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 1A

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, connect with IBM Lotus Domino App via Java Web Service to exchange static and database values.

Cheat Sheets:

Try these steps: 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 by side for Handshake, Communication...

IMPORTANT: Maintain Valid Oracle and IBM accounts for your downloads. Also create a JAR file repository to reference for each IDE, they are needed for configuring your environments; Consult our tutorials or respective company websites for instructions.

Steps:

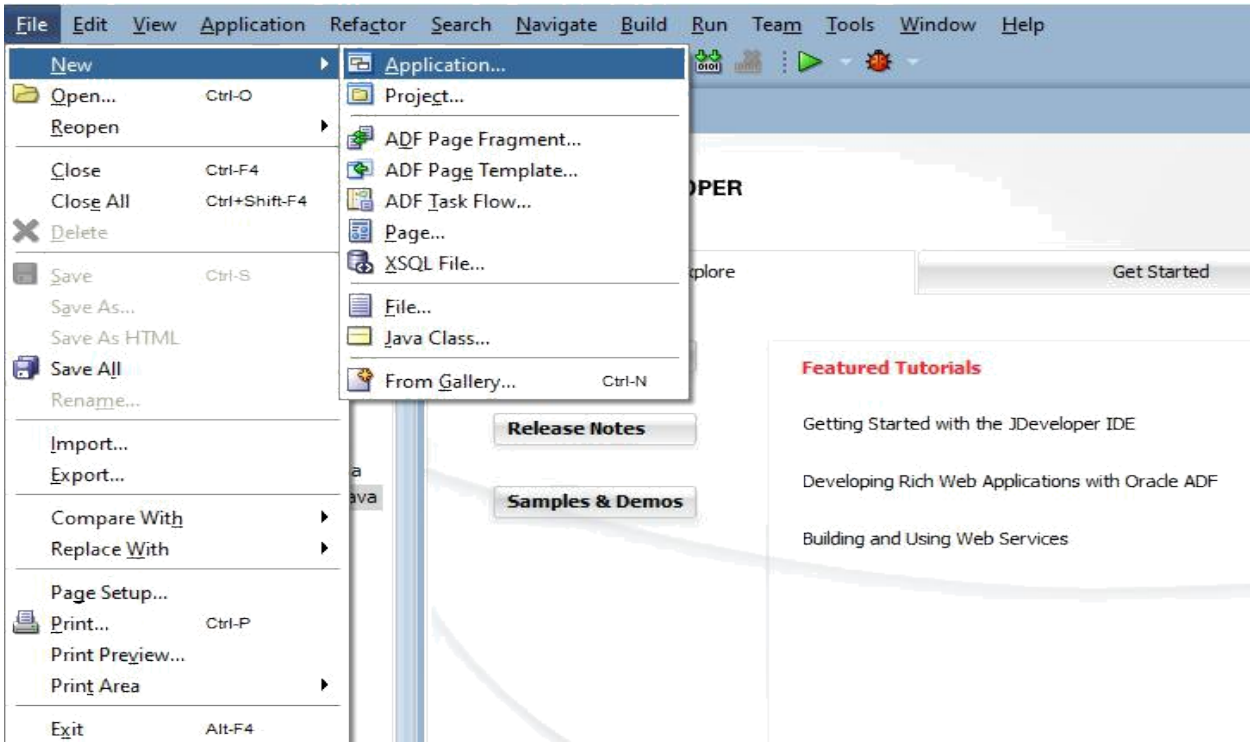
(1) Go to JDeveloper, Create a New ADF Fusion Application (Figures 1a, 1b)
(2) Name the application appropriately,
(3) Add proper package, example com.dokoll.solutions.inc.WebServs
(4) Choose right ADF technologies, accept defaults for Model & View Controller
(5) Create a Database connection to Allegation 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

