

Döcu Content

Create/Get Session

JavaBean

FacesContext sessionScope

Introduction:

Login to attempt retrieving userID record from NSF back-end, Create and Get sessionScope variable and make available userID data throughout your App to control user access to specific items, or documents from NSF back-end; item that can be used to also hide/show components on Xpages forms.

Disclaimer:

Information contained in the following is presented as is. This tutorial assumes you have basic Lotus Notes Configuration and programming knowledge.

Create/Get Session from Bean

Copy and Paste below JavaBean and XSP code to create inputText components straight away. You will be creating a skeleton Xpages file to run code and create default inputText component items, load additional items via button... You want to make sure your facesConfig file references JavaBean in question; areas of interest highlighted for convenience.

Here are the Steps, see screenshots

1. Grab JavaBean code below and include it into your App, via a preferred package
2. Create a new Xpages files, replace your code with ours
3. Save and run Xpages Login code in your Browser
4. Add incorrect userID data to Login inputText (userName)
5. Similarly, insert correct userID data saved in your view column
6. Click button, load subsequent page to view same data in xpwelcome.xsp

Copy and Paste JavaBean

TestLogJavaBean.java;

```
package com.dokoll.solutions.inc.Utils;
```

```
/**
```

```
* Copyright 2014 Dököll Solutions, Inc.  
Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at
```

```
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software
```

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

```
*  
* @AppName: DöcuContent.nsf  
* Program: TestLoginJavaBean.java  
* Purpose: Maintain variable in session, based on NSF record returned  
*         load custom error Message(s)  
* Created: 2014.02.08.4.02.PM  
*  
* TODO: Add Variable to subsequent pages  
*/
```

```
import javax.faces.context.FacesContext;  
import lotus.domino.Database;  
import lotus.domino.NotesException;  
import lotus.domino.Session;  
import lotus.domino.View;  
import lotus.domino.ViewEntry;  
import lotus.domino.ViewEntryCollection;  
  
import com.ibm.xsp.model.domino.DominoUtils;  
  
/**  
 * @author Dököll Solutions, Inc.  
 * @version 2014.02.08.4.02.PM  
 */  
public class TestLoginJavaBean {  
    //declare search/err Message variables  
    private String UserName;  
    private String MyEntries;  
    private String message;  
    //...  
    static String URLACCESS_VIEWNAME = "URLAccessView";  
  
    /**  
     * @return the userName  
     */  
    public String getUserName() {  
        return UserName;  
    }  
  
    /**  
     * @param userName  
     *         the userName to set  
     */  
    public void setUserName(String userName) {  
        UserName = userName;  
    }  
  
    /**  
     * @param myEntries  
     *         the myEntries to set  
     */  
    public void setMyEntries(String myEntries) {  
        MyEntries = myEntries;  
    }  
  
    /**  
     * @return the message  
     */  
    public String getMessage() {
```

```

        return message;
    }

    /**
     * @param message
     *         the message to set
     */
    public void setMessage(String message) {
        this.message = message;
    }

    /**
     * @return the myEntries
     */
    public String getMyEntries() {

        return MyEntries;
    }

    public TestLoginJavaBean() {
        // ...
    }

    //button code
    @SuppressWarnings("unchecked")
    public String doLoginUser() {

        // let's add a try catch here, to grab errors near the end
        try {

            // BEGIN DEBUG
            System.out
                .println("BEGIN DEBUG | TestLoginJavaBean.java -
doLoginUser started... ");
            Session session = DominoUtils.getCurrentSession(FacesContext
                .getCurrentInstance());

            System.out
                .println("docucontent.nsf | doLoginUser() in
TestLoginJavaBean.java Database Session Obtained..."
                    + session);
            // find database in question from current session
            System.out
                .println("docucontent.nsf | doLoginUser() in
TestLoginJavaBean.java based on Database Session...");
            Database database =
session.getDatabase(session.getServerName(),
                "docucontent.nsf");
            System.out
                .println("doLoginUser() in TestLoginJavaBean.java
Fetched Database..."
                    + database);
            System.out.println("Getting a database connection... ");

            System.out
                .println("TestLoginJavaBean.java | doLoginUser
Connected to "
                    + database);
            // Find the view in question
            View view = database.getView(URLACCESS_VIEWNAME);
            System.out

