Samiullah hafiz **Docent**: Serneels Frank

OPDRACHT : POSTMAN TESTING

2021-2022

# Opzoekwerk : postman

Postman is een applicatie die je op jou computer kunt downloaded (gratis) via hun website. Het is een API testing tool die wordt gebruikt om responses te krijgen van de API. Postman is een HTTP client die HTTP requests Test en als de API goed responses teruggeeft. Postman gebruikt een gebruiksvriendelijke userinterface. Waarmee we meerdere requests kunnen testen.

* GET: Obtain information
* POST: Add information
* PUT: Replace information
* PATCH: Update certain information
* DELETE: Delete information

Postman heeft de mogelijkheid om verschillende requests te groeperen. Dit noemen wij collections. Dit helpt om de testen te organiseren en een structuur aan te brengen.

Deze collections zijn folders waarin de testen zich bevinden. En daarme kan de team een structuur aanbrengen. Er is ook een mogelijkheid om te importeren of exporteren.

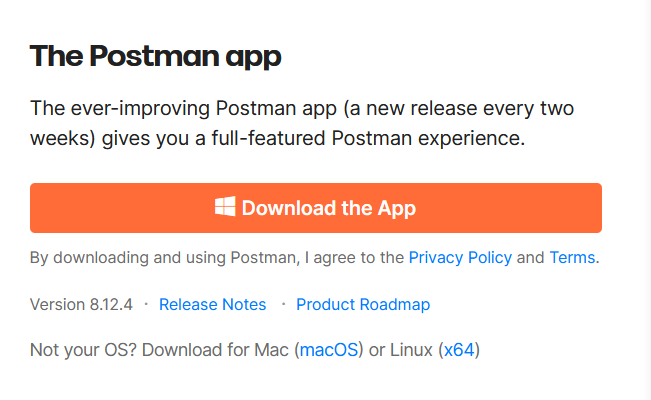
Postman laat ook toe om environments te maken. daarmee kunnen we een URL variable in meerdere environments laten testen met verschillende requests.

Met postman kan je ook debuggen namelijk door zijn console. Daarmee is het makkelijker op gegevens op te volgen en te verbeteren.

Hoe download je postman ?

Het is heel makkelijk om dit programma in te stellen.

Je kunt eerst op hun [website](https://www.postman.com/downloads/) gaan

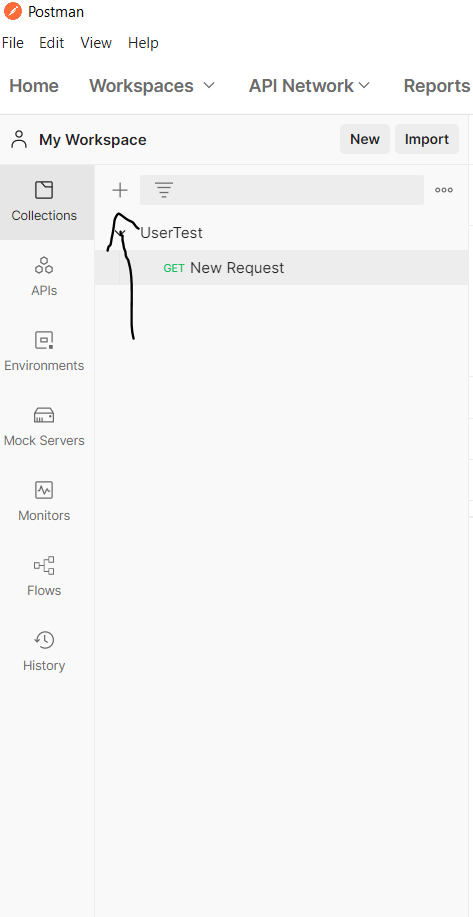


Daarna kan je hierop klikken. En daarna start de applicatie automatisch op en kan je beginnen te testen.

# Week 04/10/2021

### Collectie aanmaken

Ik heb hier aangeleerd hoe een collectie aanmaken.



Je moet hier gewoon een naam ingeven.

Une image contenant texte

Description générée automatiquement

* Daarna kan je kiezen als je een **authorization** wilt opzetten (security).
* Er zijn ook **pre-request** scripten die meestal in javascript worden geschreven die automatisch worden uitgevoerd bij een request
* Je kunt hier ook specifieke **testen** schrijven bv. Als je een request vraagt van een userid . welke userid het moet zijn
* Je kunt hier ook **variabels** vermelden welke je gaat gebruiken.

