Project Report
LAB-06 Spring framework

Submitted by:
Basedia Vivek (10721)
Pandey Anurag Kumar (10733)
EDUPUGANTI Suryavaraprasad (10405)

1. Introduction:

We have build a Spring Application that uses JPA, Spring MVC, REST Web Service with an AngularJS front end. The project is Film rental store services.

Spring allows us to build an application using Plain Old Java Objects (POJO's) and dependency injection, which helps us wire up the different components of our application. Most people prefer to use annotations, or Java Configuration and therefore stay on the pure Java side. We can also use an XML configuration file to 'wire up' our POJOs that act as beans. All of these methods are perfectly fine and supported by Spring**.**

JPA is a Java API specification which describes the management of relational data in applications using Java Platform. Where as Hibernate is a ORM (Object Relational Mapping) library which follows JPA specification. You can think JPA as a set of Rules which is implemented by Hibernate.

We have created REST Web Service, separate Business logic and DAL, used MYsql for database using MAMP.

For development Spring 4.0 has been used. To test, junit test cases are written. For testing chrome plugin /Curl dev tool is used.

.

2. Installation/ Required tools:

We used the following tools/files:

- **Spring Tool Suite 4.0.1**: https://spring.io/tools
- **MySQL Database 5.X**: M
    - Windows MAMP : https://www.mamp.info/en/downloads/
- **Database structure and content**: http://downloads.mysql.com/docs/sakila-db.zip
- **Chrome Plugin**: https://chrome.google.com/webstore/detail/hgmloofddffdnphfgcellkdfbfbjeloo

3.

Issues faced while setup:

Since we were Using Spring 4, some of the annotation which are used in the PPT were Deprecated. We have replaced them with the newer annotation and the import packages.

Deprecated one

import org.springframework.boot.test.SpringApplicationConfiguration;

@SpringApplicationConfiguration

Upgraded one

@RunWith(SpringRunner.class)

@SpringBootTest(classes =AdminPortalApplication.class)

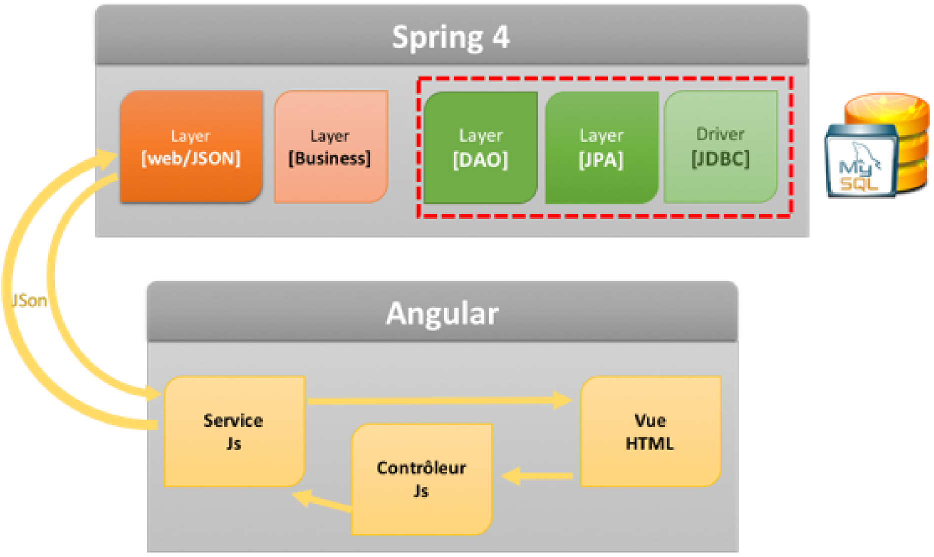
@WebAppConfiguration

Some dependency code was added to avoid error

Code added in Pom.xml :

```
<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.2.2</version>

</dependency>
```

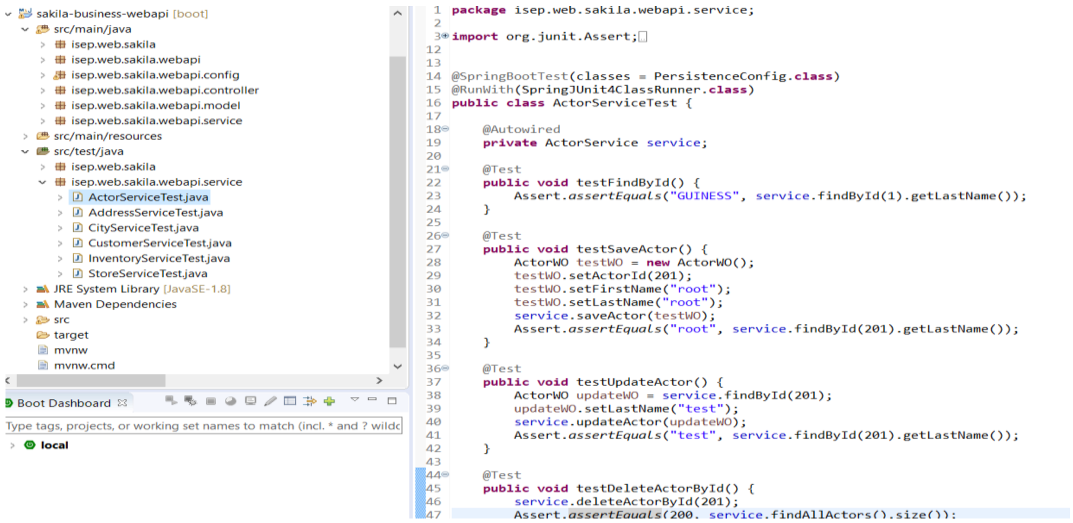
The final architecture builds a complete Web Application working application (Angular will be emulated by REST Commands). The architecture of the application will be:



The Project is to complete to rental store application. That application will be used by the rental shop in order. The rental shop is determined by the login. This is no GUI, so to test it you will have to use REST clients such as cURLou Chrome Dev environment.

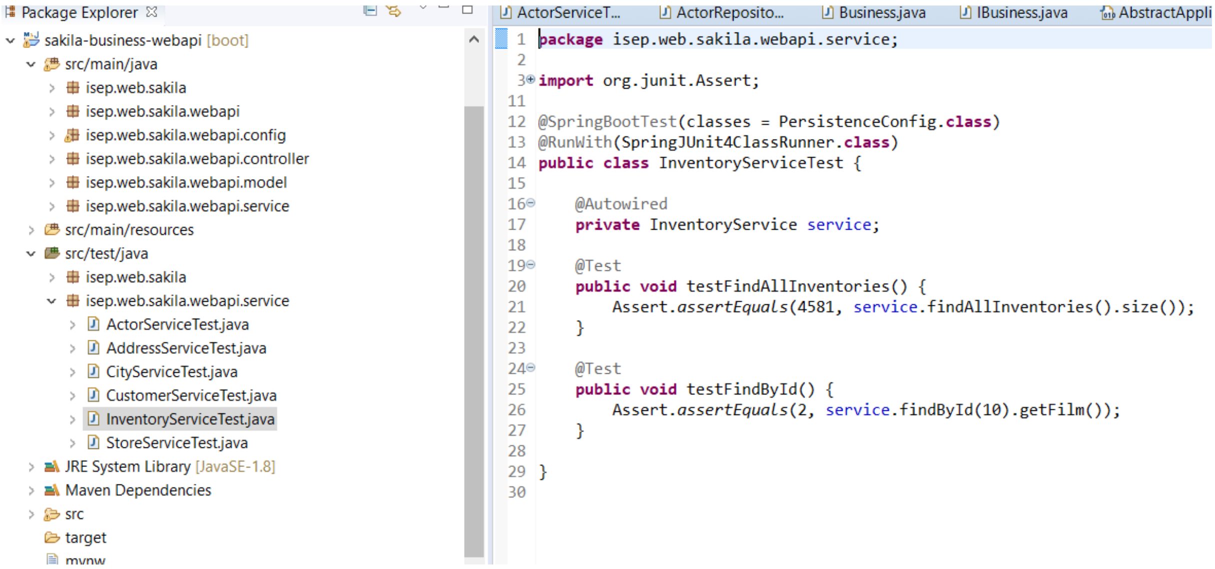
Unit Testing Scenerios:

1. Ability to create/update a customer with its address in one operation,


sakila-business-webapi [boot]
src/main/java
isep.web.sakila
isep.web.sakila.webapi
isep.web.sakila.webapi.config
isep.web.sakila.webapi.controller
isep.web.sakila.webapi.model
isep.web.sakila.webapi.service
src/main/resources
src/test/java
isep.web.sakila
isep.web.sakila.webapi.service
ActorServiceTest.java
AddressServiceTest.java
CityServiceTest.java
CustomerServiceTest.java
InventoryServiceTest.java
StoreServiceTest.java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
mvnw
mvnw.cmd
Boot Dashboard
Type tags, projects, or working set names to match (incl. * and ? wildc
local

