****

Testplan template

Integration and Communication

CODE TCIF-V2IAC1-15

HBO-ICT SIE Jaar 2 2016-2017

Cursuseigenaar: jeroen.weber@hu.nl

Auteurs

Alex Jongman

Jeroen Weber

**Review**

BAR:

Nick Stuivenberg 1667775

Datum

21 – 06 – 2017

Versie 0.1

© Hogeschool Utrecht, Utrecht, 2015



# Inhoudsopgave

[Inhoudsopgave 2](#_Toc473464760)

[1. Versie 3](#_Toc473464761)

[2. Introductie 3](#_Toc473464762)

[3. Testomgeving 3](#_Toc473464763)

[4. Smoketest 3](#_Toc473464764)

[5. Testcases black box test (integratietest of functionele test) 3](#_Toc473464765)

[6. Unit Tests 3](#_Toc473464766)

[7. Regressietests 4](#_Toc473464767)

# Versie

|  |  |  |
| --- | --- | --- |
| Datum | Versie | Aanpassing |
| 21-06-2017 | 0.1 | Gehele rapport geschreven |

# Introductie

* Voor de RESTAPI opdracht heb ik gebruik gemaakt van het Spring framework. Deze is getest met een unit test, smoke test en via rest calls via Postman

# Testomgeving

Deze applicatie maakt gebruik van één service. Dat is de CityService. De city service is verantwoordelijk voor het contact van de controller naar de Repo. De service is getest met unit tests. De applicatie wordt getest dmv een smoke test: draait de server uberhäupt?

# Smoketest

Naast de SmokeTest.Java die bekijkt of de serverbestaat heb ik de volgende restricties in het syteem gebouwd:

Een stadsnaam kan maar één keer voorkomen

Er wordt pas iets gedaan in de service als de stadsnaam in het systeem voorkomt (behalve bij Create natuurlijk)

Als de gebruiker de hoogte van de uitkering opgeeft wordt er getest of dit daadwerkelijk een dubbel is.

# Testcases black box test (integratietest of functionele test)

Voor het testen van mijn applicatie heb ik gebruik gemaakt van de applicatie Postman, enkele voorbeelden:

|  |
| --- |
|  |
| **Figuur 1: invoegen van een City object met Personen list** |

|  |
| --- |
|  |
| **Figuur 2: Uitkomst berekening** |

# Unit Tests

**MainTest.java**

package nl.hu.iac;  
  
*/\*\*  
 \* Created by Nick Stuivenberg on 21-6-2017.  
 \* Dit zei de manual.  
 \*/*import nl.hu.iac.domain.City;  
import nl.hu.iac.domain.Person;  
import nl.hu.iac.service.CityService;  
import org.junit.Test;  
import org.junit.runner.RunWith;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;  
import org.springframework.boot.test.context.SpringBootTest;  
import org.springframework.boot.test.mock.mockito.MockBean;  
import org.springframework.test.context.junit4.SpringRunner;  
import org.springframework.test.web.servlet.MockMvc;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import static org.mockito.Mockito.*when*;  
import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.*get*;  
import static org.springframework.test.web.servlet.result.MockMvcResultHandlers.*print*;  
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*content*;  
import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.*status*;  
  
import static org.hamcrest.CoreMatchers.*containsString*;  
  
@RunWith(SpringRunner.class)  
@SpringBootTest  
@AutoConfigureMockMvc  
public class MainTest {  
  
 @Autowired  
 private MockMvc mockMvc;  
  
 @MockBean  
 private CityService service;  
  
 @Test  
 public void shouldReturnDefaultMessage() throws Exception {  
 this.mockMvc.perform(*get*("/city/")).andDo(*print*()).andExpect(*status*().isOk())  
 .andExpect(*content*().string(*containsString*("Hello World")));  
 }  
 @Test  
 public void contextLoads() throws Exception {  
 }  
  
}

SmokeTest.java

package nl.hu.iac;  
  
*/\*\*  
 \* Created by Nick Stuivenberg on 21-6-2017.  
 \* Bestaat de controller?  
 \*/*import static org.assertj.core.api.Assertions.*assertThat*;  
  
import nl.hu.iac.controller.CityController;  
import org.junit.Test;  
import org.junit.runner.RunWith;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.test.context.SpringBootTest;  
import org.springframework.test.context.junit4.SpringRunner;  
  
@RunWith(SpringRunner.class)  
@SpringBootTest  
public class SmokeTest {  
  
 @Autowired  
 private CityController controller;  
  
 @Test  
 public void contexLoads() throws Exception {  
 *assertThat*(controller).isNotNull();  
 }  
}

CityServiceTest.Java

package nl.hu.iac.service;  
  
import nl.hu.iac.domain.City;  
import nl.hu.iac.repositories.CityRepo;  
import org.junit.Before;  
import org.junit.Test;  
import org.junit.runner.RunWith;  
import org.mockito.Mockito;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.test.context.TestConfiguration;  
import org.springframework.boot.test.mock.mockito.MockBean;  
import org.springframework.context.annotation.Bean;  
import org.springframework.test.context.junit4.SpringRunner;  
  
import static org.assertj.core.api.Assertions.*assertThat*;  
  
*/\*\*  
 \* Created by Nick Stuivenberg on 21-6-2017.  
 \*/*@RunWith(SpringRunner.class)  
public class CityServiceTest {  
  
 @TestConfiguration  
 static class CityServiceImplContextConf {  
 @Bean  
 public CityService cityService() {  
 return new CityService();  
 }  
 }  
  
  
 @Autowired  
 private CityService cityService;  
  
 @MockBean  
 private CityRepo cityRepo;  
  
 @Test  
 public void whenValidNameThenCityShouldBeFound() {  
 String name="London";  
 cityService = new CityService();  
 cityRepo = new CityRepo();  
 City cityObject = new City();  
 cityObject.setCityName("London");  
 cityService.addCity(cityObject);  
 City found = cityService.getCityByName(name);  
 *assertThat*(found.getCityName())  
 .isEqualTo(name);  
 }  
}

HttpRequestTest.java (Stond uit, want conflicteerde met andere testen)

package nl.hu.iac.controller;  
  
import static org.assertj.core.api.Assertions.*assertThat*;  
  
import org.junit.Test;  
import org.junit.runner.RunWith;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.context.embedded.LocalServerPort;  
import org.springframework.boot.test.context.SpringBootTest;  
import org.springframework.boot.test.context.SpringBootTest.WebEnvironment;  
import org.springframework.boot.test.web.client.TestRestTemplate;  
import org.springframework.test.context.junit4.SpringRunner;  
  
*/\*\*  
 \* Created by Nick Stuivenberg on 21-6-2017.  
 \* Deze test checkt na het opstarten van de server of de /city link "Hello World" terug geeft  
 \* Er dient echter nog steeds een server te draaien.  
 \*/*//@RunWith(SpringRunner.class)  
//@SpringBootTest(webEnvironment = WebEnvironment.RANDOM\_PORT)  
public class HttpRequestTest {  
/\*  
 @LocalServerPort  
 private int port;  
  
 @Autowired  
 private TestRestTemplate restTemplate;  
  
 @Test  
 public void itWorksShouldReturnDefaultMessage() throws Exception {  
 assertThat(this.restTemplate.getForObject("http://localhost:" + port + "/city",  
 String.class)).contains("Hello World");  
 }  
 \*/  
}

# Regressietests

Na een bugfix wordt er **mvn clean test** gerund. Dan worden alle testen van de applicatie gerund om te kijken of ze passen of failen