

DB2, JavaBean

Retrieve DB2 Data - Xpages Form (Part 3)

Xpages

System Requirements:

Download DB2 Express-C

<http://www-01.ibm.com/software/data/db2/express/download.html>

Download Domino Designer 8.5.3 Environment (DDE)

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

Introduction:

Connect to DB2 using Xpages form, JavaBean, and Backing Bean to retrieve data. Information to be used to search current DOMINODB database dynamically via links...

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...

Establish a connection to DOMINODB (DB2)

Use previous connector class for DB2 and run below code samples via Xpages form to collect resultset. Copy and paste to your environment, areas of interest have been highlighted for your convenience.

SubjectsInfo;

```
/**
```

```
* 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.
```

```
*
```

```
* @AppName: Docu.nsf  
* Program: SubjectsInfo.java
```

```
* Created: 2012.11.21.8.05.PM
*/
```

```
package com.dokoll.solutions.inc.db2.test;
```

```
/**
 *
 * @author Dököll Solutions, Inc.
 * @version 2012.11.21.8.05.PM
 */
```

```
public class SubjectsInfo {
    public String SubID;
    public String UserName;
    public String AllegationCode;
    public String Office;
    public String Investigator;
    public String DateOpen;
    public String Age;
    public String AllegationNumber;
    public String Licensee;
    public String Subject;
    public String Region;
    /**
     * @return the region
     */
    public String getRegion() {
        return Region;
    }
    /**
     * @param region the region to set
     */
    public void setRegion(String region) {
        Region = region;
    }
    /**
     * @return the subjectURL
     */
    public String getSubjectURL() {
        return SubjectURL;
    }
    /**
     * @param subjectURL the subjectURL to set
     */
    public void setSubjectURL(String subjectURL) {
        SubjectURL = subjectURL;
    }
    public String SubjectURL;

    /**
     * @return the subID
     */
    public String getSubID() {
        return SubID;
    }
    /**
     * @param subID the subID to set
```

```

    */
    public void setSubID(String subID) {
        SubID = subID;
    }
    /**
     * @return the userName
     */
    public String getUserName() {
        return UserName;
    }
    /**
     * @param userName the userName to set
     */
    public void setUserName(String userName) {
        UserName = userName;
    }
    /**
     * @return the allegationCode
     */
    public String getAllegationCode() {
        return AllegationCode;
    }
    /**
     * @param allegationCode the allegationCode to set
     */
    public void setAllegationCode(String allegationCode) {
        AllegationCode = allegationCode;
    }
    /**
     * @return the office
     */
    public String getOffice() {
        return Office;
    }
    /**
     * @param office the office to set
     */
    public void setOffice(String office) {
        Office = office;
    }
    /**
     * @return the investigator
     */
    public String getInvestigator() {
        return Investigator;
    }
    /**
     * @param investigator the investigator to set
     */
    public void setInvestigator(String investigator) {
        Investigator = investigator;
    }
    /**
     * @return the dateOpen
     */
    public String getDateOpen() {
        return DateOpen;
    }
    /**
     * @param dateOpen the dateOpen to set
     */
    public void setDateOpen(String dateOpen) {
        DateOpen = dateOpen;
    }
}

```

```

/**
 * @return the age
 */
public String getAge() {
    return Age;
}
/**
 * @param age the age to set
 */
public void setAge(String age) {
    Age = age;
}
/**
 * @return the allegationNumber
 */
public String getAllegationNumber() {
    return AllegationNumber;
}
/**
 * @param allegationNumber the allegationNumber to set
 */
public void setAllegationNumber(String allegationNumber) {
    AllegationNumber = allegationNumber;
}
/**
 * @return the licensee
 */
public String getLicensee() {
    return Licensee;
}
/**
 * @param licensee the licensee to set
 */
public void setLicensee(String licensee) {
    Licensee = licensee;
}
/**
 * @return the subject
 */
public String getSubject() {
    return Subject;
}
/**
 * @param subject the subject to set
 */
public void setSubject(String subject) {
    Subject = subject;
}
}
public SubjectsInfo(String SubID,String UserName,String
    AllegationCode,String Office, String Investigator, String
DateOpen, String Age, String AllegationNumber, String Licensee, String Subject,
String SubjectURL, String Region) {
    SubID = this.SubID;
    UserName = this.UserName;
    AllegationCode = this.AllegationCode;
    Office = this.Office;
    Investigator = this.Investigator;
    DateOpen = this.DateOpen;
    Age = this.Age;
    AllegationNumber = this.AllegationNumber;
    Licensee = this.Licensee;
    Subject = this.Subject;
    SubjectURL = this.SubjectURL;
    Region = this.Region;
}

```

```
}  
  
}
```

RetrieveSubjectsBackingBean;

```
/**
```

```
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: RetrieveSubjectsBackingBean.java  
* Created from Copy: 2012.11.21.8.05.PM  
* New Retrieval Bean from DB2 via Xpages  
*  
*/
```

