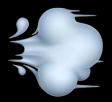
back-end

Teams, MVC and Sessions

lab 6/8



Adem in. Adem uit.

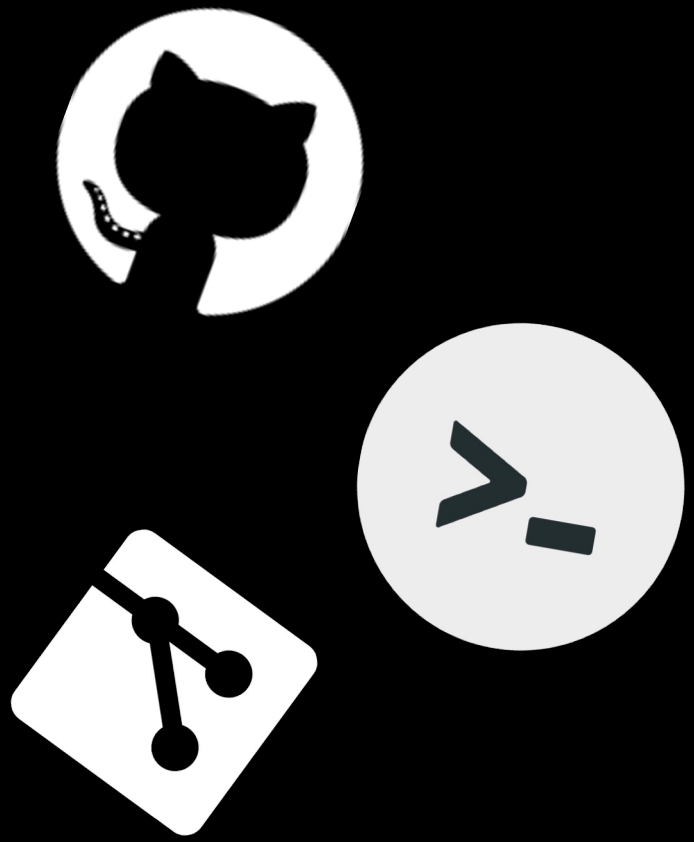
today

I. Teams assignment & A2

II. Topics

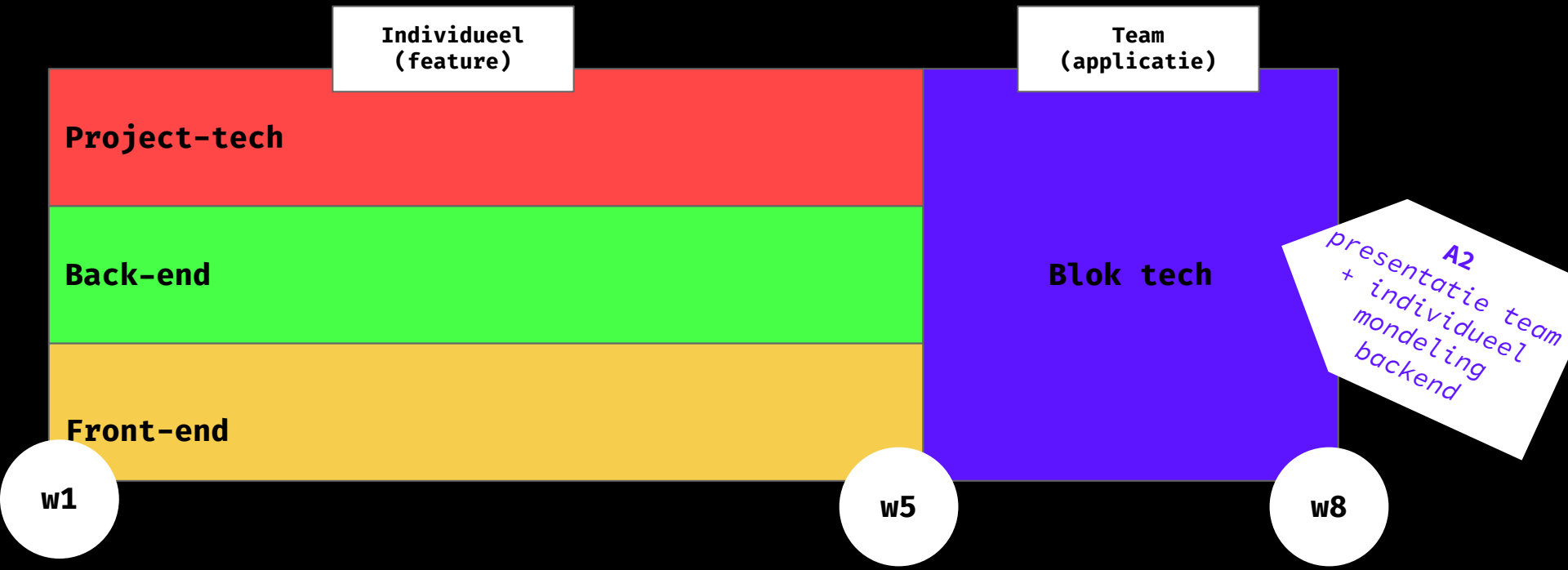
III. MVC Model

IV. Sessions and Storage



Team Assignment

tl;dr: **in een sprint van ~2 weken als team de matching-app verder uitwerken**


Individueel
(feature)
Team
(applicatie)
Project-tech
Back-end
Front-end
Blok tech
A2
presentatie team
+ individueel
mondeling
backend
w1
w5
w8

Team assigment

deliverables

- **Team prototype:** working **interactive matching application** with your team
- **Process book (wiki):** that provides insight into the **weekly iterative process** and the research

Team assigment

Grade

Team presentation at the end of the sprint to show the project. Then an assessment with each teacher from each course to show your contribution **to get an individual grade.**

A2

Grade

This is a team assignment but you'll be **graded individually!** Each prototype contains an **individually recognizable contribution** of you based on the learning goals and topics of this course. **So, you need to show you worked on something for back-end.** For example; you can't just work on the CSS of the project because that was your role in the team.

Rubric A2

The rubric for A2 is visible on github

https://github.com/cmda-bt/be-course-22-23/blob/main/grading/a2.md

Topics (bonus)

topic

Additionaly you can pick one or more topics to get **bonus points**. It's important that you really do a deep-dive into the topic. So extensive research documented in the wiki and advanced implementation.

> Pick one or more topics from the list below and work on those topics for your team assessment to get bonus points.

Synopsis

- **Homework**
- **Time**: 10:00h
- **Goals**: subgoal 10,
- **Due**: before week 8

Description

The topic can be anything related to back-end. The list below will give you some starting points. There are things like security enhancements, application structure, p[illegible]ptimizations and many more.

Pick a **topic from the list**

Or add your own!

→ List of topics for inspiration

> Something not on the list? These are topics we [illegible]h but if you always wanted to learn something [illegible]e

topic

Additionaly you can pick one or more topics to get **bonus points**. It's important that you really do a deep-d[illegible] extensive research documented in the wiki and advanced implementation.

> Pick one or more topics from the list below and work on those topics for your team assessment to get bo[illegible]

Pick a **topic from the list**

Synopsis

- **Homework**
- **Time**: 10:00h
- **Goals**: subgoal 10

Make sure members of
your team have a **different topic!**
And you **can't pick a topic you already did.**
Divide-and-conquer

Assignment (±15m)

Have a look at the topics and discuss with your team members which one you'll want to work on.

- *Does the topic match with the feature you've created?*
- *Is the topic something you want to learn?*
- *Discuss with your team which topic you'll work on.*

MVC Model

MVC Model

?

