16L14

IBM Integration Bus

MobileFirst Integration using local connections and API Connect

Featuring:

- Mobile application in MobileFirst Studio
- Connect the Mobile application to the IIB REST API
  - Explore MobileFirst adapter for local connectivity
  - Test Mobile application in Mobile Browser simulator
- Creating additional adapters for API Connect
  - IBM Secure Gateway
  - API Connect

November 2016

Hands-on lab built at product
Version 10.0.0.6

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 HR_Service. This lab uses an IIB node called TESTNODE_iibuser. To see how this REST API was developed, please refer to the REST API HR_Service Lab Guide document (16L02_IIB10006).

1.2 Outline of Lab

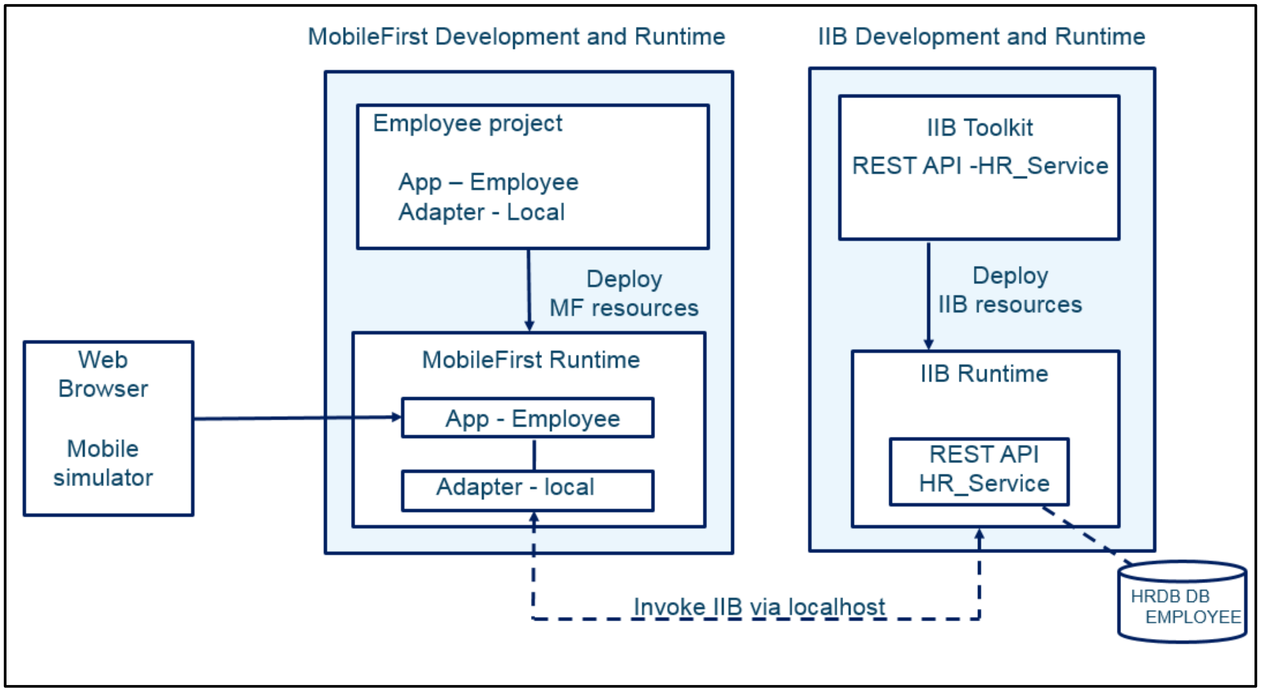
This lab will show you the following functions:

- 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
  - Explore the provided adapter
  - Test Mobile application locally in Mobile Browser simulator
- Explore additional adapters for Secure Gateway and API Connect
  - Push the deployed REST API from IIB into API Connect

1.2.1 Schematic – local invocation of IIB REST API

In the first scenario, the MobileFirst component makes a local connection to the required IIB REST API. Both systems are located on the same Windows system, and connection is direct, using localhost.

The MobileFirst project contains an adapter that provides this direct connection.



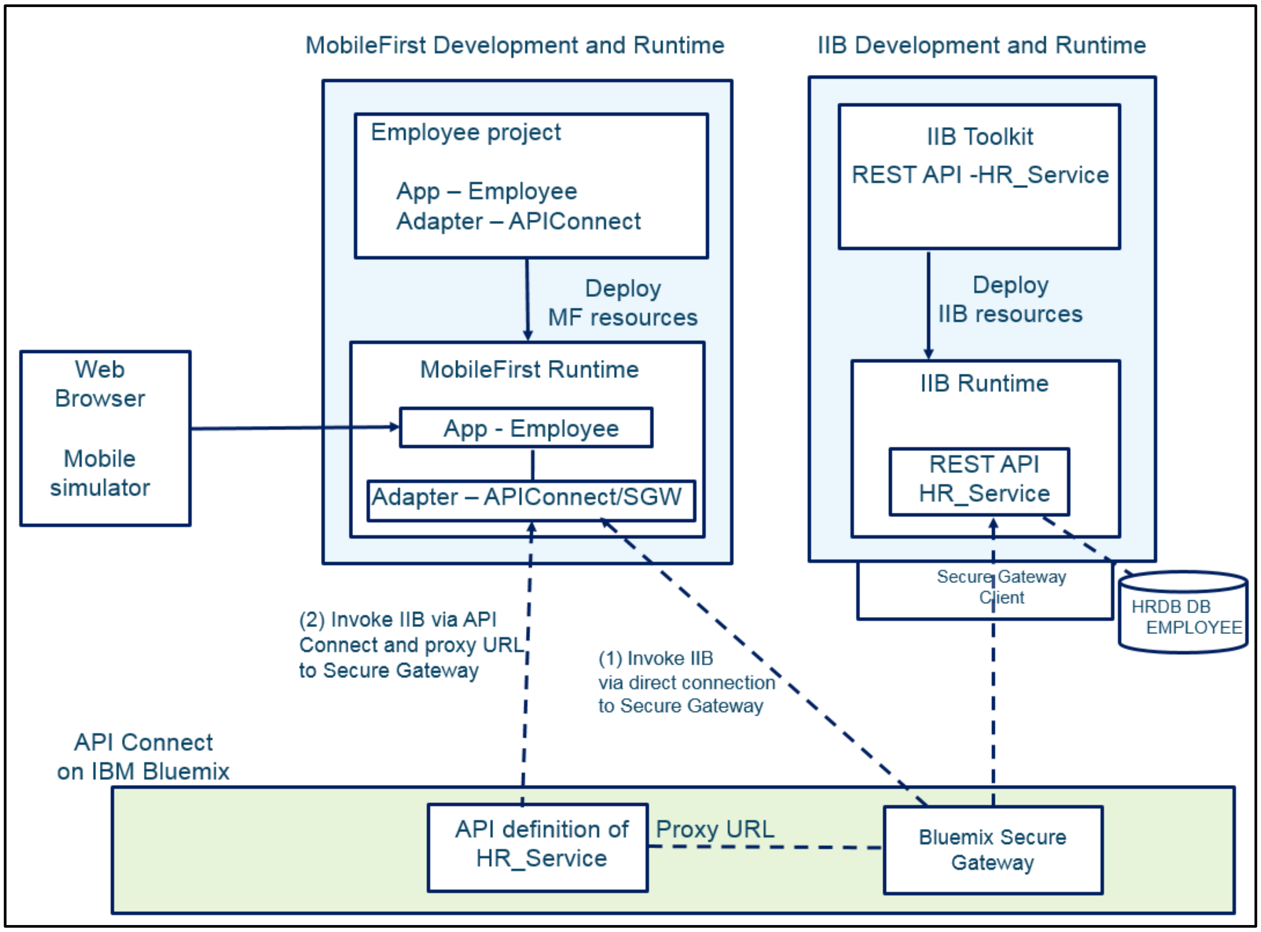
1.2.2 Schematic – invocation of IIB REST API using API Connect

In the second scenario **(Chapter 5)**, the MobileFirst app invokes the IIB REST API by directing the request via IBM Bluemix.

In the first step (shown as (1) below), the MobileFirst app will use a MobileFirst Adapter which will use a direct connection to the Secure Gateway Server on Bluemix.

The second step (shown as (2) below) will extend this by using a MobileFirst adapter that will direct the request to the API Connect service on Bluemix. The definition of the REST API on Bluemix contains a proxy URL for the target endpoint. This proxy URL directs the request to the connected Secure Gateway Server and Client, and then to the required IIB server.

As part of the second step, you will use IIB tools that will push the API definition of the deployed REST API, HR_Service, into the API Connect service on Bluemix.



1.3 Configure TESTNODE_iibuser for REST APIs

The instructions in this lab guide are based on a Windows implementation, with a user named “iibuser”.
The Windows VMWare image on which this lab is based is not available outside IBM, so you will need to provide your own software product installations where necessary.

Login to Windows as the user "iibuser", password = "passw0rd". (You may already be logged in).

Start the IIB Toolkit from the Start menu.

The IIB support for the REST API requires some special configuration for the IIB node and server. Cross-Origin Resource Scripting (CORS) must be enabled for the IIB node to execute REST applications. This is also required when testing with the SwaggerUI test tool. See http://www.w3.org/TR/cors/?cm_mc_uid=09173639950214518562833&cm_mc_sid_50200000=1452177651 for further information.

1. Ensure that TESTNODE_iibuser is started.

2. Check that CORS has been enabled on the IIB node by running the following command in an Integration Console:

```
mqsireportproperties TESTNODE_iibuser
     -e default
     -o HTTPConnector
     -r
```

3. If CORS is enabled, you will see the following lines (amongst others):

```
corsEnabled='true'
 corsAllowOrigins='*'
 corsAllowCredentials='false'
 corsExposeHeaders='Content-Type'
 corsMaxAge='-1'
 corsAllowMethods='GET,HEAD,POST,PUT,PATCH,DELETE,OPTIONS'
 corsAllowHeaders='Accept,Accept-Language,Content-Language,Content-
   Type'
```

4. If CORS has not been enabled, run the following commands:

```
mqsichangeproperties TESTNODE_iibuser
     -e default
     -o HTTPConnector
     -n corsEnabled -v true

mqsistop TESTNODE_iibuser

mqsistart TESTNODE_iibuser
```

1.4 Configure Integration Bus node to work with DB2

If you have already done a previous lab involving the HRDB database in this series of lab guides, you can skip to the next heading.

To run this lab, the Integration Bus node must be enabled to allow a JDBC connection to the HRDB database.

1. Open an IIB Integration Console (from the Start menu), and navigate to

```
c:\student10\Create_HR_database
```

2. Run the command

```
3_Create_JDBC_for_HRDB
```

   Accept the defaults presented in the script. This will create the required JDBC configurable service for the HRDB database.

3. Run the command

```
4_Create_HRDB_SecurityID
```

4. Stop and restart the node to enable the above definitions to be activated

```
mqsistop TESTNODE_iibuser

mqsistart TESTNODE_iibuser
```

This will create the necessary security credentials enabling TESTNODE_iibuser to connect to the database.

To recreate the HRDB database, skip to the Appendix.

2. Prepare the IIB application

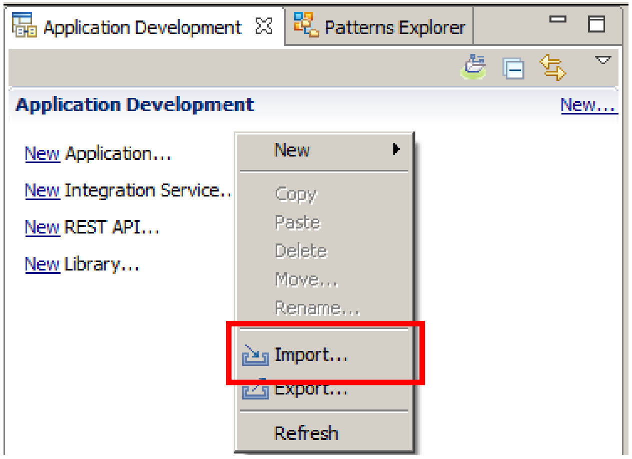
In this part of the lab, you will import and deploy the provided IIB shared library and REST API.

2.1 Import the IIB REST API

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

   In the Integration Toolkit, 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':

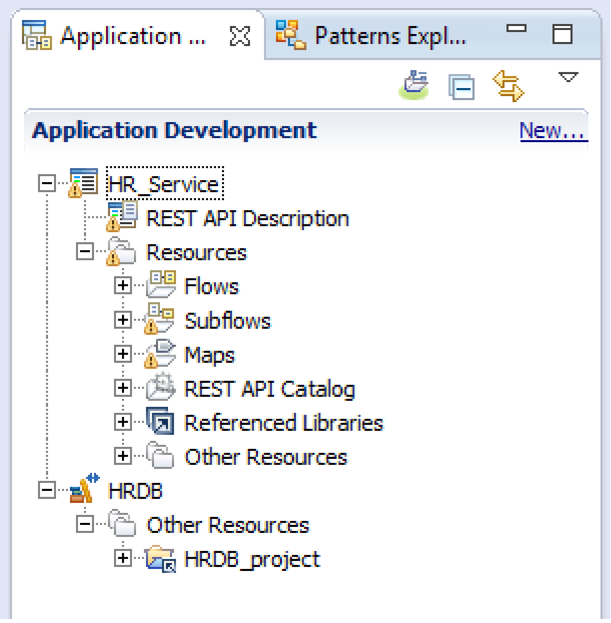


   Import the following Project Interchange (PI) file:

```
C:\student10\REST_Request_HR_Service\solution
HR_Service_getEmployee_getDepartment_getDetails.10.0.0.6.zip
```

   Note: Make sure that all three projects in this PI file are selected for import.

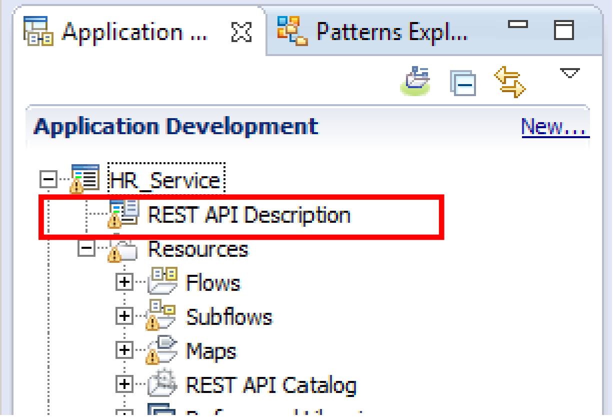
3. When imported, you should have in your workspace the HR_Service REST API and the HRDB shared library that is referenced by the REST API.



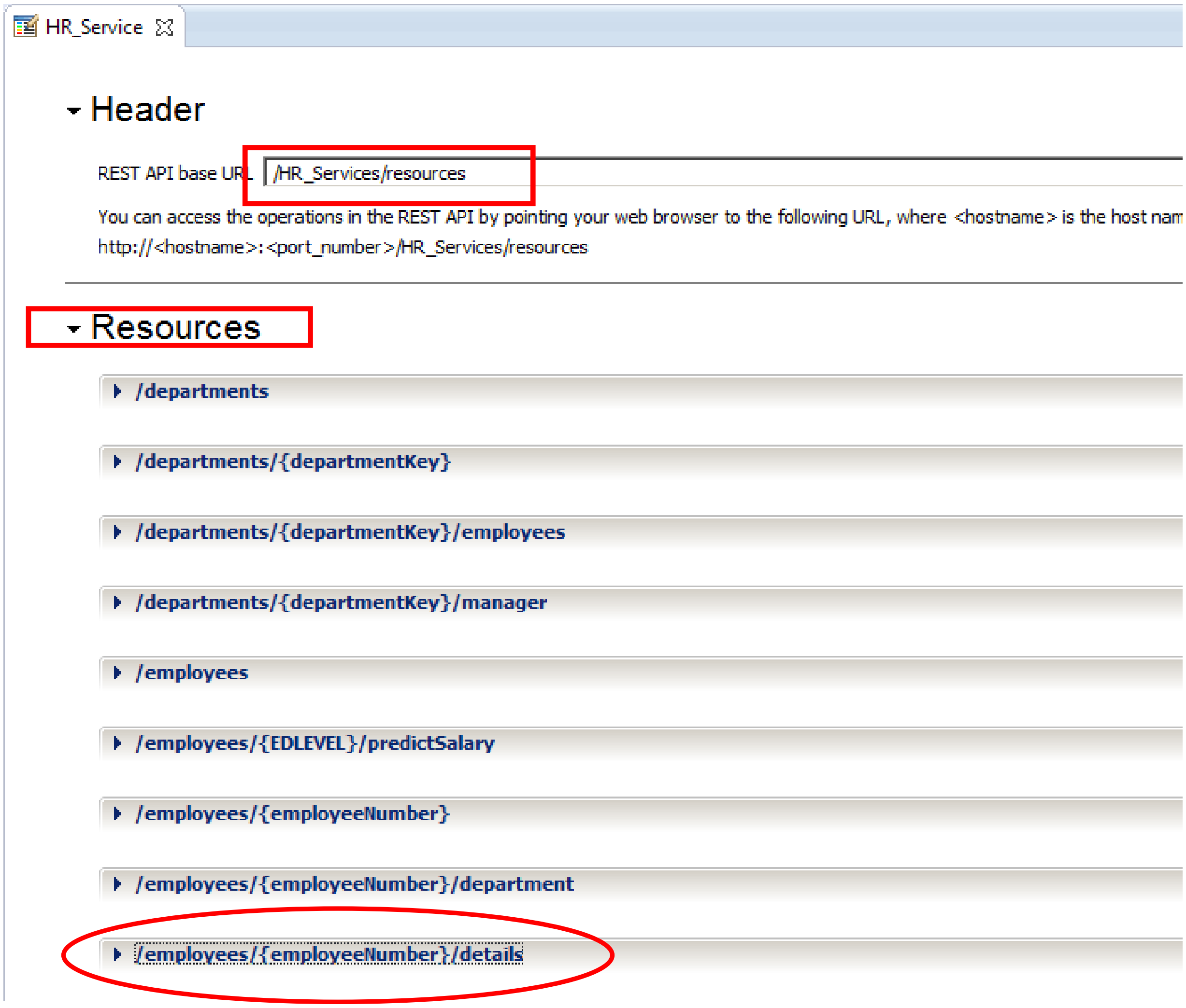
2.2 Explore the HR_Service REST API

In this section, you will explore the IIB REST API.

1. Expand the REST API Service and double-click “REST API Description”.



2. The details of the REST API are shown:



HR_Service

Header

REST API base URL /HR_Services/resources

You can access the operations in the REST API by pointing your web browser to the following URL, where <hostname> is the host nam

http://<hostname>:<port_number>/HR_Services/resources

Resources

- /departments
- /departments/{departmentKey}
- /departments/{departmentKey}/employees
- /departments/{departmentKey}/manager
- /employees
- /employees/{EDLEVEL}/predictSalary
- /employees/{employeeNumber}
- /employees/{employeeNumber}/department
- /employees/{employeeNumber}/details

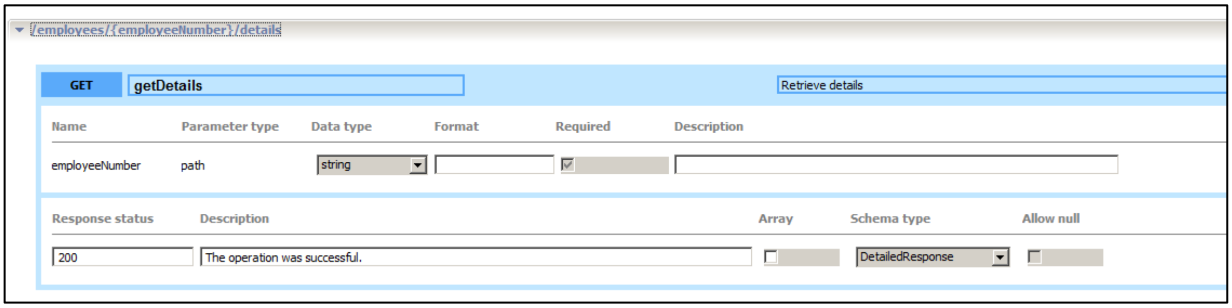
In the Header section you will see the REST API base URL as '**/HR_Services/resources'** which identifies the access path to the service over HTTP.

In the Resources section, 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 **/employees/{employeeNumber}/details** operation providing their employee number.

This operation has already been implemented in this provided service.

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

3. There is a single operation in this resource named **getDetails**.



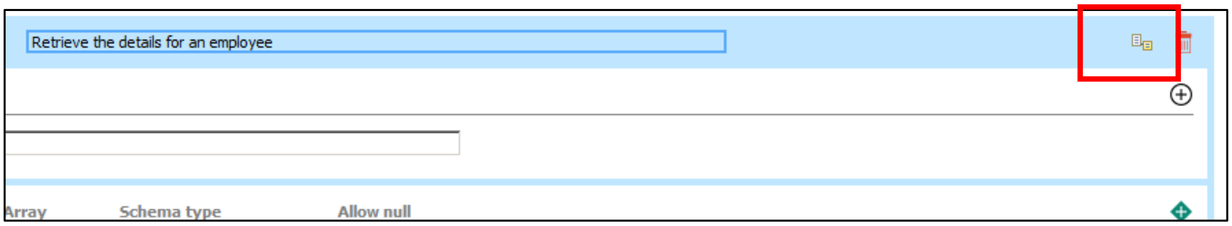
/employees/{employeeNumber}/details

GET getDetails Retrieve details

Name	Parameter type	Data type	Format	Required	Description
employeeNumber	path	string		☑	

Response status	Description	Array	Schema type	Allow null
200	The operation was successful.	☐	DetailedResponse	☐

Using the slide bar, scroll to the right, and click the icon to open the implementation of the getDetails operation.



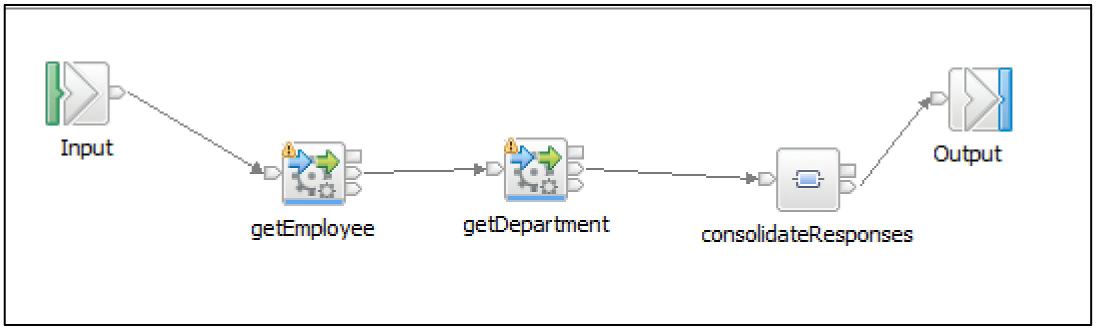
Retrieve the details for an employee

Array	Schema type	Allow null

4. This opens the implementation subflow.

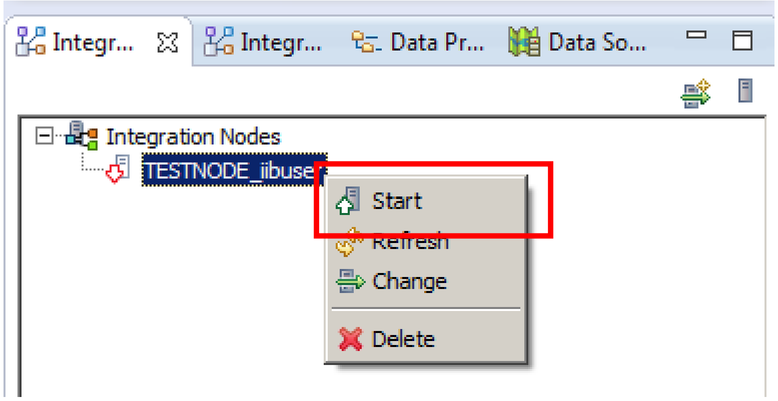
You can see that the getDetails operation itself invokes two operations, getEmployee and getDepartment. These are invoked using the IIB REST Request node (further explained in the REST Request lab guide, 16L04_IIB10006).

Finally, the results from these two operations are consolidated into a single response message for getDetails.



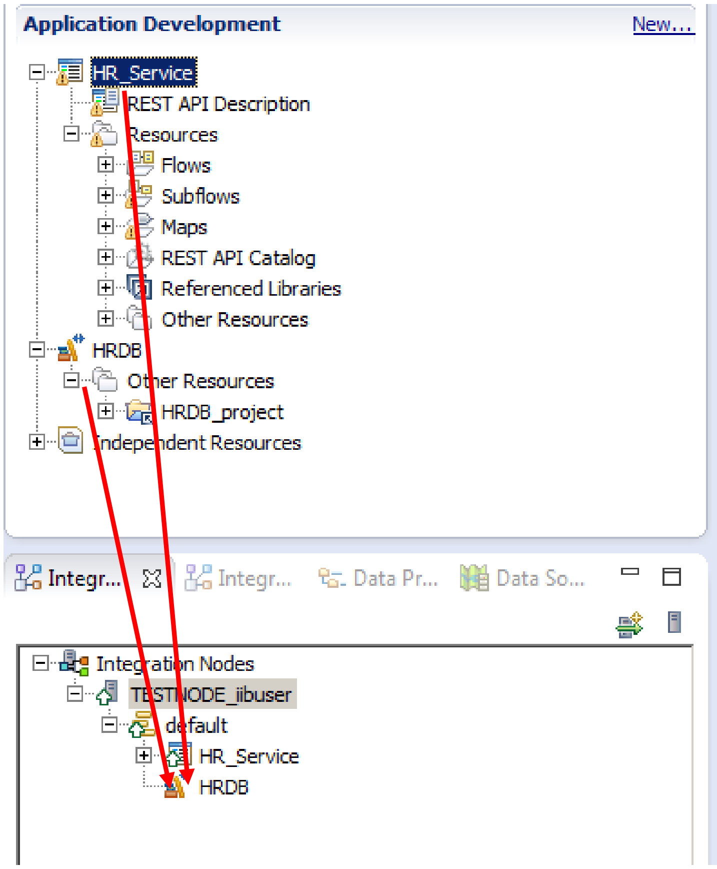
2.3 Deploy and test HR_Service

1. In the Integration Bus Toolkit, start the TESTNODE_iibuser Integration Node (if not started):



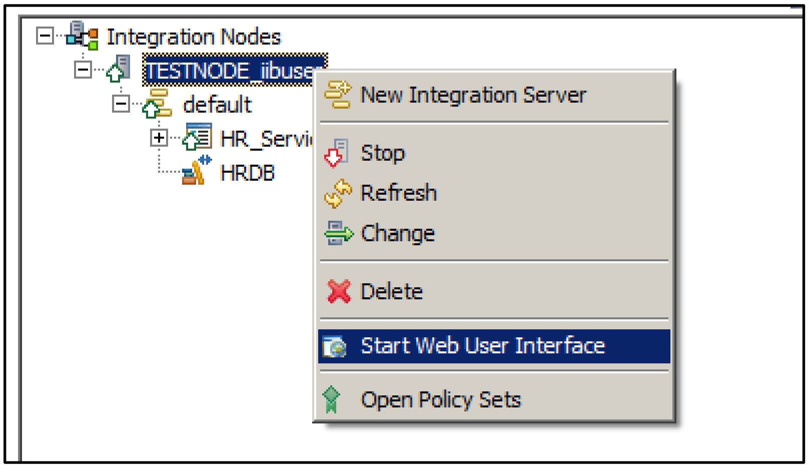
2. Deploy the library and REST API in the following order:

   - Shared Library – HRDB
   - HR_Service



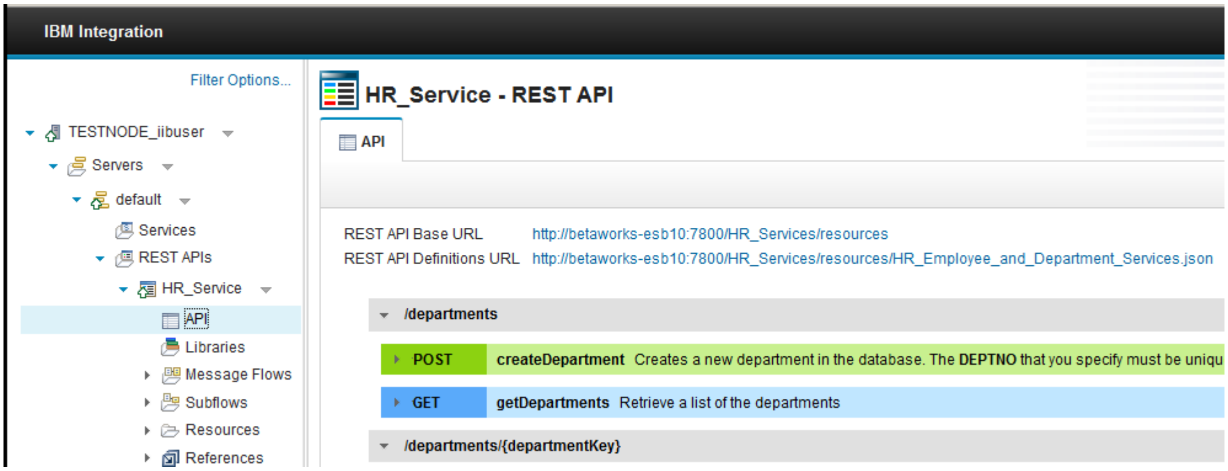
The order of deployment is important because the REST API service references the shared library and the deployment will fail if it does not find the shared library in the runtime when it is deployed.

