

Dököll Solutions, Inc.

Application Development

Project Name:	Web News Döcu Content	Start Date:	2016.10.02.7.55.AM
Purpose:	Instructions/Training	End Date:	2016.10.02.7.55.AM
Software:	DB2, Domino, JDeveloper, WebLogic	Environment:	Windows 7, 8, 10
Employee Name:	Dököll Solutions	Employee ID:	Dököll Solutions
Task(s):	JSF, Xpages, Web Services Samples	Document:	Journal Entries

JSF, Xpages, Web Services Samples Part 1B

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.dokollolutionsinc.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. Please follow these steps to begin: Create New ADF Fusion Application, make new connection to DB2 database, Start/Run Integrated WebLogic Server, Create Java Class, Create Web Service, Test and Inspect Web Service in Browser, Deploy Web Service to Integrated WebLogic Server, Stand-By for IBM Notes Domino Designer App, then run them Side for Handshake, Communication...

Create App from Scratch using Oracle JDeveloper IDE and IBM DB2

At this point we assume Oracle JDeveloper, IBM DB2 are downloaded/installed, a connection to IBM DB2 has been established. It is a good idea to first launch the Integrated Oracle WebLogic Server prior to Deploying applications. Follow instructions and screenshots below to create an application from scratch within minutes using the JDeveloper IDE; areas of interest will be highlighted for your convenience.

You can skip certain sections if already handled...

Steps:

- (1) Go to JDeveloper, Create a New ADF Fusion Application (Figures 1a, 1b)
- (2) Name the application appropriately,
- (3) Add proper package, example com.dokoll.solutions.inc.WebServs
- (4) Choose right ADF technologies, accept defaults for Model & View Controller
- (5) Create a Database connection to Allegation table in DB2 (to be used in Part 2)
- (6) Create a Java Class for Static Web Service (Figures 2a, 2b)
- (7) Convert Java Class into Web Service
- (8) Test and Deploy Web Service (Figures 3a, 3b)
- (9) Copy Web Service WSDL URL Figures 4a - 4h)
- (10) Create IBM Notes Domino Consumer Service using WSDL (Figures 5a, 5d)
- (11) Manage the Consumer Service Properties of Domino Designer
- (12) Create JavaAgent to Consume the new Web Service (Figure 6a)
- (13) Import Web Service Consumer into JavaAgent (Figure 6a)
- (14) Reference WebService Call in JavaAgent code (Figure 6a)
- (15) Run JavaAgent in IBM Notes Client (Figure 6d)
- (16) Look in JavaAgent Console for Resultset (Figure 6d, 6e)
- (17) Make Copies of Both Applications

IMPORTANT: You should have already gone through prior tutorial(s), they are incremental and are helpful if used in proper sequence... You should have already

created a JDeveloper ADF/Faces application... perhaps you have also created and tested the Web Service...

