Pom.xml:

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

</dependencies>

Application.yml:  
server:

port: 8081

spring:

security:

oauth2:

resourceserver:

jwt:

issuer-uri: <https://issuer.example.com>

ResourceServerConfig.java:

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;

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

@Configuration

@EnableWebSecurity

public class ResourceServerConfig {

@Bean

public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {

http

.authorizeHttpRequests(auth -> auth

.requestMatchers("/public").permitAll()

.anyRequest().authenticated()

)

.oauth2ResourceServer(oauth2 -> oauth2.jwt());

return http.build();

}

}

SecureController.java:

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\*\* - Access granted!";

}

@GetMapping("/public")

public String publicEndpoint() {

return "This is a public endpoint - No token required.";

}

}