twan van maastricht

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

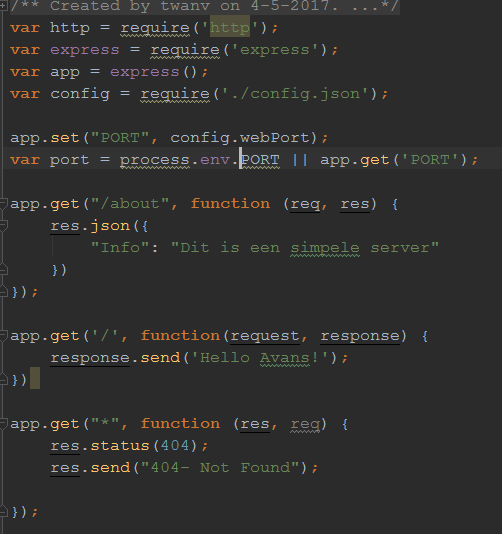
2113135

Practicum 2

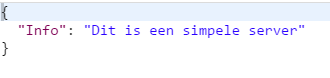
# Opdracht 1

**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:** 

**fffffff**

# Opdracht 2

# 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:

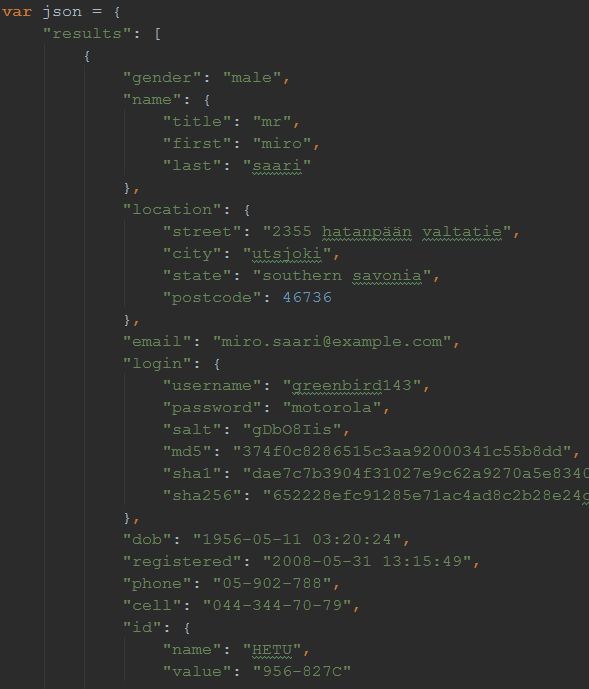


# Opdracht 7

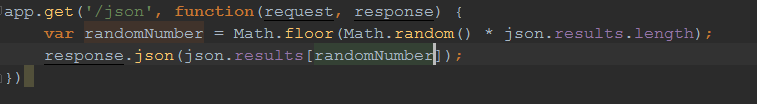
1. **Pas de code zo aan dat telkens de data van één van die gebruikers in JSON formaat wordt teruggestuurd op een GET request naar het pad ‘/api’ van je app, net als bij** [**http://randomuser.me/api**](http://randomuser.me/api)**. Als je Android app kan selecteren op gebruikers (b.v. alleen Nederlandse 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: 