

libquentier

0.8.0

Generated on Sun Jan 26 2025 16:05:44 for libquentier by Doxygen 1.15.0

Sun Jan 26 2025 16:05:44

1 libquentier	1
1.1 What's this	1
1.1.1 WARNING: libquentier is in alpha state right now, neither API nor ABI can be considered stable yet!	1
1.2 How to build/install	1
1.3 How to contribute	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	7
3.1 Class List	7
4 File Index	11
4.1 File List	11
5 Class Documentation	15
5.1 quentier::Account Class Reference	15
5.1.1 Detailed Description	17
5.1.2 Member Function Documentation	17
5.1.2.1 displayName()	17
5.1.2.2 evernoteAccountType()	17
5.1.2.3 evernoteHost()	17
5.1.2.4 id()	17
5.1.2.5 isEmpty()	17
5.1.2.6 name()	18
5.1.2.7 print()	18
5.1.2.8 setDisplayNames()	18
5.1.2.9 shardId()	18
5.1.2.10 type()	18
5.2 quentier::utility::cancelers::AnyOfCanceler Class Reference	19
5.2.1 Member Function Documentation	19
5.2.1.1 isCanceled()	19
5.3 quentier::utility::ApplicationSettings Class Reference	20
5.3.1 Detailed Description	21
5.3.2 Constructor & Destructor Documentation	21
5.3.2.1 ApplicationSettings() [1/4]	21
5.3.2.2 ApplicationSettings() [2/4]	22
5.3.2.3 ApplicationSettings() [3/4]	22
5.3.2.4 ApplicationSettings() [4/4]	22
5.3.2.5 ~ApplicationSettings()	23
5.3.3 Member Function Documentation	23
5.3.3.1 beginGroup() [1/3]	23
5.3.3.2 beginGroup() [2/3]	23

5.3.3.3 beginGroup() [3/3]	24
5.3.3.4 beginReadArray() [1/3]	24
5.3.3.5 beginReadArray() [2/3]	24
5.3.3.6 beginReadArray() [3/3]	25
5.3.3.7 beginWriteArray() [1/3]	25
5.3.3.8 beginWriteArray() [2/3]	25
5.3.3.9 beginWriteArray() [3/3]	26
5.3.3.10 contains() [1/3]	26
5.3.3.11 contains() [2/3]	26
5.3.3.12 contains() [3/3]	27
5.3.3.13 print()	27
5.3.3.14 remove() [1/3]	27
5.3.3.15 remove() [2/3]	28
5.3.3.16 remove() [3/3]	28
5.3.3.17 setValue() [1/3]	28
5.3.3.18 setValue() [2/3]	29
5.3.3.19 setValue() [3/3]	29
5.3.3.20 value() [1/3]	29
5.3.3.21 value() [2/3]	30
5.3.3.22 value() [3/3]	30
5.4 quantier::utility::ApplicationSettings::ArrayCloser Struct Reference	30
5.4.1 Detailed Description	31
5.5 quantier::synchronization::AuthenticationExpiredError Struct Reference	31
5.5.1 Detailed Description	31
5.6 quantier::synchronization::ISyncConflictResolver::ConflictResolution Struct Reference	32
5.6.1 Detailed Description	32
5.7 quantier::ErrorString Class Reference	32
5.7.1 Detailed Description	34
5.7.2 Member Function Documentation	34
5.7.2.1 print()	34
5.8 quantier::utility::EventLoopWithExitStatus Class Reference	34
5.9 quantier::utility::IKeychainService::Exception Class Reference	35
5.9.1 Detailed Description	37
5.9.2 Member Function Documentation	37
5.9.2.1 exceptionDisplayName()	37
5.10 quantier::utility::FileIOProcessorAsync Class Reference	38
5.10.1 Detailed Description	39
5.10.2 Member Function Documentation	39
5.10.2.1 onReadFileRequest	39
5.10.2.2 onWriteFileRequest	40
5.10.2.3 readFileRequestProcessed	40
5.10.2.4 setIdleTimePeriod()	40

5.10.2.5 writeFileRequestProcessed	41
5.11 quentier::utility::FileSystemWatcher Class Reference	41
5.12 quentier::utility::cancelers::FutureCanceler< T > Class Template Reference	42
5.12.1 Detailed Description	43
5.12.2 Member Function Documentation	43
5.12.2.1 isCanceled()	43
5.13 quentier::utility::ApplicationSettings::GroupCloser Struct Reference	44
5.13.1 Detailed Description	44
5.14 quentier::synchronization::IAuthenticationInfo Class Reference	45
5.14.1 Detailed Description	46
5.14.2 Member Function Documentation	46
5.14.2.1 authenticationTime()	46
5.14.2.2 authToken()	46
5.14.2.3 authTokenExpirationTime()	46
5.14.2.4 noteStoreUrl()	46
5.14.2.5 shardId()	46
5.14.2.6 userId()	47
5.14.2.7 userStoreCookies()	47
5.14.2.8 webApiUrlPrefix()	47
5.15 quentier::synchronization::IAuthenticationInfoBuilder Class Reference	47
5.16 quentier::synchronization::IAuthenticator Class Reference	48
5.17 quentier::ResourceRecognitionIndexItem::IBarcodeItem Struct Reference	48
5.18 quentier::utility::cancelers::ICanceler Class Reference	48
5.18.1 Detailed Description	49
5.19 quentier::enml::IConverter Class Reference	49
5.19.1 Detailed Description	50
5.19.2 Member Function Documentation	50
5.19.2.1 convertEnmlToHtml()	50
5.19.2.2 convertEnmlToPlainText()	51
5.19.2.3 convertEnmlToWordsList()	51
5.19.2.4 convertHtmlToDoc()	51
5.19.2.5 convertHtmlToEnml()	52
5.19.2.6 convertHtmlToXhtml()	52
5.19.2.7 convertHtmlToXml()	52
5.19.2.8 convertPlainTextToWordsList()	53
5.19.2.9 exportNotesToEnex()	53
5.19.2.10 importEnex()	53
5.19.2.11 validateAndFixupEnml()	54
5.19.2.12 validateEnml()	54
5.20 quentier::enml::IDecryptedTextCache Class Reference	54
5.21 quentier::synchronization::IDownloadNotesStatus Class Reference	55
5.21.1 Detailed Description	56

5.22	quentier::synchronization::IDownloadResourcesStatus Class Reference	57
5.22.1	Member Typedef Documentation	58
5.22.1.1	ResourceWithException	58
5.23	quentier::utility::IEncryptor Struct Reference	58
5.23.1	Detailed Description	58
5.23.2	Member Enumeration Documentation	59
5.23.2.1	Cipher	59
5.23.3	Member Function Documentation	59
5.23.3.1	decrypt()	59
5.23.3.2	encrypt()	59
5.24	quentier::enml::IENMLTagsConverter Class Reference	60
5.24.1	Detailed Description	60
5.24.2	Member Function Documentation	60
5.24.2.1	convertDecryptedText()	60
5.24.2.2	convertEncryptedText()	61
5.24.2.3	convertEnToDo()	61
5.24.2.4	convertResource()	62
5.25	quentier::synchronization::ISyncConflictResolver::ConflictResolution::IgnoreMine Struct Reference	62
5.25.1	Detailed Description	62
5.26	quentier::enml::IHtmlData Struct Reference	62
5.26.1	Detailed Description	63
5.26.2	Member Function Documentation	63
5.26.2.1	html()	63
5.26.2.2	numEnCryptNodes()	63
5.26.2.3	numEnDecryptedNodes()	64
5.26.2.4	numEnToDoNodes()	64
5.26.2.5	numHyperlinkNodes()	64
5.26.2.6	print()	64
5.27	quentier::utility::IKeychainService Class Reference	64
5.27.1	Detailed Description	65
5.27.2	Member Enumeration Documentation	65
5.27.2.1	ErrorCode	65
5.27.3	Member Function Documentation	66
5.27.3.1	deletePassword()	66
5.27.3.2	readPassword()	66
5.27.3.3	writePassword()	67
5.28	quentier::local_storage::ILocalStorage Class Reference	67
5.28.1	Member Enumeration Documentation	71
5.28.1.1	Affiliation	71
5.28.1.2	TagNotesRelation	72
5.28.2	Member Function Documentation	72
5.28.2.1	notifier()	72

5.29	quentier::local_storage::ILocalStorageNotifier Class Reference	72
5.30	quentier::INoteEditorBackend Class Reference	74
5.31	quentier::synchronization::INoteStoreFactory Class Reference	76
5.32	quentier::InvalidArgument Class Reference	77
5.32.1	Member Function Documentation	78
5.32.1.1	exceptionDisplayName()	78
5.33	quentier::ResourceRecognitionIndexItem::IObjectItem Struct Reference	78
5.34	quentier::local_storage::IPatch Class Reference	79
5.34.1	Detailed Description	79
5.34.2	Member Function Documentation	79
5.34.2.1	apply()	79
5.34.2.2	backupLocalStorage()	79
5.34.2.3	fromVersion()	80
5.34.2.4	patchLongDescription()	80
5.34.2.5	patchShortDescription()	80
5.34.2.6	removeLocalStorageBackup()	80
5.34.2.7	restoreLocalStorageFromBackup()	80
5.34.2.8	toVersion()	81
5.35	quentier::IQuentierException Class Reference	81
5.35.1	Detailed Description	82
5.35.2	Member Function Documentation	82
5.35.2.1	print()	82
5.36	quentier::synchronization::ISendStatus Class Reference	83
5.36.1	Detailed Description	84
5.36.2	Member Typedef Documentation	84
5.36.2.1	NotebookWithException	84
5.36.2.2	SavedSearchWithException	84
5.36.3	Member Function Documentation	85
5.36.3.1	failedToSendNotebooks()	85
5.36.3.2	failedToSendNotes()	85
5.36.3.3	failedToSendSavedSearches()	85
5.36.3.4	failedToSendTags()	85
5.36.3.5	needToRepeatIncrementalSync()	85
5.36.3.6	stopSynchronizationError()	86
5.36.3.7	totalAttemptedToSendNotebooks()	86
5.36.3.8	totalAttemptedToSendNotes()	86
5.36.3.9	totalAttemptedToSendSavedSearches()	86
5.36.3.10	totalAttemptedToSendTags()	86
5.36.3.11	totalSuccessfullySentNotebooks()	87
5.36.3.12	totalSuccessfullySentNotes()	87
5.36.3.13	totalSuccessfullySentSavedSearches()	87
5.36.3.14	totalSuccessfullySentTags()	87

5.37	quentier::ResourceRecognitionIndexItem::IShapeItem Struct Reference	87
5.38	quentier::enml::conversion_rules::ISkipRule Class Reference	88
5.38.1	Detailed Description	89
5.38.2	Member Enumeration Documentation	89
5.38.2.1	Target	89
5.38.3	Member Function Documentation	89
5.38.3.1	caseSensitivity()	89
5.38.3.2	includeContents()	89
5.38.3.3	matchMode()	90
5.38.3.4	print()	90
5.38.3.5	target()	90
5.38.3.6	value()	90
5.39	quentier::enml::conversion_rules::ISkipRuleBuilder Class Reference	90
5.40	quentier::synchronization::ISyncChunksDataCounters Struct Reference	91
5.40.1	Detailed Description	92
5.40.2	Member Function Documentation	92
5.40.2.1	addedLinkedNotebooks()	92
5.40.2.2	addedNotebooks()	92
5.40.2.3	addedSavedSearches()	92
5.40.2.4	addedTags()	93
5.40.2.5	expungedLinkedNotebooks()	93
5.40.2.6	expungedNotebooks()	93
5.40.2.7	expungedSavedSearches()	93
5.40.2.8	expungedTags()	93
5.40.2.9	totalExpungedLinkedNotebooks()	93
5.40.2.10	totalExpungedNotebooks()	93
5.40.2.11	totalExpungedSavedSearches()	94
5.40.2.12	totalExpungedTags()	94
5.40.2.13	totalLinkedNotebooks()	94
5.40.2.14	totalNotebooks()	94
5.40.2.15	totalSavedSearches()	94
5.40.2.16	totalTags()	94
5.40.2.17	updatedLinkedNotebooks()	94
5.40.2.18	updatedNotebooks()	95
5.40.2.19	updatedSavedSearches()	95
5.40.2.20	updatedTags()	95
5.41	quentier::synchronization::ISyncConflictResolver Class Reference	95
5.41.1	Detailed Description	96
5.41.2	Member Typedef Documentation	96
5.41.2.1	NotebookConflictResolution	96
5.41.2.2	NoteConflictResolution	96
5.41.2.3	SavedSearchConflictResolution	96

5.41.2.4 TagConflictResolution	97
5.42 quotient::synchronization::ISyncEventsNotifier Class Reference	97
5.42.1 Member Function Documentation	98
5.42.1.1 downloadFinished	98
5.42.1.2 linkedNotebookNotesDownloadProgress	98
5.42.1.3 linkedNotebookResourcesDownloadProgress	99
5.42.1.4 linkedNotebookSendStatusUpdate	99
5.42.1.5 linkedNotebookSyncChunksDataProcessingProgress	99
5.42.1.6 linkedNotebookSyncChunksDownloaded	100
5.42.1.7 linkedNotebookSyncChunksDownloadProgress	100
5.42.1.8 notesDownloadProgress	100
5.42.1.9 resourcesDownloadProgress	101
5.42.1.10 startLinkedNotebooksDataDownloading	101
5.42.1.11 syncChunksDataProcessingProgress	101
5.42.1.12 syncChunksDownloaded	101
5.42.1.13 syncChunksDownloadProgress	102
5.42.1.14 userOwnSendStatusUpdate	102
5.43 quotient::synchronization::ISynchronizer Class Reference	102
5.43.1 Member Typedef Documentation	103
5.43.1.1 SyncResult	103
5.44 quotient::synchronization::ISyncOptions Class Reference	103
5.44.1 Detailed Description	104
5.44.2 Member Function Documentation	104
5.44.2.1 downloadNoteThumbnails()	104
5.44.2.2 inkNoteImagesStorageDir()	104
5.44.2.3 maxConcurrentNoteDownloads()	104
5.44.2.4 maxConcurrentResourceDownloads()	105
5.44.2.5 requestContext()	105
5.44.2.6 retryPolicy()	105
5.45 quotient::synchronization::ISyncOptionsBuilder Class Reference	105
5.46 quotient::synchronization::ISyncResult Class Reference	106
5.47 quotient::synchronization::ISyncState Class Reference	107
5.47.1 Detailed Description	108
5.48 quotient::synchronization::ISyncStateBuilder Class Reference	108
5.49 quotient::synchronization::ISyncStateStorage Class Reference	108
5.49.1 Detailed Description	109
5.49.2 Member Function Documentation	109
5.49.2.1 notifySyncStateUpdated	109
5.50 quotient::ResourceRecognitionIndexItem::ITextItem Struct Reference	110
5.51 quotient::synchronization::IUserStoreFactory Class Reference	110
5.52 quotient::local_storage::ILocalStorage::ListGuidsFilters Struct Reference	110
5.53 quotient::local_storage::ILocalStorage::ListLinkedNotebooksOptions Struct Reference	110

5.54	quentier::local_storage::ILocalStorage::ListNotebooksOptions Struct Reference	112
5.55	quentier::local_storage::ILocalStorage::ListNotesOptions Struct Reference	113
5.56	quentier::local_storage::ILocalStorage::ListObjectsFilters Struct Reference	114
5.57	quentier::local_storage::ILocalStorage::ListOptionsBase Struct Reference	114
5.58	quentier::local_storage::ILocalStorage::ListSavedSearchesOptions Struct Reference	115
5.59	quentier::local_storage::ILocalStorage::ListTagsOptions Struct Reference	117
5.60	quentier::local_storage::LocalStorageOpenException Class Reference	118
5.60.1	Detailed Description	120
5.60.2	Member Function Documentation	120
5.60.2.1	exceptionDisplayName()	120
5.61	quentier::local_storage::LocalStorageOperationException Class Reference	120
5.61.1	Detailed Description	122
5.61.2	Member Function Documentation	122
5.61.2.1	exceptionDisplayName()	122
5.62	quentier::utility::LRUCache< Key, Value, Allocator > Class Template Reference	122
5.62.1	Member Typedef Documentation	123
5.62.1.1	const_pointer	123
5.63	quentier::utility::cancelers::ManualCanceler Class Reference	123
5.63.1	Detailed Description	124
5.63.2	Member Function Documentation	124
5.63.2.1	cancel()	124
5.63.2.2	isCanceled()	124
5.64	quentier::synchronization::tests::mocks::MockIAuthenticator Class Reference	124
5.65	quentier::utility::tests::mocks::MockIKeychainService Class Reference	125
5.66	quentier::local_storage::tests::mocks::MockILocalStorage Class Reference	127
5.67	quentier::synchronization::tests::mocks::MockINoteStoreFactory Class Reference	132
5.68	quentier::synchronization::tests::mocks::MockISyncConflictResolver Class Reference	133
5.69	quentier::synchronization::tests::mocks::MockISyncStateStorage Class Reference	135
5.70	quentier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T > Struct Template Reference	136
5.70.1	Detailed Description	136
5.70.2	Member Data Documentation	137
5.70.2.1	mine	137
5.71	quentier::NoteEditor Class Reference	137
5.71.1	Detailed Description	140
5.71.2	Member Function Documentation	141
5.71.2.1	backend()	141
5.71.2.2	clear()	141
5.71.2.3	convertToNote	141
5.71.2.4	currentNoteLocalId()	141
5.71.2.5	defaultFont()	141
5.71.2.6	defaultPalette()	141

5.71.2.7 idleTime()	142
5.71.2.8 inAppNoteLinkPasteRequested	142
5.71.2.9 initialize()	142
5.71.2.10 isEditorPageModified()	143
5.71.2.11 isModified()	143
5.71.2.12 isNoteLoaded()	143
5.71.2.13 saveNoteToLocalStorage	143
5.71.2.14 setAccount()	143
5.71.2.15 setBackend()	143
5.71.2.16 setCurrentNoteLocalId()	144
5.71.2.17 setDefaultFont	144
5.71.2.18 setDefaultPalette	144
5.71.2.19 setFocus()	145
5.71.2.20 setInitialPageHtml()	145
5.71.2.21 setNoteDeletedPageHtml()	145
5.71.2.22 setNoteLoadingPageHtml()	145
5.71.2.23 setNoteNotFoundPageHtml()	145
5.71.2.24 setNoteTitle	145
5.71.2.25 setTagIds	146
5.71.2.26 setUndoStack()	146
5.71.2.27 undoStack()	146
5.72 quantier::local_storage::NoteSearchQuery Class Reference	146
5.72.1 Member Function Documentation	149
5.72.1.1 notebookModifier()	149
5.72.1.2 print()	149
5.72.1.3 queryString()	149
5.73 quantier::OperationCanceled Class Reference	150
5.73.1 Member Function Documentation	151
5.73.1.1 exceptionDisplayName()	151
5.74 quantier::utility::Printable Class Reference	151
5.74.1 Detailed Description	153
5.75 QPromise< T > Class Template Reference	153
5.76 quantier::utility::QuantierApplication Class Reference	153
5.77 quantier::utility::QuantierUndoCommand Class Reference	154
5.77.1 Detailed Description	155
5.78 quantier::synchronization::RateLimitReachedError Struct Reference	156
5.78.1 Detailed Description	156
5.78.2 Member Data Documentation	156
5.78.2.1 rateLimitDurationSec	156
5.79 quantier::ResourceRecognitionIndexItem Class Reference	156
5.79.1 Member Function Documentation	158
5.79.1.1 print()	158

5.80	quentier::ResourceRecognitionIndices Class Reference	158
5.80.1	Member Function Documentation	159
5.80.1.1	print()	159
5.81	quentier::Result< ValueType, ErrorType, typename > Class Template Reference	160
5.81.1	Member Function Documentation	160
5.81.1.1	isValid()	160
5.82	quentier::threading::detail::ResultTypeHelper< F, Arg, Enable > Struct Template Reference	160
5.83	quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< !std::is_↵ invocable_v< std::decay_t< F >, QFuture< Arg > > > > Struct Template Reference	160
5.84	quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< std::is_↵ invocable_v< std::decay_t< F >, QFuture< Arg > > > > Struct Template Reference	161
5.85	quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< !std::is_↵ invocable_v< std::decay_t< F >, QFuture< void > > > > Struct Template Reference	161
5.86	quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< std::is_↵ invocable_v< std::decay_t< F >, QFuture< void > > > > Struct Template Reference	161
5.87	quentier::RuntimeError Class Reference	161
5.87.1	Member Function Documentation	163
5.87.1.1	exceptionDisplayName()	163
5.88	quentier::utility::ShortcutManager Class Reference	163
5.88.1	Member Function Documentation	165
5.88.1.1	defaultShortcut() [1/2]	165
5.88.1.2	defaultShortcut() [2/2]	165
5.88.1.3	shortcut() [1/2]	165
5.88.1.4	shortcut() [2/2]	165
5.88.1.5	userShortcut() [1/2]	166
5.88.1.6	userShortcut() [2/2]	166
5.89	quentier::SpellChecker Class Reference	166
5.90	quentier::utility::StringUtils Class Reference	167
5.91	quentier::utility::SysInfo Class Reference	167
5.92	quentier::threading::TrackedTask< LockableObject, Function > Class Template Reference	167
5.92.1	Detailed Description	168
5.93	quentier::utility::UidGenerator Class Reference	168
5.94	quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseMine Struct Reference	168
5.94.1	Detailed Description	168
5.95	quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseTheirs Struct Reference	168
5.95.1	Detailed Description	168
6	File Documentation	169
6.1	ISkipRule.h	169
6.2	ISkipRuleBuilder.h	170
6.3	MatchMode.h	170
6.4	HtmlUtils.h	171
6.5	IConverter.h	171

6.6 IDecryptedTextCache.h	172
6.7 IENMLTagsConverter.h	173
6.8 IHtmlData.h	174
6.9 InvalidArgument.h	175
6.10 IQuentierException.h	175
6.11 OperationCanceled.h	176
6.12 RuntimeError.h	176
6.13 enml/conversion_rules/Factory.h	177
6.14 enml/Factory.h	177
6.15 local_storage/Factory.h	178
6.16 synchronization/Factory.h	178
6.17 threading/Factory.h	179
6.18 utility/Factory.h	179
6.19 ILocalStorage.h	180
6.20 ILocalStorageNotifier.h	188
6.21 IPatch.h	189
6.22 LocalStorageOpenException.h	190
6.23 LocalStorageOperationException.h	190
6.24 NoteSearchQuery.h	191
6.25 MockILocalStorage.h	193
6.26 QuentierLogger.h	197
6.27 INoteEditorBackend.h	198
6.28 NoteEditor.h	201
6.29 SpellChecker.h	204
6.30 IAuthenticator.h	205
6.31 INoteStoreFactory.h	206
6.32 ISyncConflictResolver.h	206
6.33 ISyncEventsNotifier.h	208
6.34 ISynchronizer.h	209
6.35 ISyncStateStorage.h	210
6.36 IUserStoreFactory.h	210
6.37 MockIAuthenticator.h	211
6.38 MockINoteStoreFactory.h	211
6.39 MockISyncConflictResolver.h	212
6.40 MockISyncStateStorage.h	212
6.41 Errors.h	213
6.42 IAuthenticationInfo.h	214
6.43 IAuthenticationInfoBuilder.h	214
6.44 IDownloadNotesStatus.h	215
6.45 IDownloadResourcesStatus.h	216
6.46 ISendStatus.h	217
6.47 ISyncChunksDataCounters.h	218

6.48 ISyncOptions.h	219
6.49 ISyncOptionsBuilder.h	220
6.50 ISyncResult.h	221
6.51 ISyncState.h	222
6.52 ISyncStateBuilder.h	222
6.53 AuthenticationInfo.h	223
6.54 DownloadNotesStatus.h	223
6.55 DownloadResourcesStatus.h	224
6.56 SendStatus.h	224
6.57 SyncChunksDataCounters.h	225
6.58 SyncResult.h	225
6.59 SyncState.h	226
6.60 Future.h	226
6.61 Post.h	230
6.62 Qt5Promise.h	231
6.63 QtFutureContinuations.h	232
6.64 QtFutureHelpers.h	238
6.65 Runnable.h	241
6.66 TrackedTask.h	241
6.67 Account.h	242
6.68 ErrorString.h	244
6.69 enml/conversion_rules/Fwd.h	245
6.70 enml/Fwd.h	245
6.71 local_storage/Fwd.h	246
6.72 synchronization/Fwd.h	246
6.73 synchronization/types/Fwd.h	247
6.74 threading/Fwd.h	248
6.75 types/Fwd.h	248
6.76 utility/cancelers/Fwd.h	249
6.77 utility/Fwd.h	249
6.78 NoteUtils.h	250
6.79 RegisterMetatypes.h	250
6.80 ResourceRecognitionIndexItem.h	251
6.81 ResourceRecognitionIndices.h	252
6.82 ResourceUtils.h	253
6.83 Result.h	254
6.84 Validation.h	256
6.85 ApplicationSettings.h	257
6.86 AnyOfCanceler.h	258
6.87 FutureCanceler.h	259
6.88 ICanceler.h	259
6.89 ManualCanceler.h	260

6.90 Compat.h	260
6.91 DateTime.h	261
6.92 EventLoopWithExitStatus.h	262
6.93 FileIOProcessorAsync.h	262
6.94 FileSystem.h	263
6.95 FileSystemWatcher.h	264
6.96 IEncryptor.h	265
6.97 IKeychainService.h	265
6.98 Initialize.h	266
6.99 LRUCache.hpp	267
6.100 MessageBox.h	269
6.101 PlatformUtils.h	270
6.102 Printable.h	271
6.103 QuentierApplication.h	272
6.104 QuentierUndoCommand.h	272
6.105 ShortcutManager.h	273
6.106 Size.h	275
6.107 StandardPaths.h	275
6.108 StringUtils.h	276
6.109 SuppressWarnings.h	277
6.110 SysInfo.h	278
6.111 TagSortByParentChildRelations.h	278
6.112 MockIKeychainService.h	279
6.113 UidGenerator.h	279
6.114 Unreachable.h	280
Index	281

Chapter 1

libquentier

Set of Qt/C++ APIs for feature rich desktop clients for Evernote service

1.1 What's this

This library presents a set of Qt/C++ APIs useful for applications working as feature rich desktop clients for Evernote service. The most important and useful components of the library are the following:

- Local storage - persistence of data downloaded from Evernote service in a local SQLite database
- Synchronization - the logics of exchanging new and/or modified data with Evernote service
- Note editor - the UI component capable for notes displaying and editing

The library is based on the lower level functionality provided by [QEverCloud](#) library. It also serves as the functional core of [Quentier](#) application.

1.1.1 WARNING: libquentier is in alpha state right now, neither API nor ABI can be considered stable yet!

1.2 How to build/install

Please see the building/installation guide.

1.3 How to contribute

Please see the contribution guide for detailed info.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

quentier::utility::ApplicationSettings::ArrayCloser	30
quentier::synchronization::AuthenticationExpiredError	31
quentier::synchronization::ISyncConflictResolver::ConflictResolution	32
quentier::utility::ApplicationSettings::GroupCloser	44
quentier::synchronization::IAuthenticationInfoBuilder	47
quentier::synchronization::IAuthenticator	48
quentier::synchronization::tests::mocks::MockIAuthenticator	124
quentier::ResourceRecognitionIndexItem::IBarcodeItem	48
quentier::utility::cancelers::ICanceler	48
quentier::utility::cancelers::AnyOfCanceler	19
quentier::utility::cancelers::FutureCanceler< T >	42
quentier::utility::cancelers::ManualCanceler	123
quentier::enml::IConverter	49
quentier::enml::IDecryptedTextCache	54
quentier::utility::IEncryptor	58
quentier::enml::IENMLTagsConverter	60
quentier::synchronization::ISyncConflictResolver::ConflictResolution::IgnoreMine	62
quentier::utility::IKeychainService	64
quentier::utility::tests::mocks::MockIKeychainService	125
quentier::local_storage::ILocalStorage	67
quentier::local_storage::tests::mocks::MockILocalStorage	127
quentier::INoteEditorBackend	74
quentier::synchronization::INoteStoreFactory	76
quentier::synchronization::tests::mocks::MockINoteStoreFactory	132
quentier::ResourceRecognitionIndexItem::IObjectItem	78
quentier::local_storage::IPatch	79
quentier::ResourceRecognitionIndexItem::IShapeItem	87
quentier::enml::conversion_rules::ISkipRuleBuilder	90
quentier::synchronization::ISyncConflictResolver	95
quentier::synchronization::tests::mocks::MockISyncConflictResolver	133
quentier::synchronization::ISynchronizer	102
quentier::synchronization::ISyncOptionsBuilder	105
quentier::synchronization::ISyncStateBuilder	108

quentier::ResourceRecognitionIndexItem::ITextItem	110
quentier::synchronization::IUserStoreFactory	110
quentier::local_storage::ILocalStorage::ListGuidsFilters	110
quentier::local_storage::ILocalStorage::ListObjectsFilters	114
quentier::local_storage::ILocalStorage::ListOptionsBase	114
quentier::local_storage::ILocalStorage::ListLinkedNotebooksOptions	110
quentier::local_storage::ILocalStorage::ListNotebooksOptions	112
quentier::local_storage::ILocalStorage::ListNotesOptions	113
quentier::local_storage::ILocalStorage::ListSavedSearchesOptions	115
quentier::local_storage::ILocalStorage::ListTagsOptions	117
quentier::utility::LRUCache< Key, Value, Allocator >	122
quentier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T >	136
quentier::utility::Printable	151
quentier::Account	15
quentier::ErrorString	32
quentier::IQuentierException	81
quentier::InvalidArgument	77
quentier::OperationCanceled	150
quentier::RuntimeError	161
quentier::local_storage::LocalStorageOpenException	118
quentier::local_storage::LocalStorageOperationException	120
quentier::utility::IKeychainService::Exception	35
quentier::ResourceRecognitionIndexItem	156
quentier::ResourceRecognitionIndices	158
quentier::enml::IHtmlData	62
quentier::enml::conversion_rules::ISkipRule	88
quentier::local_storage::NoteSearchQuery	146
quentier::synchronization::IAuthenticationInfo	45
quentier::synchronization::IDownloadNotesStatus	55
quentier::synchronization::IDownloadResourcesStatus	57
quentier::synchronization::ISendStatus	83
quentier::synchronization::ISyncChunksDataCounters	91
quentier::synchronization::ISyncOptions	103
quentier::synchronization::ISyncResult	106
quentier::synchronization::ISyncState	107
quentier::utility::ApplicationSettings	20
QApplication	
quentier::utility::QuentierApplication	153
QEventLoop	
quentier::utility::EventLoopWithExitStatus	34
QException	
quentier::IQuentierException	81
QObject	
quentier::SpellChecker	166
quentier::local_storage::ILocalStorageNotifier	72
quentier::synchronization::ISyncEventsNotifier	97
quentier::synchronization::ISyncStateStorage	108
quentier::synchronization::tests::mocks::MockISyncStateStorage	135
quentier::utility::FileIOProcessorAsync	38
quentier::utility::FileSystemWatcher	41
quentier::utility::QuentierUndoCommand	154
quentier::utility::ShortcutManager	163
QPromise< T >	153
QSettings	
quentier::utility::ApplicationSettings	20
QUndoCommand	
quentier::utility::QuentierUndoCommand	154
QWidget	

quentier::NoteEditor	137
quentier::synchronization::RateLimitReachedError	156
quentier::Result< ValueType, ErrorType, typename >	160
quentier::threading::detail::ResultTypeHelper< F, Arg, Enable >	160
quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< Arg > > > >	160
quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< Arg > > > >	161
quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< void > > > >	161
quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< void > > > >	161
quentier::utility::StringUtils	167
quentier::utility::SysInfo	167
quentier::threading::TrackedTask< LockableObject, Function >	167
quentier::utility::UidGenerator	168
quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseMine	168
quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseTheirs	168

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

quentier::Account	
Encapsulates some details about the account: its name, whether it is local or synchronized to Evernote and for the latter case - some additional details like upload limit etc	15
quentier::utility::cancelers::AnyOfCanceler	19
quentier::utility::ApplicationSettings	
Enhances the functionality of QSettings, in particular it simplifies the way of working with either application-wide or account-specific settings	20
quentier::utility::ApplicationSettings::ArrayCloser	30
quentier::synchronization::AuthenticationExpiredError	31
quentier::synchronization::ISyncConflictResolver::ConflictResolution	
The ConflictResolution struct is a namespace inside which several other structs determining actual conflict resolutions	32
quentier::ErrorString	
Encapsulates two (or more) strings which are meant to contain translatable (base) and non-translatable (details) parts of the error description	32
quentier::utility::EventLoopWithExitStatus	34
quentier::utility::IKeychainService::Exception	
The IKeychainService::Exception class is the base class for exceptions returned inside QFutures from methods of IKeychainService	35
quentier::utility::FileIOProcessorAsync	
Wrapper under simple file IO operations, it is meant to be used for simple asynchronous IO	38
quentier::utility::FileSystemWatcher	41
quentier::utility::cancelers::FutureCanceler< T >	42
quentier::utility::ApplicationSettings::GroupCloser	44
quentier::synchronization::IAuthenticationInfo	
The IAuthenticationInfo interface represents the information obtained through OAuth and necessary to access Evernote API	45
quentier::synchronization::IAuthenticationInfoBuilder	47
quentier::synchronization::IAuthenticator	48
quentier::ResourceRecognitionIndexItem::IBarcodeItem	48
quentier::utility::cancelers::ICanceler	
The ICanceler interface provides <code>isCanceled</code> method which can be used to check whether some processing can be skipped because it was canceled	48
quentier::enml::IConverter	
The IConverter interface encapsulates a set of methods performing conversions between ENML and other note content formats, namely HTML	49

quentier::enml::IDecryptedTextCache	54
quentier::synchronization::IDownloadNotesStatus	
The IDownloadNotesStatus interface presents information about the status of notes downloading process	55
quentier::synchronization::IDownloadResourcesStatus	57
quentier::utility::IEncryptor	
The IEncryptor interface provides encryption and decryption functionality which is compatible with that used by Evernote service	58
quentier::enml::IENMLTagsConverter	
The IENMLTagsConverter interfaces provides methods which convert Evernote-specific markup tags such as en-crypt, en-media etc. into their counterparts which should be used in the HTML representation of note content	60
quentier::synchronization::ISyncConflictResolver::ConflictResolution::IgnoreMine	
The IgnoreMine conflict resolution means "use theirs version and ignore mine version as it doesn't really conflict with theirs version"	62
quentier::enml::IHtmlData	
The IHtmlData represents the result of ENML to HTML conversion: HTML itself plus some meta-data	62
quentier::utility::IKeychainService	
The IKeychainService interface provides the ability to interact with the storage of sensitive data - read, write and delete it	64
quentier::local_storage::ILocalStorage	67
quentier::local_storage::ILocalStorageNotifier	72
quentier::INoteEditorBackend	74
quentier::synchronization::INoteStoreFactory	76
quentier::InvalidArgument	77
quentier::ResourceRecognitionIndexItem::IObjectItem	78
quentier::local_storage::IPatch	
The IPatch interface represents patches of the local storage. Each such patch somehow changes the layout of local storage persistence so that only compliant & corresponding versions of libquentier can be used to work with it	79
quentier::IQuentierException	
Interface for exceptions specific to libquentier and applications based on it	81
quentier::synchronization::ISendStatus	
The ISendStatus interface represents the information about the attempt to send information either from user's own account or from some linked notebook to Evernote	83
quentier::ResourceRecognitionIndexItem::IShapeItem	87
quentier::enml::conversion_rules::ISkipRule	
The ISkipRule interface describes a conversion rule with regards to which some ENML/HTML element/attribute should be skipped during the conversion	88
quentier::enml::conversion_rules::ISkipRuleBuilder	90
quentier::synchronization::ISyncChunksDataCounters	
The ISyncChunksDataCounters interface provides integer counters representing the current progress on processing the data from downloaded sync chunks	91
quentier::synchronization::ISyncConflictResolver	
The ISyncConflictResolver interface provides methods used to resolve conflicts between local and remote versions of the same data item	95
quentier::synchronization::ISyncEventsNotifier	97
quentier::synchronization::ISynchronizer	102
quentier::synchronization::ISyncOptions	
Options for synchronization process	103
quentier::synchronization::ISyncOptionsBuilder	105
quentier::synchronization::ISyncResult	106
quentier::synchronization::ISyncState	
The ISyncState interface provides accessory methods to determine the sync state for the account	107
quentier::synchronization::ISyncStateBuilder	108

quentier::synchronization::ISyncStateStorage	
The ISyncStateStorage interface represents the interface of a class which stores sync state for given accounts persistently and provides access to previously stores sync states	108
quentier::ResourceRecognitionIndexItem::ITextItem	110
quentier::synchronization::IUserStoreFactory	110
quentier::local_storage::ILocalStorage::ListGuidsFilters	110
quentier::local_storage::ILocalStorage::ListLinkedNotebooksOptions	110
quentier::local_storage::ILocalStorage::ListNotebooksOptions	112
quentier::local_storage::ILocalStorage::ListNotesOptions	113
quentier::local_storage::ILocalStorage::ListObjectsFilters	114
quentier::local_storage::ILocalStorage::ListOptionsBase	114
quentier::local_storage::ILocalStorage::ListSavedSearchesOptions	115
quentier::local_storage::ILocalStorage::ListTagsOptions	117
quentier::local_storage::LocalStorageOpenException	
The LocalStorageOpenException is thrown on failure to open the local storage database	118
quentier::local_storage::LocalStorageOperationException	
The LocalStorageOperationException is thrown when the local storage encounters some internal error during the attempt to process some operation	120
quentier::utility::LRUCache< Key, Value, Allocator >	122
quentier::utility::cancelers::ManualCanceler	123
quentier::synchronization::tests::mocks::MockIAuthenticator	124
quentier::utility::tests::mocks::MockIKeychainService	125
quentier::local_storage::tests::mocks::MockILocalStorage	127
quentier::synchronization::tests::mocks::MockINoteStoreFactory	132
quentier::synchronization::tests::mocks::MockISyncConflictResolver	133
quentier::synchronization::tests::mocks::MockISyncStateStorage	135
quentier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T >	
The MoveMine conflict resolution means "before using theirs version change mine version as specified". Note: the data item inside this conflict resolution might refer to something different than mine version passed to the conflict resolution function. It can be that way because the actual conflict might be with another local data item instead of the passed one	136
quentier::NoteEditor	
Widget encapsulating all the functionality necessary for showing and editing notes	137
quentier::local_storage::NoteSearchQuery	146
quentier::OperationCanceled	150
quentier::utility::Printable	
Interface for Quentier's internal classes which should be able to write themselves into QText←Stream and/or convert to QString	151
QPromise< T >	153
quentier::utility::QuentierApplication	153
quentier::utility::QuentierUndoCommand	
Has the sole purpose of working around one quirky aspect of Qt's undo/redo framework: when you push QUndoCommand to QUndoStack, it calls "redo" method of that command. This class offers subclasses to implement their own methods for actual "undo" and "redo" commands while ignoring the attempts to "redo" anything if there were no previous "undo" call prior to that	154
quentier::synchronization::RateLimitReachedError	156
quentier::ResourceRecognitionIndexItem	156
quentier::ResourceRecognitionIndices	158
quentier::Result< ValueType, ErrorType, typename >	160
quentier::threading::detail::ResultTypeHelper< F, Arg, Enable >	160
quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFu	160
160	
quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFu	161
161	
quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFu	161
161	
quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFu	161
161	

quentier::RuntimeError	161
quentier::utility::ShortcutManager	163
quentier::SpellChecker	166
quentier::utility::StringUtils	167
quentier::utility::SysInfo	167
quentier::threading::TrackedTask< LockableObject, Function >	167
quentier::utility::UidGenerator	168
quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseMine	
The UseMine conflict resolution means "override theirs version with mine version"	168
quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseTheirs	
The UseTheirs conflict resolution means "override mine version with theirs version"	168

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

ISkipRule.h	169
ISkipRuleBuilder.h	170
MatchMode.h	170
HtmlUtils.h	171
IConverter.h	171
IDecryptedTextCache.h	172
IENMLTagsConverter.h	173
IHtmlData.h	174
InvalidArgument.h	175
IQuentierException.h	175
OperationCanceled.h	176
RuntimeError.h	176
enml/conversion_rules/Factory.h	177
enml/Factory.h	177
local_storage/Factory.h	178
synchronization/Factory.h	178
threading/Factory.h	179
utility/Factory.h	179
ILocalStorage.h	180
ILocalStorageNotifier.h	188
IPatch.h	189
LocalStorageOpenException.h	190
LocalStorageOperationException.h	190
NoteSearchQuery.h	191
MockILocalStorage.h	193
QuentierLogger.h	197
INoteEditorBackend.h	198
NoteEditor.h	201
SpellChecker.h	204
IAuthenticator.h	205
INoteStoreFactory.h	206
ISyncConflictResolver.h	206
ISyncEventsNotifier.h	208
ISynchronizer.h	209
ISyncStateStorage.h	210

IUserStoreFactory.h	210
MockIAuthenticator.h	211
MockINoteStoreFactory.h	211
MockISyncConflictResolver.h	212
MockISyncStateStorage.h	212
Errors.h	213
IAuthenticationInfo.h	214
IAuthenticationInfoBuilder.h	214
IDownloadNotesStatus.h	215
IDownloadResourcesStatus.h	216
ISendStatus.h	217
ISyncChunksDataCounters.h	218
ISyncOptions.h	219
ISyncOptionsBuilder.h	220
ISyncResult.h	221
ISyncState.h	222
ISyncStateBuilder.h	222
AuthenticationInfo.h	223
DownloadNotesStatus.h	223
DownloadResourcesStatus.h	224
SendStatus.h	224
SyncChunksDataCounters.h	225
SyncResult.h	225
SyncState.h	226
Future.h	226
Post.h	230
Qt5Promise.h	231
QtFutureContinuations.h	232
QtFutureHelpers.h	238
Runnable.h	241
TrackedTask.h	241
Account.h	242
ErrorString.h	244
enml/conversion_rules/Fwd.h	245
enml/Fwd.h	245
local_storage/Fwd.h	246
synchronization/Fwd.h	246
synchronization/types/Fwd.h	247
threading/Fwd.h	248
types/Fwd.h	248
utility/cancelers/Fwd.h	249
utility/Fwd.h	249
NoteUtils.h	250
RegisterMetatypes.h	250
ResourceRecognitionIndexItem.h	251
ResourceRecognitionIndices.h	252
ResourceUtils.h	253
Result.h	254
Validation.h	256
ApplicationSettings.h	257
AnyOfCanceler.h	258
FutureCanceler.h	259
ICanceler.h	259
ManualCanceler.h	260
Compat.h	260
DateTime.h	261
EventLoopWithExitStatus.h	262
FileIOProcessorAsync.h	262

FileSystem.h	263
FileSystemWatcher.h	264
IEncryptor.h	265
IKeychainService.h	265
Initialize.h	266
LRUCache.hpp	267
MessageBox.h	269
PlatformUtils.h	270
Printable.h	271
QuentierApplication.h	272
QuentierUndoCommand.h	272
ShortcutManager.h	273
Size.h	275
StandardPaths.h	275
StringUtils.h	276
SuppressWarnings.h	277
SysInfo.h	278
TagSortByParentChildRelations.h	278
MockIKeychainService.h	279
UidGenerator.h	279
Unreachable.h	280

Chapter 5

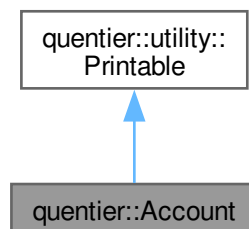
Class Documentation

5.1 `quentier::Account` Class Reference

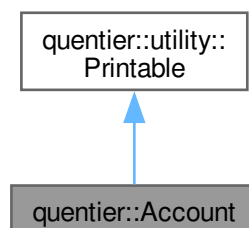
The `Account` class encapsulates some details about the account: its name, whether it is local or synchronized to Evernote and for the latter case - some additional details like upload limit etc.

```
#include <Account.h>
```

Inheritance diagram for `quentier::Account`:



Collaboration diagram for `quentier::Account`:



Public Types

- enum class **Type** { **Local** , **Evernote** }
- enum class **EvernoteAccountType** { **Free** , **Plus** , **Premium** , **Business** }

Public Member Functions

- **Account** (QString [name](#), Type [type](#), qevercloud::UserID [userId](#)=-1, EvernoteAccountType [evernoteAccountType](#)=EvernoteAccountType::Free, QString [evernoteHost](#)={}, QString [shardId](#)={})
- **Account** (const Account &other)
- **Account** (Account &&other) noexcept
- Account & **operator=** (const Account &other)
- Account & **operator=** (Account &&other) noexcept
- bool **operator==** (const Account &other) const noexcept
- bool **operator!=** (const Account &other) const noexcept
- bool [isEmpty](#) () const
- QString [name](#) () const
- void **setName** (QString [name](#))
setName sets the username to the account
- QString [displayName](#) () const
- void **setDisplayName** (QString [displayName](#))
- Type [type](#) () const
- qevercloud::UserID [id](#) () const
- EvernoteAccountType [evernoteAccountType](#) () const
- QString [evernoteHost](#) () const
- QString [shardId](#) () const
- void **setEvernoteAccountType** (EvernoteAccountType [evernoteAccountType](#))
- void **setEvernoteHost** (QString [evernoteHost](#))
- void **setShardId** (QString [shardId](#))
- qint32 [mailLimitDaily](#) () const
- qint64 [noteSizeMax](#) () const
- qint64 [resourceSizeMax](#) () const
- qint32 [linkedNotebookMax](#) () const
- qint32 [noteCountMax](#) () const
- qint32 [notebookCountMax](#) () const
- qint32 [tagCountMax](#) () const
- qint32 [noteTagCountMax](#) () const
- qint32 [savedSearchCountMax](#) () const
- qint32 [noteResourceCountMax](#) () const
- void **setEvernoteAccountLimits** (const qevercloud::AccountLimits &limits)
- QTextStream & [print](#) (QTextStream &strm) const override

Public Member Functions inherited from [quentier::utility::Printable](#)

- QString [toString](#) () const

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, Type [type](#))
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, Type [type](#))
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, EvernoteAccountType [type](#))
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, EvernoteAccountType [type](#))

5.1.1 Detailed Description

The `Account` class encapsulates some details about the account: its name, whether it is local or synchronized to Evernote and for the latter case - some additional details like upload limit etc.

5.1.2 Member Function Documentation

5.1.2.1 `displayName()`

```
QString quentier::Account::displayName () const [nodiscard]
```

Returns

Displayable user's name which is not used to uniquely identify the account, so this name may repeat across different local and Evernote accounts

5.1.2.2 `evernoteAccountType()`

```
EvernoteAccountType quentier::Account::evernoteAccountType () const [nodiscard]
```

Returns

The type of the Evernote account; if applied to free account, returns "Free"

5.1.2.3 `evernoteHost()`

```
QString quentier::Account::evernoteHost () const [nodiscard]
```

Returns

The Evernote server host with which the account is associated

5.1.2.4 `id()`

```
qevercloud::UserID quentier::Account::id () const [nodiscard]
```

Returns

User id for Evernote accounts, -1 for local accounts (as the concept of user id is not defined for local accounts)

5.1.2.5 `isEmpty()`

```
bool quentier::Account::isEmpty () const [nodiscard]
```

Returns

True if either the account is local but the name is empty or if the account is Evernote but user id is negative; in all other cases return false

5.1.2.6 name()

```
QString quentier::Account::name () const [nodiscard]
```

Returns

Username for either local or Evernote account

5.1.2.7 print()

```
QTextStream & quentier::Account::print (  
    QTextStream & strm) const [override], [virtual]
```

Implements [quentier::utility::Printable](#).

5.1.2.8 setDisplayName()

```
void quentier::Account::setDisplayName (  
    QString displayName)
```

Set the printable name of the account

5.1.2.9 shardId()

```
QString quentier::Account::shardId () const [nodiscard]
```

Returns

Shard id for Evernote accounts, empty string for local accounts (as the concept of shard id is not defined for local accounts)

5.1.2.10 type()

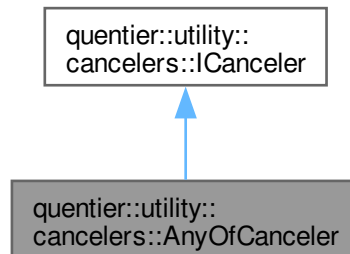
```
Type quentier::Account::type () const [nodiscard]
```

Returns

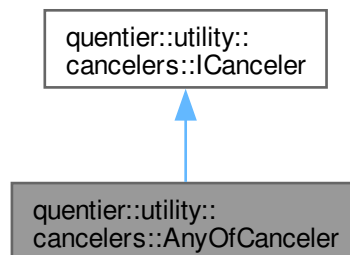
The type of the account: either local or Evernote

5.2 `quentier::utility::cancelers::AnyOfCanceler` Class Reference

Inheritance diagram for `quentier::utility::cancelers::AnyOfCanceler`:



Collaboration diagram for `quentier::utility::cancelers::AnyOfCanceler`:



Public Member Functions

- **`AnyOfCanceler`** (`QList< ICancelerPtr > cancelers`)
- **`AnyOfCanceler`** (`AnyOfCanceler &&other`) `noexcept`
- `AnyOfCanceler & operator=` (`AnyOfCanceler &&other`) `noexcept`
- `bool isCanceled ()` `const` `noexcept` override

5.2.1 Member Function Documentation

5.2.1.1 `isCanceled()`

```
bool quentier::utility::cancelers::AnyOfCanceler::isCanceled () const [nodiscard], [override],
[virtual], [noexcept]
```

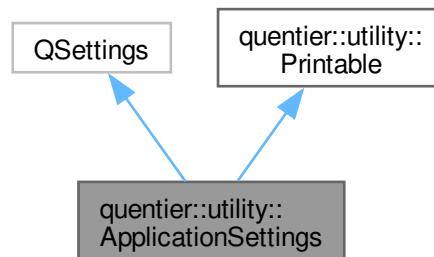
Implements [quentier::utility::cancelers::ICanceler](#).

5.3 quantier::utility::ApplicationSettings Class Reference

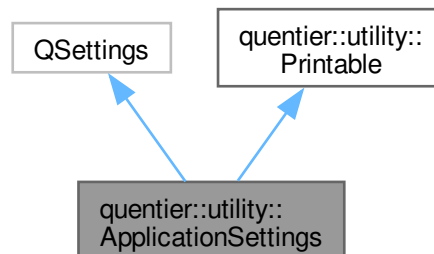
The [ApplicationSettings](#) class enhances the functionality of `QSettings`, in particular it simplifies the way of working with either application-wide or account-specific settings.

```
#include <ApplicationSettings.h>
```

Inheritance diagram for `quantier::utility::ApplicationSettings`:



Collaboration diagram for `quantier::utility::ApplicationSettings`:



Classes

- struct [ArrayCloser](#)
- struct [GroupCloser](#)

Public Member Functions

- [ApplicationSettings](#) (const QString &settingsName={})
- [ApplicationSettings](#) (const [Account](#) &account, const QString &settingsName={})
- [ApplicationSettings](#) (const [Account](#) &account, const char *settingsName, int settingsNameSize=-1)
- [ApplicationSettings](#) (const [Account](#) &account, std::string_view settingsName)
- [~ApplicationSettings](#) () override
- void [beginGroup](#) (const QString &prefix)
- void [beginGroup](#) (const char *prefix, int size=-1)
- void [beginGroup](#) (std::string_view prefix)
- int [beginReadArray](#) (const QString &prefix)
- int [beginReadArray](#) (const char *prefix, int size=-1)
- int [beginReadArray](#) (std::string_view prefix)
- void [beginWriteArray](#) (const QString &prefix, int arraySize=-1)
- void [beginWriteArray](#) (const char *prefix, int arraySize=-1, int prefixSize=-1)
- void [beginWriteArray](#) (std::string_view prefix, int arraySize=-1)
- bool [contains](#) (const QString &key) const
- bool [contains](#) (const char *key, int size=-1) const
- bool [contains](#) (std::string_view key) const
- void [remove](#) (const QString &key)
- void [remove](#) (const char *key, int size=-1)
- void [remove](#) (std::string_view key)
- void [setValue](#) (const QString &key, const QVariant &value)
- void [setValue](#) (const char *key, const QVariant &value, int keySize=-1)
- void [setValue](#) (std::string_view key, const QVariant &value)
- QVariant [value](#) (const QString &key, const QVariant &defaultValue={}) const
- QVariant [value](#) (const char *key, const QVariant &defaultValue={}, int keySize=-1) const
- QVariant [value](#) (std::string_view key, const QVariant &defaultValue={}) const
- QTextStream & [print](#) (QTextStream &strm) const override

Public Member Functions inherited from [quantier::utility::Printable](#)

- QString [toString](#) () const

5.3.1 Detailed Description

The [ApplicationSettings](#) class enhances the functionality of QSettings, in particular it simplifies the way of working with either application-wide or account-specific settings.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 ApplicationSettings() [1/4]

```
quantier::utility::ApplicationSettings::ApplicationSettings (
    const QString & settingsName = {}) [explicit]
```

Constructor for application settings not being account-specific

Parameters

<i>settingsName</i>	If not empty, the created application settings would manage the settings stored in a file with a specific name within the common settings storage; otherwise they would be stored in the default settings file for the account
---------------------	--

5.3.2.2 ApplicationSettings() [2/4]

```
quentier::utility::ApplicationSettings::ApplicationSettings (
    const Account & account,
    const QString & settingsName = {}) [explicit]
```

Constructor for application settings specific to the account

Parameters

<i>account</i>	The account for which the settings are to be stored or read
<i>settingsName</i>	If not empty, the created application settings would manage the settings stored in a file with a specific name within the account's settings storage; otherwise they would be stored in the default settings file for the account

5.3.2.3 ApplicationSettings() [3/4]

```
quentier::utility::ApplicationSettings::ApplicationSettings (
    const Account & account,
    const char * settingsName,
    int settingsNameSize = -1)
```

Constructor for application settings specific to the account

Parameters

<i>account</i>	The account for which the settings are to be stored or read
<i>settingsName</i>	If not nullptr, the created application settings would manage the settings stored in a file with a specific name within the account's settings storage; otherwise they would be stored in the default settings file for the account. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>settingsNameSize</i>	Size of the settingsName string. If negative (the default), the settingsName size is taken to be strlen(settingsName)

5.3.2.4 ApplicationSettings() [4/4]

```
quentier::utility::ApplicationSettings::ApplicationSettings (
    const Account & account,
    std::string_view settingsName)
```

Constructor for application settings specific to the account

Parameters

<i>account</i>	The account for which the settings are to be stored or read
<i>settingsName</i>	If not empty, the created application settings would manage the settings stored in a file with a specific name within the account's settings storage; otherwise they would be stored in the default settings file for the account. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8

5.3.2.5 ~ApplicationSettings()

```
quentier::utility::ApplicationSettings::~~ApplicationSettings () [override]
```

Destructor

5.3.3 Member Function Documentation**5.3.3.1 beginGroup() [1/3]**

```
void quentier::utility::ApplicationSettings::beginGroup (
    const char * prefix,
    int size = -1)
```

Appends prefix to the current group. Overload of beginGroup accepting const char * and optionally the size of the string

Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the prefix string. If negative (the default), the prefix size is taken to be strlen(prefix).

5.3.3.2 beginGroup() [2/3]

```
void quentier::utility::ApplicationSettings::beginGroup (
    const QString & prefix)
```

Appends prefix to the current group. The call is redirected to QSettings::beginGroup. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>prefix</i>	String containing the prefix name
---------------	-----------------------------------

5.3.3.3 beginGroup() [3/3]

```
void quantier::utility::ApplicationSettings::beginGroup (
    std::string_view prefix)
```

Appends prefix to the current group. Overload of beginGroup accepting std::string_view

Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the prefix string. If negative (the default), the prefix size is taken to be strlen(prefix).

5.3.3.4 beginReadArray() [1/3]

```
int quantier::utility::ApplicationSettings::beginReadArray (
    const char * prefix,
    int size = -1) [nodiscard]
```

Adds prefix to the current group and starts reading from an array. Overload of beginReadArray accepting const char * and optionally the size of the string

Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the prefix string. If negative (the default), the prefix size is taken to be strlen(prefix)

5.3.3.5 beginReadArray() [2/3]

```
int quantier::utility::ApplicationSettings::beginReadArray (
    const QString & prefix) [nodiscard]
```

Adds prefix to the current group and starts reading from an array. The call is redirected to QSettings::beginReadArray. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>prefix</i>	String containing the prefix name
---------------	-----------------------------------

Returns

The size of the array

5.3.3.6 beginReadArray() [3/3]

```
int quotientier::utility::ApplicationSettings::beginReadArray (
    std::string_view prefix) [nodiscard]
```

Adds prefix to the current group and starts reading from an array. Overload of beginReadArray accepting std::string_view

Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
---------------	---

5.3.3.7 beginWriteArray() [1/3]

```
void quotientier::utility::ApplicationSettings::beginWriteArray (
    const char * prefix,
    int arraySize = -1,
    int prefixSize = -1)
```

Adds prefix to the current group and starts writing an array of size arraySize. Overload of beginWriteArray accepting const char * and optionally the size of the string

Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>arraySize</i>	Size of the array to be written. If negative (the default), it is automatically determined based on the indexes of the entries written.
<i>prefixSize</i>	Size of the prefix string. If negative (the default), the prefix size is taken to be strlen(prefix)

5.3.3.8 beginWriteArray() [2/3]

```
void quotientier::utility::ApplicationSettings::beginWriteArray (
    const QString & prefix,
    int arraySize = -1)
```

Adds prefix to the current group and starts writing an array of size arraySize. The call is redirected to QSettings::beginWriteArray. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>prefix</i>	String containing the prefix name
<i>arraySize</i>	Size of the array to be written. If negative (the default), it is automatically determined based on the indexes of the entries written.

5.3.3.9 beginWriteArray() [3/3]

```
void quantier::utility::ApplicationSettings::beginWriteArray (
    std::string_view prefix,
    int arraySize = -1)
```

Adds prefix to the current group and starts writing an array of size arraySize. Overload of beginWriteArray accepting std::string_view

Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>arraySize</i>	Size of the array to be written. If negative (the default), it is automatically determined based on the indexes of the entries written.

5.3.3.10 contains() [1/3]

```
bool quantier::utility::ApplicationSettings::contains (
    const char * key,
    int size = -1) const [nodiscard]
```

Overload of contains accepting const char * and optionally the size of the string

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the key string. If negative (the default), the key size is taken to be strlen(key)

Returns

True if there exists a setting called key; false otherwise

5.3.3.11 contains() [2/3]

```
bool quantier::utility::ApplicationSettings::contains (
    const QString & key) const [nodiscard]
```

The call is redirected to QSettings::contains. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>key</i>	The key being checked for presence
------------	------------------------------------

Returns

True if there exists a setting called key; false otherwise

5.3.3.12 contains() [3/3]

```
bool quentier::utility::ApplicationSettings::contains (
    std::string_view key) const [nodiscard]
```

Overload of contains accepting std::string_view

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
------------	--

Returns

True if there exists a setting called key; false otherwise

5.3.3.13 print()

```
QTextStream & quentier::utility::ApplicationSettings::print (
    QTextStream & strm) const [override], [virtual]
```

Implements [quentier::utility::Printable](#).

5.3.3.14 remove() [1/3]

```
void quentier::utility::ApplicationSettings::remove (
    const char * key,
    int size = -1)
```

Removes the setting key and any sub-settings of key. Overload of remove accepting const char * and optionally the size of the string

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the key string. If negative (the default), the key size is taken to be strlen(key).

5.3.3.15 remove() [2/3]

```
void quentier::utility::ApplicationSettings::remove (
    const QString & key)
```

Removes the setting key and any sub-settings of key. The call is redirected to QSettings::remove. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>key</i>	String containing the setting name
------------	------------------------------------

5.3.3.16 remove() [3/3]

```
void quentier::utility::ApplicationSettings::remove (
    std::string_view key)
```

Removes the setting key and any sub-settings of key. Overload of remove accepting std::string_view

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
------------	--

5.3.3.17 setValue() [1/3]

```
void quentier::utility::ApplicationSettings::setValue (
    const char * key,
    const QVariant & value,
    int keySize = -1)
```

Sets the value of setting. Overload of setValue accepting const char * and optionally the size of the string

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>value</i>	Value for setting key
<i>keySize</i>	Size of the key string. If negative (the default), the key size is taken to be strlen(key).

5.3.3.18 setValue() [2/3]

```
void quantier::utility::ApplicationSettings::setValue (
    const QString & key,
    const QVariant & value)
```

Sets the value of setting. The call is redirected to QSettings::setValue. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>key</i>	String containing the setting name
<i>value</i>	Value for setting key

5.3.3.19 setValue() [3/3]

```
void quantier::utility::ApplicationSettings::setValue (
    std::string_view key,
    const QVariant & value)
```

Sets the value of setting. Overload of setValue accepting std::string_view

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>value</i>	Value for setting key

5.3.3.20 value() [1/3]

```
QVariant quantier::utility::ApplicationSettings::value (
    const char * key,
    const QVariant & defaultValue = {},
    int keySize = -1) const [nodiscard]
```

Fetches the value of setting. Overload of value accepting const char * and optionally the size of the string

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>defaultValue</i>	Default value returned if the setting doesn't exist
<i>keySize</i>	Size of the key string. If negative (the default), the key size is taken to be strlen(key)

Returns

The value for setting key. If the setting doesn't exist, returns defaultValue. If no default value is specified, a default QVariant is returned.

5.3.3.21 value() [2/3]

```
QVariant quentier::utility::ApplicationSettings::value (  
    const QString & key,  
    const QVariant & defaultValue = {}) const [nodiscard]
```

Fetches the value of setting. The call is redirected to QSettings::value. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

<i>key</i>	String containing the setting name
<i>defaultValue</i>	Default value returned if the setting doesn't exist

Returns

The value for setting key. If the setting doesn't exist, returns defaultValue. If no default value is specified, a default QVariant is returned.

5.3.3.22 value() [3/3]

```
QVariant quentier::utility::ApplicationSettings::value (  
    std::string_view key,  
    const QVariant & defaultValue = {}) const [nodiscard]
```

Fetches the value of setting. Overload of value accepting std::string_view

Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>defaultValue</i>	Default value returned if the setting doesn't exist

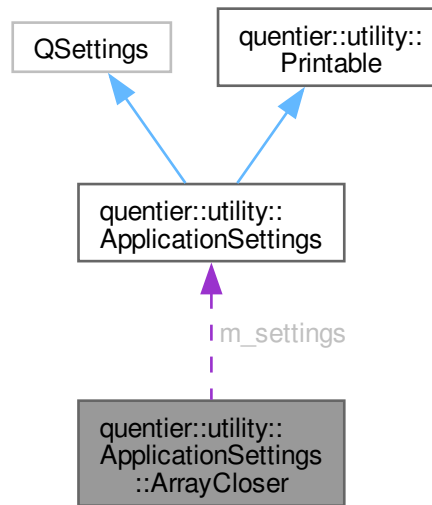
Returns

The value for setting key. If the setting doesn't exist, returns defaultValue. If no default value is specified, a default QVariant is returned.

5.4 quentier::utility::ApplicationSettings::ArrayCloser Struct Reference

```
#include <ApplicationSettings.h>
```

Collaboration diagram for `quentier::utility::ApplicationSettings::ArrayCloser`:



Public Member Functions

- **ArrayCloser** ([ApplicationSettings](#) &settings)

Public Attributes

- [ApplicationSettings](#) & **m_settings**

5.4.1 Detailed Description

Helper struct for RAIL style of ensuring the array once began would be closed even if exception is thrown after beginning the array

5.5 `quentier::synchronization::AuthenticationExpiredError` Struct Reference

```
#include <Errors.h>
```

5.5.1 Detailed Description

Authentication expired error indicates that used authentication token has expired so authentication should be repeated before the next attempt to run synchronization.

5.6 `quentier::synchronization::ISyncConflictResolver::Conflict` Resolution Struct Reference

The `ConflictResolution` struct is a namespace inside which several other structs determining actual conflict resolutions.

```
#include <ISyncConflictResolver.h>
```

Classes

- struct `UseTheirs`
The `UseTheirs` conflict resolution means "override mine version with theirs version".
- struct `UseMine`
The `UseMine` conflict resolution means "override theirs version with mine version".
- struct `IgnoreMine`
The `IgnoreMine` conflict resolution means "use theirs version and ignore mine version as it doesn't really conflict with theirs version".
- struct `MoveMine`
The `MoveMine` conflict resolution means "before using theirs version change mine version as specified". Note: the data item inside this conflict resolution might refer to something different than mine version passed to the conflict resolution function. It can be that way because the actual conflict might be with another local data item instead of the passed one.

5.6.1 Detailed Description

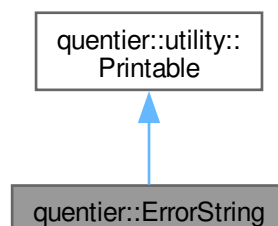
The `ConflictResolution` struct is a namespace inside which several other structs determining actual conflict resolutions.

5.7 `quentier::ErrorString` Class Reference

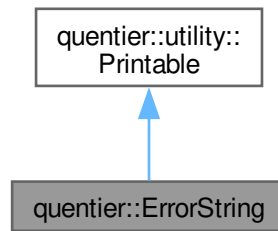
The `ErrorString` class encapsulates two (or more) strings which are meant to contain translatable (base) and non-translatable (details) parts of the error description.

```
#include <ErrorString.h>
```

Inheritance diagram for `quentier::ErrorString`:



Collaboration diagram for `quentier::ErrorString`:



Public Member Functions

- **ErrorString** (const char *error=nullptr)
- **ErrorString** (const QString &error)
- **ErrorString** (const ErrorString &other)
- **ErrorString** (ErrorString &&other) noexcept
- ErrorString & **operator=** (const ErrorString &other)
- ErrorString & **operator=** (ErrorString &&other) noexcept
- const QString & **base** () const noexcept
- QString & **base** ()
- const QStringList & **additionalBases** () const noexcept
- QStringList & **additionalBases** ()
- const QString & **details** () const noexcept
- QString & **details** ()
- void **setBase** (QString error)
- void **setBase** (const char *error)
- void **appendBase** (const QString &error)
- void **appendBase** (const QStringList &errors)
- void **appendBase** (const char *error)
- void **setDetails** (const QString &error)
- void **setDetails** (const char *error)
- bool **isEmpty** () const
- void **clear** ()
- QString **localizedString** () const
- QString **nonLocalizedString** () const
- QTextStream & **print** (QTextStream &strm) const override

Public Member Functions inherited from `quentier::utility::Printable`

- QString **toString** () const

5.7.1 Detailed Description

The [ErrorString](#) class encapsulates two (or more) strings which are meant to contain translatable (base) and non-translatable (details) parts of the error description.

1. `base()` methods return const and non-const links to the primary translatable string
2. `details()` methods return const and non-const links to non-translatable string (coming from some third party library etc)
3. `additionalBases()` methods return const and non-const links to additional translatable strings; one translatable string is not always enough because the error message might be composed from different parts

5.7.2 Member Function Documentation

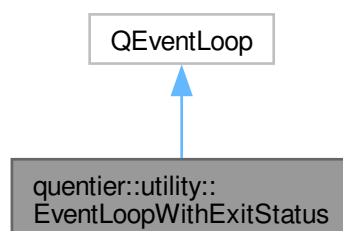
5.7.2.1 `print()`

```
QTextStream & quentier::ErrorString::print (  
    QTextStream & strm) const [override], [virtual]
```

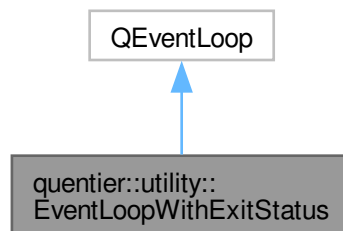
Implements [quentier::utility::Printable](#).

5.8 quentier::utility::EventLoopWithExitStatus Class Reference

Inheritance diagram for quentier::utility::EventLoopWithExitStatus:



Collaboration diagram for `quentier::utility::EventLoopWithExitStatus`:



Public Types

- enum class **ExitStatus** { **Success** , **Failure** , **Timeout** }

Public Slots

- void **exitAsSuccess** ()
- void **exitAsFailure** ()
- void **exitAsFailureWithError** (QString errorDescription)
- void **exitAsFailureWithErrorString** ([ErrorString](#) errorDescription)
- void **exitAsTimeout** ()

Public Member Functions

- **EventLoopWithExitStatus** (QObject *parent=nullptr)
- ExitStatus **exitStatus** () const
- const [ErrorString](#) & **errorDescription** () const

Friends

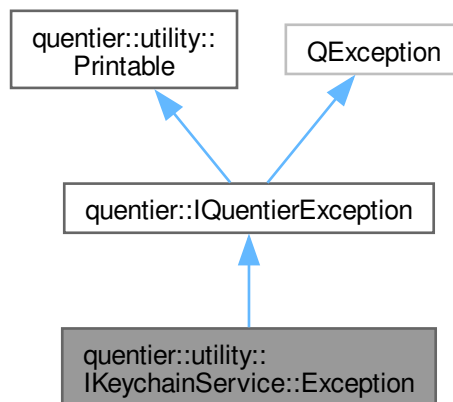
- QDebug & **operator<<** (QDebug &dbg, ExitStatus status)
- QTextStream & **operator<<** (QTextStream &strm, ExitStatus status)

5.9 `quentier::utility::IKeychainService::Exception` Class Reference

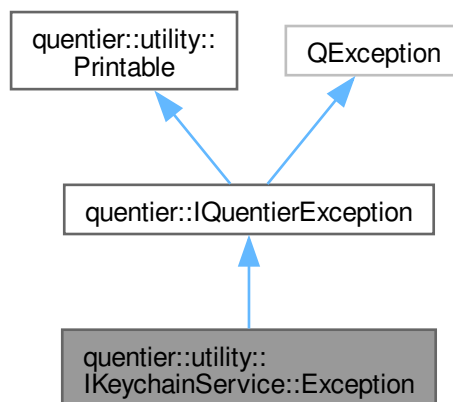
The [IKeychainService::Exception](#) class is the base class for exceptions returned inside QFutures from methods of [IKeychainService](#).

```
#include <IKeychainService.h>
```

Inheritance diagram for `quentier::utility::IKeychainService::Exception`:



Collaboration diagram for `quentier::utility::IKeychainService::Exception`:



Public Member Functions

- **Exception** ([ErrorCode](#) errorCode) noexcept
- **Exception** ([ErrorCode](#) errorCode, [ErrorString](#) errorDescription) noexcept
- [ErrorCode](#) **errorCode** () const noexcept
- [QString](#) **exceptionDisplayName** () const override
- void **raise** () const override
- [Exception](#) * **clone** () const override

Public Member Functions inherited from `quentier::IQuentierException`

- `ErrorString errorMessage () const`
- `QString localizedErrorMessage () const`
- `QString nonLocalizedErrorMessage () const`
- `const char * what () const` noexcept override
- `QTextStream & print (QTextStream &strm) const` override

Public Member Functions inherited from `quentier::utility::Printable`

- `QString toString () const`

Additional Inherited Members

Protected Member Functions inherited from `quentier::IQuentierException`

- `IQuentierException (ErrorString message)`
- `IQuentierException (const IQuentierException &other)`
- `IQuentierException & operator= (const IQuentierException &other)`

5.9.1 Detailed Description

The `IKeychainService::Exception` class is the base class for exceptions returned inside `QFutures` from methods of `IKeychainService`.

5.9.2 Member Function Documentation

5.9.2.1 `exceptionDisplayName()`

```
QString quentier::utility::IKeychainService::Exception::exceptionDisplayName () const [nodiscard],  
[override], [virtual]
```

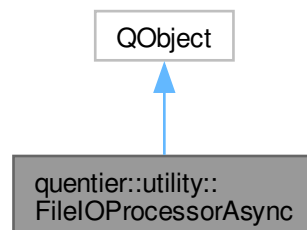
Implements `quentier::IQuentierException`.

