twan van maastricht

[ajmaastr@avans.nl](mailto:ajmaastr@avans.nl)

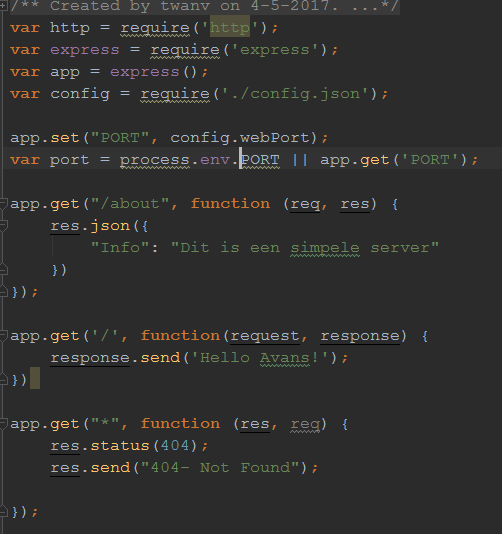
2113135

Practicum 2

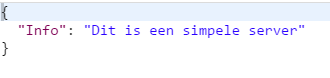
# Opdracht 1

**Maak in je projectdirectory een server.js file met een heel eenvoudige REST server die luistert op poort 8080. Kijk naar de practicumopgaven van practicum 1 en naar de presentatie van training 2 hoe dat ook alweer moest.**

**Test je REST server met een paar requests in Postman.**

**Server:** 

**Input:** http://localhost:3000/about

**JSON fragment:** 

**Input:** http://localhost:8080/

**JSON fragment:** 

# 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 5

1. **Test of je app werkt met Postman, maar geef nu de URL op die Heroku aan je app heeft toegewezen. Werkt je app op Heroku? Beschrijf het resultaat.**

Postman geeft een HTML error page terug. De app werkt dus niet op Heroku.

# Opdracht 6

1. **Ga na dat je app nu JSON output verstuurt op port 80 (de default voor HTTP).**

Resultaat van het laden van de app op Heroku:

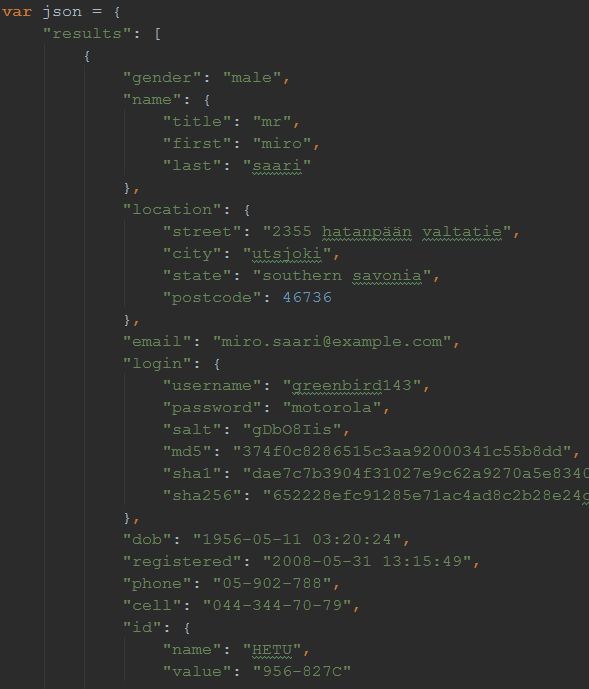


# Opdracht 7

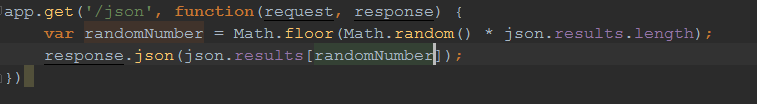
1. **Pas de code zo aan dat telkens de data van één van die gebruikers in JSON formaat wordt teruggestuurd op een GET request naar het pad ‘/api’ van je app, net als bij** [**http://randomuser.me/api**](http://randomuser.me/api)**. Als je Android app kan selecteren op gebruikers (b.v. alleen Nederlandse gebruikers of zoeken op naam), pas dan je REST server code zo aan dat ook die selectie in de API werkt.  
   Neem je gewijzigde code op in je verslag (niet de hele file, alleen het belangrijke stuk).**

**Code in server.js:**

*In var json staan 5 verschillende random users in een json array*



*/json get request geeft vervolgens een random user uit de array mee in json formaat.*



1. **Deploy je wijzigingen op Heroku en test met Postman. Zet een screenshot van Postman in je verslag om te laten zien dat het werkt.**

Resultaat: 