
spring initializr

Project

☐ Gradle - Groovy
 ☐ Gradle - Kotlin
 ☒ Maven

Language

☒ Java
 ☐ Kotlin
 ☐ Groovy

Spring Boot

☐ 3.2.0 (SNAPSHOT)
 ☐ 3.2.0 (RC2)
 ☐ 3.1.6 (SNAPSHOT)
 ☒ 3.1.5
 ☐ 3.0.13 (SNAPSHOT)
 ☐ 3.0.12
 ☐ 2.7.18 (SNAPSHOT)
 ☐ 2.7.17

Project Metadata

Group

com.in28minutes.springboot.rest.example

Artifact

spring-boot-2-jdbc-with-h2

Name

spring-boot-2-jdbc-with-h2

Description

Demo project for Spring Boot

Package name

com.in28minutes.springboot.rest.example.spring-boot-2-jdbc-with-h2

Packaging

☒ Jar
 ☐ War

Java

☐ 21
 ☒ 17
 ☐ 11
 ☐ 8

Dependencies

ADD DEPENDENCIES... CTRL + B

Spring Web

WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

JDBC API

SQL

Database Connectivity API that defines how a client may connect and query a database.

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.

Spring Boot DevTools

DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

GENERATE CTRL + G

EXPLORE CTRL + SPACE

SHARE...

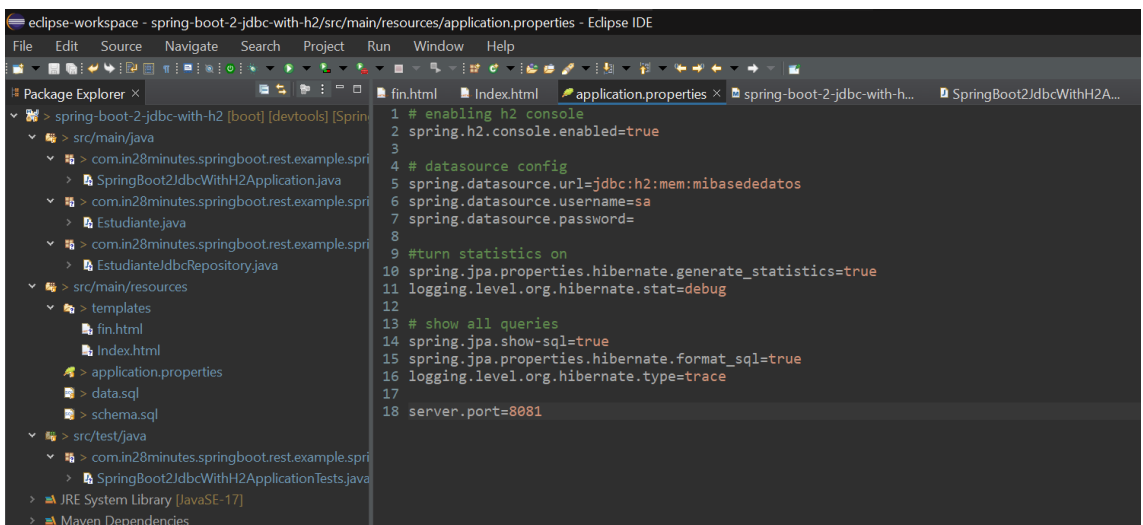
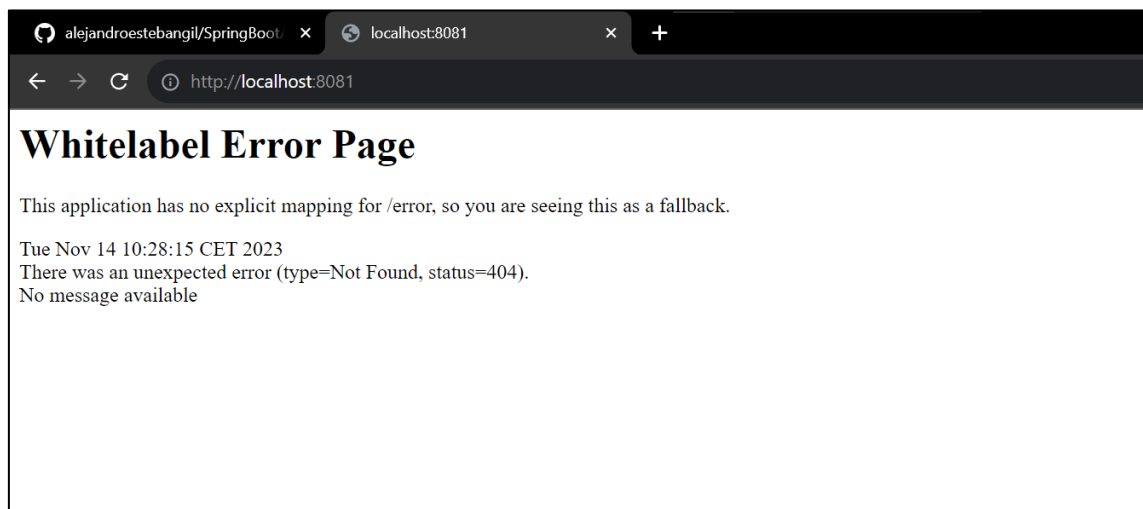
```

> spring-boot-2-jdbc-with-h2 [boot] [devtools] [Spring]
  > src/main/java
    > com.in28minutes.springboot.rest.example.springboot
      > SpringBoot2JdbcWithH2Application.java
    > com.in28minutes.springboot.rest.example.springboot
      > Estudiante.java
    > com.in28minutes.springboot.rest.example.springboot
      > EstudianteJdbcRepository.java
  > src/main/resources
    > templates
      > fin.html
      > Index.html
    > application.properties
    > data.sql
    > schema.sql
  > src/test/java
    > com.in28minutes.springboot.rest.example.springboot
      > SpringBoot2JdbcWithH2ApplicationTests.java
  > JRE System Library [JavaSE-17]
  > Maven Dependencies
    > target/generated-sources/annotations
    > target/generated-test-sources/test-annotations
  > JUnit 4
  > src
    > main
    > test
  > target
  > mvnw
  > mvnw.cmd
  > pom.xml

