Web Services e-Book Chapter 3 - Managing Enterprise Sessions

In this chapter, you'll learn how to manage an Enterprise user's session with the Enterprise system using the Session Web Service.

- Overview
- The Session Web Service
- Similarities with the Enterprise SDK
- Summary
- Additional Resources

Overview

One of the key features of the BusinessObjects Enterprise platform is security: both administrators and end-users must be authenticated (and subsequently authorized) by the Enterprise platform before completing tasks against the Enterprise system.

For example, you'll notice that the Central Management Console (CMC) and InfoView applications present the user with a logon form where the user can enter his/her Enterprise, LDAP, Windows AD or Windows NT credentials. Once authenticated, the end-user can then carry out the tasks he/she is authorized to do so within the CMC and InfoView applications.

The WS consuming application is also required to pass the end-user credentials to the Enterprise system using the WS SDK.

NOTE: You can use Windows AD, Windows NT, LDAP or Enterprise authentication to login to the Enterprise system. Windows AD, Windows NT and LDAP authentication must be configured in the Enterprise system before end-users are able to use these third party authentication plugins to log in to the Enterprise system.

The Session Web Service

The Session Web Service is used to manage Enterprise user sessions with the WS SDK; it is used to:

1. Implement the logon and logout workflow

The Session Web Service is used to carry out the log in and logout workflow on an Enterprise system. Here is a Java and .NET code snippet outlined the log in workflow:

Java Consumer API:

```java
// Create the URL for the Session Service and instantiate a new Session Service
String providerURL = "http://localhost:8080/dswsbobje/services";
URL sessConnURL = new URL(providerURL + "/session");
Connection boConnection = new Connection(sessConnURL);
Session boSession = new Session(boConnection);

// Setup the Enterprise Credentials used to log in to the Enterprise System
// cmsname - the name of the Enterprise Central Management System (CMS)
// username - the Enterprise user's user name
// password - the password for the Enterprise user
// authType - the authentication to use; can be one of the following:
// - secEnterprise: Enterprise authentication
// - secLDAP: LDAP authentication
// - secWinAD: Windows AD authentication
EnterpriseCredential boEnterpriseCredential = new EnterpriseCredential();
boEnterpriseCredential.setDomain(cmsname);
boEnterpriseCredential.setLogin(username);
boEnterpriseCredential.setPassword(password);
boEnterpriseCredential.setAuthType(authType);

// Log in to the Enterprise System and retrieve the SessionInfo
SessionInfo boSessionInfo = boSession.login(boEnterpriseCredential);

// Logout
boSession.logout();
```

.NET Consumer API:

```csharp
// Create the URL for the Session Service and instantiate a new Session Service
string sessConnURL = "http://localhost:8080/dswsbobje/services/session";
```
BusinessObjects.DSWS.Connection boConnection = new BusinessObjects.DSWS.Connection(sessConnURL);

Session boSession = new Session(boConnection);

// Setup the Enterprise Credentials used to log in to the Enterprise System
// cmsname - the name of the Enterprise Central Management System (CMS)
// username - the Enterprise user's user name
// password - the password for the Enterprise user
// authType - the authentication to use; can be one of the following:
// secEnterprise: Enterprise authentication
// secLDAP: LDAP authentication
// secWinAD: Windows AD authentication
EnterpriseCredential boEnterpriseCredential = new EnterpriseCredential();
boEnterpriseCredential.Domain = cmsname;
boEnterpriseCredential.Login = username;
boEnterpriseCredential.Password = password;
boEnterpriseCredential.AuthType = authType;

// Log in to the Enterprise System and retrieve the SessionInfo
SessionInfo boSessionInfo = boSession.Login(boEnterpriseCredential);

// Logout
boSession.Logout();

The SessionInfo object returned from the logon method call stores properties such as the session ID, user CUID, a set of user rights, etc. for the currently logged on user.

1. Represent the Enterprise Session

Once a user has been logged-on to the Enterprise system with the Session Web Service, the reference to the Session Web Service itself is the representation of the Enterprise user's session.

The Session.login() method does return a SessionInfo instance, but this object does not represent the Enterprise session.

A reference to the Session Web Service instance itself must be kept in order to maintain the Enterprise session.

1. Retrieve references to other Web Services

In order to use the other Web Services available with the WS SDK such as the BIPlatform Web Service, a valid Session Web Service reference is needed.

The following code snippets outline how you can use the Session Web Service to retrieve the BIPlatform Web Service.

Java Consumer API

// Retrieve the BIPlatform Service
String[] biPlatformURL = boSession.getAssociatedServicesURL("BIPlatform");
BIPlatform boBIPlatform = BIPlatform.getInstance(boSession, biPlatformURL[0]);

.NET Consumer API

// Retrieve the BIPlatform Service
string[] biPlatformURL = boSession.GetAssociatedServicesURL("BIPlatform");
BIPlatform boBIPlatform = BIPlatform.GetInstance(boSession, biPlatformURL[0]);

Similarities with the Enterprise SDK

If you have used the Enterprise SDK before, you'll no doubt be able to draw some comparisons between the Session Web Service and the Enterprise SDK's ISessionMgr and IEnterpriseSession classes. For those that haven't used the Enterprise SDK before, the ISessionMgr and IEnterpriseSession classes are used to manage an Enterprise user's Enterprise Session.

The ISessionMgr is used to log in to the Enterprise system where the login() method call returns a reference to an IEnterpriseSession object. The returned IEnterpriseSession instance represents the Enterprise user's Enterprise Session when using the Enterprise SDK. Furthermore, similar to how a valid Session Web Service reference must exist in order to retrieve other Web Services a valid IEnterpriseSession reference is needed in order to retrieve services such as the InfoStore with the Enterprise SDK.

So, as you can see, regardless of the SDK being used to access the Enterprise system, the same concepts apply:

- An Enterprise user must log in to the Enterprise system
- A valid Enterprise Session object is used to represent the users Enterprise Session
- A valid Enterprise Session must exist in order to retrieve and invoke the other services available
Summary

This chapter provided you with an overview of how Enterprise user sessions are managed with the WS SDK, while also providing a brief comparison to the Enterprise SDK.

In the next chapter, Completing Administrative Tasks - The BIPlatform Web Service, we'll review how to accomplish administrative tasks such as managing servers with the WS SDK.

Additional Resources

- Comparing the Enterprise and Web Service SDK Technical Paper
- Path Query Language
- Developer Library
- Supported Platforms