# **Spring Security Applnfo Plugin - Reference Documentation**

**Authors:** Burt Beckwith

Version: 2.0.0

## **Table of Contents**

1 Introduction to the Spring Security AppInfo Plugin

- 2 Security Configuration UI
- **3** General Notes

## 1 Introduction to the Spring Security Applnfo Plugin

The Spring Security AppInfo plugin provides a UI to inspect your security configuration.



If you already have the spring-security-ui plugin installed you shouldn't install this plugin, sin its own for users who want this information but not the entire UI plugin.

### **Release History**

- December 7, 2015
  - 2.0.0 release
- October 5, 2013
  - 2.0-RC2 release
- February 13, 2010
  - initial 1.0 release

# 2 Security Configuration UI

The plugin has one controller (SecurityInfoController.groovy) and is available by navigating t

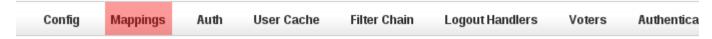
## Configuration

The Configuration menu item displays all security-related attributes in Config.groovy. The names om

Config Mappings Auth User Cache Fi	ilter Chain Logout Handlers Voters Authentica
Name	Value
active	true
adh.ajaxErrorPage	/login/ajaxDenied
adh.errorPage	/login/denied
ajaxHeader	X-Requested-With
anon.key	foo
anon.userAttribute	anonymousUser,ROLE_ANONYMOUS
apf.allowSessionCreation	true
apf.continueChainBeforeSuccessfulAuthentication	false
apf.filterProcessesUrl	/j_spring_security_check
apf.passwordParameter	j_password
apf.postOnly	true
apf.usernameParameter	j_username
atr.anonymousClass	class org.springframework.security.aut
atr.rememberMeClass	class org.springframework.security.aut
auth.ajaxLoginFormUrl	/login/authAjax
auth.forceHttps	false
auth.loginFormUrl	/login/auth
auth useForward	false

## **Mappings**

The Mappings menu item displays the current request mapping mode (Annotation, Requestmap, or Static)



#### SecurityConfigType: Annotation

Name	Value
/secure/admins/**	[ROLE_ADMIN]
/secure/superusers/**	[ROLE_SUPERUSI
/secure/users/**	[ROLE_USER]

#### **Current Authentication**

The Current Authentication menu item displays your Authentication information, mostly for reference

Config	Mappings	Auth	User Cache	Filter Chain	Logout Handlers	Voters	Authentica		
Name	Value								
Authorities	[ROLE_USE	₹]							
Credentials	password1								
Details	org.springfran	org.springframework.security.web.authentication.WebAuthenticationDetails@fffd3270: RemotelpAddres							
Principal	org.springframework.security.ldap.userdetails.LdapUserDetailsImpl@9eadb620: Dn: cn=person1,dc=d1 Enabled: true; AccountNonExpired: true; CredentialsNonExpired: true; AccountNonLocked: true; Grant								
Name	person1								

#### **User Cache**

The User Cache menu item displays information about cached users (this feature is disabled by default):

Config Mappings Auth	User Cache	Filter Chain	Logout Handlers	Voters	Authentica
----------------------	------------	--------------	-----------------	--------	------------

#### Not Caching Users

#### **Filter Chains**

The Filter Chains menu item displays your configured Filter chains. Typically there is just one chain, appli

Config	Mappings	Auth	User Cache	Filter Chain	Logout Handlers	Voters	Authentica
URL Pattern		Filters					
jsk		org.codel org.codel org.spring org.spring org.spring org.spring	haus.groovy.grails haus.groovy.grails gframework.securi gframework.securi gframework.securi gframework.securi	.plugins.spring .plugins.spring ty.web.servleta ty.web.authen ty.web.authen ty.web.access	t.SecurityContextPers gsecurity.MutableLogor gsecurity.RequestHolde api.SecurityContextHo itication.rememberme.f itication.AnonymousAu s.ExceptionTranslation s.intercept.FilterSecurit	utFilter erAuthentica IderAwareRe Rememberlv Ithentication Filter	itionFilter equestFilter leAuthenticati Filter

It is possible to have multiple URL patterns each with its own filter chain, for example when using HTTP 1

### **Logout Handlers**

The Logout Handlers menu item displays your registered LogoutHandlers. Typically there will b implementations, or a plugin might contribute one or more:

Config	Mappings	Auth	User Cache	Filter Chain	Logout Handlers	Voters	Authentica
Logout Hai	ndlers						
org.springframework.security.web.authentication.rememberme.TokenBasedRememberMeServices							
org.spring framework.security.web.authentication.logout.Security Context Logout Handler							

#### **Voters**

The Voters menu item displays your registered AccessDecisionVoters. Typically there will be implementations, or a plugin might contribute one or more:

Config	Mappings	Auth	User Cache	Filter Chain	Logout Handlers	Voters	Authentica		
Voters									
org.springfra	org.springframework.security.access.vote.AuthenticatedVoter								
org.springframework.security.access.vote.RoleHierarchyVoter									
org.codehaus.groovy.grails.plugins.springsecurity.WebExpressionVoter									

#### **Authentication Providers**

The Authentication Providers menu item displays your registered AuthenticationProviders. Typic own custom implementations, or a plugin (e.g. LDAP) might contribute one or more:

Config	Mappings	Auth	User Cache	Filter Chain	Logout Handlers	Voters	Authentica

#### **Authentication Providers**

 $org.spring framework.security.authentication.dao.Dao Authentication Provider \\ org.spring framework.security.authentication. Anonymous Authentication Provider \\ org.spring framework.security.authentication. Remember Me Authentication Provider \\ org.spring framework.security.authentication. Remember Me Authentication Provider \\ org.spring framework.security.authentication. \\ Org.spring framework.security.authentication Provider \\ org.spring framework.security.authentication. \\ Org.spring framework.security.authentication Provider \\ org.spring framework.security.authentication Provid$ 

## 3 General Notes

#### **Securing Access**

Be sure to guard access to the /securityInfo url since only authorized users should have access to th in the staticRules property in Config.groovy:

If you use database Requestmaps, create new ones:

```
new Requestmap(url: '/securityinfo', configAttribute: 'ROLE_ADMIN').save()
new Requestmap(url: '/securityinfo.*', configAttribute: 'ROLE_ADMIN').save()
new Requestmap(url: '/securityinfo/**', configAttribute: 'ROLE_ADMIN').save()
```

And if you use the interceptUrlMap approach, add mappings to that property in Config.groovy:

```
grails.plugin.springsecurity.interceptUrlMap = [
...
'/securityinfo': ['ROLE_ADMIN'],
'/securityinfo/**': ['ROLE_ADMIN'],
...
]
```