

Dököll Solutions, Inc.

IBM | Oracle

Domino Designer, JDeveloper, DB2, WebLogic,
JSF, Xpages, Web Services

Part Three - B

System Requirements

Software Used	Language Used	Protocol/Framework Used
Microsoft Windows 7, 8, 10	VBScript, Batch	Active Directory
Microsoft Internet Explorer	N/A	TCPIP, HTTP
Google Chrome	N/A	TCPIP, HTTP
Mozilla FireFox	N/A	TCPIP, HTTP
Oracle JDeveloper Version 12..x IDE	CSS, HTML, JavaScript, Java	HTTP, TCPIP, Java Server Faces
Oracle Integrated WebLogic Server 12.x	WSDL	HTTP, TCPIP
IBM DB2 Express-C v9.7.4	SQL	SQL Database Server
IBM Notes Domino Designer 8.5.3	CSS, HTML, JavaScript, Java, JavaAgent, XML	HTTP, TCPIP, Xpages

Disclaimer:

Information contained in the following is presented as is. This tutorial assumes you have basic programming and software configuration knowledge. All tutorials are based on Oracle Fusion Middleware and IBM Lotus Notes products, including and not limited to items stated in System Requirements. Should you need to familiarize yourself with JDeveloper or Domino Designer environments, prior to continuing this tutorial, please stop now, and see our Tutorials page: www.dokollsolutionsinc.com or consult your favourite or preferred Search Engine/Company sites for support...

Introduction:

Create New App with JDeveloper with a connection to existing DB2 database back-end to communicate with IBM Lotus Domino App via Java Web Service to exchange static and database records.

Steps:

- (1) Create Domino JavaAgent to Consume the Web Service (Figure 1)
- (2) Import Web Service Consumer into JavaAgent (Figure 2)
- (3) Reference Provider Service Call in Domino Agent (Figures 3, 4a, 4b)
- (4) Inject Generated Web Service classes created (Figure 5)
- (5) Create Xpage to submit to NSF back-end, for Oracle Query (Figure 6)
- (6) Run JavaAgent with IBM Notes Client, Connect to Oracle
- (7) Look in JavaAgent Web Console for Results
- (8) Make Copies of Both Applications

Brief Recap: With these instructions, you should be able to benefit from existing Domino Designer application by following screenshots, to create a Web Service consumer with a JavaAgent and run process to Query provider Web Service housed on Oracle JDeveloper.

Figure 1

Demonstrating the ID obtained from DB2 is used to query DB2 back-end to see if value exists or not...

```
*
* @AppName: DocuContent.nsf
* Program: HTTPOracleWebServiceRecordJavaAgent || OracleWebServiceJavaAgent
* jQuery Report data for Xpage, to be read by JavaBean,
* based on WebNews Web Service to be read from JDeveloper Apps
* Created: 2016.07.24.1.10.AM
*/

import java.io.PrintWriter;

import lotus.domino.AgentBase;
import lotus.domino.Database;
import lotus.domino.Document;
import lotus.domino.Session;
import lotus.domino.View;

//Web Services imports
import com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBean;
import com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBeanServiceLocator;

/**
 * @author Dokoll Solutions, Inc.
 * @version 2016.07.24.1.10.AM
 *
 */

public class OracleWebServiceJavaAgent extends AgentBase {
    final String GotoPage =
        "http://localhost/docucontent.nsf/";
    //button code...
    public void NotesMain() {
        try {
            // -----

```

OracleWebServiceJavaAgent.java

```
/**
 * Copyright 2016 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: DocuContent.nsf
 * Program: HTTPOracleWebServiceRecordJavaAgent || OracleWebServiceJavaAgent
 * jQuery Report data for Xpage, to be read by JavaBean,
 * based on WebNews Web Service to be read from JDeveloper Apps
 * Created: 2016.07.24.1.10.AM
 */

import java.io.PrintWriter;
import lotus.domino.AgentBase;
import lotus.domino.Database;
import lotus.domino.Document;
import lotus.domino.Session;
import lotus.domino.View;
//Web Services imports
import com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBean;
import com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBeanServiceLocator;
/**
 * @author Dököll Solutions, Inc.
 * @version 2016.07.24.1.10.AM
 *
 */
public class OracleWebServiceJavaAgent extends AgentBase {
    final String GotoPage =
        "http://localhost/docucontent.nsf/";
    //button code...
    public void NotesMain() {
        try {
            // Session
            Session session = this.getSession();
            // Output using PrintWriter
            PrintWriter printWriter = this.getAgentOutput();
            // ...
            Database database = session.getDatabase("", "docucontent.nsf");
            System.out.println("database obtained..." + database);
            // ensure database is not null
            if (database != null) {
                // ...
                View uLogs = database.getView("UserWebServicesViewListings");
                System.out.println("View obtained..." + uLogs);
                if (database.isFTIndexed()) {
                    System.out.println("database is indeed indexed for searching...");
                    // ensure view is not null
                    if (uLogs != null) {
                        // grab first doc