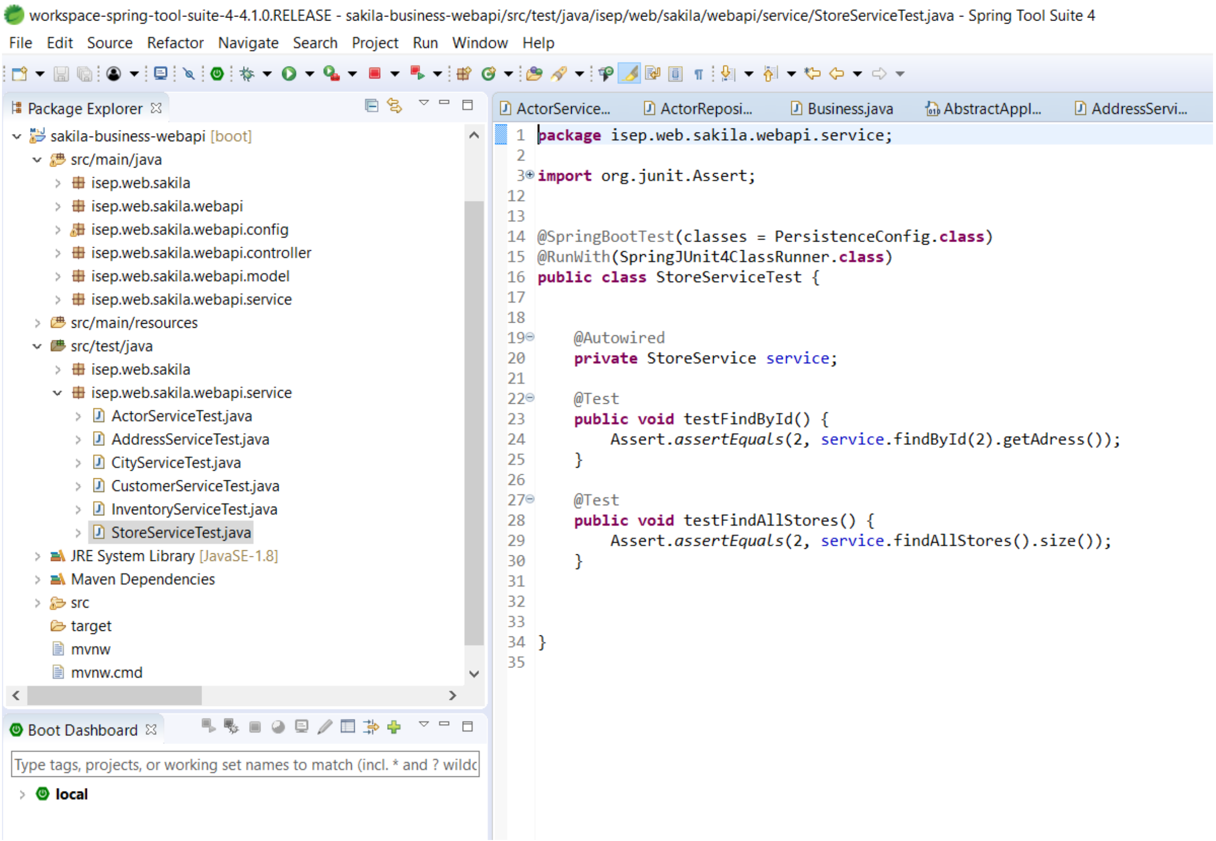
```java
package isep.web.sakila.webapi.service;

import org.junit.Assert;


@SpringBootTest(classes = PersistenceConfig.class)
@RunWith(SpringJUnit4ClassRunner.class)
public class ActorServiceTest {

    @Autowired
    private ActorService service;

    @Test
    public void testFindById() {
        Assert.assertEquals("GUINESS", service.findById(1).getLastName());
    }

    @Test
    public void testSaveActor() {
        ActorWO testWO = new ActorWO();
        testWO.setActorId(201);
        testWO.setFirstName("root");
        testWO.setLastName("root");
        service.saveActor(testWO);
        Assert.assertEquals("root", service.findById(201).getLastName());
    }

    @Test
    public void testUpdateActor() {
        ActorWO updateWO = service.findById(201);
        updateWO.setLastName("test");
        service.updateActor(updateWO);
        Assert.assertEquals("test", service.findById(201).getLastName());
    }

    @Test
    public void testDeleteActorById() {
        service.deleteActorById(201);
        Assert.assertEquals(200, service.findAllActors().size());
```

2. Ability to rent a film to a customer and to give it back



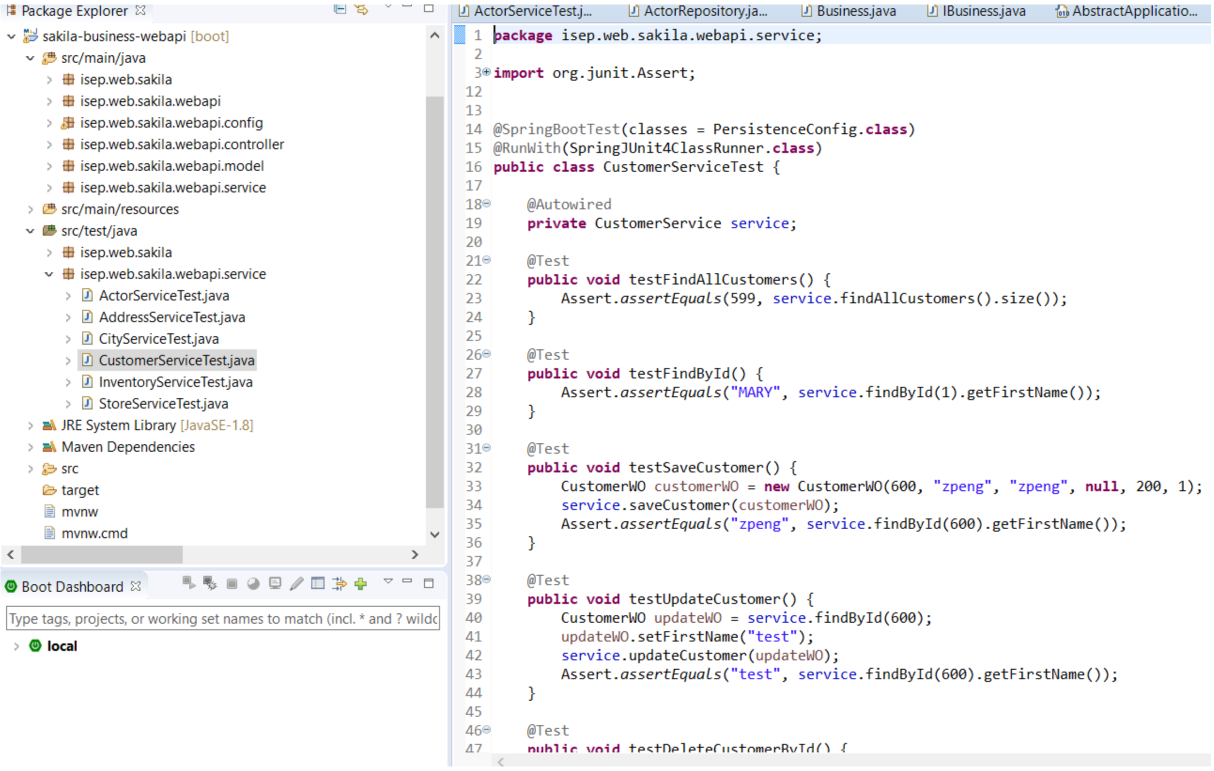
```java
package isep.web.sakila.webapi.service;

import org.junit.Assert;

@SpringBootTest(classes = PersistenceConfig.class)
@RunWith(SpringJUnit4ClassRunner.class)
public class InventoryServiceTest {

    @Autowired
    private InventoryService service;

    @Test
    public void testFindAllInventories() {
        Assert.assertEquals(4581, service.findAllInventories().size());
    }

    @Test
    public void testFindById() {
        Assert.assertEquals(2, service.findById(10).getFilm());
    }

}

```

3. Ability to Add/Remove a film from the inventory,


workspace-spring-tool-suite-4-4.1.0.RELEASE - sakila-business-webapi/src/test/java/isep/web/sakila/webapi/service/StoreServiceTest.java - Spring Tool Suite 4
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
sakila-business-webapi [boot]
src/main/java
isep.web.sakila
isep.web.sakila.webapi
isep.web.sakila.webapi.config
isep.web.sakila.webapi.controller
isep.web.sakila.webapi.model
isep.web.sakila.webapi.service
src/main/resources
src/test/java
isep.web.sakila
isep.web.sakila.webapi.service
ActorServiceTest.java
AddressServiceTest.java
CityServiceTest.java
CustomerServiceTest.java
InventoryServiceTest.java
StoreServiceTest.java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
mvnw
mvnw.cmd
Boot Dashboard
Type tags, projects, or working set names to match (incl. * and ? wildc
local
ActorService...
ActorReposi...
Business.java
AbstractAppl...
AddressServi...

