

SPRING FRAMEWORK COURSE

EXERCISE

TRANSACTION MANAGEMENT WITH SPRING FRAMEWORK



SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

Create a project to implement the handling of transactions in Spring. At the end we should observe the following:

```
-----
Running test.TestPersonService
17:26:05 [main] INFO org.springframework.test.context.support.DefaultTestContextBootstrapper - Loaded default TestExecutionListener class names from location [META-INF]
17:26:05 [main] INFO org.springframework.test.context.support.DefaultTestContextBootstrapper - Using TestExecutionListeners: [org.springframework.test.context.support]
17:26:05 [main] INFO org.springframework.jdbc.datasource.embedded.EmbeddedDatabaseFactory - Starting embedded database: url='jdbc:h2:mem:dataSource;DB_CLOSE_DELAY=-1'
17:26:06 [main] INFO org.springframework.test.context.transaction.TransactionContext - Began transaction (1) for test context [DefaultTestContext@10c626be testClass = test.TestPersonService]

17:26:06 [main] INFO - Start of test testTransactions
17:26:06 [main] INFO - Person to add: Person{idPerson=0, name=Samantha, email=samantha@mail.com}
17:26:06 [main] INFO - Modify Person : Person{idPerson=1, name=Admin, email=administrator@mail.com}
17:26:06 [main] INFO - List People:
17:26:06 [main] INFO - Person: Person{idPerson=1, name=Admin, email=administrator@mail.com}
17:26:06 [main] INFO - Person: Person{idPerson=2, name=Jhon, email=jsmith@mail.com}
17:26:06 [main] INFO - Person: Person{idPerson=3, name=Charly, email=ctyler@mail.com}
17:26:06 [main] INFO - Person: Person{idPerson=4, name=Samantha, email=samantha@mail.com}
17:26:06 [main] INFO - End of test testTransactions

17:26:06 [main] INFO org.springframework.test.context.transaction.TransactionContext - Committed transaction for test: [DefaultTestContext@10c626be testClass = test.TestPersonService]
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.017 s - in test.TestPersonService

Results:

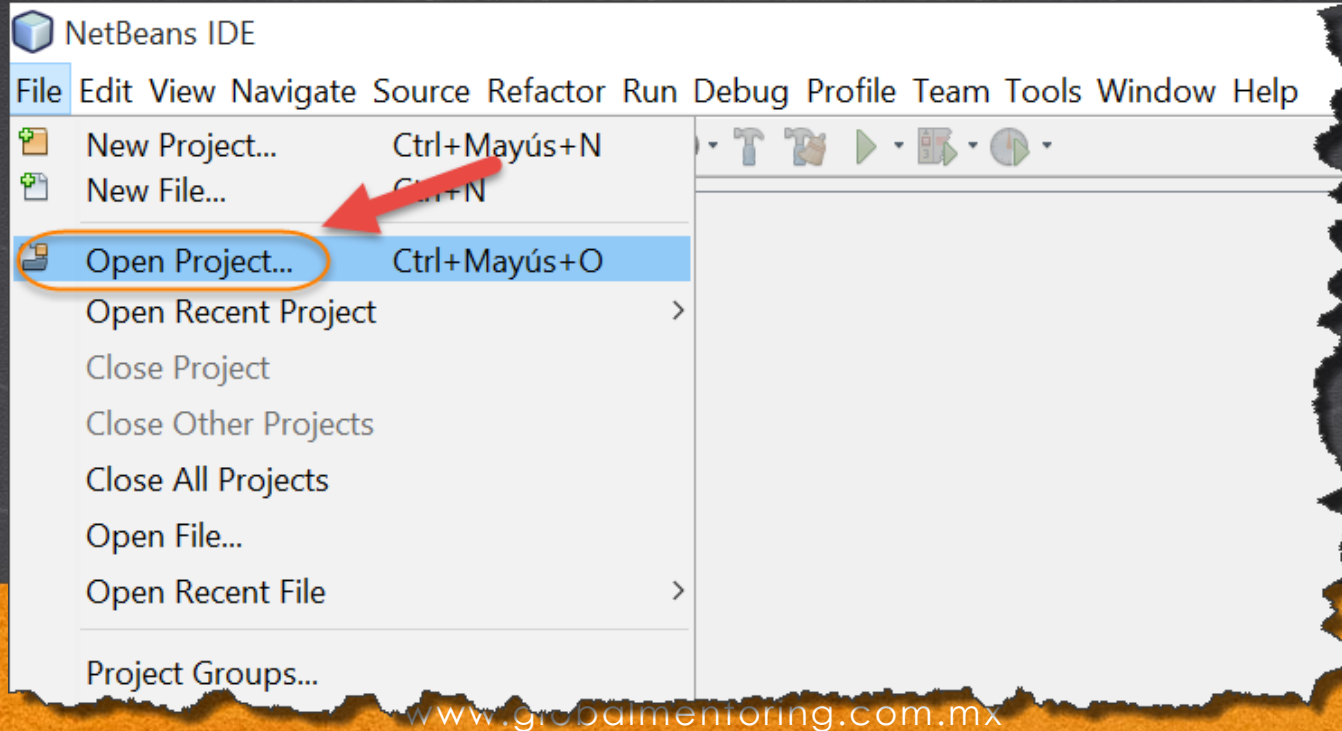
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
```

