



UNIVERSITEIT GENT

SOFTWARE ENGINEERING LAB 2

Test handleiding

Groep 2

Contents

| | | |
|----------|---------------------|----------|
| 1 | Backend | 2 |
| 2 | Frontend | 2 |
| 3 | Datadog | 3 |
| 4 | Jenkins test | 3 |

1 Backend

In de directory `"/backend/api/api"` kan je via het commando `mvnw test` automatische tests uitvoeren. We testen volgende zaken:

- `datajpatests` (test als foreign keys updaten)
- `serializationtests` (kunnen objecten naar json omgezet worden en terug?)
- `AbstractTest`
- `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 diensten opgehaald worden, kan een dienst aangemaakt worden ...)
- `Gebruikers` (Kunnen alle gebruikers opgehaald worden, kan een gebruiker aangemaakt 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.

3 Datadog

Datadog¹ 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 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/data
- sel2-2.ugent.be/app
- sel2-2.ugent.be/api/users (+ controle content type, json)

4 Jenkins test

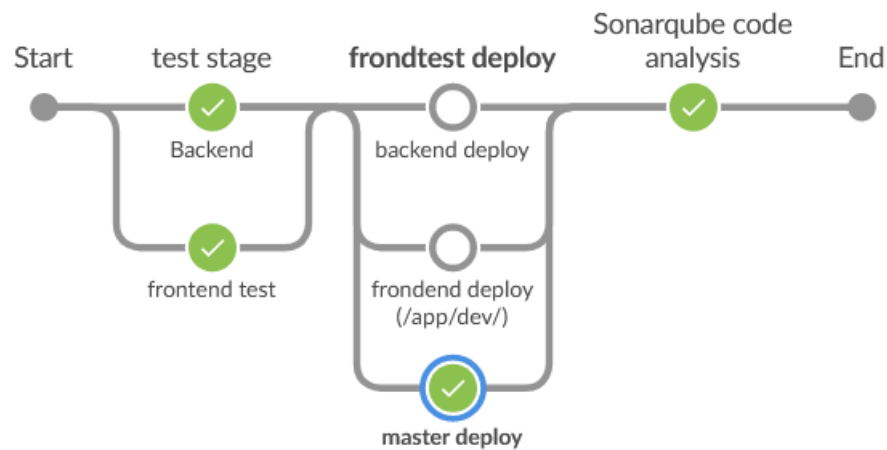


Figure 1: Voorbeeld pipeline run van jenkins

We gebruiken Jenkins voor onze continous integration. Te samen met de Blue Ocean plugin (pipeline editor) hebben we onze pipeline verdeelt in 3 stages (delen). Dit is ook te zien in onze pipeline diagram 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. Het tweede deel zal de code deployen afhankelijk van op welke branch er gecommited is². In het laatste deel wordt de code bekeken door Sonarqube.

¹<https://www.datadoghq.com/>

²master wordt gedeployed op /api