# Spring Security AppInfo Plugin - Reference Documentation

**Authors:** Burt Beckwith

**Version:** 2.0.0

## Table of Contents

# 1 Introduction to the Spring Security AppInfo 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

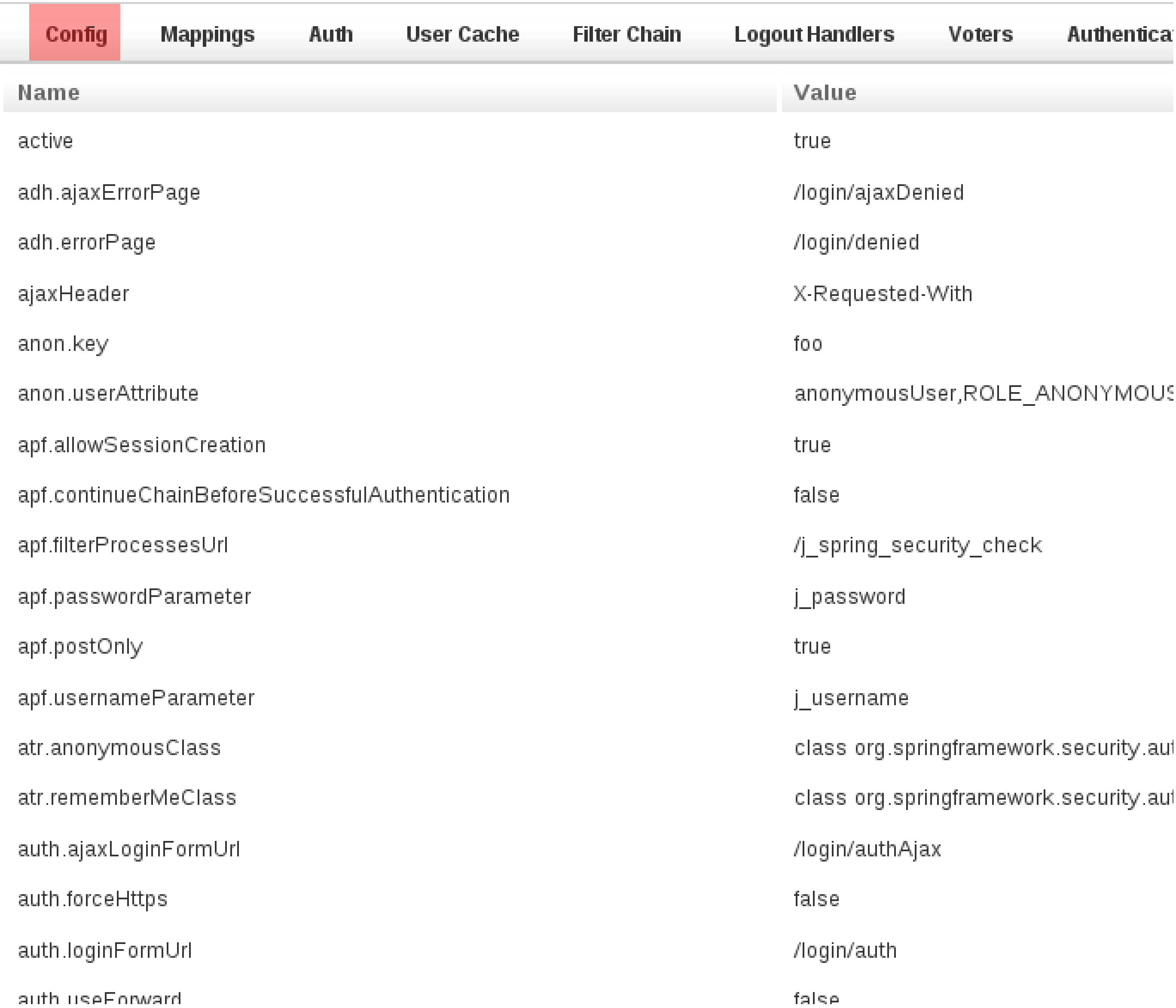
- December 7, 2015
  - 2.0.0 release
- October 5, 2013
  - 2.0-RC2 release
- February 13, 2010
  - initial 1.0 release

