SPRING FRAMEWORK COURSE

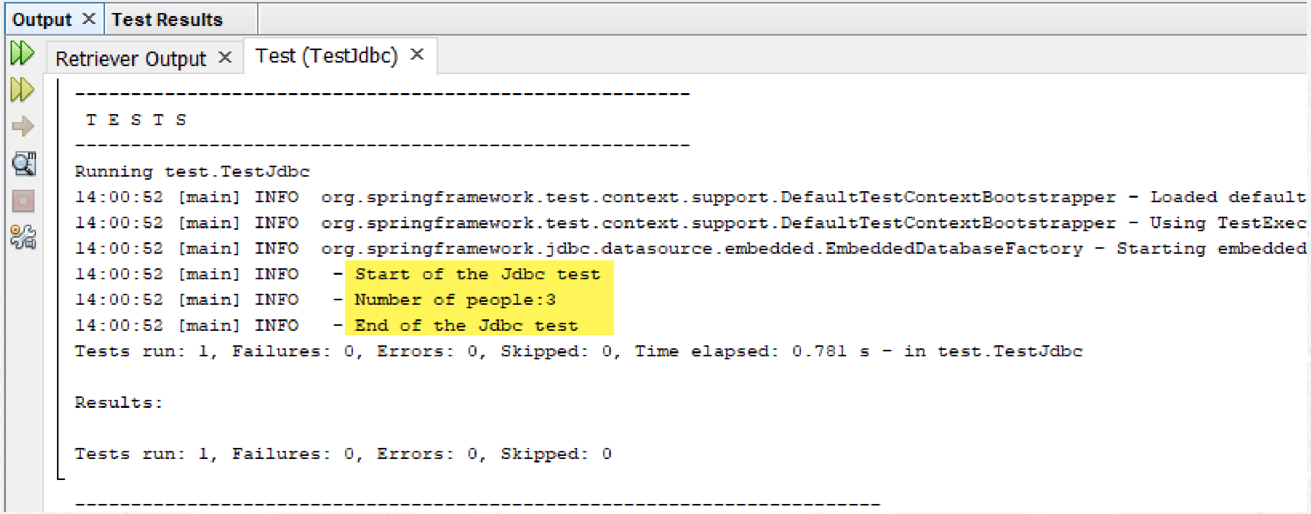
EXERCISE

SPRING JDBC CONFIGURATION

Global Mentoring

EXERCISE OBJECTIVE

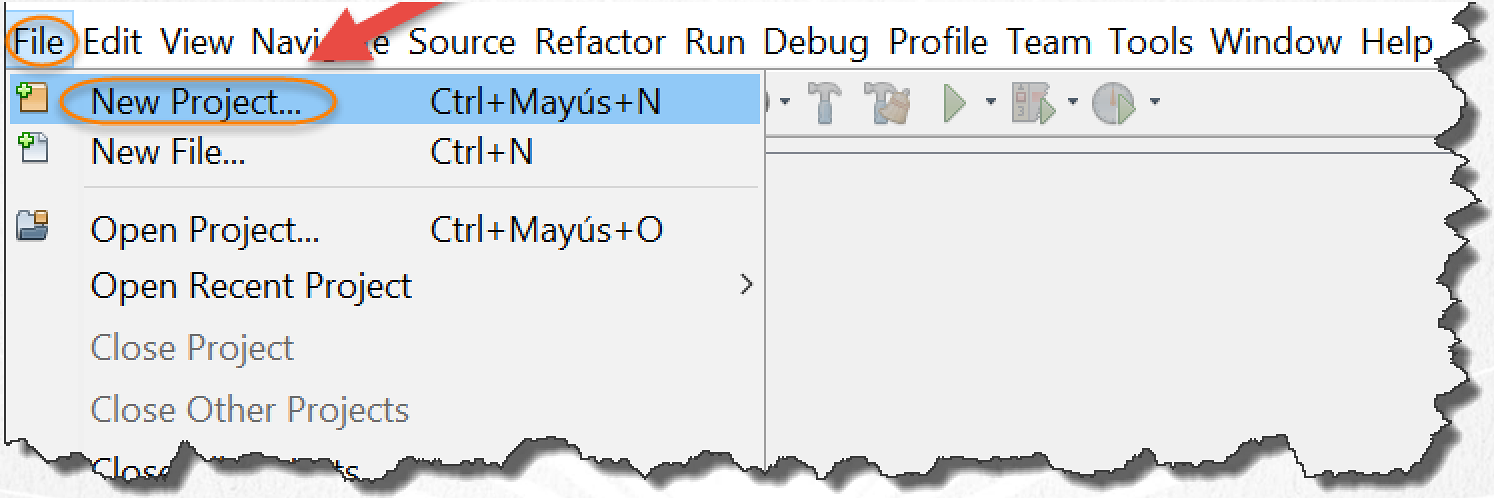
Create an exercise to configure Spring JDBC. At the end we must observe:



```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running test.TestJdbc
14:00:52 [main] INFO  org.springframework.test.context.support.DefaultTestContextBootstrapper - Loaded default
14:00:52 [main] INFO  org.springframework.test.context.support.DefaultTestContextBootstrapper - Using TestExec
14:00:52 [main] INFO  org.springframework.jdbc.datasource.embedded.EmbeddedDatabaseFactory - Starting embedded
14:00:52 [main] INFO   - Start of the Jdbc test
14:00:52 [main] INFO   - Number of people:3
14:00:52 [main] INFO   - End of the Jdbc test
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.781 s - in test.TestJdbc

Results:

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
```

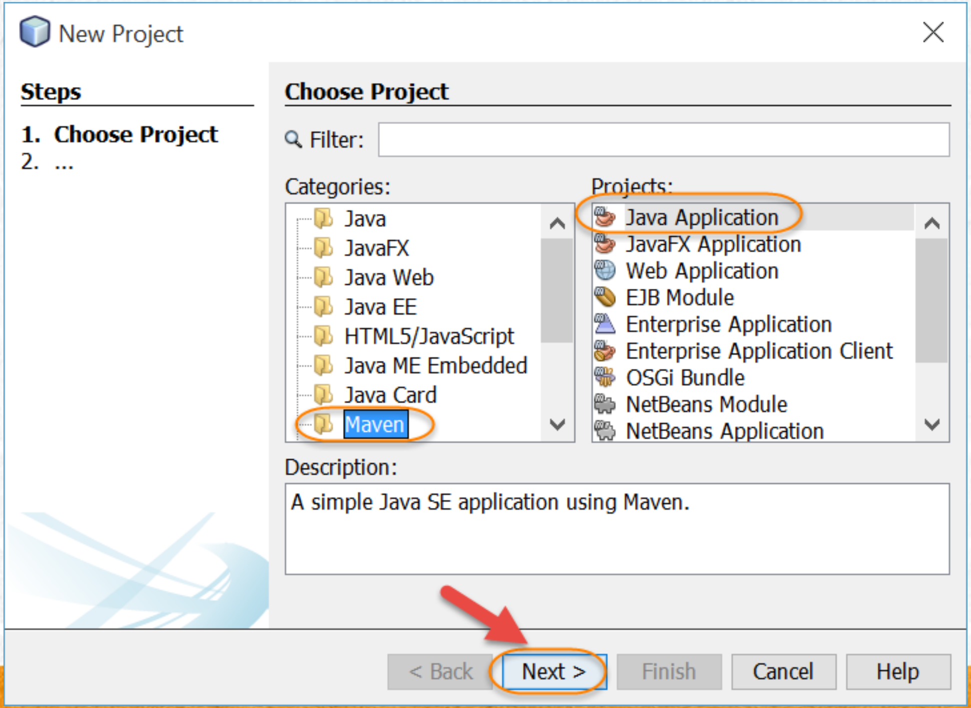
1. CREATE A NEW PROJECT

Create a Java Maven Project:


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
New Project... Ctrl+Mayús+N
New File... Ctrl+N
Open Project... Ctrl+Mayús+O
Open Recent Project
Close Project
Close Other Projects
Close All Projects