5.10 `quentier::utility::FileIOProcessorAsync` Class Reference

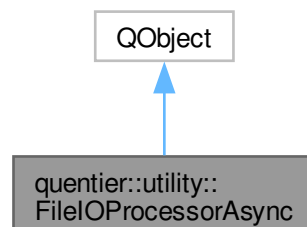
The `FileIOProcessorAsync` class is a wrapper under simple file IO operations, it is meant to be used for simple asynchronous IO.

```
#include <FileIOProcessorAsync.h>
```

Inheritance diagram for `quentier::utility::FileIOProcessorAsync`:



Collaboration diagram for `quentier::utility::FileIOProcessorAsync`:



Public Slots

- void `onWriteFileRequest` (QString absoluteFilePath, QByteArray data, QUuid requestId, bool append)
onWriteFileRequest slot processes file write requests with given request ids
- void `onReadFileRequest` (QString absoluteFilePath, QUuid requestId)
onReadFileRequest slot processes file read requests with given request ids

Signals

- void **readyForIO** ()
readyForIO signal is emitted when the queue for file IO is empty for some time (30 seconds by default, can also be configured via setIdleTimePeriod method) after the last IO event to signal listeners that they can perform some IO via the [FileIOProcessorAsync](#)
- void **writeFileRequestProcessed** (bool success, [ErrorString](#) errorDescription, QUuid requestId)
writeFileRequestProcessed signal is emitted when the file write request with given id is finished
- void **readFileRequestProcessed** (bool success, [ErrorString](#) errorDescription, QByteArray data, QUuid requestId)
readFileRequestProcessed signal is emitted when the file read request with given id is finished

Public Member Functions

- **FileIOProcessorAsync** (QObject *parent=nullptr)
- void **setIdleTimePeriod** (qint32 seconds)
setIdleTimePeriod sets time period defining the idle state of [FileIOProcessorAsync](#): once the time measured since the last IO operation is over the specified number of seconds, the class emits readyForIO signal to any interested listeners of this event. If this method is not called ever, the default idle time period would be 30 seconds.

5.10.1 Detailed Description

The [FileIOProcessorAsync](#) class is a wrapper under simple file IO operations, it is meant to be used for simple asynchronous IO.

5.10.2 Member Function Documentation

5.10.2.1 onReadFileRequest

```
void quentier::utility::FileIOProcessorAsync::onReadFileRequest (
    QString absoluteFilePath,
    QUuid requestId) [slot]
```

onReadFileRequest slot processes file read requests with given request ids

Parameters

<i>absoluteFilePath</i>	Absolute file path to be read
<i>requestId</i>	Unique identifier of the file read request

5.10.2.2 onWriteFileRequest

```
void quentier::utility::FileIOProcessorAsync::onWriteFileRequest (
    QString absoluteFilePath,
    QByteArray data,
    QUuid requestId,
    bool append) [slot]
```

onWriteFileRequest slot processes file write requests with given request ids

Parameters

<i>absoluteFilePath</i>	Absolute file path to be written
<i>data</i>	Data to be written to the file
<i>requestId</i>	Unique identifier of the file write request
<i>append</i>	If true, the data would be appended to file, otherwise the entire file would be erased before with the data is written

5.10.2.3 readFileRequestProcessed

```
void quentier::utility::FileIOProcessorAsync::readFileRequestProcessed (
    bool success,
    ErrorString errorDescription,
    QByteArray data,
    QUuid requestId) [signal]
```

readFileRequestProcessed signal is emitted when the file read request with given id is finished

Parameters

<i>success</i>	True if read operation was successful, false otherwise
<i>errorDescription</i>	Textual description of the error
<i>data</i>	Data read from file
<i>requestId</i>	Unique identifier of the file read request

5.10.2.4 setIdleTimePeriod()

```
void quentier::utility::FileIOProcessorAsync::setIdleTimePeriod (
    qint32 seconds)
```

setIdleTimePeriod sets time period defining the idle state of [FileIOProcessorAsync](#): once the time measured since the last IO operation is over the specified number of seconds, the class emits readyForIO signal to any interested listeners of this event. If this method is not called ever, the default idle time period would be 30 seconds.

Parameters

<i>seconds</i>	Number of seconds for idle time period
----------------	--

5.10.2.5 writeFileRequestProcessed

```
void quantier::utility::FileIOProcessorAsync::writeFileRequestProcessed (  
    bool success,  
    QString errorDescription,  
    QUuid requestId) [signal]
```

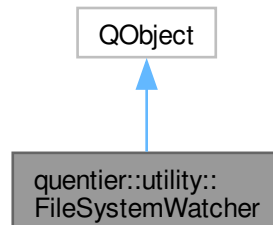
writeFileRequestProcessed signal is emitted when the file write request with given id is finished

Parameters

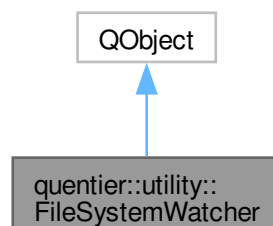
<i>success</i>	True if write operation was successful, false otherwise
<i>errorDescription</i>	Textual description of the error
<i>requestId</i>	Unique identifier of the file write request

5.11 quantier::utility::FileSystemWatcher Class Reference

Inheritance diagram for quantier::utility::FileSystemWatcher:



Collaboration diagram for quantier::utility::FileSystemWatcher:



Signals

- void **directoryChanged** (const QString &path)
- void **directoryRemoved** (const QString &path)
- void **fileChanged** (const QString &path)
- void **fileRemoved** (const QString &path)

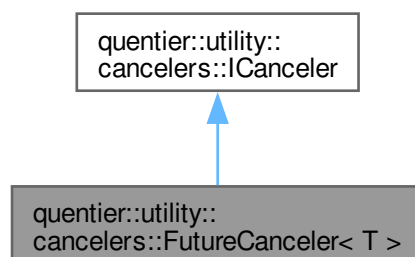
Public Member Functions

- **FileSystemWatcher** (int removalTimeoutMSec=500, QObject *parent=nullptr)
- **FileSystemWatcher** (const QStringList &paths, int removalTimeoutMSec=500, QObject *parent=nullptr)
- void **addPath** (const QString &path)
- void **addPaths** (const QStringList &paths)
- QStringList **directories** () const
- QStringList **files** () const
- void **removePath** (const QString &path)
- void **removePaths** (const QStringList &paths)

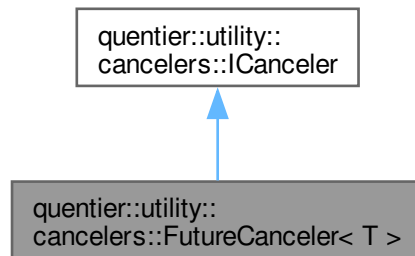
5.12 quantier::utility::cancelers::FutureCanceler< T > Class Template Reference

```
#include <FutureCanceler.h>
```

Inheritance diagram for quantier::utility::cancelers::FutureCanceler< T >:



Collaboration diagram for `quantier::utility::cancelers::FutureCanceler< T >`:



Public Member Functions

- **FutureCanceler** (`QFuture< T > future`)
- bool `isCanceled` () const noexcept override

5.12.1 Detailed Description

`template<class T>`
class `quantier::utility::cancelers::FutureCanceler< T >`

`ICanceler` implementation which tracks the canceled status of a future.

5.12.2 Member Function Documentation

5.12.2.1 `isCanceled()`

```

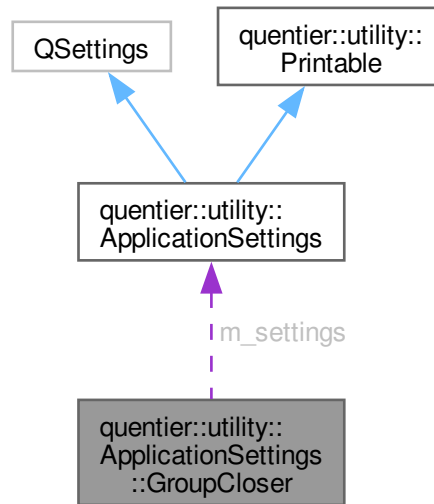
template<class T>
bool quantier::utility::cancelers::FutureCanceler< T >::isCanceled () const [inline], [nodiscard],
[override], [virtual], [noexcept]
  
```

Implements `quantier::utility::cancelers::ICanceler`.

5.13 quantier::utility::ApplicationSettings::GroupCloser Struct Reference

```
#include <ApplicationSettings.h>
```

Collaboration diagram for quantier::utility::ApplicationSettings::GroupCloser:



Public Member Functions

- **GroupCloser** ([ApplicationSettings](#) &settings)

Public Attributes

- [ApplicationSettings](#) & **m_settings**

5.13.1 Detailed Description

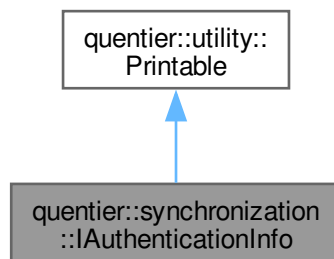
Helper struct for RAII style of ensuring the group once opened would be closed even if exception is thrown after beginning the group

5.14 `quentier::synchronization::IAuthenticationInfo` Class Reference

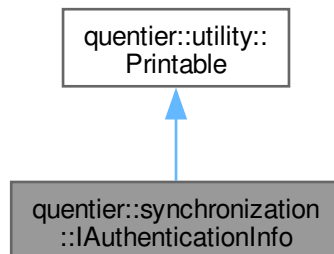
The `IAuthenticationInfo` interface represents the information obtained through OAuth and necessary to access Evernote API.

```
#include <IAuthenticationInfo.h>
```

Inheritance diagram for `quentier::synchronization::IAuthenticationInfo`:



Collaboration diagram for `quentier::synchronization::IAuthenticationInfo`:



Public Member Functions

- virtual `qevercloud::UserID` `userId` () const =0
- virtual `QString` `authToken` () const =0
- virtual `qevercloud::Timestamp` `authTokenExpirationTime` () const =0
- virtual `qevercloud::Timestamp` `authenticationTime` () const =0
- virtual `QString` `shardId` () const =0
- virtual `QString` `noteStoreUrl` () const =0
- virtual `QString` `webApiUrlPrefix` () const =0
- virtual `QList< QNetworkCookie >` `userStoreCookies` () const =0

Public Member Functions inherited from [quentier::utility::Printable](#)

- virtual QTextStream & **print** (QTextStream &strm) const =0
- QString **toString** () const

5.14.1 Detailed Description

The [IAuthenticationInfo](#) interface represents the information obtained through OAuth and necessary to access Evernote API.

5.14.2 Member Function Documentation

5.14.2.1 authenticationTime()

```
virtual qevercloud::Timestamp quentier::synchronization::IAuthenticationInfo::authenticationTime () const [nodiscard], [pure virtual]
```

Timestamp at which authentication info was received from Evernote

5.14.2.2 authToken()

```
virtual QString quentier::synchronization::IAuthenticationInfo::authToken () const [nodiscard], [pure virtual]
```

Authentication token which needs to be used for access to Evernote API

5.14.2.3 authTokenExpirationTime()

```
virtual qevercloud::Timestamp quentier::synchronization::IAuthenticationInfo::authTokenExpirationTime () const [nodiscard], [pure virtual]
```

Expiration timestamp for the authentication token

5.14.2.4 noteStoreUrl()

```
virtual QString quentier::synchronization::IAuthenticationInfo::noteStoreUrl () const [nodiscard], [pure virtual]
```

Url of the note store service for this user

5.14.2.5 shardId()

```
virtual QString quentier::synchronization::IAuthenticationInfo::shardId () const [nodiscard], [pure virtual]
```

Shard identifier which needs to be used for access to Evernote API along with the authentication token

5.14.2.6 userId()

```
virtual qevercloud::UserID quantier::synchronization::IAuthenticationInfo::userId () const
[nodiscard], [pure virtual]
```

Identifier of the authenticated user

5.14.2.7 userStoreCookies()

```
virtual QList< QNetworkCookie > quantier::synchronization::IAuthenticationInfo::userStore←
Cookies () const [nodiscard], [pure virtual]
```

The list of network cookies received during OAuth procedure. Although is is not mentioned anywhere in Evernote docs, these cookies might have to be used for access to user store. See this discussion for reference: <https://discussion.evernote.com/forums/topic/124257-calls-to-userstore-from-evernote-api-stop>

5.14.2.8 webApiUrlPrefix()

```
virtual QString quantier::synchronization::IAuthenticationInfo::webApiUrlPrefix () const [nodiscard],
[pure virtual]
```

Url prefix for Evernote Web API.

See also

qevercloud::PublicUserInfo::webApiUrlPrefix

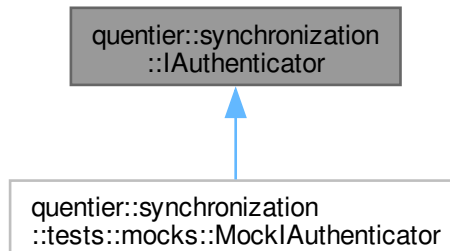
5.15 quantier::synchronization::IAuthenticationInfoBuilder Class Reference

Public Member Functions

- virtual [IAuthenticationInfoBuilder](#) & **setUserId** (qevercloud::UserID userId)=0
- virtual [IAuthenticationInfoBuilder](#) & **setAuthToken** (QString token)=0
- virtual [IAuthenticationInfoBuilder](#) & **setAuthTokenExpirationTime** (qevercloud::Timestamp expiration←Time)=0
- virtual [IAuthenticationInfoBuilder](#) & **setAuthenticationTime** (qevercloud::Timestamp authenticationTime)=0
- virtual [IAuthenticationInfoBuilder](#) & **setShardId** (QString shardId)=0
- virtual [IAuthenticationInfoBuilder](#) & **setNoteStoreUrl** (QString noteStoreUrl)=0
- virtual [IAuthenticationInfoBuilder](#) & **setWebApiUrlPrefix** (QString webApiUrlPrefix)=0
- virtual [IAuthenticationInfoBuilder](#) & **setUserStoreCookies** (QList< QNetworkCookie > cookies)=0
- virtual [IAuthenticationInfoPtr](#) **build** ()=0

5.16 `quentier::synchronization::IAuthenticator` Class Reference

Inheritance diagram for `quentier::synchronization::IAuthenticator`:



Public Member Functions

- virtual `QFuture< IAuthenticationInfoPtr >` **authenticateNewAccount** ()=0
- virtual `QFuture< IAuthenticationInfoPtr >` **authenticateAccount** ([Account](#) account)=0

5.17 `quentier::ResourceRecognitionIndexItem::IBarcodeItem` Struct Reference

Public Member Functions

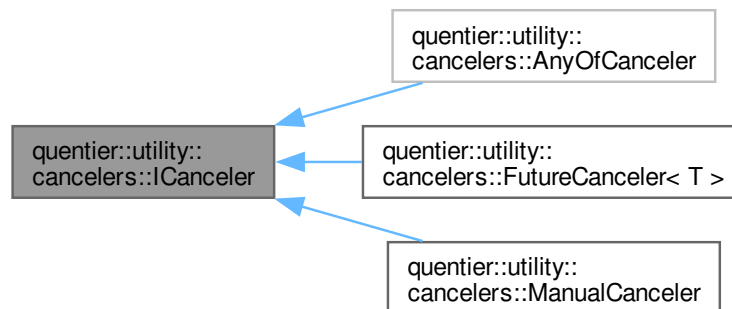
- virtual `QString` **barcode** () const =0
- virtual `int` **weight** () const =0

5.18 `quentier::utility::cancelers::ICanceler` Class Reference

The [ICanceler](#) interface provides `isCanceled` method which can be used to check whether some processing can be skipped because it was canceled.

```
#include <ICanceler.h>
```


Inheritance diagram for quantier::utility::cancelers::ICanceler:



Public Member Functions

- virtual bool **isCanceled** () const =0

5.18.1 Detailed Description

The [ICanceler](#) interface provides isCanceled method which can be used to check whether some processing can be skipped because it was canceled.

5.19 quantier::enml::IConverter Class Reference

The [IConverter](#) interface encapsulates a set of methods performing conversions between ENML and other note content formats, namely HTML.

```
#include <IConverter.h>
```

Public Types

- enum class [EnexExportTags](#) { **Yes** = 0 , **No** }

The [EnexExportTags](#) enum allows to specify whether export of note(s) to ENEX should include the names of note's tags or not.

Public Member Functions

- virtual [Result](#)< [QString](#), [ErrorString](#) > [convertHtmlToEnml](#) (const [QString](#) &html, [IDecryptedTextCache](#) &decryptedTextCache, const [QList](#)< [conversion_rules::ISkipRulePtr](#) > &skipRules={}) const =0
- virtual [Result](#)< void, [ErrorString](#) > [convertHtmlToDoc](#) (const [QString](#) &html, [QTextDocument](#) &doc, const [QList](#)< [conversion_rules::ISkipRulePtr](#) > &skipRules={}) const =0
- virtual [Result](#)< [QString](#), [ErrorString](#) > [convertHtmlToXml](#) (const [QString](#) &html) const =0
- virtual [Result](#)< [QString](#), [ErrorString](#) > [convertHtmlToXhtml](#) (const [QString](#) &html) const =0
- virtual [Result](#)< [IHtmlDataPtr](#), [ErrorString](#) > [convertEnmlToHtml](#) (const [QString](#) &enml, [IDecryptedTextCache](#) &decryptedTextCache) const =0
- virtual [Result](#)< [QString](#), [ErrorString](#) > [convertEnmlToPlainText](#) (const [QString](#) &enml) const =0
- virtual [Result](#)< [QStringList](#), [ErrorString](#) > [convertEnmlToWordsList](#) (const [QString](#) &enml) const =0
- virtual [QStringList](#) [convertPlainTextToWordsList](#) (const [QString](#) &plainText) const =0
- virtual [Result](#)< void, [ErrorString](#) > [validateEnml](#) (const [QString](#) &enml) const =0
- virtual [Result](#)< [QString](#), [ErrorString](#) > [validateAndFixupEnml](#) (const [QString](#) &enml) const =0
- virtual [Result](#)< [QString](#), [ErrorString](#) > [exportNotesToEnex](#) (const [QList](#)< [qevercloud::Note](#) > ¬es, const [QHash](#)< [QString](#), [QString](#) > &tagNamesByTagLocalIds, [EnexExportTags](#) exportTagsOption, const [QString](#) &version={}) const =0
- virtual [Result](#)< [QList](#)< [qevercloud::Note](#) >, [ErrorString](#) > [importEnex](#) (const [QString](#) &enex) const =0

5.19.1 Detailed Description

The [IConverter](#) interface encapsulates a set of methods performing conversions between ENML and other note content formats, namely HTML.

5.19.2 Member Function Documentation

5.19.2.1 [convertEnmlToHtml\(\)](#)

```
virtual Result< IHtmlDataPtr, ErrorString > quentier::enml::IConverter::convertEnmlToHtml (
    const QString & enml,
    IDecryptedTextCache & decryptedTextCache) const    [nodiscard], [pure virtual]
```

Converts ENML into HTML representation of note content

Parameters

<i>enml</i>	ENML representation of note content
<i>decryptedTextCache</i>	cache of decrypted text fragments

Returns

[Result](#) with HTML data in case of success or error string in case of failure

5.19.2.2 convertEnmlToPlainText()

```
virtual Result< QString, ErrorString > quantier::enml::IConverter::convertEnmlToPlainText (  
    const QString & enml) const [nodiscard], [pure virtual]
```

Converts ENML into plain text representation of note content

Parameters

<i>enml</i>	ENML representation of note content
-------------	-------------------------------------

Returns

[Result](#) with plain text representation of note content in case of success or error string in case of failure

5.19.2.3 convertEnmlToWordsList()

```
virtual Result< QStringList, ErrorString > quantier::enml::IConverter::convertEnmlToWordsList (  
    const QString & enml) const [nodiscard], [pure virtual]
```

Converts ENML into a list of words

Parameters

<i>enml</i>	ENML representation of note content
-------------	-------------------------------------

Returns

[Result](#) with list of words in case of success or error string in case of failure

5.19.2.4 convertHtmlToDoc()

```
virtual Result< void, ErrorString > quantier::enml::IConverter::convertHtmlToDoc (  
    const QString & html,  
    QTextDocument & doc,  
    const QList< conversion_rules::ISkipRulePtr > & skipRules = {}) const [nodiscard],  
[pure virtual]
```

Convert HTML representation of note content into QTextDocument

Parameters

<i>html</i>	HTML representation of note content
<i>doc</i>	QTextDocument into which the converted note content is put
<i>skipRules</i>	skip rules to be used during the conversion

Returns

Valid result in case of success or error string in case of failure

5.19.2.5 convertHtmlToEnml()

```
virtual Result< QString, ErrorString > quantier::enml::IConverter::convertHtmlToEnml (  
    const QString & html,  
    IDecryptedTextCache & decryptedTextCache,  
    const QList< conversion_rules::ISkipRulePtr > & skipRules = {}) const [nodiscard],  
[pure virtual]
```

Converts HTML representation of note content into ENML

Parameters

<i>html</i>	HTML representation of note content
<i>decryptedTextCache</i>	cache of decrypted text fragments
<i>skipRules</i>	skip rules to be used during the conversion

Returns

[Result](#) with ENML in case of success or error string in case of failure

5.19.2.6 convertHtmlToXhtml()

```
virtual Result< QString, ErrorString > quantier::enml::IConverter::convertHtmlToXhtml (  
    const QString & html) const [nodiscard], [pure virtual]
```

Convert HTML representation of note content into a valid XHTML document

Parameters

<i>html</i>	HTML representation of note content
-------------	-------------------------------------

Returns

[Result](#) with XHTML in case of success or error string in case of failure

5.19.2.7 convertHtmlToXml()

```
virtual Result< QString, ErrorString > quantier::enml::IConverter::convertHtmlToXml (  
    const QString & html) const [nodiscard], [pure virtual]
```

Convert HTML representation of note content into a valid XML document

Parameters

<i>html</i>	HTML representation of note content
-------------	-------------------------------------

Returns

[Result](#) with XML in case of success or error string in case of failure

5.19.2.8 convertPlainTextToWordsList()

```
virtual QStringList quotientier::enml::IConverter::convertPlainTextToWordsList (
    const QString & plainText) const [nodiscard], [pure virtual]
```

Converts plain text into a list of words

Parameters

<i>plainText</i>	plain text representation of note content
------------------	---

Returns

list of words

5.19.2.9 exportNotesToEnex()

```
virtual Result< QString, ErrorString > quotientier::enml::IConverter::exportNotesToEnex (
    const QList< qevercloud::Note > & notes,
    const QHash< QString, QString > & tagNamesByTagLocalIds,
    EnexExportTags exportTagsOption,
    const QString & version = {}) const [nodiscard], [pure virtual]
```

Exports a list of notes into ENEX

Parameters

<i>notes</i>	notes to be exported into ENEX
<i>tagNamesByTagLocalIds</i>	mapper from tag local ids into tag names
<i>exportTagsOption</i>	option controlling the export of tag names
<i>version</i>	optional version tag for ENEX, omitted if not set

Returns

[Result](#) with ENEX in case of success or error string in case of failure

5.19.2.10 importEnex()

```
virtual Result< QList< qevercloud::Note >, ErrorString > quotientier::enml::IConverter::importEnex (
    const QString & enex) const [nodiscard], [pure virtual]
```

Import notes from ENEX

Parameters

<i>enex</i>	ENEX to be used for import
-------------	----------------------------

Returns

[Result](#) with list of notes in case of success or error string in case of failure

Note

if tag names are present in ENEX, corresponding notes would have their tagName field filled

5.19.2.11 validateAndFixupEnml()

```
virtual Result< QString, ErrorString > quantier::enml::IConverter::validateAndFixupEnml (
    const QString & enml) const [nodiscard], [pure virtual]
```

Validates ENML and attempts to fix it automatically if it's not valid

Parameters

<i>enml</i>	ENML representation of note content
-------------	-------------------------------------

Returns

[Result](#) with either unchanged or fixed up ENML in case of success or error string in case of failure

5.19.2.12 validateEnml()

```
virtual Result< void, ErrorString > quantier::enml::IConverter::validateEnml (
    const QString & enml) const [nodiscard], [pure virtual]
```

Validates ENML against rules

Parameters

<i>enml</i>	ENML representation of note content
-------------	-------------------------------------

Returns

valid [Result](#) in case of success or error string in case of failure

5.20 quantier::enml::IDecryptedTextCache Class Reference**Public Types**

- enum class **RememberForSession** { **Yes** , **No** }

Public Member Functions

- virtual void **addDecryptTextInfo** (const QString &encryptedText, const QString &decryptedText, const QString &passphrase, [utility::IEncryptor::Cipher](#) cipher, RememberForSession rememberForSession)=0
- virtual std::optional< std::pair< QString, RememberForSession > > **findDecryptedTextInfo** (const QString &encryptedText) const =0
- virtual std::optional< QString > **updateDecryptedTextInfo** (const QString &originalEncryptedText, const QString &newDecryptedText)=0
- virtual bool **containsRememberedForSessionEntries** () const =0
- virtual void **removeDecryptedTextInfo** (const QString &encryptedText)=0
- virtual void **clearNonRememberedForSessionEntries** ()=0

Friends

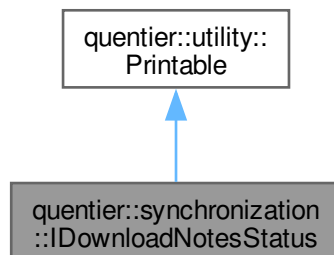
- QUENTIER_EXPORT QDebug & **operator<<** (QDebug &dbg, RememberForSession rememberForSession)
- QUENTIER_EXPORT QTextStream & **operator<<** (QTextStream &strm, RememberForSession rememberForSession)

5.21 quantier::synchronization::IDownloadNotesStatus Class Reference

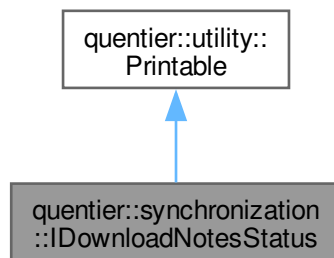
The [IDownloadNotesStatus](#) interface presents information about the status of notes downloading process.

```
#include <IDownloadNotesStatus.h>
```

Inheritance diagram for quantier::synchronization::IDownloadNotesStatus:



Collaboration diagram for `quentier::synchronization::IDownloadNotesStatus`:



Public Types

- using **QExceptionPtr** = `std::shared_ptr<QException>`
- using **NoteWithException** = `std::pair<qevercloud::Note, QExceptionPtr>`
- using **GuidWithException** = `std::pair<qevercloud::Guid, QExceptionPtr>`
- using **UpdateSequenceNumbersByGuid** = `QHash<qevercloud::Guid, qint32>`

Public Member Functions

- virtual quint64 **totalNewNotes** () const =0
- virtual quint64 **totalUpdatedNotes** () const =0
- virtual quint64 **totalExpungedNotes** () const =0
- virtual QList< NoteWithException > **notesWhichFailedToDownload** () const =0
- virtual QList< NoteWithException > **notesWhichFailedToProcess** () const =0
- virtual QList< GuidWithException > **noteGuidsWhichFailedToExpunge** () const =0
- virtual UpdateSequenceNumbersByGuid **processedNoteGuidsAndUsns** () const =0
- virtual UpdateSequenceNumbersByGuid **cancelledNoteGuidsAndUsns** () const =0
- virtual QList< qevercloud::Guid > **expungedNoteGuids** () const =0
- virtual StopSynchronizationError **stopSynchronizationError** () const =0

Public Member Functions inherited from [quentier::utility::Printable](#)

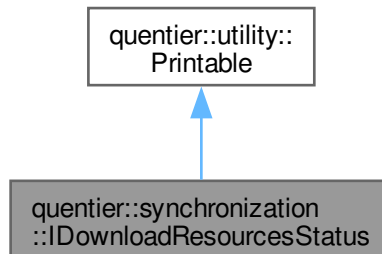
- virtual QTextStream & **print** (QTextStream &strm) const =0
- QString **toString** () const

5.21.1 Detailed Description

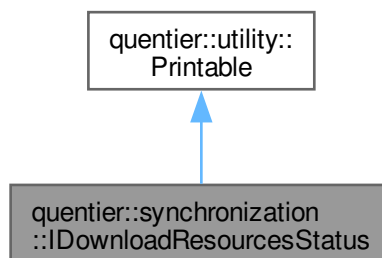
The [IDownloadNotesStatus](#) interface presents information about the status of notes downloading process.

5.22 quantier::synchronization::IDownloadResourcesStatus Class Reference

Inheritance diagram for quantier::synchronization::IDownloadResourcesStatus:



Collaboration diagram for quantier::synchronization::IDownloadResourcesStatus:



Public Types

- using **QExceptionPtr** = std::shared_ptr<QException>
- using [ResourceWithException](#)
- using **UpdateSequenceNumbersByGuid** = QHash<qevercloud::Guid, qint32>

Public Member Functions

- virtual quint64 **totalNewResources** () const =0
- virtual quint64 **totalUpdatedResources** () const =0
- virtual QList< ResourceWithException > **resourcesWhichFailedToDownload** () const =0
- virtual QList< ResourceWithException > **resourcesWhichFailedToProcess** () const =0
- virtual UpdateSequenceNumbersByGuid **processedResourceGuidsAndUsns** () const =0
- virtual UpdateSequenceNumbersByGuid **cancelledResourceGuidsAndUsns** () const =0
- virtual StopSynchronizationError **stopSynchronizationError** () const =0

Public Member Functions inherited from [quentier::utility::Printable](#)

- virtual QTextStream & **print** (QTextStream &strm) const =0
- QString **toString** () const

5.22.1 Member Typedef Documentation

5.22.1.1 ResourceWithException

```
using quentier::synchronization::IDownloadResourcesStatus::ResourceWithException
```

Initial value:

```
std::pair<qevercloud::Resource, QExceptionPtr>
```

5.23 quentier::utility::IEncryptor Struct Reference

The [IEncryptor](#) interface provides encryption and decryption functionality which is compatible with that used by Evernote service.

```
#include <IEncryptor.h>
```

Public Types

- enum class [Cipher](#) { [RC2](#) , [AES](#) }

Public Member Functions

- virtual [Result](#)< QString, [ErrorString](#) > **encrypt** (const QString &text, const QString &passphrase)=0
- virtual [Result](#)< QString, [ErrorString](#) > **decrypt** (const QString &encryptedText, const QString &passphrase, [Cipher](#) cipher)=0

Friends

- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, [Cipher](#) cipher)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, [Cipher](#) cipher)

5.23.1 Detailed Description

The [IEncryptor](#) interface provides encryption and decryption functionality which is compatible with that used by Evernote service.

5.23.2 Member Enumeration Documentation

5.23.2.1 Cipher

```
enum class quotientier::utility::IEncryptor::Cipher [strong]
```

[Cipher](#) used for encryption/decryption

Enumerator

RC2	RC2 64 bit block cipher
AES	AES 128 bit block cipher

5.23.3 Member Function Documentation

5.23.3.1 decrypt()

```
virtual Result< QString, ErrorString > quotientier::utility::IEncryptor::decrypt (  
    const QString & encryptedText,  
    const QString & passphrase,  
    Cipher cipher) [nodiscard], [pure virtual]
```

Decrypt previously encrypted text fragment

Parameters

<i>encryptedText</i>	Encrypted text to decrypt
<i>passphrase</i>	Passhprase used to encrypt text
<i>cipher</i>	Cipher used to encrypt text

Returns

[Result](#) with either decrypted text or error message

5.23.3.2 encrypt()

```
virtual Result< QString, ErrorString > quotientier::utility::IEncryptor::encrypt (  
    const QString & text,  
    const QString & passphrase) [nodiscard], [pure virtual]
```

Encrypt text fragment using AES cipher (RC2 cipher is only used for decryption)

Parameters

<i>text</i>	Text to encrypt
-------------	-----------------

<i>passphrase</i>	Passphrase which can be used to decrypt the text
-------------------	--

Returns

[Result](#) with either encrypted text or error message

5.24 quotientier::enml::IENMLTagsConverter Class Reference

The [IENMLTagsConverter](#) interfaces provides methods which convert Evernote-specific markup tags such as en-crypt, en-media etc. into their counterparts which should be used in the HTML representation of note content.

```
#include <IENMLTagsConverter.h>
```

Public Member Functions

- virtual QString [convertEnToDo](#) (bool checked, quint32 index) const =0
- virtual QString [convertEncryptedText](#) (const QString &encryptedText, const QString &hint, [utility::IEncryptor::Cipher](#) cipher, quint32 index) const =0
- virtual QString [convertDecryptedText](#) (const QString &decryptedText, const QString &encryptedText, const QString &hint, [utility::IEncryptor::Cipher](#) cipher, quint32 index) const =0
- virtual [Result](#)< QString, [ErrorString](#) > [convertResource](#) (const qevercloud::Resource &resource) const =0

5.24.1 Detailed Description

The [IENMLTagsConverter](#) interfaces provides methods which convert Evernote-specific markup tags such as en-crypt, en-media etc. into their counterparts which should be used in the HTML representation of note content.

5.24.2 Member Function Documentation

5.24.2.1 convertDecryptedText()

```
virtual QString quotientier::enml::IENMLTagsConverter::convertDecryptedText (
    const QString & decryptedText,
    const QString & encryptedText,
    const QString & hint,
    utility::IEncryptor::Cipher cipher,
    quint32 index) const [nodiscard], [pure virtual]
```

Converts already decrypted en-crypt tag into its HTML counterpart

Parameters

<i>decryptedText</i>	decrypted text from en-crypt tag
<i>encryptedText</i>	encrypted text contained within en-crypt tag
<i>hint</i>	hint to be displayed when user tries to decrypt the text

<i>cipher</i>	cipher used to encrypt the text
<i>index</i>	index of particular en-crypt tag within the note content so that different en-crypt tags can be differentiated

Returns

HTML representation of decrypted en-crypt tag

5.24.2.2 convertEncryptedText()

```
virtual QString quantier::enml::IENMLTagsConverter::convertEncryptedText (  
    const QString & encryptedText,  
    const QString & hint,  
    utility::IEncryptor::Cipher cipher,  
    quint32 index) const [nodiscard], [pure virtual]
```

Converts en-crypt tag into its HTML counterpart

Parameters

<i>encryptedText</i>	encrypted text contained within en-crypt tag
<i>hint</i>	hint to be displayed when user tries to decrypt the text
<i>cipher</i>	cipher used to encrypt the text
<i>index</i>	index of particular en-crypt tag within the note content so that different en-crypt tags can be differentiated

Returns

HTML representation of en-crypt tag

5.24.2.3 convertEnToDo()

```
virtual QString quantier::enml::IENMLTagsConverter::convertEnToDo (  
    bool checked,  
    quint32 index) const [nodiscard], [pure virtual]
```

Converts en-todo tag into its HTML counterpart

Parameters

<i>checked</i>	indicates whether this todo is checked or not
<i>index</i>	index of particular en-todo tag within the note content so that different todo tags can be differentiated

Returns

HTML representation of en-todo tag

5.24.2.4 convertResource()

```
virtual Result< QString, ErrorString > quantier::enml::IENMLTagsConverter::convertResource (
    const qevercloud::Resource & resource) const [nodiscard], [pure virtual]
```

Converts en-media tag representing a resource into its HTML counterpart

Parameters

<i>resource</i>	resource corresponding to en-media tag
-----------------	--

Returns

[Result](#) with valid HTML representing the resource/en-media tag in case of success or error string in case of failure

5.25 quantier::synchronization::ISyncConflictResolver::ConflictResolution::IgnoreMine Struct Reference

The [IgnoreMine](#) conflict resolution means "use theirs version and ignore mine version as it doesn't really conflict with theirs version".

```
#include <ISyncConflictResolver.h>
```

5.25.1 Detailed Description

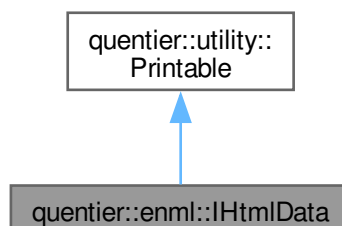
The [IgnoreMine](#) conflict resolution means "use theirs version and ignore mine version as it doesn't really conflict with theirs version".

5.26 quantier::enml::IHtmlData Struct Reference

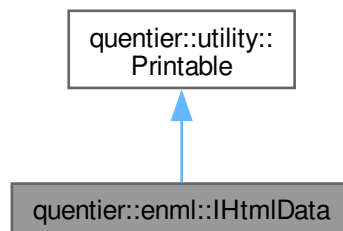
The [IHtmlData](#) represents the result of ENML to HTML conversion: HTML itself plus some metadata.

```
#include <IHtmlData.h>
```

Inheritance diagram for quantier::enml::IHtmlData:



Collaboration diagram for `quentier::enml::IHtmlData`:



Public Member Functions

- virtual `QString` `html` () const =0
- virtual `quint32` `numEnToDoNodes` () const =0
- virtual `quint32` `numHyperlinkNodes` () const =0
- virtual `quint32` `numEnCryptNodes` () const =0
- virtual `quint32` `numEnDecryptedNodes` () const =0
- `QTextStream` & `print` (`QTextStream` &strm) const override

Public Member Functions inherited from `quentier::utility::Printable`

- `QString` `toString` () const

5.26.1 Detailed Description

The `IHtmlData` represents the result of ENML to HTML conversion: HTML itself plus some metadata.

5.26.2 Member Function Documentation

5.26.2.1 `html()`

```
virtual QString quentier::enml::IHtmlData::html () const [nodiscard], [pure virtual]
```

HTML representation of note content

5.26.2.2 `numEnCryptNodes()`

```
virtual quint32 quentier::enml::IHtmlData::numEnCryptNodes () const [nodiscard], [pure virtual]
```

Number of en-crypt nodes within note HTML

5.26.2.3 numEnDecryptedNodes()

```
virtual quint32 quentier::enml::IHtmlData::numEnDecryptedNodes () const [nodiscard], [pure virtual]
```

Number of decrypted en-crypt nodes within note HTML

5.26.2.4 numEnToDoNodes()

```
virtual quint32 quentier::enml::IHtmlData::numEnToDoNodes () const [nodiscard], [pure virtual]
```

Number of ToDo nodes within note HTML

5.26.2.5 numHyperlinkNodes()

```
virtual quint32 quentier::enml::IHtmlData::numHyperlinkNodes () const [nodiscard], [pure virtual]
```

Number of hyperlink nodes within note HTML

5.26.2.6 print()

```
QTextStream & quentier::enml::IHtmlData::print (
    QTextStream & strm) const [override], [virtual]
```

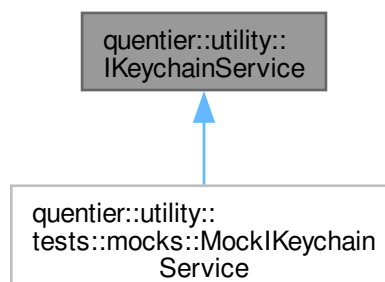
Implements [quentier::utility::Printable](#).

5.27 quentier::utility::IKeychainService Class Reference

The [IKeychainService](#) interface provides the ability to interact with the storage of sensitive data - read, write and delete it.

```
#include <IKeychainService.h>
```

Inheritance diagram for quentier::utility::IKeychainService:



Classes

- class [Exception](#)

The [IKeychainService::Exception](#) class is the base class for exceptions returned inside QFutures from methods of [IKeychainService](#).

Public Types

- enum class [ErrorCode](#) {
[NoError](#) , [EntryNotFound](#) , [CouldNotDeleteEntry](#) , [AccessDeniedByUser](#) ,
[AccessDenied](#) , [NoBackendAvailable](#) , [NotImplemented](#) , [OtherError](#) }

Public Member Functions

- virtual QFuture< void > [writePassword](#) (QString service, QString key, QString password)=0
- virtual QFuture< QString > [readPassword](#) (QString service, QString key) const =0
- virtual QFuture< void > [deletePassword](#) (QString service, QString key)=0

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, [ErrorCode](#) errorCode)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, [ErrorCode](#) errorCode)

5.27.1 Detailed Description

The [IKeychainService](#) interface provides the ability to interact with the storage of sensitive data - read, write and delete it.

5.27.2 Member Enumeration Documentation

5.27.2.1 ErrorCode

```
enum class quentier::utility::IKeychainService::ErrorCode [strong]
```

Error codes for results of operations with the keychain service

Enumerator

NoError	No error occurred, operation was successful
EntryNotFound	For the given key no data was found
CouldNotDeleteEntry	Could not delete existing secret data
AccessDeniedByUser	User denied access to keychain
AccessDenied	Access denied for some reason
NoBackendAvailable	No platform-specific keychain service available
NotImplemented	Not implemented on platform

OtherError	Something else went wrong, the error description specifies what
------------	---

5.27.3 Member Function Documentation

5.27.3.1 deletePassword()

```
virtual QFuture< void > quentier::utility::IKeychainService::deletePassword (
    QString service,
    QString key) [nodiscard], [pure virtual]
```

deletePassword potentially asynchronously deletes password from the keychain.

Parameters

<i>service</i>	Name of service within the keychain
<i>key</i>	Key under which the password is stored

Returns

Future which becomes finished when the operation is complete. If the operation fails, the future would contain an exception.

5.27.3.2 readPassword()

```
virtual QFuture< QString > quentier::utility::IKeychainService::readPassword (
    QString service,
    QString key) const [nodiscard], [pure virtual]
```

readPassword method potentially asynchronously reads password from the keychain.

Parameters

<i>service</i>	Name of service within the keychain
<i>key</i>	Key under which the password is stored

Returns

Future which becomes finished when the operation is complete. The value inside the future would be the read password. If the operation fails, the future would contain an exception.

5.27.3.3 writePassword()

```
virtual QFuture< void > quantier::utility::IKeychainService::writePassword (
    QString service,
    QString key,
    QString password) [nodiscard], [pure virtual]
```

writePassword method potentially asynchronously writes password to the keychain.

Parameters

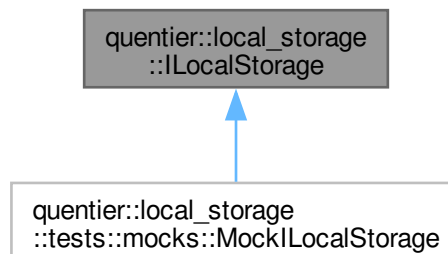
<i>service</i>	Name of service within the keychain
<i>key</i>	Key to store the password under
<i>password</i>	Password to store in the keychain

Returns

Future which becomes finished when the operation is complete. If the operation fails, the future would contain an exception.

5.28 quantier::local_storage::ILocalStorage Class Reference

Inheritance diagram for quantier::local_storage::ILocalStorage:



Classes

- struct [ListObjectsFilters](#)
- struct [ListGuidsFilters](#)
- struct [ListOptionsBase](#)
- struct [ListNotebooksOptions](#)
- struct [ListLinkedNotebooksOptions](#)
- struct [ListSavedSearchesOptions](#)
- struct [ListNotesOptions](#)
- struct [ListTagsOptions](#)

Public Types

- enum class **StartupOption** { **ClearDatabase** = 1 << 1 , **OverrideLock** = 1 << 2 }
- enum class **ListObjectsFilter** { **Include** , **Exclude** }
- enum class **OrderDirection** { **Ascending** , **Descending** }
- enum class **ListNotebooksOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByNotebookName** , **ByCreationTimestamp** , **ByModificationTimestamp** }
- enum class **ListLinkedNotebooksOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByShareName** , **ByUsername** }
- enum class **ListTagsOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByName** }
- enum class **ListNotesOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByTitle** , **ByCreationTimestamp** , **ByModificationTimestamp** , **ByDeletionTimestamp** , **ByAuthor** , **BySource** , **BySourceApplication** , **ByReminderTime** , **ByPlaceName** }
- enum class **ListSavedSearchesOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByName** , **ByFormat** }
- enum class **Affiliation** { **Any** , **User** , **AnyLinkedNotebook** , **ParticularLinkedNotebooks** }
- enum class **TagNotesRelation** { **Any** , **WithNotes** , **WithoutNotes** }
- enum class **NoteCountOption** { **IncludeNonDeletedNotes** = 1 << 1 , **IncludeDeletedNotes** = 1 << 2 }
- enum class **UpdateNoteOption** { **UpdateResourceMetadata** = 1 << 1 , **UpdateResourceBinaryData** = 1 << 2 , **UpdateTags** = 1 << 3 }
- enum class **FetchNoteOption** { **WithResourceMetadata** = 1 << 1 , **WithResourceBinaryData** = 1 << 2 }
- enum class **FetchResourceOption** { **WithBinaryData** = 1 << 1 }
- enum class **HighestUsnOption** { **WithinUserOwnContent** , **WithinUserOwnContentAndLinkedNotebooks** }

Public Member Functions

- **Q_DECLARE_FLAGS** (StartupOptions, StartupOption)
- virtual QFuture< bool > **isVersionTooHigh** () const =0
- virtual QFuture< bool > **requiresUpgrade** () const =0
- virtual QFuture< QList< IPatchPtr > > **requiredPatches** () const =0
- virtual QFuture< quint32 > **version** () const =0
- virtual QFuture< quint32 > **highestSupportedVersion** () const =0
- virtual QFuture< quint32 > **userCount** () const =0
- virtual QFuture< void > **putUser** (qevercloud::User user)=0
- virtual QFuture< std::optional< qevercloud::User > > **findUserById** (qevercloud::UserID userId) const =0
- virtual QFuture< void > **expungeUserById** (qevercloud::UserID userId)=0
- virtual QFuture< quint32 > **notebookCount** () const =0
- virtual QFuture< void > **putNotebook** (qevercloud::Notebook notebook)=0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findNotebookByLocalId** (QString notebookLocalId) const =0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findNotebookByGuid** (qevercloud::Guid guid) const =0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findNotebookByName** (QString notebookName, std::optional< qevercloud::Guid > linkedNotebookGuid=std::nullopt) const =0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findDefaultNotebook** () const =0
- virtual QFuture< void > **expungeNotebookByLocalId** (QString notebookLocalId)=0
- virtual QFuture< void > **expungeNotebookByGuid** (qevercloud::Guid notebookGuid)=0
- virtual QFuture< void > **expungeNotebookByName** (QString name, std::optional< qevercloud::Guid > linkedNotebookGuid=std::nullopt)=0
- virtual QFuture< QList< qevercloud::Notebook > > **listNotebooks** (ListNotebooksOptions options={}) const =0
- virtual QFuture< QList< qevercloud::SharedNotebook > > **listSharedNotebooks** (qevercloud::Guid notebookGuid={}) const =0

- virtual `QFuture< QSet< qevercloud::Guid > >` **listNotebookGuids** ([ListGuidsFilters](#) filters, `std::optional< qevercloud::Guid > linkedNotebookGuid={}`) `const =0`
- virtual `QFuture< quint32 >` **linkedNotebookCount** () `const =0`
- virtual `QFuture< void >` **putLinkedNotebook** (`qevercloud::LinkedNotebook linkedNotebook`)=`0`
- virtual `QFuture< std::optional< qevercloud::LinkedNotebook > >` **findLinkedNotebookByGuid** (`qevercloud::Guid guid`) `const =0`
- virtual `QFuture< void >` **expungeLinkedNotebookByGuid** (`qevercloud::Guid guid`)=`0`
- virtual `QFuture< QList< qevercloud::LinkedNotebook > >` **listLinkedNotebooks** ([ListLinkedNotebooksOptions](#) options=`{}`) `const =0`
- virtual `QFuture< quint32 >` **noteCount** (`NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)`) `const =0`
- virtual `QFuture< quint32 >` **noteCountPerNotebookLocalId** (`QString notebookLocalId, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)`) `const =0`
- virtual `QFuture< quint32 >` **noteCountPerTagLocalId** (`QString tagLocalId, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)`) `const =0`
- virtual `QFuture< QHash< QString, quint32 > >` **noteCountsPerTags** ([ListTagsOptions](#) listTagsOptions=`{}`, `NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)`) `const =0`
- virtual `QFuture< quint32 >` **noteCountPerNotebookAndTagLocalIds** (`QStringList notebookLocalIds, QStringList tagLocalIds, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)`) `const =0`
- virtual `QFuture< void >` **putNote** (`qevercloud::Note note`)=`0`
- virtual `QFuture< void >` **updateNote** (`qevercloud::Note note, UpdateNoteOptions options`)=`0`
- virtual `QFuture< std::optional< qevercloud::Note > >` **findNoteByLocalId** (`QString noteLocalId, FetchNoteOptions options`) `const =0`
- virtual `QFuture< std::optional< qevercloud::Note > >` **findNoteByGuid** (`qevercloud::Guid noteGuid, FetchNoteOptions options`) `const =0`
- virtual `QFuture< QList< qevercloud::Note > >` **listNotes** (`FetchNoteOptions fetchOptions, ListNotesOptions listOptions={}`) `const =0`
- virtual `QFuture< QList< qevercloud::Note > >` **listNotesPerNotebookLocalId** (`QString notebookLocalId, FetchNoteOptions fetchOptions, ListNotesOptions listOptions={}`) `const =0`
- virtual `QFuture< QList< qevercloud::Note > >` **listNotesPerTagLocalId** (`QString tagLocalId, FetchNoteOptions fetchOptions, ListNotesOptions listOptions={}`) `const =0`
- virtual `QFuture< QList< qevercloud::Note > >` **listNotesPerNotebookAndTagLocalIds** (`QStringList notebookLocalIds, QStringList tagLocalIds, FetchNoteOptions fetchOptions, ListNotesOptions listOptions={}`) `const =0`
- virtual `QFuture< QList< qevercloud::Note > >` **listNotesByLocalIds** (`QStringList noteLocalIds, FetchNoteOptions fetchOptions, ListNotesOptions listOptions={}`) `const =0`
- virtual `QFuture< QSet< qevercloud::Guid > >` **listNoteGuids** ([ListGuidsFilters](#) filters, `std::optional< qevercloud::Guid > linkedNotebookGuid={}`) `const =0`
- virtual `QFuture< QList< qevercloud::Note > >` **queryNotes** ([NoteSearchQuery](#) query, `FetchNoteOptions fetchOptions`) `const =0`
- virtual `QFuture< QStringList >` **queryNoteLocalIds** ([NoteSearchQuery](#) query) `const =0`
- virtual `QFuture< void >` **expungeNoteByLocalId** (`QString noteLocalId`)=`0`
- virtual `QFuture< void >` **expungeNoteByGuid** (`qevercloud::Guid noteGuid`)=`0`
- virtual `QFuture< quint32 >` **tagCount** () `const =0`
- virtual `QFuture< void >` **putTag** (`qevercloud::Tag tag`)=`0`
- virtual `QFuture< std::optional< qevercloud::Tag > >` **findTagByLocalId** (`QString tagLocalId`) `const =0`
- virtual `QFuture< std::optional< qevercloud::Tag > >` **findTagByGuid** (`qevercloud::Guid tagGuid`) `const =0`
- virtual `QFuture< std::optional< qevercloud::Tag > >` **findTagByName** (`QString tagName, std::optional< qevercloud::Guid > linkedNotebookGuid=std::nullopt`) `const =0`
- virtual `QFuture< QList< qevercloud::Tag > >` **listTags** ([ListTagsOptions](#) options=`{}`) `const =0`
- virtual `QFuture< QList< qevercloud::Tag > >` **listTagsPerNoteLocalId** (`QString noteLocalId, ListTagsOptions options={}`) `const =0`
- virtual `QFuture< QSet< qevercloud::Guid > >` **listTagGuids** ([ListGuidsFilters](#) filters, `std::optional< qevercloud::Guid > linkedNotebookGuid={}`) `const =0`
- virtual `QFuture< void >` **expungeTagByLocalId** (`QString tagLocalId`)=`0`

- virtual QFuture< void > **expungeTagByGuid** (qevercloud::Guid tagGuid)=0
- virtual QFuture< void > **expungeTagByName** (QString name, std::optional< qevercloud::Guid > linked← NotebookGuid=std::nullopt)=0
- virtual QFuture< quint32 > **resourceCount** (NoteCountOptions options=NoteCountOptions(NoteCount← Option::IncludeNonDeletedNotes)) const =0
- virtual QFuture< quint32 > **resourceCountPerNoteLocalId** (QString noteLocalId) const =0
- virtual QFuture< void > **putResource** (qevercloud::Resource resource)=0
- virtual QFuture< std::optional< qevercloud::Resource > > **findResourceByLocalId** (QString resource← LocalId, FetchResourceOptions options={}) const =0
- virtual QFuture< std::optional< qevercloud::Resource > > **findResourceByGuid** (qevercloud::Guid resourceGuid, FetchResourceOptions options={}) const =0
- virtual QFuture< void > **expungeResourceByLocalId** (QString resourceLocalId)=0
- virtual QFuture< void > **expungeResourceByGuid** (qevercloud::Guid resourceGuid)=0
- virtual QFuture< quint32 > **savedSearchCount** () const =0
- virtual QFuture< void > **putSavedSearch** (qevercloud::SavedSearch search)=0
- virtual QFuture< std::optional< qevercloud::SavedSearch > > **findSavedSearchByLocalId** (QString savedSearchLocalId) const =0
- virtual QFuture< std::optional< qevercloud::SavedSearch > > **findSavedSearchByGuid** (qevercloud::Guid guid) const =0
- virtual QFuture< std::optional< qevercloud::SavedSearch > > **findSavedSearchByName** (QString name) const =0
- virtual QFuture< QList< qevercloud::SavedSearch > > **listSavedSearches** ([ListSavedSearchesOptions](#) options={}) const =0
- virtual QFuture< QSet< qevercloud::Guid > > **listSavedSearchGuids** ([ListGuidsFilters](#) filters) const =0
- virtual QFuture< void > **expungeSavedSearchByLocalId** (QString savedSearchLocalId)=0
- virtual QFuture< void > **expungeSavedSearchByGuid** (qevercloud::Guid guid)=0
- virtual QFuture< qint32 > **highestUpdateSequenceNumber** (HighestUsnOption option) const =0
- virtual QFuture< qint32 > **highestUpdateSequenceNumber** (qevercloud::Guid linkedNotebookGuid) const =0
- virtual [ILocalStorageNotifier](#) * **notifier** () const =0

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, StartupOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, StartupOption option)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, StartupOptions options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, StartupOptions options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListObjectsFilter filter)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListObjectsFilter filter)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const [ListObjectsFilters](#) &filters)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const [ListObjectsFilters](#) &filters)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const [ListGuidsFilters](#) &filters)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const [ListGuidsFilters](#) &filters)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, OrderDirection orderDirection)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, OrderDirection orderDirection)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListNotebooksOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListNotebooksOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListLinkedNotebooksOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListLinkedNotebooksOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListTagsOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListTagsOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListNotesOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListNotesOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListSavedSearchesOrder order)

- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, ListSavedSearchesOrder order)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, Affiliation affiliation)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, Affiliation affiliation)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const ListNotebooksOptions &options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const ListNotebooksOptions &options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const ListLinkedNotebooksOptions &options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const ListLinkedNotebooksOptions &options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const ListSavedSearchesOptions &options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const ListSavedSearchesOptions &options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const ListNotesOptions &options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const ListNotesOptions &options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const ListTagsOptions &options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const ListTagsOptions &options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, NoteCountOption option)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, NoteCountOption option)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, NoteCountOptions options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, NoteCountOptions options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, UpdateNoteOption option)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, UpdateNoteOption option)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, UpdateNoteOptions options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, UpdateNoteOptions options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, FetchNoteOption option)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, FetchNoteOption option)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, FetchNoteOptions options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, FetchNoteOptions options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, FetchResourceOption option)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, FetchResourceOption option)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, FetchResourceOptions options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, FetchResourceOptions options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, HighestUsnOption option)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, HighestUsnOption option)`

5.28.1 Member Enumeration Documentation

5.28.1.1 `Affiliation`

```
enum class quentier::local\_storage::ILocalStorage::Affiliation [strong]
```

Denotes whether some data item belongs to user's own account, any of linked notebooks or particular linked notebooks

5.28.1.2 TagNotesRelation

```
enum class quantier::local_storage::ILocalStorage::TagNotesRelation [strong]
```

Denotes the relation between tag and notes - whether any note us using the given tag

Enumerator

Any	The tag might be used by some notes or it might not be.
WithNotes	The tag is used by some notes.
WithoutNotes	The tag is not used by any note.

5.28.2 Member Function Documentation

5.28.2.1 notifier()

```
virtual ILocalStorageNotifier * quantier::local_storage::ILocalStorage::notifier () const
[nodiscard], [pure virtual]
```

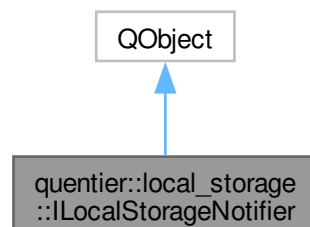
Notifications about the events occurring with the local storage are done via signals emitted by [ILocalStorageNotifier](#).

Returns

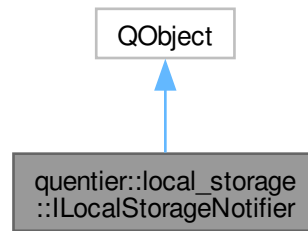
the pointer to [ILocalStorageNotifier](#) object which would be valid for at least as long as [ILocalStorage](#) object that returned it is alive.

5.29 quantier::local_storage::ILocalStorageNotifier Class Reference

Inheritance diagram for quantier::local_storage::ILocalStorageNotifier:



Collaboration diagram for quantier::local_storage::ILocalStorageNotifier:



Signals

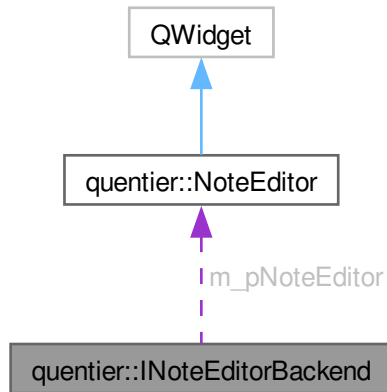
- void **userPut** (qevercloud::User user)
- void **userExpunged** (qevercloud::UserID userId)
- void **notebookPut** (qevercloud::Notebook notebook)
- void **notebookExpunged** (QString notebookLocalId)
- void **linkedNotebookPut** (qevercloud::LinkedNotebook linkedNotebook)
- void **linkedNotebookExpunged** (qevercloud::Guid linkedNotebookGuid)
- void **notePut** (qevercloud::Note note)
- void **noteUpdated** (qevercloud::Note note, ILocalStorage::UpdateNoteOptions options)
- void **noteExpunged** (QString noteLocalId)
- void **tagPut** (qevercloud::Tag tag)
- void **tagExpunged** (QString tagLocalId, QStringList expungedChildTagLocalIds)
- void **resourcePut** (qevercloud::Resource resource)
- void **resourceMetadataPut** (qevercloud::Resource resource)
- void **resourceExpunged** (QString resourceLocalId)
- void **savedSearchPut** (qevercloud::SavedSearch savedSearch)
- void **savedSearchExpunged** (QString savedSearchLocalId)

Protected Member Functions

- **ILocalStorageNotifier** (QObject *parent=nullptr)

5.30 quantier::INoteEditorBackend Class Reference

Collaboration diagram for quantier::INoteEditorBackend:



Public Types

- enum class **Rotation** { **Clockwise** , **Counterclockwise** }

Public Member Functions

- virtual void **initialize** (local_storage::ILocalStoragePtr localStorage, [SpellChecker](#) &spellChecker, const [Account](#) &account, QThread *backgroundJobsThread, enml::IDecryptedTextCachePtr decryptedTextCache)=0
- virtual QObject * **object** ()=0
- virtual QWidget * **widget** ()=0
- virtual void **setAccount** (const [Account](#) &account)=0
- virtual void **setUndoStack** (QUndoStack *pUndoStack)=0
- virtual void **setInitialPageHtml** (const QString &html)=0
- virtual void **setNoteNotFoundPageHtml** (const QString &html)=0
- virtual void **setNoteDeletedPageHtml** (const QString &html)=0
- virtual void **setNoteLoadingPageHtml** (const QString &html)=0
- virtual bool **isNoteLoaded** () const =0
- virtual qint64 **idleTime** () const =0
- virtual void **convertToNote** ()=0
- virtual void **saveNoteToLocalStorage** ()=0
- virtual void **setNoteTitle** (const QString ¬eTitle)=0
- virtual void **setTagIds** (const QStringList &tagLocalUids, const QStringList &tagGuids)=0
- virtual void **undo** ()=0
- virtual void **redo** ()=0
- virtual void **cut** ()=0
- virtual void **copy** ()=0
- virtual void **paste** ()=0
- virtual void **pasteUnformatted** ()=0

- virtual void **selectAll** ()=0
- virtual void **formatSelectionAsSourceCode** ()=0
- virtual void **fontMenu** ()=0
- virtual void **textBold** ()=0
- virtual void **textItalic** ()=0
- virtual void **textUnderline** ()=0
- virtual void **textStrikethrough** ()=0
- virtual void **textHighlight** ()=0
- virtual void **alignLeft** ()=0
- virtual void **alignCenter** ()=0
- virtual void **alignRight** ()=0
- virtual void **alignFull** ()=0
- virtual QString **selectedText** () const =0
- virtual bool **hasSelection** () const =0
- virtual void **findNext** (const QString &text, bool matchCase) const =0
- virtual void **findPrevious** (const QString &text, bool matchCase) const =0
- virtual void **replace** (const QString &textToReplace, const QString &replacementText, bool matchCase)=0
- virtual void **replaceAll** (const QString &textToReplace, const QString &replacementText, bool matchCase)=0
- virtual void **insertToDoCheckbox** ()=0
- virtual void **insertInAppNoteLink** (const QString &userId, const QString &shardId, const QString ¬eGuid, const QString &linkText)=0
- virtual void **setSpellcheck** (bool enabled)=0
- virtual bool **spellCheckEnabled** () const =0
- virtual void **setFont** (const QFont &font)=0
- virtual void **setFontHeight** (int height)=0
- virtual void **setFontColor** (const QColor &color)=0
- virtual void **setBackgroundColor** (const QColor &color)=0
- virtual QPalette **defaultPalette** () const =0
- virtual void **setDefaultPalette** (const QPalette &pal)=0
- virtual const QFont * **defaultFont** () const =0
- virtual void **setDefaultFont** (const QFont &font)=0
- virtual void **insertHorizontalLine** ()=0
- virtual void **increaseFontSize** ()=0
- virtual void **decreaseFontSize** ()=0
- virtual void **increaseIndentation** ()=0
- virtual void **decreaseIndentation** ()=0
- virtual void **insertBulletedList** ()=0
- virtual void **insertNumberedList** ()=0
- virtual void **insertTableDialog** ()=0
- virtual void **insertFixedWidthTable** (int rows, int columns, int widthInPixels)=0
- virtual void **insertRelativeWidthTable** (int rows, int columns, double relativeWidth)=0
- virtual void **insertTableRow** ()=0
- virtual void **insertTableColumn** ()=0
- virtual void **removeTableRow** ()=0
- virtual void **removeTableColumn** ()=0
- virtual void **addAttachmentDialog** ()=0
- virtual void **saveAttachmentDialog** (const QByteArray &resourceHash)=0
- virtual void **saveAttachmentUnderCursor** ()=0
- virtual void **openAttachment** (const QByteArray &resourceHash)=0
- virtual void **openAttachmentUnderCursor** ()=0
- virtual void **copyAttachment** (const QByteArray &resourceHash)=0
- virtual void **copyAttachmentUnderCursor** ()=0
- virtual void **removeAttachment** (const QByteArray &resourceHash)=0
- virtual void **removeAttachmentUnderCursor** ()=0
- virtual void **renameAttachment** (const QByteArray &resourceHash)=0

- virtual void **renameAttachmentUnderCursor** ()=0
- virtual void **rotatImageAttachment** (const QByteArray &resourceHash, Rotation rotationDirection)=0
- virtual void **rotatImageAttachmentUnderCursor** (Rotation rotationDirection)=0
- virtual void **encryptSelectedText** ()=0
- virtual void **decryptEncryptedTextUnderCursor** ()=0
- virtual void **decryptEncryptedText** (QString encryptedText, QString cipher, QString hint, QString enCrypt↔Index)=0
- virtual void **hideDecryptedTextUnderCursor** ()=0
- virtual void **hideDecryptedText** (QString encryptedText, QString decryptedText, QString cipher, QString hint, QString enDecryptedIndex)=0
- virtual void **editHyperlinkDialog** ()=0
- virtual void **copyHyperlink** ()=0
- virtual void **removeHyperlink** ()=0
- virtual void **onNoteLoadCancelled** ()=0
- virtual bool **print** (QPrinter &printer, [ErrorString](#) &errorDescription)=0
- virtual bool **exportToPdf** (const QString &absoluteFilePath, [ErrorString](#) &errorDescription)=0
- virtual bool **exportToEnex** (const QStringList &tagNames, QString &enex, [ErrorString](#) &errorDescription)=0
- virtual QString **currentNoteLocalId** () const =0
- virtual void **setCurrentNoteLocalId** (const QString ¬eLocalUid)=0
- virtual void **clear** ()=0
- virtual bool **isModified** () const =0
- virtual bool **isEditorPageModified** () const =0
- virtual void **setFocusToEditor** ()=0

Protected Member Functions

- **INoteEditorBackend** ([NoteEditor](#) *parent)

Protected Attributes

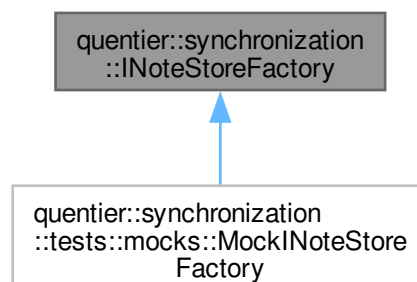
- [NoteEditor](#) * **m_pNoteEditor**

Friends

- QUENTIER_EXPORT QTextStream & **operator<<** (QTextStream &strm, Rotation rotation)
- QUENTIER_EXPORT QDebug & **operator<<** (QDebug &dbg, Rotation rotation)

5.31 quantier::synchronization::INoteStoreFactory Class Reference

Inheritance diagram for quantier::synchronization::INoteStoreFactory:

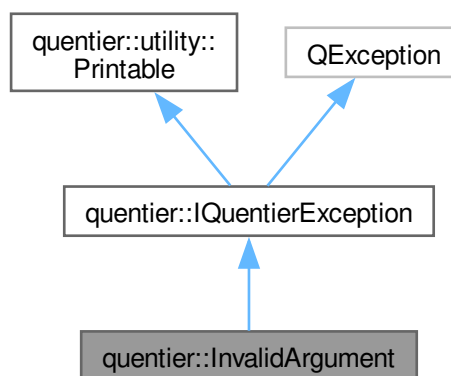


Public Member Functions

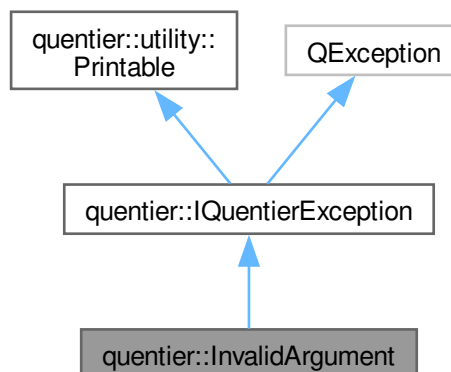
- virtual qevercloud::INoteStorePtr **createNoteStore** (QString noteStoreUrl={}, std::optional< qevercloud::↵ Guid > linkedNotebookGuid={}, qevercloud::IRequestContextPtr ctx={}, qevercloud::IRetryPolicyPtr retry↵ Policy={})=0

5.32 quantier::InvalidArgument Class Reference

Inheritance diagram for quantier::InvalidArgument:



Collaboration diagram for quantier::InvalidArgument:



Public Member Functions

- **InvalidArgument** ([ErrorString](#) message)
- InvalidArgument * **clone** () const override
- void **raise** () const override

Public Member Functions inherited from [quentier::IQuentierException](#)

- [ErrorString](#) **errorMessage** () const
- QString **localizedErrorMessage** () const
- QString **nonLocalizedErrorMessage** () const
- const char * **what** () const noexcept override
- QTextStream & [print](#) (QTextStream &strm) const override

Public Member Functions inherited from [quentier::utility::Printable](#)

- QString **toString** () const

Protected Member Functions

- QString [exceptionDisplayName](#) () const override

Protected Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** ([ErrorString](#) message)
- **IQuentierException** (const IQuentierException &other)
- IQuentierException & **operator=** (const IQuentierException &other)

5.32.1 Member Function Documentation

5.32.1.1 [exceptionDisplayName\(\)](#)

```
QString quentier::InvalidArgument::exceptionDisplayName () const [nodiscard], [override],
[protected], [virtual]
```

Implements [quentier::IQuentierException](#).

5.33 [quentier::ResourceRecognitionIndexItem::IObjectItem](#) Struct Reference

Public Member Functions

- virtual QString **objectType** () const =0
- virtual int **weight** () const =0

5.34 quantier::local_storage::IPatch Class Reference

The [IPatch](#) interface represents patches of the local storage. Each such patch somehow changes the layout of local storage persistence so that only compliant & corresponding versions of libquentier can be used to work with it.

```
#include <IPatch.h>
```

Public Member Functions

- virtual int [fromVersion](#) () const noexcept=0
- virtual int [toVersion](#) () const noexcept=0
- virtual QString [patchShortDescription](#) () const =0
- virtual QString [patchLongDescription](#) () const =0
- virtual QFuture< void > [backupLocalStorage](#) ()=0
- virtual QFuture< void > [restoreLocalStorageFromBackup](#) ()=0
- virtual QFuture< void > [removeLocalStorageBackup](#) ()=0
- virtual QFuture< void > [apply](#) ()=0

5.34.1 Detailed Description

The [IPatch](#) interface represents patches of the local storage. Each such patch somehow changes the layout of local storage persistence so that only compliant & corresponding versions of libquentier can be used to work with it.

5.34.2 Member Function Documentation

5.34.2.1 [apply\(\)](#)

```
virtual QFuture< void > quantier::local_storage::IPatch::apply () [nodiscard], [pure virtual]
```

Apply the patch to local storage

Returns

Future which can be awaited for patch application. Contains exception if patch application fails.

5.34.2.2 [backupLocalStorage\(\)](#)

```
virtual QFuture< void > quantier::local_storage::IPatch::backupLocalStorage () [nodiscard],  
[pure virtual]
```

Backup either the entire local storage or its parts affected by the particular patch, should be called before applying the patch (but can be skipped if not desired).

Returns

Future which can be awaited for the backup completion. Contains exception if backup fails.

5.34.2.3 fromVersion()

```
virtual int quentier::local_storage::IPatch::fromVersion () const [nodiscard], [pure virtual],  
[noexcept]
```

Returns

Version of local storage to which the patch needs to be applied

5.34.2.4 patchLongDescription()

```
virtual QString quentier::local_storage::IPatch::patchLongDescription () const [nodiscard],  
[pure virtual]
```

Returns

Long i.e. detailed description of the patch

5.34.2.5 patchShortDescription()

```
virtual QString quentier::local_storage::IPatch::patchShortDescription () const [nodiscard],  
[pure virtual]
```

Returns

Short description of the patch

5.34.2.6 removeLocalStorageBackup()

```
virtual QFuture< void > quentier::local_storage::IPatch::removeLocalStorageBackup () [nodiscard],  
[pure virtual]
```

Remove the previously made backup of local storage, presumably after successful application of the patch so the backup is no longer needed. It won't work if no backup was made before applying a patch, obviously.

Returns

Future which can be awaited for local storage backup removal. Contains exception if backup removal fails.

5.34.2.7 restoreLocalStorageFromBackup()

```
virtual QFuture< void > quentier::local_storage::IPatch::restoreLocalStorageFromBackup ()  
[nodiscard], [pure virtual]
```

Restore local storage from previously made backup, presumably after the failed attempt to apply a patch. Won't work if no backup was made before applying a patch, obviously.

