**Exercise 3: RESOURCE SERVER JWT**

**Project Structure:**

resource-server-jwt/

├── pom.xml # Maven dependencies for OAuth2 Resource Server

└── src/

└── main/

├── java/

│ └── com/

│ └── example/

│ └── demo/

│ ├── DemoApplication.java # Main Spring Boot application class

│ ├── ResourceServerConfig.java # Security configuration for JWT

│ └── SecureController.java # Protected REST endpoint

└── resources/

└── application.yml # Configures server port and JWT issuer

**DemoApplication.java**

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DemoApplication {

  public static void main(String[] args) {

    SpringApplication.run(DemoApplication.class, args);

  }

}

**ResourceServerConfig.java**

package com.example.demo;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

@Configuration

@EnableWebSecurity

public class ResourceServerConfig extends WebSecurityConfigurerAdapter {

  @Override

  protected void configure(HttpSecurity http) throws Exception {

    http

      .authorizeRequests()

        .anyRequest().authenticated()

        .and()

      .oauth2ResourceServer()

        .jwt();

  }

}

**SecureController.java**

package com.example.demo;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class SecureController {

  @GetMapping("/secure")

  public String secure() {

    return "This is a secure endpoint from Resource Server";

  }

}

**Application.yml**

server:

  port: 8081

spring:

  security:

    oauth2:

      resourceserver:

        jwt:

          issuer-uri: http://localhost:8081

**pom.xml**

<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

         http://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>2.7.18</version>

    <relativePath />

  </parent>

  <groupId>com.example</groupId>

  <artifactId>resource-server-jwt</artifactId>

  <version>0.0.1-SNAPSHOT</version>

  <name>resource-server-jwt</name>

  <properties>

    <java.version>11</java.version>

  </properties>

  <dependencies>

    <dependency>

      <groupId>org.springframework.boot</groupId>

      <artifactId>spring-boot-starter-security</artifactId>

    </dependency>

    <dependency>

      <groupId>org.springframework.boot</groupId>

      <artifactId>spring-boot-starter-web</artifactId>

    </dependency>

    <dependency>

      <groupId>org.springframework.boot</groupId>

      <artifactId>spring-boot-starter-oauth2-resource-server</artifactId>

    </dependency>

  </dependencies>

  <build>

    <plugins>

      <plugin>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-maven-plugin</artifactId>

        <configuration>

          <mainClass>com.example.demo.DemoApplication</mainClass>

        </configuration>

      </plugin>

    </plugins>

  </build>

</project>

**Output:**