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

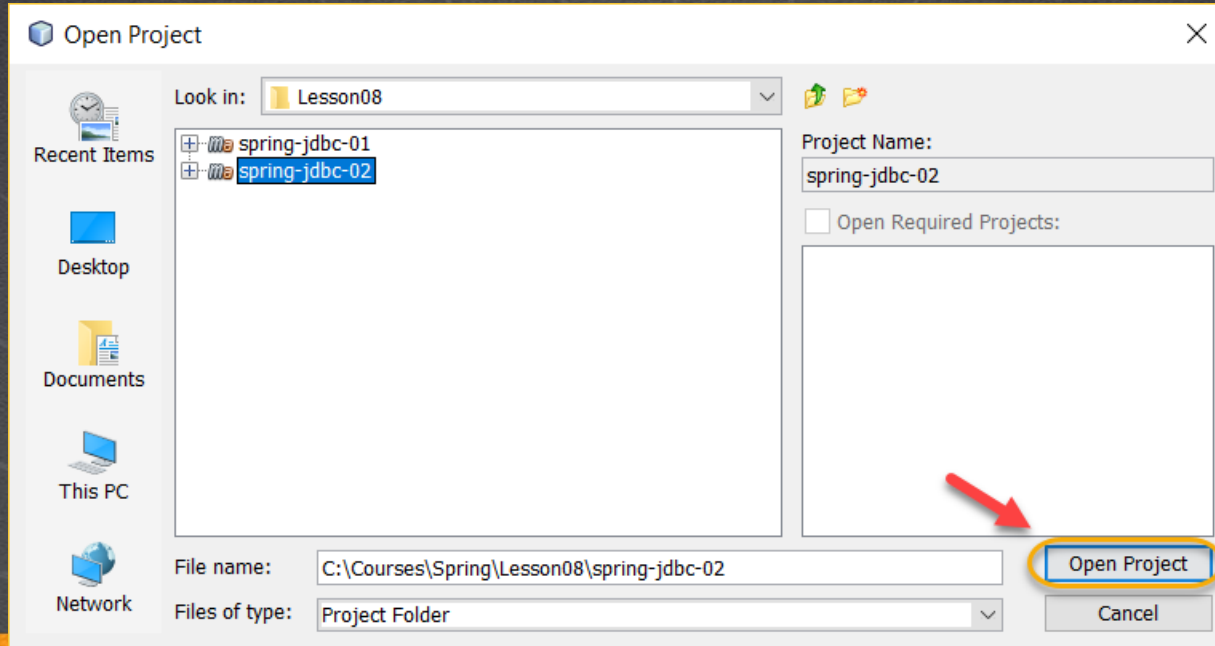
1. OPEN PROJECT

In case we do not have open the project spring-jdbc-02 we open it:



1. OPEN THE PROJECT

In case we do not have open the project spring-jdbc-02 we open it:

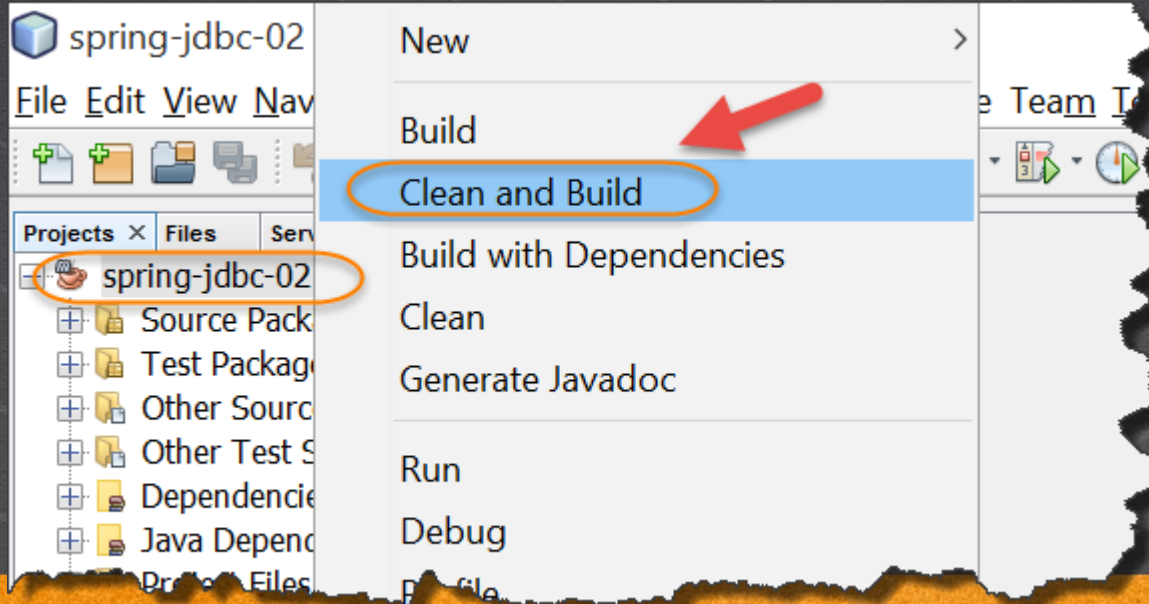


SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

1. OPEN THE PROJECT

We wait for you to fully load the project. In case the project makes a mistake, we make a Clean & Build so that all the files are shown, this step is optional:

