Universiteit Gent

Software Engineering Lab 2

Test handleiding

Groep 2

Academiejaar 2020-2021

Contents

1 Backend

In de directory "/backend/api/api" kan je via het commando *mvnw test* automatische tests uitvoeren. Het framework dat we gebruiken hiervoor is Junit4. We testen volgende zaken:

- datajpatests (test of de foreign keys updaten, of opslag werkt zoals verwacht)
- serializationtests (kunnen DTO's naar JSON omgezet worden en terug?)
- AbstractTest (Template voor testen)
- Api Applicatie (laad de applicatie ?)
- Organisaties (Kunnen alle organisaties opgehaald worden, kan een organisatie aangemaakt worden ...)
- Pakketten (Kunnen alle pakketten opgehaald worden, kan een pakket aangemaakt worden ...)
- Machtigingen (Kunnen alle machtigingen opgehaald worden, kan een machtiging aangemaakt worden ...)
- Aanvragen (Kunnen alle aanvragen opgehaald worden, kan een aanvraag aangemaakt worden ...)
- Diensten (Kunnen alle diesnten opgehaald worden, kan een dienst aangemaakt worden ...)
- Gebruikers (Kunnen alle gebruikers opgehaald worden, kan een gebruiker aangemaakt worden ...)

1.1 Milestone 2

Om makkelijker te kunnen testen hebben we bepaalde functionaliteit in utils klassen geplaast. Zo testen we een bepaalde feature zonder tussenkomst van een mock database of zonder een server op te zetten.

Foutmeldingen worden nu ook getest.

1.2 Live databank testen

Om delen van de SQL-code te testen, moet een environment variabele `LIVETESTING="1"` ingesteld staan. Dit kan op Linux met het commando `export` en op Windows met het commando `set`. Het is niet mogelijk om deze testen uit te voeren op een systeem zonder de PostgreSQL-databank eerst te installeren. Dit is de reden waarom deze testen niet standaard uitgevoerd worden.

2 Frontend

In de directory "/frontend" kan je via volgende commando's automatische tests uitvoeren:

- npm run test
- npm run test components
- npm run test views
- npm run test app
- npm run test homepage
- npm run test userpage

- npm run test organisationpage
- npm run test userform
- npm run test organisationform
- npm run test contactform

npm run test voert alle tests uit. Het optionele argument dient om enkel specifieke testen uit te voeren. We testen of de GUI correct gedrag vertoont (zoekveld filtert correct, input wordt gevalideerd, ...) Als framework gebruiken we de vue-testing-library en vue-test-utils. Er zijn wel enkele problemen omdat wij vue3 gebruiken. Dit komt doordat deze frameworks oorspronkelijk in Vue2 zijn geschreven en niet volledig gemigreerd zijn naar Vue3.

3 Datadog

Datadog[1] dient als een onafhankelijke uptime monitor. Deze controleert dus als alle diensten, die onze server levert, online zijn. Daarbij wordt ook het SSL-certificaat gecontroleerd. Zo vermijden we dat het certificaat dreigt te vervallen zonder dat wij hiervan op de hoogte zijn. We gebruiken deze dienst voor het controleren van volgende pagina's:

- sel2-2.ugent.be
- sel2-2.ugent.be/data
- sel2-2.ugent.be/app/
- sel2-2.ugent.be/api/users (+ controle content type, json)
- sel2-2.ugent.be/api/ (+ controle content type, json)
- sel2-2.ugent.be/api/proposals (+ controle content type, json)
- sel2-2.ugent.be/api/organisations (+ controle content type, json)
- sel2-2.ugent.be/api/permissions (+ controle content type, json)

4 Jenkins test

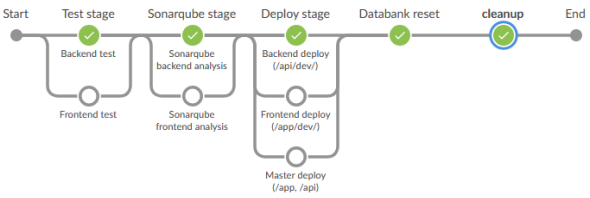

Figure 1: Voorbeeld pipeline run van jenkins

We gebruiken Jenkins voor onze continous integration. In samenwerking met de Blue Ocean plugin (pipeline editor) hebben we onze pipeline verdeelt in vier grote delen. Dit is ook te zien in onze pipeline diagram (Figure 1).

1. De eerste stage bestaat uit de frontend test en de backend tests. In de vorige secties staat hoe deze gestart worden via de command line.

[1] www.datadoghq.com/

2. In het tweede deel wordt de code geanalyseerd door sonarqube. deze berekent de code coverage.
3. Het derde deel zal de code deployen afhankelijk van op welke branch er gecommit is.
   (a) De backend branch zal gebouwd worden en de api-dev service herstarten. Zo zal /api/dev altijd van de laatste aanpassing van de branch gebruik maken.
   (b) De frontend branch zal de vue files bouwen, zo zal /app/dev altijd de laatste aanpassing weergeven.
   (c) De master branch zal hetzelfde uitvoeren als de backend en frontend branches ma zal de aanpassingen zichtbaar maken op /app en /api.
4. In "db-reset" zal men de database leegen en de tabellen verwijderen. Dan vullen we de database terug op met eventueel aangepaste data. Zo zijn zeker dat development databank altijd in data consistentie is en de juiste tabellen gebruikt.
5. In het laatstse deel wordt er een cleanup uitgevoerd. Jenkins houdt normaal een kopie van de repo bij (voor artifacten en sneller te pullen). Dit geeft echter snel problemen op een server met een beperkte opslagcapaciteit.