

Döcu Content

Notes Document View

BackingBean

Xpages File or Form Element

Introduction:

Navigate Döcu Content App via links on Xpages form, interact with documents viewing page. This tutorial makes use of existing code previously submitted code via, see:

<http://www.dokollolutionsinc.com/CutAndPasteAttachment.html/>.

Disclaimer:

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

```
RetrieveNewNoticeBackingBean.java
```

```
/**
```

```
* Copyright 2012 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.
```

```
* Program: RetrieveNewNoticeBackingBean.java
```

```
* Created from Copy: 2012.06.11.11.05.PM
```

```
* New Retrieval Bean for Toy Submissions via Xpages
```

```
*
```

```
*/
```

```
package com.osc.its.bulletin.board.JavaBeans;//...
```

```
//java imports
```

```
import java.util.Vector;
```

```
//faces imports
```

```
import javax.faces.context.FacesContext;
```

```
//servlet imports
```

```
import javax.servlet.http.HttpServletRequest;
```

```
//notes imports...
```

```
import lotus.domino.Item;
```

```
import lotus.domino.View;
```

```
import lotus.domino.local.Database;
```

```

import lotus.domino.local.Document;
/**
 * @author Dököll Solutions, Inc.
 * @version 2012.06.11.11.05.PM
 *
 */
public class RetrieveNewNoticeBackingBean {
// Declare variable, load view name
final String ViewName = "By Category";
// Get Keyword collection
@SuppressWarnings( { "null", "unchecked" })
public Keyword[] getKeywords() {
// Declare Keyword Array
Keyword[] keywords = null;
try {
// get the current database being used
Database database = (Database) FacesContext.getCurrentInstance()
.getApplication().getVariableResolver().resolveVariable(
"database");
// Get the full URL and join it with the current document
// from Notes back-end
// 2012.03.02.8.47.PM
HttpServletRequest req = (HttpServletRequest) FacesContext
.getCurrentInstance().getExternalContext().getRequest
();
//...
//Declare and variable to grab full URL
String baseURL =
req.getRequestURL().toString().replace(
req.getRequestURI(), req.getContextPath());
// Find the view in questionView view = database.getView(ViewName);
//load to console for debugging purposes
System.out.println("View Obtained..." + view);
//...
Document sDoc;
//...
Document ndoc;
//grab our first doc
sDoc = (Document) view.getFirstDocument();
//load to console for debugging purposes
System.out.println("Document Obtained..." + sDoc);
// load documents count
keywords = new Keyword[view.getEntryCount()];
// Run through Keyword document collection
int docount = 0;
while (sDoc != null) {
//initialize vector, load items
Vector<Item> itemVector = sDoc.getItems();
//run though and pull off items (attachments) one at a time
for (Item inboundItem : itemVector) {
//ensure item is of type attachment

```

```

if (inboundItem.getType() == Item.ATTACHMENT) {
//declare and initialize attachment placeholder
String AttMents = inboundItem.getValueString();
//call Keyword and fill Category, Topic,
BrowserURL, FileName variables...
Keyword keyword = new Keyword(null, null, null,
null);
// render document(s) to variables of Keyword
object
keyword.setCategory(sDoc
.getItemValueString("Categories"));
keyword.setTopic(sDoc.getItemValueString("Subject"));
keyword.setBrowserURL(baseUrl + "/" + view +
"/"
+ sDoc.getUniversalID() + "?
OpenDocument");
keyword.setFileName(baseUrl + "/" + view + "/"
+ sDoc.getUniversalID() + "/" + "$FILE/"
+ sDoc.getAttachment(AttMents));//
// Send Keyword object into Keywords Array
keywords[docount] = keyword;
// increment counts
docount += 1;// load next doc
ndoc = (Document) view.getNextDocument(sDoc);
sDoc = ndoc;
}
}
}
// return the Array
return keywords;
//catch accordingly
} catch (Exception e) {
e.printStackTrace();
}
// return nothing
return null;
}
}
}

```

Döcu Content View Documents (Xpages Style)

Get a copy of RetrieveNewNoticeBackingBean.java BackingBean through the above URL, also pick up a copy of the associating JavaBean file: Keyword.java. The file should included Xpages Form to compliment the class files. Follow instructions below to start viewing your document as a list in an Xpage.

Note: You may also need a download of any of the following Döcu Apps: Döcu (docu.nsf), or Döcu Content (docucontent.nsf), available on OpenNTF.org, through the Projects page, you can search by App name: (<http://www.openntf.org/Internal/home.nsf/projects.xsp>) or Click our download page: <http://www.dokollolutionsinc.com/>. You need to do this to fully complete the instruction below.

