

IBM Integration Bus

Developing a REST API using a Swagger JSON document (without reference to XML schemas)

Featuring:

The REST API editor for IIB The Mapping Node tools for REST API Testing with Swagger UI

September 2016 Hands-on lab built at product version 10.0.0.6

1. INTE	RODUCTION AND PREPARATION	3
1.1 1.2 1.3 1.4	INTRODUCTION OPEN THE WINDOWS LOG MONITOR FOR IIB CONFIGURE TESTNODE_IIBUSER FOR REST APIS CONFIGURE INTEGRATION BUS NODE TO WORK WITH DB2	3 3 5 6
Z. CRE		1
2.1 2.2 2.3 2.4 2.5 2.6	EXAMINE THE EMPLOYEESERVICE JSON DOCUMENT IMPORT THE HRDB SHARED LIBRARY CREATE THE NEW REST API	7 8 9 5 7 9
3. TES	T THE HR SERVICE REST API	0
3.1 3.2 3.3	DEPLOY THE SHARED LIBRARY	0 1 2
4. APP	'ENDIX	9
4.1	RECREATING THE HRDB DATABASE AND TABLES	9
END OF	LAB GUIDE	9

1. Introduction and Preparation

1.1 Introduction

In this lab you will create a new REST API, based on a provided Swagger document. The document is based on the EMPLOYEE scenario, and describes a number of REST operations. This guide will implement the Get Employee operation.

1.2 Open the Windows Log Monitor for IIB

A useful tool for IIB development on Windows is the IIB Log Viewer. This tool continuously monitors the Windows Event Log, and all messages from the log are displayed immediately.

From the Start menu, click IIB Event Log Monitor. The Monitor will open; it is useful to have this always open in the background.

💶 IIB Event Log Monitor
BIP3132I: (IB10NODE.server1) The HTTP Listener has started listening on port '
BIP2154I: (IB10NODE.server1) Execution group finished with Configuration messa
ge. [10/3/2014 3:17:24 PM] BLP2152I: (IB10NODE.server1) Configuration message received from broker. [10/3
/2014 5:07:36 PM]
BIP2153I: (IB10NODE.server1) About to ''Change'' an execution group. [10/3/201
45:07:36 PMJ BID21551: (IB10NODE convert) the device of the device of Employed vectors of Employed
oveeService JSONClient'' of type ''.APPZIP''. [10/3/2014 5:07:37 PM]
BIP2155I: (IB10NODE.server1) About to ''create '' the deployed resource ''gen.
getEmployee_EmployeeService_EmpServClient_JSON1'' of type ''.SUBFLOW''. [10/3/20
BIP21551: (IB10NODE server1) About to ''create '' the deployed resource ''EmpS
ervClient_JSON1'' of type ''.MSGFLOW''. [10/3/2014 5:07:37 PM]
BIP2154I: (IB10NODE.server1) Execution group finished with Configuration messa
ge. [10/3/2014 5:07:43 PM] BID21221: (ID10NDE UTTDIsteppen) The UTTD Listeppen has stanted listeping on m
ort ''7080'' for ''http'' connections. [10/3/2014 5:07:47 PM]
BIP2152I: < IB10NODE.server1 > Configuration message received from broker. [10/3
/2014 5:50:41 PM]
BIP21531: (IBIUNUUE.server1) Hbout to ''Change'' an execution group. [10/3/201 4 5-50-41 PM]
BIP2155I: (IB10NODE.server1) About to ''delete '' the deployed resource ''EmpS

This tool is not shipped as part of the IIB product; please contact us directly if you would like a copy.

Updates for IIB v10.0.0.3

This lab, version 10.0.0.3, has been updated significantly from earlier versions. The following changes have been made:

You should use the Windows user "iibuser". This user is a member of mqbrkrs and mqm, but is not a member of Administrators. The user "iibuser" can create new IIB nodes and do all required IIB development work. However, installation of the IIB product requires Administrator privileges (not required in this lab).

The database has been changed from the DB2 SAMPLE database to the DB2 HRDB database. HRDB contains two tables, EMPLOYEE and DEPARTMENT. These tables have been populated with data required for this lab. (The DDL for the HRDB is available in the student10 folder; we intend to provide corresponding DDL for Microsoft SQL/Server and Oracle over time).

The map node now retrieves multiple rows from the database, using an SQL "LIKE" function . Additionally, the map has been refactored to use a main map and a submap. Both the main map and submap are located in a shared library.

Input to the integration service and the REST service is now a simple schema containing just one element, the required employee number.

As a consequence, this version of the lab, and the associated solution, can only be used with the corresponding changes in other labs. Use version 10.0.0.3 of all labs in this series of lab guides.

Updates for IIB v10.0.0.5

IIB v10, fixpack 5 (10.0.0.5) has made significant changes in the Toolkit representation of REST APIs, the editor functions, and with the tools provided with the Mapping Node editor. This lab has been updated to reflect those changes.

Updates for IIB v10.0.0.6

This lab now uses JSON models throughout, and has removed the parts of the lab that were based on earlier XML schema components. The Mapping nodes use the Map Node wizard, which automatically creates input and output assemblies, based on the JSON model definitions.

Use of the pre-built submap (used in earlier versions) has been removed, and replaced with a single mapping node.

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.

- 1. Ensure that TESTNODE_iibuser is started.
- Enable Cross-Origin Resource Scripting for REST. This is required when testing with the SwaggerUI test tool. See <u>http://www.w3.org/TR/cors/?cm_mc_uid=09173639950214518562833&cm_mc_sid_502000</u> 00=1452177651 for further information.