1. CREATE A NEW PROJECT

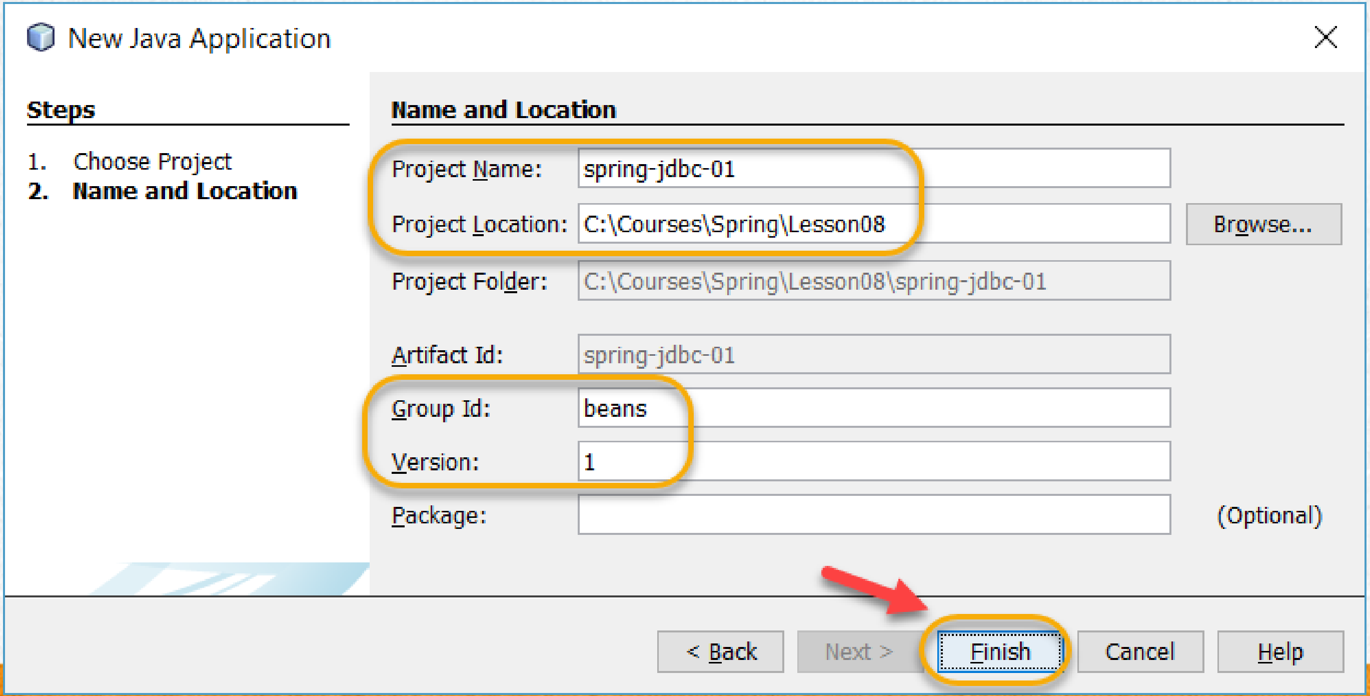
Create a Java Maven Project:



A simple Java SE application using Maven.

1. CREATE A NEW PROJECT

Create a new Java Maven Project:



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: spring-jdbc-01

Project Location: C:\Courses\Spring\Lesson08

Project Folder: C:\Courses\Spring\Lesson08\spring-jdbc-01

Artifact Id: spring-jdbc-01

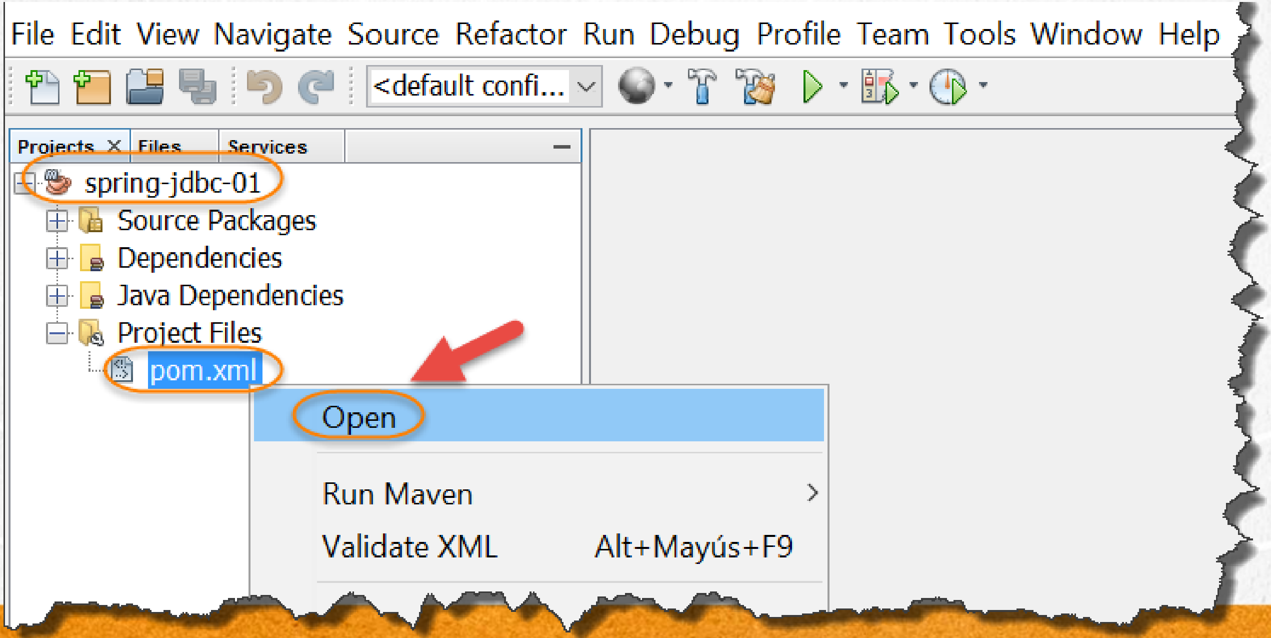
Group Id: beans

Version: 1

Package: (Optional)

2. MODIFY THE FILE

We open the file pom.xml and add the libraries that we will use:



3. MODIFY THE FILE

pom.xml:

Click to download

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>beans</groupId>
    <artifactId>spring-jdbc-01</artifactId>
    <version>1</version>
    <packaging>jar</packaging>
     <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
        <spring.version>5.1.0.RELEASE</spring.version>
        <log4j.version>2.11.1</log4j.version>
        <junit.version>5.3.1</junit.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-core</artifactId>
            <version>${spring.version}</version>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>${spring.version}</version>
        </dependency>
```

3. MODIFY THE FILE

pom.xml:

Click to download

```
        <dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-api</artifactId>
            <version>${log4j.version}</version>
        </dependency>
        <dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-core</artifactId>
            <version>${log4j.version}</version>
        </dependency>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter-api</artifactId>
            <version>${junit.version}</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter-engine</artifactId>
            <version>${junit.version}</version>
            <scope>test</scope>
        </dependency>
```

3. MODIFY THE FILE

pom.xml:

Click to download

```
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-test</artifactId>
            <version>${spring.version}</version>
            <scope>test</scope>
            <type>jar</type>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-aop</artifactId>
            <version>${spring.version}</version>
        </dependency>
        <dependency>
            <groupId>org.aspectj</groupId>
            <artifactId>aspectjweaver</artifactId>
            <version>1.9.1</version>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-jdbc</artifactId>
            <version>${spring.version}</version>
        </dependency>
```

3. MODIFY THE FILE

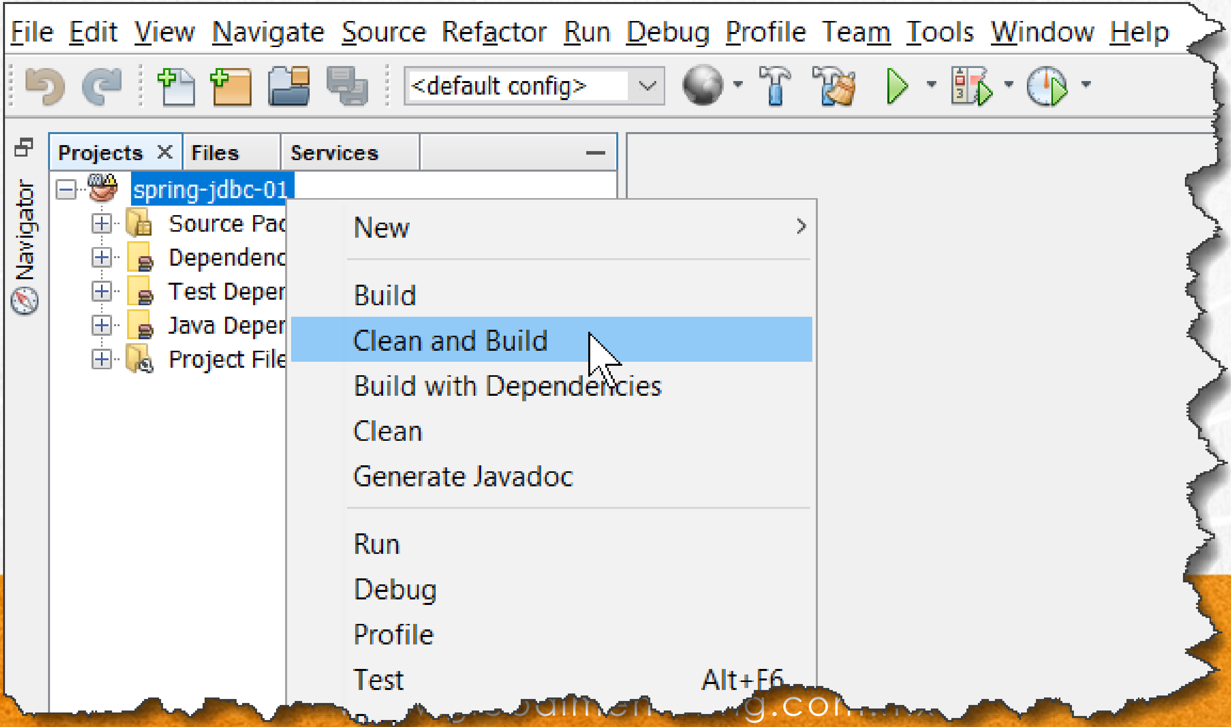
pom.xml:

Click to download

