



Testplan & testrapport Integration and Communication CODE TCIF-V2IAC1-15

HBO-ICT SIE Jaar 2 2016-2017

Naam: Wesley de Kraker

Studentnummer: 1675943

Datum
12 – 03 – 2017

Versie 1.0

© Hogeschool Utrecht, Utrecht, 2017

Inhoudsopgave

Inhoudsopgave	2
1. Introductie.....	3
2. Maven project	3
3. Testomgeving	3
4. Smoketesten	4
5. Unit testen.....	5
6. Integratietesten.....	6
7. Functionele test.....	6
8. Resultaten	7

1. Introductie

Ik heb voor de volgende opdracht gekozen: registreer windmolen (CRUD). De volgende services moeten voor deze opdracht aangeboden worden: create, read, update en delete. Ik heb ervoor gekozen om de windmolens in het geheugen op te slaan.

2. Maven project

Mijn Maven project heeft de volgende dependencies:

Group ID	Artifact ID	Version
com.sun.jersey	jersey-server	1.8
com.sun.jersey	jersey-json	1.8
com.fasterxml.jackson.jaxrs	jackson-jaxrs-json-provider	2.4.1
junit	junit	4.12

Daarnaast maak mijn project gebruik van de volgende plugins:

Group ID	Artifact ID	Version
org.apache.tomcat.maven	tomcat7-maven-plugin	2.2
org.apache.maven.plugins	maven-compiler-plugin	3.6.1

Het project is getest op Apache Tomcat 7.0.75 met de volgende commando's:

- mvn clean install
- mvn tomcat7:deploy

3. Testomgeving

Ik ga alle services die ik gebouwd heb testen, dus create, read, update en delete. Deze services zijn erg afhankelijk van elkaar. Het is bijvoorbeeld alleen mogelijk om een windmolen te verwijderen als die eerst aangemaakt is.

Vandaar dat ik ervoor gekozen heb om unit testen te maken voor de foutafhandelingen en integratietesten voor de samenwerking tussen de verschillende onderdelen. Beide testen heb ik geautomatiseerd met JUnit. Daarnaast heb ik gebruik gemaakt van SoapUI om de functionele testen te automatiseren. Tot slot heb ik mijn browser gebruikt voor de smoketesten.

4. Smoketesten

Om te bepalen of mijn webservice succesvol gedeployed was heb ik telkens de volgende testen uitgevoerd:

Test	Middel
De ReadAll methode werkt: http://localhost:8080/windmillrestservice/windmills	Browser
De Read methode werkt: http://localhost:8080/windmillrestservice/windmills/1	Browser

5. Unit testen

Om de foutafhandelingen te testen heb ik de volgende unit testen uitgevoerd:

Test	Middel
Er wordt een foutmelding gegeven als er geen windmolen gevonden wordt bij een <u>read request</u> .	JUnit
Er wordt een foutmelding gegeven als er geen windmolen gevonden wordt bij een <u>update request</u> .	JUnit
Er wordt een foutmelding gegeven als er geen windmolen gevonden wordt bij een <u>delete request</u> .	JUnit

Hieronder staat de code van deze testen:

```
@Test
public void read() throws Exception {
    assertFalse("De methode read geeft het niet aan als een object
niet gevonden is!",
        checkIfWindmill1Exists());
}

@Test
public void update() throws Exception {
    assertFalse("De methode update geeft het niet aan als een object
niet gevonden is!",
        updateWindmill1ToWindmill2());
}

@Test
public void delete() throws Exception {
    assertFalse("De methode delete geeft het niet aan als een object
niet gevonden is!",
        deleteWindmill2());
}
```

De volledige code is te vinden in deze map:

windmillrestservice/src/test/java/nl/wesleydekraker/windmillrestservice

6. Integratietesten

Hieronder staan de integratietesten, de code hiervan staat in dezelfde map als de unit testen.

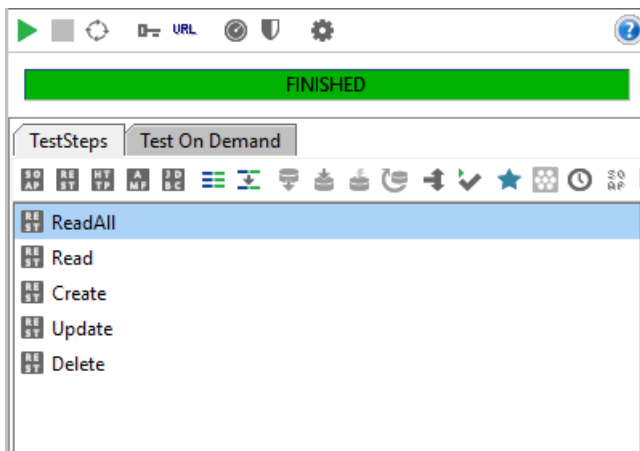
Test	Middel
Na een create request kan de windmolen ingelezen worden door een read request.	JUnit
Na een update request kan de bijgewerkte windmolen ingelezen worden met een read request.	JUnit
Na een delete request kan de windmolen niet meer opgehaald worden met een read request.	JUnit

7. Functionele test

Het SoapUI project kan teruggevonden worden in de map:

windmillrestservice /src/main/webapp/WEB-INF/rest.

Test	Middel
Een create request geeft een windmill object terug met een nieuw id.	SoapUI
Een readAll request geeft alle windmolens terug.	SoapUI
Een read request geeft één windmolen terug.	SoapUI
Na een update request wordt er een 200 code teruggestuurd.	SoapUI
Na een delete request wordt er een 200 code teruggestuurd.	SoapUI



8. Resultaten

Zoals hieronder gezien kan worden zijn alle testen geslaagd.

Smoketesten

Test	Middel	
De ReadAll methode werkt: http://localhost:8080/windmillrestservice/windmills	Browser	✓
Een create request geeft een windmill object terug.	SoapUI	✓

Unit testen

Test	Middel	
Er wordt een foutmelding gegeven als er geen windmolen gevonden wordt bij een <u>read request</u> .	JUnit	✓
Er wordt een foutmelding gegeven als er geen windmolen gevonden wordt bij een <u>update request</u> .	JUnit	✓
Er wordt een foutmelding gegeven als er geen windmolen gevonden wordt bij een <u>delete request</u> .	JUnit	✓

Integratietesten

Test	Middel	
Na een create request kan de windmolen ingelezen worden door een read request.	JUnit	✓
Na een update request kan de bijgewerkte windmolen ingelezen worden met een read request.	JUnit	✓
Na een delete request kan de windmolen niet meer opgehaald worden met een read request.	JUnit	✓

Functionele testen

Test	Middel	
Een create request geeft een windmill object terug met een nieuw id.	SoapUI	✓
Een readAll request geeft alle windmolens terug.	SoapUI	✓
Een read request geeft één windmolen terug.	SoapUI	✓
Na een update request wordt er een 200 code teruggestuurd.	SoapUI	✓

Na een delete request wordt er een 200 code teruggestuurd.	SoapUI	✓
--	--------	---