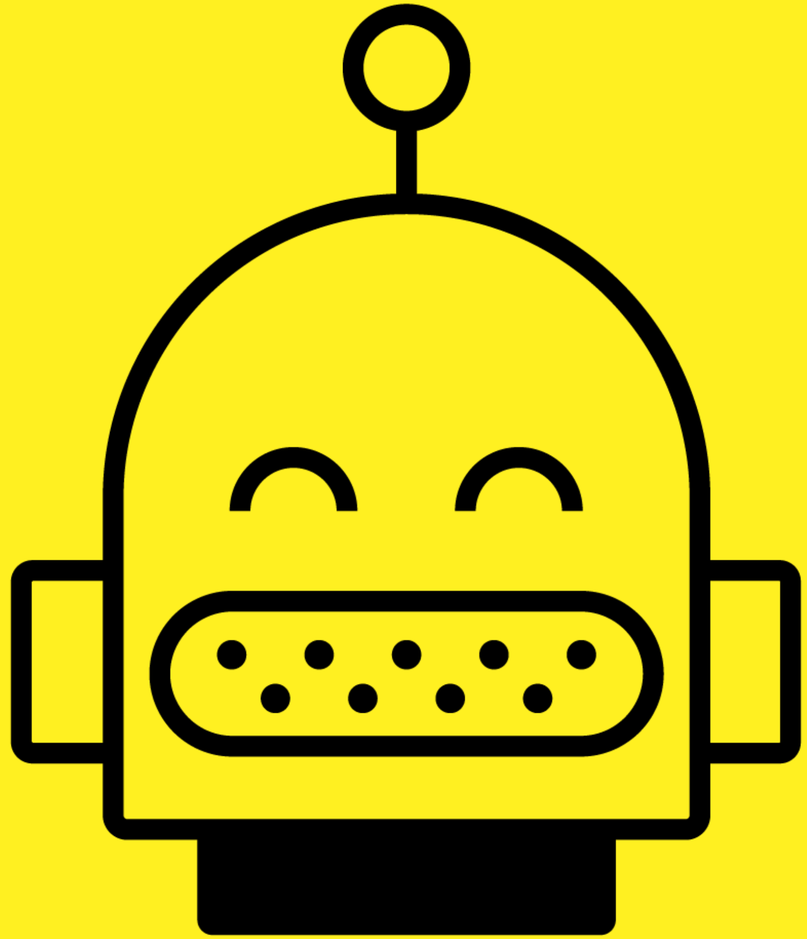
Communication and Multimedia Design | Hogeschool van Amsterdam



Project Tech

Les 3.2

Github voor teams, Formulieren verwerken in BE

Creating Tomorrow

Planning

(1)

Github voor teams

(2)

Formulieren

Theorie

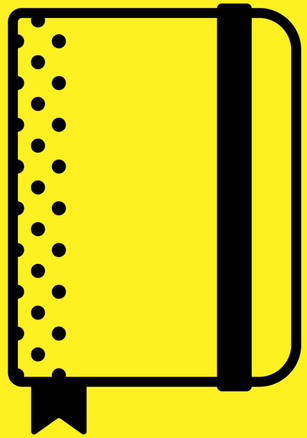
(3)

Pauze

(4)