```
        <!-- DataSource -->
        <dependency>
            <groupId>org.apache.commons</groupId>
            <artifactId>commons-dbcp2</artifactId>
            <version>2.5.0</version>
        </dependency>
        <dependency>
            <groupId>com.h2database</groupId>
            <artifactId>h2</artifactId>
            <version>1.4.197</version>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-plugin</artifactId>
                <version>2.22.0</version>
            </plugin>
        </plugins>
    </build>
</project>
```

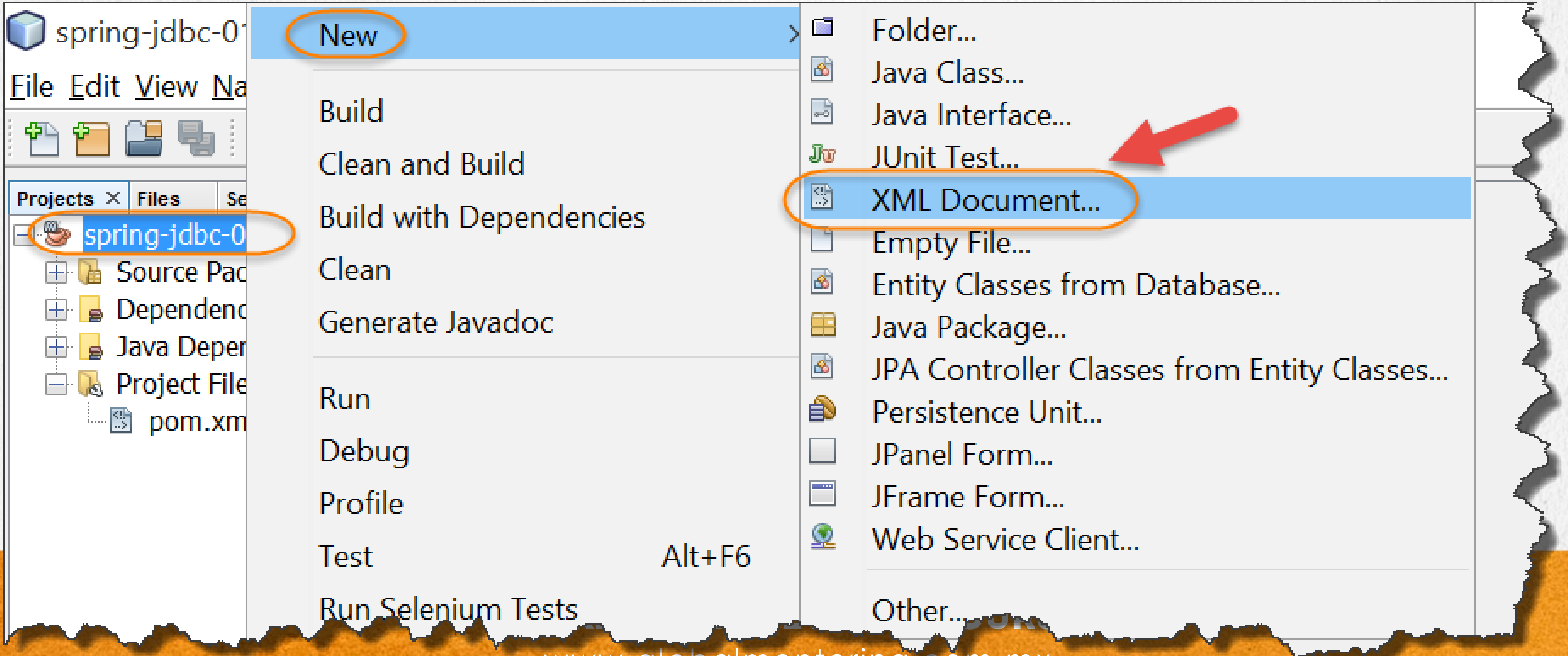
Paso 4. Execute Clean & Build

We do a clean & build to the project so that the libraries are downloaded if necessary:



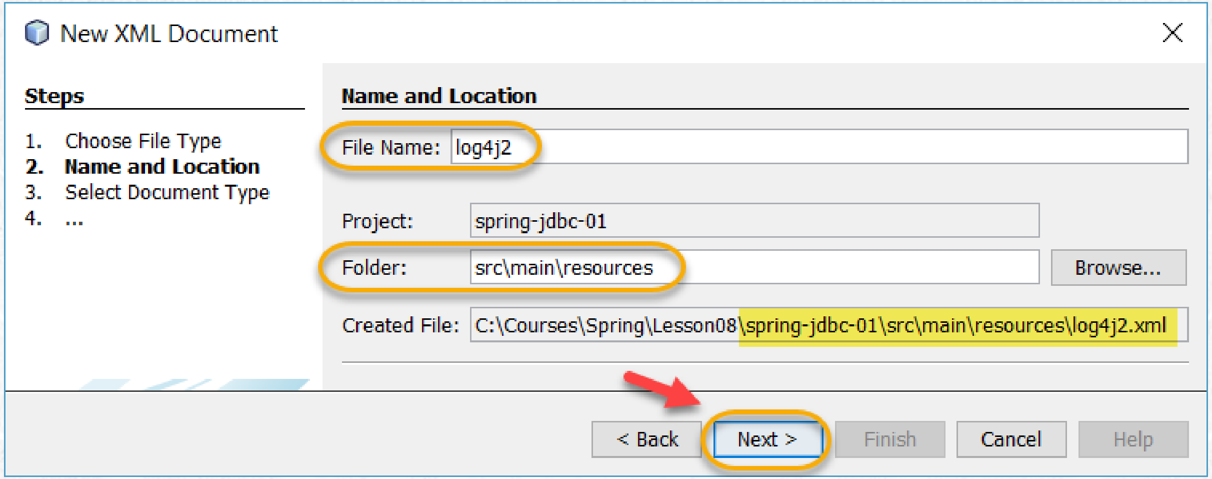
5. WE CREATE AN XML FILE

We create the log4j2.xml file:



5. CREATE AN XML FILE

We create the log4j2.xml file:



New XML Document

Steps

1. Choose File Type
2. **Name and Location**
3. Select Document Type
4. ...

Name and Location

File Name: log4j2

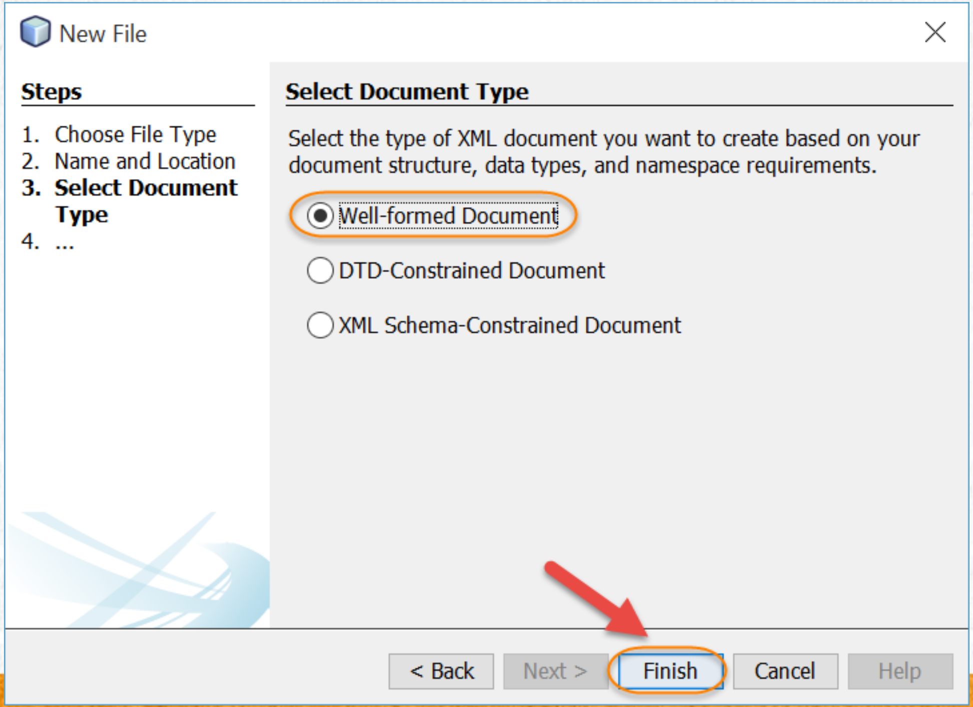
Project: spring-jdbc-01

Folder: src\main\resources

Created File: C:\Courses\Spring\Lesson08\spring-jdbc-01\src\main\resources\log4j2.xml

5. CREATE AN XML FILE

We create the log4j2.xml file:



New File

Steps

1. Choose File Type
2. Name and Location
3. **Select Document Type**
4. ...

Select Document Type

Select the type of XML document you want to create based on your document structure, data types, and namespace requirements.

● Well-formed Document

○ DTD-Constrained Document

○ XML Schema-Constrained Document

6. MODIFY THE FILE

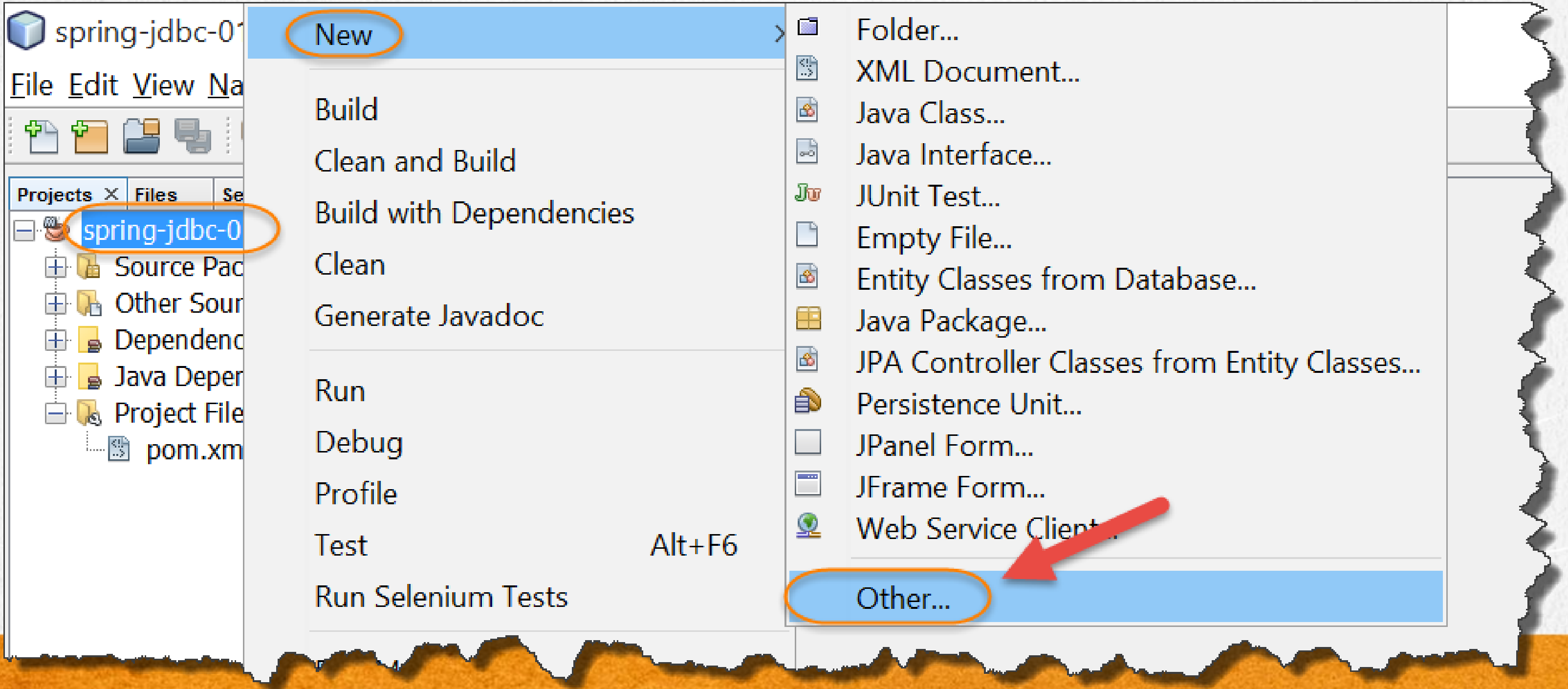
log4j2.xml:

Click to download

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="INFO">
    <Appenders>
        <Console name="Console" target="SYSTEM_OUT">
            <PatternLayout pattern="%d{HH:mm:ss} [%t] %-5level %logger{36} - %msg%n" />
        </Console>
    </Appenders>
    <Loggers>
        <Root level="info">
            <AppenderRef ref="Console" />
        </Root>
    </Loggers>
</Configuration>
```