```

```

Document doc = uLogs.getFirstDocument();
//String/Int object(s) declaration to collect records
String strUserName = null;
String strUserNumber = null;
String strPageID = null;
String strUserIP = null;
String strUserDate = null;
int c = 0;
//prepare HTML page to build jQuery report
//TODO: Ensure code is up to snuff for actual Servers
printWriter.println("<html>");
printWriter.println("<head>");
printWriter.println("<title>jQuery Report</title>");
printWriter.println("<meta charset='utf-8'>");
printWriter.println("<meta name='viewport' content='width=device-
width, initial-scale=1'> ");
printWriter.println("<link rel='stylesheet'
href='/docucontent.nsf/mobile/app/tree/css/themes/default/jquery.mobile-
1.2.0.css' /> ");
printWriter.println("<link rel='stylesheet'
href='/docucontent.nsf/mobile/app/tree/docs/_assets/css/jqm-docs.css'/>");
printWriter.println("<script
src='/docucontent.nsf/mobile/app/tree/js/jquery.js'></script>");
printWriter.println("<script
src='/docucontent.nsf/mobile/app/tree/docs/docs/_assets/js/jqm-
docs.js'></script>");
printWriter.println("<script
src='/docucontent.nsf/mobile/app/tree/js/jquery.mobile-1.2.0.js'></script>");
printWriter.println("</head>");
printWriter.println("<body>");

printWriter.println("<div align='left'>");
printWriter.println("<div data-role='content' align='left'>");
printWriter.println("<div class='content-secondary'>");
printWriter.println("<ul data-role='listview' data-theme='d' data-
divider-theme='d'>");
printWriter.println("<li data-theme='a'>Oracle Web Services
Report</li>");
//Scan view for all documents
while (doc != null) {
c += 1;
// Retrieve contents spit out HTML
// TODO: further Mix and Play with Xpages jQuery, add more
functionality to sites
strUserName = doc.getItemValueString("subjectUserName");
strUserNumber = doc
.getItemValueString("subjectUserRole");
strPageID = doc.getItemValueString("PageID");
strUserIP = doc.getItemValueString("UserIP");
strUserDate = doc.getItemValueString("subjectDateCreated");
printWriter.println("<table border='1'><tr>");
printWriter.println("<li data-role='list-divider'>" + strUserName
+ "<span class='ui-li-count'>" + " " + strUserNumber+ "</span></li>"
+ "<li><a href="+GotoPage + uLogs + "/" + doc.getUniversalID() +
"?OpenDocument" + " target='_parent'><h3>" + strPageID + "</h3><p><strong>"
strUserIP
+ "</strong></p><p><a href="+GotoPage + uLogs + "/" +
doc.getUniversalID() + "?OpenDocument" +
target='_parent'>" + doc.getUniversalID() + "</a></p><p class='ui-li-
aside'><strong>"
+ c + "</strong></p></a></li></ul></div></div></div>");

```

MyWebSrsBean stub = new

```

MyWebSrsBeanServiceLocator () .getMyWebSrsBeanPort () ;
    //do something with the result of your search
    //example submit and return value via submitEntry method from Web
Service Provider file in Oracle JDeveloper
    //2016.07.24.1.10.AM
    String BrowserValue = strUserData.toString();
    //Submit view item to Web Service variable in JDeveloper
    System.out.println("stub BrowserValue: " +
stub.submitEntry (BrowserValue) ;
        doc = uLogs.getNextDocument (doc);
    }
    } else {
        System.out.println("database not indexed for searching...");
    }
    //recycle for memory...
    uLogs.recycle();
}
// ...
database.recycle();
}
// ...
session.recycle();
} catch (Exception e) {
    e.printStackTrace();
}
}
}
}