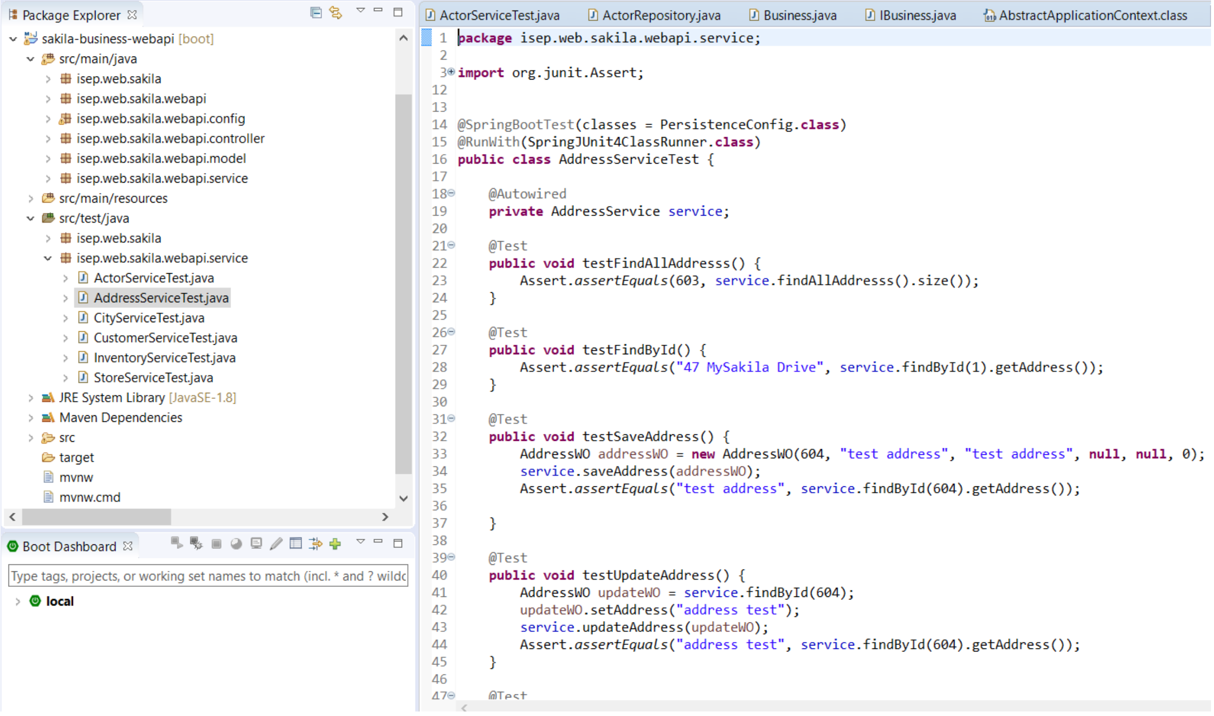
```java
package isep.web.sakila.webapi.service;

import org.junit.Assert;


@SpringBootTest(classes = PersistenceConfig.class)
@RunWith(SpringJUnit4ClassRunner.class)
public class StoreServiceTest {


    @Autowired
    private StoreService service;

    @Test
    public void testFindById() {
        Assert.assertEquals(2, service.findById(2).getAdress());
    }

    @Test
    public void testFindAllStores() {
        Assert.assertEquals(2, service.findAllStores().size());
    }



}
```

4. Ability to create/modify/delete a film and its related information. Related information are : Film Category, Language, Actors, Text

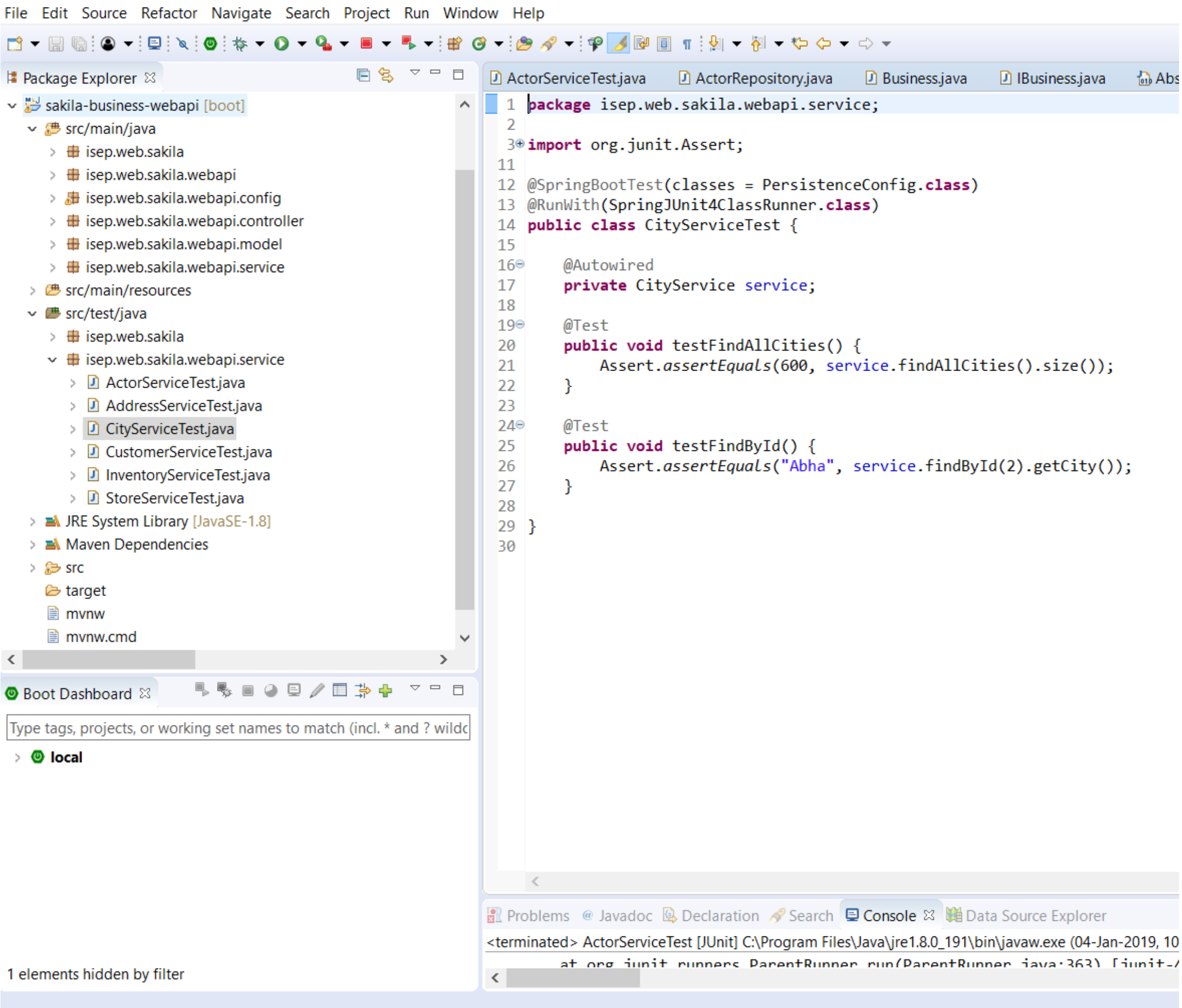


```java
package isep.web.sakila.webapi.service;

import org.junit.Assert;


@SpringBootTest(classes = PersistenceConfig.class)
@RunWith(SpringJUnit4ClassRunner.class)
public class CustomerServiceTest {

    @Autowired
    private CustomerService service;

    @Test
    public void testFindAllCustomers() {
        Assert.assertEquals(599, service.findAllCustomers().size());
    }

    @Test
    public void testFindById() {
        Assert.assertEquals("MARY", service.findById(1).getFirstName());
    }

    @Test
    public void testSaveCustomer() {
        CustomerWO customerWO = new CustomerWO(600, "zpeng", "zpeng", null, 200, 1);
        service.saveCustomer(customerWO);
        Assert.assertEquals("zpeng", service.findById(600).getFirstName());
    }

    @Test
    public void testUpdateCustomer() {
        CustomerWO updateWO = service.findById(600);
        updateWO.setFirstName("test");
        service.updateCustomer(updateWO);
        Assert.assertEquals("test", service.findById(600).getFirstName());
    }

    @Test
    public void testDeleteCustomerById() {
```

5. Ability to create/update/delete referential tables: Actor, Country, City, Language, Category



```java
package isep.web.sakila.webapi.service;

import org.junit.Assert;


@SpringBootTest(classes = PersistenceConfig.class)
@RunWith(SpringJUnit4ClassRunner.class)
public class AddressServiceTest {

	@Autowired
	private AddressService service;

	@Test
	public void testFindAllAddress() {
		Assert.assertEquals(603, service.findAllAddress().size());
	}

	@Test
	public void testFindById() {
		Assert.assertEquals("47 MySakila Drive", service.findById(1).getAddress());
	}

	@Test
	public void testSaveAddress() {
		AddressWO addressWO = new AddressWO(604, "test address", "test address", null, null, 0);
		service.saveAddress(addressWO);
		Assert.assertEquals("test address", service.findById(604).getAddress());

	}

	@Test
	public void testUpdateAddress() {
		AddressWO updateWO = service.findById(604);
		updateWO.setAddress("address test");
		service.updateAddress(updateWO);
		Assert.assertEquals("address test", service.findById(604).getAddress());
	}

	@Test
```



