Que es JWT

- En el mismo se define un mecanismo para poder propagar entre dos partes, y de forma segura, la identidad de un determinado usuario, además con una serie de claims o privilegios.
- Estos privilegios están codificados en objetos de tipo JSON, que se incrustan dentro de del payload o cuerpo de un mensaje que va firmado digitalmente.
- https://jwt.io/

JSON Web Token (JWT) estándar abierto de seguridad (RFC 7519)

spring by Pivotal

Cliente: Frontend

Servidor: API Rest

Authorization Server

Login

POST: /api/login con username y password

Valida autenticación attemptAuthentication

Error Credenciales: 401 Unauthorized

no autenticado?

si

Retorna el JWT al cliente: 200 OK

Almacena JWT LocalStore o SessionStore

Crea un JWT con un Secret

Resource Server

Acceder a un recurso
Request HTTP

Envía el JWT en las cabeceras HTTP: Authorization

Verifica la firma del JWT

no

Acceso Denegado: 403 Forbidden

token válido?

si

Obtiene datos del usuario (Claims)

Autenticación: Verifica Permisos y Roles

Acceso Denegado: 403 Forbidden

no

permisos?

si

Retorna respuesta al cliente JSON o XML: 200 OK

Accede a Servicio API REST

JSON Web Tokens are an open, industry standard **RFC 7519** method for representing claims securely between two parties.

JWT.IO allows you to decode, verify and generate JWT.

LEARN MORE ABOUT JWT

SEE JWT LIBRARIES

Algorithm HS256

Encoded

PASTE A TOKEN HERE

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6I
kpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ
.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_a
dQssw5c

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE


```
{
  "alg": "HS256",
  "typ": "JWT"
}
```


PAYLOAD: DATA


```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```


VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

Spring Boot
Spring Framework
Spring Data
Spring Cloud
Spring Cloud Data Flow
Spring Security
Spring Security Kerberos
Spring Security OAuth
Spring Security SAML
Spring GraphQL
Spring Session
Spring Integration
Spring HATEOAS

Spring Security OAuth

2.5.1.RELEASE

OVERVIEW | LEARN

Deprecation Notice

The Spring Security OAuth project is deprecated. The latest OAuth 2.0 support is provided by Spring Security. See the OAuth 2.0 Migration Guide for further details.

Spring Security OAuth provides support for using Spring Security with OAuth (1a) and OAuth2 using standard Spring and Spring Security programming models and configuration idioms.

Features

- Support for OAuth providers and OAuth consumers

Características de Spring Security

- Provee características de seguridad para aplicaciones empresariales Java EE
- Maneja componentes de Autenticación y autorización

Spring Security

- Autenticación
- Autorización(Control de acceso)

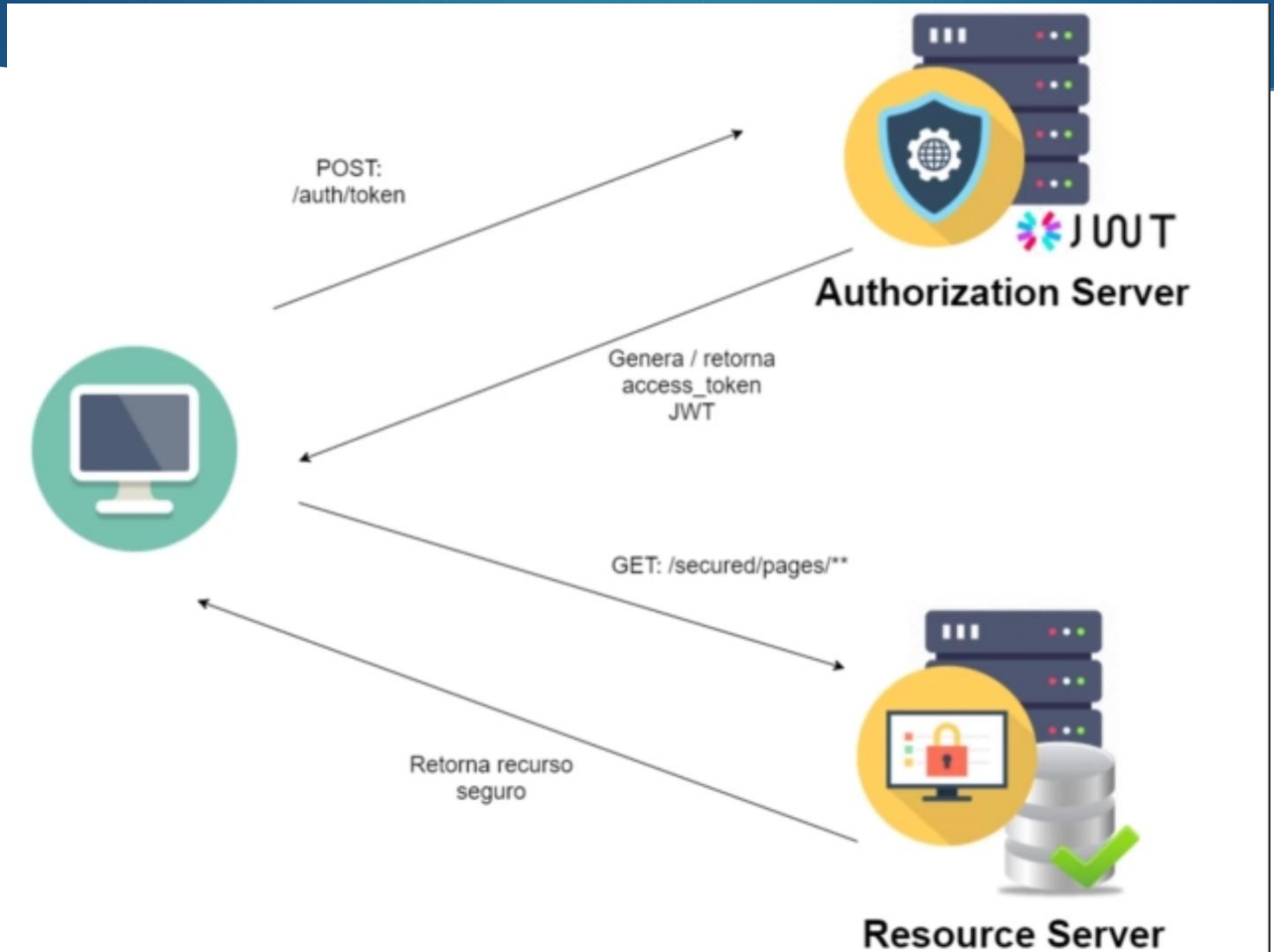
Autenticación

- Se refiere al proceso de establecer un principal(un principal significa un usuario, dispositivo o algún otro sistema el cual puede ejecutar alguna acción en nuestro sistema), en general permite a los principal autenticarse en base a cualquier proveedor de seguridad. Base de datos relacional principalmente y autenticación http.

Autorización

- Se refiere al proceso de decidir si se otorga acceso a un usuario para realizar una acción dentro de la aplicación, es decir para controlar el acceso a los recursos de la aplicación por medio de la asignación de roles y permisos a grupos de usuarios

OAuth


POST:
/auth/token
JWT
Authorization Server
Genera / retorna
access_token
JWT
GET: /secured/pages/**
Retorna recurso
seguro
Resource Server

url: POST /auth/token

header:

- Authorization: Basic Base64(client_id:client_secret)
- Content-Type: application/x-www-form-urlencoded

body:

- grant_type = password
- username = andres
- password = 12345

Authorization Server


```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "bearer",
  "refresh_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "expires_in": 3599,
  "scope": "read write",
  "jti": "58efb674-46e6-4f6b-bbf0-e92e21e4b34a"
}
```

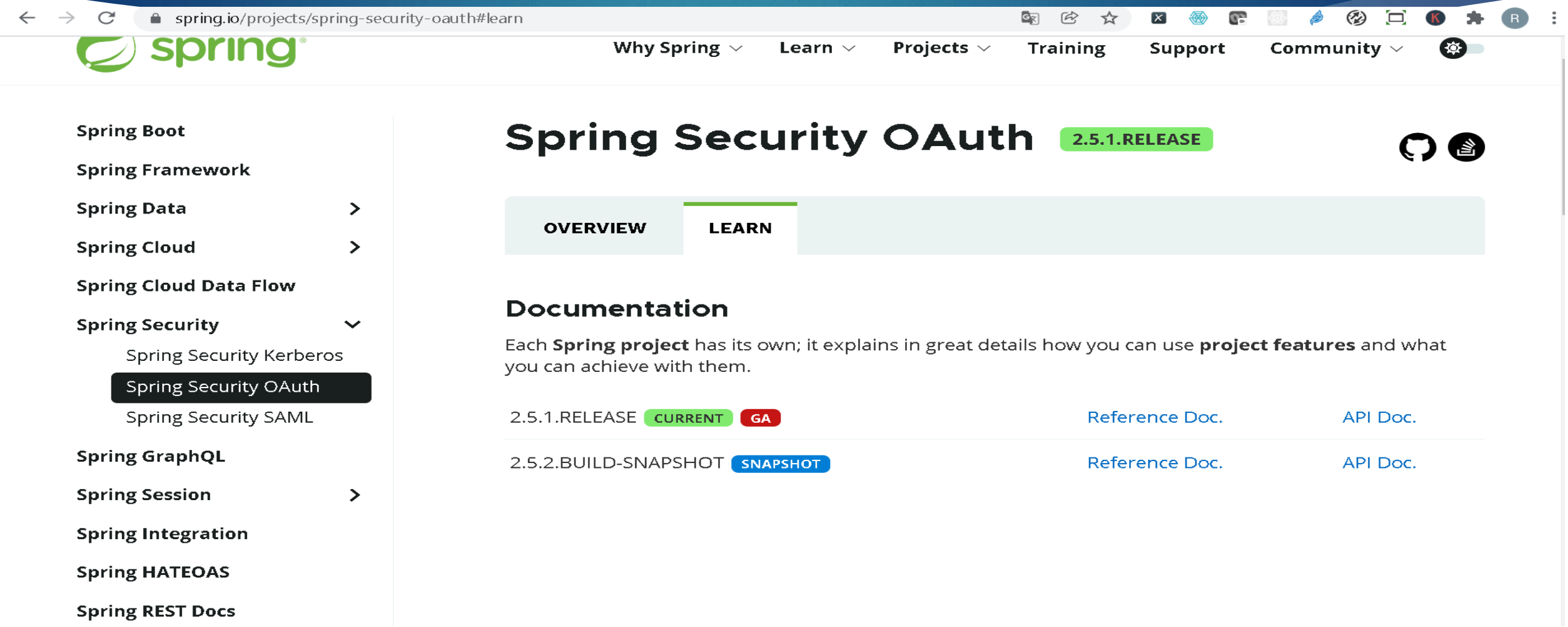
url: GET /api/clientes

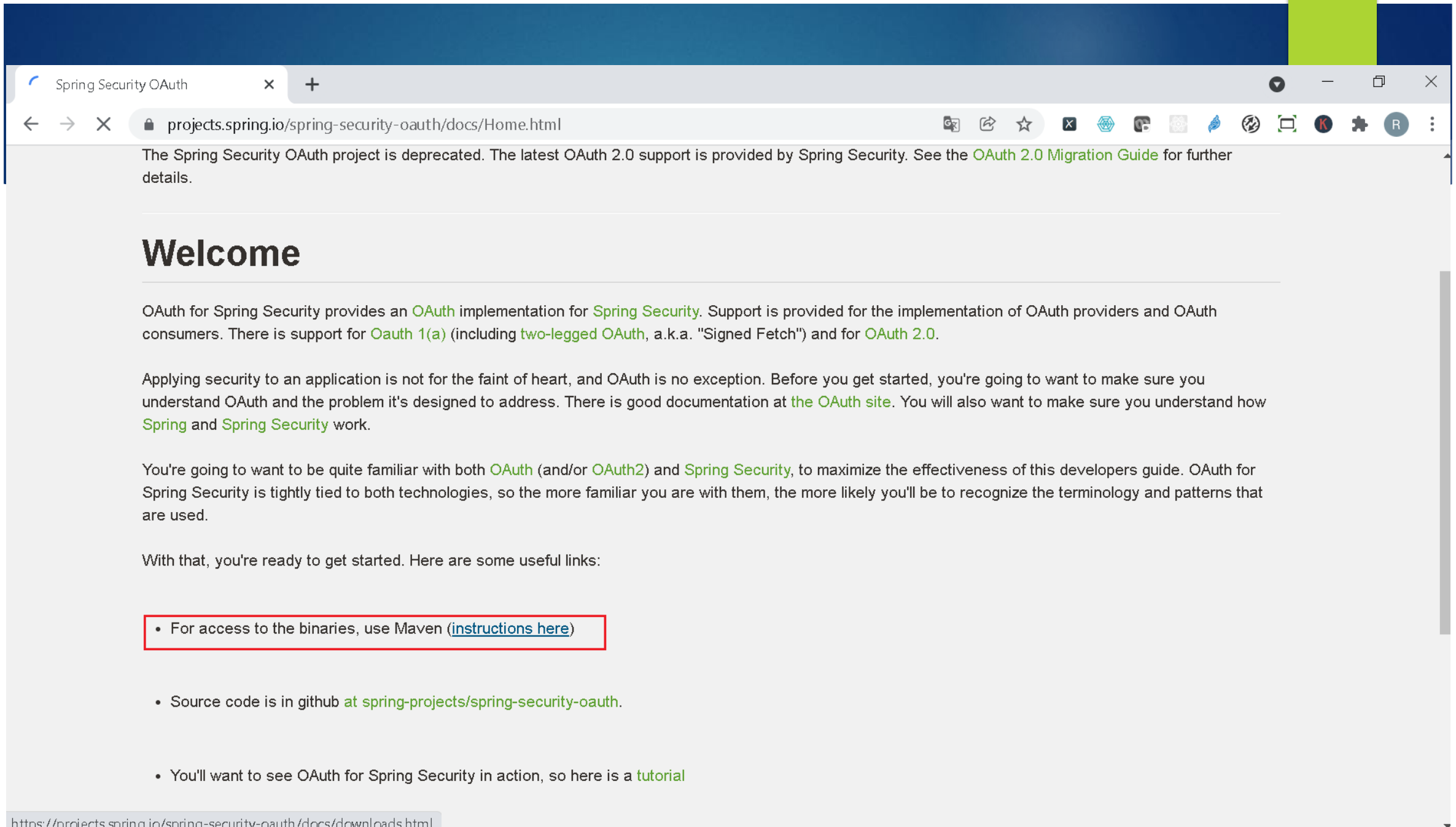
header: Authorization Bearer access_token

Resource Server


```
[
   {
      "id": 1,
      "nombre": "Andrés",
      "apellido": "Guzmán",
      "email": "a@gmail.com"

   },{
      "id": 2,
      "nombre": "Mr. John",
      "apellido": "Doe",
      "email": "doe@gmail.com"
   }
]
```

Implementamos a nuestro proyecto





The Spring Security OAuth project is deprecated. The latest OAuth 2.0 support is provided by Spring Security. See the OAuth 2.0 Migration Guide for further details.

Welcome

OAuth for Spring Security provides an OAuth implementation for Spring Security. Support is provided for the implementation of OAuth providers and OAuth consumers. There is support for Oauth 1(a) (including two-legged OAuth, a.k.a. "Signed Fetch") and for OAuth 2.0.

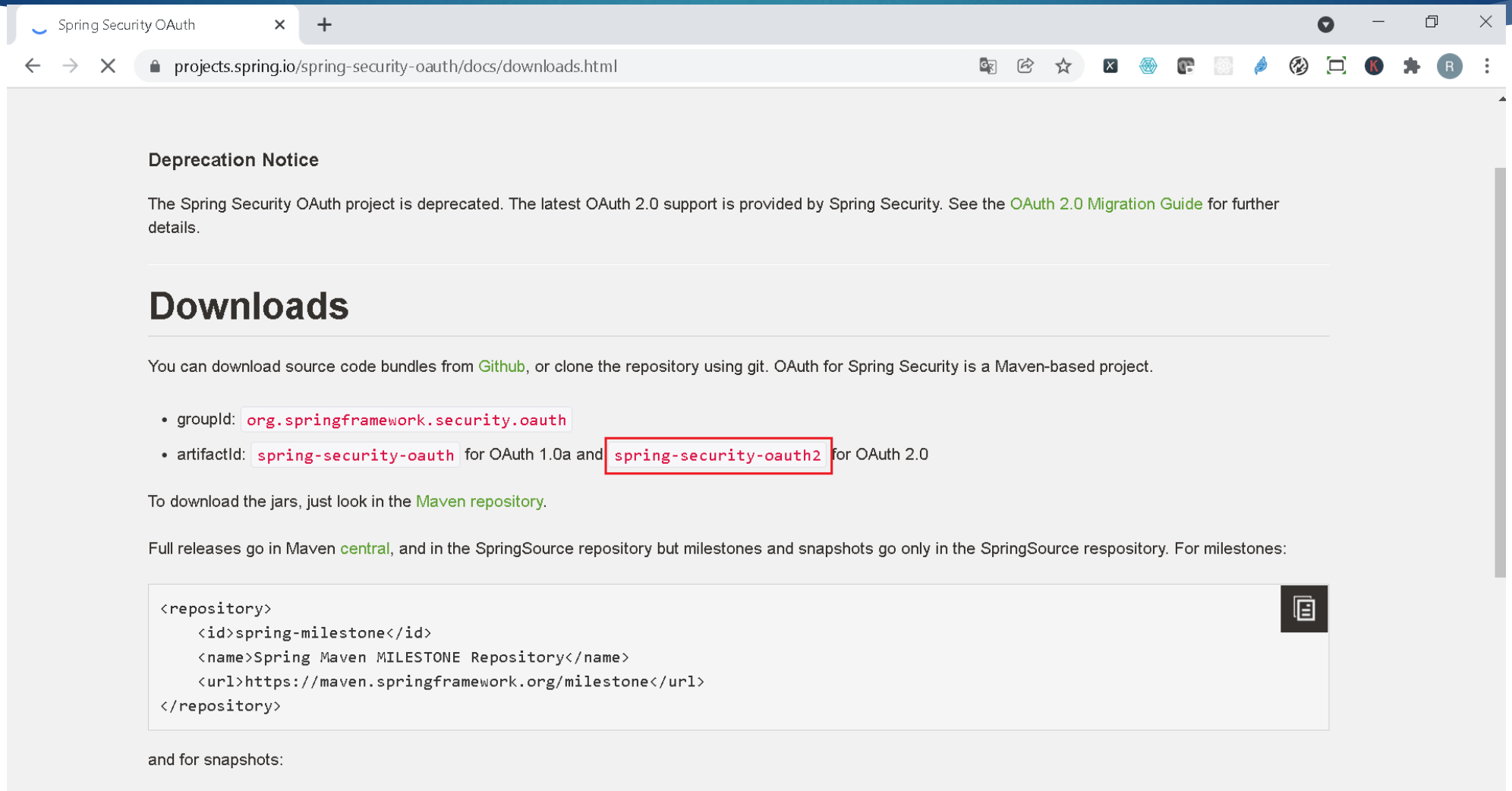
Applying security to an application is not for the faint of heart, and OAuth is no exception. Before you get started, you're going to want to make sure you understand OAuth and the problem it's designed to address. There is good documentation at the OAuth site. You will also want to make sure you understand how Spring and Spring Security work.

You're going to want to be quite familiar with both OAuth (and/or OAuth2) and Spring Security, to maximize the effectiveness of this developers guide. OAuth for Spring Security is tightly tied to both technologies, so the more familiar you are with them, the more likely you'll be to recognize the terminology and patterns that are used.

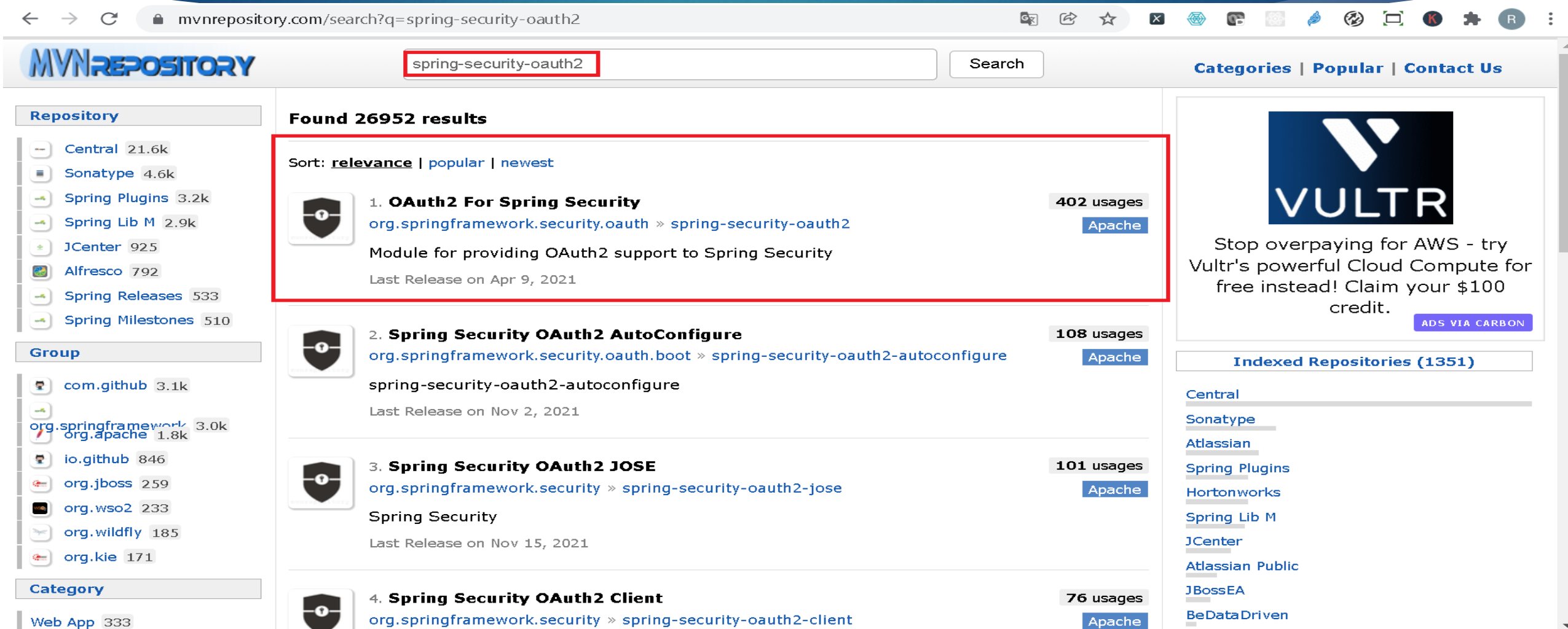
With that, you're ready to get started. Here are some useful links:

- For access to the binaries, use Maven (instructions here)
- Source code is in github at spring-projects/spring-security-oauth.
- You'll want to see OAuth for Spring Security in action, so here is a tutorial

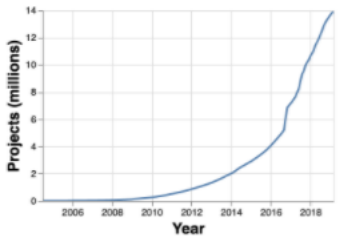
Copiamos esto



Y lo buscamos en el repositorio Maven



Indexed Artifacts (24.5M)



Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection

Home » org.springframework.security.oauth » spring-security-oauth2

OAuth2 For Spring Security

Module for providing OAuth2 support to Spring Security

License	Apache 2.0
Categories	OAuth Libraries
Tags	security spring authentication oauth
Used By	402 artifacts

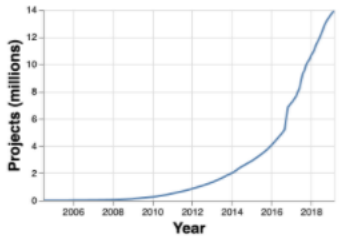
Central (54) | Spring Releases (1) | Spring Plugins (20) | ICM (2) | EBIPublic (1) | SpringFramework (2)

	Version	Repository	Usages	Date
2.5.x	2.5.1.RELEASE	Central	13	Apr, 2021
	2.5.0.RELEASE	Central	30	May, 2020
2.4.x	2.4.2.RELEASE	Central	0	Oct, 2021
	2.4.1.RELEASE	Central	11	Apr, 2020
	2.4.0.RELEASE	Central	28	Nov, 2019
	2.3.8.RELEASE	Central	25	Nov, 2019
	2.3.7.RELEASE	Central	18	Oct, 2019

Indexed Repositories (1351)

- Central
- Sonatype
- Atlassian
- Spring Plugins
- Hortonworks
- Spring Lib M
- JCenter
- Atlassian Public
- JBossEA
- BeDataDriven

Indexed Artifacts (24.5M)



Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection

Home » org.springframework.security.oauth » spring-security-oauth2 » 2.5.1.RELEASE

OAuth2 For Spring Security » 2.5.1.RELEASE

Module for providing OAuth2 support to Spring Security

License	Apache 2.0
Categories	OAuth Libraries
Date	(Apr 09, 2021)
Files	jar (491 KB) View All
Repositories	Central Scala-SBT
Used By	402 artifacts

Maven | Gradle | Gradle (Short) | Gradle (Kotlin) | SBT | Ivy | Grape | Leiningen | Buildr

```
<!-- https://mvnrepository.com/artifact/org.springframework.security.oauth/spring-security-oauth2 -->
<dependency>
    <groupId>org.springframework.security.oauth</groupId>
    <artifactId>spring-security-oauth2</artifactId>
    <version>2.5.1.RELEASE</version>
</dependency>
```

Include comment with link to declaration

Copied to clipboard!

Indexed Repositories (1351)

- Central
- Sonatype
- Atlassian
- Spring Plugins
- Hortonworks
- Spring Lib M
- JCenter
- Atlassian Public
- JBossEA
- BeDataDriven

mvnrepository.com/search?q=spring+security+jwt

MVNREPOSITORY

spring security jwt

Search

Categories | Popular | Contact Us

Repository

- Central 21.6k
- Sonatype 4.5k
- Spring Plugins 3.2k
- Spring Lib M 2.9k
- JCenter 883
- Alfresco 792
- Spring Releases 533
- Spring Milestones 510

Group

- com.github 3.1k
- org.springframework 3.0k
- org.apache 1.8k
- io.github 845
- org.jboss 260
- org.wso2 204
- org.wildfly 192
- org.kie 171

Category

- Web App 326

Found 26910 results

Sort: **relevance** | popular | newest

1. **Spring Security JWT Library** — 128 usages
   org.springframework.security » spring-security-jwt — Apache
   Spring Security JWT is a small utility library for encoding and decoding JSON Web Tokens. It belongs to the family of Spring Security crypto libraries that handle encoding and decoding text as a general, useful thing to be able to do.
   Last Release on May 28, 2020

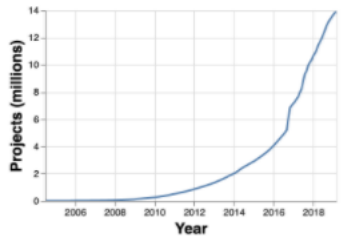
2. **Spring Context** — 11,155 usages
   org.springframework » spring-context — Apache
   Spring Context
   Last Release on Nov 11, 2021

3. **Spring Core** — 6,925 usages
   org.springframework » spring-core — Apache
   Spring Core
   Last Release on Nov 11, 2021

Indexed Repositories (1351)

- Central
- Sonatype
- Atlassian
- Spring Plugins
- Hortonworks
- Spring Lib M
- JCenter
- Atlassian Public
- JBossEA
- BeDataDriven

Indexed Artifacts (24.5M)



Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection

Home » org.springframework.security » spring-security-jwt

Spring Security JWT Library

Spring Security JWT is a small utility library for encoding and decoding JSON Web Tokens. It belongs to the family of Spring Security crypto libraries that handle encoding and decoding text as a general, useful thing to be able to do.

License	Apache 2.0
Categories	JWT Libraries
Tags	security json spring jwt
Used By	128 artifacts

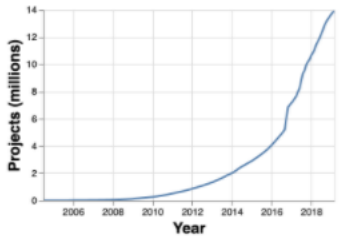
Central (14) | Spring Plugins (3) | ICM (2)

	Version	Repository	Usages	Date
1.1.x	1.1.1.RELEASE	Central	18	May, 2020
	1.1.0.RELEASE	Central	14	Nov, 2019
	1.0.11.RELEASE	Central	14	Oct, 2019
	1.0.10.RELEASE	Central	36	Jan, 2019
	1.0.9.RELEASE	Central	25	Dec, 2017
	1.0.8.RELEASE	Central	19	May, 2017
	1.0.7.RELEASE	Central	15	Nov, 2016

Indexed Repositories (1351)

- Central
- Sonatype
- Atlassian
- Spring Plugins
- Hortonworks
- Spring Lib M
- JCenter
- Atlassian Public
- JBossEA
- BeDataDriven

Indexed Artifacts (24.5M)



Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection

Spring Security JWT Library » 1.1.1.RELEASE

Spring Security JWT is a small utility library for encoding and decoding JSON Web Tokens. It belongs to the family of Spring Security crypto libraries that handle encoding and decoding text as a general, useful thing to be able to do.

License	Apache 2.0
Categories	JWT Libraries
Organization	SpringSource
HomePage	https://github.com/spring-projects/spring-security-oauth
Date	(May 28, 2020)
Files	jar (37 KB) View All
Repositories	Central
Used By	128 artifacts

Maven | Gradle | Gradle (Short) | Gradle (Kotlin) | SBT | Ivy | Grape | Leiningen | Buildr