Formulieren

Opdracht



Theorie

Git(Hub) voor teams

https://guides.github.com/introduction/flow/

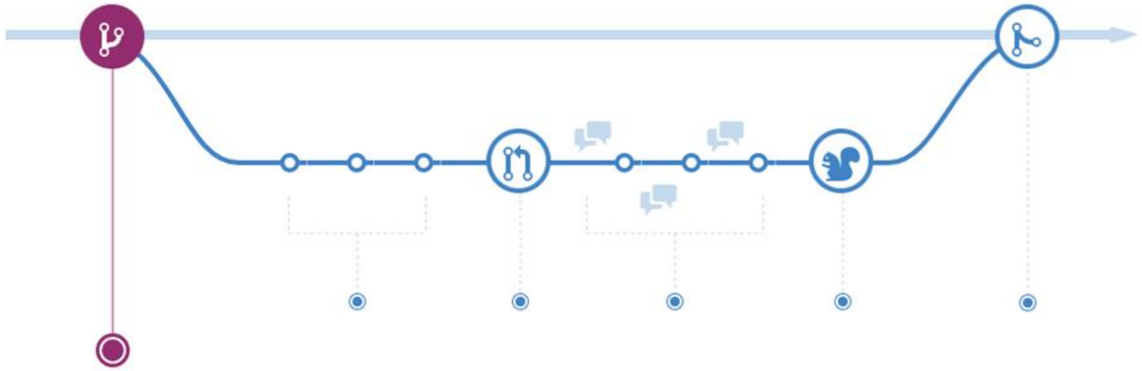
GitHub Guides

Video Guides GitHub Help GitHub.com

Understanding the GitHub flow

5 minute read Download PDF version

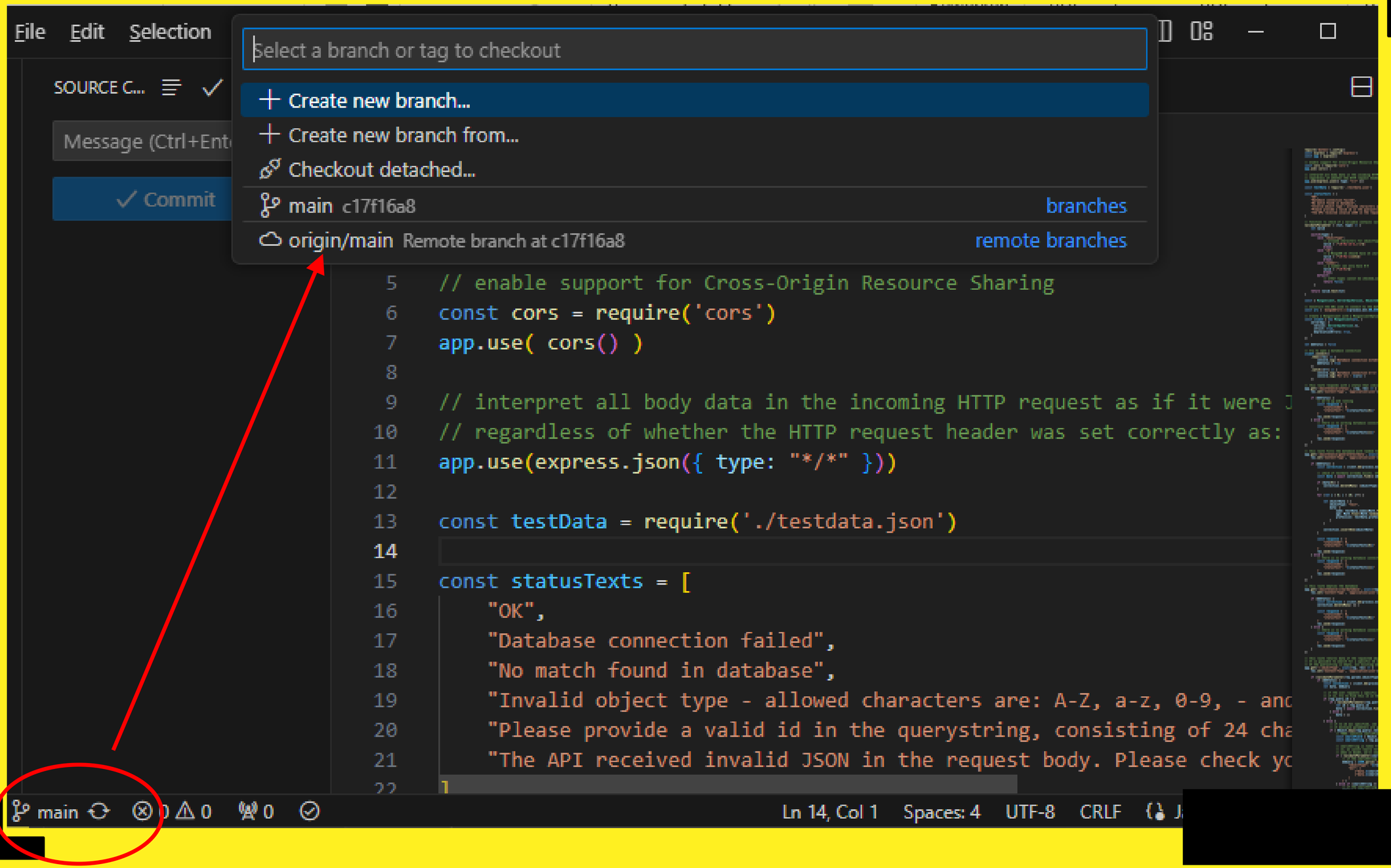
GitHub flow is a lightweight, branch-based workflow that supports teams and projects where deployments are made regularly. This guide explains how and why GitHub flow works.

