w

Afbeelding met tekst, illustratie

Automatisch gegenereerde beschrijving

Postman

**Christian Ngirinshuti**

r0813058

academiejaar: 2020-2021 Toegepaste informatica

3TI

Software Testing

Patrick Van den Broeck

Patrick Van den Broec

Franck Serneels

Inhoudsopgave

[Week 4/10-10/10 hfst 1 3](#_Toc92915523)

[15/11/2021 hfst 2 4](#_Toc92915524)

[16/11/2021 hfst 3 7](#_Toc92915525)

[17/11/2021 hfst 2 7](#_Toc92915526)

[27/11/2021 hfst 4 10](#_Toc92915527)

[28/11/2021 hfst 4 11](#_Toc92915528)

[Eindpresentatie Postman 13](#_Toc92915529)

Verslag Postman course

## Week 4/10-10/10 hfst 1

* Gesprek met de docent
* Na het gesprek heb ik een cursus aangekocht
* Ik heb een nieuwe repository aangemaakt
* Ik heb de eerste hoofdstuk bekeken
* Hoofdstuk 1: introductie tot postman
* Ik heb postman geïnstalleerd op mijn computer
* Get request uitgevoerd op een simpele url
* Kort uitleg gekregen wat json is.(data structuur)
* Geleerd hoe je parameters toegevoegd in een url

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Postman is een applicatie die wordt gebruikt voor API-testen. Het is een HTTP-client die HTTP-verzoeken test, met behulp van een grafische gebruikersinterface, waardoor we verschillende soorten reacties verkrijgen die vervolgens moeten worden gevalideerd

## 15/11/2021 hfst 2

Bij het openen van een post request , wordt de body automatisch geactiveerd. Dit komt omdat post request data kan sturen met de body.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Als je op body klikt kan je data aanpassen door op raw en json te klikken .Vervolgens kan je de ruwe data aanpassen van de server. Onderaan heb je de output van de server door op send te klikken. Je kan de request opslaan door op save te drukken. Je moet eerst een collectie aanmaken.

Je kan waardes van een request weergeven in een andere request door de requests met elkaar te linken door gebruik te maken van testen.

in de volgende voorbeeld heb je een get en een post request. De get weergeeft een ID die je in de post request wilt weergeven.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Nadien pas je de ruwe data door id toe te voegen. Zodat de data van de get request kan doorgestuurd worden naar de post request.

Afbeelding met tekst

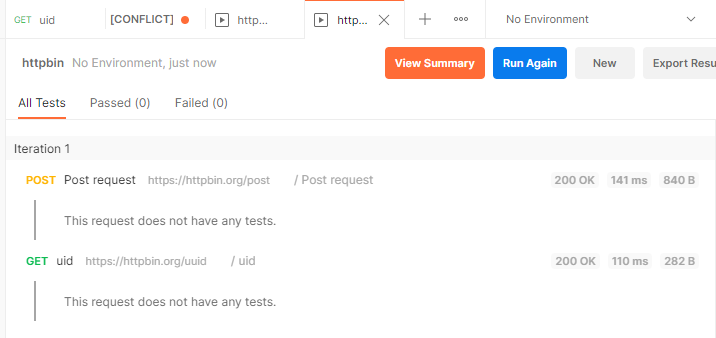
Automatisch gegenereerde beschrijving

Je kan met testen controleren of de status code gelijk is aan 200, indien het 200 dan weet je of een test werkt.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Met de run tab kan je alle request binnen een collectie laten uitvoeren. Je kan ook zelf bepalen welke request je niet wilt uitvoeren.



## 16/11/2021 hfst 3

Postman wordt gebruikt om je API te documenteren en je kan op Postman voorbeeld ontwikkelen voor request.

Een HTTP request bestaat uit 4 delen:

* Request methode
* URL
* Header
* Body

Request mode bestaat uit een lijst van methode. In begin gebruik je vooral de volgende methodes:

* Post: Update => add
* Put: Veranderen van een sigular resource
* patch : Lijst van veranderingen
* delete: Verwijderen

Postman zorgt ervoor dat je begrijpt wat er met je request gebeurt. De verschillende hulpmiddelen om de request beter te begrijpen zijn:

* statuscode
* body
* header

Met de status code kan je zien of een request of test werkt (=200). Als je op de 200 vindt je meer informatie. Naast de status code is er een tijd vermeld. Met de tijd bedoelt men hoe lang het duurt voor een request wordt uitgevoerd.

Als je een error kan in de body de dataformaat aanpassen naar bv. Json zodat het beter leesbaar is.

In de header kan je informatie terug vinden van de verschillende waardes.

