

Hibernate

Introducción

Hibernate es un poderoso **framework de mapeo objeto-relacional (ORM)** para Java que simplifica significativamente el proceso de interacción con bases de datos relacionales. Desarrollado inicialmente por Gavin King en 2001 y actualmente mantenido por Red Hat, Hibernate se ha convertido en una herramienta fundamental para desarrolladores Java que buscan gestionar la persistencia de datos de manera eficiente y transparente.

En términos simples, Hibernate permite a los desarrolladores trabajar con objetos Java en lugar de consultas SQL directas. Esto se logra mediante el mapeo de objetos Java a tablas de bases de datos y viceversa, facilitando así la manipulación y persistencia de datos en la aplicación. Hibernate se integra estrechamente con el entorno de desarrollo Java y es compatible con una amplia gama de bases de datos relacionales, lo que lo convierte en una solución flexible y escalable para aplicaciones empresariales.

Entre sus características principales se incluyen:

- **Mapeo Objeto-Relacional (ORM):** Hibernate simplifica la representación de objetos Java como entidades persistentes en bases de datos relacionales, gestionando automáticamente la correspondencia entre las estructuras de datos y los objetos en el código.
- **Transacciones y gestión de la sesión:** proporciona un mecanismo robusto para administrar transacciones de bases de datos y sesiones de trabajo, asegurando la integridad y consistencia de los datos.
- **Consultas orientadas a objetos:** permite a los desarrolladores realizar consultas utilizando el lenguaje de consultas orientado a objetos de Hibernate (HQL), que es más intuitivo y menos propenso a errores que el SQL estándar.
- **Caché de segundo nivel:** Hibernate ofrece soporte para cache de segundo nivel, lo que mejora el rendimiento al reducir la cantidad de consultas repetitivas a la base de datos.
- **Soporte para herencia y asociaciones:** permite modelar relaciones complejas entre entidades y soporta herencia en la capa de persistencia, facilitando el diseño de bases de datos relacionales más complejas.

En resumen, Hibernate simplifica el desarrollo de aplicaciones Java al proporcionar una capa de abstracción entre la lógica de negocio y la capa de persistencia de datos, permitiendo a los desarrolladores centrarse en la funcionalidad de la aplicación sin tener que preocuparse por los detalles de la manipulación de la base de datos. Esto lo convierte en una herramienta indispensable en el arsenal de cualquier desarrollador Java que trabaje con bases de datos relacionales.

Sección generada por ChatGPT

Configuration and bootstrap

Including Hibernate in your project build

Para utilizar Hibernate fuera del entorno de un contenedor como WildFly o [Quarkus](#), se necesita:

- incluir el propio Hibernate ORM, junto con el controlador JDBC apropiado, como dependencias del su proyecto
- configurar Hibernate con información sobre la base de datos, especificando propiedades de configuración.

Añadir la dependencia de Hibernate tanto en Gradle como en Maven:

```

<dependencies>
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>{version}</version>
  </dependency>
</dependencies>

```

Reemplazando `{version}` por la versión de Hibernate a utilizar.

Hibernate se divide en varios módulos/artefactos bajo el grupo `org.hibernate.orm`. El artefacto principal se llama `hibernate-core`. Algunos de los módulos se encuentran [aquí](#).

Hibernate también proporciona un módulo de plataforma (**BOM** en terminología Maven) que se puede utilizar para alinear las versiones de los módulos de Hibernate junto con las versiones de sus bibliotecas. El artefacto de la plataforma se denomina `hibernate-platform`.

Para utilizar la plataforma BOM en Maven:

```

<dependencies>
  <dependency>
    <groupId>org.hibernate.orm</groupId>
    <artifactId>hibernate-core</artifactId>
  </dependency>
  <dependency>
    <groupId>jakarta.transaction</groupId>
    <artifactId>jakarta.transaction-api</artifactId>
  </dependency>
</dependencies>
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.hibernate.orm</groupId>
      <artifactId>hibernate-platform</artifactId>
      <version>6.5.2.Final</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>

```

También se debe agregar una dependencia para el controlador JDBC de la base de datos utilizada:

- PostgreSQL or CockroachDB -> `org.postgresql:postgresql:{version}`
- MySQL or TiDB -> `com.mysql:mysql-connector-j:{version}`
- MariaDB -> `org.mariadb.jdbc:mariadb-java-client:{version}`
- DB2 -> `com.ibm.db2:jcc:{version}`
- SQL Server -> `com.microsoft.sqlserver:mssql-jdbc:${version}`
- Oracle -> `com.oracle.database.jdbc:ojdbc11:${version}`
- H2 -> `com.h2database:h2:{version}`
- HSQLDB -> `org.hsqldb:hsqldb:{version}`

Donde `{versión}` es la última versión del controlador JDBC para la base de datos.

```
<dependencies>
  <dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <version>2.2.224</version>
  </dependency>
</dependencies>
```

Optional dependencies

Opcionalmente, también se puede agregar dependencias opcionales para trabajar con Hibernate como por ejemplo [SLF4J](#) o un [Query Validator](#) para comprobar las sentencias HQL en tiempo de compilación.

La lista está disponible en la documentación oficial.

Enlaces de interés

- [🔹 Hibernate](#)
- [Hibernate Guide](#)
- [Hibernate Getting Started](#)
- <https://www.baeldung.com/tag/hibernate>

Licencia



Esta obra está bajo una [licencia de Creative Commons Reconocimiento-Compartir Igual 4.0 Internacional](#).