Testhandleiding
Vakoverschrijdend Project: Groep 6

Brecht Hendrickx Brecht Vercruyce Jonathan Deboosere
Hanne De Sutter Robin Antheunis Isaura Claeys

16 mei 2016

1 Back end

1.1 Vereisten

1.1.1 Software afhankelijkheden

De volgende zaken moeten correct geïnstalleerd zijn op het systeem vooralleer aan de testen kan begonnen worden:

- Java 8 JDK (Bij voorkeur door Oracle, maar alternatieve implementaties zoals OpenJDK gaan ook zolang ze versie 8 maar ondersteunen)
- Maven 3
- PostgreSQL database server
- MongoDB database server

1.1.2 Databank integratietesten

Aangezien de testen integratietesten bevatten die interageren met de productiedatabanken (namelijk postgreSQL en MongoDB), moeten deze eerst correct opgezet worden voor de testen kunnen draaien. We gaan er op dit punt vanuit dat er een werkende postgreSQL en MongoDB server beschikbaar is. Het is mogelijk om de connectie naar deze databanken in te stellen voor de testen met behulp van het volgende configuratiebestand: `backend/src/test/resources/backend.properties`. Hieronder staat een voorbeeld hoe de databank geconfigureerd kan worden. De poorten zijn hier bij de beide databanksystemen op hun standaardwaarde gebleven.

```
## PostgreSQL
postgres_host = localhost
postgres_port = 5432
postgres_dbname = voptestdb
postgres_user = dbuser
postgres_password = dbuser

## MongoDB
mongo_host = localhost
mongo_port = 27017
mongo_database = voptestdb
```

Belangrijke opmerking: het is ten sterkse afgeraden om het standaard admin account (*postgres*) van postgreSQL te gebruiken als eigenaar van een databank om veiligheidsredenen. U maakt dus beter een nieuw account aan speciaal voor deze toepassing met beperkte machtigingen.

Nadat de test databank zelf is aangemaakt, moet deze opgevuld worden met tabeldefinities. Hiervoor is een `SQL` script aangemaakt. Je kan dit terugvinden in `database/milestone2/sql_scripts/create_script.sql`. Zorg er zeker voor dat de eigenaar van zowel de databank als de tabellen overeenkomt met de gebruiker zoals aangegeven in het configuratiebestand.

1.2 Testen uitvoeren

Maven biedt een zeer eenvoudige command line oplossing voor het uitvoeren van alle testen en het genereren van coverage rapporten. Ga naar de hoofdfolder van het project (deze bevat het `pom.xml` bestand) en voer daar het volgende commando uit:

```
$ mvn clean org.jacoco:jacoco-maven-plugin:prepare-agent org.jacoco:
    jacoco-maven-plugin:prepare-agent-integration install test
```

De testen beginnen te lopen en er wordt gerapporteerd welke falen en slagen. Als alle testen slagen worden er automatisch rapporten aangemaakt met de coverage van de testen. Deze kunnen makkelijk in html vorm bekeken worden in `target/site/jacoco/index.html`.

Uiteraard kunnen de testen ook uitgevoerd worden met eender welke IDE met ondersteuning voor Maven. De werking verschilt tussen verschillende IDE's dus we gaan hier niet verder op in.

2 Front end

Voor het runnen van de front end tests moet npm geínstalleerd zijn, alsook het karma-jasmine package van Node,js. Men kan dan in de map *frontend/app* op de commandline het volgende commando uitvoeren `npm test`. Dit zal alle testen in een keer uitvoeren. Bij het uitvoeren van de testen wordt een map *coverage* gegenereerd. Alle informatie in verband met coverage kan gevonden worden door *coverage/lcov-report/index.html* te openen.