Returns

Future which can be awaited for the backup restoration completion. Contains exception if backup restoration fails.

5.34.2.8 toVersion()

```
virtual int quantier::local_storage::IPatch::toVersion () const [nodiscard], [pure virtual],
[noexcept]
```

Returns

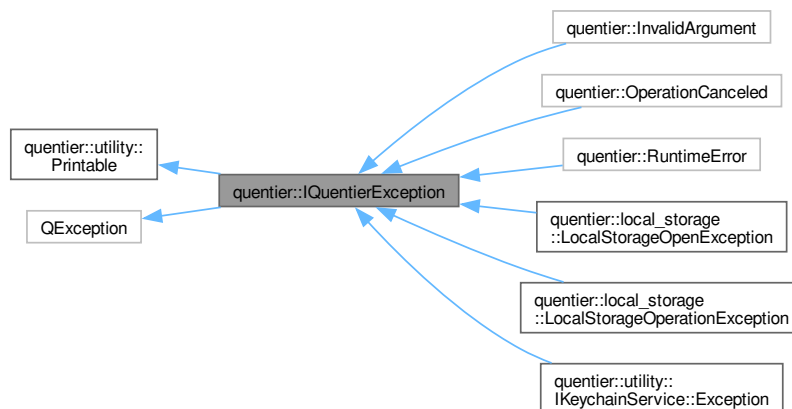
Version of local storage to which the patch would upgrade the local storage

5.35 quantier::IQuantierException Class Reference

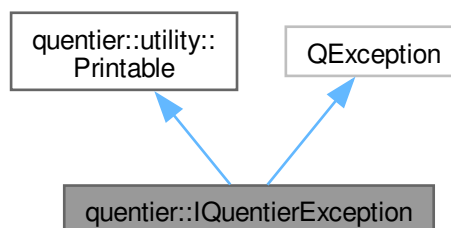
The [IQuantierException](#) class represents the interface for exceptions specific to libquantier and applications based on it.

```
#include <IQuantierException.h>
```

Inheritance diagram for quantier::IQuantierException:



Collaboration diagram for quantier::IQuantierException:



Public Member Functions

- [ErrorString](#) **errorMessage** () const
- QString **localizedErrorMessage** () const
- QString **nonLocalizedErrorMessage** () const
- const char * **what** () const noexcept override
- QTextStream & [print](#) (QTextStream &strm) const override

Public Member Functions inherited from [quentier::utility::Printable](#)

- QString **toString** () const

Protected Member Functions

- **IQuentierException** ([ErrorString](#) message)
- **IQuentierException** (const IQuentierException &other)
- IQuentierException & **operator=** (const IQuentierException &other)
- virtual QString **exceptionDisplayName** () const =0

5.35.1 Detailed Description

The [IQuentierException](#) class represents the interface for exceptions specific to libquentier and applications based on it.

In addition to standard exception features inherited from `std::exception`, [IQuentierException](#) based exceptions can provide both localized and non-localized error messages.

5.35.2 Member Function Documentation

5.35.2.1 `print()`

```
QTextStream & quentier::IQuentierException::print (  
    QTextStream & strm) const [override], [virtual]
```

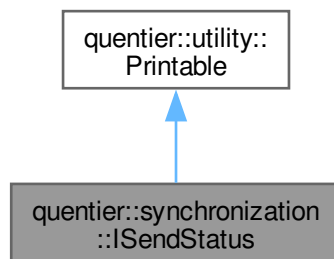
Implements [quentier::utility::Printable](#).

5.36 quantier::synchronization::ISendStatus Class Reference

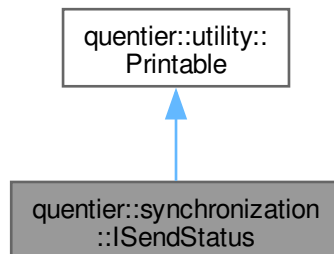
The [ISendStatus](#) interface represents the information about the attempt to send information either from user's own account or from some linked notebook to Evernote.

```
#include <ISendStatus.h>
```

Inheritance diagram for quantier::synchronization::ISendStatus:



Collaboration diagram for quantier::synchronization::ISendStatus:



Public Types

- using **QExceptionPtr** = std::shared_ptr<QException>
- using **NoteWithException** = std::pair<qevercloud::Note, QExceptionPtr>
- using [NotebookWithException](#)
- using [SavedSearchWithException](#)
- using **TagWithException** = std::pair<qevercloud::Tag, QExceptionPtr>

Public Member Functions

- virtual quint64 [totalAttemptedToSendNotes](#) () const =0
- virtual quint64 [totalAttemptedToSendNotebooks](#) () const =0
- virtual quint64 [totalAttemptedToSendSavedSearches](#) () const =0
- virtual quint64 [totalAttemptedToSendTags](#) () const =0
- virtual quint64 [totalSuccessfullySentNotes](#) () const =0
- virtual QList< NoteWithException > [failedToSendNotes](#) () const =0
- virtual quint64 [totalSuccessfullySentNotebooks](#) () const =0
- virtual QList< NotebookWithException > [failedToSendNotebooks](#) () const =0
- virtual quint64 [totalSuccessfullySentSavedSearches](#) () const =0
- virtual QList< SavedSearchWithException > [failedToSendSavedSearches](#) () const =0
- virtual quint64 [totalSuccessfullySentTags](#) () const =0
- virtual QList< TagWithException > [failedToSendTags](#) () const =0
- virtual StopSynchronizationError [stopSynchronizationError](#) () const =0
- virtual bool [needToRepeatIncrementalSync](#) () const =0

Public Member Functions inherited from [quentier::utility::Printable](#)

- virtual QTextStream & [print](#) (QTextStream &strm) const =0
- QString [toString](#) () const

5.36.1 Detailed Description

The [ISendStatus](#) interface represents the information about the attempt to send information either from user's own account or from some linked notebook to Evernote.

5.36.2 Member Typedef Documentation

5.36.2.1 NotebookWithException

```
using quentier::synchronization::ISendStatus::NotebookWithException
```

Initial value:

```
std::pair<qevercloud::Notebook, QExceptionPtr>
```

5.36.2.2 SavedSearchWithException

```
using quentier::synchronization::ISendStatus::SavedSearchWithException
```

Initial value:

```
std::pair<qevercloud::SavedSearch, QExceptionPtr>
```

5.36.3 Member Function Documentation

5.36.3.1 failedToSendNotebooks()

```
virtual QList< NotebookWithException > quantier::synchronization::ISendStatus::failedToSendNotebooks () const [nodiscard], [pure virtual]
```

Returns

list with notebooks and exceptions representing failures to send these notebooks to Evernote

5.36.3.2 failedToSendNotes()

```
virtual QList< NoteWithException > quantier::synchronization::ISendStatus::failedToSendNotes () const [nodiscard], [pure virtual]
```

Returns

list with notes and exceptions representing failures to send these notes to Evernote

5.36.3.3 failedToSendSavedSearches()

```
virtual QList< SavedSearchWithException > quantier::synchronization::ISendStatus::failedToSendSavedSearches () const [nodiscard], [pure virtual]
```

Returns

list with saved searches and exceptions representing failures to send these saved searches to Evernote

5.36.3.4 failedToSendTags()

```
virtual QList< TagWithException > quantier::synchronization::ISendStatus::failedToSendTags () const [nodiscard], [pure virtual]
```

Returns

list with tags and exceptions representing failures to send these tags to Evernote

5.36.3.5 needToRepeatIncrementalSync()

```
virtual bool quantier::synchronization::ISendStatus::needToRepeatIncrementalSync () const [nodiscard], [pure virtual]
```

If during the send step of synchronization it was found out that Evernote service's state of account has been updated since the last download step, returns true meaning that incremental download step should be repeated. Otherwise returns false.

5.36.3.6 stopSynchronizationError()

```
virtual StopSynchronizationError quentier::synchronization::ISendStatus::stopSynchronization←  
Error () const [nodiscard], [pure virtual]
```

Returns

error which might have occurred during sending the data to Evernote which has prevented further attempts to send anything to Evernote or std::monostate if no such error has occurred

5.36.3.7 totalAttemptedToSendNotebooks()

```
virtual quint64 quentier::synchronization::ISendStatus::totalAttemptedToSendNotebooks () const  
[nodiscard], [pure virtual]
```

Returns

total number of notebooks attempted to be sent to Evernote

5.36.3.8 totalAttemptedToSendNotes()

```
virtual quint64 quentier::synchronization::ISendStatus::totalAttemptedToSendNotes () const  
[nodiscard], [pure virtual]
```

Returns

total number of notes attempted to be sent to Evernote

5.36.3.9 totalAttemptedToSendSavedSearches()

```
virtual quint64 quentier::synchronization::ISendStatus::totalAttemptedToSendSavedSearches ()  
const [nodiscard], [pure virtual]
```

Returns

total number of saved searches attempted to be sent to Evernote

5.36.3.10 totalAttemptedToSendTags()

```
virtual quint64 quentier::synchronization::ISendStatus::totalAttemptedToSendTags () const  
[nodiscard], [pure virtual]
```

Returns

total number of tags attempted to be sent to Evernote

5.36.3.11 `totalSuccessfullySentNotebooks()`

```
virtual quint64 quentier::synchronization::ISendStatus::totalSuccessfullySentNotebooks ()  
const [nodiscard], [pure virtual]
```

Returns

number of notebooks which were successfully sent to Evernote

5.36.3.12 `totalSuccessfullySentNotes()`

```
virtual quint64 quentier::synchronization::ISendStatus::totalSuccessfullySentNotes () const  
[nodiscard], [pure virtual]
```

Returns

number of notes which were successfully sent to Evernote

5.36.3.13 `totalSuccessfullySentSavedSearches()`

```
virtual quint64 quentier::synchronization::ISendStatus::totalSuccessfullySentSavedSearches ()  
const [nodiscard], [pure virtual]
```

Returns

number of saved searches which were successfully sent to Evernote

5.36.3.14 `totalSuccessfullySentTags()`

```
virtual quint64 quentier::synchronization::ISendStatus::totalSuccessfullySentTags () const  
[nodiscard], [pure virtual]
```

Returns

number of tags which were successfully sent to Evernote

5.37 `quentier::ResourceRecognitionIndexItem::IShapeItem` Struct Reference

Public Member Functions

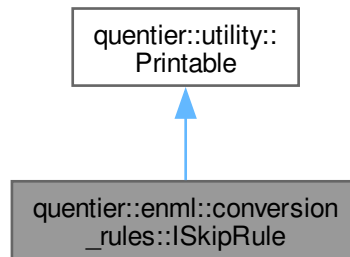
- virtual `QString` **shape** () const =0
- virtual `int` **weight** () const =0

5.38 quantier::enml::conversion_rules::ISkipRule Class Reference

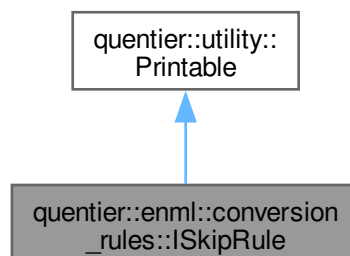
The [ISkipRule](#) interface describes a conversion rule with regards to which some ENML/HTML element/attribute should be skipped during the conversion.

```
#include <ISkipRule.h>
```

Inheritance diagram for quantier::enml::conversion_rules::ISkipRule:



Collaboration diagram for quantier::enml::conversion_rules::ISkipRule:



Public Types

- enum class [Target](#) { [Element](#) , [AttibuteName](#) , [AttributeValue](#) }

Public Member Functions

- virtual [Target](#) [target](#) () const =0
- virtual QString [value](#) () const =0
- virtual MatchMode [matchMode](#) () const =0
- virtual bool [includeContents](#) () const =0
- virtual Qt::CaseSensitivity [caseSensitivity](#) () const =0
- QTextStream & [print](#) (QTextStream &strm) const override

Public Member Functions inherited from [quantier::utility::Printable](#)

- QString **toString** () const

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, [Target target](#))
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, [Target target](#))

5.38.1 Detailed Description

The [ISkipRule](#) interface describes a conversion rule with regards to which some ENML/HTML element/attribute should be skipped during the conversion.

ENML format prohibits the use of certain HTML tags and attributes. This interface facilitates skipping these tags and attributes in the process of conversion from HTML to ENML

5.38.2 Member Enumeration Documentation

5.38.2.1 Target

```
enum class quantier::enml::conversion\_rules::ISkipRule::Target [strong]
```

[Target](#) to be affected by the skip rule

Enumerator

Element	HTML element
AttributeName	HTML attribute with specified name
AttributeValue	HTML attribute with specified value

5.38.3 Member Function Documentation

5.38.3.1 caseSensitivity()

```
virtual Qt::CaseSensitivity quantier::enml::conversion\_rules::ISkipRule::caseSensitivity ()  
const [nodiscard], [pure virtual]
```

Case sensitivity for target name/value check

5.38.3.2 includeContents()

```
virtual bool quantier::enml::conversion\_rules::ISkipRule::includeContents () const [nodiscard],  
[pure virtual]
```

Specifies whether the element contents should be included without the element itself if it needs to be skipped or not

5.38.3.3 `matchMode()`

```
virtual MatchMode quentier::enml::conversion_rules::ISkipRule::matchMode () const [nodiscard],  
[pure virtual]
```

Match mode for name or value of the target

5.38.3.4 `print()`

```
QTextStream & quentier::enml::conversion_rules::ISkipRule::print (  
    QTextStream & strm) const [override], [virtual]
```

Implements [quentier::utility::Printable](#).

5.38.3.5 `target()`

```
virtual Target quentier::enml::conversion_rules::ISkipRule::target () const [nodiscard], [pure  
virtual]
```

[Target](#) to be affected by the skip rule

5.38.3.6 `value()`

```
virtual QString quentier::enml::conversion_rules::ISkipRule::value () const [nodiscard], [pure  
virtual]
```

Name or value of the target

5.39 `quentier::enml::conversion_rules::ISkipRuleBuilder` Class Reference

Public Member Functions

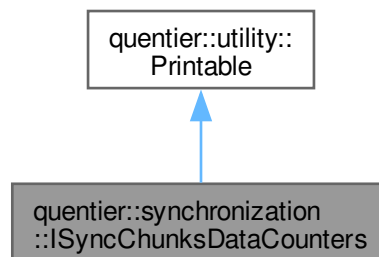
- virtual [ISkipRuleBuilder](#) & **setTarget** ([ISkipRule::Target](#) target)=0
- virtual [ISkipRuleBuilder](#) & **setValue** (QString value)=0
- virtual [ISkipRuleBuilder](#) & **setMatchMode** (MatchMode matchMode)=0
- virtual [ISkipRuleBuilder](#) & **setIncludeContents** (bool includeContents)=0
- virtual [ISkipRuleBuilder](#) & **setCaseSensitivity** (Qt::CaseSensitivity caseSensitivity)=0
- virtual [ISkipRulePtr](#) **build** ()=0

5.40 quantier::synchronization::ISyncChunksDataCounters Struct Reference

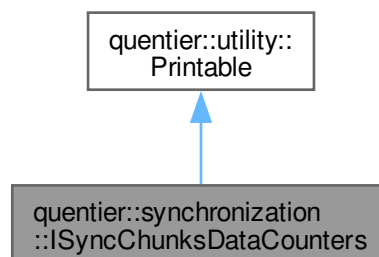
The [ISyncChunksDataCounters](#) interface provides integer counters representing the current progress on processing the data from downloaded sync chunks.

```
#include <ISyncChunksDataCounters.h>
```

Inheritance diagram for quantier::synchronization::ISyncChunksDataCounters:



Collaboration diagram for quantier::synchronization::ISyncChunksDataCounters:



Public Member Functions

- virtual quint64 [totalSavedSearches](#) () const noexcept=0
- virtual quint64 [totalExpungedSavedSearches](#) () const noexcept=0
- virtual quint64 [addedSavedSearches](#) () const noexcept=0
- virtual quint64 [updatedSavedSearches](#) () const noexcept=0
- virtual quint64 [expungedSavedSearches](#) () const noexcept=0
- virtual quint64 [totalTags](#) () const noexcept=0
- virtual quint64 [totalExpungedTags](#) () const noexcept=0

- virtual quint64 [addedTags](#) () const noexcept=0
- virtual quint64 [updatedTags](#) () const noexcept=0
- virtual quint64 [expungedTags](#) () const noexcept=0
- virtual quint64 [totalLinkedNotebooks](#) () const noexcept=0
- virtual quint64 [totalExpungedLinkedNotebooks](#) () const noexcept=0
- virtual quint64 [addedLinkedNotebooks](#) () const noexcept=0
- virtual quint64 [updatedLinkedNotebooks](#) () const noexcept=0
- virtual quint64 [expungedLinkedNotebooks](#) () const noexcept=0
- virtual quint64 [totalNotebooks](#) () const noexcept=0
- virtual quint64 [totalExpungedNotebooks](#) () const noexcept=0
- virtual quint64 [addedNotebooks](#) () const noexcept=0
- virtual quint64 [updatedNotebooks](#) () const noexcept=0
- virtual quint64 [expungedNotebooks](#) () const noexcept=0

Public Member Functions inherited from [quentier::utility::Printable](#)

- virtual QTextStream & **print** (QTextStream &strm) const =0
- QString **toString** () const

5.40.1 Detailed Description

The [ISyncChunksDataCounters](#) interface provides integer counters representing the current progress on processing the data from downloaded sync chunks.

5.40.2 Member Function Documentation

5.40.2.1 [addedLinkedNotebooks\(\)](#)

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::addedLinkedNotebooks ()
const [nodiscard], [pure virtual], [noexcept]
```

Number of linked notebooks from sync chunks added to the local storage so far

5.40.2.2 [addedNotebooks\(\)](#)

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::addedNotebooks () const
[nnodiscard], [pure virtual], [noexcept]
```

Number of notebooks from sync chunks added to the local storage so far

5.40.2.3 [addedSavedSearches\(\)](#)

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::addedSavedSearches ()
const [nodiscard], [pure virtual], [noexcept]
```

Number of saved searches from sync chunks added to the local storage so far

5.40.2.4 addedTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::addedTags () const [nodiscard],  
[pure virtual], [noexcept]
```

Number of tags from sync chunks added to the local storage so far

5.40.2.5 expungedLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::expungedLinkedNotebooks ()  
const [nodiscard], [pure virtual], [noexcept]
```

Number of linked notebooks from sync chunks expunged from the local storage so far

5.40.2.6 expungedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::expungedNotebooks () const  
[nodiscard], [pure virtual], [noexcept]
```

Number of notebooks from sync chunks expunged from the local storage so far

5.40.2.7 expungedSavedSearches()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::expungedSavedSearches ()  
const [nodiscard], [pure virtual], [noexcept]
```

Number of saved searches from sync chunks expunged from the local storage so far

5.40.2.8 expungedTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::expungedTags () const  
[nodiscard], [pure virtual], [noexcept]
```

Number of tags from sync chunks expunged from the local storage so far

5.40.2.9 totalExpungedLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalExpungedLinked←  
Notebooks () const [nodiscard], [pure virtual], [noexcept]
```

Total number of expunged saved searches in downloaded sync chunks

5.40.2.10 totalExpungedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalExpungedNotebooks ()  
const [nodiscard], [pure virtual], [noexcept]
```

Total number of expunged notebooks in downloaded sync chunks

5.40.2.11 totalExpungedSavedSearches()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalExpungedSavedSearches  
( ) const [nodiscard], [pure virtual], [noexcept]
```

Total number of expunged saved searches in downloaded sync chunks

5.40.2.12 totalExpungedTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalExpungedTags ( ) const  
[nodiscard], [pure virtual], [noexcept]
```

Total number of expunged tags in downloaded sync chunks

5.40.2.13 totalLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalLinkedNotebooks ( )  
const [nodiscard], [pure virtual], [noexcept]
```

Total number of new or updated linked notebooks in downloaded sync chunks

5.40.2.14 totalNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalNotebooks ( ) const  
[nodiscard], [pure virtual], [noexcept]
```

Total number of new or updated notebooks in downloaded sync chunks

5.40.2.15 totalSavedSearches()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalSavedSearches ( )  
const [nodiscard], [pure virtual], [noexcept]
```

Total number of new or updated saved searches in downloaded sync chunks

5.40.2.16 totalTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalTags ( ) const [nodiscard],  
[pure virtual], [noexcept]
```

Total number of new or updated tags in downloaded sync chunks

5.40.2.17 updatedLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::updatedLinkedNotebooks ( )  
const [nodiscard], [pure virtual], [noexcept]
```

Number of linked notebooks from sync chunks updated in the local storage so far

5.40.2.18 updatedNotebooks()

```
virtual quint64 quantier::synchronization::ISyncChunksDataCounters::updatedNotebooks () const
[nodiscard], [pure virtual], [noexcept]
```

Number of notebooks from sync chunks updated in the local storage so far

5.40.2.19 updatedSavedSearches()

```
virtual quint64 quantier::synchronization::ISyncChunksDataCounters::updatedSavedSearches ()
const [nodiscard], [pure virtual], [noexcept]
```

Number of saved searches from sync chunks updated in the local storage so far

5.40.2.20 updatedTags()

```
virtual quint64 quantier::synchronization::ISyncChunksDataCounters::updatedTags () const [nodiscard],
[pure virtual], [noexcept]
```

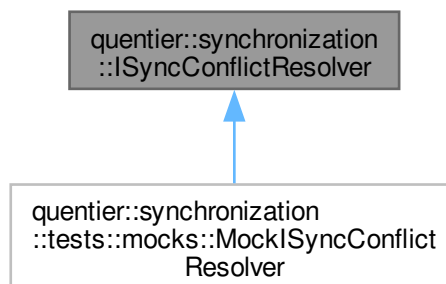
Number of tags from sync chunks updated in the local storage so far

5.41 quantier::synchronization::ISyncConflictResolver Class Reference

The [ISyncConflictResolver](#) interface provides methods used to resolve conflicts between local and remote versions of the same data item.

```
#include <ISyncConflictResolver.h>
```

Inheritance diagram for quantier::synchronization::ISyncConflictResolver:

**Classes**

- struct [ConflictResolution](#)

The [ConflictResolution](#) struct is a namespace inside which several other structs determining actual conflict resolutions.

Public Types

- using [NotebookConflictResolution](#)
- using [NoteConflictResolution](#)
- using [SavedSearchConflictResolution](#)
- using [TagConflictResolution](#)

Public Member Functions

- virtual QFuture< NotebookConflictResolution > **resolveNotebookConflict** (qevercloud::Notebook theirs, qevercloud::Notebook mine)=0
- virtual QFuture< NoteConflictResolution > **resolveNoteConflict** (qevercloud::Note theirs, qevercloud::Note mine)=0
- virtual QFuture< SavedSearchConflictResolution > **resolveSavedSearchConflict** (qevercloud::SavedSearch theirs, qevercloud::SavedSearch mine)=0
- virtual QFuture< TagConflictResolution > **resolveTagConflict** (qevercloud::Tag theirs, qevercloud::Tag mine)=0

5.41.1 Detailed Description

The [ISyncConflictResolver](#) interface provides methods used to resolve conflicts between local and remote versions of the same data item.

5.41.2 Member Typedef Documentation

5.41.2.1 NotebookConflictResolution

```
using quantier::synchronization::ISyncConflictResolver::NotebookConflictResolution
```

Initial value:

```
std::variant<
    ConflictResolution::UseTheirs, ConflictResolution::UseMine,
    ConflictResolution::IgnoreMine,
    ConflictResolution::MoveMine<qevercloud::Notebook>>
```

5.41.2.2 NoteConflictResolution

```
using quantier::synchronization::ISyncConflictResolver::NoteConflictResolution
```

Initial value:

```
std::variant<
    ConflictResolution::UseTheirs, ConflictResolution::UseMine,
    ConflictResolution::IgnoreMine,
    ConflictResolution::MoveMine<qevercloud::Note>>
```

5.41.2.3 SavedSearchConflictResolution

```
using quantier::synchronization::ISyncConflictResolver::SavedSearchConflictResolution
```

Initial value:

```
std::variant<
    ConflictResolution::UseTheirs, ConflictResolution::UseMine,
    ConflictResolution::IgnoreMine,
    ConflictResolution::MoveMine<qevercloud::SavedSearch>>
```


5.41.2.4 TagConflictResolution

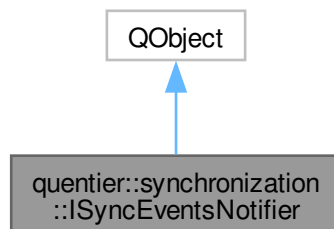
```
using quantier::synchronization::ISyncConflictResolver::TagConflictResolution
```

Initial value:

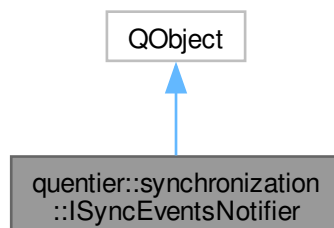
```
std::variant<  
    ConflictResolution::IgnoreMine, ConflictResolution::UseTheirs,  
    ConflictResolution::UseMine,  
    ConflictResolution::MoveMine<qevercloud::Tag>>
```

5.42 quantier::synchronization::ISyncEventsNotifier Class Reference

Inheritance diagram for quantier::synchronization::ISyncEventsNotifier:



Collaboration diagram for quantier::synchronization::ISyncEventsNotifier:



Signals

- void [syncChunksDownloadProgress](#) (qint32 highestDownloadedUsn, qint32 highestServerUsn, qint32 last← PreviousUsn)
- void [syncChunksDownloaded](#) ()
- void [syncChunksDataProcessingProgress](#) (ISyncChunksDataCountersPtr counters)

- void [startLinkedNotebooksDataDownloading](#) (const QList< [qevercloud::LinkedNotebook](#) > &linkedNotebooks)
- void [linkedNotebookSyncChunksDownloadProgress](#) (qint32 highestDownloadedUsn, qint32 highestServerUsn, qint32 lastPreviousUsn, const [qevercloud::LinkedNotebook](#) &linkedNotebook)
- void [linkedNotebookSyncChunksDownloaded](#) (const [qevercloud::LinkedNotebook](#) &linkedNotebook)
- void [linkedNotebookSyncChunksDataProcessingProgress](#) (ISyncChunksDataCountersPtr counters, const [qevercloud::LinkedNotebook](#) &linkedNotebook)
- void [notesDownloadProgress](#) (quint32 notesDownloaded, quint32 totalNotesToDownload)
- void [linkedNotebookNotesDownloadProgress](#) (quint32 notesDownloaded, quint32 totalNotesToDownload, const [qevercloud::LinkedNotebook](#) &linkedNotebook)
- void [resourcesDownloadProgress](#) (quint32 resourcesDownloaded, quint32 totalResourcesToDownload)
- void [linkedNotebookResourcesDownloadProgress](#) (quint32 resourcesDownloaded, quint32 totalResourcesToDownload, const [qevercloud::LinkedNotebook](#) &linkedNotebook)
- void [downloadFinished](#) (bool dataDownloaded)
- void [userOwnSendStatusUpdate](#) (ISendStatusPtr sendStatus)
- void [linkedNotebookSendStatusUpdate](#) (const [qevercloud::Guid](#) &linkedNotebookGuid, ISendStatusPtr sendStatus)

Protected Member Functions

- [ISyncEventsNotifier](#) (QObject *parent=nullptr)

5.42.1 Member Function Documentation

5.42.1.1 downloadFinished

```
void qentier::synchronization::ISyncEventsNotifier::downloadFinished (
    bool dataDownloaded) [signal]
```

This signal is emitted when the initial download step is finished.

Parameters

<i>dataDownloaded</i>	True if some data was actually downloaded during this step i.e. there were some updates on Evernote servers compared to local state, false otherwise.
-----------------------	---

5.42.1.2 linkedNotebookNotesDownloadProgress

```
void qentier::synchronization::ISyncEventsNotifier::linkedNotebookNotesDownloadProgress (
    quint32 notesDownloaded,
    quint32 totalNotesToDownload,
    const qevercloud::LinkedNotebook & linkedNotebook) [signal]
```

This signal is emitted on each successful download of full note data from some linked notebook.

Parameters

<i>notesDownloaded</i>	The number of notes downloaded by the moment
<i>totalNotesToDownload</i>	The total number of notes that need to be downloaded
<i>linkedNotebook</i>	The linked notebook which notes download progress is being reported

5.42.1.3 linkedNotebookResourcesDownloadProgress

```
void quantier::synchronization::ISyncEventsNotifier::linkedNotebookResourcesDownloadProgress (
    quint32 resourcesDownloaded,
    quint32 totalResourcesToDownload,
    const qevercloud::LinkedNotebook & linkedNotebook) [signal]
```

This signal is emitted on each successful download of full resource data from linked notebooks during the incremental sync (as individual resources are downloaded along with their notes during full sync).

Parameters

<i>resourcesDownloaded</i>	The number of resources downloaded by the moment
<i>totalResourcesToDownload</i>	The total number of resources that need to be downloaded
<i>linkedNotebook</i>	The linked notebook which resources download progress is being reported

5.42.1.4 linkedNotebookSendStatusUpdate

```
void quantier::synchronization::ISyncEventsNotifier::linkedNotebookSendStatusUpdate (
    const qevercloud::Guid & linkedNotebookGuid,
    ISendStatusPtr sendStatus) [signal]
```

This signal is emitted on each successful or unsuccessful attempt to send some new or locally modified data item from some linked notebook to Evernote.

Parameters

<i>linkedNotebookGuid</i>	Guid of the linked notebook for which the send status was updated
<i>sendStatus</i>	The updated send status

5.42.1.5 linkedNotebookSyncChunksDataProcessingProgress

```
void quantier::synchronization::ISyncEventsNotifier::linkedNotebookSyncChunksDataProcessingProgress (
    ISyncChunksDataCountersPtr counters,
    const qevercloud::LinkedNotebook & linkedNotebook) [signal]
```

This signal is emitted during some linked notebook's downloaded sync chunks contents processing and denotes the progress on that step.

Parameters

<i>counters</i>	Updated sync chunks data counters
<i>linkedNotebook</i>	The linked notebook which sync chunks data processing progress is being reported

5.42.1.6 linkedNotebookSyncChunksDownloaded

```
void quantier::synchronization::ISyncEventsNotifier::linkedNotebookSyncChunksDownloaded (
    const qevercloud::LinkedNotebook & linkedNotebook) [signal]
```

This signal is emitted when the sync chunks for data from some linked notebook are downloaded during "remote to local" synchronization step

Parameters

<i>linkedNotebook</i>	The linked notebook which sync chunks were downloaded
-----------------------	---

5.42.1.7 linkedNotebookSyncChunksDownloadProgress

```
void quantier::synchronization::ISyncEventsNotifier::linkedNotebookSyncChunksDownloadProgress (
    quint32 highestDownloadedUsn,
    quint32 highestServerUsn,
    quint32 lastPreviousUsn,
    const qevercloud::LinkedNotebook & linkedNotebook) [signal]
```

This signal is emitted during linked notebooks sync chunks downloading and denotes the progress of that step, individually for each linked notebook. The percentage of completeness can be computed roughly as $(\text{highestDownloadedUsn} - \text{lastPreviousUsn}) / (\text{highestServerUsn} - \text{lastPreviousUsn}) * 100\%$.

Parameters

<i>highestDownloadedUsn</i>	The highest update sequence number within data items from linked notebook sync chunks downloaded so far
<i>highestServerUsn</i>	The current highest update sequence number within the linked notebook
<i>lastPreviousUsn</i>	The last update sequence number from previous sync of the given linked notebook; if current sync is the first one, this value is zero
<i>linkedNotebook</i>	The linked notebook which sync chunks download progress is reported

5.42.1.8 notesDownloadProgress

```
void quantier::synchronization::ISyncEventsNotifier::notesDownloadProgress (
    quint32 notesDownloaded,
    quint32 totalNotesToDownload) [signal]
```

This signal is emitted on each successful download of full note data from user's own account.

Parameters

<i>notesDownloaded</i>	The number of notes downloaded by the moment
<i>totalNotesToDownload</i>	The total number of notes that need to be downloaded

5.42.1.9 resourcesDownloadProgress

```
void quantier::synchronization::ISyncEventsNotifier::resourcesDownloadProgress (
    quint32 resourcesDownloaded,
    quint32 totalResourcesToDownload) [signal]
```

This signal is emitted on each successful download of full resource data from user's own account during the incremental sync (as individual resources are downloaded along with their notes during full sync).

Parameters

<i>resourcesDownloaded</i>	The number of resources downloaded by the moment
<i>totalResourcesToDownload</i>	The total number of resources that need to be downloaded

5.42.1.10 startLinkedNotebooksDataDownloading

```
void quantier::synchronization::ISyncEventsNotifier::startLinkedNotebooksDataDownloading (
    const QList< qevercloud::LinkedNotebook > & linkedNotebooks) [signal]
```

This signal is emitted before the downloading of data corresponding to linked notebooks starts.

Parameters

<i>linkedNotebooks</i>	Linked notebooks the data from which will start being downloaded after the execution of this callback
------------------------	---

5.42.1.11 syncChunksDataProcessingProgress

```
void quantier::synchronization::ISyncEventsNotifier::syncChunksDataProcessingProgress (
    ISyncChunksDataCountersPtr counters) [signal]
```

This signal is emitted during user own account's downloaded sync chunks contents processing and denotes the progress on that step.

5.42.1.12 syncChunksDownloaded

```
void quantier::synchronization::ISyncEventsNotifier::syncChunksDownloaded () [signal]
```

This signal is emitted when the sync chunks for data from user's own account are downloaded during the download synchronization step.

5.42.1.13 syncChunksDownloadProgress

```
void quentier::synchronization::ISyncEventsNotifier::syncChunksDownloadProgress (
    qint32 highestDownloadedUsn,
    qint32 highestServerUsn,
    qint32 lastPreviousUsn) [signal]
```

This signal is emitted during user own account's sync chunks downloading and denotes the progress of that step. The percentage of completeness can be computed roughly as $(\text{highestDownloadedUsn} - \text{lastPreviousUsn}) / (\text{highestServerUsn} - \text{lastPreviousUsn}) * 100\%$.

Parameters

<i>highestDownloadedUsn</i>	The highest update sequence number within data items from sync chunks downloaded so far
<i>highestServerUsn</i>	The current highest update sequence number within the account
<i>lastPreviousUsn</i>	The last update sequence number from previous sync; if current sync is the first one, this value is zero

5.42.1.14 userOwnSendStatusUpdate

```
void quentier::synchronization::ISyncEventsNotifier::userOwnSendStatusUpdate (
    ISendStatusPtr sendStatus) [signal]
```

This signal is emitted on each successful or unsuccessful attempt to send some new or locally modified data item from user's own account to Evernote.

Parameters

<i>sendStatus</i>	The updated send status
-------------------	-------------------------

5.43 quentier::synchronization::ISynchronizer Class Reference

Public Types

- using [SyncResult](#)

Public Member Functions

- virtual QFuture< std::pair< [Account](#), IAuthenticationInfoPtr > > **authenticateNewAccount** ()=0
- virtual QFuture< IAuthenticationInfoPtr > **authenticateAccount** ([Account](#) account)=0
- virtual SyncResult **synchronizeAccount** ([Account](#) account, local_storage::ILocalStoragePtr localStorage, utility::cancelers::ICancelerPtr canceler, ISyncOptionsPtr options=nullptr, ISyncConflictResolverPtr syncConflictResolver=nullptr)=0
- virtual void **revokeAuthentication** (qevercloud::UserID userId)=0

5.43.1 Member Typedef Documentation

5.43.1.1 SyncResult

```
using quantier::synchronization::ISynchronizer::SyncResult
```

Initial value:

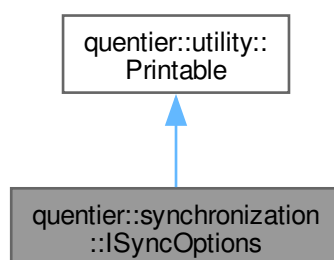
```
std::pair<QFuture<ISyncResultPtr>, ISyncEventsNotifier *>
```

5.44 quantier::synchronization::ISyncOptions Class Reference

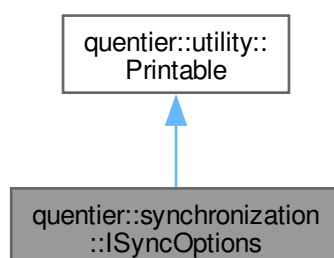
Options for synchronization process.

```
#include <ISyncOptions.h>
```

Inheritance diagram for quantier::synchronization::ISyncOptions:



Collaboration diagram for quantier::synchronization::ISyncOptions:



Public Member Functions

- virtual bool [downloadNoteThumbnails](#) () const =0
- virtual std::optional< QDir > [inkNoteImagesStorageDir](#) () const =0
- virtual qevercloud::IRequestContextPtr [requestContext](#) () const =0
- virtual qevercloud::IRetryPolicyPtr [retryPolicy](#) () const =0
- virtual std::optional< quint32 > [maxConcurrentNoteDownloads](#) () const =0
- virtual std::optional< quint32 > [maxConcurrentResourceDownloads](#) () const =0

Public Member Functions inherited from [quentier::utility::Printable](#)

- virtual QTextStream & [print](#) (QTextStream &strm) const =0
- QString [toString](#) () const

5.44.1 Detailed Description

Options for synchronization process.

5.44.2 Member Function Documentation

5.44.2.1 [downloadNoteThumbnails\(\)](#)

```
virtual bool quentier::synchronization::ISyncOptions::downloadNoteThumbnails () const [nodiscard],
[pure virtual]
```

Flag to enable or disable downloading of note thumbnails during the sync. Note thumbnails are stored inside the local storage along with other note data.

5.44.2.2 [inkNoteImagesStorageDir\(\)](#)

```
virtual std::optional< QDir > quentier::synchronization::ISyncOptions::inkNoteImagesStorageDir
() const [nodiscard], [pure virtual]
```

Directory to store the downloaded ink note images. If this method returns std::nullopt, ink note images would not be downloaded during the sync.

Ink notes images data is stored inside note's resources but the format of the data is not documented, which makes it quite hard to implement note editor able to fully handle ink notes. An easier option is to visualize a static image corresponding to the last revision of the ink note. Such images need to be downloaded separately during the sync if they are required.

Ink note images are stored right in this directory without any subdirectories, file names correspond to pattern <resource guid>.png.

5.44.2.3 [maxConcurrentNoteDownloads\(\)](#)

```
virtual std::optional< quint32 > quentier::synchronization::ISyncOptions::maxConcurrentNote↵
Downloads () const [nodiscard], [pure virtual]
```

Maximal number of concurrent note downloads. Allowing unlimited concurrent note downloads can lead to errors due to reaching the platform's limit on the number of open files. If std::nullopt is returned, the default limit is used.

5.44.2.4 `maxConcurrentResourceDownloads()`

```
virtual std::optional< quint32 > quentier::synchronization::ISyncOptions::maxConcurrentResourceDownloads () const [nodiscard], [pure virtual]
```

Maximal number of concurrent resource downloads. Allowing unlimited concurrent resource downloads can lead to errors due to reaching the platform's limit on the number of open files. If `std::nullopt` is returned, the default limit is used.

5.44.2.5 `requestContext()`

```
virtual qevercloud::IRequestContextPtr quentier::synchronization::ISyncOptions::requestContext () const [nodiscard], [pure virtual]
```

Request context with settings which should be used during the sync. If `nullptr` then request context with default settings would be used.

5.44.2.6 `retryPolicy()`

```
virtual qevercloud::IRetryPolicyPtr quentier::synchronization::ISyncOptions::retryPolicy () const [nodiscard], [pure virtual]
```

Retry policy which should be used during the sync. If `nullptr` then the default retry policy would be used.

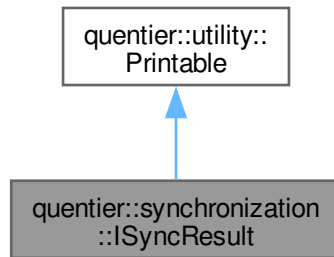
5.45 `quentier::synchronization::ISyncOptionsBuilder` Class Reference

Public Member Functions

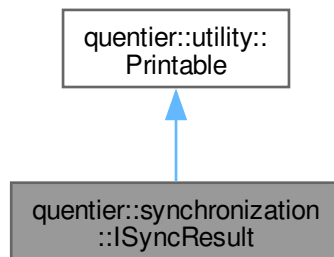
- virtual `ISyncOptionsBuilder` & **setDownloadNoteThumbnails** (bool value)=0
- virtual `ISyncOptionsBuilder` & **setInkNotelmagesStorageDir** (std::optional< QDir > dir)=0
- virtual `ISyncOptionsBuilder` & **setRequestContext** (qevercloud::IRequestContextPtr ctx)=0
- virtual `ISyncOptionsBuilder` & **setRetryPolicy** (qevercloud::IRetryPolicyPtr retryPolicy)=0
- virtual `ISyncOptionsBuilder` & **setMaxConcurrentNoteDownloads** (std::optional< quint32 > maxConcurrentNoteDownloads)=0
- virtual `ISyncOptionsBuilder` & **setMaxConcurrentResourceDownloads** (std::optional< quint32 > maxConcurrentResourceDownloads)=0
- virtual `ISyncOptionsPtr` **build** ()=0

5.46 quantier::synchronization::ISyncResult Class Reference

Inheritance diagram for quantier::synchronization::ISyncResult:



Collaboration diagram for quantier::synchronization::ISyncResult:



Public Member Functions

- virtual ISyncStatePtr **syncState** () const =0
- virtual ISyncChunksDataCountersPtr **userAccountSyncChunksDataCounters** () const =0
- virtual QHash< qevercloud::Guid, ISyncChunksDataCountersPtr > **linkedNotebookSyncChunksDataCounters** () const =0
- virtual bool **userAccountSyncChunksDownloaded** () const =0
- virtual QSet< qevercloud::Guid > **linkedNotebookGuidsWithSyncChunksDownloaded** () const =0
- virtual IDownloadNotesStatusPtr **userAccountDownloadNotesStatus** () const =0
- virtual QHash< qevercloud::Guid, IDownloadNotesStatusPtr > **linkedNotebookDownloadNotesStatuses** () const =0
- virtual IDownloadResourcesStatusPtr **userAccountDownloadResourcesStatus** () const =0
- virtual QHash< qevercloud::Guid, IDownloadResourcesStatusPtr > **linkedNotebookDownloadResourcesStatuses** () const =0
- virtual ISendStatusPtr **userAccountSendStatus** () const =0
- virtual QHash< qevercloud::Guid, ISendStatusPtr > **linkedNotebookSendStatuses** () const =0
- virtual StopSynchronizationError **stopSynchronizationError** () const =0

Public Member Functions inherited from [quentier::utility::Printable](#)

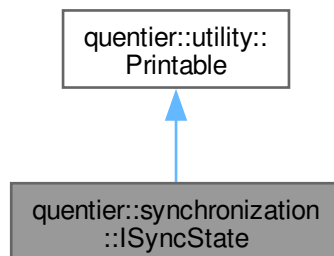
- virtual QTextStream & **print** (QTextStream &strm) const =0
- QString **toString** () const

5.47 quentier::synchronization::ISyncState Class Reference

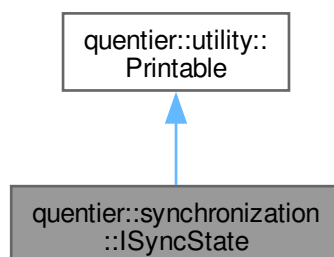
The [ISyncState](#) interface provides accessory methods to determine the sync state for the account.

```
#include <ISyncState.h>
```

Inheritance diagram for quentier::synchronization::ISyncState:



Collaboration diagram for quentier::synchronization::ISyncState:



Public Member Functions

- virtual qint32 **userDataUpdateCount** () const =0
- virtual qevercloud::Timestamp **userDataLastSyncTime** () const =0
- virtual QHash< qevercloud::Guid, qint32 > **linkedNotebookUpdateCounts** () const =0
- virtual QHash< qevercloud::Guid, qevercloud::Timestamp > **linkedNotebookLastSyncTimes** () const =0

Public Member Functions inherited from [quentier::utility::Printable](#)

- virtual QTextStream & **print** (QTextStream &strm) const =0
- QString **toString** () const

5.47.1 Detailed Description

The [ISyncState](#) interface provides accessory methods to determine the sync state for the account.

5.48 quentier::synchronization::ISyncStateBuilder Class Reference

Public Member Functions

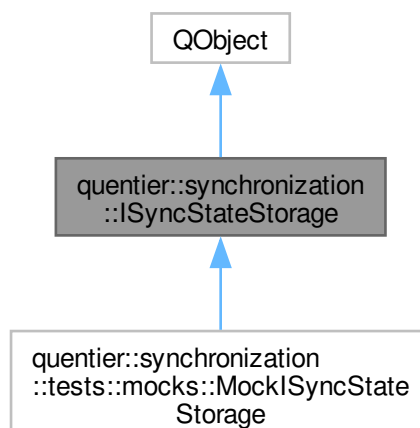
- virtual [ISyncStateBuilder](#) & **setUserDataUpdateCount** (qint32 updateCount)=0
- virtual [ISyncStateBuilder](#) & **setUserDataLastSyncTime** (qevercloud::Timestamp lastSyncTime)=0
- virtual [ISyncStateBuilder](#) & **setLinkedNotebookUpdateCounts** (QHash< qevercloud::Guid, qint32 > updateCounts)=0
- virtual [ISyncStateBuilder](#) & **setLinkedNotebookLastSyncTimes** (QHash< qevercloud::Guid, qevercloud::Timestamp > lastSyncTimes)=0
- virtual ISyncStatePtr **build** ()=0

5.49 quentier::synchronization::ISyncStateStorage Class Reference

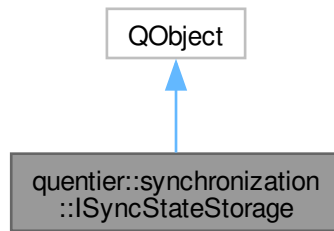
The [ISyncStateStorage](#) interface represents the interface of a class which stores sync state for given accounts persistently and provides access to previously stores sync states.

```
#include <ISyncStateStorage.h>
```

Inheritance diagram for quentier::synchronization::ISyncStateStorage:



Collaboration diagram for `quentier::synchronization::ISyncStateStorage`:



Signals

- void `notifySyncStateUpdated` (`Account` account, `ISyncStatePtr` syncState)

Public Member Functions

- virtual `ISyncStatePtr` `getSyncState` (const `Account` &account)=0
- virtual void `setSyncState` (const `Account` &account, `ISyncStatePtr` syncState)=0

Protected Member Functions

- `ISyncStateStorage` (`QObject` *parent=nullptr)

5.49.1 Detailed Description

The `ISyncStateStorage` interface represents the interface of a class which stores sync state for given accounts persistently and provides access to previously stores sync states.

5.49.2 Member Function Documentation

5.49.2.1 `notifySyncStateUpdated`

```
void quentier::synchronization::ISyncStateStorage::notifySyncStateUpdated (
    Account account,
    ISyncStatePtr syncState) [signal]
```

Classes implementing `ISyncStateStorage` interface are expected to emit `notifySyncStateUpdated` signal each time when sync state for the corresponding account is updated

5.50 `quentier::ResourceRecognitionIndexItem::ITextItem` Struct Reference

Public Member Functions

- virtual `QString text ()` const =0
- virtual `int weight ()` const =0

5.51 `quentier::synchronization::IUserStoreFactory` Class Reference

Public Member Functions

- virtual `qevercloud::IUserStorePtr createUserStore (QString userStoreUrl={}, qevercloud::IRequestContextPtr ctx={}, qevercloud::IRetryPolicyPtr retryPolicy={})`=0

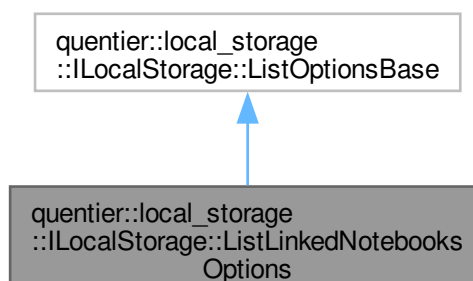
5.52 `quentier::local_storage::ILocalStorage::ListGuidsFilters` Struct Reference

Public Attributes

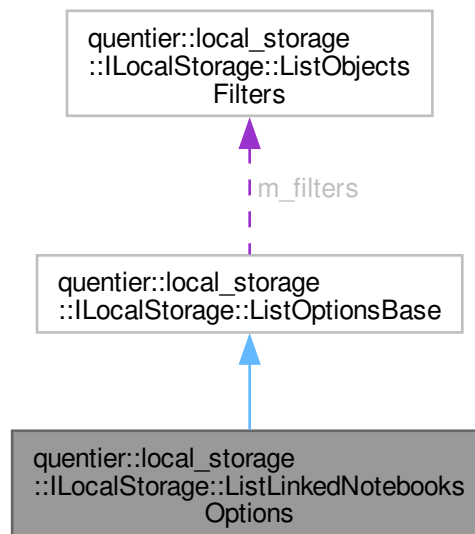
- `std::optional< ListObjectsFilter > m_locallyModifiedFilter`
- `std::optional< ListObjectsFilter > m_locallyFavoritedFilter`

5.53 `quentier::local_storage::ILocalStorage::ListLinkedNotebooksOptions` Struct Reference

Inheritance diagram for `quentier::local_storage::ILocalStorage::ListLinkedNotebooksOptions`:



Collaboration diagram for `quentier::local_storage::ILocalStorage::ListLinkedNotebooksOptions`:



Public Attributes

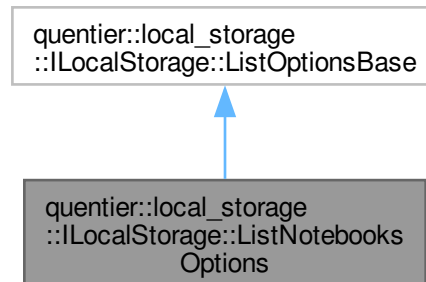
- `ListLinkedNotebooksOrder` **m_order** = `ListLinkedNotebooksOrder::NoOrder`

Public Attributes inherited from `quentier::local_storage::ILocalStorage::ListOptionsBase`

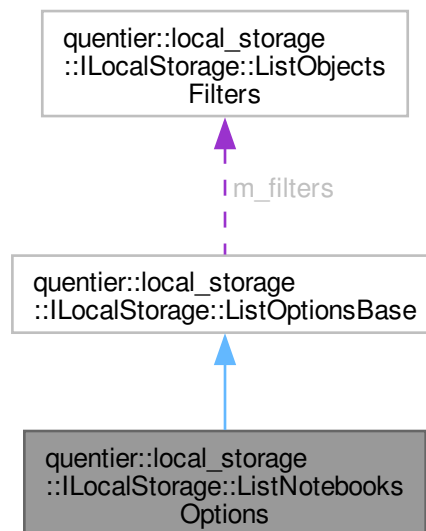
- `ListObjectsFilters` **m_filters** = {}
- quint64 **m_limit** = 0UL
- quint64 **m_offset** = 0UL
- `OrderDirection` **m_direction** = `OrderDirection::Ascending`

5.54 quantier::local_storage::ILocalStorage::ListNotebooksOptions Struct Reference

Inheritance diagram for quantier::local_storage::ILocalStorage::ListNotebooksOptions:



Collaboration diagram for quantier::local_storage::ILocalStorage::ListNotebooksOptions:



Public Attributes

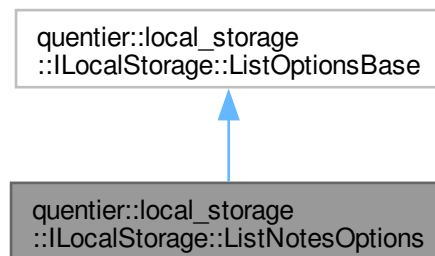
- ListNotebooksOrder **m_order** = ListNotebooksOrder::NoOrder
- [Affiliation](#) **m_affiliation** = Affiliation::Any
- `QList< qevercloud::Guid >` **m_linkedNotebookGuids**

Public Attributes inherited from [quantier::local_storage::ILocalStorage::ListOptionsBase](#)

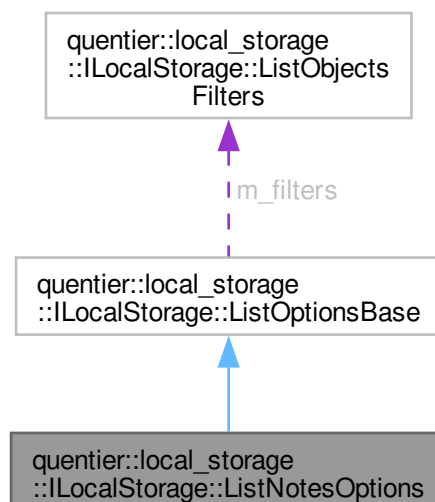
- [ListObjectsFilters](#) **m_filters** = {}
- quint64 **m_limit** = 0UL
- quint64 **m_offset** = 0UL
- OrderDirection **m_direction** = OrderDirection::Ascending

5.55 quantier::local_storage::ILocalStorage::ListNotesOptions Struct Reference

Inheritance diagram for quantier::local_storage::ILocalStorage::ListNotesOptions:



Collaboration diagram for quantier::local_storage::ILocalStorage::ListNotesOptions:



Public Attributes

- ListNotesOrder **m_order** = ListNotesOrder::NoOrder

Public Attributes inherited from [quentier::local_storage::ILocalStorage::ListOptionsBase](#)

- [ListObjectsFilters](#) **m_filters** = {}
- quint64 **m_limit** = 0UL
- quint64 **m_offset** = 0UL
- OrderDirection **m_direction** = OrderDirection::Ascending

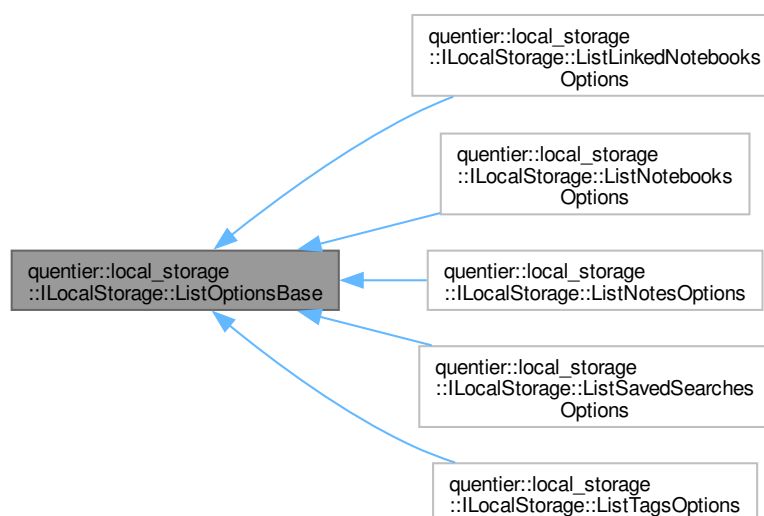
5.56 quentier::local_storage::ILocalStorage::ListObjectsFilters Struct Reference

Public Attributes

- std::optional< ListObjectsFilter > **m_locallyModifiedFilter**
- std::optional< ListObjectsFilter > **m_withGuidFilter**
- std::optional< ListObjectsFilter > **m_localOnlyFilter**
- std::optional< ListObjectsFilter > **m_locallyFavoritedFilter**

5.57 quentier::local_storage::ILocalStorage::ListOptionsBase Struct Reference

Inheritance diagram for quentier::local_storage::ILocalStorage::ListOptionsBase:



Collaboration diagram for quantier::local_storage::ILocalStorage::ListOptionsBase:

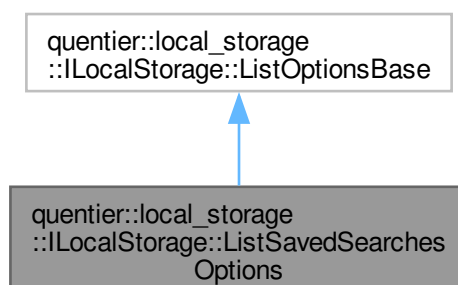


Public Attributes

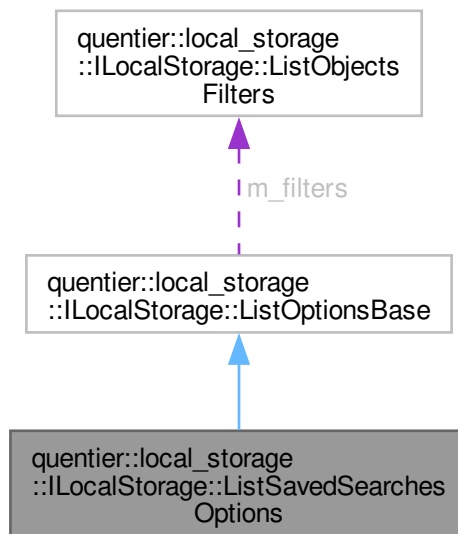
- [ListObjectsFilters](#) `m_filters` = {}
- quint64 `m_limit` = 0UL
- quint64 `m_offset` = 0UL
- OrderDirection `m_direction` = OrderDirection::Ascending

5.58 quantier::local_storage::ILocalStorage::ListSavedSearchesOptions Struct Reference

Inheritance diagram for quantier::local_storage::ILocalStorage::ListSavedSearchesOptions:



Collaboration diagram for `quentier::local_storage::ILocalStorage::ListSavedSearchesOptions`:



Public Attributes

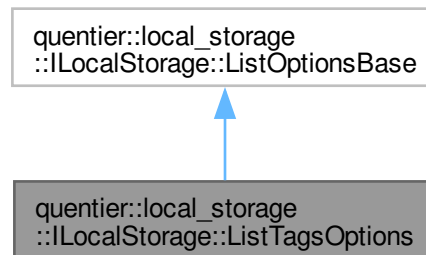
- `ListSavedSearchesOrder` **m_order** = `ListSavedSearchesOrder::NoOrder`

Public Attributes inherited from [quentier::local_storage::ILocalStorage::ListOptionsBase](#)

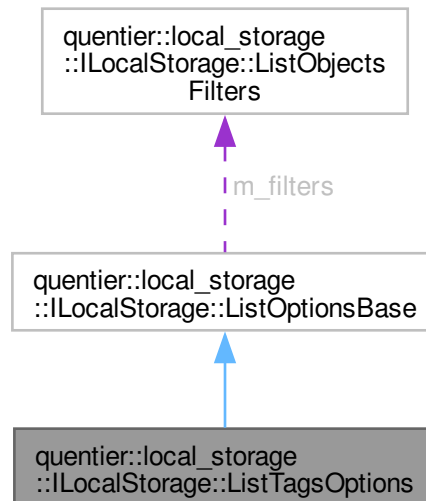
- [ListObjectsFilters](#) **m_filters** = {}
- quint64 **m_limit** = 0UL
- quint64 **m_offset** = 0UL
- `OrderDirection` **m_direction** = `OrderDirection::Ascending`

5.59 quantier::local_storage::ILocalStorage::ListTagsOptions Struct Reference

Inheritance diagram for quantier::local_storage::ILocalStorage::ListTagsOptions:



Collaboration diagram for quantier::local_storage::ILocalStorage::ListTagsOptions:



Public Attributes

- ListTagsOrder **m_order** = ListTagsOrder::NoOrder
- [Affiliation](#) **m_affiliation** = Affiliation::Any
- QList< qevercloud::Guid > **m_linkedNotebookGuids**
- [TagNotesRelation](#) **m_tagNotesRelation** = [TagNotesRelation::Any](#)

Public Attributes inherited from [quentier::local_storage::ILocalStorage::ListOptionsBase](#)

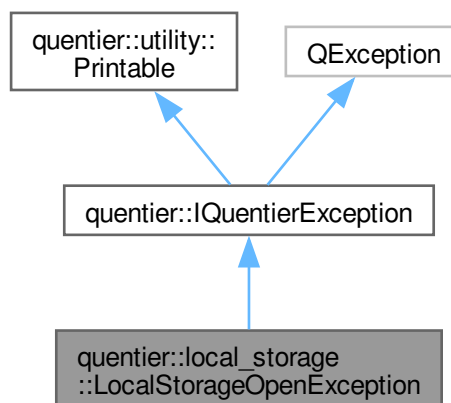
- [ListObjectsFilters](#) **m_filters** = {}
- quint64 **m_limit** = 0UL
- quint64 **m_offset** = 0UL
- OrderDirection **m_direction** = OrderDirection::Ascending

5.60 quentier::local_storage::LocalStorageOpenException Class Reference

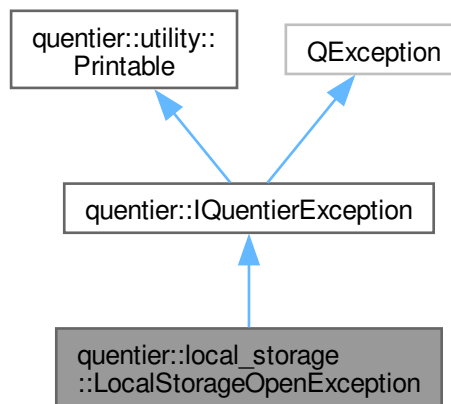
The [LocalStorageOpenException](#) is thrown on failure to open the local storage database.

```
#include <LocalStorageOpenException.h>
```

Inheritance diagram for quentier::local_storage::LocalStorageOpenException:



Collaboration diagram for `quentier::local_storage::LocalStorageOpenException`:



Public Member Functions

- **LocalStorageOpenException** (const [ErrorString](#) &message)
- LocalStorageOpenException * **clone** () const override
- void **raise** () const override

Public Member Functions inherited from [quentier::IQuentierException](#)

- [ErrorString](#) **errorMessage** () const
- QString **localizedErrorMessage** () const
- QString **nonLocalizedErrorMessage** () const
- const char * **what** () const noexcept override
- QTextStream & [print](#) (QTextStream &strm) const override

Public Member Functions inherited from [quentier::utility::Printable](#)

- QString **toString** () const

Protected Member Functions

- QString [exceptionDisplayName](#) () const override

Protected Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** ([ErrorString](#) message)
- **IQuentierException** (const IQuentierException &other)
- IQuentierException & **operator=** (const IQuentierException &other)

5.60.1 Detailed Description

The [LocalStorageOpenException](#) is thrown on failure to open the local storage database.

5.60.2 Member Function Documentation

5.60.2.1 exceptionDisplayName()

```
QString quantier::local_storage::LocalStorageOpenException::exceptionDisplayName () const
[nodiscard], [override], [protected], [virtual]
```

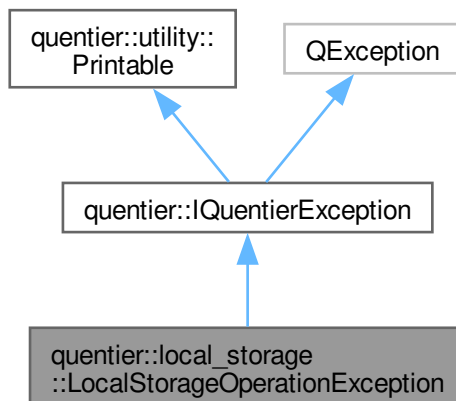
Implements [quantier::IQuantierException](#).

5.61 quantier::local_storage::LocalStorageOperationException Class Reference

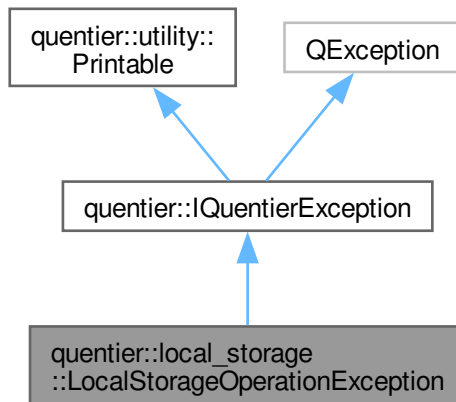
The [LocalStorageOperationException](#) is thrown when the local storage encounters some internal error during the attempt to process some operation.