Model-view-controller (MVC) is a software design pattern commonly used for developing user interfaces that divide the related program logic into **three interconnected elements.**

This is done to separate internal representations of information from the ways information is presented to and accepted from the user

wikipedia.org

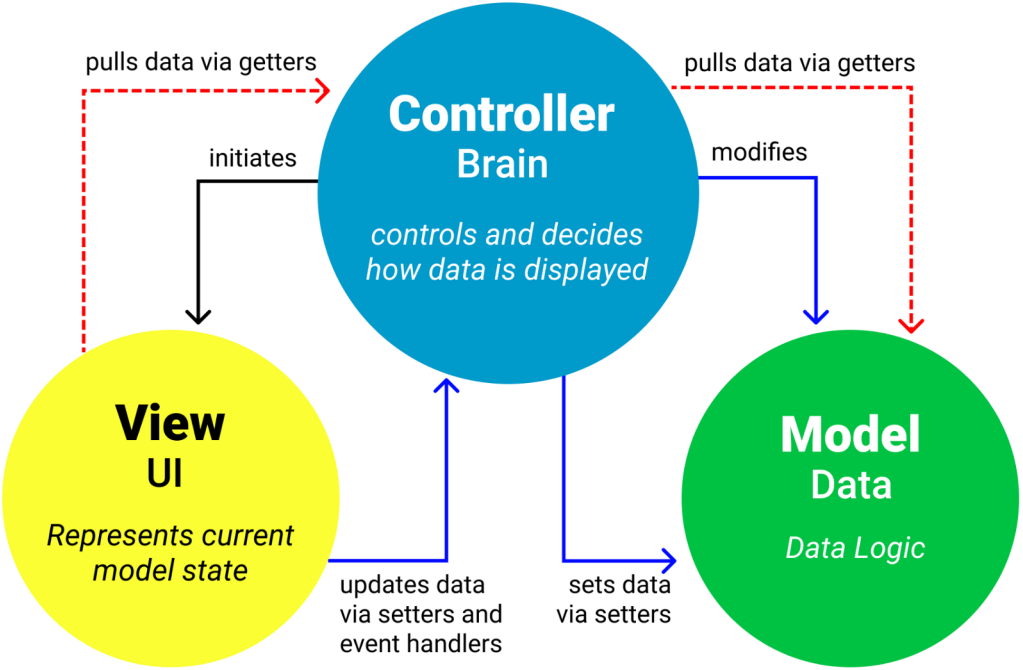
MVC Model

?

- **Model;** data structures and relations
- **View;** user interface (front-end)
- **Controller;** reacts to events

wikipedia.org

MVC Architecture Pattern



Pull requests 6 | Actions | Projects 1 | Wiki | Security | Insights

master | 20 branches | 0 tags | Go to file | Add file | Code

HappyPantss Merge pull request #54 from StanBankras/staging ... fbac353 on Apr 14, 2020 211 commits

helpers	Profile picture uploading works	2 years ago
middleware	Added authentication to routes so users must be logged in to go the...	2 years ago
public	profile pictures	2 years ago
routes	Updated gitignore	2 years ago
services	Now not matching same genders	2 years ago
views	Added required to all inputs	2 years ago
.editorconfig	Editorconfig added & linting errors fixed	2 years ago
.eslintrc	Added ESLint setup	2 years ago
.gitignore	Basic express setup	2 years ago
LICENSE	Initial commit	2 years ago
README.md	Update README.md	2 years ago
package-lock.json	api request now able to be sent and that data is being rendered on t...	2 years ago
package.json	api request now able to be sent and that data is being rendered on t...	2 years ago
server.js	Added styling to form button and fixed a chat issue	2 years ago
websockets.js	Merged	2 years ago

README.md

About

Amatch - Dating app project by Sergio Eijben, Merlijn Bergevoet & Stan Bankras

Readme
MIT License
0 stars
1 watching
1 fork

Releases

No releases published

Packages

No packages published

Contributors 3

standrd Stan
mbergevoet
HappyPantss Sergio Eijben

mongoose

elegant mongodb object modeling for node.js

read the docs | discover plugins

Star | Version 6.2.7 | Fork

Let's face it, **writing MongoDB validation, casting and business logic boilerplate is a drag**. That's why we wrote Mongoose.

