Bruce Phillips Blog on Java, ColdFusion, Flex and Spry

An Introduction to Shiro (formerly JSecurity) – A Beginner's Tutorial Part 3

Posted At : April 5, 2009 3:17 PM | Posted By : Bruce Phillips
Related Categories: **Java**

Introduction

NOTE: Updated in January 2011.

In **part 2 of this tutorial**, I demonstrated how to use Apache Shiro (formerly JSecurity and also called Ki) to add basic security to a web application. In part 3, I show how to configure Shiro to secure different parts of the web application based on a user's role.

One of Shiro's many features is an ability to restrict areas of a web application to be available not just to authenticated (logged in) users but to authenticated users that have a specific role such as admin.

Part 3 Example Application

You can download the **part 3 example project**, which is an archived Eclipse web project (uses Maven). This example is based on the example project built in the tutorial parts 1 and 2. Be sure you read parts 1 and 2 of the tutorial (links at the bottom of this page) and that you've successfully configured the Derby database (see part 1).

In part 3's example project, I reconfigured the web folders into a secure folder (secure/index.jsp) and an admin folder (admin/index.jsp and admin/users.jsp). Users who are logged in and have a role of user can access the JSPs in the secure folder, but cannot access the JSPs in the admin folder. Only users logged in and have a role of admin can view the pages in the admin folder.

You can import the downloaded archived project (named rolesecurity) into Eclipse and then run it on a Tomcat server.

You can also use the Maven jetty plugin (see reference below for how to install Maven if you've don't already have Maven) to run the web application if you're not using Eclipse and Tomcat. Just open a command window and navigate to where you unzipped the rolesecurity_mvn.zip download. Make sure you're in the rolesecurity directory. Then do the following (in this example I unzipped rolesecurity_mvn.zip to c:\jsecurity_examples):

c:\jsecurity_examples\rolesecurity\mvn clean

c:\jsecurity_examples\rolesecurity\mvn jetty:run

Once you see [INFO] Started Jetty Server in the command window, open your web browser and go to this URL: **http://localhost:8080/rolesecurity/** . You should see the contents of the index.jsp. To stop the Jetty server type control-c in the command window.

Since this web application has security based on user roles you should NOT be able to open the web pages that are in the admin folder (admin/index.jsp and admin/users.jsp) without first logging in as user **bruce@hotmail.com** and bruce (password). If you log in using user **sue@hotmail.com** and sue (password), you can only view the pages in the secure folder.

If you try to visit a web page for which you don't have the correct role, you'll be redirected to /unauthorized.jsp.

Configuring Shiro To Use Role Security

To add security based on roles, I did the following:

Add a database table named user_roles with columns named username and role_name to the Derby database. I inserted records into this table for my two users (**bruce@hotmail.com** and **sue@hotmail.com**). Bruce has a role of admin and a role of user. Sue only has a role of user. Giving Bruce both roles enables him to log in and visit pages located in both the secure and admin folders.

Because I'm following Shiro's defaults for where it looks for user roles when using a realm based on a database, my table had to be named user_roles and the columns in the table had to be named username and role_name. If your project cannot follow Shiro's defaults, you can configure Shiro to use your projects conventions (see the Shiro references below).

Note Shiro will automatically query the user_roles table when a a user logs in to determine what roles to associate with the authenticated user.

Then I had to change the configuration for IniShiroFilter (see part 2) in web.xml. Here's the code that changed:

```
[filters]
roles.unauthorizedUrl = /unauthorized.jsp

[urls]
/secure/** = authc, roles[user]
/admin/** = authc, roles[admin]
```

The changes to part 2's web.xml are in red.

The roles.unauthorizedUrl filter specifies where the IniShiroFilter should forward the user to if he tries to access a web

ARCHIVES BY SUBJECT

ColdFusion (26) [RSS]
Flex (75) [RSS]
Java (76) [RSS]
Running (10) [RSS]
Spry (13) [RSS]

CALENDAR

<< August 2012 >>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

LATEST FROM CFBLOGGERS.ORG

ColdFusion e-Seminar on Securing Application
ColdFusion Roadmap is available
Slides & Recording of e-seminar on ColdFusion 10 & security enhancements
Notes on Practical Django Projects (Part 1)
Pidgin, SIPE, and Read Error on Ubutnu 12.04

LINKS

Bruce Phillips' Main Web

Blog Home

Flex.org - The Directory for Flex

NAVIGATION

Home

RECENT ENTRIES

No recent entries.

