Front end testing report

# Our approach towards front end testing

Voordat ik ben gaan zoeken naar een front end testing framework heb ik gekeken naar de eisen waaraan deze moet voldoen voor mijn Koffie App. Hieruit zijn de volgende eisen gekomen:

* Het moet kunnen werken met Angular
* Automatisch testen moet mogelijk zijn
* Het moet kunnen werken met TeamCity

Aan de hand van de bovenstaande eisen is er gezocht naar een of meerdere passende oplossingen. Per oplossing zal ik beschrijven wat het precies inhoudt, de testen die ik heb gedaan met de oplossing en mogelijke problemen die ik ben tegen gekomen tijdens het testen van deze oplossingen.

## Protractor

Protractor is een end-to-end test framework voor Angular en Angular JS. De testen worden binnen je applicatie uitgevoerd in een echte browser. De interactie is zoals een gebruiker het zou doen. Protractor is een Node.js programma die standaard het Jasmine test framework gebruikt in combinatie met een lokale standalone Selenium server.

De tutorial op de Protractor website is heel simpel om uit te voeren en zorgt ervoor dat je binnen een korte tijd testen kan uitvoeren op hun eigen voorbeeld applicatie.

Het uitvoeren van de testen aan de hand van Protractor werkt op dit moment alleen nog maar in Chrome. Bij FireFox en Edge komt de foutmelding dat Protractor hun webbrowser drivers mist om de testen uit te kunnen voeren. Het is ook nog niet gelukt om het ook werkend te krijgen in FireFox en Edge.

Link naar de Protractor tutorial: <https://www.protractortest.org/#/tutorial>

## Karma

Karma is een tool die een webserver aanmaakt om zo de testen uit te voeren binnen iedere browsers die ermee verbonden is. De developer kan vervolgens de resultaten in een command line terug zien. Karma is gemaakt door het AngularJS team omdat de afhankelijk zijn van testen en ze altijd op zoek zijn naar betere tools om het makkelijker te maken. Op de website van Karma is een tutorial te vinden over de integratie met TeamCity.

Ook Karma is een Node.js programma. Op hun FAQ pagina van de website geven ze aan dat Karma voornamelijk ontworpen is voor low level (unit) tests. Het is mogelijk om het met een plugin in te doen maar ze raadden Protractor aan voor high-level tests.

Link naar Karma: <https://karma-runner.github.io/2.0/index.html>

# Testing framework choice

Welke + argumentatie

# How do I test my application on a regular basis

Hoe doe ik testen en met welke tools?

Bewijslast toevoegen

Indien het niet gelukt is, waarom niet en beschrijf wat er geprobeerd is

# Testcases

|  |  |  |
| --- | --- | --- |
| TESTCASE 1: Drink toevoegen | |  |
| Te gebruiken gegevens: | | |
| Breid het aanbod van de koffie automaat uit met het volgende:  Naam: Warme chocomel | | |
| Verwacht resultaat: | **Geslaagd?** | |
| Het item is toegevoegd |  | |
| Opmerkingen: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| TESTCASE 2: Drink wijzigen | |  |
| Te gebruiken gegevens: | | |
| Zoek tussen het aanbod van de koffie automaat naar “Warme chocomel” en pas deze aan naar het volgende:  Naam: Warme melk | | |
| Verwacht resultaat: | **Geslaagd?** | |
| Het item is aangepast naar warme melk. |  | |
| Opmerkingen: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| TESTCASE 3: Drink verwijderen | |  |
| Te gebruiken gegevens: | | |
| Verwijder het volgende uit het aanbod van de koffie automaat:  Naam: Warme melk | | |
| Verwacht resultaat: | **Geslaagd?** | |
| Het item is verwijderd |  | |
| Opmerkingen: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| TESTCASE 5: Order toevoegen | |  |
| Te gebruiken gegevens: | | |
| Voeg de volgende bestelling toe:  Naam werknemer: Henk  Drinken: Koffie  Suiker: 2  Melk: 1 | | |
| Verwacht resultaat: | **Geslaagd?** | |
| De bestelling is toegevoegd |  | |
| Opmerkingen: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| TESTCASE 6: Order wijzigen | |  |
| Te gebruiken gegevens: | | |
| Pas de zojuist aangemaakt bestelling van Henk aan naar het volgende:  Naam werknemer: Henk  Drinken: Cappuccino  Suiker: 1  Melk: 0 | | |
| Verwacht resultaat: | **Geslaagd?** | |
| De bestelling is aangepast |  | |
| Opmerkingen: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| TESTCASE 7: Order verwijderen | |  |
| Te gebruiken gegevens: | | |
| Verwijderd de volgende bestelling:  Naam werknemer: Henk  Drinken: Cappuccino  Suiker: 1  Melk: 0 | | |
| Verwacht resultaat: | **Geslaagd?** | |
| De bestelling is verwijderd |  | |
| Opmerkingen: | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| TESTCASE 8: Orders bekijken | |  |
| Te gebruiken gegevens: | | |
| Bekijk de lijst met alle huidige bestellingen door naar de homepagina van de koffie app te navigeren. | | |
| Verwacht resultaat: | **Geslaagd?** | |
| Er is een lijst met alle huidige bestellingen te zien. |  | |
| Opmerkingen: | | |
|  | | |

**Links die nog bekeken moeten worden:**

* <http://karma-runner.github.io/3.0/plus/teamcity.html>
* <https://medium.com/@mgd4375/building-an-angular-cli-project-in-teamcity-714a894f7083>
* <https://blog.jetbrains.com/webstorm/2017/02/your-first-unit-test-using-angular-cli-karma-and-webstorm/>
* <https://teamcity-support.jetbrains.com/hc/en-us/community/posts/115000251564-Need-help-for-set-up-build-steps-for-the-angularjs-tests>

**Canvas opdracht:**

The answer to these questions is what this assignment is all about. Make a report of your findings, approaches you have taken and results you have achieved.

You will receive a feedback on your report. The feedback will be provided via comments to this assignment.

This is an iteratively improved assignment.

Work on it and whenever you have made an improvement - resubmit it. Add short description of what you have done. This gives the teacher a notification to give you more feedback.

Form of the assignment is rather free, just keep it functional and understandable. The main story should be clearly visible.

What is expected:

* some form of an evidence of how you have approached the frontend testing, what alternatives you have looked at and why you have choosen a specific solution
* some form of an evidence that relevant tools are used properly
* some form of an evidence that frontend testing is used in your project on regular basis

The evidence can be for example designs, comparision criteria, test reports, links to a relevant test code etc.

In case you have failed to test your frontend automatically also describe what have you done. The main objective is to see you have worked in a systematic way towards achieving quality of your frontend.

It is expected that each team member has gained relevant applied knowledge. It is thus imperative that all of you work on this assignment. In case your contribution is not clearly visible you may be assigned with additional assignment to prove your level of applied knowledge.