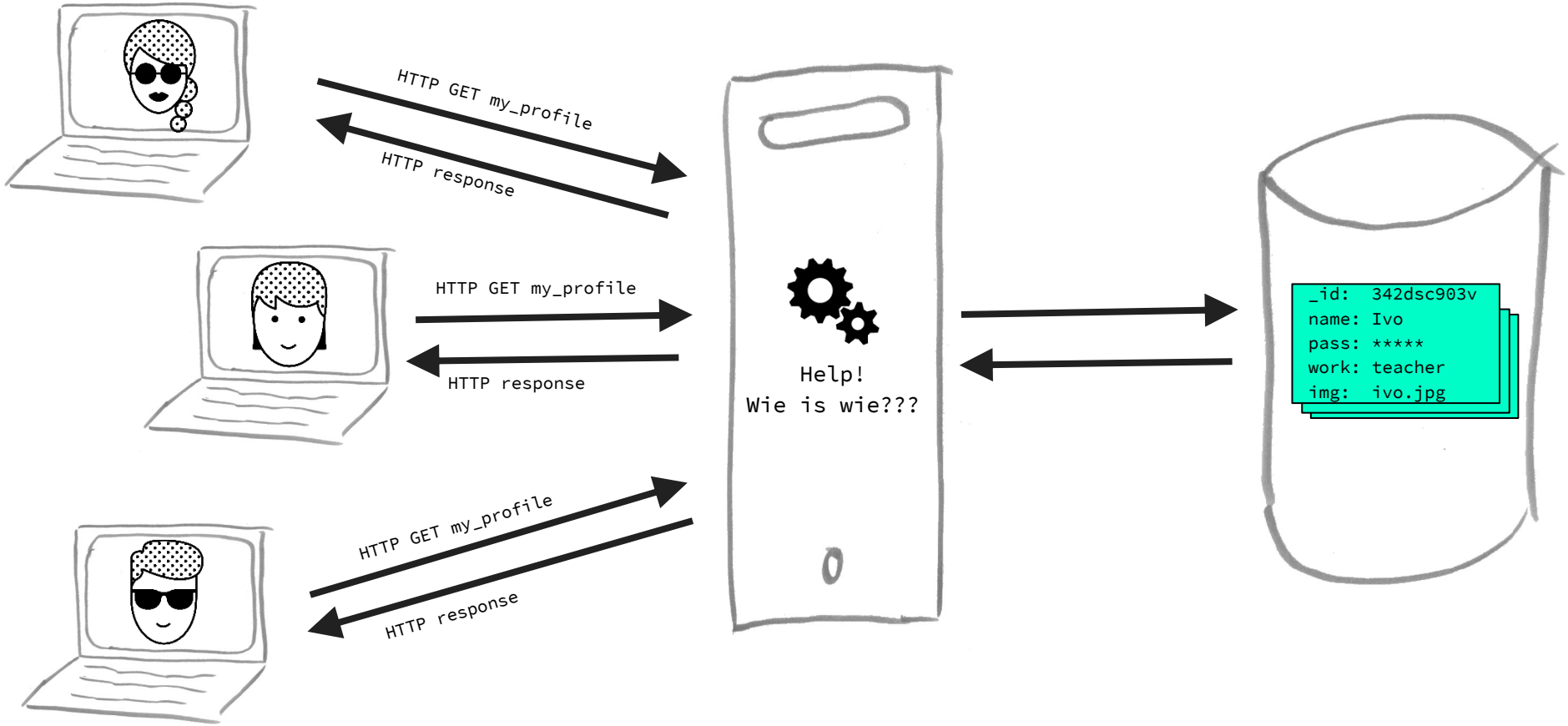
```
const mongoose = require('mongoose');
mongoose.connect('mongodb://localhost:27017/test');

const Cat = mongoose.model('Cat', { name: String });

const kitty = new Cat({ name: 'Zildjian' });
kitty.save().then(() => console.log('meow'));
```

Mongoose provides a straight-forward, schema-based solution to model your application data. It includes built-in type casting, validation, query building, business logic hooks and more, out of the box.

