Welcome to the

CoGrammar

Authentication with JWT

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.

CoGrammar

Full Stack Web Development Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

CoGrammar

Full Stack Web Development Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: www.hyperiondev.com/support
- Report a **safeguarding** incident: www.hyperiondev.com/safeguardreporting
- We would love your **feedback** on lectures: Feedback on Lectures
- You can turn on live captions on your browser settings incase you are having difficulty in hearing what's being lectured.

CoGrammar

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

What are the three parts of a JWT?

CoGrammar

What is the purpose of the header in a JWT?

CoGrammar

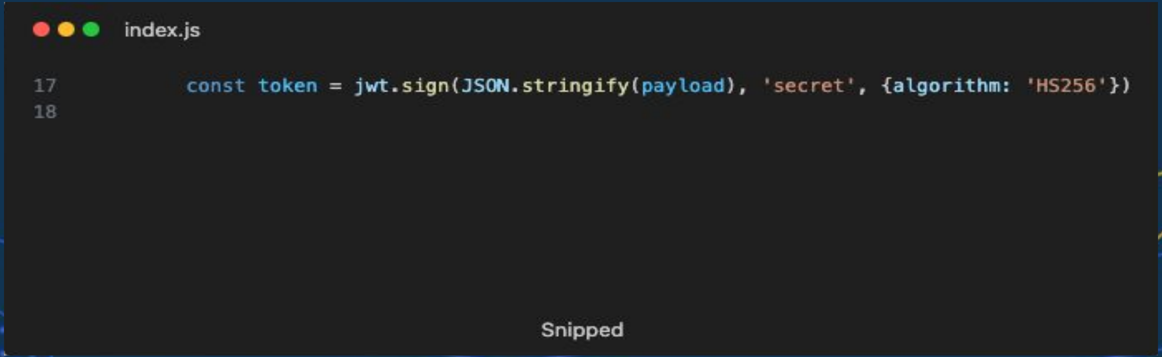
Implementing JWT

CoGrammar

JSON Web Tokens (JWT)

Implementing JWT.

- We will use a popular library to implement JWTs in our application, it makes it easier to sign the tokens and reduces boilerplate code.
- You first need to install it in an already existing express application.
  - `npm install jsonwebtoken`
- Implementing JWT with the library becomes straightforward in this manner



index.js

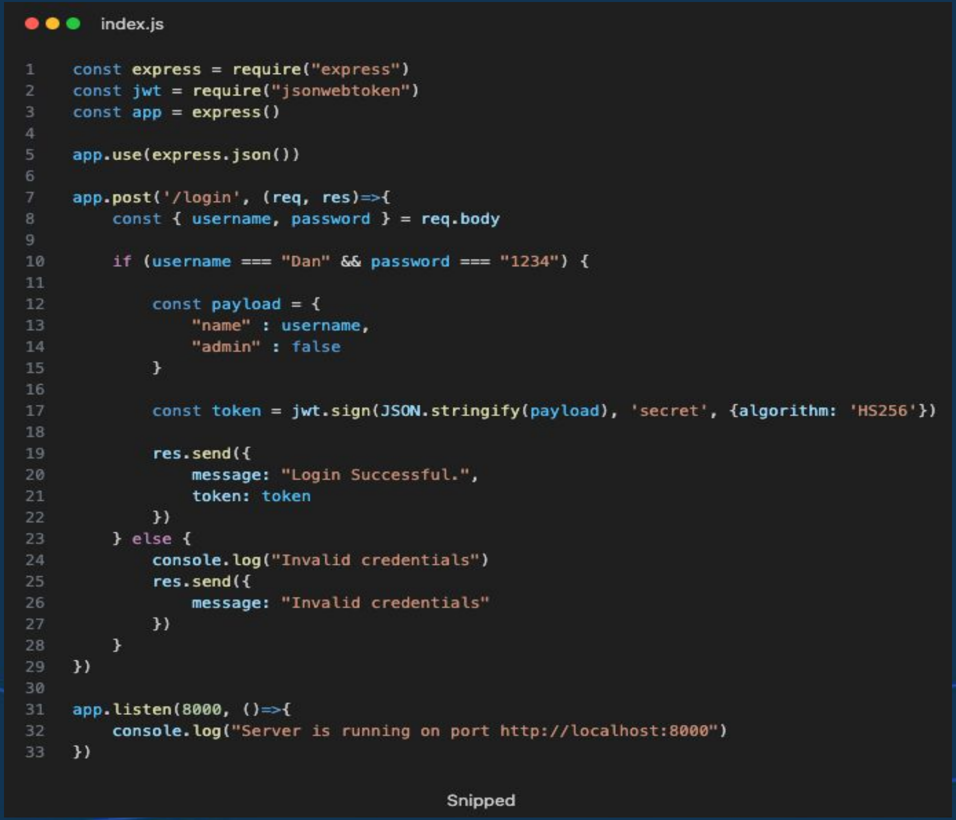
```
        const token = jwt.sign(JSON.stringify(payload), 'secret', {algorithm: 'HS256'})
```