```
<!-- https://mvnrepository.com/artifact/org.springframework.security/spring-security-jwt -->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-jwt</artifactId>
    <version>1.1.1.RELEASE</version>
</dependency>
```


Indexed Repositories (1351)

- Central
- Sonatype
- Atlassian
- Spring Plugins
- Hortonworks
- Spring Lib M
- JCenter
- Atlassian Public
- JBossEA
- BeDataDriven

mvnrepository.com/search?q=jaxb

MVNREPOSITORY

jaxb

Search

Categories | Popular | Contact Us

Repository

- Central 683
- Sonatype 191
- Spring Lib M 125
- Spring Plugins 120
- Redhat GA 50
- JBoss Releases 36
- IBiblio 30
- Spring Lib Release 27

Group

- org.apache 57
- org.jvnet 46
- com.helger 37
- com.sun 36
- com.github 34
- org.glassfish 31
- io.github 17
- janstey.sources 17

Category

Found 943 results

Sort: **relevance** | popular | newest

1. **JAXB API**
   javax.xml.bind » jaxb-api
   JAXB API
   Last Release on Aug 30, 2018
   4,741 usages
   CDDL

2. **Old JAXB Runtime**
   com.sun.xml.bind » jaxb-impl
   Old JAXB Runtime module. Contains sources required for runtime processing.
   Last Release on Aug 3, 2021
   2,746 usages
   EDL

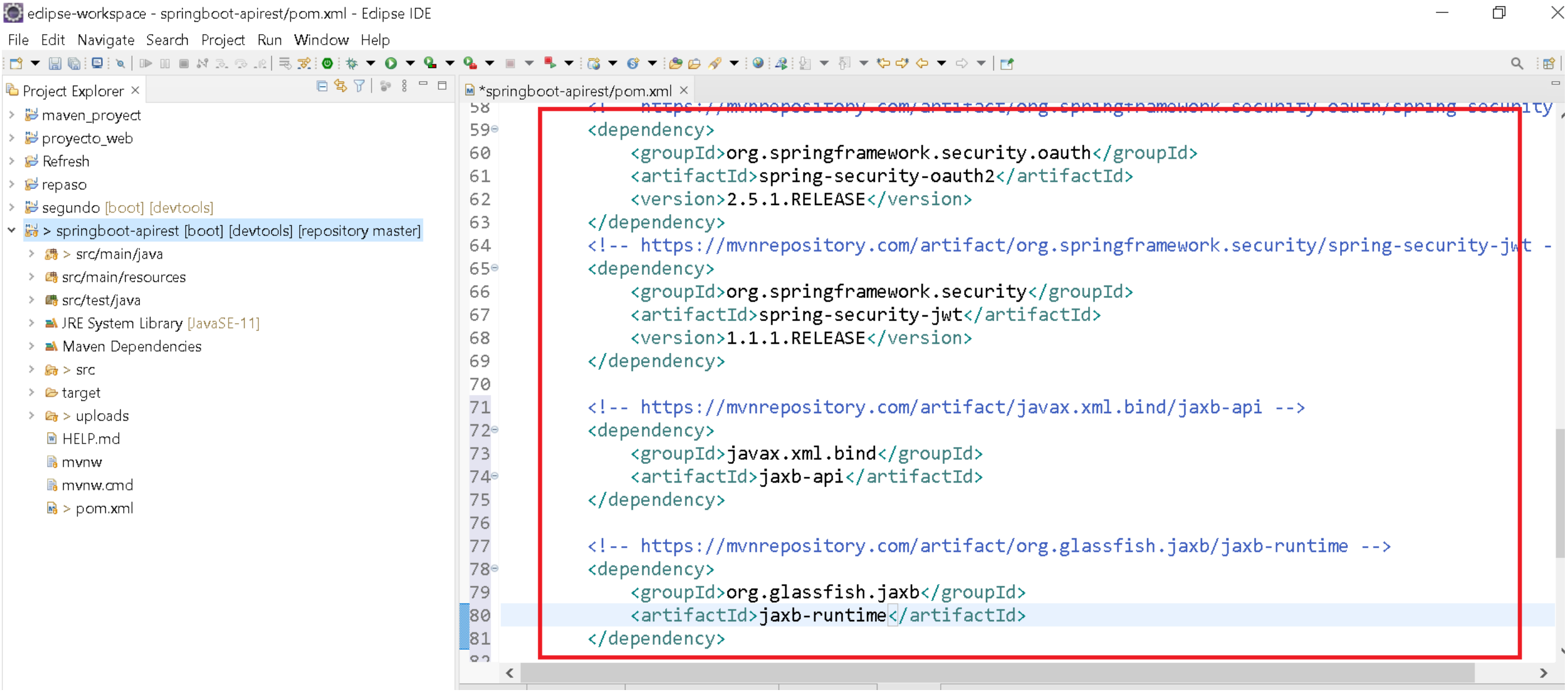
3. **JAXB Runtime**
   org.glassfish.jaxb » jaxb-runtime
   JAXB (JSR 222) Reference Implementation
   Last Release on Aug 3, 2021
   1,549 usages
   EDL

4. **Old JAXB Core**
   sun.xml.bind » jaxb-core
   1,149 usages
   EDL

Indexed Repositories (1351)

- Central
- Sonatype
- Atlassian
- Spring Plugins
- Hortonworks
- Spring Lib M
- JCenter
- Atlassian Public
- JBossEA
- BeDataDriven

Para jaxb-api y jaxb-runtime no necesitamos especificar la version



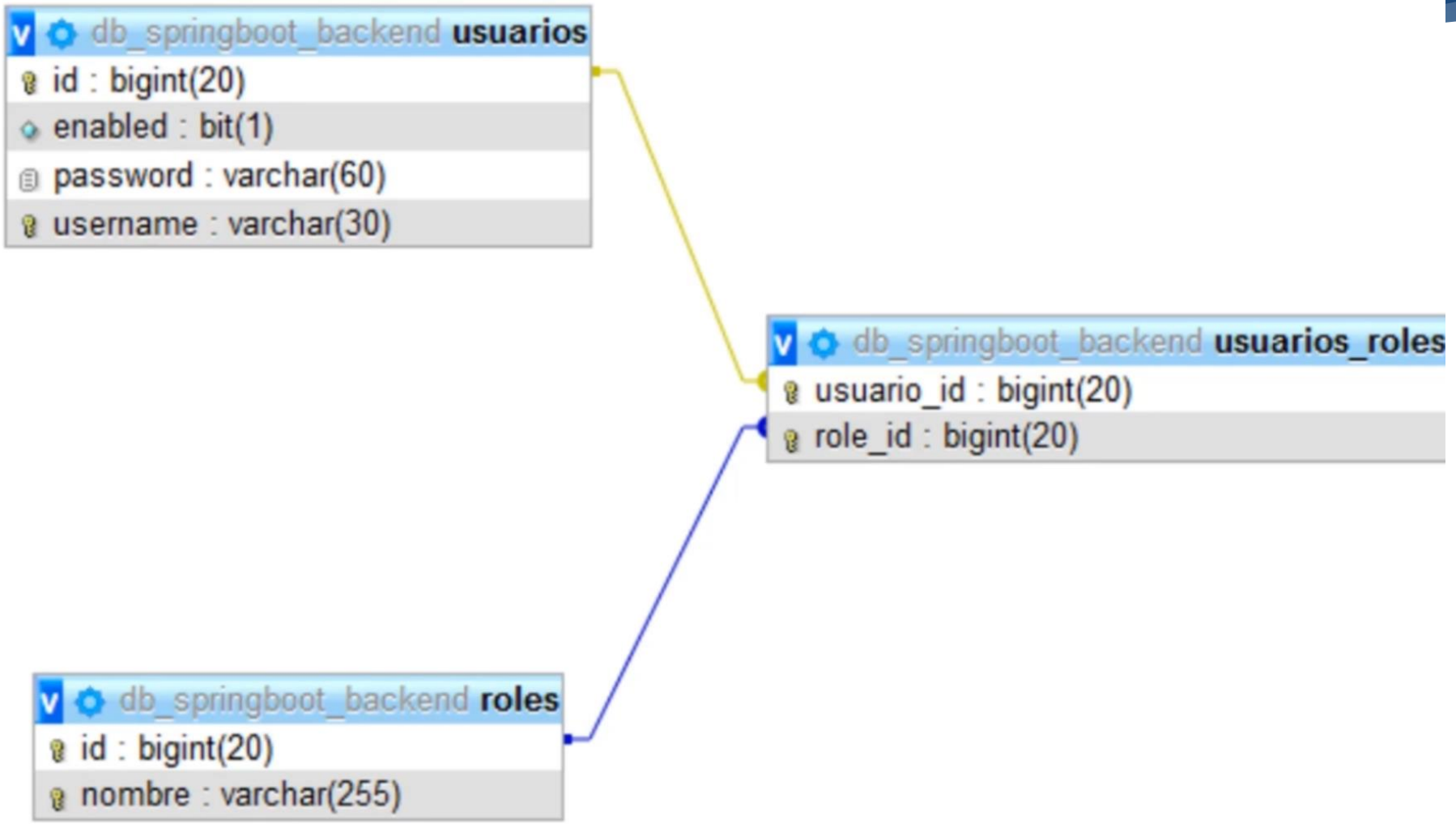
Actualizamos las dependencias

UsuarioService.java | springboot-apirest/pom.xml

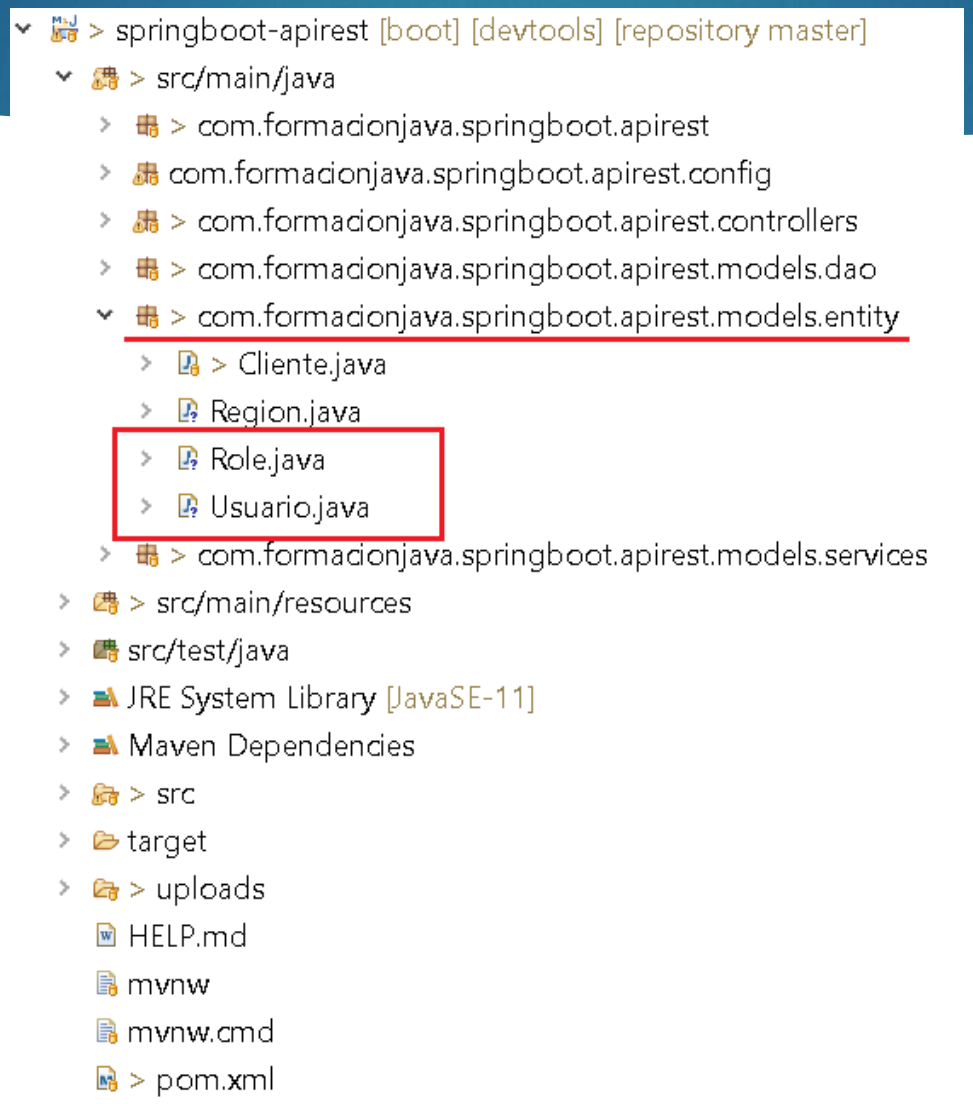
```
59        <!-- https://mvnrepository.com/artifact/org.springframework.boot/spri
60        <dependency>
61            <groupId>org.springframework.boot</groupId>
62            <artifactId>spring-boot-starter-validation</artifactId>
63        </dependency>
64
65                <!-- https://mvnrepository.com/artifact/org.springframework.s
66        <dependency>
67            <groupId>org.springframework.security.oauth</groupId>
68            <artifactId>spring-security-oauth2</artifactId>
69            <version>2.5.1.RELEASE</version>
70        </dependency>
71        <!-- https://mvnrepository.com/artifact/org.springframework.security/
72        <dependency>
73            <groupId>org.springframework.security</groupId>
74            <artifactId>spring-security-jwt</artifactId>
75            <version>1.1.1.RELEASE</version>
76        </dependency>
77
78                <!-- https://mvnrepository.com/artifact/javax.xml.bind/jaxb-api --
79        <dependency>
80            <groupId>javax.xml.bind</groupId>
81            <artifactId>jaxb-api</artifactId>
```

```
59        <!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boo
60        <dependency>
61            <groupId>org.springframework.boot</groupId>
62            <artifactId>spring-boot-starter-validation</artifactId>
63        </dependency>
64
65                <!-- https://mvnrepository.com/artifact/org.springframework.securit
66        <dependency>
67            <groupId>org.springframework.security.oauth</groupId>
68            <artifactId>spring-security-oauth2</artifactId>
69            <version>2.3.4.RELEASE</version>
70        </dependency>
71        <dependency>
72            <groupId>org.springframework.security</groupId>
73            <artifactId>spring-security-jwt</artifactId>
74            <version>1.0.9.RELEASE</version>
75        </dependency>
76
77                <!-- https://mvnrepository.com/artifact/javax.xml.bind/jaxb-api -->
78        <dependency>
79            <groupId>javax.xml.bind</groupId>
80            <artifactId>jaxb-api</artifactId>
81        </dependency>
82
```

Debemos implementar nuevas entidades



Creamos la nuevas entidades



*Usuario.java | *Role.java | IUsuarioDao.java

```java
import java.io.Serializable;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "roles")
public class Role implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String nombre;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
```

*Usuario.java × Role.java IUsuarioDao.java

```java
package com.formacionjava.springboot.apirest.models.entity;

import java.io.Serializable;

@Entity
@Table(name = "usuarios")
public class Usuario implements Serializable{

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(unique = true, length=20)
    private String username;

    @Column(length = 60)
    private String password;
    private Boolean enabled;

    @ManyToMany(fetch = FetchType.LAZY, cascade = CascadeType.ALL)
    @JoinTable(name="usuarios_roles",joinColumns= @JoinColumn(name="usuario_id"),
    inverseJoinColumns =@JoinColumn(name="role_id"),
    uniqueConstraints = {@UniqueConstraint(columnNames= {"usuario_id","role_id"})})
    private List<Role> roles;


    public Long getId() {
        return id;
    }
```

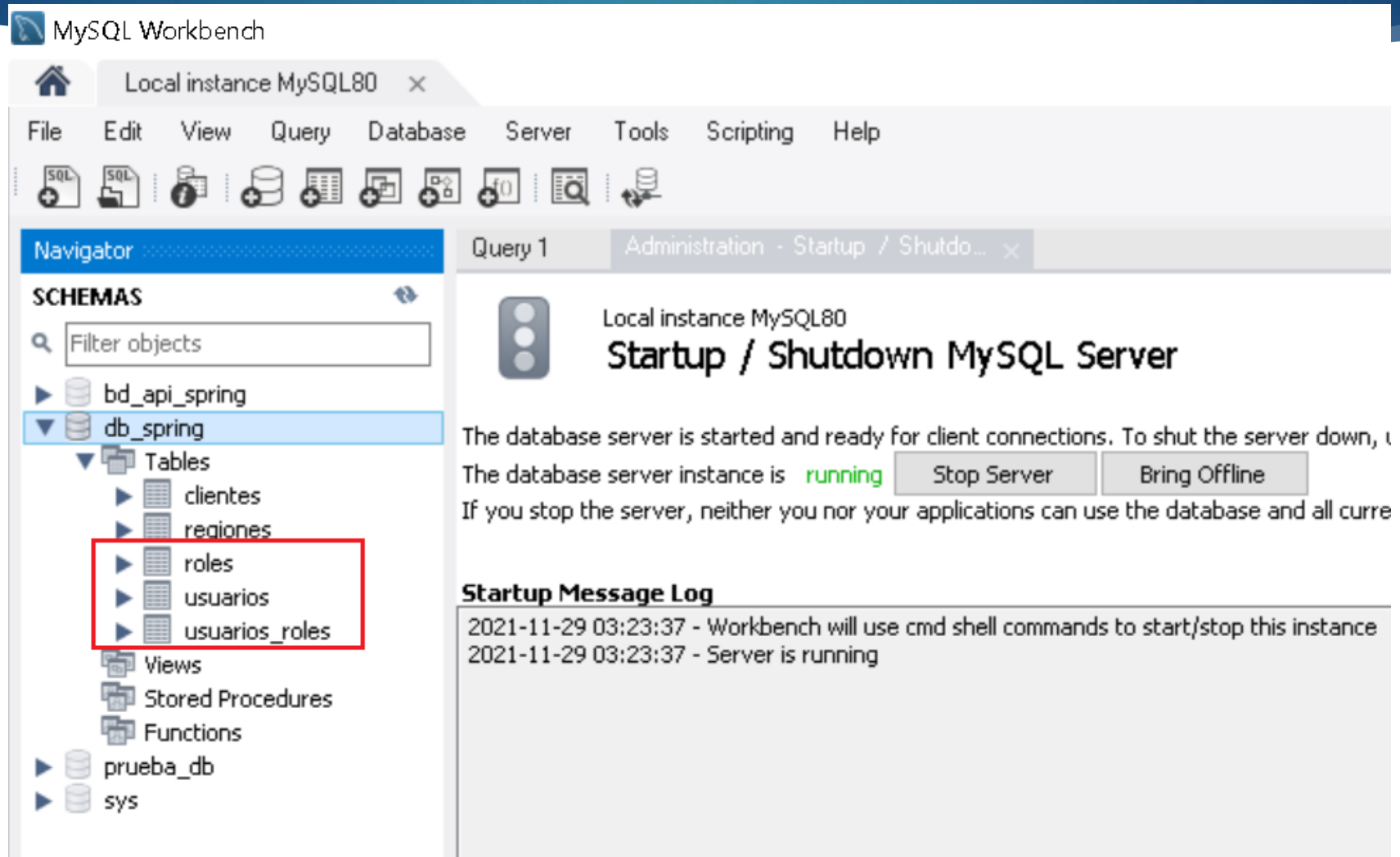
*Usuario.java × Role.java IUsuarioDao.java

```java
package com.formacionjava.springboot.apirest.models.entity;

import java.io.Serializable;
import java.util.List;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;
import javax.persistence.Table;
import javax.persistence.UniqueConstraint;

@Entity
@Table(name = "usuarios")
public class Usuario implements Serializable{

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(unique = true, length=20)
    private String username;

    @Column(length = 60)
```

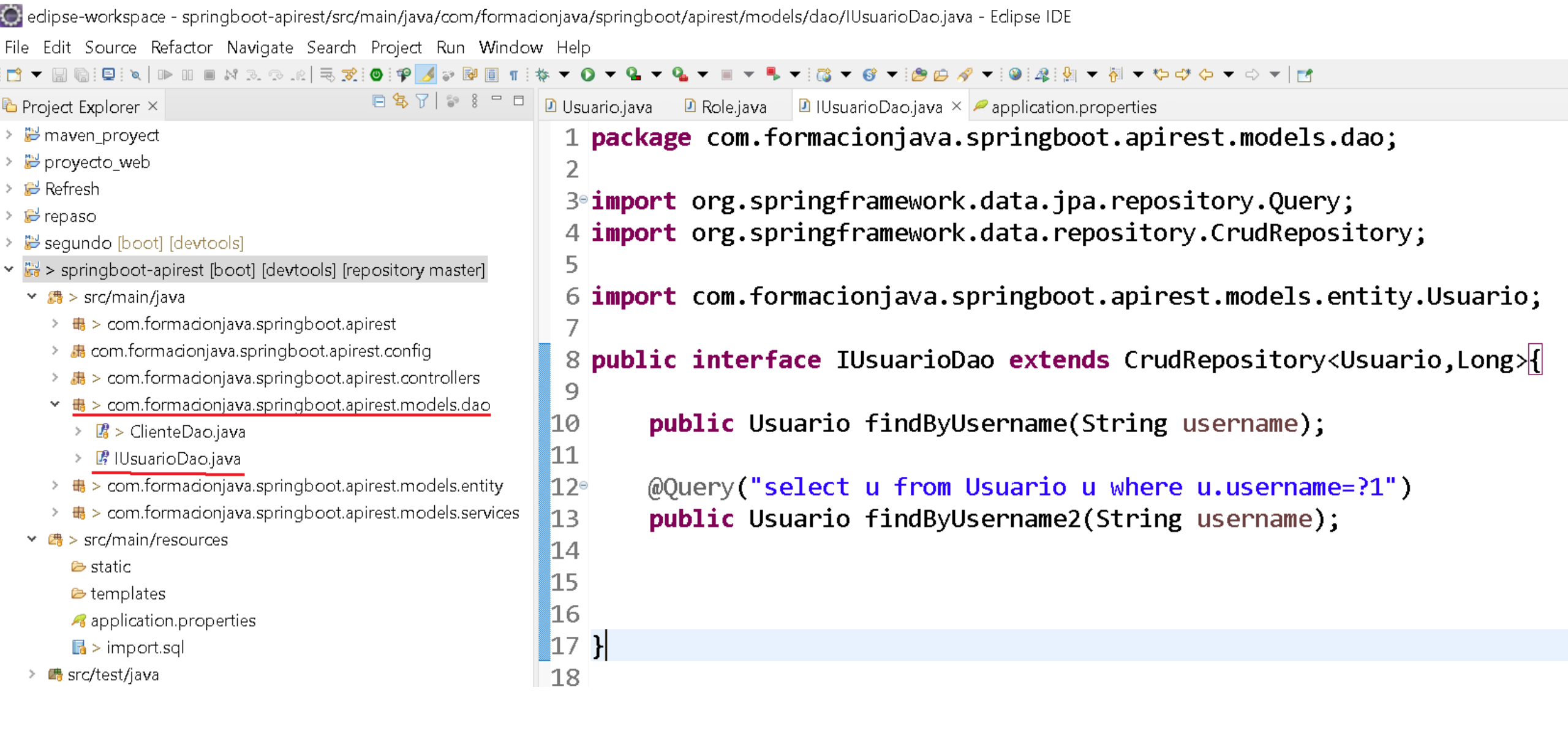
*Usuario.java × | Role.java | IUsuarioDao.java

```
41
42      public Long getId() {
43          return id;
44      }
45
46
47      public void setId(Long id) {
48          this.id = id;
49      }
50
51
52      public String getUsername() {
53          return username;
54      }
55
56
57      public void setUsername(String username) {
58          this.username = username;
59      }
60
61
62      public String getPassword() {
63          return password;
64      }
65
66
67      public void setPassword(String password) {
68          this.password = password;
69      }
70
71
72      public Boolean getEnabled() {
73          return enabled;
74      }
75
76
77      public void setEnabled(Boolean enabled) {
78          this.enabled = enabled;
79      }
80
81      private static final long serialVersionUID = 1L;
82
83  }
84
```

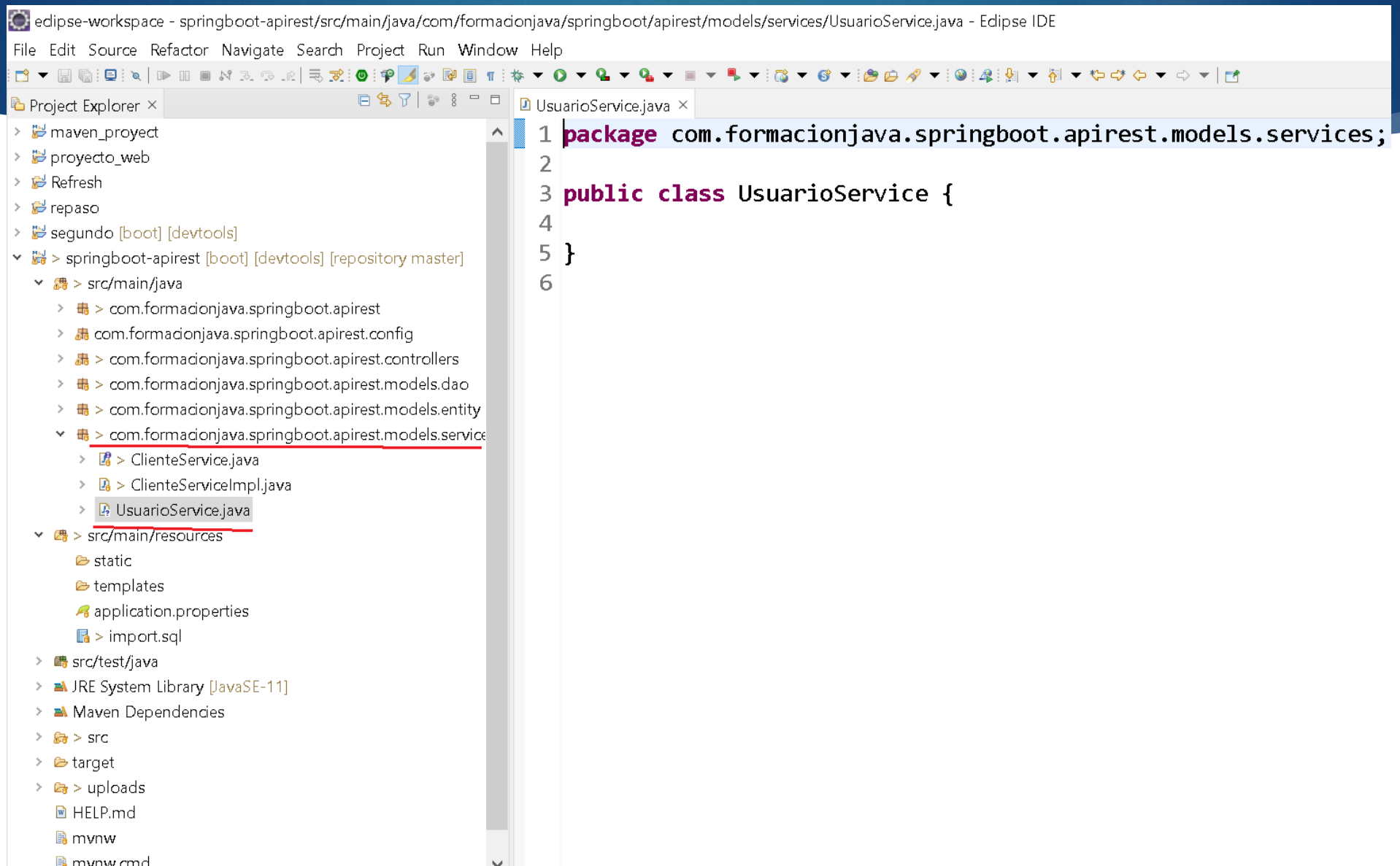
Compilamos y verificamos que nos genere las 3 tablas relacionadas



Creamos un nuevo DAO



Ahora creamos el servicio para Usuario



*UsuarioService.java

