

**Package** com.mapmyjourney.backend.security

## Class SecurityConfig

java.lang.Object<sup>↗</sup>  
com.mapmyjourney.backend.security.SecurityConfig

```
@Configuration
@EnableWebSecurity
@EnableMethodSecurity(prePostEnabled=true)
public class SecurityConfig
extends Object↗
```

Configuración de Spring Security con JWT y CORS. - Define rutas públicas (/register, /login, Swagger) - Protege endpoints privados que requieren autenticación - Configura CORS para Angular Frontend (localhost:4200) - Integra JwtAuthenticationFilter

### Constructor Summary

Constructors
Constructor
Description
<a href="#">SecurityConfig()</a>

### Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
	org.springframework.security.authentication.AuthenticationManager	
	<a href="#">authenticationManager</a> (org.springframework.security.config.annotation.web.builders.HttpSecurity http)	Configura el AuthenticationManager con DaoAuthenticationProvider.
	org.springframework.web.cors.CorsConfigurationSource	
	<a href="#">corsConfigurationSource</a> ()	Configura CORS para permitir solicitudes desde el frontend en localhost:4200.
	org.springframework.security.authentication.dao.DaoAuthenticationProvider	
	<a href="#">daoAuthenticationProvider</a> ()	Configura el proveedor de autenticación DAO.
	org.springframework.security.web.SecurityFilterChain	
	<a href="#">filterChain</a> (org.springframework.security.config.annotation.web.builders.HttpSecurity http)	Configura la cadena de filtros de seguridad HTTP.
	org.springframework.security.crypto.password.PasswordEncoder	
	<a href="#">passwordEncoder</a> ()	Define el encoder de contraseñas (BCrypt).

Methods inherited from class java.lang.Object <sup>↗</sup>
<a href="#">clone</a> <sup>↗</sup> , <a href="#">equals</a> <sup>↗</sup> , <a href="#">finalize</a> <sup>↗</sup> , <a href="#">getClass</a> <sup>↗</sup> , <a href="#">hashCode</a> <sup>↗</sup> , <a href="#">notify</a> <sup>↗</sup> , <a href="#">notifyAll</a> <sup>↗</sup> , <a href="#">toString</a> <sup>↗</sup> , <a href="#">wait</a> <sup>↗</sup> , <a href="#">wait</a> <sup>↗</sup> , <a href="#">wait</a> <sup>↗</sup>

### Constructor Details

SecurityConfig
----------------

```
public SecurityConfig()
```

## Method Details

### filterChain

```
@Bean
public org.springframework.security.web.SecurityFilterChain filterChain
(org.springframework.security.config.annotation.web.builders.HttpSecurity http)
    throws Exception
```

Configura la cadena de filtros de seguridad HTTP. Define rutas públicas y privadas, y configura CORS.

**Throws:**

[Exception](#)

### corsConfigurationSource

```
@Bean
public org.springframework.web.cors.CorsConfigurationSource corsConfigurationSource()
```

Configura CORS para permitir solicitudes desde el frontend en localhost:4200. Explícitamente permite el preflight (OPTIONS) que es obligatorio para CORS en navegadores.

### authenticationManager

```
@Bean
public org.springframework.security.authentication.AuthenticationManager authenticationManager
(org.springframework.security.config.annotation.web.builders.HttpSecurity http)
    throws Exception
```

Configura el AuthenticationManager con DaoAuthenticationProvider.

**Throws:**

[Exception](#)

### daoAuthenticationProvider

```
@Bean
public org.springframework.security.authentication.dao.DaoAuthenticationProvider daoAuthenticationProvider()
```

Configura el proveedor de autenticación DAO. Usa CustomUserDetailsService para cargar usuarios de la base de datos.

### passwordEncoder

```
@Bean
public org.springframework.security.crypto.password.PasswordEncoder passwordEncoder()
```

Define el encoder de contraseñas (BCrypt).