

Opleverdocumentatie

Inleiding

Vanuit de Hogeschool van Arnhem en Nijmegen (HAN) zijn er multidisciplinaire projectteams samengesteld om het probleem van Burgers' Zoo aan te pakken. Een eerder team heeft acht weken gewerkt aan een mogelijke oplossing. Daarbij zijn er onderzoeken gedaan, concepten ontworpen, prototypes uitgewerkt en getest. De resultaten hiervan zijn in een adviesrapport verwerkt en overhandigd aan een tweede projectteam. Dat projectteam heeft de focus gelegd op de technische haalbaarheid van de ontworpen concepten en dit ook op grotere schaal getest. Dit tweede team heeft de bevindingen hiervan verwerkt in dit document om te overhandigen aan een volgende projectgroep.

Burgers' Zoo heeft vanuit eigen onderzoek aangegeven dat bezoekers zich negatief uiten over de mobiliteit in het park en daardoor hun bezoek als minder prettig ervaren. Het bezoek wordt gezien als een dagtaak wat ook zwaar is door de lange afstanden lopen op heuvelig terrein. Hierdoor zijn bezoekers van het park minder snel geneigd om het park opnieuw te bezoeken. Burgers' Zoo heeft dit probleem voorgelegd aan de HAN, waarna meerdere projectteams hier mee aan de slag zijn gegaan.

In dit rapport wordt allereerst het concept beschreven, gevolgd door een beschrijving en uitleg van de technieken die gebruikt zijn. Daarbij is een handleiding opgenomen om een demo te starten. Daarna wordt er beschreven hoe er verder gewerkt kan worden met de software en hardware die opgeleverd is. Tot slot zijn er aanbevelingen voor een volgende projectteam. Achter aan het verslag is nog een lijst te vinden met installatieinstructies voor de genoemde software producten.

Inhoudsopgave

- [Systeemoverview \(hoe werkt dit systeem globaal?\)](#)
- [Opzethandleiding \(hoe start ik het systeem?\)](#)
- [Het starten van de applicaties](#)
 - [randvoorwaarden](#)
 - [Het starten van de web applicaties en de database](#)
 - [Development applicaties](#)
 - [productie builds \(minified\)](#)
 - [De applicaties bezoeken](#)
- [Front-end](#)
 - [Redux](#)
 - [Folderindeling \(package structuur\)](#)
- [Installatie Gateway](#)
 - [MongoDB](#)
 - [Java](#)
 - [RXTX](#)

- [Aansluiting Arduino](#)
- [Start gateway](#)
 - [Poot](#)
- [Repo commando's](#)
- [Ontwikkelhandleiding \(hoe ontwikkel ik?\)](#)
- [Bijlagen](#)
- [Begrippenlijst](#)
- [Software lijst](#)

Systeemoverview (hoe werkt dit systeem globaal?)

Opzethandleiding (hoe start ik het systeem?)

Het starten van de applicaties

Dit hoofdstuk zal beschrijven hoe alle webapplicaties, de backend en de database opgestart moeten worden. Ook zal dit hoofdstuk beschrijven hoe de database gevuld kan worden met het seedsript zodat er wat testdata in de apps staat.

randvoorwaarden

Om alle applicaties te draaien moeten er een aantal dingen geregeld worden op de pc/laptop. De tabel hieronder geeft aan welke stukken software benodigd zijn en zal, waar mogelijk, een link worden geven naar de officiële website.

- [mongo \(https://www.mongodb.com/\)](https://www.mongodb.com/)
- [docker \(https://www.docker.com/\)](https://www.docker.com/)
- [docker-compose \(https://docs.docker.com/compose/\)](https://docs.docker.com/compose/)
- [node \(https://nodejs.org/\)](https://nodejs.org/)
- [npm \(https://www.npmjs.com/\)](https://www.npmjs.com/)

Als alle bovenstaande software geïnstalleerd is dan kunnen alle apps gestart worden, om de database te vullen is er echter nog een extra stukje software nodig. De software heet mongorestore en komt , ten tijde van schrijven, mee geïnstalleerd met het mongo pakket (Op Windows met [mongotools \(https://github.com/mongodb/mongo-tools\)](https://github.com/mongodb/mongo-tools)). Bekijk [deze website \(https://docs.mongodb.com/manual/reference/program/mongorestore/\)](https://docs.mongodb.com/manual/reference/program/mongorestore/) voor meer informatie.

Het starten van de web applicaties en de database

Het volgende hoofdstuk zal uitleggen hoe de applicaties gestart kunnen worden in zowel development modus als productie modus. Voor het testen is alleen de development modus meer als genoeg.