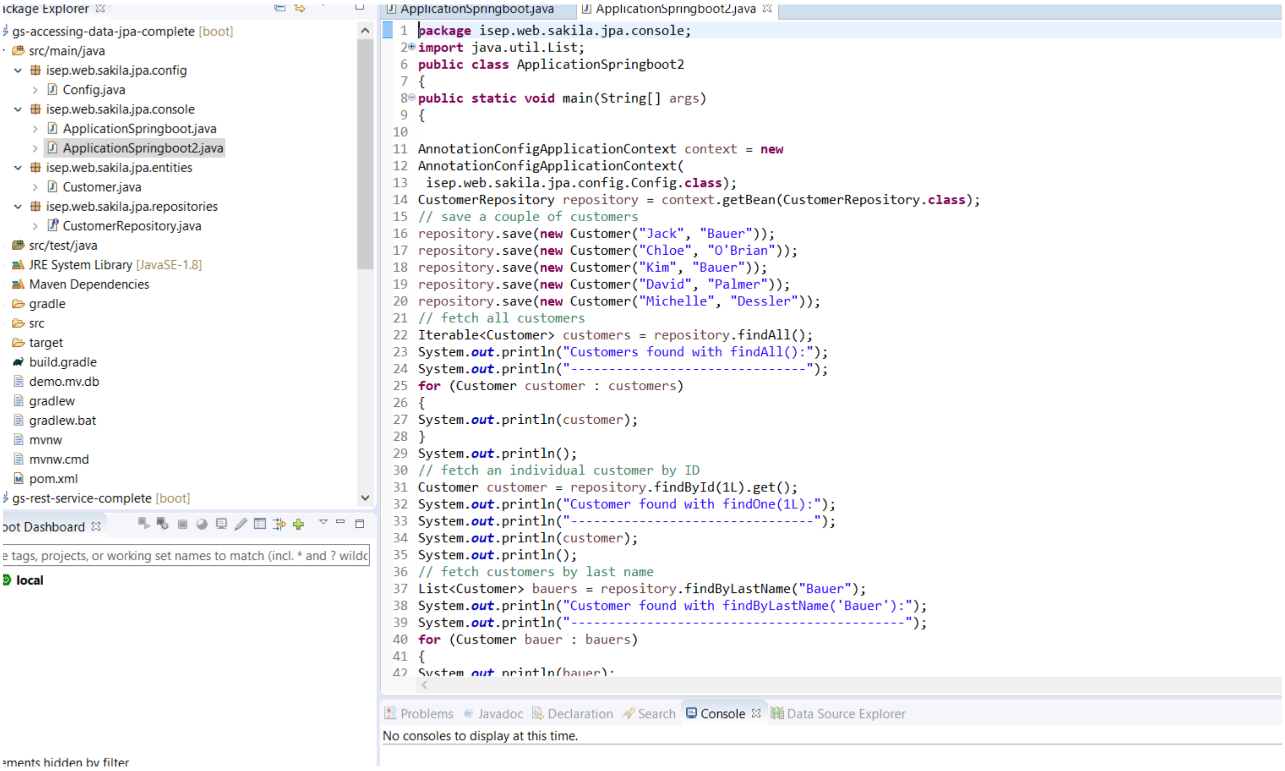
```java
package isep.web.sakila.webapi.service;

import org.junit.Assert;

@SpringBootTest(classes = PersistenceConfig.class)
@RunWith(SpringJUnit4ClassRunner.class)
public class CityServiceTest {

	@Autowired
	private CityService service;

	@Test
	public void testFindAllCities() {
		Assert.assertEquals(600, service.findAllCities().size());
	}

	@Test
	public void testFindById() {
		Assert.assertEquals("Abha", service.findById(2).getCity());
	}

}
```

```
<terminated> ActorServiceTest [JUnit] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (04-Jan-2019, 10
	at org.junit.runners.ParentRunner.run(ParentRunner.java:363) [junit-4
```

Deliverable files:

- A STS / Maven project for the DAL with Unit Testing,
- A STS / Maven project for the REST services &Business Logic with Unit Testing

3. Building the Data Access Layer

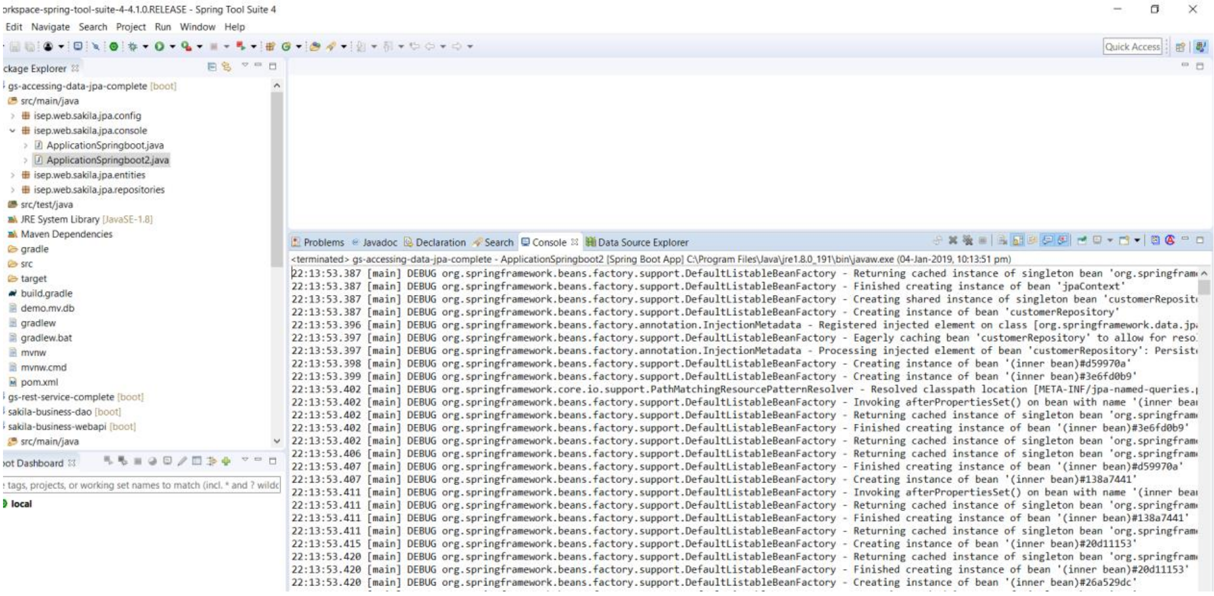


ickage Explorer
gs-accessing-data-jpa-complete [boot]
src/main/java
isep.web.sakila.jpa.config
Config.java
isep.web.sakila.jpa.console
ApplicationSpringboot.java
ApplicationSpringboot2.java
isep.web.sakila.jpa.entities
Customer.java
isep.web.sakila.jpa.repositories
CustomerRepository.java
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
gradle
src
target
build.gradle
demo.mv.db
gradlew
gradlew.bat
mvnw
mvnw.cmd
pom.xml
gs-rest-service-complete [boot]
oot Dashboard
e tags, projects, or working set names to match (incl. * and ? wildc
local
ements hidden by filter
ApplicationSpringboot.java
ApplicationSpringboot2.java
Problems
Javadoc
Declaration
Search
Console
Data Source Explorer

```
package isep.web.sakila.jpa.console;
import java.util.List;
public class ApplicationSpringboot2
{
public static void main(String[] args)
{

AnnotationConfigApplicationContext context = new
AnnotationConfigApplicationContext(
 isep.web.sakila.jpa.config.Config.class);
CustomerRepository repository = context.getBean(CustomerRepository.class);
// save a couple of customers
repository.save(new Customer("Jack", "Bauer"));
repository.save(new Customer("Chloe", "O'Brian"));
repository.save(new Customer("Kim", "Bauer"));
repository.save(new Customer("David", "Palmer"));
repository.save(new Customer("Michelle", "Dessler"));
// fetch all customers
Iterable<Customer> customers = repository.findAll();
System.out.println("Customers found with findAll():");
System.out.println("-------------------------------");
for (Customer customer : customers)
{
System.out.println(customer);
}
System.out.println();
// fetch an individual customer by ID
Customer customer = repository.findById(1L).get();
System.out.println("Customer found with findOne(1L):");
System.out.println("--------------------------------");
System.out.println(customer);
System.out.println();
// fetch customers by last name
List<Customer> bauers = repository.findByLastName("Bauer");
System.out.println("Customer found with findByLastName('Bauer'):");
System.out.println("--------------------------------------------");
for (Customer bauer : bauers)
{
System.out.println(bauer);
```

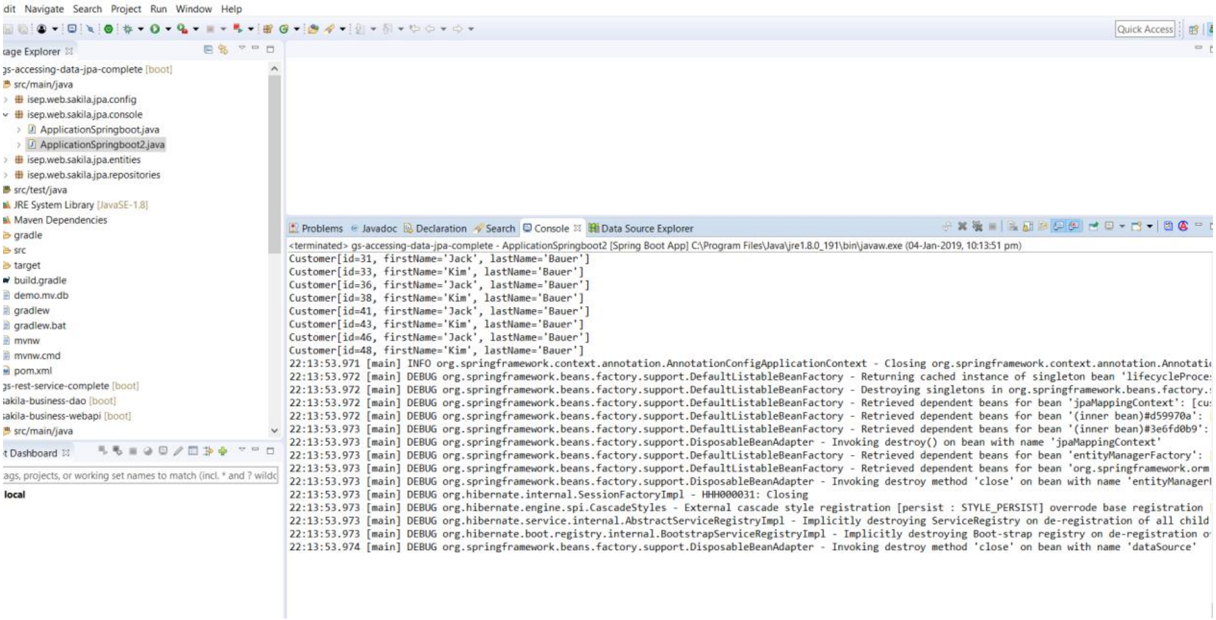
No consoles to display at this time.