Figure 3b

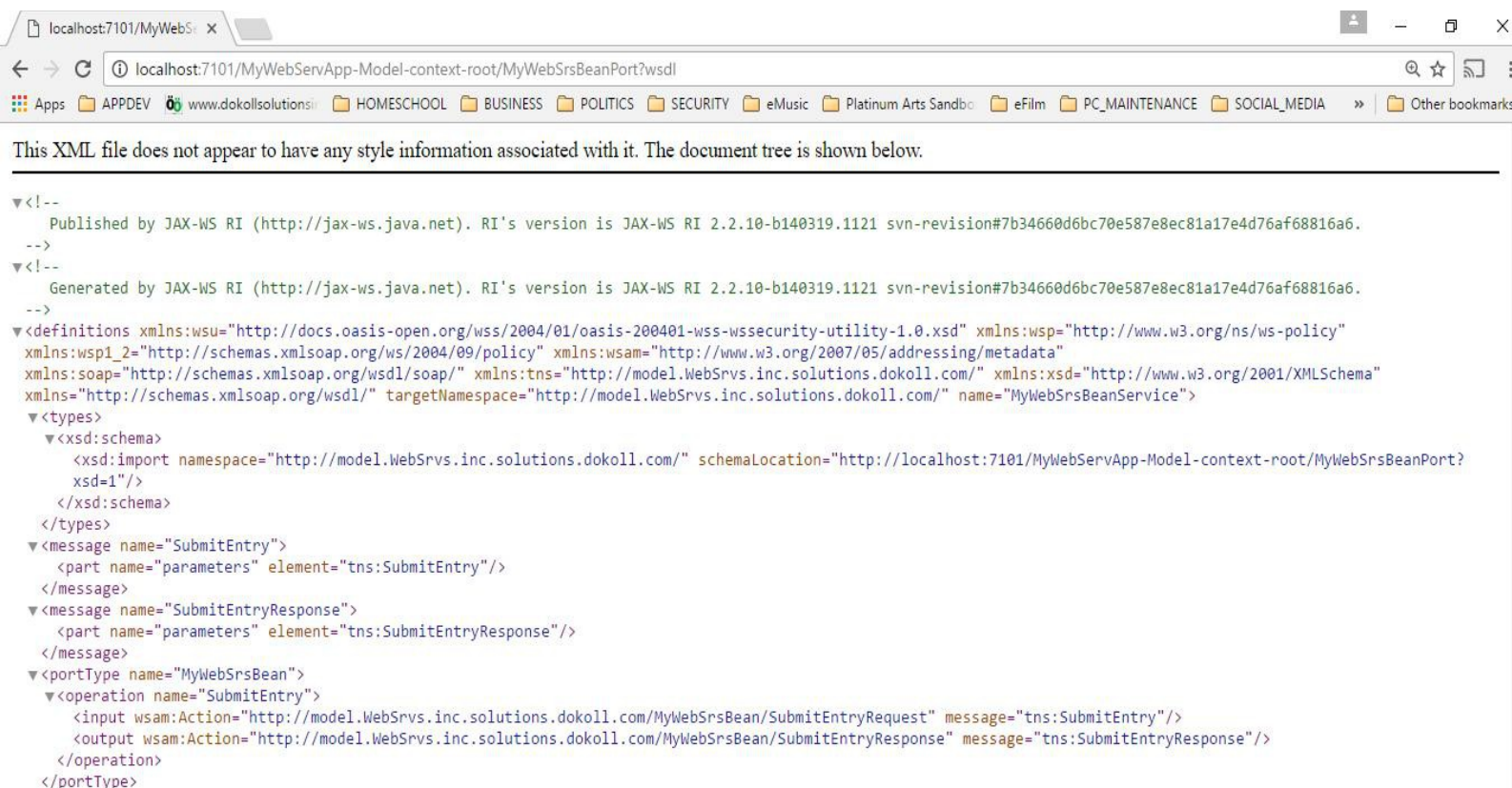
Now we can begin copying the Service URL, if you want you can look at the XML generated...



Endpoint	Information
Service Name: {http://model.WebSrvs.inc.solutions.dokoll.com}MyWebSrsBeanService	Address: http://localhost:7101/MyWebServApp-Model-context-root/MyWebSrsBeanPort
Port Name: {http://model.WebSrvs.inc.solutions.dokoll.com}MyWebSrsBeanPort	WSDL: http://localhost:7101/MyWebServApp-Model-context-root/MyWebSrsBeanPort?wsdl Test
	Implementation class: com.dokoll.solutions.inc.WebSrvs.model.MyWebSrsBean

Figure 4c

Here's our WSDL, taking a good look, inspecting the XML generated... in some cases, this would be a good time to modify address of the Web Service, see below this WSDL for more input...



```
<!--
Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.10-b140319.1121 svn-revision#7b34660d6bc70e587e8ec81a17e4d76af68816a6.
-->
<!--
Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.10-b140319.1121 svn-revision#7b34660d6bc70e587e8ec81a17e4d76af68816a6.
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://model.WebSrvs.inc.solutions.dokoll.com/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://model.WebSrvs.inc.solutions.dokoll.com/" name="MyWebSrsBeanService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://model.WebSrvs.inc.solutions.dokoll.com/" schemaLocation="http://localhost:7101/MyWebServApp-Model-context-root/MyWebSrsBeanPort?
xsd=1"/>
    </xsd:schema>
  </types>
  <message name="SubmitEntry">
    <part name="parameters" element="tns:SubmitEntry"/>
  </message>
  <message name="SubmitEntryResponse">
    <part name="parameters" element="tns:SubmitEntryResponse"/>
  </message>
  <portType name="MyWebSrsBean">
    <operation name="SubmitEntry">
      <input wsam:Action="http://model.WebSrvs.inc.solutions.dokoll.com/MyWebSrsBean/SubmitEntryRequest" message="tns:SubmitEntry"/>
      <output wsam:Action="http://model.WebSrvs.inc.solutions.dokoll.com/MyWebSrsBean/SubmitEntryResponse" message="tns:SubmitEntryResponse"/>
    </operation>
  </portType>
```