```java
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import com.formacionjava.springboot.apirest.models.dao.IUsuarioDao;
import com.formacionjava.springboot.apirest.models.entity.Usuario;

@Service
public class UsuarioService implements UserDetailsService{

    @Autowired
    private IUsuarioDao usuarioDao;


    @Override
    @Transactional(readOnly=true)
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        Usuario usuario = usuarioDao.findByUsername(username);
        return null;
    }

}

```

*UsuarioService.java

```java

@Service
public class UsuarioService implements UserDetailsService{

    private Logger logger = LoggerFactory.getLogger(UsuarioService.class);

    @Autowired
    private IUsuarioDao usuarioDao;

    @Override
    @Transactional(readOnly=true)
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        Usuario usuario = usuarioDao.findByUsername(username);


        if(usuario == null) {
            logger.error("Error en el login: no existe el usuario '"+username+"' en el sistema");
            throw new UsernameNotFoundException("Error en el login: no existe el usuario '"+username+"' en el sistema");
        }
        List<GrantedAuthority> authorities = usuario.getRoles()
                .stream()
                .map(role -> new SimpleGrantedAuthority(role.getNombre()))
                .collect(Collectors.toList());


        return new User(usuario.getUsername(), usuario.getPassword(), usuario.getEnabled(), true, true, true, authorities);
    }

}
```

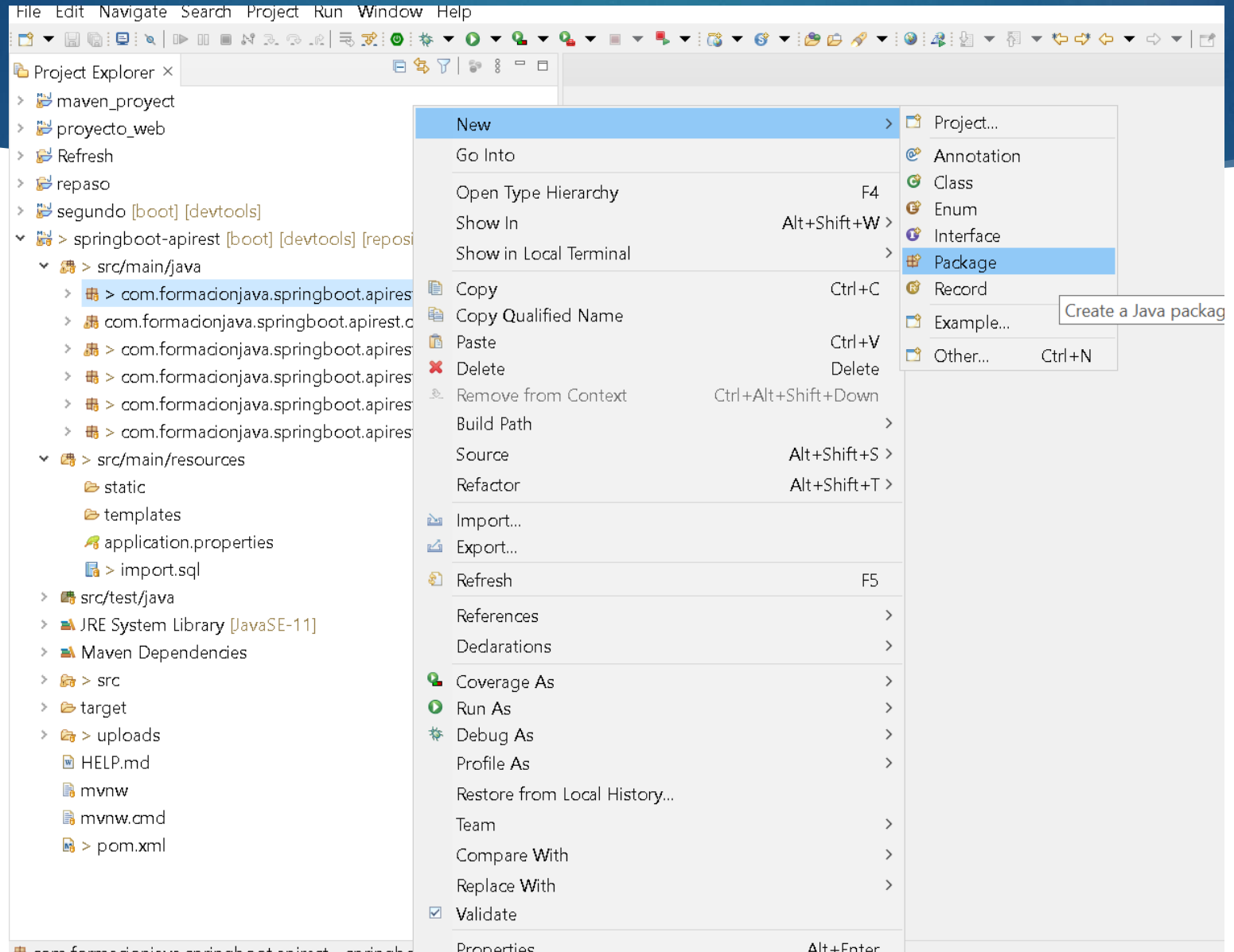
*UsuarioService.java

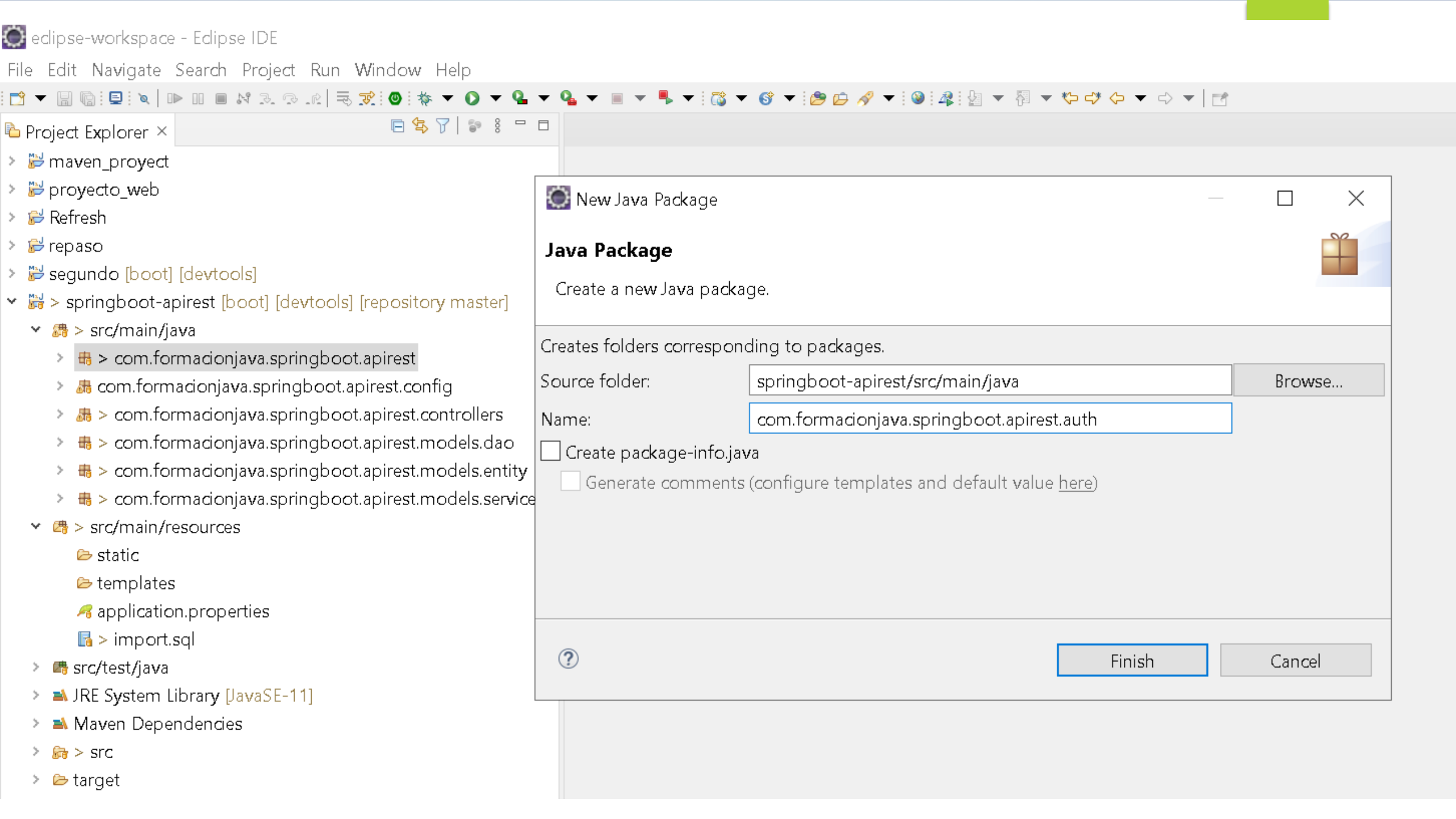
```java
package com.formacionjava.springboot.apirest.models.services;

import java.util.List;
import java.util.stream.Collectors;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.authority.SimpleGrantedAuthority;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import com.formacionjava.springboot.apirest.models.dao.IUsuarioDao;
import com.formacionjava.springboot.apirest.models.entity.Usuario;

@Service
public class UsuarioService implements UserDetailsService{

    private Logger logger = LoggerFactory.getLogger(UsuarioService.class);

    @Autowired
    private IUsuarioDao usuarioDao;

    @Override
    @Transactional(readOnly=true)
```

UsuarioService.java

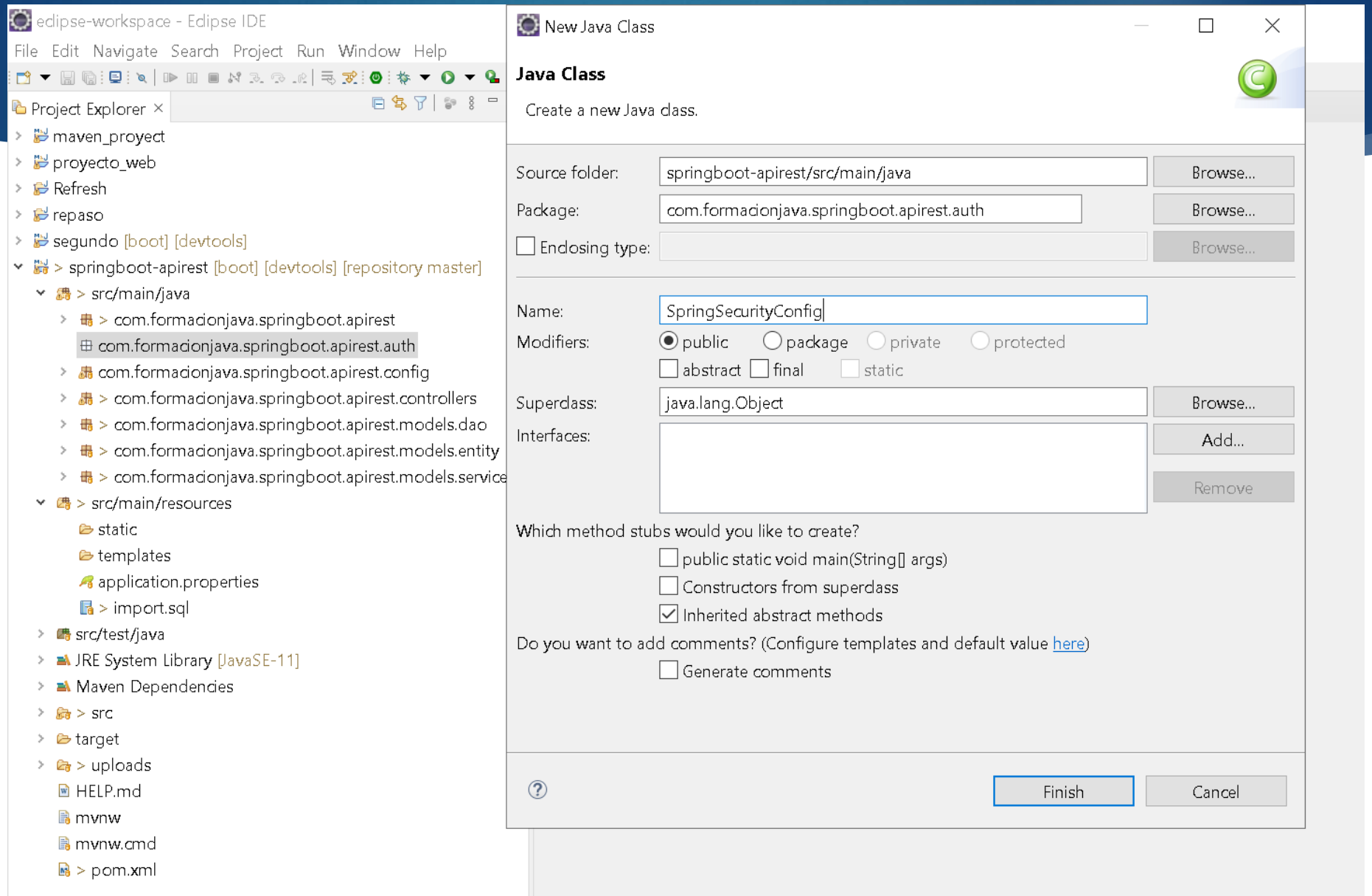
```java
@Service
public class UsuarioService implements UserDetailsService{

    private Logger logger = LoggerFactory.getLogger(UsuarioService.class);

    @Autowired
    private IUsuarioDao usuarioDao;

    @Override
    @Transactional(readOnly=true)
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        Usuario usuario = usuarioDao.findByUsername(username);


        if(usuario == null) {
            logger.error("Error en el login: no existe el usuario '"+username+"' en el sistema");
            throw new UsernameNotFoundException("Error en el login: no existe el usuario '"+username+"' en el sistema");
        }
        List<GrantedAuthority> authorities = usuario.getRoles()
                .stream()
                .map(role -> new SimpleGrantedAuthority(role.getNombre()))
                .peek(authority->logger.info("Role: "+authority.getAuthority()))
                .collect(Collectors.toList());


        return new User(usuario.getUsername(), usuario.getPassword(), usuario.getEnabled(), true, true, true, authorities);
    }

