índice

Tema 5 Manejo de conectores II.

Gestores de bases de datos embebidos e independientes

Variantes que tenemos de almacenamiento de datos en sistemas de bases de datos relacionales.

- Bases de datos en memoria: Almacena toda la información de la misma en memoria principal del sistema. (RAM). (Datos no se guardan)
- Bases de datos embebidas: bbdd integrada en la aplicación. Accederemos con JDBC driver. Recorre el mismo motor de la aplicación Java. No necesita un servidor independiente para funcionar. Para aplicaciones de pequeña escala, un usuario o pocos usuarios. SQLite, H2, HyperSQL
- Bases de datos independientes: Son base de datos que funcionan en un servidor idependiente y las aplicaciones se conectan al servidor para obtener los datos. Guardan gran tamaño de datos y muchos usuarios simultáneamente. Base de datos en un servidor interactúa a través de un JDBC o ODBC. Son los SGDB Sistemas de gestión de bases de datos relacionales.

Gestores de base de datos embebidos I: HyperSQL y ObjectDB

Sistemas gestores de bases de datos embebidos del mercado

- HyperSQL: Es una base de datos relacional. Se puede ejecutar tanto en modo embebido como en modo servidor y es estable y confiable. Esta desarrollado en Java, corre sin problema en nuestra JVM. Al descargarlo contiene un fichero .jar con el motor de la base de datos HyperSQL y el driver JDBC
- ObjectDB: También tiene los dos modos embebido y servidor. Es una base de datos orientada a objetos.

Gestores de base de datos embebidos II: Java DB, Apache Derby y H2

Otros sistemas gestores embebidos.

- Java DB y Apache Derby: son muy similares, Java DB está construida sobre el motor de la base de datos Derby. Embebida o en modo servidor. Para embeber nuestra base de datos en nuestra app java, hay que incluir el .jar en /lib.
- h2 Database: Características:

  -Es muy rápida, código abierto, JDBC API.

  -Contiene modo cliente/servidor, modo embebido y modo desplegable en memoria.

  -Se puede usar para webs

  -Muy manejable y transportable, el .jar ocupa 2MB de espacio total.

  -Soporta el MVCC

  -Se puede usar Postgress OBDC driver.

Gestores de bases de datos embebidos III: Comparativa

Podemos observar que la base de datos h2 es muy buena opción para agregar una base de datos embebida.

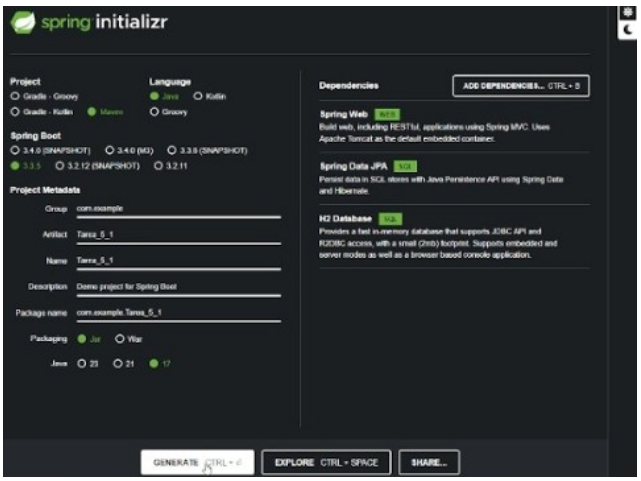
Ejemplo	h2	Derby	HYPERSQL	MYSQL	POSTGRESS
Java Puro	si	si	si	no	no
Modo memoria	si	si	si	no	no
Bbdd encriptada	si	si	si	no	no
Driver ODBC	si	no	no	si	si
Búsqueda texto completo	si	no	no	si	si
MVCC	si			si	si
Espacio (embebido)	-2MB	-3MB	-1.5MB	-	-
Espacio (cliente)	-500MB	-600MB	-1.5MB	-1MB	-700KB

Instalar una base de datos H2 en una aplicación web Java ayudándonos del Framework Spring Boot.

¿Qué es Spring Boot?

Es un módulo dentro de Spring que sirve para crear aplicaciones, seleccionando las dependencias y las auto configura. También sirve para desplegar la aplicación.