Run as SpringBoot apps ApplicationSpringboot and ApplicationSpringboot2.The result of the execution can be seen in the console. You'll find bellow a capture of the console output.



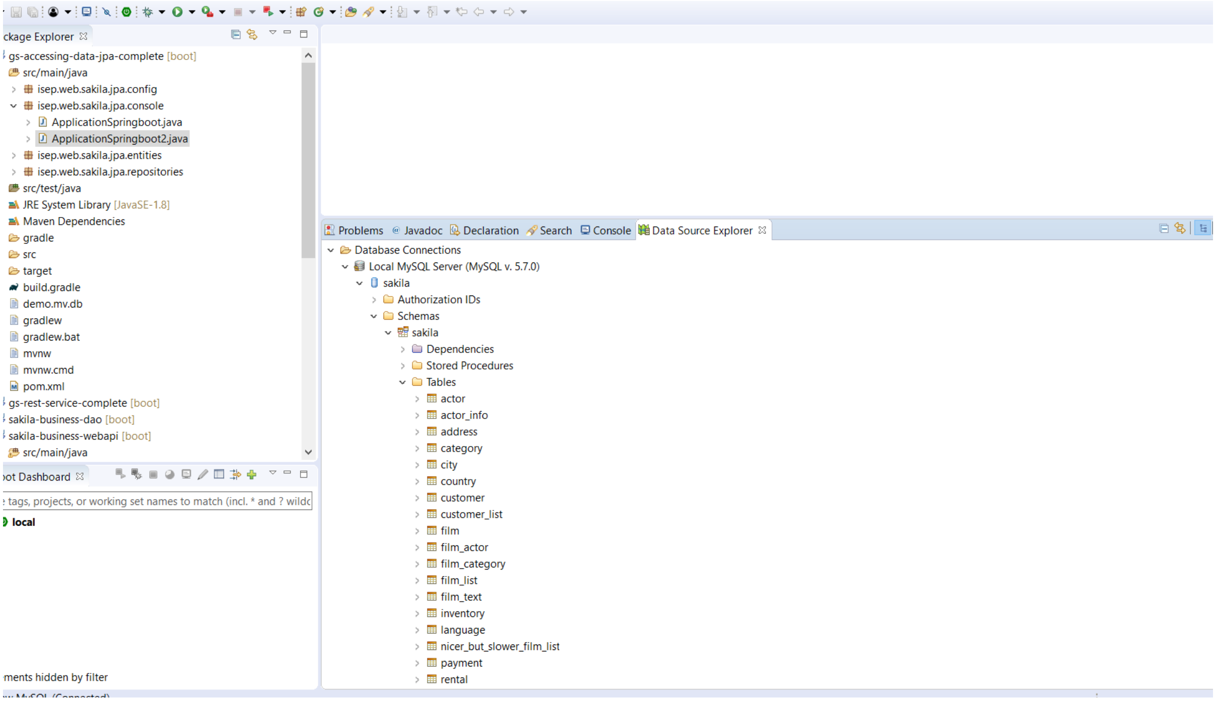
orkspace-spring-tool-suite-4-4.1.0.RELEASE - Spring Tool Suite 4
Edit Navigate Search Project Run Window Help
Quick Access
ckage Explorer
gs-accessing-data-jpa-complete [boot]
src/main/java
isep.web.sakila.jpa.config
isep.web.sakila.jpa.console
ApplicationSpringboot.java
ApplicationSpringboot2.java
isep.web.sakila.jpa.entities
isep.web.sakila.jpa.repositories
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
gradle
src
target
build.gradle
demo.mv.db
gradlew
gradlew.bat
mvnw
mvnw.cmd
pom.xml
gs-rest-service-complete [boot]
sakila-business-dao [boot]
sakila-business-webapi [boot]
src/main/java
oot Dashboard
tags, projects, or working set names to match (incl. * and ? wildc
local
Problems
Javadoc
Declaration
Search
Console
Data Source Explorer

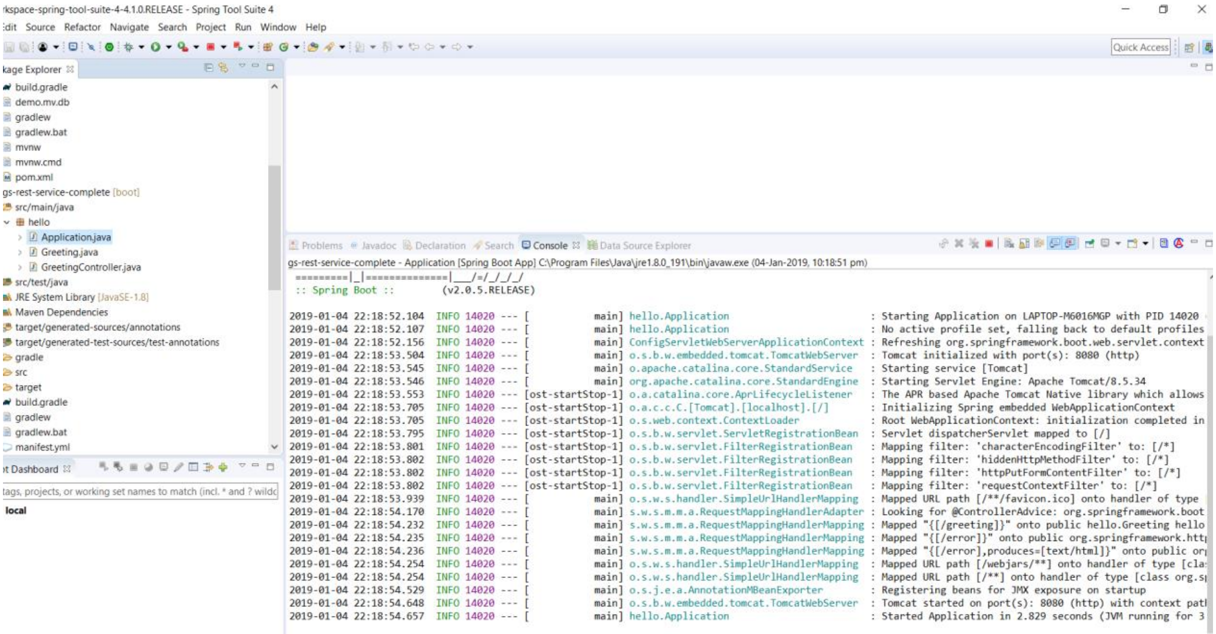
```
<terminated> gs-accessing-data-jpa-complete - ApplicationSpringboot2 [Spring Boot App] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (04-Jan-2019, 10:13:51 pm)
22:13:53.387 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Returning cached instance of singleton bean 'org.springfram
22:13:53.387 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Finished creating instance of bean 'jpaContext'
22:13:53.387 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating shared instance of singleton bean 'customerReposit
22:13:53.387 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating instance of bean 'customerRepository'
22:13:53.396 [main] DEBUG org.springframework.beans.factory.annotation.InjectionMetadata - Registered injected element on class [org.springframework.data.jp
22:13:53.397 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Eagerly caching bean 'customerRepository' to allow for reso
22:13:53.397 [main] DEBUG org.springframework.beans.factory.annotation.InjectionMetadata - Processing injected element of bean 'customerRepository': Persist
22:13:53.398 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating instance of bean '(inner bean)#d59970a'
22:13:53.399 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating instance of bean '(inner bean)#3e6fd0b9'
22:13:53.402 [main] DEBUG org.springframework.core.io.support.PathMatchingResourcePatternResolver - Resolved classpath location [META-INF/jpa-named-queries.
22:13:53.402 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Invoking afterPropertiesSet() on bean with name '(inner bea
22:13:53.402 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Returning cached instance of singleton bean 'org.springfram
22:13:53.402 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Finished creating instance of bean '(inner bean)#3e6fd0b9'
22:13:53.402 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Returning cached instance of singleton bean 'org.springfram
22:13:53.406 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Returning cached instance of singleton bean 'org.springfram
22:13:53.407 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Finished creating instance of bean '(inner bean)#d59970a'
22:13:53.407 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating instance of bean '(inner bean)#138a7441'
22:13:53.411 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Invoking afterPropertiesSet() on bean with name '(inner bea
22:13:53.411 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Returning cached instance of singleton bean 'org.springfram
22:13:53.411 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Finished creating instance of bean '(inner bean)#138a7441'
22:13:53.411 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Returning cached instance of singleton bean 'org.springfram
22:13:53.415 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating instance of bean '(inner bean)#20d11153'
22:13:53.420 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Returning cached instance of singleton bean 'org.springfram
22:13:53.420 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Finished creating instance of bean '(inner bean)#20d11153'
22:13:53.420 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating instance of bean '(inner bean)#26a529dc'
```


