HIBERNATE

Hibernate Search 7.2.3.Final

Migration Guide from 7.1

2025-03-18

Table of Contents

Introduction

The aim of this guide is to assist you migrating an existing application using any version `7.1.x` of Hibernate Search to the latest of the `7.2.x` series.

> ℹ️ 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 7.2.3.Final are mostly the same as those of Hibernate Search 7.1, with the only difference being Hibernate ORM version upgrade:

- JDK 11 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.

> ⚠️ Hibernate Search 7.2 depends on JBoss Logging 3.6 and uses its most recent APIs. If the application is using the JBoss Logging library either explicitly as a direct dependency or implicitly as a transient dependency of, e.g. Hibernate ORM or any other library, make sure that the resolved JBoss Logging version is 3.6; otherwise, you may see errors on startup.

Artifacts

The coordinates of Maven artifacts in Hibernate Search 7.2.3.Final are the same as in Hibernate Search 7.1.

Data format and schema

Indexes

The index format and schema in Hibernate Search 7.2.3.Final is backward-compatible with Hibernate Search 7.1: 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 7.2.3.Final are backward-compatible with Hibernate Search 7.1: no database schema update is necessary for these tables.

Configuration

The configuration properties in Hibernate Search 7.2.3.Final are backward-compatible with Hibernate Search 7.1.

API

The API in Hibernate Search 7.2.3.Final is backward-compatible with Hibernate Search 7.1.

With Hibernate ORM allowing non-`String` tenant identifiers, some of the Hibernate Search methods got deprecated and alternatives were introduced:

- `org.hibernate.search.mapper.orm.scope.SearchScope#workspace(String tenantId)` deprecated in favor of `org.hibernate.search.mapper.orm.scope.SearchScope#workspace(Object tenantId)`

- `org.hibernate.search.mapper.orm.scope.SearchScope#massIndexer(String tenantId)` deprecated in favor of `org.hibernate.search.mapper.orm.scope.SearchScope#massIndexer(Object tenantId)`

- `org.hibernate.search.mapper.pojo.standalone.scope.SearchScope#workspace(String tenantId)` deprecated in favor of `org.hibernate.search.mapper.pojo.standalone.scope.SearchScope#workspace(Object tenantId)`

- `org.hibernate.search.mapper.pojo.standalone.scope.SearchScope#massIndexer(String tenantId)` deprecated in favor of `org.hibernate.search.mapper.pojo.standalone.scope.SearchScope#massIndexer(Object tenantId)`

- `org.hibernate.search.mapper.pojo.standalone.session.SearchSession#tenantIdentifier()` deprecated in favor of `org.hibernate.search.mapper.pojo.standalone.session.SearchSession#tenantIdentifierValue()`

- `org.hibernate.search.mapper.pojo.standalone.session.SearchSessionBuilder#tenantId(String tenantId)` deprecated in favor of `org.hibernate.search.mapper.pojo.standalone.session.SearchSessionBuilder#tenantId(Object tenantId)`

- `org.hibernate.search.mapper.orm.outboxpolling.mapping.OutboxPollingSearchMapping#countAbortedEvents(String tenantId)` deprecated in favor of `org.hibernate.search.mapper.orm.outboxpolling.mapping.OutboxPollingSearchMapping#countAbortedEvents(Object tenantId)`

- `org.hibernate.search.mapper.orm.outboxpolling.mapping.OutboxPollingSearchMapping#reprocessAbortedEvents(String tenantId)` deprecated in favor of `org.hibernate.search.mapper.orm.outboxpolling.mapping.OutboxPollingSearchMapping#reprocessAbortedEvents(Object tenantId)`

- `org.hibernate.search.mapper.orm.outboxpolling.mapping.OutboxPollingSearchMapping#clearAllAbortedEvents(String tenantId)` deprecated in favor of `org.hibernate.search.mapper.orm.outboxpolling.mapping.OutboxPollingSearchMapping#clearAllAbortedEvents(Object tenantId)`

This version deprecates the `ValueConvert` enum used in various Search DSL methods and replaces it with a new `ValueModel` enum. For each deprecated method accepting the `ValueConvert` input,

there is now an alternative that accepts `ValueModel` instead. Replace `ValueConvert.YES` with `ValueModel.MAPPING` and `ValueConvert.NO` with `ValueModel.INDEX` in your code where the values were set explicitly.

SPI

The SPI in Hibernate Search 7.2.3.Final is backward-compatible with Hibernate Search 7.1.

Deprecated methods removed:

- `org.hibernate.search.mapper.pojo.scope.spi.PojoScopeDelegate#workspace(DetachedBackendSessionContext)`
- `org.hibernate.search.mapper.pojo.scope.spi.PojoScopeDelegate#massIndexer(PojoMassIndexingContext,DetachedBackendSessionContext)`

Behavior

The behavior of Hibernate Search 7.2.3.Final is backward-compatible with Hibernate Search 7.1.

> It is worth mentioning that with the changes in HSEARCH-5062 you may notice changes in the queries Hibernate Search creates to load entities. While we are expecting that the introduced changes should help with performance, if you notice any problems with them, we'd appreciate you reaching out to let us know.