}
```

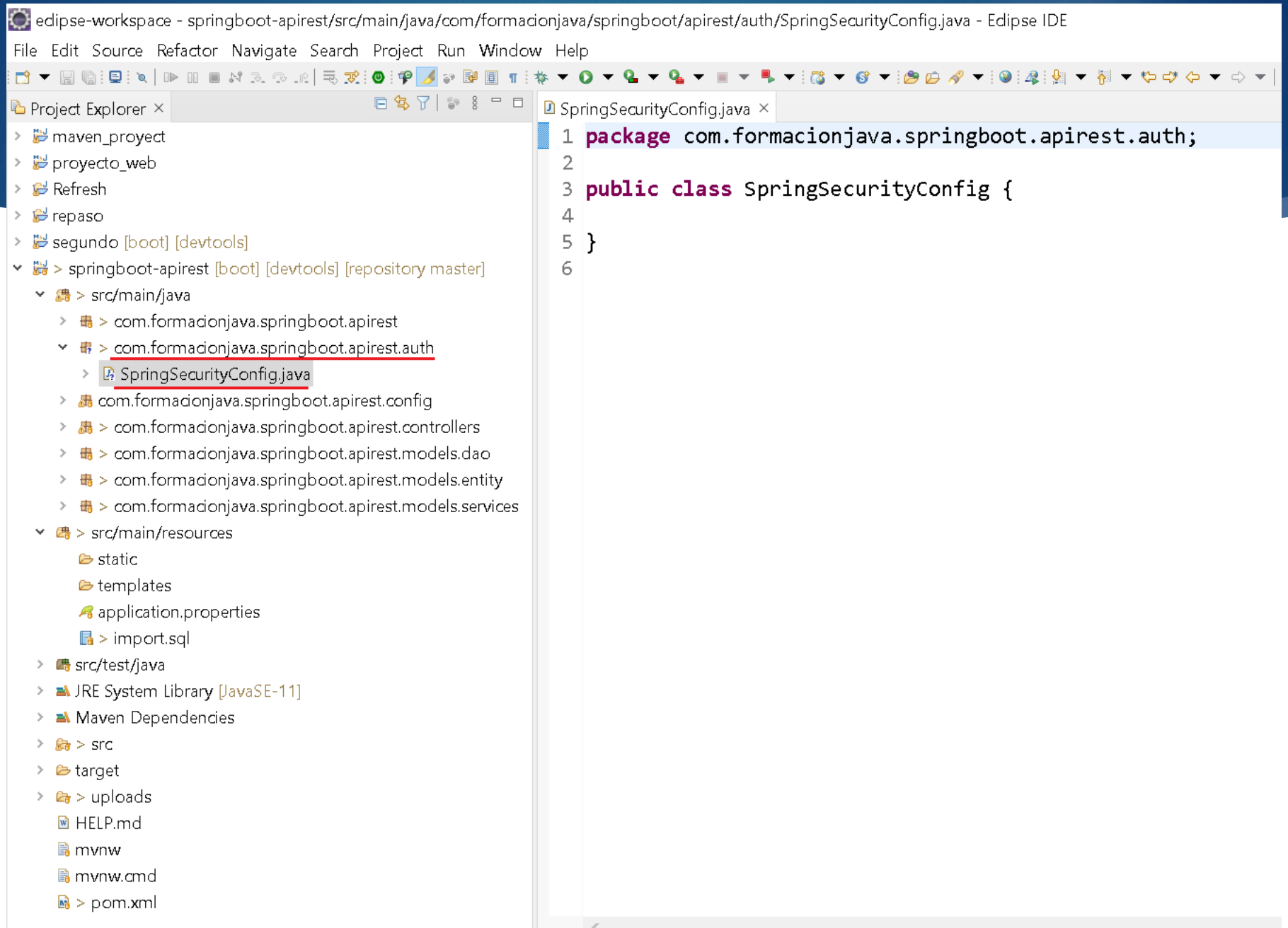
Creamos un nuevo paquete

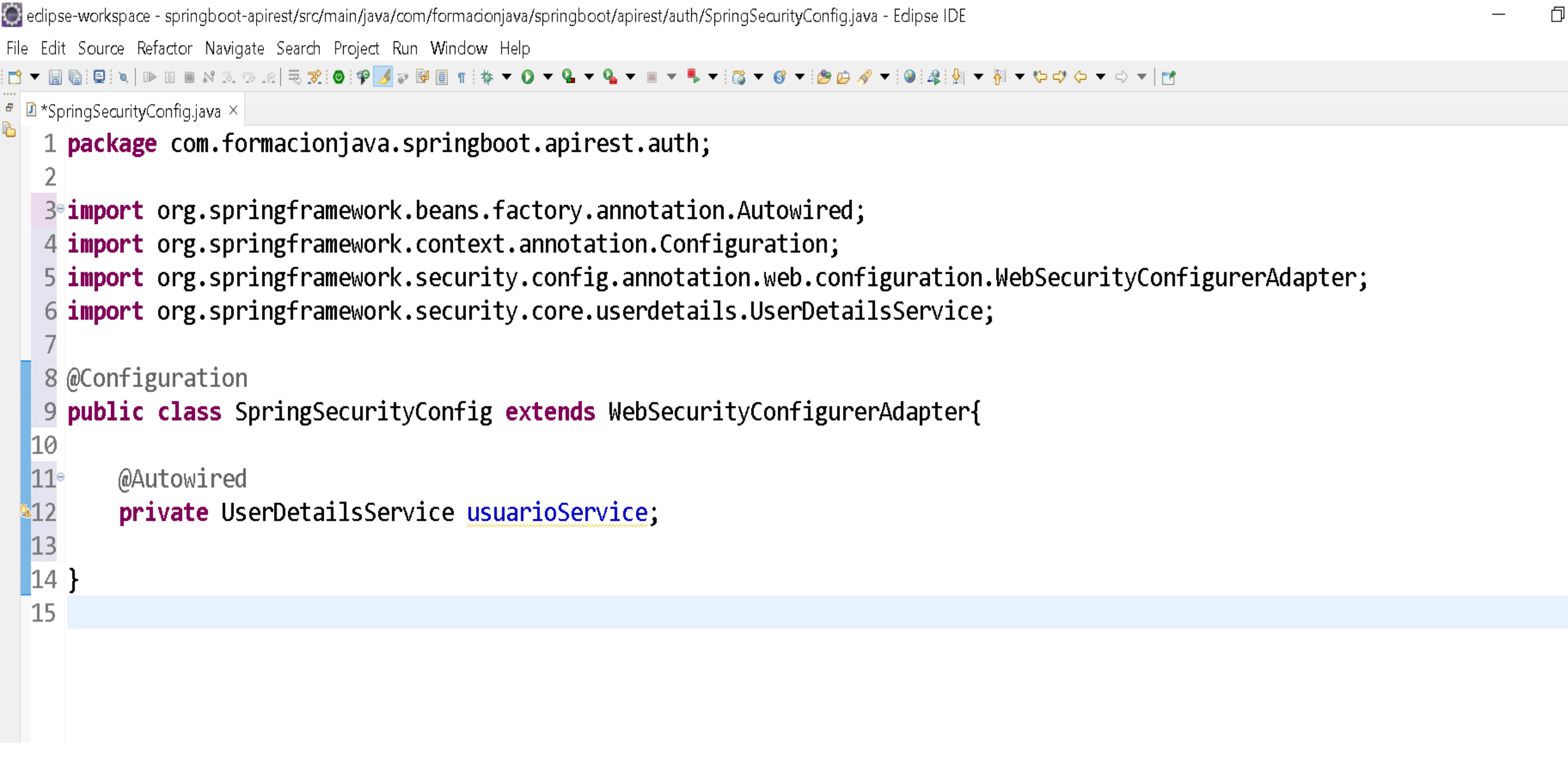


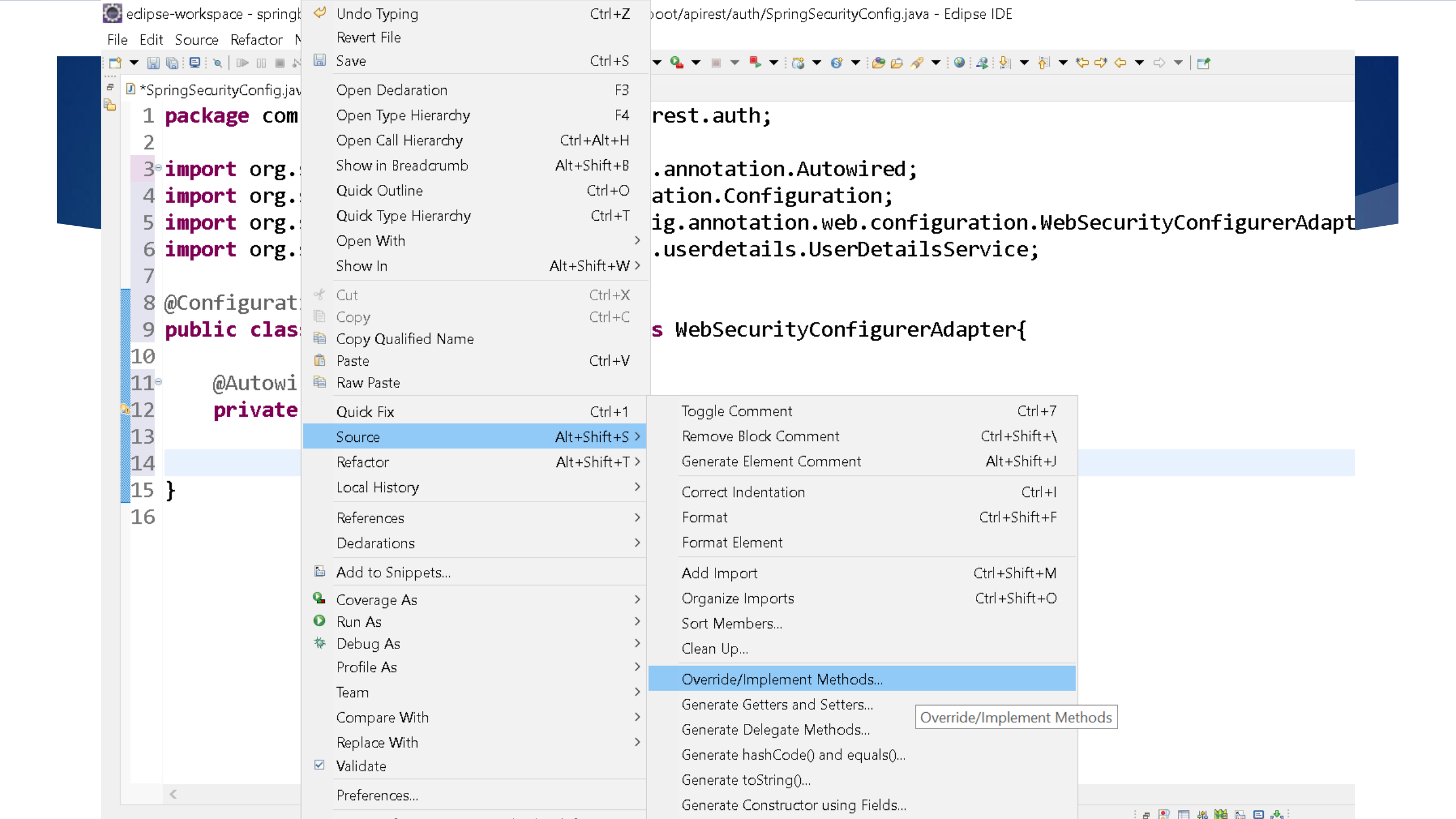

eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
maven_proyect
proyecto_web
Refresh
repaso
segundo [boot] [devtools]
> springboot-apirest [boot] [devtools] [repository master]
> src/main/java
> com.formacionjava.springboot.apirest
com.formacionjava.springboot.apirest.config
> com.formacionjava.springboot.apirest.controllers
> com.formacionjava.springboot.apirest.models.dao
> com.formacionjava.springboot.apirest.models.entity
> com.formacionjava.springboot.apirest.models.service
> src/main/resources
static
templates
application.properties
> import.sql
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
> src
target
New Java Package
Java Package
Create a new Java package.
Creates folders corresponding to packages.
Source folder:
springboot-apirest/src/main/java
Browse...
Name:
com.formacionjava.springboot.apirest.auth
Create package-info.java
Generate comments (configure templates and default value here)
Finish
Cancel

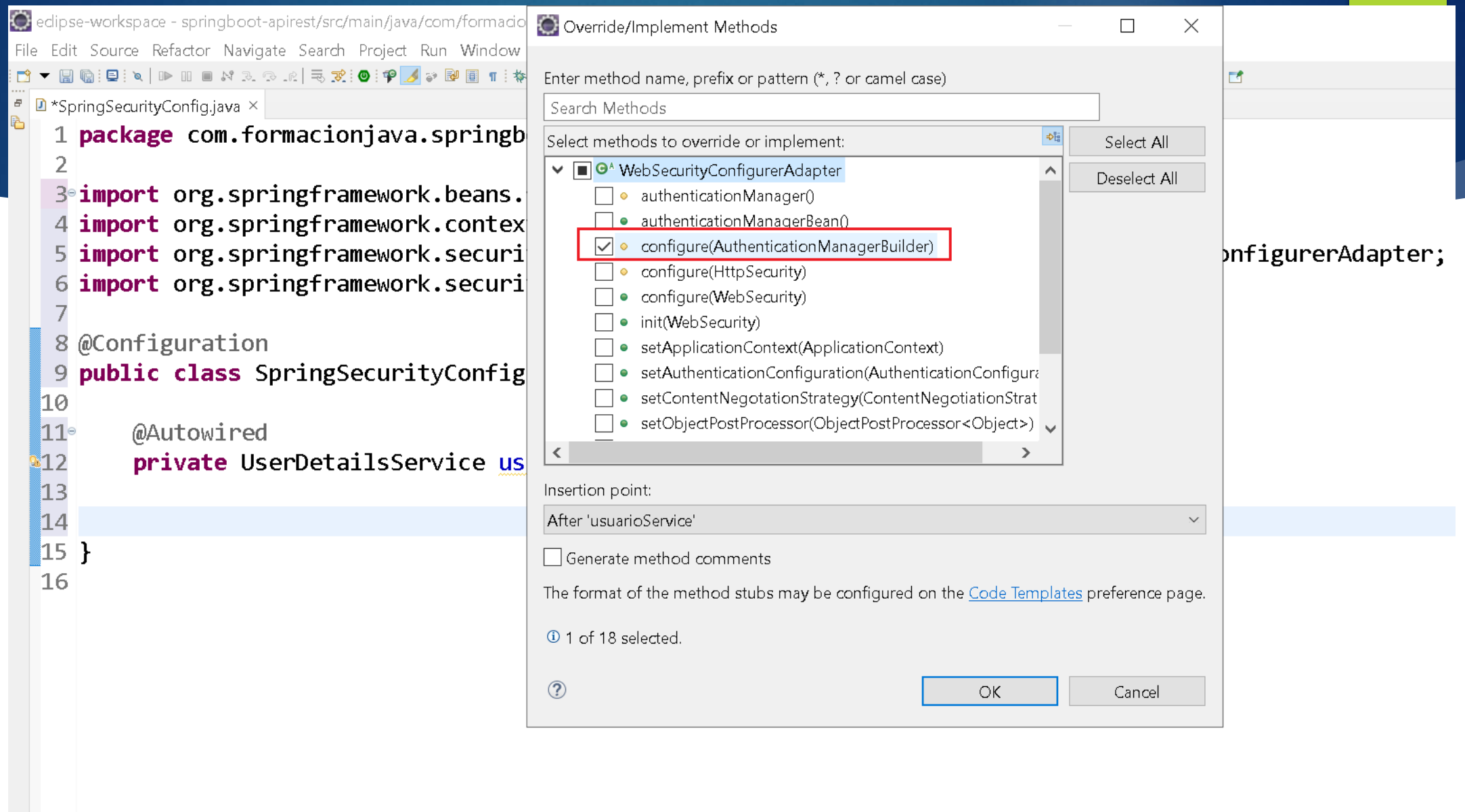
Creamos una clase

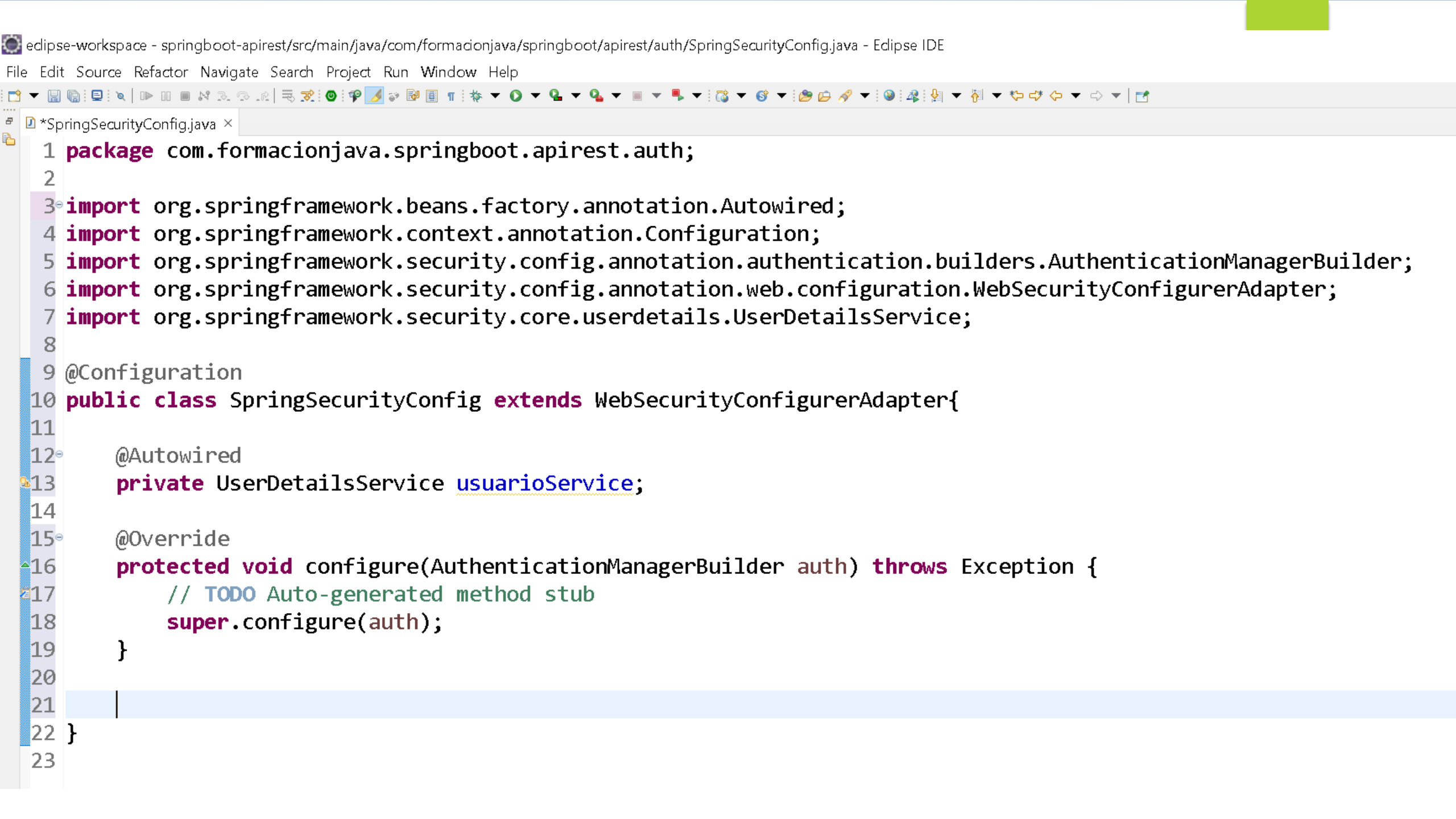



edipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/SpringSecurityConfig.java - Edipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
maven_proyect
proyecto_web
Refresh
repaso
segundo [boot] [devtools]
> springboot-apirest [boot] [devtools] [repository master]
> src/main/java
> com.formacionjava.springboot.apirest
> com.formacionjava.springboot.apirest.auth
SpringSecurityConfig.java
com.formacionjava.springboot.apirest.config
> com.formacionjava.springboot.apirest.controllers
> com.formacionjava.springboot.apirest.models.dao
> com.formacionjava.springboot.apirest.models.entity
> com.formacionjava.springboot.apirest.models.services
> src/main/resources
static
templates
application.properties
> import.sql
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
> src
target
> uploads
HELP.md
mvnw
mvnw.cmd
> pom.xml
SpringSecurityConfig.java
1 package com.formacionjava.springboot.apirest.auth;
2
3 public class SpringSecurityConfig {
4
5 }
6


eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/SpringSecurityConfig.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*SpringSecurityConfig.java
1 package com.formacionjava.springboot.apirest.auth;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.context.annotation.Configuration;
5 import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
6 import org.springframework.security.core.userdetails.UserDetailsService;
7
8 @Configuration
9 public class SpringSecurityConfig extends WebSecurityConfigurerAdapter{
10
11 @Autowired
12 private UserDetailsService usuarioService;
13
14 }
15


eclipse-workspace - springb
oot/apirest/auth/SpringSecurityConfig.java - Eclipse IDE
File Edit Source Refactor N
*SpringSecurityConfig.jav
1 package com
2
3 import org.
4 import org.
5 import org.
6 import org.
7
8 @Configurat
9 public clas
10
11 @Autowi
12 private
13
14
15 }
16
rest.auth;
.annotation.Autowired;
ation.Configuration;
ig.annotation.web.configuration.WebSecurityConfigurerAdapt
.userdetails.UserDetailsService;
s WebSecurityConfigurerAdapter{
Undo Typing Ctrl+Z
Revert File
Save Ctrl+S
Open Declaration F3
Open Type Hierarchy F4
Open Call Hierarchy Ctrl+Alt+H
Show in Breadcrumb Alt+Shift+B
Quick Outline Ctrl+O
Quick Type Hierarchy Ctrl+T
Open With
Show In Alt+Shift+W
Cut Ctrl+X
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Raw Paste
Quick Fix Ctrl+1
Source Alt+Shift+S
Refactor Alt+Shift+T
Local History
References
Declarations
Add to Snippets...
Coverage As
Run As
Debug As
Profile As
Team
Compare With
Replace With
Validate
Preferences...
Toggle Comment Ctrl+7
Remove Block Comment Ctrl+Shift+\
Generate Element Comment Alt+Shift+J
Correct Indentation Ctrl+I
Format Ctrl+Shift+F
Format Element
Add Import Ctrl+Shift+M
Organize Imports Ctrl+Shift+O
Sort Members...
Clean Up...
Override/Implement Methods...
Generate Getters and Setters...
Generate Delegate Methods...
Generate hashCode() and equals()...
Generate toString()...
Generate Constructor using Fields...
Override/Implement Methods

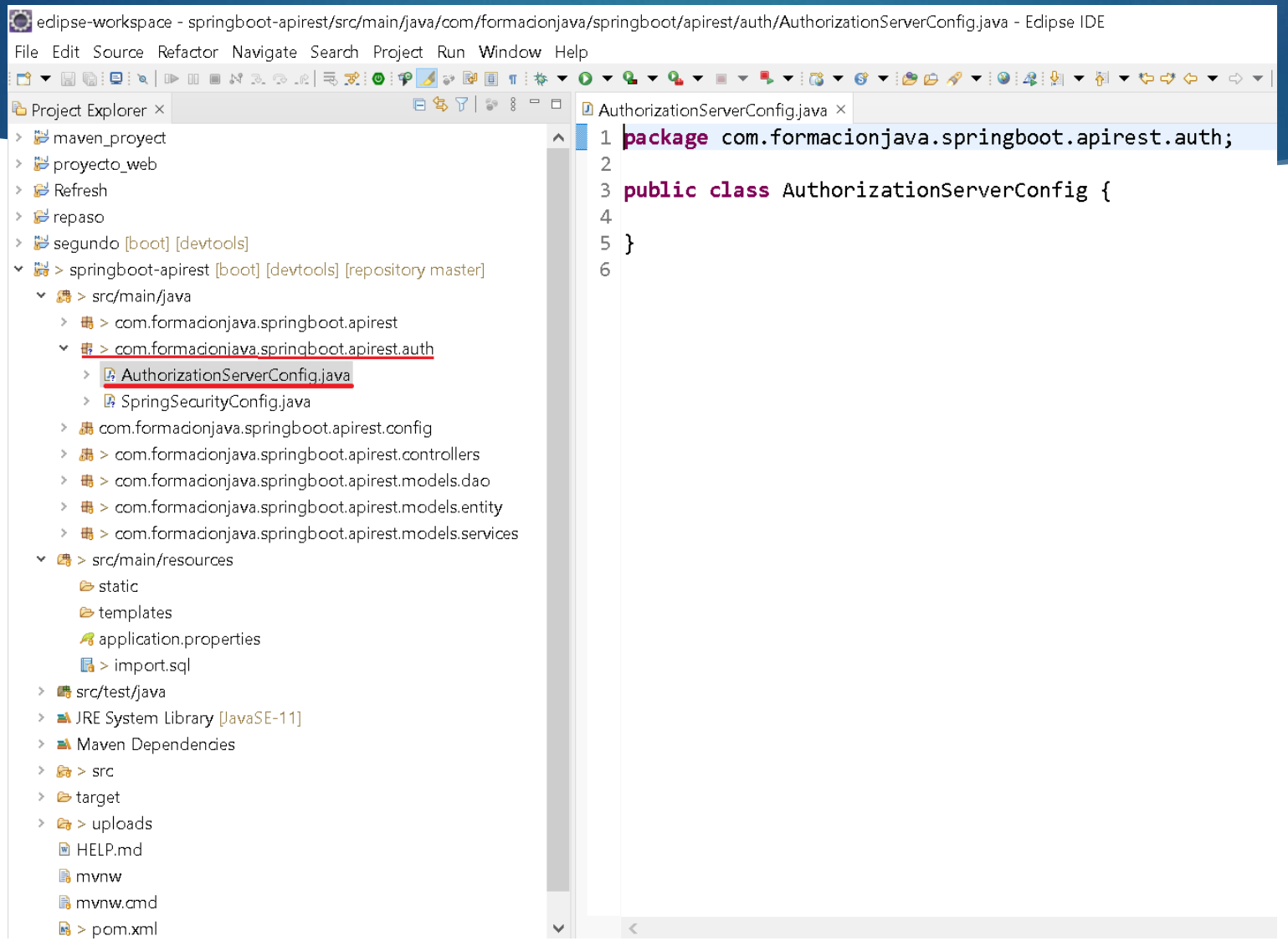

eclipse-workspace - springboot-apirest/src/main/java/com/formacio
File Edit Source Refactor Navigate Search Project Run Window
*SpringSecurityConfig.java
1 package com.formacionjava.springb
2
3 import org.springframework.beans.
4 import org.springframework.contex
5 import org.springframework.securi
6 import org.springframework.securi
7
8 @Configuration
9 public class SpringSecurityConfig
10
11 @Autowired
12 private UserDetailsService us
13
14
15 }
16
onfigurerAdapter;
Override/Implement Methods
Enter method name, prefix or pattern (*, ? or camel case)
Search Methods
Select methods to override or implement:
Select All
Deselect All
WebSecurityConfigurerAdapter
authenticationManager()
authenticationManagerBean()
configure(AuthenticationManagerBuilder)
configure(HttpSecurity)
configure(WebSecurity)
init(WebSecurity)
setApplicationContext(ApplicationContext)
setAuthenticationConfiguration(AuthenticationConfigura
setContentNegotationStrategy(ContentNegotiationStrat
setObjectPostProcessor(ObjectPostProcessor<Object>)
Insertion point:
After 'usuarioService'
Generate method comments
The format of the method stubs may be configured on the Code Templates preference page.
1 of 18 selected.
OK
Cancel

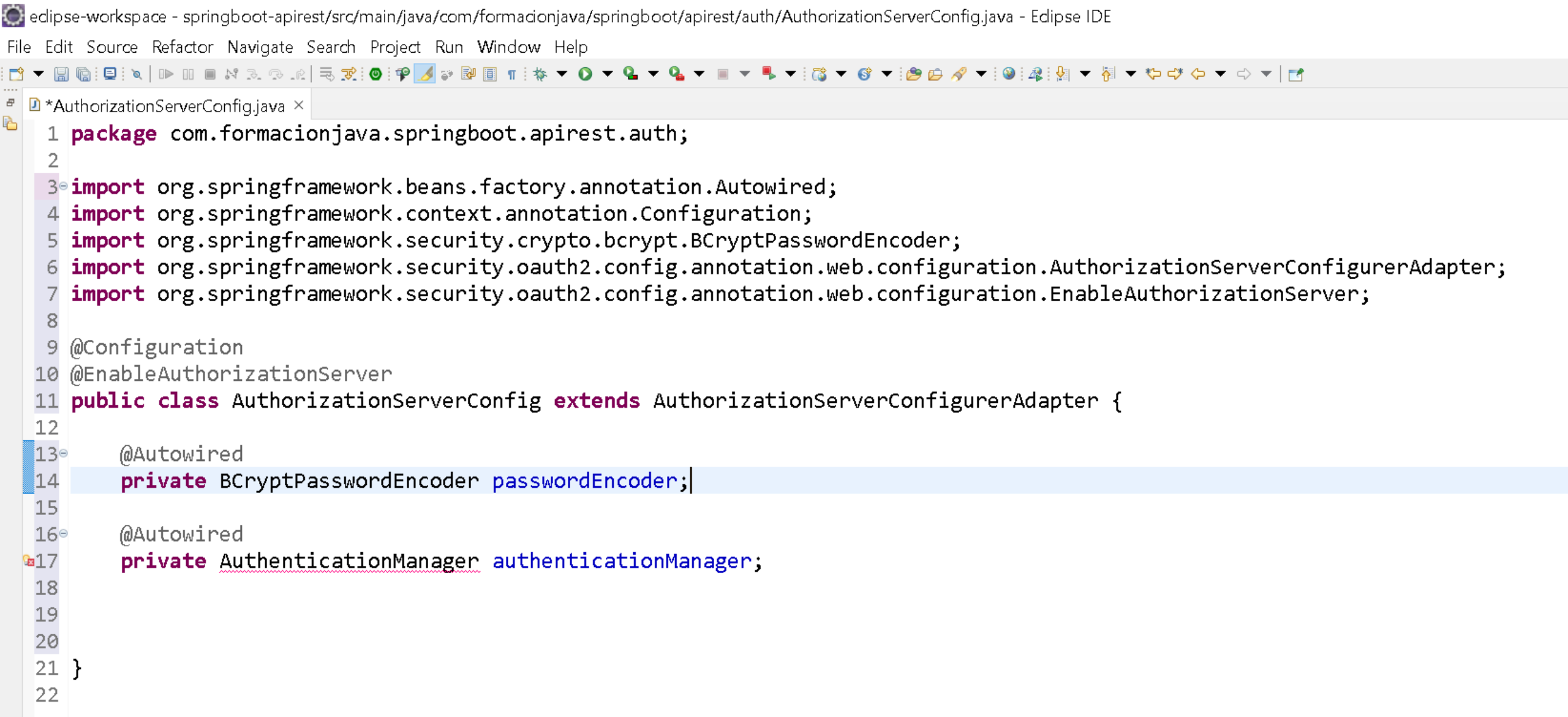

eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/SpringSecurityConfig.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*SpringSecurityConfig.java
1 package com.formacionjava.springboot.apirest.auth;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.context.annotation.Configuration;
5 import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
6 import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
7 import org.springframework.security.core.userdetails.UserDetailsService;
8
9 @Configuration
10 public class SpringSecurityConfig extends WebSecurityConfigurerAdapter{
11
12 @Autowired
13 private UserDetailsService usuarioService;
14
15 @Override
16 protected void configure(AuthenticationManagerBuilder auth) throws Exception {
17 // TODO Auto-generated method stub
18 super.configure(auth);
19 }
20
21
22 }
23

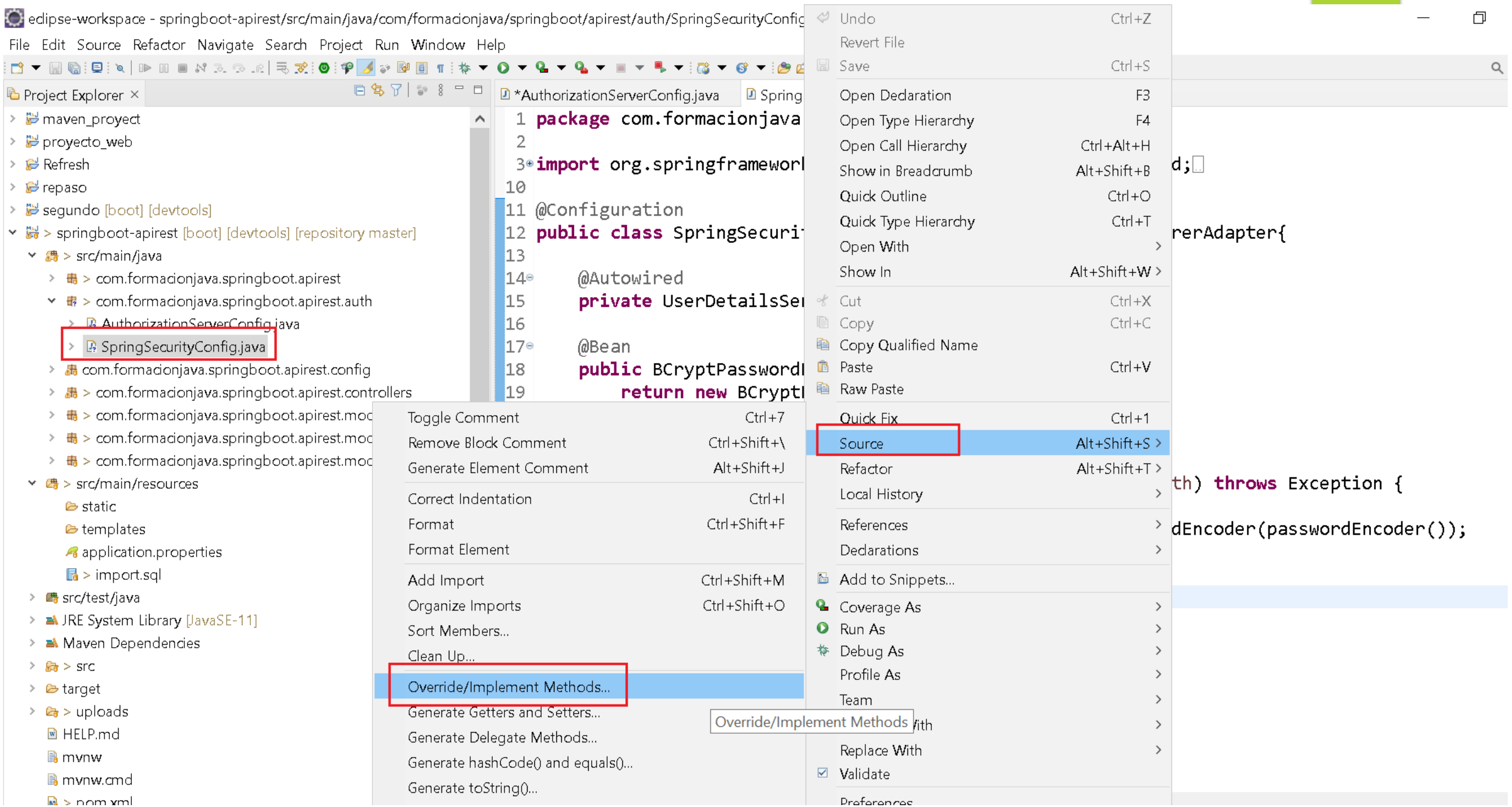
*SpringSecurityConfig.java

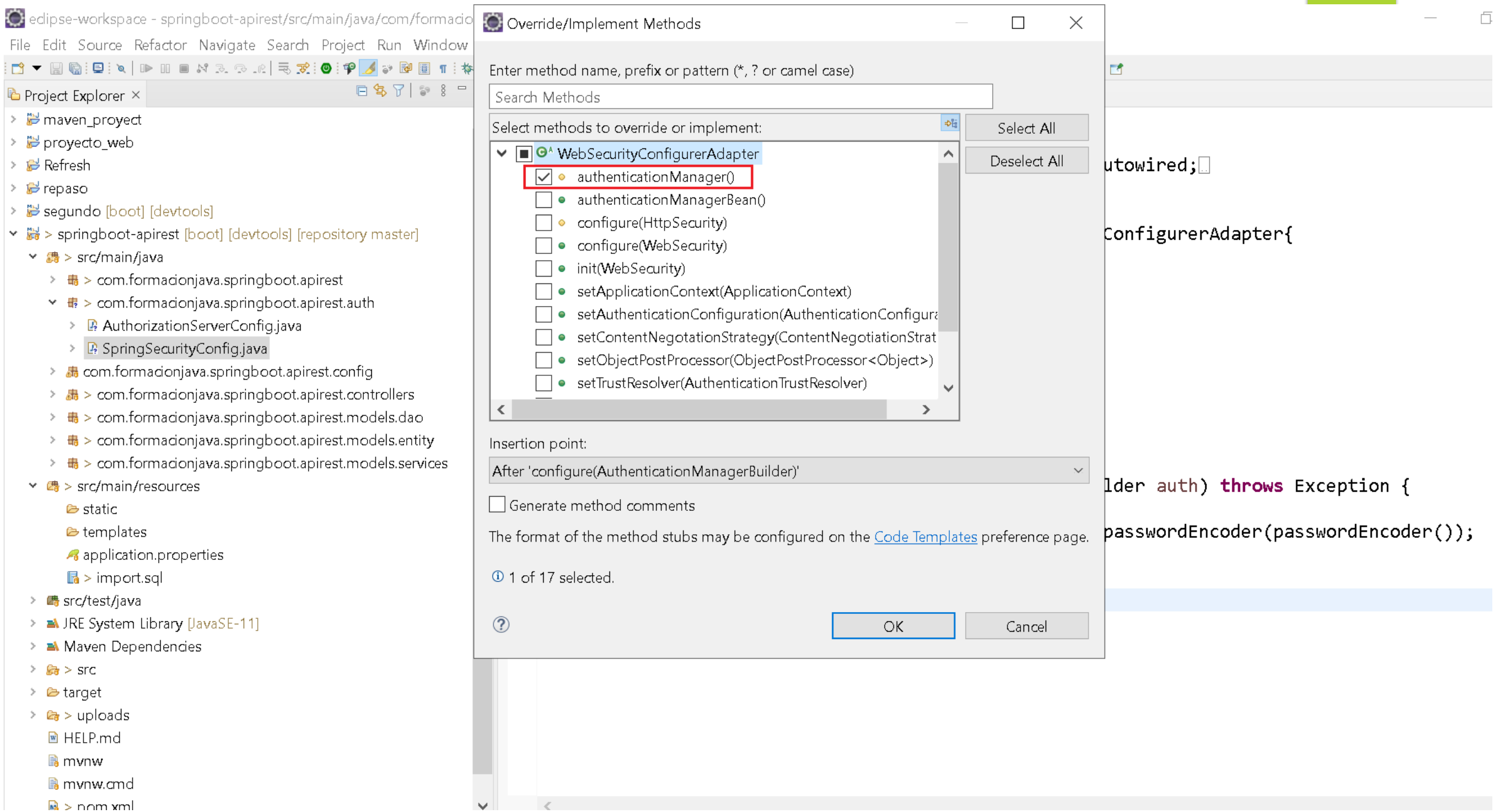
```java
package com.formacionjava.springboot.apirest.auth;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

@Configuration
public class SpringSecurityConfig extends WebSecurityConfigurerAdapter{

    @Autowired
    private UserDetailsService usuarioService;

    @Bean
    public BCryptPasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {

        auth.userDetailsService(this.usuarioService).passwordEncoder(passwordEncoder());
    }