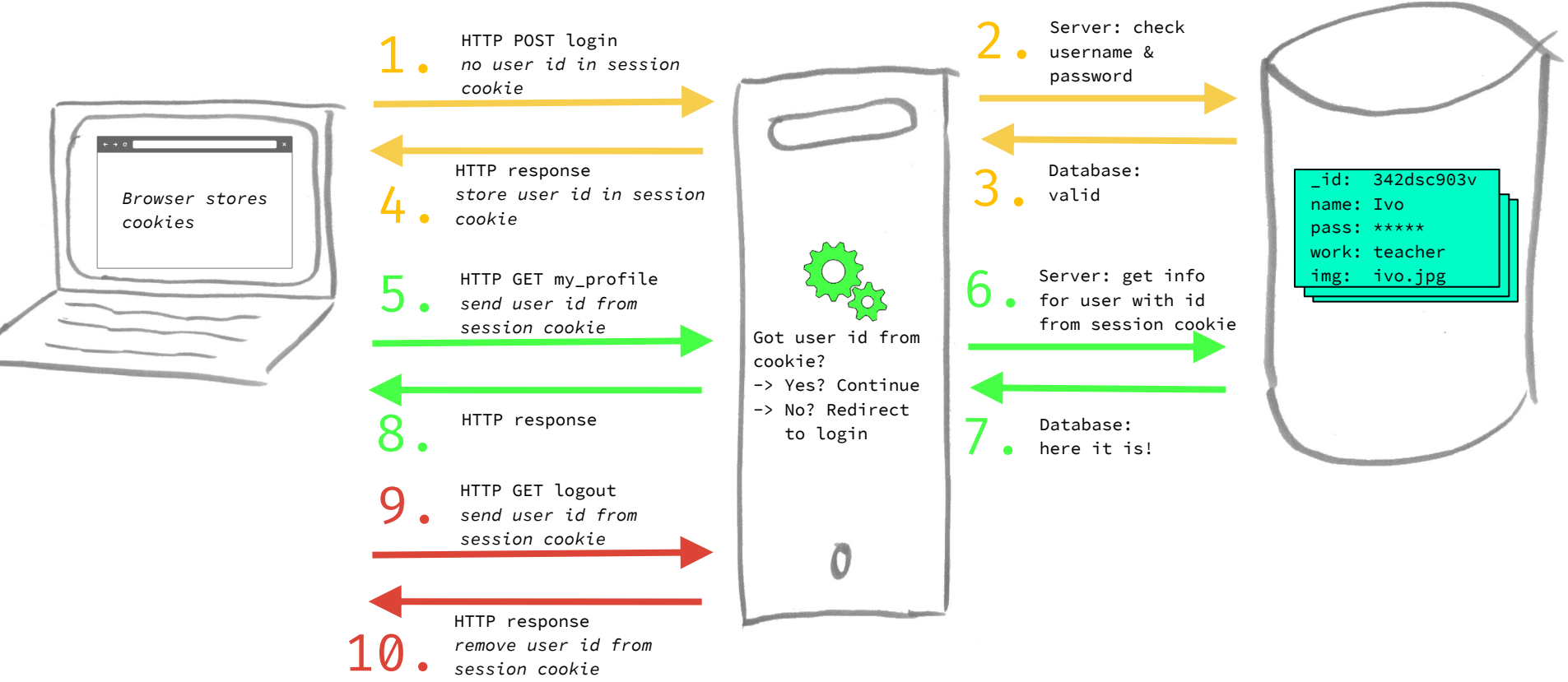
Break!

Sessions

Sessions – why?



Sessions – how?



Sessions

?

A session is **a temporary and interactive information** interchange between two or more communicating devices, or between a computer and user (see login session).

A session is established at a certain point in time, and then 'torn down' – brought to an end – at some later point.

wikipedia.org

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

Sessions

?

HTTP is stateless connection protocol, that is, the server cannot differentiate between different connections of different users. [..]

Once a client connects first time to a server, the server generates a new session ID, which later will be sent to the client as cookie value.

And from now on, **this session id will identify that client connection.**

stackoverflow

Mijn winkelwagentje

Bestel nu en je bestelling wordt gratis verzonden!

Clean Code
Robert Martin
Engelstalig - Paperback
(23)

Op voorraad. Voor 23:59 uur besteld, dinsdag in huis
+ **Select** bezorgopties

Aantal 1

€ 46,99

Verplaats naar verlanglijstje
Verwijder

Select

Neem **Select** erbij voor €9,99 p.j.
en krijg meer gemak en voordeel

Wat is Select?

