

---

# This is my Blog.

---

AUG. 23, 2011

**23 August 2011**

As it stands, Lift only has its proto\* traits for user management, and that system has its limitations and you will ultimately end up replacing it in any non-trivial application as your needs change and you need to grow. Whilst this is what those traits are designed for (quick start, short haul), you typically end up rolling your own system for users etc when using Lift, and this can often be somewhat cumbersome or not particularly easy to do well. As this whole user management piece is so often requested, I figured that i'd write a plugin library for Lift.

[Apache Shiro](#) is a Java security framework (formally known as JSecurity) and it comes with a fairly abstract set of classes for building systems that have the familiar users, roles and permissions setup. Pretty much most applications these days have some notion of users, customers or some other subject that you care about and might want to conduct access control around. This is exactly what Shiro is designed for, and it ships with out of the box inter-operation with ActiveDirectory and other such repositories commonly found in the enterprise space for managing user data.

Part of the reason that other security frameworks never really took to Lift (or vice-versa) is that Lift has its own mech for managing resource ACLs and it never made sense to separate that into a different servlet filter and somehow munge that together: its not 1990. Fortunately Shiro was fairly easy to integrate with Lift in such a way that it allows you to simply augment your existing SiteMap setup, template markup and even dispatch resources. Currently this integration project is in early stages, and you can find the source code here: [github.com/timperrett/lift-shiro](https://github.com/timperrett/lift-shiro)

Here's a quick walkthrough of the various ways you can use the integration within your project. **Example**  
Firstly, lets assume you only want to display a section of content to authenticated users:

```
<lift:has_role name="admin">  
  <p>This content is only available for admins</p>
```

```
</lift:has_role>
```

There are a range of authentication snippets that allow you to define who sees what within your templates, checkout the documentation for more on that. Nextup, what if you want to block access to an page entirely if the user is not authenticated? Just add the following to your SiteMap:

```
...  
Menu("Home") / "index" >> RequireAuthentication  
...
```

By default RequireAuthentication will redirect unauthenticated users back to the URL defined in Shiro.loginURL. Likewise, you can specify whole resources to require a particular role or permission:

```
...  
Menu("Role Test") / "restricted" >> RequireAuthentication >> HasRole("admin")  
...
```

Clearly the SiteMap functionality is implemented as LocParam, so you can implement them within your own Loc types, or simply use them declaratively within the regular SiteMap usage.

This whole integration project wraps the Shiro types, so you only need to configure shiro.ini in the root of your classpath and enter the appropriate realm information as per the regular Shiro documentation, then away you go: password files... active directory... whatever you want.

As above, this project is still early stage, but it does indeed work. I'm currently looking for feedback, so if you have some thoughts or things that would be cool to see, then please checkout the project on Github and fork away.

[blog comments powered by](#)

---

**Original URL:**

<http://timperrett.com/2011/08/23/using-apache-shiro-with-lift/>