Snipped

CoGrammar

JSON Web Tokens (JWT)

Implementing JWT.



index.js

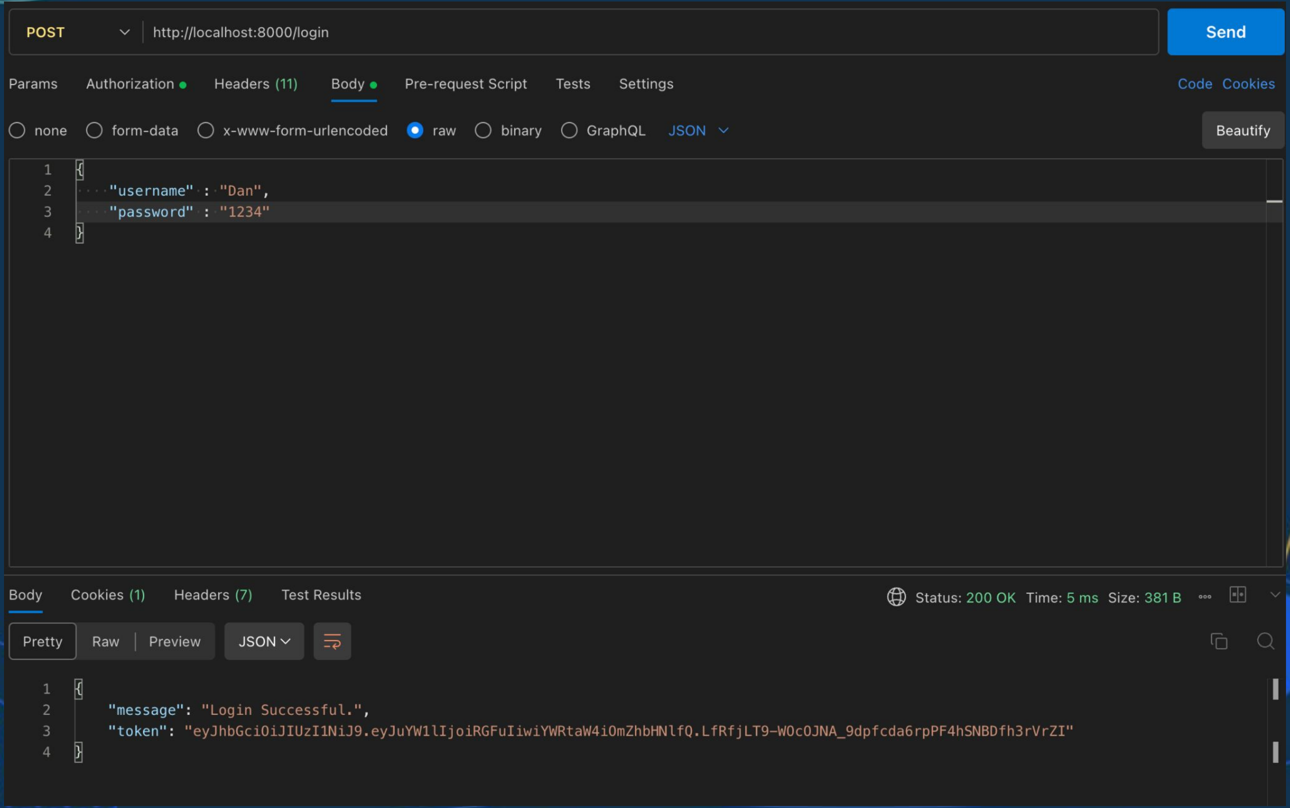
```javascript
const express = require("express")
const jwt = require("jsonwebtoken")
const app = express()

app.use(express.json())

app.post('/login', (req, res)=>{
    const { username, password } = req.body

    if (username === "Dan" && password === "1234") {

        const payload = {
            "name" : username,
            "admin" : false
        }

        const token = jwt.sign(JSON.stringify(payload), 'secret', {algorithm: 'HS256'})

        res.send({
            message: "Login Successful.",
            token: token
        })
    } else {
        console.log("Invalid credentials")
        res.send({
            message: "Invalid credentials"
        })
    }
})

app.listen(8000, ()=>{
    console.log("Server is running on port http://localhost:8000")
})
```