```
#include <LocalStorageOperationException.h>
```

Inheritance diagram for `quantier::local_storage::LocalStorageOperationException`:



Collaboration diagram for `quentier::local_storage::LocalStorageOperationException`:



Public Member Functions

- `LocalStorageOperationException` (`ErrorString` message)
- `LocalStorageOperationException * clone ()` const override
- `void raise ()` const override

Public Member Functions inherited from `quentier::IQuentierException`

- `ErrorString errorMessage ()` const
- `QString localizedErrorMessage ()` const
- `QString nonLocalizedErrorMessage ()` const
- `const char * what ()` const noexcept override
- `QTextStream & print (QTextStream &strm)` const override

Public Member Functions inherited from `quentier::utility::Printable`

- `QString toString ()` const

Protected Member Functions

- `QString exceptionDisplayName ()` const override

Protected Member Functions inherited from `quentier::IQuentierException`

- `IQuentierException` (`ErrorString` message)
- `IQuentierException` (const `IQuentierException` &other)
- `IQuentierException & operator=` (const `IQuentierException` &other)

5.61.1 Detailed Description

The [LocalStorageOperationException](#) is thrown when the local storage encounters some internal error during the attempt to process some operation.

5.61.2 Member Function Documentation

5.61.2.1 exceptionDisplayName()

```
QString quentier::local_storage::LocalStorageOperationException::exceptionDisplayName () const
[nodiscard], [override], [protected], [virtual]
```

Implements [quentier::IQuentierException](#).

5.62 quentier::utility::LRUCache< Key, Value, Allocator > Class Template Reference

Public Types

- using **key_type** = Key
- using **mapped_type** = Value
- using **allocator_type** = Allocator
- using **value_type** = std::pair<key_type, mapped_type>
- using **container_type** = std::list<value_type, allocator_type>
- using **size_type** = typename container_type::size_type
- using **difference_type** = typename container_type::difference_type
- using **iterator** = typename container_type::iterator
- using **const_iterator** = typename container_type::const_iterator
- using **reverse_iterator** = std::reverse_iterator<iterator>
- using **const_reverse_iterator** = std::reverse_iterator<const_iterator>
- using **reference** = value_type &
- using **const_reference** = const value_type &
- using **pointer** = typename std::allocator_traits<allocator_type>::pointer
- using [const_pointer](#)

Public Member Functions

- **LRUCache** (const std::size_t maxSize=100)
- iterator **begin** () noexcept
- const_iterator **begin** () const noexcept
- reverse_iterator **rbegin** () noexcept
- const_reverse_iterator **rbegin** () const noexcept
- iterator **end** () noexcept
- const_iterator **end** () const noexcept
- reverse_iterator **rend** () noexcept
- const_reverse_iterator **rend** () const noexcept
- bool **empty** () const noexcept
- std::size_t **size** () const noexcept
- std::size_t **max_size** () const noexcept
- void **clear** ()
- void **put** (const key_type &key, const mapped_type &value)
- const mapped_type * **get** (const key_type &key) const noexcept
- bool **exists** (const key_type &key) const noexcept
- bool **remove** (const key_type &key) noexcept
- void **setMaxSize** (const std::size_t maxSize)

5.62.1 Member Typedef Documentation

5.62.1.1 `const_pointer`

```
template<class Key, class Value, class Allocator = std::allocator<std::pair<Key, Value>>>>  
using quentier::utility::LRUCache< Key, Value, Allocator >::const_pointer
```

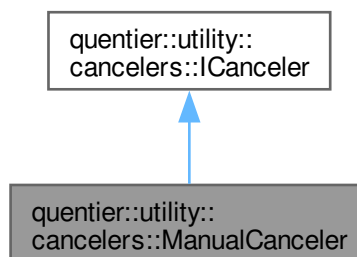
Initial value:

```
typename std::allocator_traits<allocator_type>::const_pointer
```

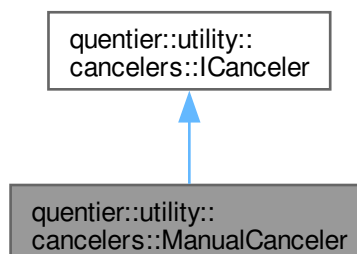
5.63 `quentier::utility::cancelers::ManualCanceler` Class Reference

```
#include <ManualCanceler.h>
```

Inheritance diagram for `quentier::utility::cancelers::ManualCanceler`:



Collaboration diagram for `quentier::utility::cancelers::ManualCanceler`:



Public Member Functions

- **ManualCanceler** (ManualCanceler &&other) noexcept
- ManualCanceler & **operator=** (ManualCanceler &&other) noexcept
- void [cancel](#) () noexcept
- bool [isCanceled](#) () const noexcept override

5.63.1 Detailed Description

[ICanceler](#) which allows one to manually call cancel method to cancel some task

5.63.2 Member Function Documentation

5.63.2.1 cancel()

```
void quantier::utility::cancelers::ManualCanceler::cancel () [noexcept]
```

Manually cancel a task

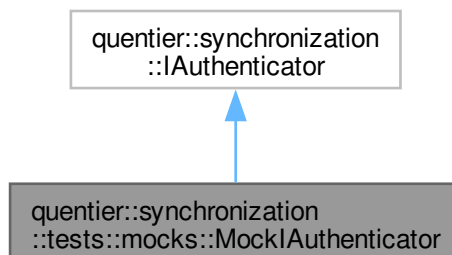
5.63.2.2 isCanceled()

```
bool quantier::utility::cancelers::ManualCanceler::isCanceled () const [nodiscard], [override], [virtual], [noexcept]
```

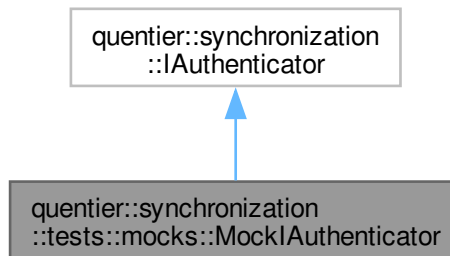
Implements [quantier::utility::cancelers::ICanceler](#).

5.64 quantier::synchronization::tests::mocks::MockIAuthenticator Class Reference

Inheritance diagram for quantier::synchronization::tests::mocks::MockIAuthenticator:



Collaboration diagram for `quentier::synchronization::tests::mocks::MockIAuthenticator`:



Public Member Functions

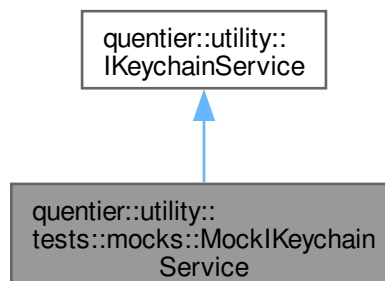
- **MOCK_METHOD** (`QFuture< IAuthenticationInfoPtr >`, `authenticateNewAccount`, `()`, `(override)`)
- **MOCK_METHOD** (`QFuture< IAuthenticationInfoPtr >`, `authenticateAccount`, `(Account account)`, `(override)`)

Public Member Functions inherited from `quentier::synchronization::IAuthenticator`

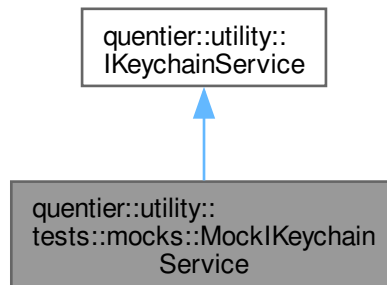
- virtual `QFuture< IAuthenticationInfoPtr >` **authenticateNewAccount** `()=0`
- virtual `QFuture< IAuthenticationInfoPtr >` **authenticateAccount** `(Account account)=0`

5.65 `quentier::utility::tests::mocks::MockIKeychainService` Class Reference

Inheritance diagram for `quentier::utility::tests::mocks::MockIKeychainService`:



Collaboration diagram for `quentier::utility::tests::mocks::MockIKeychainService`:



Public Member Functions

- **MOCK_METHOD** (`QFuture< void >`, [writePassword](#),(`QString` service, `QString` key, `QString` password),(`override`))
- **MOCK_METHOD** (`QFuture< QString >`, [readPassword](#),(`QString` service, `QString` key),(`const`, `override`))
- **MOCK_METHOD** (`QFuture< void >`, [deletePassword](#),(`QString` service, `QString` key),(`override`))

Public Member Functions inherited from [quentier::utility::IKeychainService](#)

- virtual `QFuture< void >` [writePassword](#) (`QString` service, `QString` key, `QString` password)=0
- virtual `QFuture< QString >` [readPassword](#) (`QString` service, `QString` key) `const` =0
- virtual `QFuture< void >` [deletePassword](#) (`QString` service, `QString` key)=0

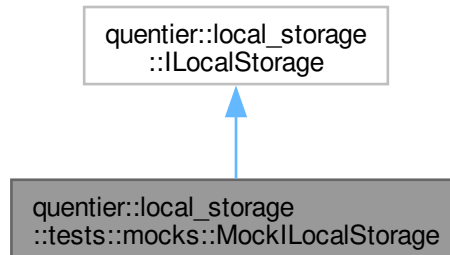
Additional Inherited Members

Public Types inherited from [quentier::utility::IKeychainService](#)

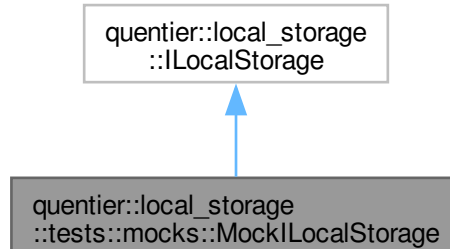
- enum class [ErrorCode](#) {
[NoError](#) , [EntryNotFound](#) , [CouldNotDeleteEntry](#) , [AccessDeniedByUser](#) ,
[AccessDenied](#) , [NoBackendAvailable](#) , [NotImplemented](#) , [OtherError](#) }

5.66 quantier::local_storage::tests::mocks::MockLocalStorage Class Reference

Inheritance diagram for quantier::local_storage::tests::mocks::MockLocalStorage:



Collaboration diagram for quantier::local_storage::tests::mocks::MockLocalStorage:



Public Member Functions

- **MOCK_METHOD** (QFuture< bool >, isVersionTooHigh,(),(const, override))
- **MOCK_METHOD** (QFuture< bool >, requiresUpgrade,(),(const, override))
- **MOCK_METHOD** (QFuture< QList< IPatchPtr > >, requiredPatches,(),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, version,(),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, highestSupportedVersion,(),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, userCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putUser,(qevercloud::User user),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::User > >, findUserById,(qevercloud::UserID userId),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeUserById,(qevercloud::UserID userId),(override))
- **MOCK_METHOD** (QFuture< quint32 >, notebookCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putNotebook,(qevercloud::Notebook notebook),(override))

- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Notebook > >, findNotebookByLocalId,(QString localId),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Notebook > >, findNotebookByGuid,(qevercloud::Guid guid),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Notebook > >, findNotebookByName,(QString name, std::optional< qevercloud::Guid > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Notebook > >, findDefaultNotebook,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeNotebookByLocalId,(QString localId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeNotebookByGuid,(qevercloud::Guid guid),(override))
- **MOCK_METHOD** (QFuture< void >, expungeNotebookByName,(QString name, std::optional< qevercloud::Guid > linkedNotebookGuid),(override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Notebook > >, listNotebooks,(ListNotebooksOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::SharedNotebook > >, listSharedNotebooks,(qevercloud::Guid notebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< QSet< qevercloud::Guid > >, listNotebookGuids,(ListGuidsFilters filters, std::optional< qevercloud::Guid > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, linkedNotebookCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putLinkedNotebook,(qevercloud::LinkedNotebook linkedNotebook),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::LinkedNotebook > >, findLinkedNotebookByGuid,(qevercloud::Guid guid),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeLinkedNotebookByGuid,(qevercloud::Guid guid),(override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::LinkedNotebook > >, listLinkedNotebooks,(ListLinkedNotebooksOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, noteCount,(NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, noteCountPerNotebookLocalId,(QString notebookLocalId, NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, noteCountPerTagLocalId,(QString tagLocalId, NoteCountOptions options),(const, override))
- **MOCK_METHOD** ((QFuture< QHash< QString, quint32 > >), noteCountsPerTags,(ListTagsOptions listTagsOptions, NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, noteCountPerNotebookAndTagLocalIds,(QStringList notebookLocalIds, QStringList tagLocalIds, NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< void >, putNote,(qevercloud::Note note),(override))
- **MOCK_METHOD** (QFuture< void >, updateNote,(qevercloud::Note note, UpdateNoteOptions options),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Note > >, findNoteByLocalId,(QString localId, FetchNoteOptions options),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Note > >, findNoteByGuid,(qevercloud::Guid guid, FetchNoteOptions options),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeNoteByLocalId,(QString localId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeNoteByGuid,(qevercloud::Guid guid),(override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotes,(FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotesPerNotebookLocalId,(QString notebookLocalId, FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotesPerTagLocalId,(QString tagLocalId, FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotesPerNotebookAndTagLocalIds,(QStringList notebookLocalIds, QStringList tagLocalIds, FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotesByLocalIds,(QStringList noteLocalIds, FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QSet< qevercloud::Guid > >, listNoteGuids,(ListGuidsFilters filters, std::optional< qevercloud::Guid > linkedNotebookGuid),(const, override))

- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, queryNotes,([NoteSearchQuery](#) query, Fetch←
NoteOptions fetchOptions),(const, override))
- **MOCK_METHOD** (QFuture< QStringList >, queryNoteLocalIds,([NoteSearchQuery](#) query),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, tagCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putTag,(qevercloud::Tag tag),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Tag > >, findTagByLocalId,(QString tagLocal←
Id),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Tag > >, findTagByGuid,(qevercloud::Guid tag←
Guid),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Tag > >, findTagByName,(QString tagName, std←
::optional< QString > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Tag > >, listTags,([ListTagsOptions](#) options),(const, over-
ride))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Tag > >, listTagsPerNoteLocalId,(QString noteLocalId,
[ListTagsOptions](#) options),(const, override))
- **MOCK_METHOD** (QFuture< QSet< qevercloud::Guid > >, listTagGuids,([ListGuidsFilters](#) filters, std←
::optional< qevercloud::Guid > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeTagByLocalId,(QString tagLocalId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeTagByGuid,(qevercloud::Guid tagGuid),(override))
- **MOCK_METHOD** (QFuture< void >, expungeTagByName,(QString name, std::optional< qevercloud::Guid
> linkedNotebookGuid),(override))
- **MOCK_METHOD** (QFuture< quint32 >, resourceCount,(NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, resourceCountPerNoteLocalId,(QString noteLocalId),(const, over-
ride))
- **MOCK_METHOD** (QFuture< void >, putResource,(qevercloud::Resource resource),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Resource > >, findResourceByLocalId,(QString
resourceLocalId, FetchResourceOptions options),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Resource > >, findResourceByGuid,(qevercloud←
::Guid resourceGuid, FetchResourceOptions options),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeResourceByLocalId,(QString resourceLocalId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeResourceByGuid,(qevercloud::Guid resourceGuid),(override))
- **MOCK_METHOD** (QFuture< quint32 >, savedSearchCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putSavedSearch,(qevercloud::SavedSearch search),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::SavedSearch > >, findSavedSearchByLocal←
Id,(QString localId),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::SavedSearch > >, findSavedSearchBy←
Guid,(qevercloud::Guid guid),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::SavedSearch > >, findSavedSearchBy←
Name,(QString name),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::SavedSearch > >, listSavedSearches,([ListSavedSearchesOptions](#)
options),(const, override))
- **MOCK_METHOD** (QFuture< QSet< qevercloud::Guid > >, listSavedSearchGuids,([ListGuidsFilters](#) fil-
ters),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeSavedSearchByLocalId,(QString localId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeSavedSearchByGuid,(qevercloud::Guid guid),(override))
- **MOCK_METHOD** (QFuture< qint32 >, highestUpdateSequenceNumber,(HighestUsnOption option),(const,
override))
- **MOCK_METHOD** (QFuture< qint32 >, highestUpdateSequenceNumber,(qevercloud::Guid linked←
NotebookGuid),(const, override))
- **MOCK_METHOD** (ILocalStorageNotifier *, [notifier](#),(),(const, override))

Public Member Functions inherited from [quentier::local_storage::ILocalStorage](#)

- **Q_DECLARE_FLAGS** (StartupOptions, StartupOption)
- virtual QFuture< bool > **isVersionTooHigh** () const =0
- virtual QFuture< bool > **requiresUpgrade** () const =0
- virtual QFuture< QList< IPatchPtr > > **requiredPatches** () const =0
- virtual QFuture< quint32 > **version** () const =0
- virtual QFuture< quint32 > **highestSupportedVersion** () const =0
- virtual QFuture< quint32 > **userCount** () const =0
- virtual QFuture< void > **putUser** (qevercloud::User user)=0
- virtual QFuture< std::optional< qevercloud::User > > **findUserById** (qevercloud::UserID userId) const =0
- virtual QFuture< void > **expungeUserById** (qevercloud::UserID userId)=0
- virtual QFuture< quint32 > **notebookCount** () const =0
- virtual QFuture< void > **putNotebook** (qevercloud::Notebook notebook)=0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findNotebookByLocalId** (QString notebookLocalId) const =0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findNotebookByGuid** (qevercloud::Guid guid) const =0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findNotebookByName** (QString notebookName, std::optional< qevercloud::Guid > linkedNotebookGuid=std::nullopt) const =0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findDefaultNotebook** () const =0
- virtual QFuture< void > **expungeNotebookByLocalId** (QString notebookLocalId)=0
- virtual QFuture< void > **expungeNotebookByGuid** (qevercloud::Guid notebookGuid)=0
- virtual QFuture< void > **expungeNotebookByName** (QString name, std::optional< qevercloud::Guid > linkedNotebookGuid=std::nullopt)=0
- virtual QFuture< QList< qevercloud::Notebook > > **listNotebooks** ([ListNotebooksOptions](#) options={}) const =0
- virtual QFuture< QList< qevercloud::SharedNotebook > > **listSharedNotebooks** (qevercloud::Guid notebookGuid={}) const =0
- virtual QFuture< QSet< qevercloud::Guid > > **listNotebookGuids** ([ListGuidsFilters](#) filters, std::optional< qevercloud::Guid > linkedNotebookGuid={}) const =0
- virtual QFuture< quint32 > **linkedNotebookCount** () const =0
- virtual QFuture< void > **putLinkedNotebook** (qevercloud::LinkedNotebook linkedNotebook)=0
- virtual QFuture< std::optional< qevercloud::LinkedNotebook > > **findLinkedNotebookByGuid** (qevercloud::Guid guid) const =0
- virtual QFuture< void > **expungeLinkedNotebookByGuid** (qevercloud::Guid guid)=0
- virtual QFuture< QList< qevercloud::LinkedNotebook > > **listLinkedNotebooks** ([ListLinkedNotebooksOptions](#) options={}) const =0
- virtual QFuture< quint32 > **noteCount** (NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< quint32 > **noteCountPerNotebookLocalId** (QString notebookLocalId, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< quint32 > **noteCountPerTagLocalId** (QString tagLocalId, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< QHash< QString, quint32 > > **noteCountsPerTags** ([ListTagsOptions](#) listTagsOptions={}, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< quint32 > **noteCountPerNotebookAndTagLocalIds** (QStringList notebookLocalIds, QStringList tagLocalIds, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< void > **putNote** (qevercloud::Note note)=0
- virtual QFuture< void > **updateNote** (qevercloud::Note note, UpdateNoteOptions options)=0
- virtual QFuture< std::optional< qevercloud::Note > > **findNoteByLocalId** (QString noteLocalId, FetchNoteOptions options) const =0
- virtual QFuture< std::optional< qevercloud::Note > > **findNoteByGuid** (qevercloud::Guid noteGuid, FetchNoteOptions options) const =0

- virtual `QFuture< QList< qevercloud::Note > >` **listNotes** (`FetchNoteOptions` fetchOptions, [ListNotesOptions](#) listOptions={}) const =0
- virtual `QFuture< QList< qevercloud::Note > >` **listNotesPerNotebookLocalId** (`QString` notebookLocalId, `FetchNoteOptions` fetchOptions, [ListNotesOptions](#) listOptions={}) const =0
- virtual `QFuture< QList< qevercloud::Note > >` **listNotesPerTagLocalId** (`QString` tagLocalId, `FetchNoteOptions` fetchOptions, [ListNotesOptions](#) listOptions={}) const =0
- virtual `QFuture< QList< qevercloud::Note > >` **listNotesPerNotebookAndTagLocalIds** (`QStringList` notebookLocalIds, `QStringList` tagLocalIds, `FetchNoteOptions` fetchOptions, [ListNotesOptions](#) listOptions={}) const =0
- virtual `QFuture< QList< qevercloud::Note > >` **listNotesByLocalIds** (`QStringList` noteLocalIds, `FetchNoteOptions` fetchOptions, [ListNotesOptions](#) listOptions={}) const =0
- virtual `QFuture< QSet< qevercloud::Guid > >` **listNoteGuids** ([ListGuidsFilters](#) filters, `std::optional< qevercloud::Guid >` linkedNotebookGuid={}) const =0
- virtual `QFuture< QList< qevercloud::Note > >` **queryNotes** ([NoteSearchQuery](#) query, `FetchNoteOptions` fetchOptions) const =0
- virtual `QFuture< QStringList >` **queryNoteLocalIds** ([NoteSearchQuery](#) query) const =0
- virtual `QFuture< void >` **expungeNoteByLocalId** (`QString` noteLocalId)=0
- virtual `QFuture< void >` **expungeNoteByGuid** (`qevercloud::Guid` noteGuid)=0
- virtual `QFuture< quint32 >` **tagCount** () const =0
- virtual `QFuture< void >` **putTag** (`qevercloud::Tag` tag)=0
- virtual `QFuture< std::optional< qevercloud::Tag > >` **findTagByLocalId** (`QString` tagLocalId) const =0
- virtual `QFuture< std::optional< qevercloud::Tag > >` **findTagByGuid** (`qevercloud::Guid` tagGuid) const =0
- virtual `QFuture< std::optional< qevercloud::Tag > >` **findTagByName** (`QString` tagName, `std::optional< qevercloud::Guid >` linkedNotebookGuid=`std::nullopt`) const =0
- virtual `QFuture< QList< qevercloud::Tag > >` **listTags** ([ListTagsOptions](#) options={}) const =0
- virtual `QFuture< QList< qevercloud::Tag > >` **listTagsPerNoteLocalId** (`QString` noteLocalId, [ListTagsOptions](#) options={}) const =0
- virtual `QFuture< QSet< qevercloud::Guid > >` **listTagGuids** ([ListGuidsFilters](#) filters, `std::optional< qevercloud::Guid >` linkedNotebookGuid={}) const =0
- virtual `QFuture< void >` **expungeTagByLocalId** (`QString` tagLocalId)=0
- virtual `QFuture< void >` **expungeTagByGuid** (`qevercloud::Guid` tagGuid)=0
- virtual `QFuture< void >` **expungeTagByName** (`QString` name, `std::optional< qevercloud::Guid >` linkedNotebookGuid=`std::nullopt`)=0
- virtual `QFuture< quint32 >` **resourceCount** (`NoteCountOptions` options=`NoteCountOptions`(`NoteCountOptions::IncludeNonDeletedNotes`)) const =0
- virtual `QFuture< quint32 >` **resourceCountPerNoteLocalId** (`QString` noteLocalId) const =0
- virtual `QFuture< void >` **putResource** (`qevercloud::Resource` resource)=0
- virtual `QFuture< std::optional< qevercloud::Resource > >` **findResourceByLocalId** (`QString` resourceLocalId, `FetchResourceOptions` options={}) const =0
- virtual `QFuture< std::optional< qevercloud::Resource > >` **findResourceByGuid** (`qevercloud::Guid` resourceGuid, `FetchResourceOptions` options={}) const =0
- virtual `QFuture< void >` **expungeResourceByLocalId** (`QString` resourceLocalId)=0
- virtual `QFuture< void >` **expungeResourceByGuid** (`qevercloud::Guid` resourceGuid)=0
- virtual `QFuture< quint32 >` **savedSearchCount** () const =0
- virtual `QFuture< void >` **putSavedSearch** (`qevercloud::SavedSearch` search)=0
- virtual `QFuture< std::optional< qevercloud::SavedSearch > >` **findSavedSearchByLocalId** (`QString` savedSearchLocalId) const =0
- virtual `QFuture< std::optional< qevercloud::SavedSearch > >` **findSavedSearchByGuid** (`qevercloud::Guid` guid) const =0
- virtual `QFuture< std::optional< qevercloud::SavedSearch > >` **findSavedSearchByName** (`QString` name) const =0
- virtual `QFuture< QList< qevercloud::SavedSearch > >` **listSavedSearches** ([ListSavedSearchesOptions](#) options={}) const =0
- virtual `QFuture< QSet< qevercloud::Guid > >` **listSavedSearchGuids** ([ListGuidsFilters](#) filters) const =0
- virtual `QFuture< void >` **expungeSavedSearchByLocalId** (`QString` savedSearchLocalId)=0
- virtual `QFuture< void >` **expungeSavedSearchByGuid** (`qevercloud::Guid` guid)=0

- virtual QFuture< qint32 > **highestUpdateSequenceNumber** (HighestUsnOption option) const =0
- virtual QFuture< qint32 > **highestUpdateSequenceNumber** (qevercloud::Guid linkedNotebookGuid) const =0
- virtual [ILocalStorageNotifier](#) * **notifier** () const =0

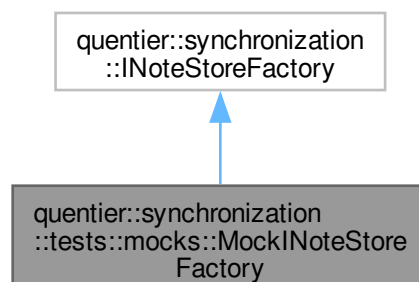
Additional Inherited Members

Public Types inherited from [quentier::local_storage::ILocalStorage](#)

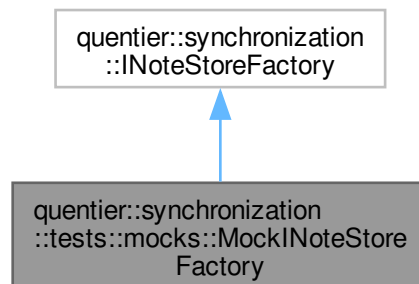
- enum class **StartupOption** { **ClearDatabase** = 1 << 1 , **OverrideLock** = 1 << 2 }
- enum class **ListObjectsFilter** { **Include** , **Exclude** }
- enum class **OrderDirection** { **Ascending** , **Descending** }
- enum class **ListNotebooksOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByNotebookName** , **ByCreationTimestamp** , **ByModificationTimestamp** }
- enum class **ListLinkedNotebooksOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByShareName** , **ByUsername** }
- enum class **ListTagsOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByName** }
- enum class **ListNotesOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByTitle** , **ByCreationTimestamp** , **ByModificationTimestamp** , **ByDeletionTimestamp** , **ByAuthor** , **BySource** , **BySourceApplication** , **ByReminderTime** , **ByPlaceName** }
- enum class **ListSavedSearchesOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByName** , **ByFormat** }
- enum class **Affiliation** { **Any** , **User** , **AnyLinkedNotebook** , **ParticularLinkedNotebooks** }
- enum class **TagNotesRelation** { **Any** , **WithNotes** , **WithoutNotes** }
- enum class **NoteCountOption** { **IncludeNonDeletedNotes** = 1 << 1 , **IncludeDeletedNotes** = 1 << 2 }
- enum class **UpdateNoteOption** { **UpdateResourceMetadata** = 1 << 1 , **UpdateResourceBinaryData** = 1 << 2 , **UpdateTags** = 1 << 3 }
- enum class **FetchNoteOption** { **WithResourceMetadata** = 1 << 1 , **WithResourceBinaryData** = 1 << 2 }
- enum class **FetchResourceOption** { **WithBinaryData** = 1 << 1 }
- enum class **HighestUsnOption** { **WithinUserOwnContent** , **WithinUserOwnContentAndLinkedNotebooks** }

5.67 quentier::synchronization::tests::mocks::MockINoteStoreFactory Class Reference

Inheritance diagram for quentier::synchronization::tests::mocks::MockINoteStoreFactory:



Collaboration diagram for `quentier::synchronization::tests::mocks::MockINoteStoreFactory`:



Public Member Functions

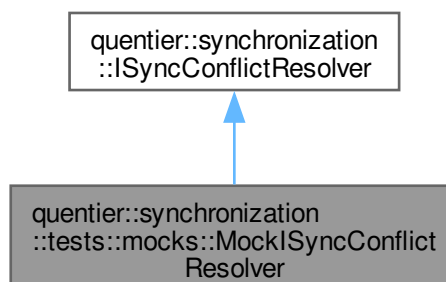
- **MOCK_METHOD** (`::qevercloud::INoteStorePtr`, `createNoteStore`, (`QString` `noteStoreUrl`, `std::optional<`
`::qevercloud::Guid` `>` `linkedNotebookGuid`, `::qevercloud::IRequestContextPtr` `ctx`, `::qevercloud::IRetryPolicyPtr` `retryPolicy`), (`override`))

Public Member Functions inherited from [quentier::synchronization::INoteStoreFactory](#)

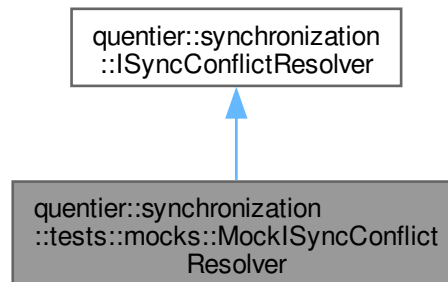
- virtual `qevercloud::INoteStorePtr` **createNoteStore** (`QString` `noteStoreUrl`=`{}`, `std::optional<` `qevercloud::Guid` `>` `linkedNotebookGuid`=`{}`, `qevercloud::IRequestContextPtr` `ctx`=`{}`, `qevercloud::IRetryPolicyPtr` `retryPolicy`=`{}`)=0

5.68 `quentier::synchronization::tests::mocks::MockISyncConflictResolver` Class Reference

Inheritance diagram for `quentier::synchronization::tests::mocks::MockISyncConflictResolver`:



Collaboration diagram for `quentier::synchronization::tests::mocks::MockISyncConflictResolver`:



Public Member Functions

- **MOCK_METHOD** (`QFuture< NotebookConflictResolution >`, `resolveNotebookConflict`, (`::qevercloud::Notebook theirs`, `::qevercloud::Notebook mine`), (`override`))
- **MOCK_METHOD** (`QFuture< NoteConflictResolution >`, `resolveNoteConflict`, (`::qevercloud::Note theirs`, `::qevercloud::Note mine`), (`override`))
- **MOCK_METHOD** (`QFuture< SavedSearchConflictResolution >`, `resolveSavedSearchConflict`, (`::qevercloud::SavedSearch theirs`, `::qevercloud::SavedSearch mine`), (`override`))
- **MOCK_METHOD** (`QFuture< TagConflictResolution >`, `resolveTagConflict`, (`::qevercloud::Tag theirs`, `::qevercloud::Tag mine`), (`override`))

Public Member Functions inherited from `quentier::synchronization::ISyncConflictResolver`

- virtual `QFuture< NotebookConflictResolution >` **resolveNotebookConflict** (`qevercloud::Notebook theirs`, `qevercloud::Notebook mine`)=0
- virtual `QFuture< NoteConflictResolution >` **resolveNoteConflict** (`qevercloud::Note theirs`, `qevercloud::Note mine`)=0
- virtual `QFuture< SavedSearchConflictResolution >` **resolveSavedSearchConflict** (`qevercloud::SavedSearch theirs`, `qevercloud::SavedSearch mine`)=0
- virtual `QFuture< TagConflictResolution >` **resolveTagConflict** (`qevercloud::Tag theirs`, `qevercloud::Tag mine`)=0

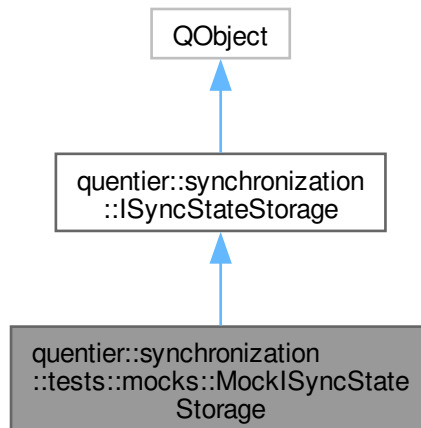
Additional Inherited Members

Public Types inherited from `quentier::synchronization::ISyncConflictResolver`

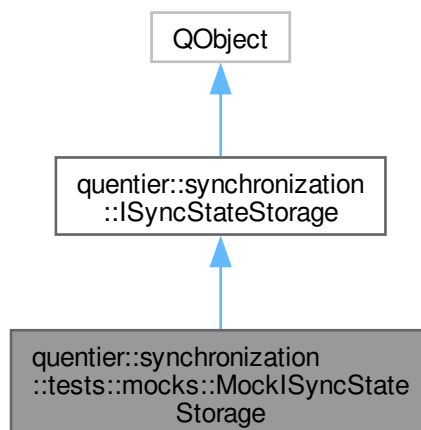
- using `NotebookConflictResolution`
- using `NoteConflictResolution`
- using `SavedSearchConflictResolution`
- using `TagConflictResolution`

5.69 quantier::synchronization::tests::mocks::MockISyncStateStorage Class Reference

Inheritance diagram for quantier::synchronization::tests::mocks::MockISyncStateStorage:



Collaboration diagram for quantier::synchronization::tests::mocks::MockISyncStateStorage:



Public Member Functions

- **MOCK_METHOD** (ISyncStatePtr, getSyncState,(const Account &account),(override))
- **MOCK_METHOD** (void, setSyncState,(const Account &account, ISyncStatePtr syncState),(override))

Public Member Functions inherited from [quentier::synchronization::ISyncStateStorage](#)

- virtual `ISyncStatePtr` **getSyncState** (const [Account](#) &account)=0
- virtual void **setSyncState** (const [Account](#) &account, `ISyncStatePtr` syncState)=0

Additional Inherited Members

Signals inherited from [quentier::synchronization::ISyncStateStorage](#)

- void [notifySyncStateUpdated](#) ([Account](#) account, `ISyncStatePtr` syncState)

Protected Member Functions inherited from [quentier::synchronization::ISyncStateStorage](#)

- `ISyncStateStorage` (`QObject` *parent=nullptr)

5.70 [quentier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine](#)< T > Struct Template Reference

The [MoveMine](#) conflict resolution means "before using theirs version change mine version as specified". Note: the data item inside this conflict resolution might refer to something different than mine version passed to the conflict resolution function. It can be that way because the actual conflict might be with another local data item instead of the passed one.

```
#include <ISyncConflictResolver.h>
```

Public Types

- using `value_type` = T

Public Attributes

- T [mine](#)

5.70.1 Detailed Description

```
template<class T>
struct quentier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T >
```

The [MoveMine](#) conflict resolution means "before using theirs version change mine version as specified". Note: the data item inside this conflict resolution might refer to something different than mine version passed to the conflict resolution function. It can be that way because the actual conflict might be with another local data item instead of the passed one.

5.70.2 Member Data Documentation

5.70.2.1 mine

```
template<class T>
T quantier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T >::mine
```

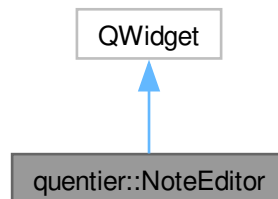
The changed value of mine data item.

5.71 quantier::NoteEditor Class Reference

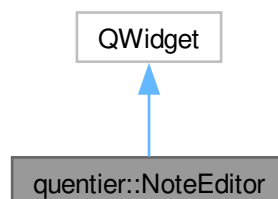
The [NoteEditor](#) class is a widget encapsulating all the functionality necessary for showing and editing notes.

```
#include <NoteEditor.h>
```

Inheritance diagram for quantier::NoteEditor:



Collaboration diagram for quantier::NoteEditor:



Public Slots

- void [convertToNote](#) ()
- void [saveNoteToLocalStorage](#) ()
- void [setNoteTitle](#) (const QString ¬eTitle)
- void [setTagIds](#) (const QStringList &tagLocalIds, const QStringList &tagGuids)
- void **undo** ()
- void **redo** ()
- void **cut** ()
- void **copy** ()
- void **paste** ()
- void **pasteUnformatted** ()
- void **selectAll** ()
- void **formatSelectionAsSourceCode** ()
- void **fontMenu** ()
- void **textBold** ()
- void **textItalic** ()
- void **textUnderline** ()
- void **textStrikethrough** ()
- void **textHighlight** ()
- void **alignLeft** ()
- void **alignCenter** ()
- void **alignRight** ()
- void **alignFull** ()
- void **findNext** (const QString &text, bool matchCase) const
- void **findPrevious** (const QString &text, bool matchCase) const
- void **replace** (const QString &textToReplace, const QString &replacementText, bool matchCase)
- void **replaceAll** (const QString &textToReplace, const QString &replacementText, bool matchCase)
- void **insertToDoCheckbox** ()
- void **insertInAppNoteLink** (const QString &userId, const QString &shardId, const QString ¬eGuid, const QString &linkText)
- void **setSpellcheck** (bool enabled)
- void **setFont** (const QFont &font)
- void **setFontHeight** (int height)
- void **setFontColor** (const QColor &color)
- void **setBackgroundColor** (const QColor &color)
- void [setDefaultPalette](#) (const QPalette &pal)
- void [setDefaultFont](#) (const QFont &font)
- void **insertHorizontalLine** ()
- void **increaseFontSize** ()
- void **decreaseFontSize** ()
- void **increaseIndentation** ()
- void **decreaseIndentation** ()
- void **insertBulletedList** ()
- void **insertNumberedList** ()
- void **insertTableDialog** ()
- void **insertFixedWidthTable** (int rows, int columns, int widthInPixels)
- void **insertRelativeWidthTable** (int rows, int columns, double relativeWidth)
- void **insertTableRow** ()
- void **insertTableColumn** ()
- void **removeTableRow** ()
- void **removeTableColumn** ()
- void **addAttachmentDialog** ()
- void **saveAttachmentDialog** (const QByteArray &resourceHash)
- void **saveAttachmentUnderCursor** ()

- void **openAttachment** (const QByteArray &resourceHash)
- void **openAttachmentUnderCursor** ()
- void **copyAttachment** (const QByteArray &resourceHash)
- void **copyAttachmentUnderCursor** ()
- void **encryptSelectedText** ()
- void **decryptEncryptedTextUnderCursor** ()
- void **editHyperlinkDialog** ()
- void **copyHyperlink** ()
- void **removeHyperlink** ()
- void **onNoteLoadCancelled** ()

Signals

- void **contentChanged** ()
contentChanged signal is emitted when the note's content (text) gets modified via manual editing (i.e. not any action like paste or cut)
- void **noteAndNotebookFoundInLocalStorage** (qevercloud::Note note, qevercloud::Notebook notebook)
noteAndNotebookFoundInLocalStorage signal is emitted when note and its corresponding notebook were found within the local storage right before the note editor starts to load the note into the editor
- void **noteNotFound** (QString noteLocalId)
noteNotFound signal is emitted when the note could not be found within the local storage by the provided local id
- void **noteDeleted** (QString noteLocalId)
noteDeleted signal is emitted when the note displayed within the note editor is deleted. The note editor stops displaying the note in this case shortly after emitting this signal
- void **noteModified** ()
noteModified signal is emitted when the note's content within the editor gets modified via some way - either via manual editing or via some action (like paste or cut)
- void **notifyError** ([ErrorString](#) error)
notifyError signal is emitted when [NoteEditor](#) encounters some problem worth letting the user to know about
- void **inAppNoteLinkClicked** (QString userId, QString shardId, QString noteGuid)
inAppNoteLinkClicked signal is emitted when the in-app note link is clicked within the note editor
- void **inAppNoteLinkPasteRequested** (QString url, QString userId, QString shardId, QString noteGuid)
- void **convertedToNote** (qevercloud::Note note)
- void **cantConvertToNote** ([ErrorString](#) error)
- void **noteEditorHtmlUpdated** (QString html)
- void **currentNoteChanged** (qevercloud::Note note)
- void **spellCheckerNotReady** ()
- void **spellCheckerReady** ()
- void **noteLoaded** ()
- void **noteSavedToLocalStorage** (QString noteLocalId)
noteSavedToLocalStorage signal is emitted when the note has been saved within the local storage. [NoteEditor](#) doesn't do this on its own unless it's explicitly asked to do this via invoking its `saveNoteToLocalStorage` slot
- void **failedToSaveNoteToLocalStorage** ([ErrorString](#) errorDescription, QString noteLocalId)
failedToSaveNoteToLocalStorage signal is emitted in case of failure to save the note to local storage
- void **textBoldState** (bool state)
- void **textItalicState** (bool state)
- void **textUnderlineState** (bool state)
- void **textStrikethroughState** (bool state)
- void **textAlignLeftState** (bool state)
- void **textAlignCenterState** (bool state)
- void **textAlignRightState** (bool state)
- void **textAlignFullState** (bool state)
- void **textInsideOrderedListState** (bool state)

- void **textInsideUnorderedListState** (bool state)
- void **textInsideTableState** (bool state)
- void **textFontFamilyChanged** (QString fontFamily)
- void **textFontSizeChanged** (int fontSize)
- void **insertTableDialogRequested** ()

Public Member Functions

- **NoteEditor** (QWidget *parent=nullptr, Qt::WindowFlags flags={})
- void **initialize** (local_storage::ILocalStoragePtr localStorage, [SpellChecker](#) &spellChecker, const [Account](#) &account, QThread *backgroundJobsThread=nullptr, enml::IDecryptedTextCachePtr decryptedTextCache=nullptr)
- [INoteEditorBackend](#) * **backend** () noexcept
- void **setBackend** ([INoteEditorBackend](#) *backend)
- void **setAccount** (const [Account](#) &account)
- const QUndoStack * **undoStack** () const noexcept
- void **setUndoStack** (QUndoStack *pUndoStack)
- void **setInitialPageHtml** (const QString &html)
- void **setNoteNotFoundPageHtml** (const QString &html)
- void **setNoteDeletedPageHtml** (const QString &html)
- void **setNoteLoadingPageHtml** (const QString &html)
- QString **currentNoteLocalId** () const
- void **setCurrentNoteLocalId** (const QString ¬eLocalId)
- void **clear** ()
- bool **isModified** () const noexcept
- bool **isEditorPageModified** () const noexcept
- bool **isNoteLoaded** () const noexcept
- qint64 **idleTime** () const noexcept
- void **setFocus** ()
- QString **selectedText** () const noexcept
- bool **hasSelection** () const noexcept
- bool **spellCheckEnabled** () const noexcept
- bool **print** (QPrinter &printer, [ErrorString](#) &errorDescription)
- bool **exportToPdf** (const QString &absoluteFilePath, [ErrorString](#) &errorDescription)
- bool **exportToEnex** (const QStringList &tagNames, QString &enex, [ErrorString](#) &errorDescription)
- QPalette **defaultPalette** () const
- const QFont * **defaultFont** () const

Protected Member Functions

- void **dragMoveEvent** (QDragMoveEvent *pEvent) override
- void **dropEvent** (QDropEvent *pEvent) override

5.71.1 Detailed Description

The [NoteEditor](#) class is a widget encapsulating all the functionality necessary for showing and editing notes.

5.71.2 Member Function Documentation

5.71.2.1 backend()

```
InNoteEditorBackend * quentier::NoteEditor::backend () [nodiscard], [noexcept]
```

Returns

the pointer to the note editor's backend

5.71.2.2 clear()

```
void quentier::NoteEditor::clear ()
```

Clear the contents of the note editor

5.71.2.3 convertToNote

```
void quentier::NoteEditor::convertToNote () [slot]
```

Invoke this slot to launch the asynchronous procedure of converting the current contents of the note editor to note; the convertedToNote signal would be emitted in response when the conversion is done

5.71.2.4 currentNoteLocalId()

```
QString quentier::NoteEditor::currentNoteLocalId () const [nodiscard]
```

Get the local id of the note currently set to the note editor

5.71.2.5 defaultFont()

```
const QFont * quentier::NoteEditor::defaultFont () const [nodiscard]
```

Returns

pointer to the default font used by the note editor; if no such font was set to the editor previously, returns null pointer

5.71.2.6 defaultPalette()

```
QPalette quentier::NoteEditor::defaultPalette () const [nodiscard]
```

Returns

palette containing default colors used by the editor; the palette is composed of colors from note editor widget's native palette but some of them might be overridden by colors from the palette specified previously via set↔DefaultPalette method: those colors from the specified palette which were valid

5.71.2.7 idleTime()

```
qint64 quentier::NoteEditor::idleTime () const [nodiscard], [noexcept]
```

Returns

the number of milliseconds since the last user's interaction with the note editor or -1 if there was no interaction or if no note is loaded at the moment

5.71.2.8 inAppNoteLinkPasteRequested

```
void quentier::NoteEditor::inAppNoteLinkPasteRequested (
    QString url,
    QString userId,
    QString shardId,
    QString noteGuid) [signal]
```

inAppNoteLinkPasteRequested signal is emitted when the note editor detects the attempt to paste the in-app note link into the note editor; the link would not be inserted right away, instead this signal would be emitted. Whatever party managing the note editor is expected to connect some slot to this signal and provide the optionally amended link information to the note editor by sending the signal connected to its insertInAppNoteLink slot - this slot accepts both the URL of the link and the link text and performs the actual link insertion into the note. If the link text is empty, the URL itself is used as the link text.

5.71.2.9 initialize()

```
void quentier::NoteEditor::initialize (
    local_storage::ILocalStoragePtr localStorage,
    SpellChecker & spellChecker,
    const Account & account,
    QThread * backgroundJobsThread = nullptr,
    enml::IDecryptedTextCachePtr decryptedTextCache = nullptr)
```

[NoteEditor](#) requires [LocalStorageManagerAsync](#), [SpellChecker](#) and [Account](#) for its work but due to the particularities of Qt's .ui files processing these can't be passed right inside the constructor, hence here's a special initialization method

Parameters

<i>localStorage</i>	Local storage
<i>spellChecker</i>	Spell checker to be used by note editor
<i>account</i>	Current account
<i>backgroundJobsThread</i>	Pointer to the thread to be used for scheduling of background jobs of NoteEditor ; if null, NoteEditor 's background jobs would take place in GUI thread
<i>decryptedTextCache</i>	Decrypted text cache to be used for note rendering and text decryption. If null, a new instance would be created and used.

5.71.2.10 isEditorPageModified()

```
bool quantier::NoteEditor::isEditorPageModified () const [nodiscard], [noexcept]
```

Returns

true if there's content within the editor not yet converted to note, false otherwise

5.71.2.11 isModified()

```
bool quantier::NoteEditor::isModified () const [nodiscard], [noexcept]
```

Returns

true if there's content within the editor not yet converted to note or not saved to local storage, false otherwise

5.71.2.12 isNoteLoaded()

```
bool quantier::NoteEditor::isNoteLoaded () const [nodiscard], [noexcept]
```

Returns

true if the note last set to the editor has been fully loaded already, false otherwise

5.71.2.13 saveNoteToLocalStorage

```
void quantier::NoteEditor::saveNoteToLocalStorage () [slot]
```

Invoke this slot to launch the asynchronous procedure of saving the modified current note back to the local storage. If no note is set to the editor or if the note is not modified, no action would be performed. Otherwise noteSaved↔ToLocalStorage signal would be emitted in case of successful saving or failedToSaveNoteToLocalStorage would be emitted otherwise

5.71.2.14 setAccount()

```
void quantier::NoteEditor::setAccount (
    const Account & account)
```

Set the current account to the note editor

5.71.2.15 setBackend()

```
void quantier::NoteEditor::setBackend (
    INoteEditorBackend * backend)
```

This method can be used to set the backend to the note editor; the note editor has the default backend so this method is not obligatory to be called

5.71.2.16 setCurrentNoteLocalId()

```
void quantier::NoteEditor::setCurrentNoteLocalId (  
    const QString & noteLocalId)
```

Set note local id to the note editor. The note is being searched for within the local storage, in case of no note being found noteNotFound signal is emitted. Otherwise note editor page starts loading.

Parameters

<i>note↔ LocalId</i>	The local id of note
--------------------------	----------------------

5.71.2.17 setDefaultFont

```
void quantier::NoteEditor::setDefaultFont (  
    const QFont & font) [slot]
```

Sets the font which would be used by the editor by default

Parameters

<i>font</i>	The font to be used by the editor by default
-------------	--

5.71.2.18 setDefaultPalette

```
void quantier::NoteEditor::setDefaultPalette (  
    const QPalette & pal) [slot]
```

Sets the palette with colors to be used by the editor. New colors are applied after the note is fully loaded. If no note is set to the editor, the palette is simply remembered for the next note to be loaded into it.

Colors within the palette and their usage:

1. WindowText - used as default font color
2. Base - used as default background color
3. HighlightedText - used as font color for selected text
4. Highlight - used as background color for selected text

Parameters

<i>pal</i>	The palette to be set. Invalid colors from it are substituted by colors from widget's palette by the editor
------------	---

5.71.2.19 setFocus()

```
void quotient::NoteEditor::setFocus ()
```

Sets the focus to the backend note editor widget

5.71.2.20 setInitialPageHtml()

```
void quotient::NoteEditor::setInitialPageHtml (  
    const QString & html)
```

Set the html to be displayed when the note is not set to the editor

5.71.2.21 setNoteDeletedPageHtml()

```
void quotient::NoteEditor::setNoteDeletedPageHtml (  
    const QString & html)
```

Set the html to be displayed when the note set to the editor was deleted from the local storage (either marked as deleted or deleted permanently i.e. expunged)

5.71.2.22 setNoteLoadingPageHtml()

```
void quotient::NoteEditor::setNoteLoadingPageHtml (  
    const QString & html)
```

Set the html to be displayed when the note set to the editor is being loaded into it

5.71.2.23 setNoteNotFoundPageHtml()

```
void quotient::NoteEditor::setNoteNotFoundPageHtml (  
    const QString & html)
```

Set the html to be displayed when the note attempted to be set to the editor was not found within the local storage

5.71.2.24 setNoteTitle

```
void quotient::NoteEditor::setNoteTitle (  
    const QString & noteTitle) [slot]
```

Invoke this slot to set the title to the note displayed via the note editor. The note editor itself doesn't manage the note title in any way so any external code using the note editor can set the title to the note editor's note which would be considered modified if the title is new and then eventually the note would be saved to local storage

Parameters

<i>noteTitle</i>	The title of the note
------------------	-----------------------

5.71.2.25 setTagIds

```
void quantier::NoteEditor::setTagIds (
    const QStringList & tagLocalIds,
    const QStringList & tagGuids) [slot]
```

Invoke this slot to set tag local ids and/or tag guids to the note displayed via the note editor. The note editor itself doesn't manage the note tags in any way so any external code using the note editor can set the tag ids to the note editor's internal note which would be considered modified if the tag ids are new and then eventually the note would be saved to local storage

Parameters

<i>tagLocalIds</i>	The list of tag local ids for the note
<i>tagGuids</i>	The list of tag guids for the note

5.71.2.26 setUndoStack()

```
void quantier::NoteEditor::setUndoStack (
    QUndoStack * pUndoStack)
```

Set the undo stack for the note editor to use

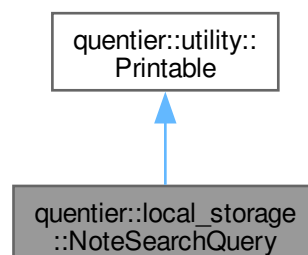
5.71.2.27 undoStack()

```
const QUndoStack * quantier::NoteEditor::undoStack () const [nodiscard], [noexcept]
```

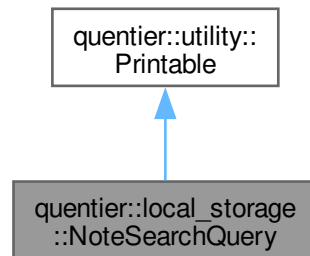
Get the undo stack serving to the note editor

5.72 quantier::local_storage::NoteSearchQuery Class Reference

Inheritance diagram for quantier::local_storage::NoteSearchQuery:



Collaboration diagram for quantier::local_storage::NoteSearchQuery:



Public Member Functions

- **NoteSearchQuery** (const NoteSearchQuery &other)
- **NoteSearchQuery** (NoteSearchQuery &&other) noexcept
- NoteSearchQuery & **operator=** (const NoteSearchQuery &other)
- NoteSearchQuery & **operator=** (NoteSearchQuery &&other) noexcept
- bool **isEmpty** () const
- void **clear** ()
- QString **queryString** () const
- bool **setQueryString** (const QString &queryString, [ErrorString](#) &error)
- QString **notebookModifier** () const
- bool **hasAnyModifier** () const
- const QStringList & **tagNames** () const
- const QStringList & **negatedTagNames** () const
- bool **hasAnyTag** () const
- bool **hasNegatedAnyTag** () const
- const QStringList & **titleNames** () const
- const QStringList & **negatedTitleNames** () const
- bool **hasAnyTitleName** () const
- bool **hasNegatedAnyTitleName** () const
- const QList< qint64 > & **creationTimestamps** () const
- const QList< qint64 > & **negatedCreationTimestamps** () const
- bool **hasAnyCreationTimestamp** () const
- bool **hasNegatedAnyCreationTimestamp** () const
- const QList< qint64 > & **modificationTimestamps** () const
- const QList< qint64 > & **negatedModificationTimestamps** () const
- bool **hasAnyModificationTimestamp** () const
- bool **hasNegatedAnyModificationTimestamp** () const
- const QStringList & **resourceMimeTypes** () const
- const QStringList & **negatedResourceMimeTypes** () const
- bool **hasAnyResourceMimeType** () const
- bool **hasNegatedAnyResourceMimeType** () const
- const QList< qint64 > & **subjectDateTimestamps** () const
- const QList< qint64 > & **negatedSubjectDateTimestamps** () const
- bool **hasAnySubjectDateTimestamp** () const
- bool **hasNegatedAnySubjectDateTimestamp** () const

- `const QList< double > & latitudes () const`
- `const QList< double > & negatedLatitudes () const`
- `bool hasAnyLatitude () const`
- `bool hasNegatedAnyLatitude () const`
- `const QList< double > & longitudes () const`
- `const QList< double > & negatedLongitudes () const`
- `bool hasAnyLongitude () const`
- `bool hasNegatedAnyLongitude () const`
- `const QList< double > & altitudes () const`
- `const QList< double > & negatedAltitudes () const`
- `bool hasAnyAltitude () const`
- `bool hasNegatedAnyAltitude () const`
- `const QStringList & authors () const`
- `const QStringList & negatedAuthors () const`
- `bool hasAnyAuthor () const`
- `bool hasNegatedAnyAuthor () const`
- `const QStringList & sources () const`
- `const QStringList & negatedSources () const`
- `bool hasAnySource () const`
- `bool hasNegatedAnySource () const`
- `const QStringList & sourceApplications () const`
- `const QStringList & negatedSourceApplications () const`
- `bool hasAnySourceApplication () const`
- `bool hasNegatedAnySourceApplication () const`
- `const QStringList & contentClasses () const`
- `const QStringList & negatedContentClasses () const`
- `bool hasAnyContentClass () const`
- `bool hasNegatedAnyContentClass () const`
- `const QStringList & placeNames () const`
- `const QStringList & negatedPlaceNames () const`
- `bool hasAnyPlaceName () const`
- `bool hasNegatedAnyPlaceName () const`
- `const QStringList & applicationData () const`
- `const QStringList & negatedApplicationData () const`
- `bool hasAnyApplicationData () const`
- `bool hasNegatedAnyApplicationData () const`
- `const QList< quint64 > & reminderOrders () const`
- `const QList< quint64 > & negatedReminderOrders () const`
- `bool hasAnyReminderOrder () const`
- `bool hasNegatedAnyReminderOrder () const`
- `const QList< quint64 > & reminderTimes () const`
- `const QList< quint64 > & negatedReminderTimes () const`
- `bool hasAnyReminderTime () const`
- `bool hasNegatedAnyReminderTime () const`
- `const QList< quint64 > & reminderDoneTimes () const`
- `const QList< quint64 > & negatedReminderDoneTimes () const`
- `bool hasAnyReminderDoneTime () const`
- `bool hasNegatedAnyReminderDoneTime () const`
- `bool hasUnfinishedToDo () const`
- `bool hasNegatedUnfinishedToDo () const`
- `bool hasFinishedToDo () const`
- `bool hasNegatedFinishedToDo () const`
- `bool hasAnyToDo () const`
- `bool hasNegatedAnyToDo () const`
- `bool hasEncryption () const`

- `bool hasNegatedEncryption () const`
- `const QStringList & contentSearchTerms () const`
- `const QStringList & negatedContentSearchTerms () const`
- `bool hasAnyContentSearchTerms () const`
- `bool isMatcheable () const`
- `QTextStream & print (QTextStream &strm) const` override

Public Member Functions inherited from `quentier::utility::Printable`

- `QString toString () const`

5.72.1 Member Function Documentation

5.72.1.1 `notebookModifier()`

```
QString quentier::local_storage::NoteSearchQuery::notebookModifier () const [nodiscard]
```

If query string has "notebook:<notebook name>" scope modifier, this method returns the name of the notebook, otherwise it returns empty string

5.72.1.2 `print()`

```
QTextStream & quentier::local_storage::NoteSearchQuery::print (  
    QTextStream & strm) const [override], [virtual]
```

Implements `quentier::utility::Printable`.

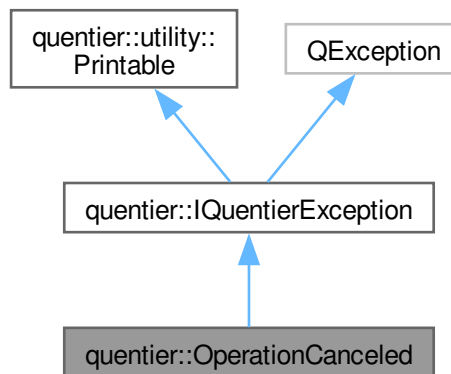
5.72.1.3 `queryString()`

```
QString quentier::local_storage::NoteSearchQuery::queryString () const [nodiscard]
```

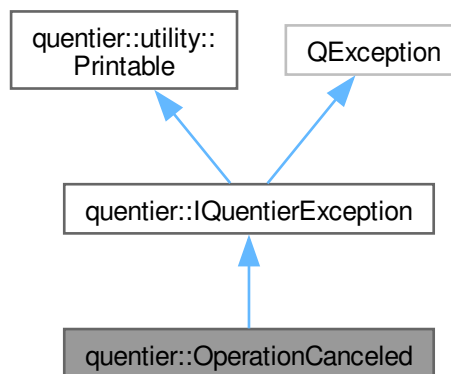
Returns the original non-parsed query string

5.73 `quentier::OperationCanceled` Class Reference

Inheritance diagram for `quentier::OperationCanceled`:



Collaboration diagram for `quentier::OperationCanceled`:



Public Member Functions

- `OperationCanceled * clone ()` const override
- `void raise ()` const override

Public Member Functions inherited from `quentier::IQuentierException`

- `ErrorString errorMessage ()` const
- `QString localizedErrorMessage ()` const
- `QString nonLocalizedErrorMessage ()` const
- `const char * what ()` const noexcept override
- `QTextStream & print (QTextStream &strm)` const override

Public Member Functions inherited from `quentier::utility::Printable`

- `QString toString () const`

Protected Member Functions

- `QString exceptionDisplayName () const override`

Protected Member Functions inherited from `quentier::IQuentierException`

- `IQuentierException (ErrorString message)`
- `IQuentierException (const IQuentierException &other)`
- `IQuentierException & operator= (const IQuentierException &other)`

5.73.1 Member Function Documentation

5.73.1.1 `exceptionDisplayName()`

```
QString quentier::OperationCanceled::exceptionDisplayName () const [nodiscard], [override],  
[protected], [virtual]
```

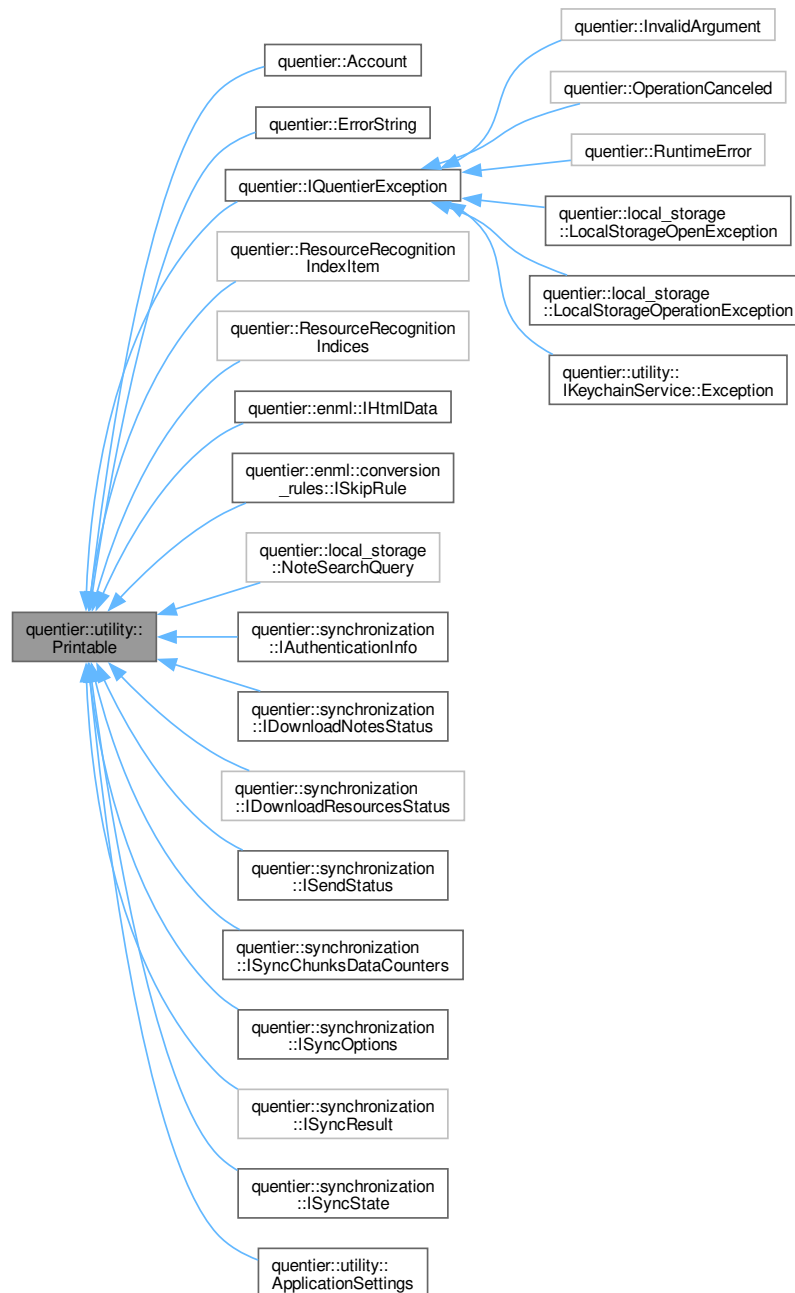
Implements `quentier::IQuentierException`.

5.74 `quentier::utility::Printable` Class Reference

The `Printable` class is the interface for Quentier's internal classes which should be able to write themselves into `QTextStream` and/or convert to `QString`.

```
#include <Printable.h>
```

Inheritance diagram for `quentier::utility::Printable`:



Public Member Functions

- virtual QTextStream & **print** (QTextStream &strm) const =0
- QString **toString** () const

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const Printable &printable)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &debug, const Printable &printable)

5.74.1 Detailed Description

The [Printable](#) class is the interface for Quentier's internal classes which should be able to write themselves into QTextStream and/or convert to QString.

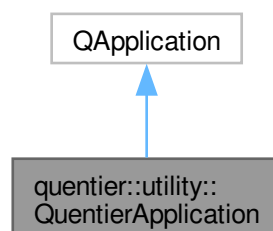
5.75 QPromise< T > Class Template Reference

Public Member Functions

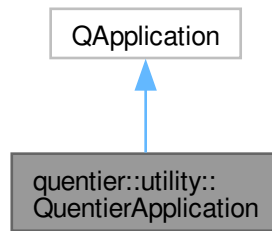
- **QPromise** ([QPromise](#)< T > &&other) noexcept
- **QPromise** (QFutureInterface< T > &other)
- [QPromise](#) & **operator=** ([QPromise](#)< T > &&other) noexcept
- QFuture< T > **future** () const
- template<typename U, typename = std::enable_if_t< std::is_same_v<U, T> || std::is_convertible_v<U, T>>>> void **addResult** (U &&result, int index=-1)
- void **setException** (const QException &e)
- void **start** ()
- void **finish** ()
- void **suspendIfRequested** ()
- bool **isCanceled** () const
- void **setProgressRange** (int minimum, int maximum)
- void **setProgressValue** (int progressValue)
- void **setProgressValueAndText** (int progressValue, const QString &progressText)
- void **swap** ([QPromise](#)< T > &other) noexcept

5.76 quentier::utility::QuentierApplication Class Reference

Inheritance diagram for quentier::utility::QuentierApplication:



Collaboration diagram for `quentier::utility::QuentierApplication`:



Public Member Functions

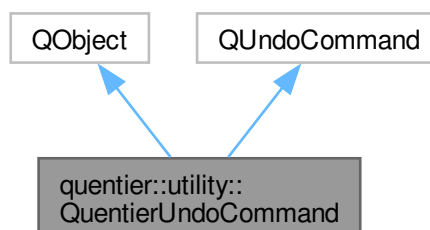
- **QuentierApplication** (int &argc, char *argv[])
- bool **notify** (QObject *object, QEvent *event) override
- bool **event** (QEvent *event) override

5.77 quentier::utility::QuentierUndoCommand Class Reference

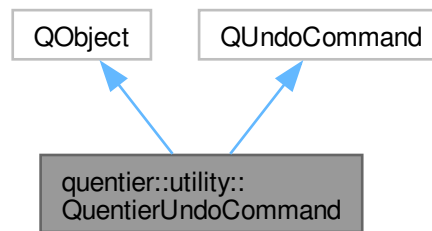
The [QuentierUndoCommand](#) class has the sole purpose of working around one quirky aspect of Qt's undo/redo framework: when you push `QUndoCommand` to `QUndoStack`, it calls "redo" method of that command. This class offers subclasses to implement their own methods for actual "undo" and "redo" commands while ignoring the attempts to "redo" anything if there were no previous "undo" call prior to that.

```
#include <QuentierUndoCommand.h>
```

Inheritance diagram for `quentier::utility::QuentierUndoCommand`:



Collaboration diagram for `quentier::utility::QuentierUndoCommand`:



Signals

- void **notifyError** ([ErrorString](#) error)

Public Member Functions

- **QuentierUndoCommand** (QUndoCommand *parent=nullptr)
- **QuentierUndoCommand** (const QString &text, QUndoCommand *parent=nullptr)
- void **undo** () final
- void **redo** () final
- bool **onceUndoExecuted** () const noexcept

Protected Member Functions

- virtual void **undoImpl** ()=0
- virtual void **redoImpl** ()=0

5.77.1 Detailed Description

The [QuentierUndoCommand](#) class has the sole purpose of working around one quirky aspect of Qt's undo/redo framework: when you push QUndoCommand to QUndoStack, it calls "redo" method of that command. This class offers subclasses to implement their own methods for actual "undo" and "redo" commands while ignoring the attempts to "redo" anything if there were no previous "undo" call prior to that.

The rationale behind the current behaviour seems to be the compliance with "command pattern behaviour" when you create the command to execute the action instead of just executing it immediately. This design is enforced by Qt's undo/redo framework, there's no option to choose not to call "redo" when pushing to the stack.

One thing which this design fails to see is the fact that the command may be already executed externally by the moment the QUndoCommand can be created. Suppose we can get the information about how to undo (and then again redo) that command. We create the corresponding QUndoCommand, set up the stuff for its undo/redo methods and push it to QUndoStack for future use... But at the same time QUndoStack calls "redo" method of the command. Really not the behaviour you'd like to have.

[QuentierUndoCommand](#) is also QObject, it is for error reporting via notifyError signal

5.78 `quentier::synchronization::RateLimitReachedError` Struct Reference

```
#include <Errors.h>
```

Public Attributes

- `std::optional<qint32>` [rateLimitDurationSec](#)

5.78.1 Detailed Description

Information about "API rate limit reached" error which Evernote servers might return if too much of their API calls were made recently. In case of such error synchronization should be repeated later, after some time passes.

5.78.2 Member Data Documentation

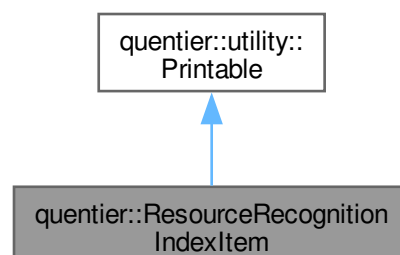
5.78.2.1 `rateLimitDurationSec`

```
std::optional<qint32> quentier::synchronization::RateLimitReachedError::rateLimitDurationSec
```

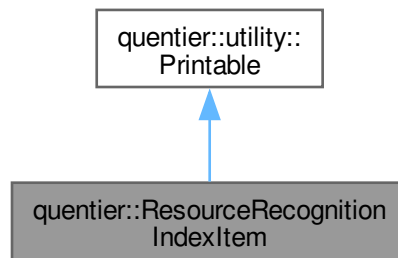
Number of seconds during which since the current moment during which any call to Evernote API would again result in "API rate limit reached" error i.e. the number of seconds to wait for before the next attempt to run synchronization

5.79 `quentier::ResourceRecognitionIndexItem` Class Reference

Inheritance diagram for `quentier::ResourceRecognitionIndexItem`:



Collaboration diagram for `quentier::ResourceRecognitionIndexItem`:



Classes

- struct [ITextItem](#)
- struct [IObjectItem](#)
- struct [IShapeItem](#)
- struct [IBarcodeItem](#)

Public Types

- using **ITextItemPtr** = `std::shared_ptr<ITextItem>`
- using **IObjectItemPtr** = `std::shared_ptr<IObjectItem>`
- using **IShapeItemPtr** = `std::shared_ptr<IShapeItem>`
- using **IBarcodeItemPtr** = `std::shared_ptr<IBarcodeItem>`

Public Member Functions

- **ResourceRecognitionIndexItem** (`const ResourceRecognitionIndexItem &other`)
- **ResourceRecognitionIndexItem** (`ResourceRecognitionIndexItem &&other`) `noexcept`
- `ResourceRecognitionIndexItem & operator=` (`const ResourceRecognitionIndexItem &other`)
- `ResourceRecognitionIndexItem & operator=` (`ResourceRecognitionIndexItem &&other`) `noexcept`
- `bool isValid` () `const`
- `int x` () `const`
- `void setX` (`int x`)
- `int y` () `const`
- `void setY` (`int y`)
- `int h` () `const`
- `void setH` (`int h`)
- `int w` () `const`
- `void setW` (`int w`)
- `int offset` () `const`
- `void setOffset` (`int offset`)
- `int duration` () `const`
- `void setDuration` (`int duration`)
- `QList< int > strokes` () `const`
- `void setStrokes` (`QList< int > strokes`)

- `QList< ITextItemPtr > textItems () const`
- `void setTextItems (QList< ITextItemPtr > textItems)`
- `QList< IObjectItemPtr > objectItems () const`
- `void setObjectItems (QList< IObjectItemPtr > objectItems)`
- `QList< IShapeItemPtr > shapeItems () const`
- `void setShapeItems (QList< IShapeItemPtr > shapeItems)`
- `QList< IBarcodeItemPtr > barcodeItems () const`
- `void setBarcodeItems (QList< IBarcodeItemPtr > barcodeItems)`
- `QTextStream & print (QTextStream &strm) const override`

Public Member Functions inherited from [quentier::utility::Printable](#)

- `QString toString () const`

5.79.1 Member Function Documentation

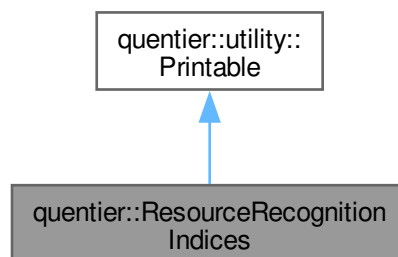
5.79.1.1 `print()`

```
QTextStream & quentier::ResourceRecognitionIndexItem::print (
    QTextStream & strm) const    [override], [virtual]
```

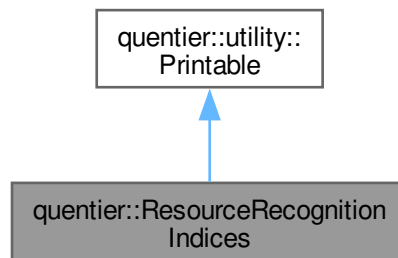
Implements [quentier::utility::Printable](#).

5.80 `quentier::ResourceRecognitionIndices` Class Reference

Inheritance diagram for `quentier::ResourceRecognitionIndices`:



Collaboration diagram for `quentier::ResourceRecognitionIndices`:



Public Member Functions

- **ResourceRecognitionIndices** (const QByteArray &rawRecognitionIndicesData)
- **ResourceRecognitionIndices** (const ResourceRecognitionIndices &other)
- **ResourceRecognitionIndices** (ResourceRecognitionIndices &&other) noexcept
- ResourceRecognitionIndices & **operator=** (const ResourceRecognitionIndices &other)
- ResourceRecognitionIndices & **operator=** (ResourceRecognitionIndices &&other) noexcept
- bool **isNull** () const
- bool **isValid** () const
- QString **objectId** () const
- QString **objectType** () const
- QString **recoType** () const
- QString **engineVersion** () const
- QString **docType** () const
- QString **lang** () const
- int **objectHeight** () const
- int **objectWidth** () const
- QVector< [ResourceRecognitionIndexItem](#) > **items** () const
- bool **setData** (const QByteArray &rawRecognitionIndicesData)
- QTextStream & [print](#) (QTextStream &strm) const override

Public Member Functions inherited from [quentier::utility::Printable](#)

- QString **toString** () const

5.80.1 Member Function Documentation

5.80.1.1 `print()`

```
QTextStream & quentier::ResourceRecognitionIndices::print (
    QTextStream & strm) const [override], [virtual]
```

Implements [quentier::utility::Printable](#).

5.81 `quantier::Result< ValueType, ErrorType, typename >` Class Template Reference

Public Member Functions

- `template<typename T1 = ValueType, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr> Result (T1 t)`
- **Result** (ErrorType error)
- **Result** (const Result< ValueType, ErrorType > &other)
- **Result** (Result< ValueType, ErrorType > &&other)
- Result & **operator=** (const Result< ValueType, ErrorType > &other)
- Result & **operator=** (Result< ValueType, ErrorType > &&other)
- bool **isValid** () const noexcept
- **operator bool** () const noexcept
- `template<typename T1 = ValueType, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr> T1 & get ()`
- `template<typename T1 = ValueType, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr> const T1 & get () const`
- `template<typename T1 = ValueType, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr> T1 & operator* ()`
- `template<typename T1 = ValueType, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr> const T1 & operator* () const`
- const ErrorType & **error** () const
- ErrorType & **error** ()

5.81.1 Member Function Documentation

5.81.1.1 `isValid()`

```
template<class ValueType, class ErrorType, typename = typename std::enable_if_t<!std::is_↵
void_v<std::decay_t<ErrorType>>>>
bool quantier::Result< ValueType, ErrorType, typename >::isValid () const [inline], [nodiscard],
[noexcept]
```

Returns

boolean value indicating whether the result contains a value

5.82 `quantier::threading::detail::ResultTypeHelper< F, Arg, Enable >` Struct Template Reference

5.83 `quantier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< Arg > > > >` Struct Template Reference

Public Types

- using **ResultType** = std::invoke_result_t<std::decay_t<F>, std::decay_t<Arg>>

5.84 `quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< Arg > > > > Struct Template Reference`

Public Types

- using **ResultType** = `std::invoke_result_t<std::decay_t<F>, QFuture<Arg>>`

5.85 `quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< void > > > > Struct Template Reference`

Public Types

- using **ResultType** = `std::invoke_result_t<std::decay_t<F>>`

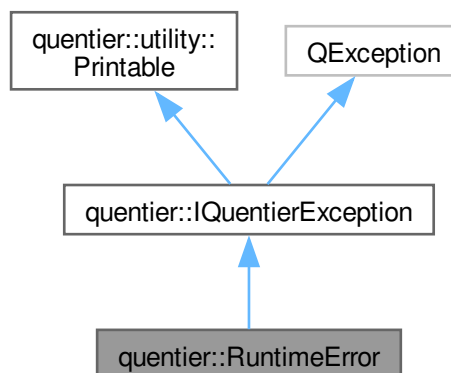
5.86 `quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< void > > > > Struct Template Reference`

Public Types

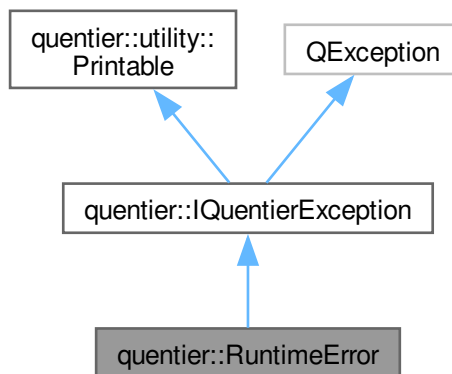
- using **ResultType** = `std::invoke_result_t<std::decay_t<F>, QFuture<void>>`

5.87 `quentier::RuntimeError` Class Reference

Inheritance diagram for `quentier::RuntimeError`:



Collaboration diagram for `quentier::RuntimeError`:



Public Member Functions

- **RuntimeError** ([ErrorString](#) message)
- `RuntimeError * clone ()` const override
- `void raise ()` const override

Public Member Functions inherited from [quentier::IQuentierException](#)

- [ErrorString](#) **errorMessage** () const
- `QString localizedErrorMessage ()` const
- `QString nonLocalizedErrorMessage ()` const
- `const char * what ()` const noexcept override
- `QTextStream & print (QTextStream &strm)` const override

Public Member Functions inherited from [quentier::utility::Printable](#)

- `QString toString ()` const

Protected Member Functions

- `QString exceptionDisplayName ()` const override

Protected Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** ([ErrorString](#) message)
- **IQuentierException** (const IQuentierException &other)
- IQuentierException & **operator=** (const IQuentierException &other)

5.87.1 Member Function Documentation

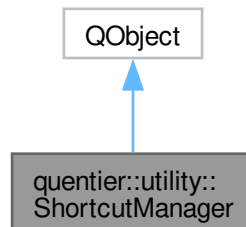
5.87.1.1 exceptionDisplayName()

```
QString quantier::RuntimeError::exceptionDisplayName () const [nodiscard], [override], [protected], [virtual]
```

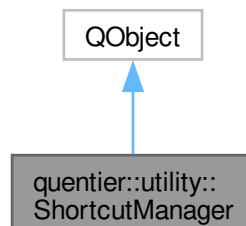
Implements [quantier::IQuantierException](#).

5.88 quantier::utility::ShortcutManager Class Reference

Inheritance diagram for quantier::utility::ShortcutManager:



Collaboration diagram for quantier::utility::ShortcutManager:



Public Types

- enum **QuentierShortcutKey** {
NewNote = 5000 , **NewTag** , **NewNotebook** , **NewSavedSearch** ,
AddAttachment , **SaveAttachment** , **OpenAttachment** , **CopyAttachment** ,
CutAttachment , **RemoveAttachment** , **RenameAttachment** , **AddAccount** ,
ExitAccount , **SwitchAccount** , **AccountInfo** , **NoteSearch** ,
NewNoteSearch , **ShowNotes** , **ShowNotebooks** , **ShowTags** ,
ShowSavedSearches , **ShowDeletedNotes** , **ShowStatusBar** , **ShowToolBar** ,
PasteUnformatted , **Font** , **UpperIndex** , **LowerIndex** ,
AlignLeft , **AlignCenter** , **AlignRight** , **AlignFull** ,
IncreaseIndentation , **DecreaseIndentation** , **IncreaseFontSize** , **DecreaseFontSize** ,
InsertNumberedList , **InsertBulletedList** , **Strikethrough** , **Highlight** ,
InsertTable , **InsertRow** , **InsertColumn** , **RemoveRow** ,
RemoveColumn , **InsertHorizontalLine** , **InsertToDoTag** , **EditHyperlink** ,
CopyHyperlink , **RemoveHyperlink** , **Encrypt** , **Decrypt** ,
DecryptPermanently , **BackupLocalStorage** , **RestoreLocalStorage** , **UpgradeLocalStorage** ,
LocalStorageStatus , **SpellCheck** , **SpellCheckIgnoreWord** , **SpellCheckAddWordToUserDictionary** ,
SavImage , **AnnotateImage** , **ImageRotateClockwise** , **ImageRotateCounterClockwise** ,
Synchronize , **FullSync** , **ImportFolders** , **Preferences** ,
ReleaseNotes , **ViewLogs** , **About** , **UnknownKey** = 100000 }

Public Slots

- void **setUserShortcut** (int key, const QKeySequence &[shortcut](#), const [Account](#) &account, QString context={})
- void **setNonStandardUserShortcut** (QString nonStandardKey, const QKeySequence &[shortcut](#), const [Account](#) &account, QString context={})
- void **setDefaultShortcut** (int key, const QKeySequence &[shortcut](#), const [Account](#) &account, QString context={})
- void **setNonStandardDefaultShortcut** (QString nonStandardKey, const QKeySequence &[shortcut](#), const [Account](#) &account, QString context={})

Signals

- void **shortcutChanged** (int key, QKeySequence [shortcut](#), const [Account](#) &account, QString context)
- void **nonStandardShortcutChanged** (QString nonStandardKey, QKeySequence [shortcut](#), const [Account](#) &account, QString context)

Public Member Functions

- **ShortcutManager** (QObject *parent=nullptr)
- QKeySequence [shortcut](#) (int key, const [Account](#) &account, const QString &context={}) const
- QKeySequence [shortcut](#) (const QString &nonStandardKey, const [Account](#) &account, const QString &context={}) const
- QKeySequence [defaultShortcut](#) (int key, const [Account](#) &account, const QString &context={}) const
- QKeySequence [defaultShortcut](#) (const QString &nonStandardKey, const [Account](#) &account, const QString &context={}) const
- QKeySequence [userShortcut](#) (int key, const [Account](#) &account, const QString &context={}) const
- QKeySequence [userShortcut](#) (const QString &nonStandardKey, const [Account](#) &account, const QString &context={}) const

5.88.1 Member Function Documentation

5.88.1.1 defaultShortcut() [1/2]

```
QKeySequence quentier::utility::ShortcutManager::defaultShortcut (
    const QString & nonStandardKey,
    const Account & account,
    const QString & context = {}) const [nodiscard]
```

Returns

Default shortcut for the non-standard key if present, otherwise empty key sequence

5.88.1.2 defaultShortcut() [2/2]

```
QKeySequence quentier::utility::ShortcutManager::defaultShortcut (
    int key,
    const Account & account,
    const QString & context = {}) const [nodiscard]
```

Returns

Default shortcut for the standard key if present, otherwise empty key sequence

5.88.1.3 shortcut() [1/2]

```
QKeySequence quentier::utility::ShortcutManager::shortcut (
    const QString & nonStandardKey,
    const Account & account,
    const QString & context = {}) const [nodiscard]
```

Returns

Active shortcut for the non-standard key - either the user defined shortcut (if present) or the default one (if present as well)

5.88.1.4 shortcut() [2/2]

```
QKeySequence quentier::utility::ShortcutManager::shortcut (
    int key,
    const Account & account,
    const QString & context = {}) const [nodiscard]
```

Returns

Active shortcut for the standard key - either the user defined shortcut (if present) or the default one (if present as well)

5.88.1.5 userShortcut() [1/2]

```
QKeySequence quentier::utility::ShortcutManager::userShortcut (  
    const QString & nonStandardKey,  
    const Account & account,  
    const QString & context = {}) const [nodiscard]
```

Returns

User defined shortcut for the non-standard key if present, otherwise empty key sequence

5.88.1.6 userShortcut() [2/2]

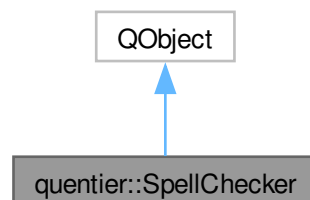
```
QKeySequence quentier::utility::ShortcutManager::userShortcut (  
    int key,  
    const Account & account,  
    const QString & context = {}) const [nodiscard]
```

Returns

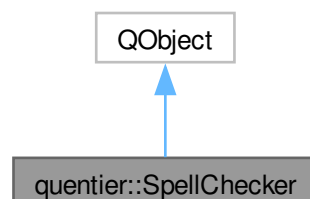
User defined shortcut for the standard key if present, otherwise empty key sequence

5.89 quentier::SpellChecker Class Reference

Inheritance diagram for quentier::SpellChecker:



Collaboration diagram for quentier::SpellChecker:



Signals

- void **ready** ()

Public Member Functions

- **SpellChecker** (`utility::FileIOProcessorAsync` *fileIOProcessorAsync, `Account` account, `QObject` *parent=nullptr, const `QString` &userDictionaryPath={})
- `QList< std::pair< QString, bool > >` **listAvailableDictionaries** () const
- void **setAccount** (const `Account` &account)
- void **enableDictionary** (const `QString` &language)
- void **disableDictionary** (const `QString` &language)
- bool **checkSpell** (const `QString` &word) const
- `QStringList` **spellCorrectionSuggestions** (const `QString` &misSpelledWord) const
- void **addToUserWordlist** (const `QString` &word)
- void **removeFromUserWordList** (const `QString` &word)
- void **ignoreWord** (const `QString` &word)
- void **removeWord** (const `QString` &word)
- bool **isReady** () const noexcept

5.90 `quentier::utility::StringUtils` Class Reference

Public Member Functions

- void **removePunctuation** (`QString` &str, const `QList< QChar >` &charactersToPreserve={}) const
- void **removeDiacritics** (`QString` &str) const
- void **removeNewlines** (`QString` &str) const

5.91 `quentier::utility::SysInfo` Class Reference

Public Member Functions

- qint64 **pageSize** ()
- qint64 **totalMemory** ()
- qint64 **freeMemory** ()
- `QString` **stackTrace** ()
- `QString` **platformName** ()

5.92 `quentier::threading::TrackedTask< LockableObject, Function >` Class Template Reference

```
#include <TrackedTask.h>
```

Public Member Functions

- `template<typename SomeLockableObject, typename SomeFunction>`
`constexpr TrackedTask (SomeLockableObject &&someLockableObject, SomeFunction &&function)`
- `template<typename... Arguments, typename = std::enable_if_t< std::is_invocable_v<Function, Arguments...> || std::is_member_↵`
`function_pointer_v<Function>>>`
`constexpr void operator() (Arguments &&... arguments)`
- `template<typename... Arguments, typename = std::enable_if_t< std::is_invocable_v<Function, Arguments...> || std::is_member_↵`
`function_pointer_v<Function>>>`
`constexpr void operator() (Arguments &&... arguments) const`

5.92.1 Detailed Description

`template<typename LockableObject, typename Function>`
class `quentier::threading::TrackedTask< LockableObject, Function >`

Wrapper class which automates checking for the state of a lockable object. With this class code like this

```
auto task = [selfWeak = weak_from_this()] { auto self = selfWeak.lock(); if (!self) { return; } // otherwise do something
};
```

can be written like this:

```
auto task = threading::TrackedTask{weak_from_this(), &MyClass::someMethod};
```

5.93 quentier::utility::UuidGenerator Class Reference

Static Public Member Functions

- static QString **generate** ()
- static QString **uidToString** (const QUuid &uid)

5.94 quentier::synchronization::ISyncConflictResolver::Conflict↵ Resolution::UseMine Struct Reference

The [UseMine](#) conflict resolution means "override theirs version with mine version".