(Helpful hint - the VM keyboard is set to UK English. If you cannot find the "\" with your keyboard, use "cd .." to move the a higher-level folder in a DOS window), or change the keyboard settings to reflect your locale.)

In an IIB Integration Console (shortcut on the Start menu), run the following command.

Note, the text should be typed on a single line - the parameters are shown on different lines here for readability; the same approach is taken throughout this and other lab documents.

mqsichangeproperties TESTNODE_iibuser

- -e default
- -o HTTPConnector
- -n corsEnabled -v true

1.4 Configure Integration Bus node to work with DB2

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

1. In 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.

2. Create the HR_Service REST API

In this section you will create a new REST API. This scenario will be based on the EmployeeService example that you may have used in other labs in this series.

2.1 Examine the EmployeeService JSON document

1. In Windows Explorer, locate the file

c:\student10\REST_API_HR_Service\resources\ HR Employee and Department Services.json

Open the file with the Notepad++ editor (right-click, select Edit with Notepad++).

We have installed a JSON document plugin into Notepad++, so this JSON document will be formatted for easy reading.

The JSON document has been constructed to define interfaces for the EMPLOYEE and DEPARTMENT tables. The document also has definitions for DBResp (for database response information, EmployeeResponse (includes DBResp and EMPLOYEE), and DepartmentResponse (includes DBResp and DEPARTMENT).

The main section of the document is a series of operations (GET, POST, PUT, etc), associated with different types of operation (getEmployee, listEmployees, etc).

Near the bottom of the document are two definitions for EMPLOYEE and DEPARTMENT, which defines the precise structure of these elements.

This document will be used as the basis of the REST API that you will build in IIB.

Look for the variable "EMPNO", which is referenced in several places in the document. This variable will be used by the IIB REST API.

The input parameter to the operation that you will implement is "employeeNumber"; this is defined as a path parameter for the required operation. Note the "required" parameter is set to "true".

Close the editor without making any changes to the document.

```
HR_Employee_and_Department_Services.json 🛛
      ł
        "swagger" : "2.0",
  2
  3
        "info" : {
  4
          "title" : "HR Employee and Department Services",
  5
          "description" : "This is the HR Swagger document for the Employee and Department Services 1
  6
          "version" : "3.0.0"
  7
        1.
  8
        "basePath" : "/HR Services/resources",
  9
        "tags" : [ {
 10
          "name" : "employees",
 11
          "description" : "Lists all of the employees at ACME"
 12
       }, {
          "name" : "departments",
 13
 14
          "description" : "Lists all of the departments at ACME"
 15
        } ],
 16
        "paths" : {
 17
          "/departments" : {
 18
            "get" : {
```

2.2 Import the HRDB Shared Library

The REST API that you will develop will use the EMPLOYEE table from the HRDB database. This requires the HRDB Database Definition project, which represents the tables schemas. This is used by the Mapping nodes that access the EMPLOYEE table. The project is contained in a Shared Library called HRDB; creating it as a shared library means that it can be shared with more than one project.

The HRDB shared library and associated project will not be developed in this lab - see Lab 01 from the 2015 series on lab guides (Creating an Integration Service) for details of how to do this. In this lab, you will import a pre-built version of this project.

1. Start the IIB Integration Toolkit, and create a new workspace. If you already have a workspace open, click File, Switch Workspace. Give the new workspace the name

```
c:\users\iibuser\IBM\IIB 10\workspace REST
```

2. In the new workspace, import the Project Interchange file:

```
C:\student10\REST_API_HR_Service\resources\HRDB.zip
```

Import both the **HRDB** shared library and **HRDB_project** from this PI file, and click Finish.

🌐 Import Project Interchange Contents	<u>_ ×</u>
Import Projects Import Projects from a zip file.	ļ.
From zip file: C:\student10\REST_API_HR_Service\resources\HRDB.zip	Browse
Project location root: C: \workspaces\FP6 ☑ ÈNRDB ☑ ÈHRDB_project	Browse
Select All Select Referenced	
Contract	Cancel

3. When imported, you will see the HRDB shared library as shown here. Note that HRDB_project is an Independent Resource (ie. it is a plain project, not a shared library).



2.3 Create the new REST API

1. In the workspace, create a new REST API..



2. Name the new service HR_Service.

Select "Import resources and operations defined in a Swagger document".

Click Next.

le Create a REST API	
Create a REST API	
A REST APT is an application that implements a REST ful interface.	
Name HR_Service	
$\ensuremath{\mathbb{C}}$ Create a REST API and define resources and operations yourself	
API base path //hr_service/v1	
Version 1.0.0	
 Import resources and operations defined in a Swagger document 	
O Kart > Finish	Cancel

Page 9 of 39

Version 10.0.0.6

3. Using the Browse button, import the JSON document

c:\student10	\REST	_API_	_HR_	Service	\resour	ces\
HR Employee	and	Depa	rtm	ent Serv	/ices.js	on

🕀 Create a REST API	<u>_ 🗆 🗙</u>
Create REST API from definition file Create a REST API from an existing Swagger 2.0 document.	
Import a Swagger 2.0 document from one of the following locations:	
Select from a file system	
Location: C:\student10\REST_API_HR_Service\resources\HR_Employee_and_Department_Services.json	Browse
C Select from your workspace	
Open Swagger Editor in the default web browser, outside of Integration Development Toolkit	>
	-
Contract Search Sear	Cancel

4. If you want to edit the HR_Employee_andDepartment_Services.json file in the Swagger Editor (editor.swagger.io), you can click on the "Open Swagger Editor ..." link on the Toolkit wizard. This will automatically start the Swagger Editor in a browser window (shown below). You can copy/paste the ".json" file into the editor in the left pane of the Swagger Editor.

