

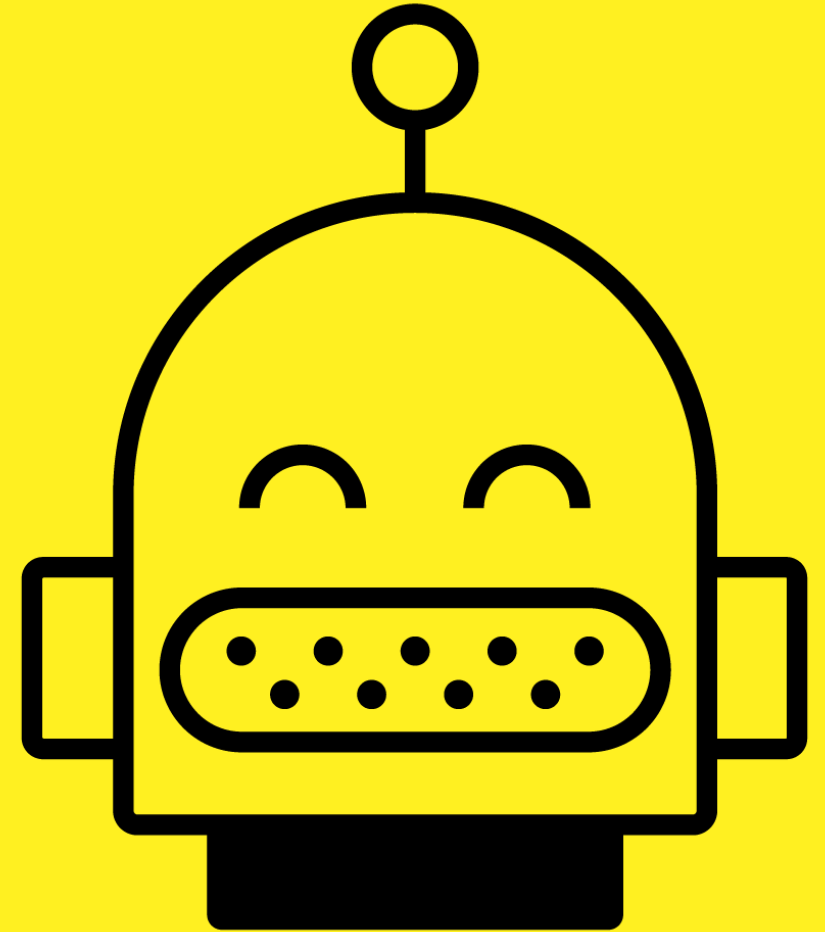


Project Tech

Verdieping Backend #4

Handige modules

Collegejaar 23/24





Modules

- In deze les bespreken we een aantal modules die handig kunnen zijn voor je project
- Of je ze kunt en wilt gebruiken, hangt mede af van je functionaliteit
- Installeer gewenste modules met `npm install` en `require` ze vervolgens in je code



Uit de workshop security

- XSS en Validator.js voor input sanitisation

```
<script>alert("hack");</script>
```



```
&lt;script&gt;alert("hack");&lt;/script&gt;
```

- Bcrypt of bcryptjs voor password hashing

Geheim



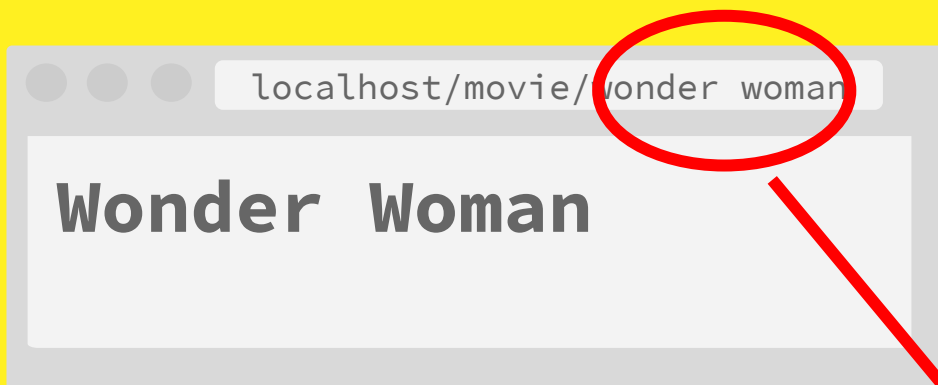
```
$2b$10$3k9m5HOUOMPh1.K.FUUIsOGCDPmd5h2Om9dRHlj7j2xOM1SUKBse2
```



