



Hibernate Search 9.0.0.Beta1

Migration Guide from 8.4

2026-09-28

Table of Contents

Introduction	1
Requirements	2
Artifacts	3
Hibernate Search 5 migration helper removed.	3
Data format and schema	4
Indexes	4
Outbox polling database tables	4
Configuration	5
API	6
SPI	7
Deprecations	7
Behavior	8

Introduction

The aim of this guide is to assist you migrating an existing application using any version 8.4.x of Hibernate Search to the latest of the 9.0.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](#).

Requirements

The requirements of Hibernate Search 9.0.0.Beta1 are the same as those of Hibernate Search 8.4.

Artifacts

Hibernate Search 5 migration helper removed

The `hibernate-search-v5migrationhelper-engine` and `hibernate-search-v5migrationhelper-orm` artifacts are no longer published. If you are still using the migration helper, you will need to migrate to the current Hibernate Search APIs.

Other than the migration helper removal, the coordinates of Maven artifacts in Hibernate Search 9.0.0.Beta1 are the same as in Hibernate Search 8.4.

Data format and schema

Indexes

The index format and schema in Hibernate Search 9.0.0.Beta1 is backward-compatible with Hibernate Search 8.4: 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 9.0.0.Beta1 are backward-compatible with Hibernate Search 8.4: no database schema update is necessary for these tables.

Configuration

The configuration properties in Hibernate Search 9.0.0.Beta1, in general, are backward-compatible with Hibernate Search 8.4, with the following exceptions:

- The deprecated `hibernate.search.automatic_indexing.*` configuration properties have been removed:
 - `hibernate.search.automatic_indexing.enabled`—use `hibernate.search.indexing.listeners.enabled` instead.
 - `hibernate.search.automatic_indexing.strategy`—use `hibernate.search.indexing.listeners.enabled` instead (note: it expects a boolean value, not a strategy name).
 - `hibernate.search.automatic_indexing.synchronization.strategy`—use `hibernate.search.indexing.plan.synchronization.strategy` instead.
 - `hibernate.search.automatic_indexing.enable_dirty_check`—removed without replacement; dirty check is now always performed.

API

The [API](#) in Hibernate Search 9.0.0.Beta1 is, in general, backward-compatible with Hibernate Search 8.4, with the following exceptions:

- Both JPA and Hibernate ORM query adapters alongside with `Search.toJpaQuery()` and `Search.toOrmQuery()` have been removed after being deprecated for a while now. Use Hibernate Search query DSL API to work with your query and results.
- The `AutomaticIndexingSynchronizationStrategy` interface, `AutomaticIndexingSynchronizationStrategyNames`, `AutomaticIndexingSynchronizationConfigurationContext`, and `AutomaticIndexingStrategyName` enum have been removed. Use `IndexingPlanSynchronizationStrategy` and related types instead. The `SearchSession#automaticIndexingSynchronizationStrategy(...)` method has been removed; use `SearchSession#indexingPlanSynchronizationStrategy(...)` instead.
- The `org.hibernate.search.mapper.orm.common.EntityReference` interface has been removed. Use `org.hibernate.search.engine.common.EntityReference` instead. The `org.hibernate.search.mapper.orm.work.SearchIndexingPlanExecutionReport` interface has been removed; use `org.hibernate.search.mapper.pojo.work.SearchIndexingPlanExecutionReport` instead.
- The deprecated converter interfaces with "Field" in the name have been removed: `FromDocumentFieldValueConverter`, `ToDocumentFieldValueConverter`, `FromDocumentFieldValueConvertContext`, `ToDocumentFieldValueConvertContext`, `FromDocumentFieldValueConvertContextExtension`, `ToDocumentFieldValueConvertContextExtension`. Use their renamed counterparts: `FromDocumentValueConverter`, `ToDocumentValueConverter`, etc.
- The `entityReferences()` and `entityReference(Object)` methods on `EntityIndexingFailureContext` and `MassIndexingEntityFailureContext` have been removed. Use `failingEntityReferences()` and `failingEntityReference(EntityReference)` instead.
- The `ValueReadHandleFactory` SPI interface has been removed. Use `ValueHandleFactory` instead.
- The `Version.getVersionString()` method has been removed. Use `Version.versionString()` instead.

SPI

The [SPI](#) in Hibernate Search 9.0.0.Beta1 is, in general, backward-compatible with Hibernate Search 8.4, with the following exceptions:

- The `AutomaticIndexingStrategyStartContext` SPI interface has been removed. It was introduced by mistake and was never used.
- The `ValueReadHandleFactory` SPI interface has been removed. Use `ValueHandleFactory` instead.
- The `ElasticsearchBackendImplSettings` class (both `cfg.impl` and `cfg.spi` variants) has been removed.
- The `HibernateOrmQueryLoader.createMultiIdentifierLoadAccess(SessionImplementor)` method has been removed. Use `HibernateOrmQueryLoader.findMultiple(SessionImplementor, List, FindOption...)` instead.
- The `HibernateOrmLoadingSessionContext.session()` method now returns `SharedSessionContractImplementor` instead of `SessionImplementor`, and has been renamed to `sessionImplementor()`. This allows the loading context to work with both `Session` and `StatelessSession`.

Deprecations

- The `HibernateOrmSessionContext.session()` method returning `Session` has been deprecated. Use `HibernateOrmSessionContext.sessionContract()` instead, which returns `SharedSessionContract` and works with both `Session` and `StatelessSession`.

Behavior

The behavior of Hibernate Search 9.0.0.Beta1 is, in general, backward-compatible with Hibernate Search 8.4, with the following exception:

- Hibernate Search no longer wraps exceptions thrown by property getters or constructor invocations in a `SearchException` with an "Exception while invoking" message. Property getter exceptions pass through unchanged, while constructor invocation exceptions are propagated as provided by Hibernate Accessor. Higher-level indexing operations may still add their usual contextual exceptions. Update exception handlers and tests that rely on the previous wrapper or cause chain.