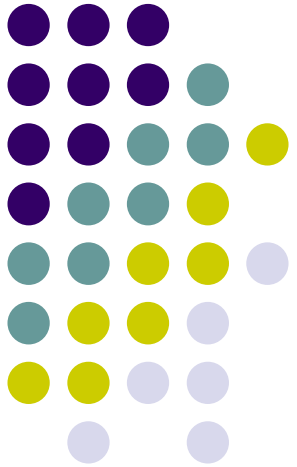
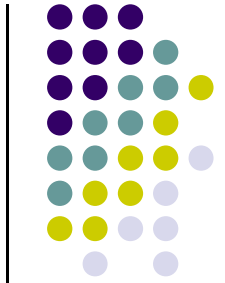
Spring

Security

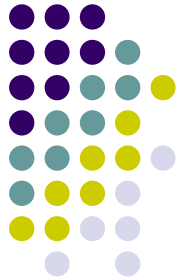


Contenidos



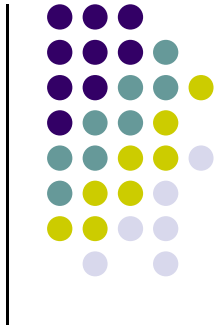
- Intro
- Ejemplo

Intro



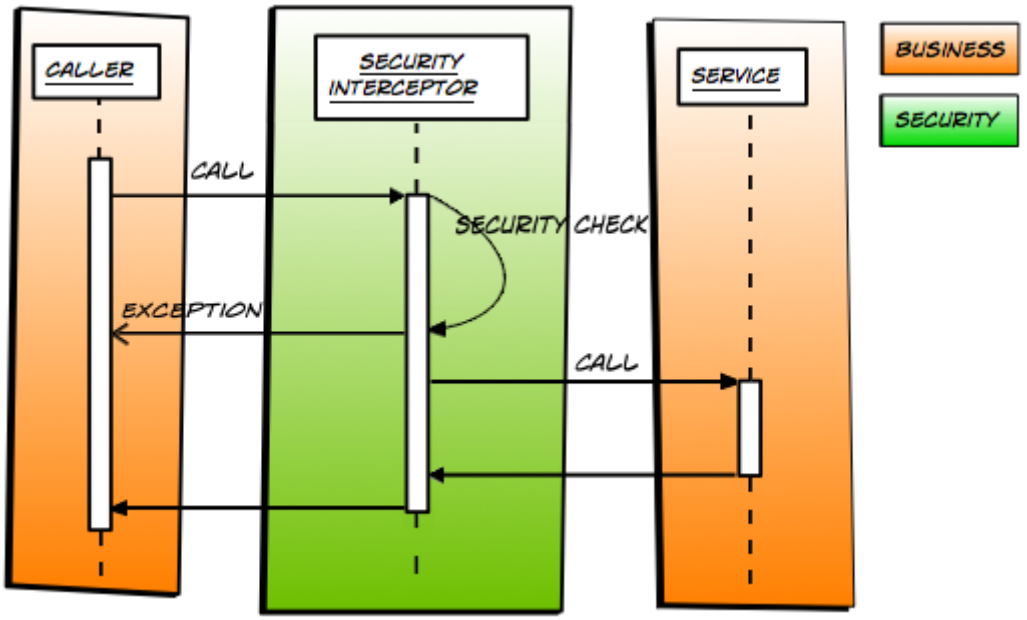
Spring Security proporciona servicios de seguridad para aplicaciones basados en J2EE, enfocado particularmente sobre proyectos construidos usando Spring Framework.