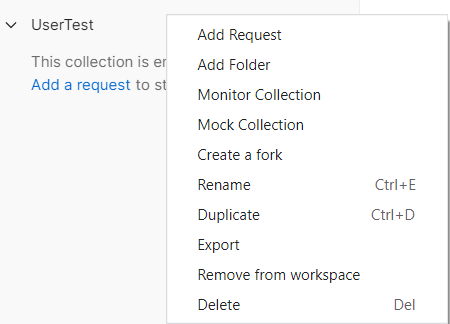
### GET request

Er zijn namelijk een paar request die je kunt aanvragen aan de hand van Postman.

De eerste die wij zullen bespreken is de GET request. Dit is een request die veel gebruikt wordt na het aanmaken van objecten.

Hoe gebeurt dit in postman?

* Eerst uw collectie selectioneren (rechter muisklik)
* Daarna de optie ADD REQUEST klikken



U kunt u API link hier invoeren met uw GET request.

Bijvoorbeeld hier beneden heb ik gevraagd om alle users te tonen op pagina nummer 2.

Hier ziet u dat er een lijst wordt teruggegeven met alle users op pagina 2.

Omdat we hier hebben gespecifieerd in de query parameters dat de parameter PAGE de value 2 moet hebben.

Une image contenant texte

Description générée automatiquement

### HTTP Status Code

Bij het uitvoeren van request krijgen we verschillende antwoorden. Bijvoorbeeld bij het uitvoeren bij de code hierboven kregen we een pagina terug met een paar users. Maar dit is niet de enige response die we terugkrijgen. We krijgen ook de response “Successful” door. Dit betekent dat de request succesvol werd uitgevoerd end at er data eruit gehaald is.

Er zijn nog verschillende responses.

* Successful : je krijgt een OK response terug
* Informational messages : dit is informatieve berichten die u kunnen toelichten
* Redirection : het stuurt u iets anders door of gaat u op een ander pagina doorsturen.
* Client Error : u heeft een fout gemaakt in het schrijven van de request
* Server error : er is iets fout gebeurd bij het uitvoeren van de request (serverside)

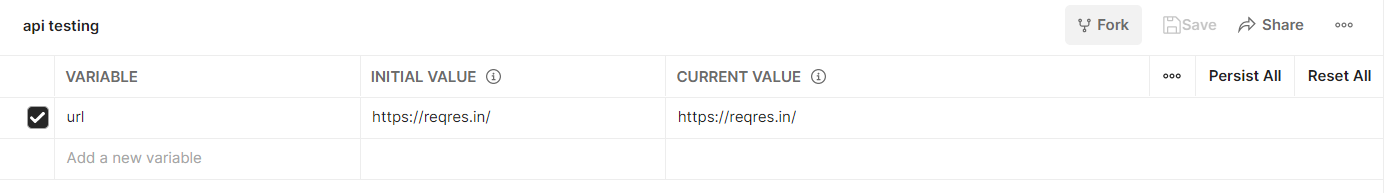
Une image contenant texte

Description générée automatiquementzoals dat u hier beneden ziet dat er op de websites een GET request wordt uitgevoerd met de status OK

### Environment

In een environment kunt u verschillende variabelen creeeren die nodig zijn voor u testing. Dit betekent dat u uw request kan verkorten en niet altijd moet copy pasten.

Voor wat dienen environments ? het zijn omgevingen waarin we requests kunnen testen. We kunnen dezelfde requests testen in verschillende environment. Bij het aanmaken van environments kunt u daar variabelen toevoegen.



Hier heb ik mijn url in een variable URL gezet zodat ik niet altijd een lange lijn moet schrijven voor dat ik mijn request schrijf.

Une image contenant texte

Description générée automatiquement

Hier ziet u dat het, hetzelfde antwoord geeft bij het uitvoeren van ons request.