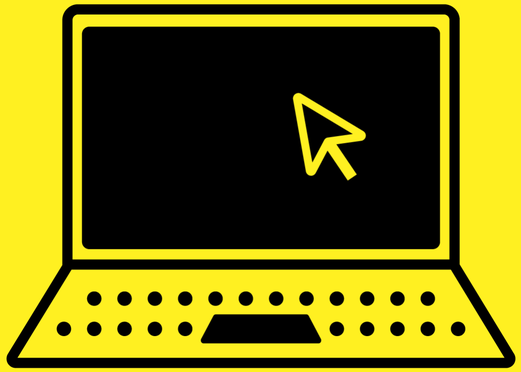


Create a branch

When you're working on a project, you're going to have a bunch of

Branches

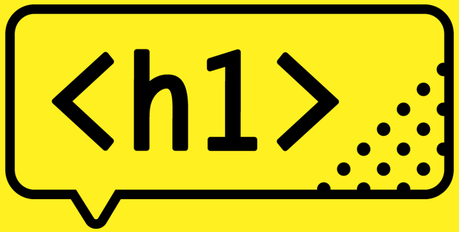

File
Edit
Selection
SOURCE C...
Message (Ctrl+Ent
Commit
Select a branch or tag to checkout
Create new branch...
Create new branch from...
Checkout detached...
main c17f16a8
branches
origin/main Remote branch at c17f16a8
remote branches
// enable support for Cross-Origin Resource Sharing
const cors = require('cors')
app.use(cors())
// interpret all body data in the incoming HTTP request as if it were
// regardless of whether the HTTP request header was set correctly as:
app.use(express.json({ type: "*/*" }))
const testData = require('./testdata.json')
const statusTexts = [
"OK",
"Database connection failed",
"No match found in database",
"Invalid object type - allowed characters are: A-Z, a-z, 0-9, - and
"Please provide a valid id in the querystring, consisting of 24 cha
"The API received invalid JSON in the request body. Please check yo
main
0
0
Ln 14, Col 1
Spaces: 4
UTF-8
CRLF



Live demo

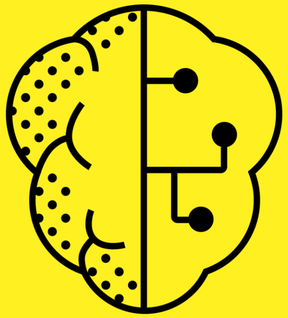
Branches & merging

in VS Code



Lesopdracht

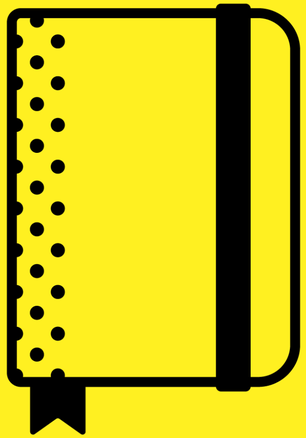
Reviewing Pull Requests & Managing Merge Conflicts



Github video’s & oefeningen

- Github flow and branches
- Branch protection
- Projects and issues
- Samenwerken op Github

Links in lesplanning

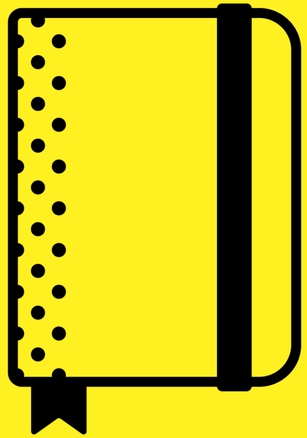


Recap theorie

Routing & templating

Laat je server code zien

Welke vragen zijn er nog?