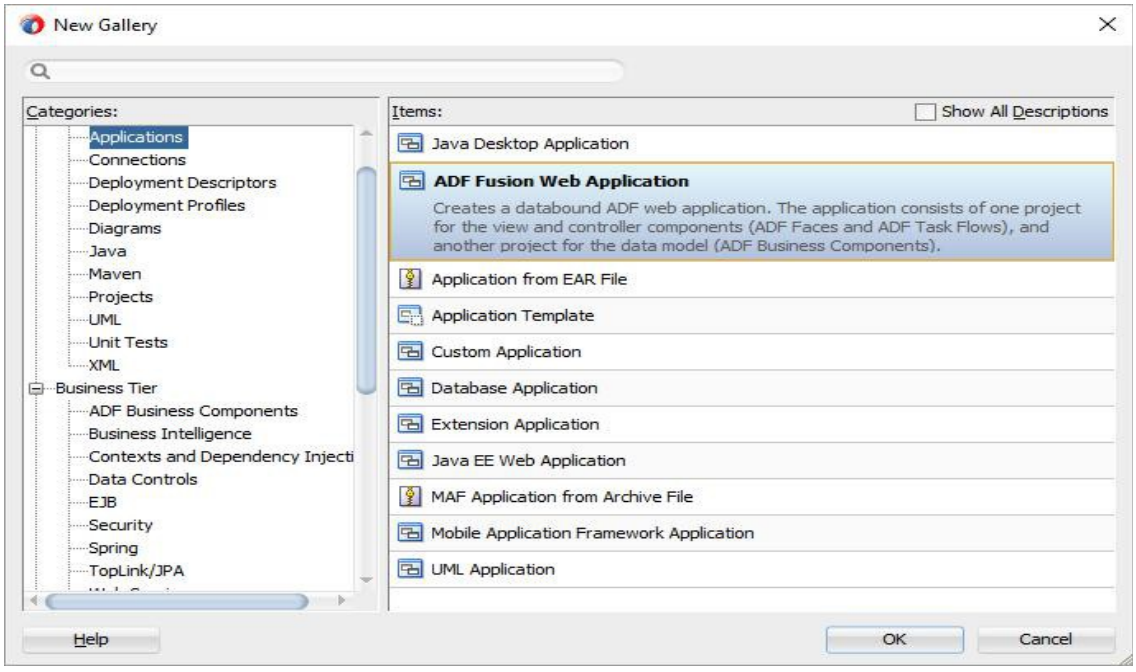
Brief Recap: You can create a New ADF application by following below screenshots. You can name the application, add a package, and attach appropriate technologies as mentioned in **Steps 1- 4**

Steps 1- 4

Figure 1a: Go to JDeveloper, Create a New ADF Fusion Application

Figure 1b: Plug in proper technologies when Creating a New ADF Fusion Application





For **Step 5**, Create a database connection to DB2 (in preparation for Tutorials Part 2)... Once completed continue below with Java class, which will be resued in Part 2 of the tutorials-

Step 5
Now, create a new Java class based on screenshots below, name it appropriately, example MyWebSrsBean