Intro

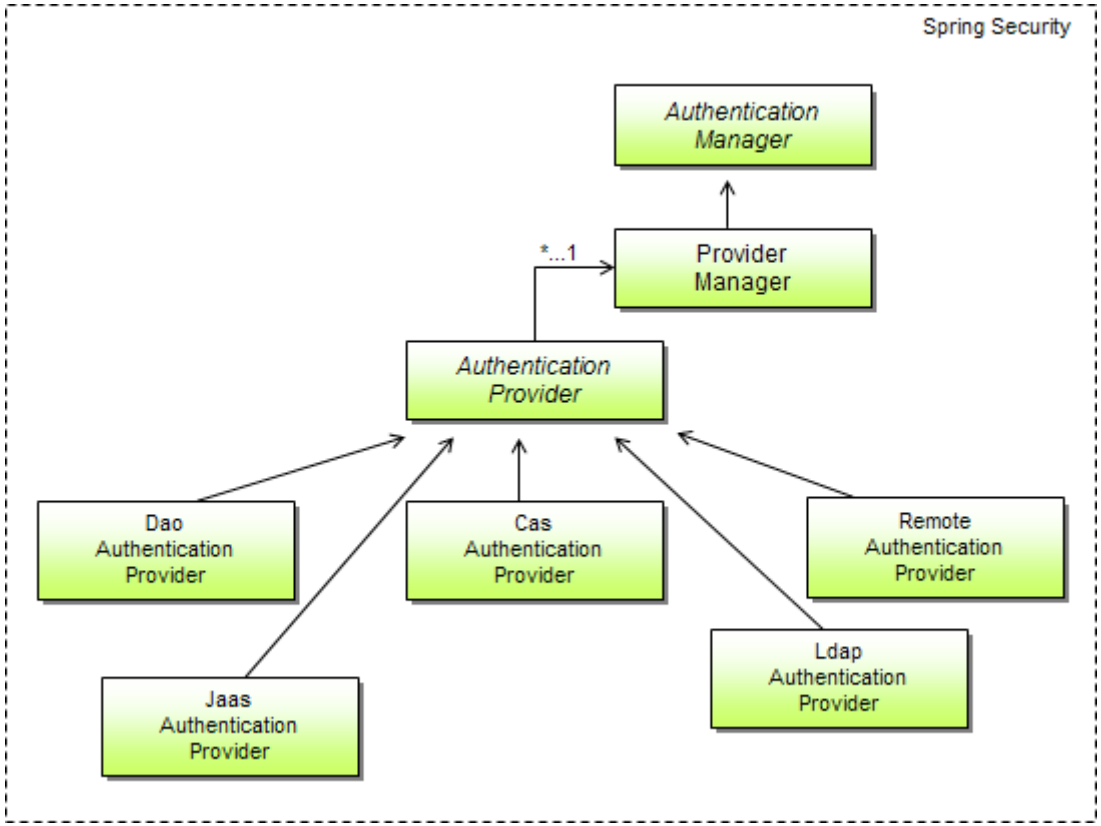


La seguridad comprende dos operaciones: La primera operación es conocida como "autenticación", por el cual se establece si un usuario(que quiere realizar una acción en nuestra aplicación) es quien dice ser, y la segunda operación es llamada "autorización" que se refiere al proceso de decidir si a un usuario le es permitido realizar una determinada acción en nuestra aplicación.

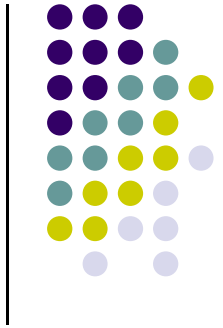
Intro


CALLER
SECURITY INTERCEPTOR
SERVICE
BUSINESS
SECURITY
CALL
SECURITY CHECK
EXCEPTION
CALL

Intro


Spring Security
Authentication Manager
*...1
Provider Manager
Authentication Provider
Dao Authentication Provider
Cas Authentication Provider
Remote Authentication Provider
Jaas Authentication Provider
Ldap Authentication Provider

Intro



HTTP BASIC authentication headers (an IEFT RFC-based standard).

HTTP Digest authentication headers (an IEFT RFC-based standard).

