Apache Roller, Acegi Security and Single Sign-on

Matt Raible
matt@raibledesigns.com
http://raibledesigns.com
Today’s Agenda

- Introductions
- Integrating Roller with LDAP and CAS on Tomcat
- Introduction to Acegi Security
- Introduction to Apache Roller
- Conclusions
- Q and A
Introductions

- Do you blog?
- Do you use Roller or JRoller?
- What do you want to get from this session?
- Experience with Acegi Security, LDAP, or SSO Solutions?
- Preferred Server: Tomcat, Geronimo, JBoss or GlassFish?
Who is Matt Raible?

- One of the first Roller users and Committers – started in August 2002
- Java Blogger since 2002
- Power user of Java Web Frameworks
- Author of *Spring Live* and *Pro JSP 2.0*
- Founder of AppFuse (http://appfuse.org)
- Member of Java EE 5, JSF 1.2 and Bean Validation Expert Groups
Integrating Roller with CAS on Tomcat

http://cwiki.apache.org/confluence/display/ROLLER/Roller+4.0+with+LDAP+and+CAS
Installing Roller on Apache Geronimo

http://cwiki.apache.org/confluence/display/ROLLER/Roller+4.0+on+Geronimo
Acegi Security

- A powerful, flexible security solution for enterprise software with a particular emphasis on applications that use Spring.
J2EE’s CMA

- Container Managed Authentication (CMA) built into the Servlet API
- Configure security-constraints in web.xml
- Configure Authentication Realm in your application server
<security-constraint>
    <web-resource-collection>
        <web-resource-name>Secure Area</web-resource-name>
        <url-pattern>*.html</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>*</role-name>
    </auth-constraint>
</security-constraint>

<login-config>
    <auth-method>FORM</auth-method>
    <form-login-config>
        <form-login-page>/login.jsp</form-login-page>
        <form-error-page>/loginError.jsp</form-error-page>
    </form-login-config>
</login-config>
Form Authentication

- /login.jsp

```html
<form id="loginForm" method="post" action="j_security_check">
  <p>
    <label for="j_username">Username:</label>
    <input type="text" name="j_username" id="j_username"/>
  </p>
  <label for="j_password">Password:</label>
  <input type="password" name="j_password" id="j_password"/>
  <button type="submit">Login</button>
</form>
```

- /loginError.jsp

```html
<p>
  Login failed - please <a href="index.jsp">try again</a>.
</p>
```
Tomcat Realms

- MemoryRealm, JDBCRealm, DataSourceRealm, JAASRealm, JNDIRealm
- JDBCRealm Example:

```xml
<Context path="" docBase="roller" debug="99"
  reloadable="true" antiJARLocking="true" antiResourceLocking="true">
  <Realm className="org.apache.catalina.realm.JDBCRealm" debug="99"
    driverName="com.mysql.jdbc.Driver"
    connectionURL="jdbc:mysql://localhost/roller?autoReconnect=true"
    connectionName="root" connectionPassword=""
    userTable="users" userNameCol="username" userCredCol="password"
    userRoleTable="user_roles" roleNameCol="rolename"/>
</Context>
```
Problems with CMA

- Not as portable as you’d think
- Every server has proprietary configuration
- Form-based authentication problems:
  - Often can’t control SQL for user/role query
  - No way to filter on /j_security_check to trap when users first login
  - Implementation different on various servers
- **However** – Vendors *love* it!
Solution: Acegi Security