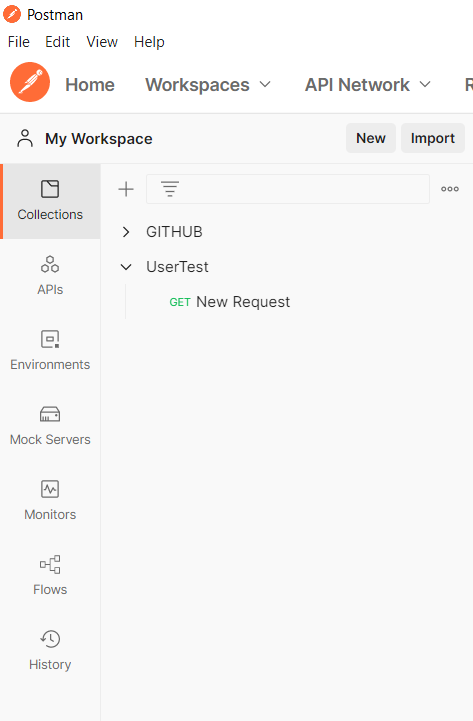
### 

# Postman crud github API

Hier gebruik ik alles wat ik geleerd heb in de theorie van POSTMAN in praktijk. Ik gebruik de demos en de theorie die hierboven vermeld is om een goede voorbeeld te realiseren. In dit voorbeeld ga ik de github API testen die het mogelijk maakt op repos met de CRUD functie te behandelen.

## Collection creëren

Ik heb hier dus eerst een collection aangemaakt waar mijn alle CRUD operaties zullen uitvoeren. Dit collection dient om alle request te groeperen en het vergemakkelijken van navigeren.



U kunt hierboven op de plusknop drukken om een nieuwe collectie aan te maken.

Une image contenant texte

Description générée automatiquement

Hier kunt u een naam en een paar opties wijzigen voor het beginnen van requests aan te maken.

## Environment aanmaken

Bij het maken van een environment is het belangerijk op lange en repititieve values in variabelen te plaatsen. In dit optie kunt u uw request in verschillende situaties plaatsen om te testen als u request nog zou werken of niet

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Dit zijn de variabelen die ik heb ingevoerd voor mijn testing.

* Username = mijn username die ik gebruik om in te loggen bij github
* Password= de password van mijn account
* url = is de api url die ik in een variabel zet om niet altijd te copy pasten
* repo = is het test repository waarmee ik werk

## authorization

Une image contenant texte

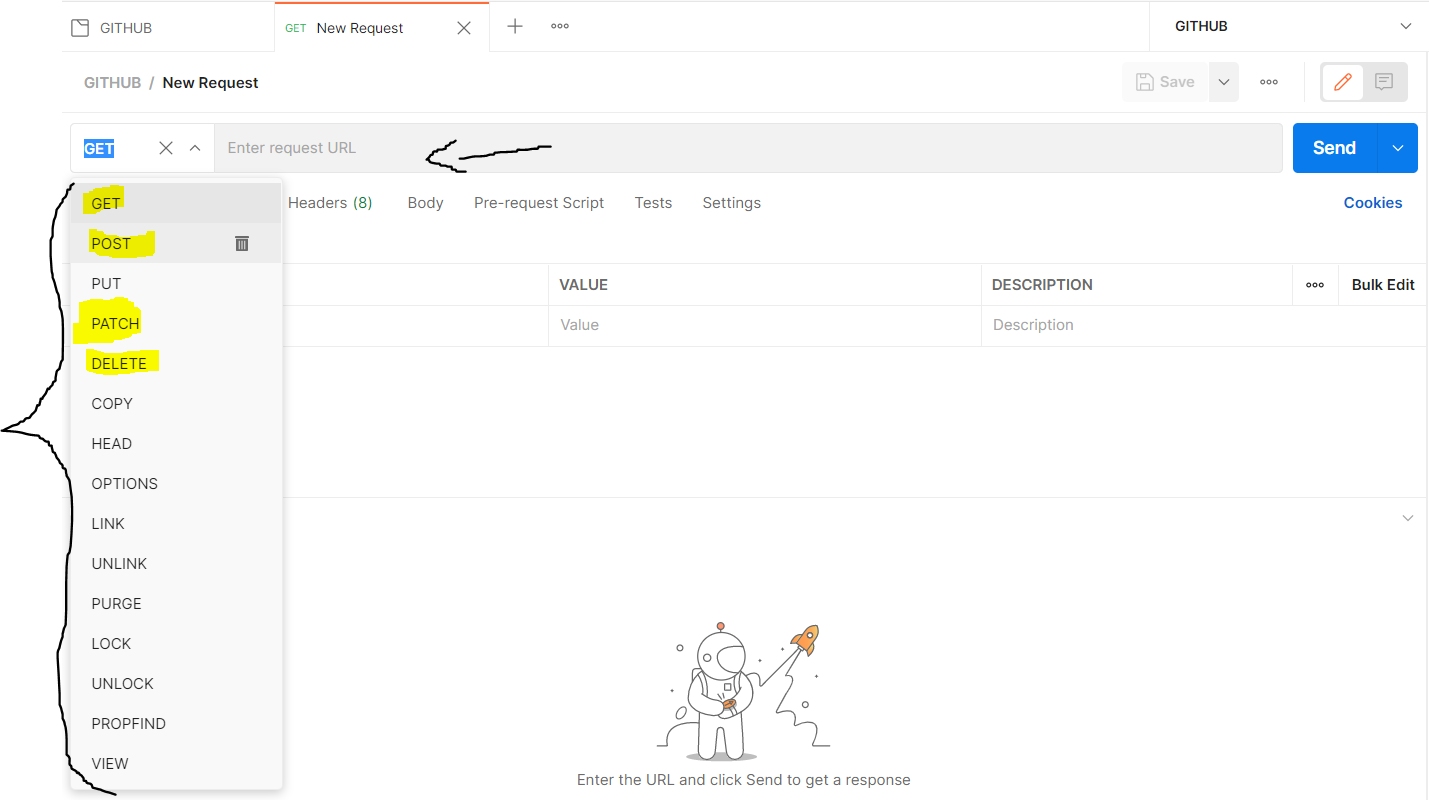
Description générée automatiquement als we nu proberen om gegevens te beheren in onze github repository zal het niet lukken. Omdat we eerst onze logingevens bij moeten geven voor een request.

