Certified Tech Developer
The Ultimate Degree

Configurando o Hibernate com o Spring Boot

Vamos ver um exemplo!

1. No pom.xml de nosso projeto Spring Boot, devemos adicionar as seguintes dependências:

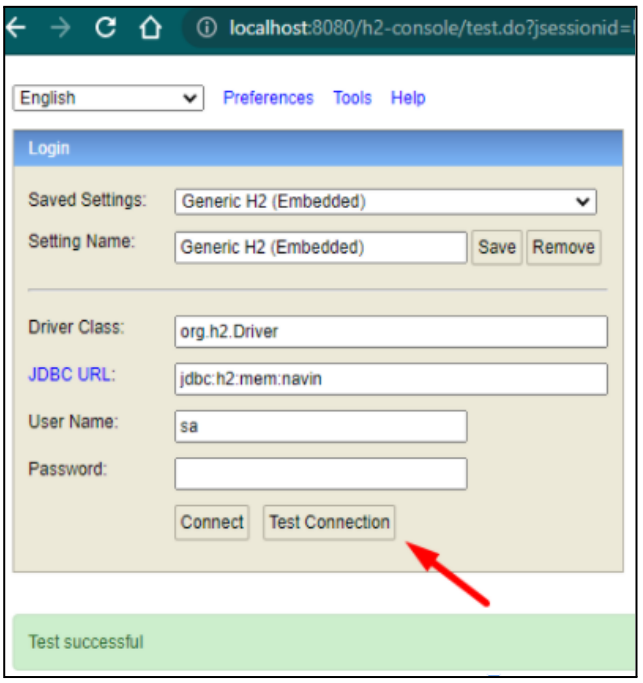
- **spring-boot-starter-data-jpa:** inclui a API JPA, a implementação JPA, JDBC e outras bibliotecas. Como a implementação padrão do JPA é o Hibernate, essa dependência também está inclusa.

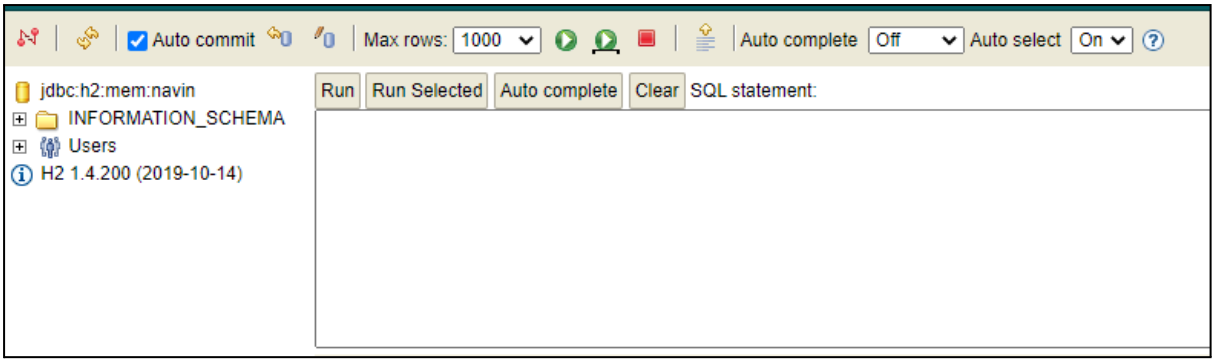
- **com.h2database:** para poder realizar um teste rapidamente, podemos adicionar o H2 (um banco de dados na memória muito leve). Em **application.properties** habilitamos o console do banco de dados H2 para acessá-lo por meio de uma UI.

```
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<dependency>
   <groupId>com.h2database</groupId>
   <artifactId>h2</artifactId>
   <scope>runtime</scope>
</dependency>
```

2. Além de incluir as dependências no pom.xml, devemos incluir as propriedades do banco de dados no arquivo **application.properties**. Este é um arquivo de configuração e é onde o **Spring** relaciona nosso projeto ao banco de dados que queremos usar. Portanto, para se conectar ao banco de dados, devemos disponibilizar algumas informações no arquivo application.properties:

3. Executamos a aplicação e verificamos se ela estabelece a conexão no navegador.





4. Criamos a entidade Student:

```
@Entity
public class Student {

@Id
@GeneratedValue(strategy=generationType.SEQUENCE)
private Long id;
private String dni;
private String name;
private String lastName;
}
```

5. Para acessar os dados com o Spring Data, só precisamos criar os repositórios. Por exemplo, para criar o repositório para a classe Student, apenas definimos a interface, **IStudentRepository** que se estende de **JpaRepository**.

```
public interface IStudentRepository extends JpaRepository <Student , Long> {
}
```

6. O que fizemos foi criar uma interface que se estende de **JpaRepository<T, ID>** onde:

- **T:** deve ser o nome da classe que será utilizada para criar o repositório (em nosso exemplo seria Student).
- **ID:** o tipo de dado que será usado como identificador ou chave primária no banco de dados (em nosso caso, Long).

Com esse Spring Data, serão criadas as operações CRUD para a entidade Student. Isso significa que agora podemos: criar, ler, atualizar e excluir um aluno (Student) do banco de dados.

7. Criamos um serviço para o repositório, no qual é injetado por meio do construtor:

```
public class StudentService {

    private final IStudentRepositoroy studentRepository;

    public StudentService(IStudentRepository studentRepository)

    {

    this.studentRepository = studentRepository;

    }

}
```

8. Executamos a aplicação e conectamos novamente no banco de dados (http://localhost:8080/h2-console).