

Indexing

Overview

Documents the indexing strategy.

Details

The domain objects are annotated to integrate with the Hibernate-search framework which combines the persistence mechanisms of Hibernate JPA with Lucene indexing. Annotations on getter methods indicate which fields are to be indexed, whether the values are to be tokenized, and whether they are actually stored in the index. In the current configuration, all identifier and textual fields are indexed. Text fields are analyzed and identifiers are not.

Indexes are generated for the following objects:

- FeedbackConversationJpa
- MapProjectJpa
- MapRecordJpa
- ReportJpa
- ConceptJpa
- TreePostionJpa
- TrackingRecordJpa

These indexes support the various modes of searching within the application.

Fielded searches into these objects are supported. For each object type, the getter methods with @Field annotations support indexing with an index field name matching the getter field name (e.g. getWorkflowStatus translates to a field of "workflowStatus"). Where @IndexEmbedded annotations are used, nested fields can also be performed (e.g. for a map record you can search the field "entries.targetId").

For information on reindexing, see the admin tool page: [Lucene Reindex](#)

Configuration

The following configuration settings affect how search works.

Property	Default Value	Notes
hibernate.search.default.indexBase	EDIT_THIS/mapping/lucene/indexes	The indexing strategy uses a filesystem-based index. This is the configuration setting for the directory where the index files live.
hibernate.search.lucene_version	LUCENE_36	The Lucene version to use.
hibernate.search.default.directory_provider	filesystem	Indicates the style of indexing strategy.

References/Links

- <http://hibernate.org/search/>