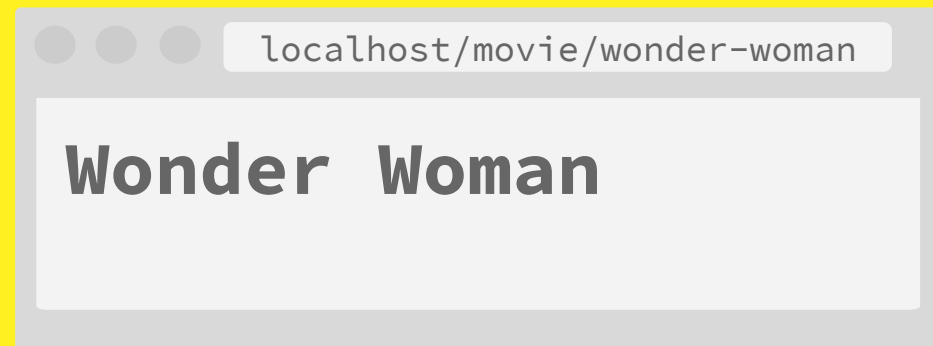
SLUG

SLUG maakt tekst URL-safe

```
const title = req.params.title  
  
<a href="movie/<%= title %>">
```



```
const title = slug(req.params.title)  
  
<a href="movie/<%= title %>">
```



Er mogen geen spaties of andere bijzondere tekens in een URL



Multer

Om via een formulier
files te uploaden naar
onze server

We can add files!

The screenshot shows a web browser window with the address bar displaying 'localhost:8000/add'. The page has a dark background and features the title 'Add a new movie' in large white text. Below the title, there are three input fields: the first contains 'Wonder Woman', the second contains 'Diana, an Amazonian warrior...', and the third is labeled 'Cover' and contains 'wonder-woman.jpg' (highlighted in red). Below these fields is a text area containing the text 'When a pilot crashes and tells of conflict in the outside world, Diana, an Amazonian warrior in training, leaves home ...'. At the bottom of the form is a white button with the text 'add'.

view/add.ejs

```
<form action="/add-movie" method="post" enctype="multipart/form-data">  
  <input name="cover" type="file" accept="image/*">  
</form>
```

Accept only images

Forms with text and files
need multipart encoding

server.js

```
const multer = require('multer')
const upload = multer({dest: 'static/upload/'})

express()
  .post('/add-movie', upload.single('cover'), add)

function add(req, res) {
  console.log(req.file.filename)
}
```

Bewaar de files hier

**Sla een file op met
upload.single()**

Multer vult req.file



Multer tips

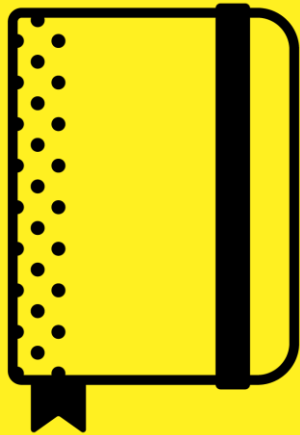
1. Gebruik Multer niet voor formulieren zonder `<input type="file">`, dan gaan dingen kapot!
2. Multer slaat de bestanden voor ons op op de webserver, wij hoeven alleen `req.file.filename` te onthouden of op te slaan in de database
3. Sla de bestanden op in een mapje onder je map met static content, zodat je server ook de nieuwe plaatjes als static content kan serveren



Recap

Wat is het verschil tussen...?