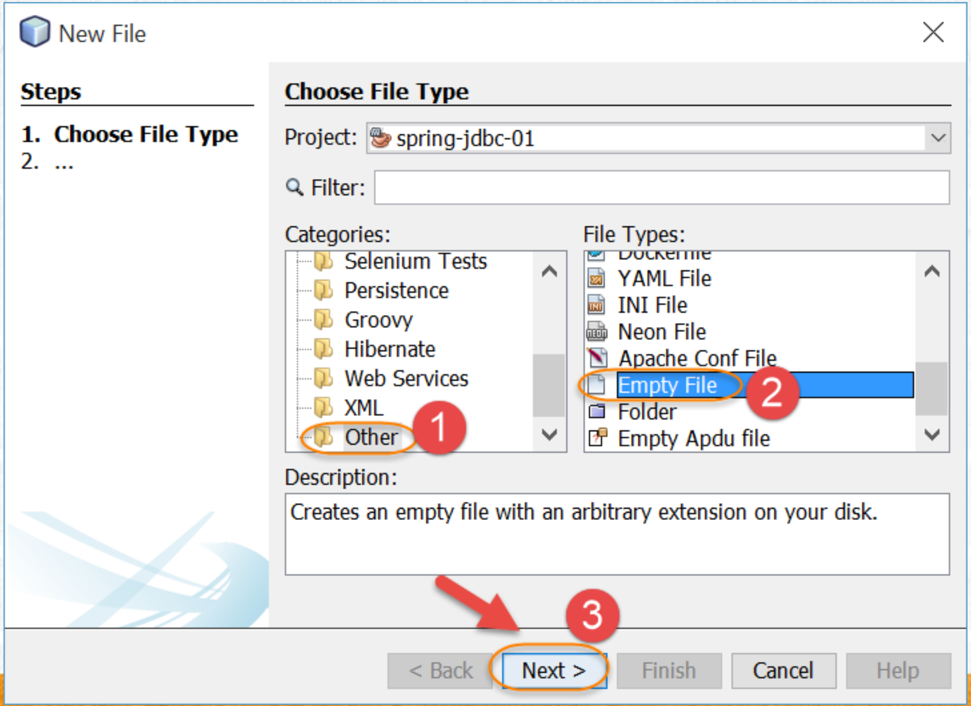
7. CREATE THE SQL FILE

We create the schema.sql file :