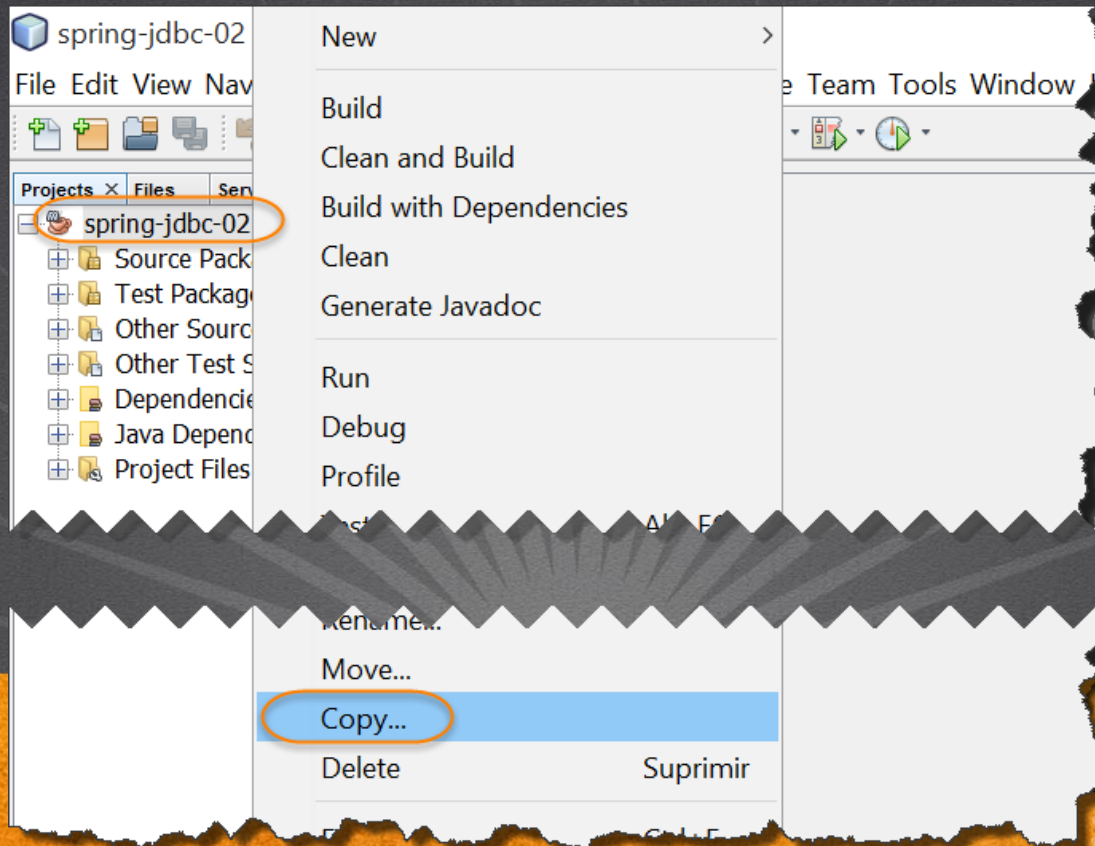


SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

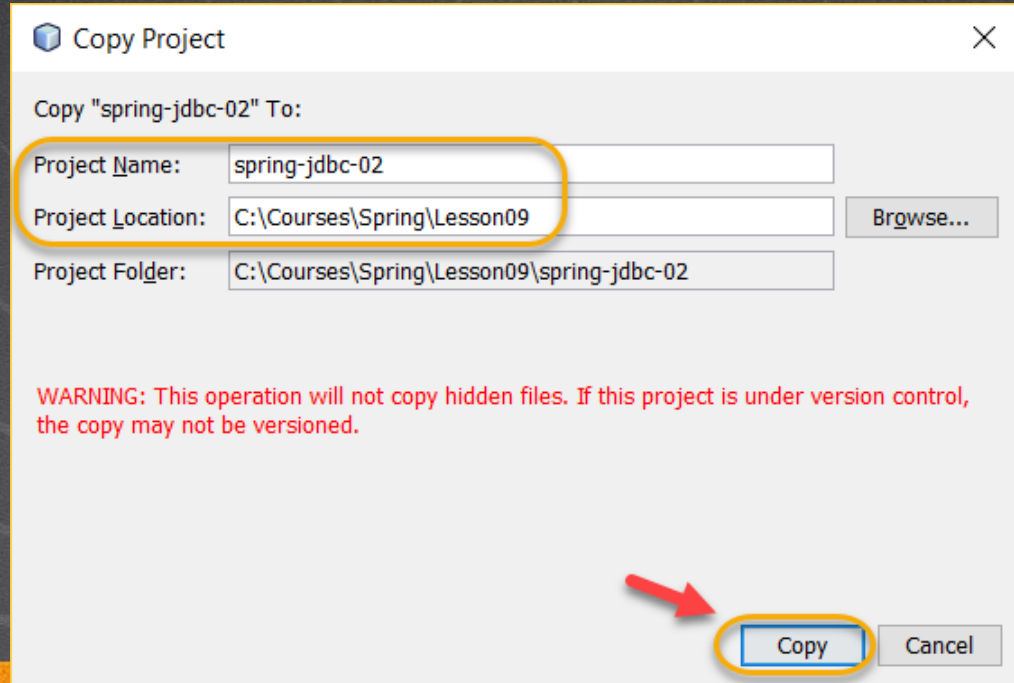
2. COPY THE PROJECT

Copy the project:



2. COPY THE PROJECT

We copy the project to put it in the new route:

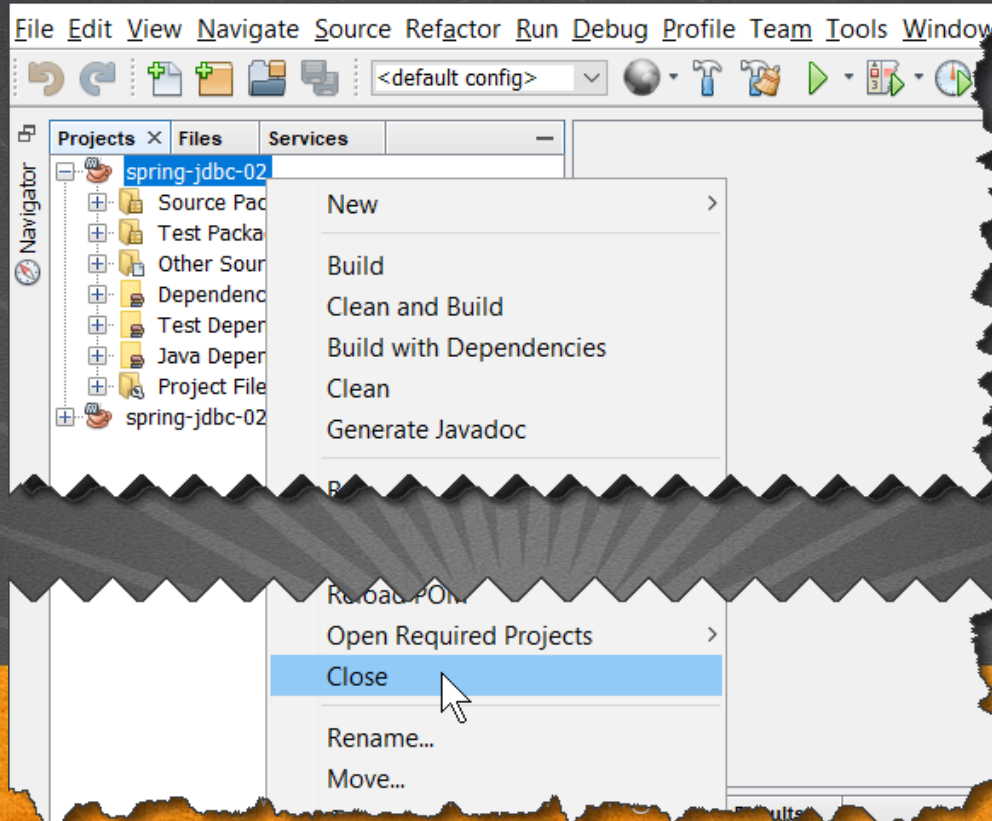


SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

3. CLOSE THE PROJECT

We closed the previous project and left only the new one:



4. MODIFY THE XML FILE

Modify the applicationContext.xml file by adding:

The namespace of tx (transactional handling)

`<tx:annotation-driven/>` (to enable transactional handling)

`<context:component-scan base-package="service" />`



SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

4. MODIFY THE CODE

applicationContext.xml:

Click to download

```
<?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:context="http://www.springframework.org/schema/context"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           http://www.springframework.org/schema/beans/spring-beans.xsd
                           http://www.springframework.org/schema/context
                           http://www.springframework.org/schema/context/spring-context.xsd
                           http://www.springframework.org/schema/tx
                           http://www.springframework.org/schema/tx/spring-tx.xsd">

    <context:component-scan base-package="jdbc" />

    <context:component-scan base-package="service" />

    <tx:annotation-driven/>

</beans>
```

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

5. MODIFY THE XML FILE

We modified the datasource-test.xml file adding the following for transactional handling:

```
<bean id="transactionManager"  
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">  
    <property name="dataSource" ref="dataSource" />  
</bean>
```


5. MODIFY THE CODE

datasource-test.xml:

Click to download

```
<?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>

    <bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
        <property name="dataSource" ref="dataSource" />
    </bean>

</beans>
```

6. MODIFY THE XML FILE

We modified the log4j2.xml file to add more jdbc information and transaction management to the Spring logger. We can do this by adding the following to our logging file. If we want to see even more detail, we can change the level to debug instead of info, as seen below:

```
<Loggers>
  <logger name="org.springframework.jdbc.core" level="info" additivity="false">
    <appender-ref ref="Console" />
  </logger>
  <logger name="org.springframework.transaction" level="info" additivity="false">
    <appender-ref ref="Console" />
  </logger>
  <Root level="info">
    <AppenderRef ref="Console" />
  </Root>
</Loggers>
```

6. MODIFY THE CODE

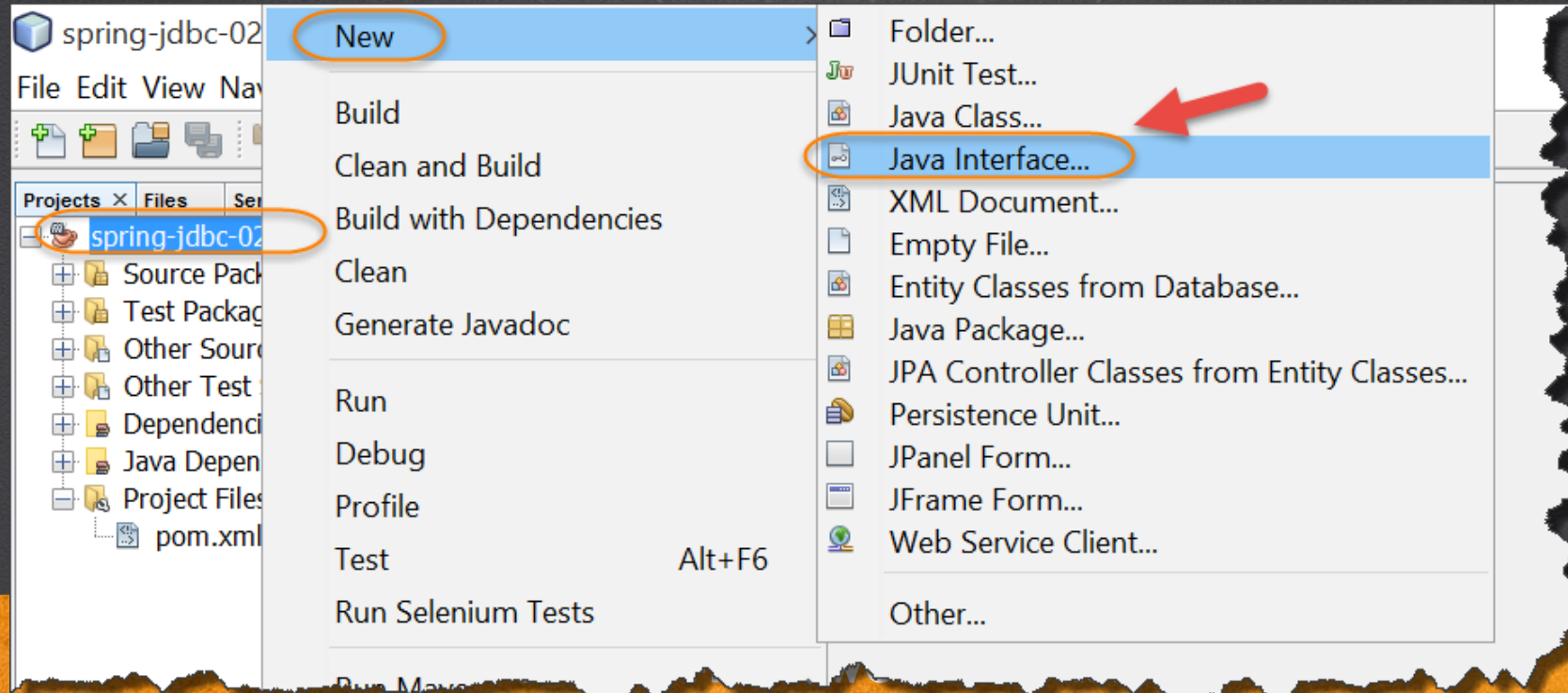
log4j2.xml:

Click to download

```
<?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>
    <logger name="org.springframework.jdbc.core" level="info" additivity="false">
      <appender-ref ref="Console" />
    </logger>
    <logger name="org.springframework.transaction" level="info" additivity="false">
      <appender-ref ref="Console" />
    </logger>
    <Root level="info">
      <AppenderRef ref="Console" />
    </Root>
  </Loggers>
</Configuration>
```


7. CREATE AN INTERFACE

We create a PersonService.java interface:



7. CREATE A NEW CLASS

We create a PersonService.java interface:

New Java Class

Steps

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

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

8. MODIFY THE CODE

PersonService.java:

[Click to download](#)

```
package service;

import java.util.List;
import jdbc.Person;

public interface PersonService {

    public List<Person> listPeople();

    public Person findPerson(Person person);

    public void addPerson(Person person);

    public void modifyPerson(Person person);

    public void deletePerson(Person person);

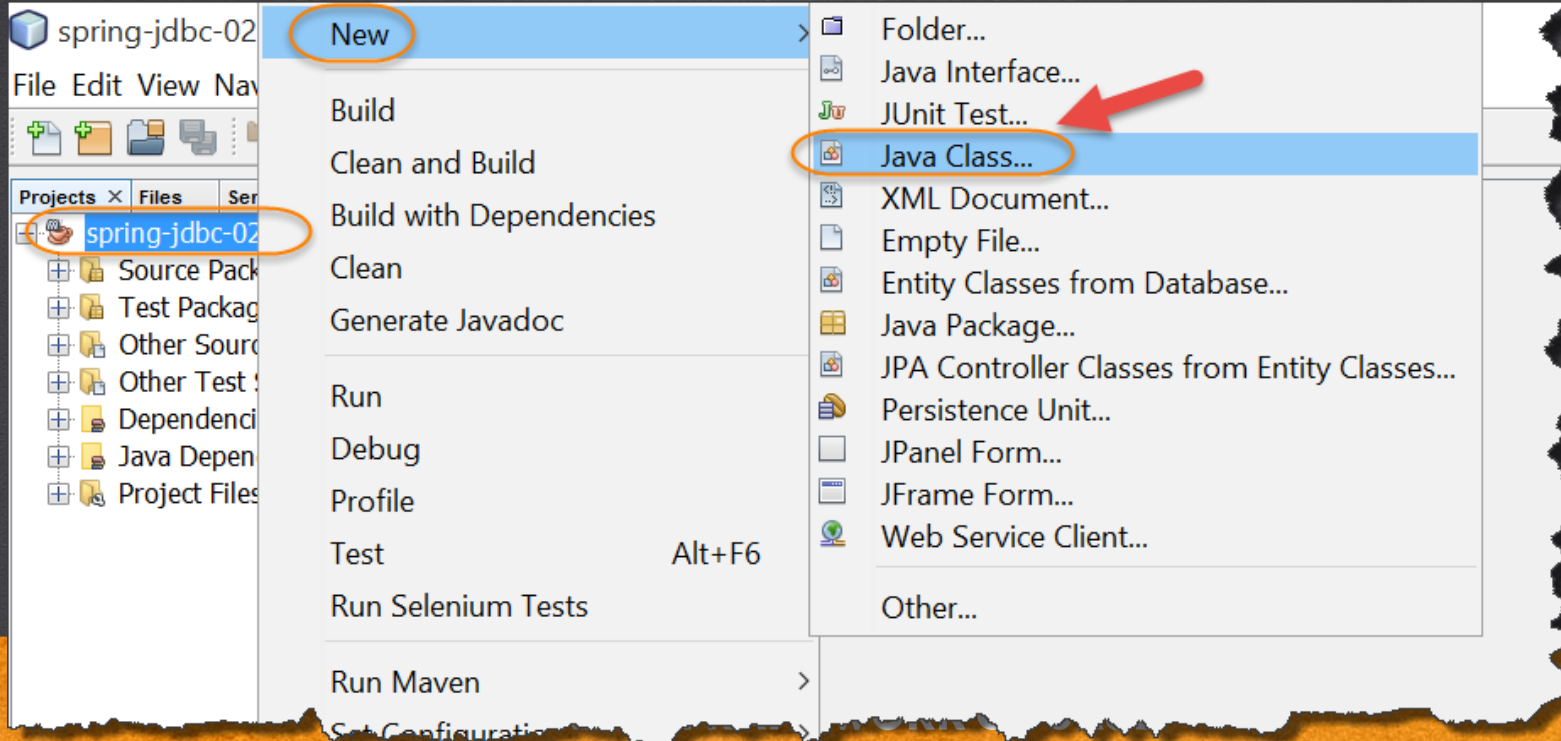
}
```

