



@task

External Login

This document describes the External Login functionality of @task v4 and provides information on creating a custom External Login module.

OVERVIEW

The External Login functionality in @task allows customization of the authentication process. This can be used to integrate with an external data source such as NDS or Active Directory.

This is done by providing a hook that allows the creation of a custom authentication interceptor that receives the username and password credentials from the login screen and authenticates against the external source, prior to @task's own authentication.

HOW IT WORKS

An external login class is created and @task is configured to use this custom class as an external authentication. This class implements the `com.attask.biz.externallogin.ExternalLogin` interface. This interface defines a single method:

```
ExternalUser login(java.lang.String username, java.lang.String password)
```

This method receives the username and password from the login page, performs its own authentication (whether against an NDS tree using LDAP, against a relational database using SQL, or just by reading a plain text file, etc.) and returns one of two things:

1. **null** - If the authentication failed
2. **An ExternalUser Object** - If the authentication was succesful. This object is created and populated by the custom external user class.

If a **null** is returned, @task will attempt to authenticate locally from the @task database. If user-supplied credentials match neither source, an invalid username/password exception is thrown. If the @task login is succesful, then the use is logged in as normal. This means that it is still possible to have @task-only users that can login to @task.

If an **ExternalUser** object is returned, @task assumes the authentication was succesful, compares the username to the @task user database and does one of two things:

1. If `login()` returns an **ExternalUser** object with a new Username not found in the @task databsae, a new @task User is created with the @task password "user". All other fields are synchronized according to instructions found in the javadocs for the **ExternalUser** class.
2. If `login()` returns an **ExternalUser** object with a matching Username found in @task, fields are synchro-nized according to instructions found in the javadocs for the **ExternalUser** class. The User's @task pass-word can only be set in the @task interface or through the @task API.

IMPLEMENTATION

You will need to create a custom class that implements the interface `com.attask.biz.externallogin.ExternalLogin`. This interface defines a single method:

```
ExternalUser login(java.lang.String username, java.lang.String password)
```

Details on the implementation of this interface can be found in the javadocs for the `com.attask.biz.externallogin` package. A sample class, `com.attask.biz.externallogin.SampleExternalLoginImpl`, is also provided in that package.

INSTALLATION

This custom class then needs to be placed in the classpath for JBoss. This will typically be:

```
<JBOSS_HOME>\server\default\deploy\redrock-app.jar\redrock.ear\redrock-ejb.jar (on Windows)
```

or

<|BOSS_HOME>/server/default/deploy/redrock-app.jar/redrock.ear/redrock-ejb.jar (on Linux, Unix, or MacOSX)

For example, on a Windows machine, to install a custom class, called NDSLogin, in the com.mybizpath.attask package, You might end up with a path that looks like this:

D:\AtTask\AtTask\Boss\server\default\deploy\redrock-app.jar\redrock.ear\redrock-ejb.jar\com\mybizpath\attask\NDSLogin.class

CONFIGURATION

Once the class is in place, @task needs to be configured to use the external login class. There are two configurations that need to be modified. To configure @task to use the class:

1. Login as an administrator
2. Click on Setup -> System Settings -> Configurations
3. Click Edit Configurations in the left contextual menu.
4. Change the **Use External Login** field to 'Yes'
5. Change the **External Login Classname** to the fully qualified name of your deployed external login class. To continue the example above, this should be set to com.mybizpath.attask.NDSLogin.

CONCLUSION

That's it. Once this setup is complete, you should be able to login using a username and password from the external authentication source.