```

Figure 3

JavaAgent properties, with Consumer service as Script Lib import

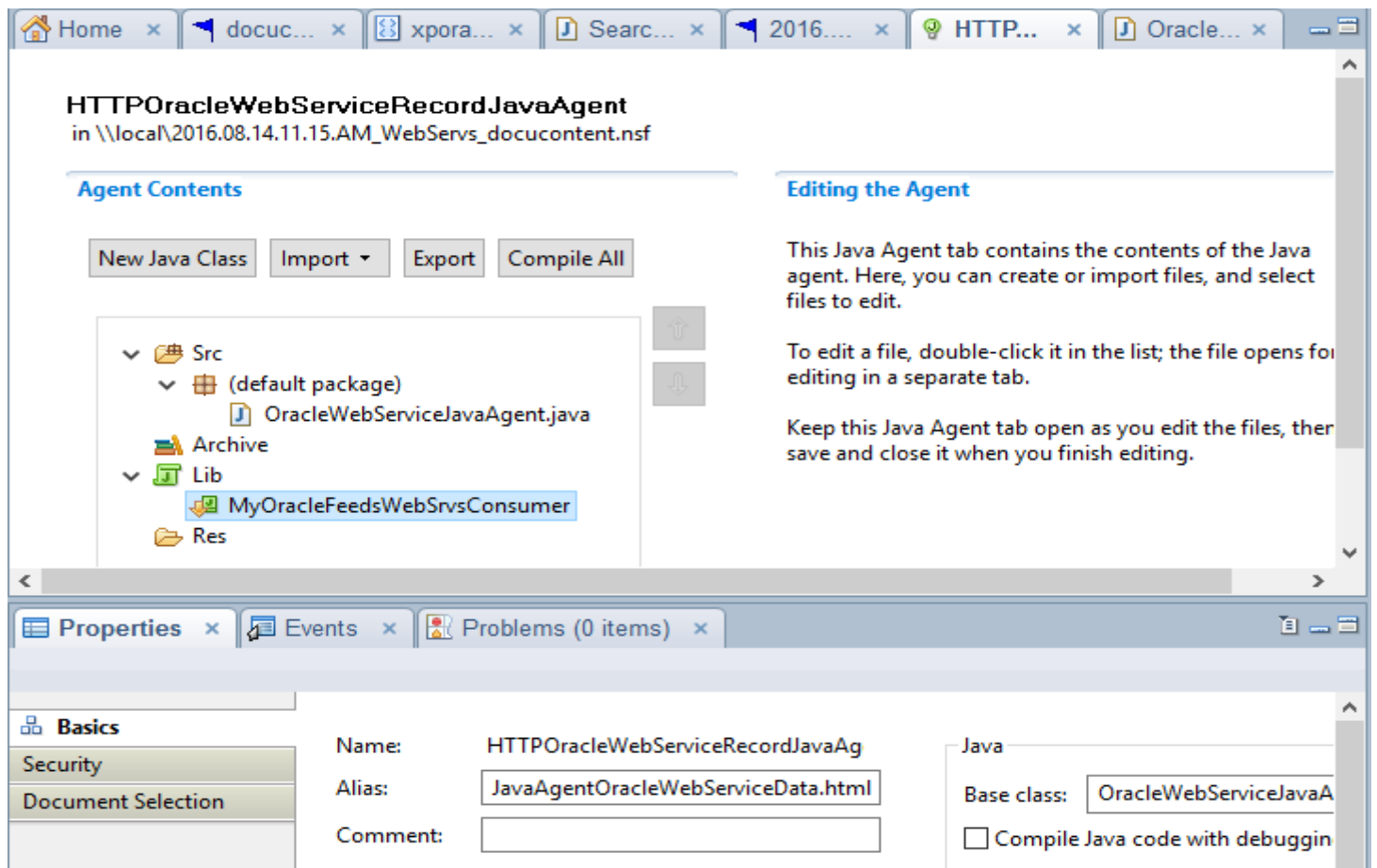


Figure 4a

Web Services Information, properties data, generated class code

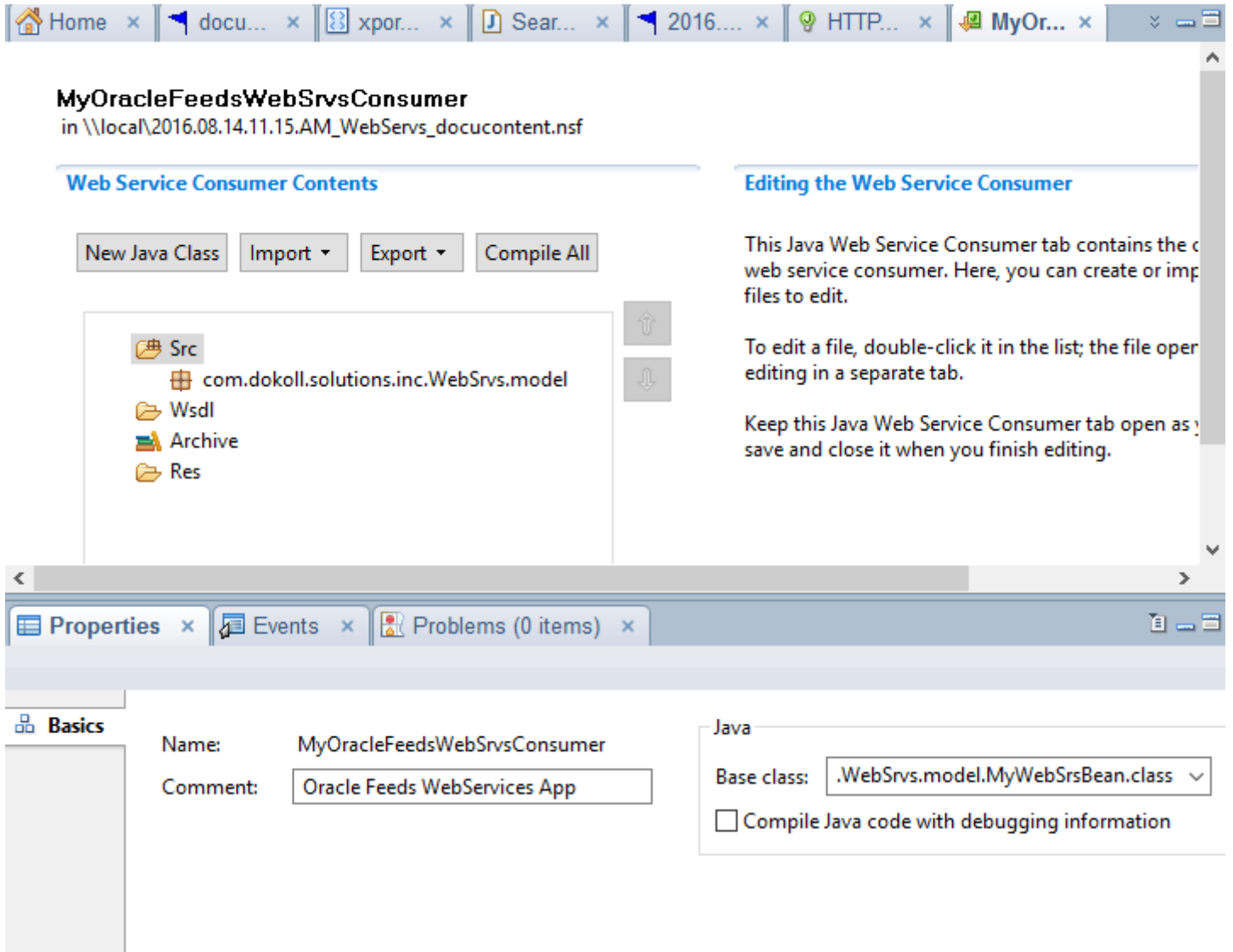