Theorie

Posting Forms


localhost:8000/add
Title
Plot
Description
Add Movie

views/add.ejs

```ejs
<%- include('head.ejs') %>

<h1>Add a new movie</h1>
<form action="/add-movie" method="POST">
  <label>Title</label>
  <input type="text" name="title">

  <label>Plot</label>
  <input type="text" name="plot">

  <label>Description</label>
  <textarea name="description"></textarea>

  <input type="submit" value="Add Movie">
</form>
```

Send a HTTP **POST** request to the URL specified in the **ACTION**...

...when the form is submitted

server.js

```
const express = require('express')
const app = express()

app.use(express.urlencoded({extended: true}))

app.get('/add', showAddForm)
app.post('/add-movie', addMovie)

function showAddForm(req, res) {
  res.render('add.ejs')
}

function addMovie(req, res) {
  res.send(`Thanks for adding the movie with:
      title: ${req.body.title},
      plot: ${req.body.plot},
      and description: ${req.body.description}
  `)
}
```

Middleware: parses form data

Route to handle the **post** request to **/add-movie**

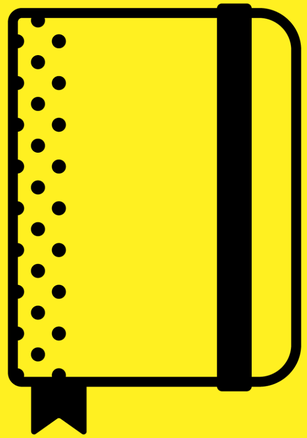
Parsed form data is stored in **req.body**

The properties of **req.body** match the **name** attributes of the inputs in the form

Opdracht: formulier posten

Breid je `node.js` server uit met code om gepost formulier te verwerken. Je kunt het inlogformulier uit de vorige les gebruiken, maar haal dan eerst de front-end JavaScript er uit. We gaan deze functionaliteit nu immers in de back-end bouwen!

1. Maak een route en een view om je formulier te tonen
2. Geef het formulier een action en een POST-method
3. Maak een route om de POST request af te handelen als het formulier wordt verstuurd
4. Maak een view om een HTTP response op de POST request te sturen. Laat in deze view de ontvangen formulier data zien.