3. When the deployment has completed, right-click the node and click Start Web User Interface.



4. In the web browser, fully expand the node, server, and the deployed REST API.

   Select the “API” item. This will display the URLs to invoke the REST API.



REST API Base URL http://betaworks-esb10:7800/HR_Services/resources

REST API Definitions URL http://betaworks-esb10:7800/HR_Services/resources/HR_Employee_and_Department_Services.json

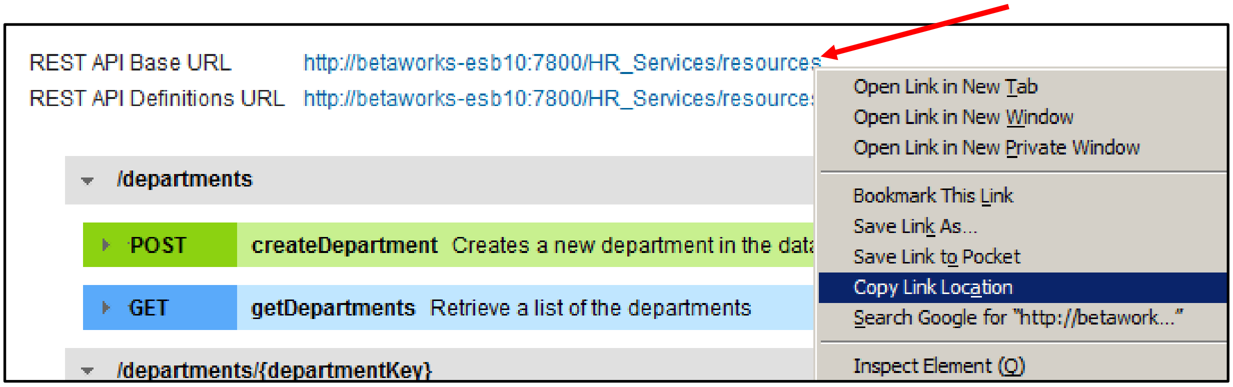
/departments

POST **createDepartment** Creates a new department in the database. The **DEPTNO** that you specify must be uniqu

GET **getDepartments** Retrieve a list of the departments

/departments/{departmentKey}

5. Right-click the “REST API Base URL” link, and select Copy Link Location.

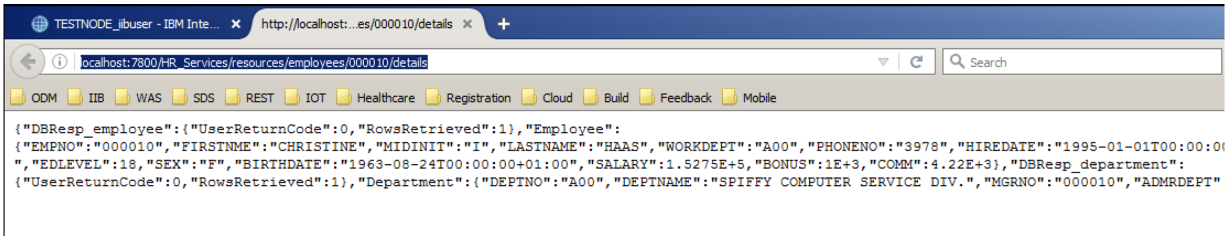

REST API Base URL
http://betaworks-esb10:7800/HR_Services/resources
REST API Definitions URL
http://betaworks-esb10:7800/HR_Services/resource:
/departments
POST
createDepartment Creates a new department in the data
GET
getDepartments Retrieve a list of the departments
/departments/{departmentKey}
Open Link in New Tab
Open Link in New Window
Open Link in New Private Window
Bookmark This Link
Save Link As…
Save Link to Pocket
Copy Link Location
Search Google for "http://betawork…"
Inspect Element (Q)

6. In a new browser tab, paste the copied URL, and append **/employees/details/000010/details**

   The final URL should be: http://localhost:7800/HR_Services/resources/employees/000010/details

   If you are using the provided VMWare image for the IIB workshop, the Mobile folder in the Firefox browser contains the URL shortcut titled “Local – 000010/details”.

   Executing this URL will invoke the REST API, with the following results.


TESTNODE_iibuser - IBM Inte…
http://localhost:…es/000010/details
localhost:7800/HR_Services/resources/employees/000010/details
Search
ODM
IIB
WAS
SDS
REST
IOT
Healthcare
Registration
Cloud
Build
Feedback
Mobile

```
{"DBResp_employee":{"UserReturnCode":0,"RowsRetrieved":1},"Employee":
{"EMPNO":"000010","FIRSTNME":"CHRISTINE","MIDINIT":"I","LASTNAME":"HAAS","WORKDEPT":"A00","PHONENO":"3978","HIREDATE":"1995-01-01T00:00:0
","EDLEVEL":18,"SEX":"F","BIRTHDATE":"1963-08-24T00:00:00+01:00","SALARY":1.5275E+5,"BONUS":1E+3,"COMM":4.22E+3},"DBResp_department":
{"UserReturnCode":0,"RowsRetrieved":1},"Department":{"DEPTNO":"A00","DEPTNAME":"SPIFFY COMPUTER SERVICE DIV.","MGRNO":"000010","ADMRDEPT"
```

   Although unformatted, note that the response message contains the four elements:

   - DBResp_employee
   - Employee
   - DBResp_department
   - Department

3. MobileFirst Integration with IBM Integration Bus

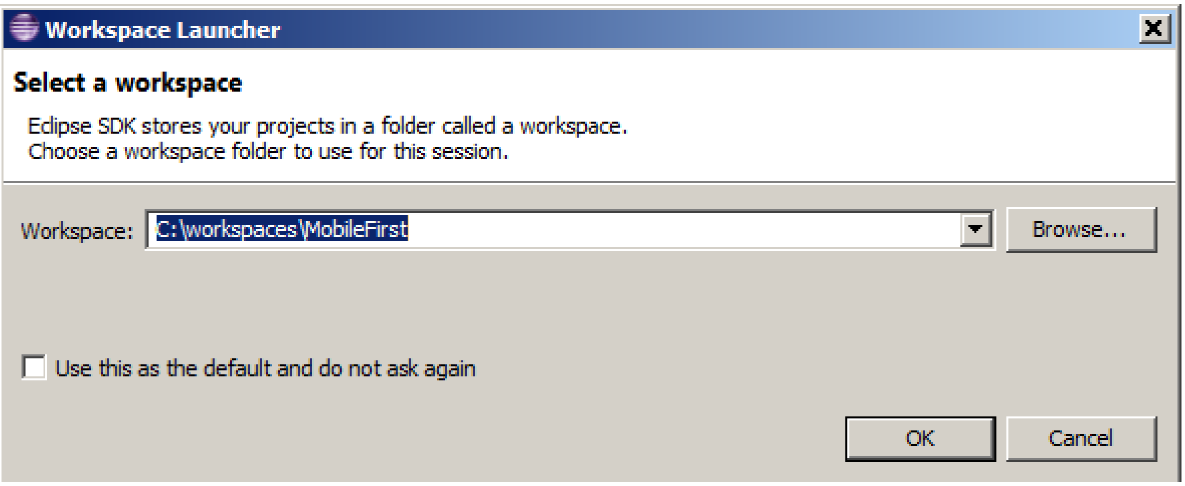
In this part of the lab, you will explore an integration adapter in the IBM MobileFirst Studio, which will connect to your application running on IBM Integration Bus.

3.1 Import the Employee Project into the MobileFirst Studio

1. Open the MobileFirst Studio (from the Windows Start menu).

   Create a new workspace, for example, c:\workspaces\MobileFirst.

   Note – the MobileFirst Studio should use a **different** workspace from the Integration Toolkit.



Workspace Launcher

Select a workspace

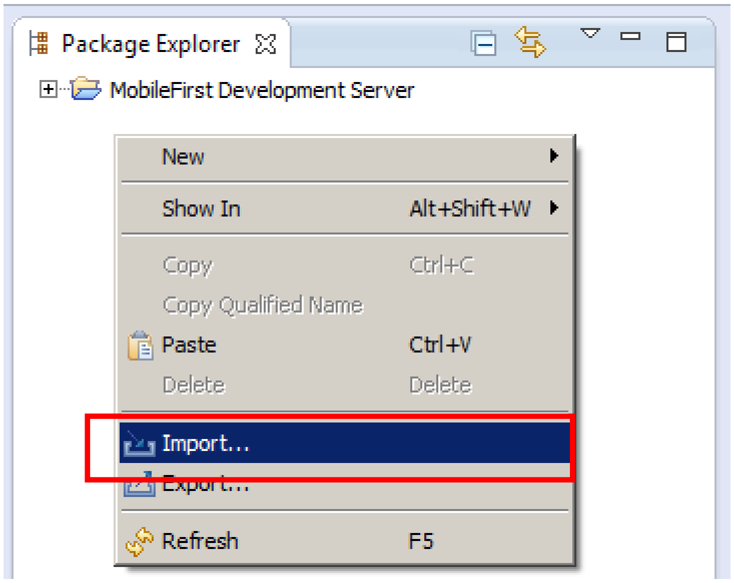
Eclipse SDK stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\workspaces\MobileFirst

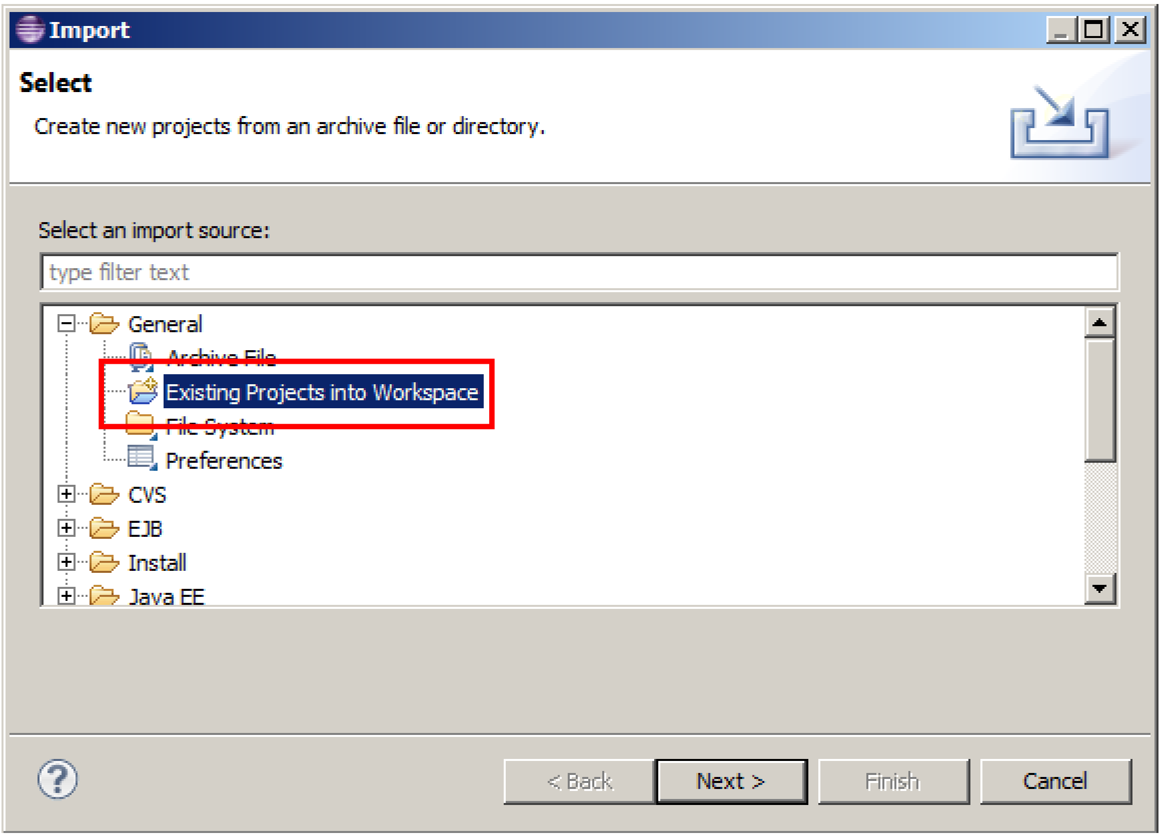
□ Use this as the default and do not ask again

2. For this lab, the Employee project for MobileFirst has already been built. The project comprises an application (app), and a choice of three adapters, required to connect to the IIB HR_Service REST API.

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



3. In the Import dialog, expand ‘General’ and select ‘Existing Projects into Workspace’. Click ‘Next’.



Import

Select

Create new projects from an archive file or directory.

Select an import source:

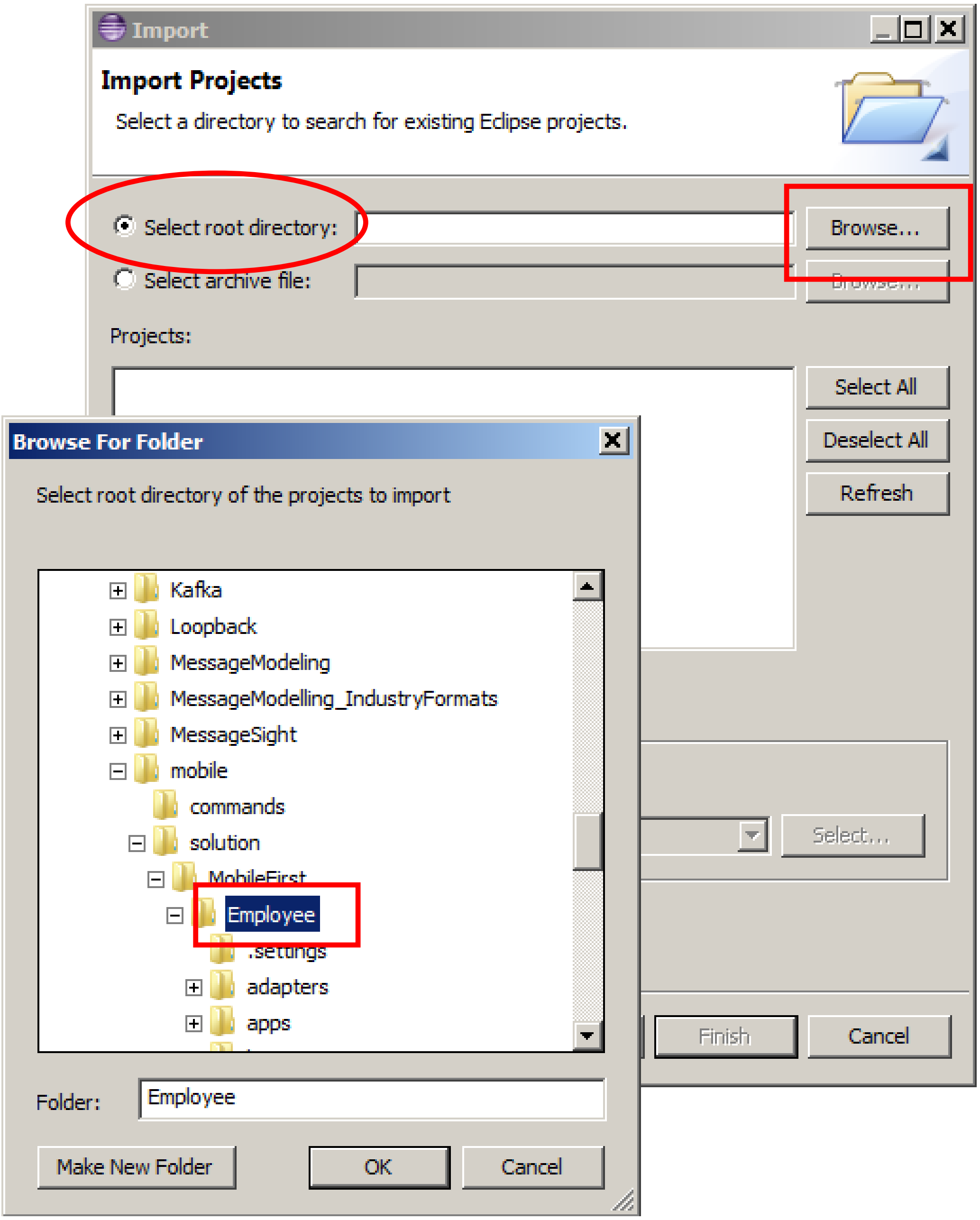
type filter text

4. Choose the “Select root directory” button, click ‘Browse’ and navigate to

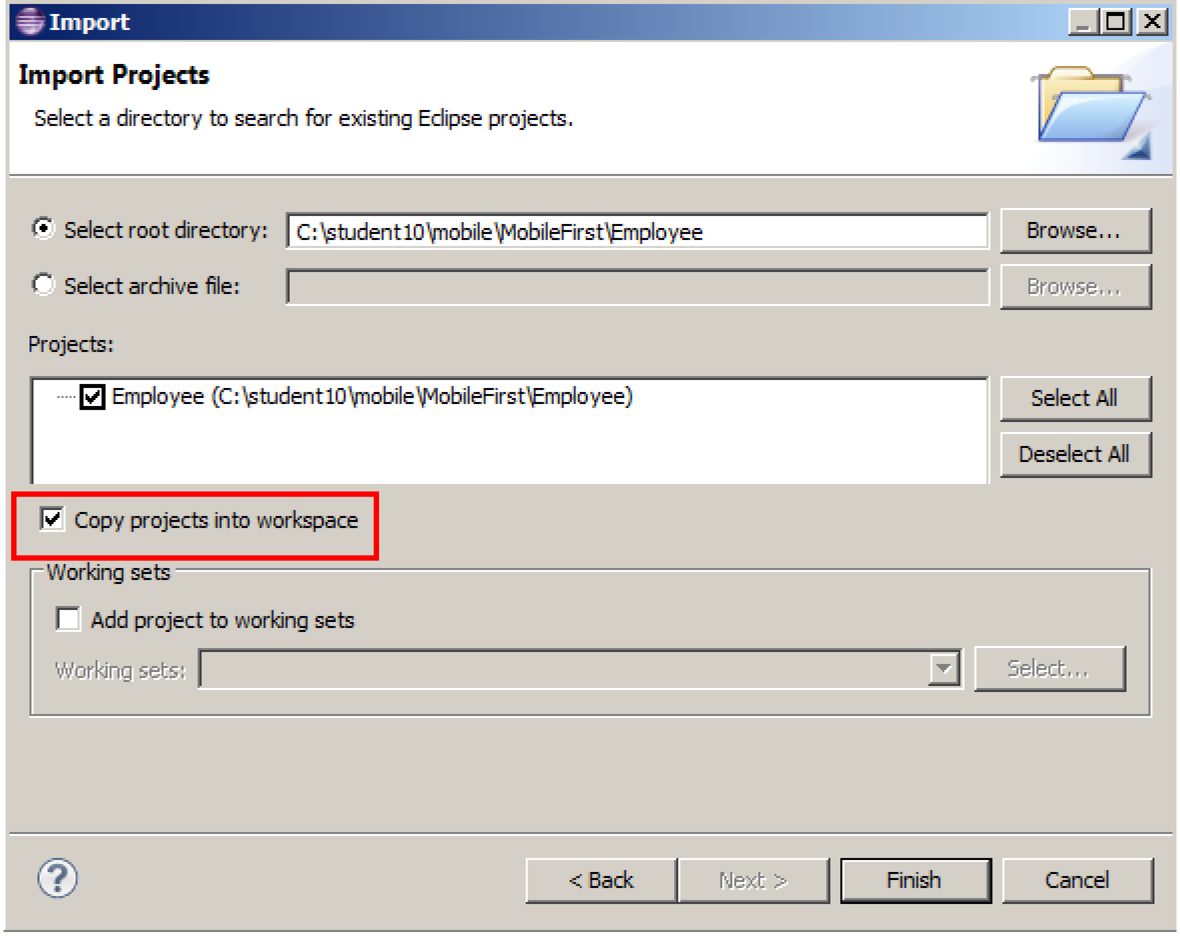
   c:\student10\mobile\solution\MobileFirst

   Select the **Employee** folder.

   Click ‘OK’.



5. Tick the box "Copy projects into workspace" and click Finish to complete the import of the project.



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\student10\mobile\MobileFirst\Employee

○ Select archive file:

Projects:

☑ Employee (C:\student10\mobile\MobileFirst\Employee)

☑ Copy projects into workspace

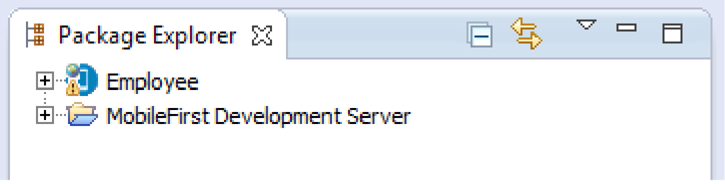
Working sets

□ Add project to working sets

Working sets:

.

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

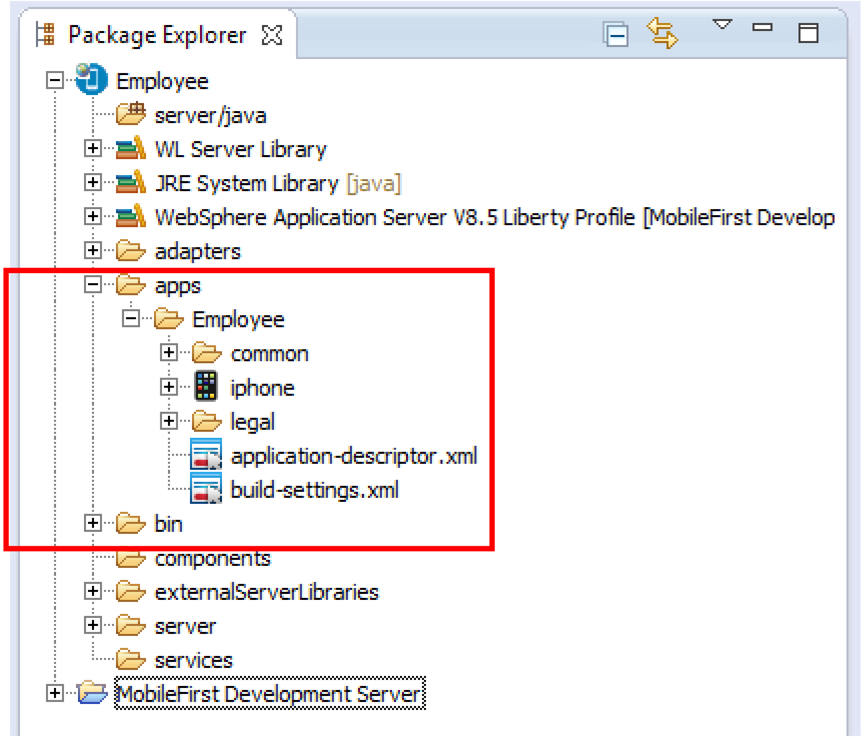


3.2 Employee app

The Mobile application that will be used in this lab has already been built. The application will invoke a MobileFirst adapter, which in turn will connect to the required operation in HR_Service.

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

1. Expand the Employee project, then ‘apps’ and ‘Employee.

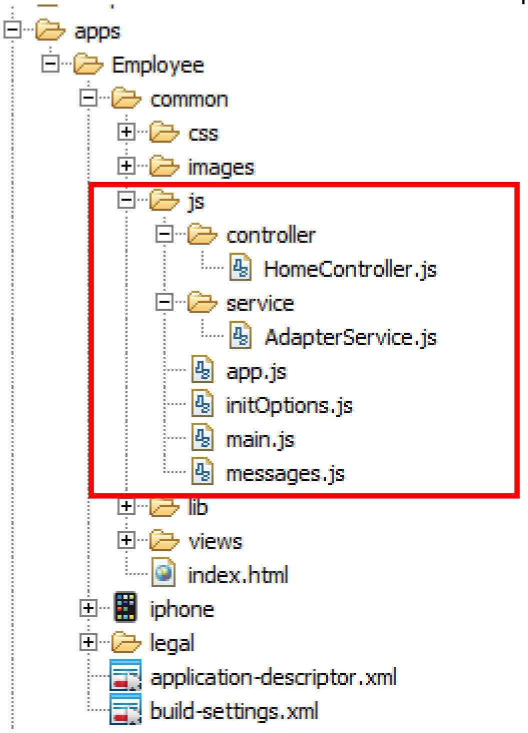


The directory contains 3 folders – common, iphone and legal:

- 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

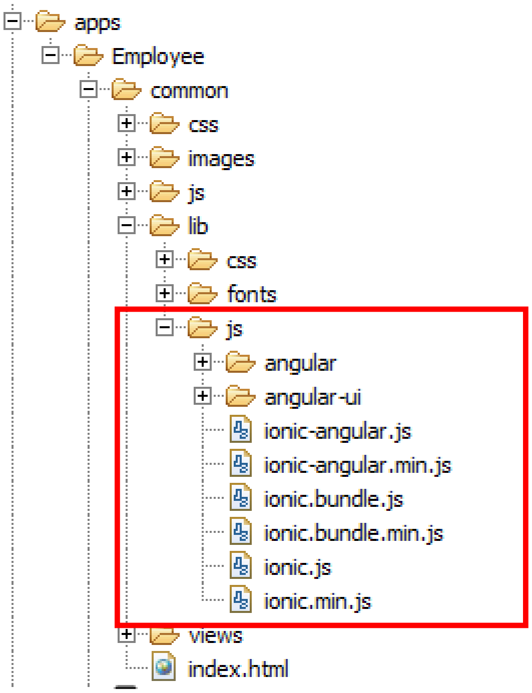
These folders are created from the MobileFirst Studio when the MobileFirst developer generates the project. Then they are customized by including resources for the Mobile application and its logic (i.e. HTML files, JavaScript files, libraries and images).