You will see warnings about unused definitions. You can ignore these (they will be implemented later).

However, in this lab, you will use the Integration Toolkit editor, so switch back to the Toolkit and click Next.

Swagger Editor × +	
(edtor.swagger.io/#/	C 🔍 Search 🗘 🖨 🗢 😑
ODM IIB WAS SDS REST IOT Healthcare Registration Cloud Build Feedba	dk
🛞 File 🗸 Preferences 🖌 Generate Server 🖌 Generate Client 🖌 Help 🗸	✓ Processed with no error
1 · { 2 "swagger": "2.0", 3 - "Info": { 4 "tile: "#RDE Employee and Department Services",	Warnings Collapse
 Description': 'INIS is the HKUB skagger document FOr the Employee and Department Services used by the IIB BetaNorks REST labs", "version": "2.0.0" >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	▲ Swagger Warning Definition is not used: #/definitions/EmployeeResponse
<pre>8 "BasePath": "/TestNeAAppresources", 9 "test": {{ 10 "mame": "employees", 12 'rescription' "Lists all of the employees at ACME" 12 'rescription' "Lists all of the employees at ACME"</pre>	Details ▶ Object
13 - { 14 "name": "departments", 15 "description": "Lists all of the departments at ACME"	A Swagger Warning Definition is not used: #/definitions/DepartmentResponse
17 - 772 17 - 50ths; { 18 - 7/4partments; { 18 - 7/4partments; { 19 - 784; { 10 - 784; { 1	Details ▶ Object
 "summary": "Retrieve a list of the departments", "description": "Retrieve a list of the departments", "operationId": "getDepartments", "produces: ("sepDication/ison"), 	▲ Swagger Warning Definition is not used: #/definitions/DetailedResponse
25- "responses": { 26- "200": { 27 "description": "OK" 28 },	becans ▶ Object
29- "500": { 30 "description": "Something wrong in Server" 31) 32)	HRDB Employee and Department Services
33), 34 "post": { 35 "tags": ["departments"],	This is the HRDB Swagger document for the Employee and Department Services used by the IIB BetaWorks REST labs
30 summary creates a new department in the database. The db >department/unber//b that you specify must be unique. The department will not have a manager assigned.,	Version 2000 Filter operations by a tag:
38 "operation[d]: "createDepartment", 39 "consumes: ["application/son"], 40 "produces: ["application/son"], 41 - "parameters: [{	employees Lists all of the employees at ACME
42 "in': "body", 43 mame: "body", 44 mdescription", ", 46 mschenzistisse, 46 states (departments Lists all of the departments at ACME
47 "\$ref": "#/definitions/DEPARTMENT" 48 } 49 }],	Paths
30° responses (51- 200°: 52 "description": "successful operation", 53 - "schema": (/departments
54 "\$ref": "#/definitions/DEPARTMENT" 55 }	

5. The summary window will show you all of the REST operations that were defined in the JSON document. These operations were constructed to match the EMPLOYEE and DEPARTMENT tables in the HRDB database.

Note there is an operation named getEmployees (ie. retrieve a list of all employees), and an operation named getEmployee. This lab will implement the **getEmployee** operation.

Click Finish.

۲	Create a REST API						
Review the imported REST API definition.							
Re	Review the list of resources and operations that are specified in the REST API						
u							
1	Title: HR Employee and Department Services						
[Description: This is the HR S	wagger do labs. It cor	cument for the Employee and Department Services used				
F	Base path: /HR Services/re	sources					
	Version: 3.0.0						
	Operation	Method	Resource				
	getDepartments	GET	/departments				
	createDepartment	POST	/departments				
	getDepartment	GET	/departments/{departmentKey}				
	updateDepartment	PUT	/departments/{departmentKey}				
	deleteDepartment	DELETE	/departments/{departmentKey}				
	getDepartmentEmploy	GET	/departments/{departmentKey}/employees				
	getDepartmentManager	GET	/departments/{departmentKey}/manager				
	setDepartmentManager	PUT	/departments/{departmentKey}/manager				
	getEmployees	GET	/employees				
	createEmployee	POST	/employees				
	getEmployee	GET	/employees/{employeeNumber}				
	updateEmployee	PUT	/employees/{employeeNumber}				
	deleteEmployee	DELETE	/employees/{employeeNumber}				
	getEmployeeDepartment	GET	/employees/{employeeNumber}/department				
	setEmployeeDepartment	PUT	/employees/{employeeNumber}/department				
	predictSalary	GET	/employees/{EDLEVEL}/predictSalary				
ļ							
C	a						
(0		Back Next > Finish Cancel				

The swagger document has now been imported into the Integration Toolkit. The import
process has also created a base REST API and a message flow that implements the REST
API.

The imported and generated items are split into five main sections in the REST API editor:

- Header containing the base URL for the REST API, title and description
- Resources containing all the resources from the swagger document, and all of the operations that are contained within each resource
- Model Definitions schema definitions for the input and output JSON objects
- Error Handling options to add some elements of runtime security
- Security basic security parameters

III HR_Service ⊠
• Header
- Resources
/departments
/departments/{departmentKey}
/departments/{departmentKey}/employees
/departments/{departmentKey}/manager
/employees
/employees/{EDLEVEL}/predictSalary
/employees/{employeeNumber}
/employees/{employeeNumber}/department
- Model Definitions
Name
(+) <enter a="" create="" model="" name="" new="" to="" unique=""></enter>

7. As an example of Resources, expand **/employees/{employeeNumber}**. (You may wish to collapse the **/departments** resource, for readability).