## 17/11/2021 hfst 2

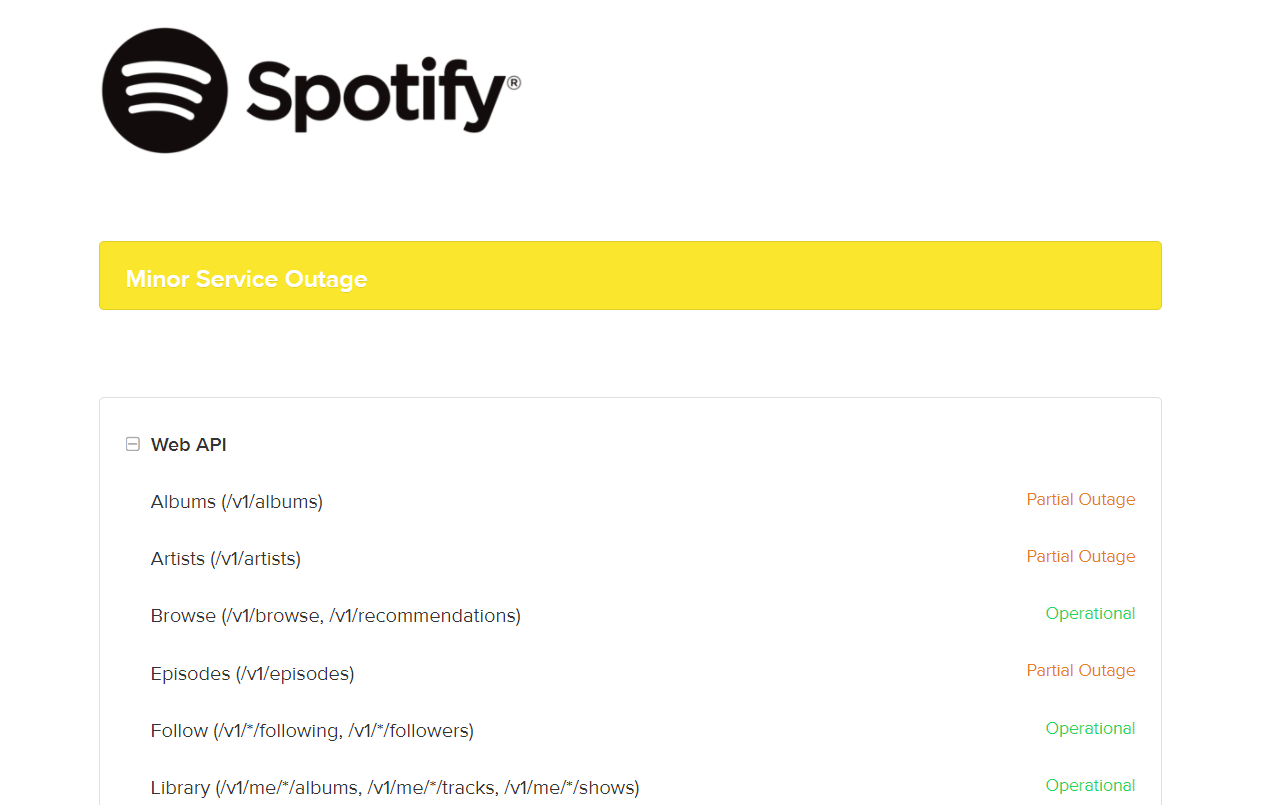
Je kan met behulp van postman troubleshooten indien een request niet lukt. Eerst moet je controleren of een geen type staat in de URL. Als de URL klopt dan moet je nagaan of de service die je aanspreekt beschikbaar is. Als de service beschikbaar is ,moet je in jouw browser de type domainnaam of IP typen.

In de onderstaande voorbeeld klopt mijn url niet. Ik type example.con in plaats van example.com.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

De status van een service kan je online altijd op zoeken een voorbeeld is de API status van Spotify.



Als er nog steeds problemen kan je de postman console gebruiken. Op deze console kan je technische informatie terugvinden over de error.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## 27/11/2021 hfst 4

Ik heb op 27-11-2021 geleerd hoe je een test moet maken op postman. Een test is simpelweg een bewering. Een test heeft als doel een waarde controleren. Je gaat de waarde vergelijken t.o.v een waarde die je normaal verwacht. Een test kan enkel werken nadat een request is gelukt.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

* get request: example.com
* in de bovenstaande afbeelding ziet u een voorbeeld van een gelukte test

Als example.con gebruik, krijg ik een andere melding.

* **Pm.test**: is een fuctie in postman om test specificaties te schrijven
* De eerst parameter in de functie is een string en de tweede parameter is callback functie
* Pm.response = reponse assertion API

Afbeelding met tekst

Automatisch gegenereerde beschrijving

* 404: wanneer het gevraagde bestand of URL niet bestaat

## 28/11/2021 hfst 4

Ik heb op 28/11/2021 geleerd hoe je een API kan testen. Hiervoor moet je eerst een account maken op Trello. Trello is een tool om projecten te beheren. Je kan op Trello boards maken.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

We gaan gebruik met van de api van Trello. Je moet eerst naar de volgende URL gaan.

