

# **IBM Integration Bus**

# **MobileFirst Integration**

Featuring:

- Explore REST API service in Integration Bus Toolkit
- Explore a Mobile application in MobileFirst Studio
- Connect the Mobile application to the REST API service
  - Create an adapter
  - Test Mobile application in Mobile Browser simulator

June 2015 Hands-on lab built at product Version 10.0.0.0

#### Content:

1. INTE	RODUCTION	3
1.1 1.2	LAB PREPARATION OUTLINE OF LAB	3 4
2. PRE	PARE THE PROVIDER APPLICATIONS	5
2.1 2.2	APPLICATION DEPLOYMENT IN IBM INTEGRATION BUS	5 6
3. IBM	MOBILEFIRST INTEGRATION WITH IBM INTEGRATION BUS	. 10
3.1 3.2 3.3 3.4 3.5 3.6	IMPORT THE PROJECT IN THE MOBILEFIRST STUDIO MOBILE APPLICATION FOR QUERYING DATABASE DATA DATA DEFINITION IN 'HOMECONTROLLER.JS' IIB DATA STRUCTURE GENERATE THE MOBILEFIRST ADAPTER ADAPTER IMPLEMENTATION FILE	. 10 . 14 . 18 . 19 . 21 . 26
4. TES	T THE MOBILE APPLICATION	. 29
4.1 4.2	VIEW THE MOBILEFIRST CONSOLE ERROR HANDLING NOTIFICATIONS	. 29 . 36
5. APP	ENDIX	. 39
END OF	LAB GUIDE	39

# 1. Introduction

IBM Integration Bus V10 allows easy integration with the IBM MobileFirst platform. It enables the building of mobile applications for a number of mobile Operating Systems. This integration uses an adapter which provides access to Web and REST services running on IBM Integration Bus.

The IBM MobileFirst platform simplifies mobile application development and utilizes features such as security and administration for mobile applications.

This lab will demonstrate a typical use case for integrating a mobile app with a REST API service running on IBM Integration Bus. The general steps in this use case are outlined below:

- Develop an IBM Integration Bus (IIB) REST API service, for querying data from a database;
- Develop a MobileFirst Application which will invoke the IIB REST API;
- Develop a MobileFirst adapter in the MobileFirst Studio which will connect the Mobile application to the IIB REST API Service.

The focus on this lab is to demonstrate the steps required for creating the MobileFirst adapter and confirming successful integration with the IIB application. Creating the IIB REST API service is the subject of a separate lab.

#### 1.1 Lab preparation

To run this lab you will use IBM MobileFirst Studio. If you are using the pre-supplied VMware image, this will already be available. The image contains an installation of Eclipse Juno v4.2.2 and IBM MobileFirst v6.3.

This lab is based on the solution of the REST API Service. This lab uses an IIB node called RESTNODE. You should have it already created. If not, refer to the Appendix at the end of this lab guide.

Start the RESTNODE.

## 1.2 Outline of Lab

This lab will show you the following functions:

- Explore REST API service in Integration Bus Toolkit
- Deploy REST API service
- Explore a Mobile application in MobileFirst Studio
- Connect the Mobile application to the REST API service
  - Create an adapter
  - Test Mobile application in Mobile Browser simulator
- Explore Error Handling



# 2. Prepare the Provider applications

In this part of the lab exercise, you will import the solutions IIB shared library and REST API Service.

## 2.1 Application Deployment in IBM Integration Bus

1. To avoid naming clashes with earlier labs, this lab will be developed using a new workspace.

If you already have a workspace open, click File, Switch Workspace. Give the new workspace the name "MobileIntegration", or similar.

2. Right-click in the 'Application Development' pane and click on 'Import':

Replication Developmen	t 🖾	🖧 Patterns Ex	plorer		
			Ë	\$	
Application Development				Ne	ew
New Application	N	lew 🕨			
New Integration Service	C	іору			
New REST API	P	aste			
New Library	D	)elete			
	ľγ	love			
	R	lename			
	è I	mport			
		xport			
	R	lefresh			

Import the following Project Interchange (PI) files:

- EmployeeServiceInterface.V10.zip from C:\student10\Integration\_service\solution folder;
- EmployeeService\_REST.V10.zip from C:\student10\REST\_service\solution folder;

Note: Make sure that all projects in these PI files are selected for import.

3. Once completed, you should have in your workspace the Employee REST API Service and a shared library that is referenced by the service.



#### 2.2 Application Deployment

In this section, you will view the IIB REST API service and where in the Mapping node one of the error messages is generated. This will be used later in this lab guide.

1. Expand the REST API Service and double-click on REST API Description:



2. The details of the REST API Service are shown:

		<u>expand all</u> / <u>collapse</u>
/departments		
/departments/{departments/	entNumber}	
/departments/{departments/	entNumber}/employees	
/departments/{departments/	entNumber}/manager	
/employees		
/employees/{employee!	umber}	
/employees/{employees	umber3/department	
,,,,,,,,,		
Error Handling		
	The subflow to which the message is routed if an exception is not handled in an operation subflow	
Implement the Catch handler	The subject to which the message is fource if an exception is not native in an operation subject.	

On the top of the page you will see the REST API base URL as '**/TestWebApp/resources**' which identifies the access path to the service over HTTP.

In addition, you can see a number of operations. Each operation is accessed by adding the specified operation extension to the base URL and adding an input parameter (if required). For example, for our Mobile app we will be using the *lemployees/{employeeNumber}* operation providing their employee number.

This operation has already been implemented in this provided service.

Expand the /employees/{employeeNumber} operation by clicking it.

3. You will see that there are three operations – Get, Put and Delete.

For the GET method, click 'Open the operation':

,,,,	ve the details for arreing	no yee	
Path Parameters	Required	Description	
employeeNumber	Yes		
PUT updateEmployee U	pdates an existing emplo	yee	Open the operation
Path Parameters	Required	Description	
employeeNumber	Yes	The employeeNumber of the employee to be updated	
DELETE deleteEmployee De	letes an existing employ	20	Open the operation
Path Parameters	Required	Description	
employeeNumber	Yes	The employeeNumber of the employee to be deleted	

4. This opens the implementation subflow, which is probably familiar.

Double-click to open the mapping node 'employeeNotFound'.

In the setEmployee.subflow ☆
Flow Exerciser: 🔟 🖼 🕅 🔍 🔍
The convertient of the convertie
Other DB error

5. Once the map has opened, click the second 'Assign' transformation.

You will see the message that will be returned from IIB if the employee is not found in the database. This will be tested in a later section of this lab.

		🗆 🖧 choice	of cast items [0*	]			
		🖳 any	[11	]			
🗎 Assign 👻		දී out_	text [11	string			
•							
Properties 🛛 🔝 Problems	🗄 Outline 🖉 Ta	asks 🔝 Deployment	Log				
Transform - Assign							
General Value: E	mployee not found	l in database					

6. In the IBM Integration Bus Toolkit, start the RESTNODE Integration Node (if not started):

E	des	
🧔 RESTNODE		11
	<b>∆</b> Start	
	🔗 Refresh	
	🖶 Change	
	💢 Delete	

- 7. Deploy your workspace resources by 'drag-and-drop' to the default integration server on RESTNODE in the following order:
  - Shared Library EmployeeServiceInterface.
  - EmployeeService\_REST.

The order of deployment is important because the REST API service references the shared library and the deployment will fail if it does not find the shared library in the runtime when they are deployed.

8. Once the deployment has completed, under the 'default' Integration server, click the deployed REST API Service.

🖧 I 🕱 况 I 🗞 D 🗰 D 🖵 🗖	🔲 Properties 🔀 🔝 Problems 🖶 Outline 🍕	Tasks III Deployment Log
📑 📑	Property	Value
Integration Nodes	- API	
	Base URL for local invocations	http://localhost:7801/TestWebApp/resources
	base ORE for remote invocations	http://192.100.211.220:7001/TestwebApp/resources
Engli Restride	Local URL for the REST API definitions	http://localhost:7801/TestWebApp/resources/EmployeeService.json
	Remote LIRL for the REST API definitions	http://192.168.211.220:7801/TestWebApp/resources/EmployeeService.json
EmployeeService_REST	🖃 Info	
EmployeeServiceInterface	Deployment Time	Fri May 08 14:27:10 BST 2015
	Full Name	EmployeeService_REST.appzip
	Last Modified	Fri May 08 14:21:08 BST 2015
	Keywords	
	BAR	C:/workspaces/IIBWorkshop2015/GeneratedBarFiles/EmployeeService_RESTp
	VERSION	
<b>▼</b>		
Local URL for the REST API definitions		

In the Properties tab you can see details about the deployed REST API Service including URLs for remote and local invocation. In addition, a URL to the REST API definitions is provided.

Write down the port number of the Integration Server, to which the REST API Service has been deployed (7801 in the screen capture above), as it will be needed later in the lab.

REST API port number:

# 3. IBM MobileFirst Integration with IBM Integration Bus

In this part of the lab, you will build an integration adapter in the IBM MobileFirst Studio, which will connect to your application running on IBM Integration Bus.

#### 3.1 Import the Project in the MobileFirst Studio

- 1. Open the MobileFirst Studio (from the Windows Start menu).
- 2. In this workshop the Mobile Application has already been built. The only missing part is an adapter, required to connect to the Integration Node and the IIB REST API Service.

In MobileFirst Studio right-click in the Package Explorer pane and select Import.

🛱 Packag	e Explorer 🔀	E \$	▽ □ □
🗄 🗁 Ma	bileFirst Development Ser	ver	
	New	+	
	Show In	Alt+Shift+W 🕨	
	Сору	Ctrl+C	
	Copy Qualified Name		I
0	Paste	Ctrl+V	I
	Delete	Delete	
Ē	🔄 Import		
Ľ	A Export		ť
q	🦻 Refresh	F5	

3. In the opened dialog, expand 'General' and select 'Existing Projects into Workspace'. Click 'Next'.

🖨 Import				_ 🗆 🗙
Select Create new projects from an archive file or direct	ctory.			è
Select an import source:				
type filter text General Archive File Existing Projects into Workspace File System Preferences CVS CVS EJB Install Doc Java EE				
?	< Back	Next >	Finish	Cancel

4. In the root directory, click 'Browse' and then select 'Employee' folder in **C:\student10\mobile\MobileFirst**.

Click 'OK'.

<b>⊜</b> Import	<u> </u>
Import Projects Select a directory to search for existing Eclipse projects.	
Select root directory:	Browse
Select archive file:	Browse
Select root directory of the projects to import	Select All
Integration_service_MessageFlowSecurity IWA JavaScript_API MessageModeling MessageModelling_IndustryFormats MessageSight MobileFirst MobileFirst Settings Jadapters Jadapters Japps	Select
Folder: Employee	
Make New Folder OK Cancel Finish	Cancel

5. Tick the box next to 'Copy projects into workspace and click 'Finish' to complete the import of the Project.

🚔 Import		
Import Projects Select a directory to searc	h for existing Edipse projects.	
Select root directory:     Select archive file:	C:\student10\mobile\MobileFirst\Employee	Browse Browse
Projects:	dapt10\mahile\MahileEirst\Employee)	Coloct All
Employee (C: )stud	Jent IO (hobile (hobilenii st (employee)	Deselect All
Copy projects into wo	rkspace	
Working sets	ing sets	Select
(?)	< Back Next > Finish	Cancel

6. Once the import has completed you will see the 'Employee' project in the Package Explorer pane.



#### 3.2 Mobile Application for querying database data

The Mobile application that will be used in this lab has already been built. The application is the front end in querying the database for an employee's record. This would be built by a MobileFirst developer, based on the data that needs to be send to the REST API Service and the returned records information.

You will first explore the main components of the project to get familiar with its structure.

1. Expand the Employee project, then 'apps' and 'EmployeeApp':



The directory contains 3 folders – common, iphone and legal:

- common contains the main artefacts for the Mobile app, which are required for any OS
- iphone artefacts for the mobile app deployed on iOS
- legal IBM legal agreements

These folders are created from the MobileFirst Studio when the MobileFirst developer generates the project. Then they are customized by including resources for the Mobile application and its logic (i.e. HTML files, JavaScript files, libraries and images).

2. Expand the 'common' folder, then 'js' and all the subfolders. The JavaScript file 'HomeController.js' is the AngularJS controller, which is used to update the data to the HTML pages. The 'app.js' file contains the logic for the transition between the different views of the mobile app.



3. If you expand the 'lib' subfolder and then 'js', you will see the required AngularJS and Ionic libraries included in the project. These libraries provide the templates for the mobile pages views when the Mobile application is deployed.



- 4. Under the 'views' folder are the Mobile app HTML pages. You will see that there are four HTML pages defined. However, for this lab you will use only two of them:
  - 'search.html' which represents the view to search for the Employee records;
  - 'result.html' which is the view for displaying the returned data from the database.

Double-click 'search.html':



5. You may see a popup message saying 'Some pages may not render correctly when using Internet Explorer as the embedded browser,

Click OK to dismiss the message.

6. In the editor, click the 'Split' tab (may be already on focus).

search.html 🕴	
😳 💷 🗄 💼 📴 🛄 📴 👘 💭 🔤 🔂 Device: Standard 👻	Skin: common 🝷
Employee number [ [{searchErrorMessage}] Get Employee Data	*
	▼  }
▲ ⊖ <ion_view title="Search"></ion_view>	
▲ ⊖ <ion-content padding="true"></ion-content>	
<pre></pre>	
⊖ <div class="list"></div>	-
<pre>     <li><label class="item item-input item-floating-label"> <span< td=""><td></td></span<></label></li></pre>	
<pre>class="input-label"&gt;Employee number <input ng-model="empNo" placeholder="Employee number" type="text"/></pre>	
<pre><div style="color: red; text-align: center">{{searchErrorMessage}}</div></pre>	
<pre></pre>	
⊖ <button <="" class="button button-block button-positive" p=""></button>	
<pre>     ng-click="search(empNo)"&gt;Get Employee Data </pre>	<b>•</b>
Design Courte Calif.	P
Design   Source   Spir.	

The editor is now open and the screen is split – in the upper part you can see the 'design', while in the bottom is the HTML code for this view.

The Mobile application's HTML pages have been created by the MobileFirst developer, based on the requirements from the Integration developer.

**Please note** that the design that is shown in the MobileFirst Studio editor is not the exact presentation of how the page will look on the mobile device. This is because the Mobile app is using the lonic framework, which currently does not render completely in the MobileFirst Studio editor. You will see the full app later in the lab using the Mobile Simulator.

When finished reviewing, close the file without making any changes.

#### 3.3 Data definition in 'HomeController.js'

The 'HomeController.js' contains a JavaScript construction function which is used in Angular.JS to augment an Angula.JS object that refers to the application model. This controller is created by the MobileFirst developer.

In the 'HomeController.js file, you can see the JSON output data structure.

1. Expand the 'common/js/controller folder. Open 'HomeController.js' file

🖻 🔂 apps	
🖻 🎓 EmployeeApp	
📮 🎲 common	
🕀 🗁 css	
🕀 🗁 images	
🛱 🗁 js	
🖻 🗁 controller	
🚯 HomeController.js	

2. In this controller an 'EmployeeError' variable is included to provide you with the error message returned from Integration Bus.

The Error message is generated in the IIB REST API Service if the employee does not exist, or if the employee number requested has more than six numbers.

The IIB error message from IIB REST API service is 'data.out\_text'

3. The EmployeeResponse JSON data structure follows the error handling code:

search.html	B HomeController.js 🔀	
	<pre>console.log("Got data");</pre>	
	<pre>//Test got valid response //if not valid stay on page and produce error message</pre>	
	EmployeeError = data.out_text; \$scope.searchErrorMessage = EmployeeError; \$scope.\$apply();	
	EmployeeResponse = data.EmployeeResponse.EMPLOYEE:	
	<pre>\$scope.name = EmployeeResponse.FIRSTNME + " " + EmployeeResponse.LASTNAME; \$scope.dob = EmployeeResponse.BIRTHDATE; \$scope.salary = EmployeeResponse.SALARY; \$scope.job = EmployeeResponse.JOB; \$scope.biredate = EmployeeResponse.HIREDATE; \$scope.bonus = EmployeeResponse.BONUS; \$scope.searchErrorMessage = ""; \$state.go("result");</pre>	
	<pre>busyIndicator.hide();</pre>	
},		
fu	<pre>inction(error) { //fail</pre>	-
4		Þ

#### 3.4 IIB data structure

1. Back in the IIB Toolkit, the JSON output data structure was generated by the IIB REST API Service. To see this, open the operation 'getEmployee' in the IIB Toolkit:

•	▼ /employees/{employeeNumber}				
	GET getEmployee Retrieve t	he details for an emp	oloyee	Open the operation	
	Path Parameters	Required	Description		
	employeeNumber	Yes			

2. Open the 'convertXMLToJSON' map:

	Be getEmployee.subflow ☆	- 6
1	Flow Exerciser: 🔟 🖗 🔍	
	Input Input extractKeyFromLocalEnv getEmp UserReturnCode = 0 ? B call was successful employeeNotFound employeeNotFound bB call had errors bB call had errors	
J	Other DB error	

3. Expand the output message assembly on the right hand side:

	□ Click to filter>		JSON
-	🛨 📌 Properties	[01]	PropertiesType
	🖃 📌 JSON	[11]	JSONMsgType
	e Padding	[01]	string
	🖃 🖧 choice of cast items	[11]	
	E Data	[11]	anyType
	🖃 🕼 Data	[1., 1]	JSONObject
	🖃 🖧 choice of cast items	[0*]	
	any	[1., 1]	
	🖃 🖳 EmployeeResponse	[1.,1]	EmployeeResponseType
_	🗉 📌 DBResp	[1.,1]	DBRespType
-	EMPLOYEE	[1.,1]	EMPLOYEEType

You will see the JSON object data tree as highlighted above as:

data.EmployeeResponse.EMPLOYEE

Look back in the 'HomeController.js' file to see where it is used.

4. Close the map and the message flow

#### 3.5 Generate the MobileFirst Adapter

In this part of the lab you will generate the adapter, which will enable the Mobile app to connect to the application, running on the Integration Node.

1. In the Mobile First Studio, right-click 'Employee' then 'New' -> 'Other'

🛱 Package Ex	plorer 🛛	□ 🔄 🔻 🗖		
Employ 	New         Go Into         Open in New Window         Open Type Hierarchy         Show In         Copy         Copy Qualified Name         Paste         Delete         Build Path         Source         Refactor         Import         Export		F4 Alt+Shift+W Ctrl+C Ctrl+V Delete Alt+Shift+S Alt+Shift+T	<ul> <li>Java Project</li> <li>Project</li> <li>Package</li> <li>Class</li> <li>Interface</li> <li>Enum</li> <li>Annotation</li> <li>Source Folder</li> <li>Java Working Set</li> <li>Folder</li> <li>File</li> <li>Untitled Text File</li> <li>JUnit Test Case</li> <li>Example</li> </ul>
	Refresh Close Project Close Unrelated Project	c	F5	📑 Other Ctrl+N

2. From the dialog, select 'MobileFirst Adapter' and click 'Next':

🌲 New	
Select a wizard A component that enables connectivity to enterprise applications and other information sources	
Wizards:	
type filter text	
Interface     Java Project     Java Project from Existing Ant Buildfile     MobileFirst Adapter     MobileFirst Application Skin     MobileFirst Environment     Show All Wizards.	×
O Kack Next > Finish	Cancel

3. Name the adapter 'EmployeeAdapter' (case sensitive) and from the drop-down menu for the Adapter Type, select 'HTTP'.

Note: Please make sure that the name of the adapter is 'EmployeeAdapter' because this is the name used in the Mobile app for invoking the adapter.

Click 'Finish':

🌐 New MobileFirst Adapte			<u>_                                    </u>
MobileFirst Adapter Create a new adapter.			
Project name:	Employee		•
Adapter type:	HTTP Adapter		
Adapter name:	EmployeeAdapter		
Create procedures for offli	e JSONStore		
	<u></u>		
Create procedures for USS	) 9 enablement		
?		< Back Next >	Finish Cancel

4. The 'EmployeeAdapter.xml' file is automatically open, wh. In the file you will make changes in order to allow connection to the REST API Service on IIB. You can also add a description if needed.

dapter Editor			
Overview		Details	G
type filter text		Name*:	EmployeeAdapter
Adapter "EmployeeAdapter"	Add Remove	The name of the Server. It can co with a letter. No cannot be modifi	e adapter. This name must be unique within the MobileFirst ontain alphanumeric characters and underscores, and must start te: after an adapter has been defined and deployed, its name fied
🖗 Procedure "getStoriesFiltered"	Up Down	Description:	EmployeeAdapter
		Additional inform MobileFirst Platfo	
		debugPort:	
		platformVersion:	
		protector in the protect	

Expand 'Connectivity'.

5. Expand 'Connectivity' and click 'Connection Policy'.

In the Details pane, you will see the policy's details and parameters.

dapter Editor			
Overview		Details	<b>F</b>
type filter text		Protocol:	http
		Domain*:	rss.cnn.com
🖃 🍟 Adapter "EmployeeAdapter"	Add	Port:	80
	Remove	Connection Timeout (in milliseconds	s): 30000
Procedure "getStories"	Up	The timeout in milliseconds to wait established	until a connection to the back-end can be
Procedure "getStoriesFiltered"		Socket Timeout (in milliseconds):	30000
	Down	The timeout in milliseconds to wait	between two consecutive packets
		SSL Certificate Alias:	
		The alias of the certificate in the s	erver key-store
		Max concurrent connections per no	ode: 50
		The maximum number of concurre back-end application	nt requests that can be performed on the
		Cookie policy:	BEST_MATCH 👻
		Sets how the HTTP adapter handle application	es cookies arriving from the back-end
		Max redirects:	10
		The maximum number of redirects useful when the back-end applicat some error, such as authentication	that the HTTP adapter should follow. This is tion sends circular redirects as a result of n failures. The default value is 10
		eventHandlerInvokerClass:	com.worklight.integration.js.JavaScrip
		procedureInvokerClass:	com.worklight.integration.js.JavaScript
		sslCertificatePassword:	

6. For this adapter, change the 'Domain' and 'Port' as follows. Change the following details as below:

#### Domain: localhost

Port: 7801 (or the one you wrote down in step 2.2.8)

Details		G.
Protocol:	http	_
Domain*:	localhost	
Port:	7801	
Connection Timeout (in milliseconds):	30000	

7. Right-click 'Procedure "getStoriesFiltered" and then 'Remove'. This was a procedure generated by default, which is not needed.

👑 *EmployeeAdapter.xml 🔀		' 🗖
Adapter Editor		
Overview	Details	æ
type filter text	Name*: getStoriesFiltered	
Adapter "EmployeeAdapter"  Add  Connectivity  Connection Policy  Procedure "getStories"  Procedure "getStoriesFiltered"  Remove  Moye Up	The name of the procedure. This name must be unique name within the adapte It can contain alphanumeric characters and underscores, and must start with a letter Display name: Description:	r.
Move Down	□ Auda	<u> </u>
	Defines whether calls to the procedure should be logged in the audit log or not. Refer to the Information Center for the location of the log file. Valid values are - checked: Calls to the procedure will be logged in the audit log - unchecked: (default)	

8. The operation that we use in our REST API Service on IIB is 'getEmployee'. This has to be specified as a procedure name in the Adapter.

Click 'Procedure "getStories" and in the Name field change it to 'getEmployee'.

Save the adapter with 'Ctrl-S' or the 'Save' button in the MobileFirst Studio:

Java - Employee/adapters/EmployeeAdapter/	EmployeeAdapter.xml - Eclipse SDK			
File Edit Navigate Search Project Run Design	Window Help			
📬 📙 🖣 📥 📵 📲 🖻 🗞 🔯 🔸	O • 9₄ •   ₩ @ •   ഈ A •   2 •	₩ - * ↔ ↔	) *   B	
🛱 Package Explorer 💥 📄 🔄 🌄 🗖	💾 *EmployeeAdapter.xml 🔀			- 8
Employee	Adapter Editor			
⊡ → JRE System Library [java]	Overview		Details	B
Handback WebSphere Application Server V8.5 Libert	type filter text	*2	Name*: getEmployee	
⊖ , , , , , , , , , , , , , , , , , , ,	Adapter "EmployeeAdapter"	Add, Remove	The name of the procedure. This name must be unique adapter. It can contain alphanumeric characters and un start with a letter	name within the derscores, and must
i filtered.xsl	🖗 Procedure "getEmployee"	Up	Display name:	
□		Down	Description:	
images				<b>V</b>
Controller Controller Controller Controller Controller Controller Controller			Defines whether calls to the procedure should be logger not. Refer to the Information Center for the location of values are:	d in the audit log or the log file. Valid

#### 3.6 Adapter Implementation File

1. You will notice in the Package Explorer that there is an error in the 'EmployeeAdapter' section (red cross). This is because we have not made any changes in the adapter implementation JavaScript file to include the procedure 'getEmployee'.

Double-click 'EmployeeAdapter-impl.js' file



2. In the editor, you will notice a large 'comment' section at the beginning describing how to edit the JavaScript code to access your remote applications. Following this are the functions that include the connection details (currently the ones generated by default when the adapter was created).

4	EmployeeAdapter.xml 🚯 EmployeeAdapter-impl.js 🕸		E
	⊖/*		
	<ul> <li>* Licensed Materials - Property of IBM</li> <li>* 5725-I43 (C) Copyright IBM Corp. 2011, 2013. All Rights Reserved.</li> <li>* US Government Users Restricted Rights - Use, duplication or</li> <li>* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.</li> <li>*/</li> </ul>		
	<pre>     /**     * WL.Server.invokeHttp(parameters) accepts the following json object as an argument:     *     *     *     {        </pre>		
	<pre>* // Mandatory * method : 'get' , 'post', 'delete' , 'put' or 'head' * path: value, *</pre>		
	<pre>* // Optional * returnedContentType: any known mime-type or one of "json", "css", "csy", "plain", "xml * returnedContentEncoding : 'encoding', * parameters: {name1: value1, }, * headers: {name1: value1, }, * cookies: {name1: value1, },</pre>	",	
	<pre>* body: {  * contentType: 'text/xml; charset=utf-8' or similar value,  * content: stringValue  * },  * transformation: {  * type: 'default', or 'xslFile',  * xslFile: fileName  * }  * }  */</pre>		
	<pre> @ /** * @param interest * must be one of the following: world, africa, sport, technology, * (The list can be found in http://edition.cnn.com/services/rss/) * @returns json list of items */  function getStories(interest) {     path = getPath(interest); </pre>		
	var input = {	Þ	•

3. Scroll all the way to the bottom of the file (you can expand the view of the file to full screen by double clicking on its name tab).

You will now edit the file to define the connection to the IIB REST API Service.

First, delete the functions 'function getStoriesFiltered(interest)' and 'function getPath(interest)':



4. On the last remaining 'function getStories(interest)' you need to do the following changes:

To ensure that changes are made correctly, we have provided this code for you.

Copy/paste the contents of C:\student10\mobile\MobileFirst\EmployeeAdapter\_REST.txt, replacing the function getStories(interest).



5. Once you have completed the changes your 'EmployeeAdapter-impl.js' file (excluding the 'comment' part) should look like below.



- 6. Save the changes with 'Ctrl-S' and close the file.
- 7. You will see that now the error in the 'adapters' section has disappeared:



# 4. Test the Mobile Application

#### 4.1 View the MobileFirst Console

1. In Package Explorer, right-click the EmployeeAdapter and select 'Run As' --> 'Deploy MobileFirst Adapter.

💾 Dackage Explorer 😒			
Employee			
server/java			
🕀 🛋 WL Server Library			
🕀 🛋 JRE System Library	[java]		
🕀 🛋 WebSphere Applica	tion Server V8.5 Liberty Profile	[MobileFirst Deve	lopment Server]
🕀 🔂 adapters			
🖻 😓 EmployeeAdapt	New		•
🔒 EmployeeA	Go Into		
🕅 filtered.xsl	Open in New Window		
🕀 🔂 🔁 apps	Show In	Alt+Shift+W	•
🗄 🗁 bin		<b>a</b> . 1. <b>a</b>	
components	Сору	Ctrl+C	
🗄 🗁 externalServerLibra	👔 🛅 Copy Qualified Name		
🗄 🔂 server	📋 Paste	Ctrl+V	
····· 🔁 services	💢 Delete	Delete	
🗄 🗁 MobileFirst Developmen			-
	Build Path		
	Refactor	Alt+Shift+T	
	ave Import		
	Export		
	🚕 Refresh	F5	
	Assian Working Sets		
			-
	Validate		
	Debug As		•
	Profile As		
	Run As		🕨 📛 1 Deploy MobileFirst Adapter 🛛 🛛 Ctrl + Alt + X, D
	Team		2 Invoke MobileFirst Back-end Service
	Compare With		

Please note that this may take a minute or two

2. Once the adapter deployment has completed, you will see the confirmation in the console.

	-
[2015-05-15	14.25.381
[2013 03 13	14.20.00]
[2015-05-15	14.25.411
[2013 03 13	14.22.41]

Server port: 10080 Adapter build and deploy finished.

**Please note** that the you may be shown the Mobile First Server console. To switch to Mobile First console, click on the arrow next to the little 'screen' and make a selection as shown below





3. Right-click the Employee project and from the menu click on 'Open MobileFirst Console'.

This will initiate the start of the server, compile the Employee project, and will deploy the EmployeeApp to the server.

This will take a few minutes. Do not be concerned if you see a few warnings or error messages in the Console.

4. Once deployment has completed, a Browser will open automatically with the log in page for IBM MobileFirst console.

**Please note** that the Firefox Browser (the default Browser) will not be able to render the mobile pages view, so it is recommended that you use Chrome (copy the URL and open in Chrome).

Enter admin/admin as User/ Password.

BM MobileFirst Platform Op ×			_ <b>—</b> X
← → C 🗋 localhost:10080/work	ightconsole/login.html#	Employee,catalog	☆ 〓
Apps   ODM Business Console			
WebSphere, software	IBM M Operat User: admin Password:  Log in Usensed Materials - Property of IB Reserved. IBM, the IBM logo, ibm International Business Machines O Other product and service names current list of IBM trademarks is a information.	ObileFirst Platform tions Console	A

5. In the console you will see the deployed 'EmployeeApp'. You are able to view the properties of the app and preview in a Mobile Browser Simulator.

In the bottom half of the view you will see the adapter that you created for connecting the Mobile application to IBM Integration Bus

Click the 'iPhone' icon as shown below:

😚 IBM M	IobileFirst Platform Op		
← →	C 192.168.211.224:10080/worklightconsole/index	.html#Employee,catalog	Q ¶ ☆ ≡
	IBM MobileFirst Platform Operations Console	Welcome, admin   Logout   About	^
	Catalog Devices Push Notifications Log Profiles Deploy application or adapter: Choose file No file chosen Submit	Analytics Dashboard >	
	EmployeeApp EmployeeApp Last deployed at: 5/14/2015 3:13 PM	× Delete	
	Version 1.0  Active Act	urity Test: Default Authentication: Disabled loe Authentication: Default - Authentication: Default d Time: S/14/2015 3:13 PM vious Build Time: No value	
	Preview as Common Resources		,
	Last deployed at: 5/15/2015 2:25 PM Show details 🗸	X Deele	
	License tracking   Audit log   Ellior log	@ Copyright IBM Corp. 2006, 2014. All Rights Reserved.	

The Mobile Browser simulator is opened in a new tab with a preview of the Mobile application on iPhone4.
 Ignore the message for 'unsupported java plug-in-.

😚 IBM MobileFirst Platform Op 🗙 🎦 Mobile Browser Simulator 🗙			X
← → C	x.html?webpage=	=/Employee/apps/services/preview/EmployeeApp/iphone/1.0/&platform=ios.iphone s	2 =
Apps      DD <sup>^</sup> Business Console	101		
Java(TM) is required to display some elements on this page. Institution of the page.	stall plug-in	Problems installing	<u>?</u> ×
Mobile Browser Simulator The Mobile Browser Simulator displays mobile web pages in a variety of mobile bro	owser sizes and shapes	25.	Î
Webpage: /Employee/apps/services/preview/EmployeeApp/iphone/1.0	Go / Refresh	Add Device 🔻   Scale All Devices: 100% 🖵   📄 Enable Useragent Switching 💟 Simulate Devic	e API
Cordova			
Device			
► Events			
Accelerometer			
▶ Battery		Search	
► Camera			
▶ Capture			
▶ Compass			
► Contacts		Employee number	
▶ File			
Geolocation			
Network		Get Employee Data	
► Scenario			
		Rota Marke	
		peravvarks	
		Apple iPhone 4	
		Арріе іг попе 4	-

7. You are now presented with the first screen of the Mobile App and you will need to enter an 'Employee Number' in order to receive the corresponding data from the database, accessed from the REST API Service running on IIB.

Enter 000010 as an Employee Number in the field and press (click) 'Get Employee Data':

	Search	
E	Employee number	
	Get Employee Data	
ß	etaW <mark>arks</mark>	
J		

8. The employee data has been returned from the database (probably already familiar to you):



9. In the Mobile Browser Simulator, click the 'Search' button to return to the mobile view to search for another employee:



## 4.2 Error handling notifications

1. Enter '000011' as an Employee Number and press 'Get Employee Data' button:



2. You will see the message returned from IIB 'Employee not found in database' in red colour font:



3. Enter a new seven digit number to search in the database – '0000111' and press 'Get Employee data':



4. This time, you will see a different message, advising you that the input number should be 6 characters in length:



5. Feel free to use another test, by entering Employee number 000020 or 000050:



6. You will see again the returned data in the mobile application 'Result' page:

-Search	Result
Empl	oyee Information
Number	000020
Name	MICHAEL THOMPSON
Date of birth	1978-02-02
Phone	3476
Salary	94250
Job	MANAGER
Bonus	800
<b>βeta</b> ₩	Varks

## 5. Appendix

To create the RESTNODE and configure it, we have provided a script. In an IIB Command Console, change directory to c:\student10\REST\_service\install.

Run the script file:

#### Create\_RESTNODE

Accept the default values for the IIB node name (RESTNODE).

This script will create the new node, and run two key commands:

• Enable HTTP embedded listeners. The REST support is only provided for embedded listeners, not the node-wide listener:

```
mqsichangeproperties RESTNODE
-e default
-o ExecutionGroup
-n httpNodesUseEmbeddedListener -v true
```

 Enable Cross-Origin Resource Scripting for REST. This is required when testing with the SwaggerUI test tool. See <u>http://en.wikipedia.org/wiki/Cross-origin resource sharing</u> for further information.

mqsichangeproperties RESTNODE -e default

- -o HTTPConnector
- -n corsEnabled -v true

The script will also configure the JDBC parameters for connection to the SAMPLE database, which will be used in this scenario.

## END OF LAB GUIDE