}
```

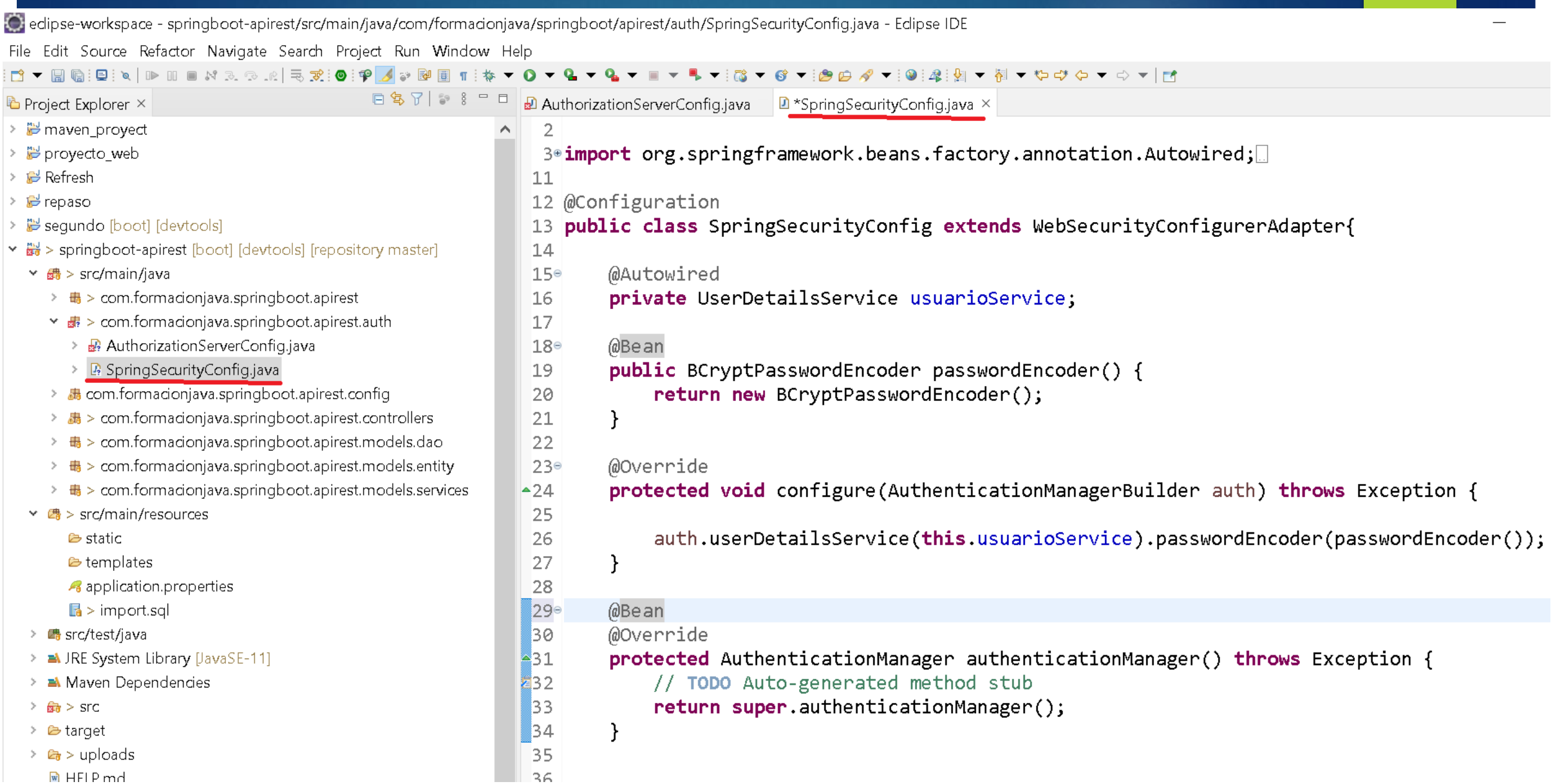
Configuración para el servidor de autorización

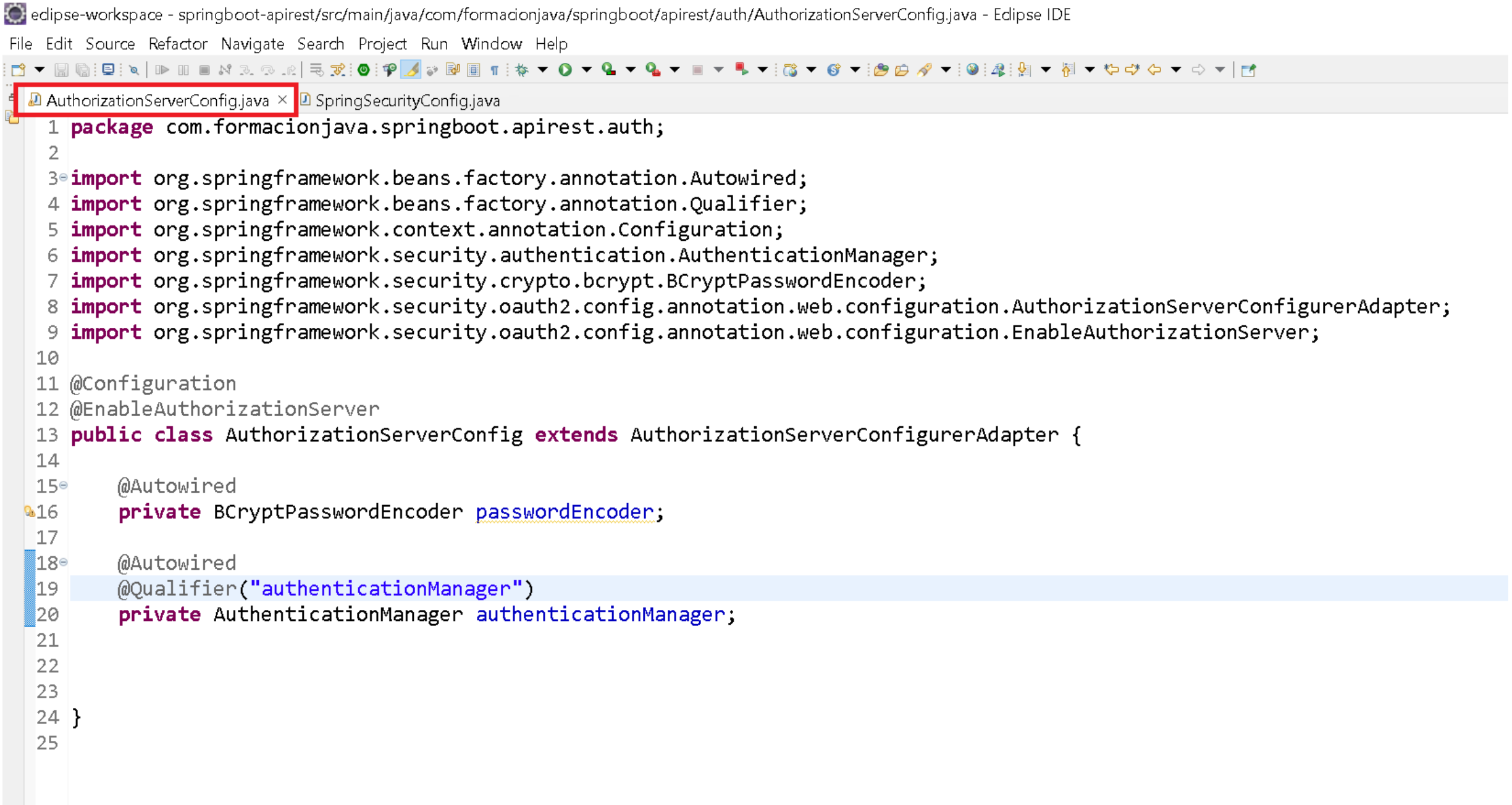



eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/AuthorizationServerConfig.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*AuthorizationServerConfig.java
1 package com.formacionjava.springboot.apirest.auth;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.context.annotation.Configuration;
5 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
6 import org.springframework.security.oauth2.config.annotation.web.configuration.AuthorizationServerConfigurerAdapter;
7 import org.springframework.security.oauth2.config.annotation.web.configuration.EnableAuthorizationServer;
8
9 @Configuration
10 @EnableAuthorizationServer
11 public class AuthorizationServerConfig extends AuthorizationServerConfigurerAdapter {
12
13 @Autowired
14 private BCryptPasswordEncoder passwordEncoder;
15
16 @Autowired
17 private AuthenticationManager authenticationManager;
18
19
20
21 }
22

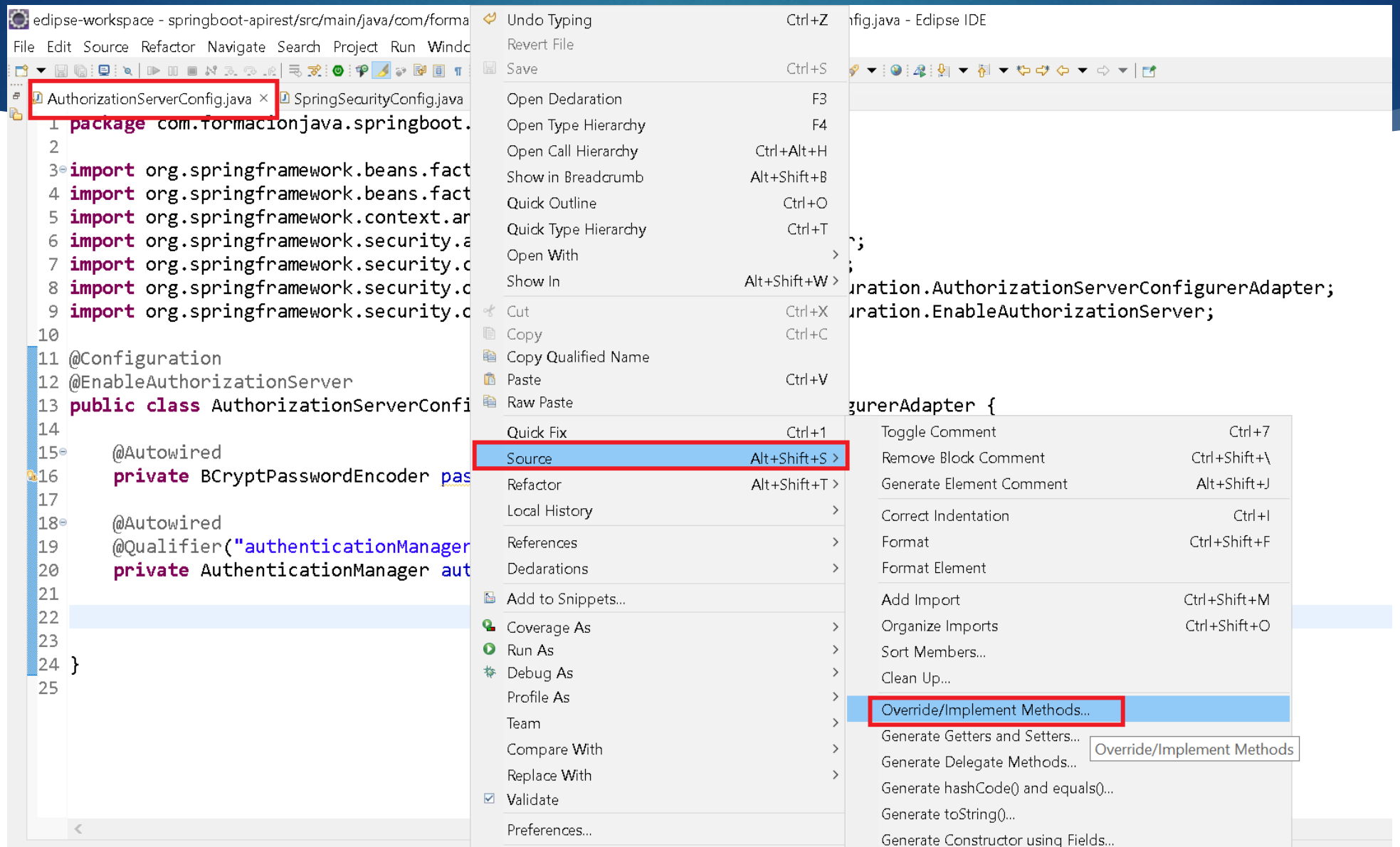

eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/SpringSecurityConfig
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
maven_proyect
proyecto_web
Refresh
repaso
segundo [boot] [devtools]
springboot-apirest [boot] [devtools] [repository master]
src/main/java
com.formacionjava.springboot.apirest
com.formacionjava.springboot.apirest.auth
AuthorizationServerConfig.java
SpringSecurityConfig.java
com.formacionjava.springboot.apirest.config
com.formacionjava.springboot.apirest.controllers
src/main/resources
static
templates
application.properties
import.sql
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
uploads
HELP.md
mvnw
mvnw.cmd
pom.xml
*AuthorizationServerConfig.java
Spring
1 package com.formacionjava
3 import org.springframeworl
10
11 @Configuration
12 public class SpringSecuri
13
14 @Autowired
15 private UserDetailsSe
16
17 @Bean
18 public BCryptPasswordI
19 return new BCryptI
Toggle Comment Ctrl+7
Remove Block Comment Ctrl+Shift+\
Generate Element Comment Alt+Shift+J
Correct Indentation Ctrl+I
Format Ctrl+Shift+F
Format Element
Add Import Ctrl+Shift+M
Organize Imports Ctrl+Shift+O
Sort Members...
Clean Up...
Override/Implement Methods...
Generate Getters and Setters...
Generate Delegate Methods...
Generate hashCode() and equals()...
Generate toString()...
Override/Implement Methods
Undo Ctrl+Z
Revert File
Save Ctrl+S
Open Declaration F3
Open Type Hierarchy F4
Open Call Hierarchy Ctrl+Alt+H
Show in Breadcrumb Alt+Shift+B
Quick Outline Ctrl+O
Quick Type Hierarchy Ctrl+T
Open With
Show In Alt+Shift+W
Cut Ctrl+X
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Raw Paste
Quick Fix Ctrl+1
Source Alt+Shift+S
Refactor Alt+Shift+T
Local History
References
Declarations
Add to Snippets...
Coverage As
Run As
Debug As
Profile As
Team
Replace With
Validate
Preferences
d;
rerAdapter{
th) throws Exception {
dEncoder(passwordEncoder());

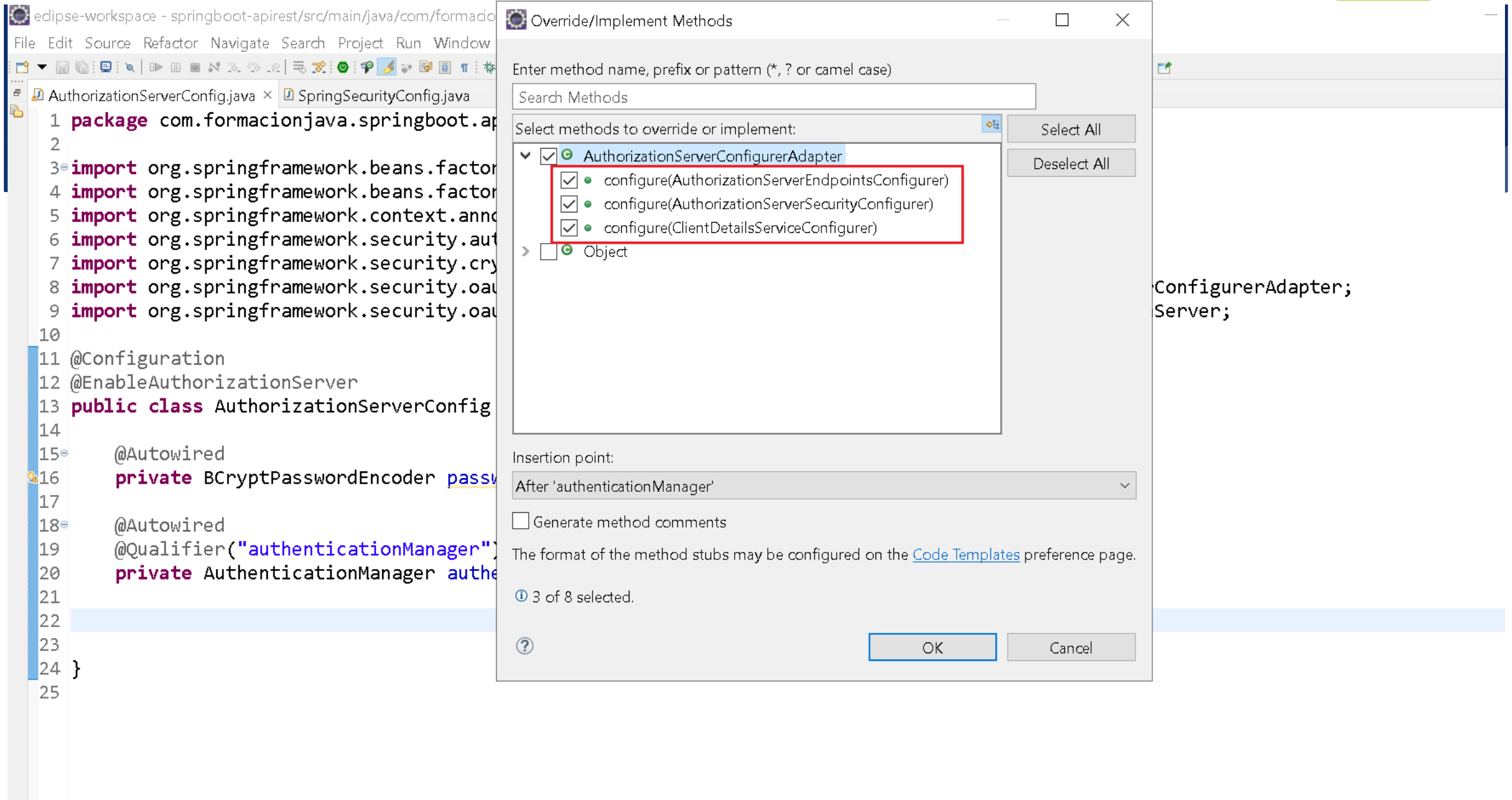

eclipse-workspace - springboot-apirest/src/main/java/com/formacio
File Edit Source Refactor Navigate Search Project Run Window
Project Explorer
maven_proyect
proyecto_web
Refresh
repaso
segundo [boot] [devtools]
springboot-apirest [boot] [devtools] [repository master]
src/main/java
com.formacionjava.springboot.apirest
com.formacionjava.springboot.apirest.auth
AuthorizationServerConfig.java
SpringSecurityConfig.java
com.formacionjava.springboot.apirest.config
com.formacionjava.springboot.apirest.controllers
com.formacionjava.springboot.apirest.models.dao
com.formacionjava.springboot.apirest.models.entity
com.formacionjava.springboot.apirest.models.services
src/main/resources
static
templates
application.properties
import.sql
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
uploads
HELP.md
mvnw
mvnw.cmd
pom.xml
Override/Implement Methods
Enter method name, prefix or pattern (*, ? or camel case)
Search Methods
Select methods to override or implement:
Select All
Deselect All
WebSecurityConfigurerAdapter
authenticationManager()
authenticationManagerBean()
configure(HttpSecurity)
configure(WebSecurity)
init(WebSecurity)
setApplicationContext(ApplicationContext)
setAuthenticationConfiguration(AuthenticationConfigura
setContentNegotationStrategy(ContentNegotiationStrat
setObjectPostProcessor(ObjectPostProcessor<Object>)
setTrustResolver(AuthenticationTrustResolver)
Insertion point:
After 'configure(AuthenticationManagerBuilder)'
Generate method comments
The format of the method stubs may be configured on the Code Templates preference page.
1 of 17 selected.
OK
Cancel
utowired;
ConfigurerAdapter{
lder auth) throws Exception {
passwordEncoder(passwordEncoder());

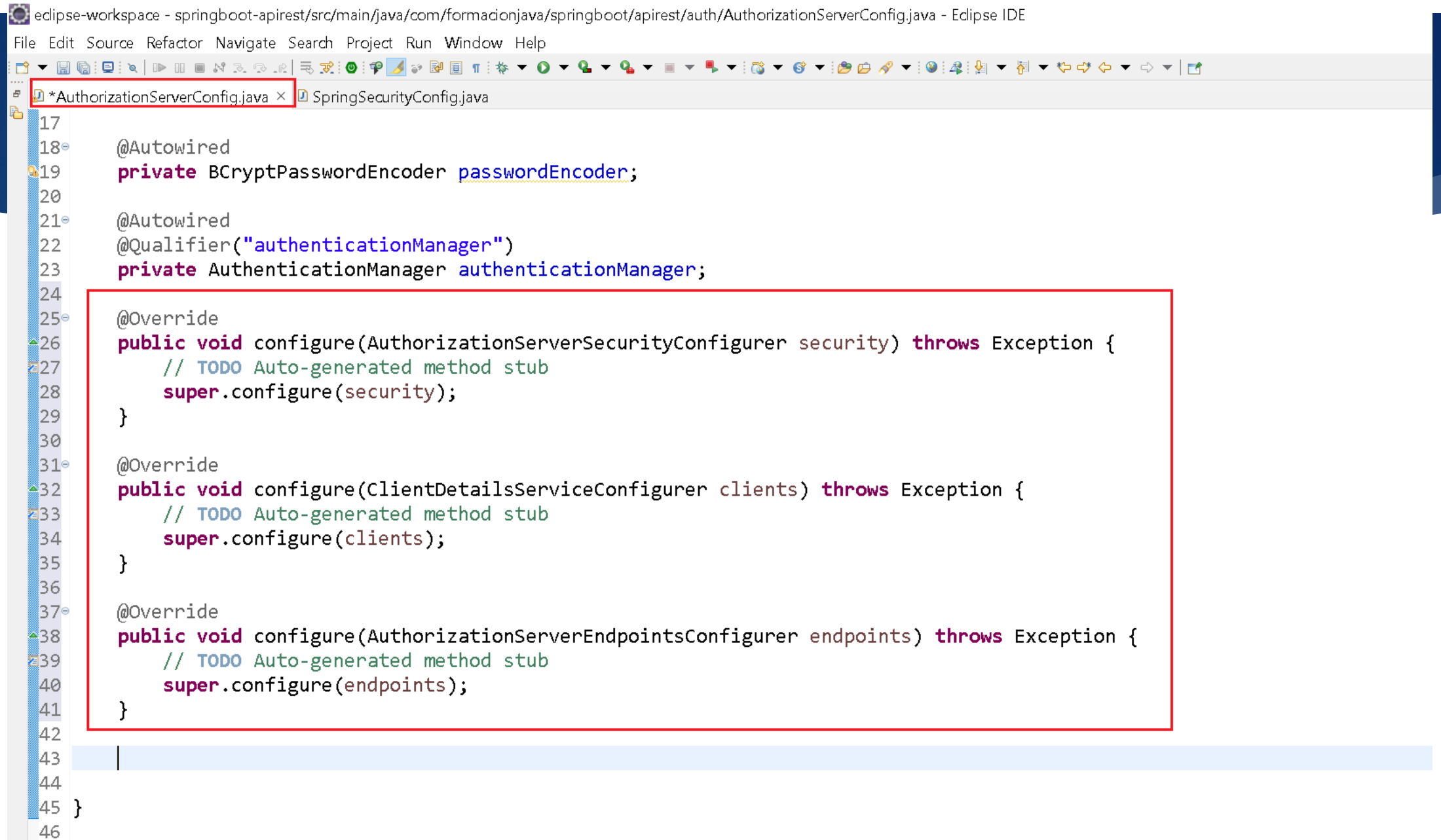

eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/SpringSecurityConfig.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
maven_proyect
proyecto_web
Refresh
repaso
segundo [boot] [devtools]
springboot-apirest [boot] [devtools] [repository master]
src/main/java
com.formacionjava.springboot.apirest
com.formacionjava.springboot.apirest.auth
AuthorizationServerConfig.java
SpringSecurityConfig.java
com.formacionjava.springboot.apirest.config
com.formacionjava.springboot.apirest.controllers
com.formacionjava.springboot.apirest.models.dao
com.formacionjava.springboot.apirest.models.entity
com.formacionjava.springboot.apirest.models.services
src/main/resources
static
templates
application.properties
import.sql
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
uploads
HELP.md
AuthorizationServerConfig.java
*SpringSecurityConfig.java
2
3 import org.springframework.beans.factory.annotation.Autowired;
11
12 @Configuration
13 public class SpringSecurityConfig extends WebSecurityConfigurerAdapter{
14
15 @Autowired
16 private UserDetailsService usuarioService;
17
18 @Bean
19 public BCryptPasswordEncoder passwordEncoder() {
20 return new BCryptPasswordEncoder();
21 }
22
23 @Override
24 protected void configure(AuthenticationManagerBuilder auth) throws Exception {
25
26 auth.userDetailsService(this.usuarioService).passwordEncoder(passwordEncoder());
27 }
28
29 @Bean
30 @Override
31 protected AuthenticationManager authenticationManager() throws Exception {
32 // TODO Auto-generated method stub
33 return super.authenticationManager();
34 }
35
36

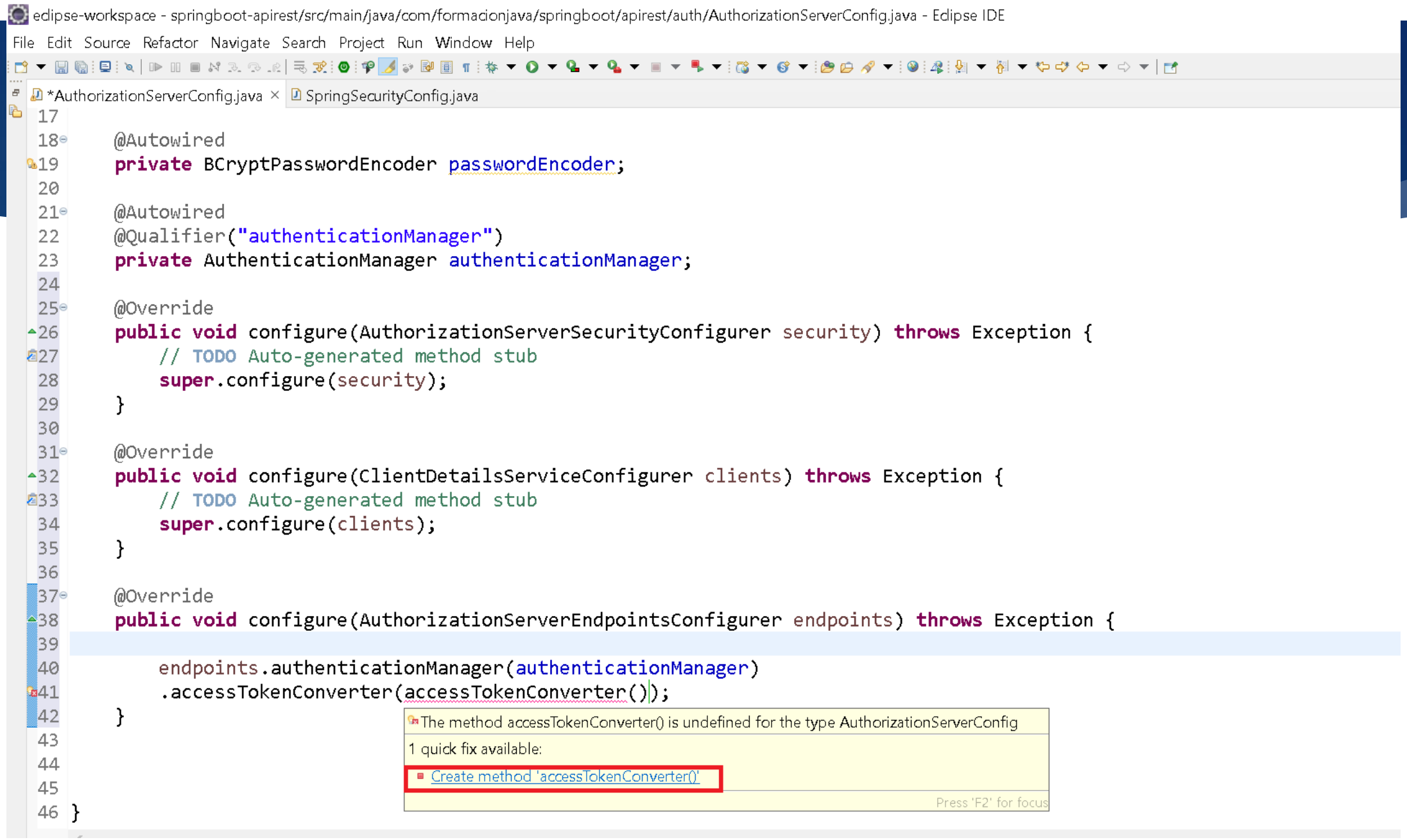

edipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/AuthorizationServerConfig.java - Edipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
AuthorizationServerConfig.java
SpringSecurityConfig.java
1 package com.formacionjava.springboot.apirest.auth;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.beans.factory.annotation.Qualifier;
5 import org.springframework.context.annotation.Configuration;
6 import org.springframework.security.authentication.AuthenticationManager;
7 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
8 import org.springframework.security.oauth2.config.annotation.web.configuration.AuthorizationServerConfigurerAdapter;
9 import org.springframework.security.oauth2.config.annotation.web.configuration.EnableAuthorizationServer;
10
11 @Configuration
12 @EnableAuthorizationServer
13 public class AuthorizationServerConfig extends AuthorizationServerConfigurerAdapter {
14
15 @Autowired
16 private BCryptPasswordEncoder passwordEncoder;
17
18 @Autowired
19 @Qualifier("authenticationManager")
20 private AuthenticationManager authenticationManager;
21
22
23
24 }
25

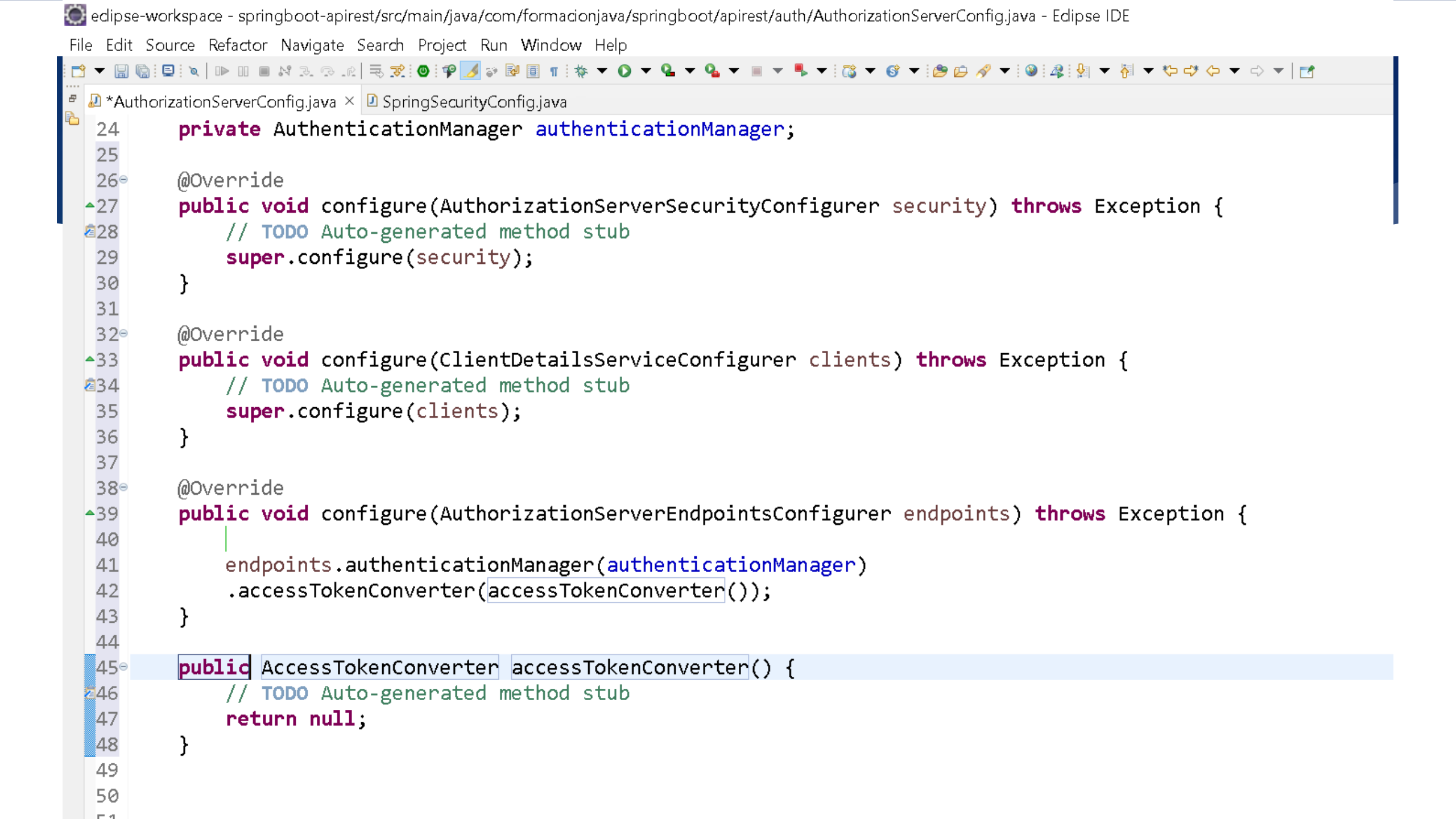
Implementamos métodos de configuración

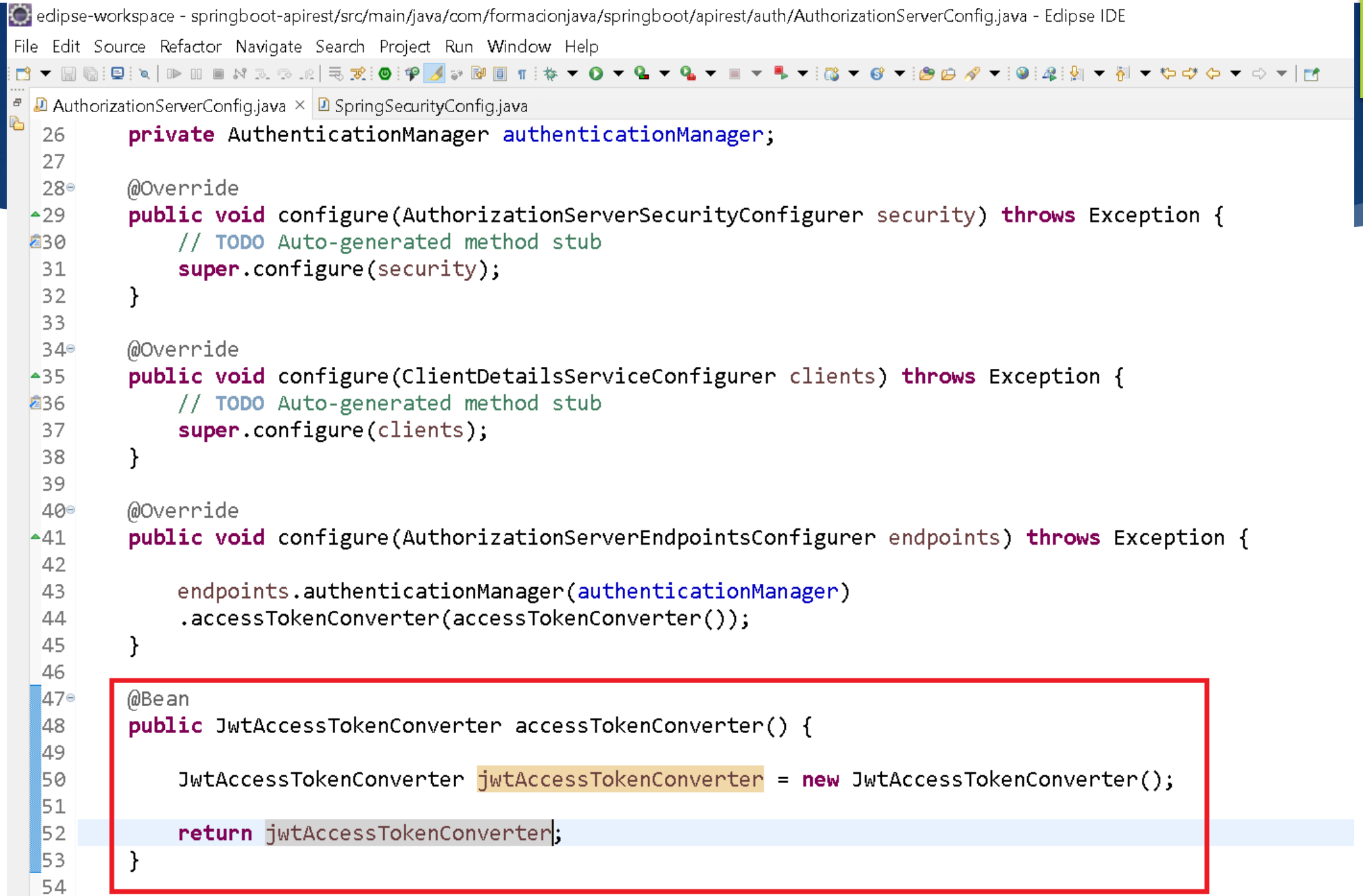



eclipse-workspace - springboot-apirest/src/main/java/com/formacio
File Edit Source Refactor Navigate Search Project Run Window
AuthorizationServerConfig.java
SpringSecurityConfig.java
1 package com.formacionjava.springboot.ap
2
3 import org.springframework.beans.factor
4 import org.springframework.beans.factor
5 import org.springframework.context.annc
6 import org.springframework.security.aut
7 import org.springframework.security.cry
8 import org.springframework.security.oau
9 import org.springframework.security.oau
10
11 @Configuration
12 @EnableAuthorizationServer
13 public class AuthorizationServerConfig
14
15 @Autowired
16 private BCryptPasswordEncoder passv
17
18 @Autowired
19 @Qualifier("authenticationManager")
20 private AuthenticationManager authe
21
22
23
24 }
25
ConfigurerAdapter;
Server;
Override/Implement Methods
Enter method name, prefix or pattern (*, ? or camel case)
Search Methods
Select methods to override or implement:
Select All
Deselect All
AuthorizationServerConfigurerAdapter
configure(AuthorizationServerEndpointsConfigurer)
configure(AuthorizationServerSecurityConfigurer)
configure(ClientDetailsServiceConfigurer)
Object
Insertion point:
After 'authenticationManager'
Generate method comments
The format of the method stubs may be configured on the Code Templates preference page.
3 of 8 selected.
OK
Cancel


eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/AuthorizationServerConfig.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*AuthorizationServerConfig.java
SpringSecurityConfig.java
@Autowired
private BCryptPasswordEncoder passwordEncoder;
@Autowired
@Qualifier("authenticationManager")
private AuthenticationManager authenticationManager;
@Override
public void configure(AuthorizationServerSecurityConfigurer security) throws Exception {
// TODO Auto-generated method stub
super.configure(security);
}
@Override
public void configure(ClientDetailsServiceConfigurer clients) throws Exception {
// TODO Auto-generated method stub
super.configure(clients);
}
@Override
public void configure(AuthorizationServerEndpointsConfigurer endpoints) throws Exception {
// TODO Auto-generated method stub
super.configure(endpoints);
}
}

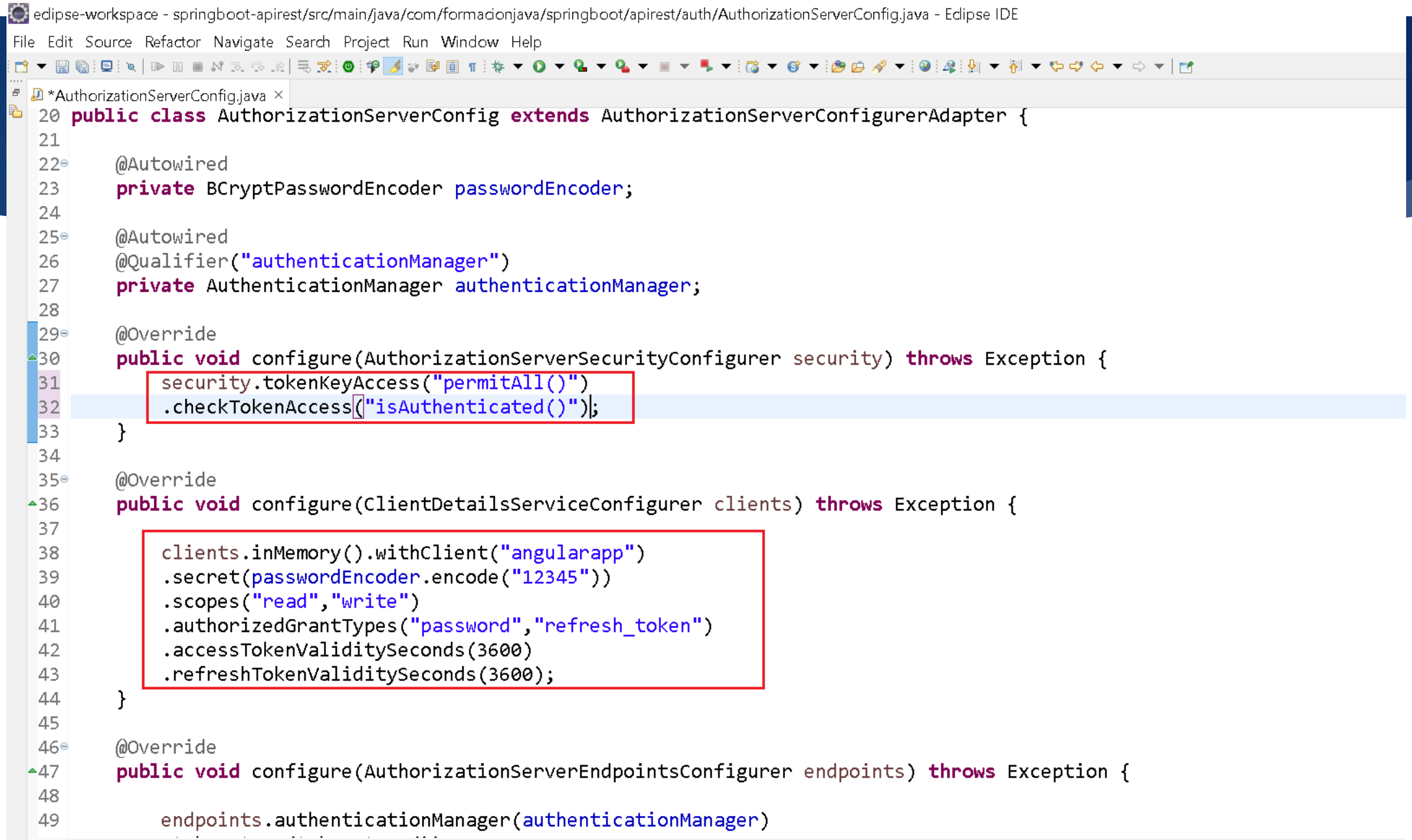

eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/AuthorizationServerConfig.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*AuthorizationServerConfig.java
SpringSecurityConfig.java
17
18 @Autowired
19 private BCryptPasswordEncoder passwordEncoder;
20
21 @Autowired
22 @Qualifier("authenticationManager")
23 private AuthenticationManager authenticationManager;
24
25 @Override
26 public void configure(AuthorizationServerSecurityConfigurer security) throws Exception {
27 // TODO Auto-generated method stub
28 super.configure(security);
29 }
30
31 @Override
32 public void configure(ClientDetailsServiceConfigurer clients) throws Exception {
33 // TODO Auto-generated method stub
34 super.configure(clients);
35 }
36
37 @Override
38 public void configure(AuthorizationServerEndpointsConfigurer endpoints) throws Exception {
39
40 endpoints.authenticationManager(authenticationManager)
41 .accessTokenConverter(accessTokenConverter());
42 }
43
44
45
46 }
The method accessTokenConverter() is undefined for the type AuthorizationServerConfig
1 quick fix available:
Create method 'accessTokenConverter()'
Press 'F2' for focus

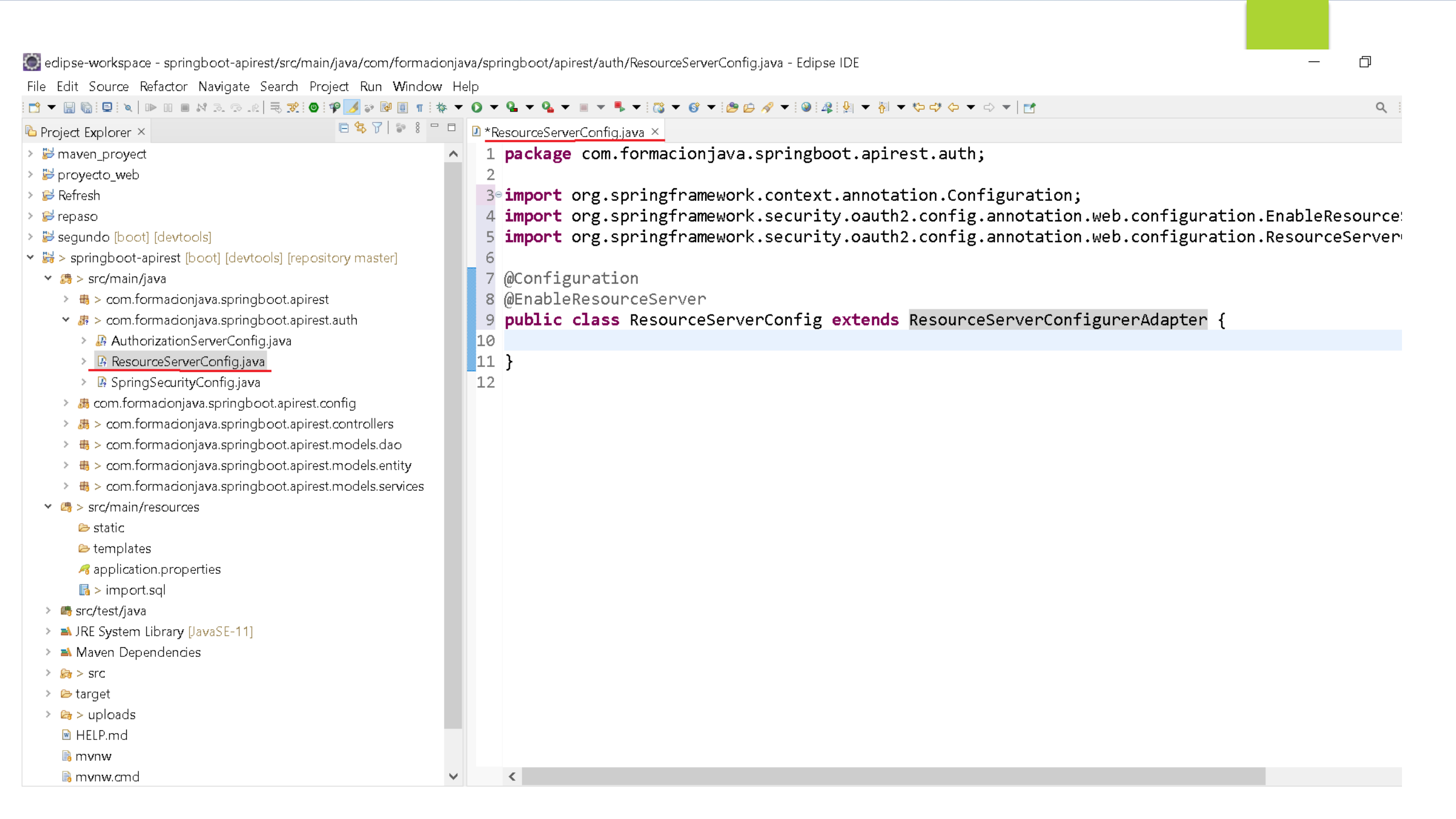

eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/AuthorizationServerConfig.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*AuthorizationServerConfig.java
SpringSecurityConfig.java
24 private AuthenticationManager authenticationManager;
25
26 @Override
27 public void configure(AuthorizationServerSecurityConfigurer security) throws Exception {
28 // TODO Auto-generated method stub
29 super.configure(security);
30 }
31
32 @Override
33 public void configure(ClientDetailsServiceConfigurer clients) throws Exception {
34 // TODO Auto-generated method stub
35 super.configure(clients);
36 }
37
38 @Override
39 public void configure(AuthorizationServerEndpointsConfigurer endpoints) throws Exception {
40
41 endpoints.authenticationManager(authenticationManager)
42 .accessTokenConverter(accessTokenConverter());
43 }
44
45 public AccessTokenConverter accessTokenConverter() {
46 // TODO Auto-generated method stub
47 return null;
48 }
49
50

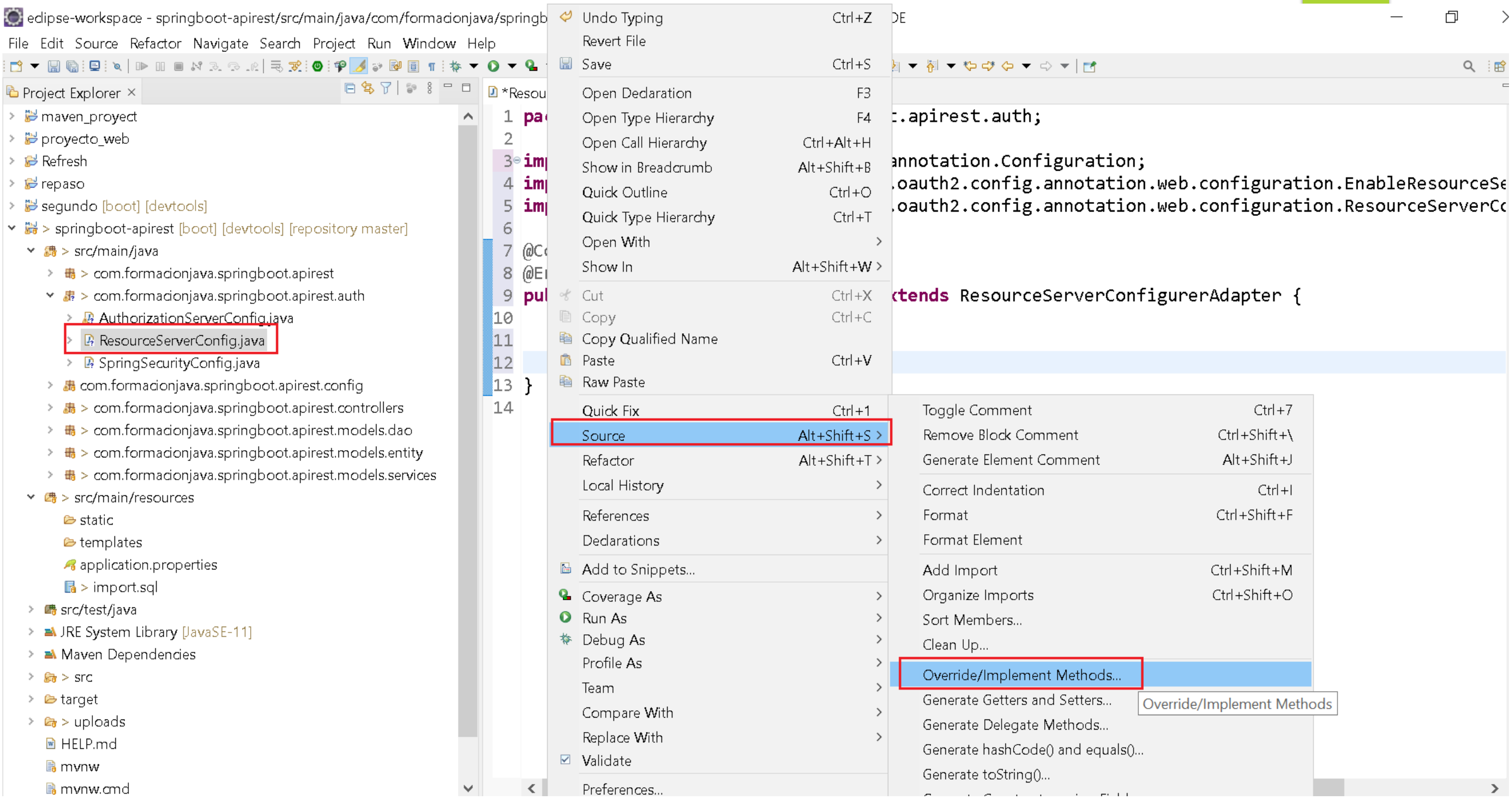

eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/AuthorizationServerConfig.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
AuthorizationServerConfig.java
SpringSecurityConfig.java
26 private AuthenticationManager authenticationManager;
27
28 @Override
29 public void configure(AuthorizationServerSecurityConfigurer security) throws Exception {
30 // TODO Auto-generated method stub
31 super.configure(security);
32 }
33
34 @Override
35 public void configure(ClientDetailsServiceConfigurer clients) throws Exception {
36 // TODO Auto-generated method stub
37 super.configure(clients);
38 }
39
40 @Override
41 public void configure(AuthorizationServerEndpointsConfigurer endpoints) throws Exception {
42
43 endpoints.authenticationManager(authenticationManager)
44 .accessTokenConverter(accessTokenConverter());
45 }
46
47 @Bean
48 public JwtAccessTokenConverter accessTokenConverter() {
49
50 JwtAccessTokenConverter jwtAccessTokenConverter = new JwtAccessTokenConverter();
51
52 return jwtAccessTokenConverter;
53 }
54

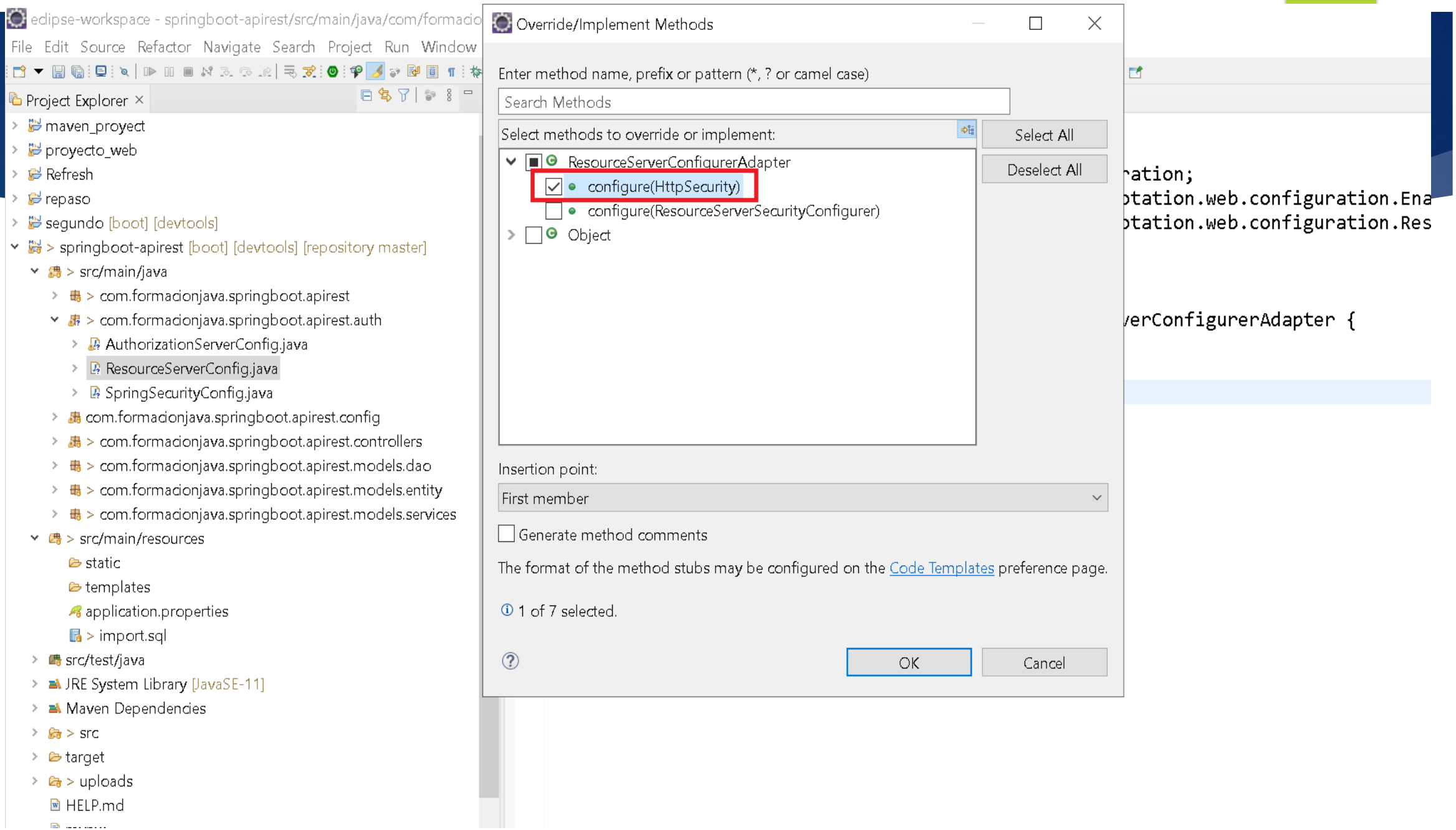
*AuthorizationServerConfig.java × SpringSecurityConfig.java