```
#include <ISyncConflictResolver.h>
```

5.94.1 Detailed Description

The [UseMine](#) conflict resolution means "override theirs version with mine version".

5.95 quentier::synchronization::ISyncConflictResolver::Conflict↵ Resolution::UseTheirs Struct Reference

The [UseTheirs](#) conflict resolution means "override mine version with theirs version".

```
#include <ISyncConflictResolver.h>
```

5.95.1 Detailed Description

The [UseTheirs](#) conflict resolution means "override mine version with theirs version".

Chapter 6

File Documentation

6.1 ISkipRule.h

```
00001 /*
00002  * Copyright 2023-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/enml/conversion_rules/MatchMode.h>
00022 #include <quentier/utility/Printable.h>
00023
00024 #include <QtGlobal>
00025
00026 namespace quentier::enml::conversion_rules {
00027
00028     class ISkipRule : public utility::Printable
00029     {
00030     public:
00031         ~ISkipRule() override;
00032
00033         enum class Target
00034         {
00035             Element,
00036             AttributeName,
00037             AttributeValue
00038         };
00039
00040         friend QUENTIER_EXPORT QTextStream & operator<<(
00041             QTextStream & strm, Target target);
00042
00043         friend QUENTIER_EXPORT QDebug & operator<<(QDebug & dbg, Target target);
00044
00045         [[nodiscard]] virtual Target target() const = 0;
00046
00047         [[nodiscard]] virtual QString value() const = 0;
00048
00049         [[nodiscard]] virtual MatchMode matchMode() const = 0;
00050
00051         [[nodiscard]] virtual bool includeContents() const = 0;
00052
00053         [[nodiscard]] virtual Qt::CaseSensitivity caseSensitivity() const = 0;
00054
00055     public: // utility::Printable
00056         QTextStream & print(QTextStream & strm) const override;
00057     };
00058 } // namespace quentier::enml::conversion_rules
```

6.2 ISkipRuleBuilder.h

```

00001 /*
00002  * Copyright 2023 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/enml/conversion_rules/Fwd.h>
00022 #include <quentier/enml/conversion_rules/ISkipRule.h>
00023 #include <quentier/utility/Linkage.h>
00024
00025 namespace quentier::enml::conversion_rules {
00026
00027     class QUENTIER_EXPORT ISkipRuleBuilder
00028     {
00029     public:
00030         virtual ~ISkipRuleBuilder();
00031
00032         virtual ISkipRuleBuilder & setTarget(ISkipRule::Target target) = 0;
00033         virtual ISkipRuleBuilder & setValue(QString value) = 0;
00034         virtual ISkipRuleBuilder & setMatchMode(MatchMode matchMode) = 0;
00035         virtual ISkipRuleBuilder & setIncludeContents(bool includeContents) = 0;
00036         virtual ISkipRuleBuilder & setCaseSensitivity(
00037             Qt::CaseSensitivity caseSensitivity) = 0;
00038
00039         [[nodiscard]] virtual ISkipRulePtr build() = 0;
00040     };
00041
00042 } // namespace quentier::enml::conversion_rules

```

6.3 MatchMode.h

```

00001 /*
00002  * Copyright 2023 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 class QDebug;
00024 class QTextStream;
00025
00026 namespace quentier::enml::conversion_rules {
00027
00031     enum class MatchMode
00032     {
00036         Equals,
00040         StartsWith,
00044         EndsWith,
00048         Contains
00049     };
00050

```

```

00051 QUENTIER_EXPORT QTextStream & operator«(
00052     QTextStream & strm, MatchMode matchMode);
00053
00054 QUENTIER_EXPORT QDebug & operator«(QDebug & dbg, MatchMode matchMode);
00055
00056 } // namespace quentier::enml::conversion_rules

```

6.4 HtmlUtils.h

```

00001 /*
00002  * Copyright 2023-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/ErrorMessage.h>
00022 #include <quentier/types/Result.h>
00023 #include <quentier/utility/Linkage.h>
00024
00025 #include <QFlags>
00026
00027 #include <memory>
00028
00029 namespace quentier::enml::utils {
00030
00031 [[nodiscard]] Result<QString, ErrorMessage> QUENTIER_EXPORT
00032     convertHtmlToXml(const QString & html);
00033
00034 [[nodiscard]] Result<QString, ErrorMessage> QUENTIER_EXPORT
00035     convertHtmlToXhtml(const QString & html);
00036
00037 [[nodiscard]] Result<QString, ErrorMessage> QUENTIER_EXPORT
00038     cleanupHtml(const QString & html);
00039
00040 enum class EscapeStringOption
00041 {
00042     Simplify = 1 « 0,
00043 };
00044
00045 Q_DECLARE_FLAGS(EscapeStringOptions, EscapeStringOption);
00046
00047 [[nodiscard]] QString QUENTIER_EXPORT htmlEscapeString(
00048     QString str, EscapeStringOptions options = EscapeStringOptions{});
00049
00050 } // namespace quentier::enml::utils

```

6.5 IConverter.h

```

00001 /*
00002  * Copyright 2023-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.

```

```

00017  */
00018
00019  #pragma once
00020
00021  #include <quentier/enml/Fwd.h>
00022  #include <quentier/enml/conversion_rules/Fwd.h>
00023  #include <quentier/types/ErrorMessage.h>
00024  #include <quentier/types/Result.h>
00025  #include <quentier/utility/Linkage.h>
00026
00027  #include <QList>
00028  #include <QStringList>
00029  #include <QTextDocument>
00030
00031  #include <qevercloud/types/Note.h>
00032
00033  namespace quentier::enml {
00034
00035  class QUENTIER_EXPORT IConverter
00036  {
00037  public:
00038      virtual ~IConverter();
00039
00040      [[nodiscard]] virtual Result<QString, ErrorMessage> convertHtmlToEnml(
00041          const QString & html, IDecryptedTextCache & decryptedTextCache,
00042          const QList<conversion_rules::ISkipRulePtr> & skipRules = {}) const = 0;
00043
00044      [[nodiscard]] virtual Result<void, ErrorMessage> convertHtmlToDoc(
00045          const QString & html, QTextDocument & doc,
00046          const QList<conversion_rules::ISkipRulePtr> & skipRules = {}) const = 0;
00047
00048      [[nodiscard]] virtual Result<QString, ErrorMessage> convertHtmlToXml(
00049          const QString & html) const = 0;
00050
00051      [[nodiscard]] virtual Result<QString, ErrorMessage> convertHtmlToXhtml(
00052          const QString & html) const = 0;
00053
00054      [[nodiscard]] virtual Result<IHtmlDataPtr, ErrorMessage> convertEnmlToHtml(
00055          const QString & enml,
00056          IDecryptedTextCache & decryptedTextCache) const = 0;
00057
00058      [[nodiscard]] virtual Result<QString, ErrorMessage> convertEnmlToPlainText(
00059          const QString & enml) const = 0;
00060
00061      [[nodiscard]] virtual Result<QStringList, ErrorMessage>
00062          convertEnmlToWordsList(const QString & enml) const = 0;
00063
00064      [[nodiscard]] virtual QStringList convertPlainTextToWordsList(
00065          const QString & plainText) const = 0;
00066
00067      [[nodiscard]] virtual Result<void, ErrorMessage> validateEnml(
00068          const QString & enml) const = 0;
00069
00070      [[nodiscard]] virtual Result<QString, ErrorMessage> validateAndFixupEnml(
00071          const QString & enml) const = 0;
00072
00073      enum class EnexExportTags
00074      {
00075          Yes = 0,
00076          No
00077      };
00078
00079      [[nodiscard]] virtual Result<QString, ErrorMessage> exportNotesToEnex(
00080          const QList<qevercloud::Note> & notes,
00081          const QHash<QString, QString> & tagNameByTagLocalIds,
00082          EnexExportTags exportTagsOption,
00083          const QString & version = {}) const = 0;
00084
00085      [[nodiscard]] virtual Result<QList<qevercloud::Note>, ErrorMessage>
00086          importEnex(const QString & enex) const = 0;
00087  };
00088
00089  } // namespace quentier::enml

```

6.6 IDecryptedTextCache.h

```

00001  /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by

```

```

00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/IEncryptor.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QString>
00025
00026 #include <optional>
00027 #include <utility>
00028
00029 class QDebug;
00030 class QTextStream;
00031
00032 namespace quentier::enml {
00033
00034 class QUENTIER_EXPORT IDecryptedTextCache
00035 {
00036 public:
00037     virtual ~IDecryptedTextCache();
00038
00039     enum class RememberForSession
00040     {
00041         Yes,
00042         No
00043     };
00044
00045     friend QUENTIER_EXPORT QDebug & operator«(
00046         QDebug & dbg, RememberForSession rememberForSession);
00047
00048     friend QUENTIER_EXPORT QTextStream & operator«(
00049         QTextStream & strm, RememberForSession rememberForSession);
00050
00051     virtual void addDecryptexTextInfo(
00052         const QString & encryptedText, const QString & decryptedText,
00053         const QString & passphrase, utility::IEncryptor::Cipher cipher,
00054         RememberForSession rememberForSession) = 0;
00055
00056     [[nodiscard]] virtual std::optional<std::pair<QString, RememberForSession>»
00057         findDecryptedTextInfo(const QString & encryptedText) const = 0;
00058
00059     [[nodiscard]] virtual std::optional<QString> updateDecryptedTextInfo(
00060         const QString & originalEncryptedText,
00061         const QString & newDecryptedText) = 0;
00062
00063     [[nodiscard]] virtual bool containsRememberedForSessionEntries() const = 0;
00064
00065     virtual void removeDecryptedTextInfo(const QString & encryptedText) = 0;
00066     virtual void clearNonRememberedForSessionEntries() = 0;
00067 };
00068
00069 } // namespace quentier::enml

```

6.7 IENMLTagsConverter.h

```

00001 /*
00002  * Copyright 2023-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018

```

```

00019 #pragma once
00020
00021 #include <quentier/types/ErrorMessage.h>
00022 #include <quentier/types/Result.h>
00023 #include <quentier/utility/IEncryptor.h>
00024 #include <quentier/utility/Linkage.h>
00025
00026 #include <qevercloud/types/Fwd.h>
00027
00028 #include <QString>
00029 #include <QtGlobal>
00030
00031 namespace quentier::enml {
00032
00033 class QUENTIER_EXPORT IENMLTagsConverter
00034 {
00035 public:
00036     virtual ~IENMLTagsConverter();
00037
00038     [[nodiscard]] virtual QString convertEnToDo(
00039         bool checked, quint32 index) const = 0;
00040
00041     [[nodiscard]] virtual QString convertEncryptedText(
00042         const QString & encryptedText, const QString & hint,
00043         utility::IEncryptor::Cipher cipher, quint32 index) const = 0;
00044
00045     [[nodiscard]] virtual QString convertDecryptedText(
00046         const QString & decryptedText, const QString & encryptedText,
00047         const QString & hint, utility::IEncryptor::Cipher cipher,
00048         quint32 index) const = 0;
00049
00050     [[nodiscard]] virtual Result<QString, ErrorMessage> convertResource(
00051         const qevercloud::Resource & resource) const = 0;
00052 };
00053
00054 } // namespace quentier::enml

```

6.8 IHtmlData.h

```

00001 /*
00002  * Copyright 2023-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022 #include <quentier/utility/Printable.h>
00023
00024 #include <QString>
00025 #include <QtGlobal>
00026
00027 namespace quentier::enml {
00028
00029 struct QUENTIER_EXPORT IHtmlData : public utility::Printable
00030 {
00031     [[nodiscard]] virtual QString html() const = 0;
00032
00033     [[nodiscard]] virtual quint32 numEnToDoNodes() const = 0;
00034
00035     [[nodiscard]] virtual quint32 numHyperlinkNodes() const = 0;
00036
00037     [[nodiscard]] virtual quint32 numEncryptNodes() const = 0;
00038
00039     [[nodiscard]] virtual quint32 numDecryptNodes() const = 0;
00040
00041 public: // utility::Printable
00042     QTextStream & print(QTextStream & strm) const override;
00043 };
00044
00045 } // namespace quentier::enml

```

6.9 InvalidArgument.h

```

00001 /*
00002  * Copyright 2021 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/exception/IQuentierException.h>
00022
00023 namespace quentier {
00024
00025 class QUENTIER_EXPORT InvalidArgument : public IQuentierException
00026 {
00027 public:
00028     explicit InvalidArgument(ErrorString message);
00029
00030     [[nodiscard]] InvalidArgument * clone() const override;
00031     void raise() const override;
00032
00033 protected:
00034     [[nodiscard]] QString exceptionDisplayName() const override;
00035 };
00036
00037 } // namespace quentier

```

6.10 IQuentierException.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/ErrorString.h>
00022 #include <quentier/utility/Printable.h>
00023
00024 #include <QException>
00025
00026 namespace quentier {
00027
00028 class QUENTIER_EXPORT IQuentierException :
00029     public utility::Printable,
00030     public QException
00031 {
00032 public:
00033     ~IQuentierException() noexcept override;
00034
00035     [[nodiscard]] ErrorString errorMessage() const;
00036     [[nodiscard]] QString localizedErrorMessage() const;
00037     [[nodiscard]] QString nonLocalizedErrorMessage() const;
00038
00039     // std::exception
00040     [[nodiscard]] const char * what() const noexcept override;

```

```

00049
00050     // utility::Printable
00051     QTextStream & print(QTextStream & strm) const override;
00052
00053 protected:
00054     explicit IQuentierException(ErrorString message);
00055     IQuentierException(const IQuentierException & other);
00056     IQuentierException & operator=(const IQuentierException & other);
00057
00058     [[nodiscard]] virtual QString exceptionDisplayName() const = 0;
00059
00060 private:
00061     ErrorString m_message;
00062     char * m_whatMessage = nullptr;
00063 };
00064
00065 } // namespace quentier

```

6.11 OperationCanceled.h

```

00001 /*
00002  * Copyright 2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/exception/IQuentierException.h>
00022
00023 namespace quentier {
00024
00025     class QUENTIER_EXPORT OperationCanceled : public IQuentierException
00026     {
00027     public:
00028         explicit OperationCanceled();
00029
00030         [[nodiscard]] OperationCanceled * clone() const override;
00031         void raise() const override;
00032
00033     protected:
00034         [[nodiscard]] QString exceptionDisplayName() const override;
00035     };
00036
00037 } // namespace quentier

```

6.12 RuntimeError.h

```

00001 /*
00002  * Copyright 2021 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018

```



```

00019 #pragma once
00020
00021 #include <quentier/exception/IQuentierException.h>
00022
00023 namespace quentier {
00024
00025 class QUENTIER_EXPORT RuntimeError : public IQuentierException
00026 {
00027 public:
00028     explicit RuntimeError(ErrorString message);
00029     ~RuntimeError() noexcept override;
00030
00031     [[nodiscard]] RuntimeError * clone() const override;
00032     void raise() const override;
00033
00034 protected:
00035     [[nodiscard]] QString exceptionDisplayName() const override;
00036 };
00037
00038 } // namespace quentier

```

6.13 enml/conversion_rules/Factory.h

```

00001 /*
00002  * Copyright 2023 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/enml/conversion_rules/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 namespace quentier::enml::conversion_rules {
00025
00026 [[nodiscard]] QUENTIER_EXPORT ISkipRuleBuilderPtr createSkipRuleBuilder();
00027
00028 } // namespace quentier::enml::conversion_rules

```

6.14 enml/Factory.h

```

00001 /*
00002  * Copyright 2023-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/enml/Fwd.h>
00022 #include <quentier/utility/Fwd.h>
00023 #include <quentier/utility/Linkage.h>
00024

```

```

00025 namespace quantier::enml {
00026
00033 [[nodiscard]] QUENTIER_EXPORT IDecryptedTextCachePtr
00034     createDecryptedTextCache(utility::IEncryptorPtr encryptor = nullptr);
00035
00039 [[nodiscard]] QUENTIER_EXPORT IENMLTagsConverterPtr createEnmlTagsConverter();
00040
00048 [[nodiscard]] QUENTIER_EXPORT IConverterPtr
00049     createConverter(IENMLTagsConverterPtr enmlTagsConverter = nullptr);
00050
00051 } // namespace quantier::enml

```

6.15 local_storage/Factory.h

```

00001 /*
00002  * Copyright 2023-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/local_storage/Fwd.h>
00022 #include <quantier/threading/Fwd.h>
00023 #include <quantier/types/Fwd.h>
00024 #include <quantier/utility/Linkage.h>
00025
00026 #include <QtGlobal>
00027
00028 class QDir;
00029
00030 namespace quantier::local_storage {
00031
00032 [[nodiscard]] QUENTIER_EXPORT ILocalStoragePtr createSqliteLocalStorage(
00033     const Account & account, const QDir & localStorageDir,
00034     threading::QThreadPtr thread = {});
00035
00036 } // namespace quantier::local_storage

```

6.16 synchronization/Factory.h

```

00001 /*
00002  * Copyright 2023-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/local_storage/Fwd.h>
00022 #include <quantier/synchronization/Fwd.h>
00023 #include <quantier/threading/Fwd.h>
00024 #include <quantier/utility/Fwd.h>
00025 #include <quantier/utility/Linkage.h>

```

```

00026
00027 #include <qevercloud/Fwd.h>
00028
00029 #include <QString>
00030 #include <QUrl>
00031 #include <QtGlobal>
00032
00033 class QWidget;
00034
00035 namespace quantier::synchronization {
00036
00037 [[nodiscard]] QUENTIER_EXPORT IAuthenticatorPtr createQEverCloudAuthenticator(
00038     QString consumerKey, QString consumerSecret, QUrl serverUrl,
00039     threading::QThreadPtr uiThread, QWidget * parentWidget = nullptr);
00040
00041 [[nodiscard]] QUENTIER_EXPORT ISynchronizerPtr createSynchronizer(
00042     const QUrl & userStoreUrl, IAuthenticatorPtr authenticator,
00043     ISyncStateStoragePtr syncStateStorage = nullptr,
00044     utility::IKeychainServicePtr keychainService = nullptr,
00045     INoteStoreFactoryPtr noteStoreFactory = nullptr,
00046     IUserStoreFactoryPtr userStoreFactory = nullptr,
00047     qevercloud::IRequestContextPtr ctx = nullptr,
00048     qevercloud::IRetryPolicyPtr retryPolicy = nullptr);
00049
00050 [[nodiscard]] QUENTIER_EXPORT ISyncConflictResolverPtr
00051     createSimpleSyncConflictResolver(
00052         local_storage::ILocalStoragePtr localStorage);
00053
00054 [[nodiscard]] QUENTIER_EXPORT ISyncStateStoragePtr
00055     createSyncStateStorage(QObject * parent = nullptr);
00056
00057 } // namespace quantier::synchronization

```

6.17 threading/Factory.h

```

00001 /*
00002  * Copyright 2023 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/threading/Fwd.h>
00022 #include <quantier/utility/Linkage.h>
00023
00024 namespace quantier::threading {
00025
00029 [[nodiscard]] QUENTIER_EXPORT QThreadPoolPtr globalThreadPool();
00030
00031 } // namespace quantier::threading

```

6.18 utility/Factory.h

```

00001 /*
00002  * Copyright 2024-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

```

```

00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QString>
00025
00026 namespace quentier::utility {
00027
00032 [[nodiscard]] QUENTIER_EXPORT IEcryptorPtr createOpenSslEncryptor();
00033
00037 [[nodiscard]] QUENTIER_EXPORT IKeychainServicePtr newQtKeychainService();
00038
00043 [[nodiscard]] QUENTIER_EXPORT IKeychainServicePtr
00044     newObfuscatingKeychainService();
00045
00055 [[nodiscard]] QUENTIER_EXPORT IKeychainServicePtr newCompositeKeychainService(
00056     QString name, IKeychainServicePtr primaryKeychain,
00057     IKeychainServicePtr secondaryKeychain);
00058
00069 [[nodiscard]] QUENTIER_EXPORT IKeychainServicePtr newMigratingKeychainService(
00070     IKeychainServicePtr sourceKeychain, IKeychainServicePtr sinkKeychain);
00071
00072 } // namespace quentier::utility

```

6.19 ILocalStorage.h

```

00001 /*
00002  * Copyright 2020-2023 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/local_storage/Fwd.h>
00022 #include <quentier/local_storage/NoteSearchQuery.h>
00023 #include <quentier/types/Fwd.h>
00024 #include <quentier/utility/Linkage.h>
00025
00026 #include <qevercloud/types/LinkedNotebook.h>
00027 #include <qevercloud/types/Note.h>
00028 #include <qevercloud/types/Notebook.h>
00029 #include <qevercloud/types/Resource.h>
00030 #include <qevercloud/types/SavedSearch.h>
00031 #include <qevercloud/types/SharedNotebook.h>
00032 #include <qevercloud/types/Tag.h>
00033 #include <qevercloud/types/User.h>
00034
00035 #include <QFlags>
00036 #include <QFuture>
00037 #include <QHash>
00038 #include <QList>
00039 #include <QStringList>
00040 #include <QThreadPool>
00041
00042 #include <optional>
00043 #include <utility>
00044
00045 class QDebug;
00046 class QTextStream;
00047 class QThreadPool;
00048
00049 namespace quentier::local_storage {
00050

```

```

00051 class QUINTIER_EXPORT ILocalStorage
00052 {
00053 public:
00054     virtual ~ILocalStorage() = default;
00055
00056 public:
00057     enum class StartupOption
00058     {
00059         ClearDatabase = 1 « 1,
00060         OverrideLock = 1 « 2
00061     };
00062     Q_DECLARE_FLAGS(StartupOptions, StartupOption);
00063
00064     friend QUINTIER_EXPORT QTextStream & operator«(
00065         QTextStream & strm, StartupOption option);
00066
00067     friend QUINTIER_EXPORT QDebug & operator«(
00068         QDebug & dbg, StartupOption option);
00069
00070     friend QUINTIER_EXPORT QTextStream & operator«(
00071         QTextStream & strm, StartupOptions options);
00072
00073     friend QUINTIER_EXPORT QDebug & operator«(
00074         QDebug & dbg, StartupOptions options);
00075
00076
00077     enum class ListObjectsFilter
00078     {
00079         Include,
00080         Exclude
00081     };
00082
00083     friend QUINTIER_EXPORT QTextStream & operator«(
00084         QTextStream & strm, ListObjectsFilter filter);
00085
00086     friend QUINTIER_EXPORT QDebug & operator«(
00087         QDebug & dbg, ListObjectsFilter filter);
00088
00089
00090     struct QUINTIER_EXPORT ListObjectsFilters
00091     {
00092         std::optional<ListObjectsFilter> m_locallyModifiedFilter;
00093         std::optional<ListObjectsFilter> m_withGuidFilter;
00094         std::optional<ListObjectsFilter> m_localOnlyFilter;
00095         std::optional<ListObjectsFilter> m_locallyFavoritedFilter;
00096     };
00097
00098     friend QUINTIER_EXPORT QTextStream & operator«(
00099         QTextStream & strm, const ListObjectsFilters & filters);
00100
00101     friend QUINTIER_EXPORT QDebug & operator«(
00102         QDebug & dbg, const ListObjectsFilters & filters);
00103
00104
00105     struct QUINTIER_EXPORT ListGuidsFilters
00106     {
00107         std::optional<ListObjectsFilter> m_locallyModifiedFilter;
00108         std::optional<ListObjectsFilter> m_locallyFavoritedFilter;
00109     };
00110
00111     friend QUINTIER_EXPORT QTextStream & operator«(
00112         QTextStream & strm, const ListGuidsFilters & filters);
00113
00114     friend QUINTIER_EXPORT QDebug & operator«(
00115         QDebug & dbg, const ListGuidsFilters & filters);
00116
00117
00118     enum class OrderDirection
00119     {
00120         Ascending,
00121         Descending
00122     };
00123
00124     friend QUINTIER_EXPORT QTextStream & operator«(
00125         QTextStream & strm, OrderDirection orderDirection);
00126
00127     friend QUINTIER_EXPORT QDebug & operator«(
00128         QDebug & dbg, OrderDirection orderDirection);
00129
00130
00131     enum class ListNotebooksOrder
00132     {
00133         NoOrder,
00134         ByUpdateSequenceNumber,
00135         ByNotebookName,
00136         ByCreationTimestamp,
00137         ByModificationTimestamp
00138     };

```

```

00143     };
00144
00145     friend QUINTIER_EXPORT QTextStream & operator<<
00146         (QTextStream & strm, ListNotebooksOrder order);
00147
00148     friend QUINTIER_EXPORT QDebug & operator<<
00149         (QDebug & dbg, ListNotebooksOrder order);
00150
00152
00153     enum class ListLinkedNotebooksOrder
00154     {
00155         NoOrder,
00156         ByUpdateSequenceNumber,
00157         ByShareName,
00158         ByUsername,
00159     };
00160
00161     friend QUINTIER_EXPORT QTextStream & operator<<
00162         (QTextStream & strm, ListLinkedNotebooksOrder order);
00163
00164     friend QUINTIER_EXPORT QDebug & operator<<
00165         (QDebug & dbg, ListLinkedNotebooksOrder order);
00166
00168
00169     enum class ListTagsOrder
00170     {
00171         NoOrder,
00172         ByUpdateSequenceNumber,
00173         ByName
00174     };
00175
00176     friend QUINTIER_EXPORT QTextStream & operator<<
00177         (QTextStream & strm, ListTagsOrder order);
00178
00179     friend QUINTIER_EXPORT QDebug & operator<<
00180         (QDebug & dbg, ListTagsOrder order);
00181
00183
00184     enum class ListNotesOrder
00185     {
00186         NoOrder,
00187         ByUpdateSequenceNumber,
00188         ByTitle,
00189         ByCreationTimestamp,
00190         ByModificationTimestamp,
00191         ByDeletionTimestamp,
00192         ByAuthor,
00193         BySource,
00194         BySourceApplication,
00195         ByReminderTime,
00196         ByPlaceName
00197     };
00198
00199     friend QUINTIER_EXPORT QTextStream & operator<<
00200         (QTextStream & strm, ListNotesOrder order);
00201
00202     friend QUINTIER_EXPORT QDebug & operator<<
00203         (QDebug & dbg, ListNotesOrder order);
00204
00206
00207     enum class ListSavedSearchesOrder
00208     {
00209         NoOrder,
00210         ByUpdateSequenceNumber,
00211         ByName,
00212         ByFormat
00213     };
00214
00215     friend QUINTIER_EXPORT QTextStream & operator<<
00216         (QTextStream & strm, ListSavedSearchesOrder order);
00217
00218     friend QUINTIER_EXPORT QDebug & operator<<
00219         (QDebug & dbg, ListSavedSearchesOrder order);
00220
00222
00225     enum class Affiliation
00226     {
00227         Any,
00228         User,
00229         AnyLinkedNotebook,
00230         ParticularLinkedNotebooks
00231     };
00232
00233     friend QUINTIER_EXPORT QTextStream & operator<<
00234         (QTextStream & strm, Affiliation affiliation);
00235
00236     friend QUINTIER_EXPORT QDebug & operator<<

```

```

00237         QDebug & dbg, Affiliation affiliation);
00238
00240
00241     struct QUINTIER_EXPORT ListOptionsBase
00242     {
00243         ListOptionsBase() noexcept {}; // NOLINT
00244
00245         ListObjectsFilters m_filters = {};
00246         quint64 m_limit = 0UL;
00247         quint64 m_offset = 0UL;
00248         OrderDirection m_direction = OrderDirection::Ascending;
00249     };
00250
00251     struct QUINTIER_EXPORT ListNotebooksOptions : public ListOptionsBase
00252     {
00253         ListNotebooksOptions() noexcept {}; // NOLINT
00254
00255         ListNotebooksOrder m_order = ListNotebooksOrder::NoOrder;
00256         Affiliation m_affiliation = Affiliation::Any;
00257         QList<qevercloud::Guid> m_linkedNotebookGuids;
00258     };
00259
00260     friend QUINTIER_EXPORT QTextStream & operator<<(
00261         QTextStream & strm, const ListNotebooksOptions & options);
00262
00263     friend QUINTIER_EXPORT QDebug & operator<<(
00264         QDebug & dbg, const ListNotebooksOptions & options);
00265
00266     struct QUINTIER_EXPORT ListLinkedNotebooksOptions : public ListOptionsBase
00267     {
00268         ListLinkedNotebooksOptions() noexcept {}; // NOLINT
00269
00270         ListLinkedNotebooksOrder m_order = ListLinkedNotebooksOrder::NoOrder;
00271     };
00272
00273     friend QUINTIER_EXPORT QTextStream & operator<<(
00274         QTextStream & strm, const ListLinkedNotebooksOptions & options);
00275
00276     friend QUINTIER_EXPORT QDebug & operator<<(
00277         QDebug & dbg, const ListLinkedNotebooksOptions & options);
00278
00279     struct QUINTIER_EXPORT ListSavedSearchesOptions : public ListOptionsBase
00280     {
00281         ListSavedSearchesOptions() noexcept {}; // NOLINT
00282
00283         ListSavedSearchesOrder m_order = ListSavedSearchesOrder::NoOrder;
00284     };
00285
00286     friend QUINTIER_EXPORT QTextStream & operator<<(
00287         QTextStream & strm, const ListSavedSearchesOptions & options);
00288
00289     friend QUINTIER_EXPORT QDebug & operator<<(
00290         QDebug & dbg, const ListSavedSearchesOptions & options);
00291
00292     struct QUINTIER_EXPORT ListNotesOptions : public ListOptionsBase
00293     {
00294         ListNotesOptions() noexcept {}; // NOLINT
00295
00296         ListNotesOrder m_order = ListNotesOrder::NoOrder;
00297     };
00298
00299     friend QUINTIER_EXPORT QTextStream & operator<<(
00300         QTextStream & strm, const ListNotesOptions & options);
00301
00302     friend QUINTIER_EXPORT QDebug & operator<<(
00303         QDebug & dbg, const ListNotesOptions & options);
00304
00307     enum class TagNotesRelation
00308     {
00310         Any,
00312         WithNotes,
00314         WithoutNotes
00315     };
00316
00317     struct QUINTIER_EXPORT ListTagsOptions : public ListOptionsBase
00318     {
00319         ListTagsOptions() noexcept {}; // NOLINT
00320
00321         ListTagsOrder m_order = ListTagsOrder::NoOrder;
00322         Affiliation m_affiliation = Affiliation::Any;
00323         QList<qevercloud::Guid> m_linkedNotebookGuids;
00324         TagNotesRelation m_tagNotesRelation = TagNotesRelation::Any;
00325     };
00326
00327     friend QUINTIER_EXPORT QTextStream & operator<<(
00328         QTextStream & strm, const ListTagsOptions & options);
00329

```

```

00330     friend QUINTIER_EXPORT QDebug & operator«(
00331         QDebug & dbg, const ListTagsOptions & options);
00332
00333     enum class NoteCountOption
00334     {
00335         IncludeNonDeletedNotes = 1 « 1,
00336         IncludeDeletedNotes = 1 « 2
00337     };
00338     Q_DECLARE_FLAGS(NoteCountOptions, NoteCountOption)
00339
00340     friend QUINTIER_EXPORT QTextStream & operator«(
00341         QTextStream & strm, NoteCountOption option);
00342
00343     friend QUINTIER_EXPORT QDebug & operator«(
00344         QDebug & dbg, NoteCountOption option);
00345
00346     friend QUINTIER_EXPORT QTextStream & operator«(
00347         QTextStream & strm, NoteCountOptions options);
00348
00349     friend QUINTIER_EXPORT QDebug & operator«(
00350         QDebug & dbg, NoteCountOptions options);
00351
00352     enum class UpdateNoteOption
00353     {
00354         UpdateResourceMetadata = 1 « 1,
00355         UpdateResourceBinaryData = 1 « 2,
00356         UpdateTags = 1 « 3
00357     };
00358     Q_DECLARE_FLAGS(UpdateNoteOptions, UpdateNoteOption)
00359
00360     friend QUINTIER_EXPORT QTextStream & operator«(
00361         QTextStream & strm, UpdateNoteOption option);
00362
00363     friend QUINTIER_EXPORT QDebug & operator«(
00364         QDebug & dbg, UpdateNoteOption option);
00365
00366     friend QUINTIER_EXPORT QTextStream & operator«(
00367         QTextStream & strm, UpdateNoteOptions options);
00368
00369     friend QUINTIER_EXPORT QDebug & operator«(
00370         QDebug & dbg, UpdateNoteOptions options);
00371
00372     friend QUINTIER_EXPORT QTextStream & operator«(
00373         QTextStream & strm, UpdateNoteOptions options);
00374
00375     friend QUINTIER_EXPORT QDebug & operator«(
00376         QDebug & dbg, UpdateNoteOptions options);
00377
00378     enum class FetchNoteOption
00379     {
00380         WithResourceMetadata = 1 « 1,
00381         WithResourceBinaryData = 1 « 2
00382     };
00383     Q_DECLARE_FLAGS(FetchNoteOptions, FetchNoteOption)
00384
00385     friend QUINTIER_EXPORT QTextStream & operator«(
00386         QTextStream & strm, FetchNoteOption option);
00387
00388     friend QUINTIER_EXPORT QDebug & operator«(
00389         QDebug & dbg, FetchNoteOption option);
00390
00391     friend QUINTIER_EXPORT QTextStream & operator«(
00392         QTextStream & strm, FetchNoteOptions options);
00393
00394     friend QUINTIER_EXPORT QDebug & operator«(
00395         QDebug & dbg, FetchNoteOptions options);
00396
00397     enum class FetchResourceOption
00398     {
00399         WithBinaryData = 1 « 1
00400     };
00401     Q_DECLARE_FLAGS(FetchResourceOptions, FetchResourceOption)
00402
00403     friend QUINTIER_EXPORT QTextStream & operator«(
00404         QTextStream & strm, FetchResourceOption option);
00405
00406     friend QUINTIER_EXPORT QDebug & operator«(
00407         QDebug & dbg, FetchResourceOption option);
00408
00409     friend QUINTIER_EXPORT QTextStream & operator«(
00410         QTextStream & strm, FetchResourceOptions options);
00411
00412     friend QUINTIER_EXPORT QDebug & operator«(
00413         QDebug & dbg, FetchResourceOptions options);
00414
00415     enum class HighestUsnOption
00416     {
00417         WithinUserOwnContent,

```



```

00422         WithinUserOwnContentAndLinkedNotebooks
00423     };
00424
00425     friend QUINTIER_EXPORT QTextStream & operator«(
00426         QTextStream & strm, HighestUsnOption option);
00427
00428     friend QUINTIER_EXPORT QDebug & operator«(
00429         QDebug & dbg, HighestUsnOption option);
00430
00431 public:
00432     // Versions/upgrade API
00433     [[nodiscard]] virtual QFuture<bool> isVersionTooHigh() const = 0;
00434     [[nodiscard]] virtual QFuture<bool> requiresUpgrade() const = 0;
00435     [[nodiscard]] virtual QFuture<QList<IPatchPtr> > requiredPatches() const = 0;
00436     [[nodiscard]] virtual QFuture<qint32> version() const = 0;
00437     [[nodiscard]] virtual QFuture<qint32> highestSupportedVersion() const = 0;
00438
00439     // Users API
00440     [[nodiscard]] virtual QFuture<quint32> userCount() const = 0;
00441     [[nodiscard]] virtual QFuture<void> putUser(qevercloud::User user) = 0;
00442
00443     [[nodiscard]] virtual QFuture<std::optional<qevercloud::User> > findUserById(
00444         qevercloud::UserID userId) const = 0;
00445
00446     [[nodiscard]] virtual QFuture<void> expungeUserById(
00447         qevercloud::UserID userId) = 0;
00448
00449     // Notebooks API
00450     [[nodiscard]] virtual QFuture<quint32> notebookCount() const = 0;
00451
00452     [[nodiscard]] virtual QFuture<void> putNotebook(
00453         qevercloud::Notebook notebook) = 0;
00454
00455     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Notebook> >
00456         findNotebookByLocalId(QString notebookLocalId) const = 0;
00457
00458     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Notebook> >
00459         findNotebookByGuid(qevercloud::Guid guid) const = 0;
00460
00461     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Notebook> >
00462         findNotebookByName(
00463             QString notebookName,
00464             std::optional<qevercloud::Guid> linkedNotebookGuid =
00465                 std::nullopt) const = 0;
00466
00467     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Notebook> >
00468         findDefaultNotebook() const = 0;
00469
00470     [[nodiscard]] virtual QFuture<void> expungeNotebookByLocalId(
00471         QString notebookLocalId) = 0;
00472
00473     [[nodiscard]] virtual QFuture<void> expungeNotebookByGuid(
00474         qevercloud::Guid notebookGuid) = 0;
00475
00476     [[nodiscard]] virtual QFuture<void> expungeNotebookByName(
00477         QString name,
00478         std::optional<qevercloud::Guid> linkedNotebookGuid = std::nullopt) = 0;
00479
00480     [[nodiscard]] virtual QFuture<QList<qevercloud::Notebook> > listNotebooks(
00481         ListNotebooksOptions options = {}) const = 0;
00482
00483     [[nodiscard]] virtual QFuture<QList<qevercloud::SharedNotebook> >
00484         listSharedNotebooks(qevercloud::Guid notebookGuid = {}) const = 0;
00485
00486     [[nodiscard]] virtual QFuture<QSet<qevercloud::Guid> > listNotebookGuids(
00487         ListGuidsFilters filters,
00488         std::optional<qevercloud::Guid> linkedNotebookGuid = {}) const = 0;
00489
00490     // Linked notebooks API
00491     [[nodiscard]] virtual QFuture<quint32> linkedNotebookCount() const = 0;
00492
00493     [[nodiscard]] virtual QFuture<void> putLinkedNotebook(
00494         qevercloud::LinkedNotebook linkedNotebook) = 0;
00495
00496     [[nodiscard]] virtual QFuture<std::optional<qevercloud::LinkedNotebook> >
00497         findLinkedNotebookByGuid(qevercloud::Guid guid) const = 0;
00498
00499     [[nodiscard]] virtual QFuture<void> expungeLinkedNotebookByGuid(
00500         qevercloud::Guid guid) = 0;
00501
00502     [[nodiscard]] virtual QFuture<QList<qevercloud::LinkedNotebook> >
00503         listLinkedNotebooks(ListLinkedNotebooksOptions options = {}) const = 0;
00504
00505     // Notes API
00506     [[nodiscard]] virtual QFuture<quint32> noteCount(
00507         NoteCountOptions options = NoteCountOptions(
00508             NoteCountOption::IncludeNonDeletedNotes)) const = 0;

```

```

00509
00510 [[nodiscard]] virtual QFuture<quint32> noteCountPerNotebookLocalId(
00511     QString notebookLocalId,
00512     NoteCountOptions options = NoteCountOptions(
00513         NoteCountOption::IncludeNonDeletedNotes)) const = 0;
00514
00515 [[nodiscard]] virtual QFuture<quint32> noteCountPerTagLocalId(
00516     QString tagLocalId,
00517     NoteCountOptions options = NoteCountOptions(
00518         NoteCountOption::IncludeNonDeletedNotes)) const = 0;
00519
00520 [[nodiscard]] virtual QFuture<QHash<QString, quint32>> noteCountsPerTags(
00521     ListTagsOptions listTagsOptions = {},
00522     NoteCountOptions options = NoteCountOptions(
00523         NoteCountOption::IncludeNonDeletedNotes)) const = 0;
00524
00525 [[nodiscard]] virtual QFuture<quint32> noteCountPerNotebookAndTagLocalIds(
00526     QStringList notebookLocalIds, QStringList tagLocalIds,
00527     NoteCountOptions options = NoteCountOptions(
00528         NoteCountOption::IncludeNonDeletedNotes)) const = 0;
00529
00530 [[nodiscard]] virtual QFuture<void> putNote(qevercloud::Note note) = 0;
00531
00532 [[nodiscard]] virtual QFuture<void> updateNote(
00533     qevercloud::Note note, UpdateNoteOptions options) = 0;
00534
00535 [[nodiscard]] virtual QFuture<std::optional<qevercloud::Note>>
00536     findNoteByLocalId(
00537     QString noteLocalId, FetchNoteOptions options) const = 0;
00538
00539 [[nodiscard]] virtual QFuture<std::optional<qevercloud::Note>>
00540     findNoteByGuid(
00541     qevercloud::Guid noteGuid, FetchNoteOptions options) const = 0;
00542
00543 [[nodiscard]] virtual QFuture<QList<qevercloud::Note>> listNotes(
00544     FetchNoteOptions fetchOptions,
00545     ListNotesOptions listOptions = {}) const = 0;
00546
00547 [[nodiscard]] virtual QFuture<QList<qevercloud::Note>>
00548     listNotesPerNotebookLocalId(
00549     QString notebookLocalId, FetchNoteOptions fetchOptions,
00550     ListNotesOptions listOptions = {}) const = 0;
00551
00552 [[nodiscard]] virtual QFuture<QList<qevercloud::Note>>
00553     listNotesPerTagLocalId(
00554     QString tagLocalId, FetchNoteOptions fetchOptions,
00555     ListNotesOptions listOptions = {}) const = 0;
00556
00557 [[nodiscard]] virtual QFuture<QList<qevercloud::Note>>
00558     listNotesPerNotebookAndTagLocalIds(
00559     QStringList notebookLocalIds, QStringList tagLocalIds,
00560     FetchNoteOptions fetchOptions,
00561     ListNotesOptions listOptions = {}) const = 0;
00562
00563 [[nodiscard]] virtual QFuture<QList<qevercloud::Note>> listNotesByLocalIds(
00564     QStringList noteLocalIds, FetchNoteOptions fetchOptions,
00565     ListNotesOptions listOptions = {}) const = 0;
00566
00567 [[nodiscard]] virtual QFuture<QSet<qevercloud::Guid>> listNoteGuids(
00568     ListGuidsFilters filters,
00569     std::optional<qevercloud::Guid> linkedNotebookGuid = {}) const = 0;
00570
00571 [[nodiscard]] virtual QFuture<QList<qevercloud::Note>> queryNotes(
00572     NoteSearchQuery query, FetchNoteOptions fetchOptions) const = 0;
00573
00574 [[nodiscard]] virtual QFuture<QStringList> queryNoteLocalIds(
00575     NoteSearchQuery query) const = 0;
00576
00577 [[nodiscard]] virtual QFuture<void> expungeNoteByLocalId(
00578     QString noteLocalId) = 0;
00579
00580 [[nodiscard]] virtual QFuture<void> expungeNoteByGuid(
00581     qevercloud::Guid noteGuid) = 0;
00582
00583 // Tags API
00584 [[nodiscard]] virtual QFuture<quint32> tagCount() const = 0;
00585 [[nodiscard]] virtual QFuture<void> putTag(qevercloud::Tag tag) = 0;
00586
00587 [[nodiscard]] virtual QFuture<std::optional<qevercloud::Tag>>
00588     findTagByLocalId(QString tagLocalId) const = 0;
00589
00590 [[nodiscard]] virtual QFuture<std::optional<qevercloud::Tag>> findTagByGuid(
00591     qevercloud::Guid tagGuid) const = 0;
00592
00593 [[nodiscard]] virtual QFuture<std::optional<qevercloud::Tag>> findTagByName(
00594     QString tagName,
00595     std::optional<qevercloud::Guid> linkedNotebookGuid =

```

```

00596         std::nullopt) const = 0;
00597
00598     [[nodiscard]] virtual QFuture<QList<qevercloud::Tag> listTags(
00599         ListTagsOptions options = {}) const = 0;
00600
00601     [[nodiscard]] virtual QFuture<QList<qevercloud::Tag>
00602         listTagsPerNoteLocalId(
00603         QString noteLocalId, ListTagsOptions options = {}) const = 0;
00604
00605     [[nodiscard]] virtual QFuture<QSet<qevercloud::Guid> listTagGuids(
00606         ListGuidsFilters filters,
00607         std::optional<qevercloud::Guid> linkedNotebookGuid = {}) const = 0;
00608
00609     [[nodiscard]] virtual QFuture<void> expungeTagByLocalId(
00610         QString tagLocalId) = 0;
00611
00612     [[nodiscard]] virtual QFuture<void> expungeTagByGuid(
00613         qevercloud::Guid tagGuid) = 0;
00614
00615     [[nodiscard]] virtual QFuture<void> expungeTagByName(
00616         QString name,
00617         std::optional<qevercloud::Guid> linkedNotebookGuid = std::nullopt) = 0;
00618
00619     // Resources API
00620     [[nodiscard]] virtual QFuture<quint32> resourceCount(
00621         NoteCountOptions options = NoteCountOptions(
00622             NoteCountOption::IncludeNonDeletedNotes)) const = 0;
00623
00624     [[nodiscard]] virtual QFuture<quint32> resourceCountPerNoteLocalId(
00625         QString noteLocalId) const = 0;
00626
00627     [[nodiscard]] virtual QFuture<void> putResource(
00628         qevercloud::Resource resource) = 0;
00629
00630     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Resource>
00631         findResourceByLocalId(
00632         QString resourceLocalId,
00633         FetchResourceOptions options = {}) const = 0;
00634
00635     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Resource>
00636         findResourceByGuid(
00637         qevercloud::Guid resourceGuid,
00638         FetchResourceOptions options = {}) const = 0;
00639
00640     [[nodiscard]] virtual QFuture<void> expungeResourceByLocalId(
00641         QString resourceLocalId) = 0;
00642
00643     [[nodiscard]] virtual QFuture<void> expungeResourceByGuid(
00644         qevercloud::Guid resourceGuid) = 0;
00645
00646     // Saved searches API
00647     [[nodiscard]] virtual QFuture<quint32> savedSearchCount() const = 0;
00648
00649     [[nodiscard]] virtual QFuture<void> putSavedSearch(
00650         qevercloud::SavedSearch search) = 0;
00651
00652     [[nodiscard]] virtual QFuture<std::optional<qevercloud::SavedSearch>
00653         findSavedSearchByLocalId(QString savedSearchLocalId) const = 0;
00654
00655     [[nodiscard]] virtual QFuture<std::optional<qevercloud::SavedSearch>
00656         findSavedSearchByGuid(qevercloud::Guid guid) const = 0;
00657
00658     [[nodiscard]] virtual QFuture<std::optional<qevercloud::SavedSearch>
00659         findSavedSearchByName(QString name) const = 0;
00660
00661     [[nodiscard]] virtual QFuture<QList<qevercloud::SavedSearch>
00662         listSavedSearches(ListSavedSearchesOptions options = {}) const = 0;
00663
00664     [[nodiscard]] virtual QFuture<QSet<qevercloud::Guid> listSavedSearchGuids(
00665         ListGuidsFilters filters) const = 0;
00666
00667     [[nodiscard]] virtual QFuture<void> expungeSavedSearchByLocalId(
00668         QString savedSearchLocalId) = 0;
00669
00670     [[nodiscard]] virtual QFuture<void> expungeSavedSearchByGuid(
00671         qevercloud::Guid guid) = 0;
00672
00673     // Synchronization API
00674     [[nodiscard]] virtual QFuture<qint32> highestUpdateSequenceNumber(
00675         HighestUsnOption option) const = 0;
00676
00677     [[nodiscard]] virtual QFuture<qint32> highestUpdateSequenceNumber(
00678         qevercloud::Guid linkedNotebookGuid) const = 0;
00679
00680     [[nodiscard]] virtual ILocalStorageNotifier * notifier() const = 0;
00681 };
00682
00688

```

```

00689 [[nodiscard]] QUENTIER_EXPORT bool operator==(
00690     const ILocalStorage::ListObjectsFilters & lhs,
00691     const ILocalStorage::ListObjectsFilters & rhs) noexcept;
00692
00693 [[nodiscard]] QUENTIER_EXPORT bool operator==(
00694     const ILocalStorage::ListOptionsBase & lhs,
00695     const ILocalStorage::ListOptionsBase & rhs) noexcept;
00696
00697 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
00698     const ILocalStorage::ListOptionsBase & lhs,
00699     const ILocalStorage::ListOptionsBase & rhs) noexcept;
00700
00701 [[nodiscard]] QUENTIER_EXPORT bool operator==(
00702     const ILocalStorage::ListNotebooksOptions & lhs,
00703     const ILocalStorage::ListNotebooksOptions & rhs) noexcept;
00704
00705 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
00706     const ILocalStorage::ListNotebooksOptions & lhs,
00707     const ILocalStorage::ListNotebooksOptions & rhs) noexcept;
00708
00709 [[nodiscard]] QUENTIER_EXPORT bool operator==(
00710     const ILocalStorage::ListLinkedNotebooksOptions & lhs,
00711     const ILocalStorage::ListLinkedNotebooksOptions & rhs) noexcept;
00712
00713 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
00714     const ILocalStorage::ListLinkedNotebooksOptions & lhs,
00715     const ILocalStorage::ListLinkedNotebooksOptions & rhs) noexcept;
00716
00717 [[nodiscard]] QUENTIER_EXPORT bool operator==(
00718     const ILocalStorage::ListSavedSearchesOptions & lhs,
00719     const ILocalStorage::ListSavedSearchesOptions & rhs) noexcept;
00720
00721 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
00722     const ILocalStorage::ListSavedSearchesOptions & lhs,
00723     const ILocalStorage::ListSavedSearchesOptions & rhs) noexcept;
00724
00725 [[nodiscard]] QUENTIER_EXPORT bool operator==(
00726     const ILocalStorage::ListNotesOptions & lhs,
00727     const ILocalStorage::ListNotesOptions & rhs) noexcept;
00728
00729 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
00730     const ILocalStorage::ListNotesOptions & lhs,
00731     const ILocalStorage::ListNotesOptions & rhs) noexcept;
00732
00733 [[nodiscard]] QUENTIER_EXPORT bool operator==(
00734     const ILocalStorage::ListTagsOptions & lhs,
00735     const ILocalStorage::ListTagsOptions & rhs) noexcept;
00736
00737 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
00738     const ILocalStorage::ListTagsOptions & lhs,
00739     const ILocalStorage::ListTagsOptions & rhs) noexcept;
00740
00741 [[nodiscard]] QUENTIER_EXPORT bool operator==(
00742     const ILocalStorage::ListGuidsFilters & lhs,
00743     const ILocalStorage::ListGuidsFilters & rhs) noexcept;
00744
00745 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
00746     const ILocalStorage::ListGuidsFilters & lhs,
00747     const ILocalStorage::ListGuidsFilters & rhs) noexcept;
00748
00749 } // namespace quentier::local_storage

```

6.20 ILocalStorageNotifier.h

```

00001 /*
00002  * Copyright 2020-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once

```

```

00020
00021 #include <quentier/local_storage/ILocalStorage.h>
00022
00023 #include <QObject>
00024
00025 namespace quentier::local_storage {
00026
00027 class QUENTIER_EXPORT ILocalStorageNotifier : public QObject
00028 {
00029     Q_OBJECT
00030 protected:
00031     explicit ILocalStorageNotifier(QObject * parent = nullptr);
00032
00033 public:
00034     ~ILocalStorageNotifier() override;
00035
00036 Q_SIGNALS:
00037     // Notifications about user related events
00038     void userPut(qevercloud::User user);
00039     void userExpunged(qevercloud::UserID userId);
00040
00041     // Notifications about notebook related events
00042     void notebookPut(qevercloud::Notebook notebook);
00043     void notebookExpunged(QString notebookLocalId);
00044
00045     // Notifications about linked notebooks
00046     void linkedNotebookPut(qevercloud::LinkedNotebook linkedNotebook);
00047     void linkedNotebookExpunged(qevercloud::Guid linkedNotebookGuid);
00048
00049     // Notifications about note related events
00050     void notePut(qevercloud::Note note);
00051
00052     void noteUpdated(
00053         qevercloud::Note note, ILocalStorage::UpdateNoteOptions options);
00054
00055     void noteExpunged(QString noteLocalId);
00056
00057     // Notifications about tag related events
00058     void tagPut(qevercloud::Tag tag);
00059
00060     void tagExpunged(QString tagLocalId, QStringList expungedChildTagLocalIds);
00061
00062     // Notifications about resource related events
00063     void resourcePut(qevercloud::Resource resource);
00064     void resourceMetadataPut(qevercloud::Resource resource);
00065     void resourceExpunged(QString resourceLocalId);
00066
00067     // Notifications about saved search related events
00068     void savedSearchPut(qevercloud::SavedSearch savedSearch);
00069     void savedSearchExpunged(QString savedSearchLocalId);
00070 };
00071
00072 } // namespace quentier::local_storage

```

6.21 IPatch.h

```

00001 /*
00002  * Copyright 2021-2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QFuture>
00024
00025 namespace quentier::local_storage {
00026
00033 class QUENTIER_EXPORT IPatch

```

```

00034 {
00035 public:
00036     virtual ~IPatch() noexcept;
00037
00042     [[nodiscard]] virtual int fromVersion() const noexcept = 0;
00043
00048     [[nodiscard]] virtual int toVersion() const noexcept = 0;
00049
00053     [[nodiscard]] virtual QString patchShortDescription() const = 0;
00054
00058     [[nodiscard]] virtual QString patchLongDescription() const = 0;
00059
00069     [[nodiscard]] virtual QFuture<void> backupLocalStorage() = 0;
00070
00080     [[nodiscard]] virtual QFuture<void> restoreLocalStorageFromBackup() = 0;
00081
00092     [[nodiscard]] virtual QFuture<void> removeLocalStorageBackup() = 0;
00093
00101     [[nodiscard]] virtual QFuture<void> apply() = 0;
00102 };
00103
00104 } // namespace quantier::local_storage

```

6.22 LocalStorageOpenException.h

```

00001 /*
00002  * Copyright 2016-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/exception/IQuantierException.h>
00022
00023 namespace quantier::local_storage {
00024
00029 class QUANTIER_EXPORT LocalStorageOpenException : public IQuantierException
00030 {
00031 public:
00032     explicit LocalStorageOpenException(const ErrorString & message);
00033
00034     [[nodiscard]] LocalStorageOpenException * clone() const override;
00035     void raise() const override;
00036
00037 protected:
00038     [[nodiscard]] QString exceptionDisplayName() const override;
00039 };
00040
00041 } // namespace quantier::local_storage

```

6.23 LocalStorageOperationException.h

```

00001 /*
00002  * Copyright 2016-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.