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

9. CREATE A NEW CLASS

We create a PersonServiceImpl.java class:



9. CREATE A NEW CLASS

We create a PersonServiceImpl.java class:

New Java Class

Steps

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

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

10. MODIFY THE CODE

PersonServiceImpl.java:

Click to download

```
package service;

import java.util.List;
import jdbc.*;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.*;

@Service
@Transactional(propagation=Propagation.SUPPORTS, readOnly=true)
public class PersonServiceImpl implements PersonService {

    @Autowired
    private PersonDao personDao;

    @Override
    public List<Person> listPeople() {
        return personDao.findAllPeople();
    }

    @Override
    public Person findPerson(Person person) {
        return personDao.findPersonById(person.getIdPerson());
    }
}
```


10. MODIFY THE CODE

PersonServiceImpl.java:

[Click to download](#)

```
@Transactional(propagation=Propagation.REQUIRED, readOnly=false)
@Override
public void addPerson(Person person) {
    personDao.insertPerson(person);
}

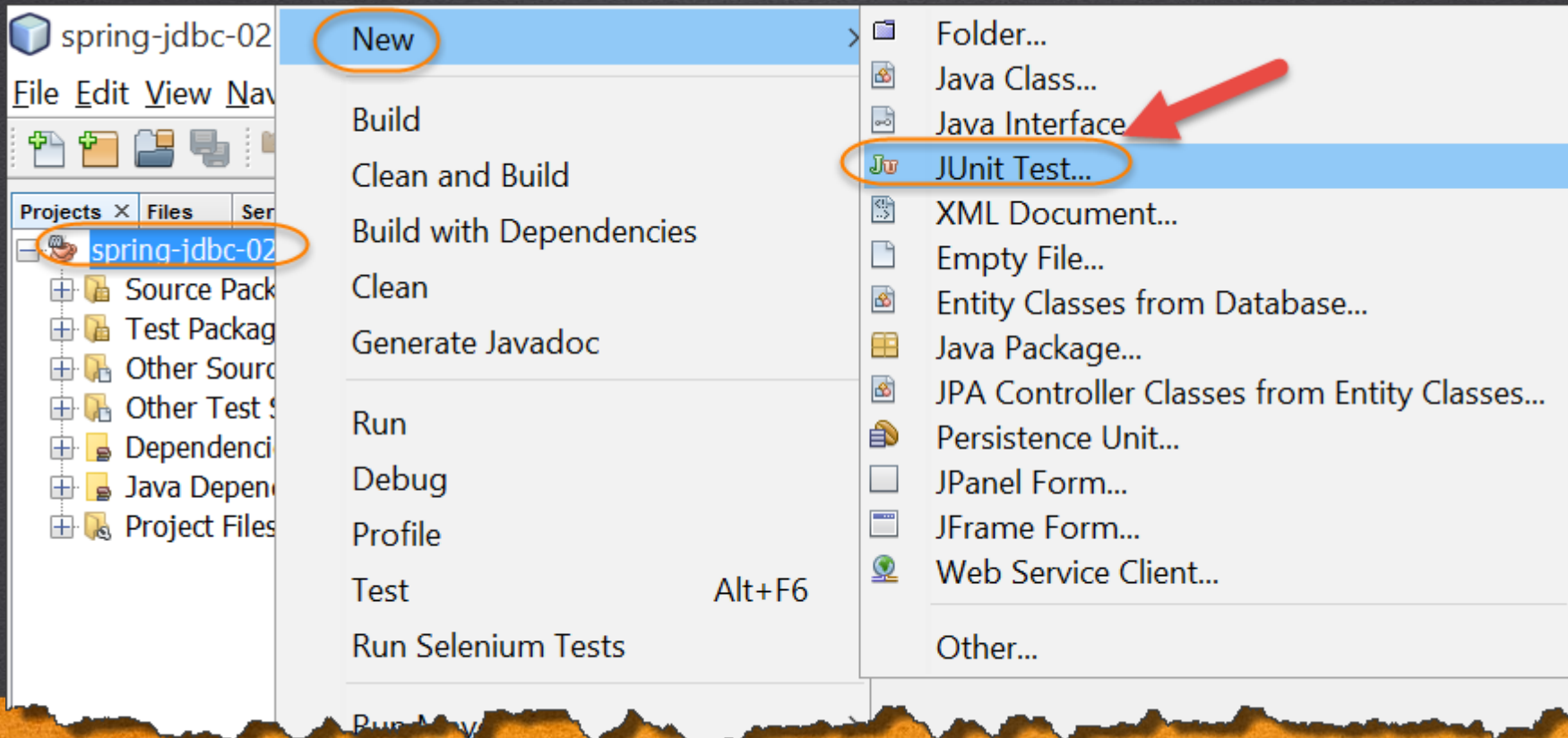
@Transactional(propagation=Propagation.REQUIRED, readOnly=false)
@Override
public void modifyPerson(Person person) {
    personDao.updatePerson(person);
}

@Transactional(propagation=Propagation.REQUIRED, readOnly=false)
@Override
public void deletePerson(Person person) {
    personDao.deletePerson(person);
}
}
```

