

# FOUNDFAVE

FOUNDFAVE React App Live Demo - <https://foundfave.online>

## INSTALLATIEHANDLEIDING FOUNDFAVE API

andre.de.groot@novi-education.nl

In opdracht van NOVI Hogeschool

Eindopdracht  
Backend Development 2023-8

Mede mogelijk gemaakt door:



# VOORWOORD

---

## DOOR ANDRÉ DE GROOT

Toen ik begon bij NOVI Hogeschool had ik al ruim een jaar of zes ervaring als freelance webdesigner, internet marketeer en SEO/SEA-consultant.

Ik dacht al aardig wat te weten van web development en tijdens de lessen van NOVI Hogeschool werd ik aangenaam verrast en ben ik geattendeerd op dingen die ik nog niet wist.

Voor de backend leerlijn is ons gevraagd een onderwerp te verzinnen en rondom dit onderwerp een Spring Boot API te bouwen.

Om tot dit ontwerp te komen heb ik diverse interesses gebundeld en ben ik op het idee gekomen om een op Marvel gebaseerde app te ontwerpen waar je Marvel karakters kunt opzoeken, aan je favorieten kunt toevoegen, en online kunt bewaren (Daarom de naam FOUNDFAVE: FOUND = Gevonden. FAVE = Favoriet)

De frontend van deze API heb ik inmiddels al afgerond tijdens de leerlijn frontend.

Deze frontend is momenteel gekoppeld aan de Marvel API en zal deze uiteindelijk koppelen aan de FOUNDFAVE API.

FOUNDFAVE API is een project gemaakt in opdracht van NOVI Hogeschool en voor de opbouw en indeling van dit document heb ik zoveel mogelijk de richtlijnen en aanwijzingen aangehouden die in de lesstof staan vermeld.

Met grote dank aan NOVI Hogeschool, mijn docenten, de SME'ers (Subject Matter Experts) en mijn medestudenten voor deze leuke en leerzame periode.

André de Groot

Amsterdam, 25 februari 2024

Titel: Installatiehandleiding FOUNDFAVE API  
Auteur: André de Groot  
Versie: 1.0  
Datum: 25 februari 2024



# INHOUDSOPGAVE

## INDEX

Inleiding .....	6
Software Installatie .....	7
IntelliJ Community Edition .....	8
JDK 17 LTS .....	9
Postman .....	10
PostgreSQL & pgAdmin .....	11
Software Configuratie .....	12
IntelliJ IDEA Configureren.....	13
Postman Configureren.....	17
pgAdmin Configureren.....	18
FOUNDFAVE API Opstarten.....	19
JWT-Token In Postman.....	20
Gebruikers, Rollen & Inlog .....	22
FOUNDFAVE API Endpoints.....	23
Troubleshooting .....	65



# COLOFON

## GEGEVENS

Opleidingsinstituut	Live Demo, Github & Figma
NOVI Hogeschool	<a href="#">Eindopdracht repository Github</a>
	<a href="https://foundfave.online">https://foundfave.online</a> (Live demo)
Opdracht	<a href="#">API online documentation</a>
Integrale eindopdracht Backend v3.4	
	Persoonlijk
	André de Groot
	FSD 2023-02 - 2023-08
	<a href="mailto:andre.de.groot@novi-education.nl">andre.de.groot@novi-education.nl</a>
Vormgeving en tekst	
André de Groot	
Illustraties	
▶ <a href="https://undraw.co">https://undraw.co</a>	
Fotografie, fonts, iconen en vectoren	
▶ Font Awesome	
▶ Google Fonts	
Tools & Software	
▶ Affinity Publisher	
▶ Affinity Photo	
▶ Affinity Designer	
▶ FontBase	



# INLEIDING

---

## DOOR ANDRÉ DE GROOT

Voor de eindopdracht, van de leerlijn frontend, heb ik een React app ontwikkeld die gegevens ophaalt van de Marvel API en deze presenteert op het scherm.

Achteraf gezien ben ik niet tevreden over de geleverde data van de Marvel API en heb daarom besloten mijn eigen API te bouwen voor de leerlijn backend, genaamd FOUNDFAVE API. Deze API zal ik uiteindelijk koppelen aan de door mij gebouwde React frontend.

Met de FOUNDFAVE API kun je Marvel karakters opzoeken met de zoekmachine, toevoegen aan je favorieten en bewaren voor later.

Het React frontend gedeelte van deze API is hier te vinden: <https://foundfave.online>.

De FOUNDFAVE API is gebouwd met de object georiënteerde programmeertaal Java en het framework Spring Boot.

In het eerste stuk van deze installatiehandleiding gaan we de **benodigde software installeren** om de API aan de praat te krijgen.

Als we alle **software configureren**, kunnen we daarna de **API en servers opstarten**, zodat we met Postman aan de slag kunnen om de endpoints van de API te testen en te bekijken welke informatie deze endpoints teruggeven.

Om de endpoints te testen is het het beste om eerst in te loggen met een bestaande gebruiker. De **gebruikers met wachtwoorden** kun je hier vinden.

Met deze bestaande gebruikers kunnen we inloggen en een **JWT-token opvragen** van de server en deze in een variabele plaatsen binnen Postman, zodat je toegang kunt krijgen tot alle endpoints

Een uitgebreid **overzicht van alle API endpoints, functionaliteit, JSON en bijhorende toelichting** is ook te vinden binnen dit document.

Mocht je tegen technische obstakels aanlopen, tijdens de installatie of configuratie, dan kun je aan het einde van dit document een lijst vinden met vragen en antwoorden in het **hoofdstuk Troubleshooting**.

# SOFTWARE INSTALLATIE

---

## LIJST VAN BENODIGDHEDEN

Om de FOUNDFAVE API te kunnen draaien zijn er enkele programma's nodig die we gaan installeren in de volgende hoofdstukken.

**De volgende software heb je nodig om de FOUNDFAVE API te kunnen draaien:**

- ▶ IntelliJ IDEA Community Edition.
- ▶ JDK 17 LTS (Java Development Kit, Long Term Support).
- ▶ Postman.
- ▶ PostgreSQL met pgAdmin.

# INTELLIJ COMMUNITY EDITION

## INSTALLATIE

Een Integrated Development Environment of IDE is computersoftware die een softwareontwikkelaar ondersteunt bij het ontwikkelen van computersoftware. Het bestaat uit een aantal onderdelen die samen een software suite vormen. IntelliJ IDEA, van het merk JetBrains, is een zeer populaire (Zie [Stack Overflow survey 2023](#)) Integrated Development Environment.

Er zijn 2 versies van IntelliJ IDEA en voor dit project gaan we de (Gratis en open source) IntelliJ IDEA Community Edition downloaden en installeren.

IntelliJ IDEA Community Edition kun je hier [downloaden van de officiële website](#).

### Stappenplan

1. Op de JetBrains website: Scroll een stukje naar onderen en je zult IntelliJ IDEA Community Edition zien staan. Klik op “Download”.
2. Na het downloaden, dubbel klikken op het exebestand om de IntelliJ IDEA installatiewizard te starten.
3. “Welcome to IntelliJ IDEA Community Edition setup” scherm. Klik op “Next”.
4. “Choose Install Location” scherm. Klik op “Next”.
5. “Install Options” scherm. Vink alle opties aan. Klik op “Next”.
6. “Choose Start Menu Folder” scherm. Klik op “Install”.
7. “Completing IntelliJ IDEA Community Edition Setup” scherm. Selecteer de optie “Reboot now” en klik op “Finish”.
8. Na reboorten en installeren, IntelliJ IDEA opstarten en krijg je de JetBrains User agreement te zien: Aanvinken: “I confirm that I have read and accept the terms of this User Agreement” en klik op “Continue”.

IntelliJ IDEA Community Edition is nu geïnstalleerd en klaar voor gebruik.



# JDK 17 LTS

## INSTALLATIE

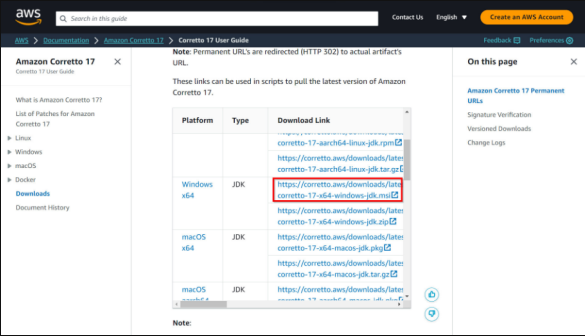
De Java Development Kit (JDK) is de software die een programmeur nodig heeft om programma's in Java te kunnen ontwikkelen. De Java Development Kit bestaat uit de Java Runtime Environment (JRE, alles wat nodig is om Java-programma's op een computer te kunnen draaien) met daarbij een reeks hulpmiddelen die voor de programmeur van belang zijn.

