Operating Manual

T-404-LOKA, Lokaverkefni, 2022

Reykjavik University - School of Computer Science, Menntavegi 1, IS-101 Reykjavík, Iceland

Elías Friðberg Guðjohnsen, Eggert Már Eggertsson, Jason Guðnason, Daníel Þór Calvi

May 12, 2022

Contents

1 Backend instructions

To setup the backend, it requires a .env file with the following parameters to be present at the root of the backend application.

```
DATABASE_URL: <Postgres Connection String>

# Audkenni API
CLIENT_ID: <Audkenni supplied>
CLIENT_SECRET: <Audkenni supplied>
BASE_URI: <Audkenni supplied>
REDIRECT_URI: <Audkenni supplied>

# Code verifier
CODE_VERIFIER_LENGTH = Integer (43 to 128 inclusive)
CODE_CHALLENGE_METHOD = String ("S256" for Sha256)

# JSON Web Token
JWT_SECRET = String (Suggested long and random)
JWT_EXPIRATION = String (Such as "1h" for 1 hour)

# JWT Token Encryption
ENCRYPT_TOKEN = String ("false" or "true")
AES256_KEY = String (Suggested long and random)

# Node
NODE_ENV = "production"

# Application
PORT = Integer (3000 is common for development)
USE_AUDKENNI_API = String ("false" or "true")
```

Once created, the backend can be started by executing

```
npm install
npm start
```

from the root of the backend application.

1.1 Endpoints

The endpoints are documented in a README document which is supplied with the git repository, under the backend root folder.

2 Environment variable information

2.1 DATABASE_URL

A connection string for a Postgresql database.

Commonly as follows.

```
postgresql://user:password@ipaddress:port/dbname
```

2.2 Audkenni API

These parameters are supplied by Auðkenni and are required for electronic ID authentication.

2.3 Code verifier

These parameters are a security challenge during electronic ID authentication.

2.4 JSON Web Token

These parameters are used for authorization, a web token is issued once a user authenticates with the system. Once the token expires, the user can no longer communicate with the backend without logging back in.

2.5 JWT Token Encryption

The token sent to the user upon authentication can be encrypted so it cannot be read by the user, information inside the token is the users social security number, name and authorization level on the system. These fields are not sensitive but encryption but encrypting the token is good practice.

2.6 Node

This parameter denotes what environment the backend is running, it's only there for internal ExpressJS purposes.

2.7 Application

These are application specific parameters, the port the backend runs on as well as if the backend should use electronic ID authentication, note, the backend has no backup authentication system and will authenticate any valid number if false.

2.8 Logging

The backend logs a lot of relevant operations to the terminal and to a file called log.txt, located at the root of the backend application. It will not be present before starting the backend once.

3 Frontend instructions

To start the frontend application run:

```
npm install
npm start
```

from the root of the frontend application. This will open up the program on localhost on port 3000.