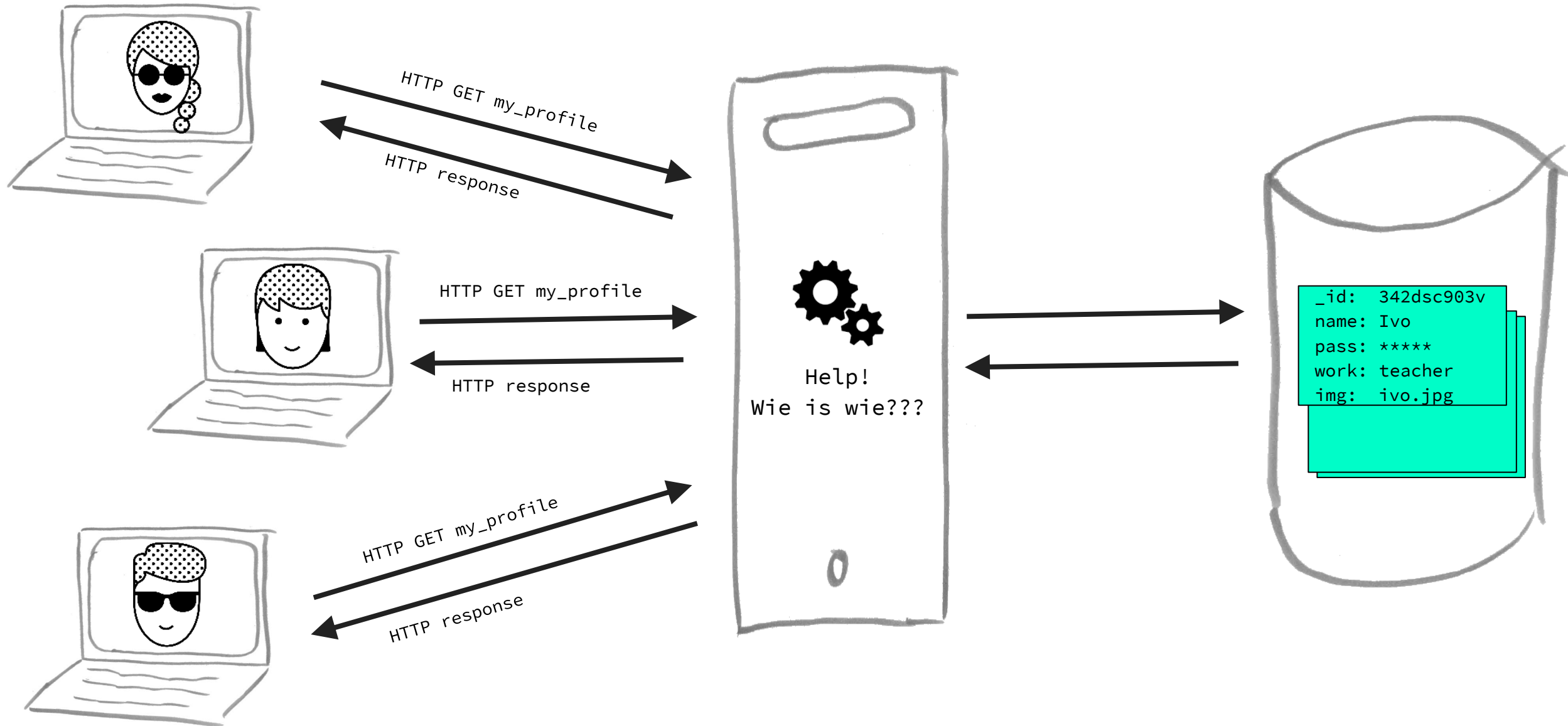
- `req.params`
- `req.body`
- `req.query`
- `req.file`