Om IntelliJ IDEA en de applicatie optimaal te kunnen laten draaien zal er een JDK (Java Development Kit) worden geïnstalleerd. DE FOUNDFAVE API applicatie runt op JDK 17 LTS (LTS – Long Term Support).

De Amazon Corretto JDK kun je [hier downloaden](#).

### Stappenplan

1. Scroll naar beneden naar de tabel en zoek in de tabel naar de rij met “Windows x64”. (Zie afbeelding)
2. Klik op het msi bestand met de naam “amazon-corretto-17-x64-windows-jdk.msi” om te downloaden.
3. Na het downloaden, dubbel klik op het .msi bestand (amazon-corretto-17.0.9.8.1-windows-x64.msi) om de JDK installatiewizard te starten.
4. “Welcome to the Amazon Corretto (x64) Setup Wizard” scherm. Klik op “Next”.
5. “Custom Setup” scherm. Klik op “Next”.
6. “Completed the Amazon Corretto (x64) Setup Wizard” scherm. Klik op “Finish” om de installatie af te ronden.



# POSTMAN INSTALLATIE

Postman is een API-platform voor het bouwen, gebruiken en testen van API's. We gaan Postman gebruiken, voor dit project, om verzoekjes te sturen naar de endpoints van de FOUNDFAVE API.

Postman downloaden en bekijk daarna het stappenplan.

## Stappenplan

1. Klik op de grote oranje knop “Windows 64-bit” om te downloaden.
2. Dubbelklik op het bestand “Postman-win64-Setup.exe” om het programma te starten. Er is verder geen sprake van een installatiewizard en het programma zal direct opstarten.

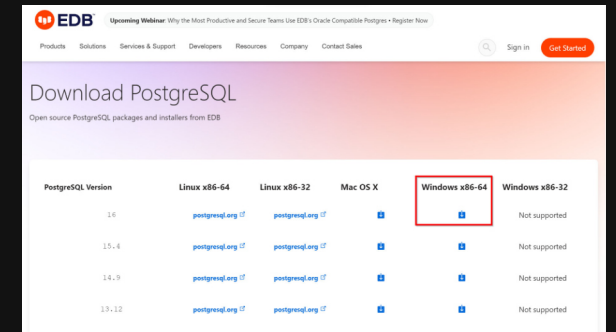
# POSTGRESQL & PGADMIN

---

## INSTALLATIE

PostgreSQL, ook bekend als Postgres, is een gratis en open-source relationeel databasebeheersysteem. pgAdmin is het beheer- en ontwikkelingsplatform voor PostgreSQL.

PostgreSQL en pgAdmin kun je [hier downloaden](#). Klik op het blauwe download-icoon onder de kolom “Windows x86-64” met de hoogste versie “PostgreSQL Version”. (Zie afbeelding)



## Stappenplan

1. Dubbelklik op het "postgresql-16.1-1-windows-x64.exe" om de installatiewizard te starten en zowel PostgreSQL als pgAdmin te installeren.
2. "Setup – PostgreSQL" scherm. Klik op "Next".
3. "Installation Directory" scherm. Klik op "Next".
4. "Select Components" scherm. Alles opties aanvinken, behalve de "Stack Builder" optie. Klik op "Next".
5. "Data Directory" scherm. Klik op "Next".
6. "Password" scherm. Vul tweemaal een password in en klik op "Next".
7. "Port" scherm. Port 5432 en klik op "Next".
8. "Advanced Options" scherm. "Default locale" en klik op "Next".
9. "Pre Installation Summary" scherm. Klik op "Next".
10. "Ready to Install" scherm. Klik op "Next".
11. Na installeren klik op "Finish".

# SOFTWARE CONFIGURATIE

## INLEIDING

In de volgende hoofdstukken gaan we de diverse software pakketten, die we net hebben gedownload en geïnstalleerd, configureren.



# INTELIJ IDEA CONFIGUREREN

## CONFIGURATIE

We gaan nu de net geïnstalleerde JDK 17 LTS koppelen aan het FOUNDFAVE API project. Ook in het projectbestand, de pom.xml, wordt er verwezen naar JDK 17.

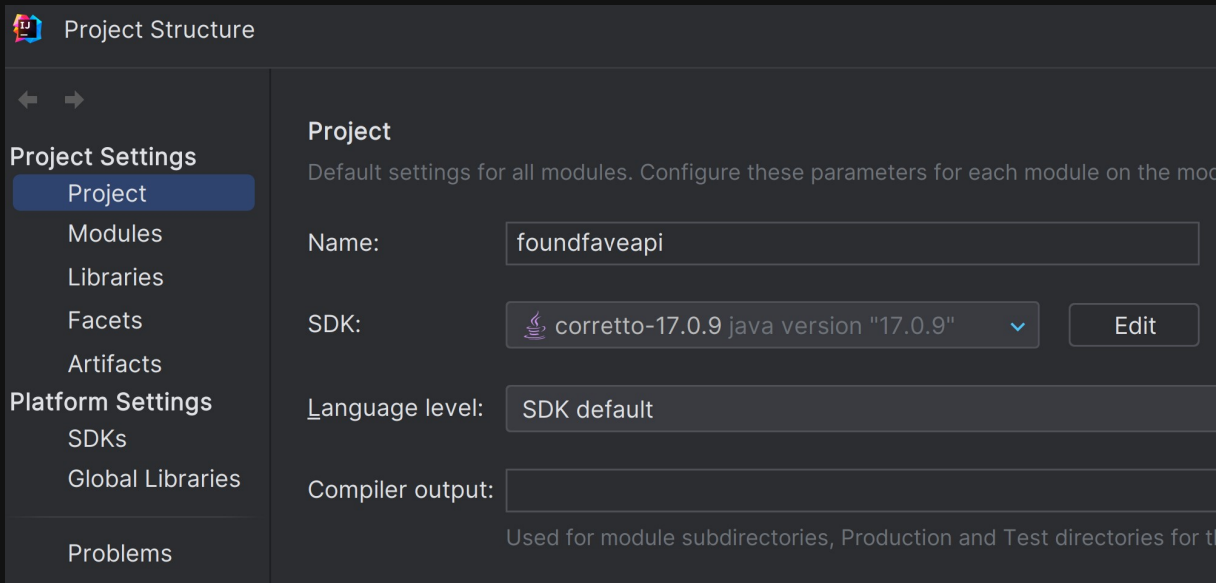
```
<properties>

    <java.version>17</java.version>

</properties>
```

### Stappenplan

1. Open het FOUNDFAVE API project met IntelliJ IDEA.
2. In het settings (...) menu: “File > Project structure”.
3. Klik op het dropdownmenu onder het Project SDK en selecteer “corretto-17.0.9”. Klik op “OK” om de wijzigingen op te slaan. (Zie afbeelding)



# INTELLIJ IDEA CONFIGUREREN

## CONFIGURATIE (VERVOLG)

### Configureren van het application.properties bestand

Open het project in IntelliJ en navigeer naar het 'application.properties' bestand. Dit bestand staat in het mapje: \src\main\resources.

### Stel de volgende waarden in het application.properties bestand:

```
spring.datasource.url=jdbc:postgresql://localhost:5432/foundfave-api/  
  
spring.datasource.username=postgres  
  
spring.datasource.password=password  
  
spring.datasource.driver-class-name=org.postgresql.Driver
```



# INTELLIJ IDEA CONFIGUREREN

## CONFIGURATIE (VERVOLG)

Nu het application.properties bestand ingesteld is kunnen we de data sources en drivers binnen IntelliJ gaan instellen.