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.



```
apps
  Employee
    common
      css
      images
      js
        controller
          HomeController.js
        service
          AdapterService.js
        app.js
        initOptions.js
        main.js
        messages.js
      lib
      views
      index.html
    iphone
    legal
    application-descriptor.xml
    build-settings.xml
```

3. 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.

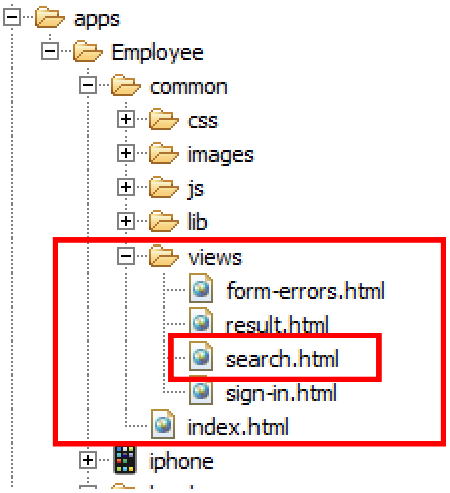


```
apps
  Employee
    common
      css
      images
      js
      lib
        css
        fonts
        js
          angular
          angular-ui
          ionic-angular.js
          ionic-angular.min.js
          ionic.bundle.js
          ionic.bundle.min.js
          ionic.js
          ionic.min.js
      views
      index.html
```

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

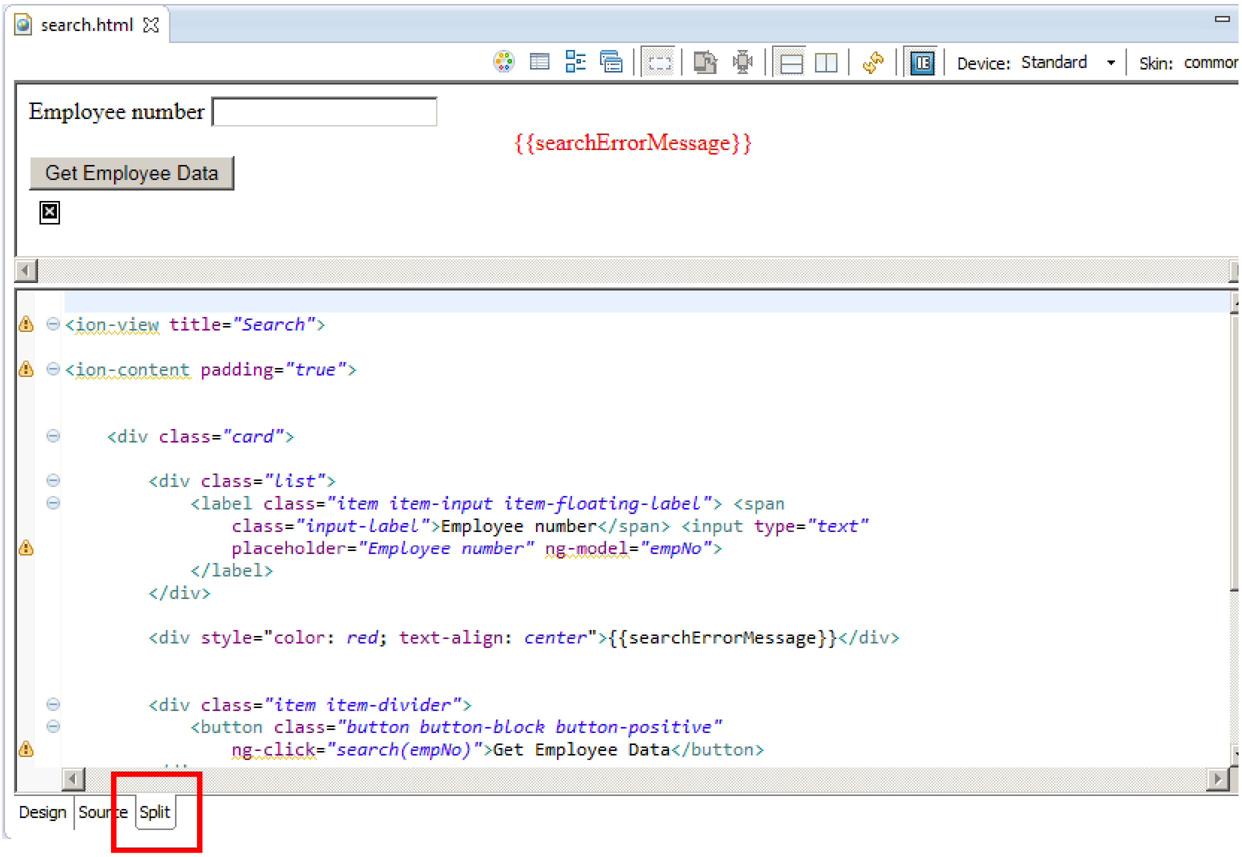
    Double-click search.html.



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).



```
<ion-view title="Search">

<ion-content padding="true">


    <div class="card">

        <div class="list">
            <label class="item item-input item-floating-label"> <span
                class="input-label">Employee number</span> <input type="text"
                placeholder="Employee number" ng-model="empNo">
            </label>
        </div>

        <div style="color: red; text-align: center">{{searchErrorMessage}}</div>


        <div class="item item-divider">
            <button class="button button-block button-positive"
                ng-click="search(empNo)">Get Employee Data</button>
```

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 Ionic 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.

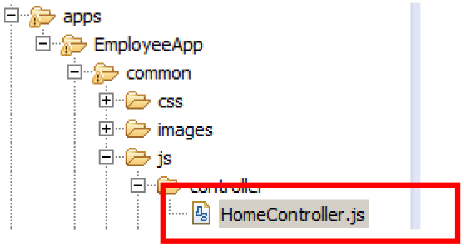
When finished reviewing, close the file without making any changes.

3.3 Data definition in ‘HomeController.js’

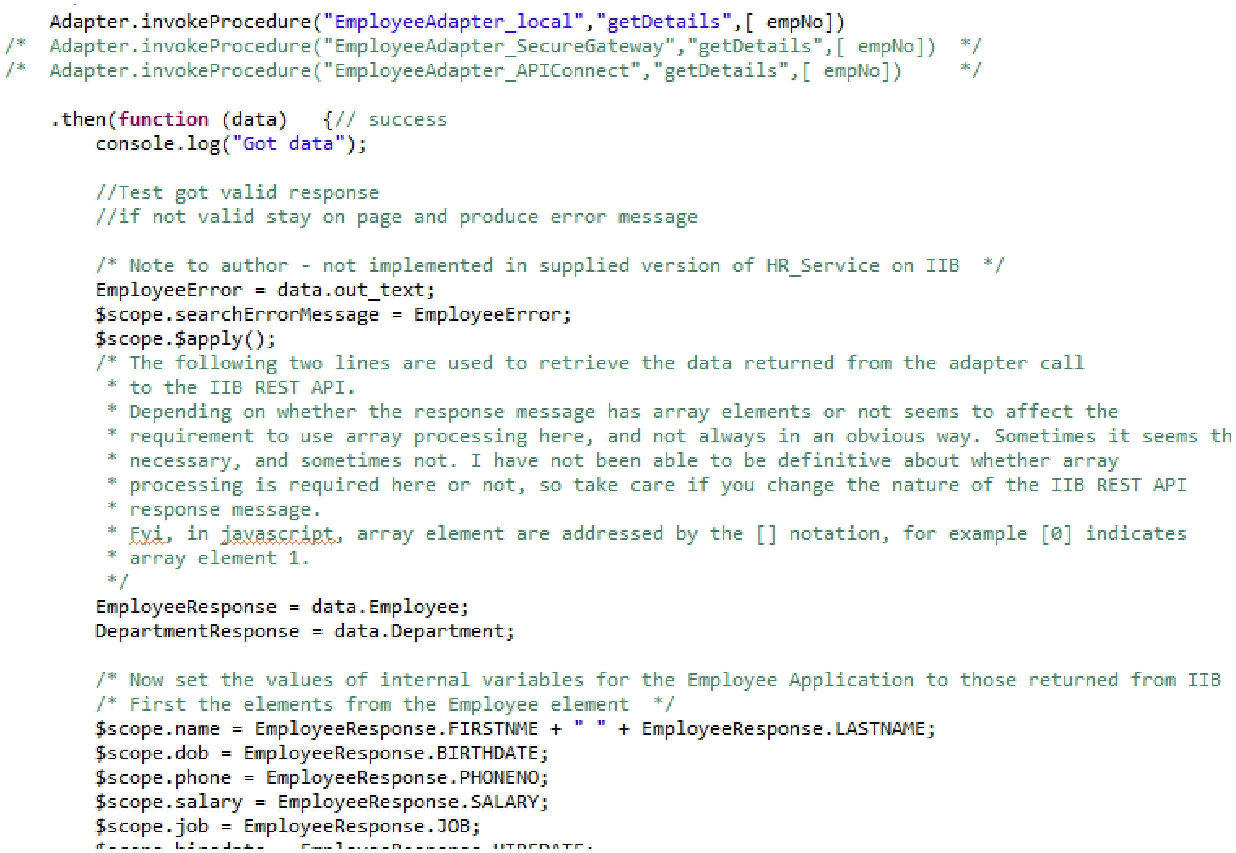
The ‘HomeController.js’ contains a JavaScript construction function which is used in Angular.JS to augment an Angula.JS object that refers to the application model. This controller is created by the MobileFirst developer.

1. Expand the folder **common/js/controlle**r.

   Open HomeController.js.



2. This javascript code primarily invokes the MobileFirst adapter. This is currently set to EmployeeAdapter_local. Data from the adapter is returned in the “data” variable.

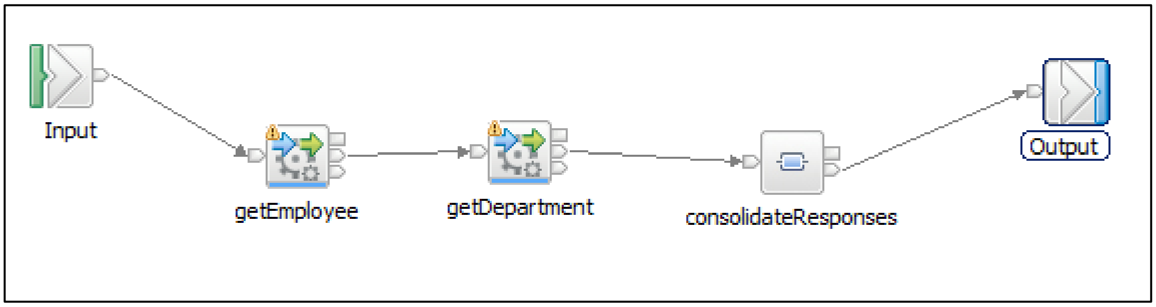


```
    Adapter.invokeProcedure("EmployeeAdapter_local","getDetails",[ empNo])
/*  Adapter.invokeProcedure("EmployeeAdapter_SecureGateway","getDetails",[ empNo])  */
/*  Adapter.invokeProcedure("EmployeeAdapter_APIConnect","getDetails",[ empNo])     */

    .then(function (data)   {// success
        console.log("Got data");

        //Test got valid response
        //if not valid stay on page and produce error message

        /* Note to author - not implemented in supplied version of HR_Service on IIB  */
        EmployeeError = data.out_text;
        $scope.searchErrorMessage = EmployeeError;
        $scope.$apply();
        /* The following two lines are used to retrieve the data returned from the adapter call
         * to the IIB REST API.
         * Depending on whether the response message has array elements or not seems to affect the
         * requirement to use array processing here, and not always in an obvious way. Sometimes it seems th
         * necessary, and sometimes not. I have not been able to be definitive about whether array
         * processing is required here or not, so take care if you change the nature of the IIB REST API
         * response message.
         * Fyi, in javascript, array element are addressed by the [] notation, for example [0] indicates
         * array element 1.
         */
        EmployeeResponse = data.Employee;
        DepartmentResponse = data.Department;

        /* Now set the values of internal variables for the Employee Application to those returned from IIB
        /* First the elements from the Employee element  */
        $scope.name = EmployeeResponse.FIRSTNME + " " + EmployeeResponse.LASTNAME;
        $scope.dob = EmployeeResponse.BIRTHDATE;
        $scope.phone = EmployeeResponse.PHONENO;
        $scope.salary = EmployeeResponse.SALARY;
        $scope.job = EmployeeResponse.JOB;
        $scope.hiredate = EmployeeResponse.HIREDATE;
```

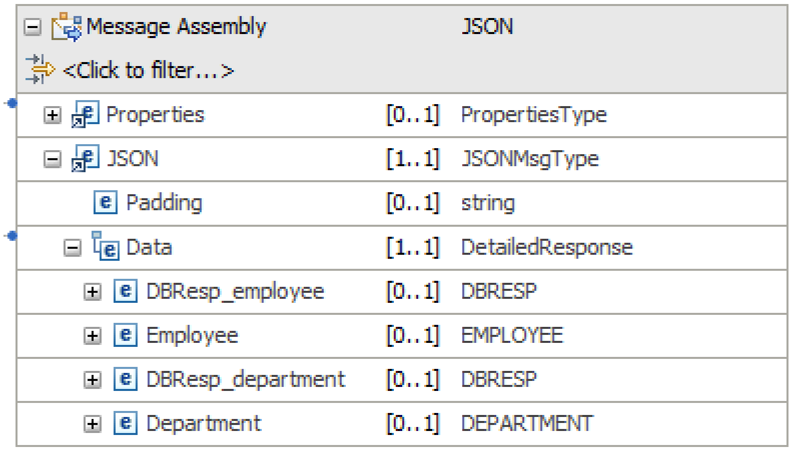
3.4 IIB data structure

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 “getDetails”.

   Open the **consolidateResponses** map.



2. Expand the output message assembly on the right hand side.



Message Assembly		JSON
<Click to filter…>		
Properties	[0..1]	PropertiesType
JSON	[1..1]	JSONMsgType
Padding	[0..1]	string
Data	[1..1]	DetailedResponse
DBResp_employee	[0..1]	DBRESP
Employee	[0..1]	EMPLOYEE
DBResp_department	[0..1]	DBRESP
Department	[0..1]	DEPARTMENT

   You will see the output JSON object **Data** tree as shown above.

   Back in the MobileFirst workbench, in the Employee app **HomeController.js** file, this data is referenced by the following statements:

```
EmployeeResponse = data.Employee;
DepartmentResponse = data.Department;
```

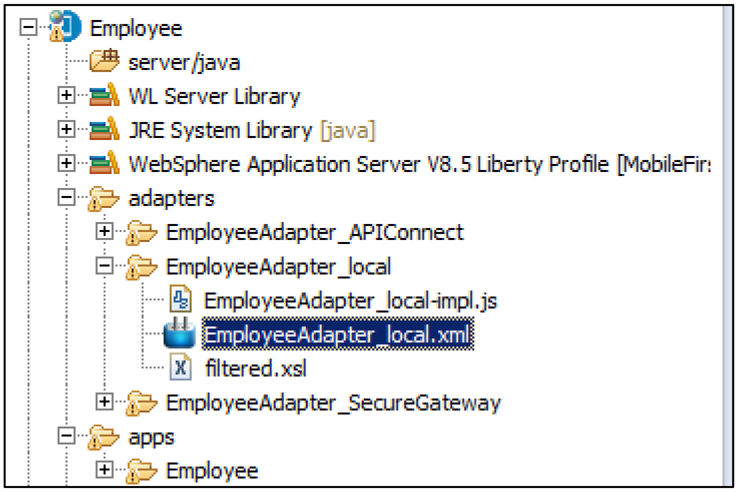
   Close the map and the message flow.

3.5 Review the MobileFirst adapter EmployeeAdapter_local

In this part of the lab you will explore, deploy and test the EmployeeAdapter_local adapter. This will establish a local connection to the HR_Service REST API (using localhost), and will be used by the Employee MobileFirst app.

1. In the MobileFirst Studio, expand adapters, and EmployeeAdapter_local.

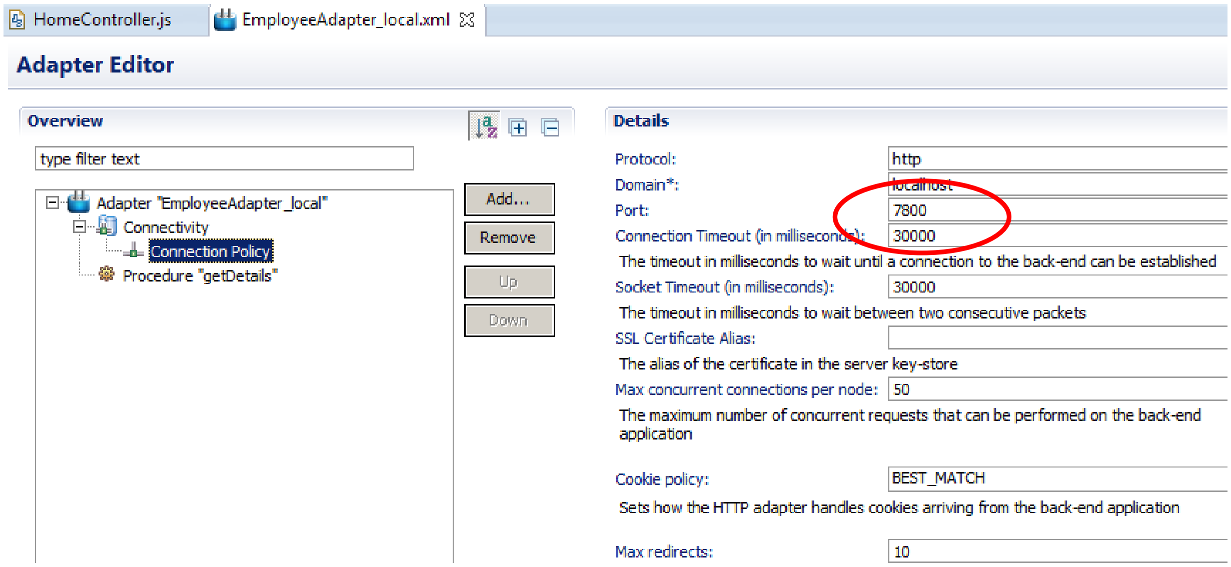
   Open the EmployeeAdapter_local.xml file



```
Employee
  server/java
  WL Server Library
  JRE System Library [java]
  WebSphere Application Server V8.5 Liberty Profile [MobileFir:
  adapters
    EmployeeAdapter_APIConnect
    EmployeeAdapter_local
      EmployeeAdapter_local-impl.js
      EmployeeAdapter_local.xml
      filtered.xsl
    EmployeeAdapter_SecureGateway
  apps
    Employee
```

2. Expand the adapter and Connectivity, and click Connection Policy.

   Note the following settings:
   - Domain:
   - Port:



Protocol: http

Domain*: localhost

Port: 7800

Connection Timeout (in milliseconds): 30000

The timeout in milliseconds to wait until a connection to the back-end can be established

Socket Timeout (in milliseconds): 30000

The timeout in milliseconds to wait between two consecutive packets

SSL Certificate Alias:

The alias of the certificate in the server key-store

Max concurrent connections per node: 50

The maximum number of concurrent requests that can be performed on the back-end application

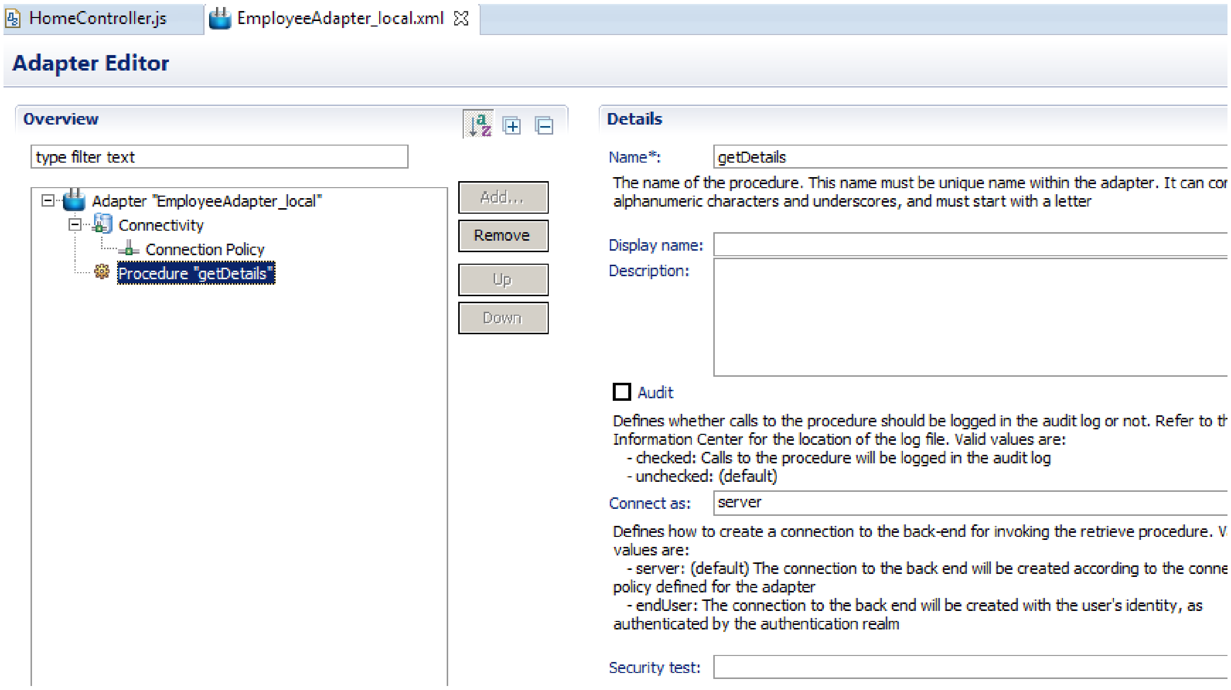
Cookie policy: BEST_MATCH

Sets how the HTTP adapter handles cookies arriving from the back-end application

Max redirects: 10

3. Click **“Procedure getDetails**”.

   Note that the name of the procedure is getDetails, which matches the name of the target operation in the IIB HR_Service. This is not strictly necessary, but is good practice for overall end-to-end consistency.



Name*: getDetails

The name of the procedure. This name must be unique name within the adapter. It can cor alphanumeric characters and underscores, and must start with a letter

Display name:

Description:

□ Audit

Defines whether calls to the procedure should be logged in the audit log or not. Refer to th 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)

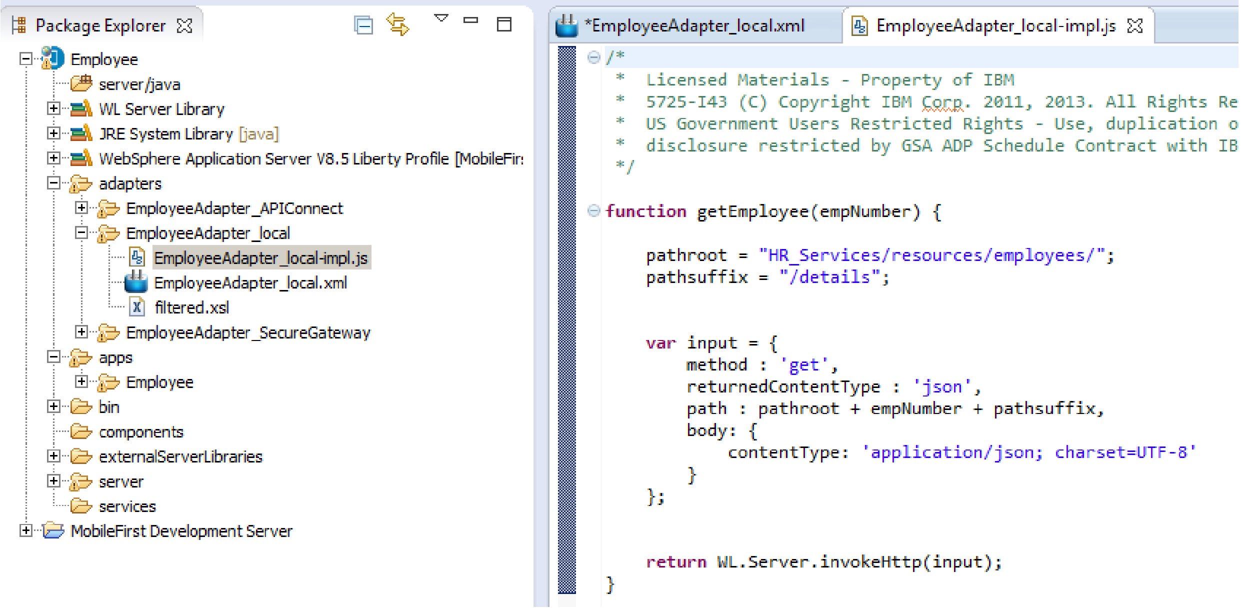
Connect as: server

Defines how to create a connection to the back-end for invoking the retrieve procedure. V values are:
- server: (default) The connection to the back end will be created according to the conne policy defined for the adapter
- endUser: The connection to the back end will be created with the user's identity, as authenticated by the authentication realm

Security test:

3.6 Adapter Implementation File

1. Open the ‘EmployeeAdapter_local-impl.js’ file.


Package Explorer
Employee
server/java
WL Server Library
JRE System Library [java]
WebSphere Application Server V8.5 Liberty Profile [MobileFir
adapters
EmployeeAdapter_APIConnect
EmployeeAdapter_local
EmployeeAdapter_local-impl.js
EmployeeAdapter_local.xml
filtered.xsl
EmployeeAdapter_SecureGateway
apps
Employee
bin
components
externalServerLibraries
server
services
MobileFirst Development Server
*EmployeeAdapter_local.xml
EmployeeAdapter_local-impl.js

```
/*
 *  Licensed Materials - Property of IBM
 *  5725-I43 (C) Copyright IBM Corp. 2011, 2013. All Rights Re
 *  US Government Users Restricted Rights - Use, duplication o
 *  disclosure restricted by GSA ADP Schedule Contract with IB
 */

function getEmployee(empNumber) {

    pathroot = "HR_Services/resources/employees/";
    pathsuffix = "/details";


    var input = {
        method : 'get',
        returnedContentType : 'json',
        path : pathroot + empNumber + pathsuffix,
        body: {
            contentType: 'application/json; charset=UTF-8'
        }
    };


    return WL.Server.invokeHttp(input);
}
```

2. The default generated adapter code has been changed because the getDetails operation has a URL suffix (“/details”), which must be added after the empNumber element (note – the “empNumber” variable in MobileFirst corresponds to the “employeeNumber” element in the Integration Bus schema.

