**Creating Secured REST APIs**

**Overview**

REST APIs allow different systems to communicate over the internet using standard HTTP methods like GET, POST, PUT, and DELETE. Since these APIs handle data exchange, they must be secured to prevent unauthorized access.

**Securing REST APIs**

To protect APIs, we implement:

* **Authentication**: Verifies the identity of the client.
* **Authorization**: Ensures the client has permission to perform an action.

A common method for securing APIs is **JSON Web Tokens (JWTs)**. JWTs are digitally signed JSON objects used for:

* **Authentication**: A client logs in and receives a token.
* **Authorization**: The token is sent with every request for verification.

**Implementation**

This example consists of two main endpoints:

1. **Authentication Service (/auth/login)**: Generates a JWT for the user.
2. **Test Service (/test)**: Verifies the JWT before granting access.

**Step 1: Set Up Authentication (/auth/login)**

Create an authentication endpoint that validates user credentials and returns a JWT.



**Step 2: Verify Tokens in Test Service (/test)**

Create an endpoint that verifies the JWT before granting access.

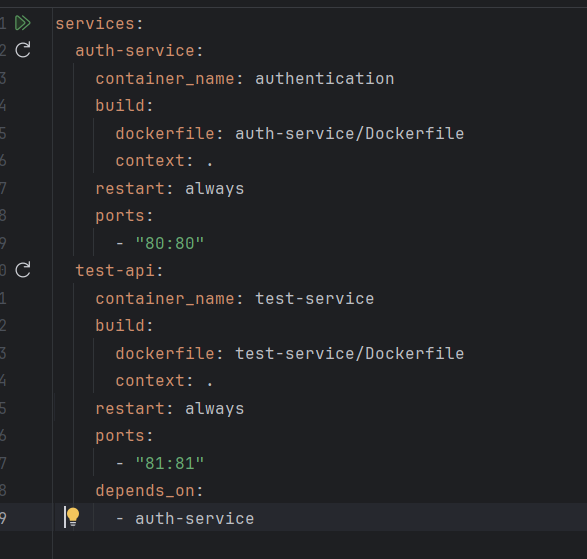


**Step 3: Test the API with Postman**

1. **Obtain a token** – Send a POST request to /auth/login with { "email": "admin", "password": "admin" }.
2. **Verify access** – Use the received token in a POST request to /test.

**Step 4: Deploy with Docker**

Use Docker to containerize the services. Define them in docker-compose.yml:



Run the application with:

docker-compose up –build