```

```
~/DDI/SpringBootAPP-Practica/spring-boot-2-jdbc-with-h2 master* 4m 5s
> mvn spring-boot:run
[INFO] Scanning for projects...
[INFO]
--> com.in28minutes.springboot.rest.example:spring-boot-2-jdbc-with-h2 >--
[INFO] Building spring-boot-2-jdbc-with-h2 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
>>> spring-boot-maven-plugin:3.1.5:run (default-cli) > test-compile @ spring-boot-2-jdbc-with-h2 >>>
[INFO]
--- maven-resources-plugin:3.3.1:resources (default-resources) @ spring-boot-2-jdbc-with-h2 ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 4 resources from src/main/resources to target/classes
[INFO]
--- maven-compiler-plugin:3.11.0:compile (default-compile) @ spring-boot-2-jdbc-with-h2 ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
--- maven-resources-plugin:3.3.1:testResources (default-testResources) @ spring-boot-2-jdbc-with-h2 ---
[INFO] skip non existing resourceDirectory /home/alejandro/DDI/SpringBootAPP-Practica/spring-boot-2-jdbc-with-h2/src/test/resources
[INFO]
--- maven-compiler-plugin:3.11.0:testCompile (default-testCompile) @ spring-boot-2-jdbc-with-h2 ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
<<< spring-boot-maven-plugin:3.1.5:run (default-cli) < test-compile @ spring-boot-2-jdbc-with-h2 <<<
[INFO]
--- spring-boot-maven-plugin:3.1.5:run (default-cli) @ spring-boot-2-jdbc-with-h2 ---
[INFO] Attaching agents: []

  ____ _
 / ___ \ | |
/ /___\ \| |
 \___ \_|_|
  _____|_|
:: Spring Boot ::
                (v3.1.5)
```