- Everything can be configured in your application
- Secure URLs by role with regular expressions
- URL patterns can be regular expressions or Ant-style patterns (i.e. /**/admin*.html)
- Authentication methods supported: Basic, Digest, Form, Yale Central Authentication Service (CAS)
- Authentication Providers: JDBC, XML, LDAP, CAS
Configuration: web.xml

```xml
<filter>
    <filter-name>securityFilter</filter-name>
    <filter-class>org.acegisecurity.util.FilterToBeanProxy</filter-class>
    <init-param>
        <param-name>targetClass</param-name>
        <param-value>org.acegisecurity.util.FilterChainProxy</param-value>
    </init-param>
</filter>

<filter-mapping>
    <filter-name>securityFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```
security.xml

- The **filterChainProxy** bean contains the filter list that will process the authentication process. These filters each perform specific duties:
  - **httpSessionContextIntegrationFilter**: This filter is responsible for communicating with the user's session to store the user's authentication in the SecurityContextHolder.
  - **basicProcessingFilter**: This filter processes an HTTP request's BASIC authorization headers, placing the result into the SecurityContextHolder.
  - **exceptionTranslationFilter**: Defines exceptions and entry point (URL and SSL)
<bean id="filterChainProxy" class="org.acegisecurity.util.FilterChainProxy">
  <property name="filterInvocationDefinitionSource">
    <value>
      CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
      PATTERN_TYPE_APACHE_ANT
      /**=httpSessionContextIntegrationFilter,authenticationProcessingFilter,
       remoteUserFilter,rememberMeProcessingFilter,anonymousProcessingFilter,
       exceptionTranslationFilter,filterInvocationInterceptor
    </value>
  </property>
</bean>
Form Authentication

- Changing from Basic to Form-based authentication is just XML configuration
- Login and Error pages can be same as CMA pages
- No code needed to support Remember Me and Password Encryption – just XML
- Can configure SSL “channels” based on URL-pattern
Authentication Providers

- **Custom**: write your own
- **In-Memory**: credentials in XML file
- **JAAS**: provided by LoginModule
- **JDBC**: credentials in database
- **LDAP**: credentials in database
- **OpenID**: experimental support, see SEC-432
- **Windows NT**: experimental support, see SEC-8
- **SSO**: Yale’s CAS, SiteMinder
<bean id="daoAuthenticationProvider"
class="org.acegisecurity.providers.dao.DaoAuthenticationProvider">
  <property name="userDetailsService" ref="inMemoryDaoImpl"/>
</bean>

... 

<bean id="inMemoryDaoImpl"
class="org.acegisecurity.providers.dao.memory.InMemoryDaoImpl">
  <property name="userMap">
    <value>
      tomcat=tomcat,ROLE_USER  
springlive=springlive,ROLE_USER
    </value>
  </property>
</bean>
Password Encryption

```xml
<bean id="inMemoryDaoImpl"
    class="org.acegisecurity.providers.dao.memory.InMemoryDaoImpl">
    <property name="userMap">
        <value>
            tomcat=536c0b339345616c1b33caf454454d8b8a190d6c,ROLE_USER
            springlive=2a9152c0f1d25b5bbaa3e5fbc7acdc6905c9f251,ROLE_USER
        </value>
    </property>
</bean>

<bean id="daoAuthenticationProvider"
    class="org.acegisecurity.providers.dao.DaoAuthenticationProvider">
    <property name="userDetailsService" ref="inMemoryDaoImpl"/>
    <property name="passwordEncoder" ref="passwordEncoder"/>
</bean>

<bean id="passwordEncoder"
    class="org.acegisecurity.providers.encoding.ShaPasswordEncoder"/>
```
To use JDBC, just define a bean with a dataSource dependency:

```xml
<bean id="jdbcDaoImpl" class="org.acegisecurity.providers.dao.jdbc.JdbcDaoImpl">
   <property name="dataSource" ref="dataSource"/>
</bean>
```

Default SQL for select users and roles:

```
"SELECT username,password,enabled FROM users WHERE username = ?";
"SELECT username,authority FROM authorities WHERE username = ?";
```
Cool Features

- Event Listeners
- SecurityContextAwareRequestFilter
- Secure Methods by Role
- Remember Me
- SSL Switching
Other Features

- **Anonymous Filter**: Creates Authentication object with anonymous user information
- **Access Control Lists (ACLs)**: Control permissions per object
- **AfterMethodInvocation Interceptor**: Removes objects from collections when user can’t read them
- **Auditing and Event Logging**:

  ![Cygwin](image)
### J2EE vs. Acegi Security

<table>
<thead>
<tr>
<th>Security Framework</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>J2EE Security</td>
<td>It is easy to set up from an application perspective.</td>
<td>It can be difficult to port from one application server to the other.</td>
</tr>
<tr>
<td></td>
<td>User Realm configuration is in the hands of the deployer.</td>
<td>Even though the application-developer configuration is standardized, the realm configuration for servers is not.</td>
</tr>
<tr>
<td></td>
<td>Because it's a standard, many sources of documentation are available.</td>
<td>Service layer methods can only be secured if using EJ Bs.</td>
</tr>
</tbody>
</table>

---

Matt Raible — Apache Roller, Acegi Security and Single Sign-on  
Colorado Software Summit: October 21 – 26, 2007  
© Copyright 2007, Raible Designs
# J2EE vs. Acegi Security Framework

<table>
<thead>
<tr>
<th>Security Framework</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
</table>
| Acegi Security     | Security configuration is completely self-contained in the application – you don't have to worry about application server portability.  
It solves many of the shortcomings of J2EE security and allows all the same things, with the option to customize.  
It supports single sign-on with CAS.  
It’s evolving and improving very rapidly.  
It allows you to secure methods of any Spring-managed bean and filtering objects based on their ACLs. | It requires a lot of XML to configure.  
The learning curve can be a little steep and seem overwhelming at first.  
Realm information is packaged with the application, making it tough for a deployer to change.                                                                                             |
Apache Roller

- What is Apache Roller?
- Installing Roller
- Roller Architecture
  - Blog Customization
  - Server Customization
- Other Features: Clients and Planet
What is Apache Roller?

- Apache Roller is a full-featured, multi-user and group-blog server suitable for blog sites large and small

Cool Features:
- Multi-user and Group blogging
- Comment moderation and spam prevention
- Complete control over UI with templates
- Built-in Search
- RSS 2.0 and Atom 1.0 Support
- Pluggable Cache and Rendering System
Started by Dave Johnson in 2000 as “Homeport”

http://rollerweblogger.org/roller/date/20021231
History of Roller

- In 2002, Dave ditched EJBs and HAHTsite IDE for open source tools (Ant, Castor, Struts, and Velocity)
- April 5, 2002: Roller 0.9.0 Released
- April 17, 2002: O’Reilly Article “Building an Open Source J2EE Weblogger” Published
- July 31 – August 7, 2002: The world starts blogging with Roller

http://rollerweblogger.org/roller/date/20021231
Roller since 2002

- Roller now powers jroller.com, blogs.sun.com, IBM developerWorks blogs and many others
- Dave hired by Sun to work on Roller full-time in September 2004
- Roller began incubation at Apache in June 2005
- April 23, 2007: Graduated and released 3.1
But what *is* Roller?

- Roller is a **blogging** engine
- **Blogs** make web publishing easy
  - Everyone can do it
  - No need for IT or Webmasters
  - Tools are in the users hands