Cadeaukaartcode invoeren

Totaal artikelen (1)	**€ 46,99**
Verzendkosten	**Geen**
Totaal	**€ 46,99**

Verder naar bestellen

Sessions

Examples

- Are used to check if s**omebody has logged in**
- **Show recently visited** product on webshop
- **Multi-page forms** to avoid database calls

Sessions

Examples

Local Storage	Session Storage	Cookies
The storage capacity of local storage is 5MB/10MB	The storage capacity of session storage is 5MB	The storage capacity of Cookies is 4KB
As it is not session-based, it must be deleted via javascript or manually	It's session-based and works per window or tab. This means that data is stored only for the duration of a session, i.e., until the browser (or tab) is closed	Cookies expire based on the setting and working per tab and window
The client can only read local storage	The client can only read local storage	Both clients and servers can read and write the cookies
There is no transfer of data to the server	There is no transfer of data to the server	Data transfer to the server is exist
There are fewer old browsers that support it	There are fewer old browsers that support it	It is supported by all the browser including older browser

geeksforgeeks.com

sessions

set-up

```
// Files
auth-server/
├─ node_modules/
├─ static/
│  ├─ index.css
│  ├─ index.js
│  └─ upload/
├─ view/
│  ├─ add.ejs
│  ├─ detail.ejs
│  ├─ head.ejs
│  ├─ list.ejs
│  ├─ log-in.ejs
│  ├─ not-found.ejs
│  ├─ sign-up.ejs
│  └─ tail.ejs
├─ .env
├─ .gitignore
├─ index.js
└─ package.json
```

bash

```
$ npm install express-session

+ express-session@1.15.6
added 5 packages in 1.749s

$
```

sessions set-up

```
// Files
auth-server/
├─ node_modules/
├─ static/
│  ├─ index.css
│  ├─ index.js
│  └─ upload/
├─ view/
│  ├─ add.ejs
│  ├─ detail.ejs
│  ├─ head.ejs
│  ├─ list.ejs
│  ├─ log-in.ejs
│  ├─ not-found.ejs
│  ├─ sign-up.ejs
│  └─ tail.ejs
├─ .env
├─ .gitignore
├─ index.js
└─ package.json
```

express-session handles sessions through cookies

bash

```
$ npm install express-session

+ express-session@1.15.6
added 5 packages in 1.749s

$
```

sessions set-up

```
// Files
auth-server/
├─ node_modules/
├─ static/
│  ├─ index.css
│  ├─ index.js
│  └─ upload/
├─ view/
│  ├─ add.ejs
│  ├─ detail.ejs
│  ├─ head.ejs
│  ├─ list.ejs
│  ├─ log-in.ejs
│  ├─ not-found.ejs
│  ├─ sign-up.ejs
│  └─ tail.ejs
├─ .env
├─ .gitignore
├─ index.js
└─ package.json
```

.env

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

index.js

```
…
var session = require('express-session')

…

express()
  …

  .use(session({
    resave: false,
    saveUninitialized: true,
    secret: process.env.SESSION_SECRET
  }))
  …
  .listen(8000)
```

Developer Tools - http://localhost:8000/

Elements | Console | Sources | Network | Performance | Memory | Application | Security | Audits | ChromeLens | aXe

View: | Group by frame | Preserve log | Disable cache | Offline | Online

Filter | Hide data URLs | All | XHR | JS | CSS | Img | Media | Font | Doc | WS | Manifest | Other

20 ms | 40 ms | 60 ms | 80 ms | 100 ms | 120 ms | 140 ms | 160 ms | 180 ms | 200 ms

Name

- index.css
- index.js
- localhost

Headers | Preview | Response | Cookies | Timing

General

Request URL: http://localhost:8000/
Request Method: GET
Status Code: 200 OK
Remote Address: [::1]:8000
Referrer Policy: no-referrer-when-downgrade