dit Navigate Search Project Run Window Help
Quick Access
age Explorer
js-accessing-data-jpa-complete [boot]
src/main/java
isep.web.sakila.jpa.config
isep.web.sakila.jpa.console
ApplicationSpringboot.java
ApplicationSpringboot2.java
isep.web.sakila.jpa.entities
isep.web.sakila.jpa.repositories
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
gradle
src
target
build.gradle
demo.mv.db
gradlew
gradlew.bat
mvnw
mvnw.cmd
pom.xml
js-rest-service-complete [boot]
akila-business-dao [boot]
akila-business-webapi [boot]
src/main/java
t Dashboard
ags, projects, or working set names to match (incl. * and ? wildc
local
Problems
Javadoc
Declaration
Search
Console
Data Source Explorer

```
<terminated> gs-accessing-data-jpa-complete - ApplicationSpringboot2 [Spring Boot App] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (04-Jan-2019, 10:13:51 pm)
Customer[id=31, firstName='Jack', lastName='Bauer']
Customer[id=33, firstName='Kim', lastName='Bauer']
Customer[id=36, firstName='Jack', lastName='Bauer']
Customer[id=38, firstName='Kim', lastName='Bauer']
Customer[id=41, firstName='Jack', lastName='Bauer']
Customer[id=43, firstName='Kim', lastName='Bauer']
Customer[id=46, firstName='Jack', lastName='Bauer']
Customer[id=48, firstName='Kim', lastName='Bauer']
22:13:53.971 [main] INFO org.springframework.context.annotation.AnnotationConfigApplicationContext - Closing org.springframework.context.annotation.Annotati
22:13:53.972 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Returning cached instance of singleton bean 'lifecycleProce
22:13:53.972 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Destroying singletons in org.springframework.beans.factory.
22:13:53.972 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Retrieved dependent beans for bean 'jpaMappingContext': [cu
22:13:53.972 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Retrieved dependent beans for bean '(inner bean)#d59970a':
22:13:53.973 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Retrieved dependent beans for bean '(inner bean)#3e6fd0b9':
22:13:53.973 [main] DEBUG org.springframework.beans.factory.support.DisposableBeanAdapter - Invoking destroy() on bean with name 'jpaMappingContext'
22:13:53.973 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Retrieved dependent beans for bean 'entityManagerFactory':
22:13:53.973 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Retrieved dependent beans for bean 'org.springframework.orm
22:13:53.973 [main] DEBUG org.springframework.beans.factory.support.DisposableBeanAdapter - Invoking destroy method 'close' on bean with name 'entityManager
22:13:53.973 [main] DEBUG org.hibernate.internal.SessionFactoryImpl - HHH000031: Closing
22:13:53.973 [main] DEBUG org.hibernate.engine.spi.CascadeStyles - External cascade style registration [persist : STYLE_PERSIST] overrode base registration
22:13:53.973 [main] DEBUG org.hibernate.service.internal.AbstractServiceRegistryImpl - Implicitly destroying ServiceRegistry on de-registration of all child
22:13:53.973 [main] DEBUG org.hibernate.boot.registry.internal.BootstrapServiceRegistryImpl - Implicitly destroying Boot-strap registry on de-registration o
22:13:53.974 [main] DEBUG org.springframework.beans.factory.support.DisposableBeanAdapter - Invoking destroy method 'close' on bean with name 'dataSource'
```

Database MY sql Sakila import Spring configure:

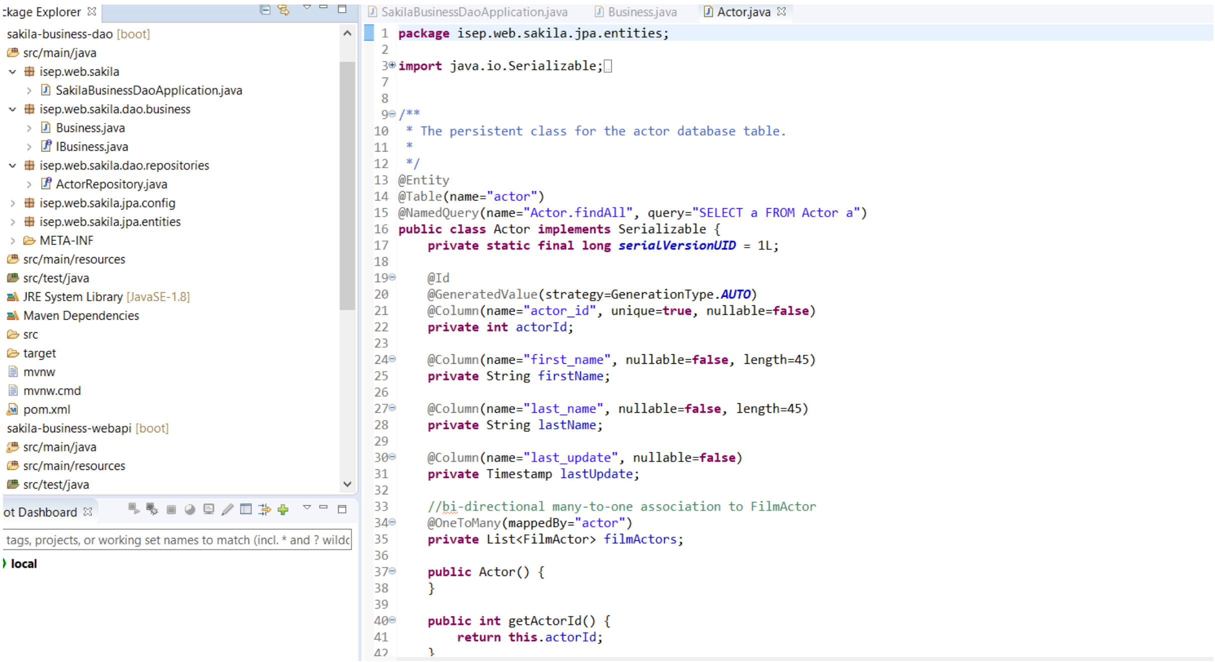




rkspace-spring-tool-suite-4-4.1.0.RELEASE - Spring Tool Suite 4
Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
kage Explorer
build.gradle
demo.mv.db
gradlew
gradlew.bat
mvnw
mvnw.cmd
pom.xml
gs-rest-service-complete [boot]
src/main/java
hello
Application.java
Greeting.java
GreetingController.java
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
gradle
src
target
build.gradle
gradlew
gradlew.bat
manifest.yml
ot Dashboard
tags, projects, or working set names to match (incl. * and ? wildc
local
Problems
Javadoc
Declaration
Search
Console
Data Source Explorer

```
gs-rest-service-complete - Application [Spring Boot App] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (04-Jan-2019, 10:18:51 pm)
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.0.5.RELEASE)

2019-01-04 22:18:52.104  INFO 14020 --- [           main] hello.Application                        : Starting Application on LAPTOP-M6016MGP with PID 14020
2019-01-04 22:18:52.107  INFO 14020 --- [           main] hello.Application                        : No active profile set, falling back to default profiles
2019-01-04 22:18:52.156  INFO 14020 --- [           main] ConfigServletWebServerApplicationContext : Refreshing org.springframework.boot.web.servlet.context
2019-01-04 22:18:53.504  INFO 14020 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2019-01-04 22:18:53.545  INFO 14020 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2019-01-04 22:18:53.546  INFO 14020 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet Engine: Apache Tomcat/8.5.34
2019-01-04 22:18:53.553  INFO 14020 --- [ost-startStop-1] o.a.catalina.core.AprLifecycleListener   : The APR based Apache Tomcat Native library which allows
2019-01-04 22:18:53.705  INFO 14020 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2019-01-04 22:18:53.705  INFO 14020 --- [ost-startStop-1] o.s.web.context.ContextLoader            : Root WebApplicationContext: initialization completed in
2019-01-04 22:18:53.795  INFO 14020 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean  : Servlet dispatcherServlet mapped to [/]
2019-01-04 22:18:53.801  INFO 14020 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter: 'characterEncodingFilter' to: [/*]
2019-01-04 22:18:53.802  INFO 14020 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter: 'hiddenHttpMethodFilter' to: [/*]
2019-01-04 22:18:53.802  INFO 14020 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter: 'httpPutFormContentFilter' to: [/*]
2019-01-04 22:18:53.802  INFO 14020 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter: 'requestContextFilter' to: [/*]
2019-01-04 22:18:53.939  INFO 14020 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/**/favicon.ico] onto handler of type
2019-01-04 22:18:54.170  INFO 14020 --- [           main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot
2019-01-04 22:18:54.232  INFO 14020 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/greeting]}" onto public hello.Greeting hello
2019-01-04 22:18:54.235  INFO 14020 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.htt
2019-01-04 22:18:54.236  INFO 14020 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error],produces=[text/html]}" onto public or
2019-01-04 22:18:54.254  INFO 14020 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/webjars/**] onto handler of type [cla
2019-01-04 22:18:54.254  INFO 14020 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/**] onto handler of type [class org.s
2019-01-04 22:18:54.529  INFO 14020 --- [           main] o.s.j.e.a.AnnotationMBeanExporter        : Registering beans for JMX exposure on startup
2019-01-04 22:18:54.648  INFO 14020 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context pat
2019-01-04 22:18:54.657  INFO 14020 --- [           main] hello.Application                        : Started Application in 2.829 seconds (JVM running for 3
```