Snipped

CoGrammar

JSON Web Tokens (JWT)

Login Request With Postman



POST http://localhost:8000/login

Body: ( ) none ( ) form-data ( ) x-www-form-urlencoded [✓ raw] ( ) binary ( ) GraphQL JSON

```json
{
    "username" : "Dan",
    "password" : "1234"
}
```

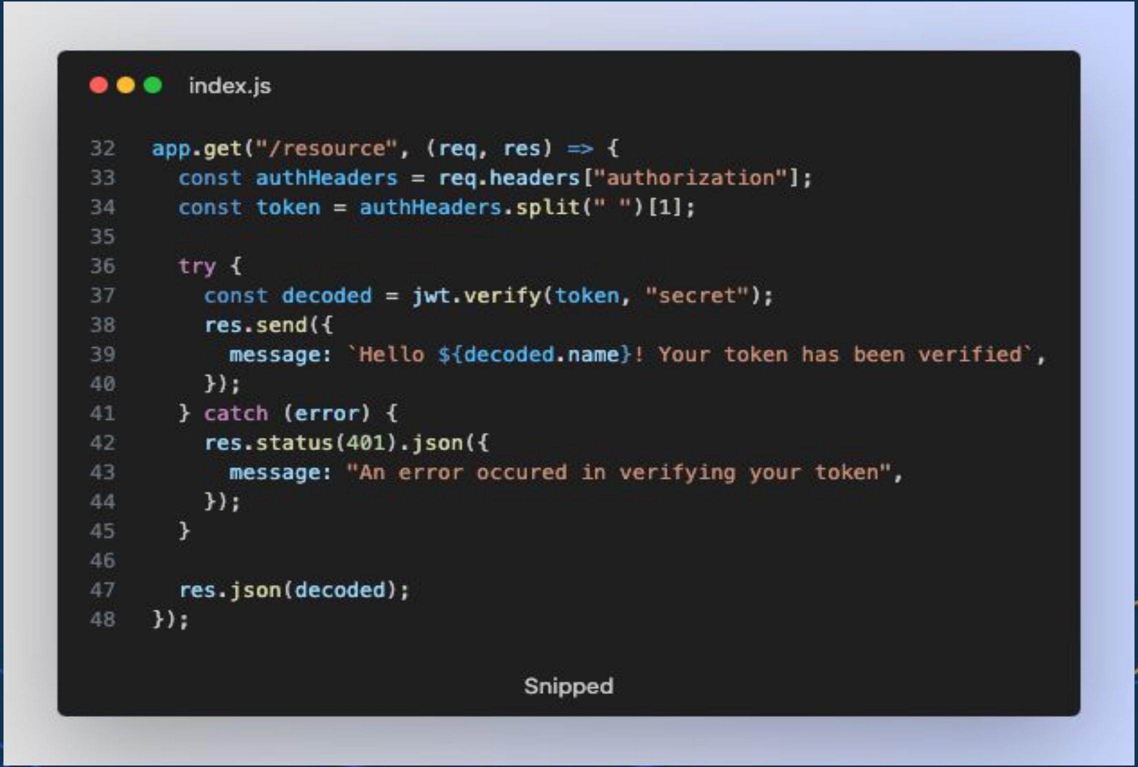
Status: 200 OK Time: 5 ms Size: 381 B

```json
{
    "message": "Login Successful.",
    "token": "eyJhbGciOiJIUzI1NiJ9.eyJuYW1lIjoiRGFuIiwiYWRtaW4iOmZhbHNlfQ.LfRfjLT9-WOcOJNA_9dpfcda6rpPF4hSNBDfh3rVrZI"
}
```

CoGrammar

JSON Web Tokens (JWT)

