* Server en client draait op hetzelfde computer
* Het is erg belangrijk om te weten welk code op de server draait en welke code op de client.
  + Server side
    - Express
    - Node
  + Client side
    - React
    - Javascript
    - Typescript
    - HTML/CSS
* Server response bestaat voornamelijk uit JSON
* De JSON data hangt af van de aanvraag van de gebruiker.
* Express: App.listen gaat voor app.get
* Process.env.PORT is een omgevingvariabele, die veranderd op basis van welk apparaat het programma zich bevindt.
* In een request is het altijd belangrijk om een response terug te sturen.
  + Je mag maar 1 antwoord teruggeven
  + De response bestaat uit headers en content
  + Het maakt niet erg uit of de response via .json of via .send gebeurt.
    - De header wordt automatisch naar application/json geconverteerd.
    - Send past de header aan op basis van de content
  + De belangrijkste header is de status code
* Het is geen probleem als een URL onhandig lang is.
* De request is wat de client naar de server toestuurt, en de response is wat de server naar de client terugstuurt na aanleiding van de request.
  + Res.status(code)
  + Res.json([‘item1’ , ‘item2’]). Express zet dit om naar JSON
* In de URL kun je de naam van de property van een object in de url zetten met een dubbele punt. Bijvoorbeeld /action/:naam/goto
  + Als een variabele naar een entiteit/resource verwijst, of als je een entiteit uniek kan identificeren, dan is het handig om dit te gebruiken.
* Na res.json mag je nog wel andere code uitvoeren, zolang het niet meer een response terugstuurt naar de client.
* Express().use wordt altijd als eerst uitgevoerd, maar je moet wel aangeven dat deze functie verder moet. Dit geef je aan door de functie next.
* Het is mogelijk om een error af te handelen met een middleware, door middel van app.use. Wel rekening houden met de status code.
* Middleware body parser:
  + Alle velden in een google formulier wordt niet in de URL gestopt, maar in de body van de request.
  + Body parser moet als eerst geinstalleerd worden als dit nog niet gedaan is
  + bodyParser retourneert een functie, die je configureert om iets te doen.

Samenvatting:

* Client stuurt een HTTP request
* Wordt opgevangne door express
* Express heeft een lijst van functies die wordt uitgevoerd op basis van de request
* Voert altijd de eerste uit als het niet een request is, dus use.
* De body kan in elk handler gebruikt worden met req.body
* Zonder middleware kun je req.params gebruiken. Aangegeven door :