NOTE! de commando's zijn bedacht voor Linux en Mac OS X, hieronder wordt beschreven hoe het werkt voor alle drie de systemen al is het zeer aan te raden om een Linux Virtual machine op te zetten. (klik [hier \(https://www.storagecraft.com/blog/the-dead-simple-guide-to-installing-a-linux-virtual-machine-on-windows/\)](https://www.storagecraft.com/blog/the-dead-simple-guide-to-installing-a-linux-virtual-machine-on-windows/) voor uitleg)

Development applicaties

Om de applicaties in development modus te starten (in Docker) moet het volgende commando worden uitgevoerd:

```
npm start
```

Dit zal, onder water, het volgende draaien:

```
npm run copy-endpoint-dev && docker-compose down && docker-compose up
```

Het commando "copy-endpoint-dev" gaat echter **fout** op een Windows systeem omdat Windows geen fatsoenlijke copy tool op de command line heeft. Als het toch op windows moet draaien moet je de kopieerstep even zelf uitvoeren. De stappen zijn dan als volgt:

1. Kopieer het bestand ./config/dev.json naar de volgende twee locaties:
 - ./apps/admin/src/constants/external-variables/endpoints.json
 - ./apps/ranger/src/constants/external-variables/endpoints.json
2. Draai het docker-compose down commando
3. Draai het docker-compose up commando

productie builds (minified)

Om de applicaties in productie modus te starten (in Docker) moet het volgende commando worden uitgevoerd:

```
npm run build
```

Dit zal, onder water, het volgende draaien:

```
npm run copy-endpoint-prod && docker-compose -f ./dockerfiles/prod/docker-compose.yml down && docker-compose -f ./dockerfiles/prod/docker-compose.yml up && bash seedsript.sh -h servers.rickvanlieshout.com:8019
```

Vervang hier de server (servers.rickvanlieshout.com:8019) met de juiste productie server en verander ook de variabelen in ./config/prod.json naar de benodigde productie variabelen.

Het commando "copy-endpoint-dev" gaat echter **fout** op een Windows systeem omdat Windows geen fatsoenlijke copy tool op de command line heeft. Als het toch op windows moet draaien moet je de kopieerstep even zelf uitvoeren. De stappen zijn dan als volgt:

1. Kopieer het bestand ./config/prod.json naar de volgende twee locaties:
 - ./apps/admin/src/constants/external-variables/endpoints.json
 - ./apps/ranger/src/constants/external-variables/endpoints.json

2. Draai het docker-compose down commando
3. Draai het docker-compose up commando
4. Draai het bash seedsript.sh SERVERURL bestand. (op Windows heb je hier de [bash shell voor Windows \(https://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/\)](https://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/) voor nodig.)

De applicaties bezoeken

Om de applicaties te bezoeken, en ze te gebruiken, ga je naar de volgende web adressen:

Applicatie	Adres in develop modus	Adres in productie modus
Back-end api	http://localhost:8001	http://localhost:8011
Admin / educatie applicatie	http://localhost:8002	http://localhost:8012
Ranger applicatie	http://localhost:8003	http://localhost:8013
De database	http://localhost:8009 mongo://localhost:8009	http://localhost:8009 mongo://localhost:8009

Front-end

De front-end applicaties zijn opgebouwd met een React seed van [Corey House \(https://github.com/coryhouse\)](https://github.com/coryhouse), namelijk [react-slingshot \(https://github.com/coryhouse/react-slingshot\)](https://github.com/coryhouse/react-slingshot). Corey heeft op zijn Github repository een [uitgebreide uitleg \(https://github.com/coryhouse/react-slingshot/blob/master/README.md#get-started\)](https://github.com/coryhouse/react-slingshot/blob/master/README.md#get-started) staan over het werken met zijn slingshot seed. De gekozen bundler voor dit project is [Webpack \(https://webpack.js.org/\)](https://webpack.js.org/).

Redux

Het maken van een applicatie in React kan snel uit de hand lopen als je applicatie groter wordt. Om structuur in de applicatie aan te brengen is gekozen om Redux te gebruiken.

Met Redux beheer je state op applicatieniveau. Redux bestaat uit een aantal onderdelen maar het belangrijkste dat je moet onthouden zijn de volgende drie:

- **store:** opslagplaats van alle data als één groot object (POJO).
- **reducers:** pure functies die de app-state bewerken op basis van binnengekregen data uit zgn. action creators.
- **action creators:** functies die iets doen, bijvoorbeeld een API benaderen of iets uitrekenen. De uitkomst geven ze door aan de reducers. Dat doorgeven wordt '**dispatchen**' genoemd.

Folderindeling (package structuur)

De folderindeling hanteert de naamgevingen van Redux, zodat iedereen die weet wat Redux is meteen snapt waar files moeten komen te staan.