You will see three operations, GET, PUT and DELETE.

For some of the operations (for example, the updateEmployee PUT operation in this resource), the Schema type has been set (in this case to EMPLOYEE).

For some other operations (for example the getEmployee GET operation), the input parameter is specified (employeeNumber). Because {employeeNumber} has been specified as part of the REST API URL path, the input parameter name (employeeNumber) is not editable.

The Schema Type of the successful (200) operation has been set to EmployeeResponse (originally specified in the swagger doc).

You can use the Schema Type dropdowns to change the required schema for the operation. The available values are derived from the Model Definitions section. If no schema type is specified, the REST operation can dynamically specify the format of the output message.

		- 15 - 1							
	GET	getEmployee					Retriev	e the details for an employe	e
/	Name	Parameter type	Data type	Format	Required	Description			
	employeeNur	mber path	string						
	Neponse	tatus Description					Array	Schema type	Allow null
	200	ОК						Employee ^D coponse	
	500	Something wron	g in Server						v
	404	The employee ca	annot be found						
	PUT	updateEmployee					Update	s an existing employee in the	e database.
	Name	Parameter type	Data type	Format	Required	Description			
	employeeNur	mber path	string	•	V	The employee	umber (emr	oloveeNumber) of the emplo	vee to be updated
						Time employed			,
	Boguest by	- du			(School	an huno			,
	Request bo	ody			Schen	na type	Allow	null	,
	Request bo	ody			Schen	na type	Allow	null	,
	Request bo	ody status Description			Schen	na type	Allow	ull Schema type	Allow null
	Request bo	status Description			Schen	na type	Allow I	schema type	Allow null
	Request bo	status Description	red updating the emp	loyee	Schen	na type	Allow I	Schema type	Allow null
	Request bo	status Description Updated A problem occur There was a pro	red updating the emp	xloyee t	Schen	na type	Allow I	Schema type	Allow null
	Request bo	status Description Updated A problem occur There was a pro The employee ca	red updating the emp blem with the reques annot be found	loyee t	Schen	na type	Array	schema type	Allow null
	Request bo	status Description Updated A problem occur There was a pro	red updating the emp blem with the reques annot be found	xloyee t	Schen	na type	Allow 1 Allow 1 Array	Schema type	Allow null
	Request bo	status Description Updated A problem occur There was a pro The employee ca deleteEmployee	red updating the emp blem with the reques	koyee t	Schen	na type	Array	schema type	Allow null
	Request bo	status Description Updated A problem occur There was a pro	red updating the emp	xloyee t	Schen	na type	Array	Schema type	Allow null

Version 10.0.0.6

8. Before proceeding with the implementation, the REST API project has to reference the HRDB shared library. This is because you will use a map to retrieve data from the database; the map references elements in the HRDB.dbm file.

In the navigator, right-click HR_Service and select "Manage Library references".

🔚 Application De	velopment 🙁 🐯 Patterns Explorer
Application Deve	elopment
□ ·· 🖅 HR_Service □ ·· 💯 REST / ⊡ ·· 🔁 Resour □ ·· 🐒 HRDB ⊡ ·· 😭 Other ⊡ ·· 😭 Independe	New Manage Library references Manage included projects Focus on REST API

Tick HRDB and click OK.

Manage Library References	×
Select the shared or static libraries to be referenced. Any other static libraries referenced from the selected static library will also be included.	
Shared libraries	1
Static libraries	7
The following static libraries will also be included because they are	
referenced by the selected static libraries.	
P	
OK Cancel	1

2.4 Implement the getEmployee operation

You will now implement the getEmployee operation that was defined by the EmployeeService json document.

1. In the Integration Toolkit, switch to the HR_Service REST API.

Expand the REST API Resources, and position the editor at the **/employees/{employeeNumber}** resource.

The getEmployee operation is the first operation in this resource.

employees/{employ	veeNumber}							
GET getE	mployee					Retrieve	the details for an employee	
Name	Parameter type	Data type	Format	Required	Description			
employeeNumber	path	string	•					
Response status	Description					Array	Schema type	
200	ОК						EmployeeResponse	•
500	Something wrong) in Server						•
404	The employee ca	nnot be found						•

2. In the top right part of the description, click on the icon to "Create a subflow for the operation".

Retrieve the details	for an employee		E	Ē
Required	Description		\widetilde{I}	\oplus

This will create a skeleton subflow where you will provide the logic to implement the operation:



3. Drop a Mapping node onto the flow editor as shown. As soon as you drop the node onto the flow editor, change the name of the node to getEmployee, and click Return (this should automatically name the map file, as shown in the node properties, below. The map name is formed by concatenating the REST operation name (getEmployee in this example) with the label of the mapping node (also getEmployee in this example).

🗾 HR_Service	🗉 *getEmpl	oyee.subflow 🕱	
👌 😲 Palette	F	ow Exerciser: 💷 🚰 🚧 🛛 🔍 🔍	
SCA SCA SCA SCA SCA SCA SCA SCA	Adapters	Input	Output
Construction	. ▼		
Properties S	3 🖳 Problems	🗄 Outline 🖉 Tasks 🔠 Deployment Log	
🕀 Mapping	Node Propertie	es - getEmployee	
Description			
Basic	Mapping routine*	{default}:getEmployee_getEmployee	
Validation	Transaction*	Automatic	
Monitoring			

2.5 Implement the map

1. Double-click the Mapping node that you have placed on the flow editor.

Note that the new Map wizard has provided two options:

- Message map for input/output REST operation getEmployee. This option was introduced in IIB v10.0.0.5, and is specifically designed for implementing REST APIs with the Mapping node. Ensure this option is selected.
- Simple message map called by message flow node. This option enables the more general Mapping Node options. It could be used for REST APIs, but the map designer would then be responsible for manually specifying the inputs and outputs correctly.