7. CREATE THE SQL FILE

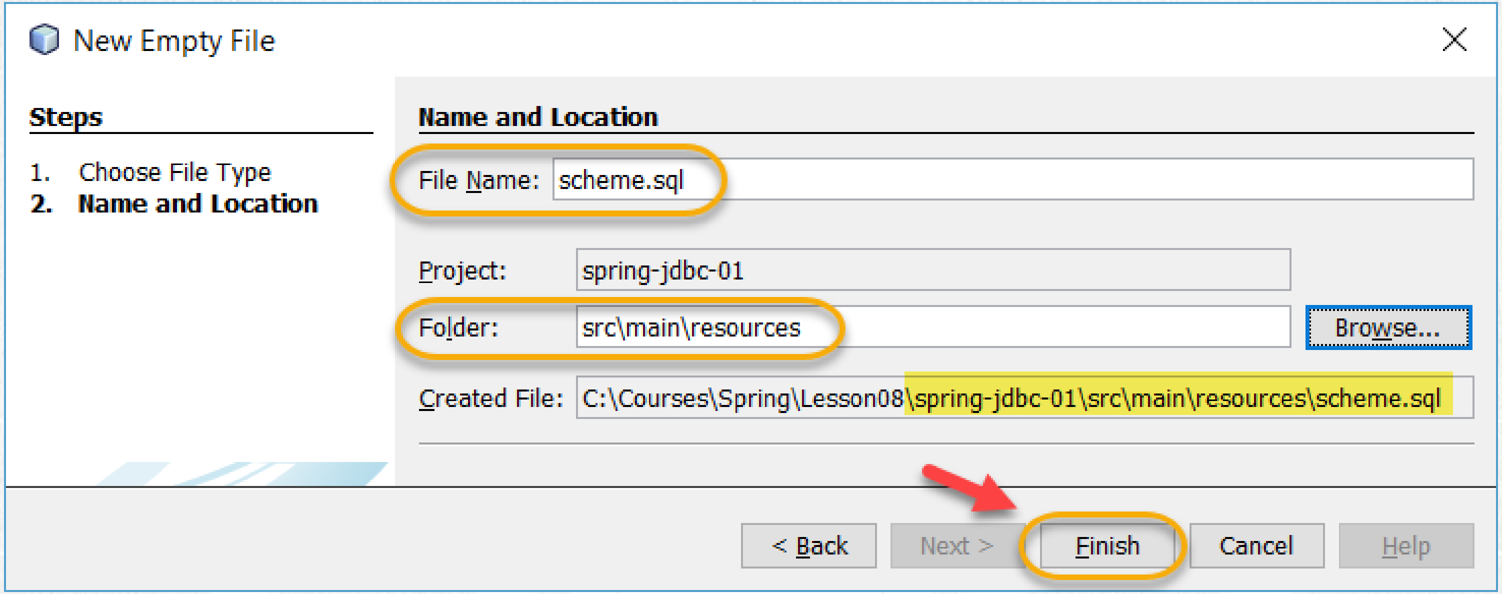
We create the schema.sql file :



Creates an empty file with an arbitrary extension on your disk.

7. CREATE THE SQL FILE

We create the schema.sql file :



New Empty File

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name: scheme.sql

Project: spring-jdbc-01

Folder: src\main\resources

Created File: C:\Courses\Spring\Lesson08\spring-jdbc-01\src\main\resources\scheme.sql

8. MODIFY THE FILE

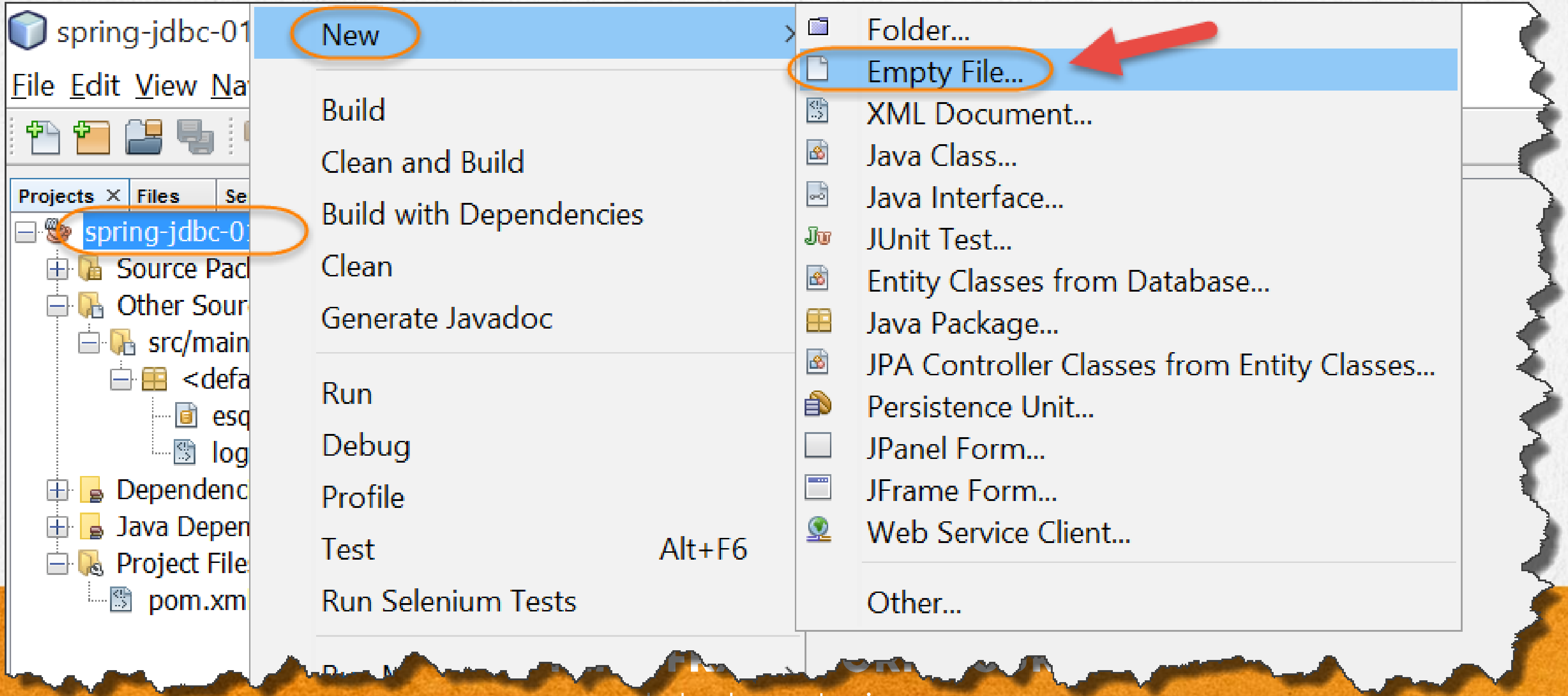
scheme.sql:

Click to download

```
drop table if exists person;

create table person (
    id_person identity primary key,
    name varchar(50) not null,
    email varchar(50) not null
);
```

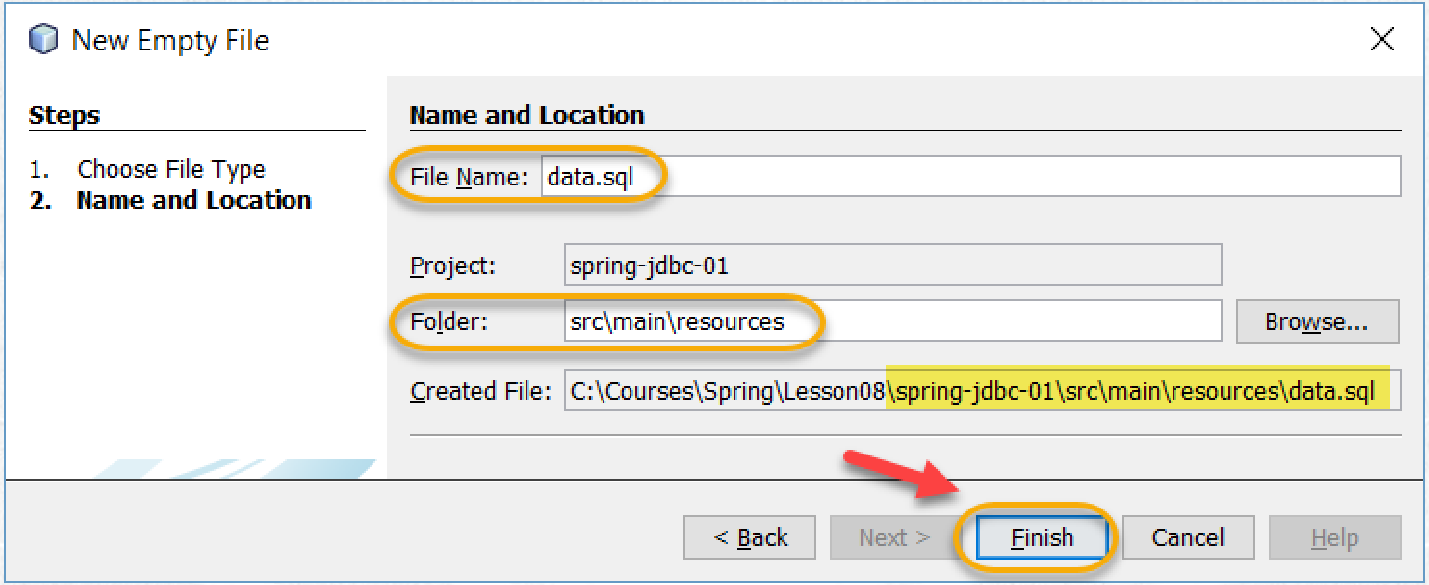
8. CREATE THE SQL FILE

We create the file data.sql:



8. CREATE THE SQL FILE

We create the file datos.sql:



New Empty File

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name: data.sql

Project: spring-jdbc-01

Folder: src\main\resources

Created File: C:\Courses\Spring\Lesson08\spring-jdbc-01\src\main\resources\data.sql

9. MODIFY THE FILE

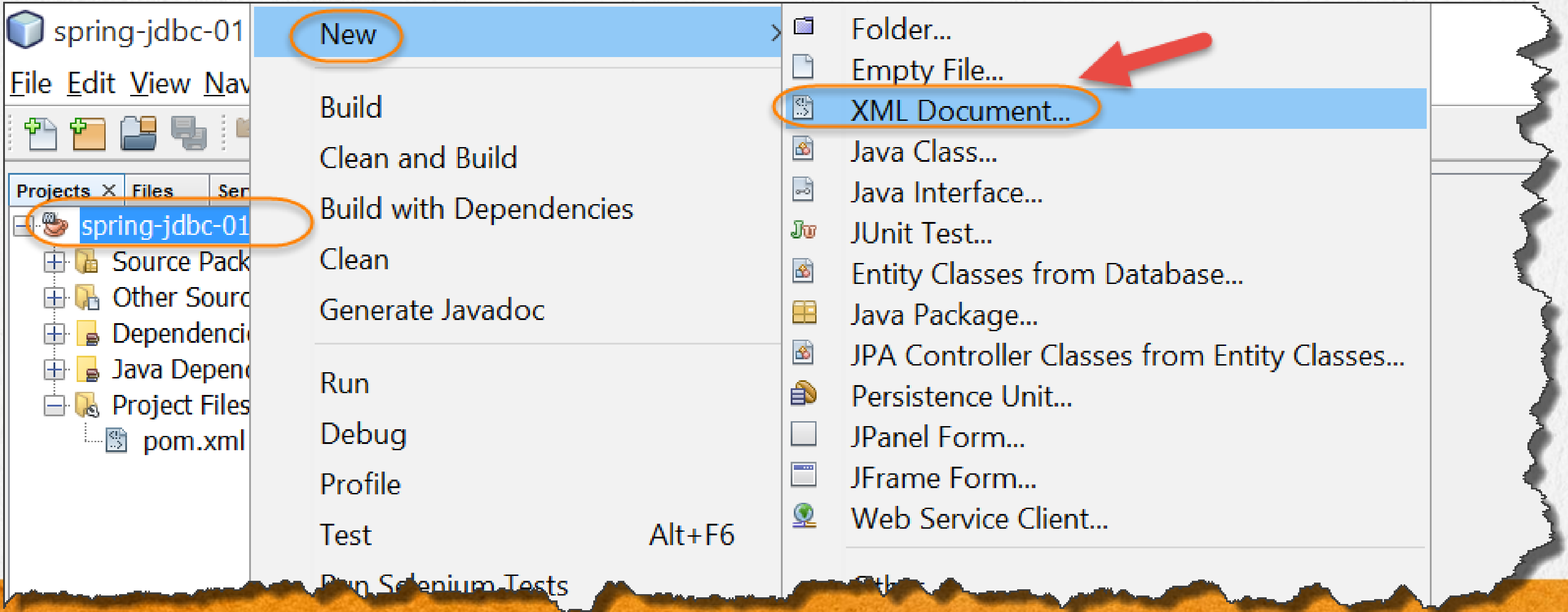
data.sql:

Click to download

```sql
insert into person (name, email) values ('Admin', 'admin@icursos.net');
insert into person (name, email) values ('Jhon', 'jsmith@mail.com');
insert into person (name, email) values ('Charly', 'ctyler@mail.com');
```

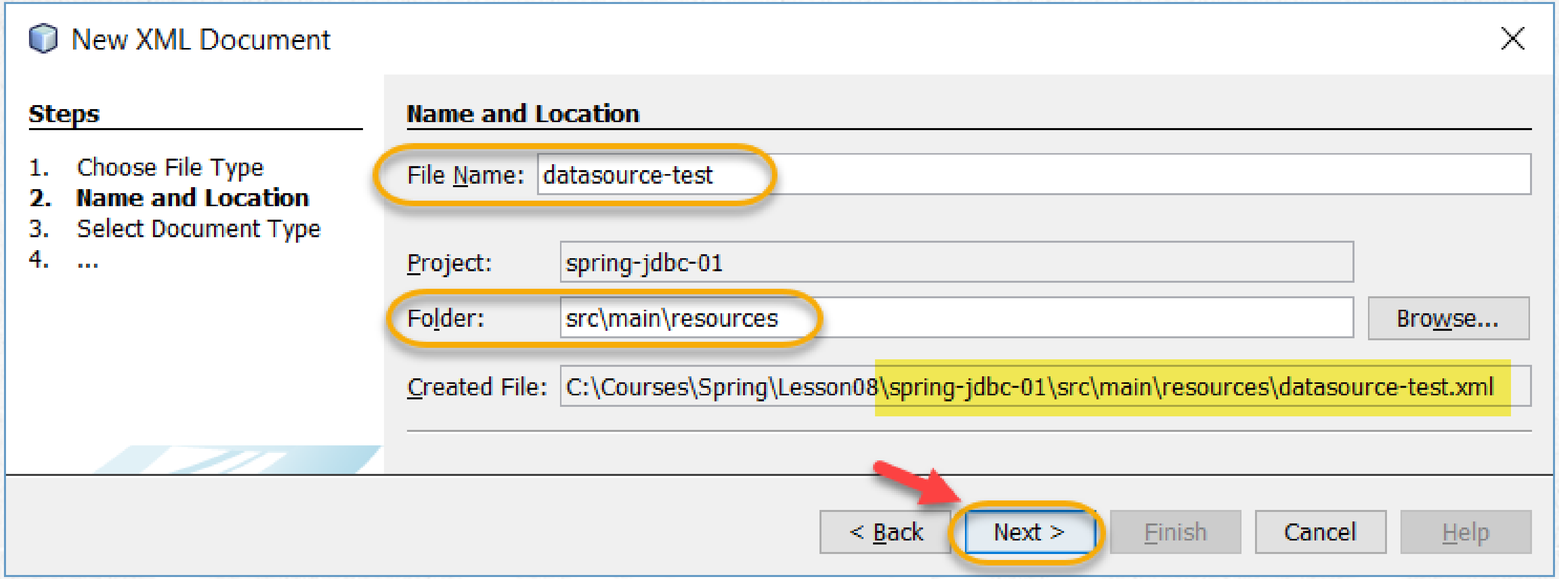
10. CREATE THE DATASOURCE FILE

We create the datasource-test.xml file:



10. CREATE THE DATASOURCE FILE

We create the datasource-test.xml file:



New XML Document

Steps

1. Choose File Type
2. **Name and Location**
3. Select Document Type
4. ...

Name and Location

File Name: datasource-test

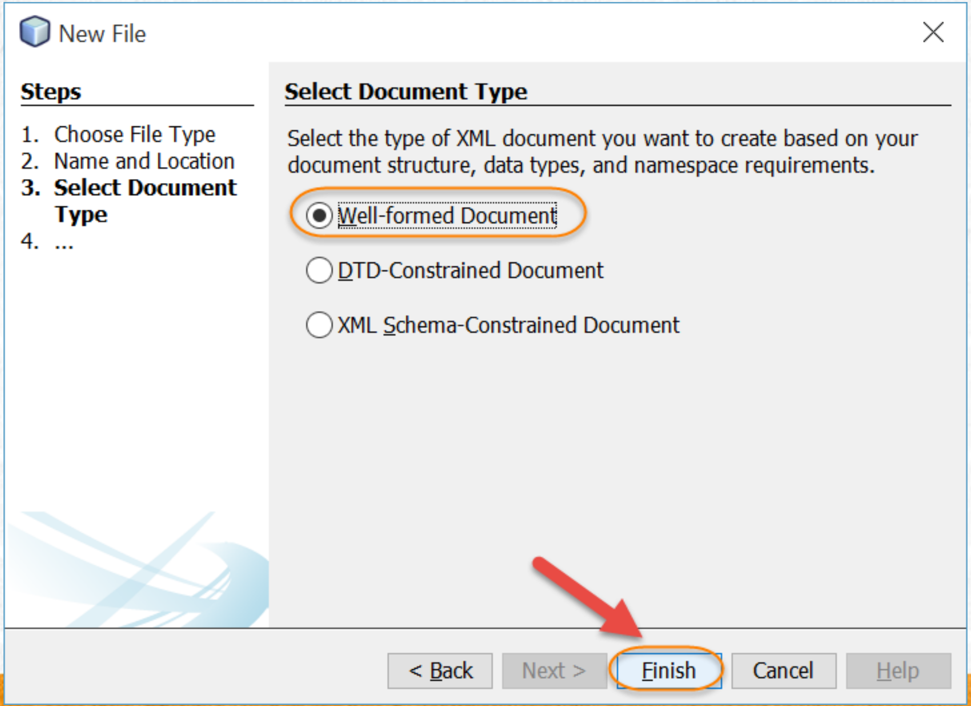
Project: spring-jdbc-01

Folder: src\main\resources

Created File: C:\Courses\Spring\Lesson08\spring-jdbc-01\src\main\resources\datasource-test.xml

10. CREATE THE DATASOURCE FILE

We create the datasource-test.xml file:



New File

Steps

1. Choose File Type
2. Name and Location
3. **Select Document Type**
4. ...

Select Document Type

Select the type of XML document you want to create based on your document structure, data types, and namespace requirements.

