|  |
| --- |
|  |
| 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

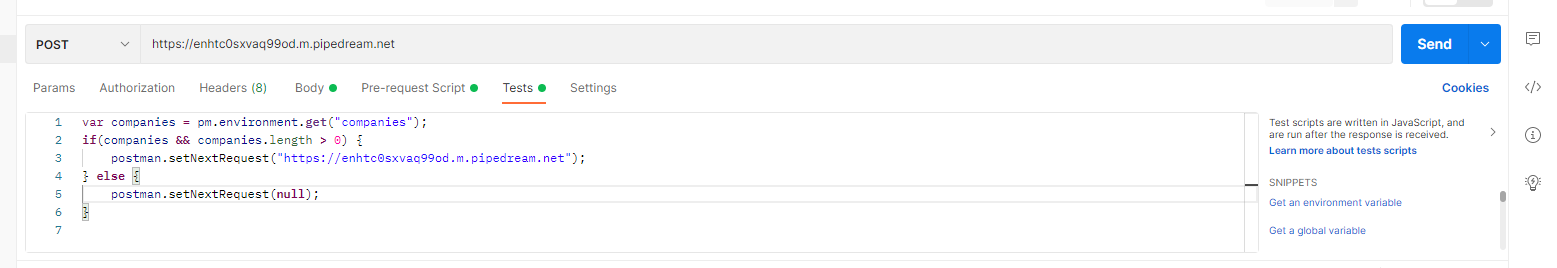
A picture containing graphical user interface

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:

De oefeningen van deze section heb ik niet gemaakt, maar ik zal deze later wel toepassen in mijn opdracht. Dit lijkt mij meer zinvol aangezien de tutorial in udemy gebruik maakte van een persoonlijke backend die ik niet ter beschikking heb.

# Section 14:

Postman biedt een API aan voor basic authentication, nl postman echo. Link: <https://postman-echo.com/basic-auth>.

Graphical user interface, text, application, email, Teams

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Deze geeft als response terug unauthorized. De headers geeft aan dat ik basic authentication moet meegeven. Dit doe ik door via de headers een authorization en passwoord mee te geven.

Graphical user interface, text, application

Description automatically generated

Nu lukt de request wel.

Background pattern

Description automatically generated with low confidence

Dit is een korte intro over basic authentication in Postman.

Voor de volgende oefening zal ik via postman basic authentication uitvoeren op een bestaande api. Hiervoor gebruik ik de imgur api.

Graphical user interface, text, application, email

Description automatically generated

Dit geeft mij een unauthorized response terug. Dit komt omdat ik geen token heb meegegeven. Ik open een nieuwe tab in postman en ga naar de authorization tab.

Graphical user interface, text, application

Description automatically generated

Ik ga naar get new access token. Via deze tab kan een nieuwe authorization code aanmaken. Deze geeft ook een callback url mee die ik aan de form van imgur moet meegeven.

Graphical user interface, application

Description automatically generated

Vervolgens zal ik een registratie form invullen bij imgur.

Graphical user interface, text

Description automatically generated

In de authorization callback url zal ik de callback url meegeven die postman meegaf: <https://app.getpostman.com/oauth2/callback>. Als alles gelukt is, krijg ik een client id en secret.

Text

Description automatically generated

Deze geef ik mee aan postman:

Graphical user interface, text, application

Description automatically generated

Als alle velden ingevuld zijn, klik ik op request new token:

Graphical user interface, text, application, chat or text message

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Dan moet ik mij authenticeren via imgur, deze pop up komt automatisch. Dit laat toe dat postman gegevens gebruikt van mijn imgur account. Vervolgens krijg ik van postman een token. Deze geef ik dan mee in mijn oorspronkelijke get request, waardoor de request werkt:

Graphical user interface, text, application, email

Description automatically generated

# Section 15:

Geen oefeningen, aangezien het hoofdstuk meer een Q&A was.

# Section 16:

Oefeningen niet gemaakt, want ik zal hier geen gebruik van maken.

# Section 17:

Ik kan al met javascript onderweg, aangezien ik veel over javascript heb gezien in mijn school carrière. Ik heb echter een aantal simpele zaken geleerd van udemy die heel handig zijn.

Nieuwe zaken die ik heb geleerd van Udemy over javascript:

Ik heb geleerd dat men data types kan opvragen via de console in een browser:

Graphical user interface, text, application

Description automatically generated

Ik kan ook simpele zaken controleren, dit is zeer handig:

Graphical user interface, application

Description automatically generated

Oefening:

Random email generator:

Graphical user interface, text, application, email, Teams

Description automatically generated

Oefening:

Geef alle actieve employees die 10 jaar of langer werken:

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

De oefeningen rond deze section was niet zo moeilijk, aangezien ik al veel van javascript heb gezien in Odisee. Het interessante aan dit hoofdstuk is dat basic javascript ook in postman gebruikt kan worden.