Click Finish (Next is not available as an option, since the map wizard automatically creates the inputs and outputs).

🌐 New Message Map		<u>_ 0 ×</u>
Specify a new message map file Select map type, container, name, and broker	schema for the new map.	
Type of map that you want to create: • Message map with the input and output for • Simple message map called by a message • Submap called by another map	or REST API operation getEmployee flow node	
Container: HR_Service		New
Map name: getEmployee_getEmployee		
Map organization		
Schema: (default broker schema)		-
?	<back next=""> Finish</back>	Cancel

2. The basic map is shown. Because the getEmployee operation is a GET method, the input is defined as a BLOB and the output message is a JSON message. The input message can be defined as a BLOB because for a GET operation, the input parameters are provided in the message header, and it is not necessary to parse the message body.



Version 10.0.0.6

3. Note that the input message assembly is already shown with the LocalEnvironment folder. This is because IIB places some parts of the input of a REST API into the LocalEnvironment, and the Mapping Node will almost always need access to this for developing application logic.

Expand the LocalEnvironment folder, and fully expand the REST subfolder.

Note that employeeNumber has already been added to the REST folder - this was derived from the input Schema Type on the swagger definition of the getEmployee operation.

Also note that IIB 10.0.0.6 has added the Response folder under the REST folder.

🖃 🚰 Message Assembly		BLOB
$\rightarrow 1$ <click filter="" to=""></click>		
🗄 📌 Properties	[01]	PropertiesType
🖃 📌 LocalEnvironment	[01]	_LocalEnvironmentType
	[01]	_LocalEnvironmentDestinationType
	[01]	_LocalEnvironmentWrittenDestinationType
e Aggregation	[01]	_LocalEnvironmentAggregationType
	•	
🖃 🖻 REST	[01]	_LocalEnvironmentRESTType
🖃 💽 Input	[01]	_RESTInputType
e Method	[01]	string
e Operation	[01]	string
e Path	[01]	string
e URI	[01]	string
Parameters	[01]	<anonymous></anonymous>
🖃 📇 choice of cast items	[0*]	
e any	[11]	
employeeNumber	[11]	string
E Response	[0_1]	RESTRESponseType
e CorrelationID	[01]	hexBinary
e StatusCode	[01]	integer
e ResponseHeadersSize	[01]	integer
e ResponseBodySize	[01]	integer
	[01]	<anonymous></anonymous>
∎	[01]	_LocalEnvironmentTimeoutRequestType
∎ e XSL	[01]	_LocalEnvironmentXSLType
ny 👔	[0*]	
E 🛃 BLOB	[11]	BLOB_Msg_type

4. Now focus on the output message assembly. The Map wizard has provided a JSON output assembly (because the getEmployee operation specified a schema type of EmployeeResponse for the response message).

Fully expand the output assembly. The JSON Data is of type EmployeeResponse. You will see the DBResp (JSON object) and Employee (JSON array) items underneath the Data element.

Note the Item element (of type EMPLOYEE) is a repeating element.

Ensure the Data (type EmployeeResponse) element is highlighted. This is important for the subsequent step which will automatically create a Select transform, and connect it to the highlighted element.

	🖃 🔩 Message Assembly		JSON
	≩ <click filter="" to=""></click>		
•	🗄 📌 Properties	[01]	PropertiesType
	🖃 📌 JSON	[11]	JSONMsgType
	e Padding	[01]	string
	🖃 🗓 Data	[11]	EmployeeResponse
	🖃 🖻 DBResp	[01]	DBRESP
	e UserReturnCode	[01]	<integer></integer>
	e RowsRetrieved	[01]	<integer></integer>
	e RowsAdded	[01]	<integer></integer>
	e RowsUpdated	[01]	<integer></integer>
	e RowsDeleted	[01]	<integer></integer>
	e SQLCODE_Errorcode	[01]	<integer></integer>
	e SQLSTATE_SQLState	[01]	<string></string>
	e SQL_Error_Message	[01]	<string></string>
	🖃 🖻 Employee	[01]	JSONArray_Employee
	🖃 🖻 Item	[0*]	EMPLOYEE
	e EMPNO	[11]	<string></string>
	e FIRSTNME	[11]	<string></string>
	e MIDINIT	[01]	<string></string>
	e LASTNAME	[01]	<string></string>
	e WORKDEPT	[01]	<string></string>
	PHONENO	[01]	<strina></strina>

5. Now add a transform to retrieve data from the HRDB database. Click the "Select rows from a database" icon at the top of the map editor.



6. In the New Database Select window, the HRDB database should already be available and selected (because you set the HRDB Library reference).

Select the EMPLOYEE table.

🌐 New Database Select			×
Hew Database Select Choose a database to select from Select a database available to the map, or obtain a different database HRDB Add database Choose the columns to include You must choose at least one column. Our must choose at least one column. Department Department Department Department Department	Define a where clause The where dause is used to extract only the key column in the database table. The value evaluate to a boolean. Table columns Table columns Tabl	Operators Operators AND OR NOT C C C C C C C C C C C C C C C C C C C	cified condition, which is often the value of a puts in the map. The expression must Available inputs for column values ■
► Classify SQL warning	Plac XPath expression		Edit Add Remove
?			OK Cancel

7. Remove the "1=1" statement from the "SQL where clause".



- 8. Create a new WHERE clause, as follows:
 - Double-click the table column EMPNO
 - Double-click the operator "LIKE".
 - Double-click the Available input
 LocalEnvironment/REST/Input/Parameters/choice..../employeeNumber

The SQL Where clause will be generated as follows: IIBADMIN.EMPLOYEE.EMPNO LIKE ?