● Well-formed Document

○ DTD-Constrained Document

○ XML Schema-Constrained Document

11. MODIFY THE CODE

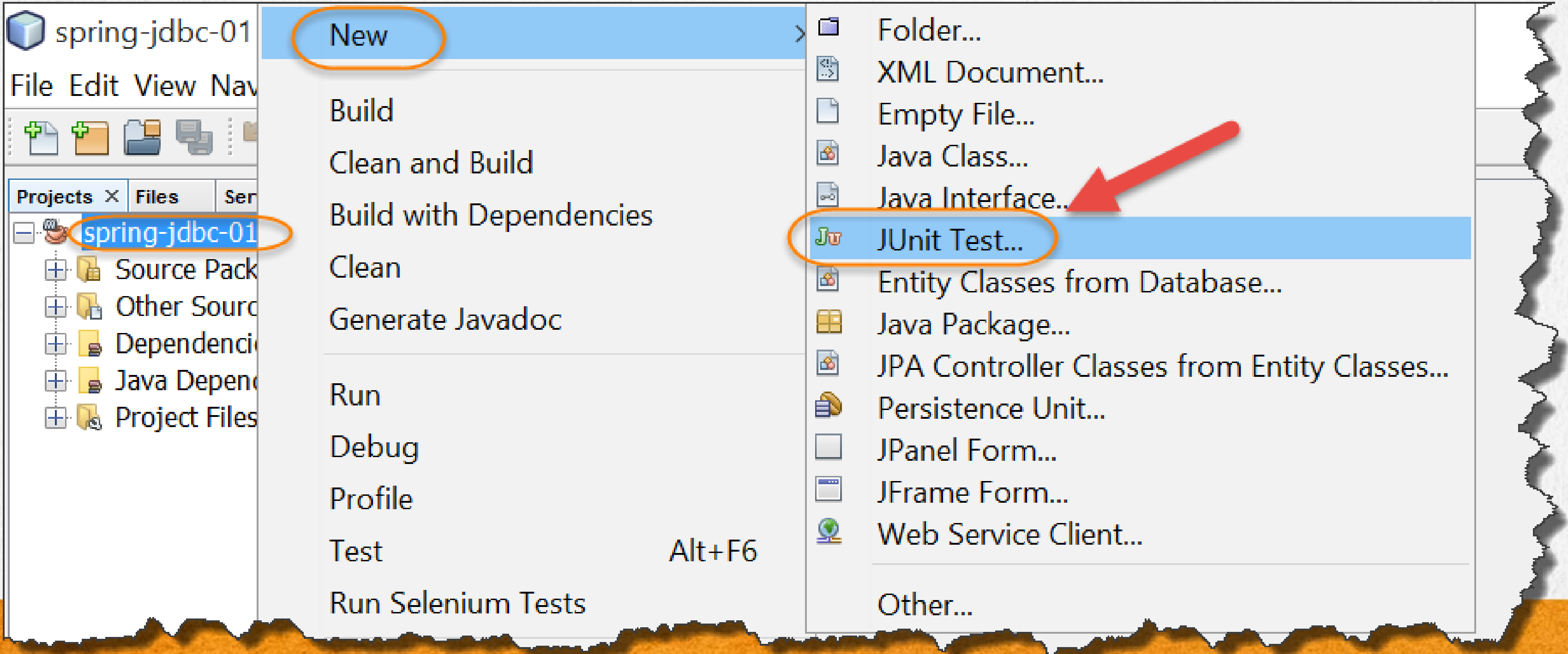
datasource-test.xml:

Click to download

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:jdbc="http://www.springframework.org/schema/jdbc"
       xsi:schemaLocation="http://www.springframework.org/schema/jdbc
                http://www.springframework.org/schema/jdbc/spring-jdbc.xsd
                http://www.springframework.org/schema/beans
                http://www.springframework.org/schema/beans/spring-beans.xsd">

    <jdbc:embedded-database id="dataSource" type="H2">
        <jdbc:script location="classpath:scheme.sql" />
        <jdbc:script location="classpath:data.sql" />
    </jdbc:embedded-database>

    <bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
        <constructor-arg ref="dataSource" />
    </bean>
</beans>
```

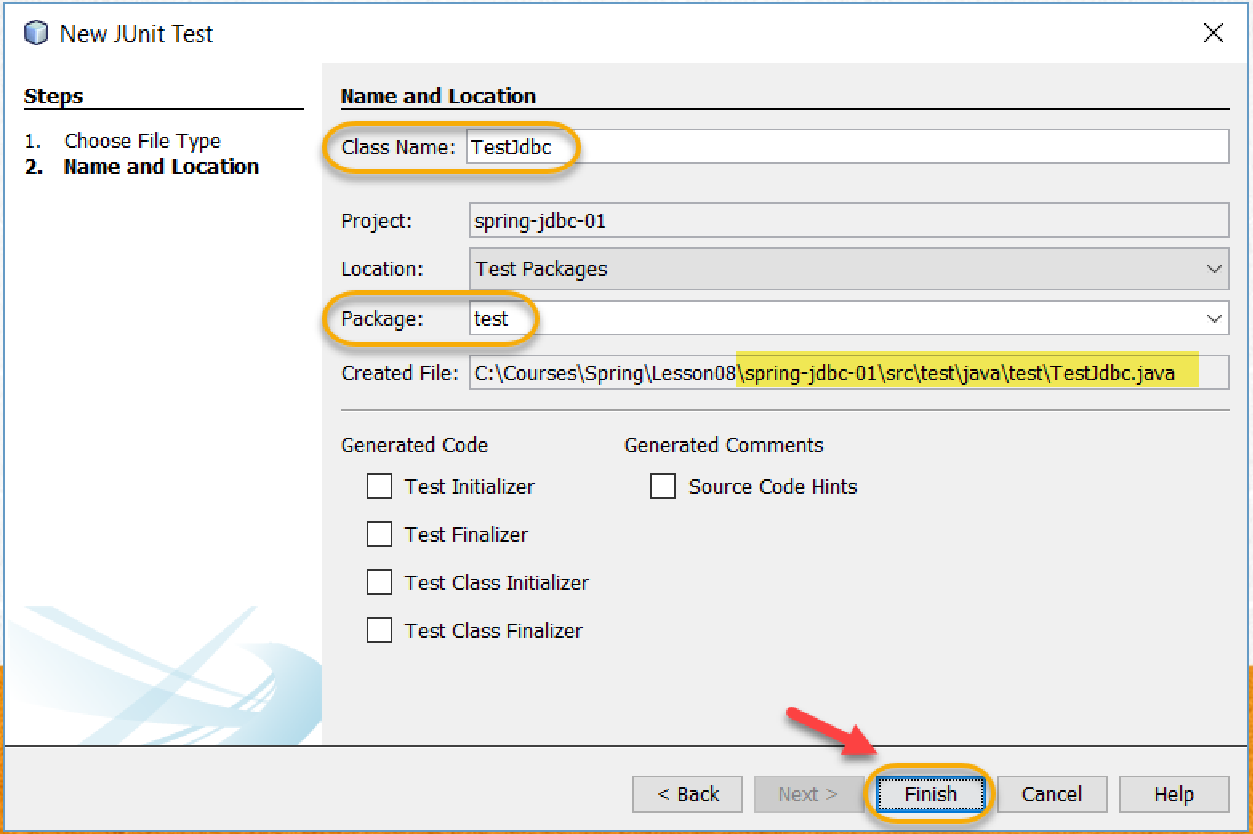
12. CREATE A NEW CLASS

We create a unit test class TestJdbc.java:



12. CREATE A NEW CLASS

We create a unit test class TestJdbc.java:



New JUnit Test

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: TestJdbc

Project: spring-jdbc-01

Location: Test Packages

Package: test

Created File: C:\Courses\Spring\Lesson08\spring-jdbc-01\src\test\java\test\TestJdbc.java

Generated Code

- ☐ Test Initializer
- ☐ Test Finalizer
- ☐ Test Class Initializer
- ☐ Test Class Finalizer

Generated Comments

- ☐ Source Code Hints

13. MODIFY THE FILE

TestJdbc.java:

Click to download