Xpages File Instructions, see screenshots for areas of interest

1. Declare a variable `final String GotoPage = "xpbycategorylist.xsp";` for view page
2. From the above BackingBean, go to `public Keyword[] getKeywords()`
3. Change `keyword.setBrowserURL(baseUrl + "/" + view + "/" +`
4. To `keyword.setBrowserURL(baseUrl + "/" + GotoPage + "?" + "documentId=" +`
5. Create a New Xpages Form the easy way, by dragging and dropping controls on the page
6. Call this Xpage 'xpbycategorylist.xsp', use it to Open a Document in this case
7. DataSource should be based on 'Bulletins' Form in docu.nsf or docucontent.nsf back-end
8. Set Default property in Form (Bulletins) within the On Web Access section to HTML
9. Open DDE and fire Xpages Form called xpbycategory, this feeds into xpbycategorylist
10. Click any row/link under Type in the xpbycategory.xsp Xpage
11. `xpbycategorylist.xsp` should now load to Browser with specific document accessed

Screenshots, BackingBean excerpts, Xpages, and Form elements

RetrieveNewNoticeBackingBean.java;

(1) Declaring variable to hold Xpages Form name

```

58 public class RetrieveNewNoticeBackingBean {
59
60     // Declare variable, load view name
61
62     final String ViewName = "Notices By Category";
63
64     final String GotoPage = "xpbycategorylist.xsp";
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

(2) Viewing Documents via Form element

```

keyword.setBrowserURL(baseUrl + "/" + view + "/"
+ sDoc.getUniversalID() + "?OpenDocument");

```

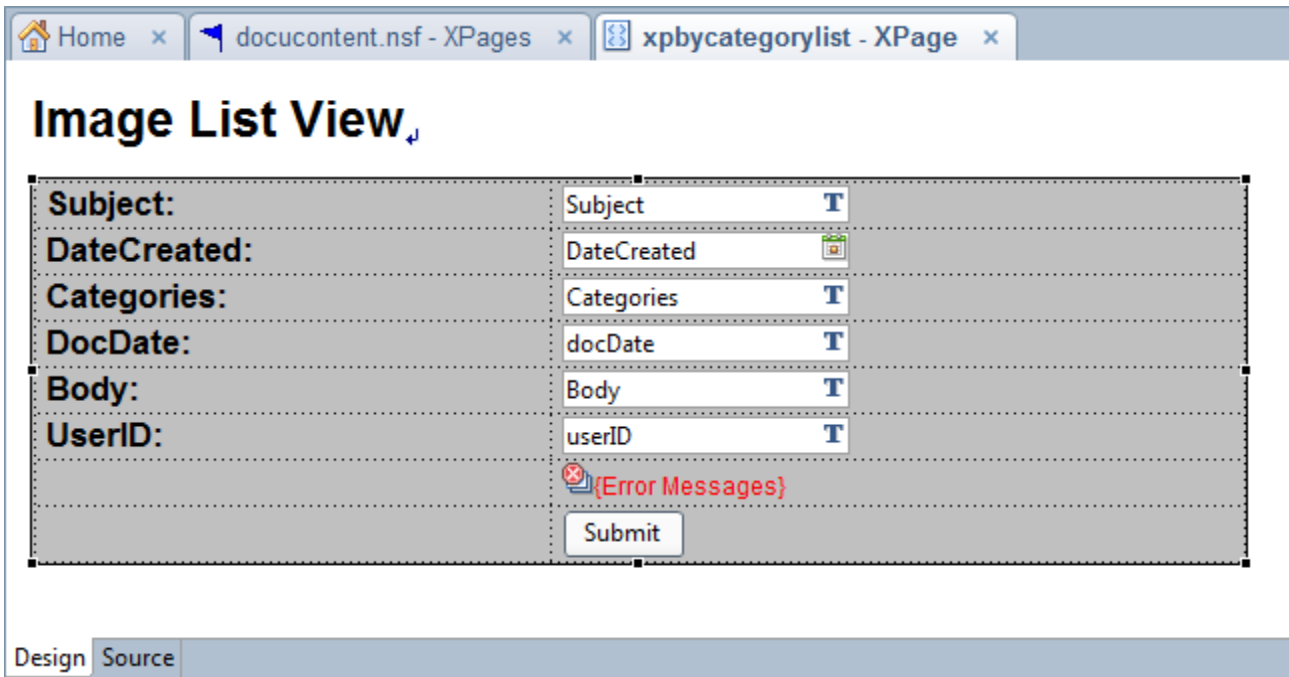
(3) Viewing Documents via Xpages element

