|  |
| --- |
|  |
| William Wong |
| Bachelor Toegepaste Informatica  Afstudeerrichting? |
|  |
|  |
| Software Testing  Frank Serneels |
|  |
| Academiejaar 2021-2022 |

|  |  |
| --- | --- |
| Udemy cursus - week 6: toepassing geziene leerstof | |
|  |

# Week 6 Software testing: Oefeningen Udemy

In dit deel zal ik, per section van de Udemy cursus, toelichten welke oefeningen ik heb gemaakt.

# Section 10 oefeningen:

In de eerste oefening zal ik een request meerdere malen proberen uit te voeren door gebruik te maken van pre-request scripts en een data set. De bedoeling is dat de request even vaak wordt uitgevoerd als er gegevens zijn in de data set.

A picture containing chart

Description automatically generated

Allereerst geef ik een url en een body mee, die gebruik zal maken van een global variable, die ik in de pre request script zal definiëren.

Graphical user interface, text, application

Description automatically generated

In de pre request script definieer ik een data set met meerdere gegevens, nl companies. Ook definieer ik een variable met currentcompany die aangeeft welke company er momenteel in gebruik is.

Graphical user interface

Description automatically generated with medium confidence

Vervolgens schrijf ik een test die de data set (array) doorloopt en telkens als er een item is, moet er een post request gebeuren.

Graphical user interface, application, Teams

Description automatically generated

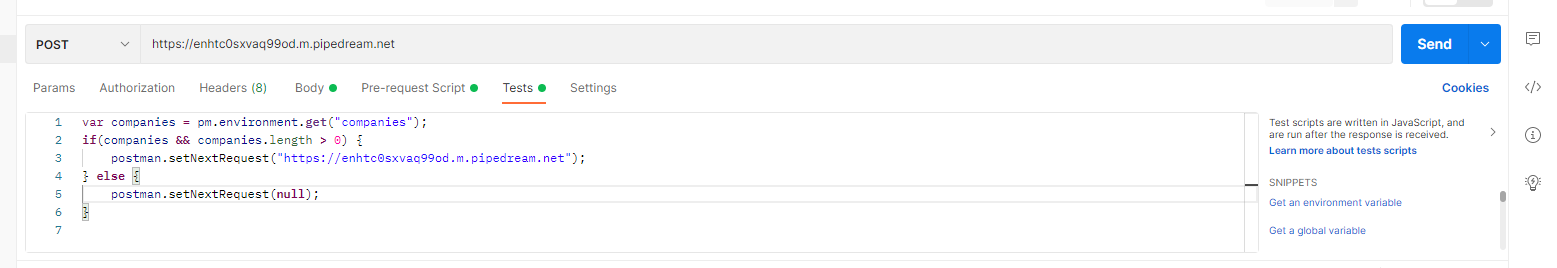
A picture containing graphical user interface

Description automatically generated

Rectangle

Description automatically generated with medium confidence

Nu probeer ik de tests te runnen, maar dit geeft echter een fout. Ik had geen permission om een post request te doen op de url, dus heb ik zelf een requestbin moeten aanmaken. Dit heb ik ook gedaan, waardoor het wel werkt. Ik heb de tests dan aangepast met de correcte url.



Graphical user interface, text, application, email

Description automatically generated

Er zijn 4 gegevens in onze data set en er worden ook 4 post requests uitgevoerd.

In de tweede oefening zal ik gebruik maken van externe data file om een flow te creëren.

Graphical user interface, text, application, email

Description automatically generated

De eerste get request zal een reference creëren, die gebruikt zal worden in de volgende request, namelijk: create order request.

In de create order request, zal ik een order creëren met een reference id, die ik dynamisch krijg van de vorige get reference request. Bij deze create order request zal ik ook een customer id en product id meegeven.

Graphical user interface, text, application, email

Description automatically generated

De body die ik meegeef bij deze request is hard coded. Deze zal dan ook gecontroleerd worden in de tests die geschreven zijn.

Graphical user interface, text, application, Teams

Description automatically generated Graphical user interface, application

Description automatically generated

De derde request zal gewoon de status checken om na te gaan of alles gelukt is.

Graphical user interface, text

Description automatically generated with medium confidence

Ik zal nu de collection runnen.

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Alles werkt. Maar wat als ik een iteratie wil van request met verschillende dynamisch gegevens. Dit lukt dan niet aangezien de body bij post request (create order) hard coded is. We moeten dus de body aanpassen om gegevens dynamisch te kunnen meegeven:

Graphical user interface, application

Description automatically generated

Vervolgens zullen we de collection weer uitvoeren, maar we geven een externe data file mee.

Graphical user interface, application

Description automatically generated

We klikken op preview om onze data van de externe file te bekijken.

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated with medium confidence

In de data file zitten er drie rijen, waardoor er ook 3 iteraties gebeuren. Onze tests werken dus.

# Section 12:

Voor deze section zal ik een mock server aanmaken in Postman, via de stappenplan in de Udemy cursus.

Allereerst moet ik een url meegeven en een response body voor de mock server:

Graphical user interface, text, application, email

Description automatically generated

Nadien krijg ik een url mee van postman:

Graphical user interface, text, application

Description automatically generated

Om de mock server uit te testen, moet ik eerst de environment selecteren, die automatisch door postman wordt aangemaakt na het aanmaken van een mock server. Doe ik dit niet, dan krijg ik een error omdat de url variable nergens anders is gedefinieerd. De request werkt ook, wat wilt zeggen dat de mock server ook werkt.

A screenshot of a computer

Description automatically generated

A picture containing table

Description automatically generated

Bovenstaande pijl geeft de url variable van de aangemaakte environment weer.

Dit is misschien een simpele oefening, maar wel erg handig voor als ik enkel een front end heb van de applicatie. Dit kan dan helpen om de back end te bepalen. Bovendien kan ik dan ook testen uitvoeren voor de front end, zonder een bestaande api op een server en ik kan een werkende front end hebben, zonder rekening te houden met het feit dat de API nog niet bestaat.

# Section 13:

Ik heb toch besloten om deze opdracht te doen, omdat het belangrijk is voor mijn project. Ik zal in deze opdracht meerdere files posten met behulp van Newman. Hiervoor ga ik aan automated testing doen via Newman. Om automated tests met newman uit te voeren, moet ik eerst de collection van Postman exporteren.

Graphical user interface, application, Teams

Description automatically generated

Vervolgens moet ik mijn command line interface openen en naar de folder gaan waar ik mijn files heb opgeslagen.