Figure 2a

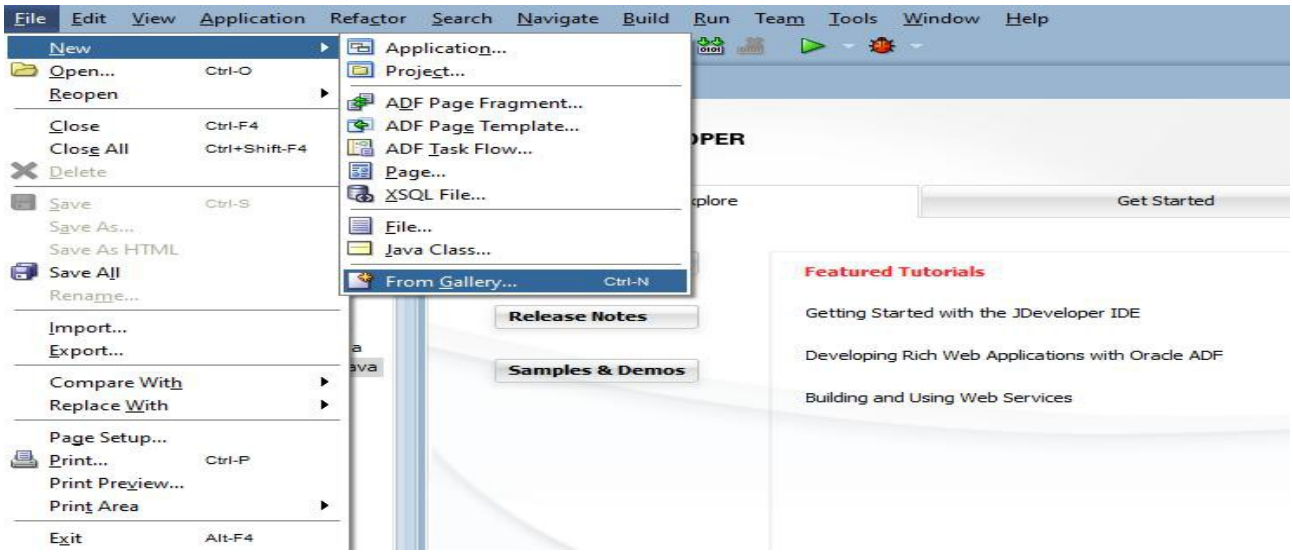
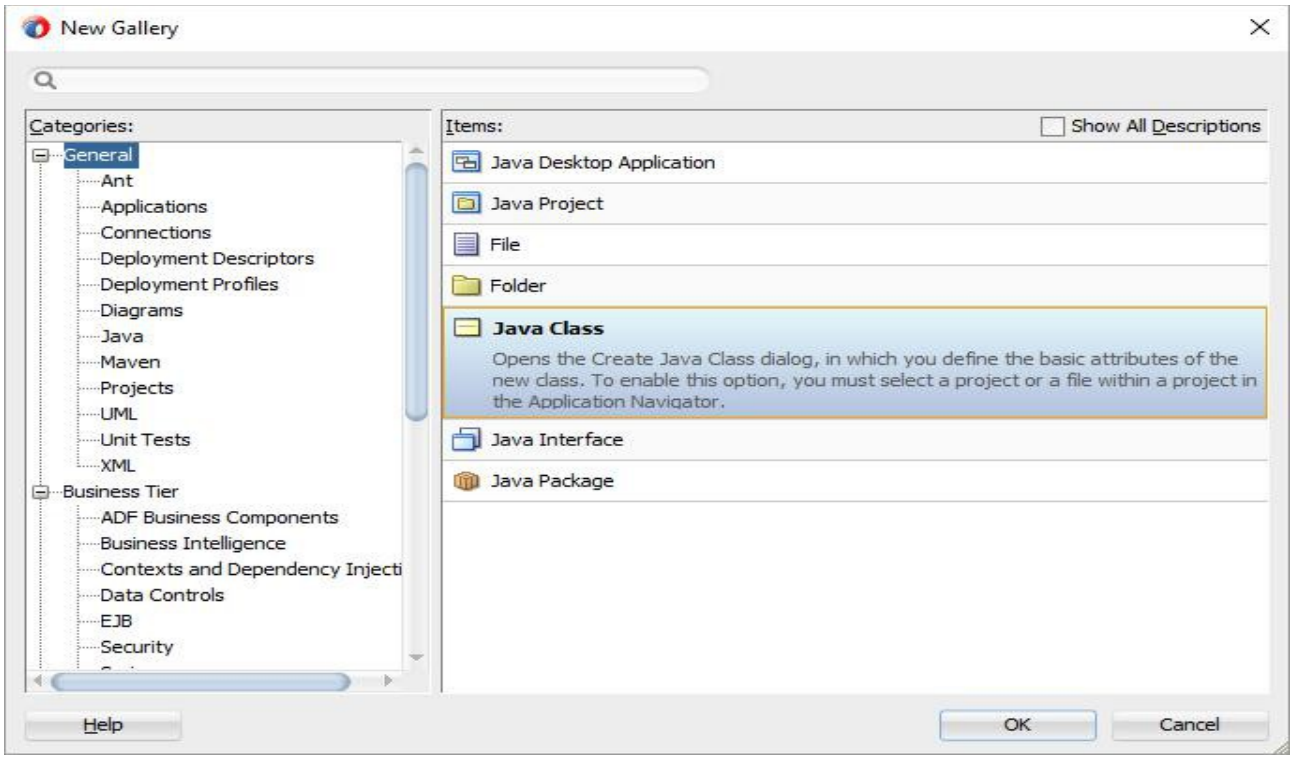


Figure 2b: Pretty basic stuff, just go with the defaults...



Quick look at new Java class
MyWebSrsBean.java

Objective: Once your Java class has been created, you will add a method to call a String variable. See **Before** and **After** code below, complete **Step 6**

Step 6
Your new Java class will generate the following code

BEFORE

++>>

```
public class MyWebSrsBean {

    public MyWebSrsBean() {
        super();
    }
   //...
   //Add your method here...

}
```

++>>

//then you will add your own method

++>>

```
public class MyWebSrsBean {

    public MyWebSrsBean() {
        super();
    }
   //...
   //your method begins here...

    public String SubmitEntry (String inputSiteNo){

        return "Howdy, your Site No is: " + inputSiteNo;
```

```
    }
}
```

++>>

Step 7

Objective: Once you have create own method, you need to convert this Java class into a Web Service program.

You can do this by right-clicking on the actual Java class you created and choose Create Web Service in the menu. The resulting code is below for Step 7

AFTER

++>>

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

import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;

@WebService
public class MyWebSrsBean {

    public MyWebSrsBean() {
        super();
    }

   //...
   //Add your method here...

    @WebMethod
    public String SubmitEntry (@WebParam(name = "arg0") String inputSiteNo){

        return "Howdy, your Site No is: " + inputSiteNo;

    }
}
```

++>>

Step 8

Objective: You now want to Test the Web Service to make sure it compiles okay, and be able to inspect the XML (WSDL) generated and copy URL...

Figure 3a: Right-click on class in Model and select Test Web Service

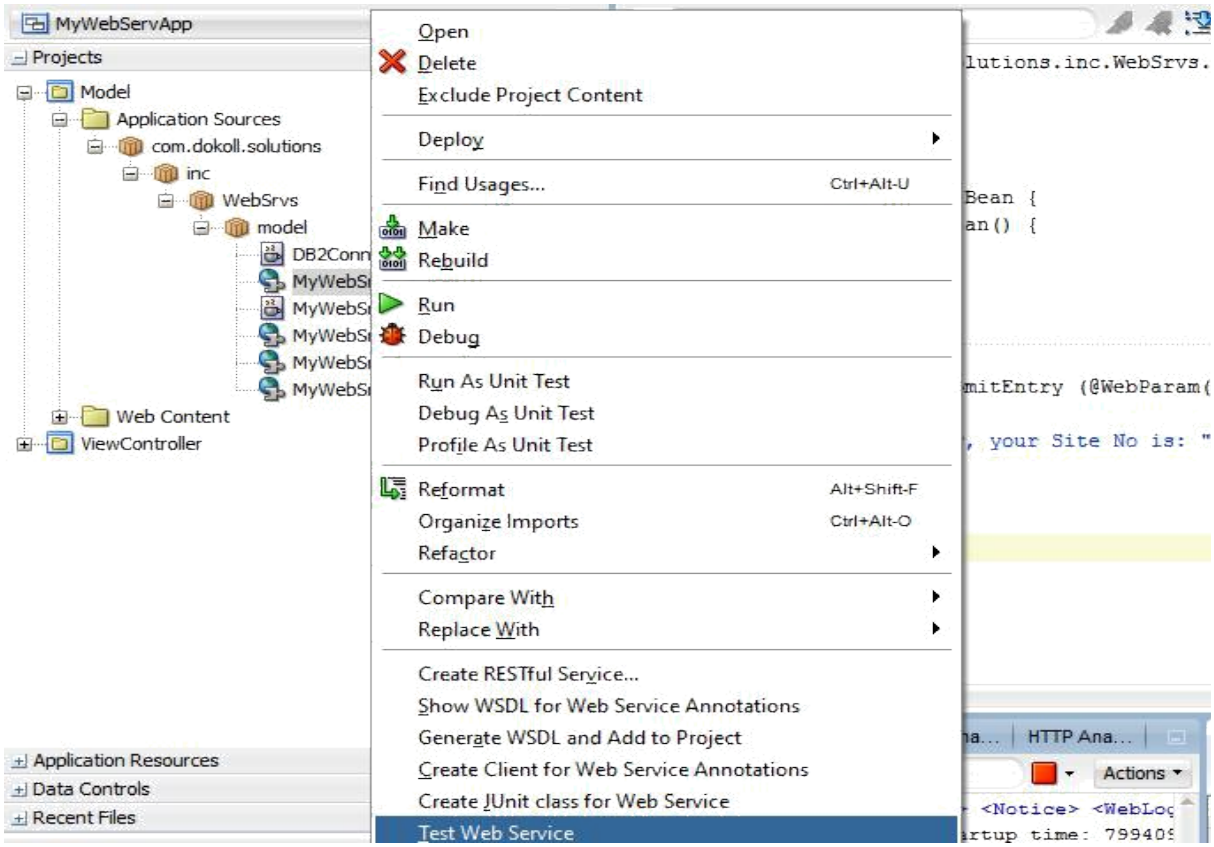


Figure 3b: Copy the Web Service URL to make available in IBM Lotus Notes Domino programs, be sure to also inspect the WSDL, see what's being transmitted, in some cases you would need to modify the URL in the service-

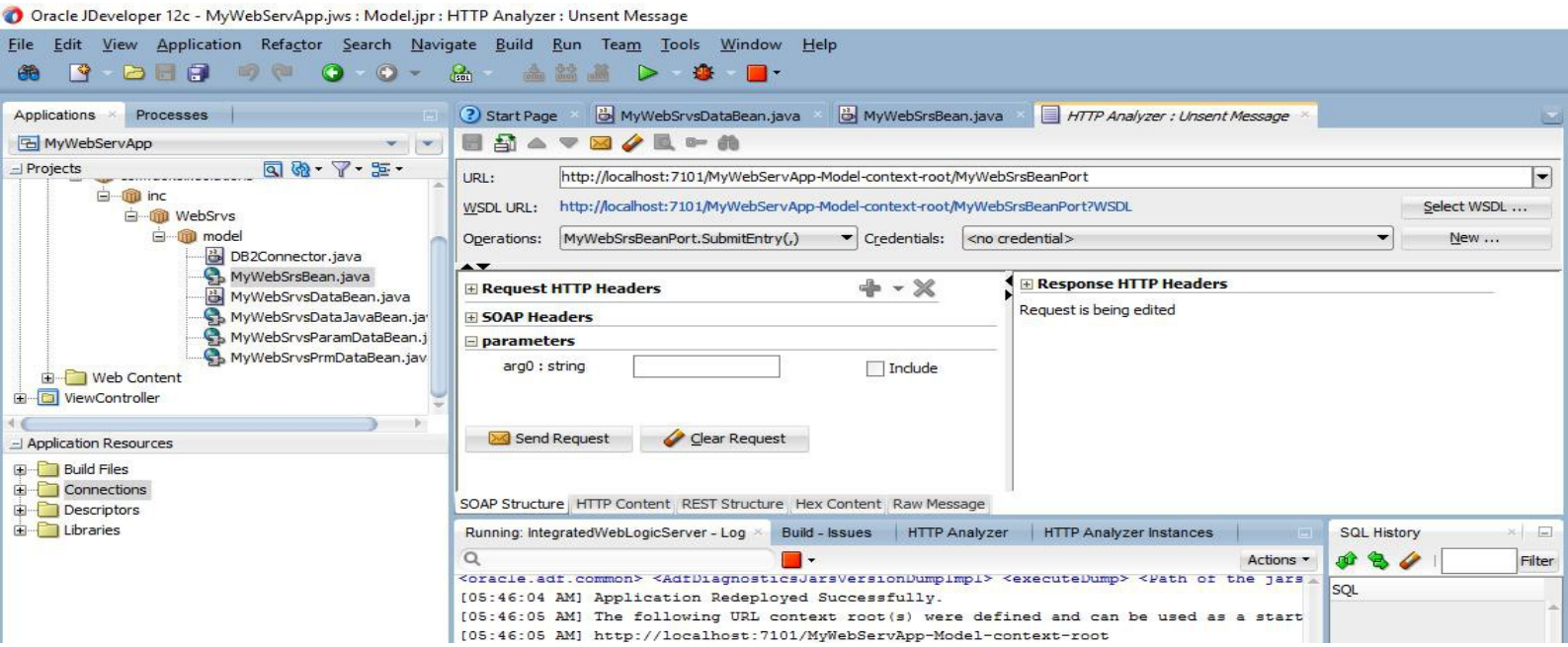
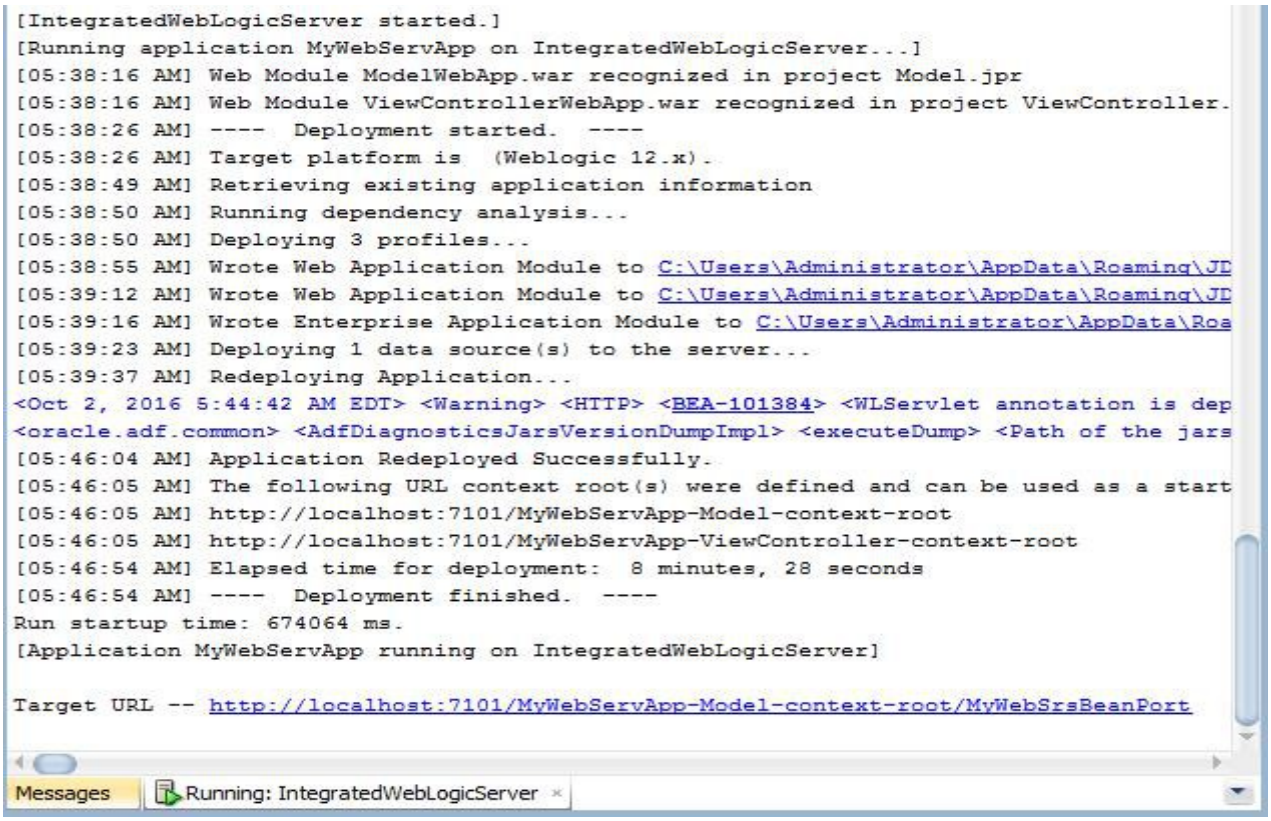


Figure 4a

Taking a quick look at the Console to make sure there are no errors...



Conclusion

You are now able to prepare communication between two different IDEs via Web Services, with Oracle JDeveloper as Provider Web Service, and IBM Notes Domino as Recipient Service (Consumer)... Create and Test successful in Oracle JDeveloper environment- Looking ahead to Sending a Request via HTTP to test returned value-

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=2kBNIaKDWwg&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.dokollsolutionsinc.com/CutAndPasteDB2DataLoad.html

IBM DB2 DataSource
http://www.dokollsolutionsinc.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.dokollsolutionsinc.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.dokollsolutionsinc.com
version:2016.10.02.7.55.AM