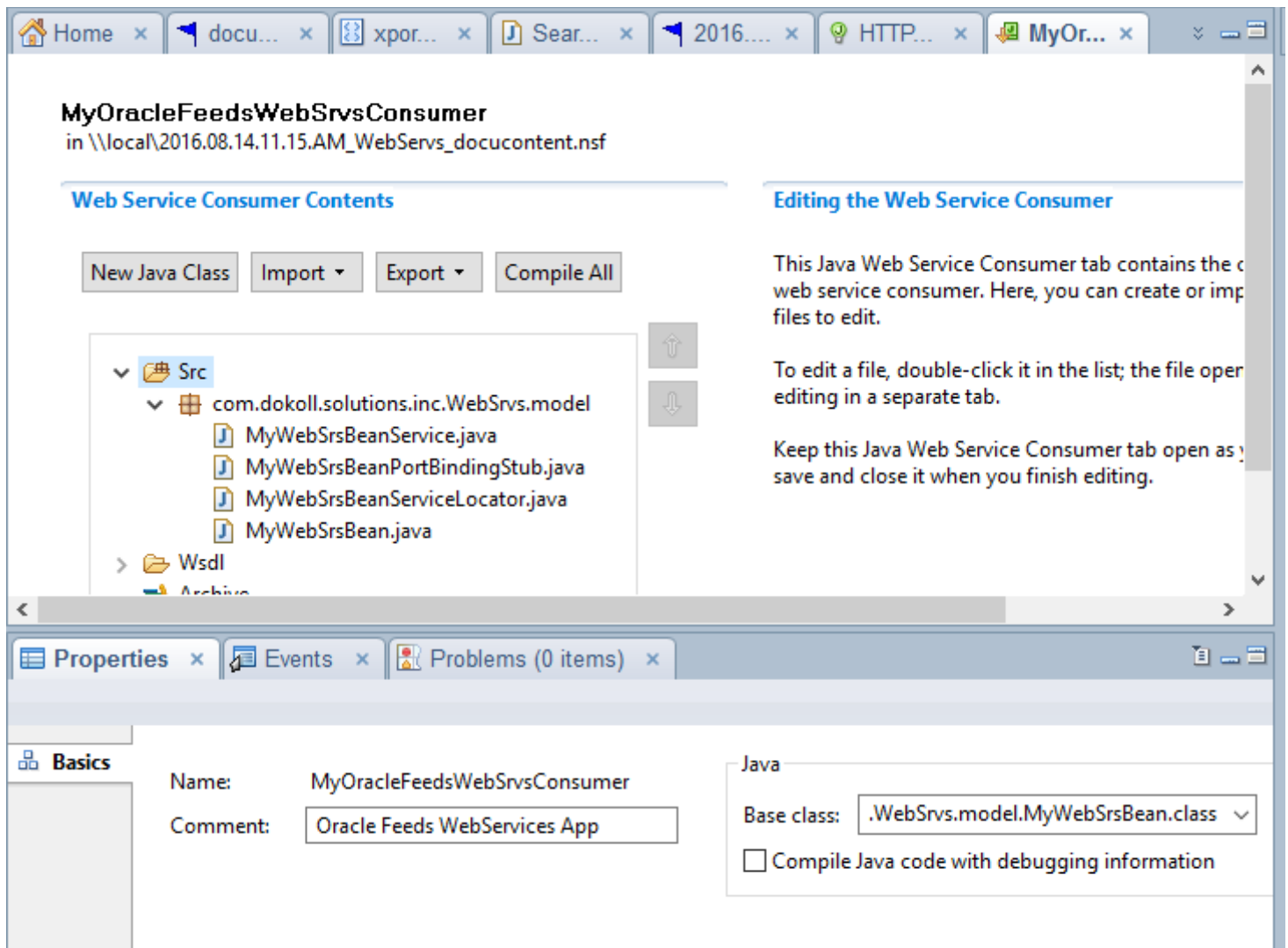


Figure 4b

Web Services Information, properties data, generated class code



Brief Recap: if no errors and the JavaAgent compiles okay, with the Consumer Web Service attached, inspect following code and keep going...