Figure 4e

this WebService file was generated from a Java class file

MyWebSrsBeanPort class.forName

```
<!--
Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI
2.2.10-b140319.1121 svn-revision#7b34660d6bc70e587e8ec81a17e4d76af68816a6.
-->
<!--
Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI
2.2.10-b140319.1121 svn-revision#7b34660d6bc70e587e8ec81a17e4d76af68816a6.
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-
policy"xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="htt
p://www.w3.org/2007/05/addressing/metadata"xmlns:soap="http://schemas.xmlsoap.org/
wsdl/soap/" xmlns:tns="http://model.WebSrvs.inc.solutions.dokoll.com/" xmlns:xsd="
http://www.w3.org/2001/XMLSchema"xmlns="http://schemas.xmlsoap.org/wsdl/" targetNa
mespace="http://model.WebSrvs.inc.solutions.dokoll.com/" name="MyWebSrsBeanService
">
<types>
<xsd:schema>
<xsd:import namespace="http://model.WebSrvs.inc.solutions.dokoll.com/"
schemaLocat ion="http://localhost:7101/MyWebServApp-Model-context-
root/MyWebSrsBeanPort? xsd=1"/>
</xsd:schema>
</types>
<message name="SubmitEntry">
<part name="parameters"
element="tns:SubmitEntry"/> </message>
<message name="SubmitEntryResponse">
<part name="parameters"
element="tns:SubmitEntryResponse"/> </message>
<portType name="MyWebSrsBean">
<operation name="SubmitEntry">
<input wsam:Action="http://model.WebSrvs.inc.solutions.dokoll.com/MyWebSrsBean/Sub
mitEntryRequest" message="tns:SubmitEntry"/>
<output wsam:Action="http://model.WebSrvs.inc.solutions.dokoll.com/MyWebSrsBean/Su
bmitEntryResponse" message="tns:SubmitEntryResponse"/>
</operation>
</portType>
<binding name="MyWebSrsBeanPortBinding" type="tns:MyWebSrsBean">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<operation name="SubmitEntry">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="MyWebSrsBeanService">
<port name="MyWebSrsBeanPort" binding="tns:MyWebSrsBeanPortBinding">
<soap:address location="http://localhost:7101/MyWebServApp-Model-
context-root/MyWebSrsBeanPort"/>
</port>
</service>
```

</definitions>

Figure 4f

When the Web Service compiles, it produces the following form to allow a developer to Test the Web Service