Creating the Real DAO Project:



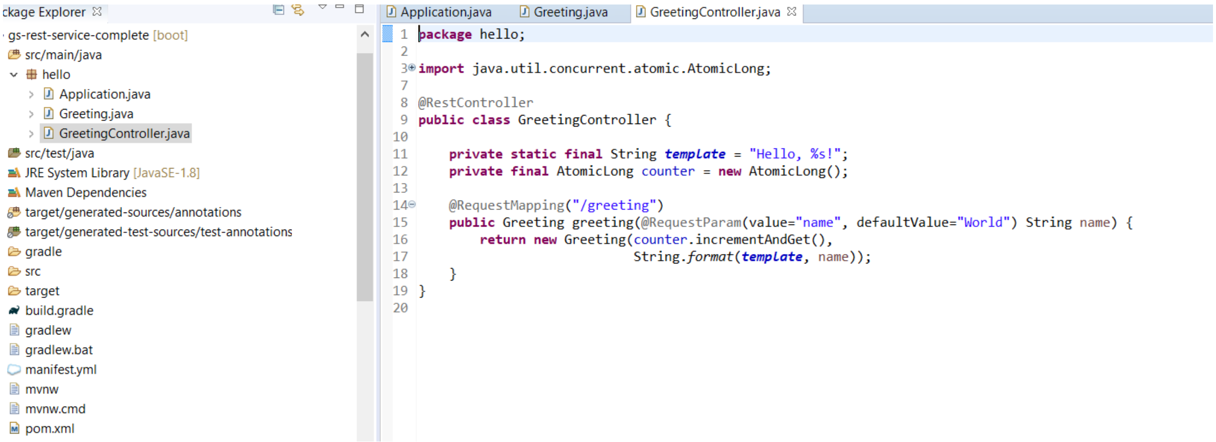
kage Explorer
sakila-business-dao [boot]
src/main/java
isep.web.sakila
SakilaBusinessDaoApplication.java
isep.web.sakila.dao.business
Business.java
IBusiness.java
isep.web.sakila.dao.repositories
ActorRepository.java
isep.web.sakila.jpa.config
isep.web.sakila.jpa.entities
META-INF
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
mvnw
mvnw.cmd
pom.xml
sakila-business-webapi [boot]
src/main/java
src/main/resources
src/test/java
ot Dashboard
tags, projects, or working set names to match (incl. * and ? wildc
local
SakilaBusinessDaoApplication.java
Business.java
Actor.java

```java
package isep.web.sakila.jpa.entities;

import java.io.Serializable;


/**
 * The persistent class for the actor database table.
 * 
 */
@Entity
@Table(name="actor")
@NamedQuery(name="Actor.findAll", query="SELECT a FROM Actor a")
public class Actor implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name="actor_id", unique=true, nullable=false)
    private int actorId;

    @Column(name="first_name", nullable=false, length=45)
    private String firstName;

    @Column(name="last_name", nullable=false, length=45)
    private String lastName;

    @Column(name="last_update", nullable=false)
    private Timestamp lastUpdate;

    //bi-directional many-to-one association to FilmActor
    @OneToMany(mappedBy="actor")
    private List<FilmActor> filmActors;

    public Actor() {
    }

    public int getActorId() {
        return this.actorId;
    }
```

In order to expose the Business Logic to your clients, you'll need a specific interface. In our project this will be completed by the Web/JSon layer.

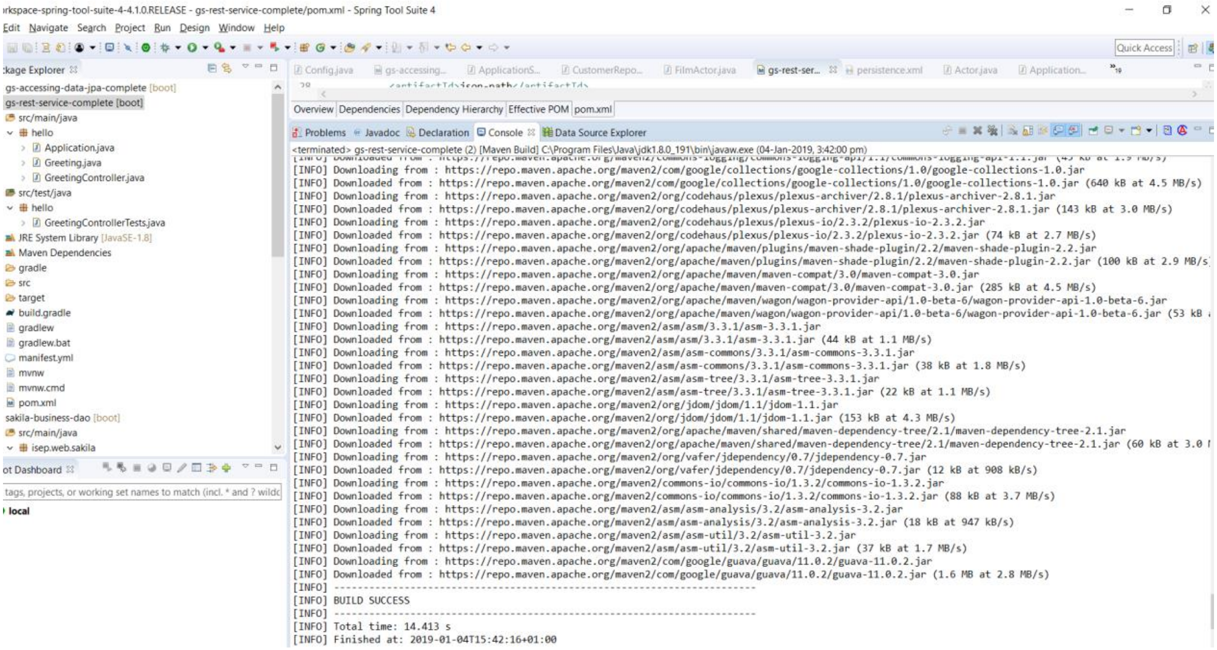
That Layer will expose Web Services at the REST format. Those Web Services will answer to the queries text formatted in JSON (JavaScript Object Notation). That kind of web application are often called Web API. We are going to implement that Web Application with Spring MVC.

Create The Service Layer:



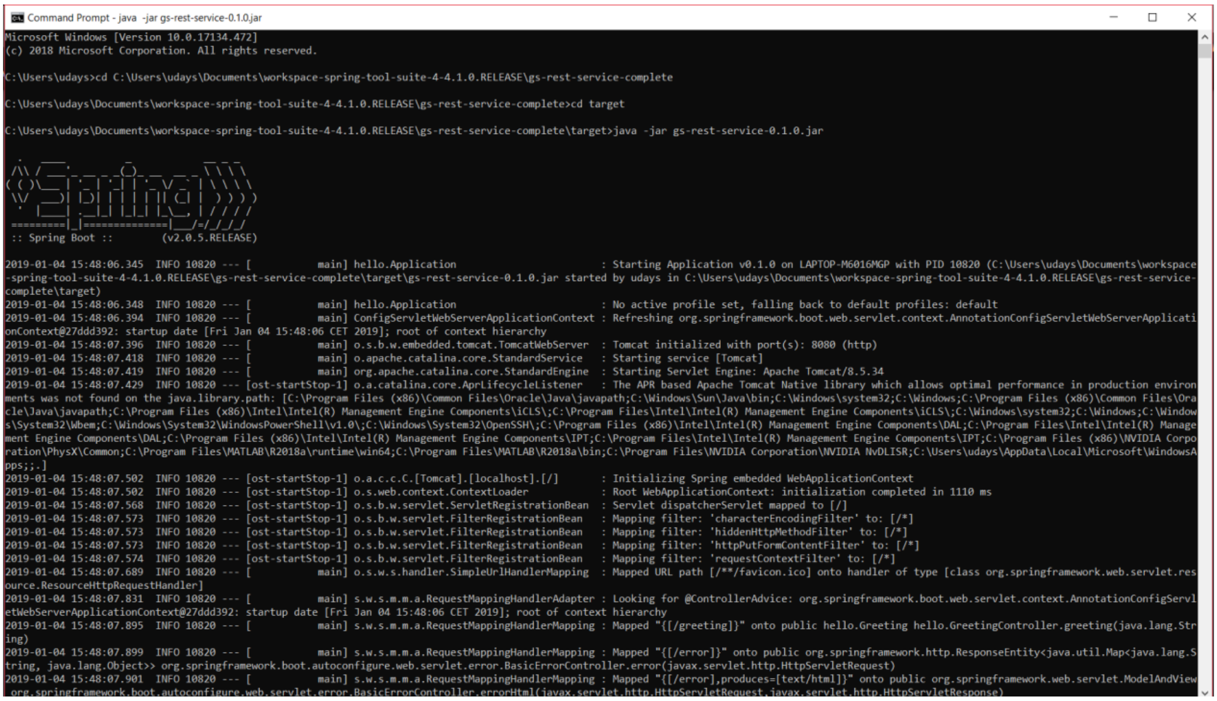
```java
package hello;

import java.util.concurrent.atomic.AtomicLong;

@RestController
public class GreetingController {

    private static final String template = "Hello, %s!";
    private final AtomicLong counter = new AtomicLong();

    @RequestMapping("/greeting")
    public Greeting greeting(@RequestParam(value="name", defaultValue="World") String name) {
        return new Greeting(counter.incrementAndGet(),
                            String.format(template, name));
    }
}