MyWebSrsBeanServiceLocator.java

```

package com.dokoll.solutions.inc.WebSrvs.model;

public class MyWebSrsBeanServiceLocator extends
lotus.domino.websvc.client.Service implements
com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBeanService {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    public MyWebSrsBeanServiceLocator() {
        super("HttpModelWebSrvsIncSolutionsDokollComMyWebSrsBeanService");
    }

    // Use to get a proxy class for MyWebSrsBeanPort

```

```

    private final java.lang.String MyWebSrsBeanPort_address =
"http://localhost:7101/MyWebServApp-Model-context-root/MyWebSrsBeanPort";

    public java.lang.String getMyWebSrsBeanPortAddress() {
        return MyWebSrsBeanPort_address;
    }

    // The WSDD service name defaults to the port name.
    private final java.lang.String MyWebSrsBeanPortWSDDServiceName =
"MyWebSrsBeanService.MyWebSrsBeanPort";

    public com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBean
getMyWebSrsBeanPort() throws javax.xml.rpc.ServiceException {
    java.net.URL endpoint;
    try {
        endpoint = new java.net.URL(MyWebSrsBeanPort_address);
    }
    catch (java.net.MalformedURLException e) {
        throw new javax.xml.rpc.ServiceException(e);
    }
    return getMyWebSrsBeanPort(endpoint);
}

    public com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBean
getMyWebSrsBeanPort(java.net.URL portAddress) throws
javax.xml.rpc.ServiceException {
    try {
        com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBeanPortBindingStub
_stub = new
com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBeanPortBindingStub(portAddress,
this);
        _stub.setPortName(MyWebSrsBeanPortWSDDServiceName);
        return _stub;
    }
    catch (lotus.domino.types.Fault e) {
        return null;
    }
}

/**
 * For the given interface, get the stub implementation.
 * If this service has no port for the given interface,
 * then ServiceException is thrown.
 */
    public java.rmi.Remote getPort(Class serviceEndpointInterface) throws
javax.xml.rpc.ServiceException {
    try {
        if
(com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBean.class.isAssignableFrom(serv
iceEndpointInterface)) {
            com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBeanPortBindingSt
ub _stub = new
com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBeanPortBindingStub(new
java.net.URL(MyWebSrsBeanPort_address), this);
            _stub.setPortName(MyWebSrsBeanPortWSDDServiceName);
            return _stub;
        }
    }
    catch (java.lang.Throwable t) {
        throw new javax.xml.rpc.ServiceException(t);
    }
    throw new javax.xml.rpc.ServiceException("There is no stub
implementation for the interface: " + (serviceEndpointInterface == null ?
>null" : serviceEndpointInterface.getName()));
}

```



```

    }

    /**
     * For the given interface, get the stub implementation.
     * If this service has no port for the given interface,
     * then ServiceException is thrown.
     */
    public java.rmi.Remote getPort(javax.xml.namespace.QName portName, Class
serviceEndpointInterface) throws javax.xml.rpc.ServiceException {
        if (portName == null) {
            return getPort(serviceEndpointInterface);
        }
        String inputPortName = portName.getLocalPart();
        if ("MyWebSrsBeanPort".equals(inputPortName)) {
            return getMyWebSrsBeanPort();
        }
        else {
            java.rmi.Remote _stub = getPort(serviceEndpointInterface);
            ((lotus.domino.websvc.client.Stub) _stub).setPortName(portName);
            return _stub;
        }
    }

    public javax.xml.namespace.QName getServiceName() {
        return new
javax.xml.namespace.QName("http://model.WebSrvs.inc.solutions.dokoll.com/",
"MyWebSrsBeanService");
    }

    private java.util.HashSet ports = null;

    public java.util.Iterator getPorts() {
        if (ports == null) {
            ports = new java.util.HashSet();
            ports.add(new javax.xml.namespace.QName("MyWebSrsBeanPort"));
        }
        return ports.iterator();
    }
}

```

MyWebSrsBeanPortBindingStub.java

```

package com.dokoll.solutions.inc.WebSrvs.model;

public class MyWebSrsBeanPortBindingStub extends lotus.domino.websvc.client.Stub
implements com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBean {

    public MyWebSrsBeanPortBindingStub(java.net.URL endpointURL,
javax.xml.rpc.Service service) throws lotus.domino.types.Fault {
        super(endpointURL, service);
    }

    public java.lang.String submitEntry(java.lang.String arg0) throws
java.rmi.RemoteException {
        lotus.domino.websvc.client.Call _call = createCall("submitEntry");
        java.lang.Object _resp = _call.invoke(new java.lang.Object[] {arg0});
        return (java.lang.String) _call.convert(_resp, java.lang.String.class);
    }
}

