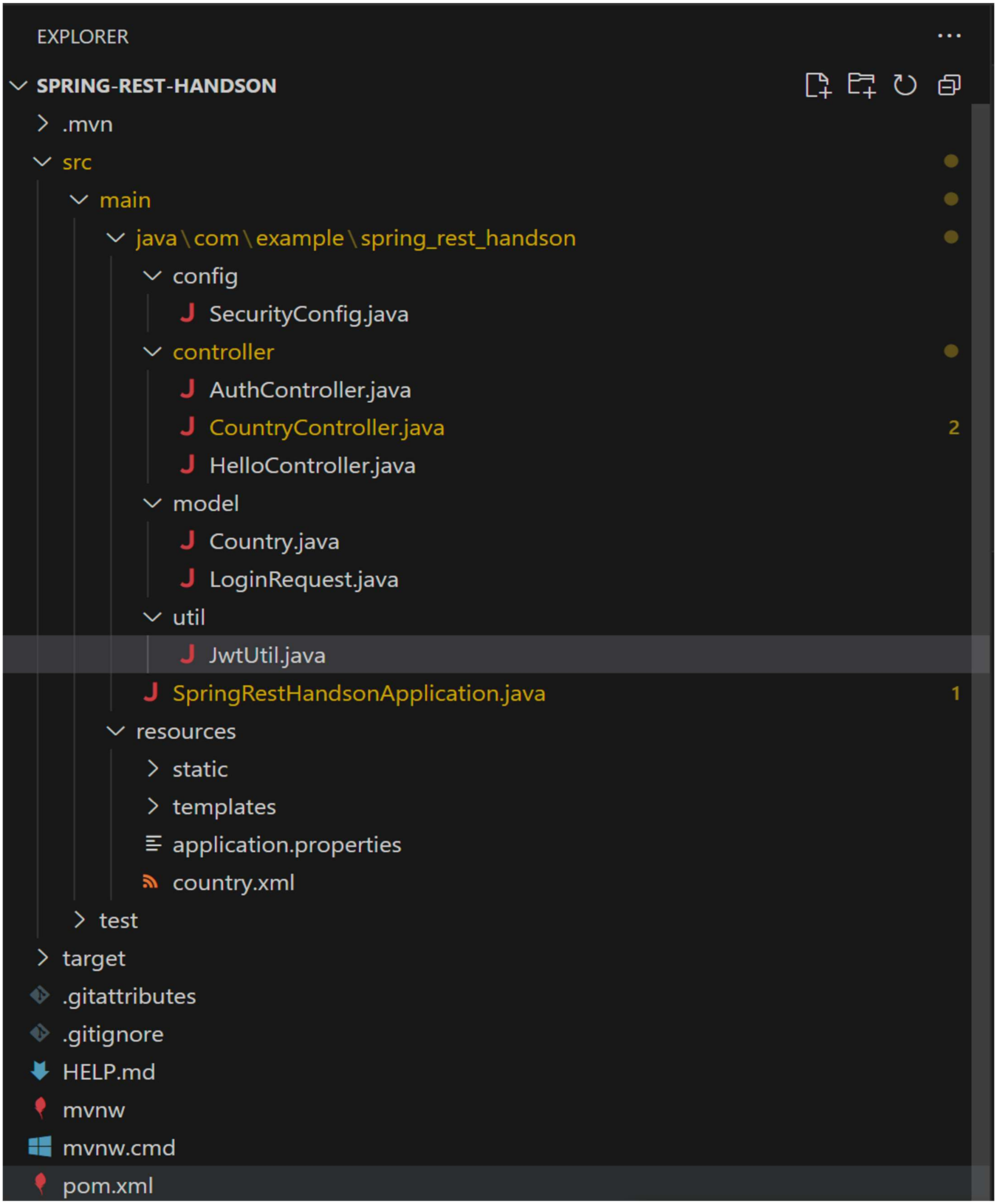
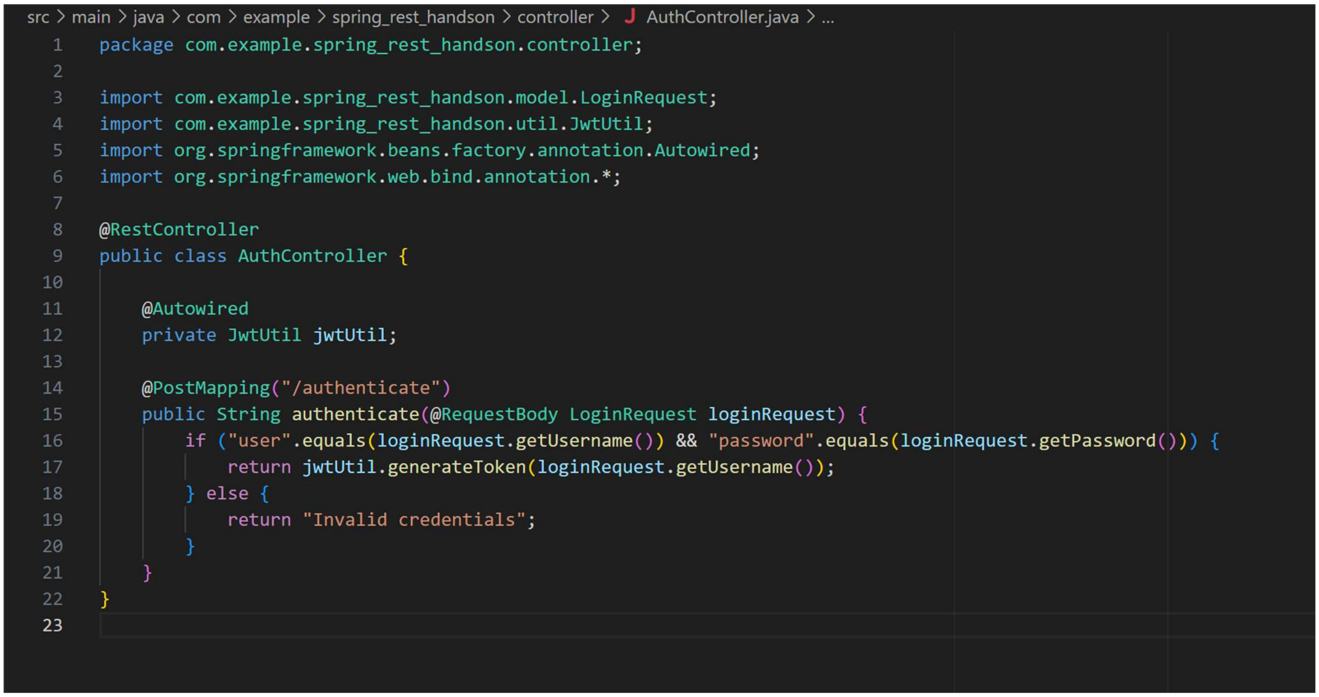
Week4_Spring Rest using Spring Boot 3_6_ Create authentication service that returns JWT

