CHAPTER 6

Security

Security is a natural part of nearly every online application. Any application that requires some kind of personal service needs to know the identity of the user. Preferences and data can't be associated with you unless the application knows who "you" are. This is the goal of authentication: securely identifying the users of a system.

Applications are typically put online so that many people can use them, not just one. It is very rare, however, that you want everyone using your application to have the same access rights. Some people have more or less access than others, depending on the roles and permissions that they have been given. Authorization provides the tools for assigning and verifying the access rights of users.

In this chapter, we are going to explore Seam's support for authenticating and authorizing users. First, I'll give you a high-level overview of the capabilities provided by Seam. Then I'll describe some security-related extensions required in the next version of the Gadget Catalog, and we'll implement them in the rest of the chapter using Seam's security services.

Seam Security Support

Seam directly supports the integration of security measures into your applications. Specifically, it facilitates adding authentication and authorization features to your Seam applications. As any security expert will tell you, there are many other aspects to be considered when it comes to security (identity management, encryption, intrusion detection, etc.). But Seam focuses on the most common application needs, and the rest can be integrated through other means.

Authentication

Seam helps you inject login functionality into your application, in areas where you need to identify users and/or check to see whether they have rights to access particular pages, functions, or data. Seam gives you the ability to require authentication at various levels (page, JSF control, component, action method) using configuration file entries and/or

code annotations. The login process can be implemented using standard Seam components and JSF forms, in conjunction with built-in components provided by Seam. Seam also provides built-in components that support the handling of the user's identity for easy access from within the Seam contextual component model.

Authorization

You often need to authenticate users because you need to check their access rights. Again, Seam helps you to authorize users at various levels in the application—you can specify access limits for groups of pages, single pages, specific JSF controls, entire Seam components, or specific action methods on components. These access rights are specified using roles and/or permissions. Roles in Seam are simple named role assignments, like "admin" or "sales-rep", while permissions are named actions that can be performed in general or on specific entities. Like the authentication services within Seam, you can specify access rules for entities using configuration file entries and/or code annotations. It's also possible to do more advanced authorization management using JBoss Rules rulesets.

Seam Security vs. Java EE Security

Readers who are familiar with Java EE security features for web and EJB components might be wondering how Seam's security features relate to them. The short answer is that they don't. Seam's security services are an independent system that is not integrated with Java EE's declarative or programmatic security features. User identities and roles in Seam are sourced from Seam components and services, while Java EE uses the concept of realms configured in the application server. Seam's authentication is configured through `components.xml` and the identity is stored in a Seam component, while Java EE uses `login-config` elements in `web.xml` and stores the identity in the user's runtime web/EJB context. Seam's programmatic role checking is done through EL expressions and/or Seam component methods, while Java EE provides the `isUserInRole()` methods in web and EJB components. And so on.

Seam does offer some crude integration with the Java Authentication and Authorization Services (JAAS), which are the backing services behind the Java EE security services. In the current released version of Seam (version 1.2.1), however, this integration is definitely not seamless (pardon the pun) and involves some fairly complicated configuration gymnastics. These configuration details aren't provided here, because they seem to be very preliminary in nature, and not broadly useful in their current form. The integration is also limited, since it only provides a common source for identities and roles, but no shared authorization or authentication configuration.

In order to avoid complex configuration details, runtime conflicts, and potential confusion, my recommendation would be to stick to one security model or the other until there is better integration between the two. If you have a specific requirement to use