



Hibernate Search 7.1.2.Final

Migration Guide from 7.0

2024-08-30

Table of Contents

Introduction	1
Requirements	2
Artifacts	3
Data format and schema	4
Indexes	4
Outbox polling database tables	4
Configuration	8
API	9
SPI	10
Behavior	11

Introduction

The aim of this guide is to assist you migrating an existing application using any version **7.0.x** of Hibernate Search to the latest of the **7.1.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 7.1.2.Final are the same as those of Hibernate Search 7.0:

- JDK 11 or later;
- Lucene 9 for its Lucene backend;
- Elasticsearch 7.10+ or OpenSearch 1.3+ for its Elasticsearch backend;
- Hibernate ORM 6.4.x for the Hibernate ORM integration.

Artifacts

The coordinates of Maven artifacts in Hibernate Search 7.1.2.Final are the same as in Hibernate Search 7.0.

Data format and schema

Indexes

The index format and schema in Hibernate Search 7.1.2.Final is backward-compatible with Hibernate Search 7.0: older indexes can be read from and written to without reindexing.

Outbox polling database tables

If you use the incubating [outbox-polling coordination strategy](#), you may be impacted by the changes to entity mapping for entities that represents the outbox event and agent, requiring database schema changes.

Changes are only necessary if configuration properties `hibernate.search.coordination.entity.mapping.*.uuid_type` are not set or are set to `default`. If schema changes are not possible to apply at the moment, the following settings can be provided to specify the type to represent UUIDs in the database:

```
hibernate.search.coordination.entity.mapping.outboxevent.uuid_type=CHAR
hibernate.search.coordination.entity.mapping.agent.uuid_type=CHAR
```

The default type for UUIDs in outbox-polling tables now match [the default type for UUIDs in Hibernate ORM](#). See suggested migration scripts:

Postgresql:

```
-- change outbox event `id` column type to uuid:
alter table hsearch_outbox_event
  alter column id TYPE uuid USING cast(id as uuid);

-- change agent `id` column type to uuid:
alter table hsearch_agent
  alter column id TYPE uuid USING cast(id as uuid);
```

CockroachDB:

```
-- change outbox event `id` column type to uuid:
-- altering type directly is not supported: https://go.crdp.dev/issue-v/47636/v22.1
alter table hsearch_outbox_event
  add tmp uuid;
update hsearch_outbox_event
set tmp = cast(id as uuid)
where 1 = 1;
alter table hsearch_outbox_event
  alter column tmp set not null;
alter table hsearch_outbox_event
  alter primary key using columns (tmp);
alter table hsearch_outbox_event
  drop column id;
alter table hsearch_outbox_event
  rename column tmp to id;

-- change agent `id` column type to char:
alter table hsearch_agent
```

```

    add tmp uuid;
update hsearch_agent
set tmp = cast(id as uuid)
where 1 = 1;
alter table hsearch_agent
    alter column tmp set not null;
alter table hsearch_agent
    alter primary key using columns (tmp);
alter table hsearch_agent
    drop column id;
alter table hsearch_agent
    rename column tmp to id;

```

MySQL:

```

-- change outbox event `id` column type to binary(16):
alter table hsearch_outbox_event
    add column tmp binary(16);
update hsearch_outbox_event
set tmp = UUID_TO_BIN(id)
where 1 = 1;
alter table hsearch_outbox_event
    drop column id,
    rename column tmp to id;
alter table hsearch_outbox_event
    add primary key (id);

-- change agent `id` column type to binary(16):
alter table hsearch_agent
    add column tmp binary(16);
update hsearch_agent
set tmp = UUID_TO_BIN(id)
where 1 = 1;
alter table hsearch_agent
    drop column id,
    rename column tmp to id;
alter table hsearch_outbox_event
    add primary key (id);

```

MariaDB:

```

-- change outbox event `id` column type to uuid:
alter table hsearch_outbox_event
    modify column id uuid;

-- change agent `id` column type to uuid:
alter table hsearch_agent
    modify column id uuid;

```

DB2:

```

-- change outbox event `id` column type to binary(16):
alter table hsearch_outbox_event
    drop primary key;
alter table hsearch_outbox_event
    drop column id;
alter table hsearch_outbox_event
    add column id binary(16);
update hsearch_outbox_event
set id = generate_unique()
where 1 = 1;
alter table hsearch_outbox_event
    alter column id set not null;

```

```

-- make this call if the adding constraint fails:
call sysproc.admin_cmd('reorg table hsearch_outbox_event');
alter table hsearch_outbox_event
    add constraint hsearch_outbox_event_pkey primary key (id);

-- change agent `id` column type to binary(16):
alter table hsearch_agent
    drop primary key;
alter table hsearch_agent
    drop column id;
alter table hsearch_agent
    add column id binary(16);
update hsearch_agent
set id = generate_unique()
where 1 = 1;
alter table hsearch_agent
    alter column id set not null;
-- make this call if the adding constraint fails:
call sysproc.admin_cmd('reorg table hsearch_agent');
alter table hsearch_agent
    add constraint hsearch_agent_pkey primary key (id);