HTTP X.509 client certificate exchange (an IEFT RFC-based standard).

LDAP (un enfoque muy comun para necesidades de autenticación multiplataforma, especificamente en entornos extensos).

Form-based authentication (necesario para interfaces de usuario simples).

OpenID authentication.

Java Authentication and Authorization Service (JAAS)

Container integration with JBoss, Jetty, Resin and Tomcat (tambien podemos usar autenticación gestionada por el contenedor)

OpenNMS Network Management Platform *

AppFuse *

AndroMDA *

Mule ESB

Direct Web Request (DWR) *

Grails

Tapestry

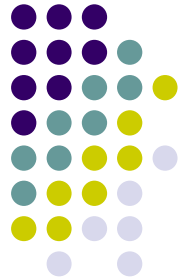
JTrac

Nuestros propios sistemas de autenticación.

Intro

Independientemente de como la autenticación fué realizada, Spring Security proporciona un conjunto amplio de capacidades de autorización. Existen tres áreas principales de interés respecto a la autorización, que son; **autorización basado en solicitudes web**, **autorización basada en que métodos** pueden ser invocados y **autorización de acceso a instancias de objetos** que pertenecen a un dominio individual.

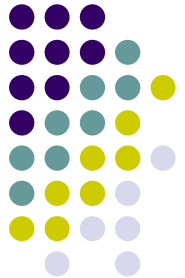
Ejemplo – Web XML



```
<filter>
    <filter-name>springSecurityFilterChain</filter-name>
    <filter-
class>org.springframework.web.filter.DelegatingFilterProxy</filter-
class>
</filter>

<filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

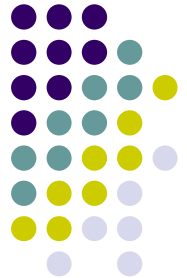
Ejemplo – ApplicationContext-security XML



```
<http use-expressions="true" access-denied-page="/seguridad/deniedpage.html" >
<intercept-url pattern="/usuarios/**"  access="hasRole('Administrador') or hasRole('Gerente')" />
<intercept-url pattern="/**" access="isAuthenticated()" />
</http>
```

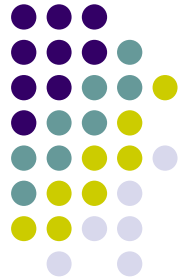
```
<form-login login-page="/seguridad/login.html"  login-processing-url="/j_spring_security_check"
default-target-url="/usuarios/listar.html"   always-use-default-target="true" authentication-failure-
url="/seguridad/login.html?login_error=true" />
<logout  logout-success-url="/seguridad/logoutsuccess.html" />
```

Ejemplo – ApplicationContext-security XML



```
<global-method-security pre-post-annotations="enabled">
<!-- AspectJ pointcut
<protect-pointcut expression="execution(* bigbank.*Service.post*(..))“ access="Administrador"/> -->
</global-method-security>
```

Ejemplo – ApplicationContext-security XML



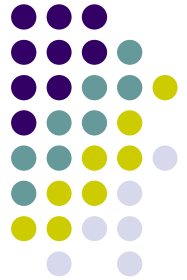
```
<authentication-manager>
  <authentication-provider>
   <password-encoder hash="md5" />
      <user-service>
        <user name=“pedro" password="a564de63c2d0da68cf47586ee05984d7"
            authorities=“Administrador" />
        </user-service>
    </authentication-provider>
</authentication-manager>
```

Ejemplo – Vistas

```
<sec:authentication property="principal.username" />

<sec:authorize access="hasRole('Administrador')">
Ejemplo: Contenido solo para administradores....
</sec:authorize>
```

Ejemplo – Código



```
String nombreUsuario = SecurityContextHolder.getContext()
   .getAuthentication().getName();
```

```
@PreAuthorize("hasRole('Administrador')")
```

+ info http://static.springsource.org/spring-security/site/docs/3.0.x/reference/el-access.html