```

```
}  
}
```

MyWebSrsBeanService.java

```
/*  
 * To instantiate a callable instance for invoking this service, copy and paste  
from the following:  
  
import com.dokoll.solutions.inc.WebSrvs.model.*;  
  
// Refer to com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBean.java for  
service interface methods:  
MyWebSrsBean stub = new MyWebSrsBeanServiceLocator().getMyWebSrsBeanPort();  
*/  
  
package com.dokoll.solutions.inc.WebSrvs.model;  
  
public interface MyWebSrsBeanService extends javax.xml.rpc.Service {  
  
    public java.lang.String getMyWebSrsBeanPortAddress();  
  
    public com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBean  
getMyWebSrsBeanPort() throws javax.xml.rpc.ServiceException;  
  
    public com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBean  
getMyWebSrsBeanPort(java.net.URL portAddress) throws  
javax.xml.rpc.ServiceException;  
}
```

MyWebSrsBean.java

```
package com.dokoll.solutions.inc.WebSrvs.model;  
  
public interface MyWebSrsBean extends lotus.domino.types.PortTypeBase,  
java.rmi.Remote {  
    public java.lang.String submitEntry(java.lang.String arg0) throws  
java.rmi.RemoteException;  
}
```

Figure 6

Xpage to submit record to back-end View so that JavaAgent can have a value to Query Web Service Provider in JDeveloper

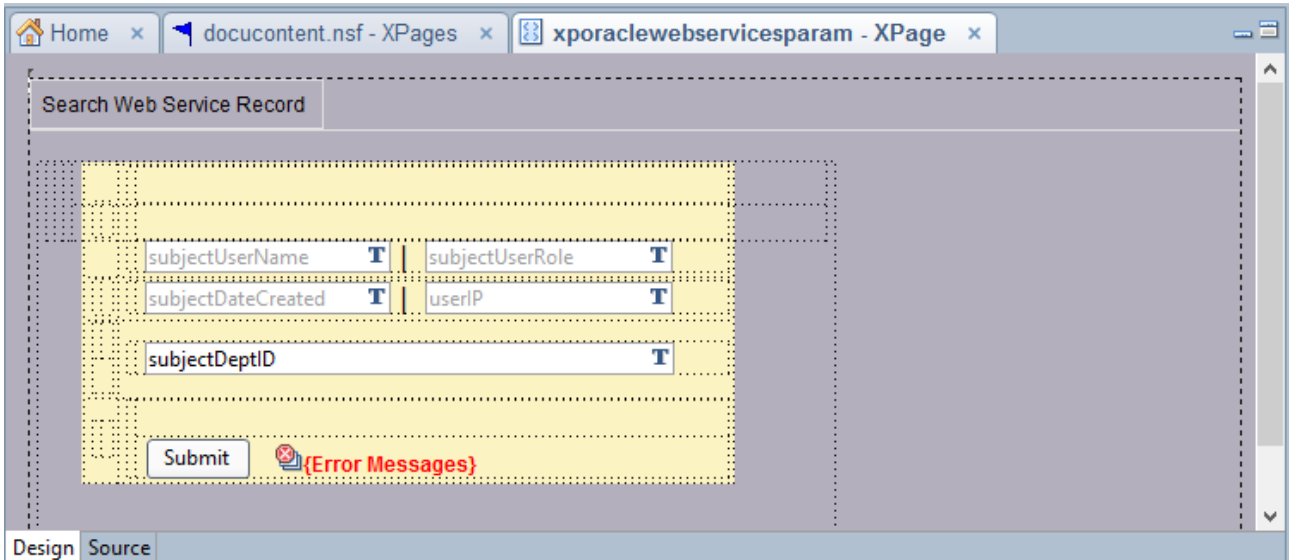
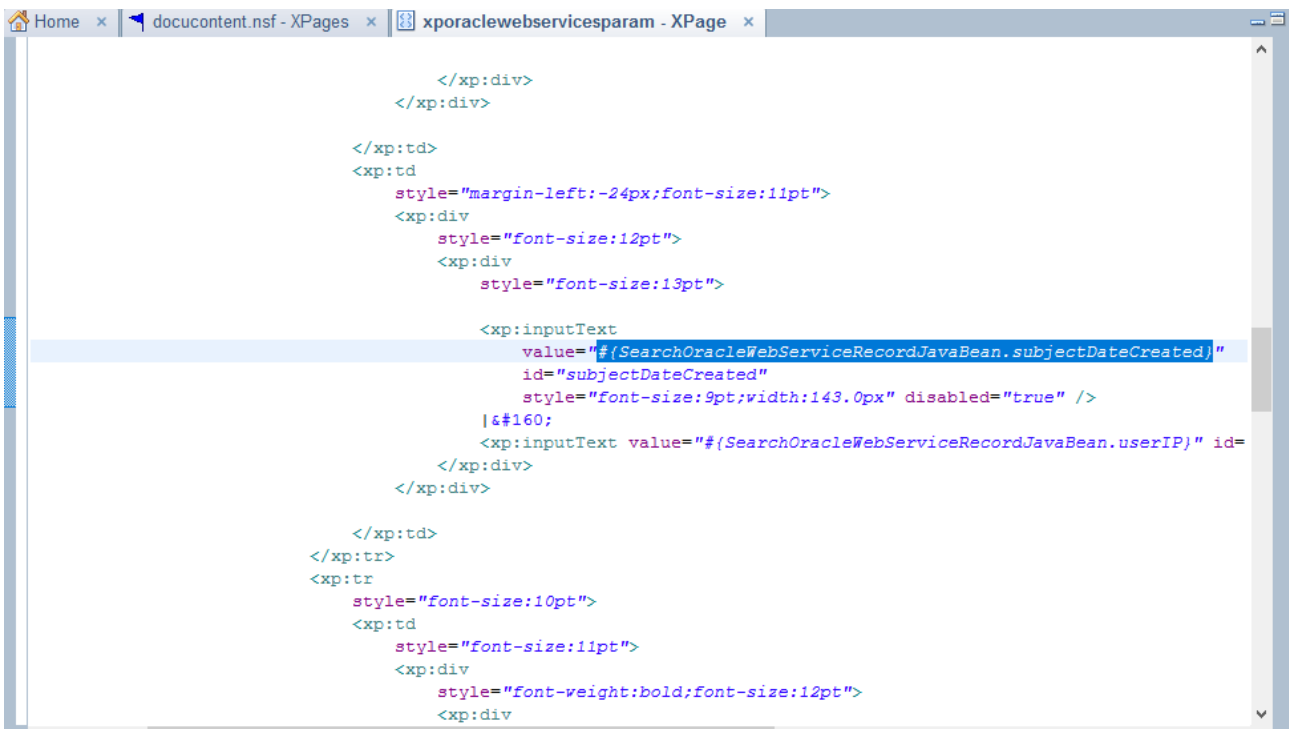