Verifying token



index.js

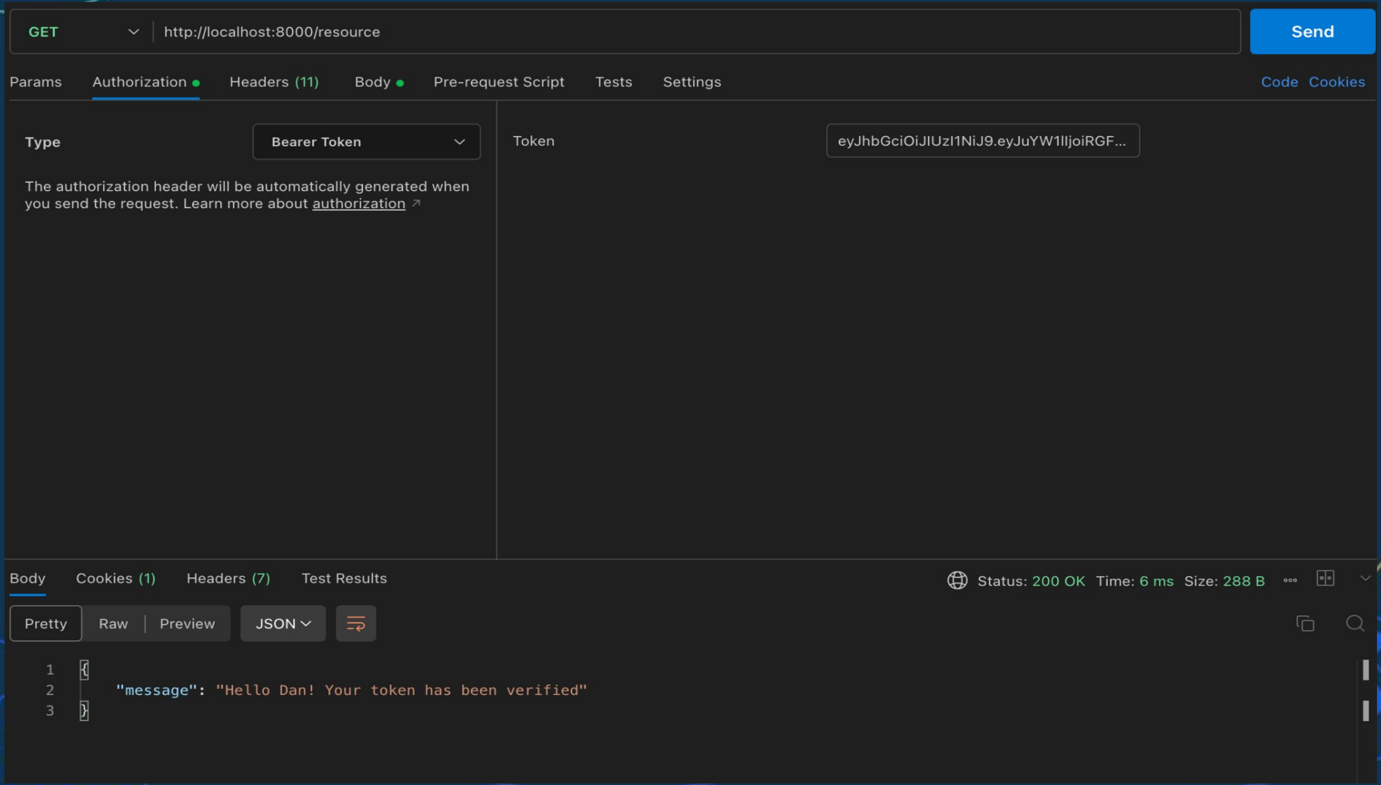
```javascript
app.get("/resource", (req, res) => {
  const authHeaders = req.headers["authorization"];
  const token = authHeaders.split(" ")[1];

  try {
    const decoded = jwt.verify(token, "secret");
    res.send({
      message: `Hello ${decoded.name}! Your token has been verified`,
    });
  } catch (error) {
    res.status(401).json({
      message: "An error occured in verifying your token",
    });
  }

  res.json(decoded);
});
```

Snipped

CoGrammar

JSON Web Tokens (JWT)

Accessing and verifying request with POSTMAN



http://localhost:8000/resource

Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about authorization

Token: eyJhbGciOiJIUzI1NiJ9.eyJuYW1lIjoiRGF...