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

11. CREATE A NEW CLASS



SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

11. CREATE A NEW CLASS

We create a unit test class TestPersonaServiceImpl.java:

New JUnit Test

Steps

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

Name and Location

Class Name: TestPersonService

Project: spring-jdbc-02

Location: Test Packages

Package: test

Created File: C:\Courses\Spring\Lesson09\spring-jdbc-02\src\test\java\test\TestPersonService.java

Generated Code

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

Generated Comments

- ☐ Source Code Hints

< Back Next > **Finish** Cancel Help

12. MODIFY THE CODE

TestPersonService.java:

[Click to download](#)

```
package test;

import java.util.List;
import jdbc.Person;
import jdbc.PersonDao;
import org.apache.logging.log4j.*;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.ExtendWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.test.annotation.Commit;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit.jupiter.SpringExtension;
import org.springframework.transaction.annotation.Transactional;
import service.PersonService;

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

    private final Logger logger = LogManager.getRootLogger();

    @Autowired
    private PersonService personService;
```

TestPersonService.java:

Click to download

[illegible]

12. MODIFY THE CODE

TestPersonService.java:

Click to download

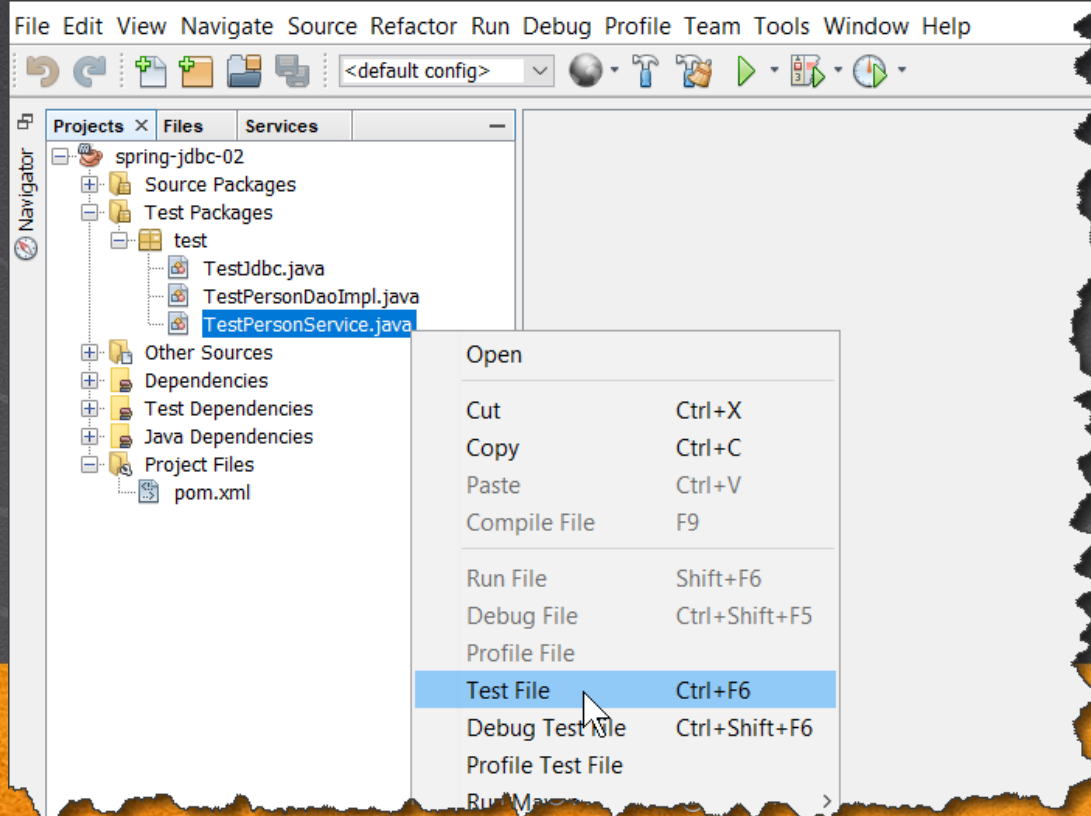
```
    } catch (Exception e) {  
        logger.error("Error Service: ", e);  
    }  
}  
  
private int displayPeople() {  
    List<Person> persons = personService.listPeople();  
    int counPeople = 0;  
    for (Person person : persons) {  
        logger.info("Person: " + person);  
        counPeople++;  
    }  
    return counPeople;  
}  
}
```

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

13. EXECUTE THE PROJECT

Execute the test:



13. EXECUTE THE PROJECT

We execute the application. The result is similar to the following:

```
Running test.TestPersonService
17:26:05 [main] INFO org.springframework.test.context.support.DefaultTestContextBootstrapper - Loaded default TestExecutionListener class names from location [META-
17:26:05 [main] INFO org.springframework.test.context.support.DefaultTestContextBootstrapper - Using TestExecutionListeners: [org.springframework.test.context.supp
17:26:05 [main] INFO org.springframework.jdbc.datasource.embedded.EmbeddedDatabaseFactory - Starting embedded database: url='jdbc:h2:mem:dataSource;DB_CLOSE_DELAY=
17:26:06 [main] INFO org.springframework.test.context.transaction.TransactionContext - Began transaction (1) for test context [DefaultTestContext@10c626be testClass=

17:26:06 [main] INFO - Start of test testTransactions
17:26:06 [main] INFO - Person to add:Person{idPerson=0, name=Samantha, email=samantha@mail.com}
17:26:06 [main] INFO - Modify Person : Person{idPerson=1, name=Admin, email=administrator@mail.com}
17:26:06 [main] INFO - List People:
17:26:06 [main] INFO - Person: Person{idPerson=1, name=Admin, email=administrator@mail.com}
17:26:06 [main] INFO - Person: Person{idPerson=2, name=Jhon, email=jsmith@mail.com}
17:26:06 [main] INFO - Person: Person{idPerson=3, name=Charly, email=ctyler@mail.com}
17:26:06 [main] INFO - Person: Person{idPerson=4, name=Samantha, email=samantha@mail.com}
17:26:06 [main] INFO - End of test testTransactions

17:26:06 [main] INFO org.springframework.test.context.transaction.TransactionContext - Committed transaction for test: [DefaultTestContext@10c626be testClass = Test
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.017 s - in test.TestPersonService

Results:

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

SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

13. EXECUTE THE PROJECT

If we enable the error (line 34) and comment on line 35, we observe the case when rollback is done:

```
17:29:29 [main] INFO org.springframework.test.context.support.DefaultTestContextBootstrapper - Loaded default TestExecutionListener class names from location [META-INF/spring.handlers]
17:29:29 [main] INFO org.springframework.test.context.support.DefaultTestContextBootstrapper - Using TestExecutionListeners: [org.springframework.test.context.support.DefaultTestExecutionListener, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DirtiesContextTestExecutionListener, org.springframework.test.context.support.MethodOutputTestExecutionListener, org.springframework.test.context.support.ResultAwareTestExecutionListener, org.springframework.test.context.support.TransactionAwareTestExecutionListener]
17:29:29 [main] INFO org.springframework.jdbc.datasource.embedded.EmbeddedDatabaseFactory - Starting embedded database: url='jdbc:h2:mem:dataSource;DB_CLOSE_DELAY=10000'
17:29:29 [main] INFO org.springframework.test.context.transaction.TransactionContext - Began transaction (1) for test context [DefaultTestContext@2f0cc3 testClass=org.springframework.test.context.transaction.TransactionalTestExecutionListener]

17:29:29 [main] INFO - Start of test testTransactions
17:29:30 [main] ERROR - Error Service:
org.springframework.dao.DataIntegrityViolationException: PreparedStatementCallback; SQL [INSERT INTO person (name, email) values (?, ?)]; Value too long for column 'NAME VARCHAR(50) NOT NULL': INSERT INTO person (name, email) values (?, ?) [22001-197]; nested exception is org.h2.jdbc.JdbcSQLException: Value too long for column "NAME VARCHAR(50) NOT NULL": INSERT INTO person (name, email) values (?, ?) [22001-197]
    at org.springframework.jdbc.support.SQLExceptionTranslator.doTranslate(SQLExceptionTranslator.java:246) ~[spring-jdbc-5.1.0.RELEASE.jar:5.1.0.RELEASE]
    at org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFallbackSQLExceptionTranslator.java:72) ~[spring-jdbc-5.1.0.RELEASE.jar:5.1.0.RELEASE]
17:29:30 [main] WARN org.springframework.test.context.TestContextManager - Caught exception while invoking 'afterTestMethod' callback on TestExecutionListener [org.springframework.transaction.UnexpectedRollbackException: Transaction rolled back because it has been marked as rollback-only]
    at org.springframework.transaction.support.AbstractPlatformTransactionManager.processRollback(AbstractPlatformTransactionManager.java:873) ~[spring-tx-5.1.0.RELEASE.jar:5.1.0.RELEASE]
    at org.springframework.transaction.support.AbstractPlatformTransactionManager.commit(AbstractPlatformTransactionManager.java:710) ~[spring-tx-5.1.0.RELEASE.jar:5.1.0.RELEASE]
    at org.springframework.test.context.transaction.TransactionContext.endTransaction(TransactionContext.java:131) ~[spring-test-5.1.0.RELEASE.jar:5.1.0.RELEASE]
    at org.springframework.test.context.transaction.TransactionalTestExecutionListener.afterTestMethod(TransactionalTestExecutionListener.java:216) ~[spring-test-5.1.0.RELEASE.jar:5.1.0.RELEASE]
    at org.springframework.test.context.TestContextManager.afterTestMethod(TestContextManager.java:443) ~[spring-test-5.1.0.RELEASE.jar:5.1.0.RELEASE]
    at org.springframework.test.context.junit4.SpringExtension.afterEach(SpringExtension.java:139) ~[spring-test-5.1.0.RELEASE.jar:5.1.0.RELEASE]
    at org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor.lambda$invokeAfterEachCallbacks$11(TestMethodTestDescriptor.java:218) ~[junit-jupiter-engine-5.3.1.jar:5.3.1]
    at org.junit.platform.engine.support.hierarchical.ThrowableCollector.execute(ThrowableCollector.java:72) ~[junit-platform-engine-1.3.1.jar:1.3.1]
    at org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor.lambda$invokeAllAfterMethodsOrCallbacks$13(TestMethodTestDescriptor.java:230) ~[junit-jupiter-engine-5.3.1.jar:5.3.1]
    at java.util.ArrayList.forEach(ArrayList.java:1257) ~[?:1.8.0_171]
    at org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor.invokeAllAfterMethodsOrCallbacks(TestMethodTestDescriptor.java:228) ~[junit-jupiter-engine-5.3.1.jar:5.3.1]
    at org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor.invokeAfterEachCallbacks(TestMethodTestDescriptor.java:217) ~[junit-jupiter-engine-5.3.1.jar:5.3.1]
```


EXERCISE CONCLUSION

- With this exercise we have put into practice the concept of Transactional Management using Spring Framework.
- We implemented both an error transaction, to see how the rollback of the transaction is executed, and we also tried executing a query where there is no error in the transaction and therefore it executes and ends correctly.



SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx

ONLINE COURSE

SPRING FRAMEWORK

By: Eng. Ubaldo Acosta



SPRING FRAMEWORK COURSE

www.globalmentoring.com.mx