

Programmeren 6: Fullstack Webdevelopment (React & Node.js)

Knowledgebase: <https://luukftf.github.io/knowledgebase>

(code: <https://github.com/LuukFTF/knowledgebase>)

By: Lucas van der Vegt

Leerdoelen

Index

- Programmeren 6: Fullstack Webdevelopment (React & Node.js)
 - Leerdoelen
 - Index
 - A. backend - Nodejs & Express & MongoDB
 - NPM Packages
 - Installing
 - .ENV
 - Endpoint
 - Middleware
 - Models
 - Checker
 - B. HTTP, RESTfull API & OAuth
 - HTTP
 - Software
 - Methods
 - URIs
 - Representatieformaten
 - JSON
 - XML
 - YAML
 - Request
 - Response
 - Basic Networking
 - IP/TCP & OSI model
 - Statuscodes
 - 2XX good
 - 3XX recoverable error
 - 4XX client error
 - 5XX server error
 - RESTfull API
 - API Documentation
 - HATEOAS (Linking)
 - HAL
 - link relation types
 - Pagination
 - Response Categories
 - Type RESTFULL Resources
 - Queries
 - OAuth
 - C. operations - VPS & Linux
 - Virtual Private Server (VPS)
 - Basic Networking / VPS commands
 - Linux
 - Basic BASH commands

- Backend MERN
 - D. frontend - React
 - E. frontend - Sass
 - Links
-

A. backend - Nodejs & Express & MongoDB

<https://www.youtube.com/watch?v=ENrzd9HAZK4>

NPM Packages

Express Mongoose Nodemon Dotenv

Installing

installing needed software (windows)

```
winget install npm # install npm
winget install nodejs # install nodejs
winget install mongodb # install mongodb

npm -v # check if npm is correctly installed
nodejs -v # check if nodejs is correctly installed
mongodb -v # check if mongodb is correctly installed
```

new project

```
git clone # clone corresponding git repo

npm install express mongoose dotenv # add express & mongoose dependency to project
npm install --save-dev nodemon # add express & mongoose dependency to project
development

npm i # install repo packages
```

.ENV

Endpoint

Middleware

Models

Database

Mongoose Schema

Checker

B. HTTP, RESTfull API & OAuth

<https://www.youtube.com/watch?v=-MTSQjw5DrM>

API < an Application Programming Interface is an interface that defines interactions between multiple software applications or mixed hardware-software intermediaries. >

idempotency < Idempotence is the property of certain operations in mathematics and computer science whereby they can be applied multiple times without changing the result beyond the initial application. >

Safe Method < Deze method veranderd niks op de server >

HTTP

Hypertext Transfer Protocol

Uniform Interface

stateless

Cacheable

Software

Postman Insomnia

VScode extension: REST Client

Methods

name	function	safe	idempotent
POST	create		x
GET	read	x	x
PUT	create / update (geheel)		x
PATCH	update (deel)		
DELETE	destroy		x
OPTIONS	get possible methods	x	x
HEAD	get without body	x	x
TRACE			
CONNECT			

POST overloading < sending HTTP requests that doesnt exist can be done with a POST request. You send a method with the head that specifies what custom request you are using (document this correctly)>

Custom Methods < another way to use create new request methods, is to use custom http methods, just exchange *post* for something else like *undelete* (document this correctly)>

URIs

< *Uniform Resource Identifier* > <https://www.slideshare.net/landlessness/teach-a-dog-to-rest>

protocol://userinfo@subdomain.domain.tld:port/path?query#fragment

<https://api.com/v2/comet>

network_location/resource

structuur / hierarchie (pad) nevenschikking (😊 leesbaarheid (- _) geen extensies (of extensies over de inhoud, liever application/json /xml (.json .xml))

routing verbergt techniek

Representatieformaten

Mensen: html

Mensen & Machines: html met microformats

Machines: json, xml, yaml, rss

JSON

```
{
  "widget": {
    "debug": "on",
    "window": {
      "title": "Sample Konfabulator Widget",
      "name": "main_window",
      "width": 500,
      "height": 500
    },
    "image": {
      "src": "Images/Sun.png",
      "name": "sun1",
      "hOffset": 250,
      "vOffset": 250,
      "alignment": "center"
    },
    "text": {
      "data": "Click Here",
      "size": 36,
      "style": "bold",
      "name": "text1",
      "hOffset": 250,
      "vOffset": 100,

```

```

        "alignment": "center",
        "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
    }
}
}