```

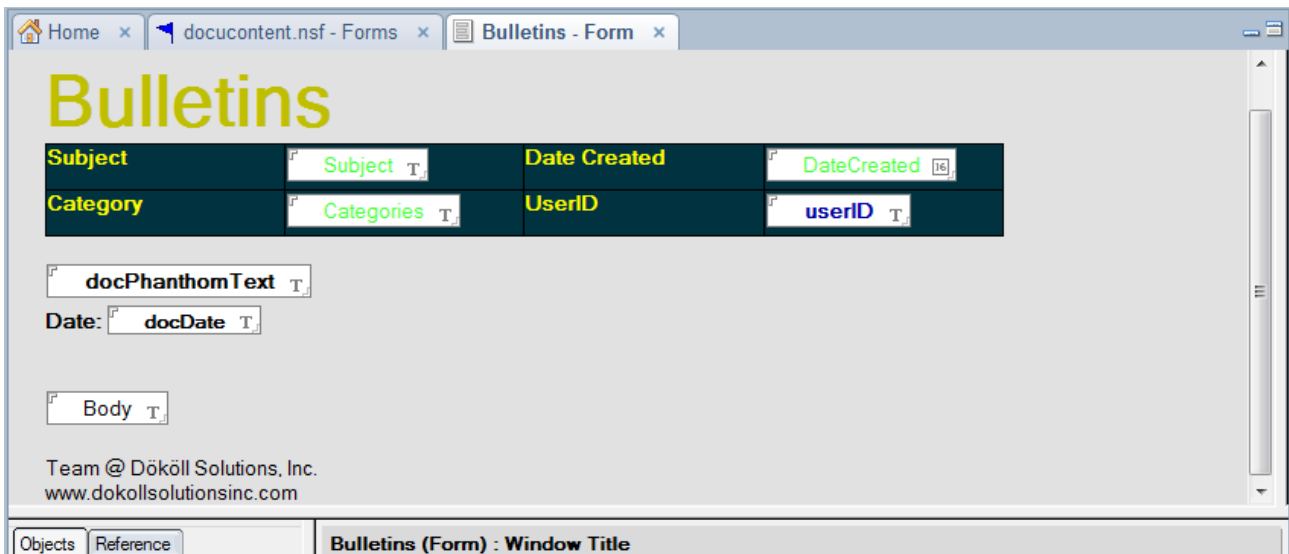
keyword.setBrowserURL(baseUrl + "/" + GotoPage + "?" + "documentId="
+ sDoc.getUniversalID() + "&action="+ "EditDocument");

```

(4) Creating Xpages Form to view documents



(5) Connecting to Bulletins Form as DataSource item



(6) Setting property from Notes to HTML

Form [? X]

Formulas inherit values from selected document
 Inherit entire selected document into rich text field:

On Create
 Automatically enable Edit Mode
 Show context pane
 Display XPage instead: []

On Close
 Present mail send dialog

On Web Access
 Display XPage instead: []
 Content type: Notes
 HTML
 Other
 Character set: Default []
 Generate HTML for all fields
 Active link: [Red] [] Unvisited link: [Blue] [] Visited link: [Purple] []





Data Source Options [Browse] []
 Default data connection: []
 Default metadata object: []