Crear un proyecto con las dependencias elegidas.



Base de datos embebida en aplicación web a través de Spring Boot I: Añadir BBDD

Web: https://start.spring.io/

- Elegimos Maven
- Lenguaje Jaava
- Spring boot la versión que venga por defecto
- En project Metadata podemos cambiar nombre del proyecto y ponerle una descripcion
- Packaging elegimos Jar
- En dependencies elegimos H2 Database (embebida)
- Una vez añadamos todas las dependencias le damos a generate

Base de datos embebida en aplicación web a través de Spring Boot II: IDE

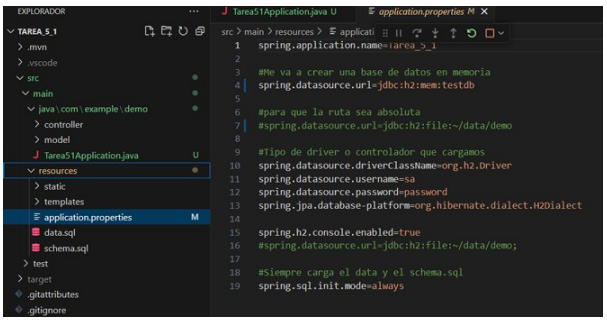
- Añadimos también la dependencia de Spring Web.
- Le damos a generate.
- Nos descargara un zip y lo descomprimimos.
- Spring Boot nos genera un fichero dentro del proyecto llamado "Application.properties"

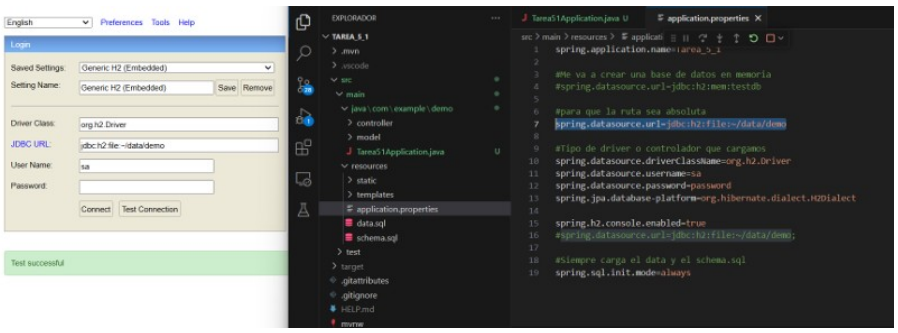
  Donde se configura varias propiedades y ajustes de la aplicación como la configuración de la bbdd, seguridad, logging…

  - base de datos en memoria Ram: spring.datasource.url=jdbc:h2:file:~/data/demo
  - bbdd en archivo

    spring.datasource.url=jdbc:h2: file:~/data/demo
  - usuario y contraseña: username y password
  - para poder acceder también poner la línea:

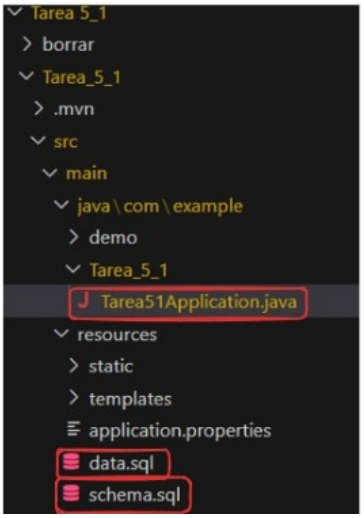
    spring.h2.console.enabled=true





Base de datos embebida en aplicación web a través de Spring Boot III: Inicio BBDD

- Creamos un **data.sql** donde iran los insert y un **schema.sql** donde irán las tablas



en resources creamos data.sql y schema.sql

- acceder a la aplicación: localhost puerto 8080

http://localhost:8080/h2-console

Gestores de base de datos independientes

La principal diferencia de las bases independientes y las embebidas es que una **bbdd independiente no puede ejecutarse bajo la misma máquina virtual** de Java. Siempre **consumirá más recursos que una embebida**

BBDD independientes más conocidas:

- SQL Server DataBase
- Oracle
- Postgres SQL
- MySql