Ik gebruik hier basic authorization. Ik heb hier mijn variabelen username en password gebruikt. Zodat ik dit ook kan testen in andere environments met andere accounts.

Het is hier ook mogelijk om met een token te werk te gaan. Maar dit zou dus onze werkfactor limiteren. Dus kunnen we ook niet met verschillende account of environments werken.

## Request aanmaken

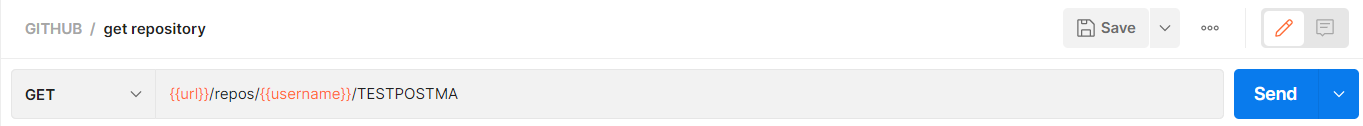
Bij het aanmaken van de environment en de collectie zelf moeten we nog een paar stappen uitvoeren voor dat we verder kunnen. Hier zullen een request toevoegen voor ons testing. Eens dat we dit hebben voltooid kunnen we ook beginnen aan onze volgende stappen van ons api testing



Als we nu op add request klikken van ons collective gaat u zich op deze pagina bevinden. U kunt bovenaan (pijl) u API url invullen en u kunt kiezen welke request dat u wilt uitvoeren of testen. Wij zullen hier GET,POST,PATCH,DELETE gebruiken de basic CRUD operaties

## Get testing met GITHUB (mijn) account

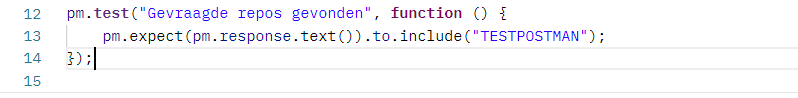
Hier gaan we get testing doen. Hier zien jullie dat ik de URL in een variable heb gezet. Ik gebruik mijn username om een repository te vinden die al gemaakt is in mijn GITHUB account. Ik heb mijn repository naam achteraan gezet.



We zullen een paar testen uitvoeren om te zien als de request zelf werkt of niet

Une image contenant texte

Description générée automatiquement



Hier heb ik verschillende testen geschreven :

* Status code is 200 : is om te zien als ons test goed doorloopt
* Status code is 404 : is om te zien als ons test faalt
* To have header “server”: is als de header de variable Server heeft om te zien als ons api goed verbonden is
* Gevraagde repos gevonden: is voor het vinden van de repos TESTPOSTMAN in de body

We zien hier dat twee testen gelukt zijn en 2 gefaald.

Dit betekent dat de api goed verbonden is maar dat hij geen repository heeft gevonden.