Theorie

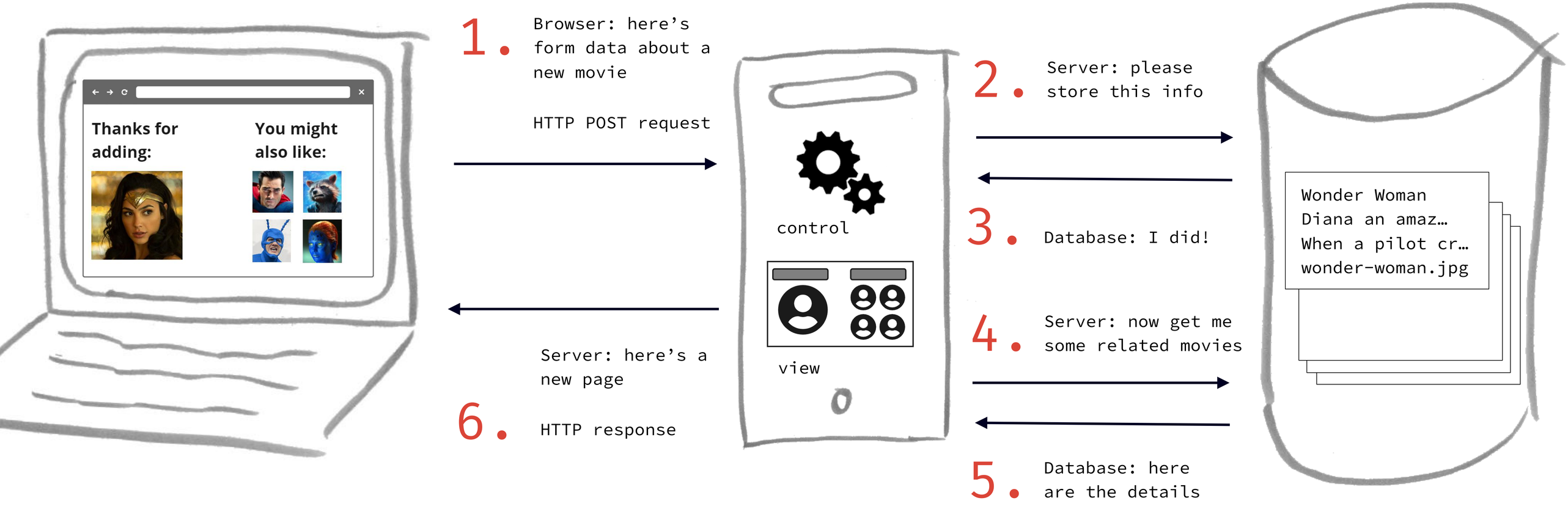
Database

Bronnenlijst

Client
(browser)

Server
(Webserver met Node.js)

Database
(MongoDB)



