



Hibernate Search 8.0.0.Alpha1

Migration Guide from 7.2

2024-12-17

Table of Contents

Introduction	1
Requirements	2
Artifacts	3
Data format and schema	4
Indexes	4
Outbox polling database tables	4
Configuration	5
API	6
SPI	7
Behavior	8

Introduction

The aim of this guide is to assist you migrating an existing application using any version **7. 2. x** of Hibernate Search to the latest of the **8. 0. x** series.



If you think something is missing or something does not work, please [contact us](#).

If you're looking to migrate from an earlier version, you should migrate step-by-step, from one minor version to the next, following the migration guide of [each version](#).



To Hibernate Search 5 users

Be aware that a lot of APIs have changed since Hibernate Search 5, some only because of a package change, others because of more fundamental changes (like moving away from using Lucene types in Hibernate Search APIs).

When migrating from Hibernate Search 5, you are encouraged to migrate first to Hibernate Search 6.0 using the [6.0 migration guide](#), and only then to later versions (which will be significantly easier).

Requirements

The requirements of Hibernate Search 8.0.0.Alpha1 are mostly the same as those of Hibernate Search 7.2, with the differences being Hibernate ORM and minimum JDK version upgrade:

- ¥ JDK 17 or later;
- ¥ Lucene 9 for its Lucene backend;
- ¥ Elasticsearch 7.10+ or OpenSearch 1.3+ for its Elasticsearch backend;
- ¥ Hibernate ORM 6.6.x for the Hibernate ORM integration.

Artifacts

The coordinates of Maven artifacts in Hibernate Search 8.0.0.Alpha1 are the same as in Hibernate Search 7.2.

Data format and schema

Indexes

The index format and schema in Hibernate Search 8.0.0.Alpha1 is backward-compatible with Hibernate Search 7.2: older indexes can be read from and written to without reindexing.

Outbox polling database tables

The event and agent database tables used for [outbox-polling](#) in Hibernate Search 8.0.0.Alpha1 are backward-compatible with Hibernate Search 7.2: no database schema update is necessary for these tables.

Configuration

The configuration properties in Hibernate Search 8.0.0.Alpha1 , in general, are backward-compatible with Hibernate Search 7.2. But some default values have changed:

¥ The default value of the Elasticsearch backend property `hibernate.search.backend.max_connections` is now set to **40** instead of **20**.

¥ The default value of the Elasticsearch backend property `hibernate.search.backend.max_connections_per_route` is now set to **20** instead of **10**.

API

The [API](#) in Hibernate Search 8.0.0.Alpha1 is, in general, backward-compatible with Hibernate Search 7.2. But there are next changes:

¥ An incubating `org.hibernate.search.mapper.pojo.standalone.loading.MassIdentifierLoader` changed the return type of `#totalCount()` from `Long` to `Optional Long`. This was done to address the scenarios where the total number of identifiers to load is not known ahead of time.

¥ Deprecated `org.hibernate.search.mapper.orm.massindexing.MassIndexingFailureHandler`, `org.hibernate.search.mapper.orm.massindexing.MassIndexingMonitor` interfaces are removed in this version. They have their alternatives in a `org.hibernate.search.mapper.pojo.massindexing` for a while now.

¥ `org.hibernate.search.mapper.pojo.massindexing.MassIndexingMonitor#addToTotalCount(..)` gets deprecated for removal. Instead, we are introducing the `org.hibernate.search.mapper.pojo.massindexing.MassIndexingTypeGroupMonitor` that can be obtained through `org.hibernate.search.mapper.pojo.massindexing.MassIndexingMonitor#typeGroupMonitor(..)`. This new type group monitor has more flexibility and also allows implementors to skip total count computations if needed.

¥ `multi()` methods exposed in various projection DSL steps are deprecated in favour of a `collector(ProjectionCollector.Provider)`, or one of the "shortcut-methods": `.list()` / `.set()` / `.sortedSet()` É Check the `ProjectionCollector` factory methods to see the list of built-in collectors that provide support for nullable/optional single-valued projections and for multivalued ones such as lists, sets arrays and more.

¥ The `org.hibernate.search.elasticsearch.request` logging category is now renamed to `org.hibernate.search.elasticsearch.client.request` and `org.hibernate.search.backend.lucene.infostream` to `org.hibernate.search.lucene.infostream` to be in a more consistent format compared with other logging categories.

¥ Hibernate Search now uses logging categories instead of class names to log messages. See [Appendix B: List of all available logging categories](#) to find out what categories are available.

SPI

The [SPI](#) in Hibernate Search 8.0.0.Alpha1 is, in general, backward-compatible with Hibernate Search 7.2. But there are next changes:

¥ [org.hibernate.search.mapper.pojo.loading.spi.PojoMassIndexerLoader](#) also changed the return type of `#totalCount()` from `Long` to `Optional Long` to reflect the changes in [the](#) [org.hibernate.search.mapper.pojo.standalone.loading.MassIndexerLoader](#)

Behavior

The behavior of Hibernate Search 8.0.0.Alpha1 is, in general, backward-compatible with Hibernate Search 7.2.

- ¥ The default mass indexer logging monitor updated the format of the logged messages to provide the information in a more condense form.
- ¥ In a few places related to the discovery of the inverse side of an association (in the ORM mapper) that previously logged warnings, Hibernate Search now will throw exceptions instead. This is related to [HSEARCH-4708](#).