```

Oracle:

```

-- change outbox event `id` column type to raw:
alter table hsearch_outbox_event
    add tmp raw(16);
update hsearch_outbox_event
set tmp = SYS_GUID()
where 1 = 1;
alter table hsearch_outbox_event
    modify tmp not null;
alter table hsearch_outbox_event
    drop column id;
alter table hsearch_outbox_event
    rename column tmp to id;
alter table hsearch_outbox_event
    add constraint hsearch_outbox_event_pkey primary key (id);

-- change agent `id` column type to raw:
alter table hsearch_agent
    add tmp raw(16);
update hsearch_agent
set tmp = SYS_GUID()
where 1 = 1;
alter table hsearch_agent
    modify tmp not null;
alter table hsearch_agent
    drop column id;
alter table hsearch_agent
    rename column tmp to id;
alter table hsearch_agent
    add constraint hsearch_agent_pkey primary key (id);

```

MSSQL:

```

-- change outbox event `id` column type to uniqueidentifier:
alter table hsearch_outbox_event
    drop constraint if exists hsearch_outbox_event_pkey;
alter table hsearch_outbox_event
    alter column id uniqueidentifier not null;
alter table hsearch_outbox_event
    add constraint hsearch_outbox_event_pkey primary key (id);

```



```
-- change agent `id` column type to uuid:  
alter table hsearch_agent  
    drop constraint if exists hsearch_agent_pkey;  
alter table hsearch_agent  
    alter column id uniqueidentifier not null;  
alter table hsearch_agent  
    add constraint hsearch_agent_pkey primary key (id);
```

H2:

```
-- change outbox event `id` column type to uuid:  
alter table hsearch_outbox_event  
    alter column id uuid not null;  
  
-- change agent `id` column type to uuid:  
alter table hsearch_agent  
    alter column id uuid not null;
```

Configuration

The configuration properties in Hibernate Search 7.1.2.Final are backward-compatible with Hibernate Search 7.0.

API

All stable [API](#) in Hibernate Search 7.1.2.Final is backward-compatible with Hibernate Search 7.0.

However, some **incubating** APIs changed:

- `SearchMapping.builder(...)` in the Standalone POJO Mapper expects different arguments. See [this section of the documentation](#) for usage.
- `org.hibernate.search.mapper.pojo.standalone.mapping.metadata.EntityConfigurer`, `org.hibernate.search.mapper.pojo.standalone.mapping.metadata.EntityConfigurationContext`, as well as `StandalonePojoMappingConfigurationContext#addEntity(...)` methods have been deprecated.

The Standalone POJO Mapper no longer expects entities and their loading strategies to be primarily defined using a [mapping configurer](#), but rather using the `@SearchEntity` [annotation](#).

Programmatic definition of entities is still available, but is now [fully integrated in the programmatic mapping API](#).

SPI

Most [SPI](#) in Hibernate Search 7.1.2.Final is backward-compatible with Hibernate Search 7.0.

However, a few SPIs are not:

- `org.hibernate.search.mapper.pojo.mapping.spi.PojoMappingDelegate.createPojoScope` has been deprecated.

Consider taking advantage of `org.hibernate.search.mapper.pojo.mapping.spi.PojoMappingDelegate#createPojoScopeForClasses` or `org.hibernate.search.mapper.pojo.mapping.spi.PojoMappingDelegate.createPojoScopeForEntityNames`, which should greatly simplify the implementation of a mapper.

- `markAsEntity(...)` and `markAsIndexed(...)` `org.hibernate.search.mapper.pojo.model.additionalmetadata.building.spi.PojoAdditionalMetadataCollectorTypeNode` have been deprecated in favor of methods with the same name but not taking any argument.

Use methods in the returned `PojoAdditionalMetadataCollectorEntityTypeNode` and `PojoAdditionalMetadataCollectorIndexedTypeNode` to pass this information instead.

- A few SPIs related to loading in `hibernate-search-mapper-poj-base` have changed in [pull request #3954](#).

In particular, `pojo-base` now exposes entity definition APIs (including loading configuration), and handles most of the loading during search and mass indexing; mappers are only expected to implement `org.hibernate.search.mapper.pojo.mapping.building.spi.PojoTypeExtendedMappingCollector#applyLoadingBinder` to turn a user-provided (or mapper-provided) binder into loading strategies on startup.

As a consequence, mappers are no longer expected to provide loading strategies at runtime, e.g. in `PojoMassIndexingContext`, and some SPI types/methods have been removed in that area.

Loading strategy interfaces (`PojoSelectionLoadingStrategy`, `PojoMassLoadingStrategy`), experienced some changes as well, mostly to provide them with more context coming from the mapper, and to move them to the loading package (`org.hibernate.search.mapper.pojo.loading.spi`).

- A few SPIs related to loading in the Hibernate ORM mapper have changed in [pull request #3954](#).
- Deprecated `org.hibernate.search.mapper.pojo.model.path.spi.PojoPathsDefinition` class has been removed.

Feel free to [reach out to the Hibernate Search team](#) if you integrate with those SPIs and need help upgrading.

Behavior

The behavior of Hibernate Search 7.1.2.Final is backward-compatible with Hibernate Search 7.0.