



SpringBoot

Jaime Navas Carmona



En esta práctica voy a aplicar Spring boot con una base de datos.

Lo primero que hago es meterme en spring.io y ajustarlo a mi medida. El tipo de lenguaje que es java, el proyecto que es Maven y añado 3 dependencias: SPRING web. Spring Data JPA y H2 Database. Y le pongo el nombre que yo quiera, en mi caso TiendaDb_Jaime.

The screenshot shows the Spring Initializr configuration page. It is divided into several sections:

- Project:** Includes radio buttons for 'Gradle - Groovy', 'Gradle - Kotlin', and 'Maven' (which is selected).
- Language:** Includes radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'.
- Spring Boot:** Includes radio buttons for various versions: '4.0.0 (SNAPSHOT)', '4.0.0 (RC1)', '3.5.8 (SNAPSHOT)', '3.5.7' (selected), '3.4.12 (SNAPSHOT)', and '3.4.11'.
- Project Metadata:** A form with fields for:
 - Group: `com.medac`
 - Artifact: `TiendaDb_Jaime`
 - Name: `TiendaDb_Jaime`
 - Description: `Práctica de Acceso a Datos con H2`
 - Package name: `com.medac.TiendaDb_Jaime`
- Dependencies:** A list of dependencies with a button 'ADD DEPENDENCIES... CTRL + B'. The listed dependencies are:
 - Spring Web** (WEB): Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
 - Spring Data JPA** (SQL): Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
 - H2 Database** (SQL): Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.



Ahora con lo que hemos hecho antes le daremos a generar y se nos generara un proyecto vacío de java pero con las características de SpringBoot. Ahora hay que hacer lo mismo que para conectarse a una BBDD pero con las credenciales de SpringBoot.

```
SpringApplication.run(TiendaDbJaimeA  
    String DRIVER = "org.h2.Driver";  
    String url = "jdbc:h2:mem:testdb";  
    String usuario = "sa";  
    String contra = "password";  
    int opcion = 0;
```

Ahora creamos un archivo application.properties y en este archivo meteremos lo siguiente:

Nombre de la aplicación, la url, el driver, el nombre de usuario, la contraseña, la plataforma, el modo de inicio, el puerto y la consola.

```
1  spring.application.name=TiendaDb_Jaime  
2  spring.datasource.url=jdbc:h2:mem:testdb  
3  spring.datasource.driverClassName=org.h2.Driver  
4  spring.datasource.username=sa  
5  spring.datasource.password=password  
6  spring.jpa.database-platform=org.hibernate.dialect.H2Dialect  
7  spring.sql.init.mode = always  
8  server.port= 8080  
9  spring.h2.console.enabled=true  
10
```



Ahora ya la podemos desplegar en el navegador, metemos de username sa y de password password y accedemos a nuestra base de datos.

English Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:testdb

User Name: sa

Password:

Connect Test Connection

Hacemos una consulta para comprobar que es nuestra BBBDD.

Run Run Selected Auto complete Clear SQL statement:

select * from pedidos;

ID_PEDIDO	FECHA	ID_CLIENTE
1	2024-01-10	1
2	2024-01-11	2
3	2024-01-12	3
4	2024-05-05	1

(4 rows, 4 ms)