# 2 Security Configuration UI

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

## Configuration

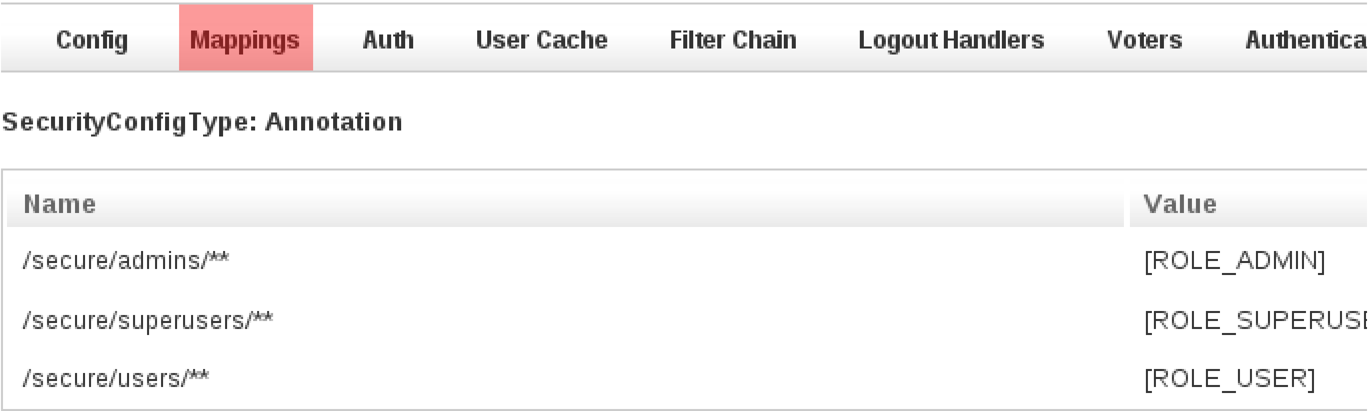
The Configuration menu item displays all security-related attributes in `Config.groovy`. The names om



| 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)

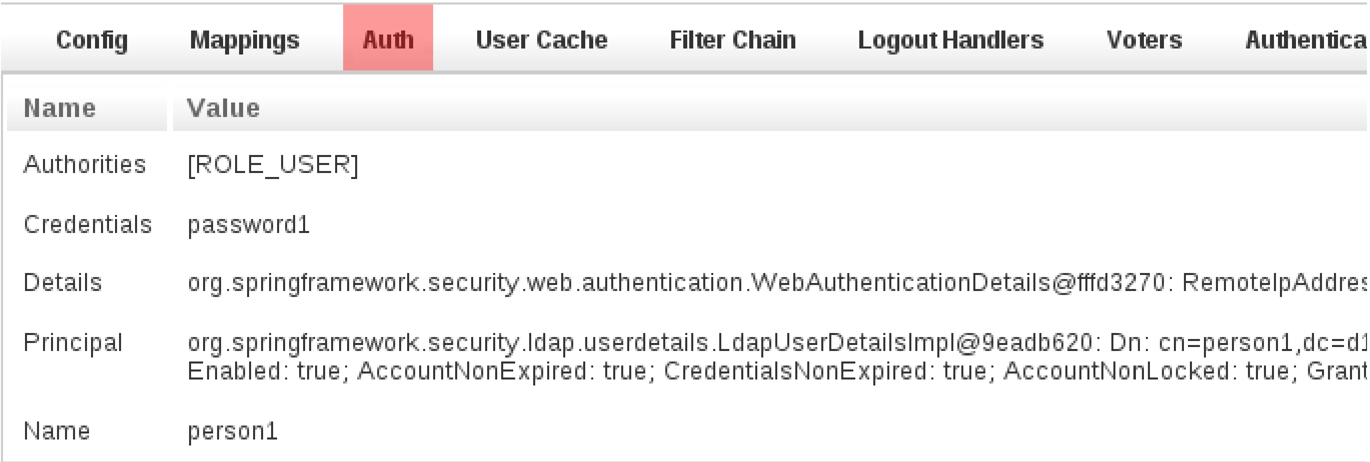


**SecurityConfigType: Annotation**

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

## Current Authentication

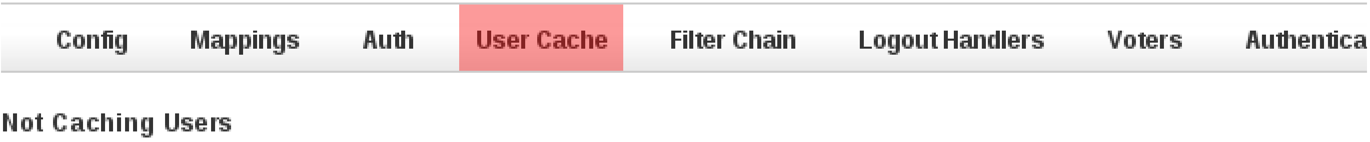
The Current Authentication menu item displays your `Authentication` information, mostly for referen



| Name | Value |
|---|---|
| Authorities | [ROLE_USER] |
| Credentials | password1 |
| Details | org.springframework.security.web.authentication.WebAuthenticationDetails@fffd3270: RemoteIpAddres |
| 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