RECENT COMMENTS

Android App Development - Using The ActionBar Widget
mike said: That was fantastic, been looking for that for a while now. thanks for putting in the effort. regards **[More]**

Using Blackboard Learn 9 Web Services - Part 4 Getting, Creating, Updating, Deleting Users
Bruce said: Baci - I"ve not used that method before and don't know much about course organizations. **[More]**

Using Blackboard Learn 9 Web Services - Part 4 Getting, Creating, Updating, Deleting Users
Baci said: Bruce, Thanks for being really the only resource as we're trying to implement some Bb webservic stuf... **[More]**

An Introduction to Shiro (formerly

page but doesn't have the correct role.

The line /secure/** = authc, roles[admin] tell the IniShiroFilter to restrict the JSPs in the /secure/ folder to only users logged and that have a role of user. The line after that one does the same for the admin folder, but a logged in user must have a role of admin to visit those pages. Note you CANNOT do: /secure/** = authc, roles[user,admin] to indicate that users with either the admin or the user role can view pages in the secure folder.

In my Servlet GetAllUsers, I now needed to verify that the user making the request is logged in and has the role of admin. If you examine the code in the doPost method of GetAllUsers you'll see that I just needed to call the Subject class's hasRole method. This method returns true if the Subject has the role passed in as a parameter, otherwise it returns false.

Summary

It was simple to refactor the part 2 web application to use Shiro's role security. Please note that I'm only scratching the surface of Shiro's capabilities. Consult the references below for much more information. When you download Shiro, you'll receive several sample applications that use more advanced features. Also be sure to review the JavaDoc for the Shiro classes as they contain much useful information.

What's Next?

In **part 4 of the tutorial**, I'll explain how to use some of the custom tags provided by Shiro.

References:

1. An Introduction to Shiro (formerly JSecurity) – A Beginner's Tutorial Part 2, **http://www.brucephillips.name/blog/index.cfm/2009/4/5/An-Introduction-to-Ki-formerly-JSecurity--A-Beginners--Tutorial-Part-2**
2. Role Security Example Application, **http://www.brucephillips.name/jsecurity_examples/rolesecurity_mvn.zip**
3. Apache Shiro **http://shiro.apache.org/**
4. Apache Shiro API, **http://shiro.apache.org/static/current/apidocs/**
5. Apache Shiro Mailing Lists, **http://shiro.apache.org/mailing-lists.html**
6. Presentation on JSecurity to the Charlotte Java Users Group, **http://www.jsecurity.org/files/JSecurity.pdf**
7. Apache Derby, **http://db.apache.org/derby/**
8. Apache Tomcat, **http://tomcat.apache.org/**
9. Jetty, **http://jetty.mortbay.org/jetty5/index.html**
10. Apache Software Foundation, Apache incubator, **http://incubator.apache.org/projects/ki.html**
11. Maven: The Definitive Guide, **http://www.sonatype.com/books/maven-book/reference/public-book.html**
12. Developing with Eclipse and Maven, **http://www.sonatype.com/books/m2eclipse-book/reference/index.html**

Related Blog Entries

- **An Introduction to Shiro (formerly JSecurity) – A Beginner's Tutorial Part 4** (April 5, 2009)
- **An Introduction to Shiro (formerly JSecurity) – A Beginner's Tutorial Part 2** (April 5, 2009)

Comments (Comment Moderation is enabled. Your comment will not appear until approved.)

It seems Shiro need another table - ROLES_PERMISSIONS - in the DerbyDB in order to make the role based authorization. This table should have the following two columns:
ROLE_NAME
PERMISSION

 # Posted By Richard | 9/18/09 3:38 PM

Richard:

Did the example application from part 3 not work for you?

In part 5 I discuss using permissions and describe the changes needed for the database.

For part 3 you don't need the roles_permissions table.

 # Posted By Bruce | 9/18/09 3:46 PM

JSecurity) – A Beginner's Tutorial Part 4
mark said: In secure/index.jsp had to change jsec:principal to shiro:principal. **[More]**

NetBeans Java IDE 7.2 - Comments From An Eclipse User
Ralf Eichinger said: I used Eclipse for years now and tried to switch to Netbeans 7.1.2. Basically Netbeans does the job,... **[More]**

RSS

RSS FEED

SEARCH

Search

SUBSCRIBE

Enter your email address to subscribe to this blog.

Subscribe

TAGS

coldfusion **flex** **java** running **spry**