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

```
//...  
//java imports  
import java.sql.Connection;  
import java.sql.ResultSet;  
import java.sql.Statement;  
//faces imports  
import javax.faces.context.FacesContext;  
import javax.servlet.http.HttpServletRequest;  
//Docu imports  
import com.dokoll.solutions.inc.Utills.DB2Connector;  
import com.dokoll.solutions.inc.db2.test.SubjectsInfo;
```

```
/**  
* @author Dököll Solutions, Inc.  
* @version 2012.11.21.8.05.PM  
*  
*/
```

```
@ManagedBean  
@RequestScoped
```

```
public class RetrieveSubjectsBackingBean {
```

```
// Get SubjectsInfo collection  
@SuppressWarnings( { "null", "unchecked" })
```

```

public SubjectsInfo[] getKeywords() {

    // Declare SubjectsInfo Array
    SubjectsInfo[] keywords = null;

    try {

        //...
        Connection connection = DB2Connector.getConnection();

        Statement stat = connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);

        connection.setAutoCommit(true);

        ResultSet rs = stat.executeQuery("select * from DB2ADMIN.USER_DETAILS");
        //...

        //declare and initialize counts
        int docount = 0;
        int rowcount = 0;
        if (rs.last()) {
            rowcount = rs.getRow();
            rs.beforeFirst();

        keywords = new SubjectsInfo[rowcount];

        //submit results to Xpages form
        while (rs.next()) {

            //call String SubID,String UserName,String AllegationCode,String Office, String
            Investigator....
            SubjectsInfo keyword = new SubjectsInfo(null, null, null, null, null, null,
            null, null, null, null,null,null);
            //loads DB2 results into SubjectsInfo object...
            keyword.setSubID(rs.getString("ID"));
            keyword.setUserName(rs.getString("UserName"));
            keyword.setAllegationCode(rs.getString("AllegationCode"));
            keyword.setInvestigator(rs.getString("Investigator"));
            keyword.setDateOpen(rs.getString("DateOpen"));
            keyword.setOffice(rs.getString("Office"));
            keyword.setAge(rs.getString("Age"));
            keyword.setOffice(rs.getString("Office"));
            keyword.setAllegationNumber(rs.getString("AllegationNumber"));
            keyword.setLicensee(rs.getString("Licensee"));
            keyword.setSubject(rs.getString("Subject"));

            keywords[docount] = keyword;

            // increment counts
            docount += 1;

        }

        rs.close();
        connection.close();

    }

    // return the Array
    return keywords;
}

```

```

//catch accordingly
} catch (Exception e) {

e.printStackTrace();

}

// return nothing
return null;

}

}

```

Retrieve DB2 Data into Xpages form

Retrieve data from DB2 database (DOMINODB). Copy and paste to your environment, areas of interest have been highlighted for your convenience.

xpdb2bycategory.xsp;

```

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

    <xp:dataTable rows="5" id="siteTable" var="contact"

        value="#{javascript:RetrieveSubjectsBackingBean.getKeywords()}"
        style="width:50%">

        <xp:this.facets>

            <xp:pager layout="Previous Group Next" xp:key="header"
id="pager1"
                for="siteTable">

            </xp:pager>

            <xp:pager layout="Previous Group Next" xp:key="footer"
id="pager2"
                for="siteTable" partialRefresh="true">

            </xp:pager>

        </xp:this.facets>

        <xp:column id="userNameColumn" style="font-weight:bold;font-
size:10pt">

            <xp:this.facets>
                <xp:span xp:key="header">UserName</xp:span>
            </xp:this.facets>

            <xp:text escape="true" id="userNameField"
value="#{contact.userName}">

            </xp:text>

        </xp:column>

        <xp:column id="allegationNumberColumn" style="font-weight:bold;font-
size:10pt">

```

```

        <xp:this.facets>
            <xp:span xp:key="header">Allegation Code</xp:span>
        </xp:this.facets>

        <xp:text escape="true" id="allegationNumberField"
value="#{contact.allegationNumber}">

            </xp:text>

        </xp:column>

        <xp:column id="officeColumn" style="font-weight:bold;font-
size:10pt">
            <xp:this.facets>
                <xp:span xp:key="header">Office</xp:span>
            </xp:this.facets>

            <xp:text escape="true" id="officeField"
value="#{contact.office}">

                </xp:text>

            </xp:column>

        </xp:dataTable>

    </xp:view>

```

UserName	Allegation Code	Office
username14	2010.12.19.1.10.AM	Montréal
username16	2011.10.11.11.10.AM	Chcago
username15	2011.10.19.1.10.AM	Chicago
username16	2011.10.11.11.10.AM	Chcago
username15	2011.10.19.1.10.AM	Chicago

Conclusion:

You can now use an Xpages form to connect to DB2 database to retrieve data back into the front-end. Information to be used to search current DOMINODB database dynamically via links...

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

Thank you for coming...