Task 9: Authentication with JWT

Objective:

Implement user authentication using JSON Web Tokens (JWT) without using Express. Secure API routes and verify users can only access them with valid tokens.

Prerequisites:

- Basic understanding of JavaScript and Node.js.
- Node.js installation.

Concepts:

- **JSON Web Tokens (JWT):**
  - JWT is a compact, URL-safe means of representing claims to be transferred between two parties.
  - JWTs can be signed using a secret or a public/private key pair.

  Example:

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

  // Create a token
  const token = jwt.sign({ userId: 123 }, 'your_secret_key', { expiresIn: '1h' });
  console.log(token);

  // Verify a token
  try {
    const decoded = jwt.verify(token, 'your_secret_key');
    console.log(decoded);
  } catch (err) {
    console.error('Invalid token');
  }
  ```

Setup:

- **Install Node.js:**
  - Ensure Node.js is installed on your machine. You can download it from nodejs.org.

Tasks:

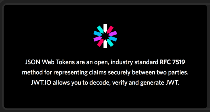
1. **Set Up HTTP Server:**
   - **Task:**
     - Create a file named `server.js` and set up a basic HTTP server using the built-in `http` module.
     - Install the required packages: `jsonwebtoken`, `body-parser`.
     - Create a few routes, including a protected route.
   - **Outcome:**
     - Ensure the server runs correctly and routes are accessible.

2. **User Registration and Login:**
   - **Task:**
     - Create a file named `auth.js` and write code to handle user registration and login.
     - Use JWT to generate tokens for authenticated users.
   - **Outcome:**
     - Ensure users can register and login, receiving a valid JWT on successful authentication.
3. **Protecting Routes:**
   - **Task:**
     - Create a middleware named `authMiddleware.js` to protect routes by verifying JWTs.
     - Apply this middleware to the protected route in `server.js` .
   - **Outcome:**
     - Ensure only users with a valid JWT can access the protected route.

Instructions:

- **Perform the following tasks:**
  - Write the required code in separate files (`server.js` , `auth.js` , `authMiddleware.js`).
  - Run the server and test the routes using tools like Postman to ensure the code executes without errors and demonstrates the use of JWTs.

Resources:

- 

  JSON Web Tokens are an open, industry standard **RFC 7519** method for representing claims securely between two parties. JWT.IO allows you to decode, verify and generate JWT.

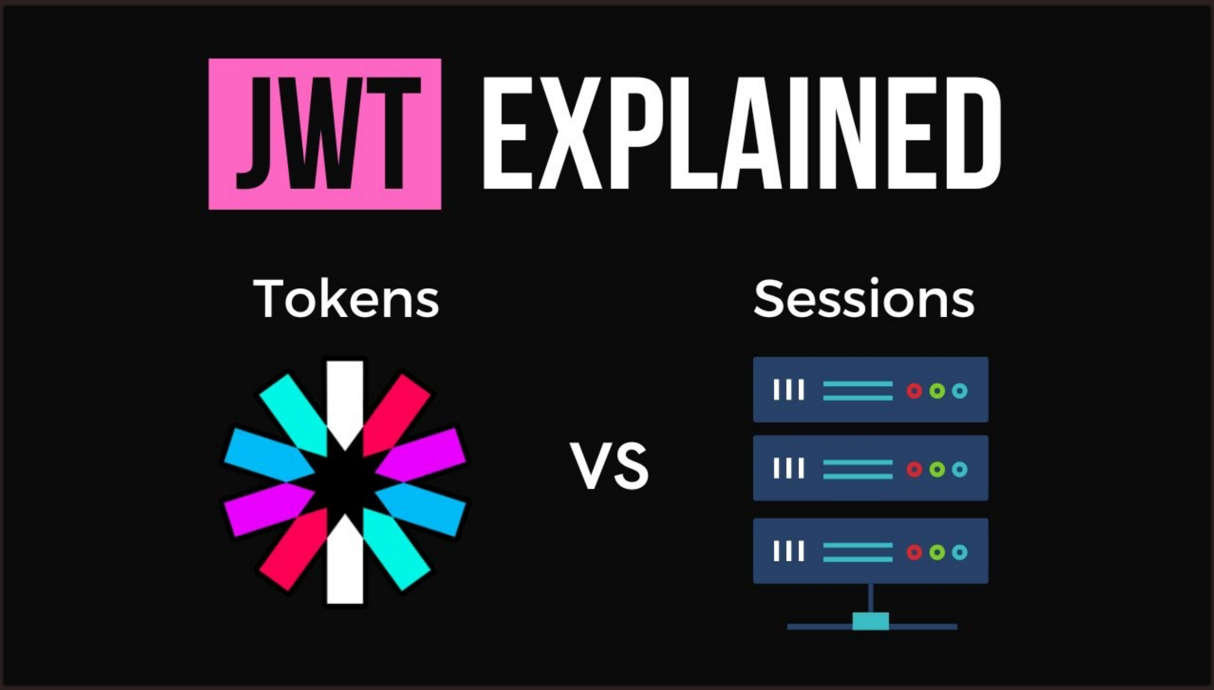
  JWT.IO - JSON Web Tokens Introduction

  Learn about JSON Web Tokens, what are they, how they work, when and why you should...

  Author **auth0.com**
- Index | Node.js v22.3.0 Documentation

  https://nodejs.org/en/docs/

Videos:



Nikita Dev 11:27 min 18,272 Views 672 Likes

Web Dev Simplified 27:36 min 1,017,426 Views 21,983 Likes

GitHub Instructions:

1. **Open in Visual Studio Code:**
   - After clicking on the "Open in Visual Studio Code" button from the GitHub Classroom confirmation page, Visual Studio Code (VSCode) will open the repository directly.
   - If prompted, select "Open" or "Allow" to open the repository in VSCode.
2. **Complete the Task:**

   - Write your solution in the respective files (`server.js` , `auth.js` , `authMiddleware.js`).
3. **Run and Test Your Code:**
   - Run your code to ensure it works correctly. Use the following command:

   ```
   node server.js
   ```

4. **Commit Your Changes:**
   - Commit your changes with a meaningful message:

   ```
   git commit -m "Completed Authentication with JWT task"
   ```

5. **Push Your Changes to Your Forked Repository:**
   - Push your changes to your forked repository:

   ```
   git push origin main
   ```

6. **Create a Pull Request:**
   - Go to your forked repository on GitHub.
   - Click on the "Pull Requests" tab.
   - Click the "New Pull Request" button.
   - Ensure the base repository is the original template repository and the base branch is `main` .
   - Ensure the head repository is your forked repository and the compare branch is `main` .
   - Click "Create Pull Request".
   - Add a title and description for your pull request and submit it.

Summary of Commands:

```
# Fork the repository on GitHub

# Clone the forked repository
git clone https://github.com/your-github-username/repository-name.git
cd repository-name

# Complete the task by writing and running the code in the specified files

# Add, commit, and push your changes
git commit -m "Completed Authentication with JWT task"
git push origin main

# Create a pull request on GitHub
```