Status: 200 OK Time: 6 ms Size: 288 B

```
{
    "message": "Hello Dan! Your token has been verified"
}
```

CoGrammar

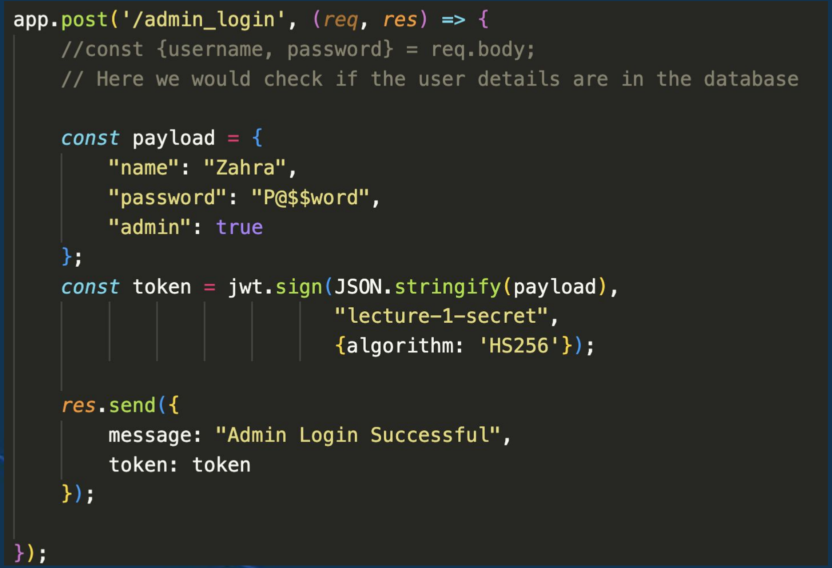
User Permissions

CoGrammar

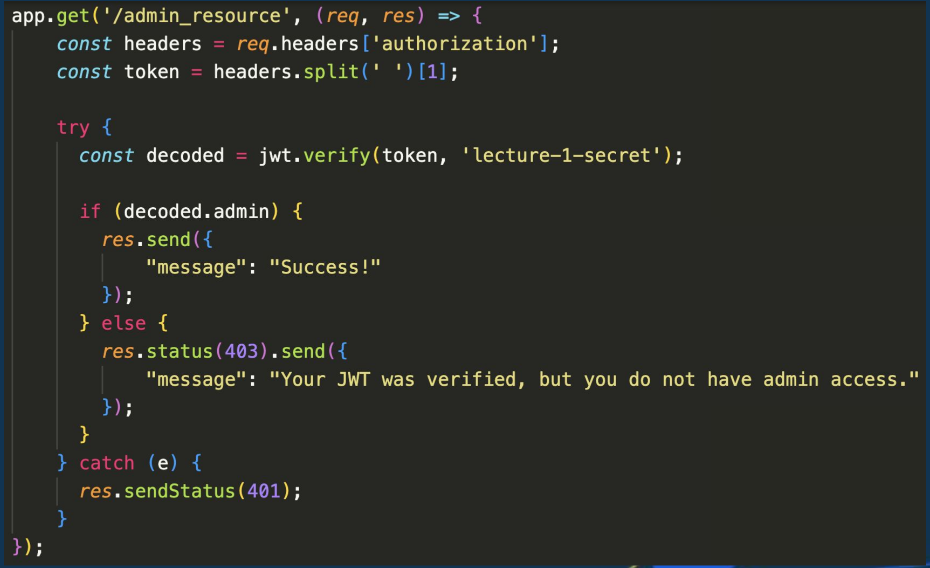
User Permissions

- ❖ By adding an **admin** attribute to the **payload of the auth endpoint**, we can implement user permissions i.e. features or resources only accessible to users with certain privileges.
- ❖ The admin attribute can **only** be added at the endpoint,

User Permissions



```
app.post('/admin_login', (req, res) => {
    //const {username, password} = req.body;
    // Here we would check if the user details are in the database

    const payload = {
        "name": "Zahra",
        "password": "P@$$word",
        "admin": true
    };
    const token = jwt.sign(JSON.stringify(payload),
                           "lecture-1-secret",
                           {algorithm: 'HS256'});

    res.send({
        message: "Admin Login Successful",
        token: token
    });

});
```



```
app.get('/admin_resource', (req, res) => {
    const headers = req.headers['authorization'];
    const token = headers.split(' ')[1];

    try {
      const decoded = jwt.verify(token, 'lecture-1-secret');

      if (decoded.admin) {
        res.send({
            "message": "Success!"
        });
      } else {
        res.status(403).send({
            "message": "Your JWT was verified, but you do not have admin access."
        });
      }
    } catch (e) {
      res.sendStatus(401);
    }
});
```

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar