

Jasig CAS support for the Spring Security plugin.

# Spring Security ACL Plugin - Reference Documentation

**Authors:** Burt Beckwith

**Version:** 1.0.3

## Table of Contents

- 1** Introduction
  - 1.1** History
- 2** Usage
- 3** Configuration

# 1 Introduction

The CAS plugin adds [CAS](#) single sign-on support to a Grails application that uses Spring Security. It depends on the CAS client library.

Once you have configured a CAS server and have configured your Grails application(s) as clients, you can use the CAS client library to connect to the CAS server and be automatically authenticated to all other clients.

## 1.1 History

- Version 1.0.3
  - released July 4, 2012
- Version 1.0.2
  - released February 12, 2011
- Version 1.0.1
  - released September 1, 2010
- Version 1.0
  - released July 27, 2010
- Version 0.1
  - released June 18, 2010

## 2 Usage



Configuring your CAS server is beyond the scope of this document. There are many different configurations that most likely be done by IT staff. It's assumed here that you already have a running CAS server.

[CAS](#) is a popular single sign-on implementation. It's open source and has an Apache-like license, and is highly configurable. In addition it has clients written in Java, .Net, PHP, Perl, and other languages.

There isn't much that you need to do in your application to be a CAS client. Just install this plugin, and configure the optional parameters you want in Config.groovy. These are described in detail in [Chapter 3](#) but typically you

```
grails.plugins.springsecurity.cas.loginUri = '/login'
grails.plugins.springsecurity.cas.serviceUrl = 'http://localhost:8080/your-app-name'
grails.plugins.springsecurity.cas.serverUrlPrefix = 'https://your-cas-server/cas'
grails.plugins.springsecurity.cas.proxyCallbackUrl = 'http://localhost:8080/your-app-name'
grails.plugins.springsecurity.cas.proxyReceptorUrl = '/secure/receptor'
```

where "your-app-name" is the Grails application context (will be blank if deployed as the default context) and "your-cas-server" is the URL of the CAS server.

### Single Signout

Single signout is enabled by default and enables signing out for all CAS-managed applications with one click. It is configured by the `afterLogoutUrl` parameter, for example:

```
grails.plugins.springsecurity.logout.afterLogoutUrl =
    'https://your-cas-server/cas/logout?url=http://localhost:8080/your-app-name/'
```

With this configuration, when a user logs out locally by navigating to `/logout/` they'll then be redirected to the local URL. When the whole process is finished they'll be redirected to the subsequent secure URLs at the local server or other CAS-managed servers will require a new login.

If you don't want the single signout filter registered, you can disable the feature:

```
grails.plugins.springsecurity.cas.useSingleSignout = false
```

## 3 Configuration

There are a few configuration options for the CAS plugin.



All of these property overrides must be specified in `grails-app/conf/Config.groovy` with the `grails.plugins.springsecurity` suffix, for example

```
grails.plugins.springsecurity.cas.serverUrlPrefix =  
    'https://cas-server/cas'
```

Name	Default	Meaning
<code>cas.active</code>	<code>true</code>	whether the plugin is enabled or not (e.g. to disable)
<code>cas.serverUrlPrefix</code>	<code>null</code> , must be set	the 'root' of all CAS server URLs, e.g. <code>https://</code>
<code>cas.loginUri</code>	<code>null</code> , must be set	the login URI, relative to <code>cas.serverUrlPrefix</code>
<code>cas.sendRenew</code>	<code>false</code>	if true, ticket validation will only succeed if it was issued from a single sign-on session. Analogue to <code>Spring Security</code>
<code>cas.serviceUrl</code>	<code>null</code> , must be set	the local application URL, e.g. <code>http://localhost:8080/myapp/j_spring_security_check</code>
<code>cas.key</code>	'grails-spring-security-cas', should be changed	used by <code>CasAuthenticationProvider</code> to identify tickets
<code>cas.artifactParameter</code>	'ticket'	the ticket login url parameter
<code>cas.serviceParameter</code>	'service'	the service login url parameter
<code>cas.filterProcessesUrl</code>	'/j_spring_cas_security_check'	the URL that the filter intercepts for login
<code>cas.proxyCallbackUrl</code>	<code>null</code> , should be set	proxy callback url, e.g. 'http://localhost:8080/myapp/secure/receptor'
<code>cas.proxyReceptorUrl</code>	<code>null</code> , should be set	proxy receptor url, e.g. '/secure/receptor'
<code>cas.useSingleSignout</code>	<code>true</code>	if true a <code>org.jasig.cas.client.session.SessionManager</code> is used in <code>web.xml</code>