AcegiSecurity Plugin - Upgrading from 0.2.x to 0.3

Added by Burt Beckwith, last edited by Burt Beckwith on May 02, 2008

> A user upgrading from v0.2.1 to v0.3.0 saw some strange behavior because one of the updated scripts wasn't compiled and the old cached script was running. Before upgrading run 'grails clean' and delete the script cache (C:\.grails\1.0.2\scriptCache on Windows, $home/.grails/1.0.2/scriptCache on Linux).

If you're using version 0.2.x of the plugin and want to upgrade to version 0.3.0, you'll need to make a few changes. The install script only unpacks the plugin zip, so there's not much to do to upgrade:

- delete the current plugin zip and expanded folder from the plugins folder
- install the new plugin:

```
grails install-plugin acegi
```

- rename your application's grails-app/conf/AcegiConfig.groovy to SecurityConfig.groovy
- in SecurityConfig.groovy, change

```
acegi {
   loadAcegi = true
   ...
}
```

to

```
security {
   active = true
   ...
}
```

- in RegisterController.groovy, change

```
def cnf = authenticateService.getAcegiConfig()
def defaultRole = cnf.acegi.defaultRole
def useMail = cnf.acegi.useMail
```

to

```
def config = authenticateService.securityConfig
def defaultRole = config.security.defaultRole
def useMail = config.security.useMail
```

- in LoginController.groovy, update the Ajax check from

```
//is ajax access?
def cnf = authenticateService.getAcegiConfig()
def ajaxHeader = cnf.acegi.ajaxHeader
def AJAX_HEADER="${ajaxHeader}"
```

```
boolean isAjax=false;
if(session["ACEGI_SAVED_REQUEST_KEY"]){
  def itr =  session["ACEGI_SAVED_REQUEST_KEY"].getHeaderValues(AJAX_HEADER)
  if(itr.hasNext()) isAjax=true
}
```

to

```
boolean isAjax = authenticateService.isAjax(request)
```

- in LogoutController.groovy, change

```
j_acegi_logout
```

to

```
j_spring_security_logout
```

- in auth.gsp, change

```
j_acegi_security_check

and

_acegi_security_remember_me
```

to

```
j_spring_security_check

and

_spring_security_remember_me
```

- Replace session keys 'ACEGI_SECURITY_LAST_EXCEPTION' and 'ACEGI_SECURITY_LAST_USERNAME' with 'SPRING_SECURITY_LAST_EXCEPTION' and 'SPRING_SECURITY_LAST_USERNAME' in LoginController.groovy and any of your code that uses them
- User.email_show was renamed to User.emailShow, so change all references to this variable (views/user/*.gsp, views/register/*.gsp, RegisterController.groovy, User.groovy (or whatever your User/Person class is called)
- if you're calling AuthBase.groovy, AuthenticateService.groovy, or AuthorizeTagLib.groovy from your code, they've been moved out of the default package, so you'll have to add imports for them:
  - import org.grails.plugins.springsecurity.service.AuthenticateService
  - import org.grails.plugins.springsecurity.controller.AuthBase
  - import org.grails.plugins.springsecurity.taglib.AuthorizeTagLib
- If you're referencing older Acegi classes (org.acegisecurity) in your code, update the packages to the new names (org.springframework.security), e.g. org.acegisecurity.DisabledException in LoginController.groovy
- If you've configured logging in Config.groovy, you'll need to adjust the packages there (log4j.logger.org.acegisecurity -> log4j.logger.org.springframework.security)

> ℹ After making the above code changes, search your entire codebase for 'acegi' - there should be no instances other than in a few of the comments. Once Acegi became an official Spring module, everything was renamed and no code should have 'acegi' in it, and the plugin follows the same convention.

Finally, if you're using Eclipse, remove the old plugin's source folders and jar files from your classpath and add these:
- add these plugin folders as Source Folders:
  - plugins/acegi-0.3/grails-app/controllers
  - plugins/acegi-0.3/grails-app/services
  - plugins/acegi-0.3/grails-app/taglib
  - plugins/acegi-0.3/src/groovy
  - plugins/acegi-0.3/src/java

- and these jars:
  - plugins/acegi-0.3/lib/commons-codec-1.3.jar
  - plugins/acegi-0.3/lib/spring-security-core-2.0.0.jar
  - plugins/acegi-0.3/lib/spring-security-core-tiger-2.0.0.jar
  - plugins/acegi-0.3/lib/spring-security-openid-2.0.0.jar
- and bin-groovy (or wherever you've configured the Groovy plugin to compile classes in the 'Groovy compiler output location' property) as a Class Folder

If you're using another IDE, perform the corresponding steps.

Comments (2)

ejy says: After you delete the zip file and the expanded acegi dir in the plugins dir, and...   Apr 30, 2008

After you delete the zip file and the expanded acegi dir in the plugins dir, and then

When you issue the command

grails install-plugin acegi

It upgrades to acegi-0.2.1 not acegi-0.3

I think the command should be

grails install-plugin [path to acegi-0.3.zip]

This will install the acegi-0.3 plugin.

Burt Beckwith says: Sorry, the docs are a little out of sync with the plugin - I updated them for th...   Apr 30, 2008

Sorry, the docs are a little out of sync with the plugin - I updated them for the 0.3 release, but that hasn't happened yet - it'll be tagged soon. For now you can download the plugin manually (http://plugins.grails.org/grails-acegi/trunk/grails-acegi-0.3.zip) and run "grails install-plugin grails-acegi-0.3.zip)

Powered by a free Atlassian Confluence Open Source Project License granted to The Codehaus. Evaluate Confluence today.

Printed by Atlassian Confluence 2.9.2, the Enterprise Wiki.