



#30DaysOfCode | Backend Track

Day 17

Auth Auth II

*Using ExpressJs or the native Http module, write a server that takes in username, email and password from a SIGNUP route to create a user, and has a LOGIN route to authenticate the user using the password and the **email** or **username**.*

As a step farther, reimplement the signup route to be able save the time and date the user signed up and have that data ready on a protected **"getuser"** route. The **"getuser"** route will take in the user's email.

On a user's authenticated login through the **"login"**, return a session token (tokenize the email) as the response to make sure only an authenticated user with that token can access the **"getuser"** route; this token will be sent with the **"getuser"** request in the request header as the value of the **"Authorization"** key; it will have a "Bearer" prefix.

The token will be verified on each **"getuser"** request to be sure the session token is valid and has not expired.

Enjoy :-)

>> Remember to encrypt and compare the password with bcrypt but don't return it in any responses...

>> All requests and responses in JSON.

>> Check out JSON Web Tokens.

>> Refer to Google, Youtube, StackOverflow and JWT Docs for help.

>> Contact the support group for questions.

Request Format

Path: [GET]https://anyserver/getuser

Example:

Header: "Authorization": "Bearer *exampletoken*"

Body:

```
> {  
  "email": "david@aol.com"  
}
```

Response Format

Path: [GET]https://anyserver/getuser

Example:

```
> {  
  "email": "david@aol.com",  
  "date": "09-04-2020"  
  "time": "12:22pm"  
}
```