(7) Running Xpages with List of Documents

Home x docucontent.nsf - XPages x xpbycategory - XPage x

Image List

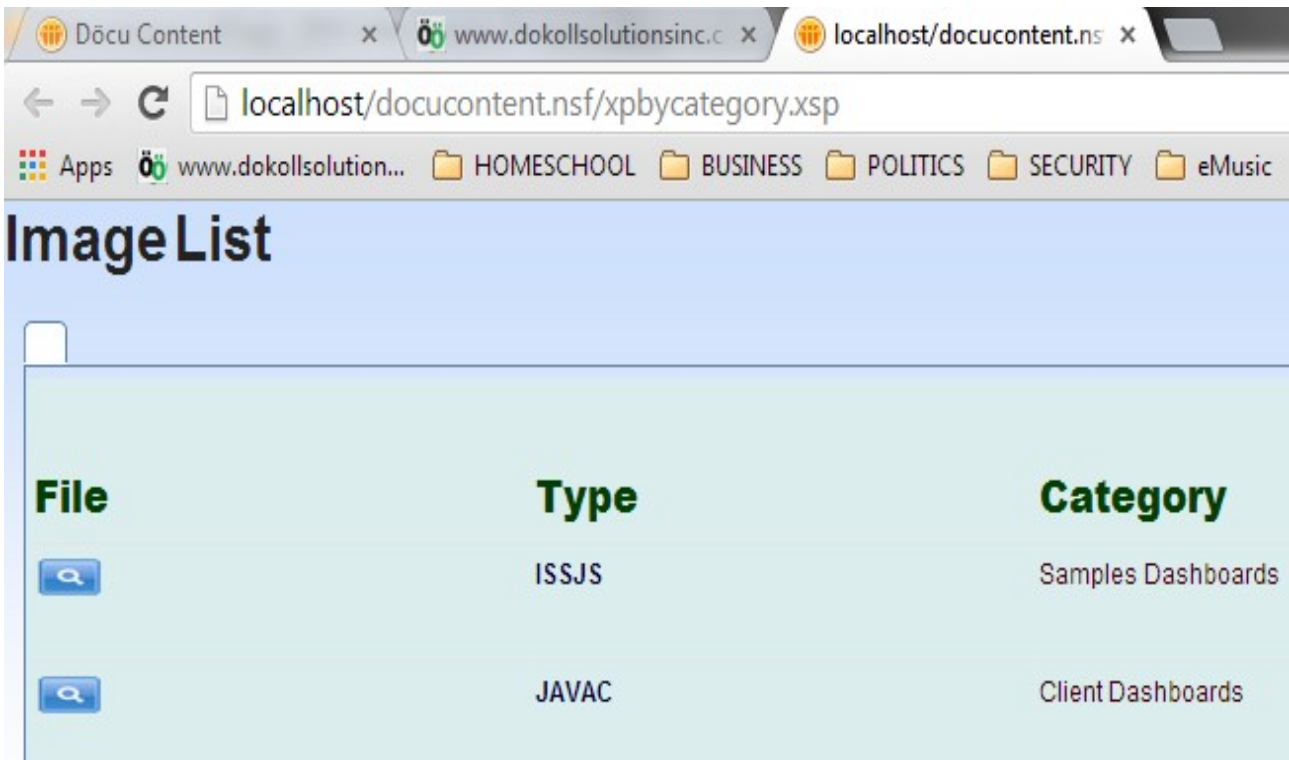
Page1|2|3|4|5

File	Type	Category	Image
 	 {link2}	{computedField2}	

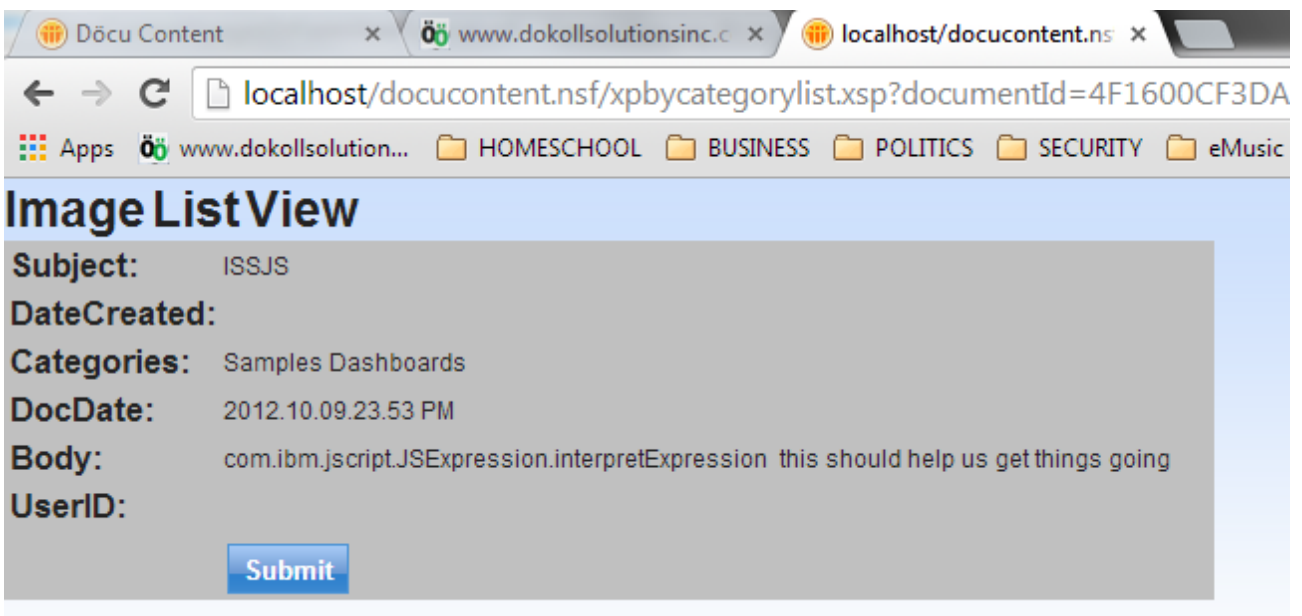
Page1|2|3|4|5

Design Source

(8) Clicking link to make available specific Doc



(9) Viewing specific Doc



TIP: You should make a copy of teh Bulletins Form to toggle viewing documents via Form or Xpages depending on security access of your users.

Conclusion:

You can now view documents via Xpages using documentId rather than a view/form. This should facilitate setting pages that only certain users need to see, while leaving your main Form design intact.

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

Thank you for coming...

Version:2014.01.04.12.28.PM