```

XML

```

<widget>
  <debug>on</debug>
  <window title="Sample Konfabulator Widget">
    <name>main_window</name>
    <width>500</width>
    <height>500</height>
  </window>
  <image src="Images/Sun.png" name="sun1">
    <hOffset>250</hOffset>
    <vOffset>250</vOffset>
    <alignment>center</alignment>
  </image>
  <text data="Click Here" size="36" style="bold">
    <name>text1</name>
    <hOffset>250</hOffset>
    <vOffset>100</vOffset>
    <alignment>center</alignment>
    <onMouseUp>
      sun1.opacity = (sun1.opacity / 100) * 90;
    </onMouseUp>
  </text>
</widget>

```

YAML

```

widget:
  debug: 'on'
  window:
    title: Sample Konfabulator Widget
    name: main_window
    width: 500
    height: 500
  image:
    src: Images/Sun.png
    name: sun1
    hOffset: 250
    vOffset: 250
    alignment: center
  text:
    data: Click Here

```



```
size: 36
style: bold
name: text1
hOffset: 250
vOffset: 100
alignment: center
onMouseUp: sun1.opacity = (sun1.opacity / 100) * 90;
```

Request

```
POST https://api.com/v2/comet HTTP/1.1
Accept: application/json
Authorization: <token>
Connection: keep-alive

{
  "body": "body"
}
```

VERB / resource uri / protocol

Response

```
HTTP/1.1 200 OK
Age: 2323
Connection: keep-alive

{
  "id": "2"
  "status": "3"
}
```

protocol / statuscode

Basic Networking

IP/TCP & OSI model

Packets (information)

7 layers

L7 - Application {L6 - Presentation} {L5 - Session}

L4 - Transport - protocol / ports (TCP & UDP + https:443, http:80, ssh:22, ftp) L3 - Network - ip adressen L2 - Data Link - mac adressen L1 - Physical - ethernet ports

Statuscodes

Uniform Interface kan errors afhandelen

2XX good

200 - OK 201 - Created

3XX recoverable error

302 - Found (redirect) 304 - Not Modified

4XX client error

401 - Unauthorized 403 - Forbidden 404 - Not found

5XX server error

500 - Internal Server Error 503 - Service Unavailable

RESTfull API

RESTful < *Representational State Transfer, invented by Roy Fielding in 2000* >

JSON

API Documentation

OpenAPI Specification <https://swagger.io/specification/>

Postman

HATEOAS (Linking)

< *Hypermedia as the Engine of Application State* > <https://en.wikipedia.org/wiki/HATEOAS>

HAL

https://en.wikipedia.org/wiki/Hypertext_Application_Language

```
{
  "_links": {
    "self": { "href" : "http://api....." },
    "collection": { "href" : "http://api....." }
  }
}
```

link relation types

self collection alternate edit related previous & next first & last

Pagination

start (begin bij 1) limit (aantal)

```
GET /items?start=6&limit=5
Accept: application/json
```

pagina 6 tot en met 10

```
{
  "pagination": {
    "currentPage": 2,
    "currentItems": 5,
    "totalPages": 2,
    "totalItems": 10,
    "links": {
      "first": {
        "page": 1,
        "href": "/items?start=1&limit=5"
      },
      "last": {
        "page": 2,
        "href": "/items?start=6&limit=5"
      },
      "previous": {
        "page": 1,
        "href": "/items?start=1&limit=5"
      },
      "next": {
        "page": 2,
        "href": "/items?start=6&limit=5"
      },
    },
  },
}
```

Response Categories

items links pagination

```
GET /items/
Accept: application/json
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "items": [
    {
      "id": "200",
      "title": "test",
      "links": {
        "self": "/items/200"
      }
    },
    {
      "id": "201",
      "title": "test",
      "links": {
        "self": "/items/201"
      }
    },
  ],
  "links": {
    "self": "/items/"
  },
  "pagination": {
    "currentPage": 1,
    "currentItems": 4,
    "totalPages": 1,
    "totalItems": 4,
    "links": {
      "first": {
        "page": 1,
        "href": "/items/"
      },
      "last": {
        "page": 1,
        "href": "/items/"
      },
      "previous": {
        "page": 1,
        "href": "/items/"
      },
      "next": {
        "page": 1,
        "href": "/items/"
      }
    }
  }
}
```

Type RESTFULL Resources

REST Resource < *Het type resource wat in de body teruggestuurd word* >

Detail < *all data of specific item* >

```
GET https://pokeapi.co/api/v2/pokemon/4
Accept: application/json
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 4,
  "name": "charmander",
  "base_experience": 62,
  "height": 6,
  "weight": 85,
  "location_area_encounters": "https://pokeapi.co/api/v2/pokemon/4/encounters",
  "stats": [
    {
      "base_stat": 39,
      "stat": {
        "name": "hp",
        "url": "https://pokeapi.co/api/v2/stat/1/"
      }
    },
    {
      "base_stat": 52,
      "stat": {
        "name": "attack",
        "url": "https://pokeapi.co/api/v2/stat/2/"
      }
    }
  ],
}
```

