Operations Manual

Machine Learning Repository

ML-Repo

Team:

Gabríela Jóna Ólafsdóttir
Guðmundur Hermann Þórlaugarson
Jóhann Hinrik Jónsson
Katrín Guðmundsdóttir
Magnús Freyr Sveinsson

Supervisor

Gunnar Sigurðsson

Examiner

Sigurjón Ingi Garðarsson

1 Introduction

This operation manual will go over all the necessary steps to continue development and deployment of the OZ ML Repo. This includes all the services used and installed, the projects directory structure, git repository link, installing dependencies and how to integrate the project.

2 Web Application

Here we will go into the web application, how to set up the project on your machine and how to continue the development and deploy on heroku.

2.1 Setup

1. Clone the oz-app repository onto your local computer by running:

```
$ git clone git@github.com:thorlaugarson/oz-app.git
```

2. Install node.js and npm globally:
   https://nodejs.org/en/download/
3. In the project directory, run:

```
$ npm install
```

   This will add npm and packages needed to run the site.

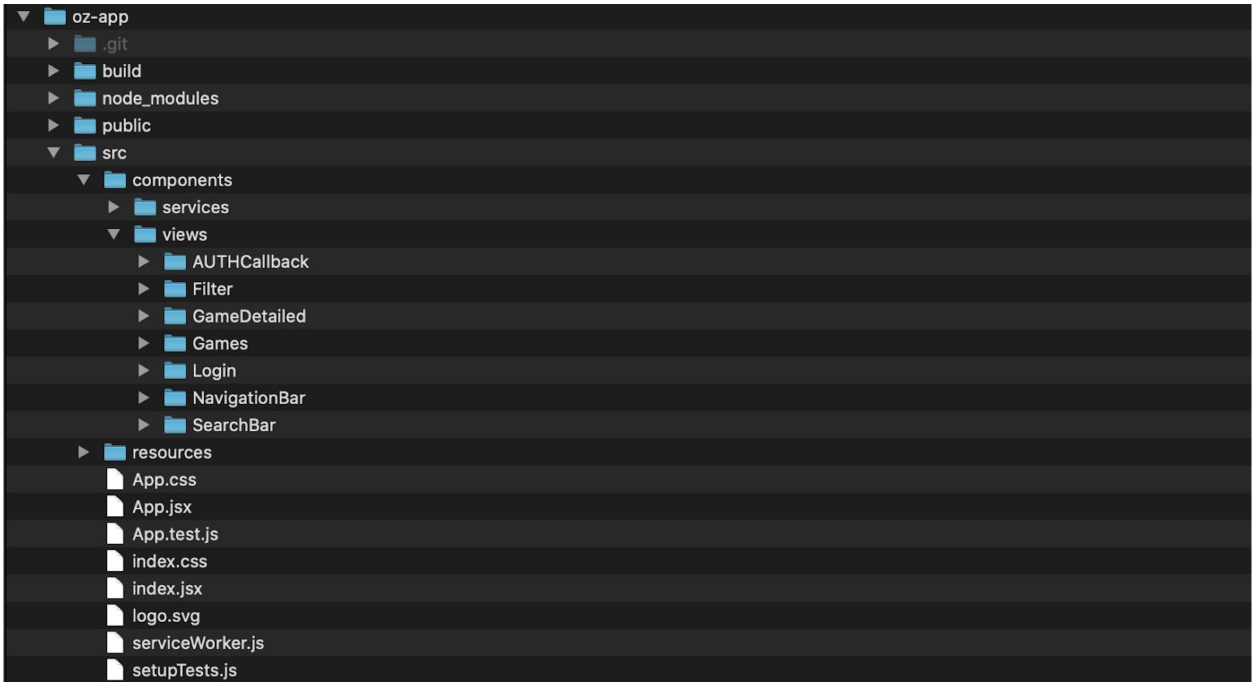
4. In the project directory, run:

```
$ npm start
```

   This will run the app in the development mode.

2.2 Folder structure

Here we see the basic folder structure of the web application. Main files are the *index.jsx* and *app.jsx* are in the *src/* folder and then all of the components are split into either *service/* or *view/* components where the view components have their individual folders with *jsx* and *css* files.



```
▼ oz-app
    ▶ .git
    ▶ build
    ▶ node_modules
    ▶ public
    ▼ src
        ▼ components
            ▶ services
            ▼ views
                ▶ AUTHCallback
                ▶ Filter
                ▶ GameDetailed
                ▶ Games
                ▶ Login
                ▶ NavigationBar
                ▶ SearchBar
        ▶ resources
          App.css
          App.jsx
          App.test.js
          index.css
          index.jsx
          logo.svg
          serviceWorker.js
          setupTests.js
```

2.3 Development and Deployment

1. All development is done on the ‘development’ branch on github and before deployment it will be pushed onto the ‘master’ branch.
2. The project is developed using ReactJs
3. The front end code is located at:
   oz-app/src/index.jsx then *oz-app/src/app.jsx*
   Then from src everything is split into components in the component folder:
   a. There are service components split into different jsx files
   b. All of the view components split into individual directories with their respected JSX files and CSS of each component.
4. When the development merges to the ‘master’ branch the heroku app deploys automatically at https://oz-app.herokuapp.com/

3 Authentication

Login on the web application is done with auth0 and to add new users you need to:

1. Go to https://auth0.com/ and login with the owner account.
2. Click Users & Roles in the sidebar then Users
3. Click + CREATE USER.
4. Fill in email and password then select OZ-Users in connection

4 Maintaining the downloading functionality

4.1 Instance maintenance

Make sure instance "i-0c8679416c8354ed0" is running on Frankfurt EC2

If the instance is down, relaunch it. If it has been removed or is corrupted. Attempt attaching EBS "vol-0849fb72133c2a2f4" to a new machine, t3a.nano instance type with default config is recommended.

In the case of both the instance and EBS being nowhere to be found. To get this back up and running. First create a new t3a.nano instance, with 200GB magnetic drive. Install ffmpeg and python. Import the "dl.py" script into the home directory. Install the python3 version of aws-cli, this tutorial provides a good step by step guide on how "linuxhint.com/install_aws_cli_ubuntu/". Finally set up a cronjob that runs the python script at "0 4 * * *".

4.2 Python script

The python script connects to the “https://core.oz.com/channels/$ChannelSlug/pastEvents” API call, this call is just used for this script. As long as said API call is active, the script should behave correctly. In the event that the API goes through a rewrite. This script depends on the channelId and matchId parameters. As long as those 2 are provided, the script can function correctly.

4.3 AWS

Generate an Access Key ID that only has permissions to upload and download from the bucket oz-test-johann. Import this key into the python-script and the main project if the past key has been revoked.

5 Services

Here is a list of all the tools and services used in the project.

- **Github**: Version control system.
- **Node.js**: Server environment (JavaScript runtime built on Chrome's V8 JavaScript engine).
- **npm**: Package manager.
- **ESLint**: Static code analysis tool.
- **React**: JavaScript library.
- **aws-sdk**: software development kit for Node.js
- **Express.js**: Web application framework for Node.js.
- **Javascript**: Frontend language.
- **Heroku**: Deployment tool.
- **Amazon Web Service**: Host for video database