.	
— /build/	# De folder voor gecompileerde output
— /node_modules/	# 3rd-party libraries en utilities
— /src/	# The source code of the application
— /actions/	# Action creators zoals beschreven in Redx
— /components/	# React components die enkel UI logica bevatten
— /constants/	# Constantes die over meerdere files gebruikt
worden	
— /containers/	# React components die toegang hebben tot de
Redux app state	
— /reducers/	# Redux reducers
— /routes/	# Page/screen components met routing informatie
— /store/	# Bevat configuratie die nodig is om de Redux
store op te bouwen	
— /styles/	# Bevat styling files die over meerdere
componenten gebruikt wordt	
— /index.ejs	# Template voor de index.html file, wordt door
webpack gebruikt	
— /index.jsx	# Startup script, koppelt React aan de DOM
— ...	# Overige core files die door webpack gebruikt
worden.	
— /tools/	# Build automation scripts en utilities die
webpack gebruikt.	
— package.json	# De lijst met 3rd party libraries en utilities

Installatie Gateway

Om de gateway werkend te krijgen zijn er een aantal vereisten:

- Raspberry Pi 3.
- Arduino Nano, Mega of Uno met een aangesloten NRF24L01+.
- Er is een werkende versie van RASPBIAN STRETCH WITH DESKTOP geïnstalleerd.
- Er is toegang via SSH of direct op de Raspberry Pi 3 terminal toegang.

MongoDB

Omdat er het een en ander wordt opgeslagen op de Raspberry Pi 3 moet er een database geïnstalleerd worden, in dit geval MongoDB.

Dit kan op de Raspberry Pi 3 gedaan worden met de volgende commando's in de terminal:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install mongodb-server
```

Het upgrade process kan een tijdje duren.

Als de MongoDB server succesvol geïnstalleerd is, kan deze service gestart worden door:

```
$ sudo service mongodb start
```

Java

De gateway draait in een JVM en het is dus nodig om de juiste Java installatie te installeren.

```
sudo su
echo "deb http://ppa.launchpad.net/webupd8team/java/ubuntu xenial main" | tee
/etc/apt/sources.list.d/webupd8team-java.list
apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys EEA14886
```

Mocht bij het uitvoeren van het bovenstaande een error naar voren komen over het ontbreken van dirmngr dan kan dat gefixed worden door dit te installeren:

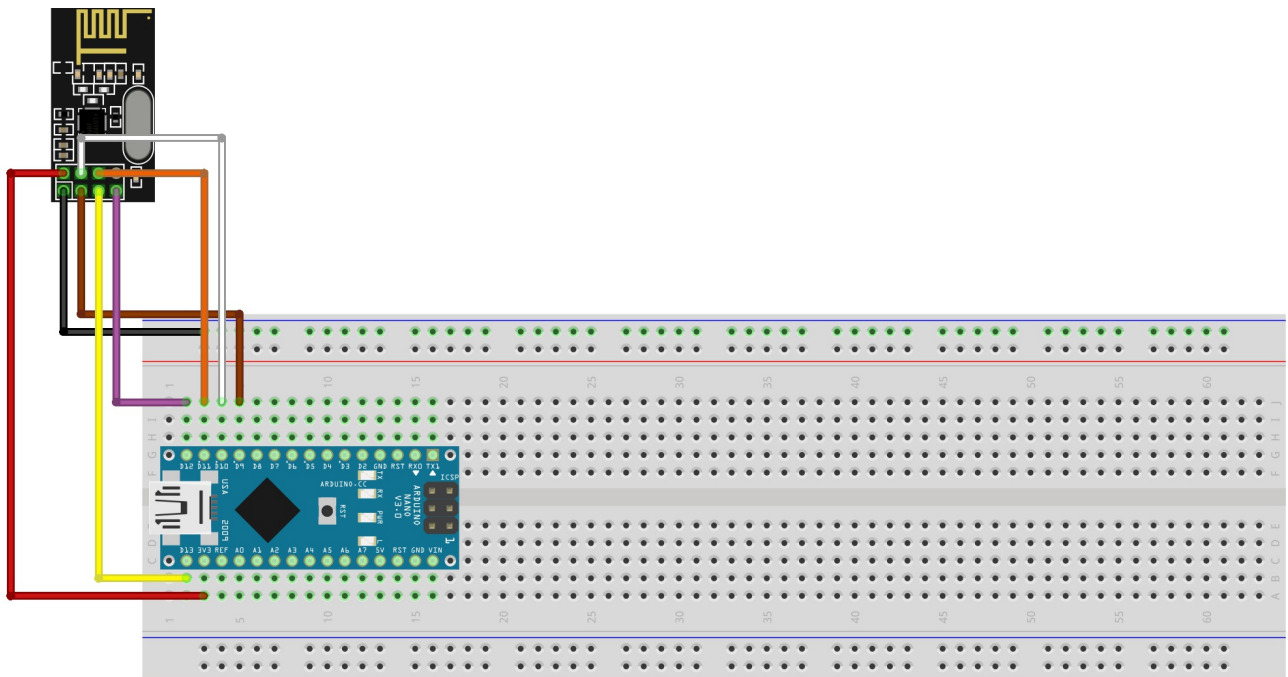
```
sudo apt-get install dirmngr
```