* <https://developer.atlassian.com/cloud/trello/guides/power-ups/your-first-power-up/>

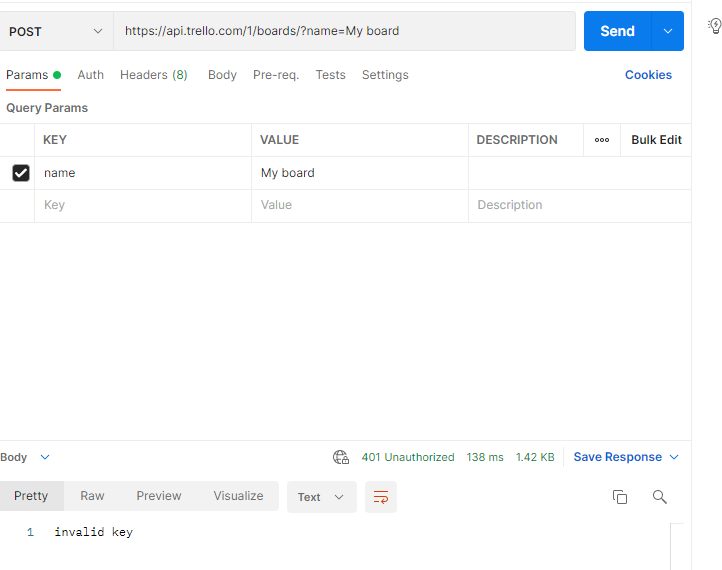
Dan klik je op references. In References klik je bij board.

Als oefening maken we een board via postman hiervoor gebruik je de url die op de website staat.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Nadien ga je op postman en maak je post request. Hiervoor heb je een Key nodig.



Je vindt de key terug als in logt en klikt op : <https://trello.com/app-key>

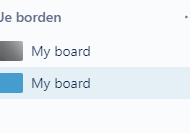
* je hebt ook een token nodig

Dit is het resultaat. Dankzij de post request wordt een nieuwe board aangemaakt.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Op trolleo zou je nu 2 verschillende borden moet hebben.



# Eindpresentatie Postman

Mijn presentatie gaat over een voorbeeld van een automated testing met postman. Hiervoor gebruik een simple API van iemand zijn github repository. De API bevat een lijst met films.

Eerst moet je node js installeren. En dan ga je naar je terminal en daar verplaats naar de folder. En dan voer je de volgende commando uit:

* Install npm
* Np m run dev

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Vervolgens ga je op postman om de APi testen.

Eerst controleren we of de server werkt. Hiervoor maken we postrequest met de url: <http://localhost:3000/>

Als je op send klikt, krijg je als repons.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Dit kom omdat je in de header een key value bent vergeten toe te voegen.

Key value:



Afbeelding met tekst

Automatisch gegenereerde beschrijving

Nu weet je dat de client kan communiceren met de server. En weet je dat de API werkt.

Nadien maak je met de API een user aan met een Post request.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Dan vraag ik een token aan voor de nieuwe user die ik net heb aangemaakt (met een post request).

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Met een post request kan je een lijst opvragen met alle films

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Je kan ook een specifieke film opvragen door in de url met id nummer van een specifiek film.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Aanmaak van een nieuwe film. In de body voeg je json toe en in de header plaats je de bearer token.

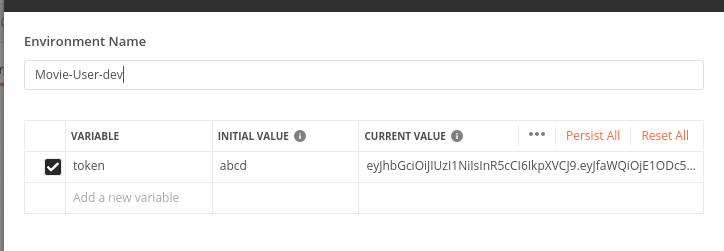
Afbeelding met tekst

Automatisch gegenereerde beschrijving

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Dan maak ik een dev enviroment aan. Waarin ik variabels kan opslaan. Doordat ik geregeld een token moet gebruik, heb ik besloten om variable aan te maken voor de Bearer token.





Ik heb request aangemaakt om een film te verwijderen.

* DELETE http://localhost:3000/api/movies/:id

Nadien maak je een collectie waarin je alle verzoek bijeen verzamelt. Ik heb elke verzoek een naam geven

Nu voegen we testen toe.

Fail check

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Check gelukt

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Nieuwe user aangemaakt

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Ik heb een prescript aangemaakt. Om te voorkomen dat ik de volgende melding krijg : “email already in use “(duplicate) telkens als de collectie herstart.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Login en Aanvraag token

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Eerst test je of de user is ingelogd. Nadien kijkt ment

Opvragen lijst films

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Nieuwe film aanmaken

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Specifieke film aanvragen

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Film verwijderen