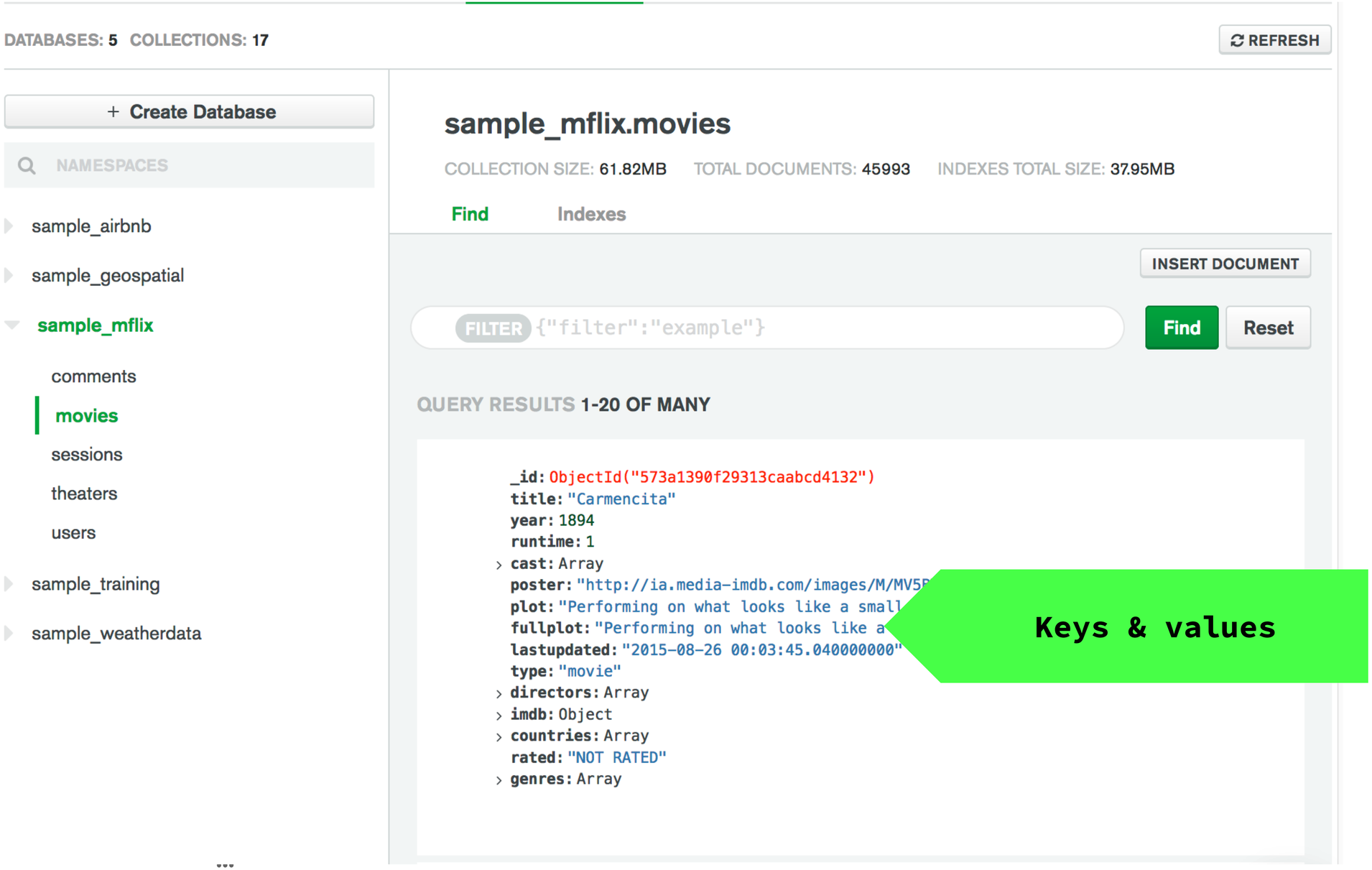
MongoDB

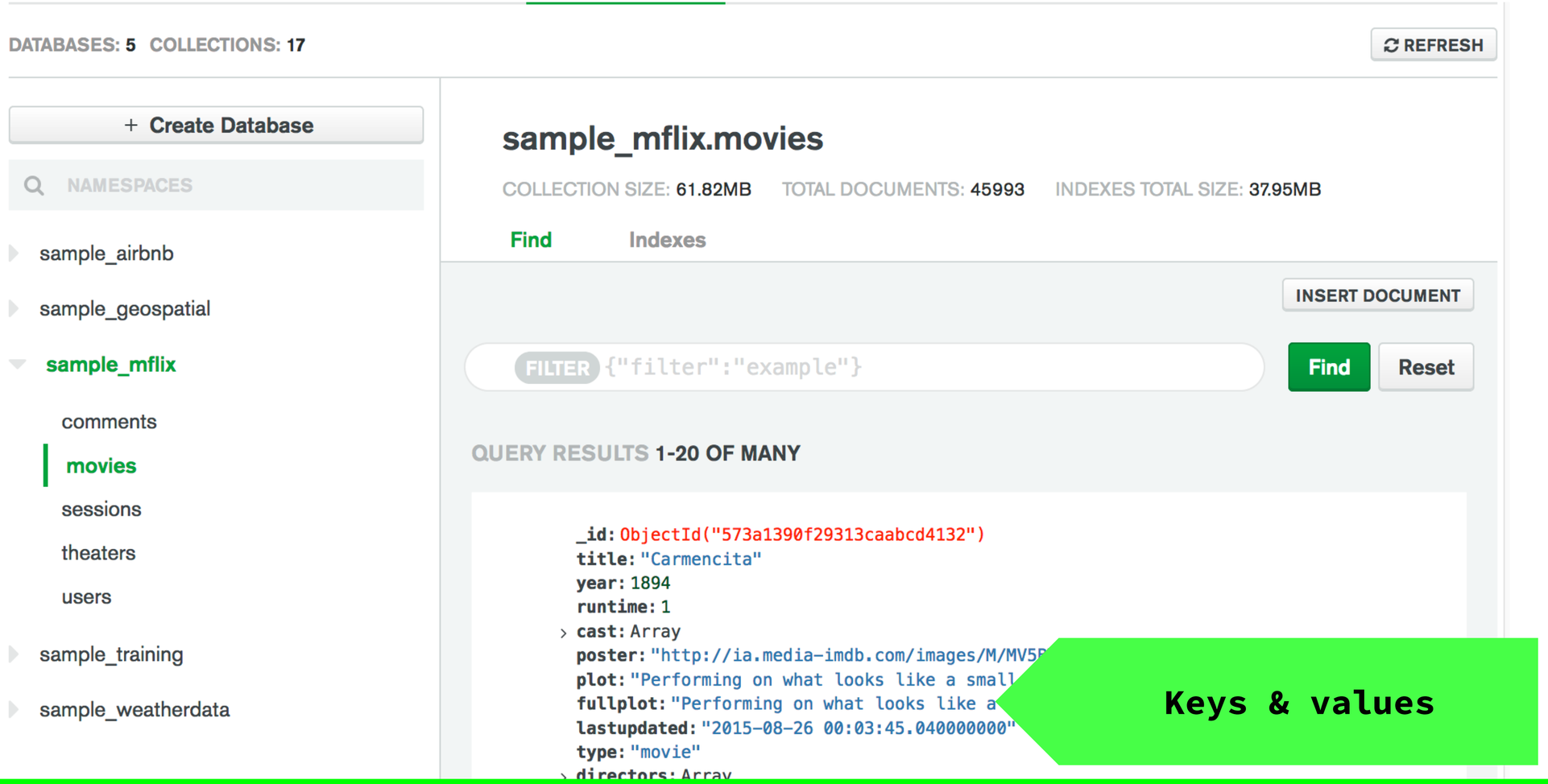
MongoDB (from humongous) is a free and **open-source** cross-platform document-oriented database program. Classified as a **NoSQL** database program, MongoDB uses **JSON-like documents** with schemas. MongoDB is developed by MongoDB Inc. […]