A new XPath expression will have been created so that the SQL clause can reference the required input element from the LocalEnvironment (employeeNumber).



9. Refine the XPath expression.

The map will be built to retrieve all employee records that partially match the provided key. For example, if the provided key is "0020", we want the map to retrieve rows with a value of "000020", "00201", "00204", etc.

This is done by extending the SQL LIKE statement, in conjunction with the "%" character, appended both as a prefix and a suffix. Hence the SQL statement "SELECT xxxx LIKE %0020%" will achieve the result described above.

To do this, edit the XPath expression as follows:

```
fn:concat('%', $ComIbmMessage..../employeeNumber, '%')
```

The result should look like this:

SQL where	dause		
IIBADMIN	.EMPLOYEE.EMPNO LIKE ?		
·			
Plac	XPath expression	Edit.	
?	fn:concat('%', \$ComIbmMessageAssembly_BLOB/LocalEnvironment/REST/Input/Parameters//employeeNumber, '%')		
			Add
			Remove
•		Þ	

Now double-check that you made the XPath changes exactly as shown above !!

Click OK to close the Database Select window.

10. A Select transform will have been placed into the map, connecting to the output "Data" element (of type EmployeeResponse). If you didn't previously highlight the Data element, the connection will be in the wrong place; you can adjust the connector by highlighting it, and dragging to the desired target.

Click the Select transform (click the Select text, not the surrounding transform icon). This is where you will define the precise element mappings that you need. Clicking Select will take the editor to the next lower logical level of the map (synonymous with a "Do ... End" programming construct).

▼getEmployee_getEmploy	yee_REST 🛛 🔑 🦆 🕌) 🚯 😫 🕾 🔍 🔍	0	🎟 🏭 🛅 🏭 🗯	0-	
🖃 🔠 Message Assembly	BLOB		[🖃 🖧 Message Assembly		JSON
$\Rightarrow +$ <click filter="" to=""></click>				Click to filter>		
🗈 📌 Properties	[01] PropertiesType	Move 🔻	•	🗉 📌 Properties	[01]	PropertiesType
	[01] _LocalEnvironmentType		[🖃 📌 JSON	[11]	ISONMsgType
E BLOB	[11] BLOB_Msg_type			e Padding	[01]	le_g
				🖃 🗓 Data	[1.,1]	EmployeeResponse
Select from HRDB				🕀 🖻 DBResp	[01]	DBRESP
→ <click filter="" to=""></click>				😑 🖻 Employee	[01]	JSONArray_Employee
표 🐻 ResultSet	[0*] Result Set Row			🗉 🖻 Item	[0*]	EMPLOYEE
		1				

11. Expand the output Employee element to expose the **Item** element.

Connect **ResultSet** Row to the **Item** array element in the output Data. This will generate a "For each" transform; this means that the transform will be performed for each row returned from the database. However, you still haven't defined the individual element mappings – see the next step.

🖃 📸 ResultSet	[0*] Result Set Row	tia For each → *	🖃 🔓 Data	EmployeeResponse
⇒ <click filter="" to=""></click>			Sector Secto	
E EMPNO	[1.,1] CHAR		🖃 🖻 DBResp	[01] DBRESP
E EIRSTNME	[1., 1] VARCHAR		C UserReturnCode	[01] <integer></integer>
			e RowsRetrieved	[01] <integer></integer>
			e RowsAdded	[01] <integer></integer>
			e RowsUpdated	[01] <integer></integer>
			e RowsDeleted	[01] <integer></integer>
	[1] CHAR		e SQLCODE Errorcode	[01] <integer></integer>
HIREDATE	[11] DATE		SOLSTATE SOLState	[0., 1] <string></string>
JOB	[11] CHAR		SQL Error Message	[0 1] <string></string>
EDLEVEL	[11] SMALLINT			
E SEX	[11] CHAR			[0BONArray_Employee
BIRTHDATE	[11] DATE		🕒 🖻 Item	[0*] EMPLOYEE
SALARY	[11] DECIMAL		e EMPNO	[11] <string></string>
BONUS	[1., 1] DECIMAL		e FIRSTNME	[11] <string></string>
E COMM	[1., 1] DECIMAL		e MIDINIT	[01] <string></string>
	[1	e LASTNAME	[01] <string></string>
			e WORKDEPT	[01] <string></string>

12. Click the text "For each" to specify the individual element mappings. This will take you to the next lower logical level of the map.