```

```

        .println("TestLoginJavaBean.java | doLoginUser
Connected to "
                + view);
// determine database state for searching
if (database.isFTIndexed())
    database.updateFTIndex(false);
else
    database.updateFTIndex(true);

//perform search via ViewEntryCollection
ViewEntryCollection vec =
view.getAllEntriesByKey(UserName, true);
// ...
// put the userName variable in session to be picked up
through App

```

**FacesContext.getCurrentInstance().getExternalContext().
getSessionMap().put("userName", UserName);**

```

//at this point we assume, the userID column in back-end
(URLACCESS_VIEWNAME) has the value searched
//load count for debugging purposes
//Note, count can be 0 here, thus err Message(s) should
load...
System.out
.println("TestLoginJavaBean.java |
ViewEntryCollection Counts Obtained...");
System.out.println("TestLoginJavaBean.java | ViewEntries "
    + vec.getCount() + " Account(s)");
System.out
.println("TestLoginJavaBean.java |
ViewEntryCollection completed successfully...");

ViewEntry tmpentry = null;
ViewEntry entry = vec.getFirstEntry();

// ensure UserID field/column returned is not null (record not
found) from NSF back-end,
// if so, load text/err Message(s) to user
if (entry == null) {

    // add Message value...
    // TODO: Do something else with the current component
    // example, combine with built-in items
    //...
MyEntries = "Invalid UserID";
//load to custom errMessage label control
message = MyEntries.toString();

// load friendlier debug message to log
// in which case, inputText or Label should load
'Invalid UserID to

// form
// TODO: Create set of custom messages to load to Xpages_
form

System.out
.println("TestLoginJavaBean.java |
ViewEntryCollection completed with error..."
        + MyEntries);

```

```

        // otherwise load the value needed to let user continue
    } else {

        // add value needed for pairing with session, and so
on...
MyEntries = entry.getDocument().getItemValueString("UserID");
        // DEBUG ONLY:
        // load friendly debug message to log
        // System.out
        // .println("TestLoginJavaBean.java |
ViewEntryCollection completed with valid ID..."
        // + MyEntries);
        if (MyEntries.equalsIgnoreCase(Username)) {
            //grab the session and make available in other
pages
            //this should only happen if in fact NSF userID
column has value searched via userName component
        Username = (String) FacesContext.getCurrentInstance()
.getExternalContext().getSessionMap().get("userName");
        System.out.println("Session obtained from Bean... "
            + Username);
        tmpentry = vec.getNextEntry();
        entry.recycle();
        entry = tmpentry;

        System.out
            .println("END DEBUG |
TestLoginJavaBean.java - doLoginUser releasing access to controls... ");

        return "Xsp-success";
        // END DEBUG
    }

}

} catch (NotesException e) {
    System.out.println(e.id + " " + e.text);
    e.printStackTrace();

} catch (Exception e) {

    e.printStackTrace();
}
// return Username;
return Username;
}
}
// ...
// end of program

```

Copy and Paste XSP Code

```

<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core">

    <xp:span style="font-weight:bold;font-size:18pt">UserID</xp:span>
    <xp:span style="font-weight:bold;font-size:18pt"> Login</xp:span>
    <xp:br></xp:br>

    <xp:table>

```

```

        <xp:tr>
            <xp:td>
                <xp:inputText id="userName1" style="width:275.0px"
                    required="true"
value="#{TestLoginJavaBean.userName}"
                    disableClientSideValidation="true">
                </xp:inputText>
            </xp:td>
            <xp:td>
                <xp:label
value="#{ TestLoginJavaBean.message}" id="errMessage"
                    style="width:124.0px;color:rgb(117,11,11);font-
weight:bold"></xp:label>
            </xp:td>
        </xp:tr>
        <xp:tr>
            <xp:td>
                <xp:button value="Login" id="button1">
                    <xp:eventHandler event="onclick" submit="true"
                        refreshMode="complete" immediate="false"
save="true" id="eventHandler1">
                        <xp:this.action><!
[CDATA[#{javascript:TestLoginJavaBean.doLoginUser()}]]></xp:this.ac
tion>
                    </xp:eventHandler>
                </xp:button>
            </xp:td>
            <xp:td></xp:td>
        </xp:tr>
    </xp:table>
    <xp:this.navigationRules>

        <xp:navigationRule outcome="Xsp-success"

viewId="/xpwelcome.xsp"></xp:navigationRule>
    </xp:this.navigationRules>

</xp:view>