### Stappenplan

1. Klik op het gele database icoontje aan de rechterkant binnen IntelliJ. (Zie afbeelding rechts)
2. Klik op de blauwe + icoon (New).
3. Klik op "Data source" > PostgreSQL. Je krijgt dan onderstaand scherm te zien. (Zie afbeelding onder)

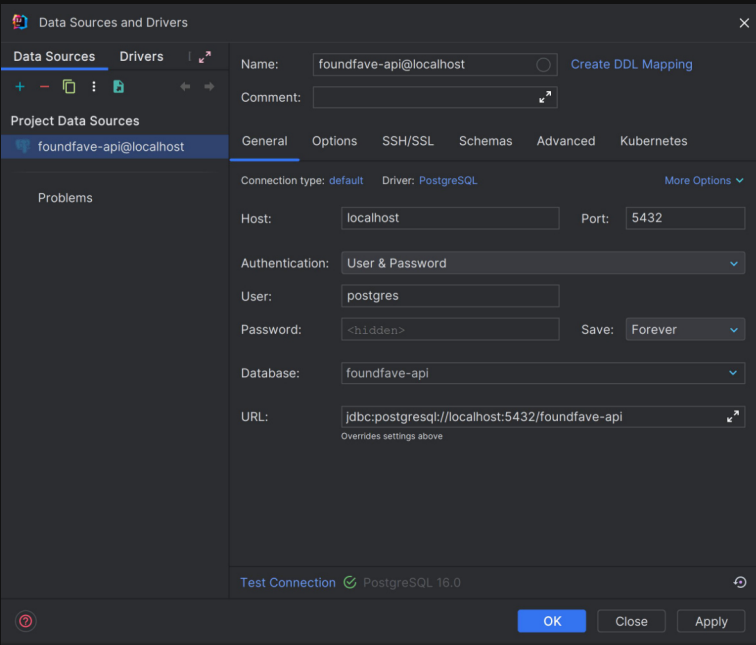
### 4. Vul de volgende waarden in:

**User:** postgres

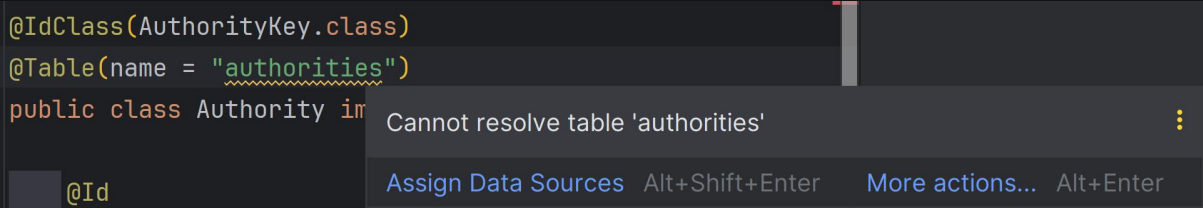
**Password:** password (Het password die je had ingesteld voor pgAdmin)

**Database:** foundfave-api

5. Klik op "Test Connection".
6. Als de test is geslaagd, klik op "OK".



7. Ga naar het bestand Authority.class in package "models".
8. Klik op de gele waarschuwing Table(name="authorities") en selecteer "Assign Data Sources". (Zie afbeelding)

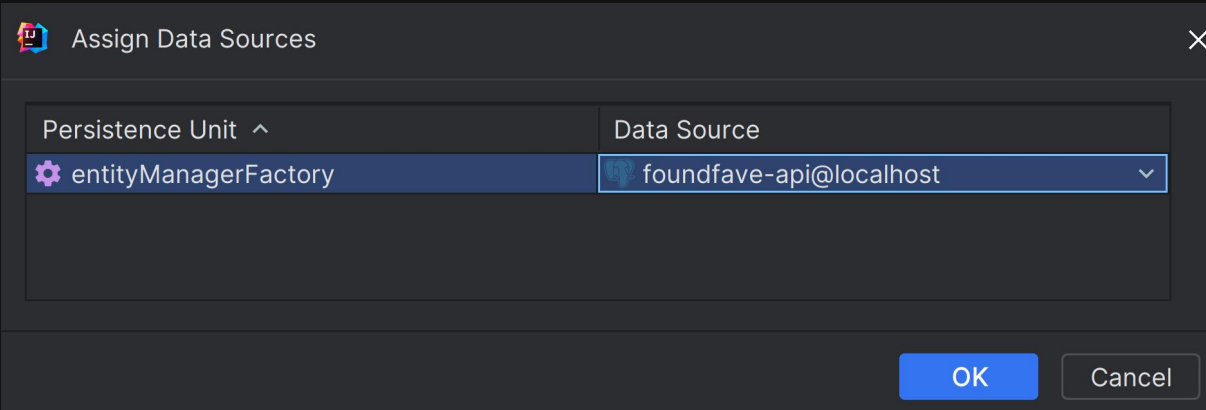




# INTELLIJ IDEA CONFIGUREREN

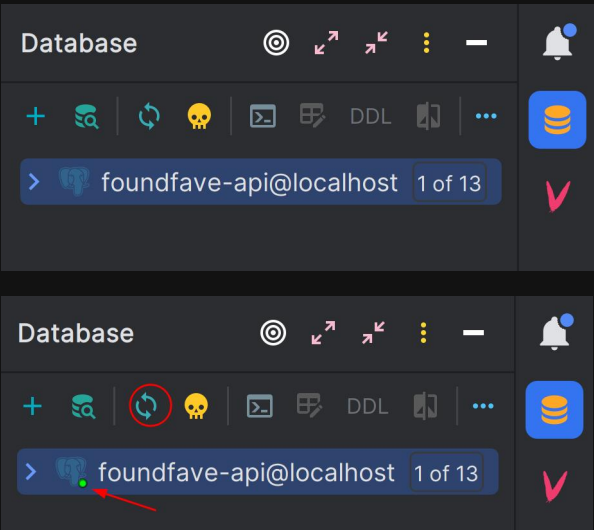
## CONFIGURATIE (VERVOLG)

9. Klik op “Data Source” en selecteer “foundfave-api@localhost” en klik op “OK”. (Zie afbeelding)



10. Klik nu op “Run FoundfaveapiAppliaction” of SHIFT + F10.

11. Klik nu op “Refresh” of CTRL + F5. Als het goed is gaat er een groen ‘lampje’ branden bij het Postgres olifantje en verdwijnt het gele streepje onder Authorities. (Zie afbeeldingen)



# POSTMAN CONFIGUREREN

## CONFIGURATIE

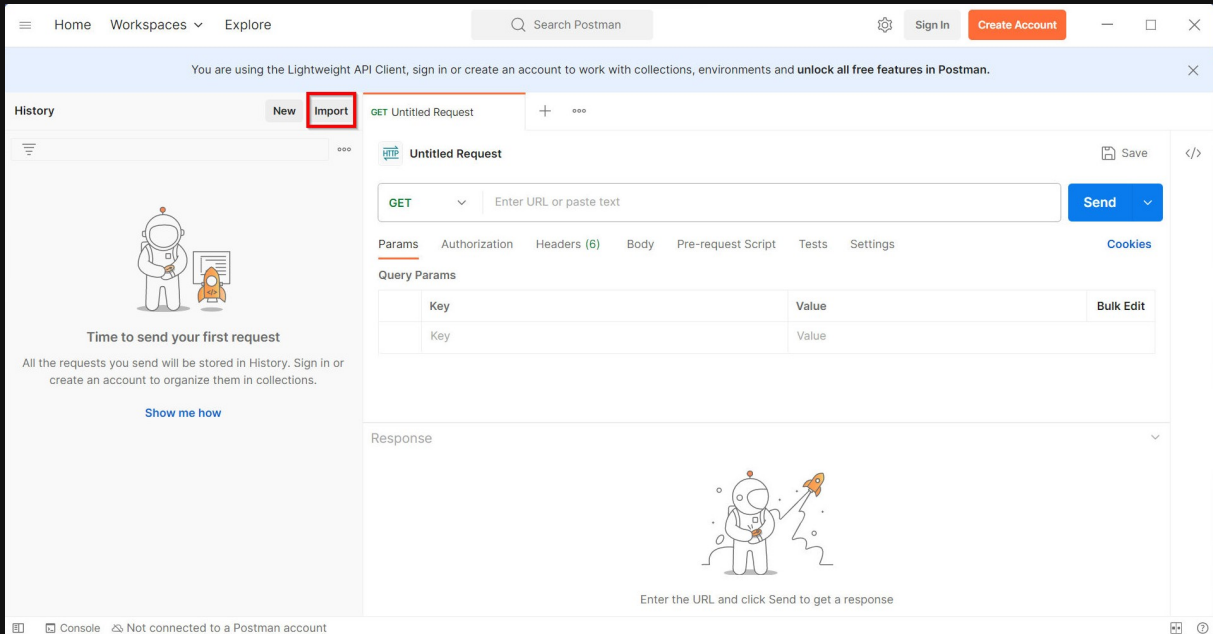
Door het meegeleverde JSON-bestand, die je kunt importeren in Postman, hoef je niet handmatig alle endpoints van de FOUNDFAVE API in te voeren. Om dit JSON-bestand te kunnen importeren kun je eerst een **Postman account aanmaken** en inloggen binnen de Postman applicatie.

Je kunt het JSON-bestand eventueel downloaden, mocht je het nog niet zijn tegengekomen binnen de bestanden van het project:

[Download JSON-bestand van Github repository.](#)

### Stappenplan

1. Rechtsboven, binnen de Postman applicatie, kun je inloggen. Na inloggen is het mogelijk het JSON-bestand te importeren.
2. Klik op de “Import” optie in het linkerpaneel om het meegeleverde JSON-bestand te importeren. (Zie afbeelding)
3. Een popup scherm komt naar voren waar je de optie krijgt om het JSON-bestand te importeren. Na het importeren verschijnt de collectie in het linkerpaneel en kun je er mee aan de slag.