```

```

00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/exception/IQuentierException.h>
00022
00023 namespace quentier::local_storage {
00024
00025 class QUENTIER_EXPORT LocalStorageOperationException : public IQuentierException
00026 {
00027 public:
00028     explicit LocalStorageOperationException(ErrorString message);
00029
00030     [[nodiscard]] LocalStorageOperationException * clone() const override;
00031     void raise() const override;
00032
00033 protected:
00034     [[nodiscard]] QString exceptionDisplayName() const override;
00035 };
00036
00037 } // namespace quentier::local_storage

```

6.24 NoteSearchQuery.h

```

00001  /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/ErrorString.h>
00022
00023 #include <QList>
00024 #include <QSharedDataPointer>
00025
00026 namespace quentier::local_storage {
00027
00028 class QUENTIER_EXPORT NoteSearchQuery : public utility::Printable
00029 {
00030 public:
00031     explicit NoteSearchQuery();
00032
00033     NoteSearchQuery(const NoteSearchQuery & other);
00034     NoteSearchQuery(NoteSearchQuery && other) noexcept;
00035
00036     NoteSearchQuery & operator=(const NoteSearchQuery & other);
00037     NoteSearchQuery & operator=(NoteSearchQuery && other) noexcept;
00038
00039     ~NoteSearchQuery() override;
00040
00041     [[nodiscard]] bool isEmpty() const;
00042
00043     void clear();
00044
00045     [[nodiscard]] QString queryString() const;
00046
00047     [[nodiscard]] bool setQueryString(
00048         const QString & queryString, ErrorString & error);
00049
00050     [[nodiscard]] QString notebookModifier() const;
00051
00052     [[nodiscard]] bool hasAnyModifier() const;
00053
00054     [[nodiscard]] const QStringList & tagNames() const;
00055     [[nodiscard]] const QStringList & negatedTagNames() const;
00056     [[nodiscard]] bool hasAnyTag() const;

```

```

00065 [[nodiscard]] bool hasNegatedAnyTag() const;
00066
00067 [[nodiscard]] const QStringList & titleNames() const;
00068 [[nodiscard]] const QStringList & negatedTitleNames() const;
00069 [[nodiscard]] bool hasAnyTitleName() const;
00070 [[nodiscard]] bool hasNegatedAnyTitleName() const;
00071
00072 [[nodiscard]] const QList<qint64> & creationTimestamps() const;
00073 [[nodiscard]] const QList<qint64> & negatedCreationTimestamps() const;
00074 [[nodiscard]] bool hasAnyCreationTimestamp() const;
00075 [[nodiscard]] bool hasNegatedAnyCreationTimestamp() const;
00076
00077 [[nodiscard]] const QList<qint64> & modificationTimestamps() const;
00078 [[nodiscard]] const QList<qint64> & negatedModificationTimestamps() const;
00079 [[nodiscard]] bool hasAnyModificationTimestamp() const;
00080 [[nodiscard]] bool hasNegatedAnyModificationTimestamp() const;
00081
00082 [[nodiscard]] const QStringList & resourceMimeTypes() const;
00083 [[nodiscard]] const QStringList & negatedResourceMimeTypes() const;
00084 [[nodiscard]] bool hasAnyResourceMimeType() const;
00085 [[nodiscard]] bool hasNegatedAnyResourceMimeType() const;
00086
00087 [[nodiscard]] const QList<qint64> & subjectDateTimestamps() const;
00088 [[nodiscard]] const QList<qint64> & negatedSubjectDateTimestamps() const;
00089 [[nodiscard]] bool hasAnySubjectDateTimestamp() const;
00090 [[nodiscard]] bool hasNegatedAnySubjectDateTimestamp() const;
00091
00092 [[nodiscard]] const QList<double> & latitudes() const;
00093 [[nodiscard]] const QList<double> & negatedLatitudes() const;
00094 [[nodiscard]] bool hasAnyLatitude() const;
00095 [[nodiscard]] bool hasNegatedAnyLatitude() const;
00096
00097 [[nodiscard]] const QList<double> & longitudes() const;
00098 [[nodiscard]] const QList<double> & negatedLongitudes() const;
00099 [[nodiscard]] bool hasAnyLongitude() const;
00100 [[nodiscard]] bool hasNegatedAnyLongitude() const;
00101
00102 [[nodiscard]] const QList<double> & altitudes() const;
00103 [[nodiscard]] const QList<double> & negatedAltitudes() const;
00104 [[nodiscard]] bool hasAnyAltitude() const;
00105 [[nodiscard]] bool hasNegatedAnyAltitude() const;
00106
00107 [[nodiscard]] const QStringList & authors() const;
00108 [[nodiscard]] const QStringList & negatedAuthors() const;
00109 [[nodiscard]] bool hasAnyAuthor() const;
00110 [[nodiscard]] bool hasNegatedAnyAuthor() const;
00111
00112 [[nodiscard]] const QStringList & sources() const;
00113 [[nodiscard]] const QStringList & negatedSources() const;
00114 [[nodiscard]] bool hasAnySource() const;
00115 [[nodiscard]] bool hasNegatedAnySource() const;
00116
00117 [[nodiscard]] const QStringList & sourceApplications() const;
00118 [[nodiscard]] const QStringList & negatedSourceApplications() const;
00119 [[nodiscard]] bool hasAnySourceApplication() const;
00120 [[nodiscard]] bool hasNegatedAnySourceApplication() const;
00121
00122 [[nodiscard]] const QStringList & contentClasses() const;
00123 [[nodiscard]] const QStringList & negatedContentClasses() const;
00124 [[nodiscard]] bool hasAnyContentClass() const;
00125 [[nodiscard]] bool hasNegatedAnyContentClass() const;
00126
00127 [[nodiscard]] const QStringList & placeNames() const;
00128 [[nodiscard]] const QStringList & negatedPlaceNames() const;
00129 [[nodiscard]] bool hasAnyPlaceName() const;
00130 [[nodiscard]] bool hasNegatedAnyPlaceName() const;
00131
00132 [[nodiscard]] const QStringList & applicationData() const;
00133 [[nodiscard]] const QStringList & negatedApplicationData() const;
00134 [[nodiscard]] bool hasAnyApplicationData() const;
00135 [[nodiscard]] bool hasNegatedAnyApplicationData() const;
00136
00137 [[nodiscard]] const QList<qint64> & reminderOrders() const;
00138 [[nodiscard]] const QList<qint64> & negatedReminderOrders() const;
00139 [[nodiscard]] bool hasAnyReminderOrder() const;
00140 [[nodiscard]] bool hasNegatedAnyReminderOrder() const;
00141
00142 [[nodiscard]] const QList<qint64> & reminderTimes() const;
00143 [[nodiscard]] const QList<qint64> & negatedReminderTimes() const;
00144 [[nodiscard]] bool hasAnyReminderTime() const;
00145 [[nodiscard]] bool hasNegatedAnyReminderTime() const;
00146
00147 [[nodiscard]] const QList<qint64> & reminderDoneTimes() const;
00148 [[nodiscard]] const QList<qint64> & negatedReminderDoneTimes() const;
00149 [[nodiscard]] bool hasAnyReminderDoneTime() const;
00150 [[nodiscard]] bool hasNegatedAnyReminderDoneTime() const;
00151

```



```

00152     [[nodiscard]] bool hasUnfinishedToDo() const;
00153     [[nodiscard]] bool hasNegatedUnfinishedToDo() const;
00154
00155     [[nodiscard]] bool hasFinishedToDo() const;
00156     [[nodiscard]] bool hasNegatedFinishedToDo() const;
00157
00158     [[nodiscard]] bool hasAnyToDo() const;
00159     [[nodiscard]] bool hasNegatedAnyToDo() const;
00160
00161     [[nodiscard]] bool hasEncryption() const;
00162     [[nodiscard]] bool hasNegatedEncryption() const;
00163
00164     [[nodiscard]] const QStringList & contentSearchTerms() const;
00165     [[nodiscard]] const QStringList & negatedContentSearchTerms() const;
00166     [[nodiscard]] bool hasAnyContentSearchTerms() const;
00167
00168     [[nodiscard]] bool isMatcheable() const;
00169
00170     // utility::Printable
00171     QTextStream & print(QTextStream & strm) const override;
00172
00173 private:
00174     class Data;
00175     QSharedDataPointer<Data> d;
00176 };
00177
00178 [[nodiscard]] QUENTIER_EXPORT bool operator==(
00179     const NoteSearchQuery & lhs, const NoteSearchQuery & rhs) noexcept;
00180
00181 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
00182     const NoteSearchQuery & lhs, const NoteSearchQuery & rhs) noexcept;
00183
00184 } // namespace quentier::local_storage

```

6.25 MockILocalStorage.h

```

00001 /*
00002  * Copyright 2021-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/local_storage/ILocalStorage.h>
00022
00023 #include <gmock/gmock.h>
00024
00025 namespace quentier::local_storage::tests::mocks {
00026
00027     class MockILocalStorage : public ILocalStorage
00028     {
00029     public:
00030         MOCK_METHOD(QFuture<bool>, isVersionTooHigh, (), (const, override));
00031         MOCK_METHOD(QFuture<bool>, requiresUpgrade, (), (const, override));
00032
00033         MOCK_METHOD(
00034             QFuture<QList<IPatchPtr>>, requiredPatches, (), (const, override));
00035
00036         MOCK_METHOD(QFuture<qint32>, version, (), (const, override));
00037
00038         MOCK_METHOD(
00039             QFuture<qint32>, highestSupportedVersion, (), (const, override));
00040
00041         MOCK_METHOD(QFuture<quint32>, userCount, (), (const, override));
00042         MOCK_METHOD(QFuture<void>, putUser, (qevercloud::User user), (override));
00043
00044         MOCK_METHOD(
00045             QFuture<std::optional<qevercloud::User>>, findUserById,
00046             (qevercloud::UserID userId), (const, override));
00047

```

```

00048     MOCK_METHOD (
00049         QFuture<void>, expungeUserById, (qevercloud::UserID userId),
00050         (override));
00051
00052     MOCK_METHOD (QFuture<quint32>, notebookCount, (), (const, override));
00053
00054     MOCK_METHOD (
00055         QFuture<void>, putNotebook, (qevercloud::Notebook notebook),
00056         (override));
00057
00058     MOCK_METHOD (
00059         QFuture<std::optional<qevercloud::Notebook>, findNotebookByLocalId,
00060         (QString localId), (const, override));
00061
00062     MOCK_METHOD (
00063         QFuture<std::optional<qevercloud::Notebook>, findNotebookByGuid,
00064         (qevercloud::Guid guid), (const, override));
00065
00066     MOCK_METHOD (
00067         QFuture<std::optional<qevercloud::Notebook>, findNotebookByName,
00068         (QString name, std::optional<qevercloud::Guid> linkedNotebookGuid),
00069         (const, override));
00070
00071     MOCK_METHOD (
00072         QFuture<std::optional<qevercloud::Notebook>, findDefaultNotebook, (),
00073         (const, override));
00074
00075     MOCK_METHOD (
00076         QFuture<void>, expungeNotebookByLocalId, (QString localId), (override));
00077
00078     MOCK_METHOD (
00079         QFuture<void>, expungeNotebookByGuid, (qevercloud::Guid guid),
00080         (override));
00081
00082     MOCK_METHOD (
00083         QFuture<void>, expungeNotebookByName,
00084         (QString name, std::optional<qevercloud::Guid> linkedNotebookGuid),
00085         (override));
00086
00087     MOCK_METHOD (
00088         QFuture<QList<qevercloud::Notebook>, listNotebooks,
00089         (ListNotebooksOptions options), (const, override));
00090
00091     MOCK_METHOD (
00092         QFuture<QList<qevercloud::SharedNotebook>, listSharedNotebooks,
00093         (qevercloud::Guid notebookGuid), (const, override));
00094
00095     MOCK_METHOD (
00096         QFuture<QSet<qevercloud::Guid>, listNotebookGuids,
00097         (ListGuidsFilters filters,
00098         std::optional<qevercloud::Guid> linkedNotebookGuid),
00099         (const, override));
00100
00101     MOCK_METHOD (QFuture<quint32>, linkedNotebookCount, (), (const, override));
00102
00103     MOCK_METHOD (
00104         QFuture<void>, putLinkedNotebook,
00105         (qevercloud::LinkedNotebook linkedNotebook), (override));
00106
00107     MOCK_METHOD (
00108         QFuture<std::optional<qevercloud::LinkedNotebook>,
00109         findLinkedNotebookByGuid, (qevercloud::Guid guid), (const, override));
00110
00111     MOCK_METHOD (
00112         QFuture<void>, expungeLinkedNotebookByGuid, (qevercloud::Guid guid),
00113         (override));
00114
00115     MOCK_METHOD (
00116         QFuture<QList<qevercloud::LinkedNotebook>, listLinkedNotebooks,
00117         (ListLinkedNotebooksOptions options), (const, override));
00118
00119     MOCK_METHOD (
00120         QFuture<quint32>, noteCount, (NoteCountOptions options),
00121         (const, override));
00122
00123     MOCK_METHOD (
00124         QFuture<quint32>, noteCountPerNotebookLocalId,
00125         (QString notebookLocalId, NoteCountOptions options), (const, override));
00126
00127     MOCK_METHOD (
00128         QFuture<quint32>, noteCountPerTagLocalId,
00129         (QString tagLocalId, NoteCountOptions options), (const, override));
00130
00131     MOCK_METHOD (
00132         (QFuture<QHash<QString, quint32>, noteCountsPerTags,
00133         (ListTagsOptions listTagsOptions, NoteCountOptions options),
00134         (const, override));

```

```

00135
00136     MOCK_METHOD (
00137         QFuture<quint32>, noteCountPerNotebookAndTagLocalIds,
00138         (QStringList notebookLocalIds, QStringList tagLocalIds,
00139          NoteCountOptions options),
00140         (const, override));
00141
00142     MOCK_METHOD (QFuture<void>, putNote, (qevercloud::Note note), (override));
00143
00144     MOCK_METHOD (
00145         QFuture<void>, updateNote,
00146         (qevercloud::Note note, UpdateNoteOptions options), (override));
00147
00148     MOCK_METHOD (
00149         QFuture<std::optional<qevercloud::Note>, findNoteByLocalId,
00150         (QString localId, FetchNoteOptions options), (const, override));
00151
00152     MOCK_METHOD (
00153         QFuture<std::optional<qevercloud::Note>, findNoteByGuid,
00154         (qevercloud::Guid guid, FetchNoteOptions options), (const, override));
00155
00156     MOCK_METHOD (
00157         QFuture<void>, expungeNoteByLocalId, (QString localId), (override));
00158
00159     MOCK_METHOD (
00160         QFuture<void>, expungeNoteByGuid, (qevercloud::Guid guid), (override));
00161
00162     MOCK_METHOD (
00163         QFuture<QList<qevercloud::Note>, listNotes,
00164         (FetchNoteOptions fetchOptions, ListNotesOptions options),
00165         (const, override));
00166
00167     MOCK_METHOD (
00168         QFuture<QList<qevercloud::Note>, listNotesPerNotebookLocalId,
00169         (QString notebookLocalId, FetchNoteOptions fetchOptions,
00170          ListNotesOptions options),
00171         (const, override));
00172
00173     MOCK_METHOD (
00174         QFuture<QList<qevercloud::Note>, listNotesPerTagLocalId,
00175         (QString tagLocalId, FetchNoteOptions fetchOptions,
00176          ListNotesOptions options),
00177         (const, override));
00178
00179     MOCK_METHOD (
00180         QFuture<QList<qevercloud::Note>, listNotesPerNotebookAndTagLocalIds,
00181         (QStringList notebookLocalIds, QStringList tagLocalIds,
00182          FetchNoteOptions fetchOptions, ListNotesOptions options),
00183         (const, override));
00184
00185     MOCK_METHOD (
00186         QFuture<QList<qevercloud::Note>, listNotesByLocalIds,
00187         (QStringList noteLocalIds, FetchNoteOptions fetchOptions,
00188          ListNotesOptions options),
00189         (const, override));
00190
00191     MOCK_METHOD (
00192         QFuture<QSet<qevercloud::Guid>, listNoteGuids,
00193         (ListGuidsFilters filters,
00194          std::optional<qevercloud::Guid> linkedNotebookGuid),
00195         (const, override));
00196
00197     MOCK_METHOD (
00198         QFuture<QList<qevercloud::Note>, queryNotes,
00199         (NoteSearchQuery query, FetchNoteOptions fetchOptions),
00200         (const, override));
00201
00202     MOCK_METHOD (
00203         QFuture<QStringList>, queryNoteLocalIds, (NoteSearchQuery query),
00204         (const, override));
00205
00206     MOCK_METHOD (QFuture<quint32>, tagCount, (), (const, override));
00207     MOCK_METHOD (QFuture<void>, putTag, (qevercloud::Tag tag), (override));
00208
00209     MOCK_METHOD (
00210         QFuture<std::optional<qevercloud::Tag>, findTagByLocalId,
00211         (QString tagLocalId), (const, override));
00212
00213     MOCK_METHOD (
00214         QFuture<std::optional<qevercloud::Tag>, findTagByGuid,
00215         (qevercloud::Guid tagGuid), (const, override));
00216
00217     MOCK_METHOD (
00218         QFuture<std::optional<qevercloud::Tag>, findTagByName,
00219         (QString tagName, std::optional<QString> linkedNotebookGuid),
00220         (const, override));
00221

```

```

00222     MOCK_METHOD (
00223         QFuture<QList<qevercloud::Tag>, listTags, (ListTagsOptions options),
00224         (const, override));
00225
00226     MOCK_METHOD (
00227         QFuture<QList<qevercloud::Tag>, listTagsPerNoteLocalId,
00228         (QString noteLocalId, ListTagsOptions options), (const, override));
00229
00230     MOCK_METHOD (
00231         QFuture<QSet<qevercloud::Guid>, listTagGuids,
00232         (ListGuidsFilters filters,
00233         std::optional<qevercloud::Guid> linkedNotebookGuid),
00234         (const, override));
00235
00236     MOCK_METHOD (
00237         QFuture<void>, expungeTagByLocalId, (QString tagLocalId), (override));
00238
00239     MOCK_METHOD (
00240         QFuture<void>, expungeTagByGuid, (qevercloud::Guid tagGuid),
00241         (override));
00242
00243     MOCK_METHOD (
00244         QFuture<void>, expungeTagByName,
00245         (QString name, std::optional<qevercloud::Guid> linkedNotebookGuid),
00246         (override));
00247
00248     MOCK_METHOD (
00249         QFuture<quint32>, resourceCount, (NoteCountOptions options),
00250         (const, override));
00251
00252     MOCK_METHOD (
00253         QFuture<quint32>, resourceCountPerNoteLocalId, (QString noteLocalId),
00254         (const, override));
00255
00256     MOCK_METHOD (
00257         QFuture<void>, putResource, (qevercloud::Resource resource),
00258         (override));
00259
00260     MOCK_METHOD (
00261         QFuture<std::optional<qevercloud::Resource>, findResourceByLocalId,
00262         (QString resourceLocalId, FetchResourceOptions options),
00263         (const, override));
00264
00265     MOCK_METHOD (
00266         QFuture<std::optional<qevercloud::Resource>, findResourceByGuid,
00267         (qevercloud::Guid resourceGuid, FetchResourceOptions options),
00268         (const, override));
00269
00270     MOCK_METHOD (
00271         QFuture<void>, expungeResourceByLocalId, (QString resourceLocalId),
00272         (override));
00273
00274     MOCK_METHOD (
00275         QFuture<void>, expungeResourceByGuid, (qevercloud::Guid resourceGuid),
00276         (override));
00277
00278     MOCK_METHOD (QFuture<quint32>, savedSearchCount, (), (const, override));
00279
00280     MOCK_METHOD (
00281         QFuture<void>, putSavedSearch, (qevercloud::SavedSearch search),
00282         (override));
00283
00284     MOCK_METHOD (
00285         QFuture<std::optional<qevercloud::SavedSearch>,
00286         findSavedSearchByLocalId, (QString localId), (const, override));
00287
00288     MOCK_METHOD (
00289         QFuture<std::optional<qevercloud::SavedSearch>, findSavedSearchByGuid,
00290         (qevercloud::Guid guid), (const, override));
00291
00292     MOCK_METHOD (
00293         QFuture<std::optional<qevercloud::SavedSearch>, findSavedSearchByName,
00294         (QString name), (const, override));
00295
00296     MOCK_METHOD (
00297         QFuture<QList<qevercloud::SavedSearch>, listSavedSearches,
00298         (ListSavedSearchesOptions options), (const, override));
00299
00300     MOCK_METHOD (
00301         QFuture<QSet<qevercloud::Guid>, listSavedSearchGuids,
00302         (ListGuidsFilters filters), (const, override));
00303
00304     MOCK_METHOD (
00305         QFuture<void>, expungeSavedSearchByLocalId, (QString localId),
00306         (override));
00307
00308     MOCK_METHOD (

```

```

00309         QFuture<void>, expungeSavedSearchByGuid, (qevercloud::Guid guid),
00310         (override));
00311
00312     MOCK_METHOD (
00313         QFuture<qint32>, highestUpdateSequenceNumber, (HighestUsnOption option),
00314         (const, override));
00315
00316     MOCK_METHOD (
00317         QFuture<qint32>, highestUpdateSequenceNumber,
00318         (qevercloud::Guid linkedNotebookGuid), (const, override));
00319
00320     MOCK_METHOD (ILocalStorageNotifier *, notifier, (), (const, override));
00321 };
00322
00323 } // namespace quentier::local_storage::tests::mocks

```

6.26 QuentierLogger.h

```

00001 /*
00002  * Copyright 2016-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QDebug>
00024 #include <QRegularExpression>
00025 #include <QString>
00026 #include <QTextStream>
00027
00028 namespace quentier {
00029
00030     enum class LogLevel
00031     {
00032         Trace,
00033         Debug,
00034         Info,
00035         Warning,
00036         Error
00037     };
00038
00039     QUINTIER_EXPORT QDebug & operator<<(QDebug & dbg, LogLevel logLevel);
00040
00041     QUINTIER_EXPORT QTextStream & operator<<(QTextStream & strm, LogLevel logLevel);
00042
00043     void QUINTIER_EXPORT QuentierInitializeLogging();
00044
00045     void QUINTIER_EXPORT QuentierAddLogEntry(
00046         const QString & sourceFileName, int sourceFileLineNumber,
00047         const QString & component, const QString & message, LogLevel logLevel);
00048
00049     LogLevel QUINTIER_EXPORT QuentierMinLogLevel();
00050
00051     void QUINTIER_EXPORT QuentierSetMinLogLevel(LogLevel logLevel);
00052
00053     void QUINTIER_EXPORT QuentierAddStdOutLogDestination();
00054
00055     [[nodiscard]] bool QUINTIER_EXPORT QuentierIsLogLevelActive(LogLevel logLevel);
00056
00057     [[nodiscard]] QString QUINTIER_EXPORT QuentierLogFilesDirPath();
00058
00059     void QUINTIER_EXPORT QuentierRestartLogging();
00060
00061     [[nodiscard]] QRegularExpression QUINTIER_EXPORT QuentierLogComponentFilter();
00062
00063     void QUINTIER_EXPORT
00064         QuentierSetLogComponentFilter(const QRegularExpression & filter);
00065
00066 }

```

```

00108 } // namespace quantier
00109
00110 #define QNLOG_PRIVATE_BASE(component, message, level)           \
00111     if (quantier::QuantierIsLogLevelActive(quantier::LogLevel::level)) { \
00112         QString msg; \
00113         QDebug dbg(&msg); \
00114         dbg.nospace(); \
00115         dbg.noquote(); \
00116         dbg < message; \
00117         quantier::QuantierAddLogEntry( \
00118             QStringLiteral(__FILE__), __LINE__, QString::fromUtf8(component), \
00119             msg, quantier::LogLevel::level); \
00120     } \
00121     // QNLOG_PRIVATE_BASE
00122
00123 #define QNTRACE(component, message) \
00124     QNLOG_PRIVATE_BASE(component, message, Trace) \
00125     // QNTRACE
00126
00127 #define QNDEBUG(component, message) \
00128     QNLOG_PRIVATE_BASE(component, message, Debug) \
00129     // QNDEBUG
00130
00131 #define QNINFO(component, message) \
00132     QNLOG_PRIVATE_BASE(component, message, Info) \
00133     // QNINFO
00134
00135 #define QNWARNING(component, message) \
00136     QNLOG_PRIVATE_BASE(component, message, Warning) \
00137     // QNWARNING
00138
00139 #define QNERROR(component, message) \
00140     QNLOG_PRIVATE_BASE(component, message, Error) \
00141     // QNERROR
00142
00143 #define QUENTIER_SET_MIN_LOG_LEVEL(level) \
00144     quantier::QuantierSetMinLogLevel(quantier::LogLevel::level) \
00145     // QUENTIER_SET_MIN_LOG_LEVEL
00146
00147 #define QUENTIER_INITIALIZE_LOGGING() quantier::QuantierInitializeLogging() \
00148     // QUENTIER_INITIALIZE_LOGGING
00149
00150 // clang-format off
00151 #define QUENTIER_ADD_STDOUT_LOG_DESTINATION() \
00152     quantier::QuantierAddStdOutLogDestination() \
00153     // QUENTIER_ADD_STDOUT_LOG_DESTINATION
00154 // clang-format on
00155
00156 #define QNLOG_FILE_LINENUMBER_DELIMITER ":"

```

6.27 INoteEditorBackend.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/enml/Fwd.h>
00022 #include <quantier/local_storage/Fwd.h>
00023 #include <quantier/utility/Linkage.h>
00024
00025 #include <QPalette>
00026 #include <QPrinter>
00027 #include <QStringList>
00028 #include <QThread>
00029 #include <QWidget>
00030
00031 class QTextStream;

```

```

00032 class QUndoStack;
00033
00034 namespace quentier {
00035
00036 class Account;
00037 class ErrorString;
00038 class NoteEditor;
00039 class SpellChecker;
00040
00041 class QUINTIER_EXPORT INoteEditorBackend
00042 {
00043 public:
00044     virtual ~INoteEditorBackend() noexcept;
00045
00046     virtual void initialize(
00047         local_storage::ILocalStoragePtr localStorage,
00048         SpellChecker & spellChecker, const Account & account,
00049         QThread * backgroundJobsThread,
00050         enml::IDecryptedTextCachePtr decryptedTextCache) = 0;
00051
00052     [[nodiscard]] virtual QObject * object() = 0; // provide QObject interface
00053     [[nodiscard]] virtual QWidget * widget() = 0; // provide QWidget interface
00054
00055     virtual void setAccount(const Account & account) = 0;
00056     virtual void setUndoStack(QUndoStack * pUndoStack) = 0;
00057
00058     virtual void setInitialPageHtml(const QString & html) = 0;
00059     virtual void setNoteNotFoundPageHtml(const QString & html) = 0;
00060     virtual void setNoteDeletedPageHtml(const QString & html) = 0;
00061     virtual void setNoteLoadingPageHtml(const QString & html) = 0;
00062
00063     [[nodiscard]] virtual bool isNoteLoaded() const = 0;
00064     [[nodiscard]] virtual quint64 idleTime() const = 0;
00065
00066     virtual void convertToNote() = 0;
00067     virtual void saveNoteToLocalStorage() = 0;
00068     virtual void setNoteTitle(const QString & noteTitle) = 0;
00069
00070     virtual void setTagIds(
00071         const QStringList & tagLocalUids, const QStringList & tagGuids) = 0;
00072
00073     virtual void undo() = 0;
00074     virtual void redo() = 0;
00075     virtual void cut() = 0;
00076     virtual void copy() = 0;
00077     virtual void paste() = 0;
00078     virtual void pasteUnformatted() = 0;
00079     virtual void selectAll() = 0;
00080
00081     virtual void formatSelectionAsSourceCode() = 0;
00082
00083     virtual void fontMenu() = 0;
00084     virtual void textBold() = 0;
00085     virtual void textItalic() = 0;
00086     virtual void textUnderline() = 0;
00087     virtual void textStrikethrough() = 0;
00088     virtual void textHighlight() = 0;
00089
00090     virtual void alignLeft() = 0;
00091     virtual void alignCenter() = 0;
00092     virtual void alignRight() = 0;
00093     virtual void alignFull() = 0;
00094
00095     [[nodiscard]] virtual QString selectedText() const = 0;
00096     [[nodiscard]] virtual bool hasSelection() const = 0;
00097
00098     virtual void findNext(const QString & text, bool matchCase) const = 0;
00099
00100     virtual void findPrevious(const QString & text, bool matchCase) const = 0;
00101
00102     virtual void replace(
00103         const QString & textToReplace, const QString & replacementText,
00104         bool matchCase) = 0;
00105
00106     virtual void replaceAll(
00107         const QString & textToReplace, const QString & replacementText,
00108         bool matchCase) = 0;
00109
00110     virtual void insertToDoCheckbox() = 0;
00111
00112     virtual void insertInAppNoteLink(
00113         const QString & userId, const QString & shardId,
00114         const QString & noteGuid, const QString & linkText) = 0;
00115
00116     virtual void setSpellcheck(bool enabled) = 0;
00117     [[nodiscard]] virtual bool spellCheckEnabled() const = 0;
00118

```

```

00119     virtual void setFont(const QFont & font) = 0;
00120     virtual void setFontHeight(int height) = 0;
00121     virtual void setFontColor(const QColor & color) = 0;
00122     virtual void setBackgroundColor(const QColor & color) = 0;
00123
00124     [[nodiscard]] virtual QPalette defaultPalette() const = 0;
00125     virtual void setDefaultPalette(const QPalette & pal) = 0;
00126
00127     [[nodiscard]] virtual const QFont * defaultFont() const = 0;
00128     virtual void setDefaultFont(const QFont & font) = 0;
00129
00130     virtual void insertHorizontalLine() = 0;
00131
00132     virtual void increaseFontSize() = 0;
00133     virtual void decreaseFontSize() = 0;
00134
00135     virtual void increaseIndentation() = 0;
00136     virtual void decreaseIndentation() = 0;
00137
00138     virtual void insertBulletedList() = 0;
00139     virtual void insertNumberedList() = 0;
00140
00141     virtual void insertTableDialog() = 0;
00142
00143     virtual void insertFixedWidthTable(
00144         int rows, int columns, int widthInPixels) = 0;
00145
00146     virtual void insertRelativeWidthTable(
00147         int rows, int columns, double relativeWidth) = 0;
00148
00149     virtual void insertTableRow() = 0;
00150     virtual void insertTableColumn() = 0;
00151     virtual void removeTableRow() = 0;
00152     virtual void removeTableColumn() = 0;
00153
00154     virtual void addAttachmentDialog() = 0;
00155     virtual void saveAttachmentDialog(const QByteArray & resourceHash) = 0;
00156     virtual void saveAttachmentUnderCursor() = 0;
00157     virtual void openAttachment(const QByteArray & resourceHash) = 0;
00158     virtual void openAttachmentUnderCursor() = 0;
00159     virtual void copyAttachment(const QByteArray & resourceHash) = 0;
00160     virtual void copyAttachmentUnderCursor() = 0;
00161     virtual void removeAttachment(const QByteArray & resourceHash) = 0;
00162     virtual void removeAttachmentUnderCursor() = 0;
00163     virtual void renameAttachment(const QByteArray & resourceHash) = 0;
00164     virtual void renameAttachmentUnderCursor() = 0;
00165
00166     enum class Rotation
00167     {
00168         Clockwise,
00169         Counterclockwise
00170     };
00171
00172     friend QUENTIER_EXPORT QTextStream & operator<<(
00173         QTextStream & strm, Rotation rotation);
00174
00175     friend QUENTIER_EXPORT QDebug & operator<<(QDebug & dbg, Rotation rotation);
00176
00177     virtual void rotateImageAttachment(
00178         const QByteArray & resourceHash, Rotation rotationDirection) = 0;
00179
00180     virtual void rotateImageAttachmentUnderCursor(
00181         Rotation rotationDirection) = 0;
00182
00183     virtual void encryptSelectedText() = 0;
00184
00185     virtual void decryptEncryptedTextUnderCursor() = 0;
00186
00187     virtual void decryptEncryptedText(
00188         QString encryptedText, QString cipher, QString hint,
00189         QString encryptIndex) = 0;
00190
00191     virtual void hideDecryptedTextUnderCursor() = 0;
00192
00193     virtual void hideDecryptedText(
00194         QString encryptedText, QString decryptedText, QString cipher,
00195         QString hint, QString decryptIndex) = 0;
00196
00197     virtual void editHyperlinkDialog() = 0;
00198     virtual void copyHyperlink() = 0;
00199     virtual void removeHyperlink() = 0;
00200
00201     virtual void onNoteLoadCancelled() = 0;
00202
00203     [[nodiscard]] virtual bool print(
00204         QPrinter & printer, ErrorString & errorDescription) = 0;
00205

```



```

00206     [[nodiscard]] virtual bool exportToPdf(
00207         const QString & absoluteFilePath, ErrorString & errorDescription) = 0;
00208
00209     [[nodiscard]] virtual bool exportToEnex(
00210         const QStringList & tagNames, QString & enex,
00211         ErrorString & errorDescription) = 0;
00212
00213     [[nodiscard]] virtual QString currentNoteLocalId() const = 0;
00214     virtual void setCurrentNoteLocalId(const QString & noteLocalUid) = 0;
00215
00216     virtual void clear() = 0;
00217
00218     [[nodiscard]] virtual bool isModified() const = 0;
00219     [[nodiscard]] virtual bool isEditorPageModified() const = 0;
00220
00221     virtual void setFocusToEditor() = 0;
00222
00223 protected:
00224     INoteEditorBackend(NoteEditor * parent);
00225     NoteEditor * m_pNoteEditor;
00226 };
00227
00228 } // namespace quentier

```

6.28 NoteEditor.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/enml/Fwd.h>
00022 #include <quentier/local_storage/Fwd.h>
00023 #include <quentier/types/ErrorString.h>
00024 #include <quentier/utility/Linkage.h>
00025
00026 #include <qevercloud/types/Note.h>
00027 #include <qevercloud/types/Notebook.h>
00028
00029 #include <QPrinter>
00030 #include <QStringList>
00031 #include <QThread>
00032 #include <QWidget>
00033
00034 class QUndoStack;
00035
00036 namespace quentier {
00037
00038     class Account;
00039     class INoteEditorBackend;
00040     class SpellChecker;
00041
00042     class QUENTIER_EXPORT NoteEditor : public QWidget
00043     {
00044     public:
00045         explicit NoteEditor(
00046             QWidget * parent = nullptr,
00047             #if QT_VERSION >= QT_VERSION_CHECK(5, 15, 0)
00048             Qt::WindowFlags flags = {});
00049         #else
00050             Qt::WindowFlags flags = 0); // NOLINT
00051         #endif
00052
00053         ~NoteEditor() noexcept override;
00054
00055         void initialize(
00056             local_storage::ILocalStoragePtr localStorage,

```

```

00082         SpellChecker & spellChecker, const Account & account,
00083         QThread * backgroundJobsThread = nullptr,
00084         enml::IDecryptedTextCachePtr decryptedTextCache = nullptr);
00085
00086 [[nodiscard]] INoteEditorBackend * backend() noexcept;
00087
00088 void setBackend(INoteEditorBackend * backend);
00089
00090 void setAccount(const Account & account);
00091
00092 [[nodiscard]] const QUndoStack * undoStack() const noexcept;
00093
00094 void setUndoStack(QUndoStack * pUndoStack);
00095
00096 void setInitialPageHtml(const QString & html);
00097
00098 void setNoteNotFoundPageHtml(const QString & html);
00099
00100 void setNoteDeletedPageHtml(const QString & html);
00101
00102 void setNoteLoadingPageHtml(const QString & html);
00103
00104 [[nodiscard]] QString currentNoteLocalId() const;
00105
00106 void setCurrentNoteLocalId(const QString & noteLocalId);
00107
00108 void clear();
00109
00110 [[nodiscard]] bool isModified() const noexcept;
00111
00112 [[nodiscard]] bool isEditorPageModified() const noexcept;
00113
00114 [[nodiscard]] bool isNoteLoaded() const noexcept;
00115
00116 [[nodiscard]] qint64 idleTime() const noexcept;
00117
00118 void setFocus();
00119
00120 [[nodiscard]] QString selectedText() const noexcept;
00121 [[nodiscard]] bool hasSelection() const noexcept;
00122
00123 [[nodiscard]] bool spellCheckEnabled() const noexcept;
00124
00125 [[nodiscard]] bool print(
00126     QPrinter & printer, ErrorString & errorDescription);
00127
00128 [[nodiscard]] bool exportToPdf(
00129     const QString & absoluteFilePath, ErrorString & errorDescription);
00130
00131 [[nodiscard]] bool exportToEnex(
00132     const QStringList & tagNames, QString & enex,
00133     ErrorString & errorDescription);
00134
00135 [[nodiscard]] QPalette defaultPalette() const;
00136
00137 [[nodiscard]] const QFont * defaultFont() const;
00138
00139 Q_SIGNALS:
00140 void contentChanged();
00141
00142 void noteAndNotebookFoundInLocalStorage(
00143     qevercloud::Note note, qevercloud::Notebook notebook);
00144
00145 void noteNotFound(QString noteLocalId);
00146
00147 void noteDeleted(QString noteLocalId);
00148
00149 void noteModified();
00150
00151 void notifyError(ErrorString error);
00152
00153 void inAppNoteLinkClicked(
00154     QString userId, QString shardId, QString noteGuid);
00155
00156 void inAppNoteLinkPasteRequested(
00157     QString url, QString userId, QString shardId, QString noteGuid);
00158
00159 void convertedToNote(qevercloud::Note note);
00160 void cantConvertToNote(ErrorString error);
00161
00162 void noteEditorHtmlUpdated(QString html);
00163
00164 void currentNoteChanged(qevercloud::Note note);
00165
00166 void spellCheckerNotReady();
00167 void spellCheckerReady();

```

```

00288     void noteLoaded();
00289
00296     void noteSavedToLocalStorage(QString noteLocalId);
00297
00302     void failedToSaveNoteToLocalStorage(
00303         QString errorDescription, QString noteLocalId);
00304
00305     // Signals to notify anyone interested of the formatting at the current
00306     // cursor position
00307     void textBoldState(bool state);
00308     void textItalicState(bool state);
00309     void textUnderlineState(bool state);
00310     void textStrikethroughState(bool state);
00311     void textAlignLeftState(bool state);
00312     void textAlignCenterState(bool state);
00313     void textAlignRightState(bool state);
00314     void textAlignFullState(bool state);
00315     void textInsideOrderedListState(bool state);
00316     void textInsideUnorderedListState(bool state);
00317     void textInsideTableState(bool state);
00318
00319     void textFontFamilyChanged(QString fontFamily);
00320     void textFontSizeChanged(int fontSize);
00321
00322     void insertTableDialogRequested();
00323
00324 public Q_SLOTS:
00330     void convertToNote();
00331
00340     void saveNoteToLocalStorage();
00341
00351     void setNoteTitle(const QString & noteTitle);
00352
00364     void setTagIds(
00365         const QStringList & tagLocalIds, const QStringList & tagGuids);
00366
00367     void undo();
00368     void redo();
00369     void cut();
00370     void copy();
00371     void paste();
00372     void pasteUnformatted();
00373     void selectAll();
00374
00375     void formatSelectionAsSourceCode();
00376
00377     void fontMenu();
00378     void textBold();
00379     void textItalic();
00380     void textUnderline();
00381     void textStrikethrough();
00382     void textHighlight();
00383
00384     void alignLeft();
00385     void alignCenter();
00386     void alignRight();
00387     void alignFull();
00388
00389     void findNext(const QString & text, bool matchCase) const;
00390     void findPrevious(const QString & text, bool matchCase) const;
00391
00392     void replace(
00393         const QString & textToReplace, const QString & replacementText,
00394         bool matchCase);
00395
00396     void replaceAll(
00397         const QString & textToReplace, const QString & replacementText,
00398         bool matchCase);
00399
00400     void insertToDoCheckbox();
00401
00402     void insertInAppNoteLink(
00403         const QString & userId, const QString & shardId,
00404         const QString & noteGuid, const QString & linkText);
00405
00406     void setSpellcheck(bool enabled);
00407
00408     void setFont(const QFont & font);
00409     void setFontHeight(int height);
00410     void setFontColor(const QColor & color);
00411     void setBackgroundColor(const QColor & color);
00412
00428     void setDefaultPalette(const QPalette & pal);
00429
00435     void setDefaultFont(const QFont & font);
00436
00437     void insertHorizontalLine();

```

```

00438
00439     void increaseFontSize();
00440     void decreaseFontSize();
00441
00442     void increaseIndentation();
00443     void decreaseIndentation();
00444
00445     void insertBulletedList();
00446     void insertNumberedList();
00447
00448     void insertTableDialog();
00449
00450     void insertFixedWidthTable(int rows, int columns, int widthInPixels);
00451
00452     void insertRelativeWidthTable(int rows, int columns, double relativeWidth);
00453
00454     void insertTableRow();
00455     void insertTableColumn();
00456     void removeTableRow();
00457     void removeTableColumn();
00458
00459     void addAttachmentDialog();
00460     void saveAttachmentDialog(const QByteArray & resourceHash);
00461     void saveAttachmentUnderCursor();
00462     void openAttachment(const QByteArray & resourceHash);
00463     void openAttachmentUnderCursor();
00464     void copyAttachment(const QByteArray & resourceHash);
00465     void copyAttachmentUnderCursor();
00466
00467     void encryptSelectedText();
00468     void decryptEncryptedTextUnderCursor();
00469
00470     void editHyperlinkDialog();
00471     void copyHyperlink();
00472     void removeHyperlink();
00473
00474     void onNoteLoadCancelled();
00475
00476 protected:
00477     void dragMoveEvent(QDragMoveEvent * pEvent) override;
00478     void dropEvent(QDropEvent * pEvent) override;
00479
00480 private:
00481     INoteEditorBackend * m_backend;
00482 };
00483
00484 } // namespace quantier

```

6.29 SpellChecker.h

```

00001 /*
00002  * Copyright 2017-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/utility/Fwd.h>
00022 #include <quantier/utility/Linkage.h>
00023
00024 #include <QList>
00025 #include <QObject>
00026
00027 #include <utility>
00028
00029 namespace quantier {
00030
00031     class Account;
00032     class FileIOProcessorAsync;
00033     class SpellCheckerPrivate;

```

```

00034
00035 class QUENTIER_EXPORT SpellChecker : public QObject
00036 {
00037     Q_OBJECT
00038 public:
00039     SpellChecker(
00040         utility::FileIOProcessorAsync * fileIOProcessorAsync, Account account,
00041         QObject * parent = nullptr, const QString & userDictionaryPath = {});
00042
00043     // The second bool in the pair indicates whether the dictionary
00044     // is enabled or disabled
00045     [[nodiscard]] QList<std::pair<QString, bool>> listAvailableDictionaries()
00046         const;
00047
00048     void setAccount(const Account & account);
00049
00050     void enableDictionary(const QString & language);
00051     void disableDictionary(const QString & language);
00052
00053     [[nodiscard]] bool checkSpell(const QString & word) const;
00054
00055     [[nodiscard]] QStringList spellCorrectionSuggestions(
00056         const QString & missSpelledWord) const;
00057
00058     void addToUserWordlist(const QString & word);
00059     void removeFromUserWordList(const QString & word);
00060     void ignoreWord(const QString & word);
00061     void removeWord(const QString & word);
00062
00063     [[nodiscard]] bool isReady() const noexcept;
00064
00065 Q_SIGNALS:
00066     void ready();
00067
00068 private:
00069     SpellCheckerPrivate * const d_ptr;
00070     Q_DECLARE_PRIVATE(SpellChecker)
00071 };
00072
00073 } // namespace quentier

```

6.30 IAuthenticator.h

```

00001 /*
00002  * Copyright 2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Fwd.h>
00022 #include <quentier/types/Account.h>
00023 #include <quentier/utility/Linkage.h>
00024
00025 #include <QFuture>
00026
00027 namespace quentier::synchronization {
00028
00029 class QUENTIER_EXPORT IAuthenticator
00030 {
00031 public:
00032     virtual ~IAuthenticator() noexcept;
00033
00034     [[nodiscard]] virtual QFuture<IAuthenticationInfoPtr>
00035         authenticateNewAccount() = 0;
00036
00037     [[nodiscard]] virtual QFuture<IAuthenticationInfoPtr> authenticateAccount(
00038         Account account) = 0;
00039 };
00040
00041 } // namespace quentier::synchronization

```

6.31 INoteStoreFactory.h

```

00001 /*
00002  * Copyright 2022-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <qevercloud/Fwd.h>
00024 #include <qevercloud/services/Fwd.h>
00025 #include <qevercloud/types/TypeAliases.h>
00026
00027 #include <optional>
00028
00029 namespace quentier::synchronization {
00030
00031 class QUENTIER_EXPORT INoteStoreFactory
00032 {
00033 public:
00034     virtual ~INoteStoreFactory();
00035
00036     [[nodiscard]] virtual qevercloud::INoteStorePtr createNoteStore(
00037         QString noteStoreUrl = {},
00038         std::optional<qevercloud::Guid> linkedNotebookGuid = {},
00039         qevercloud::IRequestContextPtr ctx = {},
00040         qevercloud::IRetryPolicyPtr retryPolicy = {}) = 0;
00041 };
00042
00043 } // namespace quentier::synchronization

```

6.32 ISyncConflictResolver.h

```

00001 /*
00002  * Copyright 2021-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <qevercloud/types/Note.h>
00024 #include <qevercloud/types/Notebook.h>
00025 #include <qevercloud/types/SavedSearch.h>
00026 #include <qevercloud/types/Tag.h>
00027
00028 #include <QFuture>
00029
00030 #include <variant>
00031
00032 class QDebug;
00033 class QTextStream;
00034

```

```

00035 namespace quantier::synchronization {
00036
00041 class QUENTIER_EXPORT ISyncConflictResolver
00042 {
00043 public:
00048     struct QUENTIER_EXPORT ConflictResolution
00049     {
00054         struct QUENTIER_EXPORT UseTheirs
00055         {};
00056
00061         struct QUENTIER_EXPORT UseMine
00062         {};
00063
00069         struct QUENTIER_EXPORT IgnoreMine
00070         {};
00071
00080         template <class T>
00081         struct MoveMine
00082         {
00083             using value_type = T;
00084
00088             T mine;
00089         };
00090     };
00091
00092     using NotebookConflictResolution = std::variant<
00093         ConflictResolution::UseTheirs, ConflictResolution::UseMine,
00094         ConflictResolution::IgnoreMine,
00095         ConflictResolution::MoveMine<qevercloud::Notebook>>;
00096
00097     using NoteConflictResolution = std::variant<
00098         ConflictResolution::UseTheirs, ConflictResolution::UseMine,
00099         ConflictResolution::IgnoreMine,
00100         ConflictResolution::MoveMine<qevercloud::Note>>;
00101
00102     using SavedSearchConflictResolution = std::variant<
00103         ConflictResolution::UseTheirs, ConflictResolution::UseMine,
00104         ConflictResolution::IgnoreMine,
00105         ConflictResolution::MoveMine<qevercloud::SavedSearch>>;
00106
00107     using TagConflictResolution = std::variant<
00108         ConflictResolution::IgnoreMine, ConflictResolution::UseTheirs,
00109         ConflictResolution::UseMine,
00110         ConflictResolution::MoveMine<qevercloud::Tag>>;
00111
00112 public:
00113     virtual ~ISyncConflictResolver() noexcept;
00114
00115     [[nodiscard]] virtual QFuture<NotebookConflictResolution>
00116         resolveNotebookConflict(
00117         qevercloud::Notebook theirs, qevercloud::Notebook mine) = 0;
00118
00119     [[nodiscard]] virtual QFuture<NoteConflictResolution> resolveNoteConflict(
00120         qevercloud::Note theirs, qevercloud::Note mine) = 0;
00121
00122     [[nodiscard]] virtual QFuture<SavedSearchConflictResolution>
00123         resolveSavedSearchConflict(
00124         qevercloud::SavedSearch theirs, qevercloud::SavedSearch mine) = 0;
00125
00126     [[nodiscard]] virtual QFuture<TagConflictResolution> resolveTagConflict(
00127         qevercloud::Tag theirs, qevercloud::Tag mine) = 0;
00128 };
00129
00130 QUENTIER_EXPORT QTextStream & operator<<(
00131     QTextStream & strm,
00132     const ISyncConflictResolver::NotebookConflictResolution & resolution);
00133
00134 QUENTIER_EXPORT QDebug & operator<<(
00135     QDebug & dbg,
00136     const ISyncConflictResolver::NotebookConflictResolution & resolution);
00137
00138 QUENTIER_EXPORT QTextStream & operator<<(
00139     QTextStream & strm,
00140     const ISyncConflictResolver::NoteConflictResolution & resolution);
00141
00142 QUENTIER_EXPORT QDebug & operator<<(
00143     QDebug & dbg,
00144     const ISyncConflictResolver::NoteConflictResolution & resolution);
00145
00146 QUENTIER_EXPORT QTextStream & operator<<(
00147     QTextStream & strm,
00148     const ISyncConflictResolver::SavedSearchConflictResolution & resolution);
00149
00150 QUENTIER_EXPORT QDebug & operator<<(
00151     QDebug & dbg,
00152     const ISyncConflictResolver::SavedSearchConflictResolution & resolution);
00153

```

```

00154 QUENTIER_EXPORT QTextStream & operator«(
00155     QTextStream & strm,
00156     const ISyncConflictResolver::TagConflictResolution & resolution);
00157
00158 QUENTIER_EXPORT QDebug & operator«(
00159     QDebug & dbg,
00160     const ISyncConflictResolver::TagConflictResolution & resolution);
00161
00162 } // namespace quentier::synchronization

```

6.33 ISyncEventsNotifier.h

```

00001 /*
00002  * Copyright 2023-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/Fwd.h>
00022 #include <quentier/synchronization/types/Fwd.h>
00023 #include <quentier/utility/Linkage.h>
00024
00025 #include <qevercloud/types/LinkedNotebook.h>
00026
00027 #include <QList>
00028 #include <QObject>
00029
00030 namespace quentier::synchronization {
00031
00032 class QUENTIER_EXPORT ISyncEventsNotifier : public QObject
00033 {
00034     Q_OBJECT
00035 protected:
00036     explicit ISyncEventsNotifier(QObject * parent = nullptr);
00037
00038 public:
00039     ~ISyncEventsNotifier() override;
00040
00041     Q_SIGNALS:
00042     void syncChunksDownloadProgress(
00043         qint32 highestDownloadedUsn, qint32 highestServerUsn,
00044         qint32 lastPreviousUsn);
00045
00046     void syncChunksDownloaded();
00047
00048     void syncChunksDataProcessingProgress(ISyncChunksDataCountersPtr counters);
00049
00050     void startLinkedNotebooksDataDownloading(
00051         const QList<qevercloud::LinkedNotebook> & linkedNotebooks);
00052
00053     void linkedNotebookSyncChunksDownloadProgress(
00054         qint32 highestDownloadedUsn, qint32 highestServerUsn,
00055         qint32 lastPreviousUsn,
00056         const qevercloud::LinkedNotebook & linkedNotebook);
00057
00058     void linkedNotebookSyncChunksDownloaded(
00059         const qevercloud::LinkedNotebook & linkedNotebook);
00060
00061     void linkedNotebookSyncChunksDataProcessingProgress(
00062         ISyncChunksDataCountersPtr counters,
00063         const qevercloud::LinkedNotebook & linkedNotebook);
00064
00065     void notesDownloadProgress(
00066         quint32 notesDownloaded, quint32 totalNotesToDownload);
00067
00068     void linkedNotebookNotesDownloadProgress(
00069         quint32 notesDownloaded, quint32 totalNotesToDownload,
00070         const qevercloud::LinkedNotebook & linkedNotebook);
00071
00072 };

```



```

00165     void resourcesDownloadProgress(
00166         quint32 resourcesDownloaded, quint32 totalResourcesToDownload);
00167
00179     void linkedNotebookResourcesDownloadProgress(
00180         quint32 resourcesDownloaded, quint32 totalResourcesToDownload,
00181         const qevercloud::LinkedNotebook & linkedNotebook);
00182
00191     void downloadFinished(bool dataDownloaded);
00192
00199     void userOwnSendStatusUpdate(ISendStatusPtr sendStatus);
00200
00209     void linkedNotebookSendStatusUpdate(
00210         const qevercloud::Guid & linkedNotebookGuid, ISendStatusPtr sendStatus);
00211 };
00212
00213 } // namespace quantier::synchronization

```

6.34 ISynchronizer.h

```

00001 /*
00002  * Copyright 2021-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/local_storage/Fwd.h>
00022 #include <quantier/synchronization/Fwd.h>
00023 #include <quantier/synchronization/types/Fwd.h>
00024 #include <quantier/types/Account.h>
00025 #include <quantier/utility/Linkage.h>
00026 #include <quantier/utility/cancelers/Fwd.h>
00027
00028 #include <qevercloud/types/TypeAliases.h>
00029
00030 #include <QFuture>
00031
00032 #include <memory>
00033 #include <utility>
00034
00035 namespace quantier {
00036
00037     class Account;
00038
00039 } // namespace quantier
00040
00041 namespace quantier::synchronization {
00042
00043     class QUENTIER_EXPORT ISynchronizer
00044     {
00045     public:
00046         virtual ~ISynchronizer() noexcept;
00047
00048         [[nodiscard]] virtual QFuture<std::pair<Account, IAuthenticationInfoPtr>>
00049             authenticateNewAccount() = 0;
00050
00051         [[nodiscard]] virtual QFuture<IAuthenticationInfoPtr> authenticateAccount(
00052             Account account) = 0;
00053
00054         using SyncResult =
00055             std::pair<QFuture<ISyncResultPtr>, ISyncEventsNotifier *>;
00056
00057         [[nodiscard]] virtual SyncResult synchronizeAccount(
00058             Account account, local_storage::ILocalStoragePtr localStorage,
00059             utility::cancelers::ICancelerPtr canceler,
00060             ISyncOptionsPtr options = nullptr,
00061             ISyncConflictResolverPtr syncConflictResolver = nullptr) = 0;
00062
00063         virtual void revokeAuthentication(qevercloud::UserID userId) = 0;
00064     };

```

```
00065
00066 } // namespace quantier::synchronization
```

6.35 ISyncStateStorage.h

```
00001 /*
00002  * Copyright 2020-2023 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/synchronization/types/Fwd.h>
00022 #include <quantier/types/Account.h>
00023 #include <quantier/utility/Linkage.h>
00024
00025 #include <QObject>
00026
00027 namespace quantier::synchronization {
00028
00034 class QUANTIER_EXPORT ISyncStateStorage : public QObject
00035 {
00036     Q_OBJECT
00037 protected:
00038     explicit ISyncStateStorage(QObject * parent = nullptr);
00039
00040 public:
00041     ~ISyncStateStorage() override;
00042
00043     [[nodiscard]] virtual ISyncStatePtr getSyncState(
00044         const Account & account) = 0;
00045
00046     virtual void setSyncState(
00047         const Account & account, ISyncStatePtr syncState) = 0;
00048
00049 Q_SIGNALS:
00055     void notifySyncStateUpdated(Account account, ISyncStatePtr syncState);
00056 };
00057
00058 } // namespace quantier::synchronization
```

6.36 IUserStoreFactory.h

```
00001 /*
00002  * Copyright 2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/utility/Linkage.h>
00022
```

```

00023 #include <qevercloud/Fwd.h>
00024 #include <qevercloud/services/Fwd.h>
00025
00026 #include <QString>
00027
00028 namespace quantier::synchronization {
00029
00029     class QUENTIER_EXPORT IUserStoreFactory
00030     {
00031     public:
00032         virtual ~IUserStoreFactory();
00033
00034         [[nodiscard]] virtual qevercloud::IUserStorePtr createUserStore(
00035             QString userStoreUrl = {}, qevercloud::IRequestContextPtr ctx = {},
00036             qevercloud::IRetryPolicyPtr retryPolicy = {}) = 0;
00037     };
00038
00039 } // namespace quantier::synchronization

```

6.37 MockIAuthenticator.h

```

00001 /*
00002  * Copyright 2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/synchronization/IAuthenticator.h>
00022
00023 #include <gmock/gmock.h>
00024
00025 namespace quantier::synchronization::tests::mocks {
00026
00027     class MockIAuthenticator : public IAuthenticator
00028     {
00029     public:
00030         MOCK_METHOD(
00031             QFuture<IAuthenticationInfoPtr>, authenticateNewAccount, (),
00032             (override));
00033
00034         MOCK_METHOD(
00035             QFuture<IAuthenticationInfoPtr>, authenticateAccount, (Account account),
00036             (override));
00037     };
00038
00039 } // namespace quantier::synchronization::tests::mocks

```

6.38 MockINoteStoreFactory.h

```

00001 /*
00002  * Copyright 2022-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License

```

```

00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/INoteStoreFactory.h>
00022
00023 #include <gmock/gmock.h>
00024
00025 namespace quentier::synchronization::tests::mocks {
00026
00027 class MockINoteStoreFactory : public INoteStoreFactory
00028 {
00029 public:
00030     MOCK_METHOD(
00031         ::qevercloud::INoteStorePtr, createNoteStore,
00032         (QString noteStoreUrl,
00033          std::optional<::qevercloud::Guid> linkedNotebookGuid,
00034          ::qevercloud::IRequestContextPtr ctx,
00035          ::qevercloud::IRetryPolicyPtr retryPolicy),
00036         (override));
00037 };
00038
00039 } // namespace quentier::synchronization::tests::mocks

```

6.39 MockISyncConflictResolver.h

```

00001 /*
00002  * Copyright 2022-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/ISyncConflictResolver.h>
00022
00023 #include <gmock/gmock.h>
00024
00025 namespace quentier::synchronization::tests::mocks {
00026
00027 class MockISyncConflictResolver : public ISyncConflictResolver
00028 {
00029 public:
00030     MOCK_METHOD(
00031         QFuture<NotebookConflictResolution>, resolveNotebookConflict,
00032         (::qevercloud::Notebook theirs, ::qevercloud::Notebook mine),
00033         (override));
00034
00035     MOCK_METHOD(
00036         QFuture<NoteConflictResolution>, resolveNoteConflict,
00037         (::qevercloud::Note theirs, ::qevercloud::Note mine), (override));
00038
00039     MOCK_METHOD(
00040         QFuture<SavedSearchConflictResolution>, resolveSavedSearchConflict,
00041         (::qevercloud::SavedSearch theirs, ::qevercloud::SavedSearch mine),
00042         (override));
00043
00044     MOCK_METHOD(
00045         QFuture<TagConflictResolution>, resolveTagConflict,
00046         (::qevercloud::Tag theirs, ::qevercloud::Tag mine), (override));
00047 };
00048
00049 } // namespace quentier::synchronization::tests::mocks

```

6.40 MockISyncStateStorage.h

```

00001 /*

```

```

00002  * Copyright 2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/ISyncStateStorage.h>
00022
00023 #include <gmock/gmock.h>
00024
00025 namespace quentier::synchronization::tests::mocks {
00026
00027 class MockISyncStateStorage : public ISyncStateStorage
00028 {
00029     Q_OBJECT
00030 public:
00031     MOCK_METHOD(
00032         ISyncStatePtr, getSyncState, (const Account & account), (override));
00033
00034     MOCK_METHOD(
00035         void, setSyncState, (const Account & account, ISyncStatePtr syncState),
00036         (override));
00037 };
00038
00039 } // namespace quentier::synchronization::tests::mocks

```

6.41 Errors.h

```

00001 /*
00002  * Copyright 2022-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QtGlobal>
00024
00025 #include <optional>
00026 #include <variant>
00027
00028 namespace quentier::synchronization {
00029
00030 struct QUENTIER_EXPORT RateLimitReachedError
00031 {
00032     std::optional<qint32> rateLimitDurationSec;
00033 };
00034
00035 [[nodiscard]] QUENTIER_EXPORT bool operator==(
00036     const RateLimitReachedError & lhs,
00037     const RateLimitReachedError & rhs) noexcept;
00038
00039 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
00040     const RateLimitReachedError & lhs,
00041     const RateLimitReachedError & rhs) noexcept;
00042
00043 }

```

```

00060 struct QUINTIER_EXPORT AuthenticationExpiredError
00061 {};
00062
00063 [[nodiscard]] QUINTIER_EXPORT bool operator==(
00064     const AuthenticationExpiredError & lhs,
00065     const AuthenticationExpiredError & rhs) noexcept;
00066
00067 [[nodiscard]] QUINTIER_EXPORT bool operator!=(
00068     const AuthenticationExpiredError & lhs,
00069     const AuthenticationExpiredError & rhs) noexcept;
00070
00076 using StopSynchronizationError = std::variant<
00077     RateLimitReachedError, AuthenticationExpiredError, std::monostate>;
00078
00079 } // namespace quentier::synchronization

```

6.42 IAuthenticationInfo.h

```

00001 /*
00002  * Copyright 2022-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023 #include <quentier/utility/Printable.h>
00024
00025 #include <gevercloud/types/TypeAliases.h>
00026
00027 #include <QList>
00028 #include <QNetworkCookie>
00029 #include <QString>
00030
00031 namespace quentier::synchronization {
00032
00037 class QUINTIER_EXPORT IAuthenticationInfo : public utility::Printable
00038 {
00039 public:
00043     [[nodiscard]] virtual gevercloud::UserID userId() const = 0;
00044
00048     [[nodiscard]] virtual QString authToken() const = 0;
00049
00053     [[nodiscard]] virtual gevercloud::Timestamp authTokenExpirationTime()
00054         const = 0;
00055
00059     [[nodiscard]] virtual gevercloud::Timestamp authenticationTime() const = 0;
00060
00065     [[nodiscard]] virtual QString shardId() const = 0;
00066
00070     [[nodiscard]] virtual QString noteStoreUrl() const = 0;
00071
00076     [[nodiscard]] virtual QString webApiUrlPrefix() const = 0;
00077
00084     [[nodiscard]] virtual QList<QNetworkCookie> userStoreCookies() const = 0;
00085 };
00086
00087 } // namespace quentier::synchronization

```

6.43 IAuthenticationInfoBuilder.h

```

00001 /*
00002  * Copyright 2023 Dmitry Ivanov
00003  *

```

```

00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <qevercloud/types/TypeAliases.h>
00025
00026 #include <QList>
00027 #include <QNetworkCookie>
00028 #include <QString>
00029
00030 namespace quentier::synchronization {
00031
00032 class QUENTIER_EXPORT IAuthenticationInfoBuilder
00033 {
00034 public:
00035     virtual ~IAuthenticationInfoBuilder() noexcept;
00036
00037     virtual IAuthenticationInfoBuilder & setUserId(
00038         qevercloud::UserID userId) = 0;
00039
00040     virtual IAuthenticationInfoBuilder & setAuthToken(QString token) = 0;
00041
00042     virtual IAuthenticationInfoBuilder & setAuthTokenExpirationTime(
00043         qevercloud::Timestamp expirationTime) = 0;
00044
00045     virtual IAuthenticationInfoBuilder & setAuthenticationTime(
00046         qevercloud::Timestamp authenticationTime) = 0;
00047
00048     virtual IAuthenticationInfoBuilder & setShardId(QString shardId) = 0;
00049
00050     virtual IAuthenticationInfoBuilder & setNoteStoreUrl(
00051         QString noteStoreUrl) = 0;
00052
00053     virtual IAuthenticationInfoBuilder & setWebApiUrlPrefix(
00054         QString webApiUrlPrefix) = 0;
00055
00056     virtual IAuthenticationInfoBuilder & setUserStoreCookies(
00057         QList<QNetworkCookie> cookies) = 0;
00058
00059     [[nodiscard]] virtual IAuthenticationInfoPtr build() = 0;
00060 };
00061
00062 [[nodiscard]] QUENTIER_EXPORT IAuthenticationInfoBuilderPtr
00063 createAuthenticationInfoBuilder();
00064
00065 } // namespace quentier::synchronization

```

6.44 IDownloadNotesStatus.h

```

00001 /*
00002  * Copyright 2022-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018

```

```

00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Errors.h>
00022 #include <quentier/synchronization/types/Fwd.h>
00023 #include <quentier/utility/Linkage.h>
00024 #include <quentier/utility/Printable.h>
00025
00026 #include <qevercloud/types/Note.h>
00027 #include <qevercloud/types/TypeAliases.h>
00028
00029 #include <QException>
00030 #include <QList>
00031
00032 #include <memory>
00033 #include <utility>
00034
00035 namespace quentier::synchronization {
00036
00041 class QUINTIER_EXPORT IDownloadNotesStatus : public utility::Printable
00042 {
00043 public:
00044     using QExceptionPtr = std::shared_ptr<QException>;
00045     using NoteWithException = std::pair<qevercloud::Note, QExceptionPtr>;
00046     using GuidWithException = std::pair<qevercloud::Guid, QExceptionPtr>;
00047     using UpdateSequenceNumbersByGuid = QHash<qevercloud::Guid, qint32>;
00048
00049     [[nodiscard]] virtual quint64 totalNewNotes() const = 0;
00050     [[nodiscard]] virtual quint64 totalUpdatedNotes() const = 0;
00051     [[nodiscard]] virtual quint64 totalExpungedNotes() const = 0;
00052
00053     [[nodiscard]] virtual QList<NoteWithException> notesWhichFailedToDownload()
00054         const = 0;
00055
00056     [[nodiscard]] virtual QList<NoteWithException> notesWhichFailedToProcess()
00057         const = 0;
00058
00059     [[nodiscard]] virtual QList<GuidWithException>
00060         noteGuidsWhichFailedToExpunge() const = 0;
00061
00062     [[nodiscard]] virtual UpdateSequenceNumbersByGuid
00063         processedNoteGuidsAndUsns() const = 0;
00064
00065     [[nodiscard]] virtual UpdateSequenceNumbersByGuid
00066         cancelledNoteGuidsAndUsns() const = 0;
00067
00068     [[nodiscard]] virtual QList<qevercloud::Guid> expungedNoteGuids() const = 0;
00069
00070     [[nodiscard]] virtual StopSynchronizationError stopSynchronizationError()
00071         const = 0;
00072 };
00073
00074 } // namespace quentier::synchronization

```

6.45 IDownloadResourcesStatus.h

```

00001 /*
00002  * Copyright 2022-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Errors.h>
00022 #include <quentier/utility/Linkage.h>
00023 #include <quentier/utility/Printable.h>
00024
00025 #include <qevercloud/types/Resource.h>
00026 #include <qevercloud/types/TypeAliases.h>
00027
00028 #include <QException>

```



```

00029
00030 #include <memory>
00031 #include <utility>
00032
00033 namespace quentier::synchronization {
00034
00035 class QUENTIER_EXPORT IDownloadResourcesStatus : public utility::Printable
00036 {
00037 public:
00038     ~IDownloadResourcesStatus() noexcept override;
00039
00040     using QExceptionPtr = std::shared_ptr<QException>;
00041
00042     using ResourceWithException =
00043         std::pair<qevercloud::Resource, QExceptionPtr>;
00044
00045     using UpdateSequenceNumbersByGuid = QHash<qevercloud::Guid, qint32>;
00046
00047     [[nodiscard]] virtual quint64 totalNewResources() const = 0;
00048     [[nodiscard]] virtual quint64 totalUpdatedResources() const = 0;
00049
00050     [[nodiscard]] virtual QList<ResourceWithException>
00051         resourcesWhichFailedToDownload() const = 0;
00052
00053     [[nodiscard]] virtual QList<ResourceWithException>
00054         resourcesWhichFailedToProcess() const = 0;
00055
00056     [[nodiscard]] virtual UpdateSequenceNumbersByGuid
00057         processedResourceGuidsAndUsns() const = 0;
00058
00059     [[nodiscard]] virtual UpdateSequenceNumbersByGuid
00060         cancelledResourceGuidsAndUsns() const = 0;
00061
00062     [[nodiscard]] virtual StopSynchronizationError stopSynchronizationError()
00063         const = 0;
00064 };
00065
00066 } // namespace quentier::synchronization

```

6.46 ISendStatus.h

```

00001 /*
00002  * Copyright 2022-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Errors.h>
00022 #include <quentier/utility/Linkage.h>
00023 #include <quentier/utility/Printable.h>
00024
00025 #include <qevercloud/types/Note.h>
00026 #include <qevercloud/types/Notebook.h>
00027 #include <qevercloud/types/SavedSearch.h>
00028 #include <qevercloud/types/Tag.h>
00029 #include <qevercloud/types/TypeAliases.h>
00030
00031 #include <QException>
00032 #include <QList>
00033
00034 #include <memory>
00035 #include <utility>
00036
00037 namespace quentier::synchronization {
00038
00039 class QUENTIER_EXPORT ISendStatus : public utility::Printable
00040 {
00041 public:
00042     using QExceptionPtr = std::shared_ptr<QException>;

```

```

00048
00049     using NoteWithException = std::pair<qevercloud::Note, QExceptionPtr>;
00050
00051     using NotebookWithException =
00052         std::pair<qevercloud::Notebook, QExceptionPtr>;
00053
00054     using SavedSearchWithException =
00055         std::pair<qevercloud::SavedSearch, QExceptionPtr>;
00056
00057     using TagWithException = std::pair<qevercloud::Tag, QExceptionPtr>;
00058
00059 public:
00060     // Total
00061
00062     [[nodiscard]] virtual quint64 totalAttemptedToSendNotes() const = 0;
00066
00067     [[nodiscard]] virtual quint64 totalAttemptedToSendNotebooks() const = 0;
00071
00072     [[nodiscard]] virtual quint64 totalAttemptedToSendSavedSearches() const = 0;
00076
00077     [[nodiscard]] virtual quint64 totalAttemptedToSendTags() const = 0;
00081
00082     // Notes
00083
00084     [[nodiscard]] virtual quint64 totalSuccessfullySentNotes() const = 0;
00088
00089     [[nodiscard]] virtual QList<NoteWithException> failedToSendNotes()
00090         const = 0;
00094
00095     // Notebooks
00096
00097     [[nodiscard]] virtual quint64 totalSuccessfullySentNotebooks() const = 0;
00101
00102     [[nodiscard]] virtual QList<NotebookWithException> failedToSendNotebooks()
00103         const = 0;
00107
00108     // Saved searches
00109
00110     [[nodiscard]] virtual quint64 totalSuccessfullySentSavedSearches()
00111         const = 0;
00115
00116     [[nodiscard]] virtual QList<SavedSearchWithException>
00117         failedToSendSavedSearches() const = 0;
00121
00122     // Tags
00123
00124     [[nodiscard]] virtual quint64 totalSuccessfullySentTags() const = 0;
00128
00129     [[nodiscard]] virtual QList<TagWithException> failedToSendTags() const = 0;
00133
00134     // General
00135
00136     [[nodiscard]] virtual StopSynchronizationError stopSynchronizationError()
00137         const = 0;
00141
00142     [[nodiscard]] virtual bool needToRepeatIncrementalSync() const = 0;
00146
00147 };
00151
00152 } // namespace quantier::synchronization

```

6.47 ISyncChunksDataCounters.h

```

00001 /*
00002  * Copyright 2021-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/utility/Linkage.h>

```

```

00022 #include <quentier/utility/Printable.h>
00023
00024 #include <QtGlobal>
00025
00026 namespace quentier::synchronization {
00027
00033 struct QUENTIER_EXPORT ISyncChunksDataCounters : public utility::Printable
00034 {
00035     // ===== Saved searches =====
00036
00040     [[nodiscard]] virtual quint64 totalSavedSearches() const noexcept = 0;
00041
00045     [[nodiscard]] virtual quint64 totalExpungedSavedSearches()
00046         const noexcept = 0;
00047
00052     [[nodiscard]] virtual quint64 addedSavedSearches() const noexcept = 0;
00053
00058     [[nodiscard]] virtual quint64 updatedSavedSearches() const noexcept = 0;
00059
00064     [[nodiscard]] virtual quint64 expungedSavedSearches() const noexcept = 0;
00065
00066     // ===== Tags =====
00067
00071     [[nodiscard]] virtual quint64 totalTags() const noexcept = 0;
00072
00076     [[nodiscard]] virtual quint64 totalExpungedTags() const noexcept = 0;
00077
00081     [[nodiscard]] virtual quint64 addedTags() const noexcept = 0;
00082
00086     [[nodiscard]] virtual quint64 updatedTags() const noexcept = 0;
00087
00091     [[nodiscard]] virtual quint64 expungedTags() const noexcept = 0;
00092
00093     // ===== Linked notebooks =====
00094
00098     [[nodiscard]] virtual quint64 totalLinkedNotebooks() const noexcept = 0;
00099
00103     [[nodiscard]] virtual quint64 totalExpungedLinkedNotebooks()
00104         const noexcept = 0;
00105
00110     [[nodiscard]] virtual quint64 addedLinkedNotebooks() const noexcept = 0;
00111
00116     [[nodiscard]] virtual quint64 updatedLinkedNotebooks() const noexcept = 0;
00117
00122     [[nodiscard]] virtual quint64 expungedLinkedNotebooks() const noexcept = 0;
00123
00124     // ===== Notebooks =====
00125
00129     [[nodiscard]] virtual quint64 totalNotebooks() const noexcept = 0;
00130
00134     [[nodiscard]] virtual quint64 totalExpungedNotebooks() const noexcept = 0;
00135
00139     [[nodiscard]] virtual quint64 addedNotebooks() const noexcept = 0;
00140
00144     [[nodiscard]] virtual quint64 updatedNotebooks() const noexcept = 0;
00145
00150     [[nodiscard]] virtual quint64 expungedNotebooks() const noexcept = 0;
00151 };
00152
00153 } // namespace quentier::synchronization

```

6.48 ISyncOptions.h

```

00001 /*
00002  * Copyright 2022-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020

```

```

00021 #include <quentier/utility/Linkage.h>
00022 #include <quentier/utility/Printable.h>
00023
00024 #include <qevercloud/Fwd.h>
00025
00026 #include <QDir>
00027 #include <QtGlobal>
00028
00029 #include <optional>
00030
00031 namespace quentier::synchronization {
00032
00033     class QUINTIER_EXPORT ISyncOptions : public utility::Printable
00034     {
00035     public:
00036         ~ISyncOptions() noexcept override;
00037
00038         [[nodiscard]] virtual bool downloadNoteThumbnails() const = 0;
00039
00040         [[nodiscard]] virtual std::optional<QDir> inkNoteImagesStorageDir()
00041             const = 0;
00042
00043         [[nodiscard]] virtual qevercloud::IRequestContextPtr requestContext()
00044             const = 0;
00045
00046         [[nodiscard]] virtual qevercloud::IRetryPolicyPtr retryPolicy() const = 0;
00047
00048         [[nodiscard]] virtual std::optional<quint32> maxConcurrentNoteDownloads()
00049             const = 0;
00050
00051         [[nodiscard]] virtual std::optional<quint32>
00052             maxConcurrentResourceDownloads() const = 0;
00053     };
00054 } // namespace quentier::synchronization

```

6.49 ISyncOptionsBuilder.h

```

00001 /*
00002  * Copyright 2022-2023 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <qevercloud/Fwd.h>
00025
00026 #include <QDir>
00027
00028 #include <optional>
00029
00030 namespace quentier::synchronization {
00031
00032     class QUINTIER_EXPORT ISyncOptionsBuilder
00033     {
00034     public:
00035         virtual ~ISyncOptionsBuilder() noexcept;
00036
00037         virtual ISyncOptionsBuilder & setDownloadNoteThumbnails(bool value) = 0;
00038
00039         virtual ISyncOptionsBuilder & setInkNoteImagesStorageDir(
00040             std::optional<QDir> dir) = 0;
00041
00042         virtual ISyncOptionsBuilder & setRequestContext(
00043             qevercloud::IRequestContextPtr ctx) = 0;
00044
00045         virtual ISyncOptionsBuilder & setRetryPolicy(

```

```

00046         qevercloud::IRetryPolicyPtr retryPolicy) = 0;
00047
00048     virtual ISyncOptionsBuilder & setMaxConcurrentNoteDownloads(
00049         std::optional<quint32> maxConcurrentNoteDownloads) = 0;
00050
00051     virtual ISyncOptionsBuilder & setMaxConcurrentResourceDownloads(
00052         std::optional<quint32> maxConcurrentResourceDownloads) = 0;
00053
00054     [[nodiscard]] virtual ISyncOptionsPtr build() = 0;
00055 };
00056
00057 [[nodiscard]] QUENTIER_EXPORT ISyncOptionsBuilderPtr createSyncOptionsBuilder();
00058
00059 } // namespace quantier::synchronization

```

6.50 ISyncResult.h

```

00001 /*
00002  * Copyright 2022-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/synchronization/Fwd.h>
00022 #include <quantier/synchronization/types/Errors.h>
00023 #include <quantier/synchronization/types/Fwd.h>
00024 #include <quantier/utility/Linkage.h>
00025 #include <quantier/utility/Printable.h>
00026
00027 #include <qevercloud/types/TypeAliases.h>
00028
00029 #include <QHash>
00030 #include <QSet>
00031
00032 namespace quantier::synchronization {
00033
00034     class QUENTIER_EXPORT ISyncResult : public utility::Printable
00035     {
00036     public:
00037         [[nodiscard]] virtual ISyncStatePtr syncState() const = 0;
00038
00039         [[nodiscard]] virtual ISyncChunksDataCountersPtr
00040             userAccountSyncChunksDataCounters() const = 0;
00041
00042         [[nodiscard]] virtual QHash<qevercloud::Guid, ISyncChunksDataCountersPtr>
00043             linkedNotebookSyncChunksDataCounters() const = 0;
00044
00045         [[nodiscard]] virtual bool userAccountSyncChunksDownloaded() const = 0;
00046
00047         [[nodiscard]] virtual QSet<qevercloud::Guid>
00048             linkedNotebookGuidsWithSyncChunksDownloaded() const = 0;
00049
00050         [[nodiscard]] virtual IDownloadNotesStatusPtr
00051             userAccountDownloadNotesStatus() const = 0;
00052
00053         [[nodiscard]] virtual QHash<qevercloud::Guid, IDownloadNotesStatusPtr>
00054             linkedNotebookDownloadNotesStatuses() const = 0;
00055
00056         [[nodiscard]] virtual IDownloadResourcesStatusPtr
00057             userAccountDownloadResourcesStatus() const = 0;
00058
00059         [[nodiscard]] virtual QHash<qevercloud::Guid, IDownloadResourcesStatusPtr>
00060             linkedNotebookDownloadResourcesStatuses() const = 0;
00061
00062         [[nodiscard]] virtual ISendStatusPtr userAccountSendStatus() const = 0;
00063
00064         [[nodiscard]] virtual QHash<qevercloud::Guid, ISendStatusPtr>
00065             linkedNotebookSendStatuses() const = 0;
00066

```

```

00067     [[nodiscard]] virtual StopSynchronizationError stopSynchronizationError()
00068         const = 0;
00069 };
00070
00071 } // namespace quantier::synchronization

```

6.51 ISyncState.h

```

00001 /*
00002  * Copyright 2022-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/utility/Linkage.h>
00022 #include <quantier/utility/Printable.h>
00023
00024 #include <qevercloud/types/TypeAliases.h>
00025
00026 #include <QHash>
00027 #include <QString>
00028
00029 namespace quantier::synchronization {
00030
00031     class QUANTIER_EXPORT ISyncState : public utility::Printable
00032     {
00033     public:
00034         [[nodiscard]] virtual qint32 userDataUpdateCount() const = 0;
00035
00036         [[nodiscard]] virtual qevercloud::Timestamp userDataLastSyncTime()
00037             const = 0;
00038
00039         [[nodiscard]] virtual QHash<qevercloud::Guid, qint32>
00040             linkedNotebookUpdateCounts() const = 0;
00041
00042         [[nodiscard]] virtual QHash<qevercloud::Guid, qevercloud::Timestamp>
00043             linkedNotebookLastSyncTimes() const = 0;
00044     };
00045
00046 } // namespace quantier::synchronization

```

6.52 ISyncStateBuilder.h

```

00001 /*
00002  * Copyright 2023 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/synchronization/types/Fwd.h>

```

```

00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <qevercloud/types/TypeAliases.h>
00025
00026 #include <QHash>
00027
00028 namespace quentier::synchronization {
00029
00030 class QUINTIER_EXPORT ISyncStateBuilder
00031 {
00032 public:
00033     virtual ~ISyncStateBuilder() noexcept;
00034
00035     virtual ISyncStateBuilder & setUserDataUpdateCount(qint32 updateCount) = 0;
00036
00037     virtual ISyncStateBuilder & setUserDataLastSyncTime(
00038         qevercloud::Timestamp lastSyncTime) = 0;
00039
00040     virtual ISyncStateBuilder & setLinkedNotebookUpdateCounts(
00041         QHash<qevercloud::Guid, qint32> updateCounts) = 0;
00042
00043     virtual ISyncStateBuilder & setLinkedNotebookLastSyncTimes(
00044         QHash<qevercloud::Guid, qevercloud::Timestamp> lastSyncTimes) = 0;
00045
00046     [[nodiscard]] virtual ISyncStatePtr build() = 0;
00047 };
00048
00049 [[nodiscard]] QUINTIER_EXPORT ISyncStateBuilderPtr createSyncStateBuilder();
00050
00051 } // namespace quentier::synchronization

```

6.53 AuthenticationInfo.h

```

00001 /*
00002  * Copyright 2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QJsonObject>
00025
00026 namespace quentier::synchronization {
00027
00031 [[nodiscard]] QJsonObject QUINTIER_EXPORT
00032     serializeAuthenticationInfoToJson(const IAuthenticationInfo & info);
00033
00039 [[nodiscard]] IAuthenticationInfoPtr QUINTIER_EXPORT
00040     deserializeAuthenticationInfoFromJson(const QJsonObject & json);
00041
00042 } // namespace quentier::synchronization

