

Bruce Phillips Blog on Java, ColdFusion, Flex and Spry

An Introduction to Shiro (formerly JSecurity) – A Beginner's Tutorial Part 1

Posted At : April 5, 2009 12:22 PM | Posted By : Bruce Phillips

Related Categories: [Java](#)

Introduction

NOTE: Updated in January 2011.

I recently took over a project that used Apache Shiro for web application security. Shiro was previously known as JSecurity (and briefly also called Ki). Not having used Shiro before, I needed to do some research and learn the basics. This blog entry is designed to assist other developers in applying Shiro to a web application. Please understand that I'm just a beginner in using Shiro. Any mistakes in my explanations or code are my responsibility. If you do notice that something is wrong, please post a comment.

From the Shiro web site:

Apache Shiro is a powerful and easy-to-use Java security framework that performs authentication, authorization, cryptography, and session management.
(<http://shiro.apache.org> - accessed January 2011)

In March 2009, JSecurity's developers changed the JSecurity name to Ki and then to Shiro for various reasons. So you may hear/read about reference to all three names. According to the Apache Software Foundation: "Ki entered the Apache Incubator in June, 2008" (<http://incubator.apache.org/projects/ki.html>, accessed on April 5, 2009). In September 2010, Shiro became a Apache Top Level Project. The current release (January 2011) of Shiro is 1.1.0.

Shiro provides powerful capabilities with minimal setup--if you're able to use Shiro's configuration defaults in your project. But even if you're not able to use the defaults, you can provide your configuration to Shiro or you can extend one or more of the Shiro classes to customize how it works.

Tutorial Description and Setup

Part 1 of this tutorial starts with a basic web application that has no security. You can download an archived [Eclipse dynamic web project here](#). The project uses Maven to manage dependencies. If you're not familiar with Maven, see the references below. Eclipse has an excellent Maven plugin (see: <http://www.sonatype.com/books/m2eclipse-book/reference/index.html>). You should be able to import the archived project directly into Eclipse. The project's name is nosecurity. If you don't use Eclipse, just unzip the downloaded file (nosecurity_mvn.zip) and copy the source code to your own Java IDE.

You must also download the [separate Apache Derby](#) database as I'm going to show you how to use Shiro when your web application is storing usernames and passwords in a database. Note that Shiro comes with support for many other kinds of storage mechanisms. Unzip this download into c:/derby on your computer.

To use the Derby database with Tomcat and connection pooling you'll need to download the Derby lib files at <http://db.apache.org/derby/releases/release-10.4.2.0.html>. After unzipping the download copy the derby.jar to your Tomcat's /lib folder. The nosecurity web application uses connection pooling to manage connections to the Derby database. For Tomcat to find the correct driver class to connect to Derby the derby jar file must be in Tomcat's lib directory.

The no security web application includes an index.jsp page and a separate folder (named secure) that has two web pages (users.jsp and index.jsp). The pages under the secure folder are supposed to be available only to logged in (authenticated) users. Under the src/main/java folder are packages with the data access, model, and Servlet classes.

Testing The Database Connection

After you've got everything setup you can test the application's connection to Derby by running the TestConnectionFactory class that is in the dao package. This class is just a standard Java class with a main method. In the main method, it uses the ConnectionFactory class to get a connection to the Derby database, then displays some information about Derby, and shows the results of querying the users table. The users table is where the project is storing the usernames and passwords for the tutorial. Note if you did not unzip the Derby database (securityDB) into c:/derby you will need to change the path to securityDB in class ConnectionFactory, method getConnection, in the context.xml file under the META-INF folder, and in jetty-env.xml in WEB-INF folder.

Running The Tutorial

After confirming that the project can connect to the Derby database, you can build and deploy the web application to a Java Servlet container and web server such as Tomcat or Jetty. The tutorial was tested on Tomcat version 6 and Jetty. Remember if you're using Eclipse, you'll need the Maven Eclipse plugin so that all the dependent jars will be provided to the project. See the reference below for how to get the Maven Eclipse plugin.