Click the Automap icon to invoke the wizard to create the mappings automatically.

getEmployee_getEm	ployee_REST	🖾 🗮 🎲 🖼 🛤 🖬 🗗 🗳 🕼 유 🧠 📜
		Ê
🖃 📸 ResultSet	Result Set Row	EMPLOYEE
*i ∗i <click filter="" to=""></click>		⇒ <click filter="" to=""></click>
EMPNO	[11] CHAR	E EMPNO [11] <string></string>
FIRSTNME	[11] VARCHAR	e FIRSTNME [11] <string></string>
MIDINIT	[11] CHAR	e MIDINIT [01] <string></string>
LASTNAME	[11] VARCHAR	LASTNAME [01] <string></string>
WORKDEPT	[11] CHAR	WORKDEPT [01] <string></string>
PHONENO	[11] CHAR	PHONENO [01] <string></string>
HIREDATE	[11] DATE	e HIREDATE [11] <string></string>
JOB	[11] CHAR	e JOB [01] <string></string>
EDLEVEL	[11] SMALLINT	e EDLEVEL [01] <integer></integer>
SEX	[11] CHAR	E SEX [01] <string></string>
BIRTHDATE	[11] DATE	BIRTHDATE [11] <string></string>
SALARY	[11] DECIMAL	E SALARY [01] <double></double>
BONUS	[11] DECIMAL	BONUS [01] <double></double>
COMM	[11] DECIMAL	COMM [01] <double></double>

13. Accept the generated mappings, and click Finish.

🌐 Auto Map		
Automatically map inputs to outputs		===
Choose the options to automatically map the selected input and output elements. Click Next to select the transforms to create, or dick Finish to create the transforms for all the match	ing elements.	
Mapping Scope		_
Map all simple descendants of the selected elements		
Group transforms into nested maps		
C Map the immediate children of the selected elements		
Name Matching Options		
Case sensitive		
Alphanumeric characters (Letters and digits only)		
Mapping Criteria		
Press F1 for more information when the names of inputs and outputs satisfy more than one crite	rion.	
Create transforms when the names of inputs and outputs are the same		
Create transforms when the names of inputs and outputs are more similar than		
	▶ 60 % match	Default
Create d'ansionns when the input and output names are matched to synonyms denned in a n	iic	
I		Browse,
(?)	ik Next >	Finish Cancel

14. Return to the higher level of the map by clicking the yellow up arrow.

🖃 📸 ResultSet	Result Set Row	□ 🖳 Item	EMPLOYEE
\Rightarrow <click filter="" to=""></click>		⇒tion <click filter="" to=""></click>	
EMPNO	[11] CHAR	Move - (11)	<string></string>
EIRSTNME	[1.,1] VARCHAR	Bove Image: A second	<string></string>
	[]	@ MIDINIT [01]	<string></string>
MIDINIT	[11] CHAR	Move ELASTNAME [01]	<string></string>
LASTNAME	[11] VARCHAR	● Move ▼ (01)	<string></string>
WORKDEPT	[1., 1] CHAR	Bove PHONENO [01]	<string></string>
		e HIREDATE [11]	<string></string>
PHONENO	[11] CHAR	• • • • • • • • • • • • • • • • • • •	<string></string>
HIREDATE	[11] DATE	Move - EDLEVEL [01]	<integer></integer>
5 JOB	[1., 1] CHAR	● SEX [01]	<string></string>
		BIRTHDATE [11]	<string></string>
EDLEVEL	[11] SMALLINT	• • • • • • • • • • • • • • • • • • •	<double></double>
SEX	[11] CHAR		<double></double>
BIRTHDATE	[11] DATE	• • • • • • • • • • • • • • • • • • •	<double></double>

15. We want to set the value for RowsRetrieved.

Expand DBResp, and connect the input ResultSet to the output DBResp/RowsRetrieved. Again, this will initially create a "For each" transform; the transform will show a warning.

getEmployee_getEmp	loyee_REST	/ P1 .	P 🎝 🕯) JD1	ţı⊳	X -	P 🛱	🏠	⁰₿	1				8 i		0-
														Ŷ		
🖃 😼 ResultSet	[0*] Result Se	et Row			For each				िंह्र D > <cli< td=""><td>ata ck to fi</td><td>ter:</td><td>></td><td></td><td></td><td></td><td>EmployeeResponse</td></cli<>	ata ck to fi	ter:	>				EmployeeResponse
⇒ <click filter="" to=""></click>									- e	DBRes	^p			[0	01]	DBRESP
EMPNO	[11] CHAR								[e Use	rRetur	nCod	e	[0	01]	<integer></integer>
FIRSTNME	[11] VARCHA	R							[e Row	sRetri	eved		[0	01]	<integer></integer>
	[11] CHAR								[e Row	sAdde	d		[0	01]	<integer></integer>
LASTNAME	[1.,1] VARCHA	R							[e Row	sUpda	ted		[0	01]	<integer></integer>
	[ana] mittain								ſ	e Dow	eNalat	had		Fr	n 11	ZintenerN

16. Click the drop-down arrow on the transform, and change the transform to "Custom XPath".

Eor each -	EmployeeRespons
6	If Co Select ve Custom Transforms Custom ESQL Custom Java Custom Java Custom XPath Custom String Functions Boolean Functions
All ex to	ows you to enter your own XPath pressions in the property pages be used in the transform.

17. In the map editor, make sure the Custom XPath transform is selected.

On the Properties tab for this transform, select "General" and type the following into the XPath editor:

fn:count(

Then, invoke the content assist function (Ctrl-space) which will show you the possible values for completion. Select the value that shows \$ResultSet, or similar. Depending on whether you have made other editing changes, the value shown may be \$ResultSet or have a suffixed number. The example shown below is \$ResultSet.

Complete the XPath expression with a ")". This expression will calculate the number of rows retrieved from the database.

🔲 Properties 🔀 🚦	Problems	🖶 Outline 🛛 🕢 Tasks 🏢 Deployment Log
Transform - Cus	tom XPath	
General	fn:count(Insert Simple XPath
Namespaces		m: xy fn:abs (numeric?) : numeric?
Cardinality		<pre>xy fn:adjust-dateTime-to-timezone (xs:dateTime?) : xs:dateTime?</pre>
Variables		xy fn:adjust-dateTime-to-timezone (xs:dateTime? ,xs:dayTimeDuration?) : xs
Condition		xy fn:adjust-date-to-timezone (xs:date?) : xs:date?
Sort		<pre>wide in the second second</pre>
~		xy fn:adjust-time-to-timezone (xs:time?,xs:dayTimeDuration?): xs:time?

