twan van maastricht

2113135

ajmaastr@avans.nl

programmeren 4 Practicum 1

Werken met Node.js, Git en Heroku

# Opdracht 1

**Wat is de naam van Pokémon nummer 7 en nummer 18? Laat het JSON fragment zien met het antwoord (meerdere mogelijkheden).**

**Input:** <http://pokeapi.co/api/v2/pokemon/7>

**JSON fragment:** 

**Input:** http://pokeapi.co/api/v2/pokemon/18

**JSON fragment:** 

**Wat is de naam van planeet nummer 7 en nummer 18 in de Star Wars database? Laat het JSON fragment zien.**

Input: <https://swapi.co/api/planets/7>

JSON Fragment: https://lh4.googleusercontent.com/wChatWbJp6wtwmr6gEvJhVcV4iaa-UEFLpqwZMh6hgAgAbmONSJ9TURL0rQW9u2xtYH03ZJW-SqEnTetxw_ZqCxdU3u768bIYqNzHYFVyEWYvhUKD-uVlBX1VRdVgzc0lm7SoGzm

Input: [https://swapi.co/api/planets/](https://swapi.co/api/planets/7)18

JSON Fragment: https://lh3.googleusercontent.com/Viy139hFUe5EAc_gLmnhfT-kmIV7FGxq8wgcf_QwdvAonxIB8rKfRAdVCFEXI-VYGrrH42qxBFNB9McFx1B7kORgOIC-gtVW_7ZFJoagoazbSP-tBeVIS9RP_nveWVRlqr6dTYry

# Opdracht 2

1. **Aan welke eisen moet de "name" van je project voldoen?**

kleine letters, geen spaties, (lage) streepjes mogen

1. **Wat betekent de '^' (*caret* in het Engels) voor het typenummer achter "express":? Als er "^4.15.2" staat, welke nieuwe versies van express zijn dan voor dit project (deze package.json) toegestaan en welke niet:**

**4.0.0 4.15.0 4.15.3 4.16.0 4.16.2 5.0.0 5.15.2 6.3.4**

Staat modificaties toe aan de alle cijfers behalve het meest linkse, niet nul, cijfer in de [major, minor, patch] tuple.

Toegestane versies: 4.0.0, 4.15.0, 4.16.0, 4.16.2

1. **Welke response zie je in Postman? Klopt dat met de Javascript source code?**

Verwacht antwoord:

**GET:** Hello Avans!

POST: Hello Avans, POST request received!

PUT: Hello Avans, PUT request received!

Resultaat:

**GET:** 

**POST:** 

**PUT:** 

1. **Probeer een aantal paden in combinatie met localhost:3000 en beschrijf wat je ziet:**

<http://localhost:3000/about>**:** 

<http://localhost:3000/about/>: “ “

<http://localhost:3000/About>: “ “

<http://localhost:3000/ABOUT>: “ “

<http://localhost:3000/AboutThisApp>: krijg een HTML error pagina terug

<http://localhost/about>: Postman kan geen response terugkrijgen

<http://localhost:3000/api>: krijg een HTML error pagina terug

<http://localhost:3000/api/>: krijg een HTML error pagina terug

1. **Lees de documentatie op https://expressjs.com/en/guide/routing.html en vat je conclusies kort samen (<= 10 regels tekst).**

Routing is de definitie van applicatie endpoints en hoe ze reageren op verzoeken van een client. Een route methode is afgeleid van http methodes. De express class kan deze methodes gebruiken. App.all() is een speciale methode die niet is afgeleid van een http methode. Deze methode wordt aangeroepen na alle http methodes. In combinatie met request methoden verzorgen request paths de endpoints van een API. Route parameters zijn keys die een value vangen die bij die keys horen. Route handlers zijn callback functies die een request af kunnen handelen, fungerende als tussenpersonen. Response methodes zijn methodes die de client een antwoord geven.

1. **Test met een GET request naar localhost:3000/api in Postman. Werkt het ook voor een POST naar localhost:3000/api?**

Dit werkt niet. Bij een post krijg je een Error HTML page terug.

1. **Als we app.get('\*', ...) vervangen door app.all('\*', ...), lost dit dan het probleem op met de POST? Test in Postman en toon je resultaat.**

Ja, dit lost ook het probleem op met PUT:

1. **Wat gebeurt er als we de app.all('\*', ...) bovenaan zetten in de file, boven de eerste app.get(...)? Test met Postman, beschrijf je resultaat.**

Alle methodes die nu onder app.all(‘\*’, …) staan worden niet meer uitgevoerd. Alle get, post en put request krijgen nu het volgende terug als antwoord: 

# Opdracht 3