Figure 7

Demonstrating button code placement on the Xpage App, used to query DB2 back-end...



Related Info: You can create a New Xpages application by following below screenshots and code samples. You can name the application, add a package, and attach appropriate technologies

xporaclewebservicessparam.xsp

```
<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core"
  pageTitle="Bulletin Posting Page"
  xmlns:xc="http://www.ibm.com/xsp/custom"
  style="background-color:rgb(180,175,188) ">

  <xp:this.resources>
    <xp:script src="/docucontentGACodeTest.js" clientSide="true" />
  </xp:this.resources>

  <xp:this.navigationRules>
    <xp:navigationRule outcome="xsp-success"
      viewId="/xporaclefeedsroom.xsp" />

    <xp:navigationRule outcome="xsp-failure"
      viewId="/xpgraboraclefeedslist.xsp" />
  </xp:this.navigationRules>

  <xp:tabbedPanel id="tabbedPanell1">
    <xp:tabPanel label="Search Web Service Record" id="tabPanell1">
      <xp:panel
        style="width:459.0px;text-align:center;font-
size:18pt;height:222.0px">
        <p style="text-align:center;font-size:19pt">
          <xp:div style="text-align:center;font-size:20pt">
            <xp:div
              style="text-align:center;font-
size:21pt">
              <b style="font-size:22pt">
                <xp:div
                  style="text-
align:left;font-size:23pt">
                  <xp:div
                    style="text-
align:center;font-size:24pt">
                    <xp:span
                      style="font-
size:20pt;font-weight:bold" />
                    <xp:span style="font-
size:20pt" />
                    <xp:table
```

```

align="left"

style="background-color:rgb(252,243,194);width:83.0%;font-size:9pt">
    <xp:tr

style="font-size:10pt">
    <xp:td

style="font-size:11pt">

<xp:div
style="font-weight:bold;font-size:12pt">
<xp:div
    style="font-size:13pt">

</xp:div>
</xp:div>
    </xp:td>
    <xp:td

style="margin-left:-24px;font-size:11pt">
<xp:div
style="font-size:12pt">
<xp:div
    style="font-size:13pt">

</xp:div>
<xp:div
    style="font-size:13pt">

</xp:div>
<xp:div
    style="font-size:13pt">
    <xp:inputText

value="#{SearchOracleWebServiceRecordJavaBean.subjectUserName}"
    id="subjectUserName" style="font-size:9pt;width:143.0px"
    rendered="true"

defaultValue="#{SearchOracleWebServiceRecordJavaBean.userNameValue}"

```

```

                disabled="true" />
            |&#160;
            <xp:inputText
value="#{SearchOracleWebServiceRecordJavaBean.subjectUserRole}"
                id="subjectUserRole" style="font-size:9pt;width:144.0px"
                rendered="true"
defaultValue="#{SearchOracleWebServiceRecordJavaBean.roleNameValue}"
                disabled="true" />
        </xp:div>
    </xp:div>
                </xp:td>
            </xp:tr>
            <xp:tr
style="font-size:10pt">
                <xp:td
style="font-size:11pt">
            <xp:div
style="font-weight:bold;font-size:12pt">
            <xp:div
                style="font-size:13pt">
            </xp:div>
        </xp:div>
                </xp:td>
            <xp:td
style="margin-left:-24px;font-size:11pt">
            <xp:div
style="font-size:12pt">
            <xp:div
                style="font-size:13pt">
            <xp:inputText