# PGADMIN CONFIGUREREN

## CONFIGURATIE

In dit hoofdstuk gaan we in pgAdmin een nieuwe database aanmaken die we straks gaan koppelen aan de Spring Boot FOUNDFAVE API binnen IntelliJ.

### Stappenplan

1. Open pgAdmin. (Opstarten van pgAdmin kan even duren!)
2. Klik in het Object Explorer paneel op “Servers” zodat deze uitklapt.
3. Klik op “PostgreSQL 16” en voer het wachtwoord in dat je eerder had opgegeven tijdens de installatie. Mocht de server niet willen verbinden, na het invoeren van het wachtwoord, kijk dan even naar het hoofdstuk [Troubleshooting](#).
4. Maak een nieuwe database aan door met de rechtermuisknop te klikken op “Databases” (Onder PostgreSQL) en te kiezen: “Create > Database...”
5. Voer een naam in voor deze database: foundfave-api en klik op “Save”.

# FOUNDFAVE API OPSTARTEN

## INLEIDING

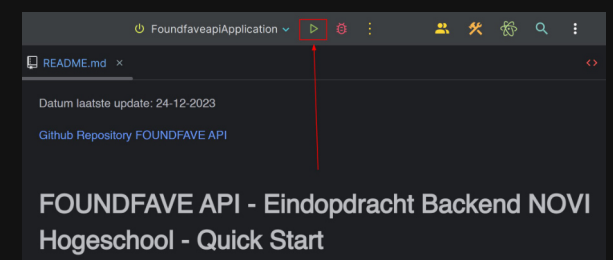
Na het installeren en configureren van IntelliJ, JDK 17, Postman, PostgreSQL en pgAdmin kunnen we nu alle software en servers gaan opstarten.

### pgAdmin opstarten

1. pgAdmin opstarten duurt meestal een paar minuten.
2. Wanneer de applicatie opstart, verschijnen er, als het goed is, in de database X tabellen.
3. Klik binnen pgAdmin op: PostgreSQL 16 > foundfave-api > Schemas > public > Tables.
4. De tabellen a, b en c zouden nu beschikbaar moeten zijn.
5. Mochten de tabellen niet zichtbaar zijn, klik dan met de rechtermuisknop op “Tables” en kies “Refresh”. Bekijk nogmaals of de tabellen nu wel zichtbaar zijn.
6. Mocht je daarna alsnog geen tabellen zien, bekijk dan even de [Troubleshooting](#) sectie.

### IntelliJ IDEA opstarten

1. Open het FOUNDFAVE API project in IntelliJ.
2. Klik op “Run FoundfaveapiApplication” (Zie afbeelding)



### Postman opstarten

We hebben in de vorige hoofdstukken, Postman geïnstalleerd, geconfigureerd en het JSON bestand geïmporteerd, en kunnen nu de eerste verzoeken gaan sturen naar de endpoints van de FOUNDFAVE API.

# JWT-TOKEN IN POSTMAN

## JWT-TOKEN VERKRIJGEN

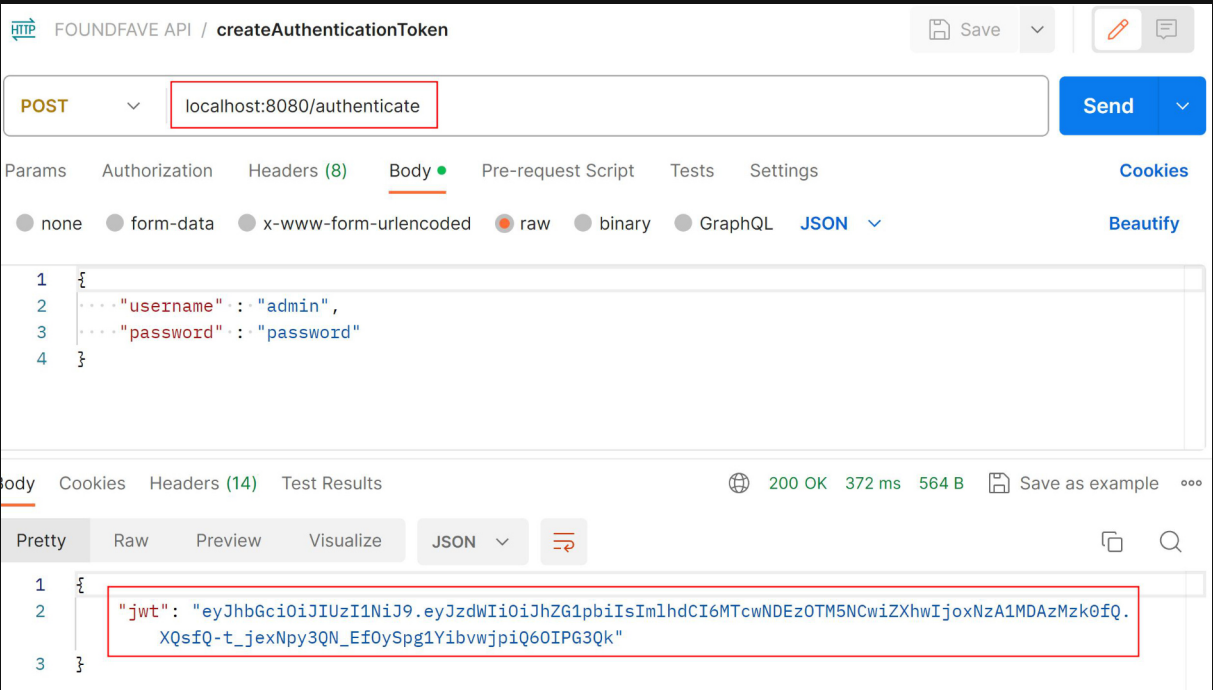
JSON Web Tokens (JWT) zijn een populaire methode voor het veilig overbrengen van informatie tussen partijen in de vorm van JSON-objecten.

### Stappenplan

1. Het JWT-token kun je opvragen door op het volgende request in Postman te klikken: “signIn”. Je ziet in dit POST request de uri: localhost:8080/authenticate, met de volgende Body:

```
{
  "username": "admin",
  "password": "password"
}
```

2. Klik op “Send” en je krijgt een Jwt-token terug in de vorm van JSON. (Zie afbeelding) - **Opmerking:** Momenteel zijn JWT-tokens 10 dagen geldig.



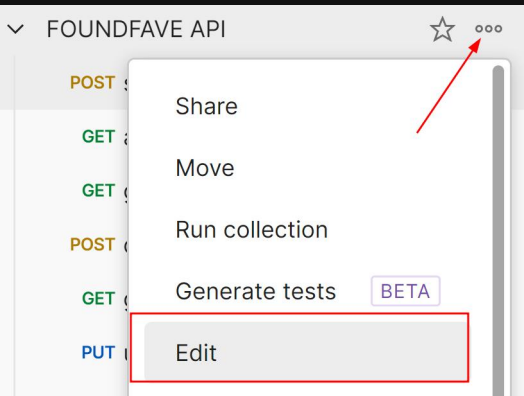
3. Selecteer het Jwt-token en druk op CTRL + C (kopieren).



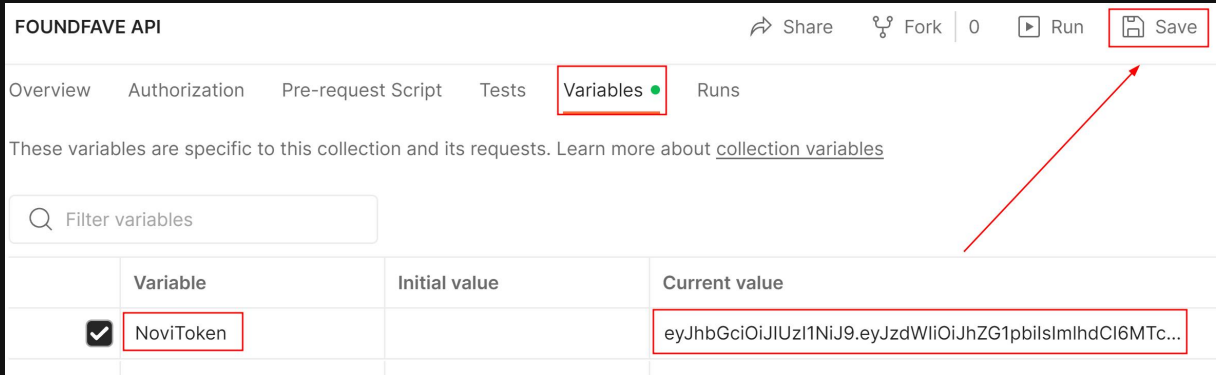
# JWT-TOKEN IN POSTMAN

## JWT-TOKEN VERKRIJGEN (VERVOLG)

4. Klik rechts van “FOUNDFAVE API”, in de collectie, op het menu (...). Klik op “Edit”. (Zie afbeelding rechts)
5. Klik op tabblad “Variables”. (Zie afbeelding onder)
6. Het “Collection details” venster zit in de weg. Klik dat venster weg.