Project Folder



```
EXPLORER
SPRING-REST-HANDSON
  .mvn
  src
    main
      java\com\example\spring_rest_handson
        config
          SecurityConfig.java
        controller
          AuthController.java
          CountryController.java                2
          HelloController.java
        model
          Country.java
          LoginRequest.java
        util
          JwtUtil.java
        SpringRestHandsonApplication.java       1
      resources
        static
        templates
        application.properties
        country.xml
    test
  target
  .gitattributes
  .gitignore
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

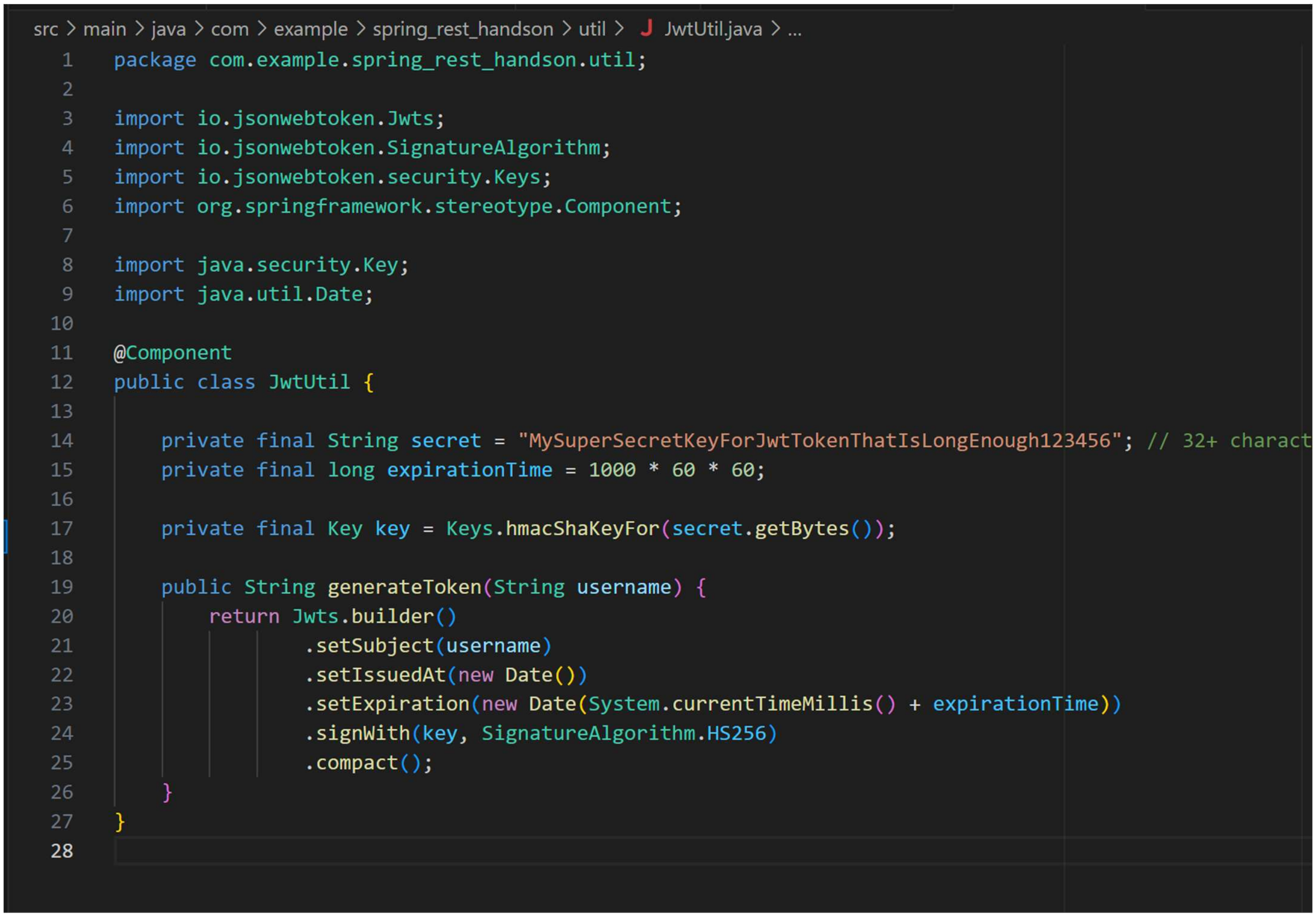
AuthController.java



src > main > java > com > example > spring_rest_handson > controller > AuthController.java > ...

```java
package com.example.spring_rest_handson.controller;

import com.example.spring_rest_handson.model.LoginRequest;
import com.example.spring_rest_handson.util.JwtUtil;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

@RestController
public class AuthController {

    @Autowired
    private JwtUtil jwtUtil;

    @PostMapping("/authenticate")
    public String authenticate(@RequestBody LoginRequest loginRequest) {
        if ("user".equals(loginRequest.getUsername()) && "password".equals(loginRequest.getPassword())) {
            return jwtUtil.generateToken(loginRequest.getUsername());
        } else {
            return "Invalid credentials";
        }
    }
}

```

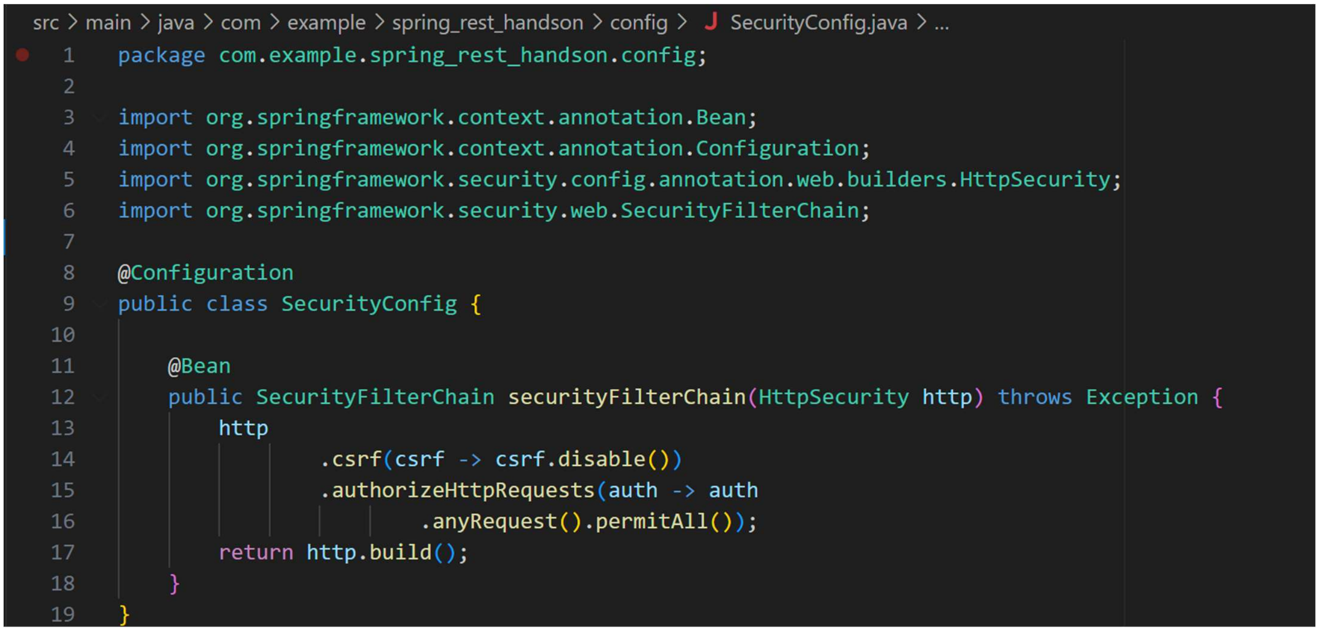
JwtUtil.java



src > main > java > com > example > spring_rest_handson > util > JwtUtil.java > ...

```java
package com.example.spring_rest_handson.util;

import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import io.jsonwebtoken.security.Keys;
import org.springframework.stereotype.Component;

import java.security.Key;
import java.util.Date;

@Component
public class JwtUtil {

    private final String secret = "MySuperSecretKeyForJwtTokenThatIsLongEnough123456"; // 32+ charact
    private final long expirationTime = 1000 * 60 * 60;

    private final Key key = Keys.hmacShaKeyFor(secret.getBytes());

    public String generateToken(String username) {
        return Jwts.builder()
                .setSubject(username)
                .setIssuedAt(new Date())
                .setExpiration(new Date(System.currentTimeMillis() + expirationTime))
                .signWith(key, SignatureAlgorithm.HS256)
                .compact();
    }
}

```

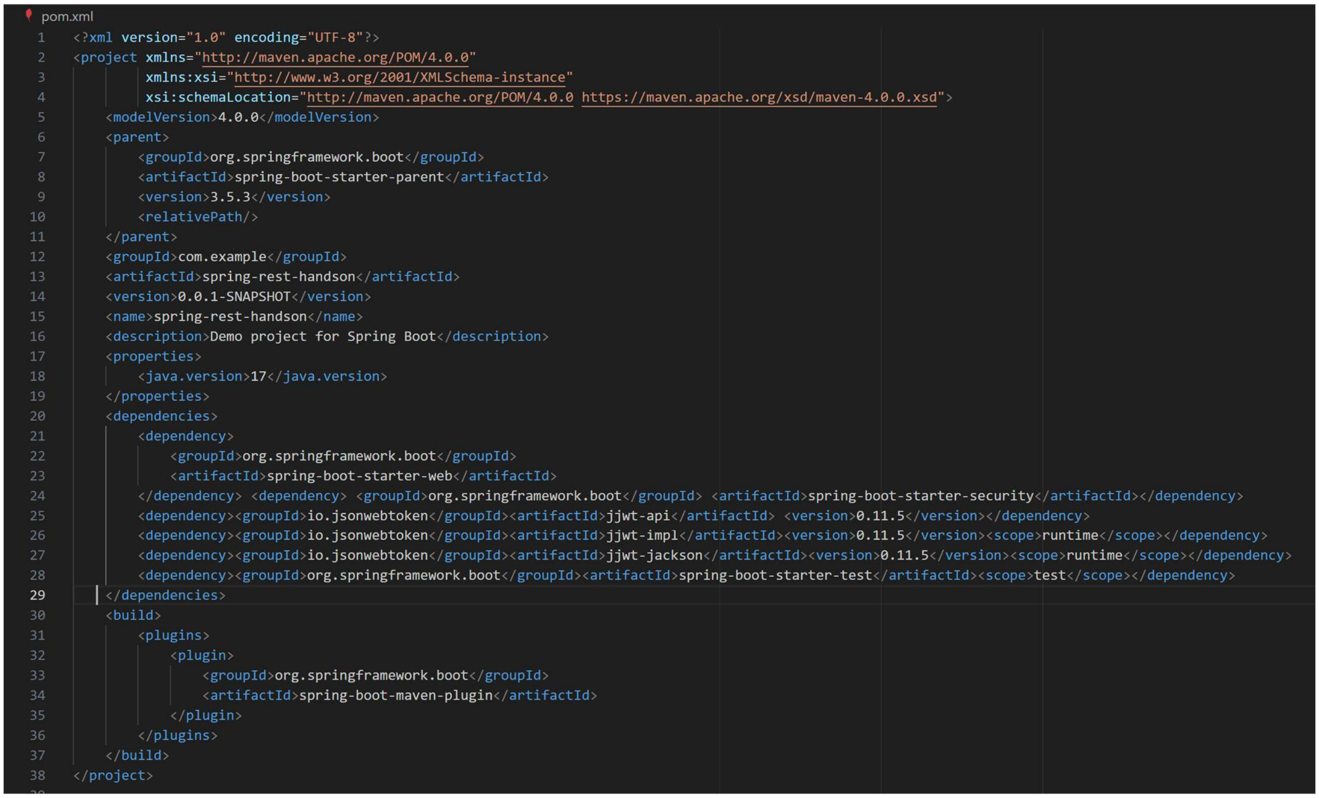
SecurityConfig.java



src > main > java > com > example > spring_rest_handson > config > SecurityConfig.java > ...

```java
package com.example.spring_rest_handson.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
public class SecurityConfig {