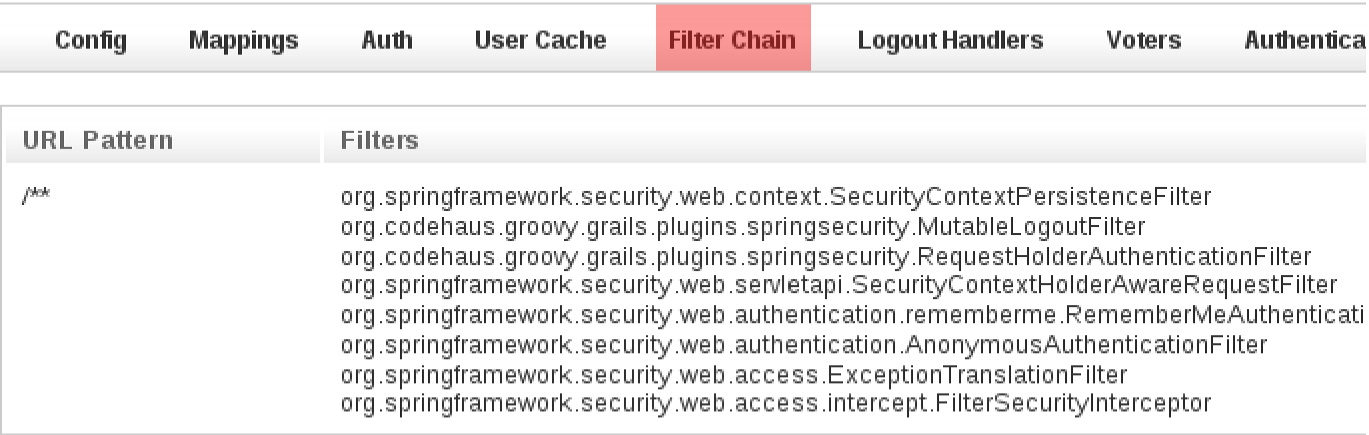
The User Cache menu item displays information about cached users (this feature is disabled by default):



**Not Caching Users**

## Filter Chains

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

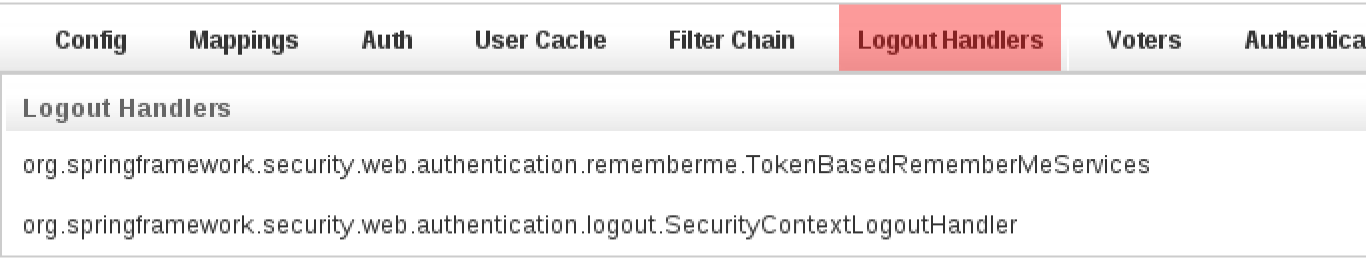


| URL Pattern | Filters |
|---|---|
| /** | org.springframework.security.web.context.SecurityContextPersistenceFilter<br>org.codehaus.groovy.grails.plugins.springsecurity.MutableLogoutFilter<br>org.codehaus.groovy.grails.plugins.springsecurity.RequestHolderAuthenticationFilter<br>org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter<br>org.springframework.security.web.authentication.rememberme.RememberMeAuthenticati<br>org.springframework.security.web.authentication.AnonymousAuthenticationFilter<br>org.springframework.security.web.access.ExceptionTranslationFilter<br>org.springframework.security.web.access.intercept.FilterSecurityInterceptor |

