

**Solution**

**application.yml**

spring:  
 security:  
 oauth2:  
 client:  
 registration:  
 google:   
 client-id: 740499750539-i975fd0s7j0r7lasn73dn1thkin4tfi4.apps.googleusercontent.com   
 client-secret: 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   
 scope: openid,profile,email  
 authorization-grant-type: authorization\_code  
 redirect-uri: "{baseUrl}/login/oauth2/code/{registrationId}"  
 provider:  
 google: # This should match the registrationId if using common provider names  
 authorization-uri: https://accounts.google.com/o/oauth2/auth  
 token-uri: https://oauth2.googleapis.com/token  
 user-info-uri: https://openidconnect.googleapis.com/v1/userinfo  
 user-name-attribute: sub

**pom.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/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.example</groupId>  
 <artifactId>CentralAuthApp</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>CentralAuthApp</name>  
 <description>Spring Boot Centralized Authentication Application</description>  
 <url/>  
 <licenses>  
 <license/>  
 </licenses>  
 <developers>  
 <developer/>  
 </developers>  
 <scm>  
 <connection/>  
 <developerConnection/>  
 <tag/>  
 <url/>  
 </scm>  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-oauth2-client</artifactId>  
 </dependency>  
 <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-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.security</groupId>  
 <artifactId>spring-security-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>

**SecurityConfig.java**

package com.example.CentralAuthApp;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.Customizer;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;  
import org.springframework.security.web.SecurityFilterChain;  
  
@Configuration  
@EnableWebSecurity  
public class SecurityConfig {  
  
 @Bean  
 public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {  
 http  
 .authorizeHttpRequests(authorize -> authorize  
 .anyRequest().authenticated()  
 )  
 .oauth2Login(Customizer.withDefaults());  
 return http.build();  
 }  
}

**UserController.java**

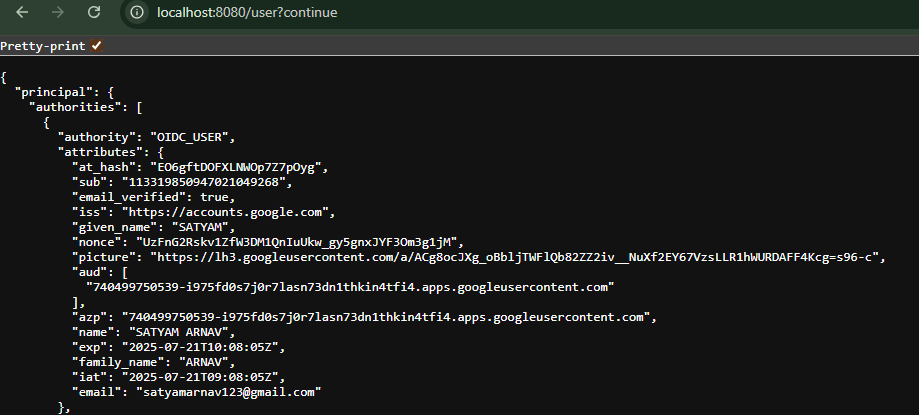
package com.example.CentralAuthApp;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
import java.security.Principal;  
  
@RestController  
public class UserController {  
  
 @GetMapping("/user")  
 public Principal user(Principal principal) {  
 return principal;  
 }  
}

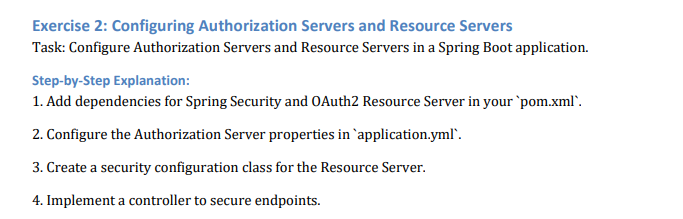
**CentralAuthAppApplication.java**

package com.example.CentralAuthApp;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class CentralAuthAppApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(CentralAuthAppApplication.class, args);  
 }  
}

**Output**

{  
 "principal": {  
 "authorities": [  
 {  
 "authority": "OIDC\_USER",  
 "attributes": {  
 "at\_hash": "EO6gftDOFXLNWOp7Z7pOyg",  
 "sub": "113319850947021049268",  
 "email\_verified": true,  
 "iss": "https://accounts.google.com",  
 "given\_name": "SATYAM",  
 "nonce": "UzFnG2Rskv1ZfW3DM1QnIuUkw\_gy5gnxJYF3Om3g1jM",  
 "picture": "https://lh3.googleusercontent.com/a/ACg8ocJXg\_oBbljTWFlQb82ZZ2iv\_\_NuXf2EY67VzsLLR1hWURDAFF4Kcg=s96-c",  
 "aud": [  
 "740499750539-i975fd0s7j0r7lasn73dn1thkin4tfi4.apps.googleusercontent.com"  
 ],  
 "azp": "740499750539-i975fd0s7j0r7lasn73dn1thkin4tfi4.apps.googleusercontent.com",  
 "name": "SATYAM ARNAV",  
 "exp": "2025-07-21T10:08:05Z",  
 "family\_name": "ARNAV",  
 "iat": "2025-07-21T09:08:05Z",  
 "email": "satyamarnav123@gmail.com"  
 }



**Solution**

**pom.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>ResourceServerApp</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>ResourceServerApp</name>  
 <description>Spring Boot Resource Server Application</description>  
 <url/>  
 <licenses>  
 <license/>  
 </licenses>  
 <developers>  
 <developer/>  
 </developers>  
 <scm>  
 <connection/>  
 <developerConnection/>  
 <tag/>  
 <url/>  
 </scm>  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
 <dependencies>  
 <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-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-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.security</groupId>  
 <artifactId>spring-security-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>

**application.yml**

server:

port: 8081

spring:  
 security:  
 oauth2:  
 resourceserver:  
 jwt:  
 issuer-uri: <https://issuer.example.com>

**ResourceServerConfig.java**

package com.example.ResourceServerApp;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.Customizer;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;  
import org.springframework.security.web.SecurityFilterChain;  
  
@Configuration  
@EnableWebSecurity  
public class ResourceServerConfig {  
  
 @Bean  
 public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {  
 http  
 .authorizeHttpRequests(authorize -> authorize  
 .anyRequest().authenticated()  
 )  
 .oauth2ResourceServer(oauth2ResourceServer -> oauth2ResourceServer  
 .jwt(Customizer.withDefaults()) // Enables JWT support for resource server  
 );  
 return http.build();  
 }  
}

**SecureController.java**

package com.example.ResourceServerApp;  
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";  
 }  
}

**ResourceServerAppApplication.java**

package com.example.ResourceServerApp;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class ResourceServerAppApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(ResourceServerAppApplication.class, args);  
 }  
  
}

**Output**