```
32          super.configure(security);
33      }
34
35      @Override
36      public void configure(ClientDetailsServiceConfigurer clients) throws Exception {
37          // TODO Auto-generated method stub
38          super.configure(clients);
39      }
40
41      @Override
42      public void configure(AuthorizationServerEndpointsConfigurer endpoints) throws Exception {
43
44          endpoints.authenticationManager(authenticationManager)
45          .tokenStore(tokenStore())
46          .accessTokenConverter(accessTokenConverter());
47      }
48
49      @Bean
50      public JwtTokenStore tokenStore() {
51          return new JwtTokenStore(accessTokenConverter());
52      }
53
54      @Bean
55      public JwtAccessTokenConverter accessTokenConverter() {
56
57          JwtAccessTokenConverter jwtAccessTokenConverter = new JwtAccessTokenConverter();
58
59          return jwtAccessTokenConverter;
60      }
61
```


eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/AuthorizationServerConfig.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
*AuthorizationServerConfig.java
20 public class AuthorizationServerConfig extends AuthorizationServerConfigurerAdapter {
21
22 @Autowired
23 private BCryptPasswordEncoder passwordEncoder;
24
25 @Autowired
26 @Qualifier("authenticationManager")
27 private AuthenticationManager authenticationManager;
28
29 @Override
30 public void configure(AuthorizationServerSecurityConfigurer security) throws Exception {
31 security.tokenKeyAccess("permitAll()")
32 .checkTokenAccess("isAuthenticated()");
33 }
34
35 @Override
36 public void configure(ClientDetailsServiceConfigurer clients) throws Exception {
37
38 clients.inMemory().withClient("angularapp")
39 .secret(passwordEncoder.encode("12345"))
40 .scopes("read","write")
41 .authorizedGrantTypes("password","refresh_token")
42 .accessTokenValiditySeconds(3600)
43 .refreshTokenValiditySeconds(3600);
44 }
45
46 @Override
47 public void configure(AuthorizationServerEndpointsConfigurer endpoints) throws Exception {
48
49 endpoints.authenticationManager(authenticationManager)


edipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/ResourceServerConfig.java - Edipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
maven_proyect
proyecto_web
Refresh
repaso
segundo [boot] [devtools]
springboot-apirest [boot] [devtools] [repository master]
src/main/java
com.formacionjava.springboot.apirest
com.formacionjava.springboot.apirest.auth
AuthorizationServerConfig.java
ResourceServerConfig.java
SpringSecurityConfig.java
com.formacionjava.springboot.apirest.config
com.formacionjava.springboot.apirest.controllers
com.formacionjava.springboot.apirest.models.dao
com.formacionjava.springboot.apirest.models.entity
com.formacionjava.springboot.apirest.models.services
src/main/resources
static
templates
application.properties
import.sql
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
uploads
HELP.md
mvnw
mvnw.cmd
*ResourceServerConfig.java
1 package com.formacionjava.springboot.apirest.auth;
2
3 import org.springframework.context.annotation.Configuration;
4 import org.springframework.security.oauth2.config.annotation.web.configuration.EnableResource
5 import org.springframework.security.oauth2.config.annotation.web.configuration.ResourceServer
6
7 @Configuration
8 @EnableResourceServer
9 public class ResourceServerConfig extends ResourceServerConfigurerAdapter {
10
11 }
12


eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springb
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
maven_proyect
proyecto_web
Refresh
repaso
segundo [boot] [devtools]
springboot-apirest [boot] [devtools] [repository master]
src/main/java
com.formacionjava.springboot.apirest
com.formacionjava.springboot.apirest.auth
AuthorizationServerConfig.java
ResourceServerConfig.java
SpringSecurityConfig.java
com.formacionjava.springboot.apirest.config
com.formacionjava.springboot.apirest.controllers
com.formacionjava.springboot.apirest.models.dao
com.formacionjava.springboot.apirest.models.entity
com.formacionjava.springboot.apirest.models.services
src/main/resources
static
templates
application.properties
import.sql
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
uploads
HELP.md
mvnw
mvnw.cmd
*Resou
1 pa
2
3 im
4 im
5 im
6
7 @C
8 @E
9 pu
10
11
12
13 }
14
.apirest.auth;
annotation.Configuration;
.oauth2.config.annotation.web.configuration.EnableResourceSe
.oauth2.config.annotation.web.configuration.ResourceServerCo
xtends ResourceServerConfigurerAdapter {
Undo Typing Ctrl+Z
Revert File
Save Ctrl+S
Open Declaration F3
Open Type Hierarchy F4
Open Call Hierarchy Ctrl+Alt+H
Show in Breadcrumb Alt+Shift+B
Quick Outline Ctrl+O
Quick Type Hierarchy Ctrl+T
Open With
Show In Alt+Shift+W
Cut Ctrl+X
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Raw Paste
Quick Fix Ctrl+1
Source Alt+Shift+S
Refactor Alt+Shift+T
Local History
References
Declarations
Add to Snippets...
Coverage As
Run As
Debug As
Profile As
Team
Compare With
Replace With
Validate
Preferences...
Toggle Comment Ctrl+7
Remove Block Comment Ctrl+Shift+\
Generate Element Comment Alt+Shift+J
Correct Indentation Ctrl+I
Format Ctrl+Shift+F
Format Element
Add Import Ctrl+Shift+M
Organize Imports Ctrl+Shift+O
Sort Members...
Clean Up...
Override/Implement Methods...
Generate Getters and Setters...
Generate Delegate Methods...
Generate hashCode() and equals()...
Generate toString()...
Override/Implement Methods


eclipse-workspace - springboot-apirest/src/main/java/com/formacio
File Edit Source Refactor Navigate Search Project Run Window
Project Explorer
maven_proyect
proyecto_web
Refresh
repaso
segundo [boot] [devtools]
springboot-apirest [boot] [devtools] [repository master]
src/main/java
com.formacionjava.springboot.apirest
com.formacionjava.springboot.apirest.auth
AuthorizationServerConfig.java
ResourceServerConfig.java
SpringSecurityConfig.java
com.formacionjava.springboot.apirest.config
com.formacionjava.springboot.apirest.controllers
com.formacionjava.springboot.apirest.models.dao
com.formacionjava.springboot.apirest.models.entity
com.formacionjava.springboot.apirest.models.services
src/main/resources
static
templates
application.properties
import.sql
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
uploads
HELP.md
Override/Implement Methods
Enter method name, prefix or pattern (*, ? or camel case)
Search Methods
Select methods to override or implement:
Select All
Deselect All
ResourceServerConfigurerAdapter
configure(HttpSecurity)
configure(ResourceServerSecurityConfigurer)
Object
Insertion point:
First member
Generate method comments
The format of the method stubs may be configured on the Code Templates preference page.
1 of 7 selected.
OK
Cancel
ration;
otation.web.configuration.Ena
otation.web.configuration.Res
verConfigurerAdapter {

eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/ResourceServerConfig.java - Eclipse IDE

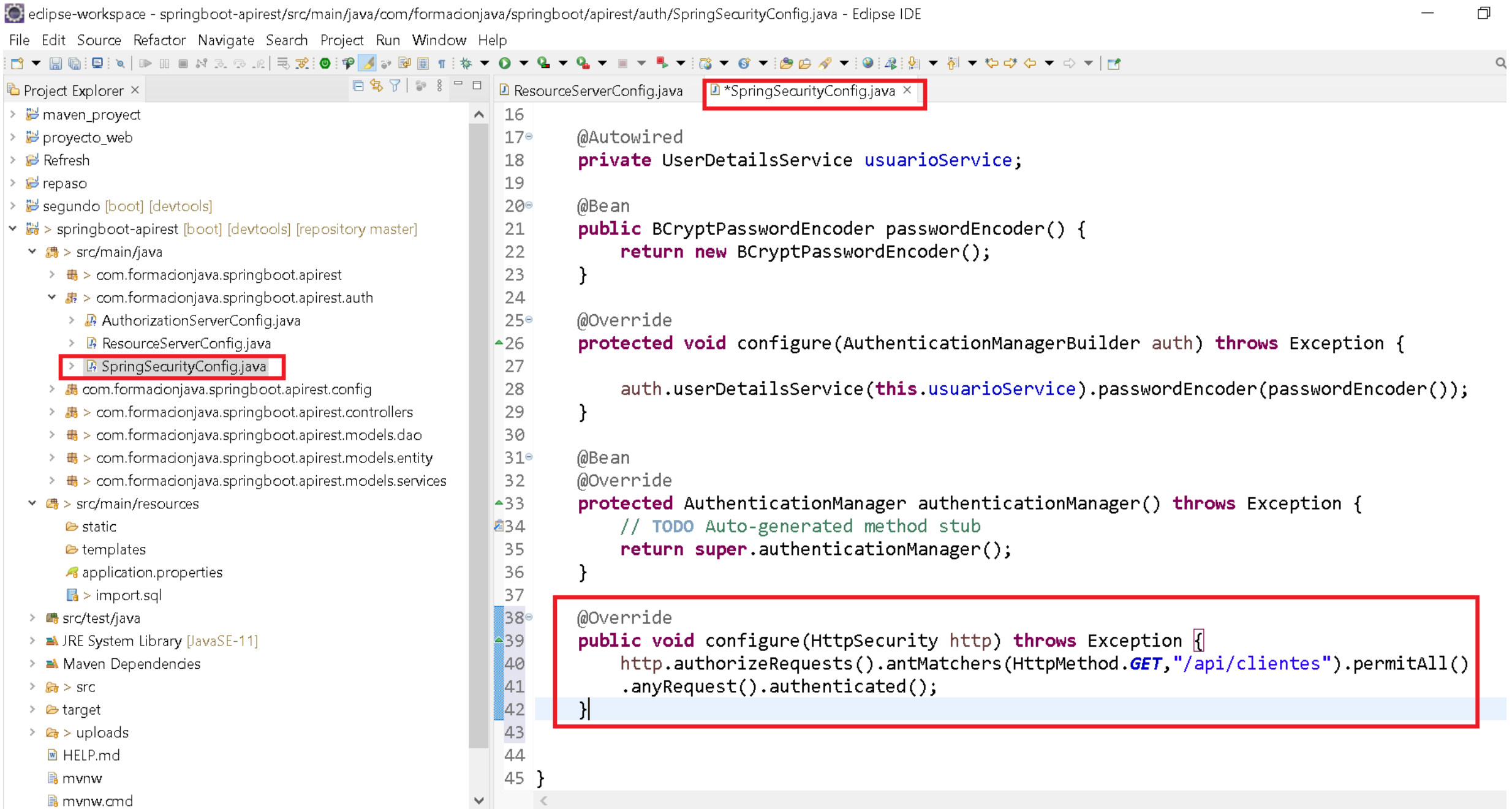
File Edit Source Refactor Navigate Search Project Run Window Help

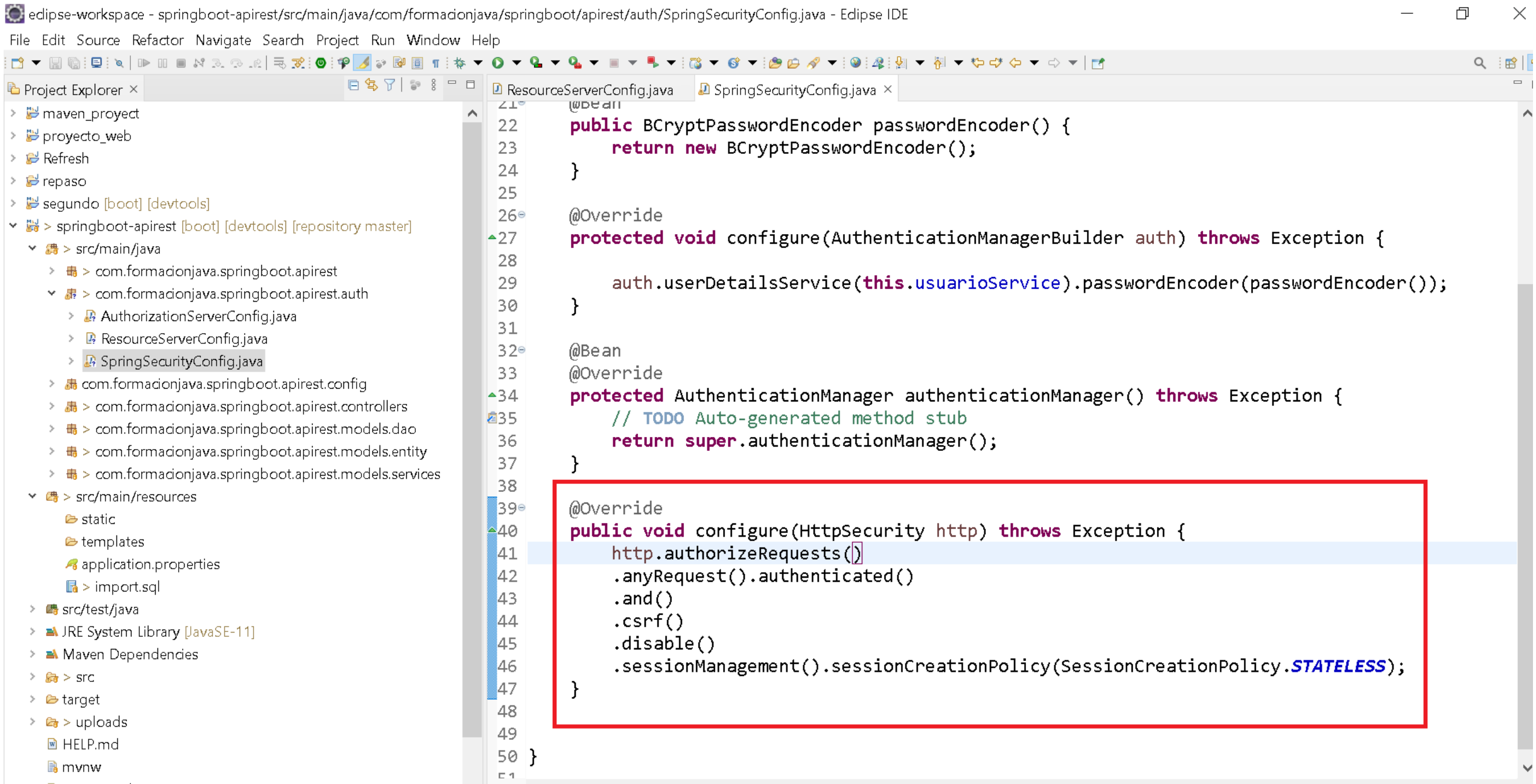
Project Explorer

- maven_proyect
- proyecto_web
- Refresh
- repaso
- segundo [boot] [devtools]
- springboot-apirest [boot] [devtools] [repository master]
  - src/main/java
    - com.formacionjava.springboot.apirest
    - com.formacionjava.springboot.apirest.auth
      - AuthorizationServerConfig.java
      - ResourceServerConfig.java
      - SpringSecurityConfig.java
    - com.formacionjava.springboot.apirest.config
    - com.formacionjava.springboot.apirest.controllers
    - com.formacionjava.springboot.apirest.models.dao
    - com.formacionjava.springboot.apirest.models.entity
    - com.formacionjava.springboot.apirest.models.services
  - src/main/resources
    - static
    - templates
    - application.properties
    - import.sql
  - src/test/java
  - JRE System Library [JavaSE-11]
  - Maven Dependencies
  - src
  - target
  - uploads
  - HELP.md
  - mvnw
  - mvnw.cmd

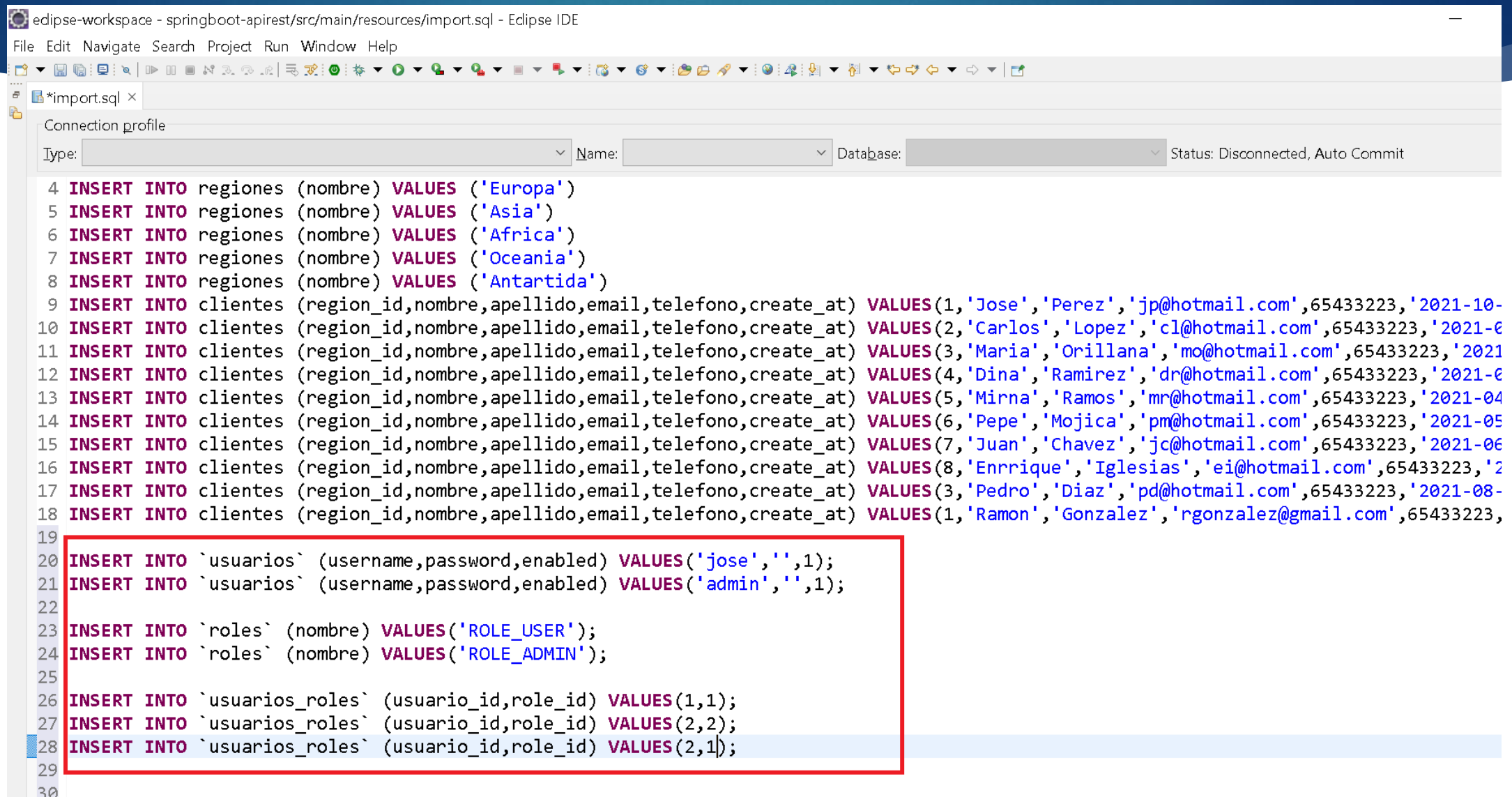
*ResourceServerConfig.java