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
        http
                .csrf(csrf -> csrf.disable())
                .authorizeHttpRequests(auth -> auth
                        .anyRequest().permitAll());
        return http.build();
    }
}
```

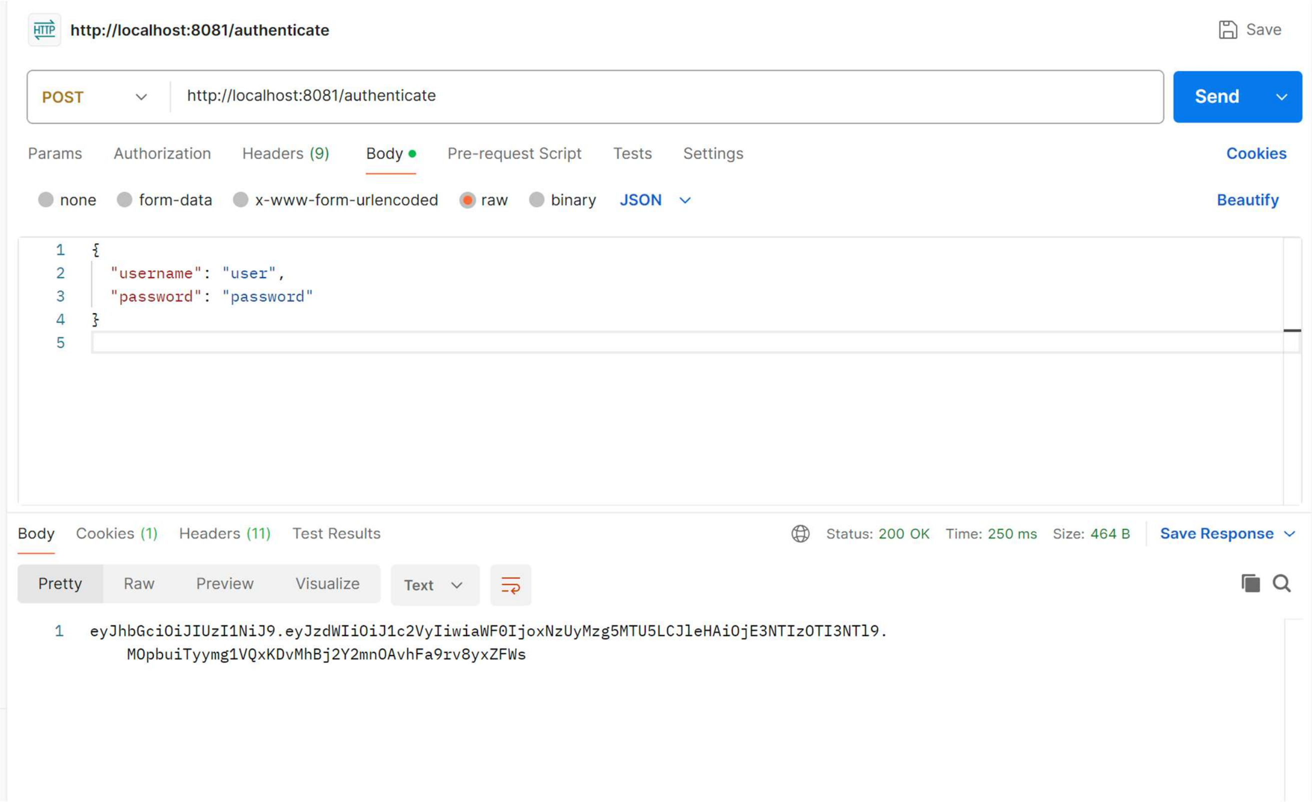
Pom.xml



pom.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.5.3</version>
        <relativePath/>
    </parent>
    <groupId>com.example</groupId>
    <artifactId>spring-rest-handson</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>spring-rest-handson</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency> <dependency> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-starter-security</artifactId></dependency>
        <dependency><groupId>io.jsonwebtoken</groupId><artifactId>jjwt-api</artifactId> <version>0.11.5</version></dependency>
        <dependency><groupId>io.jsonwebtoken</groupId><artifactId>jjwt-impl</artifactId><version>0.11.5</version><scope>runtime</scope></dependency>
        <dependency><groupId>io.jsonwebtoken</groupId><artifactId>jjwt-jackson</artifactId><version>0.11.5</version><scope>runtime</scope></dependency>
        <dependency><groupId>org.springframework.boot</groupId><artifactId>spring-boot-starter-test</artifactId><scope>test</scope></dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

Output



POST http://localhost:8081/authenticate

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON

```
{
  "username": "user",
  "password": "password"
}

```

Status: 200 OK Time: 250 ms Size: 464 B

Text

```
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNzUyMzg5MTU5LCJleHAiOjE3NTIzOTI3NTl9.
    MOpbuiTyymg1VQxKDvMhBj2Y2mnOAvhFa9rv8yxZFWs
```