```

Copy and Paste XSP Code

Create additional XSP file called **xpwelcome.xsp**, copy and paste XML code

```

<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core">
    <xp:table>
        <xp:tr>
            <xp:td>
                <xp:label id="label2" value="Logged in As:"
                    style="font-
weight:bold;color:rgb(240,222,0);background-color:rgb(0,64,128)"></xp:label>
            </xp:td>
            <xp:td>
                <xp:label
value="#{ sessionScope.TestLoginJavaBean.userName}"
                    id="label1"
                    style="background-
color:rgb(0,64,128);color:rgb(255,255,0);width:120.0px">

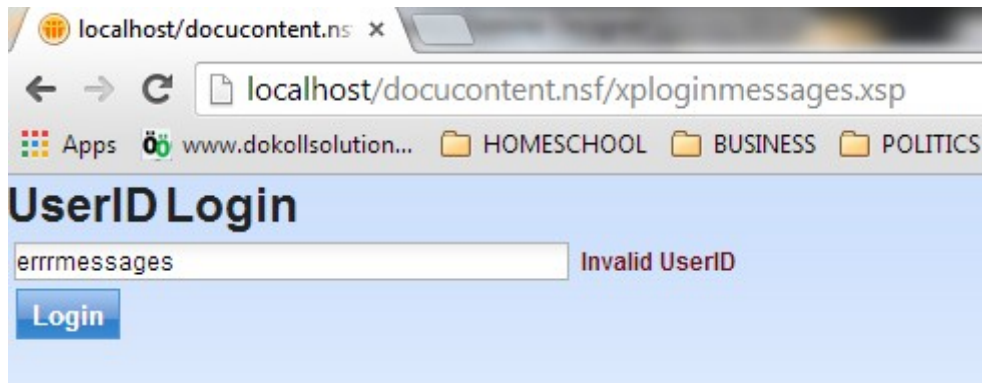
```

```
        </xp:label>
    </xp:td>
</xp:tr>
</xp:table>
<xp:br></xp:br>

</xp:view>
```

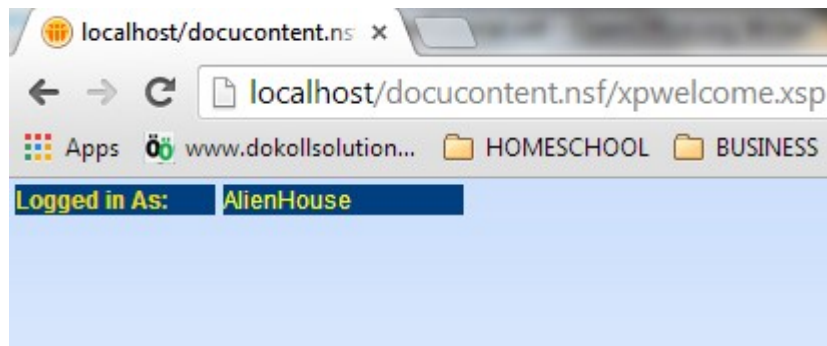
Run to Browser

Be sure to Reference your TestLoginJavaBean.java in facesConfig XML file, and run code



View in Browser

Click button to with own userID from NSF back-end to facilitate creating and grabbing session to view welcome page...



Conclusion:

You can now create and grab session from variable and show it on subsequent Xpages forms; information that can be used to ensure specific users see their own stuff.

Questions, comments, please post a brief message on our [Contact](#) form on the main site.

Thank you for coming...

Version:2014.02.14.1.57.PM