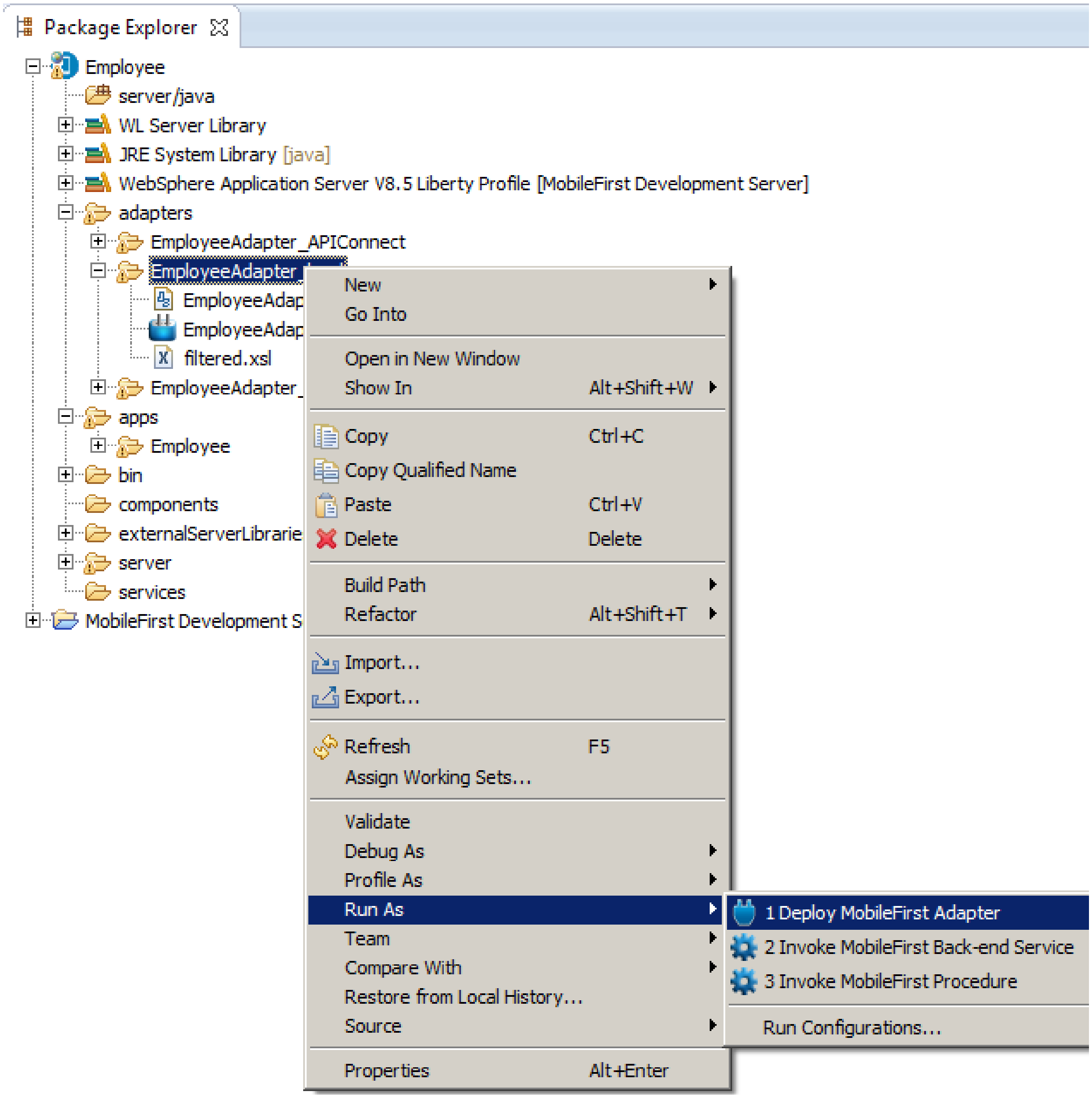
   This is done by using two javascript variables, pathroot and pathsuffix, as shown below. If you need to invoke a REST API with a different URL, you will need to make appropriate changes to this javascript.

```
function getDetails(empNumber) {

    pathroot = "HR_Services/resources/employees/";
    pathsuffix = "/details";


    var input = {
        method : 'get',
        returnedContentType : 'json',
        path : pathroot + empNumber + pathsuffix,
        body: {
          contentType: 'application/json; charset=UTF-8'
        }
    };


    return WL.Server.invokeHttp(input);
```

4. Test the Mobile Application using a Local Connection

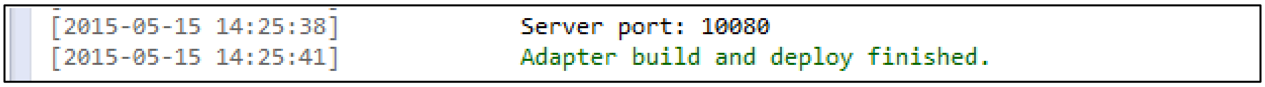
4.1 Deploy and test the local adapter

1. In the MobileFirst Package Explorer, right-click the EmployeeAdapter_local and select ‘Run As’ --> ‘Deploy MobileFirst Adapter.



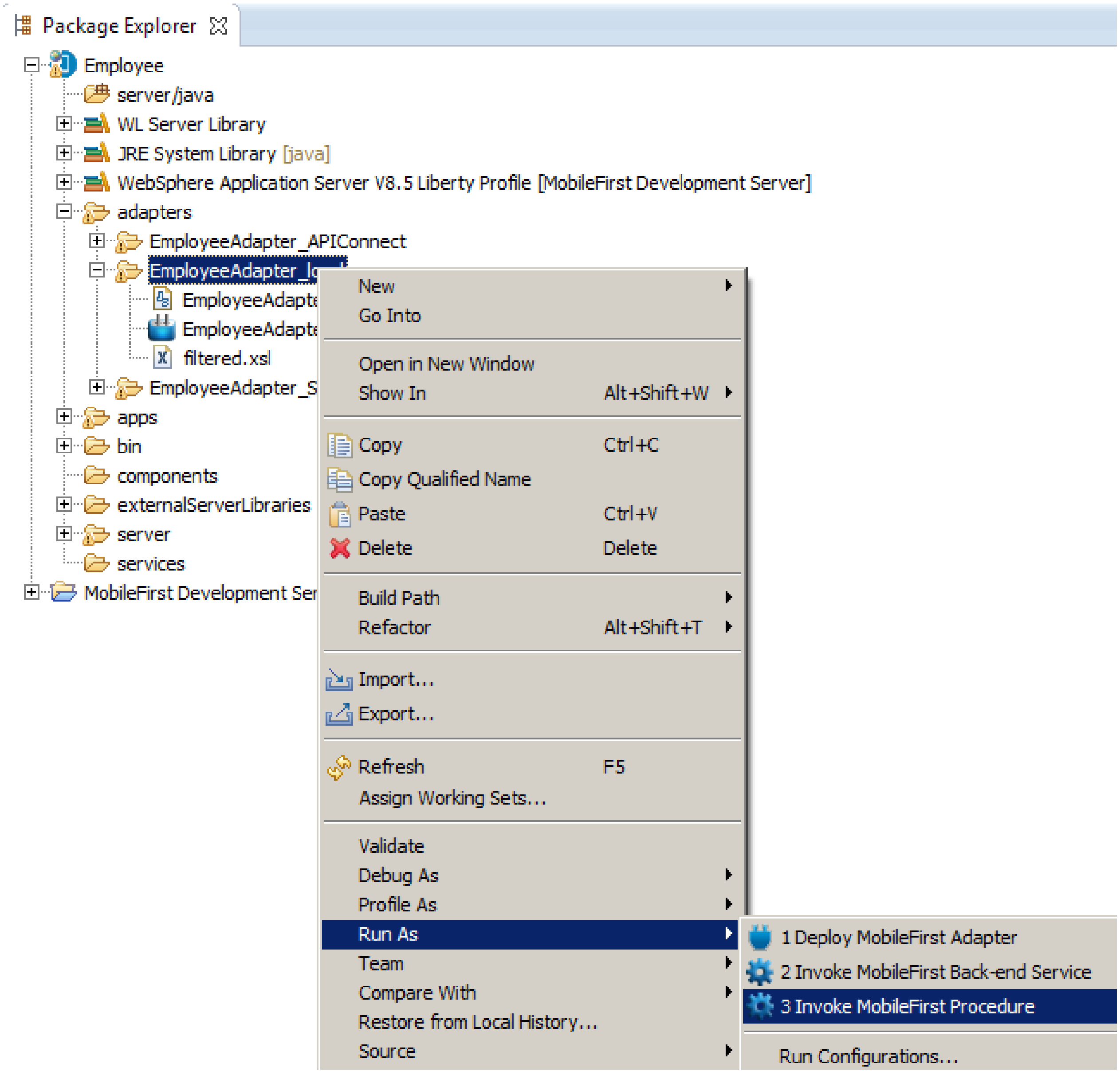
Please note that this may take a minute or two.

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



```
[2015-05-15 14:25:38]            Server port: 10080
[2015-05-15 14:25:41]            Adapter build and deploy finished.
```

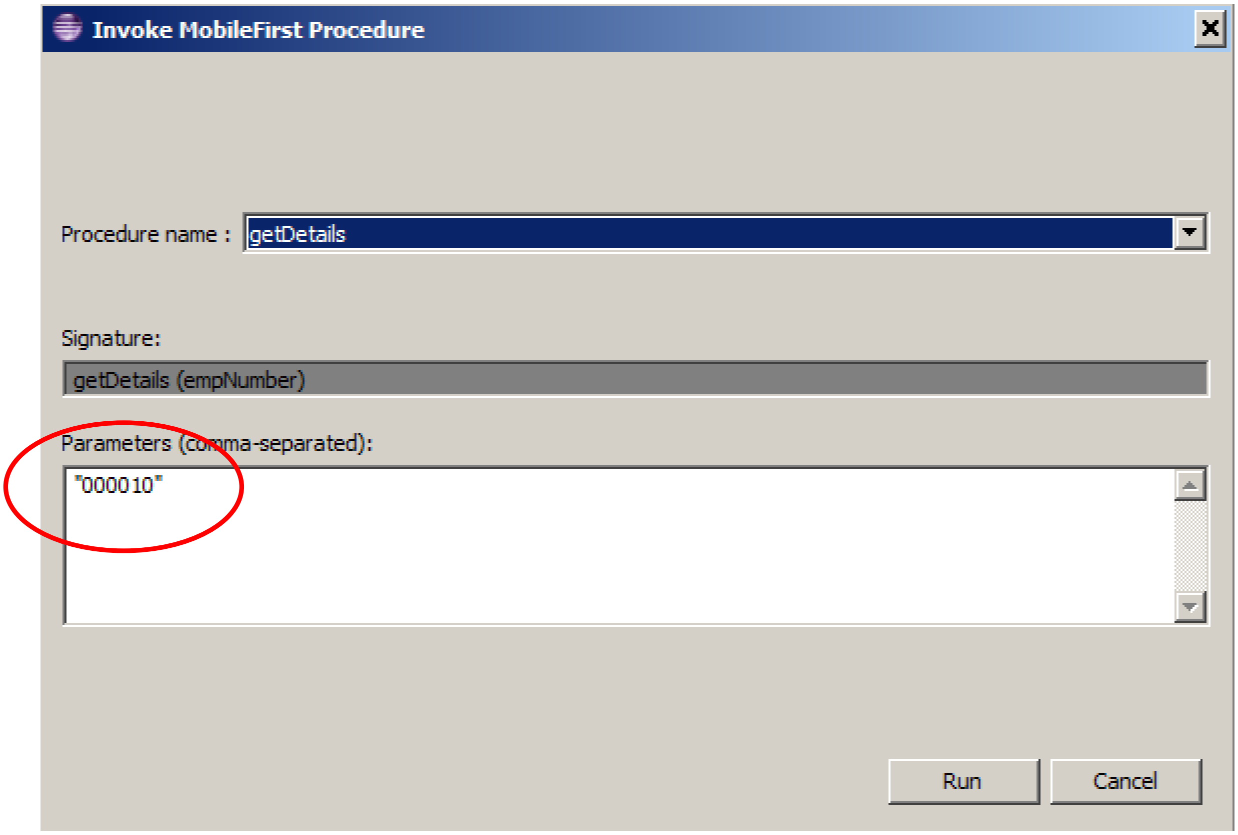
3. The adapter can be tested individually, without needing an app to invoke it. To perform a local test of the deployed adapter, right-click the EmployeeAdapter_local, and select Run As, Invoke MobileFirst Procedure.



4. In the new window, provide the input data a shown. Note the double apostrophes are required.

“000010”

Click Run.

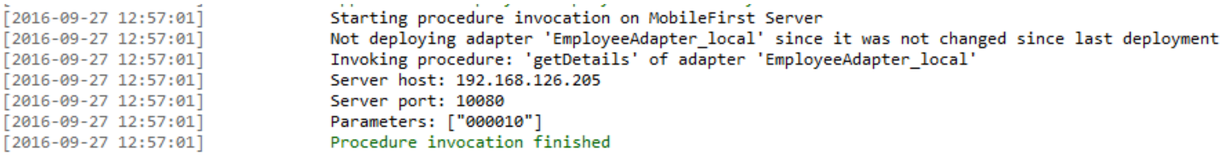


Procedure name : getDetails

Signature: getDetails (empNumber)

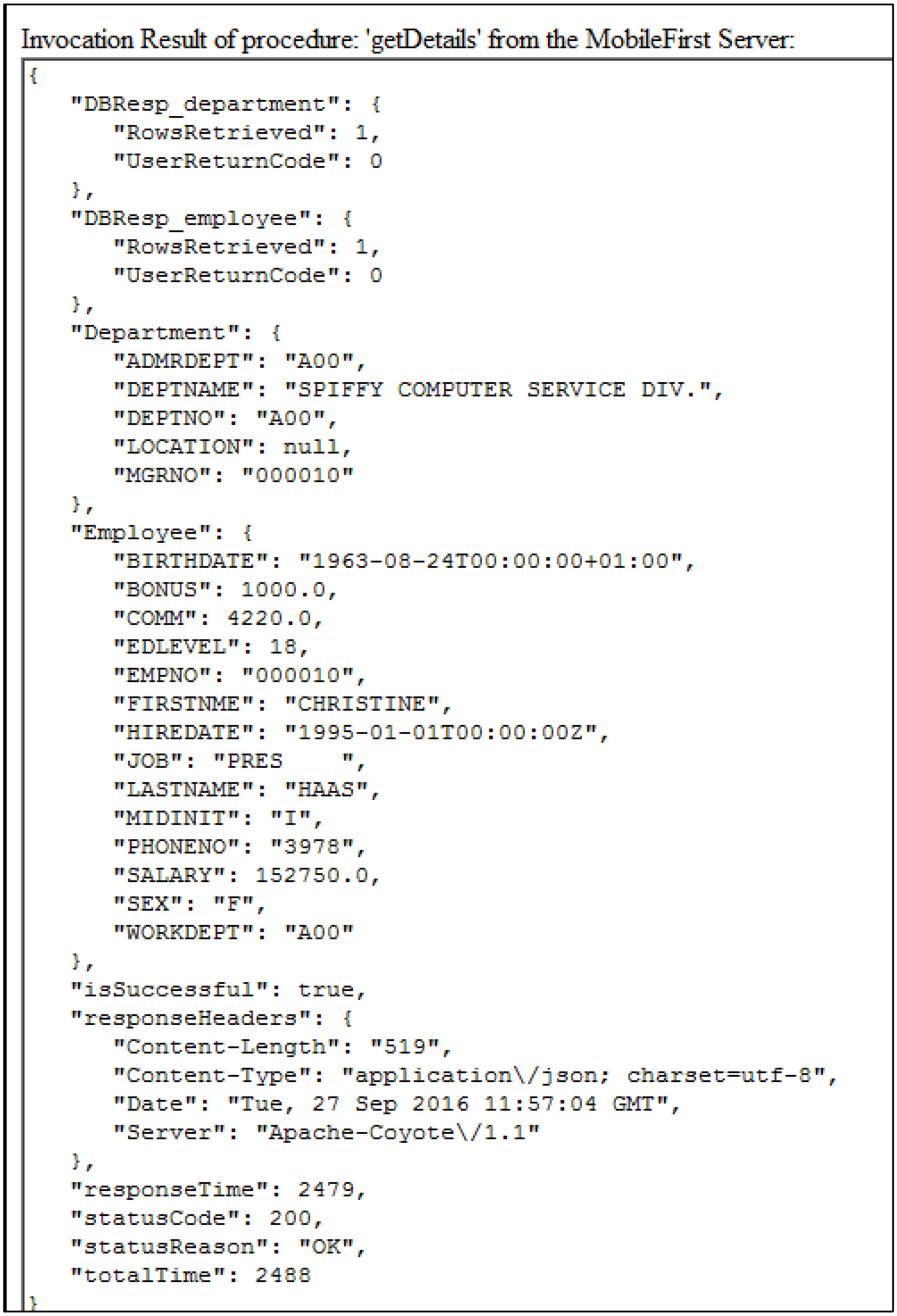
Parameters (comma-separated): "000010"

5. In the MobileFirst Studio, In the Console output, you will see the adapter being invoked.



```
[2016-09-27 12:57:01]             Starting procedure invocation on MobileFirst Server
[2016-09-27 12:57:01]             Not deploying adapter 'EmployeeAdapter_local' since it was not changed since last deployment
[2016-09-27 12:57:01]             Invoking procedure: 'getDetails' of adapter 'EmployeeAdapter_local'
[2016-09-27 12:57:01]             Server host: 192.168.126.205
[2016-09-27 12:57:01]             Server port: 10080
[2016-09-27 12:57:01]             Parameters: ["000010"]
[2016-09-27 12:57:01]             Procedure invocation finished
```

6. Switching to the browser tab that automatically opened, you will see the response data returned from HR_Service.

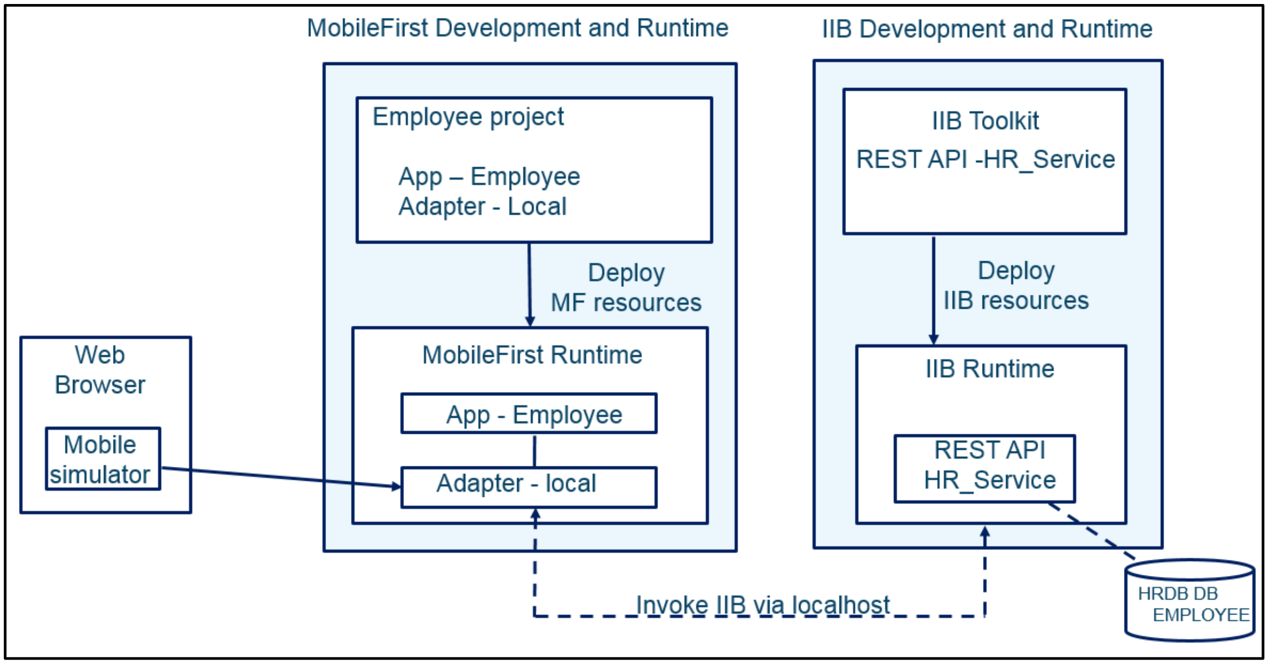


Invocation Result of procedure: 'getDetails' from the MobileFirst Server:

```
{
   "DBResp_department": {
      "RowsRetrieved": 1,
      "UserReturnCode": 0
   },
   "DBResp_employee": {
      "RowsRetrieved": 1,
      "UserReturnCode": 0
   },
   "Department": {
      "ADMRDEPT": "A00",
      "DEPTNAME": "SPIFFY COMPUTER SERVICE DIV.",
      "DEPTNO": "A00",
      "LOCATION": null,
      "MGRNO": "000010"
   },
   "Employee": {
      "BIRTHDATE": "1963-08-24T00:00:00+01:00",
      "BONUS": 1000.0,
      "COMM": 4220.0,
      "EDLEVEL": 18,
      "EMPNO": "000010",
      "FIRSTNME": "CHRISTINE",
      "HIREDATE": "1995-01-01T00:00:00Z",
      "JOB": "PRES    ",
      "LASTNAME": "HAAS",
      "MIDINIT": "I",
      "PHONENO": "3978",
      "SALARY": 152750.0,
      "SEX": "F",
      "WORKDEPT": "A00"
   },
   "isSuccessful": true,
   "responseHeaders": {
      "Content-Length": "519",
      "Content-Type": "application\/json; charset=utf-8",
      "Date": "Tue, 27 Sep 2016 11:57:04 GMT",
      "Server": "Apache-Coyote\/1.1"
   },
   "responseTime": 2479,
   "statusCode": 200,
   "statusReason": "OK",
   "totalTime": 2488
}
```

You now know that the adapter has successfully connected to the HR_Service in IIB. You can now proceed to using the adapter from within a MobileFirst app.

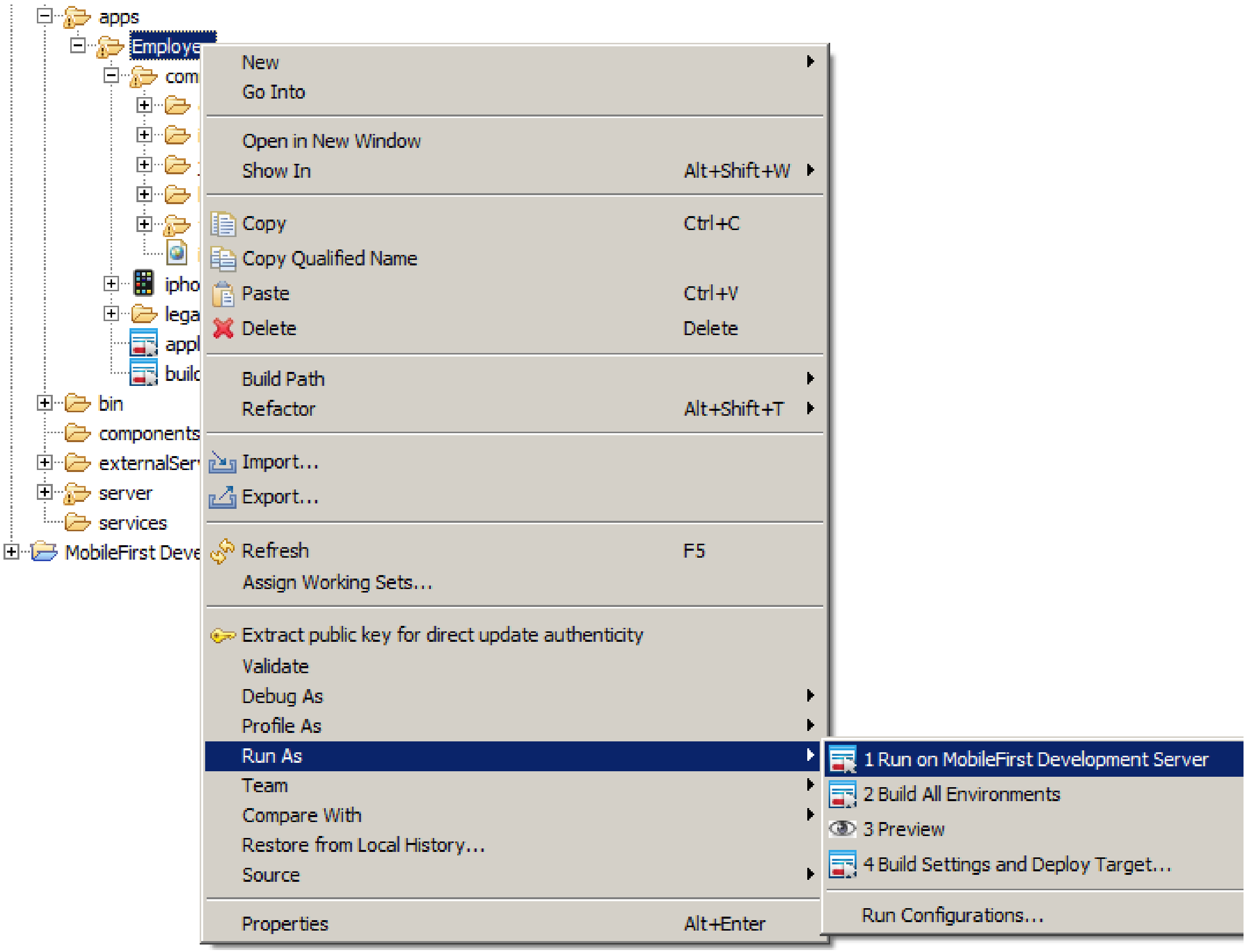
7. As a reminder, you have just used the MobileFirst Adapter simulator to directly invoke the Local Adapter for MobileFirst. This connects directly to the IIB REST API.



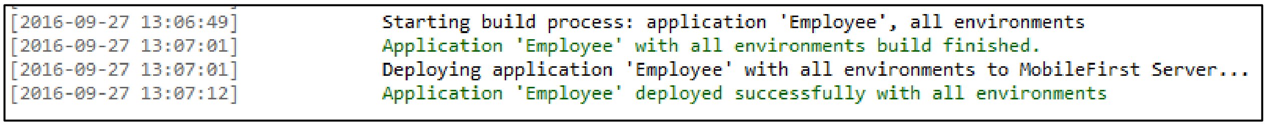
4.2 Deploy and Test MobileFirst Employee App

The next task is to deploy the Employee app, and look at the deployed MobileFirst assets through the Operations Console.