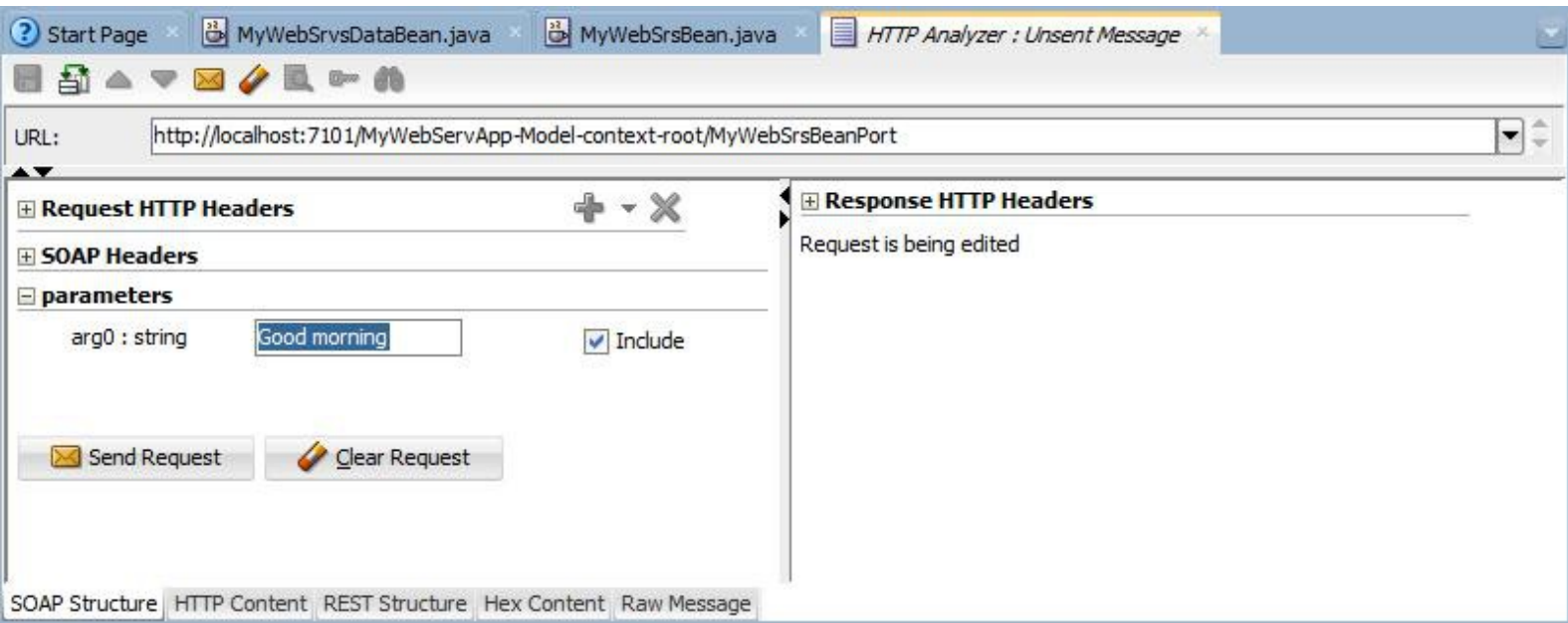


Figure 4g

Provided we have added text to **arg0: string textbox** and submitting using the Send Request button, the result should show when process has completed

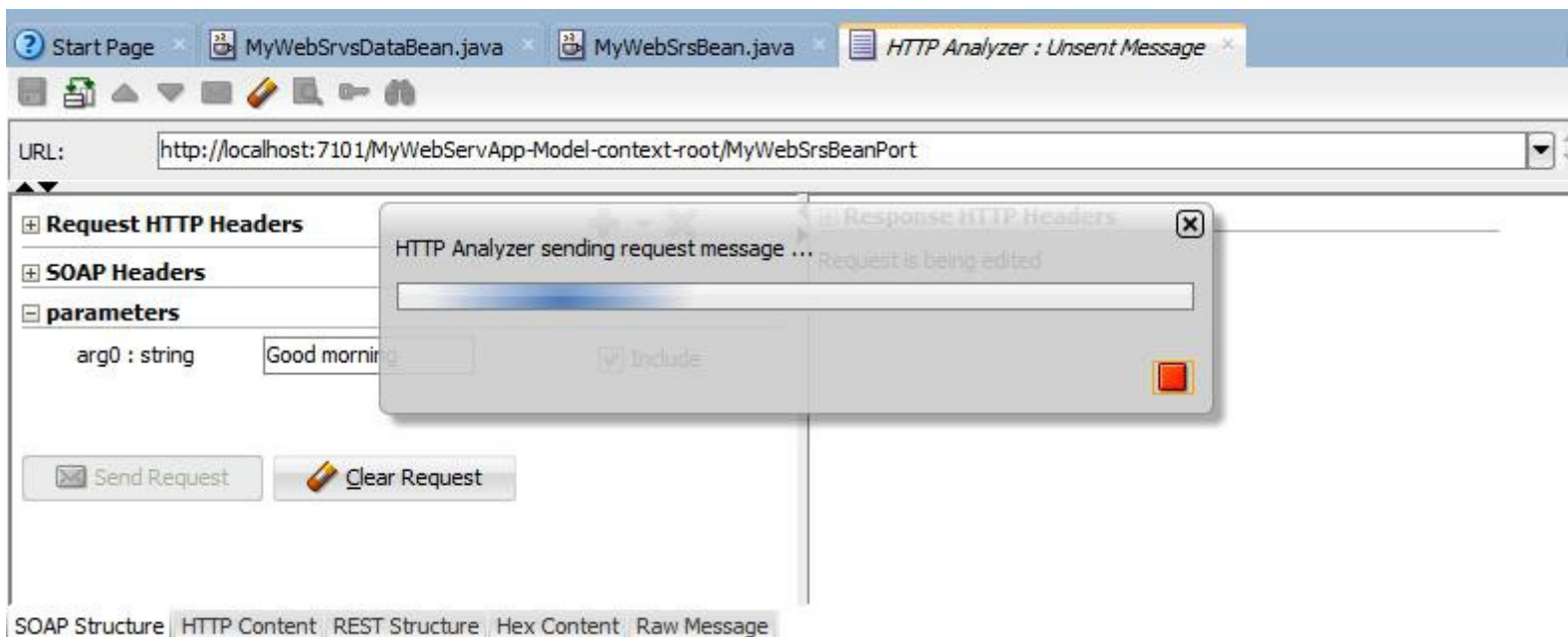
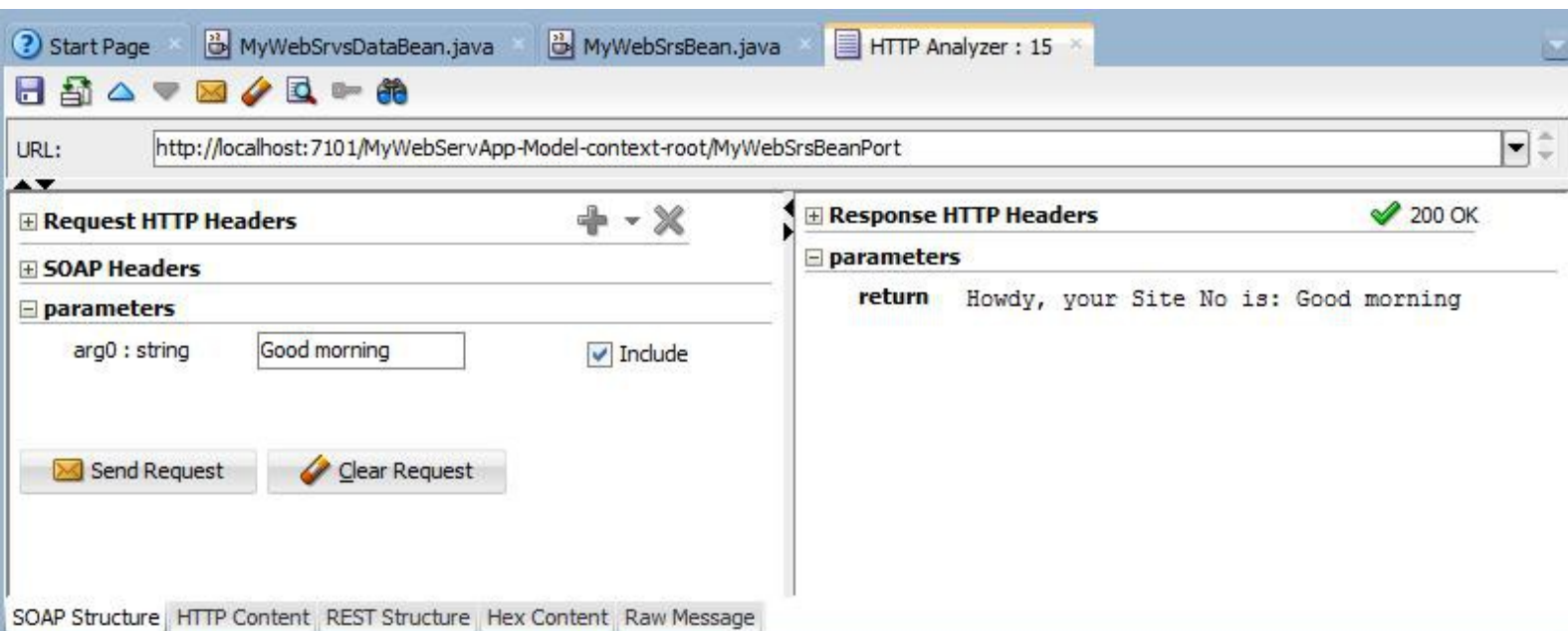


Figure 4h

Response HTTP Headers return successful



Test completed successfully, see next tutorial to complete the whole process...

Conclusion

You are now able to prepare communication between two different IDEs via Web Services, with Oracle JDeveloper as the Provider Web Service, and IBM Notes Domino as the Recipient (Consumer)... Handshake between Oracle JDeveloper environment and IBM Notes Domino Designer environment is successful, Looking ahead to creating the Consumer programs from scratch, starting with a form, view, JavaAgent, JavaBean, Xpages, etc...

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=S2TcjuXij4&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: 2016.10.02.7.55.AM