

Introducción a Spring Security

Implementa la seguridad de tu API Rest con Spring Boot

Spring Security

- Proyecto paraguas (integra muchos proyectos)
- Ofrece servicios de seguridad para aplicaciones Java EE
- Integración sencilla inmediata con proyectos Spring MVC a través de Spring Boot.
- Responde a dos cuestiones:
 - *Autenticación*: ¿quién eres?
 - *Autorización*: ¿para qué tienes permiso?

pom.xml

- Añadiendo dependencias starter

```
<dependency>
```

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

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

```
</dependency>
```

Spring Security

- Gran aceptación por la comunidad de desarrolladores por su *flexibilidad* en los modelos de autenticación.
- Rápida integración sin necesidad de una migración de los sistemas a un entorno de terceros.
- Plataforma abierta y en constante evolución.

Módulos

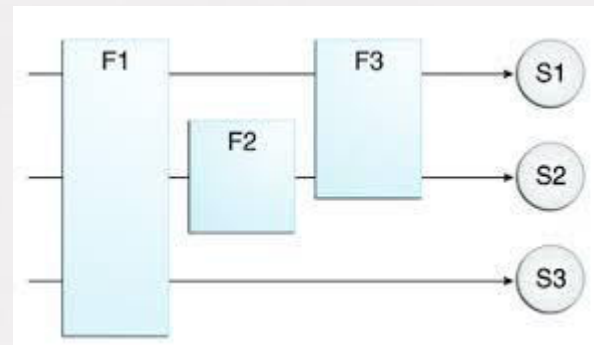
- Desde Spring Security 3, el código se encuentra dividido en diferentes .jars, para separar claramente funcionalidades y dependencias de terceros.
- Algunos de ellos son:
 - Core: contiene los elementos centrales de autenticación y control de acceso.
 - Web: contiene los filtros y el código que articula la infraestructura de seguridad de una aplicación web.

Módulos

- Algunos de ellos son:
 - LDAP: provee los mecanismos necesarios para la autenticación vía LDAP.
 - OAuth 2.0 Core: provee las clases e interfaces para dar soporte al framework OAuth 2.0 y OpenID Connect Core 1.0
 - OAuth 2.0 Client: proporciona el soporte cliente para OAuth 2.0 y OpenID Connect Core 1.0.

Java Filter

- Funcionalidad que se coloca entre el cliente y un servlet.
- Permite dejar pasar una petición, rechazarla o añadir una determinada funcionalidad.
- Uno de sus usos clásicos es la seguridad.



Seguridad web: Filtro

- Durante todo el curso desarrollaremos aplicaciones web (REST) basadas en el API Servlet de Java.
- Spring Security se integra con el contenedor de servlets utilizando un filtro (Filter) estándar.
- Solo necesitamos un contenedor de servlets para utilizar Spring Security.
- De hecho, no es necesario utilizar Spring para poder usar Spring Security:O

Spring Boot y Spring Security

- Al ejecutar un proyecto Spring Boot con la dependencia de Spring Security, suceden varias cosas *automáticamente*.
 - Se habilita la configuración por defecto, a través de un filtro, llamado *springSecurityFilterChain*.
 - Se crea un bean de tipo *UserDetailsService* con un usuario llamado *user* y una contraseña aleatoria que se imprime por consola.
 - Se registra el filtro en el contenedor de servlets para todas las peticiones.

Spring Boot y Spring Security

- Aunque no ha configurado mucho, tiene muchas consecuencias
 - Requiere autenticación para interactuar con nuestra aplicación
 - Genera un formulario de login por defecto.
 - Genera un mecanismo de logout
 - Protege el almacenamiento de la contraseña con BCrypt.
 - Prevé contra ataques CSRF, Session Fixation, Clickjacking...
 - ...