Ahora vamos a meter nuestro programa de java anteriormente creado en nuestro proyecto de SpringBoot de java. Ahora vamos a insertar un cliente desde la consola de java y se nos guardara en nuestra base de datos de forma perezcedera ya que cuando apaguemos el programa el insert ya no estará ya que lo hemos guardado en memoria. Ya que como vemos en el codigo he puesto en la url mem:testdb eso es que se va a guardar en memoria.

```
1  spring.application.name=TiendaDb_Jaime
2  spring.datasource.url=jdbc:h2:mem:testdb
3  spring.datasource.driverClassName=org.h2.Driver
4  spring.datasource.username=sa
5  spring.datasource.password=password
6  spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
7  spring.sql.init.mode = always
8  server.port= 8080
9  spring.h2.console.enabled=true
10
```

```
SpringApplication.run(TiendaDbJaimeA
| String DRIVER = "org.h2.Driver";
| String url = "jdbc:h2:mem:testdb";
| String usuario = "sa";
| String contra = "password";
| int opcion = 0;
```



```
1.-Crear cliente
2.-Crear producto
3.-Registrar nuevo pedido
4.-Consultar detalles pedido
5.-Eliminar un producto
6.-Salir
Elige una opcion
1
Nombre del cliente:
ivan garcia
Edad del cliente:
44
Ciudad del cliente:
marruecos
Cliente registrado
```

```
select * from cliente;
```

```
select * from cliente;
```

ID_CLIENTE	NOMBRE	EDAD	CIUDAD
1	Ana Lopez	30	Madrid
2	Carlos Ruiz	45	Sevilla
3	Lucía Gómez	22	Granada
4	ivan garcia	44	marruecos

(4 rows, 5 ms)



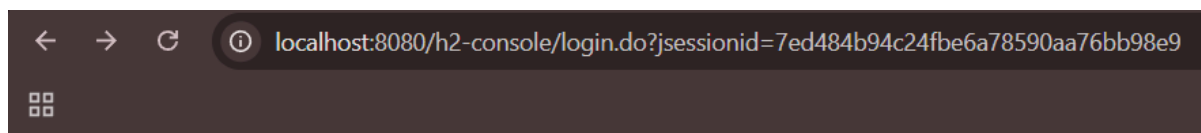
Ahora en cambio voy a hacer que los datos que se inserten se queden guardados de forma permanente en la BBDD, para eso simplemente cambié la url y en vez de poner mem:testdb, pongo file:~/data/demo tanto en el archivo application como en la variable url del main.

```
1  spring.application.name=TiendaDb_Jaime
2  spring.datasource.url=jdbc:h2:file:~/data/demo
3  spring.datasource.driverClassName=org.h2.Driver
4  spring.datasource.username=sa
5  spring.datasource.password=password
6  spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
7  spring.sql.init.mode = always
8  server.port= 8080
9  spring.h2.console.enabled=true
10
```

```
SpringApplication.run(TiendaDbJaimeApplication.class, true);
String DRIVER = "org.h2.Driver";
String url = "jdbc:h2:file:~/data/demo";
String usuario = "sa";
String contra = "password";
int opcion = 0;
```



Ahora una vez que lo desplegamos en el navegador tendremos que cambiar la url del jdbc a la que hemos puesto anteriormente. Ahora una vez que hagamos el insert y el select se nos guardara en la base de datos para siempre.



English Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:file:~/data/demo

User Name: sa

Password:

Connect Test Connection

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:file:~/data/demo

- CLIENTE
- PEDIDOS
- PEDIDO_PRODUCTO
- PRODUCTO
- INFORMATION_SCHEMA
- Users
- H2 2.3.232 (2024-08-11)

Run Run Selected Auto complete Clear SQL statement:

select * from cliente

select * from cliente;

ID_CLIENTE	NOMBRE	EDAD	CIUDAD
1	Ana Lopez	30	Madrid
2	Carlos Ruiz	45	Sevilla
3	Lucía Gómez	22	Granada
4	raul	66	murcia

(4 rows, 5 ms)



Por último una vez desplegada la base de datos en el navegador para guardar los archivos de forma permanente nos tendremos que ir al archivo application.properties y cambiar el init mode a never ya que se borrarán los datos y vuelve a cargar los datos iniciales.

```
1  spring.application.name=TiendaDb_Jaime
2  spring.datasource.url=jdbc:h2:file:~/data/demo
3  spring.datasource.driverClassName=org.h2.Driver
4  spring.datasource.username=sa
5  spring.datasource.password=password
6  spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
7  spring.sql.init.mode = never
8  server.port= 8080
9  spring.h2.console.enabled=true
10
```