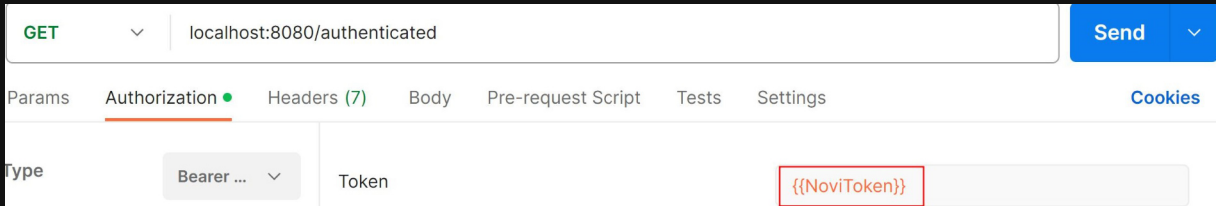


7. Vul in het veld “Variabele” de waarde: “NoviToken” in.
8. CTRL + V (Paste) de Jwt-token die nog op je klembord zit in het veld “Current value.”.
9. Klik op “Save” als je klaar bent.



10. Als het goed is werken de API verzoeken nu. In elk verzoek zit een {{NoviToken}} variabele ingebakken onder het tabblad “Authorization” > Type: Bearer Token > {{NoviToken}} (Zie afbeelding onder)

Mocht je tegen technische problemen aanlopen, raadpleeg dan het hoofdstuk **Troubleshooting**.



# GEBRUIKERS, ROLLEN & INLOG

## INLOGGEVENS TEST ACCOUNTS

Binnen de FOUNDFAVE API zijn er 2 rollen. De User en Admin. Hieronder een korte toelichting.

GEBRUIKERSROLLEN	
Rol	Toelichting
ROLE_USER	Deze gebruiker kan maar bij een beperkt aantal endpoints.
ROLE_ADMIN	Deze gebruiker kan alle endpoints, met de daarbijhorende requests, bereiken.

### Inloggegevens

Met onderstaande inloggegevens kun je inloggen binnen Postman.

INLOGGEGEVENS		
Gebruikersnaam	Wachtwoord	Rol
admin	password	ADMIN
user	password	USER



# FOUNDFAVE API ENDPOINTS

## OVERZICHT ENDPOINTS

In dit hoofdstuk een lijst van alle REST-endpoints, inclusief voorbeelden van de JSON-requests. De JSON-requests in dit document kun je gemakkelijk selecteren, kopiëren en plakken.

### Tabelinformatie van de endpoints

Per controller die in de applicatie aanwezig is, wordt er een tabel weergegeven met de volgende informatie:

- **“Endpoint”**: Dit zijn endpoints, zoals bijvoorbeeld /users of /info.
- **“Methode”**: Dit is het type HTTP request, zoals GET, POST, PUT, etc.
- **“Toegang”**: Door wie een bepaalde endpoint mag worden benaderd: USER of ADMIN.
- **“JSON”**: De JSON-body van een specifiek request. Vul hier de gewenste key-value pairs in.
- **“Toelichting”**: Wat de functie is van een endpoint en wat het doet. Welke method is geassocieerd met deze endpoint en andere nuttige informatie.

### Postman

De diverse endpoints zijn ook op te vragen via Postman op het volgende publieke endpoint: localhost:8080/info

Een overzicht van alle endpoints kun je ook hier vinden (Na het opstarten van de server):

<http://localhost:8080/documentation/index.html> (Engelstalig)

Download de laatste versie van het **Postman collectie JSON bestand**.



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS

PUBLIEKE ENDPOINTS				
Endpoint	Methode	Toegang	JSON	Toelichting
/test	GET	Publiek	N/A	Postman: <b>checkIfApiOnline</b> (Public)  Dit endpoint laat zien of de FOUNDFAVE API online is.
/info	GET	Publiek	N/A	Postman: <b>showApiInfo</b> (Public)  Dit endpoint geeft je een overzicht met alle beschikbare endpoints
/queries	GET	Publiek	N/A	Postman: <b>showAllAvailableQueries</b> (Public)  Dit endpoint geeft je een overzicht met alle beschikbare zoekopdrachten: Characters en Movies
/documentation/index.html	GET	Public	N/A	Postman: <b>API documentation</b> (Public)  Dit endpoint geeft je een gedetailleerd overzicht met alle beschikbare endpoints.



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

AUTHENTICATION ENDPOINTS				
Endpoint	Methode	Toegang	JSON	Toelichting
/login	POST	Publiek	{ "username": "admin", "password": "password" }	Postman: <b>Login</b> (Public)  Met dit endpoint kun je een JWT-token verkrijgen en inloggen op de service..
/authenticated	GET	Authenticated	N/A	Postman: <b>authenticated</b>  Op dit endpoint zie je welke bevoegdheden je hebt, als je geauthenticeerd bent, je gebruikersnaam en andere informatie.



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

CHARACTERS ENDPOINTS				
Endpoint	Methode	Toegang	JSON	Toelichting
/characters	GET	Admin, User	N/A	Postman: <b>getAllCharacters</b>  Met dit endpoint kun je alle Marvel-personages bekijken die momenteel in de database staan.
/characters/{characterId}	GET	Admin, User	N/A	Postman: <b>getCharacterById</b>  Met dit endpoint kun je een specifiek Marvel-personage bekijken door de ID ervan op te geven op de locatie van de padvariabele {characterId}.  <b>Voorbeeld:</b> /characters/1000
/characters/search/starting-with	GET	Admin, User	N/A	Postman: <b>findCharactersByName StartingWith</b>  Met dit endpoint kun je Marvel-personages zoeken met een query parameter.  <b>Voorbeeld:</b> /characters/search/starting-with?name=vision



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

CHARACTERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/characters/search/contains	GET	Admin, User	N/A	Postman: <b>findCharactersByName Contains</b>  Met dit endpoint kun je zoeken naar Marvel-personages met een query parameter. Deze methode zoekt naar personages waarvan de naam de inhoud bevat die als query parameter is ingevuld.  <b>Voorbeeld:</b> /characters/search/contains?name=vision
/characters/search/sorted-asc	GET	Admin, User	N/A	Postman: <b>findCharactersByName SortedAsc</b>  Met dit endpoint kun je zoeken naar Marvel-personages met een query parameter. Deze methode retourneert de zoekresultaten in oplopende volgorde.  <b>Voorbeeld:</b> /characters/search/sorted-asc?name=vision





# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

CHARACTERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/characters/ search/sorted- desc	GET	Admin, User	N/A	Postman: <b>findCharactersByName SortedAsc</b>  Met dit endpoint kun je zoeken naar Marvel-personages met een query parameter. Deze methode retourneert de zoekresultaten in aflopende volgorde.  <b>Voorbeeld:</b> /characters/search/sorted-desc?name=vision
				Postman: <b>findCharactersByActor NameContains</b>  Met dit endpoint kun je zoeken naar Marvel-personages, gespeeld door bepaalde acteurs, met een bepaalde query parameter.  <b>Voorbeeld:</b> /characters/search/actor-name?name=robert



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

CHARACTERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/characters/ character/ {characterId}	PUT	Admin	{ "characterAliasName": "Personage alias naam.", "characterRealName": "Personage echte naam.", "characterActorName": "Naam acteur.", "characterTitle": "Personage titel hier.", "characterGender": "Male", "characterSummary": "Samenvatting hier.", "characterDescription": "Beschrijving hier." }	Postman: <b>updateCharacterById</b>  Met dit endpoint kun je een Marvel-personage bijwerken. De character-ID moet worden ingevoerd op de locatie van de padvariabele {characterId}.  Voer de vereiste informatie in de JSON-body in.  Het geslacht heeft 3 mogelijke opties: Male, Female en X.  <b>Voorbeeld:</b> /characters/character/1000



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

CHARACTERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/characters/associate/movie/{characterId}	PUT	Admin	{ "id": 1000 }	<p>Postman: <b>associateMovieAndCharacter</b></p> <p>Met dit endpoint kun je een personage en een film combineren. Het personage wordt dan aan die film toegevoegd en die film wordt aan dat personage toegevoegd. De ID in de JSON-body is de ID van de film. De character-id moet worden ingevoerd op de locatie van de padvariabele {characterId}.</p> <p><b>Voorbeeld:</b> /characters/associate/movie/1007</p>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

CHARACTERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/characters	POST	Admin	{ "characterAliasName": "Naam karakter.", "characterRealName": "Echte naam.", "characterActorName": "Naam van acteur.", "characterTitle": "Jouw titel.", "characterGender": "Female", "characterSummary": "Samenvatting.", "characterDescription": "Omschrijving..", }	<p>Postman: <b>createCharacter</b></p> <p>Met dit endpoint kun je een Marvel-personage maken met de opgegeven waarden en attributen.</p> <p>Het geslacht heeft 3 mogelijke opties: Male, Female en X.</p>





# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

CHARACTERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/characters/{characterId}	DELETE	Admin	N/A	<p>Postman: <b>deleteCharacterById</b></p> <p>Met dit endpoint kun je een Marvel-personage met een bepaalde ID verwijderen. De character-ID moet worden ingevoerd op de locatie van de padvariabele {characterId}.</p> <p><b>Notitie:</b> Momenteel mogen karakters die zijn toegewezen aan gebruikers (Favorieten) of films niet worden verwijderd.</p> <p>Gebruik eerst de <b>removeFavoriteCharacterFromUser</b> en <b>disassociateMovieAndCharacter</b> methoden voordat je een Marvel-personage probeert te verwijderen.</p> <p><b>Voorbeeld:</b> /characters/1000</p>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

CHARACTERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/characters/disassociate/movie/{characterId}	DELETE	Admin	{ "id": 1005 }	<p>Postman: <b>disassociateMovieAndCharacter</b></p> <p>Met dit endpoint kun je een personage en een film loskoppelen. Het personage wordt dan uit die film verwijderd en die film wordt uit dat personage verwijderd. De ID in de JSON-body is de ID van de film. De character-ID moet worden ingevoerd op de locatie van de padvariabele {characterId}.</p> <p><b>Voorbeeld:</b> /characters/disassociate/movie/1000</p>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

CONTACT FORMS ENDPOINTS				
Endpoint	Methode	Toegang	JSON	Toelichting
/contactforms	GET	Admin	N/A	Postman: <b>getAllContactForms</b>  Met dit endpoint kun je een lijst bekijken met alle ingediende contactformulieren.
/contactforms/search/name	GET	Admin, User	N/A	Postman: <b>findContactFormsByNameContains</b>  Met dit endpoint kun je een contactformulier vinden door een deel van een naam in te voeren als query parameter.  <b>Voorbeeld:</b> /contactforms/search/name?name=an
/contactforms/search/email	GET	Admin, User	N/A	Postman: <b>findContactFormsByEmailContains</b>  Met dit endpoint kun je een contactformulier vinden door een deel van een e-mailadres in te voeren.  <b>Voorbeeld:</b> /contactforms/search/email?email=novi



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

CONTACT FORMS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/contactforms/search/comments	GET	Admin, User	N/A	Postman: <b>findContactFormsByCommentsContains</b>  Met dit endpoint kun je een contactformulier vinden door een deel van een trefwoord in te voeren als query parameter.  <b>Voorbeeld:</b> /contactforms/search/comments?comments=jul
/contactforms/{contactFormId}	GET	Admin, User	N/A	Postman: <b>getContactFormById</b>  Met dit endpoint kun je een contactformulier vinden door de ID ervan in te voeren. De contactformulier-ID moet worden ingevoerd op de locatie van de padvariabele {contactFormId}.  <b>Voorbeeld:</b> /contactforms/1



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

CONTACT FORMS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/contactforms/post	POST	Admin, User	{ "name": "Novi", "email": "test@novi.nl", "comments": "Dit is commentaar" }	<p>Postman: <b>createContactForm</b></p> <p>Met dit endpoint kun je een contactformulier maken en plaatsen door de JSON-body in te vullen.</p>
/contactforms	DELETE	Admin	N/A	<p>Postman: <b>deleteAllContactForms</b></p> <p>Met dit endpoint kun je alle ingediende contactformulieren verwijderen.</p>
/contactforms/{contactFormId}	DELETE	Admin, User	N/A	<p>Postman: <b>deleteContactFormById</b></p> <p>Met dit endpoint kun je een contactformulier met een bepaalde ID verwijderen. De contactformulier-ID moet worden ingevoerd op de locatie van de padvariabele {contactFormId}.</p> <p><b>Voorbeeld:</b> /contactforms/1</p>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

FILE ENDPOINTS				
Endpoint	Methode	Toegang	JSON	Toelichting
/download-profile-image/{fileName}	GET	Public	N/A	<p>Postman: <b>downloadProfileImage</b> (Public)</p> <p>Met dit endpoint kun je een profielafbeelding downloaden. De naam van het bestand moet worden ingevoerd op de locatie van de padvariabele {fileName}.</p> <p><b>Voorbeeld:</b> /download-profile-image/shield-profile.jpg</p>
/download-character-image/{fileName}	GET	Public	N/A	<p>Postman: <b>downloadCharacterImage</b> (Public)</p> <p>Met dit endpoint kun je een afbeelding van een karakter downloaden. De naam van het bestand moet worden ingevoerd op de locatie van de padvariabele {fileName}.</p> <p><b>Voorbeeld:</b> /download-character-image/hulk-character.jpg</p>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

FILE ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/download-movie-image/{fileName}	GET	Public	N/A	<p>Postman: <b>downloadMovieImage</b> (Public)</p> <p>Met dit endpoint kun je een afbeelding van een film downloaden. De naam van het bestand moet worden ingevoerd op de locatie van de padvariabele {fileName}.</p> <p><b>Voorbeeld:</b> /download-movie-image/thor-movie.jpg</p>
/upload-profile-image/{profileId}	POST	Admin, User		<p>Postman: <b>uploadProfileImageById</b></p> <p>Met dit endpoint kun je een afbeelding uploaden naar een bepaald profiel met een ID. De profiel-ID moet worden ingevoerd op de locatie van de padvariabele {profileId}.</p> <p>Je kunt .jpg- of .png-bestanden uploaden.</p> <p><b>Voorbeeld:</b> /upload-profile-image/1</p>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

FILE ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/upload-character-image/{characterId}	POST	Admin	N/A	<p>Postman: <b>uploadCharacterImageByld</b></p> <p>Met dit endpoint kun je een afbeelding uploaden naar een bepaald personage met een id. De character-ID moet worden ingevoerd op de locatie van de padvariabele {characterId}. Je kunt .jpg- of .png bestanden uploaden.</p> <p><b>Voorbeeld:</b> /upload-character-image/1000</p>
/upload-movie-image/{movieId}	POST	Admin	N/A	<p>Postman: <b>uploadMovieImageByld</b></p> <p>Met dit endpoint kun je een afbeelding met een id naar een bepaalde film uploaden. De film-ID moet worden ingevoerd op de locatie van de padvariabele {movieId}.</p> <p><b>Voorbeeld:</b> /upload-movie-image/1000</p>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

FILE ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/delete-profile-image/{profileId}	DELETE	Admin, User	N/A	<p>Postman: <b>deleteProfileImageById</b></p> <p>Met dit endpoint kun je een afbeelding op een profiel met een bepaalde ID verwijderen. De profiel-ID moet worden ingevoerd op de locatie van de padvariabele {profileId}.</p> <p><b>Voorbeeld:</b> /delete-profile-image/1</p>
/delete-character-image/{characterId}	DELETE	Admin	N/A	<p>Postman: <b>deleteCharacterImageById</b></p> <p>Met dit endpoint kun je een afbeelding verwijderen die is gekoppeld aan een personage met een bepaalde ID. De character-ID moet worden ingevoerd op de locatie van de padvariabele {characterId}.</p> <p><b>Voorbeeld:</b> /delete-character-image/1000</p>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

FILE ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/delete-movie-image/{movieId}	DELETE	Admin	N/A	<p>Postman: <b>deleteMovieImageById</b></p> <p>Met dit endpoint kun je een afbeelding verwijderen die is gekoppeld aan een film met een bepaalde ID. De film-ID moet worden ingevoerd op de locatie van de padvariabele {movieId}.</p> <p><b>Voorbeeld:</b> /delete-movie-image/1000</p>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

MOVIES ENDPOINTS				
Endpoint	Methode	Toegang	JSON	Toelichting
/movies	GET	Admin, User	N/A	Postman: <b>getAllMovies</b>  Met dit endpoint kun je alle films bekijken die momenteel in de database staan.
/movies/{movieId}	GET	Admin, User	N/A	Postman: <b>getMovieById</b>  Met dit endpoint kun je een bepaalde film bekijken met een id. De film-ID moet worden ingevoerd op de locatie van de padvariabele {movieId}.  <b>Voorbeeld:</b> /movies/1
/movies/search/starting-with	GET	Admin, User	N/A	Postman: <b>findMoviesByTitleStartingWith</b>  Met dit endpoint kun je naar een film zoeken door een titel als query parameter in te vullen.  <b>Voorbeeld:</b> /movies/search/starting-with?title=cap