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

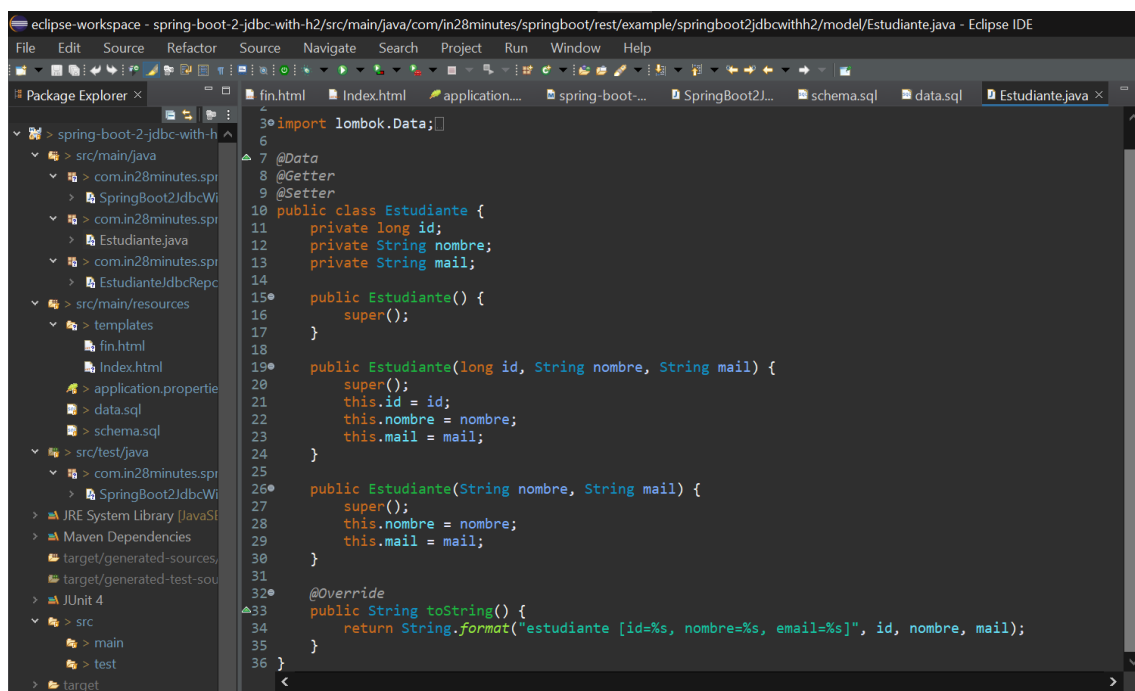
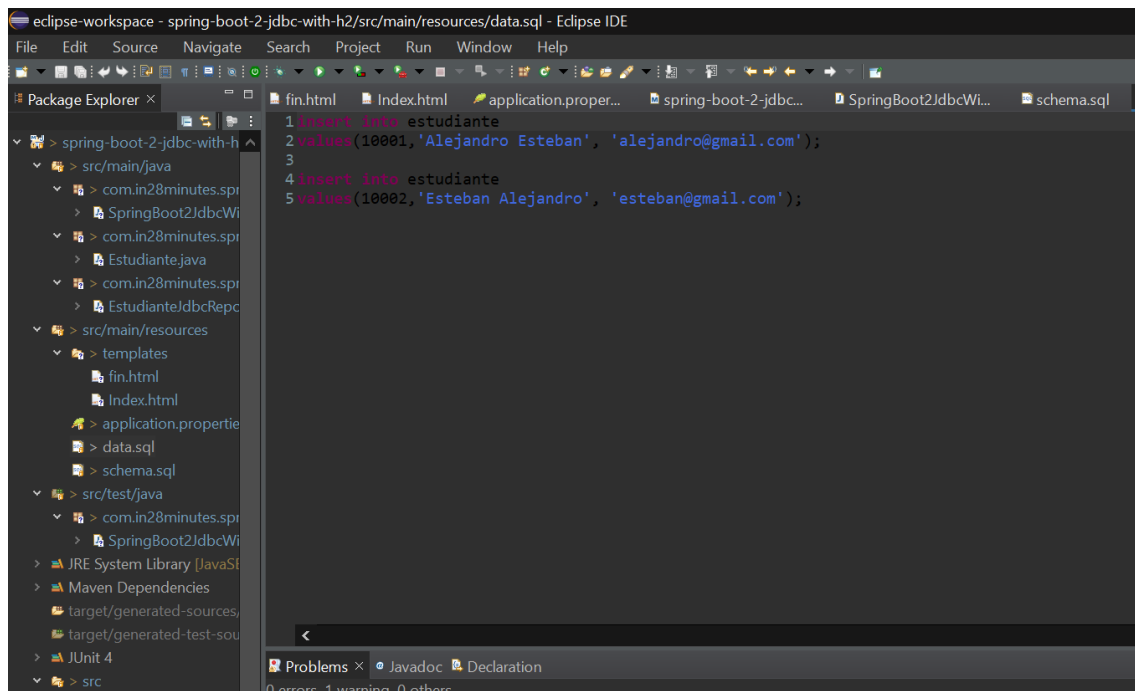
eclipse-workspace - spring-boot-2-jdbc-with-h2/src/main/resources/schema.sql - Eclipse IDE