(wikipedia.org)

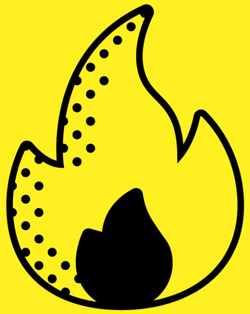
Waarom MongoDB?

- Het werkt goed samen met JavaScript / Node.js
- Je kunt je data flexibel structureren op een JSON-achtige manier, zonder SQL of formeel database ontwerp te hoeven leren
- Het is gratis ;)


DATABASES: 5 COLLECTIONS: 17
REFRESH
+ Create Database
NAMESPACES
sample_airbnb
sample_geospatial
sample_mflix
comments
movies
sessions
theaters
users
sample_training
sample_weatherdata
sample_mflix.movies
COLLECTION SIZE: 61.82MB
TOTAL DOCUMENTS: 45993
INDEXES TOTAL SIZE: 37.95MB
Find
Indexes
INSERT DOCUMENT
FILTER {"filter":"example"}
Find
Reset
QUERY RESULTS 1-20 OF MANY
_id: ObjectId("573a1390f29313caabcd4132")
title: "Carmencita"
year: 1894
runtime: 1
cast: Array
poster: "http://ia.media-imdb.com/images/M/MV5
plot: "Performing on what looks like a small
fullplot: "Performing on what looks like a
lastupdated: "2015-08-26 00:03:45.040000000"
type: "movie"
directors: Array
imdb: Object
countries: Array
rated: "NOT RATED"
genres: Array
Keys & values



Think about how you want to structure you data.
This is called **data modelling.**



Maar eerst: .env

- Je inloggegevens voor de database wil je niet in je code zetten
- Als ze op GitHub staan, kan de hele wereld bij je database (en dat gaan ze doen ook)
- Zelfs als ze alleen in een oude commit staan (dat was het fijne van Git)
- Daarom zetten we dit soort gegeven in een apart bestand, genaamd: `.env`
- En zetten `.env` in onze `.gitignore` (!!)
- Ook handig als er iets verandert: een nieuw database account, deployen naar een nieuwe webserver etc. Je hoeft dit dan maar op 1 plek aan te passen.

.env

```
DB_HOST=dbhost.somewhere.com
DB_NAME=mydatabase
DB_USERNAME=myusername
DB_PASSWORD=mypassword
DB_COLLECTION=mycollection
```

We gebruiken meestal hoofdletters

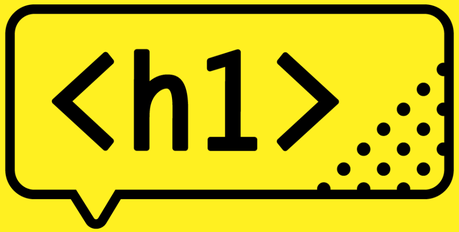
Geen “ ” (.env is geen JavaScript)

.gitignore

```
node_modules/
.DS_Store
.env
```

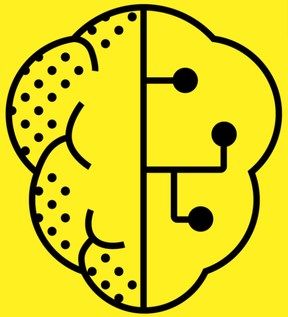
Huiswerkopdracht: database

1. Maak een eigen database account aan op `mongodb.com` volgens de instructies op onze GitHub
2. Maak een `.env` file met je eigen inloggegevens
3. En zet je `.env` in je `.gitignore`
4. Installeer met `npm install` de modules `dotenv` en `mongodb`



Lesopdracht

Formulier verwerken



Toelichting lesopdracht

Zie lesplanning op Github

Einde van de les

Tot de volgende keer!