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

MOVIES ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/movies/search/contains	GET	Admin, User	N/A	Postman: <b>findMoviesByTitleContains</b>  Met dit endpoint kun je naar een film zoeken door een titel als query parameter in te vullen.  <b>Voorbeeld:</b> /movies/search/contains?title=amer
/movies/search/sorted-asc	GET	Admin, User	N/A	Postman: <b>findMoviesByTitleSortedAsc</b>  Met dit endpoint kun je naar een film zoeken door een titel in te vullen als query parameter. De resultaten verschijnen in oplopende volgorde.  <b>Voorbeeld:</b> /movies/search/sorted-asc?title=avengers



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

MOVIES ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/movies/ search/sorted- desc	GET	Admin, User	N/A	Postman: <b>findMoviesByTitleSortedDesc</b>  Met dit endpoint kun je naar een film zoeken door een titel in te vullen als query parameter. De resultaten komen in aflopende volgorde terug.  <b>Voorbeeld:</b> /movies/search/sorted-desc?title=Avengers
				Postman: <b>findMovieByYearOfRelease</b>  Met dit endpoint kun je naar een film zoeken door als query parameter het jaar in te vullen waarin de film is uitgebracht.  <b>Voorbeeld:</b> /movies/search/year?year=2008



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

MOVIES ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/movies/movie/ {movieId}	PUT	Admin	{  "movieTitle": "Titel van de film.",  "movieSummary": "Samenvatting van de film.",  "movieYearOfRelease": "2008"  }	Postman: <b>updateMovieById</b>  Met dit endpoint kun je een film bijwerken met een bepaalde ID en de vereiste JSON-body invullen. De film-ID moet worden ingevoerd op de locatie van de padvariabele {movieId}.  <b>Voorbeeld:</b> /movies/movie/1000
				Postman: <b>createMovie</b>  Met dit endpoint kun je een film maken door de vereiste JSON-body in te vullen.



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

MOVIES ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/movies/{movieId}	DELETE	Admin	N/A	Postman: <b>deleteMovieById</b>
				Met dit endpoint kun je een film verwijderen door de vereiste ID in te vullen. De film-ID moet worden ingevoerd op de locatie van de padvariabele {movieId}.
				Momenteel mogen films die aan een personage of personages zijn gekoppeld, niet worden verwijderd. Gebruik eerst de <b>disassociateMovieAndCharacter</b> -methode en probeer vervolgens opnieuw de film te verwijderen.  <b>Voorbeeld:</b> /movies/1000



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

PROFILES ENDPOINTS				
Endpoint	Methode	Toegang	JSON	Toelichting
/profiles	GET	Admin	N/A	Postman: <b>getAllProfiles</b>
				Met dit endpoint kun je alle profielen opvragen en bekijken die zich momenteel in de database bevinden.
				Postman: <b>getProfileById</b>
/profiles/{profileId}	GET	Admin	N/A	Met dit endpoint kun je een specifiek profiel met een ID bekijken. De profiel-ID moet worden ingevoerd op de locatie van de padvariabele {profileId}.  <b>Voorbeeld:</b> /profiles/1
				Postman: <b>findProfileByFirstnameContains</b>
/profiles/search/firstname	GET	Admin	N/A	Met dit endpoint kun je een profiel zoeken door de voornaam van een gebruiker in te voeren als query parameter.
				<b>Voorbeeld:</b> /profiles/search/firstname?firstname=p





# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

PROFILES ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/profiles/search/lastname	GET	Admin	N/A	Postman: <b>findProfileByLastName</b>  Contains  Met dit endpoint kun je een profiel zoeken door de achternaam van een gebruiker in te voeren als query parameter.  <b>Voorbeeld:</b> /profiles/search/lastname?lastname=groot
				Postman: <b>getProfileByUsername</b>  Met dit endpoint kun je een profiel zoeken door de gebruikersnaam van een gebruiker in te voeren. De gebruikersnaam van de gebruiker moet worden ingevoerd op de locatie van de padvariabele {username}.  <b>Voorbeeld:</b> /profiles/profile/andre



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

PROFILES ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/profiles/image/user/{username}	GET	Admin	N/A	Postman: <b>getProfileImageByUsername</b>  Met dit endpoint kun je een specifieke profielafbeelding bekijken door een gebruikersnaam in te voeren. De gebruikersnaam van de gebruiker moet worden ingevoerd op de locatie van de padvariabele {username}.  <b>Voorbeeld:</b> /profiles/profile/andre
				Postman: <b>getProfileImageByProfileId</b>  Met dit endpoint kun je een specifieke profielafbeelding bekijken door een ID in te voeren. De profiel-ID moet worden ingevoerd op de locatie van de padvariabele {profileId}.  <b>Voorbeeld:</b> /profiles/image/profile/1000



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

PROFILES ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/profiles/profile/{profileId}	PUT	Admin, User		Postman: <b>updateProfileById</b>
			{ "firstName": "André", "lastName": "de Groot", "gender": "Male", "dateOfBirth": "01-01-2000" }	Met dit endpoint kun je een specifiek profiel bijwerken door een ID en de vereiste JSON-body in te voeren. De profiel-ID moet worden ingevoerd op de locatie van de padvariabele {profileId}.  Het geslacht heeft 3 mogelijke opties: Male, Female en X.  <b>Voorbeeld:</b> /profiles/profile/1000
/profiles	POST	Admin, User	{ "firstName": "André", "lastName": "de Groot", "gender": "Male", "dateOfBirth": "01-01-2000" }	Postman: <b>createProfile</b>  Met dit endpoint kun je een profiel maken door de vereiste informatie in de JSON-body in te voeren.  Het geslacht heeft 3 mogelijke opties: Male, Female en X.



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

PROFILES ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/profiles/profile/{profileId}	DELETE	Admin	N/A	Postman: <b>deleteProfileById</b>
				Met dit endpoint kun je een profiel verwijderen door een ID op te geven. De profiel-ID moet worden ingevoerd op de locatie van de padvariabele {profileId}.
/profiles/{profileId}				Momenteel mogen profielen die aan gebruikers zijn toegewezen, niet worden verwijderd. Gebruik eerst de <b>detachProfileFromUser</b> -methode en probeer vervolgens opnieuw het profiel te verwijderen.  <b>Voorbeeld:</b> /profiles/1



LIJST ENDPOINTS

OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS				
Endpoint	Methode	Toegang	JSON	Toelichting
/users	GET	Admin	N/A	<div>Postman: <b>getAllUsers</b></div> <div>Met dit endpoint kun je alle gebruikersaccounts bekijken.</div>
/users/{username}	GET	Admin, User	N/A	<div>Postman: <b>getUserByUsername</b></div> <div>Met dit endpoint kun je een specifiek gebruikersaccount bekijken door een gebruikersnaam op te geven. De gebruikersnaam van de gebruiker moet worden ingevoerd op de locatie van de padvariabele {username}.</div> <div>Voorbeeld: /users/andre</div> <div>Postman: <b>getActiveUsers</b></div>
/users/active	GET	Admin	N/A	<div>Postman: <b>getActiveUsers</b></div> <div>Met dit endpoint kun je alle actieve gebruikers bekijken van wie het account is ingeschakeld.</div>



LIJST ENDPOINTS

OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/users/favorites/{username}	GET	Admin	N/A	<div>Postman: <b>getAllFavoritesFromUser</b></div> <div>Met dit endpoint kun je alle favorieten (Characters) van een specifieke gebruiker bekijken door de gebruikersnaam van dat account op te geven. De gebruikersnaam van de gebruiker moet worden ingevoerd op de locatie van de padvariabele {username}.</div> <div>Voorbeeld: /users/favorites/andre</div> <div>Postman: <b>findUserByEmail</b></div>
/users/search/email	GET	Admin	N/A	<div>Postman: <b>findUserByEmail</b></div> <div>Met dit endpoint kun je naar een gebruiker zoeken door een e-mailadres op te geven.</div> <div>Voorbeeld: /users/search?email=admin@foundfave.online</div>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/users/search/email	GET	Admin	N/A	Postman: <b>findUserByEmailContains</b>  Met dit endpoint kun je naar een gebruiker of gebruikers zoeken door een e-mailadres of een deel van een e-mailadres op te geven.  <b>Voorbeeld:</b> /users/search/email?email=admin
				Postman: <b>findUserByUsernameContains</b>  Met dit endpoint kun je naar een gebruiker zoeken door een gebruikersnaam of een deel van een gebruikersnaam op te geven.  <b>Voorbeeld:</b> /users/search/username?username=admin



