

JavaBean/Backing Bean

Show NSF Secure User Access

Xpages

System Requirements:

Download Domino Designer 8.5.2 Environment (DDE)

<http://www.ibm.com/developerworks/downloads/ls/dominodesigner/>

Introduction:

Build a straight through connection to Lotus Notes Domino Database located on C Drive, fetch/load documents containing security access information only accessible to Admin users. Supply an Xpage the logged-in user security access. Hide/show link control(s) on the Xpages form at will...

Disclaimer:

Information contained in the following is presented as is. This tutorial assumes you have basic programming knowledge. All tutorials are based on an Eclipse/Eclipse-based software. Should you need to familiarize yourself with a certain Eclipse environment, prior to continuing this tutorial, please stop now and see our Tutorials page...

Copy and Paste Xpages inputText /dataTable controls

At this point we assume Domino Designer 8.5.2 is downloaded/installed, you've looked at the LoginBeanSQL class, and you added a link control to your Xpages form(xpnavlinks.xsp). Copy and paste below code samples to your environment.

xpsecurityaccesscheck.xsp

```
<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core"><xp:div style="text-align:center"><xp:span style="font-weight:bold;font-size:19pt">
    Security Access Search</xp:span>
</xp:div><xp:div style="text-align:center"><xp:span style="font-weight:bold;font-size:12pt">Fetch By UserID</xp:span>
</xp:div>

<xp:dataTable rows="30" id="dataTable1" var="thisUser"
    value="#{SearchNSFJavaBean}">
    <xp:this.facets>
        <xp:pager partialRefresh="true" layout="Previous Group Next"
            xp:key="header" id="pager1" for="dataTable1">
        </xp:pager>
        <xp:pager partialRefresh="true" layout="Previous Group Next"
            xp:key="footer" id="pager2" for="dataTable1">
        </xp:pager>
    </xp:this.facets>
```

```

    <xp:column id="column2">
        <xp:this.facets>
            <xp:label value="UserID" id="computedField1"
                xp:key="header" style="font-weight:bold;font-
size:12pt">
                </xp:label>
            </xp:this.facets>
            <xp:text escape="true" id="computedField2"
                value="#{thisUser.userID}">
            </xp:text>
        </xp:column>

    <xp:column id="column1">
        <xp:this.facets>
            <xp:label value="RoleID" id="label1" xp:key="header"
style="font-weight:bold;font-size:12pt">
            </xp:label>
        </xp:this.facets>
        <xp:text escape="true" id="computedField3"
            value="#{thisUser.roleID}">
        </xp:text>
    </xp:column>
    <xp:column id="column3"><xp:this.facets><xp:label value="ViewAccess"
id="label2" xp:key="header" style="font-weight:bold;font-size:12pt">
    </xp:label></xp:this.facets><xp:text escape="true"
id="computedField4" value="#{thisUser.viewAccess}">
    </xp:text></xp:column>
    <xp:column id="column4"><xp:this.facets><xp:label value="AddAccess"
id="label3" xp:key="header" style="font-weight:bold;font-size:12pt">
    </xp:label></xp:this.facets><xp:text escape="true"
id="computedField5" value="#{thisUser.addAccess}">
    </xp:text></xp:column>
    <xp:column id="column5"><xp:this.facets><xp:label value="UpdateAccess"
id="label4" xp:key="header" style="font-weight:bold;font-size:12pt">
    </xp:label></xp:this.facets><xp:text escape="true"
id="computedField6" value="#{thisUser.updateAccess}">
    </xp:text></xp:column>
    <xp:column id="column6"><xp:this.facets><xp:label value="DeleteAccess"
id="label5" xp:key="header" style="font-weight:bold;font-size:12pt">
    </xp:label></xp:this.facets><xp:text escape="true"
id="computedField7" value="#{thisUser.deleteAccess}">
    </xp:text></xp:column>
    <xp:column id="column7"><xp:this.facets><xp:label value="DateCreated"
id="label6" xp:key="header" style="font-weight:bold;font-size:12pt">
    </xp:label></xp:this.facets><xp:text escape="true"
id="computedField8" value="#{thisUser.dateCreated}">
    </xp:text></xp:column>
</xp:dataTable>
<xp:br></xp:br>

    <xp:inputText value="#{SearchNSFJavaBean.searchText}"
id="inputText5" title="Insert UserID"
    style="font-size:9pt;width:131.0px" rendered="true"/>
    <xp:br></xp:br><xp:button value="Search" id="button1">
        <xp:eventHandler event="onclick" submit="true"
            refreshMode="complete" immediate="false" save="true"
            id="eventHandler1">

        <xp:this.action><!--

```

```
[CDATA[#{javascript:SearchNSFJavaBean.doSearchNSFDocs()}]></xp:this.action>
    </xp:eventHandler>
</xp:button>

</xp:view>
```

Copy and Paste Search Feature Code

Copy the following class into your application and search NSF back-end for Unique UserID with specific Security Access. Items that can be used to further restrict access to controls or pages. Copy and paste below code samples to your environment, areas of interest have been highlighted for your convenience.

