

サーブレットの使い方

この機能は、Release10.5以降で利用可能です。

NXJアプリケーションは、NXJ以外で開発されたサーブレットを利用することができます。

例えば、ユーザを確認するためのメソッドを呼び出す以下の **Login.java** サーブレットを作成したと仮定します。サーブレットは、“Welcome”と名づけられたエントリフォームから呼び出されます。

Login.java サーブレットは以下を含みます。

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Login extends HttpServlet{
public void doPost( HttpServletRequest req,
                  HttpServletResponse res)
throws ServletException, IOException{

String userId = req.getParameter("userId");
String password = req.getParameter("password");
/*****
** call a method to validate the password which will
** return the User Name for authorized users and a null
** string for unauthorized users.
*****/

String uName = validateUser(userId, password);
// if uName is null .. user is not authorized.
if (uName == null)
{
PrintWriter ot = res.getWriter();
ot.println(" Please verify the Userid and password");
ot.close();
```

```
    }

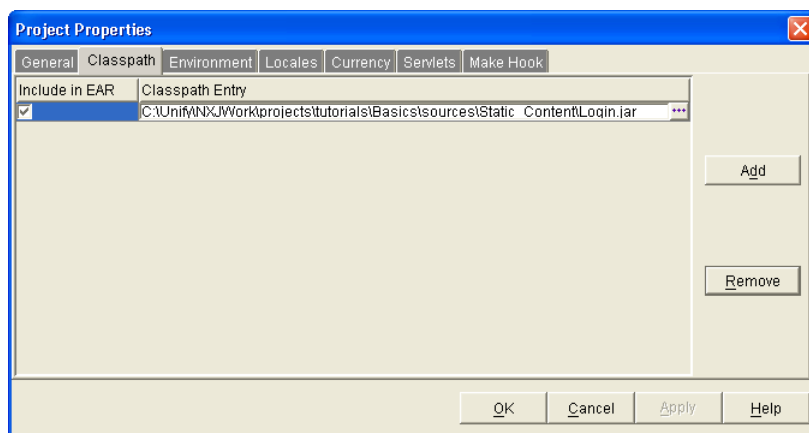
else
{
    // So the user is valid let's create a session
    // for this user.
    HttpSession userSession = req.getSession(true);
    // put the user name session variable.

    userSession.setAttribute("userName", uName);
    // now we need to transfer the control to welcome.jsp
    RequestDispatcher rd =
    getServletContext().getRequestDispatcher("/welcome.jsp");

    if (rd != null)
        {
            rd.forward(req,res);
        }
}
} // end of doPost

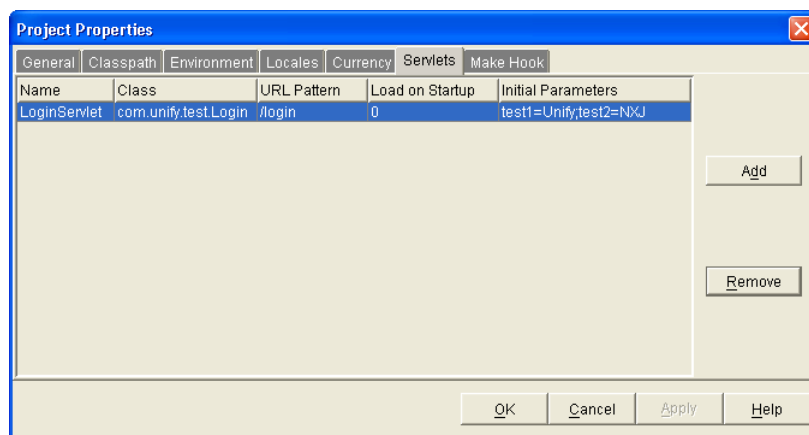
String validateUser(String userName, String password)
{
    // This is a dummy method so return Unify for all accounts.
    return "Unify";
}
} // end of servlet class
```

サーブレットを作成した後、JAR ファイルにサーブレットを追加して、次に **プロジェクト > プロパティ > クラスパス** パネルを使って、プロジェクトのクラスパスに JAR ファイルを追加します。“EAR に含める” チェックボックスをセットしてください。



また、プロジェクトの Sources/Classes フォルダにサーブレットクラスを追加することができます。

次に、**プロジェクト > プロパティ > Servlets** パネルを使って、プロジェクトに **LoginServlet** プロパティを追加します。以下の例を参照してください。



このパネルについての詳細は、『Unify NXJ 開発者ガイド』の第 3 章の「プロジェクトプロパティ」の「Servlets」を参照してください。

これで、サーブレットは実行時に NXJ アプリケーションを利用することができます。