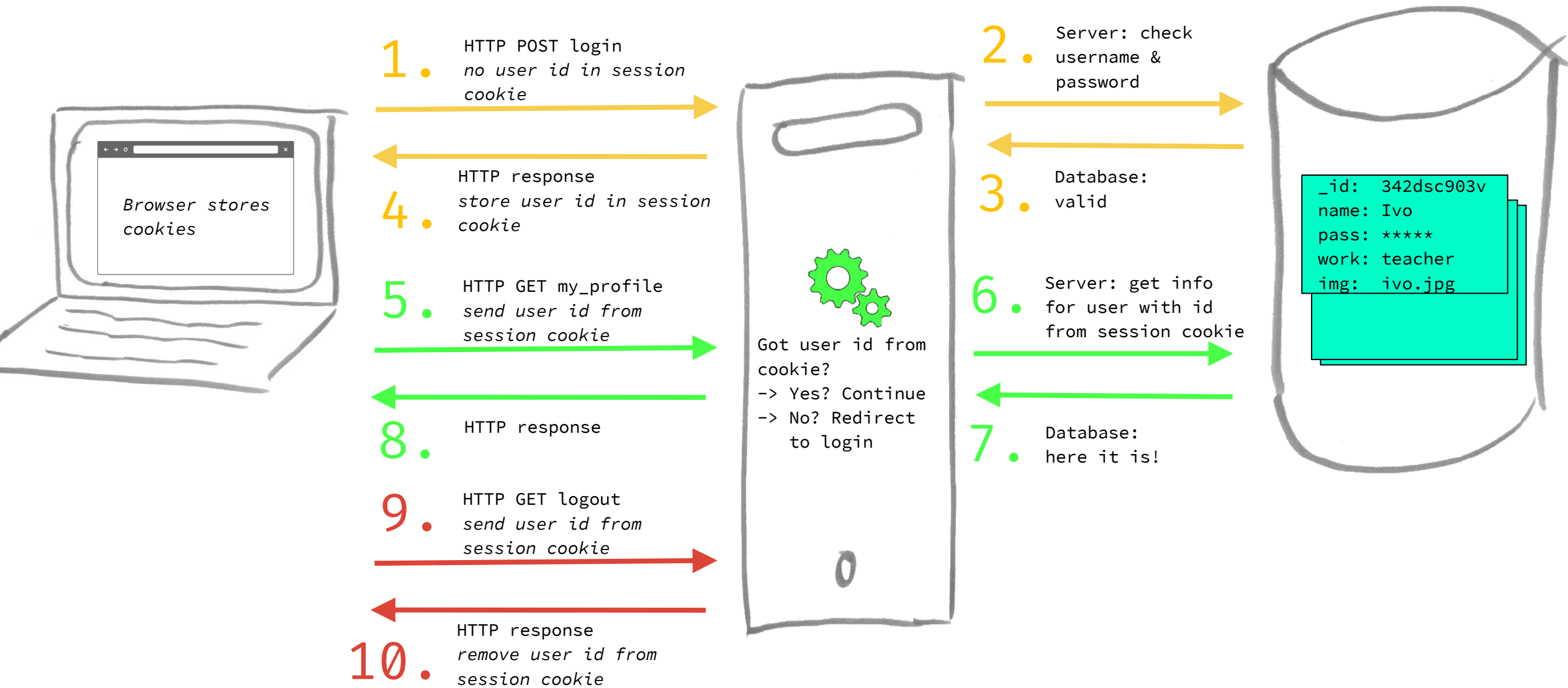
Theorie

Sessions

Sessions – waarom?



Sessions – hoe?





Sessions

Client-side sessions use cookies and cryptographic techniques to maintain state without storing much data on the server.

When presenting a dynamic web page, the server sends the current state data to the client (web browser) in the form of a cookie. The client saves the cookie in memory or on disk.

([wikipedia.org](https://en.wikipedia.org))



Sessions

Kunnen bijvoorbeeld worden gebruikt:

- Om te kijken of iemand ingelogd is
- Om bij te houden wie iemand is
- Te onthouden welke producten je bekeken hebt
- Om de voortgang bij te houden bij het invullen van een formulier van meerdere pagina's
- ...



Using Sessions

- Gebruik de module `express-session`
- De sessies worden versleuteld met een door jou te kiezen secret (een soort geheim wachtwoord)
- Zet je dit secret in je code?



(Waar dan wel?)

server.js

```
const session = require('express-session')

app
  .use(session({
    resave: false,
    saveUninitialized: true,
    secret: process.env.SESSION_SECRET
  })))
```

.env

```
DB_HOST=localhost
DB_USER=root
DB_NAME=mymoviewebsite
DB_PASSWORD=mypassword
SESSION_SECRET=ilikecats
```