SearchNSFJavaBean;

```
package com.dokoll.solutions.inc.search;
/**
 * Created from copy: 2012.08.25.9.14.PM
 * SearchNSFJavaBean
 * Read View data into Xpage (JavaBean Search Feature)
 */
//load import for Domino classes
import lotus.domino.*;
import lotus.domino.local.Database;
import lotus.domino.local.Document;
//import for JavaServer Faces classes
import javax.faces.context.FacesContext;
/**
 * @author Dököll Solutions, Inc.
 * @version 2012.08.25.9.14.PM
 */
// begin class
public class SearchNSFJavaBean extends AgentBase {
    //declare Search Criteria variables
    private String UserID;
    private String RoleID;
    private String SearchText;
    //declare variables for page security loads
    private String ViewAccess;
    private String AddAccess;
    private String UpdateAccess;
    private String DeleteAccess;
    private DateTime DateCreated;
    /**
     * @return the dateCreated
     */
    public DateTime getDateCreated() {
        return DateCreated;
    }
    /**
     * @param dateCreated the dateCreated to set
     */
    public void setDateCreated(DateTime dateCreated) {
        DateCreated = dateCreated;
    }
}
```

```

/**
 * @return the viewAccess
 */
public String getViewAccess() {
    return ViewAccess;
}
/**
 * @param viewAccess the viewAccess to set
 */
public void setViewAccess(String viewAccess) {
    ViewAccess = viewAccess;
}
/**
 * @return the addAccess
 */
public String getAddAccess() {
    return AddAccess;
}
/**
 * @param addAccess the addAccess to set
 */
public void setAddAccess(String addAccess) {
    AddAccess = addAccess;
}
/**
 * @return the updateAccess
 */
public String getUpdateAccess() {
    return UpdateAccess;
}
/**
 * @param updateAccess the updateAccess to set
 */
public void setUpdateAccess(String updateAccess) {
    UpdateAccess = updateAccess;
}
/**
 * @return the deleteAccess
 */
public String getDeleteAccess() {
    return DeleteAccess;
}
/**
 * @param deleteAccess the deleteAccess to set
 */
public void setDeleteAccess(String deleteAccess) {
    DeleteAccess = deleteAccess;
}
/**
 * @return the userID
 */
public String getUserID() {
    return UserID;
}
/**
 * @param userID the userID to set
 */
public void setUserID(String userID) {
    UserID = userID;
}

```

```

    }
    /**
     * @return the roleID
     */
    public String getRoleID() {
        return RoleID;
    }
    /**
     * @param roleID the roleID to set
     */
    public void setRoleID(String roleID) {
        RoleID = roleID;
    }
    /**
     * @return the searchText
     */
    public String getSearchText() {
        return SearchText;
    }
    /**
     * @param searchText the searchText to set
     */
    public void setSearchText(String searchText) {
        SearchText = searchText;
    }
}

// open method, this actually runs the whole App
public void doSearchNSFDocs() {
    // let's add a try catch here, to grab errors near the end
    try {
        // Declare Variables, one to hold Documents/ and page through
        results with the other
        Document doc;
        //BEGIN DEBUG
        Database database = (Database) FacesContext.getCurrentInstance()
            .getApplication().getVariableResolver().resolveVariable(
                FacesContext.getCurrentInstance(), "database");
        System.out.println("Database Obtained..." + database);
        // Find the view in question
        View view = database.getView("SecurityAccessView");
        System.out.println("View Obtained..." + view);
        //initialize search through collection
        ViewEntryCollection vec =
view.getAllEntriesByKey("SearchText", true); //
        System.out.println("Collection Obtained..." + vec);
        doc = (Document) view.getFirstDocument();

        System.out.println("Doc Obtained..." + doc);
        System.out.println("Loading to Xpage...");
        // process the document in the View
        while (vec != null) {
            //Load security document to Xpapes form
            UserID = doc.getItemValueString("UserID");
            RoleID = doc.getItemValueString("RoleID");
            ViewAccess = doc.getItemValueString("ViewAccess");
            AddAccess = doc.getItemValueString("AddAccess");
            UpdateAccess =
doc.getItemValueString("UpdateAccess");

```

```

DeleteAccess =
doc.getItemValueString("DeleteAccess");
DateCreated = doc.getCreated();
System.out.println("Loaded to Xpage...");
//...
//load to debug console for testing purposes
System.out.println("Searching UserID "+SearchText);

System.out.println("Found UserID "+ UserID);
System.out.println("Created @ "+ DateCreated);
//...
//recycle to free up mem
doc.recycle();
//...
//this is necessary to avoid NullPointer
//We do not expect another instance of that userID
return;
}
} catch (Exception e) {
    e.printStackTrace();
}
}
}
}

```

SecurityAccessView Screenshot

UserID	RoleID	ViewAccess	AddAccess	UpdateAccess	DeleteAccess	DateCreated
username101	Admin	Yes	Yes	Yes	Yes	08/18/2012 10:18:46 PM
username102	User	Yes	Yes	No	No	08/18/2012 10:14:02 PM

Conclusion:

You can now retrieve Unique User Access to pages and/or controls on demand. Security access information restricted in secured (Admin ONLY) documents can now be maintained. Stay tuned for ability to delete, add, update from the SecurityAccessView.

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

Thank you for coming...