GET, PUT/PATCH, DELTE, OPTIONS

Collection < *list of items with indexable information* >

```
GET https://pokeapi.co/api/v2/pokemon
Accept: application/json
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "results": [
```

```
{
  "name": "bulbasaur",
  "url": "https://pokeapi.co/api/v2/pokemon/1/"
},
{
  "name": "ivysaur",
  "url": "https://pokeapi.co/api/v2/pokemon/2/"
},
{
  "name": "venusaur",
  "url": "https://pokeapi.co/api/v2/pokemon/3/"
},
{
  "name": "charmander",
  "url": "https://pokeapi.co/api/v2/pokemon/4/"
},
{
  "name": "charmeleon",
  "url": "https://pokeapi.co/api/v2/pokemon/5/"
},
{
  "name": "charizard",
  "url": "https://pokeapi.co/api/v2/pokemon/6/"
},
{
  "name": "squirtle",
  "url": "https://pokeapi.co/api/v2/pokemon/7/"
}
]
"pagination": {
  "currentPage": 1,
  "currentItems": 7,
  "totalPages": 160,
  "count": 1118,
}
}
```

GET, POST, OPTIONS

Composition < a combination of different types of resources >

```
GET https://pokeapi.co/api/v2/location/2/
Accept: application/json
```

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "id": 2,
```

```

    "name": "eterna-city",
    "region": {
      "name": "sinnoh",
      "url": "https://pokeapi.co/api/v2/region/4/"
    }
    "areas": [
      {
        "name": "eterna-city-area",
        "url": "https://pokeapi.co/api/v2/location-area/2/"
      },
      {
        "name": "eterna-city-west-gate",
        "url": "https://pokeapi.co/api/v2/location-area/788/"
      }
    ],
    "game_indices": [
      {
        "game_index": 9,
        "generation": {
          "name": "generation-iv",
          "url": "https://pokeapi.co/api/v2/generation/4/"
        }
      }
    ],
  ],
}

```

GET, OPTIONS, (PUT/PATCH)

Function < *functional resource, custom input generates custom output* */distance/rdam;adam* >

```

GET https://pokeapi.co/api/v2/battle/4;6
Accept: application/json

```

```
// calcutation who wins the battle with stats and winner
```

GET, OPTIONS

Controller < *special function* >

```

GET https://pokeapi.co/api/v2/hit
Accept: application/json

```

```
// calcutation if a pokemon is still alive after a hit, amount left & stats
```

POST, OPTIONS

Queries

Filter

/pokemon?type=fire

OAuth

C. operations - VPS & Linux

Virtual Private Server (VPS)

Basic Networking / VPS commands

```
ssh username@ip
```

Linux

<https://cheatography.com/davechild/cheat-sheets/linux-command-line/>

Basic BASH commands

```
pwd # Show current directory
mkdir [dir] # Make directory
cd [dir] # Change directory to dir
cd .. # Go up a directory
ls # List files
-a # Show all (including hidden)
-t # Sort by last modified
-S # Sort by file size
cp [dir/name] [newdir/name] # copy
mv [dir/name] [newdir/name] # move
rm [dir/name] # remove
touch [name] # new file

ps # process snapshot
kill [pid] # kill process
uptime # Show uptime
uname -a # Show system and kernel
whoami # Show your username
help / man # manuals and information
clear # clear screen

CTRL-C # Stop dcurent running command
```

Backend MERN

installing needed software (linux)

```
sudo apt update # update apt packages
sudo apt install npm # install npm
sudo apt install nodejs # install nodejs
sudo apt install mongodb # install mongoddb

npm -v # check if npm is correctly installed
nodejs -v # check if nodejs is correctly installed
mongodb -v # check if mongodb is correctly installed
```

installing on server / locally

```
git clone / pull [repo] # clone or pull repository

cd [dir] # change directory to repo dir

npm i # install repo packages
node . # start node index.js

sudo systemctl start mongodb # start mongodb server

sudo systemctl status mongodb # check status of mongodb server
```

configuration

```
mongo --eval 'db.runCommand({ connectionStatus: 1 })' # diagnostic mongo command

sudo systemctl stop mongodb # stop mongodb server
sudo systemctl restart mongodb # restart mongodb server

sudo ufw status # check firewall status

sudo nano /etc/mongodb.conf # edit mongodb config
```

D. frontend - React

E. frontend - Sass

styleguides <http://styleguides.io/>

<https://web.archive.org/web/20170523012226/http://codepen.io/guide/#one>

Links

<https://www.youtube.com/watch?v=fgTGADljAeg>

end of file

publish date: 0000-00-00

modified date: 0000-00-00