```

```

value="#{SearchOracleWebServiceRecordJavaBean.subjectDateCreated}"
        id="subjectDateCreated"
        style="font-size:9pt;width:143.0px" disabled="true" />
    |&#160;
        <xp:inputText value="#{SearchOracleWebServiceRecordJavaBean.userIP}"
id="inputText1" style="font-size:9pt;width:144.0px" rendered="true"
disabled="true" />
    </xp:div>
</xp:div>
</xp:td>
</xp:tr>
<xp:tr
style="font-size:10pt">
    <xp:td
style="font-size:11pt">
<xp:div
style="font-weight:bold;font-size:12pt">
<xp:div
style="font-size:13pt">
</xp:div>
</xp:div>
<xp:div
style="font-size:12pt" />
</xp:td>
<xp:td
style="margin-left:-24px;font-size:11pt">
<xp:div
style="font-size:12pt">
<xp:div
style="font-size:13pt">
<xp:inputText

```

```

value="#{SearchOracleWebServiceRecordJavaBean.subjectDeptID}
"

        id="subjectDeptID" style="font-size:9pt;width:308.0px" />

</xp:div>
</xp:div>

</xp:td>

</xp:tr>

<xp:tr

style="font-size:10pt">
        <xp:td

style="font-size:11pt">

<xp:div
style="font-weight:bold;font-size:12pt">
<xp:div
        style="font-size:13pt">

</xp:div>
</xp:div>

</xp:td>
<xp:td

style="margin-left:-24px;font-size:11pt">
<xp:div
style="font-size:12pt">
<xp:div
        style="font-size:13pt">

</xp:div>
<xp:div

```



```

        style="font-size:13pt">

        <xp:button
            value="Submit" id="button1" style="font-size:9pt">
            <xp:eventHandler
                event="onclick" submit="true" refreshMode="complete"
                immediate="false" save="true" id="eventHandler1">

                <xp:this.action><!
[CDATA[#{javascript:SearchOracleWebServiceRecordJavaBean.submitEn
try()}]]></xp:this.action>

                </xp:eventHandler>
            </xp:button>

            &#160;

            <xp:messages
                layout="table"
                style="color:red;font-size:9pt;font-weight:bold"
                id="messages1" />

        </xp:div>
    </xp:div>

                                                    </xp:td>
                                                </xp:tr>

                                                    </xp:table>
                                                </xp:div>
                                            <xp:div
                                                style="text-
align:center;font-style:italic;font-size:24pt">

                                                    <xp:span
                                                        style="font-
weight:bold;font-style:italic;font-size:17pt" />

                                                </xp:div>
                                            </xp:div>

                                                    </b>
                                                </xp:div>
                                            </xp:div>

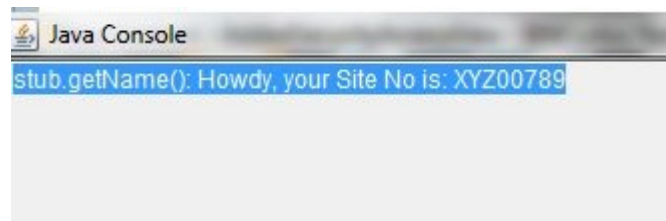
                                                    </p>
                                                </xp:panel>
                                            </xp:tabPanel>
                                        </xp:tabbedPanel>

</xp:view>

```

Figure 8

Xpages form has Run in Browser and button accessed to Query DB2 using IBM Notes Domino Designer Consumer Web Service and Oracle JDeveloper Provider Web Service, to reveal Java Web Console results, based on that single IBM DB2 value...



Conclusion:

You can connect Oracle JDeveloper and IBM Notes Domino to IBM DB2 and run ADF/JSF/Xpages Pages to Browser to Exchange data statically and dynamically using Provider and consumer Web Services.

Tutorials:

IBM DB2 in Oracle Products

https://www.youtube.com/watch?v=7C8mU4ap47Y&index=4&list=PLL0J_OmDhsPSWnwEXFCqkHMNjuo2p-

Oracle SQL Developer

https://www.youtube.com/watch?v=S2TcjulXij4&index=1&list=PLL0J_OmDhsPSWnwEXFCqkHMNjuo2p-UAC

Oracle Jdeveloper

https://www.youtube.com/watch?v=2kBNlaKDWwg&list=PLL0J_OmDhsPSWnwEXFCqkHMNjuo2p-UAC&index=3

Related Info:

Visit prior video tutorials to create connection to IBM DB2 Express-C

IBM DB2 Database

<http://www.dokollolutionsinc.com/CutAndPasteDB2DataLoad.html>

IBM DB2 DataSource

<http://www.dokollolutionsinc.com/CutAndPasteDB2DataSourceLoad.html>

For all Questions and comments, please add a Quick note to our Contact form, or visit our social media networks

Contact

<http://www.dokollolutionsinc.com/apptrendscontactemail.php>

Facebook

<https://www.facebook.com/Dököll-Solutions-Inc-233555900032117/>

Google+

<https://plus.google.com/u/0/+DököllSolutions/posts>

Twitter

<https://twitter.com/DokollSolutions>

YouTube

<https://www.youtube.com/channel/UCSImDTpK0oe7QrPsYOE4nww>

Dököll Solutions, Inc.

www.dokollolutionsinc.com

version:2016.10.03.8.17.PM