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

## Logout Handlers

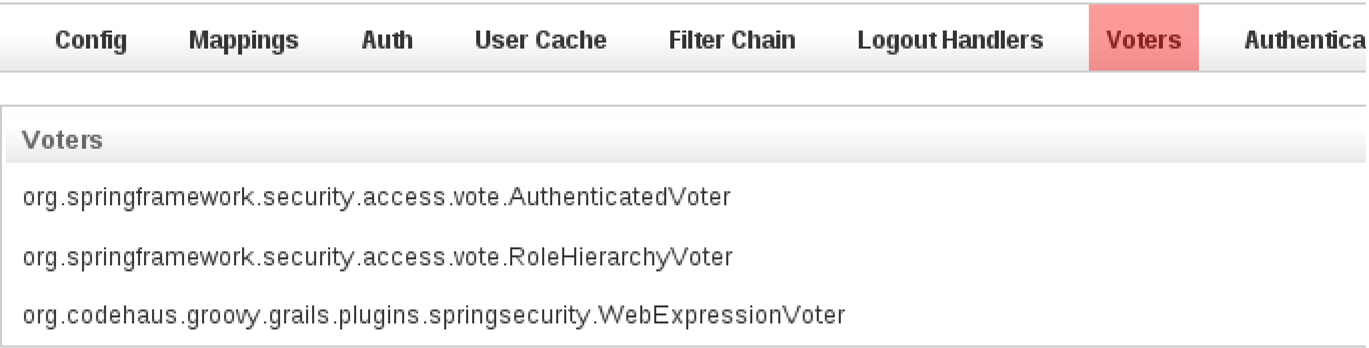
The Logout Handlers menu item displays your registered `LogoutHandlers`. Typically there will b implementations, or a plugin might contribute one or more:



| Logout Handlers |
|---|
| org.springframework.security.web.authentication.rememberme.TokenBasedRememberMeServices |
| org.springframework.security.web.authentication.logout.SecurityContextLogoutHandler |

## Voters

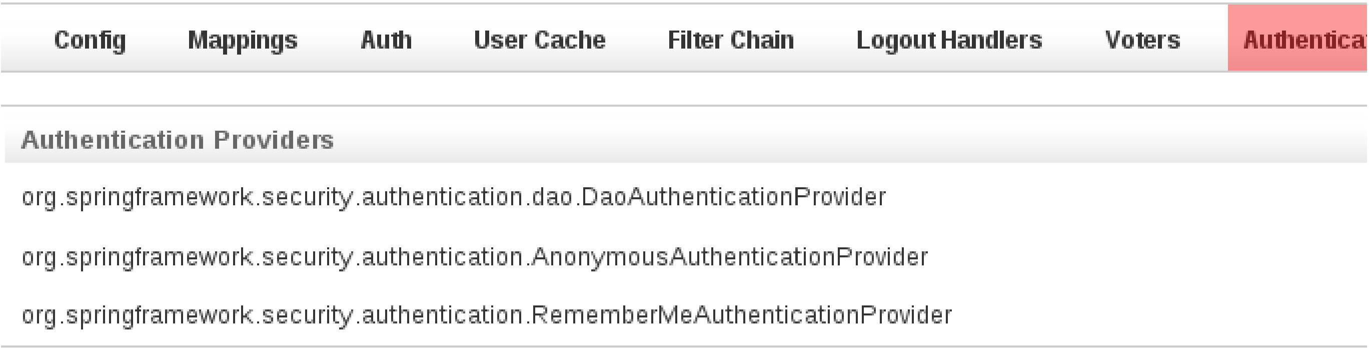
The Voters menu item displays your registered `AccessDecisionVoters`. Typically there will be implementations, or a plugin might contribute one or more:



| Voters |
|---|
| 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`. Typi own custom implementations, or a plugin (e.g. LDAP) might contribute one or more:



Authentication Providers

org.springframework.security.authentication.dao.DaoAuthenticationProvider

org.springframework.security.authentication.AnonymousAuthenticationProvider

org.springframework.security.authentication.RememberMeAuthenticationProvider

# 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`:

```
grails.plugin.springsecurity.controllerAnnotations.staticRules = [
   …
   '/securityinfo':    ['ROLE_ADMIN'],
   '/securityinfo/**': ['ROLE_ADMIN'],
   …
]
```

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'],
   …
]
```