

Hassan Karar

07/17/2022

## Service Layers

### Overview

In the frontend of my web application, I will be using React.js. It is an open-source JavaScript, and I am familiar with it. And for the backend I will be using Node.js and Express. Because it is efficient open-source and making it fast and easy for me to build my web application.

### Specifications:

For the backend the controllers will be for the business logic layer, this will contain the objects execute the business functions and solve all the problems that related to the business. And the Routing will be using for the endpoints. I am going to need a controller for each HTTP request.

I will be using the GET, POST, PATCH, and DELETE methods as well. For example, the HTTP method GET will be used as to get the customer by ID.

## Auth/Signin

Method: POST

URL: <https://localhost:4059/api/v1/auth/signin>

Purpose: this endpoint is responsible to sign in a user, the http request will be sent along with the username and password, then check if the user exists and the provided password is correct for that specific user, if everything goes okay, a token and a refresh token will be sent as a response from the server with a 200 status, if not an appropriate error message will be sent then.

Example requests:

```
curl --request GET --url https://localhost:4059/api/v1/auth/signin
```

success response:

[illegible]

Error response:

```
Body  Cookies (1)  Headers (9)  Test Results
Pretty  Raw  Preview  Visualize  JSON
1  {
2    "message": "Please check your username or password!"
3  }
```

## Auth/signup

Method: POST

URL: [https:// localhost:4059/api/v1/auth/signup](https://localhost:4059/api/v1/auth/signup)

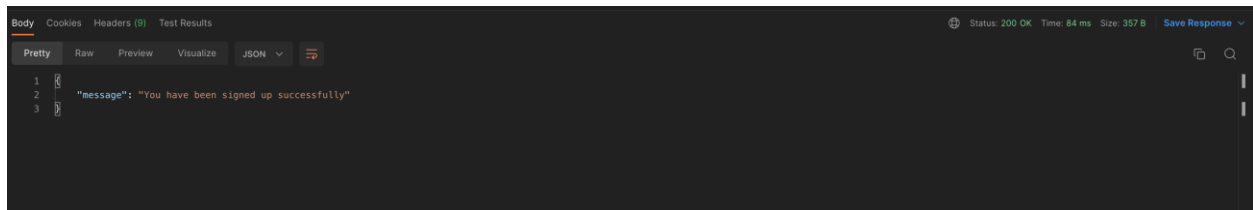
Purpose:

This endpoint is responsible to sign up a new user (owner or customer) by getting all the required information from the frontend.

Example requests:

curl --request POST --url <https:// localhost:4059/api/v1/auth/signup>

success response



Error response:



## Refresh token

Method: POST

URL: <https://localhost:4059/auth/refresh/>

Purpose:

This endpoint generates a new access token for the current session from the refresh token, so all the users can access protected route when send along

Example requests:

```
Curl -request POST -url https://localhost:4059/auth/refresh/
```

### Success response



## Error response

There is no error response since the cookie is used to generate the new access token

## Logout

Method: DELETE

URL: <https://localhost:4059/auth/logout/>

Purpose:

This endpoint logs the user out of the system by deleting the de secured cookie containing the refresh token and deleting it from the database

Example requests:

Curl `-request POST -url https://localhost:4059/auth/logout/`

## Success response



## Error response

There is no error response since the cookie is used to delete the refresh token

## Appointment

Method: POST

URL: <https://localhost:4059/api/v1/appointment>

Purpose:

This endpoint helps create new appointment, by providing the date, time and note.

Example requests:

```
curl --request POST --url https://localhost:4059/api/v1/appointment/
```

success response



Error response



## Appointment cancellation

Method: DELETE

URL: <https://localhost:4059/api/v1/appointment/2>

Purpose:

This endpoint is responsible to help the customer to delete an existing appointment and all the services that go along with it.

Example requests:

```
curl --request DELETE --url https://localhost:4059/api/v1/appointment/
```

## Success response



## Error response



## Appointment /update

Method: PATCH

URL: <https://localhost:4059/api/v1/appointment/4>

Purpose:

This endpoint is responsible to help the customer update an existing appointment, such as time, date and note by sending a number representing the appointment id that needs to be updated.

Example requests:

`curl -request PATCH -url https://localhost:4059/api/v1/appointment/4`

### Success response



### Error response