```

6.54 DownloadNotesStatus.h

```

00001 /*
00002  * Copyright 2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.

```

```

00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QJsonObject>
00025
00026 namespace quentier::synchronization {
00027
00031 [[nodiscard]] QJsonObject QUENTIER_EXPORT
00032     serializeDownloadNotesStatusToJson(const IDownloadNotesStatus & status);
00033
00039 [[nodiscard]] IDownloadNotesStatusPtr QUENTIER_EXPORT
00040     deserializeDownloadNotesStatusFromJson(const QJsonObject & json);
00041
00042 } // namespace quentier::synchronization

```

6.55 DownloadResourcesStatus.h

```

00001 /*
00002  * Copyright 2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QJsonObject>
00025
00026 namespace quentier::synchronization {
00027
00031 [[nodiscard]] QJsonObject QUENTIER_EXPORT
00032     serializeDownloadResourcesStatusToJson(
00033         const IDownloadResourcesStatus & status);
00034
00040 [[nodiscard]] IDownloadResourcesStatusPtr QUENTIER_EXPORT
00041     deserializeDownloadResourcesStatusFromJson(const QJsonObject & json);
00042
00043 } // namespace quentier::synchronization

```

6.56 SendStatus.h

```

00001 /*
00002  * Copyright 2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of

```



```

00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QJsonObject>
00025
00026 namespace quentier::synchronization {
00027
00031 [[nodiscard]] QJsonObject QUENTIER_EXPORT
00032     serializeSendStatusToJson(const ISendStatus & sendStatus);
00033
00039 [[nodiscard]] ISendStatusPtr QUENTIER_EXPORT
00040     deserializeSendStatusFromJson(const QJsonObject & json);
00041
00042 } // namespace quentier::synchronization

```

6.57 SyncChunksDataCounters.h

```

00001 /*
00002  * Copyright 2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QJsonObject>
00025
00026 namespace quentier::synchronization {
00027
00031 [[nodiscard]] QJsonObject QUENTIER_EXPORT serializeSyncChunksDataCountersToJson(
00032     const ISyncChunksDataCounters & counters);
00033
00039 [[nodiscard]] ISyncChunksDataCountersPtr QUENTIER_EXPORT
00040     deserializeSyncChunksDataCountersFromJson(const QJsonObject & json);
00041
00042 } // namespace quentier::synchronization

```

6.58 SyncResult.h

```

00001 /*
00002  * Copyright 2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License

```

```

00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QJsonObject>
00025
00026 namespace quentier::synchronization {
00027
00031 [[nodiscard]] QJsonObject QUENTIER_EXPORT
00032     serializeSyncResultToJson(const ISyncResult & syncResult);
00033
00039 [[nodiscard]] ISyncResultPtr QUENTIER_EXPORT
00040     deserializeSyncResultFromJson(const QJsonObject & json);
00041
00042 } // namespace quentier::synchronization

```

6.59 SyncState.h

```

00001 /*
00002  * Copyright 2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/synchronization/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QJsonObject>
00025
00026 namespace quentier::synchronization {
00027
00031 [[nodiscard]] QJsonObject QUENTIER_EXPORT
00032     serializeSyncStateToJson(const ISyncState & syncState);
00033
00039 [[nodiscard]] ISyncStatePtr QUENTIER_EXPORT
00040     deserializeSyncStateFromJson(const QJsonObject & json);
00041
00042 } // namespace quentier::synchronization

```

6.60 Future.h

```

00001 /*
00002  * Copyright 2021-2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once

```

```

00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QAbstractEventDispatcher>
00024 #include <QFuture>
00025 #include <QFutureWatcher>
00026 #include <QMutex>
00027 #include <QMutexLocker>
00028 #include <QObject>
00029 #include <QPointer>
00030
00031 #include <quentier/threading/QtFutureContinuations.h>
00032
00033 #if QT_VERSION >= QT_VERSION_CHECK(6, 0, 0)
00034 #include <QPromise>
00035 #include <exception>
00036 #else
00037 #include <quentier/threading/Qt5Promise.h>
00038 #endif
00039
00040 #include <algorithm>
00041 #include <cmath>
00042 #include <memory>
00043 #include <type_traits>
00044 #include <utility>
00045
00046 namespace quentier::threading {
00047
00051 template <class T>
00052 [[nodiscard]] std::enable_if_t<
00053     std::negation_v<std::is_same<std::decay_t<T>, void>>,
00054     QFuture<std::decay_t<T>>
00055     makeReadyFuture(T t)
00056 {
00057     QPromise<std::decay_t<T>> promise;
00058     QFuture<std::decay_t<T>> future = promise.future();
00059
00060     promise.start();
00061     promise.addResult(std::move(t));
00062     promise.finish();
00063
00064     return future;
00065 }
00066
00067 [[nodiscard]] QFuture<void> QUENTIER_EXPORT makeReadyFuture();
00068
00073 template <class T, class E>
00074 [[nodiscard]] std::enable_if_t<std::is_base_of_v<QException, E>, QFuture<T>>
00075     makeExceptionalFuture(const E & e)
00076 {
00077     QPromise<std::decay_t<T>> promise;
00078     QFuture<std::decay_t<T>> future = promise.future();
00079
00080     promise.start();
00081     promise.setException(e);
00082     promise.finish();
00083
00084     return future;
00085 }
00086
00087 #if QT_VERSION >= QT_VERSION_CHECK(6, 0, 0)
00092 template <class T>
00093 [[nodiscard]] QFuture<T> makeExceptionalFuture(std::exception_ptr e)
00094 {
00095     QPromise<std::decay_t<T>> promise;
00096     QFuture<std::decay_t<T>> future = promise.future();
00097
00098     promise.start();
00099     promise.setException(std::move(e));
00100     promise.finish();
00101
00102     return future;
00103 }
00104 #endif // QT_VERSION
00105
00110 template <class T, class U>
00111 void bindCancellation(const QFuture<T> & from, QFuture<U> to)
00112 {
00113     auto watcher = std::make_unique<QFutureWatcher<T>>();
00114     auto * rawWatcher = watcher.get();
00115
00116     QObject::connect(
00117         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
00118         [rawWatcher, to]() mutable {
00119             to.cancel();
00120             rawWatcher->deleteLater();
00121         });

```

```

00122
00123     QObject::connect(
00124         rawWatcher, &QFutureWatcher<T>::finished, rawWatcher,
00125         [rawWatcher] { rawWatcher->deleteLater(); });
00126
00127     watcher->setFuture(from);
00128     Q_UNUSED(watcher.release());
00129 }
00130
00131 [[nodiscard]] QFuture<void> QUINTIER_EXPORT
00132     whenAll(QList<QFuture<void>> futures);
00133
00134 template <class T>
00135 [[nodiscard]] std::enable_if_t<
00136     !std::is_void_v<std::decay_t<T>, QFuture<QList<std::decay_t<T>>>
00137     whenAll(QList<QFuture<std::decay_t<T>>> futures)
00138 {
00139     if (Q_UNLIKELY(futures.isEmpty())) {
00140         return makeReadyFuture<QList<std::decay_t<T>>>({});
00141     }
00142
00143     auto promise = std::make_shared<QPromise<QList<std::decay_t<T>>>>();
00144     auto future = promise->future();
00145
00146     for (auto & f: futures) {
00147         threading::bindCancellation(future, f);
00148     }
00149
00150     const auto totalItemCount = futures.size();
00151     promise->setProgressRange(0, static_cast<int>(totalItemCount));
00152     promise->setProgressValue(0);
00153
00154     promise->start();
00155
00156     auto resultIndexedList =
00157         std::make_shared<QList<std::pair<int, std::decay_t<T>>>>();
00158
00159     auto processedItemsCount = std::make_shared<int>(0);
00160     auto exceptionFlag = std::make_shared<bool>(false);
00161     auto mutex = std::make_shared<QMutex>();
00162
00163     for (int i = 0; i < futures.size(); ++i) {
00164         auto & f = futures[i];
00165         auto thenFuture = then(
00166             std::move(f),
00167             [promise, processedItemsCount, totalItemCount, exceptionFlag, mutex,
00168             resultIndexedList, i](std::decay_t<T> result) {
00169                 if (promise->isCanceled()) {
00170                     return;
00171                 }
00172
00173                 int count = 0;
00174                 {
00175                     const QMutexLocker locker{mutex.get()};
00176
00177                     if (*exceptionFlag) {
00178                         return;
00179                     }
00180
00181                     ++(*processedItemsCount);
00182                     count = *processedItemsCount;
00183                     promise->setProgressValue(count);
00184
00185                     resultIndexedList->append(
00186                         std::make_pair(i, std::move(result)));
00187                 }
00188
00189                 if (count == totalItemCount) {
00190                     std::sort(
00191                         resultIndexedList->begin(), resultIndexedList->end(),
00192                         [](const auto & lhs, const auto & rhs) {
00193                             return lhs.first < rhs.first;
00194                         });
00195
00196                     auto resultList =
00197                         std::make_shared<QList<std::decay_t<T>>>();
00198                     resultList->reserve(resultIndexedList->size());
00199                     for (auto & [i, v]: *resultIndexedList) {
00200                         resultList->append(std::move(v));
00201                     }
00202
00203                     promise->addResult(*resultList);
00204                     promise->finish();
00205                 }
00206             });
00207     }
00208
00209     onFailed(

```

```

00225         std::move(thenFuture),
00226         [promise, mutex, exceptionFlag](const QException & e) {
00227             if (promise->isCanceled()) {
00228                 return;
00229             }
00230
00231             {
00232                 const QMutexLocker locker(mutex.get());
00233
00234                 if (*exceptionFlag) {
00235                     return;
00236                 }
00237
00238                 *exceptionFlag = true;
00239             }
00240
00241             promise->setException(e);
00242             promise->finish();
00243         });
00244     }
00245
00246     return future;
00247 }
00248
00254 template <class T, class U>
00255 void mapFutureProgress(
00256     const QFuture<T> & future, const std::shared_ptr<QPromise<U> & promise)
00257 {
00258     const auto futureProgressMinimum = future.progressMinimum();
00259     const auto futureProgressRange =
00260         future.progressMaximum() - futureProgressMinimum;
00261
00262     Q_ASSERT(futureProgressRange >= 0);
00263
00264     const auto promiseFuture = promise->future();
00265     const auto promiseProgressMinimum = promiseFuture.progressMinimum();
00266     const auto promiseProgressMaximum = promiseFuture.progressMaximum();
00267
00268     const auto promiseProgressRange =
00269         promiseProgressMaximum - promiseProgressMinimum;
00270
00271     Q_ASSERT(promiseProgressRange >= 0);
00272
00273     auto futureWatcher = std::make_unique<QFutureWatcher<T>>();
00274
00275     QObject::connect(
00276         futureWatcher.get(), &QFutureWatcher<T>::progressValueChanged,
00277         futureWatcher.get(),
00278         [promise, futureProgressMinimum, futureProgressRange,
00279          promiseProgressRange, promiseProgressMinimum,
00280          promiseProgressMaximum](int progressValue) {
00281             if (Q_UNLIKELY(futureProgressRange == 0)) {
00282                 promise->setProgressValue(0);
00283                 return;
00284             }
00285
00286             const auto progressPart =
00287                 static_cast<double>(progressValue - futureProgressMinimum) /
00288                 static_cast<double>(futureProgressRange);
00289
00290             const auto mappedProgressValue = static_cast<int>(
00291                 std::round(progressPart * promiseProgressRange));
00292
00293             promise->setProgressValue(std::clamp(
00294                 promiseProgressMinimum + mappedProgressValue,
00295                 promiseProgressMinimum, promiseProgressMaximum));
00296         });
00297
00298     QObject::connect(
00299         futureWatcher.get(), &QFutureWatcher<T>::finished, futureWatcher.get(),
00300         [futureWatcherWeak = QPointer<QFutureWatcher<T>>(futureWatcher.get())] {
00301             if (!futureWatcherWeak.isNull()) {
00302                 futureWatcherWeak->deleteLater();
00303             }
00304         });
00305
00306     QObject::connect(
00307         futureWatcher.get(), &QFutureWatcher<T>::canceled, futureWatcher.get(),
00308         [futureWatcherWeak = QPointer<QFutureWatcher<T>>(futureWatcher.get())] {
00309             if (!futureWatcherWeak.isNull()) {
00310                 futureWatcherWeak->deleteLater();
00311             }
00312         });
00313
00314     futureWatcher->setFuture(future);
00315     Q_UNUSED(futureWatcher.release());
00316 }

```

```

00317
00318 } // namespace quentier::threading

```

6.61 Post.h

```

00001 /*
00002  * Copyright 2021-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <QAbstractEventDispatcher>
00022 #include <QMetaObject>
00023 #include <QObject>
00024
00025 #include <QThread>
00026
00027 #include <memory>
00028 #include <utility>
00029
00030 namespace quentier::threading {
00031
00032 template <typename Function>
00033 void postToObject(QObject * object, Function && function)
00034 {
00035     Q_ASSERT(object);
00036
00037     QMetaObject::invokeMethod(
00038         object, std::forward<Function>(function), Qt::QueuedConnection);
00039 }
00040
00041 template <typename Function>
00042 void postToThread(QThread * pThread, Function && function)
00043 {
00044     Q_ASSERT(pThread);
00045     Q_ASSERT(!pThread->isFinished());
00046
00047     QObject * pObject = QAbstractEventDispatcher::instance(pThread);
00048     if (!pObject) {
00049         // Thread's event loop has not been started yet. Create a dummy QObject,
00050         // move it to the target thread, set things up so that it would be
00051         // destroyed after the job is done and use postToObject.
00052         auto pDummyObj = std::make_unique<QObject>();
00053         pDummyObj->moveToThread(pThread);
00054         postToObject(
00055             pDummyObj.get(),
00056             [pObj = pDummyObj.get(),
00057              function = std::forward<Function>(function)]() mutable {
00058                 pObj->deleteLater();
00059                 function();
00060             });
00061         Q_UNUSED(pDummyObj.release()) // NOLINT
00062         return;
00063     }
00064
00065     if (pThread == QThread::currentThread()) {
00066         // Already on the target thread, executing the function right away
00067         function();
00068         return;
00069     }
00070
00071     QMetaObject::invokeMethod(pObject, std::forward<Function>(function));
00072 }
00073
00074 } // namespace quentier::threading

```

6.62 Qt5Promise.h

```

00001 /*
00002  * Copyright 2021-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <QFutureInterface>
00022
00023 #include <type_traits>
00024
00025 // Partial backport of QPromise from Qt6 to Qt5
00026 template <typename T>
00027 class QPromise
00028 {
00029     static_assert(
00030         std::is_copy_constructible_v<T> || std::is_move_constructible_v<T> ||
00031         std::is_same_v<T, void>,
00032         "Type with copy or move constructors or type void is required");
00033
00034 public:
00035     QPromise() = default;
00036
00037     Q_DISABLE_COPY(QPromise)
00038
00039     QPromise(QPromise<T> && other) noexcept : d(other.d)
00040     {
00041         other.d = QFutureInterface<T>();
00042     }
00043
00044     QPromise(QFutureInterface<T> & other) : d(other) {}
00045
00046     QPromise & operator=(QPromise<T> && other) noexcept
00047     {
00048         QPromise<T> tmp(std::move(other));
00049         tmp.swap(*this);
00050         return *this;
00051     }
00052
00053     ~QPromise()
00054     {
00055         // If QFutureInterface has no state, there is nothing to be done
00056         if (d.queryState(QFutureInterfaceBase::State::NoState)) {
00057             return;
00058         }
00059
00060         // Otherwise, if computation is not finished at this point, cancel
00061         // potential waits
00062         if (!d.queryState(QFutureInterfaceBase::State::Finished)) {
00063             d.cancel();
00064             finish(); // required to finalize the state
00065         }
00066     }
00067
00068     // Core QPromise APIs
00069     QFuture<T> future() const
00070     {
00071         return d.future();
00072     }
00073
00074     template <
00075         typename U,
00076         typename = std::enable_if_t<
00077             std::is_same_v<U, T> || std::is_convertible_v<U, T>>
00078     >
00079     void addResult(U && result, int index = -1)
00080     {
00081         d.reportResult(std::forward<U>(result), index);
00082     }
00083
00084     void setException(const QException & e)
00085     {
00086         d.reportException(e);
00087     }

```

```

00085     }
00086
00087     void start()
00088     {
00089         d.reportStarted();
00090     }
00091     void finish()
00092     {
00093         d.reportFinished();
00094     }
00095
00096     void suspendIfRequested()
00097     {
00098         d.suspendIfRequested();
00099     }
00100
00101     bool isCanceled() const
00102     {
00103         return d.isCanceled();
00104     }
00105
00106     // Progress methods
00107     void setProgressRange(int minimum, int maximum)
00108     {
00109         d.setProgressRange(minimum, maximum);
00110     }
00111     void setProgressValue(int progressValue)
00112     {
00113         d.setProgressValue(progressValue);
00114     }
00115     void setProgressValueAndText (
00116         int progressValue, const QString & progressText)
00117     {
00118         d.setProgressValueAndText (progressValue, progressText);
00119     }
00120
00121     void swap(QPromise<T> & other) noexcept
00122     {
00123         qSwap(this->d, other.d);
00124     }
00125
00126 private:
00127     mutable QFutureInterface<T> d = QFutureInterface<T>();
00128 };
00129
00130 template <typename T>
00131 inline void swap(QPromise<T> & a, QPromise<T> & b) noexcept
00132 {
00133     a.swap(b);
00134 }

```

6.63 QtFutureContinuations.h

```

00001 /*
00002  * Copyright 2021-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <QtGlobal>
00022
00023 #include <QFutureWatcher>
00024 #include <QRunnable>
00025 #include <QThreadPool>
00026 #include <quentier/exception/RuntimeError.h>
00027 #include <quentier/threading/Post.h>
00028 #include <quentier/threading/QtFutureHelpers.h>
00029
00030 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)

```



```

00031 #include <quentier/threading/Qt5Promise.h>
00032 #endif // QT_VERSION
00033
00034 #include <quentier/threading/Runnable.h>
00035
00036 #include <boost/core/demangle.hpp>
00037
00038 #include <memory>
00039 #include <type_traits>
00040 #include <typeinfo>
00041
00042 namespace quentier::threading {
00043
00044 // NOTE: "native" implementation of continuations for Qt6 is currently disabled
00045 // due to bugs in their implementation, in particular (but not limited to)
00046 // https://bugreports.qt.io/browse/QTBUG-119579 and
00047 // https://bugreports.qt.io/browse/QTBUG-117918. It's a shame but it is what it
00048 // is.
00049 /*
00050 #if QT_VERSION >= QT_VERSION_CHECK(6, 0, 0)
00051
00052 template <class T, class Function>
00053 QFuture<typename QtPrivate::ResultTypeHelper<Function, T>::ResultType> then(
00054     QFuture<T> && future, Function && function)
00055 {
00056     return future.then(std::forward<decltype(function)>(function));
00057 }
00058
00059 template <class T, class Function>
00060 QFuture<typename QtPrivate::ResultTypeHelper<Function, T>::ResultType> then(
00061     QFuture<T> && future, QtFuture::Launch policy, Function && function)
00062 {
00063     return future.then(policy, std::forward<decltype(function)>(function));
00064 }
00065
00066 template <class T, class Function>
00067 QFuture<typename QtPrivate::ResultTypeHelper<Function, T>::ResultType> then(
00068     QFuture<T> && future, QThreadPool * pool, Function && function)
00069 {
00070     return future.then(pool, std::forward<decltype(function)>(function));
00071 }
00072
00073 template <class T, class Function>
00074 QFuture<typename QtPrivate::ResultTypeHelper<Function, T>::ResultType> then(
00075     QFuture<T> && future, QObject * context, Function && function)
00076 {
00077     return future.then(context, std::forward<decltype(function)>(function));
00078 }
00079
00080 template <class T, class Function>
00081 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, QFuture<T>
00082     onFailed(QFuture<T> && future, Function && handler)
00083 {
00084     return future.onFailed(std::forward<decltype(handler)>(handler));
00085 }
00086
00087 template <class T, class Function>
00088 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, QFuture<T>
00089     onFailed(QFuture<T> && future, QObject * context, Function && handler)
00090 {
00091     return future.onFailed(context, std::forward<decltype(handler)>(handler));
00092 }
00093
00094 #else // QT_VERSION
00095
00096 // implementation for Qt5
00097 */
00098
00099 namespace detail {
00100
00101 template <class T, class Function>
00102 void processParentFuture(
00103     std::shared_ptr<
00104         QPromise<typename ResultTypeHelper<Function, T>::ResultType>
00105     > promise,
00106     QFuture<T> && future, Function && function)
00107 {
00108     Q_ASSERT(promise);
00109
00110     using ResultType = typename ResultTypeHelper<Function, T>::ResultType;
00111
00112     promise->start();
00113
00114     // If future contains exception, just forward it to the promise and
00115     // don't call the function at all
00116     try {
00117         future.waitForFinished();

```

```

00118     }
00119     catch (const QException & e) {
00120         promise->setException(e);
00121         promise->finish();
00122         return;
00123     }
00124     // NOTE: there cannot be other exception types in this context in Qt5
00125     // because exception store can only contain QExceptions
00126
00127     // Try to run the handler, in case of success forward the result to promise
00128     // (unless it is void), catch possible exceptions and if caught put them
00129     // to the promise
00130     try {
00131         if constexpr (std::is_void_v<ResultType>) {
00132             if constexpr (std::is_void_v<T>) {
00133                 function();
00134             }
00135             else {
00136                 if (future.resultCount() == 0) {
00137                     promise->setException(RuntimeError{ErrorString{
00138                         QString::fromUtf8(
00139                             "Invalid future continuation: detected future "
00140                             "without result for type %1"
00141                             .arg(QString::fromStdString(std::string{
00142                                 boost::core::demangle(typeid(T).name())
00143                             })));
00144                     }
00145                     promise->finish();
00146                     return;
00147                 }
00148                 function(future.result());
00149             }
00150             else {
00151                 if constexpr (std::is_void_v<T>) {
00152                     promise->addResult(function());
00153                 }
00154                 else {
00155                     promise->addResult(function(future.result()));
00156                 }
00157             }
00158         }
00159         catch (const QException & e) {
00160             promise->setException(e);
00161         }
00162         catch (const std::exception & e) {
00163             ErrorString error{QT_TRANSLATE_NOOP(
00164                 "utility", "Unknown std::exception in then future handler")};
00165             error.details() = QString::fromStdString(std::string{e.what()});
00166             promise->setException(RuntimeError{std::move(error)});
00167         }
00168         catch (...) {
00169             ErrorString error{QT_TRANSLATE_NOOP(
00170                 "utility", "Unknown exception in then future handler")};
00171             promise->setException(RuntimeError{std::move(error)});
00172         }
00173     }
00174     promise->finish();
00175 }
00176
00177 } // namespace detail
00178
00179 template <class T, class Function>
00180 QFuture<typename detail::ResultTypeHelper<Function, T>::ResultType> then(
00181     QFuture<T> && future, Function && function)
00182 {
00183     using ResultType =
00184         typename detail::ResultTypeHelper<Function, T>::ResultType;
00185
00186     auto promise = std::make_shared<QPromise<ResultType>>();
00187     auto result = promise->future();
00188
00189     if (future.isFinished()) {
00190         detail::processParentFuture(
00191             std::move(promise), std::move(future),
00192             std::forward<decltype(function)>(function));
00193         return result;
00194     }
00195
00196     auto watcher = std::make_unique<QFutureWatcher<T>>();
00197     auto * rawWatcher = watcher.get();
00198     QObject::connect(
00199         rawWatcher, &QFutureWatcher<T>::finished, rawWatcher,
00200         [rawWatcher, function = std::forward<decltype(function)>(function),
00201             promise = std::move(promise)]() mutable {
00202             detail::processParentFuture(
00203                 std::move(promise), rawWatcher->future(),
00204                 std::forward<decltype(function)>(function));

```

```

00205         rawWatcher->deleteLater();
00206     });
00207
00208     QObject::connect(
00209         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
00210         [rawWatcher] { rawWatcher->deleteLater(); });
00211
00212     watcher->setFuture(std::move(future));
00213     Q_UNUSED(watcher.release())
00214
00215     return result;
00216 }
00217
00218 template <class T, class Function>
00219 QFuture<typename detail::ResultTypeHelper<Function, T>::ResultType> then(
00220     QFuture<T> && future, QtFuture::Launch policy, Function && function)
00221 {
00222     if (policy == QtFuture::Launch::Sync) {
00223         return then(
00224             std::move(future), std::forward<decltype(function)>(function));
00225     }
00226
00227     return then(
00228         std::move(future), QThreadPool::globalInstance(),
00229         std::forward<decltype(function)>(function));
00230 }
00231
00232 template <class T, class Function>
00233 QFuture<typename detail::ResultTypeHelper<Function, T>::ResultType> then(
00234     QFuture<T> && future, QThreadPool * pool, Function && function)
00235 {
00236     using ResultType =
00237         typename detail::ResultTypeHelper<Function, T>::ResultType;
00238
00239     auto promise = std::make_shared<QPromise<ResultType>>();
00240     auto result = promise->future();
00241
00242     if (future.isFinished()) {
00243         auto * runnable = createFunctionRunnable(
00244             [future = std::move(future), promise = std::move(promise),
00245              function = std::forward<decltype(function)>(function)]() mutable {
00246                 detail::processParentFuture(
00247                     std::move(promise), std::move(future),
00248                     std::forward<decltype(function)>(function));
00249             });
00250         runnable->setAutoDelete(true);
00251         pool->start(runnable);
00252         return result;
00253     }
00254
00255     auto watcher = std::make_unique<QFutureWatcher<T>>();
00256     auto * rawWatcher = watcher.get();
00257     QObject::connect(
00258         rawWatcher, &QFutureWatcher<T>::finished, rawWatcher,
00259         [rawWatcher, function = std::forward<decltype(function)>(function),
00260          promise = std::move(promise), pool]() mutable {
00261             auto * runnable = createFunctionRunnable(
00262                 [function = std::forward<decltype(function)>(function),
00263                  promise = std::move(promise),
00264                  future = rawWatcher->future()]() mutable {
00265                     detail::processParentFuture(
00266                         std::move(promise), std::move(future),
00267                         std::forward<decltype(function)>(function));
00268                 });
00269             runnable->setAutoDelete(true);
00270             pool->start(runnable);
00271             rawWatcher->deleteLater();
00272         });
00273
00274     QObject::connect(
00275         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
00276         [rawWatcher] { rawWatcher->deleteLater(); });
00277
00278     watcher->setFuture(std::move(future));
00279     Q_UNUSED(watcher.release())
00280
00281     return result;
00282 }
00283
00284 template <class T, class Function>
00285 QFuture<typename detail::ResultTypeHelper<Function, T>::ResultType> then(
00286     QFuture<T> && future, QObject * context, Function && function)
00287 {
00288     using ResultType =
00289         typename detail::ResultTypeHelper<Function, T>::ResultType;
00290
00291     auto promise = std::make_shared<QPromise<ResultType>>();

```

```

00292     auto result = promise->future();
00293
00294     if (future.isFinished()) {
00295         postToObject(
00296             context,
00297             [future = std::move(future), promise = std::move(promise),
00298              function = std::forward<decltype(function)>(function)]() mutable {
00299                 detail::processParentFuture(
00300                     std::move(promise), std::move(future),
00301                     std::forward<decltype(function)>(function));
00302             });
00303         return result;
00304     }
00305
00306     auto watcher = std::make_unique<QFutureWatcher<T>>();
00307     auto * rawWatcher = watcher.get();
00308
00309     QObject::connect(
00310         rawWatcher, &QFutureWatcher<T>::finished, context,
00311         [context, rawWatcher,
00312          function = std::forward<decltype(function)>(function),
00313          promise = std::move(promise)]() mutable {
00314             postToObject(
00315                 context,
00316                 [function = std::forward<decltype(function)>(function),
00317                  promise = std::move(promise),
00318                  future = rawWatcher->future()]() mutable {
00319                     detail::processParentFuture(
00320                         std::move(promise), std::move(future),
00321                         std::forward<decltype(function)>(function));
00322                 });
00323             rawWatcher->deleteLater();
00324         });
00325
00326     QObject::connect(
00327         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
00328         [rawWatcher] { rawWatcher->deleteLater(); });
00329
00330     watcher->setFuture(std::move(future));
00331     Q_UNUSED(watcher.release())
00332
00333     return result;
00334 }
00335
00336 namespace detail {
00337
00338 template <class T, class Function>
00339 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, void>
00340 processPossibleFutureException(
00341     std::shared_ptr<QPromise<T> > promise, QFuture<T> && future,
00342     Function && handler)
00343 {
00344     Q_ASSERT(promise);
00345
00346     using ArgType = typename QtPrivate::ArgResolver<Function>::First;
00347     using ResultType =
00348         typename ResultTypeHelper<Function, std::decay_t<ArgType>>::ResultType;
00349     static_assert(std::is_convertible_v<ResultType, T>);
00350
00351     promise->start();
00352
00353     try {
00354         try {
00355             future.waitForFinished();
00356         }
00357         catch (const ArgType & e) {
00358             try {
00359                 if constexpr (std::is_void_v<ResultType>) {
00360                     handler(e);
00361                 }
00362                 else {
00363                     promise->addResult(handler(e));
00364                 }
00365             }
00366             catch (const QException & e) {
00367                 promise->setException(e);
00368             }
00369             catch (const std::exception & e) {
00370                 QString error{QT_TRANSLATE_NOOP(
00371                     "utility",
00372                     "Unknown std::exception in onFailed future handler")};
00373                 error.details() = QString::fromStdString(std::string{e.what()});
00374                 promise->setException(RuntimeError{std::move(error)});
00375             }
00376             catch (...) {
00377                 QString error{QT_TRANSLATE_NOOP(
00378                     "utility", "Unknown exception in onFailed future handler")};

```

```

00379         promise->setException(RuntimeError{std::move(error)});
00380     }
00381 }
00382 }
00383 // Exception doesn't match with handler's argument type, propagate
00384 // the exception to be handled later.
00385 catch (const QException & e) {
00386     promise->setException(e);
00387 }
00388 catch (const std::exception & e) {
00389     ErrorString error{QT_TRANSLATE_NOOP(
00390         "utility",
00391         "Unknown std::exception which did not match with onFailed "
00392         "future handler")};
00393     error.details() = QString::fromStdString(std::string{e.what()});
00394     promise->setException(RuntimeError{std::move(error)});
00395 }
00396 catch (...) {
00397     ErrorString error{QT_TRANSLATE_NOOP(
00398         "utility",
00399         "Unknown which did not match with onFailed "
00400         "future handler")};
00401     promise->setException(RuntimeError{std::move(error)});
00402 }
00403 }
00404 promise->finish();
00405 }
00406 }
00407 } // namespace detail
00408
00409 // WARNING! "Chaining" of onFailed calls would only work properly with Qt5 if
00410 // all involved exceptions subclass QException. It is due to the way exception
00411 // storage is implemented in Qt5. In Qt6 it was made to store std::exception_ptr
00412 // so there's no requirement to use QExceptions in Qt6.
00413
00414 template <class T, class Function>
00415 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, QFuture<T>
00416     onFailed(QFuture<T> && future, Function && handler)
00417 {
00418     auto promise = std::make_shared<QPromise<T>>();
00419     auto result = promise->future();
00420
00421     if (future.isFinished()) {
00422         detail::processPossibleFutureException(
00423             std::move(promise), std::move(future),
00424             std::forward<decltype(handler)>(handler));
00425         return result;
00426     }
00427
00428     auto watcher = std::make_unique<QFutureWatcher<T>>();
00429     auto * rawWatcher = watcher.get();
00430     QObject::connect(
00431         rawWatcher, &QFutureWatcher<T>::finished, rawWatcher,
00432         [rawWatcher, promise = std::move(promise),
00433          handler = std::forward<decltype(handler)>(handler)]() mutable {
00434             auto future = rawWatcher->future();
00435             rawWatcher->deleteLater();
00436             detail::processPossibleFutureException(
00437                 std::move(promise), std::move(future),
00438                 std::forward<decltype(handler)>(handler));
00439         });
00440
00441     QObject::connect(
00442         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
00443         [rawWatcher] { rawWatcher->deleteLater(); });
00444
00445     watcher->setFuture(std::move(future));
00446     Q_UNUSED(watcher.release())
00447
00448     return result;
00449 }
00450
00451 template <class T, class Function>
00452 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, QFuture<T>
00453     onFailed(QFuture<T> && future, QObject * context, Function && handler)
00454 {
00455     auto promise = std::make_shared<QPromise<T>>();
00456     auto result = promise->future();
00457
00458     if (future.isFinished()) {
00459         postToObject(
00460             context,
00461             [promise = std::move(promise), future = std::move(future),
00462              handler = std::forward<decltype(handler)>(handler)]() mutable {
00463                 detail::processPossibleFutureException(
00464                     std::move(promise), std::move(future),
00465                     std::forward<decltype(handler)>(handler));

```

```

00466         });
00467         return result;
00468     }
00469
00470     auto watcher = std::make_unique<QFutureWatcher<T>>();
00471     auto * rawWatcher = watcher.get();
00472     QObject::connect(
00473         rawWatcher, &QFutureWatcher<T>::finished, context,
00474         [context, rawWatcher, promise = std::move(promise),
00475          handler = std::forward<decltype(handler)>(handler)]() mutable {
00476             postToObject(
00477                 context,
00478                 [promise = std::move(promise), future = rawWatcher->future(),
00479                  handler = std::forward<decltype(handler)>(handler)]() mutable {
00480                     detail::processPossibleFutureException(
00481                         std::move(promise), std::move(future),
00482                         std::forward<decltype(handler)>(handler));
00483                     });
00484             rawWatcher->deleteLater();
00485         });
00486
00487     QObject::connect(
00488         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
00489         [rawWatcher] { rawWatcher->deleteLater(); });
00490
00491     watcher->setFuture(std::move(future));
00492     Q_UNUSED(watcher.release())
00493
00494     return result;
00495 }
00496
00497 // #endif // QT_VERSION
00498
00499 // Convenience functions for both Qt versions
00500
00501 template <class T, class U, class Function>
00502 void thenOrFailed(
00503     QFuture<T> && future, std::shared_ptr<QPromise<U>> promise,
00504     Function && function)
00505 {
00506     auto thenFuture =
00507         then(std::move(future), std::forward<decltype(function)>(function));
00508
00509     onFailed(std::move(thenFuture), [promise](const QException & e) {
00510         promise->setException(e);
00511         promise->finish();
00512     });
00513 }
00514
00515 template <class T, class U, class Function>
00516 void thenOrFailed(
00517     QFuture<T> && future, QThread * thread,
00518     std::shared_ptr<QPromise<U>> promise, Function && function)
00519 {
00520     auto thenFuture =
00521         then(std::move(future), thread, std::forward<Function>(function));
00522
00523     onFailed(std::move(thenFuture), thread, [promise](const QException & e) {
00524         promise->setException(e);
00525         promise->finish();
00526     });
00527 }
00528
00529 template <class T, class U>
00530 void thenOrFailed(QFuture<T> && future, std::shared_ptr<QPromise<U>> promise)
00531 {
00532     thenOrFailed(std::move(future), promise, [promise] { promise->finish(); });
00533 }
00534
00535 template <class T, class U>
00536 void thenOrFailed(
00537     QFuture<T> && future, QThread * thread,
00538     std::shared_ptr<QPromise<U>> promise)
00539 {
00540     thenOrFailed(
00541         std::move(future), thread, promise, [promise] { promise->finish(); });
00542 }
00543
00544 } // namespace quantier::threading

```

6.64 QtFutureHelpers.h

```
00001 /*
```

```

00002  * Copyright 2021-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <QFuture>
00022
00023 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
00024 #include "Qt5Promise.h"
00025 #endif
00026
00027 #include <type_traits>
00028
00029 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
00030 // Backports of some helpers for QFuture continuations from Qt6 to Qt5
00031 namespace QtFuture {
00032
00033 // Inherit option from Qt6 is not supported in Qt5
00034 enum class Launch
00035 {
00036     Sync,
00037     Async,
00038 };
00039
00040 } // namespace QtFuture
00041
00042 namespace QtPrivate {
00043
00044 template <typename...>
00045 struct ArgsType;
00046
00047 template <typename Arg, typename... Args>
00048 struct ArgsType<Arg, Args...>
00049 {
00050     using First = Arg;
00051     using PromiseType = void;
00052     using IsPromise = std::false_type;
00053     static const bool HasExtraArgs = (sizeof...(Args) > 0);
00054
00055     template <class Class, class Callable>
00056     static const bool CanInvokeWithArgs =
00057         std::is_invocable_v<Callable, Class, Arg, Args...>;
00058 };
00059
00060 template <typename Arg, typename... Args>
00061 struct ArgsType<QPromise<Arg> &, Args...>
00062 {
00063     using First = QPromise<Arg> &;
00064     using PromiseType = Arg;
00065     using IsPromise = std::true_type;
00066     static const bool HasExtraArgs = (sizeof...(Args) > 0);
00067
00068     template <class Class, class Callable>
00069     static const bool CanInvokeWithArgs =
00070         std::is_invocable_v<Callable, Class, QPromise<Arg> &, Args...>;
00071 };
00072
00073 template <>
00074 struct ArgsType<>
00075 {
00076     using First = void;
00077     using PromiseType = void;
00078     using IsPromise = std::false_type;
00079     static const bool HasExtraArgs = false;
00080     using AllArgs = void;
00081
00082     template <class Class, class Callable>
00083     static const bool CanInvokeWithArgs = std::is_invocable_v<Callable, Class>;
00084 };
00085
00086 template <typename F>
00087 struct ArgResolver : ArgResolver<decltype(&std::decay_t<F>::operator())>
00088 {};

```

```

00089
00090 template <typename F>
00091 struct ArgResolver<std::reference_wrapper<F> :
00092     ArgResolver<decltype(&std::decay_t<F>::operator())>
00093     >;
00094
00095 template <typename R, typename... Args>
00096 struct ArgResolver<R (Args...)> : public ArgsType<Args...>
00097 {};
00098
00099 template <typename R, typename... Args>
00100 struct ArgResolver<R (*) (Args...)> : public ArgsType<Args...>
00101 {};
00102
00103 template <typename R, typename... Args>
00104 struct ArgResolver<R (&) (Args...)> : public ArgsType<Args...>
00105 {};
00106
00107 template <typename R, typename... Args>
00108 struct ArgResolver<R (*const) (Args...)> : public ArgsType<Args...>
00109 {};
00110
00111 template <typename R, typename... Args>
00112 struct ArgResolver<R (&) (Args...)> : public ArgsType<Args...>
00113 {};
00114
00115 template <typename Class, typename R, typename... Args>
00116 struct ArgResolver<R (Class::*) (Args...)> : public ArgsType<Args...>
00117 {};
00118
00119 template <typename Class, typename R, typename... Args>
00120 struct ArgResolver<R (Class::*) (Args...) noexcept> : public ArgsType<Args...>
00121 {};
00122
00123 template <typename Class, typename R, typename... Args>
00124 struct ArgResolver<R (Class::*) (Args...) const> : public ArgsType<Args...>
00125 {};
00126
00127 template <typename Class, typename R, typename... Args>
00128 struct ArgResolver<R (Class::*) (Args...) const noexcept> :
00129     public ArgsType<Args...>
00130 {};
00131
00132 template <typename Class, typename R, typename... Args>
00133 struct ArgResolver<R (Class::*const) (Args...) const> : public ArgsType<Args...>
00134 {};
00135
00136 template <typename Class, typename R, typename... Args>
00137 struct ArgResolver<R (Class::*const) (Args...) const noexcept> :
00138     public ArgsType<Args...>
00139 {};
00140
00141 } // namespace QtPrivate
00142 #endif // QT_VERSION
00143
00144 namespace quantier::threading::detail {
00145
00146 template <typename F, typename Arg, typename Enable = void>
00147 struct ResultTypeHelper
00148 {};
00149
00150 // The callable takes an argument of type Arg
00151 template <typename F, typename Arg>
00152 struct ResultTypeHelper<
00153     F, Arg,
00154     typename std::enable_if_t<
00155         !std::is_invocable_v<std::decay_t<F>, QFuture<Arg>>
00156     > {
00157     using ResultType = std::invoke_result_t<std::decay_t<F>, std::decay_t<Arg>;
00158 };
00159
00160 // The callable takes an argument of type QFuture<Arg>
00161 template <class F, class Arg>
00162 struct ResultTypeHelper<
00163     F, Arg,
00164     typename std::enable_if_t<
00165         std::is_invocable_v<std::decay_t<F>, QFuture<Arg>>
00166     > {
00167     using ResultType = std::invoke_result_t<std::decay_t<F>, QFuture<Arg>;
00168 };
00169
00170 // The callable takes an argument of type QFuture<void>
00171 template <class F>
00172 struct ResultTypeHelper<
00173     F, void,
00174     typename std::enable_if_t<
00175         std::is_invocable_v<std::decay_t<F>, QFuture<void>>

```



```

00176 {
00177     using ResultType = std::invoke_result_t<std::decay_t<F>, QFuture<void>;
00178 };
00179
00180 // The callable doesn't take argument
00181 template <class F>
00182 struct ResultTypeHelper<
00183     F, void,
00184     typename std::enable_if_t<
00185         !std::is_invocable_v<std::decay_t<F>, QFuture<void>>>
00186 {
00187     using ResultType = std::invoke_result_t<std::decay_t<F>;
00188 };
00189
00190 } // namespace quentier::threading::detail

```

6.65 Runnable.h

```

00001 /*
00002  * Copyright 2021 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <functional>
00024
00025 class QRunnable;
00026
00027 namespace quentier::threading {
00028
00029 [[nodiscard]] auto QUINTIER_EXPORT
00030     createFunctionRunnable(std::function<void()> function) -> QRunnable *;
00031
00032 } // namespace quentier::threading

```

6.66 TrackedTask.h

```

00001 /*
00002  * Copyright 2022-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <functional>
00022 #include <type_traits>
00023 #include <utility>
00024
00025 namespace quentier::threading {

```

```

00026
00027 namespace detail {
00028
00029 template <typename LockableObject, typename Function, typename... Arguments>
00030 constexpr std::enable_if_t<std::is_invocable_v<Function, Arguments...> invoke(
00031     LockableObject & lockableObject, Function & function,
00032     Arguments &&... arguments)
00033 {
00034     const auto lockedObject = lockableObject.lock();
00035     if (lockedObject) {
00036         std::invoke(function, std::forward<Arguments>(arguments)...);
00037     }
00038 }
00039
00040 template <typename LockableObject, typename Function, typename... Arguments>
00041 constexpr std::enable_if_t<
00042     !std::is_invocable_v<Function, Arguments...> &&
00043     std::is_member_function_pointer_v<Function>
00044     invoke(
00045         LockableObject & lockableObject, Function & function,
00046         Arguments &&... arguments)
00047 {
00048     const auto lockedObject = lockableObject.lock();
00049     if (lockedObject) {
00050         std::invoke(
00051             function, *lockedObject, std::forward<Arguments>(arguments)...);
00052     }
00053 }
00054
00055 } // namespace detail
00056
00073 template <typename LockableObject, typename Function>
00074 class TrackedTask
00075 {
00076 public:
00077     template <typename SomeLockableObject, typename SomeFunction>
00078     constexpr TrackedTask(
00079         SomeLockableObject && someLockableObject, SomeFunction && function) :
00080         m_lockableObject{std::forward<SomeLockableObject>(someLockableObject)},
00081         m_function{std::forward<SomeFunction>(function)}
00082     {}
00083
00084     template <
00085         typename... Arguments,
00086         typename = std::enable_if_t<
00087             std::is_invocable_v<Function, Arguments...> ||
00088             std::is_member_function_pointer_v<Function>>
00089     > constexpr void operator()(Arguments &&... arguments)
00090     {
00091         detail::invoke(
00092             m_lockableObject, m_function,
00093             std::forward<Arguments>(arguments)...);
00094     }
00095
00096     template <
00097         typename... Arguments,
00098         typename = std::enable_if_t<
00099             std::is_invocable_v<Function, Arguments...> ||
00100             std::is_member_function_pointer_v<Function>>
00101     > constexpr void operator()(Arguments &&... arguments) const
00102     {
00103         detail::invoke(
00104             m_lockableObject, m_function,
00105             std::forward<Arguments>(arguments)...);
00106     }
00107
00108 private:
00109     LockableObject m_lockableObject;
00110     Function m_function;
00111 };
00112
00113 template <typename LockableObject, typename Function>
00114 TrackedTask(LockableObject, Function) -> TrackedTask<LockableObject, Function>;
00115
00116 } // namespace quantier::threading

```

6.67 Account.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *

```

```

00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Printable.h>
00022
00023 #include <qevercloud/QEverCloud.h>
00024
00025 #include <QSharedDataPointer>
00026 #include <QString>
00027
00028 namespace quentier {
00029
00030 class AccountData;
00031
00032 class QUENTIER_EXPORT Account : public utility::Printable
00033 {
00034 public:
00035     enum class Type
00036     {
00037         Local,
00038         Evernote
00039     };
00040
00041     friend QUENTIER_EXPORT QTextStream & operator<<(
00042         QTextStream & strm, Type type);
00043
00044     friend QUENTIER_EXPORT QDebug & operator<<(QDebug & dbg, Type type);
00045
00046     enum class EvernoteAccountType
00047     {
00048         Free,
00049         Plus,
00050         Premium,
00051         Business
00052     };
00053
00054     friend QUENTIER_EXPORT QTextStream & operator<<(
00055         QTextStream & strm, EvernoteAccountType type);
00056
00057     friend QUENTIER_EXPORT QDebug & operator<<(
00058         QDebug & dbg, EvernoteAccountType type);
00059 public:
00060     explicit Account();
00061
00062     explicit Account(
00063         QString name, Type type, qevercloud::UserID userId = -1,
00064         EvernoteAccountType evernoteAccountType = EvernoteAccountType::Free,
00065         QString evernoteHost = {}, QString shardId = {});
00066
00067     Account(const Account & other);
00068     Account(Account && other) noexcept;
00069
00070     Account & operator=(const Account & other);
00071     Account & operator=(Account && other) noexcept;
00072
00073     ~Account() noexcept override;
00074
00075     [[nodiscard]] bool operator==(const Account & other) const noexcept;
00076     [[nodiscard]] bool operator!=(const Account & other) const noexcept;
00077
00078     [[nodiscard]] bool isEmpty() const;
00079
00080     [[nodiscard]] QString name() const;
00081
00082     void setName(QString name);
00083
00084     [[nodiscard]] QString displayName() const;
00085
00086     void setDisplayName(QString displayName);
00087
00088     [[nodiscard]] Type type() const;
00089
00090     [[nodiscard]] qevercloud::UserID id() const;
00091
00092

```

```

00129     [[nodiscard]] EvernoteAccountType evernoteAccountType() const;
00130
00135     [[nodiscard]] QString evernoteHost() const;
00136
00142     [[nodiscard]] QString shardId() const;
00143
00144     void setEvernoteAccountType(EvernoteAccountType evernoteAccountType);
00145     void setEvernoteHost(QString evernoteHost);
00146     void setShardId(QString shardId);
00147
00148     [[nodiscard]] quint32 mailLimitDaily() const;
00149     [[nodiscard]] quint64 noteSizeMax() const;
00150     [[nodiscard]] quint64 resourceSizeMax() const;
00151     [[nodiscard]] quint32 linkedNotebookMax() const;
00152     [[nodiscard]] quint32 noteCountMax() const;
00153     [[nodiscard]] quint32 notebookCountMax() const;
00154     [[nodiscard]] quint32 tagCountMax() const;
00155     [[nodiscard]] quint32 noteTagCountMax() const;
00156     [[nodiscard]] quint32 savedSearchCountMax() const;
00157     [[nodiscard]] quint32 noteResourceCountMax() const;
00158
00159     void setEvernoteAccountLimits(const qevercloud::AccountLimits & limits);
00160
00161     // utility::Printable
00162     QTextStream & print(QTextStream & strm) const override;
00163
00164 private:
00165     QSharedDataPointer<AccountData> d;
00166 };
00167
00168 } // namespace quotientier

```

6.68 ErrorString.h

```

00001 /*
00002  * Copyright 2017-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquotientier
00005  *
00006  * libquotientier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquotientier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquotientier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quotientier/utility/Printable.h>
00022
00023 #include <QSharedDataPointer>
00024
00025 namespace quotientier {
00026
00027 class ErrorStringData;
00028
00042 class QUENTIER_EXPORT ErrorString : public utility::Printable
00043 {
00044 public:
00045     explicit ErrorString(const char * error = nullptr);
00046     explicit ErrorString(const QString & error);
00047
00048     ErrorString(const ErrorString & other);
00049     ErrorString(ErrorString && other) noexcept;
00050
00051     ErrorString & operator=(const ErrorString & other);
00052     ErrorString & operator=(ErrorString && other) noexcept;
00053
00054     ~ErrorString() override;
00055
00056     [[nodiscard]] const QString & base() const noexcept;
00057     [[nodiscard]] QString & base();
00058
00059     [[nodiscard]] const QStringList & additionalBases() const noexcept;
00060     [[nodiscard]] QStringList & additionalBases();
00061
00062     [[nodiscard]] const QString & details() const noexcept;

```

```

00063     [[nodiscard]] QString & details();
00064
00065     void setBase(QString error);
00066     void setBase(const char * error);
00067
00068     void appendBase(const QString & error);
00069     void appendBase(const QStringList & errors);
00070     void appendBase(const char * error);
00071
00072     void setDetails(const QString & error);
00073     void setDetails(const char * error);
00074
00075     [[nodiscard]] bool isEmpty() const;
00076     void clear();
00077
00078     [[nodiscard]] QString localizedString() const;
00079     [[nodiscard]] QString nonLocalizedString() const;
00080
00081     QTextStream & print(QTextStream & strm) const override;
00082
00083 private:
00084     QSharedDataPointer<ErrorStringData> d;
00085 };
00086
00087 [[nodiscard]] QUENTIER_EXPORT bool operator==(
00088     const ErrorString & lhs, const ErrorString & rhs) noexcept;
00089
00090 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
00091     const ErrorString & lhs, const ErrorString & rhs) noexcept;
00092
00093 } // namespace quantier

```

6.69 enml/conversion_rules/Fwd.h

```

00001 /*
00002  * Copyright 2023 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <memory>
00022
00023 namespace quantier::enml::conversion_rules {
00024
00025     class ISkipRule;
00026     using ISkipRulePtr = std::shared_ptr<ISkipRule>;
00027
00028     class ISkipRuleBuilder;
00029     using ISkipRuleBuilderPtr = std::shared_ptr<ISkipRuleBuilder>;
00030
00031 } // namespace quantier::enml::conversion_rules

```

6.70 enml/Fwd.h

```

00001 /*
00002  * Copyright 2016-2023 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,

```

```

00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <memory>
00022
00023 namespace quentier::enml {
00024
00025 class IConverter;
00026 using IConverterPtr = std::shared_ptr<IConverter>;
00027
00028 class IDecryptedTextCache;
00029 using IDecryptedTextCachePtr = std::shared_ptr<IDecryptedTextCache>;
00030
00031 class IENMLTagsConverter;
00032 using IENMLTagsConverterPtr = std::shared_ptr<IENMLTagsConverter>;
00033
00034 struct IHtmlData;
00035 using IHtmlDataPtr = std::shared_ptr<IHtmlData>;
00036
00037 } // namespace quentier::enml

```

6.71 local_storage/Fwd.h

```

00001 /*
00002  * Copyright 2020-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <memory>
00022
00023 namespace quentier::local_storage {
00024
00025 class ILocalStorage;
00026 using ILocalStoragePtr = std::shared_ptr<ILocalStorage>;
00027
00028 class ILocalStorageNotifier;
00029
00030 class IPatch;
00031 using IPatchPtr = std::shared_ptr<IPatch>;
00032
00033 class NoteSearchQuery;
00034
00035 } // namespace quentier::local_storage

```

6.72 synchronization/Fwd.h

```

00001 /*
00002  * Copyright 2020-2023 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,

```

```

00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <memory>
00022
00023 namespace quentier::synchronization {
00024
00025     class IAuthenticator;
00026     using IAuthenticatorPtr = std::shared_ptr<IAuthenticator>;
00027
00028     class INoteStoreFactory;
00029     using INoteStoreFactoryPtr = std::shared_ptr<INoteStoreFactory>;
00030
00031     class ISyncConflictResolver;
00032     using ISyncConflictResolverPtr = std::shared_ptr<ISyncConflictResolver>;
00033
00034     class ISynchronizer;
00035     using ISynchronizerPtr = std::shared_ptr<ISynchronizer>;
00036
00037     class ISyncEventsNotifier;
00038
00039     class ISyncOptions;
00040     using ISyncOptionsPtr = std::shared_ptr<ISyncOptions>;
00041
00042     class ISyncStateStorage;
00043     using ISyncStateStoragePtr = std::shared_ptr<ISyncStateStorage>;
00044
00045     class IUserStoreFactory;
00046     using IUserStoreFactoryPtr = std::shared_ptr<IUserStoreFactory>;
00047
00048 } // namespace quentier::synchronization

```

6.73 synchronization/types/Fwd.h

```

00001  /*
00002  * Copyright 2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <memory>
00022
00023 namespace quentier::synchronization {
00024
00025     class IAuthenticationInfo;
00026     using IAuthenticationInfoPtr = std::shared_ptr<IAuthenticationInfo>;
00027
00028     class IAuthenticationInfoBuilder;
00029     using IAuthenticationInfoBuilderPtr =
00030         std::shared_ptr<IAuthenticationInfoBuilder>;
00031
00032     class IDownloadNotesStatus;
00033     using IDownloadNotesStatusPtr = std::shared_ptr<IDownloadNotesStatus>;
00034
00035     class IDownloadResourcesStatus;
00036     using IDownloadResourcesStatusPtr = std::shared_ptr<IDownloadResourcesStatus>;
00037
00038     class ISendStatus;
00039     using ISendStatusPtr = std::shared_ptr<ISendStatus>;
00040
00041     struct ISyncChunksDataCounters;
00042     using ISyncChunksDataCountersPtr = std::shared_ptr<ISyncChunksDataCounters>;

```

```

00043
00044 class ISyncOptions;
00045 using ISyncOptionsPtr = std::shared_ptr<ISyncOptions>;
00046
00047 class ISyncOptionsBuilder;
00048 using ISyncOptionsBuilderPtr = std::shared_ptr<ISyncOptionsBuilder>;
00049
00050 class ISyncResult;
00051 using ISyncResultPtr = std::shared_ptr<ISyncResult>;
00052
00053 class ISyncState;
00054 using ISyncStatePtr = std::shared_ptr<ISyncState>;
00055
00056 class ISyncStateBuilder;
00057 using ISyncStateBuilderPtr = std::shared_ptr<ISyncStateBuilder>;
00058
00059 } // namespace quantier::synchronization

```

6.74 threading/Fwd.h

```

00001 /*
00002  * Copyright 2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <memory>
00022
00023 class QMutex;
00024 class QThread;
00025 class QThreadPool;
00026
00027 namespace quantier::threading {
00028
00029 using QMutexPtr = std::shared_ptr<QMutex>;
00030 using QThreadPtr = std::shared_ptr<QThread>;
00031 using QThreadPoolPtr = std::shared_ptr<QThreadPool>;
00032
00033 } // namespace quantier::threading

```

6.75 types/Fwd.h

```

00001 /*
00002  * Copyright 2023 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 namespace quantier {
00022

```



```

00023 class Account;
00024 class ErrorString;
00025 class ResourceRecognitionIndexItem;
00026 class ResourceRecognitionIndices;
00027
00028 } // namespace quantier

```

6.76 utility/cancelers/Fwd.h

```

00001 /*
00002  * Copyright 2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <memory>
00022
00023 namespace quantier::utility::cancelers {
00024
00025 class AnyOfCanceler;
00026 using AnyOfCancelerPtr = std::shared_ptr<AnyOfCanceler>;
00027
00028 class ICanceler;
00029 using ICancelerPtr = std::shared_ptr<ICanceler>;
00030
00031 class ManualCanceler;
00032 using ManualCancelerPtr = std::shared_ptr<ManualCanceler>;
00033
00034 } // namespace quantier::utility::cancelers

```

6.77 utility/Fwd.h

```

00001 /*
00002  * Copyright 2020-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <memory>
00022
00023 namespace quantier::utility {
00024
00025 class ApplicationSettings;
00026 class FileIOProcessorAsync;
00027 class QuantierApplication;
00028
00029 struct IEncryptor;
00030 using IEncryptorPtr = std::shared_ptr<IEncryptor>;
00031
00032 class IKeychainService;

```

```
00033 using IKeychainServicePtr = std::shared_ptr<IKeychainService>;
00034
00035 } // namespace quantier::utility
```

6.78 NoteUtils.h

```
00001 /*
00002  * Copyright 2020-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/types/Fwd.h>
00022 #include <quantier/utility/Linkage.h>
00023
00024 #include <qevercloud/types/Fwd.h>
00025
00026 #include <QStringList>
00027
00028 #include <utility>
00029
00030 namespace quantier {
00031
00032 [[nodiscard]] QUENTIER_EXPORT bool isInkNote(const qevercloud::Note & note);
00033
00034 [[nodiscard]] QUENTIER_EXPORT bool noteContentContainsCheckedToDo(
00035     const QString & noteContent);
00036
00037 [[nodiscard]] QUENTIER_EXPORT bool noteContentContainsUncheckedToDo(
00038     const QString & noteContent);
00039
00040 [[nodiscard]] QUENTIER_EXPORT bool noteContentContainsToDo(
00041     const QString & noteContent);
00042
00043 [[nodiscard]] QUENTIER_EXPORT bool noteContentContainsEncryptedFragments(
00044     const QString & noteContent);
00045
00046 [[nodiscard]] QUENTIER_EXPORT QString noteContentToPlainText(
00047     const QString & noteContent, ErrorString * errorDescription = nullptr);
00048
00049 [[nodiscard]] QUENTIER_EXPORT QStringList noteContentToListOfWords(
00050     const QString & noteContent, ErrorString * errorDescription = nullptr);
00051
00052 [[nodiscard]] QUENTIER_EXPORT std::pair<QString, QStringList>
00053     noteContentToPlainTextAndListOfWords(
00054         const QString & noteContent, ErrorString * errorDescription = nullptr);
00055
00056 } // namespace quantier
```

6.79 RegisterMetatypes.h

```
00001 /*
00002  * Copyright 2016-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
```

```

00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 namespace quentier {
00024
00025 QUINTIER_EXPORT void registerMetatypes();
00026
00027 } // namespace quentier

```

6.80 ResourceRecognitionIndexItem.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022 #include <quentier/utility/Printable.h>
00023
00024 #include <QByteArray>
00025 #include <QList>
00026 #include <QSharedDataPointer>
00027
00028 #include <memory>
00029
00030 namespace quentier {
00031
00032 class ResourceRecognitionIndexItemData;
00033
00034 class QUINTIER_EXPORT ResourceRecognitionIndexItem : public utility::Printable
00035 {
00036 public:
00037     explicit ResourceRecognitionIndexItem();
00038
00039     ResourceRecognitionIndexItem(const ResourceRecognitionIndexItem & other);
00040
00041     ResourceRecognitionIndexItem(
00042         ResourceRecognitionIndexItem && other) noexcept;
00043
00044     ResourceRecognitionIndexItem & operator=(
00045         const ResourceRecognitionIndexItem & other);
00046
00047     ResourceRecognitionIndexItem & operator=(
00048         ResourceRecognitionIndexItem && other) noexcept;
00049
00050     ~ResourceRecognitionIndexItem() override;
00051
00052     [[nodiscard]] bool isValid() const;
00053
00054     [[nodiscard]] int x() const;
00055     void setX(int x);
00056
00057     [[nodiscard]] int y() const;
00058     void setY(int y);
00059
00060     [[nodiscard]] int h() const;
00061     void setH(int h);
00062
00063     [[nodiscard]] int w() const;
00064     void setW(int w);
00065
00066     [[nodiscard]] int offset() const;

```

```

00067     void setOffset(int offset);
00068
00069     [[nodiscard]] int duration() const;
00070     void setDuration(int duration);
00071
00072     [[nodiscard]] QList<int> strokes() const;
00073     void setStrokes(QList<int> strokes);
00074
00075     struct QUINTIER_EXPORT ITextItem
00076     {
00077         virtual ~ITextItem();
00078
00079         [[nodiscard]] virtual QString text() const = 0;
00080         [[nodiscard]] virtual int weight() const = 0;
00081     };
00082
00083     using ITextItemPtr = std::shared_ptr<ITextItem>;
00084
00085     [[nodiscard]] QList<ITextItemPtr> textItems() const;
00086     void setTextItems(QList<ITextItemPtr> textItems);
00087
00088     struct QUINTIER_EXPORT IObjectItem
00089     {
00090         virtual ~IObjectItem();
00091
00092         [[nodiscard]] virtual QString objectType() const = 0;
00093         [[nodiscard]] virtual int weight() const = 0;
00094     };
00095
00096     using IObjectItemPtr = std::shared_ptr<IObjectItem>;
00097
00098     [[nodiscard]] QList<IObjectItemPtr> objectItems() const;
00099     void setObjectItems(QList<IObjectItemPtr> objectItems);
00100
00101     struct QUINTIER_EXPORT IShapeItem
00102     {
00103         virtual ~IShapeItem();
00104
00105         [[nodiscard]] virtual QString shape() const = 0;
00106         [[nodiscard]] virtual int weight() const = 0;
00107     };
00108
00109     using IShapeItemPtr = std::shared_ptr<IShapeItem>;
00110
00111     [[nodiscard]] QList<IShapeItemPtr> shapeItems() const;
00112     void setShapeItems(QList<IShapeItemPtr> shapeItems);
00113
00114     struct QUINTIER_EXPORT IBarcodeItem
00115     {
00116         virtual ~IBarcodeItem();
00117
00118         [[nodiscard]] virtual QString barcode() const = 0;
00119         [[nodiscard]] virtual int weight() const = 0;
00120     };
00121
00122     using IBarcodeItemPtr = std::shared_ptr<IBarcodeItem>;
00123
00124     [[nodiscard]] QList<IBarcodeItemPtr> barcodeItems() const;
00125     void setBarcodeItems(QList<IBarcodeItemPtr> barcodeItems);
00126
00127     // utility::Printable
00128     QTextStream & print(QTextStream & strm) const override;
00129
00130 private:
00131     QSharedDataPointer<ResourceRecognitionIndexItemData> d;
00132 };
00133
00134 } // namespace quentier

```

6.81 ResourceRecognitionIndices.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

```

```

00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/ResourceRecognitionIndexItem.h>
00022
00023 #include <QByteArray>
00024 #include <QSharedDataPointer>
00025 #include <QVector>
00026
00027 namespace quentier {
00028
00029 class ResourceRecognitionIndicesData;
00030
00031 class QUINTIER_EXPORT ResourceRecognitionIndices : public utility::Printable
00032 {
00033 public:
00034     explicit ResourceRecognitionIndices();
00035
00036     explicit ResourceRecognitionIndices(
00037         const QByteArray & rawRecognitionIndicesData);
00038
00039     ResourceRecognitionIndices(const ResourceRecognitionIndices & other);
00040     ResourceRecognitionIndices(ResourceRecognitionIndices && other) noexcept;
00041
00042     ResourceRecognitionIndices & operator=(
00043         const ResourceRecognitionIndices & other);
00044
00045     ResourceRecognitionIndices & operator=(
00046         ResourceRecognitionIndices && other) noexcept;
00047
00048     ~ResourceRecognitionIndices() override;
00049
00050     [[nodiscard]] bool isNull() const;
00051     [[nodiscard]] bool isValid() const;
00052
00053     [[nodiscard]] QString objectId() const;
00054     [[nodiscard]] QString objectType() const;
00055     [[nodiscard]] QString recoType() const;
00056     [[nodiscard]] QString engineVersion() const;
00057     [[nodiscard]] QString docType() const;
00058     [[nodiscard]] QString lang() const;
00059
00060     [[nodiscard]] int objectHeight() const;
00061     [[nodiscard]] int objectWidth() const;
00062
00063     [[nodiscard]] QVector<ResourceRecognitionIndexItem> items() const;
00064
00065     bool setData(const QByteArray & rawRecognitionIndicesData);
00066
00067     // utility::Printable
00068     QTextStream & print(QTextStream & strm) const override;
00069
00070 private:
00071     QSharedDataPointer<ResourceRecognitionIndicesData> d;
00072 };
00073
00074 } // namespace quentier

```

6.82 ResourceUtils.h

```

00001 /*
00002  * Copyright 2020-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018

```

```

00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <qevercloud/types/Fwd.h>
00024
00025 #include <QString>
00026
00027 namespace quentier {
00028
00029 [[nodiscard]] QUENTIER_EXPORT QString
00030     resourceDisplayName(const qevercloud::Resource & resource);
00031
00032 [[nodiscard]] QUENTIER_EXPORT QString
00033     preferredFileSuffix(const qevercloud::Resource & resource);
00034
00035 } // namespace quentier

```

6.83 Result.h

```

00001 /*
00002  * Copyright 2023-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/exception/RuntimeError.h>
00022 #include <quentier/types/ErrorString.h>
00023
00024 #include <type_traits>
00025 #include <variant>
00026
00027 namespace quentier {
00028
00029 template <
00030     class ValueType, class ErrorType,
00031     typename =
00032         typename std::enable_if_t<!std::is_void_v<std::decay_t<ErrorType>>>
00033 class Result
00034 {
00035 private:
00036     template <typename T>
00037     struct ValueWrapper
00038     {
00039         T value;
00040     };
00041
00042     using ValueWrapperInnerType = std::conditional_t<
00043         std::is_void_v<std::decay_t<ValueType>, std::nullptr_t,
00044         std::decay_t<ValueType>;
00045
00046 public:
00047     template <
00048         typename T1 = ValueType,
00049         typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>>> * =
00050             nullptr>
00051     explicit Result(T1 t) :
00052         m_valueOrError{ValueWrapper<std::decay_t<ValueType>>{std::move(t)}}
00053     {}
00054
00055     template <
00056         typename T1 = ValueType,
00057         typename std::enable_if_t<std::is_void_v<std::decay_t<T1>>> * = nullptr>
00058     explicit Result() : m_valueOrError{ValueWrapper<std::nullptr_t>{}}
00059     {}
00060
00061     explicit Result(ErrorType error) : m_valueOrError{std::move(error)} {}
00062
00063     Result(const Result<ValueType, ErrorType> & other) :

```

```

00064     m_valueOrError{other.m_valueOrError}
00065     {}
00066
00067     Result(Result<ValueType, ErrorType> && other) :
00068         m_valueOrError{std::move(other.m_valueOrError)}
00069     {}
00070
00071     Result & operator=(const Result<ValueType, ErrorType> & other)
00072     {
00073         if (this != &other) {
00074             m_valueOrError = other.m_valueOrError;
00075         }
00076         return *this;
00077     }
00078
00079     Result & operator=(Result<ValueType, ErrorType> && other)
00080     {
00081         if (this != &other) {
00082             m_valueOrError = std::move(other.m_valueOrError);
00083         }
00084         return *this;
00085     }
00086
00087
00088 [[nodiscard]] bool isValid() const noexcept
00089 {
00090     return std::holds_alternative<ValueWrapper<ValueWrapperInnerType>>(
00091         m_valueOrError);
00092 }
00093
00094 operator bool() const noexcept
00095 {
00096     return isValid();
00097 }
00098
00099 template <
00100     typename T1 = ValueType,
00101     typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>>> * =
00102         nullptr>
00103 [[nodiscard]] T1 & get()
00104 {
00105     // NOTE: std::get also performs the check of what is stored inside the
00106     // variant but it throws std::bad_variant_access which doesn't implement
00107     // QException so this exception is not representable inside QFuture
00108     // in Qt5. Due to this for Qt5 also performing another check and using
00109     // another exception type
00110     #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
00111         if (Q_UNLIKELY(!isValid())) {
00112             throw RuntimeError{
00113                 ErrorString{"Detected attempt to get value from empty Result"}};
00114         }
00115     #endif
00116     return std::get<ValueWrapper<std::decay_t<ValueType>>>(m_valueOrError)
00117         .value;
00118 }
00119
00120 template <
00121     typename T1 = ValueType,
00122     typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>>> * =
00123         nullptr>
00124 [[nodiscard]] const T1 & get() const
00125 {
00126     // NOTE: std::get also performs the check of what is stored inside the
00127     // variant but it throws std::bad_variant_access which doesn't implement
00128     // QException so this exception is not representable inside QFuture
00129     // in Qt5. Due to this for Qt5 also performing another check and using
00130     // another exception type
00131     #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
00132         if (Q_UNLIKELY(!isValid())) {
00133             throw RuntimeError{
00134                 ErrorString{"Detected attempt to get value from empty Result"}};
00135         }
00136     #endif
00137     return std::get<ValueWrapper<std::decay_t<ValueType>>>(m_valueOrError)
00138         .value;
00139 }
00140
00141 template <
00142     typename T1 = ValueType,
00143     typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>>> * =
00144         nullptr>
00145 [[nodiscard]] T1 & operator*()
00146 {
00147     return get();
00148 }

```

```

00154     }
00155
00156     template <
00157         typename T1 = ValueType,
00158         typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>> * =
00159             nullptr>
00160     [[nodiscard]] const T1 & operator*() const
00161     {
00162         return get();
00163     }
00164
00165     [[nodiscard]] const ErrorType & error() const
00166     {
00167         // NOTE: std::get also performs the check of what is stored inside the
00168         // variant but it throws std::bad_variant_access which doesn't implement
00169         // QException so this exception is not representable inside QFuture
00170         // in Qt5. Due to this for Qt5 also performing another check and using
00171         // another exception type
00172 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
00173         if (Q_UNLIKELY(isValid())) {
00174             throw RuntimeError{ErrorString{
00175                 "Detected attempt to get error from non-empty Result"}};
00176         }
00177 #endif
00178
00179         return std::get<ErrorType>(m_valueOrError);
00180     }
00181
00182     [[nodiscard]] ErrorType & error()
00183     {
00184         // NOTE: std::get also performs the check of what is stored inside the
00185         // variant but it throws std::bad_variant_access which doesn't implement
00186         // QException so this exception is not representable inside QFuture
00187         // in Qt5. Due to this for Qt5 also performing another check and using
00188         // another exception type
00189 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
00190         if (Q_UNLIKELY(isValid())) {
00191             throw RuntimeError{ErrorString{
00192                 "Detected attempt to get error from non-empty Result"}};
00193         }
00194 #endif
00195
00196         return std::get<ErrorType>(m_valueOrError);
00197     }
00198
00199 private:
00200     std::variant<ValueWrapper<ValueWrapperInnerType>, ErrorType> m_valueOrError;
00201 };
00202
00203 } // namespace quentier

```

6.84 Validation.h

```

00001 /*
00002  * Copyright 2020-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 class QString;
00025
00026 namespace quentier {
00027
00028 [[nodiscard]] QUINTIER_EXPORT bool validateNoteTitle(
00029     const QString & noteTitle,
00030     ErrorString * errorDescription = nullptr) noexcept;

```



```

00041
00052 [[nodiscard]] QUINTIER_EXPORT bool validateNotebookName(
00053     const QString & notebookName,
00054     QString * errorDescription = nullptr) noexcept;
00055
00067 [[nodiscard]] QUINTIER_EXPORT bool validateSavedSearchName(
00068     const QString & savedSearchName,
00069     QString * errorDescription = nullptr) noexcept;
00070
00081 [[nodiscard]] QUINTIER_EXPORT bool validateTagName(
00082     const QString & tagName, QString * errorDescription = nullptr) noexcept;
00083
00084 } // namespace quentier

```

6.85 ApplicationSettings.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/Account.h>
00022
00023 #include <QSettings>
00024
00025 #include <string_view>
00026
00027 namespace quentier::utility {
00028
00034 class QUINTIER_EXPORT ApplicationSettings :
00035     public QSettings,
00036     public utility::Printable
00037 {
00038     Q_OBJECT
00039 public:
00049     explicit ApplicationSettings(const QString & settingsName = {});
00050
00062     explicit ApplicationSettings(
00063         const Account & account, const QString & settingsName = {});
00064
00081     ApplicationSettings(
00082         const Account & account, const char * settingsName,
00083         int settingsNameSize = -1);
00084
00098     ApplicationSettings(const Account & account, std::string_view settingsName);
00099
00103     ~ApplicationSettings() override;
00104
00109     struct ArrayCloser
00110     {
00111         ArrayCloser(ApplicationSettings & settings) : m_settings(settings) {}
00112
00113         ~ArrayCloser()
00114         {
00115             m_settings.endArray();
00116             m_settings.sync();
00117         }
00118
00119         ApplicationSettings & m_settings;
00120     };
00121
00126     struct GroupCloser
00127     {
00128         GroupCloser(ApplicationSettings & settings) : m_settings(settings) {}
00129
00130         ~GroupCloser()
00131         {
00132             m_settings.endGroup();