```
sudo apt-get update
sudo apt-get install oracle-java8-installer
```

RXTX

```
sudo apt-get install librx-tx-java
```

Aansluiting Arduino



fritzing

Aansluitschema Arduino Nano met een NRF24L01+

De Arduino moet vervolgens verbonden worden via een USB kabel met de Raspberry Pi 3.

Start gateway

Om de gateway te starten moet eerst de MongoDB aan staan. Dat kan met het eerste commando.

Als deze draait kan daarna de gateway zelf gestart worden.

```
sudo service mongod start #starts mongo service
```

```
java -Djava.library.path=/usr/lib/jni -jar gateway.jar #gateway.jar is te  
vervangen met de jar naam van de gateway
```

Bij het tweede gedeelte, om Java te starten, is het belangrijk dat de volgende regel voor de -jar komt: -Djava.library.path=/usr/lib/jni. Anders wordt de RXTX library niet goed geladen.

Poot

Repo commando's

In dit hoofdstuk wordt uitgelegd wat de "repo commando's" inhouden.
Een "repo commando" is één van de scripts die in de package.json staan.

Dat levert de volgende items op:

Commando	resultaat	Notities
start	Start de apps in development modus	
build	Start de apps in productie modus (en bouwt productie files)	
build-docker	Bouwt zowel de dev als de productie docker images.	
build-docker-dev	Bouwt de dev docker image.	
build-docker-prod	Bouwt de productie docker image	
compile-deliverables	Bouwt alle documentatie	
compile-images	Verzamelt alle images in de deliverables/images map zodat ze gebruikt kunnen worden in de documentatie	! werkt niet op Windows
compile-pva	Bouwt het Plan van Aanpak	Wordt gebouwt in de deliverables map
compile-fo	Bouwt het Functioneel ontwerp	Wordt gebouwt in de deliverables map
compile-to	Bouwt het technisch ontwerp	Wordt gebouwt in de deliverables map
compile-testplan	Bouwt het testplan	Wordt gebouwt in de deliverables map
compile-opleverdocumentatie	Bouwt de opleverdocumentatie	Wordt gebouwt in de deliverables map
generate-pdfs	Zet alle gebouwde bestanden om naar een .pdf	Wordt gebouwt in de deliverables/pdfs map
copy-endpoint-prod	Dit kopieërd alle development instellingen naar de apps	! werkt niet op Windows
copy-endpoint-dev	Dit kopieërd alle development instellingen naar de apps	! werkt niet op Windows
postinstall	Dit script draait NA een npm install en zal de "build-docker" taak uitvoeren	

Ontwikkelhandleiding (hoe ontwikkel ik?)

Op Linux en Mac OS X zit verder nog een limiet op het aantal bestanden / mappen waar een gebruiker tegelijk naar mag "luisteren" voor veranderingen. Om dat op te lossen moet je het volgende commando uitvoeren:

```
echo fs.inotify.max_user_watches=524288 | sudo tee -a /etc/sysctl.conf && sudo sysctl -p
```

Voor technische info klik [hier \(https://github.com/emcrisostomo/fswatch\)](https://github.com/emcrisostomo/fswatch), voor sysctl uitleg [klik \(https://wiki.archlinux.org/index.php/sysctl\)](https://wiki.archlinux.org/index.php/sysctl) hier.

Bijlagen

Begrippenlijst

Begrip Uitleg

Software lijst

In deze lijst vindt je voor de meeste software links naar installatiehandleidingen.

Product	Windows	Mac OS
Mongo	link (https://docs.mongodb.com/manual/installation/)	link (https://docs.mongodb.com)
Docker	link (https://docs.docker.com/toolbox/toolbox_install_windows/)	link (https://docs.docker.com/mac/install/)
docker-compose	link (https://docs.docker.com/compose/install/)	link (https://docs.docker.com)
node	link (http://blog.teamtreehouse.com/install-node-js-npm-windows)	link (http://nodesource.com/nodejs-tutorial-mac-os-x/)
npm	link (https://www.npmjs.com/package/npm)	link (https://www.npmjs.com)
bash	link (https://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/)	-