Une image contenant texte

Description générée automatiquement

Ik heb voor de fouten gezocht en ik heb gevonden dat ik de N ben vergeten te schrijven bij TESTPOSTMAN

Une image contenant texte

Description générée automatiquement

Na het verbeteren van de fout ziet u dat de 3 testen erdoor zijn.hier betekent dus dat de request succesvol is doorgegaan en dat hij ook een variable server heeft gevonden in de header en heeft ook de gevraagde repository gevonden in de body. Er is 1 test gefaald en dit is de test 404 error. Dit gaat niet door omdat de test gelukt is.

Une image contenant texte

Description générée automatiquement

## POST testing

Voor de volgende stap gaan we het POST testen in postman. We gaan dit uitvoeren en alles goed verloopt gaan we een repository maken op mijn eigen github. Bij het uitvoeren van deze POST testing gaat er een repository gemaakt worden die niet private is en een naam erbij heeft. De inhoud van de repository gaat standard zijn. Dus u gaat gewoon een read.me document erbij staan.

Une image contenant texte

Description générée automatiquement

Hier ziet u dat ik een repository aanmaak met de naam TESTPOSTMAN en als description “dit is een repos voor testing”.

Une image contenant texte

Description générée automatiquement

Hier ziet u dat ik nog geen repository heb met de naam TESTPOSTMAN. Dit zullen we nu aanmaken aan de hand van ons code.

Une image contenant texte

Description générée automatiquement

Hier maken we testen aan voor de POST request.

Voor wat dienen al die testen?

* Successful POST request : is als de code 201,202 is en dat de repository goed is aangemaakt.
* Repository is correct aangemaakt: dit is als de body de naam TESTPOSTMAN meegeeft e nook de description
* Validatie is niet gelukt: is als we de code 422 terug krijgen en dat de validatie niet is gelukt bv. De repos bestaat al, de title is teg root,...

Une image contenant texte

Description générée automatiquement

Hier ziet u dat ons repository goed is aangemaakt. Behalve 1 test die gefaald is. De gefaalde test is omdat de validatie gelukt is . en kregen we de code 201 terug en niet 422.

Une image contenant texte

Description générée automatiquement

Bij het reloaden van mijn github profiel ziet u dat de repository is toegevoegd. Dit is een succesvolle resultaat. De descriptie staat er ook natuurlijk bij.

Als ik nu de test uitvoer met dezelfde gegevens dan zullen de meeste testen falen.

Une image contenant texte

Description générée automatiquement

Hier ziet u bijvoorbeeld wanneer ik de test terug uitvoer met dezelfde gegevens. 2 testen falen en 1 is gelukt.

* De eerste test is gefaald omdat: ik geen code 201 of 202 heb teruggekregen. Dit betekent dat mijn POST request gefaald is.
* De tweede test is gefaald omdat : de body niet de zinnen bevat die ik had gevraagd.
* De derde test is gelukt: omdat ik de code 422 terug heb gekregen end it betekent dat de validatie gefaald is.

## Delete testing

bij het deleten gebruik ik de url de repos de username en de repository zelf. Ik moet hier de repository meegeven om dit te laten verwijderen.



Hier zal ik dezelfde repository verwijderen die ik net had aangemaakt. Dus de repository TESTPOSTMAN.

Une image contenant texte

Description générée automatiquement

Hier ziet u dat de repository nog op mijn account staat. We gaan hiervoor een paar testen schrijven voordat we dit uitvoeren. Zodat wij kunnen weten als de DELETE request goed gelukt is of niet.

Une image contenant texte

Description générée automatiquement

Hier heb ik 2 testen geschreven.

* De eerste test : als de status code 204 is (success) dan is die test groen
* De tweede test : als de status code 404 is (gefaald) dan is die test groen

Une image contenant texte

Description générée automatiquement

Bij het uitvoeren van de code ziet u dat de eerste gelukt niet en de tweede niet. Dus dit betekent dat de repository verwijdert is. Dit zullen we eens checken.

Une image contenant texte

Description générée automatiquement

Hier ziet u dat mijn repository succesvol verwijdert is.

## PATCH testing

Wij zullen hier de request patch gebruiken. Als we nu het verschil zouden vragen tussen patch en put.

Patch : is het sturen van een stuk data die de object niet gaat verwijderen en gewoon aanpassen

Put : maakt een hele nieuwe object aan met de aangepaste gegevens.

We zullen hier de request patch gebruiken omdat we hier kleine gegevens moeten wijzigen zoals de titel , de description enzovoort.



Hier geeft ik de url door met de repos en de username en de repository die u wilt updaten. Voor dat we dit uitvoeren gaan we eerst de repository aanmaken door de post zelf.