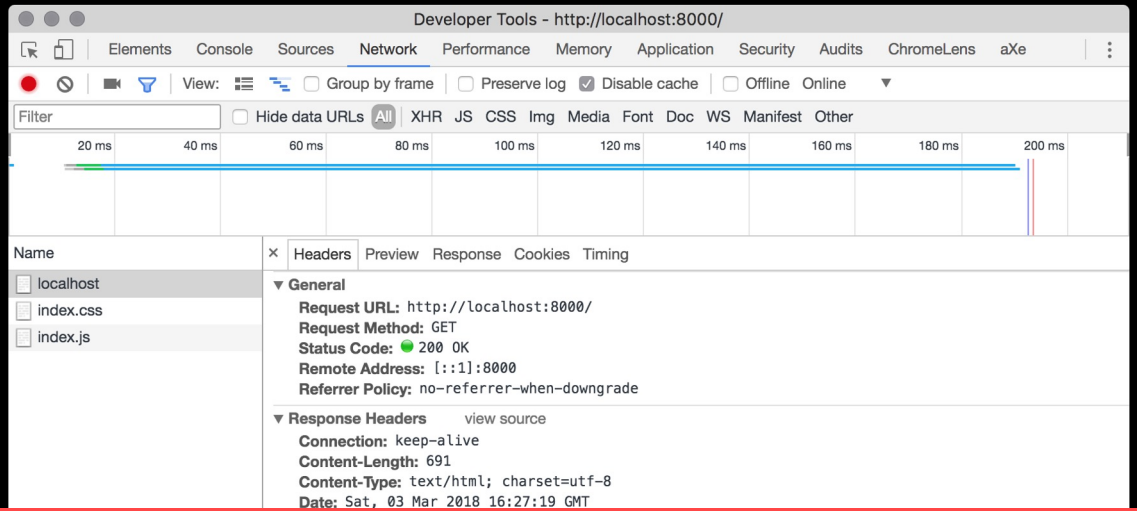
Response Headers view source

Connection: keep-alive
Content-Length: 691
Content-Type: text/html; charset=utf-8
Date: Sat, 03 Mar 2018 16:34:24 GMT
ETag: W/"2b3-AFSSiIE4IiJ/cIAFfEtX3K5zPxg"
set-cookie: connect.sid=s%3A6CgnaCco_jYNmLQ-plPaLrn3mUocEqLV.NyvXixDalI6RdkS9ogo6vVGntocjaKRzGtv%2BAYxZ74I; Path=/; HttpOnly
X-Powered-By: Express

Request Headers view source

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en;q=0.9,en-US;q=0.8,nl;q=0.7
Cache-Control: no-cache
Connection: keep-alive
DNT: 1
Host: localhost:8000
Pragma: no-cache
Referer: http://localhost:8000/
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.186 Safari/537.36

3 requests | 2.5 KB transferred | F...



▼ **General**

Request URL: `http://localhost:8000/`
Request Method: `GET`
Status Code: ● `200 OK`
Remote Address: `[::1]:8000`
Referrer Policy: `no-referrer-when-downgrade`

▼ **Response Headers** view source

Connection: `keep-alive`
Content-Length: `691`
Content-Type: `text/html; charset=utf-8`
Date: `Sat, 03 Mar 2018 16:27:19 GMT`

Localhost uses cookies tl;dr

sluiten

view/list.ejs

```
<% include head.ejs %>
<title>Movies - My movie website</title>
<h1>Movies</h1>
<p>
  <% if (user) { %>
    Hello <%= user.username %>!
  <% } else { %>
    <a href=/log-in>Log in</a>
    or
    <a href=/sign-up>Sign up</a>
  <% } %>
</p>

…

<% if (user) { %>
    <a href=/add>Add a movie</a>
<% } %>
<% include tail.ejs %>
```

Assignment (±30m)

Research more about the MVC Model, write it down and try to explain it in your own terms.

- *Can you find real-life examples?*
- *Are there similar ‘architecture’ models based on mvc?*
- *How does MVC work in a node.js / mongodb project?*

Assignment (±30m)

Think of use cases in your team application where you could use sessions.

- *Where do you need temporary storage?*
- *Login / Register flow?*
- *Multi-page forms?*

Laatste les volgende keer…

Geen nieuwe onderwerpen, alleen teambesprekingen. Kom met vragen!

Questions?

Feedback?

exit;

see you in **lab-7**!