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 nosecurity_mvn.zip download. Make sure you're in the nosecurity directory. Then do the following (in this example I unzipped nosecurity_mvn.zip to c:\security_examples):

```
c:\security_examples\nosecurity\mvn clean
```

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

[Why I Still Love ColdFusion](#)
[Java Shots 9 - Java EE7 and HTML 5, SAML Explained, and Git Internals](#)
[New Product Marketing Manager for ColdFusion](#)
[Getting HTML5 Ready - ContentEditable](#)
[NCDevCon Session Spotlight: Designing the mobile experience](#)

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](#)

c:\jsecurity_examples\nosecurity\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/nosecurity/>. 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 no security you should be able to click on all the links and all the web pages should display, including the secure/users.jsp which displays the records from the users table that is the Derby database.

What's Next?

I realize this seems like a lot of setup just to get a simple web application to run. But I want to ensure that you can run this web application that uses a Derby database to store user information. In future tutorials, I'll be adding to this web application features from Shiro to enable security. So it's important that you get the basic web application working correctly in your development environment so that if you run into problems in the future tutorials you'll be able to more easily determine the cause.

In [part 2 of this tutorial](#), I'll add basic security using Shiro to the web application to prevent users who have not logged in from viewing the pages in the secure folder.

References

1. No Security Example Application, http://www.brucephillips.name/jsecurity_examples/nosecurity_mvn.zip
2. Apache Shiro <http://shiro.apache.org/>
3. Apache Shiro API, <http://shiro.apache.org/static/current/apidocs/>
4. Apache Shiro Mailing Lists, <http://shiro.apache.org/mailling-lists.html>
5. Apache Derby, <http://db.apache.org/derby/>
6. Apache Tomcat, <http://tomcat.apache.org/>
7. Jetty, <http://jetty.mortbay.org/jetty5/index.html>
8. Maven: The Definitive Guide, <http://www.sonatype.com/books/maven-book/reference/public-book.html>
9. Developing with Eclipse and Maven, <http://www.sonatype.com/books/m2eclipse-book/reference/index.html>

 [Comments \(4\)](#) |  [Print](#) | [Send](#) |  [del.icio.us](#) |  [Digg It!](#) |  [Linking Blogs](#) | 26165 Views

Related Blog Entries

- [Tutorials On Shiro-A Java Security Framework-Updated](#) (January 9, 2011)
- [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.)

[\[Add Comment\]](#) [\[Subscribe to Comments\]](#)

Any suggestions on running the example on glassfish v.3? i was unable to view the user list page (it throws null pointer exception), when the app was run. the home page is displayed correctly.

Posted By Umair Ishaq | 3/14/10 4:33 PM

Hi Bruce,

I am working on implementing Shiro and was looking at your tutorial. I may be missing something but when I run the TestConnectionFactory I get an exception stating that the USERS schema does not exist.

I connected to the DB supplied and noticed that this is correct - the USERS schema doesn't exist. I did unzip securityDB to c:\derby as instructed.

Any possibility the uploaded data did not contain this table?

Posted By Keith | 2/20/12 7:11 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](#)

Keith:

This line in class ConnectionFactory:

```
String sourceURL = "jdbc:derby:/derby/" + dbName + ";create=true";
```

needs to be changed to

```
String sourceURL = "jdbc:derby:c:/derby/" + dbName + ";create=true";
```

if you are running this on a Windows computer and have copied the no security database to c:/derby.

When I first built this example app I was on Windows, but when I updated two years later I was on a Mac.

Let me know if that fixes your issue.

Posted By Bruce | 2/20/12 8:31 PM

Thanks Bruce - I figured it out 5 minutes after I posted - I missed that there is a separate download for the derby DB data!

It's working fine now and I'm moving on to part two.

Thanks Again.

Keith

Posted By Keith | 2/20/12 8:48 PM

[\[Add Comment\]](#)

BlogCFC was created by [Raymond Camden](#). This blog is running version 5.9.1.002. [Contact Blog Owner](#)