1. Right-click the Employee app, and select **Run As, Run on MobileFirst Development Server**.

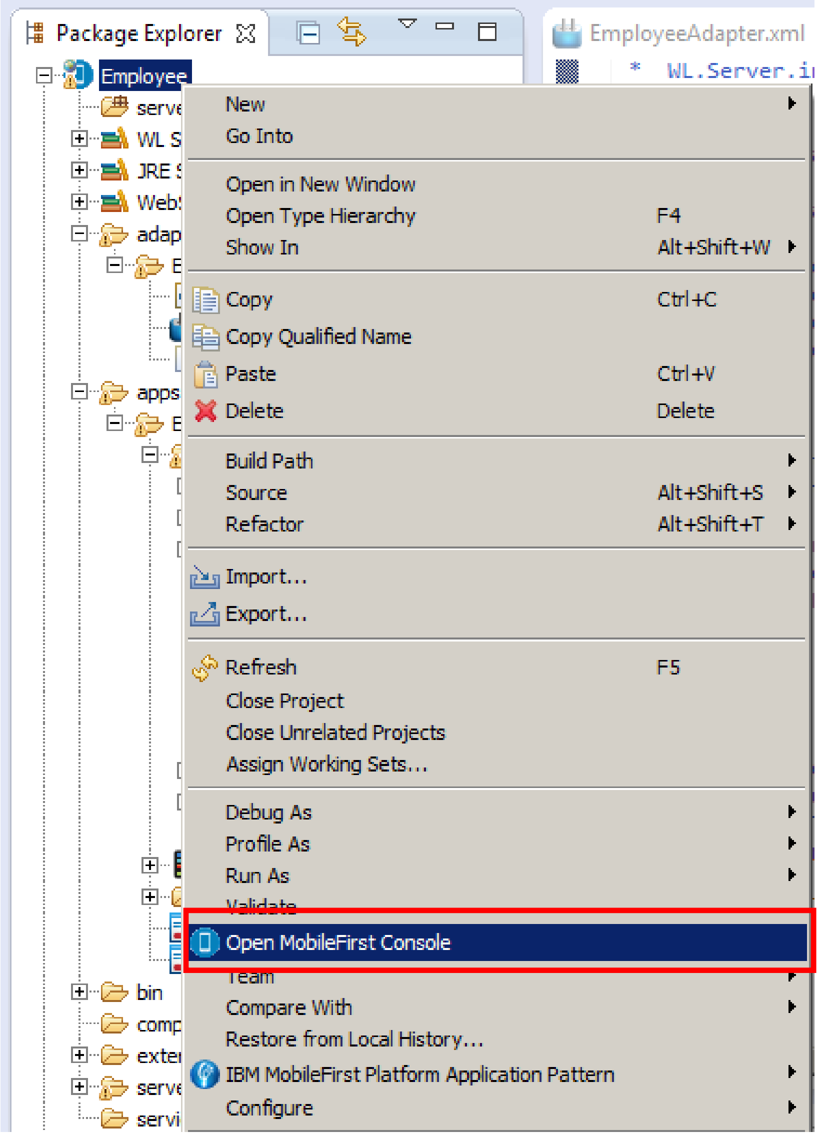


2. Progress messages will be shown in the Console log.



```
[2016-09-27 13:06:49]           Starting build process: application 'Employee', all environments
[2016-09-27 13:07:01]           Application 'Employee' with all environments build finished.
[2016-09-27 13:07:01]           Deploying application 'Employee' with all environments to MobileFirst Server...
[2016-09-27 13:07:12]           Application 'Employee' deployed successfully with all environments
```

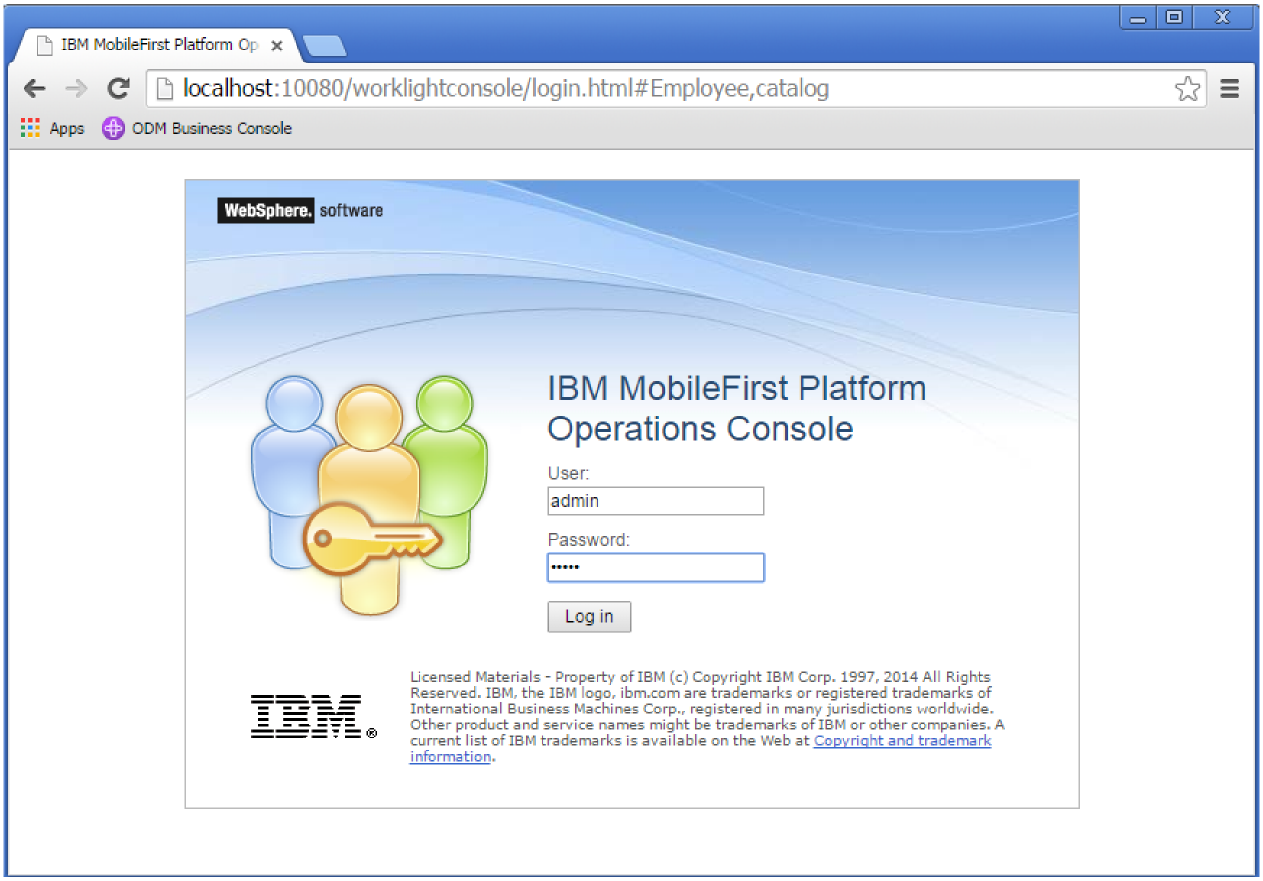
3. To open the MobileFirst Console, right-click the **Employee project** (not the Employee app) and from the menu click on ‘Open MobileFirst Console’.



4. A web browser tab will open automatically with the log in page for IBM MobileFirst console.

   Please note that the Firefox Browser (the default Browser on the workshop VMWare image) **will not be able to render the mobile pages view.** Google Chrome will be able to render these pages, so if Chrome is not your default browser, copy the generated URL and open in Chrome.

   Enter admin/admin as User/ Password.



localhost:10080/worklightconsole/login.html#Employee,catalog

IBM MobileFirst Platform Operations Console

User: admin

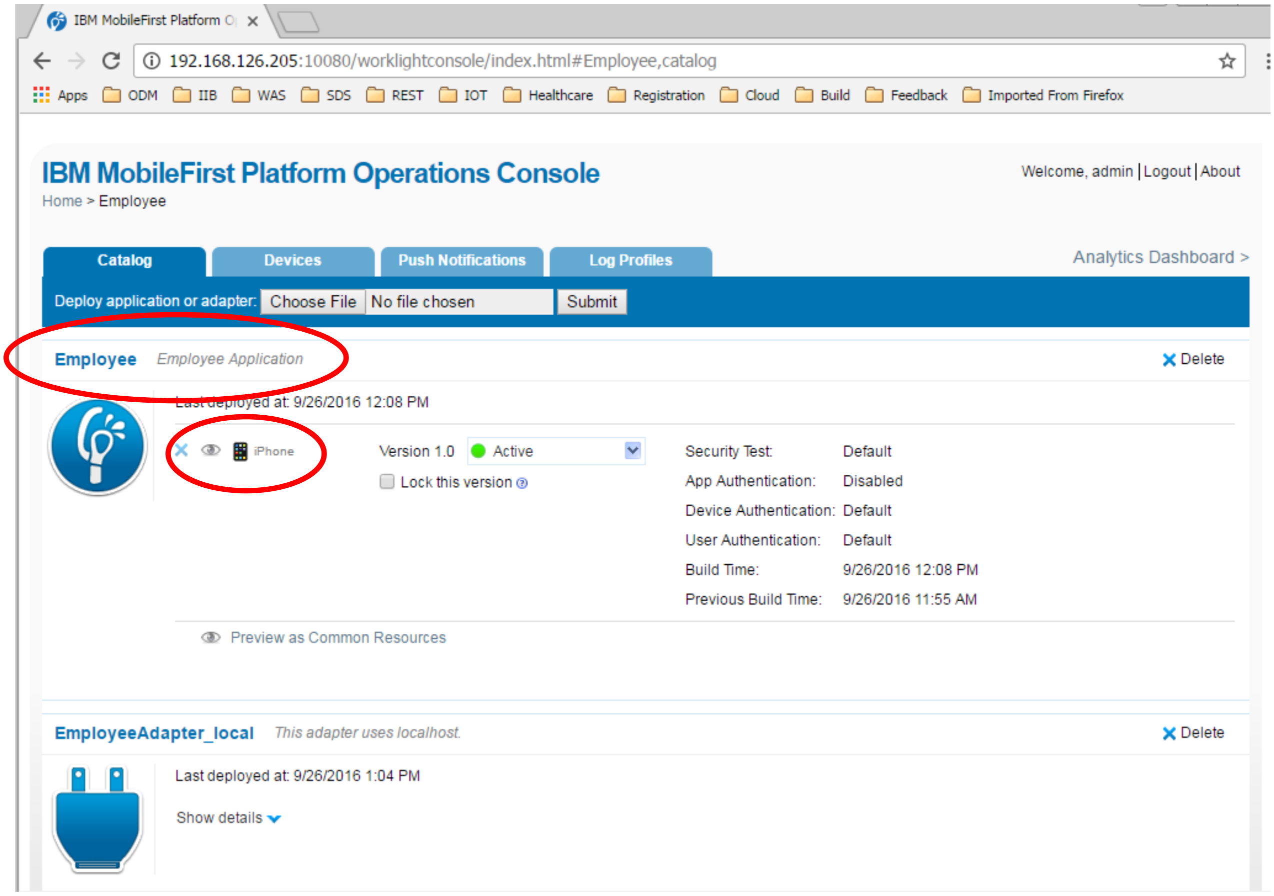
Password: •••••

Licensed Materials - Property of IBM (c) Copyright IBM Corp. 1997, 2014 All Rights Reserved. IBM, the IBM logo, ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information.

5. In the console you will see the deployed Employee app. 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 uses the local connection to the IIB REST API.

   Click the ‘iPhone’ icon as shown below:



IBM MobileFirst Platform Operations Console

Deploy application or adapter: No file chosen

Employee *Employee Application*

Last deployed at: 9/26/2016 12:08 PM

Version 1.0 ● Active

□ Lock this version

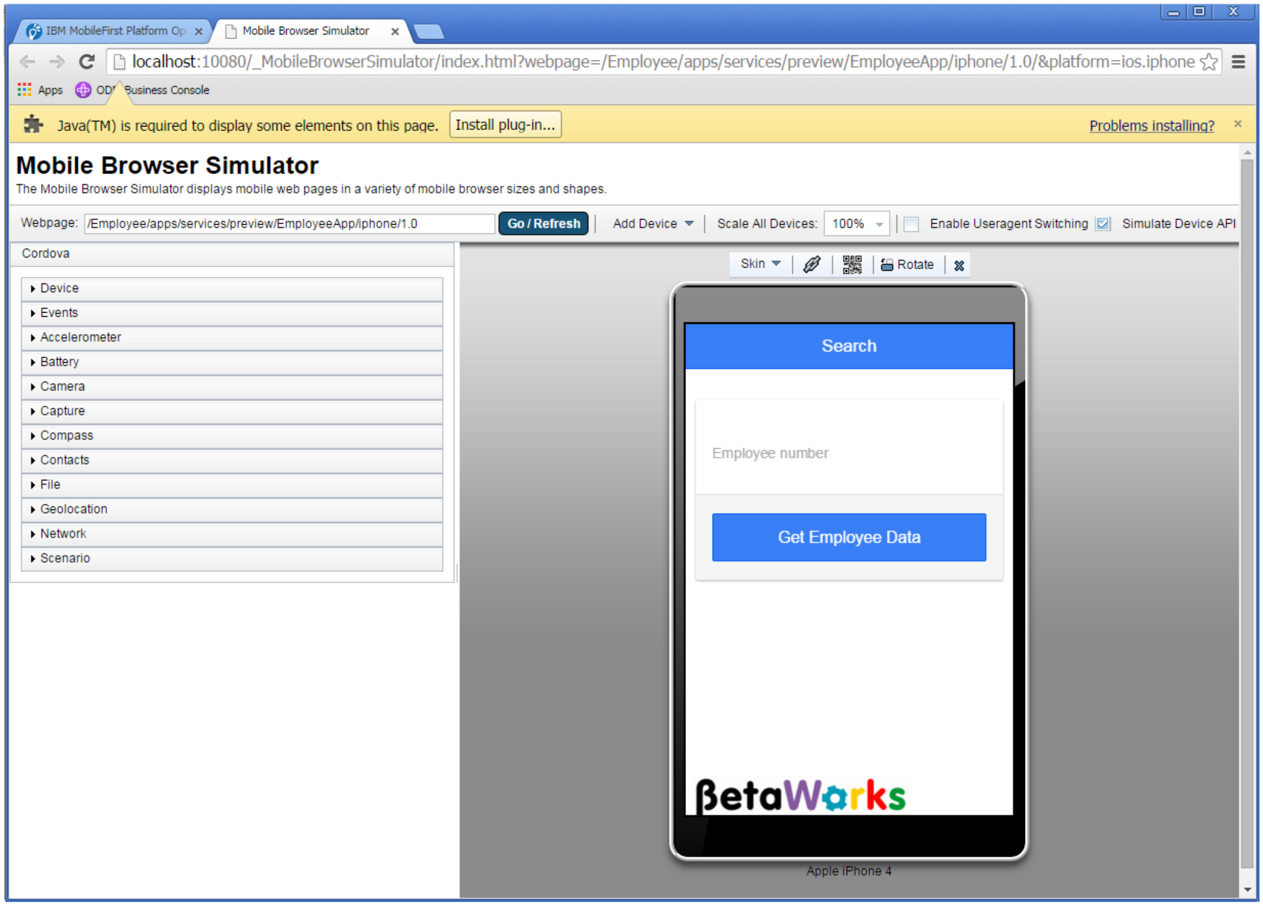
Security Test:	Default
App Authentication:	Disabled
Device Authentication:	Default
User Authentication:	Default
Build Time:	9/26/2016 12:08 PM
Previous Build Time:	9/26/2016 11:55 AM

EmployeeAdapter_local *This adapter uses localhost.*

Last deployed at: 9/26/2016 1:04 PM

6. 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-.



localhost:10080/_MobileBrowserSimulator/index.html?webpage=/Employee/apps/services/preview/EmployeeApp/iphone/1.0/&platform=ios.iphone

Java(TM) is required to display some elements on this page. Install plug-in... Problems installing?

Mobile Browser Simulator

The Mobile Browser Simulator displays mobile web pages in a variety of mobile browser sizes and shapes.

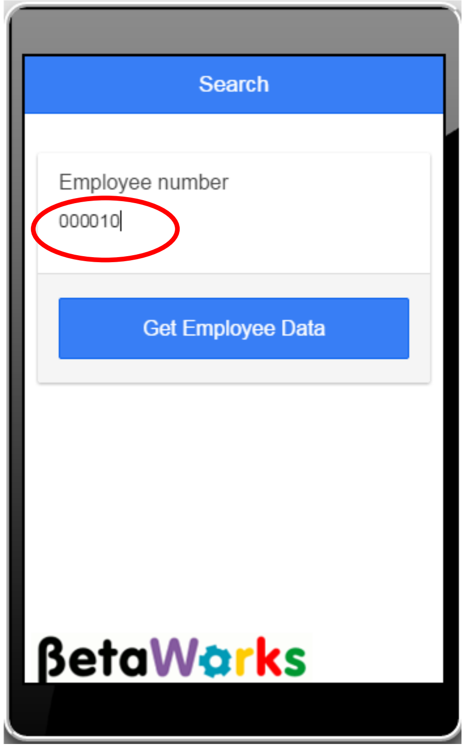
Webpage: /Employee/apps/services/preview/EmployeeApp/iphone/1.0

Scale All Devices: 100%

☐ Enable Useragent Switching ☑ Simulate Device API

7. You are now presented with the first screen of the Mobile App

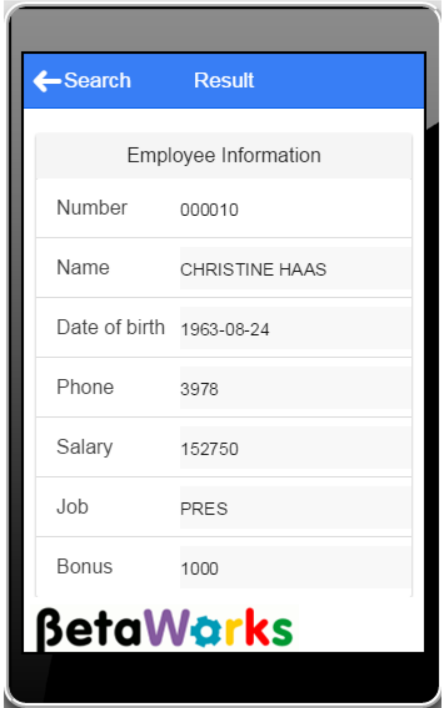
   Enter 000010 as an Employee Number and press (click) ‘Get Employee Data’:



Employee number

000010

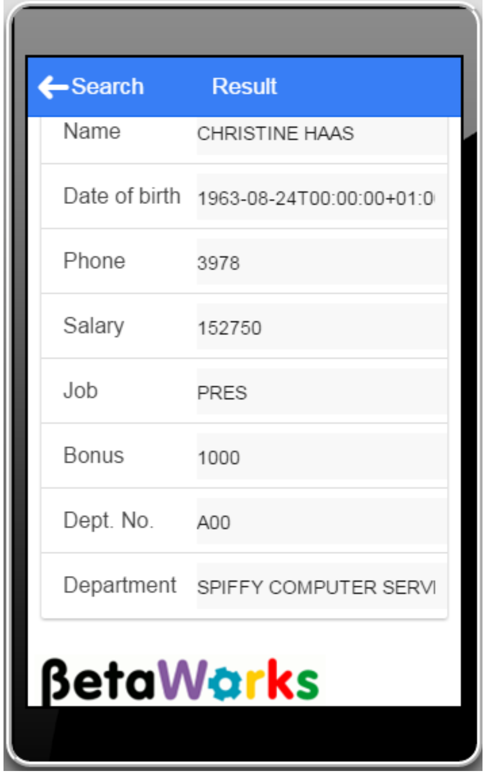
8. The employee data is returned from the HR_Service REST API in IIB.



Employee Information

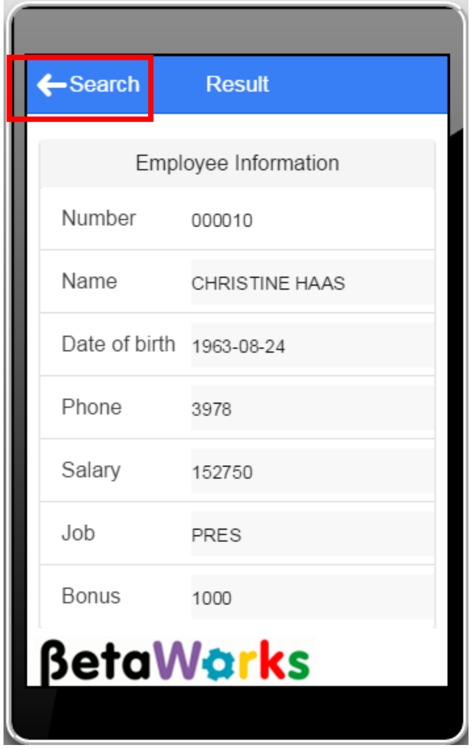
Number	000010
Name	CHRISTINE HAAS
Date of birth	1963-08-24
Phone	3978
Salary	152750
Job	PRES
Bonus	1000

9. Using the down arrow, or the mouse or the mouse scrolling wheel, will move the display to the bottom, where you will see the additional department data that was returned.



Name	CHRISTINE HAAS
Date of birth	1963-08-24T00:00:00+01:0
Phone	3978
Salary	152750
Job	PRES
Bonus	1000
Dept. No.	A00
Department	SPIFFY COMPUTER SERVI

10. In the Mobile Browser Simulator, click the ‘Search’ button to return to the mobile view to search for another employee:



Employee Information

Number	000010
Name	CHRISTINE HAAS
Date of birth	1963-08-24
Phone	3978
Salary	152750
Job	PRES
Bonus	1000

5. Using adapters for access to IIB using API-Connect

The imported Employee project in the MobileFirst workbench contains three adapters. You have already use the first one of these, EmployeeAdapter_local, which provides direct access to the IIB REST API on the same system (localhost).

The two other adapters are used as follows:

- EmployeeAdapter_SecureGateway – provides direct access to the IBM Secure Gateway connection. This is hosted on IBM Bluemix, and will connect through to the required backend IIB system.
- EmployeeAdapter_APIConnect – provides access to a nominated API Connect service running on IBM Bluemix. In turn, this will use the Secure Gateway connection to execute the REST API on the backend IIB system.

Note that both of these adapters assume that the Secure Gateway and API Connect configurations for IIB have been done. This is fully described in the Lab Guide **"IIB with API Connect" (10006_16L07)**, and this lab should be performed in its entirety, before proceeding with the remainder of this lab.

Important note

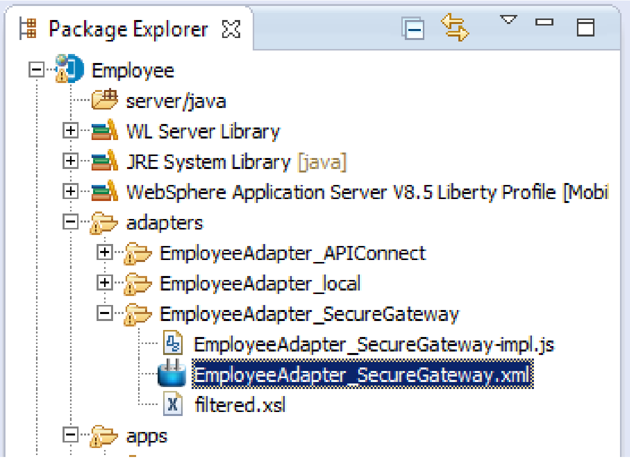
This MobileFirst lab uses the getDetails operation of the HR_Service REST API. Since the "Integration with API Connect" lab uses the getEmployee operation, you will need to follow the deployment and runtime steps in the "Integration with API Connect" lab document, but substitute the operation name as appropriate.

Use the version of HR_Service that is supplied for the MobileFirst lab.

5.1 The Secure Gateway adapter

5.1.1 Explore and deploy the MobileFirst SecureGateway adapter

1. Expand the EmployeeAdapter_SecureGateway adapter, and open the ".xml" configuration file.



Package Explorer

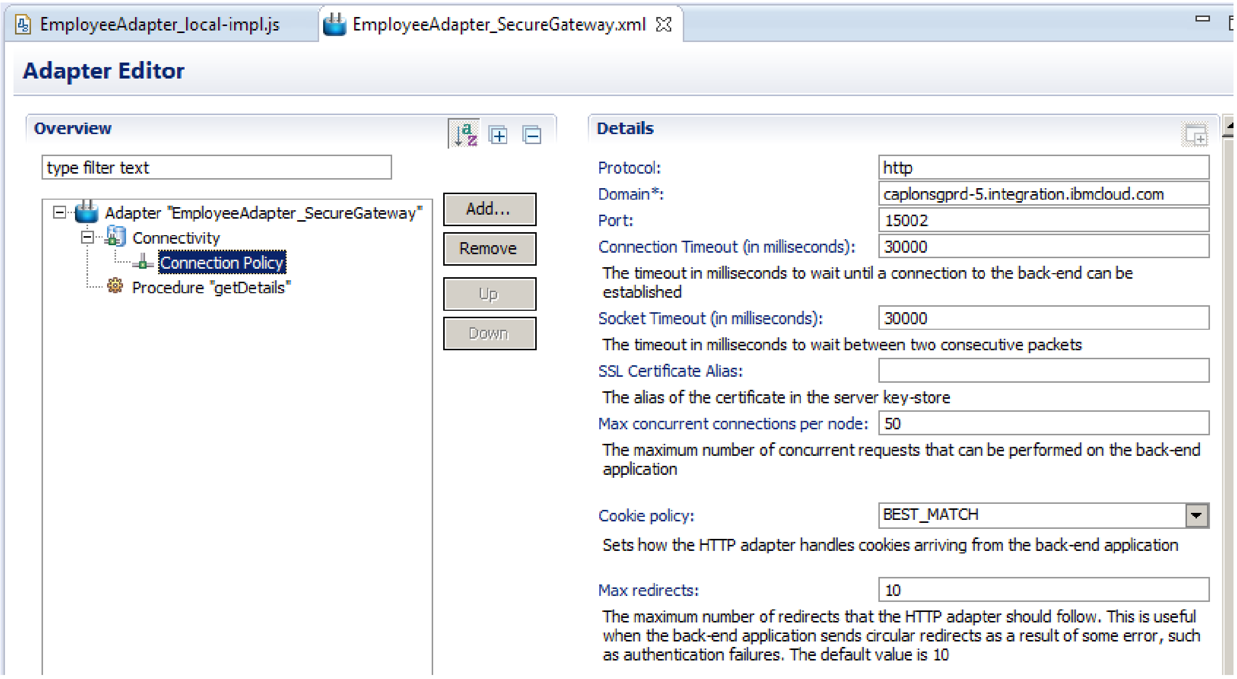
- Employee
  - server/java
  - WL Server Library
  - JRE System Library [java]
  - WebSphere Application Server V8.5 Liberty Profile [Mobi
  - adapters
    - EmployeeAdapter_APIConnect
    - EmployeeAdapter_local
    - EmployeeAdapter_SecureGateway
      - EmployeeAdapter_SecureGateway-impl.js
      - EmployeeAdapter_SecureGateway.xml
      - filtered.xsl
  - apps

2. In the Connectivity Policy, note that the following values have been set:

- Domain – caplonsprd5-5.integration.ibmcloud.com
- Port - 15002

These are the values that the Bluemix Secure Gateway uses in the test scenario for this lab guide. You will need to check the values for your own configuration, and make appropriate changes to this MobileFirst adapter. Refer to the corresponding lab guide for the IIB API Connect scenario.

If you do need to make changes, save them with Ctrl-S.



EmployeeAdapter_local-impl.js | EmployeeAdapter_SecureGateway.xml

Adapter Editor

Overview

type filter text

- Adapter "EmployeeAdapter_SecureGateway"
  - Connectivity
    - Connection Policy
  - Procedure "getDetails"

Add... | Remove | Up | Down

Details

Protocol: http

Domain*: caplonsgprd-5.integration.ibmcloud.com

Port: 15002

Connection Timeout (in milliseconds): 30000

The timeout in milliseconds to wait until a connection to the back-end can be established

Socket Timeout (in milliseconds): 30000

The timeout in milliseconds to wait between two consecutive packets

SSL Certificate Alias:

The alias of the certificate in the server key-store

Max concurrent connections per node: 50

The maximum number of concurrent requests that can be performed on the back-end application

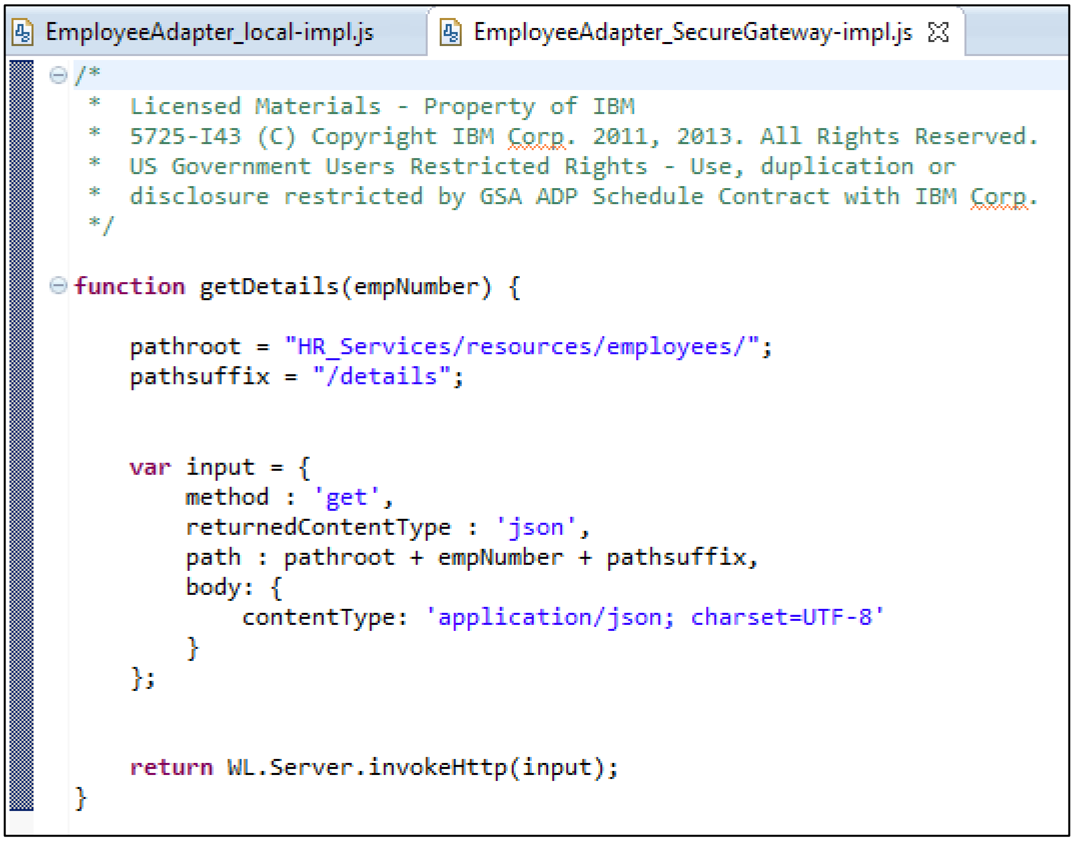
Cookie policy: BEST_MATCH

Sets how the HTTP adapter handles cookies arriving from the back-end application

Max redirects: 10

The maximum number of redirects that the HTTP adapter should follow. This is useful when the back-end application sends circular redirects as a result of some error, such as authentication failures. The default value is 10

3. As a reminder, review the adapter javascript implementation. This is the same javascript logic as the local adapter. Recall that the full path statement uses a concatenation, because the MobileFirst wizard does not allow for path suffix components.


EmployeeAdapter_local-impl.js
EmployeeAdapter_SecureGateway-impl.js

```
/*
 *  Licensed Materials - Property of IBM
 *  5725-I43 (C) Copyright IBM Corp. 2011, 2013. All Rights Reserved.
 *  US Government Users Restricted Rights - Use, duplication or
 *  disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
 */

function getDetails(empNumber) {

    pathroot = "HR_Services/resources/employees/";
    pathsuffix = "/details";


    var input = {
        method : 'get',
        returnedContentType : 'json',
        path : pathroot + empNumber + pathsuffix,
        body: {
            contentType: 'application/json; charset=UTF-8'
        }
    };


    return WL.Server.invokeHttp(input);
}
```

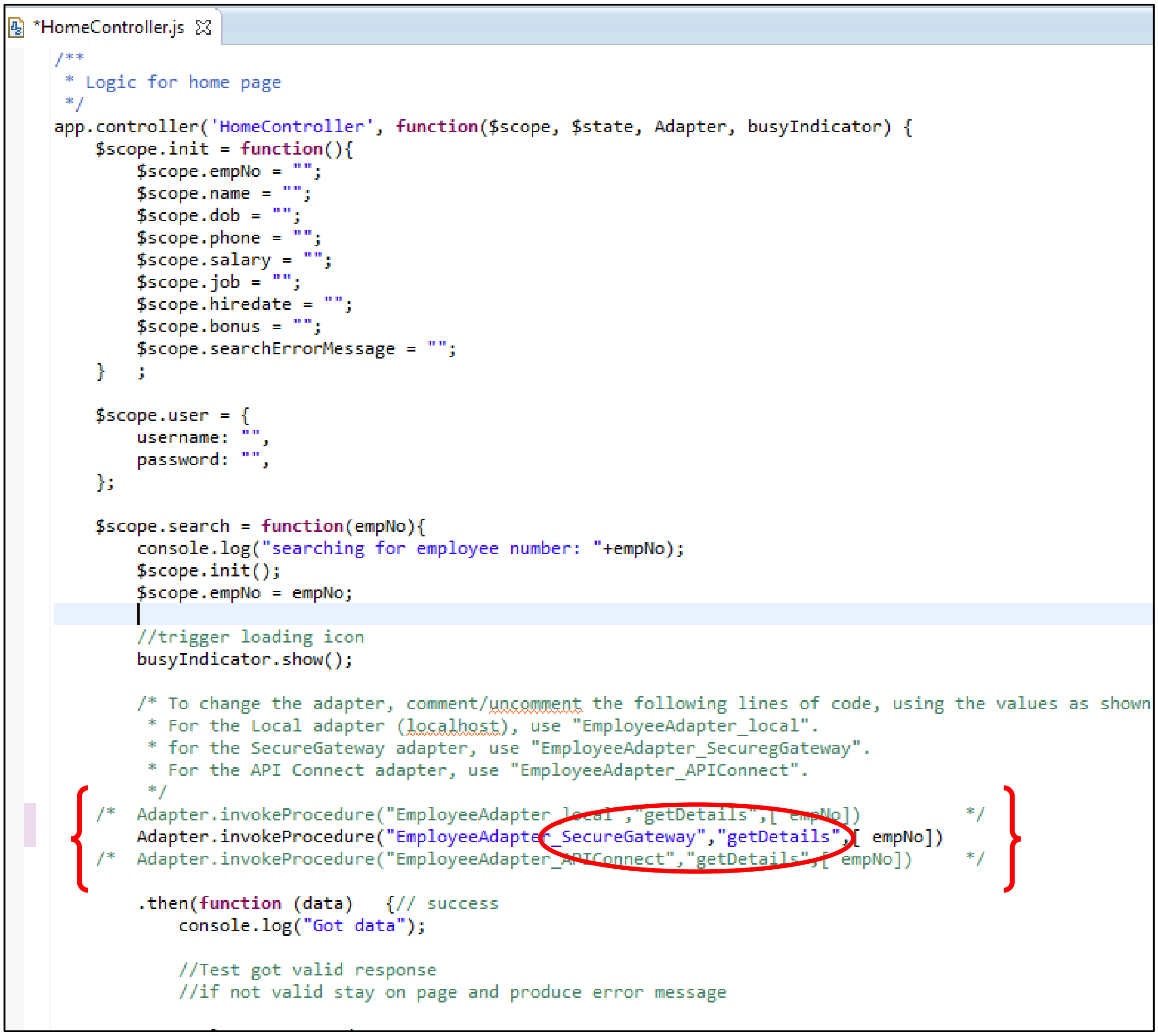
data':

4. The Employee app must be changed to use the Secure Gateway adapter.

   In the Employee project, expand apps/Employee/common/js/controller, and open HomeController.js.

   Change the Adapter.invokeProcedure line to set the name of the required adapter to EmployeeAdapter_SecureGateway, as described in the comments below.

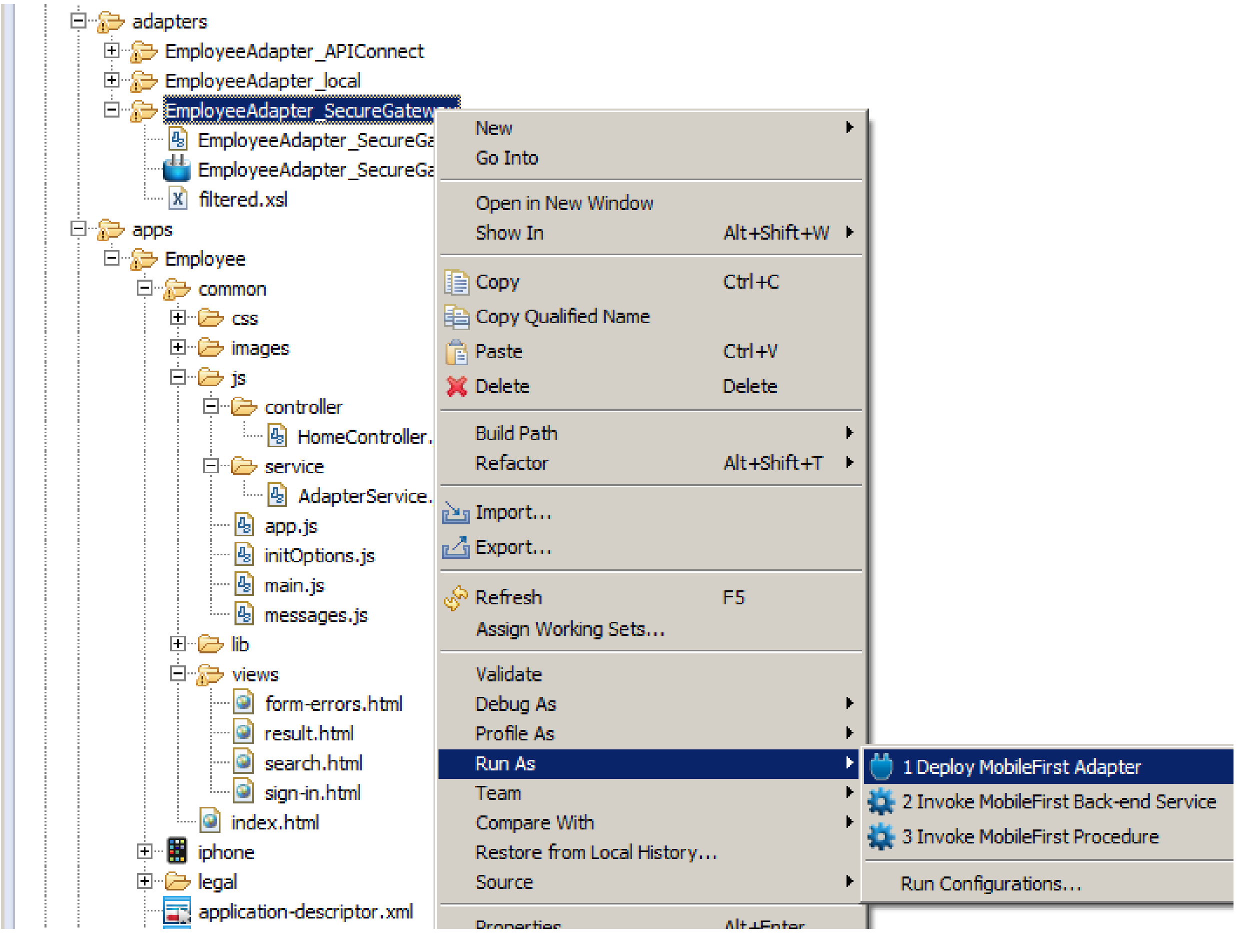
   Save the changes to the HomeController.js file.



```
/**
 * Logic for home page
 */
app.controller('HomeController', function($scope, $state, Adapter, busyIndicator) {
    $scope.init = function(){
        $scope.empNo = "";
        $scope.name = "";
        $scope.dob = "";
        $scope.phone = "";
        $scope.salary = "";
        $scope.job = "";
        $scope.hiredate = "";
        $scope.bonus = "";
        $scope.searchErrorMessage = "";
    }   ;

    $scope.user = {
        username: "",
        password: "",
    };

    $scope.search = function(empNo){
        console.log("searching for employee number: "+empNo);
        $scope.init();
        $scope.empNo = empNo;

        //trigger loading icon
        busyIndicator.show();

        /* To change the adapter, comment/uncomment the following lines of code, using the values as shown
         * For the Local adapter (localhost), use "EmployeeAdapter_local".
         * for the SecureGateway adapter, use "EmployeeAdapter_SecuregGateway".
         * For the API Connect adapter, use "EmployeeAdapter_APIConnect".
         */
    /*  Adapter.invokeProcedure("EmployeeAdapter_local","getDetails",[ empNo])          */
        Adapter.invokeProcedure("EmployeeAdapter_SecureGateway","getDetails",[ empNo])
    /*  Adapter.invokeProcedure("EmployeeAdapter_APIConnect","getDetails",[ empNo])     */

        .then(function (data)   {// success
            console.log("Got data");

            //Test got valid response
            //if not valid stay on page and produce error message
```

5. Deploy the Secure Gateway Adapter.

   Right-click the EmployeeAdapter_SecureGateway, and select **Run As, Deploy MobileFirst Adapter**.

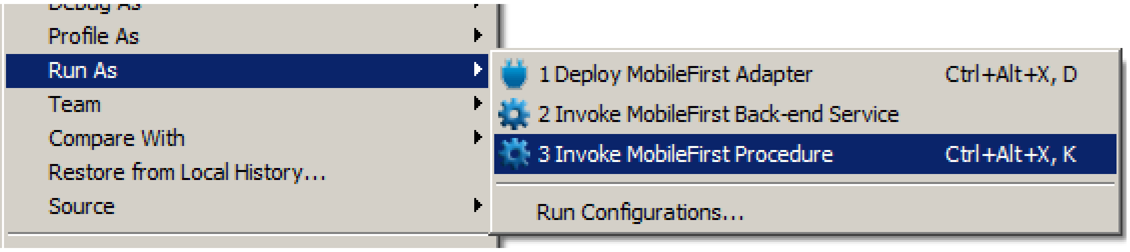


6. **The following steps assume that the Bluemix Secure Gateway Client has been installed and started, and is connected to the Bluemix Secure Gateway server. Depending on how you have installed and configured this component, you may need to start this service with a user that has Windows administrator privileges.**

   For details, refer to the IIB API Connect lab (16L07_10006_Integration_API_Connect).

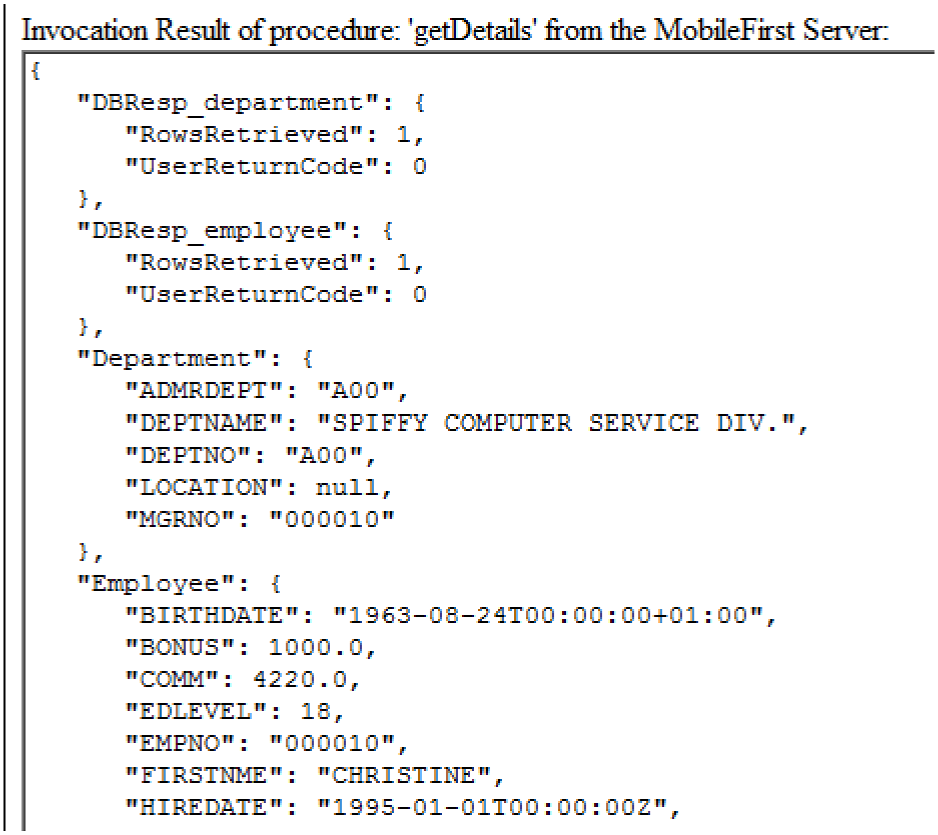
   When the Secure Gateway Client is started, switch back to the user that you are using for this lab, and proceed with the next step.

7. As before, perform a test of the adapter, Right-click the adapter, and select **Run As, Invoke ….**


Debug As
Profile As
Run As
Team
Compare With
Restore from Local History...
Source
1 Deploy MobileFirst Adapter Ctrl+Alt+X, D
2 Invoke MobileFirst Back-end Service
3 Invoke MobileFirst Procedure Ctrl+Alt+X, K
Run Configurations...

Provide input data: "000010".

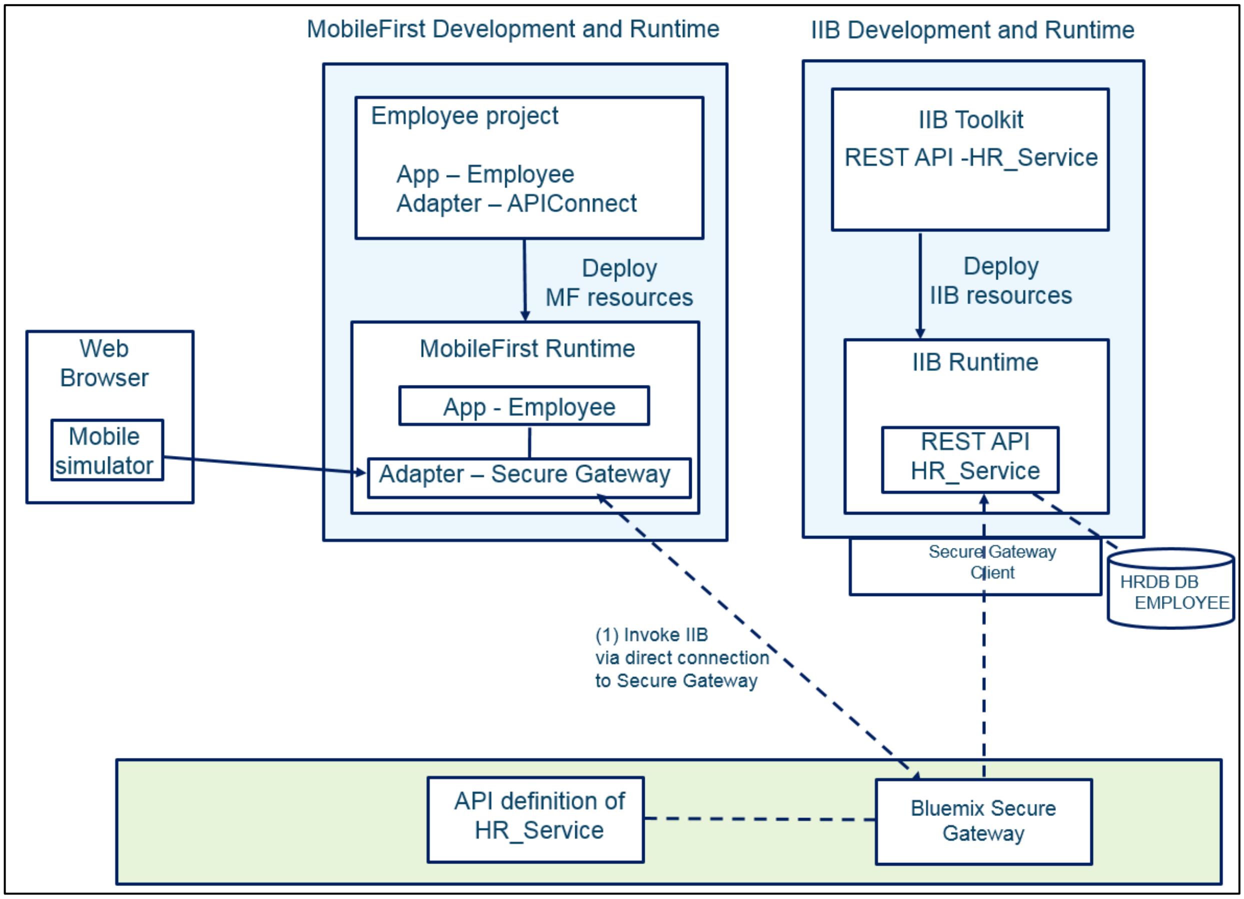
If successful, the returned data will be presented through a new browser window.



Invocation Result of procedure: 'getDetails' from the MobileFirst Server:

```
{
   "DBResp_department": {
      "RowsRetrieved": 1,
      "UserReturnCode": 0
   },
   "DBResp_employee": {
      "RowsRetrieved": 1,
      "UserReturnCode": 0
   },
   "Department": {
      "ADMRDEPT": "A00",
      "DEPTNAME": "SPIFFY COMPUTER SERVICE DIV.",
      "DEPTNO": "A00",
      "LOCATION": null,
      "MGRNO": "000010"
   },
   "Employee": {
      "BIRTHDATE": "1963-08-24T00:00:00+01:00",
      "BONUS": 1000.0,
      "COMM": 4220.0,
      "EDLEVEL": 18,
      "EMPNO": "000010",
      "FIRSTNME": "CHRISTINE",
      "HIREDATE": "1995-01-01T00:00:00Z",
```

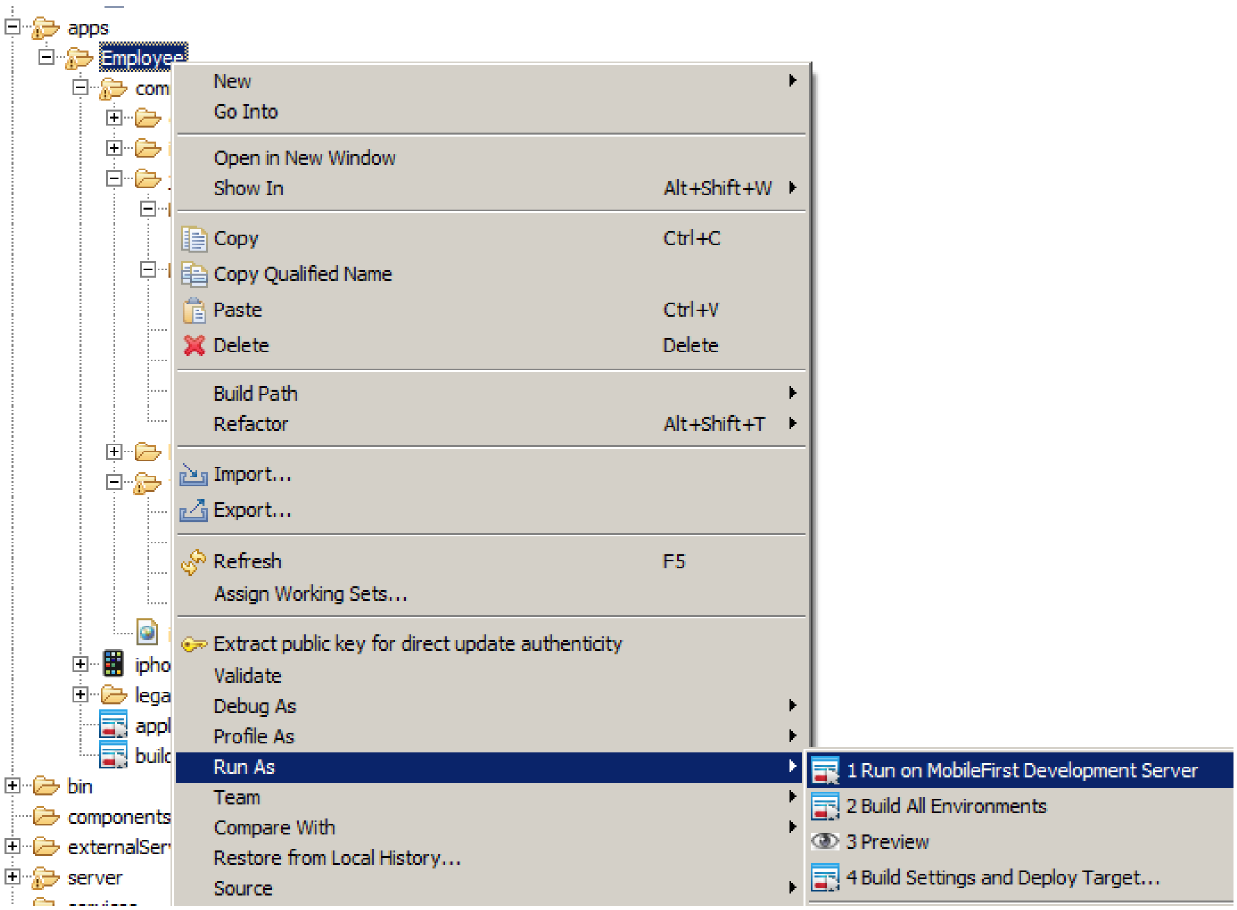
8. As a reminder, this schematic shows the various components, and how the MobileFirst Secure Gateway adapter has been invoked.



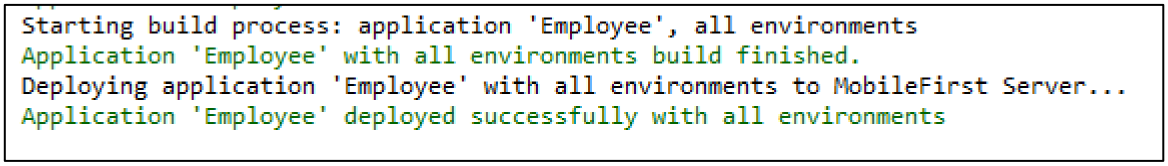
5.1.2 Deploy and test Secure Gateway Adapter with MobileFirst app

1. Return to the MobileFirst Studio, and deploy the updated Employee app.

   Right-click on the Employee app, and select **Run As, Run on MobileFirst Development Server.**



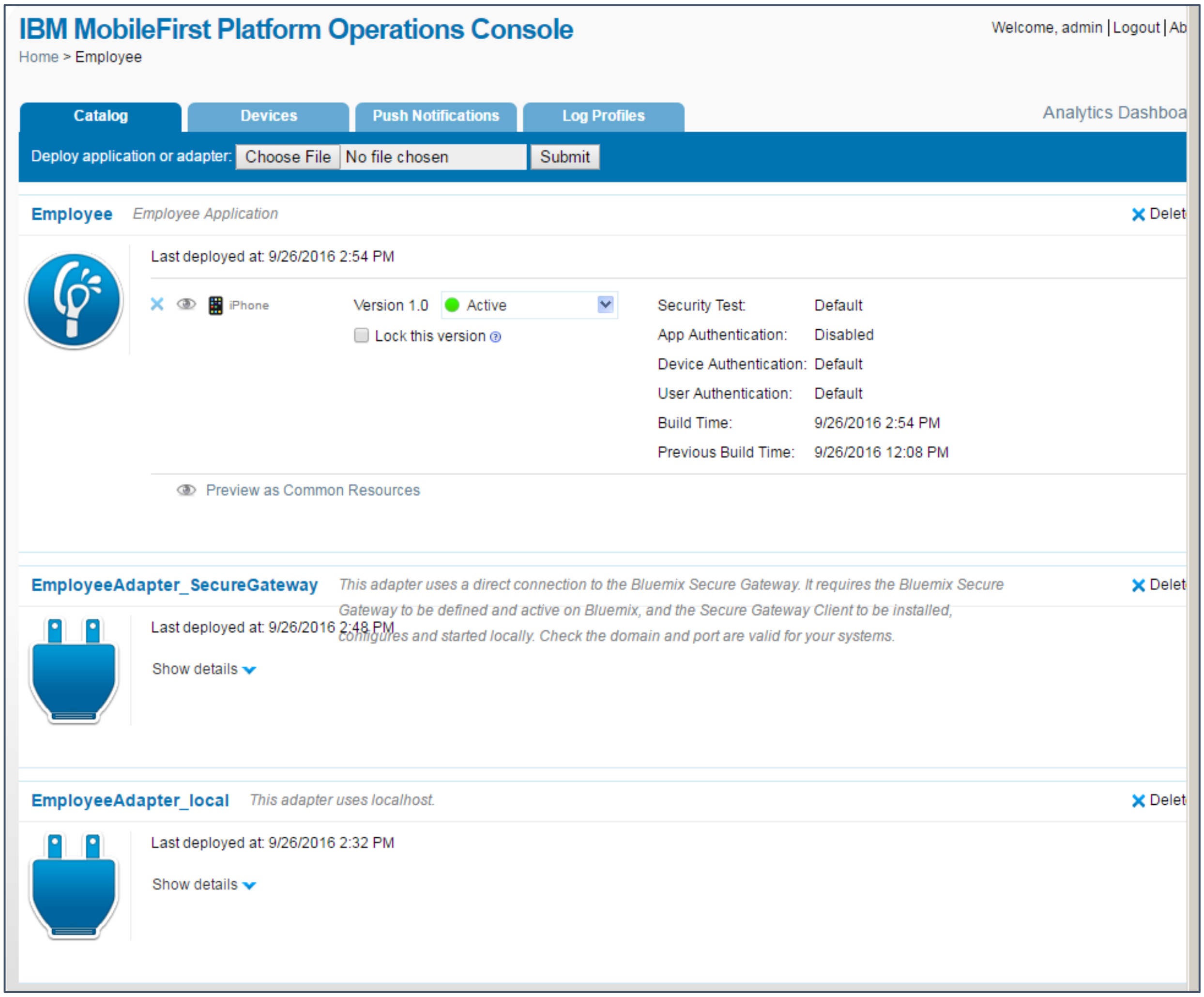
When finished, in the Console you will see



```
Starting build process: application 'Employee', all environments
Application 'Employee' with all environments build finished.
Deploying application 'Employee' with all environments to MobileFirst Server...
Application 'Employee' deployed successfully with all environments
```

2. In the Chrome browser, in the MobileFirst Operations Console, press F5 to refresh the operations console.

   You will see the newly deployed Secure Gateway adapter, and the Employee app should have a new deployment time.



IBM MobileFirst Platform Operations Console

Welcome, admin | Logout | Ab

Home > Employee

Deploy application or adapter: Choose File No file chosen Submit

Employee *Employee Application* ✕ Delet

Last deployed at: 9/26/2016 2:54 PM

iPhone Version 1.0 ● Active

□ Lock this version

Security Test:	Default
App Authentication:	Disabled
Device Authentication:	Default
User Authentication:	Default
Build Time:	9/26/2016 2:54 PM
Previous Build Time:	9/26/2016 12:08 PM

Preview as Common Resources

EmployeeAdapter_SecureGateway *This adapter uses a direct connection to the Bluemix Secure Gateway. It requires the Bluemix Secure Gateway to be defined and active on Bluemix, and the Secure Gateway Client to be installed, configures and started locally. Check the domain and port are valid for your systems.* ✕ Delet

Last deployed at: 9/26/2016 2:48 PM

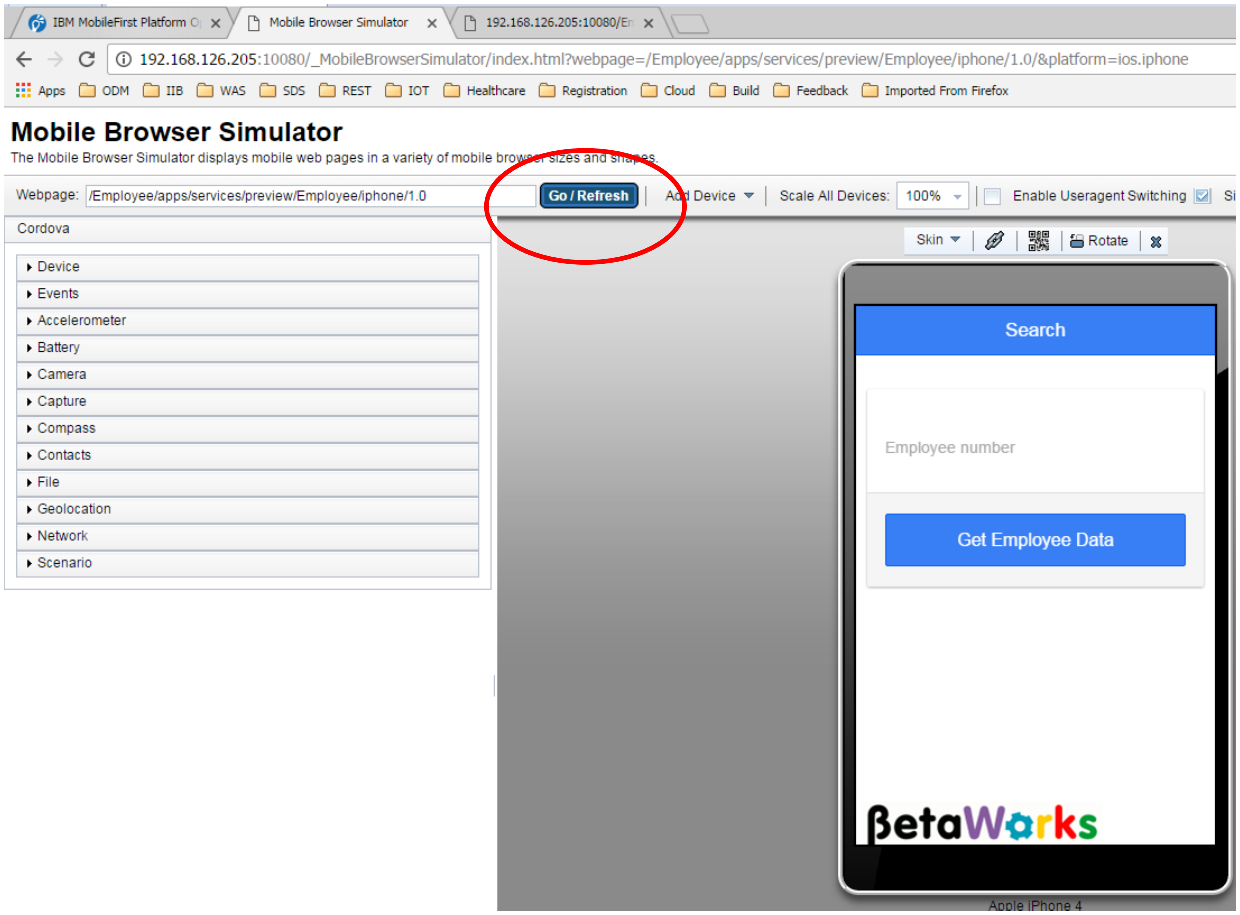
Show details

EmployeeAdapter_local *This adapter uses localhost.* ✕ Delet

Last deployed at: 9/26/2016 2:32 PM

Show details

3. Switch to the iPhone simulator tab (probably still open from last time), and click the "**Go / Refresh**" button. This is necessary to ensure the simulator reloads correctly.



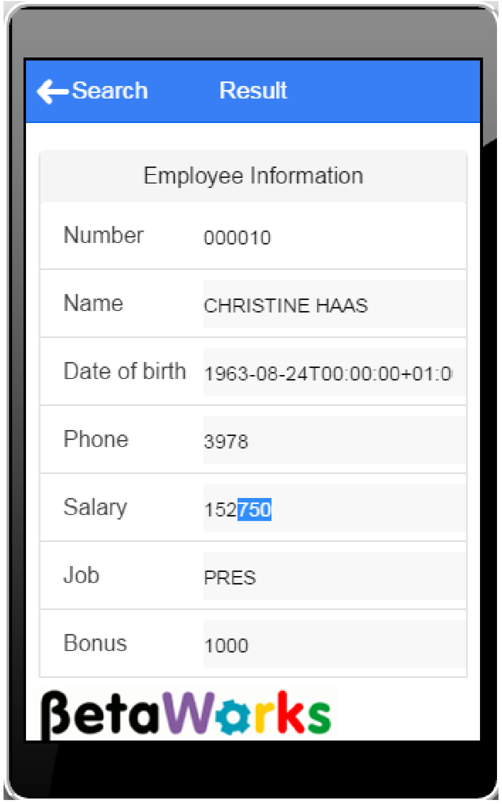
192.168.126.205:10080/_MobileBrowserSimulator/index.html?webpage=/Employee/apps/services/preview/Employee/iphone/1.0/&platform=ios.iphone

Mobile Browser Simulator

The Mobile Browser Simulator displays mobile web pages in a variety of mobile browser sizes and shapes.

Webpage: /Employee/apps/services/preview/Employee/iphone/1.0

4. Provide the employee number 000010, and click Get Employee Data.



Employee Information

Number	000010
Name	CHRISTINE HAAS
Date of birth	1963-08-24T00:00:00+01:0
Phone	3978
Salary	152750
Job	PRES
Bonus	1000

5.2 The API Connect Adapter

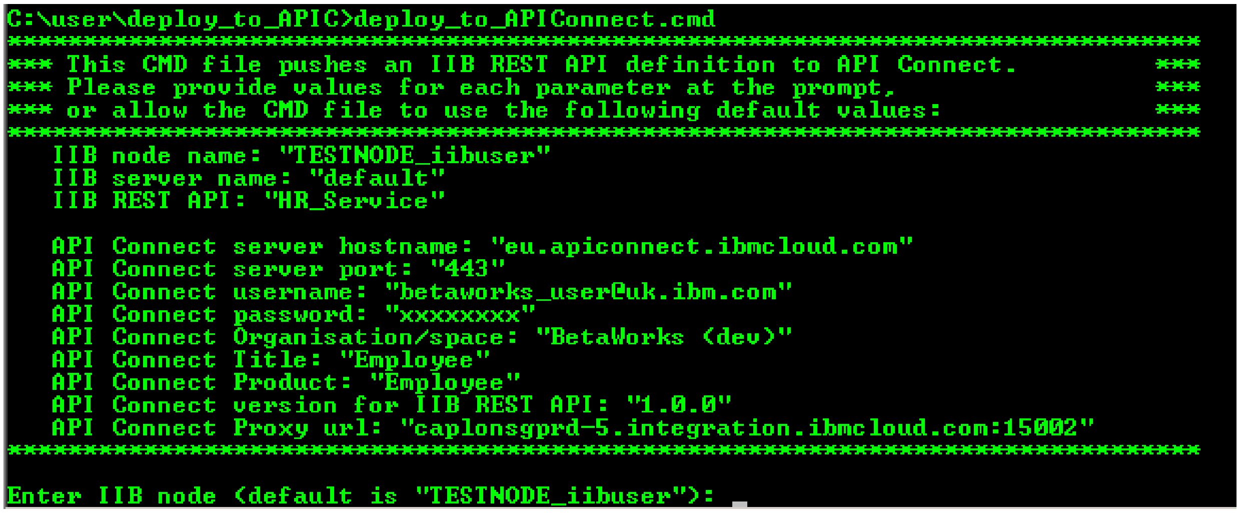
5.2.1 Publish the IIB REST API to API Connect

1. Before running the MobileFirst adapter, you must first publish the REST API from IIB into the API Connect server. You can do this either by using the IIB Web UI tools, or by using the IIB command interface.
   To use the IIB web UI tools, please follow the instructions in the IIB API Connect lab guide (16L07_IIB10006).

   To use the IIB command tools, use the command file **deploy_to_APIConnect.cmd**, supplied in the student10/mobile folder:

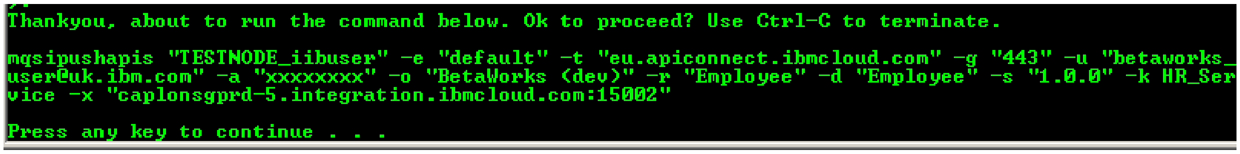
   - In an IIB Command Console, change directory to c:\student10\IIB_APIC\commands.

   - Run the command file **deploy_to_APIConnect.cmd**. Default values are provided for most of the required parameters, appropriate for this lab scenario. You can override these using the prompts during execution, or you can permanently override the defaults by editing the cmd file.



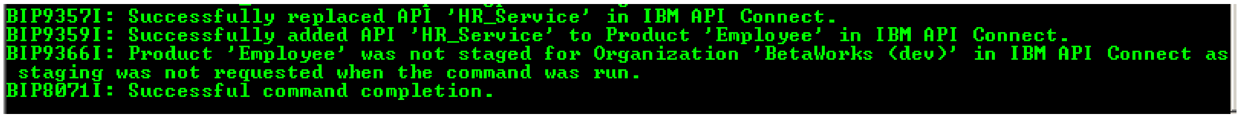
```
C:\user\deploy_to_APIC>deploy_to_APIConnect.cmd
*******************************************************************************
*** This CMD file pushes an IIB REST API definition to API Connect.         ***
*** Please provide values for each parameter at the prompt,                 ***
*** or allow the CMD file to use the following default values:              ***
*******************************************************************************
   IIB node name: "TESTNODE_iibuser"
   IIB server name: "default"
   IIB REST API: "HR_Service"

   API Connect server hostname: "eu.apiconnect.ibmcloud.com"
   API Connect server port: "443"
   API Connect username: "betaworks_user@uk.ibm.com"
   API Connect password: "xxxxxxxx"
   API Connect Organisation/space: "BetaWorks (dev)"
   API Connect Title: "Employee"
   API Connect Product: "Employee"
   API Connect version for IIB REST API: "1.0.0"
   API Connect Proxy url: "caplonsgprd-5.integration.ibmcloud.com:15002"
*******************************************************************************

Enter IIB node (default is "TESTNODE_iibuser"): _
```

   When all parameters have been provided, the final "mqsipushapis" command to be executed will be shown. Check this carefully, and if correct, press Return to push the API from IIB into API Connect.



```
Thankyou, about to run the command below. Ok to proceed? Use Ctrl-C to terminate.

mqsipushapis "TESTNODE_iibuser" -e "default" -t "eu.apiconnect.ibmcloud.com" -g "443" -u "betaworks_
user@uk.ibm.com" -a "xxxxxxxx" -o "BetaWorks (dev)" -r "Employee" -d "Employee" -s "1.0.0" -k HR_Ser
vice -x "caplonsgprd-5.integration.ibmcloud.com:15002"

Press any key to continue . . .
```

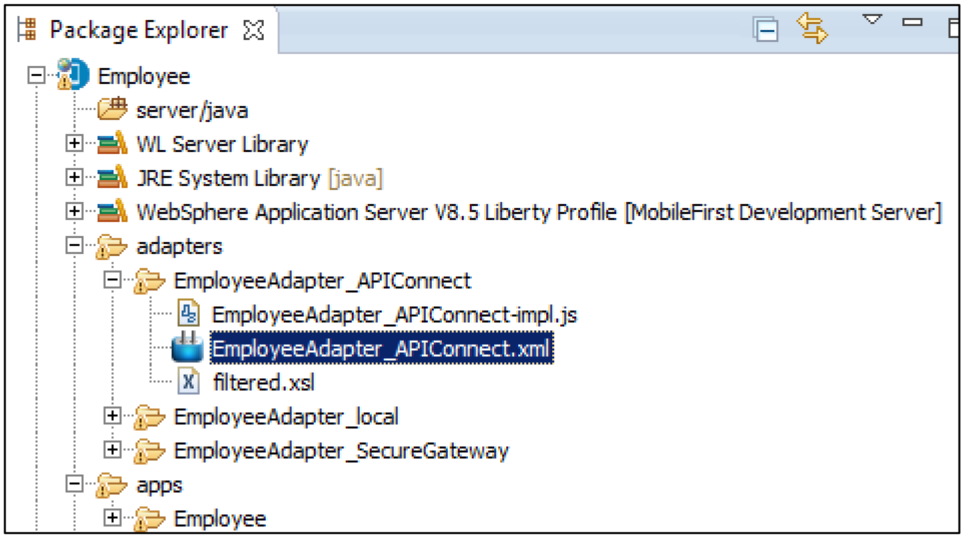
   When the command has run, confirmation of the successful command will be shown.



```
BIP9357I: Successfully replaced API 'HR_Service' in IBM API Connect.
BIP9359I: Successfully added API 'HR_Service' to Product 'Employee' in IBM API Connect.
BIP9366I: Product 'Employee' was not staged for Organization 'BetaWorks (dev)' in IBM API Connect as
 staging was not requested when the command was run.
BIP8071I: Successful command completion.
```

5.2.2 Explore and deploy the MobileFirst APIConnect adapter

1. Expand the EmployeeAdapter_APIConnect adapter, and open the ".xml" configuration file.

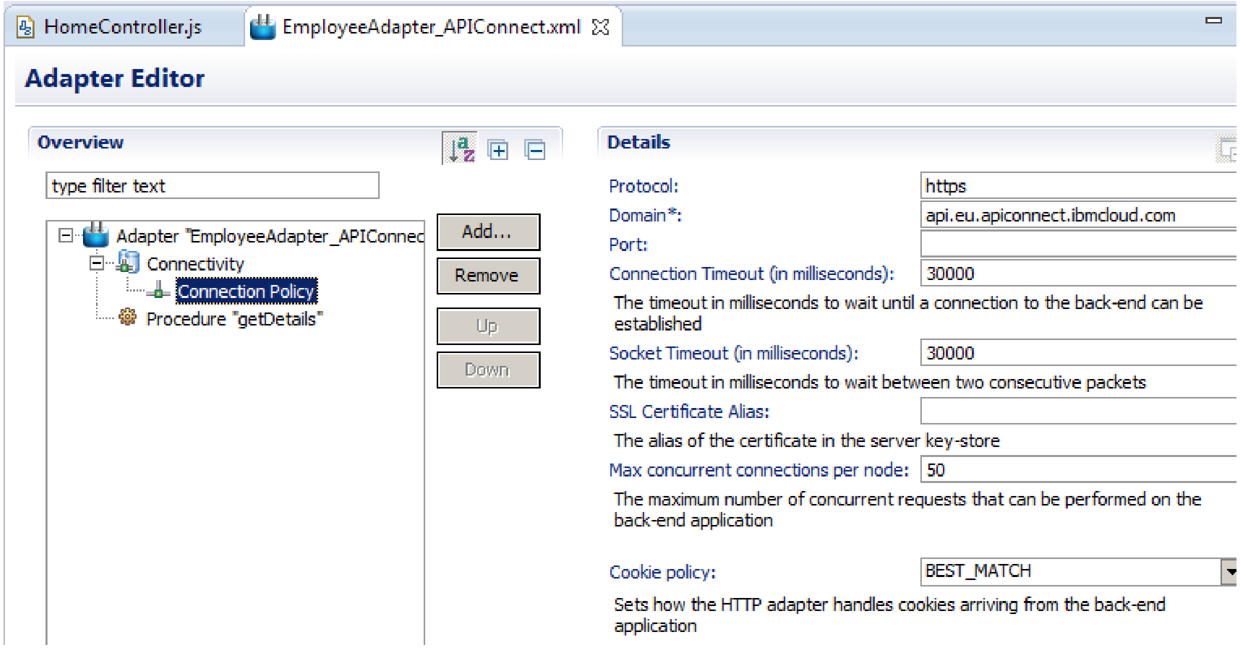

Package Explorer
Employee
server/java
WL Server Library
JRE System Library [java]
WebSphere Application Server V8.5 Liberty Profile [MobileFirst Development Server]
adapters
EmployeeAdapter_APIConnect
EmployeeAdapter_APIConnect-impl.js
EmployeeAdapter_APIConnect.xml
filtered.xsl
EmployeeAdapter_local
EmployeeAdapter_SecureGateway
apps
Employee

2. In the Connectivity Policy, note that the following values have been set:

   - Protocol: https
   - Domain: api.eu.apiconnect.ibmcloud.com
   - Port: left blank

   These are the values that the Bluemix API Connect system used when this lab guide document was being developed. You will need to check the values for your own API Connect configuration, and make appropriate changes to this MobileFirst adapter. Refer to the corresponding lab guide for the IIB API Connect scenario (10006_16L07).

   When you have made these changes, save them with Ctrl-S.



Protocol: https

Domain*: api.eu.apiconnect.ibmcloud.com

Port:

Connection Timeout (in milliseconds): 30000

The timeout in milliseconds to wait until a connection to the back-end can be established

Socket Timeout (in milliseconds): 30000

The timeout in milliseconds to wait between two consecutive packets

SSL Certificate Alias:

The alias of the certificate in the server key-store

Max concurrent connections per node: 50

The maximum number of concurrent requests that can be performed on the back-end application

Cookie policy: BEST_MATCH

Sets how the HTTP adapter handles cookies arriving from the back-end application

3. Review the javascript implementation for the APIConnect adapter. In this case, additional logic has to be provided to add an additional element to the URL. This is because API Connect augments the URL of the backend system with path elements that represent the Bluemix Organization, Space and Bluemix runtime environment.

   In the case shown here, API Connect has added “**betaworks-dev/sb**” to the URL. This is derived from our Bluemix organization (betaworks), our Bluemix space (dev), and the API Connect Sandbox runtime system (sb).

   You will need to make corresponding changes to this implementation script to take account of your own API Connect system.

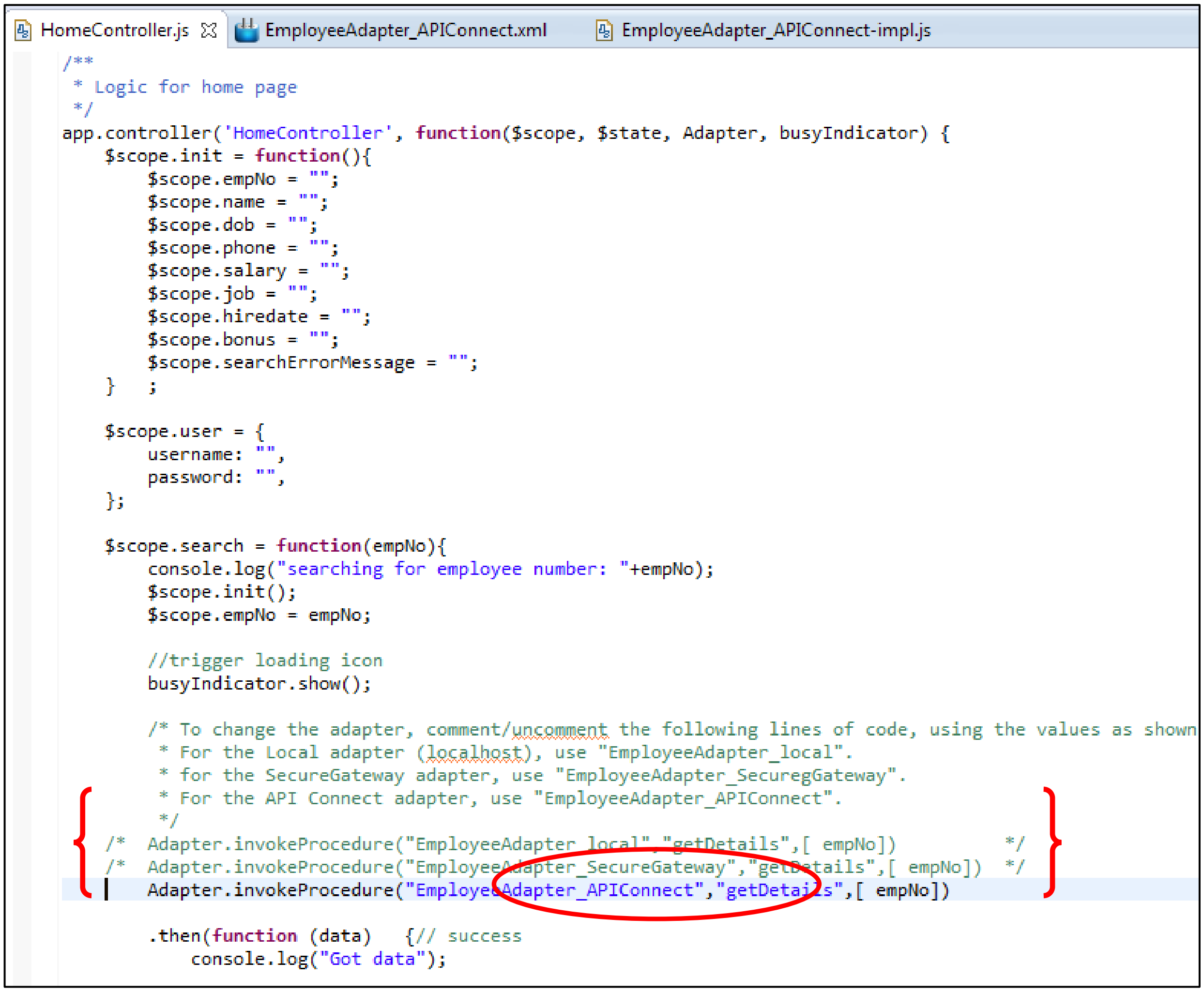
```
function getDetails(empNumber) {

/* Because API Connect adds an additional part to the URL path, this needs
to be manually added here (because MobileFirst does not permit this to be
provided through the config panel options).

For this scenario, using the BetaWorks login details for API Connect, this
string is "betaworks-dev/sb". Other API Connect systems will be different,
so this code will have to be changed accordingly.
The remainder of the path is the same one that is used in all of the
adapters that invokes HR_Service on IIB.
*/


      pathroot = "betaworks-dev/sb/HR_Services/resources/employees/";
      pathsuffix = "/details";

      var input = {
          method : 'get',
          returnedContentType : 'json',
          path : pathroot + empNumber + pathsuffix,
          body: {
            contentType: 'application/json; charset=UTF-8'
          }
      };
```

4. The Employee app must be changed to use the EmployeeAdapter_APIConnect adapter.

   In the Employee project, expand apps/Employee/common/js/controller, and open HomeController.js.

   Change the Adapter.invokeProcedure line to set the name of the required adapter to EmployeeAdapter_APIConnect, as described in the comments below.

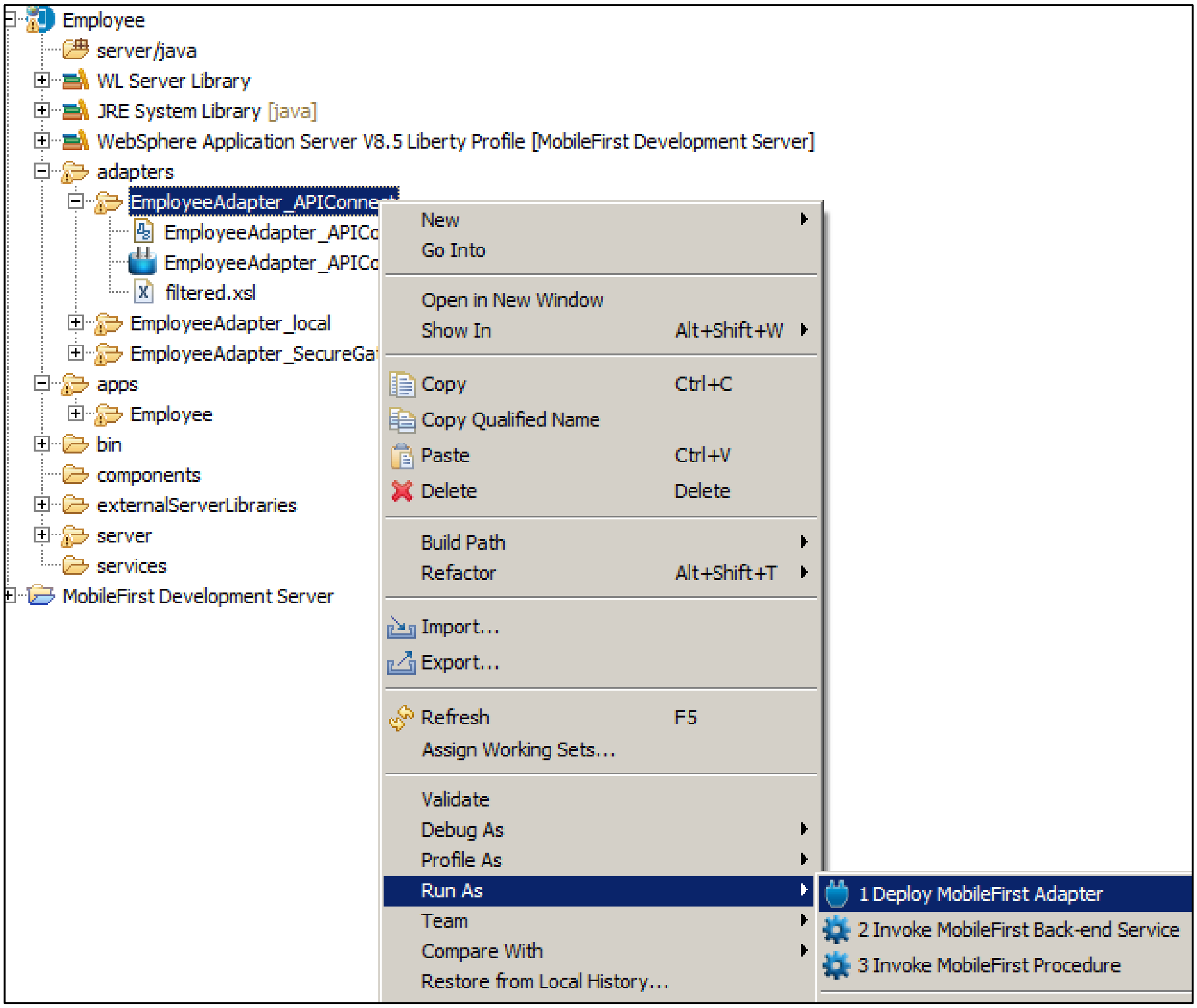
   Save the changes to the HomeController.js file.


HomeController.js
EmployeeAdapter_APIConnect.xml
EmployeeAdapter_APIConnect-impl.js

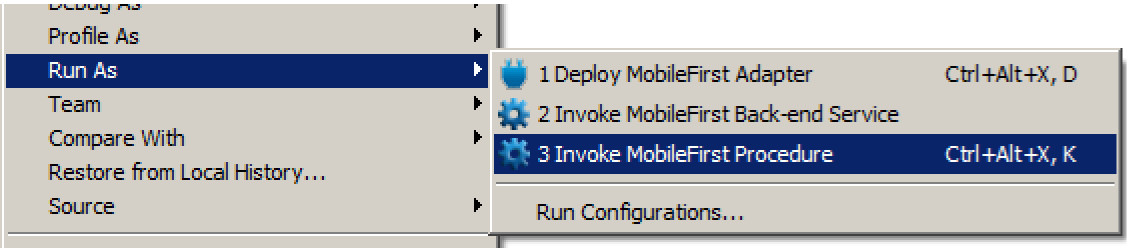
```
/**
 * Logic for home page
 */
app.controller('HomeController', function($scope, $state, Adapter, busyIndicator) {
    $scope.init = function(){
        $scope.empNo = "";
        $scope.name = "";
        $scope.dob = "";
        $scope.phone = "";
        $scope.salary = "";
        $scope.job = "";
        $scope.hiredate = "";
        $scope.bonus = "";
        $scope.searchErrorMessage = "";
    }   ;

    $scope.user = {
        username: "",
        password: "",
    };

    $scope.search = function(empNo){
        console.log("searching for employee number: "+empNo);
        $scope.init();
        $scope.empNo = empNo;

        //trigger loading icon
        busyIndicator.show();

        /* To change the adapter, comment/uncomment the following lines of code, using the values as shown
         * For the Local adapter (localhost), use "EmployeeAdapter_local".
         * for the SecureGateway adapter, use "EmployeeAdapter_SecuregGateway".
         * For the API Connect adapter, use "EmployeeAdapter_APIConnect".
         */
    /*  Adapter.invokeProcedure("EmployeeAdapter_local","getDetails",[ empNo])          */
    /*  Adapter.invokeProcedure("EmployeeAdapter_SecureGateway","getDetails",[ empNo])  */
        Adapter.invokeProcedure("EmployeeAdapter_APIConnect","getDetails",[ empNo])

        .then(function (data)   {// success
            console.log("Got data");
```

5. Deploy the MobileFirst APIConnect Adapter.

Right-click the EmployeeAdapter_APIConnect, and select **Run As, Deploy MobileFirst Adapter**.

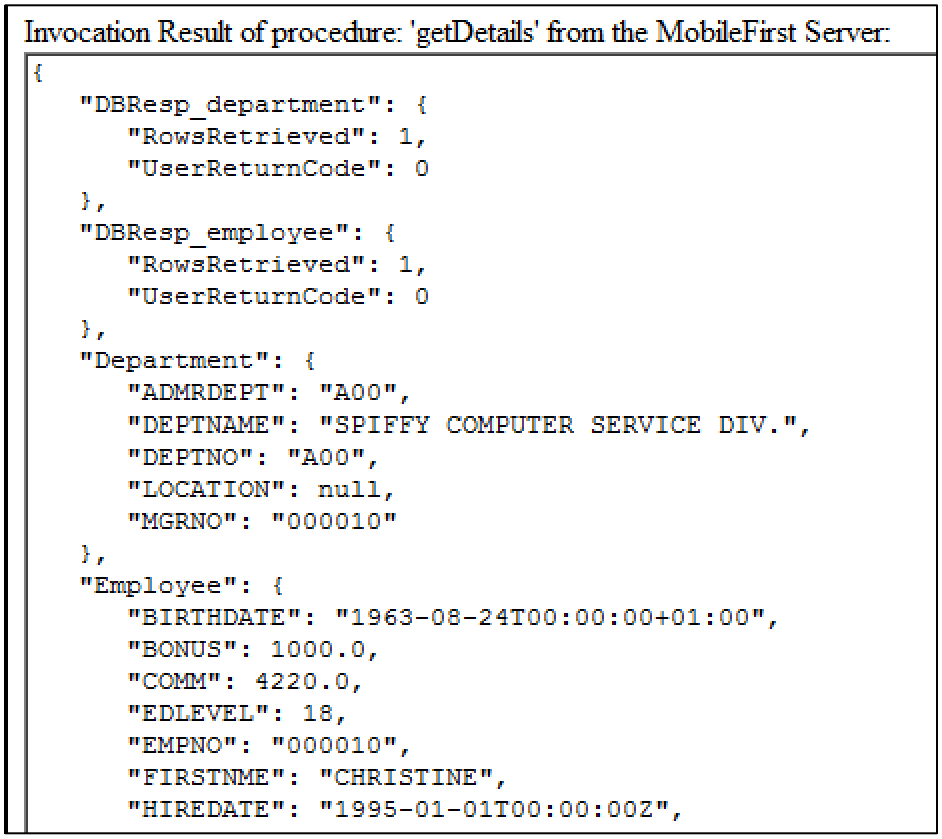


6. As before, perform a test of the adapter. Right-click the adapter, and select **Run As, Invoke ….**


Debug As
Profile As
Run As
Team
Compare With
Restore from Local History…
Source
1 Deploy MobileFirst Adapter Ctrl+Alt+X, D
2 Invoke MobileFirst Back-end Service
3 Invoke MobileFirst Procedure Ctrl+Alt+X, K
Run Configurations…

Provide input data: “000010”.

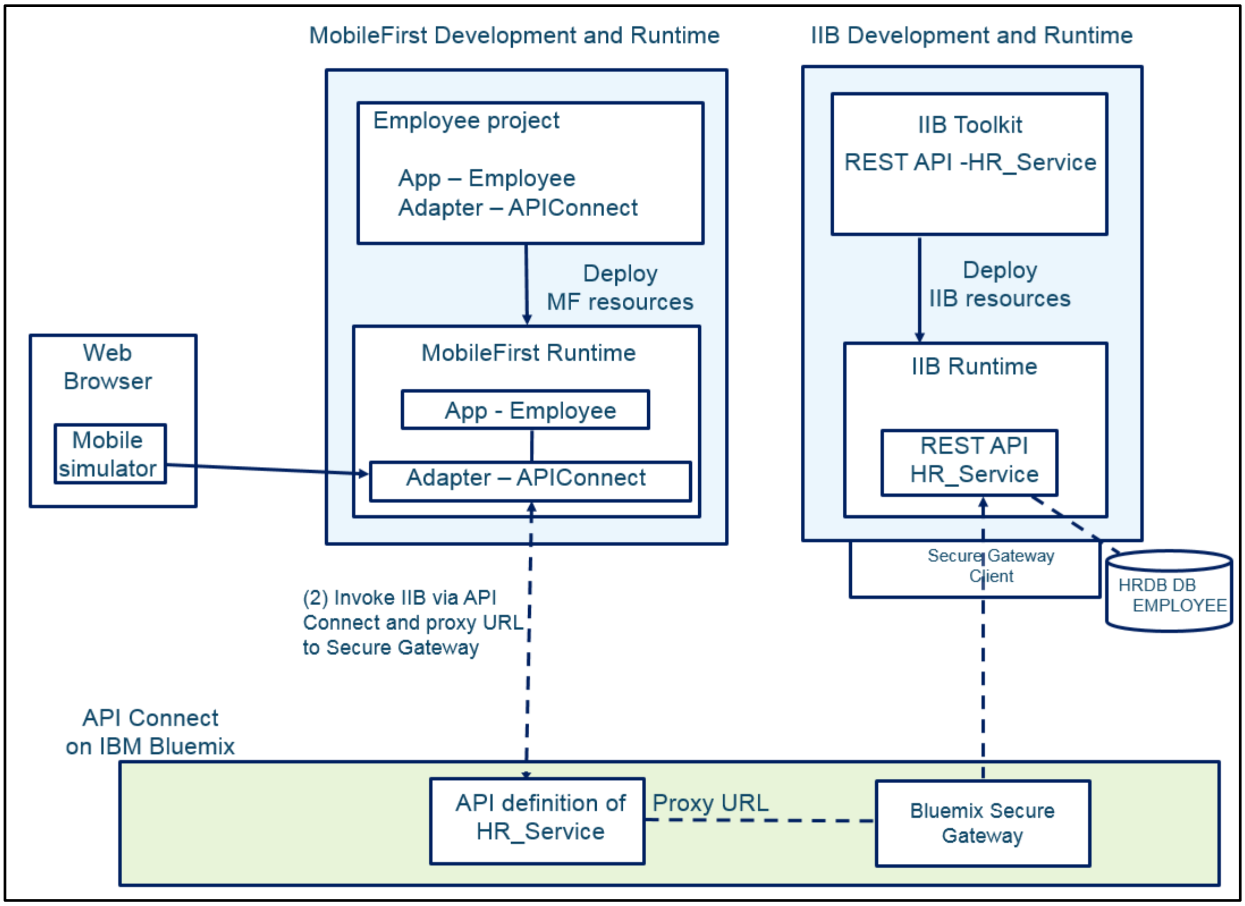
If successful, the returned data will be presented through a new browser window.



Invocation Result of procedure: 'getDetails' from the MobileFirst Server:

```
{
   "DBResp_department": {
      "RowsRetrieved": 1,
      "UserReturnCode": 0
   },
   "DBResp_employee": {
      "RowsRetrieved": 1,
      "UserReturnCode": 0
   },
   "Department": {
      "ADMRDEPT": "A00",
      "DEPTNAME": "SPIFFY COMPUTER SERVICE DIV.",
      "DEPTNO": "A00",
      "LOCATION": null,
      "MGRNO": "000010"
   },
   "Employee": {
      "BIRTHDATE": "1963-08-24T00:00:00+01:00",
      "BONUS": 1000.0,
      "COMM": 4220.0,
      "EDLEVEL": 18,
      "EMPNO": "000010",
      "FIRSTNME": "CHRISTINE",
      "HIREDATE": "1995-01-01T00:00:00Z",
```

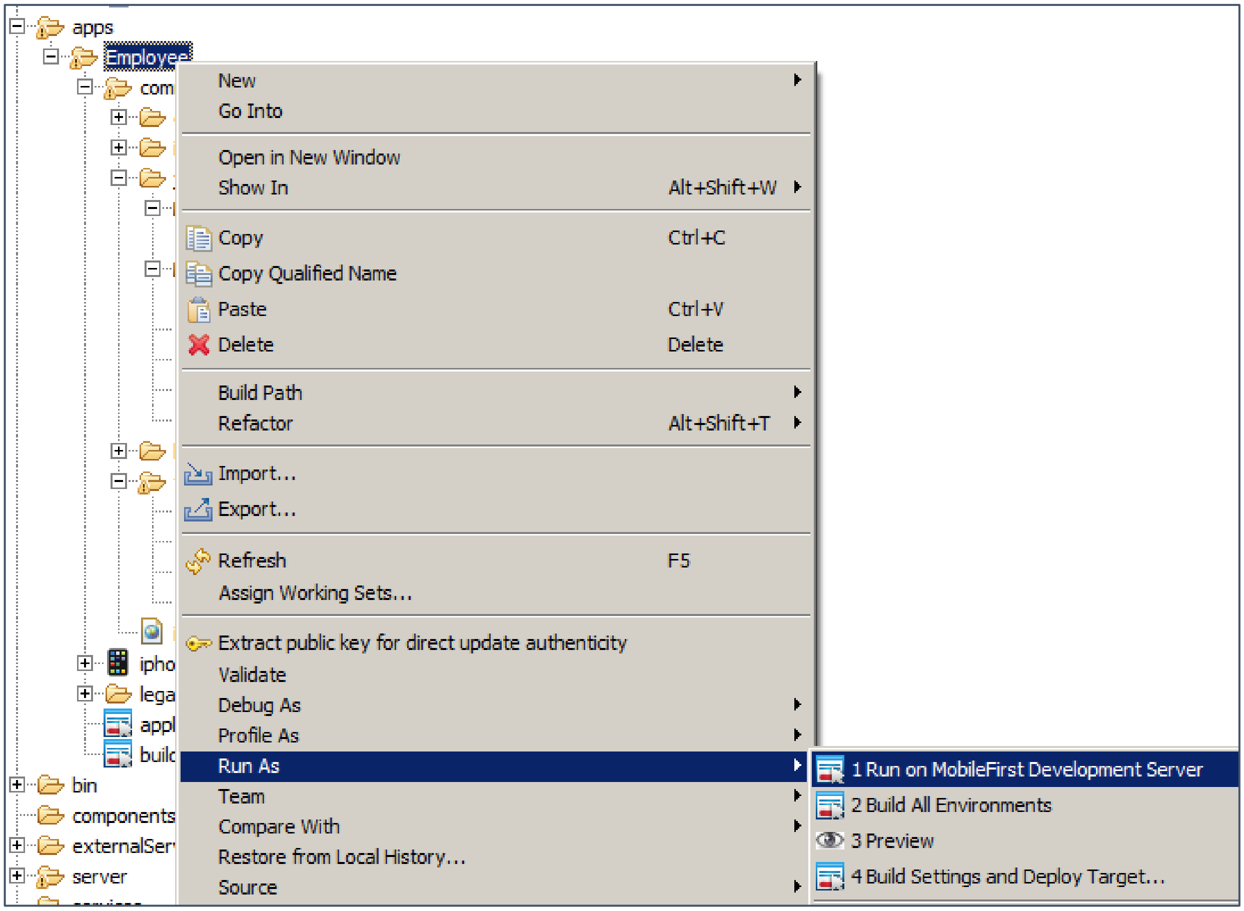
7. As a reminder, the MobileFirst adapter has been invoked directly by the MobileFirst Adapter simulator.



5.2.3 Deploy and retest the updated MobileFirst Employee App

1. Return to the MobileFirst Studio, and deploy the updated Employee app.

   Right-click on the **Employee** app, and select **Run As, Run on MobileFirst Development Server**.

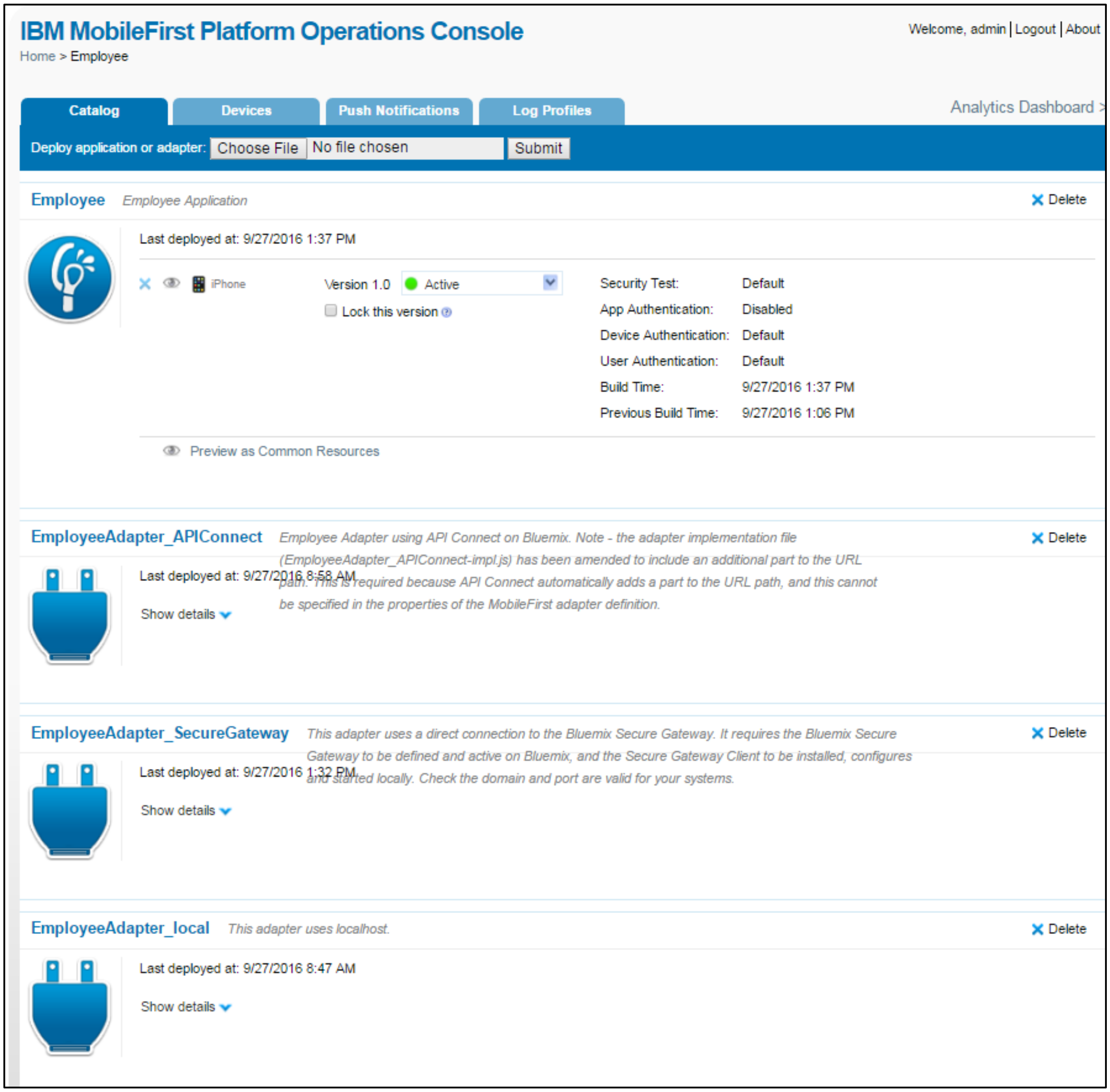


   When finished, in the Console you will see

```
Starting build process: application 'Employee', all environments
Application 'Employee' with all environments build finished.
Deploying application 'Employee' with all environments to MobileFirst Server...
Application 'Employee' deployed successfully with all environments
```

2. In the Chrome browser, in the MobileFirst Operations Console, press F5 to refresh the operations console.

   You will see the newly deployed API Connect adapter, and the Employee app should have a new deployment time.



Employee *Employee Application*

Last deployed at: 9/27/2016 1:37 PM

Version 1.0 ● Active

□ Lock this version

Security Test:	Default
App Authentication:	Disabled
Device Authentication:	Default
User Authentication:	Default
Build Time:	9/27/2016 1:37 PM
Previous Build Time:	9/27/2016 1:06 PM

EmployeeAdapter_APIConnect *Employee Adapter using API Connect on Bluemix. Note - the adapter implementation file (EmployeeAdapter_APIConnect-impl.js) has been amended to include an additional part to the URL path. This is required because API Connect automatically adds a part to the URL path, and this cannot be specified in the properties of the MobileFirst adapter definition.*

Last deployed at: 9/27/2016 8:58 AM

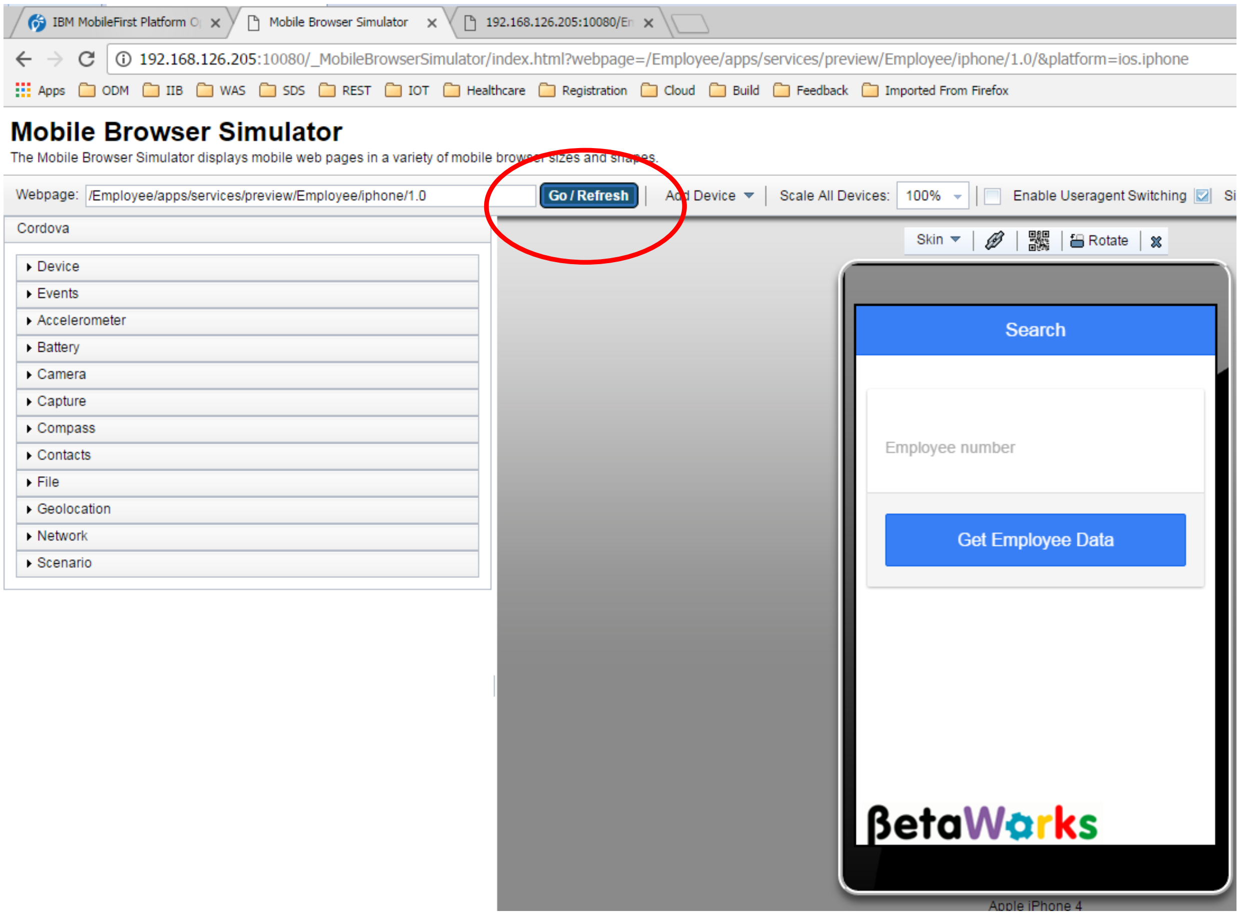
EmployeeAdapter_SecureGateway *This adapter uses a direct connection to the Bluemix Secure Gateway. It requires the Bluemix Secure Gateway to be defined and active on Bluemix, and the Secure Gateway Client to be installed, configures and started locally. Check the domain and port are valid for your systems.*

Last deployed at: 9/27/2016 1:32 PM

EmployeeAdapter_local *This adapter uses localhost.*

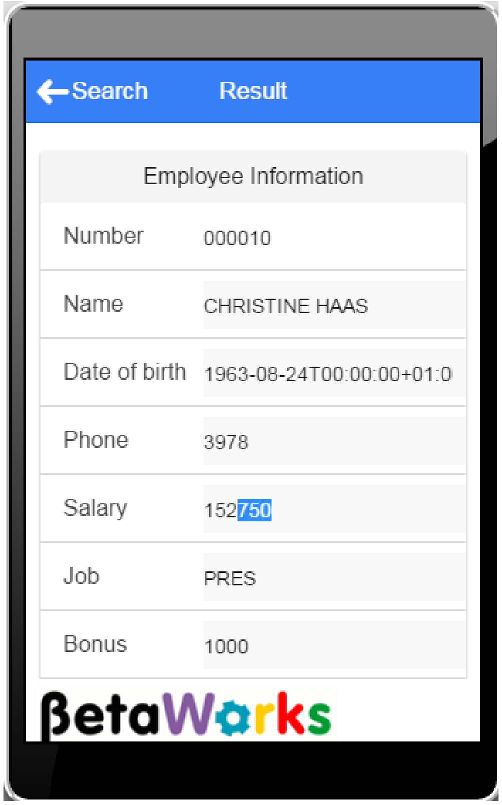
Last deployed at: 9/27/2016 8:47 AM

3. Switch to the iPhone simulator tab (probably still open from last time), and click the **"Go/Refresh"** button. This is necessary to ensure the simulator reloads correctly.



IBM MobileFirst Platform O
Mobile Browser Simulator
192.168.126.205:10080/E
192.168.126.205:10080/_MobileBrowserSimulator/index.html?webpage=/Employee/apps/services/preview/Employee/iphone/1.0/&platform=ios.iphone
Apps ODM IIB WAS SDS REST IOT Healthcare Registration Cloud Build Feedback Imported From Firefox
Mobile Browser Simulator
The Mobile Browser Simulator displays mobile web pages in a variety of mobile browser sizes and shapes.
Webpage: /Employee/apps/services/preview/Employee/iphone/1.0
Go / Refresh
Add Device
Scale All Devices: 100%
Enable Useragent Switching
Cordova
Device
Events
Accelerometer
Battery
Camera
Capture
Compass
Contacts
File
Geolocation
Network
Scenario
Skin
Rotate
Search
Employee number
Get Employee Data
BetaWorks
Apple iPhone 4

Provide the employee number 000010, and click Get Employee Data.



Employee Information

Number	000010
Name	CHRISTINE HAAS
Date of birth	1963-08-24T00:00:00+01:0
Phone	3978
Salary	152750
Job	PRES
Bonus	1000

In this test, the mobile phone app (simulated in the browser) has connected to the MobileFirst server, and invoked the MobileFirst APIConnect adapter. This adapter has connected to API Connect on Bluemix, which has forwarded the request to the IIB REST API, via the Secure Gateway Client.

END OF LAB GUIDE

.