```

```

00133         m_settings.sync();
00134     }
00135
00136     ApplicationSettings & m_settings;
00137 };
00138
00145 void beginGroup(const QString & prefix);
00146
00157 void beginGroup(const char * prefix, int size = -1);
00158
00168 void beginGroup(std::string_view prefix);
00169
00177 [[nodiscard]] int beginReadArray(const QString & prefix);
00178
00189 [[nodiscard]] int beginReadArray(const char * prefix, int size = -1);
00190
00198 [[nodiscard]] int beginReadArray(std::string_view prefix);
00199
00210 void beginWriteArray(const QString & prefix, int arraySize = -1);
00211
00226 void beginWriteArray(
00227     const char * prefix, int arraySize = -1, int prefixSize = -1);
00228
00240 void beginWriteArray(std::string_view prefix, int arraySize = -1);
00241
00249 [[nodiscard]] bool contains(const QString & key) const;
00250
00262 [[nodiscard]] bool contains(const char * key, int size = -1) const;
00263
00272 [[nodiscard]] bool contains(std::string_view key) const;
00273
00280 void remove(const QString & key);
00281
00292 void remove(const char * key, int size = -1);
00293
00301 void remove(std::string_view key);
00302
00310 void setValue(const QString & key, const QVariant & value);
00311
00323 void setValue(const char * key, const QVariant & value, int keySize = -1);
00324
00333 void setValue(std::string_view key, const QVariant & value);
00334
00345 [[nodiscard]] QVariant value(
00346     const QString & key, const QVariant & defaultValue = {}) const;
00347
00362 [[nodiscard]] QVariant value(
00363     const char * key, const QVariant & defaultValue = {},
00364     int keySize = -1) const;
00365
00377 [[nodiscard]] QVariant value(
00378     std::string_view key, const QVariant & defaultValue = {}) const;
00379
00380 // utility::Printable
00381 QTextStream & print(QTextStream & strm) const override;
00382
00383 private:
00384     Q_DISABLE_COPY(ApplicationSettings)
00385 };
00386
00387 } // namespace quantier::utility

```

6.86 AnyOfCanceler.h

```

00001 /*
00002  * Copyright 2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once

```

```

00020
00021 #include <quentier/utility/cancelers/Fwd.h>
00022 #include <quentier/utility/cancelers/ICanceler.h>
00023
00024 #include <QList>
00025
00026 namespace quentier::utility::cancelers {
00027
00028 class QUINTIER_EXPORT AnyOfCanceler : public ICanceler
00029 {
00030 public:
00031     explicit AnyOfCanceler(QList<ICancelerPtr> cancelers);
00032     AnyOfCanceler(AnyOfCanceler && other) noexcept;
00033     AnyOfCanceler & operator=(AnyOfCanceler && other) noexcept;
00034     ~AnyOfCanceler() noexcept override;
00035
00036     [[nodiscard]] bool isCanceled() const noexcept override;
00037
00038 private:
00039     class Impl;
00040     std::unique_ptr<Impl> m_impl;
00041 };
00042
00043 } // namespace quentier::utility::cancelers

```

6.87 FutureCanceler.h

```

00001 /*
00002  * Copyright 2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/cancelers/ICanceler.h>
00022
00023 #include <QFuture>
00024
00025 namespace quentier::utility::cancelers {
00026
00027 template <class T>
00028 class FutureCanceler : public ICanceler
00029 {
00030 public:
00031     explicit FutureCanceler(QFuture<T> future) : m_future{std::move(future)} {}
00032
00033     [[nodiscard]] bool isCanceled() const noexcept override
00034     {
00035         return m_future.isCanceled();
00036     }
00037
00038 private:
00039     QFuture<T> m_future;
00040 };
00041
00042 } // namespace quentier::utility::cancelers

```

6.88 ICanceler.h

```

00001 /*
00002  * Copyright 2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify

```

```

00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 namespace quentier::utility::cancelers {
00024
00025 class QUENTIER_EXPORT ICanceler
00026 {
00027 public:
00028     virtual ~ICanceler() = default;
00029
00030     [[nodiscard]] virtual bool isCanceled() const = 0;
00031 };
00032
00033 } // namespace quentier::utility::cancelers

```

6.89 ManualCanceler.h

```

00001 /*
00002  * Copyright 2022 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/cancelers/ICanceler.h>
00022
00023 #include <atomic>
00024 #include <memory>
00025
00026 namespace quentier::utility::cancelers {
00027
00028 class QUENTIER_EXPORT ManualCanceler : public ICanceler
00029 {
00030 public:
00031     ManualCanceler();
00032     ManualCanceler(ManualCanceler && other) noexcept;
00033     ManualCanceler & operator=(ManualCanceler && other) noexcept;
00034     ~ManualCanceler() noexcept override;
00035
00036     void cancel() noexcept;
00037
00038     [[nodiscard]] bool isCanceled() const noexcept override;
00039
00040 private:
00041     class Impl;
00042     std::unique_ptr<Impl> m_impl;
00043 };
00044
00045 } // namespace quentier::utility::cancelers

```

6.90 Compat.h

```

00001 /*

```

```

00002  * Copyright 2020-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <QHash>
00022 #include <QString>
00023 #include <QtGlobal>
00024
00025 // Compatibility with boost parts which require to take a hash of QString
00026
00027 inline std::size_t hash_value(const QString & x) noexcept
00028 {
00029     return qHash(x);
00030 }

```

6.91 DateTime.h

```

00001 /*
00002  * Copyright 2020-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QFlags>
00024
00025 namespace quentier::utility {
00026
00027 [[nodiscard]] constexpr int secondsToMilliseconds(int seconds) noexcept
00028 {
00029     return seconds * 1000;
00030 }
00031
00032 enum class DateTimePrintOption
00033 {
00034     IncludeNumericTimestamp = 1 « 1,
00035     IncludeMilliseconds = 1 « 2,
00036     IncludeTimezone = 1 « 3
00037 };
00038
00039 Q_DECLARE_FLAGS(DateTimePrintOptions, DateTimePrintOption)
00040 Q_DECLARE_OPERATORS_FOR_FLAGS(DateTimePrintOptions)
00041
00042 [[nodiscard]] QString QUENTIER_EXPORT printableDateTimeFromTimestamp(
00043     qint64 timestamp,
00044     DateTimePrintOptions options = DateTimePrintOptions(
00045         DateTimePrintOption::IncludeNumericTimestamp |
00046         DateTimePrintOption::IncludeMilliseconds |
00047         DateTimePrintOption::IncludeTimezone),
00048     const char * customFormat = nullptr);
00049
00050 } // namespace quentier::utility

```

6.92 EventLoopWithExitStatus.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/ErrorMessage.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QEventLoop>
00025
00026 class QDebug;
00027 class QTextStream;
00028
00029 namespace quentier::utility {
00030
00031 class QUENTIER_EXPORT EventLoopWithExitStatus : public QEventLoop
00032 {
00033     Q_OBJECT
00034 public:
00035     explicit EventLoopWithExitStatus(QObject * parent = nullptr);
00036
00037     enum class ExitStatus
00038     {
00039         Success,
00040         Failure,
00041         Timeout
00042     };
00043
00044     friend QDebug & operator<<(QDebug & dbg, ExitStatus status);
00045     friend QTextStream & operator<<(QTextStream & strm, ExitStatus status);
00046
00047     [[nodiscard]] ExitStatus exitStatus() const;
00048     [[nodiscard]] const ErrorMessage & errorDescription() const;
00049
00050 public Q_SLOTS:
00051     void exitAsSuccess();
00052     void exitAsFailure();
00053     void exitAsFailureWithError(QString errorDescription);
00054     void exitAsFailureWithError(ErrorMessage errorDescription);
00055     void exitAsTimeout();
00056
00057 private:
00058     ExitStatus m_exitStatus;
00059     ErrorMessage m_errorDescription;
00060 };
00061
00062 } // namespace quentier::utility

```

6.93 FileIOProcessorAsync.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License

```

```

00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/ErrorString.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QByteArray>
00025 #include <QIODevice>
00026 #include <QObject>
00027 #include <QString>
00028 #include <QUuid>
00029
00030 namespace quentier::utility {
00031
00032 class FileIOProcessorAsyncPrivate;
00033
00034 class QUENTIER_EXPORT FileIOProcessorAsync : public QObject
00035 {
00036     Q_OBJECT
00037 public:
00038     explicit FileIOProcessorAsync(QObject * parent = nullptr);
00039
00040     void setIdleTimePeriod(qint32 seconds);
00041
00042 Q_SIGNALS:
00043     void readyForIO();
00044
00045     void writeFileRequestProcessed(
00046         bool success, ErrorString errorDescription, QUuid requestId);
00047
00048     void readFileRequestProcessed(
00049         bool success, ErrorString errorDescription, QByteArray data,
00050         QUuid requestId);
00051
00052 public Q_SLOTS:
00053     void onWriteFileRequest(
00054         QString absoluteFilePath, QByteArray data, QUuid requestId,
00055         bool append);
00056
00057     void onReadFileRequest(QString absoluteFilePath, QUuid requestId);
00058
00059 private:
00060     FileIOProcessorAsyncPrivate * const d_ptr;
00061     Q_DECLARE_PRIVATE(FileIOProcessorAsync)
00062 };
00063
00064 } // namespace quentier::utility

```

6.94 FileSystem.h

```

00001 /*
00002  * Copyright 2020-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QString>
00025
00026 namespace quentier::utility {
00027
00028 [[nodiscard]] QString QUENTIER_EXPORT relativePathFromAbsolutePath(
00029     const QString & absolutePath, const QString & relativePathRootFolderPath);
00030
00031 [[nodiscard]] bool QUENTIER_EXPORT removeFile(const QString & filePath);

```

```

00054
00064 [[nodiscard]] bool QUENTIER_EXPORT removeDir(const QString & dirPath);
00065
00077 [[nodiscard]] QByteArray QUENTIER_EXPORT
00078     readFileContents(const QString & filePath, ErrorString & errorDescription);
00079
00094 [[nodiscard]] bool QUENTIER_EXPORT renameFile(
00095     const QString & from, const QString & to, ErrorString & errorDescription);
00096
00097 } // namespace quentier::utility

```

6.95 FileSystemWatcher.h

```

00001 /*
00002  * Copyright 2016–2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QObject>
00024 #include <QStringList>
00025
00026 namespace quentier::utility {
00027
00028 class QUENTIER_EXPORT FileSystemWatcher : public QObject
00029 {
00030     Q_OBJECT
00031 public:
00032     explicit FileSystemWatcher(
00033         int removalTimeoutMSec = 500, QObject * parent = nullptr);
00034
00035     explicit FileSystemWatcher(
00036         const QStringList & paths, int removalTimeoutMSec = 500,
00037         QObject * parent = nullptr);
00038
00039     ~FileSystemWatcher() override;
00040
00041     void addPath(const QString & path);
00042     void addPaths(const QStringList & paths);
00043
00044     [[nodiscard]] QStringList directories() const;
00045     [[nodiscard]] QStringList files() const;
00046
00047     void removePath(const QString & path);
00048     void removePaths(const QStringList & paths);
00049
00050     Q_SIGNALS:
00051         void directoryChanged(const QString & path);
00052         void directoryRemoved(const QString & path);
00053
00054         void fileChanged(const QString & path);
00055         void fileRemoved(const QString & path);
00056
00057 private:
00058     Q_DISABLE_COPY(FileSystemWatcher)
00059
00060 private:
00061     class FileSystemWatcherPrivate;
00062
00063     FileSystemWatcherPrivate * d_ptr;
00064     Q_DECLARE_PRIVATE(FileSystemWatcher)
00065 };
00066
00067 } // namespace quentier::utility

```


6.96 IEncryptor.h

```

00001 /*
00002  * Copyright 2024-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/ErrorMessage.h>
00022 #include <quentier/types/Result.h>
00023 #include <quentier/utility/Linkage.h>
00024
00025 #include <QString>
00026
00027 class QDebug;
00028 class QTextStream;
00029
00030 namespace quentier::utility {
00031
00032 struct QUENTIER_EXPORT IEncryptor
00033 {
00034     virtual ~IEncryptor() noexcept;
00035
00036     enum class Cipher
00037     {
00038         RC2,
00039         AES
00040     };
00041
00042     friend QUENTIER_EXPORT QDebug & operator<<(QDebug & dbg, Cipher cipher);
00043
00044     friend QUENTIER_EXPORT QTextStream & operator<<(
00045         QTextStream & strm, Cipher cipher);
00046
00047     [[nodiscard]] virtual Result<QString, ErrorMessage> encrypt(
00048         const QString & text, const QString & passphrase) = 0;
00049
00050     [[nodiscard]] virtual Result<QString, ErrorMessage> decrypt(
00051         const QString & encryptedText, const QString & passphrase,
00052         Cipher cipher) = 0;
00053 };
00054
00055 } // namespace quentier::utility

```

6.97 IKeychainService.h

```

00001 /*
00002  * Copyright 2018-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/exception/IQuentierException.h>
00022 #include <quentier/types/ErrorMessage.h>

```

```

00023
00024 #include <quentier/utility/Fwd.h>
00025 #include <quentier/utility/Linkage.h>
00026
00027 #include <QFuture>
00028
00029 class QDebug;
00030
00031 namespace quentier::utility {
00032
00033     class QUENTIER_EXPORT IKeychainService
00034     {
00035     public:
00036         virtual ~IKeychainService() noexcept;
00037
00038         enum class ErrorCode
00039         {
00040             NoError,
00041             EntryNotFound,
00042             CouldNotDeleteEntry,
00043             AccessDeniedByUser,
00044             AccessDenied,
00045             NoBackendAvailable,
00046             NotImplemented,
00047             OtherError
00048         };
00049
00050         friend QUENTIER_EXPORT QTextStream & operator<<(
00051             QTextStream & strm, ErrorCode errorCode);
00052
00053         friend QUENTIER_EXPORT QDebug & operator<<(
00054             QDebug & dbg, ErrorCode errorCode);
00055
00056         class QUENTIER_EXPORT Exception : public IQuentierException
00057         {
00058         public:
00059             explicit Exception(ErrorCode errorCode) noexcept;
00060
00061             explicit Exception(
00062                 ErrorCode errorCode, QString errorDescription) noexcept;
00063
00064             [[nodiscard]] ErrorCode errorCode() const noexcept;
00065             [[nodiscard]] QString exceptionDisplayName() const override;
00066
00067             void raise() const override;
00068             [[nodiscard]] Exception * clone() const override;
00069
00070         private:
00071             const ErrorCode m_errorCode;
00072         };
00073
00074     public:
00075         [[nodiscard]] virtual QFuture<void> writePassword(
00076             QString service, QString key, QString password) = 0;
00077
00078         [[nodiscard]] virtual QFuture<QString> readPassword(
00079             QString service, QString key) const = 0;
00080
00081         [[nodiscard]] virtual QFuture<void> deletePassword(
00082             QString service, QString key) = 0;
00083     };
00084 } // namespace quentier::utility

```

6.98 Initialize.h

```

00001 /*
00002  * Copyright 2020–2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */

```

```

00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 namespace quentier::utility {
00024
00025 void QUENTIER_EXPORT initializeLibquentier();
00030
00031 } // namespace quentier::utility

```

6.99 LRUCache.hpp

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <QHash>
00022
00023 #include <cstddef>
00024 #include <list>
00025
00026 namespace quentier::utility {
00027
00028 template <
00029     class Key, class Value,
00030     class Allocator = std::allocator<std::pair<Key, Value>>
00031 class LRUCache
00032 {
00033 public:
00034     explicit LRUCache(const std::size_t maxSize = 100) : m_maxSize{maxSize} {}
00035
00036     using key_type = Key;
00037     using mapped_type = Value;
00038     using allocator_type = Allocator;
00039     using value_type = std::pair<key_type, mapped_type>;
00040     using container_type = std::list<value_type, allocator_type>;
00041     using size_type = typename container_type::size_type;
00042     using difference_type = typename container_type::difference_type;
00043     using iterator = typename container_type::iterator;
00044     using const_iterator = typename container_type::const_iterator;
00045     using reverse_iterator = std::reverse_iterator<iterator>;
00046     using const_reverse_iterator = std::reverse_iterator<const_iterator>;
00047
00048     using reference = value_type &;
00049     using const_reference = const value_type &;
00050     using pointer = typename std::allocator_traits<allocator_type>::pointer;
00051
00052     using const_pointer =
00053         typename std::allocator_traits<allocator_type>::const_pointer;
00054
00055     [[nodiscard]] iterator begin() noexcept
00056     {
00057         return m_container.begin();
00058     }
00059
00060     [[nodiscard]] const_iterator begin() const noexcept
00061     {
00062         return m_container.begin();
00063     }
00064
00065     [[nodiscard]] reverse_iterator rbegin() noexcept
00066     {
00067         return m_container.rbegin();
00068     }
00069
00070     [[nodiscard]] const_reverse_iterator rbegin() const noexcept

```

```

00071     {
00072         return m_container.rbegin();
00073     }
00074
00075     [[nodiscard]] iterator end() noexcept
00076     {
00077         return m_container.end();
00078     }
00079
00080     [[nodiscard]] const_iterator end() const noexcept
00081     {
00082         return m_container.end();
00083     }
00084
00085     [[nodiscard]] reverse_iterator rend() noexcept
00086     {
00087         return m_container.rend();
00088     }
00089
00090     [[nodiscard]] const_reverse_iterator rend() const noexcept
00091     {
00092         return m_container.rend();
00093     }
00094
00095     [[nodiscard]] bool empty() const noexcept
00096     {
00097         return m_container.empty();
00098     }
00099
00100     [[nodiscard]] std::size_t size() const noexcept
00101     {
00102         return m_currentSize;
00103     }
00104
00105     [[nodiscard]] std::size_t max_size() const noexcept
00106     {
00107         return m_maxSize;
00108     }
00109
00110     void clear()
00111     {
00112         m_container.clear();
00113         m_mapper.clear();
00114         m_currentSize = 0;
00115     }
00116
00117     void put(const key_type & key, const mapped_type & value)
00118     {
00119         Q_UNUSED(remove(key))
00120
00121         m_container.push_front(value_type(key, value));
00122         m_mapper[key] = m_container.begin();
00123         ++m_currentSize;
00124
00125         fixupSize();
00126     }
00127
00128     [[nodiscard]] const mapped_type * get(const key_type & key) const noexcept
00129     {
00130         auto mapperIt = m_mapper.find(key);
00131         if (mapperIt == m_mapper.end()) {
00132             return nullptr;
00133         }
00134
00135         auto it = mapperIt.value();
00136         if (it == m_container.end()) {
00137             return nullptr;
00138         }
00139
00140         m_container.splice(m_container.begin(), m_container, it);
00141         mapperIt.value() = m_container.begin();
00142         return &(mapperIt.value()->second);
00143     }
00144
00145     [[nodiscard]] bool exists(const key_type & key) const noexcept
00146     {
00147         const auto mapperIt = m_mapper.find(key);
00148         if (mapperIt == m_mapper.end()) {
00149             return false;
00150         }
00151
00152         const auto it = mapperIt.value();
00153         return (it != m_container.end());
00154     }
00155
00156     bool remove(const key_type & key) noexcept
00157     {

```

```

00158         const auto mapperIt = m_mapper.find(key);
00159         if (mapperIt == m_mapper.end()) {
00160             return false;
00161         }
00162
00163         const auto it = mapperIt.value();
00164         Q_UNUSED(m_container.erase(it))
00165         Q_UNUSED(m_mapper.erase(mapperIt))
00166
00167         if (m_currentSize != 0) {
00168             --m_currentSize;
00169         }
00170
00171         return true;
00172     }
00173
00174     void setMaxSize(const std::size_t maxSize)
00175     {
00176         if (maxSize >= m_maxSize) {
00177             m_maxSize = maxSize;
00178             return;
00179         }
00180
00181         std::size_t diff = m_maxSize - maxSize;
00182         for (std::size_t i = 0; (i < diff) && !m_container.empty(); ++i) {
00183             auto lastIt = m_container.end();
00184             --lastIt;
00185
00186             const key_type & lastElementKey = lastIt->first;
00187             Q_UNUSED(m_mapper.remove(lastElementKey))
00188             Q_UNUSED(m_container.erase(lastIt))
00189
00190             if (m_currentSize != 0) {
00191                 --m_currentSize;
00192             }
00193         }
00194     }
00195
00196 private:
00197     void fixupSize()
00198     {
00199         if (m_currentSize <= m_maxSize) {
00200             return;
00201         }
00202
00203         if (Q_UNLIKELY(m_container.empty())) {
00204             return;
00205         }
00206
00207         auto lastIt = m_container.end();
00208         --lastIt;
00209
00210         const key_type & lastElementKey = lastIt->first;
00211
00212         Q_UNUSED(m_mapper.remove(lastElementKey))
00213         Q_UNUSED(m_container.erase(lastIt))
00214
00215         if (m_currentSize != 0) {
00216             --m_currentSize;
00217         }
00218     }
00219
00220 private:
00221     mutable container_type m_container;
00222     std::size_t m_currentSize = 0;
00223     std::size_t m_maxSize;
00224
00225     mutable QHash<Key, iterator> m_mapper;
00226 };
00227
00228 } // namespace quantier::utility

```

6.100 MessageBox.h

```

00001 /*
00002  * Copyright 2017-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *

```

```

00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QMessageBox>
00024
00025 namespace quentier::utility {
00026
00027 int QUENTIER_EXPORT genericMessageBox(
00028     QWidget * parent, const QString & title, const QString & briefText,
00029     const QString & detailedText = {},
00030     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
00031
00032 int QUENTIER_EXPORT informationMessageBox(
00033     QWidget * parent, const QString & title, const QString & briefText,
00034     const QString & detailedText = {},
00035     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
00036
00037 int QUENTIER_EXPORT warningMessageBox(
00038     QWidget * parent, const QString & title, const QString & briefText,
00039     const QString & detailedText = {},
00040     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
00041
00042 int QUENTIER_EXPORT criticalMessageBox(
00043     QWidget * parent, const QString & title, const QString & briefText,
00044     const QString & detailedText = {},
00045     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
00046
00047 [[nodiscard]] int QUENTIER_EXPORT questionMessageBox(
00048     QWidget * parent, const QString & title, const QString & briefText,
00049     const QString & detailedText = {},
00050     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok |
00051     QMessageBox::Cancel);
00052
00053 void QUENTIER_EXPORT
00054     internalErrorMessageBox(QWidget * parent, QString detailedText = {});
00055
00056 } // namespace quentier::utility

```

6.101 PlatformUtils.h

```

00001 /*
00002  * Copyright 2020-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QString>
00024 #include <QUrl>
00025
00026 namespace quentier::utility {
00027
00028 [[nodiscard]] QString QUENTIER_EXPORT getCurrentUserName();
00029
00030 [[nodiscard]] QString QUENTIER_EXPORT getCurrentUserFullName();
00031
00032 void QUENTIER_EXPORT openUrl(const QUrl & url);
00033
00034 } // namespace quentier::utility

```

6.102 Printable.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QDebug>
00024 #include <QHash>
00025 #include <QIODevice>
00026 #include <QSet>
00027 #include <QString>
00028 #include <QTextStream>
00029
00030 namespace quentier::utility {
00031
00032     class QUINTIER_EXPORT Printable
00033     {
00034     public:
00035         virtual ~Printable() noexcept;
00036
00037         virtual QTextStream & print(QTextStream & strm) const = 0;
00038
00039         [[nodiscard]] QString toString() const;
00040
00041         friend QUINTIER_EXPORT QTextStream & operator<<(
00042             QTextStream & strm, const Printable & printable);
00043
00044         friend QUINTIER_EXPORT QDebug & operator<<(
00045             QDebug & debug, const Printable & printable);
00046     };
00047
00048 } // namespace quentier::utility
00049
00050 // printing operators for existing classes not inheriting from Printable
00051
00052 template <class T>
00053 [[nodiscard]] QString ToString(const T & object)
00054 {
00055     QString str;
00056     QTextStream strm(&str, QIODevice::WriteOnly);
00057     strm << object;
00058     return str;
00059 }
00060
00061 template <class TKey, class TValue>
00062 [[nodiscard]] QString ToString(const QHash<TKey, TValue> & object)
00063 {
00064     QString str;
00065     QTextStream strm(&str, QIODevice::WriteOnly);
00066     strm << QStringLiteral("QHash: \n");
00067
00068     using CIter = typename QHash<TKey, TValue>::const_iterator;
00069     CIter hashEnd = object.end();
00070     for (CIter it = object.begin(); it != hashEnd; ++it) {
00071         strm << QStringLiteral("[") << it.key() << QStringLiteral(" = ")
00072             << it.value() << QStringLiteral("; \n");
00073     }
00074     return str;
00075 }
00076
00077 template <class T>
00078 [[nodiscard]] QString ToString(const QSet<T> & object)
00079 {
00080     QString str;
00081     QTextStream strm(&str, QIODevice::WriteOnly);
00082     strm << QStringLiteral("QSet: \n");
00083
00084     using CIter = typename QSet<T>::const_iterator;

```

```

00090     CIter setEnd = object.end();
00091     for (CIter it = object.begin(); it != setEnd; ++it) {
00092         strm « QStringLiteral("[") « *it « QStringLiteral("];\n");
00093     }
00094     return str;
00095 }
00096
00097 #define QUENTIER_DECLARE_PRINTABLE(type, ...)
00098     QUENTIER_EXPORT QTextStream & operator<(
00099         QTextStream & strm, const type & obj);
00100     inline QDebug & operator<(QDebug & debug, const type & obj)
00101     {
00102         debug « ToString<type, #__VA_ARGS__>(obj);
00103         return debug;
00104     }
00105     // QUENTIER_DECLARE_PRINTABLE

```

6.103 QuentierApplication.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QApplication>
00024
00025 namespace quentier::utility {
00026
00027 class QUENTIER_EXPORT QuentierApplication : public QApplication
00028 {
00029     Q_OBJECT
00030 public:
00031     QuentierApplication(int & argc, char * argv[]); // NOLINT
00032     ~QuentierApplication() noexcept override;
00033
00034     [[nodiscard]] bool notify(QObject * object, QEvent * event) override;
00035     [[nodiscard]] bool event(QEvent * event) override;
00036 };
00037
00038 } // namespace quentier::utility

```

6.104 QuentierUndoCommand.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018

```



```

00019 #pragma once
00020
00021 #include <quentier/types/ErrorMessage.h>
00022
00023 #include <QObject>
00024 #include <QUndoCommand>
00025
00026 namespace quentier::utility {
00027
00053 class QuentierUndoCommand : public QObject, public QUndoCommand
00054 {
00055     Q_OBJECT
00056 public:
00057     QuentierUndoCommand(QUndoCommand * parent = nullptr);
00058     QuentierUndoCommand(const QString & text, QUndoCommand * parent = nullptr);
00059     ~QuentierUndoCommand() noexcept override;
00060
00061     void undo() final;
00062     void redo() final;
00063
00064     [[nodiscard]] bool onceUndoExecuted() const noexcept
00065     {
00066         return m_onceUndoExecuted;
00067     }
00068
00069     Q_SIGNALS:
00070         void notifyError(ErrorMessage error);
00071
00072 protected:
00073     virtual void undoImpl() = 0;
00074     virtual void redoImpl() = 0;
00075
00076 private:
00077     bool m_onceUndoExecuted = false;
00078 };
00079
00080 } // namespace quentier::utility

```

6.105 ShortcutManager.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/Account.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QKeySequence>
00025 #include <QObject>
00026
00027 namespace quentier::utility {
00028
00029 class QUENTIER_EXPORT ShortcutManager : public QObject
00030 {
00031     Q_OBJECT
00032 public:
00033     explicit ShortcutManager(QObject * parent = nullptr);
00034
00035     enum QuentierShortcutKey
00036     {
00037         NewNote = 5000,
00038         NewTag,
00039         NewNotebook,
00040         NewSavedSearch,
00041         AddAttachment,
00042         SaveAttachment,
00043         OpenAttachment,

```

```

00044     CopyAttachment,
00045     CutAttachment,
00046     RemoveAttachment,
00047     RenameAttachment,
00048     AddAccount,
00049     ExitAccount,
00050     SwitchAccount,
00051     AccountInfo,
00052     NoteSearch,
00053     NewNoteSearch,
00054     ShowNotes,
00055     ShowNotebooks,
00056     ShowTags,
00057     ShowSavedSearches,
00058     ShowDeletedNotes,
00059     ShowStatusBar,
00060     ShowToolBar,
00061     PasteUnformatted,
00062     Font,
00063     UpperIndex,
00064     LowerIndex,
00065     AlignLeft,
00066     AlignCenter,
00067     AlignRight,
00068     AlignFull,
00069     IncreaseIndentation,
00070     DecreaseIndentation,
00071     IncreaseFontSize,
00072     DecreaseFontSize,
00073     InsertNumberedList,
00074     InsertBulletedList,
00075     Strikethrough,
00076     Highlight,
00077     InsertTable,
00078     InsertRow,
00079     InsertColumn,
00080     RemoveRow,
00081     RemoveColumn,
00082     InsertHorizontalLine,
00083     InsertToDoTag,
00084     EditHyperlink,
00085     CopyHyperlink,
00086     RemoveHyperlink,
00087     Encrypt,
00088     Decrypt,
00089     DecryptPermanently,
00090     BackupLocalStorage,
00091     RestoreLocalStorage,
00092     UpgradeLocalStorage,
00093     LocalStorageStatus,
00094     SpellCheck,
00095     SpellCheckIgnoreWord,
00096     SpellCheckAddWordToUserDictionary,
00097     SaveImage,
00098     AnnotateImage,
00099     ImageRotateClockwise,
00100     ImageRotateCounterClockwise,
00101     Synchronize,
00102     FullSync,
00103     ImportFolders,
00104     Preferences,
00105     ReleaseNotes,
00106     ViewLogs,
00107     About,
00108     UnknownKey = 100000
00109 };
00110
00116 [[nodiscard]] QKeySequence shortcut(
00117     int key, const Account & account, const QString & context = {}) const;
00118
00124 [[nodiscard]] QKeySequence shortcut(
00125     const QString & nonStandardKey, const Account & account,
00126     const QString & context = {}) const;
00127
00132 [[nodiscard]] QKeySequence defaultShortcut(
00133     int key, const Account & account, const QString & context = {}) const;
00134
00139 [[nodiscard]] QKeySequence defaultShortcut(
00140     const QString & nonStandardKey, const Account & account,
00141     const QString & context = {}) const;
00142
00147 [[nodiscard]] QKeySequence userShortcut(
00148     int key, const Account & account, const QString & context = {}) const;
00149
00154 [[nodiscard]] QKeySequence userShortcut(
00155     const QString & nonStandardKey, const Account & account,
00156     const QString & context = {}) const;

```

```

00157
00158 Q_SIGNALS:
00159     void shortcutChanged(
00160         int key, QKeySequence shortcut, const Account & account,
00161         QString context);
00162
00163     void nonStandardShortcutChanged(
00164         QString nonStandardKey, QKeySequence shortcut, const Account & account,
00165         QString context);
00166
00167 public Q_SLOTS:
00168     void setUserShortcut(
00169         int key, const QKeySequence & shortcut, const Account & account,
00170         QString context = {});
00171
00172     void setNonStandardUserShortcut(
00173         QString nonStandardKey, const QKeySequence & shortcut,
00174         const Account & account, QString context = {});
00175
00176     void setDefaultShortcut(
00177         int key, const QKeySequence & shortcut, const Account & account,
00178         QString context = {});
00179
00180     void setNonStandardDefaultShortcut(
00181         QString nonStandardKey, const QKeySequence & shortcut,
00182         const Account & account, QString context = {});
00183
00184 private:
00185     class ShortcutManagerPrivate;
00186
00187     ShortcutManagerPrivate * const d_ptr;
00188     Q_DECLARE_PRIVATE(ShortcutManager)
00189 };
00190
00191 } // namespace quantier::utility

```

6.106 Size.h

```

00001 /*
00002  * Copyright 2020-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quantier/utility/Linkage.h>
00022
00023 #include <QString>
00024
00025 namespace quantier::utility {
00026
00036 [[nodiscard]] QString QUENTIER_EXPORT humanReadableSize(quint64 bytes);
00037
00038 } // namespace quantier::utility

```

6.107 StandardPaths.h

```

00001 /*
00002  * Copyright 2017-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.

```

```

00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/Account.h>
00022 #include <quentier/utility/Linkage.h>
00023
00028 #define LIBQUENTIER_PERSISTENCE_STORAGE_PATH \
00029     "LIBQUENTIER_PERSISTENCE_STORAGE_PATH"
00030
00031 namespace quentier::utility {
00032
00033 constexpr const char * gLibquentierPersistenceStoragePath =
00034     "LIBQUENTIER_PERSISTENCE_STORAGE_PATH";
00035
00044 [[nodiscard]] QString QUENTIER_EXPORT
00045     applicationPersistentStoragePath(bool * nonStandardLocation = nullptr);
00046
00056 [[nodiscard]] QString QUENTIER_EXPORT
00057     accountPersistentStoragePath(const Account & account);
00058
00063 [[nodiscard]] QString QUENTIER_EXPORT applicationTemporaryStoragePath();
00064
00070 [[nodiscard]] QString QUENTIER_EXPORT homePath();
00071
00075 [[nodiscard]] QString QUENTIER_EXPORT documentsPath();
00076
00077 } // namespace quentier::utility

```

6.108 StringUtils.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QList>
00024 #include <QSet>
00025 #include <QString>
00026
00027 namespace quentier::utility {
00028
00029 class QUENTIER_EXPORT StringUtils
00030 {
00031 public:
00032     StringUtils();
00033     ~StringUtils() noexcept;
00034
00035     void removePunctuation(
00036         QString & str, const QList<QChar> & charactersToPreserve = {}) const;
00037
00038     void removeDiacritics(QString & str) const;
00039     void removeNewlines(QString & str) const;
00040
00041 private:
00042     class StringUtilsPrivate;
00043     StringUtilsPrivate * const d_ptr;
00044     Q_DECLARE_PRIVATE(StringUtils);

```

```
00045 };
00046
00047 } // namespace quantier::utility
```

6.109 SuppressWarnings.h

```
00001 /*
00002  * Copyright 2020-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00022 // Common macros
00024
00025 #define STRINGIFY(a) #a
00026
00027 // Define empty macros doing nothing for supported compilers, they would be used
00028 // as fallback when any of these compilers are not actually used
00029
00030 #define SAVE_WARNINGS
00031
00032 #define CLANG_SUPPRESS_WARNING(warning)
00033 #define GCC_SUPPRESS_WARNING(warning)
00034 #define MSVC_SUPPRESS_WARNING(warning)
00035
00036 #define RESTORE_WARNINGS
00037
00039 // Clang implementation
00041
00042 #if defined(__clang__)
00043
00044 #undef CLANG_SUPPRESS_WARNING
00045
00046 #define CLANG_SUPPRESS_WARNING(warning) \
00047     _Pragma(Stringify(clang diagnostic ignored #warning))
00048
00049 #undef SAVE_WARNINGS
00050
00051 #define SAVE_WARNINGS _Pragma("clang diagnostic push")
00052
00053 #undef RESTORE_WARNINGS
00054
00055 #define RESTORE_WARNINGS _Pragma("clang diagnostic pop")
00056
00057 #endif // clang
00058
00060 // GCC implementation
00062
00063 // Clang can mimic gcc so need to ensure it's indeed gcc
00064 #if defined(__GNUC__) && !defined(__clang__)
00065
00066 #undef GCC_SUPPRESS_WARNING
00067
00068 #define GCC_SUPPRESS_WARNING(warning) \
00069     _Pragma(Stringify(GCC diagnostic ignored #warning))
00070
00071 #undef SAVE_WARNINGS
00072
00073 #define SAVE_WARNINGS _Pragma("GCC diagnostic push")
00074
00075 #undef RESTORE_WARNINGS
00076
00077 #define RESTORE_WARNINGS _Pragma("GCC diagnostic pop")
00078
00079 #endif // GCC
00080
00082 // MSVC implementation
00084
00085 #if defined(_MSC_VER)
```

```

00086
00087 #undef MSVC_SUPPRESS_WARNING
00088
00089 #define MSVC_SUPPRESS_WARNING(number) __pragma(warning(disable : number))
00090
00091 #undef SAVE_WARNINGS
00092
00093 #define SAVE_WARNINGS __pragma(warning(push))
00094
00095 #undef RESTORE_WARNINGS
00096
00097 #define RESTORE_WARNINGS __pragma(warning(pop))
00098
00099 #endif // MSVC

```

6.110 SysInfo.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QString>
00024
00025 namespace quentier::utility {
00026
00027 class QUINTIER_EXPORT SysInfo
00028 {
00029 public:
00030     SysInfo();
00031     ~SysInfo() noexcept;
00032
00033     [[nodiscard]] quint64 pageSize();
00034     [[nodiscard]] quint64 totalMemory();
00035     [[nodiscard]] quint64 freeMemory();
00036
00037     [[nodiscard]] QString stackTrace();
00038
00039     [[nodiscard]] QString platformName();
00040
00041 private:
00042     Q_DISABLE_COPY(SysInfo)
00043
00044 private:
00045     class SysInfoPrivate;
00046     SysInfoPrivate * const d_ptr;
00047     Q_DECLARE_PRIVATE(SysInfo)
00048 };
00049
00050 } // namespace quentier::utility

```

6.111 TagSortByParentChildRelations.h

```

00001 /*
00002  * Copyright 2017-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.

```

```

00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/types/Fwd.h>
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <qevercloud/types/Fwd.h>
00025
00026 #include <QList>
00027
00028 namespace quentier::utility {
00029
00041 bool QUENTIER_EXPORT sortTagsByParentChildRelations(
00042     QList<qevercloud::Tag> & tagList, ErrorString & errorDescription);
00043
00044 } // namespace quentier::utility

```

6.112 MockIKeychainService.h

```

00001 /*
00002  * Copyright 2022-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #include <quentier/utility/IKeychainService.h>
00020
00021 #include <gmock/gmock.h>
00022
00023 namespace quentier::utility::tests::mocks {
00024
00025 class MockIKeychainService : public IKeychainService
00026 {
00027 public:
00028     MOCK_METHOD(
00029         QFuture<void>, writePassword,
00030         (QString service, QString key, QString password), (override));
00031
00032     MOCK_METHOD(
00033         QFuture<QString>, readPassword, (QString service, QString key),
00034         (const, override));
00035
00036     MOCK_METHOD(
00037         QFuture<void>, deletePassword, (QString service, QString key),
00038         (override));
00039 };
00040
00041 } // namespace quentier::utility::tests::mocks

```

6.113 UidGenerator.h

```

00001 /*
00002  * Copyright 2016-2025 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify

```

```

00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <quentier/utility/Linkage.h>
00022
00023 #include <QString>
00024 #include <QUuid>
00025
00026 namespace quentier::utility {
00027
00028 class QUENTIER_EXPORT UidGenerator
00029 {
00030 public:
00031     [[nodiscard]] static QString generate();
00032     [[nodiscard]] static QString uidToString(const QUuid & uid);
00033 };
00034
00035 } // namespace quentier::utility

```

6.114 Unreachable.h

```

00001 /*
00002  * Copyright 2022-2024 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #pragma once
00020
00021 #include <QtGlobal>
00022
00023 #ifdef _MSC_VER
00024 #define UNREACHABLE
00025     do {
00026         Q_ASSERT(false);
00027         __assume(0);
00028     } while (false)
00029 #else
00030 #define UNREACHABLE
00031     do {
00032         Q_ASSERT(false);
00033         __builtin_unreachable();
00034     } while (false)
00035 #endif

```


Index

- ~ApplicationSettings
 - quentier::utility::ApplicationSettings, [23](#)
- AccessDenied
 - quentier::utility::IKeychainService, [65](#)
- AccessDeniedByUser
 - quentier::utility::IKeychainService, [65](#)
- Account.h, [242](#)
- addedLinkedNotebooks
 - quentier::synchronization::ISyncChunksDataCountersCipher
 - [92](#)
- addedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters,
 - [92](#)
- addedSavedSearches
 - quentier::synchronization::ISyncChunksDataCounters,
 - [92](#)
- addedTags
 - quentier::synchronization::ISyncChunksDataCounters,
 - [92](#)
- AES
 - quentier::utility::IEncryptor, [59](#)
- Affiliation
 - quentier::local_storage::ILocalStorage, [71](#)
- Any
 - quentier::local_storage::ILocalStorage, [72](#)
- AnyOfCanceler.h, [258](#)
- ApplicationSettings
 - quentier::utility::ApplicationSettings, [21](#), [22](#)
- ApplicationSettings.h, [257](#)
- apply
 - quentier::local_storage::IPatch, [79](#)
- AttributeName
 - quentier::enml::conversion_rules::ISkipRule, [89](#)
- AttributeValue
 - quentier::enml::conversion_rules::ISkipRule, [89](#)
- AuthenticationInfo.h, [223](#)
- authenticationTime
 - quentier::synchronization::IAuthenticationInfo, [46](#)
- authToken
 - quentier::synchronization::IAuthenticationInfo, [46](#)
- authTokenExpirationTime
 - quentier::synchronization::IAuthenticationInfo, [46](#)
- backend
 - quentier::NoteEditor, [141](#)
- backupLocalStorage
 - quentier::local_storage::IPatch, [79](#)
- beginGroup
 - quentier::utility::ApplicationSettings, [23](#)
- beginReadArray
 - quentier::utility::ApplicationSettings, [24](#)
- beginWriteArray
 - quentier::utility::ApplicationSettings, [25](#)
- cancel
 - quentier::utility::cancelers::ManualCanceler, [124](#)
- caseSensitivity
 - quentier::enml::conversion_rules::ISkipRule, [89](#)
- clear
 - quentier::NoteEditor, [141](#)
- Compat.h, [260](#)
- const_pointer
 - quentier::utility::LRUCache< Key, Value, Allocator>, [123](#)
- contains
 - quentier::utility::ApplicationSettings, [26](#)
- convertDecryptedText
 - quentier::enml::IENMLTagsConverter, [60](#)
- convertEncryptedText
 - quentier::enml::IENMLTagsConverter, [61](#)
- convertEnmlToHtml
 - quentier::enml::IConverter, [50](#)
- convertEnmlToPlainText
 - quentier::enml::IConverter, [50](#)
- convertEnmlToWordsList
 - quentier::enml::IConverter, [51](#)
- convertEnToDo
 - quentier::enml::IENMLTagsConverter, [61](#)
- convertHtmlToDoc
 - quentier::enml::IConverter, [51](#)
- convertHtmlToEnml
 - quentier::enml::IConverter, [51](#)
- convertHtmlToXhtml
 - quentier::enml::IConverter, [52](#)
- convertHtmlToXml
 - quentier::enml::IConverter, [52](#)
- convertPlainTextToWordsList
 - quentier::enml::IConverter, [52](#)
- convertResource
 - quentier::enml::IENMLTagsConverter, [61](#)
- convertToNote
 - quentier::NoteEditor, [141](#)
- CouldNotDeleteEntry
 - quentier::utility::IKeychainService, [65](#)
- currentNoteLocalId
 - quentier::NoteEditor, [141](#)

- DateTime.h, 261
- decrypt
 - quentier::utility::IEncryptor, 59
- defaultFont
 - quentier::NoteEditor, 141
- defaultPalette
 - quentier::NoteEditor, 141
- defaultShortcut
 - quentier::utility::ShortcutManager, 165
- deletePassword
 - quentier::utility::IKeychainService, 66
- displayName
 - quentier::Account, 17
- downloadFinished
 - quentier::synchronization::ISyncEventsNotifier, 98
- DownloadNotesStatus.h, 223
- downloadNoteThumbnails
 - quentier::synchronization::ISyncOptions, 104
- DownloadResourcesStatus.h, 224
- Element
 - quentier::enml::conversion_rules::ISkipRule, 89
- encrypt
 - quentier::utility::IEncryptor, 59
- EntryNotFound
 - quentier::utility::IKeychainService, 65
- ErrorCode
 - quentier::utility::IKeychainService, 65
- Errors.h, 213
- ErrorString.h, 244
- EventLoopWithExitStatus.h, 262
- evernoteAccountType
 - quentier::Account, 17
- evernoteHost
 - quentier::Account, 17
- exceptionDisplayName
 - quentier::InvalidArgument, 78
 - quentier::local_storage::LocalStorageOpenException, 120
 - quentier::local_storage::LocalStorageOperationException, 122
 - quentier::OperationCanceled, 151
 - quentier::RuntimeError, 163
 - quentier::utility::IKeychainService::Exception, 37
- exportNotesToEnex
 - quentier::enml::IConverter, 53
- expungedLinkedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters, 93
- expungedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters, 93
- expungedSavedSearches
 - quentier::synchronization::ISyncChunksDataCounters, 93
- expungedTags
 - quentier::synchronization::ISyncChunksDataCounters, 93
- Factory.h, 177–179
- failedToSendNotebooks
 - quentier::synchronization::ISendStatus, 85
- failedToSendNotes
 - quentier::synchronization::ISendStatus, 85
- failedToSendSavedSearches
 - quentier::synchronization::ISendStatus, 85
- failedToSendTags
 - quentier::synchronization::ISendStatus, 85
- FileIOProcessorAsync.h, 262
- FileSystem.h, 263
- FileSystemWatcher.h, 264
- fromVersion
 - quentier::local_storage::IPatch, 79
- Future.h, 226
- FutureCanceler.h, 259
- Fwd.h, 245–249
- html
 - quentier::enml::IHtmlData, 63
- HtmlUtils.h, 171
- IAuthenticationInfo.h, 214
- IAuthenticationInfoBuilder.h, 214
- IAuthenticator.h, 205
- ICanceler.h, 259
- IConverter.h, 171
- id
 - quentier::Account, 17
- IDecryptedTextCache.h, 172
- idleTime
 - quentier::NoteEditor, 141
- IDownloadNotesStatus.h, 215
- IDownloadResourcesStatus.h, 216
- IEncryptor.h, 265
- IENMLTagsConverter.h, 173
- IHtmlData.h, 174
- IKeychainService.h, 265
- ILocalStorage.h, 180
- ILocalStorageNotifier.h, 188
- importEnex
 - quentier::enml::IConverter, 53
- inAppNoteLinkPasteRequested
 - quentier::NoteEditor, 142
- includeContents
 - quentier::enml::conversion_rules::ISkipRule, 89
- initialize
 - quentier::NoteEditor, 142
- Initialize.h, 266
- inkNoteImagesStorageDir
 - quentier::synchronization::ISyncOptions, 104
- INoteEditorBackend.h, 198
- INoteStoreFactory.h, 206
- InvalidArgument.h, 175
- IPatch.h, 189
- IQuentierException.h, 175
- isCanceled
 - quentier::utility::cancelers::AnyOfCanceler, 19

- quenter::utility::cancelers::FutureCanceler< T >, 43
 - quenter::utility::cancelers::ManualCanceler, 124
- isEditorPageModified
 - quenter::NoteEditor, 142
- isEmpty
 - quenter::Account, 17
- ISendStatus.h, 217
- ISkipRule.h, 169
- ISkipRuleBuilder.h, 170
- isModified
 - quenter::NoteEditor, 143
- isNoteLoaded
 - quenter::NoteEditor, 143
- isValid
 - quenter::Result< ValueType, ErrorType, typename >, 160
- ISyncChunksDataCounters.h, 218
- ISyncConflictResolver.h, 206
- ISyncEventsNotifier.h, 208
- ISynchronizer.h, 209
- ISyncOptions.h, 219
- ISyncOptionsBuilder.h, 220
- ISyncResult.h, 221
- ISyncState.h, 222
- ISyncStateBuilder.h, 222
- ISyncStateStorage.h, 210
- IUserStoreFactory.h, 210
- libquenter, 1
- linkedNotebookNotesDownloadProgress
 - quenter::synchronization::ISyncEventsNotifier, 98
- linkedNotebookResourcesDownloadProgress
 - quenter::synchronization::ISyncEventsNotifier, 99
- linkedNotebookSendStatusUpdate
 - quenter::synchronization::ISyncEventsNotifier, 99
- linkedNotebookSyncChunksDataProcessingProgress
 - quenter::synchronization::ISyncEventsNotifier, 99
- linkedNotebookSyncChunksDownloaded
 - quenter::synchronization::ISyncEventsNotifier, 100
- linkedNotebookSyncChunksDownloadProgress
 - quenter::synchronization::ISyncEventsNotifier, 100
- LocalStorageOpenException.h, 190
- LocalStorageOperationException.h, 190
- LRUCache.hpp, 267
- ManualCanceler.h, 260
- matchMode
 - quenter::enml::conversion_rules::ISkipRule, 89
- MatchMode.h, 170
- maxConcurrentNoteDownloads
 - quenter::synchronization::ISyncOptions, 104
- maxConcurrentResourceDownloads
 - quenter::synchronization::ISyncOptions, 104
- MessageBox.h, 269
- mine
 - quenter::synchronization::ISyncConflictResolver::ConflictResolution: T >, 137
- MockIAuthenticator.h, 211
- MockIKeychainService.h, 279
- MockILocalStorage.h, 193
- MockINoteStoreFactory.h, 211
- MockISyncConflictResolver.h, 212
- MockISyncStateStorage.h, 212
- name
 - quenter::Account, 17
- needToRepeatIncrementalSync
 - quenter::synchronization::ISendStatus, 85
- NoBackendAvailable
 - quenter::utility::IKeychainService, 65
- NoError
 - quenter::utility::IKeychainService, 65
- NotebookConflictResolution
 - quenter::synchronization::ISyncConflictResolver, 96
- notebookModifier
 - quenter::local_storage::NoteSearchQuery, 149
- NotebookWithException
 - quenter::synchronization::ISendStatus, 84
- NoteConflictResolution
 - quenter::synchronization::ISyncConflictResolver, 96
- NoteEditor.h, 201
- notesDownloadProgress
 - quenter::synchronization::ISyncEventsNotifier, 100
- NoteSearchQuery.h, 191
- noteStoreUrl
 - quenter::synchronization::IAuthenticationInfo, 46
- NoteUtils.h, 250
- notifier
 - quenter::local_storage::ILocalStorage, 72
- notifySyncStateUpdated
 - quenter::synchronization::ISyncStateStorage, 109
- NotImplemented
 - quenter::utility::IKeychainService, 65
- numEncryptNodes
 - quenter::enml::IHtmlData, 63
- numDecryptNodes
 - quenter::enml::IHtmlData, 63
- numEnToDoNodes
 - quenter::enml::IHtmlData, 64
- numHyperlinkNodes
 - quenter::enml::IHtmlData, 64
- onReadFileRequest
 - quenter::utility::FileIOProcessorAsync, 39
- onWriteFileRequest
 - quenter::utility::FileIOProcessorAsync, 39
- OperationCanceled.h, 176
- OtherError
 - quenter::utility::IKeychainService, 66
- patchLongDescription

- quantier::local_storage::IPatch, 80
- patchShortDescription
 - quantier::local_storage::IPatch, 80
- PlatformUtils.h, 270
- Post.h, 230
- print
 - quantier::Account, 18
 - quantier::enml::conversion_rules::ISkipRule, 90
 - quantier::enml::IHtmlData, 64
 - quantier::ErrorString, 34
 - quantier::IQuantierException, 82
 - quantier::local_storage::NoteSearchQuery, 149
 - quantier::ResourceRecognitionIndexItem, 158
 - quantier::ResourceRecognitionIndices, 159
 - quantier::utility::ApplicationSettings, 27
- Printable.h, 271
- QPromise< T >, 153
- Qt5Promise.h, 231
- QtFutureContinuations.h, 232
- QtFutureHelpers.h, 238
- quantier::Account, 15
 - displayName, 17
 - evernoteAccountType, 17
 - evernoteHost, 17
 - id, 17
 - isEmpty, 17
 - name, 17
 - print, 18
 - setDisplayName, 18
 - shardId, 18
 - type, 18
- quantier::enml::conversion_rules::ISkipRule, 88
 - AttributeName, 89
 - AttributeValue, 89
 - caseSensitivity, 89
 - Element, 89
 - includeContents, 89
 - matchMode, 89
 - print, 90
 - Target, 89
 - target, 90
 - value, 90
- quantier::enml::conversion_rules::ISkipRuleBuilder, 90
- quantier::enml::IConverter, 49
 - convertEnmlToHtml, 50
 - convertEnmlToPlainText, 50
 - convertEnmlToWordsList, 51
 - convertHtmlToDoc, 51
 - convertHtmlToEnml, 51
 - convertHtmlToXhtml, 52
 - convertHtmlToXml, 52
 - convertPlainTextToWordsList, 52
 - exportNotesToEnex, 53
 - importEnex, 53
 - validateAndFixupEnml, 54
 - validateEnml, 54
- quantier::enml::IDecryptedTextCache, 54
- quantier::enml::IENMLTagsConverter, 60
 - convertDecryptedText, 60
 - convertEncryptedText, 61
 - convertEnToDo, 61
 - convertResource, 61
- quantier::enml::IHtmlData, 62
 - html, 63
 - numEncryptNodes, 63
 - numDecryptNodes, 63
 - numToDoNodes, 64
 - numHyperlinkNodes, 64
 - print, 64
- quantier::ErrorString, 32
 - print, 34
- quantier::INoteEditorBackend, 74
- quantier::InvalidArgument, 77
 - exceptionDisplayName, 78
- quantier::IQuantierException, 81
 - print, 82
- quantier::local_storage::ILocalStorage, 67
 - Affiliation, 71
 - Any, 72
 - notifier, 72
 - TagNotesRelation, 71
 - WithNotes, 72
 - WithoutNotes, 72
- quantier::local_storage::ILocalStorage::ListGuidsFilters, 110
- quantier::local_storage::ILocalStorage::ListLinkedNotebooksOptions, 110
- quantier::local_storage::ILocalStorage::ListNotebooksOptions, 112
- quantier::local_storage::ILocalStorage::ListNotesOptions, 113
- quantier::local_storage::ILocalStorage::ListObjectsFilters, 114
- quantier::local_storage::ILocalStorage::ListOptionsBase, 114
- quantier::local_storage::ILocalStorage::ListSavedSearchesOptions, 115
- quantier::local_storage::ILocalStorage::ListTagsOptions, 117
- quantier::local_storage::ILocalStorageNotifier, 72
- quantier::local_storage::IPatch, 79
 - apply, 79
 - backupLocalStorage, 79
 - fromVersion, 79
 - patchLongDescription, 80
 - patchShortDescription, 80
 - removeLocalStorageBackup, 80
 - restoreLocalStorageFromBackup, 80
 - toVersion, 80
- quantier::local_storage::LocalStorageOpenException, 118
 - exceptionDisplayName, 120
- quantier::local_storage::LocalStorageOperationException, 120
 - exceptionDisplayName, 122
- quantier::local_storage::NoteSearchQuery, 146

- notebookModifier, 149
- print, 149
- queryString, 149
- quentier::local_storage::tests::mocks::MockLocalStorage, 127
- quentier::NoteEditor, 137
 - backend, 141
 - clear, 141
 - convertToNote, 141
 - currentNoteLocalId, 141
 - defaultFont, 141
 - defaultPalette, 141
 - idleTime, 141
 - inAppNoteLinkPasteRequested, 142
 - initialize, 142
 - isEditorPageModified, 142
 - isModified, 143
 - isNoteLoaded, 143
 - saveNoteToLocalStorage, 143
 - setAccount, 143
 - setBackend, 143
 - setCurrentNoteLocalId, 143
 - setDefaultFont, 144
 - setDefaultPalette, 144
 - setFocus, 144
 - setInitialPageHtml, 145
 - setNoteDeletedPageHtml, 145
 - setNoteLoadingPageHtml, 145
 - setNoteNotFoundPageHtml, 145
 - setNoteTitle, 145
 - setTagIds, 146
 - setUndoStack, 146
 - undoStack, 146
- quentier::OperationCanceled, 150
 - exceptionDisplayName, 151
- quentier::ResourceRecognitionIndexItem, 156
 - print, 158
- quentier::ResourceRecognitionIndexItem::IBarcodeItem, 48
- quentier::ResourceRecognitionIndexItem::IObjectItem, 78
- quentier::ResourceRecognitionIndexItem::IShapeItem, 87
- quentier::ResourceRecognitionIndexItem::ITextItem, 110
- quentier::ResourceRecognitionIndices, 158
 - print, 159
- quentier::Result< ValueType, ErrorType, typename >, 160
 - isValid, 160
- quentier::RuntimeError, 161
 - exceptionDisplayName, 163
- quentier::SpellChecker, 166
- quentier::synchronization::AuthenticationExpiredError, 31
- quentier::synchronization::IAuthenticationInfo, 45
 - authenticationTime, 46
 - authToken, 46
 - authTokenExpirationTime, 46
 - noteStoreUrl, 46
 - shardId, 46
 - userId, 46
 - userStoreCookies, 47
 - webApiUrlPrefix, 47
- quentier::synchronization::IAuthenticationInfoBuilder, 47
- quentier::synchronization::IAuthenticator, 48
- quentier::synchronization::IDownloadNotesStatus, 55
- quentier::synchronization::IDownloadResourcesStatus, 57
 - ResourceWithException, 58
- quentier::synchronization::INoteStoreFactory, 76
- quentier::synchronization::ISendStatus, 83
 - failedToSendNotebooks, 85
 - failedToSendNotes, 85
 - failedToSendSavedSearches, 85
 - failedToSendTags, 85
 - needToRepeatIncrementalSync, 85
 - NotebookWithException, 84
 - SavedSearchWithException, 84
 - stopSynchronizationError, 85
 - totalAttemptedToSendNotebooks, 86
 - totalAttemptedToSendNotes, 86
 - totalAttemptedToSendSavedSearches, 86
 - totalAttemptedToSendTags, 86
 - totalSuccessfullySentNotebooks, 86
 - totalSuccessfullySentNotes, 87
 - totalSuccessfullySentSavedSearches, 87
 - totalSuccessfullySentTags, 87
- quentier::synchronization::ISyncChunksDataCounters, 91
 - addedLinkedNotebooks, 92
 - addedNotebooks, 92
 - addedSavedSearches, 92
 - addedTags, 92
 - expungedLinkedNotebooks, 93
 - expungedNotebooks, 93
 - expungedSavedSearches, 93
 - expungedTags, 93
 - totalExpungedLinkedNotebooks, 93
 - totalExpungedNotebooks, 93
 - totalExpungedSavedSearches, 93
 - totalExpungedTags, 94
 - totalLinkedNotebooks, 94
 - totalNotebooks, 94
 - totalSavedSearches, 94
 - totalTags, 94
 - updatedLinkedNotebooks, 94
 - updatedNotebooks, 94
 - updatedSavedSearches, 95
 - updatedTags, 95
- quentier::synchronization::ISyncConflictResolver, 95
 - NotebookConflictResolution, 96
 - NoteConflictResolution, 96
 - SavedSearchConflictResolution, 96
 - TagConflictResolution, 96

quantier::synchronization::ISyncConflictResolver::ConflictResolution, std::decay_t< F >, QFuture< Arg > > > >, 32
 quantier::synchronization::ISyncConflictResolver::ConflictResolution, std::decay_t< F >, QFuture< Arg > > > >, 62
 quantier::synchronization::ISyncConflictResolver::ConflictResolution, std::decay_t< F >, QFuture< Arg > > > >, T >, 136
 mine, 137
 quantier::synchronization::ISyncConflictResolver::ConflictResolution, type Mine, std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< void > > > >, std::decay_t< F >, QFuture< void > > > >, 168
 quantier::synchronization::ISyncConflictResolver::ConflictResolution, type Mine, std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< void > > > >, std::decay_t< F >, QFuture< void > > > >, 168
 quantier::synchronization::ISyncEventsNotifier, 97
 downloadFinished, 98
 linkedNotebookNotesDownloadProgress, 98
 linkedNotebookResourcesDownloadProgress, 99
 linkedNotebookSendStatusUpdate, 99
 linkedNotebookSyncChunksDataProcessingProgress, 99
 linkedNotebookSyncChunksDownloaded, 100
 linkedNotebookSyncChunksDownloadProgress, 100
 notesDownloadProgress, 100
 resourcesDownloadProgress, 101
 startLinkedNotebooksDataDownloading, 101
 syncChunksDataProcessingProgress, 101
 syncChunksDownloaded, 101
 syncChunksDownloadProgress, 101
 userOwnSendStatusUpdate, 102
 quantier::synchronization::ISynchronizer, 102
 SyncResult, 103
 quantier::synchronization::ISyncOptions, 103
 downloadNoteThumbnails, 104
 inkNoteImagesStorageDir, 104
 maxConcurrentNoteDownloads, 104
 maxConcurrentResourceDownloads, 104
 requestContext, 105
 retryPolicy, 105
 quantier::synchronization::ISyncOptionsBuilder, 105
 quantier::synchronization::ISyncResult, 106
 quantier::synchronization::ISyncState, 107
 quantier::synchronization::ISyncStateBuilder, 108
 quantier::synchronization::ISyncStateStorage, 108
 notifySyncStateUpdated, 109
 quantier::synchronization::IUserStoreFactory, 110
 quantier::synchronization::RateLimitReachedError, 156
 rateLimitDurationSec, 156
 quantier::synchronization::tests::mocks::MockIAuthenticator, 124
 quantier::synchronization::tests::mocks::MockINoteStoreFactory, 132
 quantier::synchronization::tests::mocks::MockISyncConflictResolver, 133
 quantier::synchronization::tests::mocks::MockISyncStateStorage, 135
 quantier::threading::detail::ResultTypeHelper< F, Arg, Enable >, 160
 quantier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< void > > > >, std::decay_t< F >, QFuture< void > > > >, 161
 quantier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< void > > > >, std::decay_t< F >, QFuture< void > > > >, 161
 quantier::threading::TrackedTask< LockableObject, Function >, 167
 quantier::utility::ApplicationSettings, 20
 ~ApplicationSettings, 23
 ApplicationSettings, 21, 22
 beginGroup, 23
 beginReadArray, 24
 beginWriteArray, 25
 contains, 26
 print, 27
 remove, 27, 28
 setValue, 28, 29
 value, 29, 30
 quantier::utility::ApplicationSettings::ArrayCloser, 30
 quantier::utility::ApplicationSettings::GroupCloser, 44
 quantier::utility::cancelers::AnyOfCanceler, 19
 isCanceled, 19
 quantier::utility::cancelers::FutureCanceler< T >, 42
 isCanceled, 43
 quantier::utility::cancelers::ICanceler, 48
 quantier::utility::cancelers::ManualCanceler, 123
 cancel, 124
 isCanceled, 124
 quantier::utility::EventLoopWithExitStatus, 34
 quantier::utility::FileIOProcessorAsync, 38
 onReadFileRequest, 39
 onWriteFileRequest, 39
 readFileRequestProcessed, 40
 setIdleTimePeriod, 40
 writeFileRequestProcessed, 41
 quantier::utility::FileSystemWatcher, 41
 quantier::utility::IEncryptor, 58
 AES, 59
 Cipher, 59
 decrypt, 59
 encrypt, 59
 quantier::utility::IKeychainService, 64
 AccessDenied, 65
 AccessDeniedByUser, 65
 CouldNotDeleteEntry, 65
 deletePassword, 66
 EntryNotFound, 65
 ErrorCode, 65

- NoBackendAvailable, 65
- NoError, 65
- NotImplemented, 65
- OtherError, 66
- readPassword, 66
- writePassword, 66
- quentier::utility::IKeychainService::Exception, 35
 - exceptionDisplayName, 37
- quentier::utility::LRUCache< Key, Value, Allocator >, 122
 - const_pointer, 123
- quentier::utility::Printable, 151
- quentier::utility::QuentierApplication, 153
- quentier::utility::QuentierUndoCommand, 154
- quentier::utility::ShortcutManager, 163
 - defaultShortcut, 165
 - shortcut, 165
 - userShortcut, 165, 166
- quentier::utility::StringUtils, 167
- quentier::utility::SysInfo, 167
- quentier::utility::tests::mocks::MockIKeychainService, 125
- quentier::utility::UidGenerator, 168
- QuentierApplication.h, 272
- QuentierLogger.h, 197
- QuentierUndoCommand.h, 272
- queryString
 - quentier::local_storage::NoteSearchQuery, 149
- rateLimitDurationSec
 - quentier::synchronization::RateLimitReachedError, 156
- RC2
 - quentier::utility::IEncryptor, 59
- readFileRequestProcessed
 - quentier::utility::FileIOProcessorAsync, 40
- readPassword
 - quentier::utility::IKeychainService, 66
- RegisterMetatypes.h, 250
- remove
 - quentier::utility::ApplicationSettings, 27, 28
- removeLocalStorageBackup
 - quentier::local_storage::IPatch, 80
- requestContext
 - quentier::synchronization::ISyncOptions, 105
- ResourceRecognitionIndexItem.h, 251
- ResourceRecognitionIndices.h, 252
- resourcesDownloadProgress
 - quentier::synchronization::ISyncEventsNotifier, 101
- ResourceUtils.h, 253
- ResourceWithException
 - quentier::synchronization::IDownloadResourcesStatus, 58
- restoreLocalStorageFromBackup
 - quentier::local_storage::IPatch, 80
- Result.h, 254
- retryPolicy
 - quentier::synchronization::ISyncOptions, 105
- Runnable.h, 241
- RuntimeError.h, 176
- SavedSearchConflictResolution
 - quentier::synchronization::ISyncConflictResolver, 96
- SavedSearchWithException
 - quentier::synchronization::ISendStatus, 84
- saveNoteToLocalStorage
 - quentier::NoteEditor, 143
- SendStatus.h, 224
- setAccount
 - quentier::NoteEditor, 143
- setBackend
 - quentier::NoteEditor, 143
- setCurrentNoteLocalId
 - quentier::NoteEditor, 143
- setDefaultFont
 - quentier::NoteEditor, 144
- setDefaultPalette
 - quentier::NoteEditor, 144
- setDisplayName
 - quentier::Account, 18
- setFocus
 - quentier::NoteEditor, 144
- setIdleTimePeriod
 - quentier::utility::FileIOProcessorAsync, 40
- setInitialPageHtml
 - quentier::NoteEditor, 145
- setNoteDeletedPageHtml
 - quentier::NoteEditor, 145
- setNoteLoadingPageHtml
 - quentier::NoteEditor, 145
- setNoteNotFoundPageHtml
 - quentier::NoteEditor, 145
- setNoteTitle
 - quentier::NoteEditor, 145
- setTagIds
 - quentier::NoteEditor, 146
- setUndoStack
 - quentier::NoteEditor, 146
- setValue
 - quentier::utility::ApplicationSettings, 28, 29
- shardId
 - quentier::Account, 18
 - quentier::synchronization::IAuthenticationInfo, 46
- shortcut
 - quentier::utility::ShortcutManager, 165
- ShortcutManager.h, 273
- Size.h, 275
- SpellChecker.h, 204
- StandardPaths.h, 275
- startLinkedNotebooksDataDownloading
 - quentier::synchronization::ISyncEventsNotifier, 101
- stopSynchronizationError
 - quentier::synchronization::ISendStatus, 85
- StringUtils.h, 276
- SuppressWarnings.h, 277

- SyncChunksDataCounters.h, [225](#)
- syncChunksDataProcessingProgress
 - quentier::synchronization::ISyncEventsNotifier, [101](#)
- syncChunksDownloaded
 - quentier::synchronization::ISyncEventsNotifier, [101](#)
- syncChunksDownloadProgress
 - quentier::synchronization::ISyncEventsNotifier, [101](#)
- SyncResult
 - quentier::synchronization::ISynchronizer, [103](#)
- SyncResult.h, [225](#)
- SyncState.h, [226](#)
- SysInfo.h, [278](#)
- TagConflictResolution
 - quentier::synchronization::ISyncConflictResolver, [96](#)
- TagNotesRelation
 - quentier::local_storage::ILocalStorage, [71](#)
- TagSortByParentChildRelations.h, [278](#)
- Target
 - quentier::enml::conversion_rules::ISkipRule, [89](#)
- target
 - quentier::enml::conversion_rules::ISkipRule, [90](#)
- totalAttemptedToSendNotebooks
 - quentier::synchronization::ISendStatus, [86](#)
- totalAttemptedToSendNotes
 - quentier::synchronization::ISendStatus, [86](#)
- totalAttemptedToSendSavedSearches
 - quentier::synchronization::ISendStatus, [86](#)
- totalAttemptedToSendTags
 - quentier::synchronization::ISendStatus, [86](#)
- totalExpungedLinkedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters, [93](#)
- totalExpungedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters, [93](#)
- totalExpungedSavedSearches
 - quentier::synchronization::ISyncChunksDataCounters, [93](#)
- totalExpungedTags
 - quentier::synchronization::ISyncChunksDataCounters, [94](#)
- totalLinkedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters, [94](#)
- totalNotebooks
 - quentier::synchronization::ISyncChunksDataCounters, [94](#)
- totalSavedSearches
 - quentier::synchronization::ISyncChunksDataCounters, [94](#)
- totalSuccessfullySentNotebooks
 - quentier::synchronization::ISendStatus, [86](#)
- totalSuccessfullySentNotes
 - quentier::synchronization::ISendStatus, [87](#)
- totalSuccessfullySentSavedSearches
 - quentier::synchronization::ISendStatus, [87](#)
- totalSuccessfullySentTags
 - quentier::synchronization::ISendStatus, [87](#)
- totalTags
 - quentier::synchronization::ISyncChunksDataCounters, [94](#)
- toVersion
 - quentier::local_storage::IPatch, [80](#)
- TrackedTask.h, [241](#)
- type
 - quentier::Account, [18](#)
- UidGenerator.h, [279](#)
- undoStack
 - quentier::NoteEditor, [146](#)
- Unreachable.h, [280](#)
- updatedLinkedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters, [94](#)
- updatedNotebooks
 - quentier::synchronization::ISyncChunksDataCounters, [94](#)
- updatedSavedSearches
 - quentier::synchronization::ISyncChunksDataCounters, [95](#)
- updatedTags
 - quentier::synchronization::ISyncChunksDataCounters, [95](#)
- userId
 - quentier::synchronization::IAuthenticationInfo, [46](#)
- userOwnSendStatusUpdate
 - quentier::synchronization::ISyncEventsNotifier, [102](#)
- userShortcut
 - quentier::utility::ShortcutManager, [165](#), [166](#)
- userStoreCookies
 - quentier::synchronization::IAuthenticationInfo, [47](#)
- validateAndFixupEnml
 - quentier::enml::IConverter, [54](#)
- validateEnml
 - quentier::enml::IConverter, [54](#)
- Validation.h, [256](#)
- value
 - quentier::enml::conversion_rules::ISkipRule, [90](#)
 - quentier::utility::ApplicationSettings, [29](#), [30](#)
- webApiUrlPrefix
 - quentier::synchronization::IAuthenticationInfo, [47](#)
- WithNotes
 - quentier::local_storage::ILocalStorage, [72](#)
- WithoutNotes
 - quentier::local_storage::ILocalStorage, [72](#)
- writeFileRequestProcessed
 - quentier::utility::FileIOProcessorAsync, [41](#)
- writePassword
 - quentier::utility::IKeychainService, [66](#)