File Edit Source Navigate Search Project Run Window Help

Package Explorer

- spring-boot-2-jdbc-with-h2
 - src/main/java
 - com.in28minutes.springboot.jdbcwithh2
 - SpringBoot2JdbcWithH2Application.java
 - Estudiante.java
 - EstudianteJdbcRepository.java
 - src/main/resources
 - templates
 - fin.html
 - Index.html
 - application.properties
 - data.sql
 - schema.sql
 - src/test/java

```
1 create table estudiante
2 (
3     id integer not null,
4     nombre varchar(255) not null,
5     mail varchar(255) not null,
6     primary key(id)
7 );
8
```



eclipse-workspace - spring-boot-2-jdbc-with-h2/src/main/java/com/in28minutes/springboot/rest/example/springboot2jdbcwithh2/repository/EstudianteJdbcRepository.java - Eclipse IDE

```
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer
spring-boot-2-jdbc-with-h2
src/main/java
com.in28minutes.springboot
SpringBoot2JdbcWithH2Application
com.in28minutes.springboot
Estudiante.java
com.in28minutes.springboot
EstudianteJdbcRepository.java
src/main/resources
templates
fin.html
index.html
application.properties
data.sql
schema.sql
src/test/java
com.in28minutes.springboot
SpringBoot2JdbcWithH2Application
JRE System Library [JavaSE-8]
Maven Dependencies
target/generated-sources
target/generated-test-sources
JUnit 4
src
main
test
target
```

```
15 @Controller
16 public class EstudianteJdbcRepository {
17
18     @Autowired
19     JdbcTemplate jdbcTemplate;
20
21     class EstudianteRowMapper implements RowMapper<Estudiante> {
22         @Override
23         public Estudiante mapRow(ResultSet rs, int rowNum) throws SQLException {
24             Estudiante estudiante = new Estudiante();
25             estudiante.setId(rs.getLong("id"));
26             estudiante.setNombre(rs.getString("nombre"));
27             estudiante.setMail(rs.getString("mail"));
28             return estudiante;
29         }
30     }
31
32     public Estudiante findById(long id) {
33         return jdbcTemplate.queryForObject("select * from estudiante where id=?",
34             new BeanPropertyRowMapper<Estudiante>(Estudiante.class), new Object[] { id });
35     }
36
37     public List<Estudiante> findAll() {
38         return jdbcTemplate.query("SELECT * FROM estudiante", new EstudianteRowMapper());
39     }
40
41     public int deleteById(long id) {
42         return jdbcTemplate.update("delete from estudiante where id=?", new Object[] { id });
43     }
44
45     public int insert(Estudiante estudiante) {
46         return jdbcTemplate.update("insert into estudiante (id, nombre, mail) " + "values(?, ?, ?)",
```

eclipse-workspace - spring-boot-2-jdbc-with-h2/src/main/java/com/in28minutes/springboot/rest/example/springboot2jdbcwithh2/SpringBoot2JdbcWithH2Application.java - Eclipse IDE

```
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer
spring-boot-2-jdbc-with-h2
src/main/java
com.in28minutes.springboot
SpringBoot2JdbcWithH2Application
com.in28minutes.springboot
Estudiante.java
com.in28minutes.springboot
EstudianteJdbcRepository.java
src/main/resources
templates
fin.html
index.html
application.properties
data.sql
schema.sql
src/test/java
com.in28minutes.springboot
SpringBoot2JdbcWithH2Application
JRE System Library [JavaSE-8]
Maven Dependencies
target/generated-sources
target/generated-test-sources
JUnit 4
src
main
test
target
```

```
1 package com.in28minutes.springboot.rest.example.springboot2jdbcwithh2;
2
3 import org.slf4j.Logger;
4
5 @SpringBootApplication
6 public class SpringBoot2JdbcWithH2Application implements CommandLineRunner {
7
8     private Logger logger = LoggerFactory.getLogger(getClass());
9
10    @Autowired
11    EstudianteJdbcRepository repository;
12
13    public static void main(String[] args) {
14        SpringApplication.run(SpringBoot2JdbcWithH2Application.class, args);
15    }
16
17    @Override
18    public void run(String... args) throws Exception {
19        logger.info("estudiante id 10001 -> {}", repository.findById(10001));
20        logger.info("all users 1 -> {}", repository.findAll());
21        logger.info("inserting -> {}",
22            repository.insert(new Estudiante(10010, "Paco Sanz", "pacosanz@gmail.com")));
23
24        logger.info("update 10001 -> {}",
25            repository.update(new Estudiante(10001, "Sanz Paco", "sanzpaco@gmail.com")));
26
27        repository.deleteById(10002);
28
29        logger.info("all users 2 -> {}", repository.findAll());
30    }
31}
```