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/users/exists/{username}	GET	Admin	N/A	Postman: <b>doesUserExist</b>  Met dit endpoint kun je zien of een gebruiker bestaat door de gebruikersnaam op te geven. Het resultaat is waar of onwaar. De gebruikersnaam van de gebruiker moet worden ingevoerd op de locatie van de padvariabele {username}.  <b>Voorbeeld:</b> /users/exists/andre
				Postman: <b>getUserAuthorities</b>  Met dit endpoint kun je zien welke rollen een gebruiker heeft door de gebruikersnaam op te geven. De gebruikersnaam van de gebruiker moet worden ingevoerd op de locatie van de padvariabele {username}.  <b>Voorbeeld:</b> /users/andre/authorities



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/users/admin/{username}	PUT	Admin	{ "password": "bAn@nen1" }	Postman: <b>updateUserPasswordAdmin</b> (Admin only)
				Met dit endpoint kan de beheerder het wachtwoord van een gebruiker overschrijven door een nieuw wachtwoord op te geven in de JSON-body. De gebruikersnaam van de gebruiker moet worden ingevoerd op de locatie van de padvariabele {username}.  <b>Password moet het volgende bevatten:</b>  1. Minimaal 1 kleine letter. 2. Minimaal 1 hoofdletter/ 3. Minimaal 1 nummer. 4. Minimaal 1 symbool. 5. Het moet tussen de 8 en 20 tekens lang zijn.  <b>Voorbeeld:</b> /users/admin/andre



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/users/user/{username}	PUT	User	{ "password": "bAn@nen2", "apikey": "12345678", "email": "andre@novi.nl" }	Postman: <b>updateUserDetails</b> (User only)
				Met dit endpoint kan de gebruiker diens account bijwerken. Werk de gebruikersgegevens bij door de vereiste gegevens op te geven in de JSON-body. De gebruikersnaam van de gebruiker moet worden ingevoerd op de locatie van de padvariabele {username}.  <b>Password moet het volgende bevatten:</b>  1. Minimaal 1 kleine letter. 2. Minimaal 1 hoofdletter/ 3. Minimaal 1 nummer. 4. Minimaal 1 symbool. 5. Het moet tussen de 8 en 20 tekens lang zijn.  <b>Voorbeeld:</b> /users/user/andre



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/users/assign/profile/{username}	PUT	Admin	{ "id": 1 }	<p>Postman: <b>assignProfileToUser</b></p> <p>Met dit endpoint kan de beheerder een nieuw profiel aan een gebruiker toewijzen door de vereiste profiel-ID op te geven. De gebruikersnaam moet worden ingevoerd op de locatie van de padvariabele {username}.</p> <p><b>Voorbeeld:</b> /users/assign/profile/andre</p>
/users/detach/profile/{username}	PUT	Admin	N/A	<p>Postman: <b>detachProfileFromToUser</b></p> <p>Met dit endpoint kan de beheerder een profiel loskoppelen van een gebruiker. De gebruikersnaam moet worden ingevoerd op de locatie van de padvariabele {username}.</p> <p><b>Voorbeeld:</b> /users/detach/profile/andre</p>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/users/add/character/{username}	PUT	Admin, User	{ "id": 1 }	<p>Postman: <b>addFavoriteCharacterToUser</b></p> <p>Met dit endpoint kan de gebruiker een Marvel-personage aan zijn of haar favorietenlijst toevoegen door de ID van het personage in de JSON-body in te voeren. De gebruikersnaam van de gebruiker moet worden ingevoerd op de locatie van de padvariabele {username}.</p> <p><b>Voorbeeld:</b> /users/add/character/andre</p>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/users	POST	Public	{ "username": "andre", "password": "pAssword1@", "email": "andre4@novi.nl" }	<p>Postman: <b>createUser</b></p> <p>Met dit endpoint kan de gebruiker een nieuw gebruikersaccount maken door de vereiste informatie in de JSON-body in te voeren.</p> <p><b>Password moet het volgende bevatten:</b></p> <ol style="list-style-type: none"><li>1. Minimaal 1 kleine letter.</li><li>2. Minimaal 1 hoofdletter/</li><li>3. Minimaal 1 nummer.</li><li>4. Minimaal 1 symbool.</li><li>5. Het moet tussen de 8 en 20 tekens lang zijn.</li></ol>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/users/ {username}/ authorities	POST	Admin	{ "authority": "ROLE_ADMIN" }	<p>Postman: <b>addUserAuthority</b></p> <p>Met dit endpoint kan de beheerder een gebruiker promoveren tot beheerder door de vereiste informatie in de JSON-body in te voeren. De gebruikersnaam van de gebruiker moet worden ingevoerd op de locatie van de padvariabele {username}.</p> <p><b>Voorbeeld:</b> /users/ andre4/authorities</p>



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/users/{username}	DELETE	Admin	N/A	Postman: <b>deleteUser</b>  Met dit endpoint kan de beheerder een gebruiker verwijderen door de gewenste gebruikersnaam in te voeren. De gebruikersnaam van de gebruiker moet worden ingevoerd op de locatie van de padvariabele {username}.  Momenteel mogen gebruikers aan wie profielen of contactformulieren zijn toegewezen, niet worden verwijderd. Gebruik eerst de <b>detachProfileFromUser</b> -methode of probeer de <b>deleteContactFormById</b> -methode en probeer vervolgens opnieuw de gebruiker te verwijderen.  <b>Voorbeeld:</b> /users/andre



# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/users/{username}/authorities/{authority}	DELETE	Admin	N/A	Postman: <b>removeUserAuthority</b>  Met dit endpoint kan de beheerder een beheerder degraderen tot gebruiker door de gewenste gebruikersnaam en ROLE_ADMIN in te voeren. De gebruikersnaam en autoriteit (ROLE_ADMIN) van de gebruiker moeten worden ingevoerd op de locaties van de padvariabelen {username} en {authority}.  <b>Voorbeeld:</b> /users/andre4/authorities/ROLE_ADMIN





# LIJST ENDPOINTS

## OVERZICHT ENDPOINTS (VERVOLG)

USERS ENDPOINTS (VERVOLG)				
Endpoint	Methode	Toegang	JSON	Toelichting
/users/remove/character/{username}	DELETE	Admin, User	N/A	<p>Postman: <b>removeFavoriteCharacterFromUser</b></p> <p>Met dit endpoint kan de beheerder of gebruiker een Marvel-personage uit zijn of haar favorietenlijst verwijderen. De gebruikersnaam van de gebruiker moet worden ingevoerd op de locatie van de padvariabele {username}.</p> <p><b>Voorbeeld:</b> /users/remove/character/andre</p>



# TROUBLESHOOTING

## INLEIDING

**Probleem:** PgAdmin: De server (PostgreSQL 16) wilt niet verbinden na het invoeren van het wachtwoord.

**Antwoord:** Het poortnummer staat mogelijk niet goed ingesteld. Deze kun je instellen door met de rechtermuisknop op “PostgreSQL” te klikken en daarna “Properties...”. Daarna tabblad Connection > Port. Vul hier 5432 in.

**Probleem:** Er zijn geen tabellen zichtbaar in de foundfave-api database.

**Antwoord:** Rechtermuisknop op “Tables” en kies daarna “Refresh”. Mocht dat niet werken:

Zijn de gegevens correct ingevuld in het application.properties bestand? Staat de IntelliJ server aan? Geeft de terminal foutmeldingen weer?

**Probleem:** “The SDK is not specified for module” foutmelding.

**Antwoord:** Ga naar Menu: File > Project Structure... > SDK > Kies of voeg een JDK toe.

**Probleem:** De Postman API verzoeken doen het niet en ik krijg een 403 response.

**Antwoord:** Het is mogelijk dat je de Jwt-token code hebt overgenomen van de afbeelding in het stappenplan. Herhaal stap 1 tot en met 10 in het hoofdstuk **JWT-Token in Postman** (En dit keer niet de code van de afbeelding overnemen!)

**Antwoord 2:** Het is ook mogelijk dat de geldigheid van het JWT-token is verlopen. JWT-tokens zijn momenteel ingesteld om na 10 dagen te verlopen (In verband met diverse test doeleinden).

Om de geldigheid van het JWT-token te verlengen kun je opnieuw inloggen, met het / login endpoint. Als Postman het JWT-token retourneert, kun je deze plakken in je variabele. Bekijk het hoofdstuk **JWT-Token in Postman** voor meer details.



# TROUBLESHOOTING

---

## TROUBLESHOOTING (VERVOLG)

**Probleem:** Help! Ik kan het Postman collectie JSON bestand niet vinden.

**Antwoord:** Het Postman collectie JSON bestand kun je [hier downloaden](#) en klik rechtsboven "Download raw file" om het bestand te downloaden van Github.

Het Postman collectie JSON bestand staat ook in de root van het project.