JWT: Json web token

🡺 JWT makes it possible to communicate securely between two bodies

🡺 JWT is used for Authorization

How to use JWT?

**1st:**

install jsonwebtoken 🡪 npm i jsonwebtoken

**2nd:**

🡪Import jwt using require

**const jwt = require('jsonwebtoken');**

**3rd:**

🡪 create token:

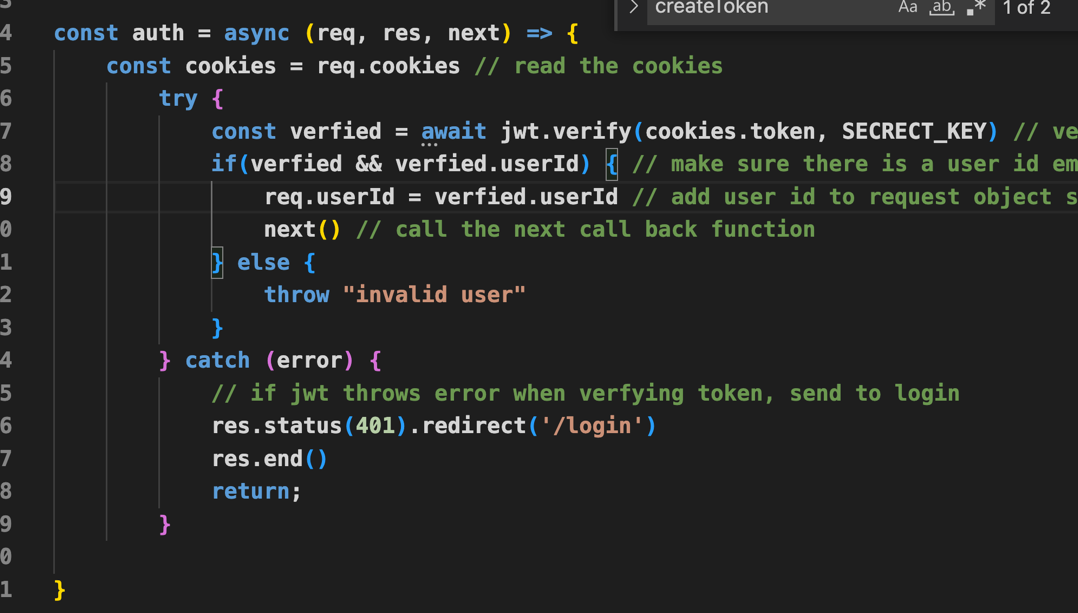
/usermodules/auth.js

Text

Description automatically generated

1. **createToken** method take **userId** as aparameter
2. jwt which we import it before, we can send the payload as we want like userId, or email
   1. in our case: jwt.**sign** method takes three params.
      1. userId 🡪 payload of data
      2. SECRET\_KEY 🡪 var which we stored our secretkey
      3. expiresIn 🡪how much the token will be valid

**4th:**

****

This auth method as **middleware** function🡪 we can use it before any request to check the authentication.

**🡺next third param:**

The middleware function **auth** do the verification code, then passes on the request to the next middleware function in the stack by calling the **next()** function.

**const verfied = await jwt.verify(cookies.token, SECRECT\_KEY) // verify token was created by us**

j**wt.verify**🡪 to verify the given **token** using a **secretkey** and returns **JWT payload**

so, we can verify if the **userId** in the **req** is equal to **the userId(payload)** in the **token**

**🡪 in service.js**

We used **createToken** function inside login method after checking the user email and password and returning object of userId and token

Text

Description automatically generated

**🡪 in app.js**

Ex to use auth middleware fun:

**Text

Description automatically generated**

So here, auth as a second param will go to execute the verification, then if it is succussed, it will pass to the next fun in the stack which is getting the data of the user