```

Spring Boot



```
<terminated> gs-rest-service-complete (2) [Maven Build] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (04-Jan-2019, 3:42:00 pm)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/com/google/collections/google-collections/1.0/google-collections-1.0.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/com/google/collections/google-collections/1.0/google-collections-1.0.jar (640 kB at 4.5 MB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/2.8.1/plexus-archiver-2.8.1.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/2.8.1/plexus-archiver-2.8.1.jar (143 kB at 3.0 MB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.3.2/plexus-io-2.3.2.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.3.2/plexus-io-2.3.2.jar (74 kB at 2.7 MB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-shade-plugin/2.2/maven-shade-plugin-2.2.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-shade-plugin/2.2/maven-shade-plugin-2.2.jar (100 kB at 2.9 MB/s
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/apache/maven/maven-compat/3.0/maven-compat-3.0.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/apache/maven/maven-compat/3.0/maven-compat-3.0.jar (285 kB at 4.5 MB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/apache/maven/wagon/wagon-provider-api/1.0-beta-6/wagon-provider-api-1.0-beta-6.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/apache/maven/wagon/wagon-provider-api/1.0-beta-6/wagon-provider-api-1.0-beta-6.jar (53 kB
[INFO] Downloading from : https://repo.maven.apache.org/maven2/asm/asm/3.3.1/asm-3.3.1.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/asm/asm/3.3.1/asm-3.3.1.jar (44 kB at 1.1 MB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/asm/asm-commons/3.3.1/asm-commons-3.3.1.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/asm/asm-commons/3.3.1/asm-commons-3.3.1.jar (38 kB at 1.8 MB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/asm/asm-tree/3.3.1/asm-tree-3.3.1.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/asm/asm-tree/3.3.1/asm-tree-3.3.1.jar (22 kB at 1.1 MB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/jdom/jdom/1.1/jdom-1.1.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/jdom/jdom/1.1/jdom-1.1.jar (153 kB at 4.3 MB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-dependency-tree/2.1/maven-dependency-tree-2.1.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-dependency-tree/2.1/maven-dependency-tree-2.1.jar (60 kB at 3.0
[INFO] Downloading from : https://repo.maven.apache.org/maven2/org/vafer/jdependency/0.7/jdependency-0.7.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/org/vafer/jdependency/0.7/jdependency-0.7.jar (12 kB at 908 kB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/commons-io/commons-io/1.3.2/commons-io-1.3.2.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/commons-io/commons-io/1.3.2/commons-io-1.3.2.jar (88 kB at 3.7 MB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/asm/asm-analysis/3.2/asm-analysis-3.2.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/asm/asm-analysis/3.2/asm-analysis-3.2.jar (18 kB at 947 kB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/asm/asm-util/3.2/asm-util-3.2.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/asm/asm-util/3.2/asm-util-3.2.jar (37 kB at 1.7 MB/s)
[INFO] Downloading from : https://repo.maven.apache.org/maven2/com/google/guava/guava/11.0.2/guava-11.0.2.jar
[INFO] Downloaded from : https://repo.maven.apache.org/maven2/com/google/guava/guava/11.0.2/guava-11.0.2.jar (1.6 MB at 2.8 MB/s)
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 14.413 s
[INFO] Finished at: 2019-01-04T15:42:16+01:00
```

Cmd prompt se Maven project build output Clean pakcage



```
Microsoft Windows [Version 10.0.17134.472]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\udays>cd C:\Users\udays\Documents\workspace-spring-tool-suite-4-4.1.0.RELEASE\gs-rest-service-complete

C:\Users\udays\Documents\workspace-spring-tool-suite-4-4.1.0.RELEASE\gs-rest-service-complete>cd target

C:\Users\udays\Documents\workspace-spring-tool-suite-4-4.1.0.RELEASE\gs-rest-service-complete\target>java -jar gs-rest-service-0.1.0.jar

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.0.5.RELEASE)

2019-01-04 15:48:06.345  INFO 10820 --- [           main] hello.Application                        : Starting Application v0.1.0 on LAPTOP-M6016MGP with PID 10820 (C:\Users\udays\Documents\workspace
-spring-tool-suite-4-4.1.0.RELEASE\gs-rest-service-complete\target\gs-rest-service-0.1.0.jar started by udays in C:\Users\udays\Documents\workspace-spring-tool-suite-4-4.1.0.RELEASE\gs-rest-service-
complete\target)
2019-01-04 15:48:06.348  INFO 10820 --- [           main] hello.Application                        : No active profile set, falling back to default profiles: default
2019-01-04 15:48:06.394  INFO 10820 --- [           main] ConfigServletWebServerApplicationContext : Refreshing org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicati
onContext@27ddd392: startup date [Fri Jan 04 15:48:06 CET 2019]; root of context hierarchy
2019-01-04 15:48:07.396  INFO 10820 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2019-01-04 15:48:07.418  INFO 10820 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2019-01-04 15:48:07.419  INFO 10820 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet Engine: Apache Tomcat/8.5.34
2019-01-04 15:48:07.429  INFO 10820 --- [ost-startStop-1] o.a.catalina.core.AprLifecycleListener   : The APR based Apache Tomcat Native library which allows optimal performance in production environ
ments was not found on the java.library.path: [C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Windows\Sun\Java\bin;C:\Windows\system32;C:\Windows;C:\Program Files (x86)\Common Files\Ora
cle\Java\javapath;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\iCLS\;C:\Program Files\Intel\Intel(R) Management Engine Components\iCLS\;C:\Windows\system32;C:\Windows;C:\Window
s\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Windows\System32\OpenSSH\;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files\Intel\Intel(R) Manage
ment Engine Components\DAL;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files (x86)\NVIDIA Corpo
ration\PhysX\Common;C:\Program Files\MATLAB\R2018a\runtime\win64;C:\Program Files\MATLAB\R2018a\bin;C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR;C:\Users\udays\AppData\Local\Microsoft\WindowsA
pps;;.]
2019-01-04 15:48:07.502  INFO 10820 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2019-01-04 15:48:07.502  INFO 10820 --- [ost-startStop-1] o.s.web.context.ContextLoader            : Root WebApplicationContext: initialization completed in 1110 ms
2019-01-04 15:48:07.568  INFO 10820 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean  : Servlet dispatcherServlet mapped to [/]
2019-01-04 15:48:07.573  INFO 10820 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter: 'characterEncodingFilter' to: [/*]
2019-01-04 15:48:07.573  INFO 10820 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter: 'hiddenHttpMethodFilter' to: [/*]
2019-01-04 15:48:07.573  INFO 10820 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter: 'httpPutFormContentFilter' to: [/*]
2019-01-04 15:48:07.574  INFO 10820 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean   : Mapping filter: 'requestContextFilter' to: [/*]
2019-01-04 15:48:07.689  INFO 10820 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.res
ource.ResourceHttpRequestHandler]
2019-01-04 15:48:07.831  INFO 10820 --- [           main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.web.servlet.context.AnnotationConfigServl
etWebServerApplicationContext@27ddd392: startup date [Fri Jan 04 15:48:06 CET 2019]; root of context hierarchy
2019-01-04 15:48:07.895  INFO 10820 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/greeting]}" onto public hello.Greeting hello.GreetingController.greeting(java.lang.Str
ing)
2019-01-04 15:48:07.899  INFO 10820 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.http.ResponseEntity<java.util.Map<java.lang.S
tring, java.lang.Object>> org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorController.error(javax.servlet.http.HttpServletRequest)
2019-01-04 15:48:07.901  INFO 10820 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error],produces=[text/html]}" onto public org.springframework.web.servlet.ModelAndView
 org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorController.errorHtml(javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResponse)
```

Learnings from the Project:

The goal of the Spring Data JPA project is:

Implementing a data access layer of an application has been cumbersome for quite a while. Too much boilerplate code has to be written to execute simple queries as well as perform pagination, and auditing. Spring JPA aims to significantly improve the implementation of data access layers by reducing the effort to the amount that's actually needed. As a developer we write our repository interfaces, including custom finder methods, and Spring will provide the implementation automatically.

Spring Data JPA has following advantages over the "old school" method of building JPA repositories:

It provides CRUD capabilities to any domain object without the need of any boilerplate code.

It minimizes the amount of source code needed to write custom queries.

It offers simple abstractions for performing common tasks like sorting an pagination.

The thing is that implementing these functions have forced the developers to write a lot of boilerplate code in the past. Spring Data JPA changes all this. It minimizes the amount of code needed for implementing repositories.

JPA provides a database independent abstraction on top of SQL. As long as we not using any native queries, we don't have to worry about database portability. Our JPA implementation adapts the generated SQL statements in each API call or JPQL query to the specific database dialect and handles the different database-specific data types.