```java
package com.formacionjava.springboot.apirest.auth;

import org.springframework.context.annotation.Configuration;
import org.springframework.http.HttpMethod;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.oauth2.config.annotation.web.configuration.EnableResourceSe
import org.springframework.security.oauth2.config.annotation.web.configuration.ResourceServerCo

@Configuration
@EnableResourceServer
public class ResourceServerConfig extends ResourceServerConfigurerAdapter {

    @Override
    public void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests().antMatchers(HttpMethod.GET,"/api/clientes").permitAll()
        .anyRequest().authenticated();
    }


}

```

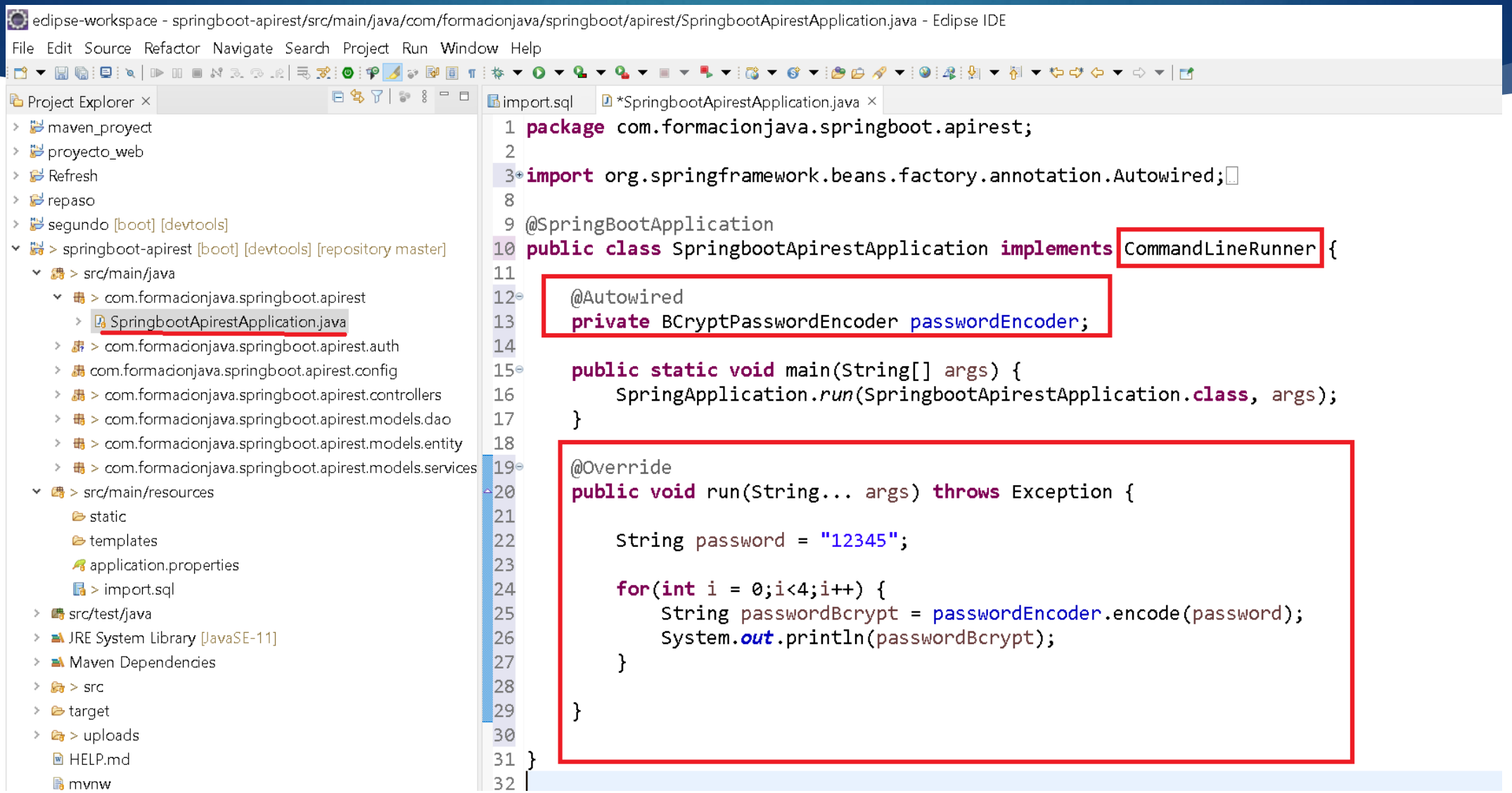

eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/SpringSecurityConfig.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
maven_proyect
proyecto_web
Refresh
repaso
segundo [boot] [devtools]
springboot-apirest [boot] [devtools] [repository master]
src/main/java
com.formacionjava.springboot.apirest
com.formacionjava.springboot.apirest.auth
AuthorizationServerConfig.java
ResourceServerConfig.java
SpringSecurityConfig.java
com.formacionjava.springboot.apirest.config
com.formacionjava.springboot.apirest.controllers
com.formacionjava.springboot.apirest.models.dao
com.formacionjava.springboot.apirest.models.entity
com.formacionjava.springboot.apirest.models.services
src/main/resources
static
templates
application.properties
import.sql
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
uploads
HELP.md
mvnw
mvnw.cmd
ResourceServerConfig.java
*SpringSecurityConfig.java
@Autowired
private UserDetailsService usuarioService;
@Bean
public BCryptPasswordEncoder passwordEncoder() {
return new BCryptPasswordEncoder();
}
@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
auth.userDetailsService(this.usuarioService).passwordEncoder(passwordEncoder());
}
@Bean
@Override
protected AuthenticationManager authenticationManager() throws Exception {
// TODO Auto-generated method stub
return super.authenticationManager();
}
@Override
public void configure(HttpSecurity http) throws Exception {
http.authorizeRequests().antMatchers(HttpMethod.GET,"/api/clientes").permitAll()
.anyRequest().authenticated();
}
}

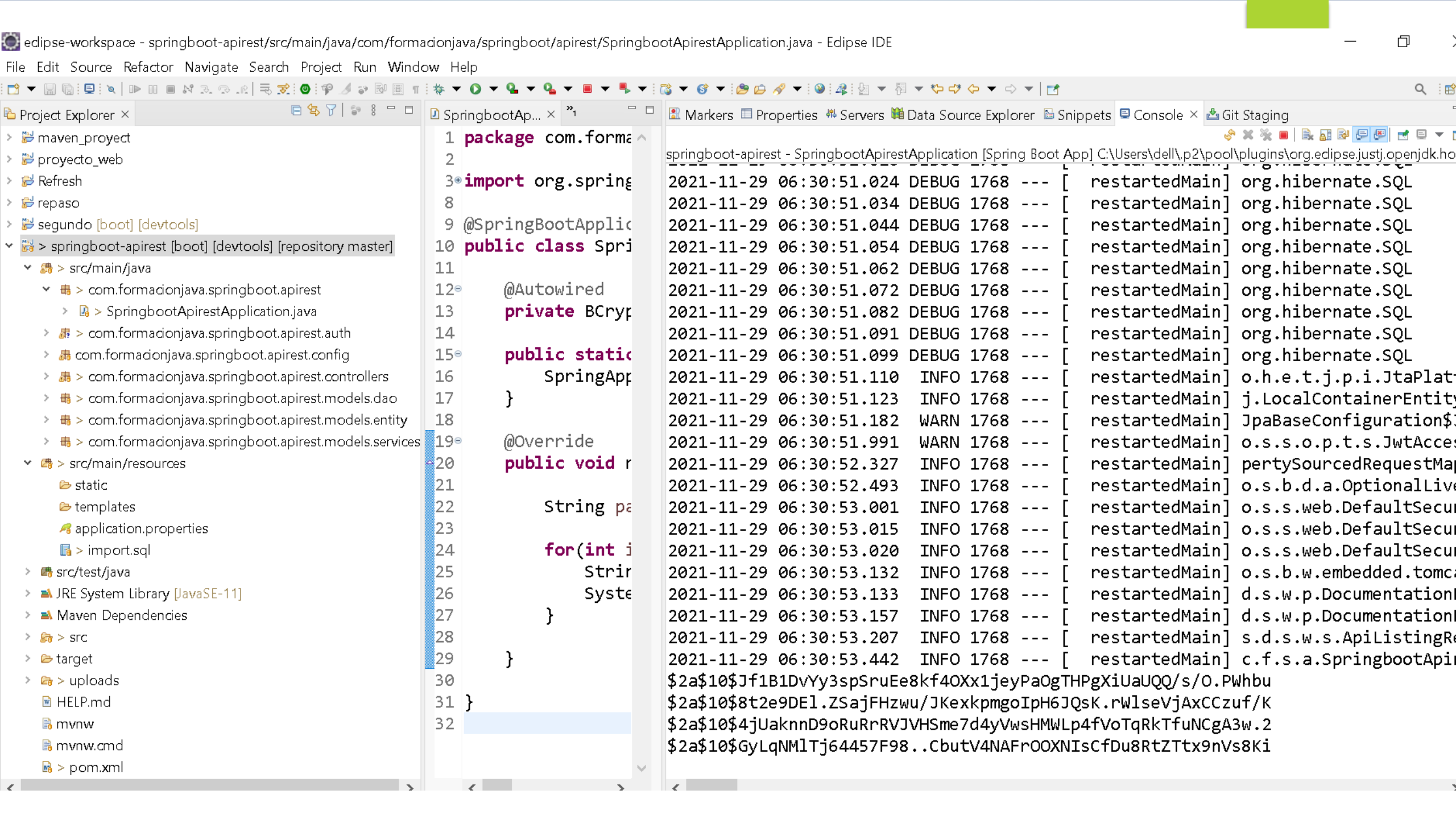

eclipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/auth/SpringSecurityConfig.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
maven_proyect
proyecto_web
Refresh
repaso
segundo [boot] [devtools]
springboot-apirest [boot] [devtools] [repository master]
src/main/java
com.formacionjava.springboot.apirest
com.formacionjava.springboot.apirest.auth
AuthorizationServerConfig.java
ResourceServerConfig.java
SpringSecurityConfig.java
com.formacionjava.springboot.apirest.config
com.formacionjava.springboot.apirest.controllers
com.formacionjava.springboot.apirest.models.dao
com.formacionjava.springboot.apirest.models.entity
com.formacionjava.springboot.apirest.models.services
src/main/resources
static
templates
application.properties
import.sql
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
uploads
HELP.md
mvnw
ResourceServerConfig.java
SpringSecurityConfig.java
@Bean
public BCryptPasswordEncoder passwordEncoder() {
return new BCryptPasswordEncoder();
}
@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
auth.userDetailsService(this.usuarioService).passwordEncoder(passwordEncoder());
}
@Bean
@Override
protected AuthenticationManager authenticationManager() throws Exception {
// TODO Auto-generated method stub
return super.authenticationManager();
}
@Override
public void configure(HttpSecurity http) throws Exception {
http.authorizeRequests()
.anyRequest().authenticated()
.and()
.csrf()
.disable()
.sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS);
}
}

Creamos datos de pruebas



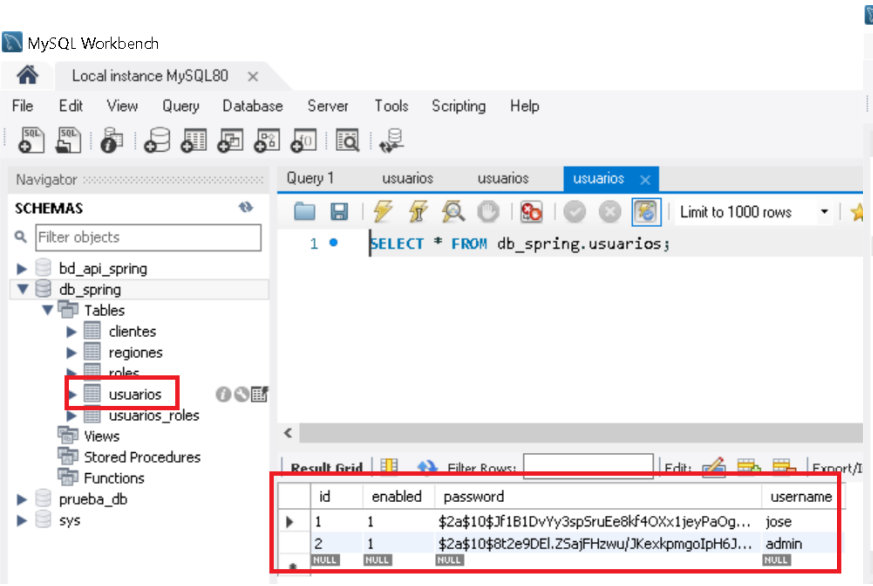
Generamos los password encriptados

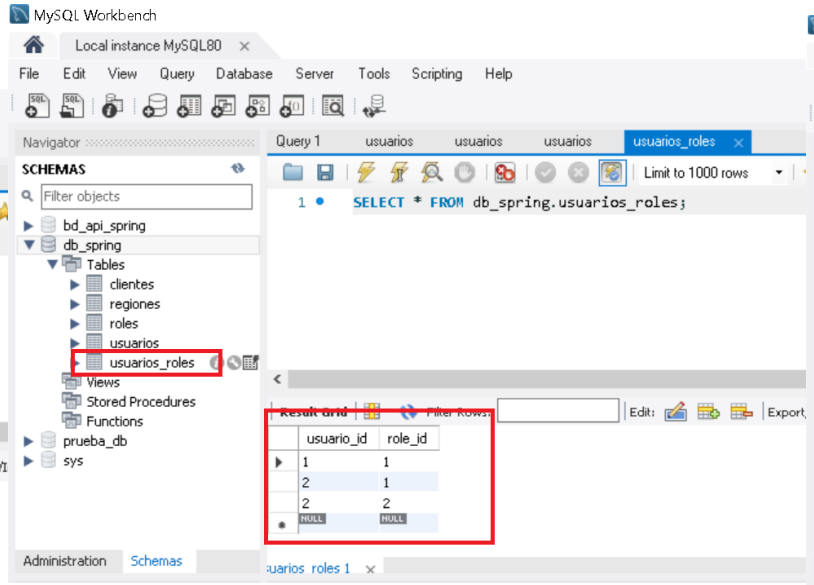


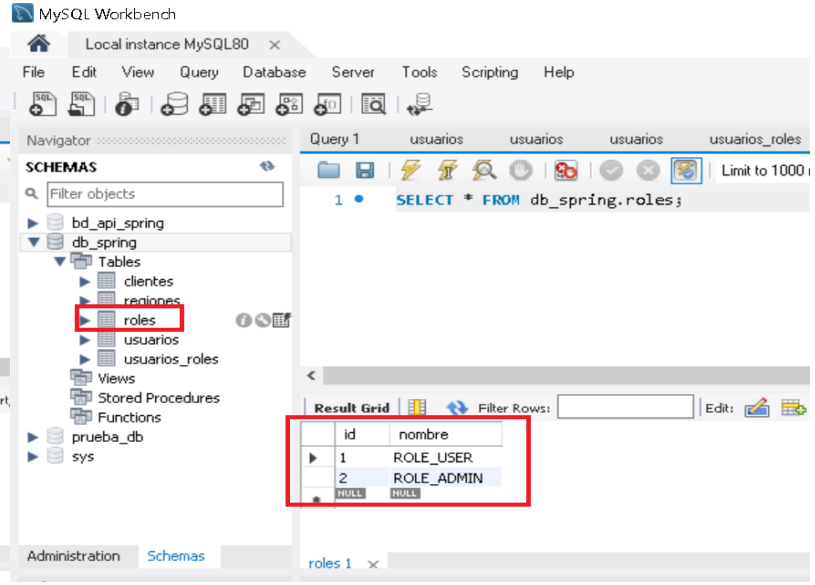

edipse-workspace - springboot-apirest/src/main/java/com/formacionjava/springboot/apirest/SpringbootApirestApplication.java - Edipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
maven_proyect
proyecto_web
Refresh
repaso
segundo [boot] [devtools]
> springboot-apirest [boot] [devtools] [repository master]
> src/main/java
> com.formacionjava.springboot.apirest
> SpringbootApirestApplication.java
> com.formacionjava.springboot.apirest.auth
com.formacionjava.springboot.apirest.config
> com.formacionjava.springboot.apirest.controllers
> com.formacionjava.springboot.apirest.models.dao
> com.formacionjava.springboot.apirest.models.entity
> com.formacionjava.springboot.apirest.models.services
> src/main/resources
static
templates
application.properties
> import.sql
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
> src
target
> uploads
HELP.md
mvnw
mvnw.cmd
> pom.xml
SpringbootAp...
1 package com.forma
2
3 import org.spring
8
9 @SpringBootApplic
10 public class Spri
11
12 @Autowired
13 private BCryp
14
15 public static
16 SpringApp
17 }
18
19 @Override
20 public void r
21
22 String pa
23
24 for(int i
25 Strir
26 Syste
27 }
28
29 }
30
31 }
32
Markers Properties Servers Data Source Explorer Snippets Console Git Staging
springboot-apirest - SpringbootApirestApplication [Spring Boot App] C:\Users\dell\.p2\pool\plugins\org.eclipse.justj.openjdk.ho
2021-11-29 06:30:51.024 DEBUG 1768 --- [restartedMain] org.hibernate.SQL
2021-11-29 06:30:51.034 DEBUG 1768 --- [restartedMain] org.hibernate.SQL
2021-11-29 06:30:51.044 DEBUG 1768 --- [restartedMain] org.hibernate.SQL
2021-11-29 06:30:51.054 DEBUG 1768 --- [restartedMain] org.hibernate.SQL
2021-11-29 06:30:51.062 DEBUG 1768 --- [restartedMain] org.hibernate.SQL
2021-11-29 06:30:51.072 DEBUG 1768 --- [restartedMain] org.hibernate.SQL
2021-11-29 06:30:51.082 DEBUG 1768 --- [restartedMain] org.hibernate.SQL
2021-11-29 06:30:51.091 DEBUG 1768 --- [restartedMain] org.hibernate.SQL
2021-11-29 06:30:51.099 DEBUG 1768 --- [restartedMain] org.hibernate.SQL
2021-11-29 06:30:51.110 INFO 1768 --- [restartedMain] o.h.e.t.j.p.i.JtaPlat
2021-11-29 06:30:51.123 INFO 1768 --- [restartedMain] j.LocalContainerEntity
2021-11-29 06:30:51.182 WARN 1768 --- [restartedMain] JpaBaseConfiguration$
2021-11-29 06:30:51.991 WARN 1768 --- [restartedMain] o.s.s.o.p.t.s.JwtAcces
2021-11-29 06:30:52.327 INFO 1768 --- [restartedMain] pertySourcedRequestMap
2021-11-29 06:30:52.493 INFO 1768 --- [restartedMain] o.s.b.d.a.OptionalLive
2021-11-29 06:30:53.001 INFO 1768 --- [restartedMain] o.s.s.web.DefaultSecur
2021-11-29 06:30:53.015 INFO 1768 --- [restartedMain] o.s.s.web.DefaultSecur
2021-11-29 06:30:53.020 INFO 1768 --- [restartedMain] o.s.s.web.DefaultSecur
2021-11-29 06:30:53.132 INFO 1768 --- [restartedMain] o.s.b.w.embedded.tomca
2021-11-29 06:30:53.133 INFO 1768 --- [restartedMain] d.s.w.p.DocumentationP
2021-11-29 06:30:53.157 INFO 1768 --- [restartedMain] d.s.w.p.DocumentationP
2021-11-29 06:30:53.207 INFO 1768 --- [restartedMain] s.d.s.w.s.ApiListingRe
2021-11-29 06:30:53.442 INFO 1768 --- [restartedMain] c.f.s.a.SpringbootApir
$2a$10$Jf1B1DvYy3spSruEe8kf4OXx1jeyPaOgTHPgXiUaUQQ/s/O.PWhbu
$2a$10$8t2e9DEl.ZSajFHzwu/JKexkpmgoIpH6JQsK.rWlseVjAxCCzuf/K
$2a$10$4jUaknnD9oRuRrRVJVHSme7d4yVwsHMWLp4fVoTqRkTfuNCgA3w.2
$2a$10$GyLqNMlTj64457F98..CbutV4NAFrOOXNIsCfDu8RtZTtx9nVs8Ki