- **Feeds** make reading blogs easy
  - Feeds are XML-based: RSS or Atom
  - You *subscribe* to a feed in a **Feed Reader**
  - **Feed Readers** are like inboxes for the web
Posting a Weblog Entry

Editing weblog entry in weblog rd

Title: What a Trip - Amsterdam was a blast!
Status: Published (Last updated 5/8/07 6:25 PM)
Permalink: http://raibledesigns.com/rd/entry/what_a_trip_amsterdam_was
Category: /General
Tags: amsterdam apachecon europe germany

Content:

Last week's trip to Europe for ApacheCon EU was nothing short of spectacular. Amsterdam was an incredible city that amazed me with its awesome biking system (and usage!), incredible atmosphere and a nice sense of relaxation. I think the picture below sums up our trip nicely. The weather couldn't have been better - sunny and warm with a cool breeze.
Viewing a Weblog Entry

ApacheCon EU: Comparo... | Main | Off to Florida

What a Trip - Amsterdam was a blast!

Last week’s trip to Europe for ApacheCon EU was nothing short of spectacular. Amsterdam was an incredible city that amazed me with its awesome biking system (and usage), incredible atmosphere and a nice sense of relaxation. I think the picture below sums up our trip nicely. The weather couldn’t have been better - sunny and warm with a cool breeze.

ApacheCon was likely an excellent conference, but I’m proud to say I didn’t attend a single session. I did manage to make it to Sun’s party on Thursday night (thanks Dave!), but that was about it. At the party, the Wicket guys had a BOF, which my dad and I sat in on. Thanks to Martin, Elco and others for the good conversation, even if we did have to shout.
Reading a Weblog Entry

Last week's trip to Europe for ApacheCon EU was nothing short of spectacular. Amsterdam was an incredible city that amazed me with its awesome biking system (and usage!), incredible atmosphere and a nice sense of relaxation. I think the picture below sums up our trip nicely. The weather couldn't have been better - sunny and warm with a cool breeze.

http://raibledesigns.com/rd/entry/what_a_trip_amsterdam_was
Why choose Roller?

- Proven, full-featured blogging solution for big sites
  - Used by Sun, IBM, Yale University, Covalent and ESRI
- Open Source and Apache Licensed
- Active and growing community at Apache
- Standard Java Web Application Architecture
Why choose Roller?

- It works great if you know what you’re doing
- Nice looking example sites:
  - http://blogs.usd.edu/jrbethke
  - http://rollerweblogger.org/roller
  - http://raibledesigns.com
  - http://ryandelaplante.com
- **Awesome** themes at http://rollerthemes.com!
Installing Roller

- Download Roller 3.1 from http://cwiki.apache.org/confluence/display/ROLLER/Roller+Downloads
- Download Hibernate and other JARs from https://roller.dev.java.net/servlets/ProjectDocumentList?folderID=6962
- Copy JARs from java.net download into apache-roller-3.1/webapp/roller/WEB-INF/lib
Installing Roller: Java & MySQL

- Download and Install Java 5 from:
  - http://java.sun.com/javase/downloads

- Download and install MySQL 5 from:
  - http://dev.mysql.com/downloads

- Create database with files in WEB-INF/dbscripts:
  - mysqladmin -u root -p create roller
  - cd webapp/roller/WEB-INF/dbscripts/mysql
  - mysql -u root -p roller < createdb.sql

**NOTE:** Use /WEB-INF/classes/dbscripts in Roller 4.0.
Installing Roller: Tomcat

- Download and install Tomcat 6 from
  - http://tomcat.apache.org/download-60.cgi
- Copy `apache-roller-3.1/webapp/roller` to `$CATALINA_HOME/webapps/ROOT`
- Copy `activation.jar`, `mail.jar` and `mysql-connector-java-5.0.3-bin.jar` to `$CATALINA_HOME/lib` (common/lib for Tomcat 5.x)

»Continued on next page
Installing Roller: Tomcat

(Continued)

- Create ROOT/META-INF/context.xml with the following contents:

  ```xml
  <Context path="" reloadable="false" antiJARLocking="true"
    antiResourceLocking="false" allowLinking="true">
  <Resource name="jdbc/rollerdb" auth="Container"
    type="javax.sql.DataSource"
    maxActive="20" maxIdle="10" maxWait="100"
    driverClassName="com.mysql.jdbc.Driver"
    username="root" password=""
    url="jdbc:mysql://localhost/roller"/>
  <Resource name="mail/Session" auth="Container"
    type="javax.mail.Session"
    mail.smtp.host="localhost"/>
  </Context>
  ```
Roller Install: Startup

Start Tomcat and create your weblog at http://localhost:8080

Welcome to Roller!
Follow these steps to finalize your Roller installation:

• Create a user
Before you can start to use Roller, you need to create a user so you can login, manage Roller and start blogging. Note that the first user you create will be given the Global Admin rights necessary to manage Roller, manage Roller users and designate other Global Admins.
Create your first user via the New User Registration Page.

• Create a weblog
Before you can start blogging, you need to create at least one weblog. Just so you know, you can create as many as you want. Each Roller user can have multiple weblogs and each Roller weblog can have multiple authors.
Create your first weblog via the New Weblog Creation Page.

• Designate a frontpage weblog
You must specify a weblog to serve as the frontpage of your Roller site. You can do this by going to the Server Admin -> Configuration page. In the field labelled "Handle of the weblog to server as the frontpage" enter the handle of the weblog you want to front your site. Once you're done you won't see this irritating page any longer.
Designate a frontpage weblog on the Server Admin Page

Powered by Apache Roller 3.1 | Report an Issue | User Guide | Mailing Lists
Create a User

New User Registration

Use this form to register a new Roller user, please create only one user account per person.

Full Name: Matt Raible

Username: mraible

Password: ********

Password (Confirm): ********

Email: matt@raibledesigns.com

Locale: English (United States)

Timezone: MDT – America/Denver

Your full name (with no HTML).

A short one word username for your user account. Please limit it to simple ASCII alphanumeric characters (a-z, A-Z and 0-9) and do not use HTML.

Your password.

Confirm your password.

Please enter valid email address, the site administrator may disable your account if he/she cannot reach you via email.

Your preferred locale.

Your preferred timezone.

Register User Cancel
Create a Weblog

Create Weblog
Use this form to create a new weblog for you or for you and a group of your friends.

Name: Matt's Weblog

Description:

Handle: blog
URL: http://localhost:8080/blog

Email Address: matt@raibledesigns.com

Locale: English (United States)

Timezone: MDT – America/Denver

Theme: brushedmetal

The name is the title of your weblog, it will be displayed at the top of your weblog page, and in the title field of your weblog's newsfeed. This field should not include HTML.

The description of your weblog may be displayed at the top of your weblog (depending on the theme you choose) and it will be used in description or subtitle field of your newsfeed. This field should not include HTML.

The handle is a short one word name for your weblog. It will be used in your URL, so please limit it to simple ASCII alphanumeric characters (a-z, A-Z and 0-9) and do not use HTML.

Your email address or the email address of the person responsible for the weblog that you are creating.

The default locale to be used for the display of this weblog when the browser does not specify a locale.

The timezone to be used for entry and display of this weblog.

The theme to be used to display this weblog. Later, you can easily switch to another theme or edit the themes templates to customize it to your liking.
The obligatory first post

Matt's Weblog

Tuesday Jun 19, 2007

First Post
This is my first post using Apache Roller. It was pretty easy to setup after watching Covalent's Webinar.

Posted at 07:16AM Jun 19, 2007 by Matt Raible in General | Edit | Comments[0]

Comments:
Post a Comment:

Name:
E-Mail:
URL:

[ ] Remember Information?

Your Comment:
Roller Architecture: Enterprise

- Web UI via Java Servlets and JSP
  - Front controller, Web MVC and Open Session In View patterns
- Persistence via JDBC
  - Factory, Façade and Data Mapper patterns
Roller Architecture: Geek Speak

- **Roller Web: Web and UI Layer**
  - Editor UI *via Struts* and *JSP*, blog and feed rendering *via Velocity*
  - Feed parsing *via ROME*, Blogger API *via Apache XML-RPC*

- **Roller Beans: Business and Persistence Layer**
  - *Hibernate/JPA* for DBMS, *Lucene* for search
What’s New in Roller 4.0

- Easier Theme Customization
- Easy Installation
- Java 5, Open JPA and Struts 2
- [http://cwiki.apache.org/confluence/display/ROLLER/What%27s+New+in+Roller+4.0](http://cwiki.apache.org/confluence/display/ROLLER/What%27s+New+in+Roller+4.0)
Conclusion

Apache Roller is a full-featured blogging system
Installing Roller is Easy
Integrating with SSO is Painless
Blogging is fun – and great for your career!
Additional Resources

- Acegi Security Forums:

- Yale’s CAS:

- Roller User Mailing List:
  - [user@roller.apache.org](mailto:user@roller.apache.org)
Questions?

- matt@raibledesigns.com
- http://raibledesigns.com

Download presentation from:
http://raibledesigns.com/rd/page/publications