```
package test;

import org.apache.logging.log4j.*;
import static org.junit.jupiter.api.Assertions.assertEquals;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.ExtendWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit.jupiter.SpringExtension;

@ExtendWith(SpringExtension.class)
@ContextConfiguration(locations = {"classpath:datasource-test.xml"})
public class TestJdbc {

    private final Logger logger = LogManager.getRootLogger();

    @Autowired
    JdbcTemplate jdbcTemplate;
```

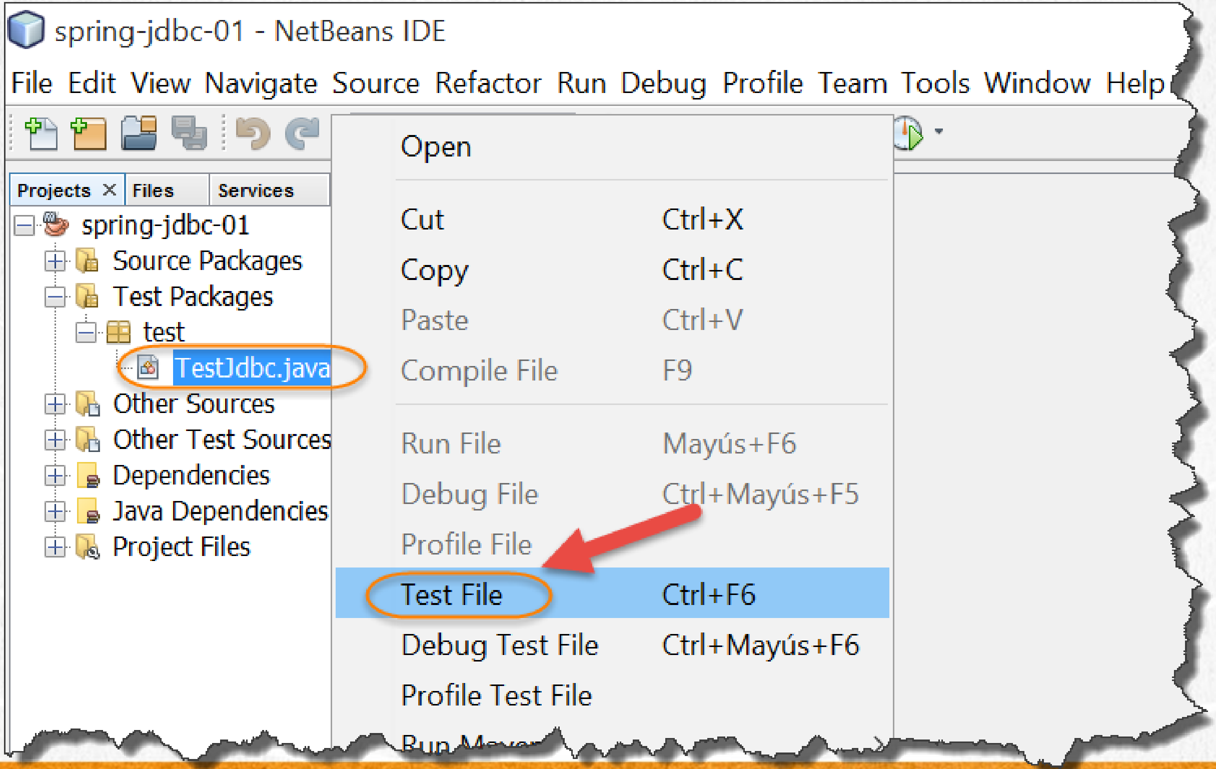
13. MODIFY THE FILE

TestJdbc.java:

Click to download

```
    @Test
    public void testJdbc() {
        logger.info("Start of the Jdbc test");
        String sql = "select count(*) from person";
        int noPersonas = jdbcTemplate.queryForObject(sql, Integer.class);
        logger.info("Number of people:" + noPersonas);
        assertEquals(3, noPersonas);
        logger.info("End of the Jdbc test");
    }
}
```

14. EXECUTE THE PROJECT

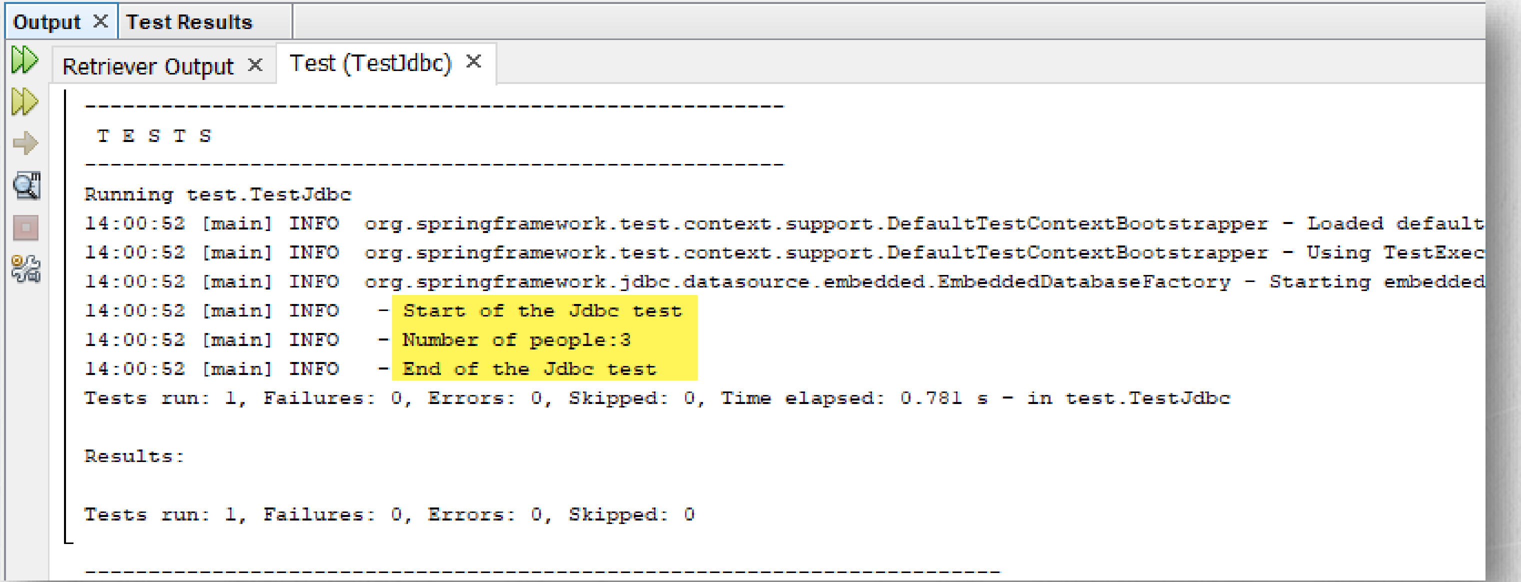


SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

14. EXECUTE THE PROJECT

The output of the project is as follows:



```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running test.TestJdbc
14:00:52 [main] INFO  org.springframework.test.context.support.DefaultTestContextBootstrapper - Loaded default
14:00:52 [main] INFO  org.springframework.test.context.support.DefaultTestContextBootstrapper - Using TestExec
14:00:52 [main] INFO  org.springframework.jdbc.datasource.embedded.EmbeddedDatabaseFactory - Starting embedded
14:00:52 [main] INFO   - Start of the Jdbc test
14:00:52 [main] INFO   - Number of people:3
14:00:52 [main] INFO   - End of the Jdbc test
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.781 s - in test.TestJdbc

Results:

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
```

EXERCISE CONCLUSION

With this exercise put into practice the concept of JDBC with Spring.

We use a test database called H2. This simplifies the process of doing the exercises and testing with the Spring JDBC API. However it is very easy to make the change to a database like MySql or any other database engine of our choice, since the configuration is practically the same.

ONLINE COURSE

SPRING FRAMEWORK

By: Eng. Ubaldo Acosta

Global Mentoring

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx