Documentation: MVC Code Base

Summary

This is the codebase for a basic webserver using the MVC pattern. This webserver handles Hubspot OAuth method in both posgresql and mongodb. This webserver also has 2 basic test methods that can serve as a base to test functions and endpoints.

Possible issues

The server sometimes has issues connecting to a MongoDB, this is to be fixed in future versions.

Description of the modules

This server consists of 5 main modules, each of which will be explained here.

▶ Details

Controllers

This module contains two files, endpoints.js and oauth-controller.js

```
File: endpoints.js
Purpose: This file serves as an endpoint controller and contains functions to get
the account users of a Hubspot account and to give authorization to an account.
```

```
File: oauth-controller.js
Purpose: This file serves as an authentication controller and contains functions
to install the app in hubspot, handle oauth callback and check for authorization.
This file uses 3 variables from .env: CLIENT_ID, CLIENT_SECRET and SCOPES
```

▶ Details

Models

This module contains six files, encryption.js, hubspot.js, hubspotMongo.js, mongo.js, posgres.js, server.js.

```
File: encryption.js
Purpose: This file serves an encryption model. It handles the encryption and
decryption of JSON.
```

```
File: hubspot.js
Purpose: This file serves as an authentication model. It handles getting, setting
```

```
and refreshing of tokens for authorization. This file works with Postgresql,
please keep it if you're using this db type.
```

```
File: hubspotMongo.js
Purpose: This file serves as an authentication model. It handles getting, setting
and refreshing of tokens for authorization. This files works with MongoDB, please
keep it if you're using this db type.
```

```
File: mongo.js
Purpose: This file serves as an database model. It handles db connection, db
insertion, db getting and db updating. This files works with MongoDB, please keep
it if you're using this db type.
```

```
File: postgres.js
Purpose: This file serves as an database model. It handles db connection, db
insertion, db getting and db updating. This files works with Postgresql, please
keep it if you're using this db type.
```

```
File: server.js
Purpose: This file serves as a server model. It handles server middlewares, route
handling and server listening.
```

▶ Details

Routes

This module contains two files, api.js and root.js

```
File: api.js
Purpose: This file serves as route definer. It contains one endpoint that gets an
account's users.
```

```
File: root.js
Purpose: This file serves as a route definer. It cointains five endpoints that
provide a home page, an error page, an authorization endpoint, an app installation
endpoint and an Oauth callback endpoint.
```

▶ Details

Tests

This module contains three files, basic.js, basic.tests.js and endpoints.tests.js

```
File: basic.js
Purpose: This file serves as a javascript sample function. This function returns a
message that will be tested to showcase a unit testing of a simple function.
```

```
File: basic.tests.js
Purpose: This file serves as a test file for the basic.js function. This test file
tests if the function returns a specific string, and showcases how you can use the
expect and to match functions to test a simple custom function.
```

```
File: endpoints.tests.js
Purpose: This file serves as a test file for endpoints. This test file contains a
test for an uncreated endpoint and serves as a template to check endpoints using
the supertest library.
```

▶ Details

Views

This module contains two files, display-information.js and index.html.

```
File: display-information.js
Purpose: This file has a single function that sets errors in html.
```

```
File: index.html
Purpose: This file serves as a front page. It contains a button that serves as an
app installer and changes depending on the authorization check.
```

User guide

This server will serve as a template you can use to have a basic mvc for your proyects.

- To start the server you can use **npm run start**.
- To run tests you can use **npm run test**. Please note that only the basic test will pass, as it only shows how basic testing can be done on custom functions. To have the endpoints test pass you need to customize it to your needs by replacing the endpoint url you're going to test and changing the

functions inside. The before all and after all are functions designed to run before and after the tests. This allows us to for example create a database record before a test and delete it after the test is completed. You can delete these two functions at will.

- To run a dev environment you can use **npm run dev**. This will run nodemon.
- If you are going to use PostgreSQL, you can delete the hubspotMongo.js and mongo.js files and perform uninstall of the mongoose package.
- If you are going to use MongoDB, you can delete or replace the hubspot.js and postgres.js files and perform uninstall of the pg package.