```sql
INSERT INTO clientes (region_id,nombre,apellido,email,telefono,create_at) VALUES(1,'Ramon','Gonzalez','rgonzalez@gmail.com',65433223,'202

INSERT INTO `usuarios` (username,password,enabled) VALUES('jose','$2a$10$Jf1B1DvYy3spSruEe8kf4OXx1jeyPaOgTHPgXiUaUQQ/s/O.PWhbu',1);
INSERT INTO `usuarios` (username,password,enabled) VALUES('admin','$2a$10$8t2e9DEl.ZSajFHzwu/JKexkpmgoIpH6JQsK.rWlseVjAxCCzuf/K',1);

INSERT INTO `roles` (nombre) VALUES('ROLE_USER');
INSERT INTO `roles` (nombre) VALUES('ROLE_ADMIN');

INSERT INTO `usuarios_roles` (usuario_id,role_id) VALUES(1,1);
INSERT INTO `usuarios_roles` (usuario_id,role_id) VALUES(2,2);
INSERT INTO `usuarios_roles` (usuario_id,role_id) VALUES(2,1);
```

Compilamos y probarlo, deberíamos verlo en nuestra base de datos

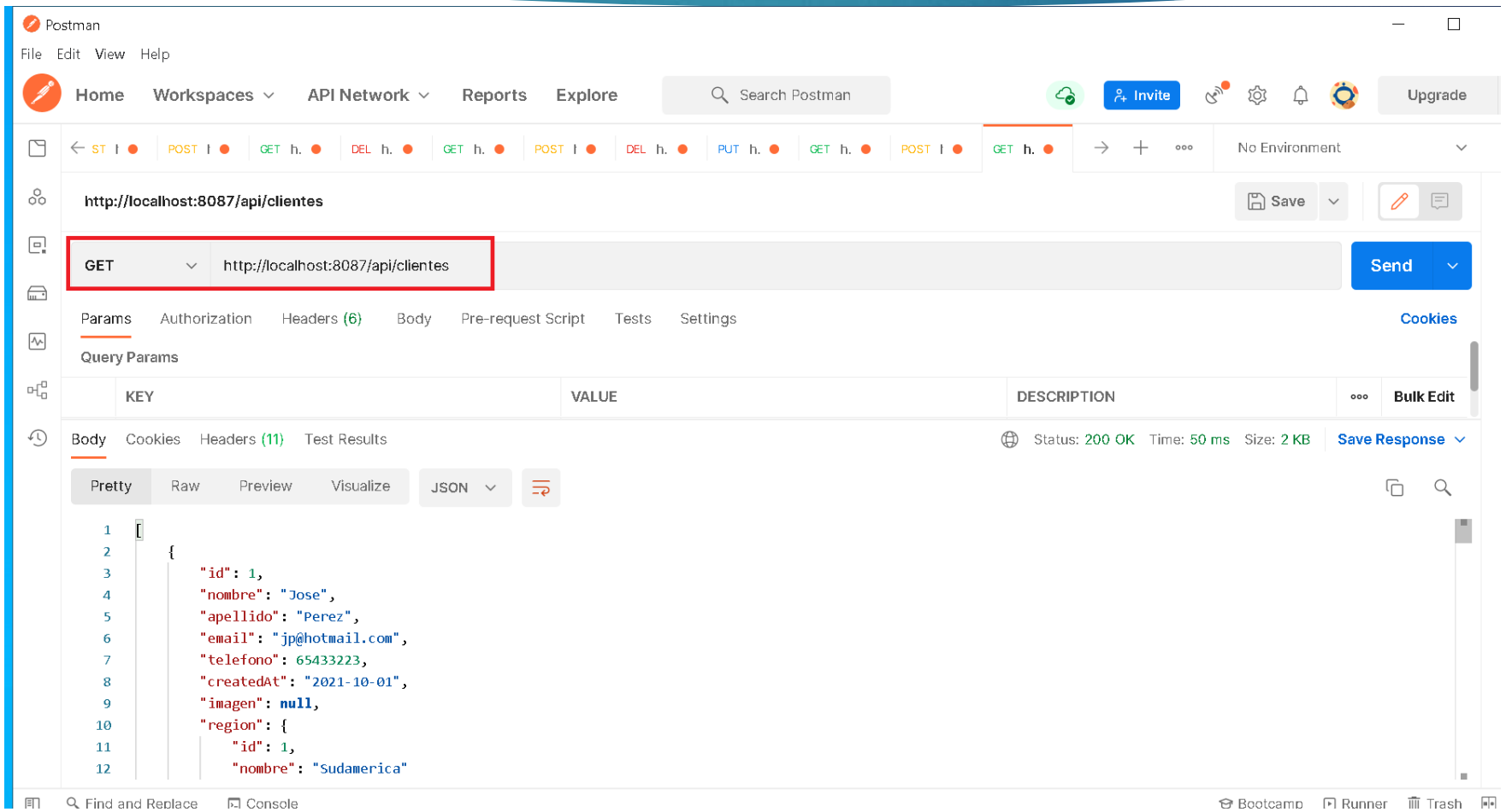


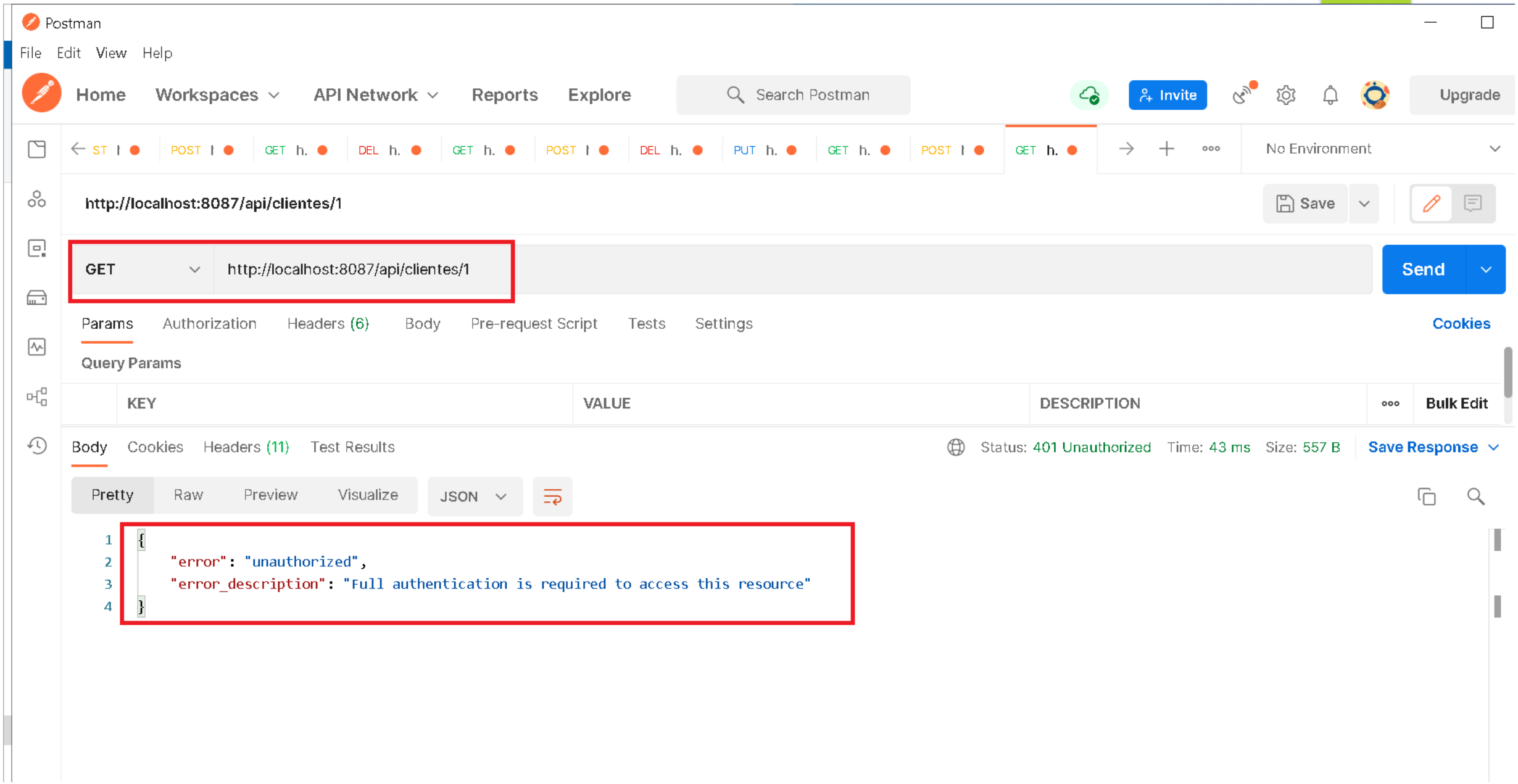




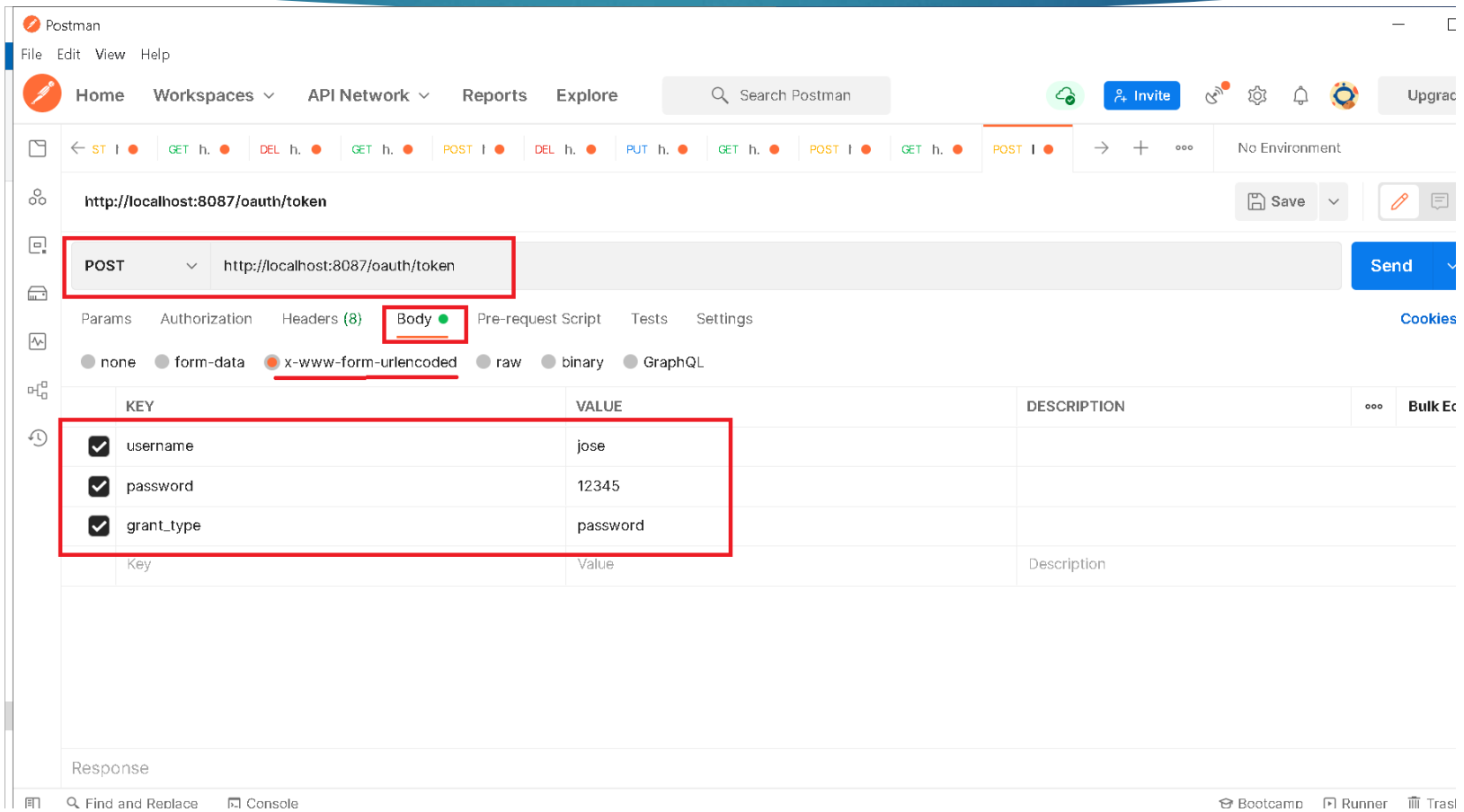
Probamos con Postman

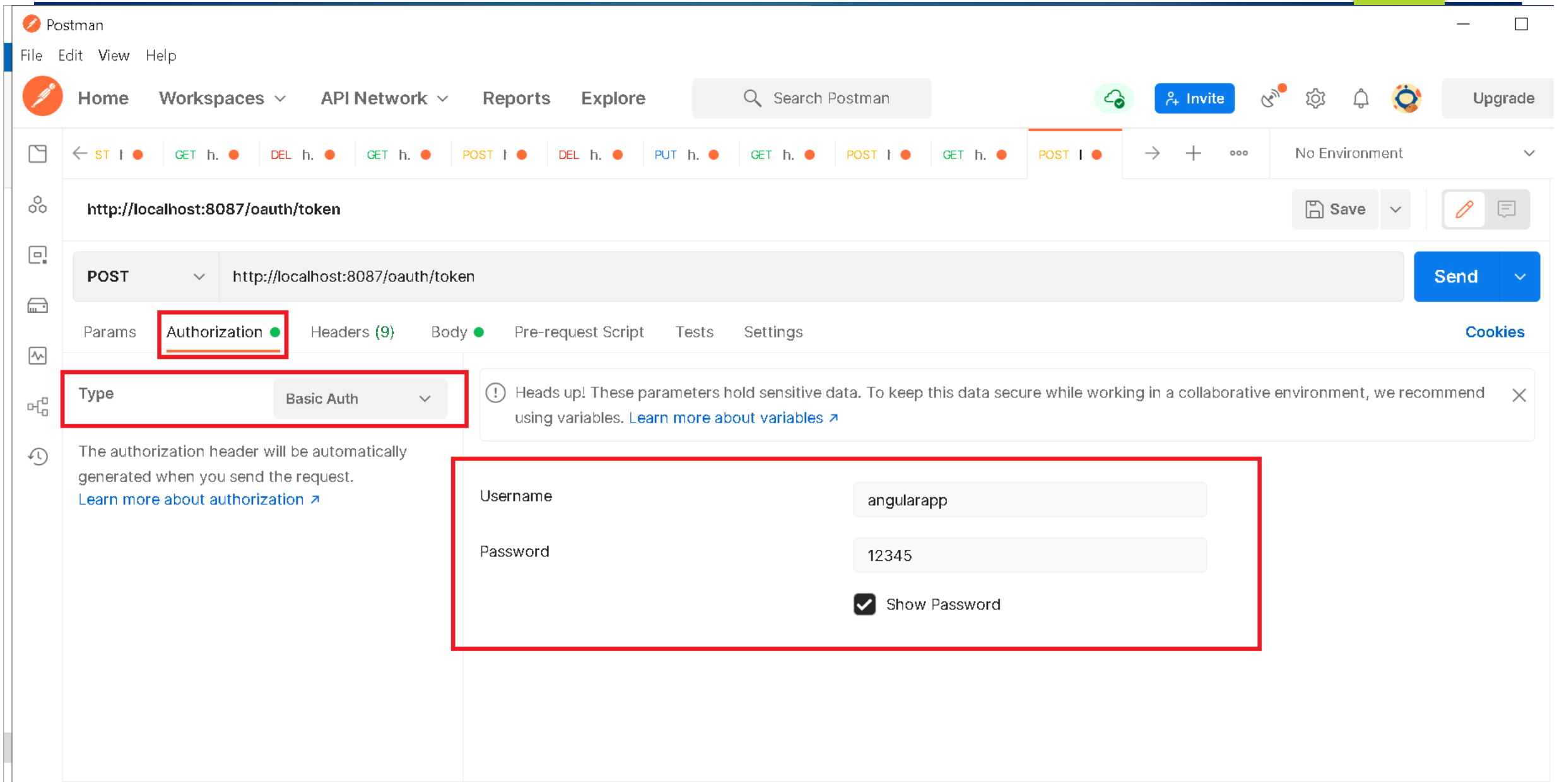
- El único api habilitado es el listado de todos los clientes los demás estarías deshabilitados




Postman
File Edit View Help
Home Workspaces API Network Reports Explore
Search Postman
Invite
Upgrade
No Environment
http://localhost:8087/api/clientes/1
Save
GET
http://localhost:8087/api/clientes/1
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
Query Params
KEY VALUE DESCRIPTION Bulk Edit
Body Cookies Headers (11) Test Results
Status: 401 Unauthorized Time: 43 ms Size: 557 B
Save Response
Pretty Raw Preview Visualize JSON
{
"error": "unauthorized",
"error_description": "Full authentication is required to access this resource"
}

Con Spring Security realizamos la petición de un Token para poder utilizar las demás APIS




Postman
File Edit View Help
Home Workspaces API Network Reports Explore
Search Postman
Invite
Upgrade
No Environment
http://localhost:8087/oauth/token
Save
POST
http://localhost:8087/oauth/token
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies
Type
Basic Auth
Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables
The authorization header will be automatically generated when you send the request.
Learn more about authorization
Username
angularapp
Password
12345
Show Password

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman Invite Upgrade

ST GET h. DEL h. GET h. POST DEL h. PUT h. GET h. POST GET h. POST No Environment

http://localhost:8087/oauth/token Save

POST http://localhost:8087/oauth/token Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (10) Test Results Status: 200 OK Time: 586 ms Size: 1.08 KB Save Response

Pretty Raw Preview Visualize JSON


```
{
    "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJleHAiOjE2MzgyNDU3NDcsInVzZXJfbmFtZSI6Impvc2UiLCJhdXRob3JpdGllcyI6WyJST0xFX1VTRVIiXSwianRpIjoiNzI2NTIzNmMtZGRhOS00YWIzLTlmZjItN2Q5ZTJjYmZkZmFkIiwiY2xpZW50X2lkIjoiYW5ndWxhcmFwcCIsInNjb3BlIjpbInJlYWQiLCJ3cml0ZSJdfQ.Kybr4bUaafg_EmgIYP3VH2sMUuEtTQEmbE_Ke5oc1rs",
    "token_type": "bearer",
    "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJ1c2VyX25hbWUiOiJqb3NlIiwic2NvcGUiOlsicmVhZCIsIndyaXRlIl0sImF0aSI6IjcyNjUyMzZjLWRkYTktNGFiMy05ZmYyLTdkOWUyY2JmZGZhZCIsImV4cCI6MTYzODI0NTc0NywiYXV0aG9yaXRpZXMiOlsiUk9MRV9VU0VSIl0sImp0aSI6ImFkZTkzYjFiLWJkMGItNGRmMS05YzQ0LTg1ZjFlOTU3ZDI3MyIsImNsaWVudF9pZCI6ImFuZ3VsYXJhcHAifQ.
        BwqlFNw7df-90mGKXMQIo5WWixdSoqz-mpHdKqUTA1U",
    "expires_in": 3599,
    "scope": "read write",
    "jti": "7265236c-dda9-4ab3-9ff2-7d9e2cbfdfad"
}
```


Find and Replace Console Bootcamp Runner Trash

Con el token podemos utilizar las todas las apis