18. Set the value of UserReturnCode to 0. Do this by using an Assign transform.

Expand the output assembly DBResp, and right-click UserReturnCode. Select Add Assign.

🖃 🗓 Data	EmployeeResponse	
⇒ <click filter="" to=""></click>		
🖃 🖻 DBResp	[01] DBRESP	
e UserReturnCode	In 11 Zintegers	
e RowsRetrieved	C Undo Change code	
e RowsAdded	Revert	
e RowsUpdated		
e RowsDeleted	Cut	Ctrl+X
SQLCODE_Errorcod	Copy	Ctri+C
SQLSTATE_SQLStat		
SQL_Error_Message	Delete	
🗉 🖻 Employee	i Open Information Popup	Ctrl+Shift+I
	Open Declaration	F3
	🗈 Cast	
	- Add Connection	
	🛱 Accept All Auto-mapped Transforms	
	🖀 Reject All Auto-mapped Transforms	
	📮 Auto Map	
	Quick Link from Input	Ctrl+L
	Database	•
	Cache	•
	순 Up a Level	
	Expand All	
	Expand All Sort Transforms	•
	Expand All Sort Transforms View) }

19. The default value for an Assign is "0", so leave this unchanged.

Properties 🔀	R Problems	E Outline	Tasks	🖽 Deployment Log	Progress
Transform - As	sign				
General	Value: 0				
Cardinality					
Variables					
Condition					
Sort					
Order					
Documentation					
	1				

20. Finally, go up one level in the map (yellow up arrow), and delete the Task transform that was added by the Map Wizard.

💼 Task 🔻	Undo Change submap	
	Revert	
	of Cut	Ctrl+X
	📄 Сору	Ctrl+C
	Paste	⊂trl+∀
	💢 Delete	

The main map is now complete, so save (Ctrl-S) and close the map.

2.6 Complete the subflow

1. Connect the nodes as shown. Save and close the subflow.



3. Test the HR_Service REST API

This chapter will show you how to use the SwaggerUI tool to send a REST request into the REST API that you have just created.

3.1 Deploy the Shared Library

1. In the IIB Toolkit navigator, deploy the HRDB shared library to the default server.



3.2 Deploy the service

2. In the IIB Toolkit navigator, deploy HR_Service to the default server.



3.3 Test the service

1. Open the IIB web UI by right-clicking TESTNODE_iibuser and selecting Start Web User Interface.



2. You will be switched to the default browser. Fully expand TESTNODE_iibuser, down to the HR_Service REST API, as shown below.

Under HR_Service, click "API", which will show you the available operations in the REST API, and whether they have been implemented. Check that you have implemented the correct operation – it should be getEmployee.

It will also show you the URLs for the REST API and the definitions (the .json file).

IBM Integration					
Filter Options	HR_Service - REST API				
▼	API				
🗸 🖻 Servers 👻					
🔻 🛃 default 📼					
Services	REST API Base URL http://betaworks-esb10:7800/HR_Services/resources				
▼ (III REST APIS	REST API Definitions URL http://betaworks-esb10:7800/HR_Services/resources/HR_Employee_and_Department_Services.json				
▼ Æ HR_Service →					
	✓ /departments				
🕭 Libraries	POST createDepartment Creates a new department in the database. The DEPTNO that you specify must be unique. The d				
 Message Flows Subflows 	GET getDepartments Retrieve a list of the departments				
 Resources 					
▶ B References	 /departments/{DEPTNO} 				
Applications	DELETE deleteDepartment Deletes an existing department in the database.				
) Libraries	CET actionartment. Batrieve the datails for a department				
 Mared Libraries 					
Message Flows	PUT updateDepartment Updates an existing department in the database.				

3. On the "REST API Definitions URL", right-click and select "Copy Link Location".

HR_Service - REST API	СОРҮ
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_Der	Open Link in New <u>T</u> ab
✓ /departments	Open Link in New Private Window
POST createDepartment Creates a new department in the database. The DEPTNO that y	Bookmark This <u>L</u> ink Save Lin <u>k</u> As
GET getDepartments Retrieve a list of the departments	Save Link to Pocket
✓ /departments/{DEPTNO}	Search Google for "http://betawor Inspect Element (Q)

4. In Firefox, open a new tab, and open the SwaggerUI tool (using the bookmark in the REST folder).

By default, this will open the Petstore Swagger document.



5. In the entry field (not the browser address field), paste the contents of the clipboard and click Explore.

The two high-level functions, departments and employees, will be shown.



HR Employee and Department Services

This is the HR Swagger document for the Employee and Department Services used by the IIB BetaWorks REST labs. It contain definitions and JSON model definitions.

departments : Lists all of the departments at ACME	Show/Hide List Operations E
employees : Lists all of the employees at ACME	Show/Hide List Operations E
[BASE URL: /HR Services/resources , API VERSION: 3.0.0]	

6. We are concerned with the getEmployee operation so click "List Operations" to show the operations related to employees.

Note that SwaggerUI does not have any knowledge at this point of whether the operation has been implemented.

HR Employee and Department Services

This is the HR Swagger document for the Employee and Department Services used by the IIB BetaWorks REST labs. It contains resource definitions and JSON model definitions.

depai	rtments : Lists all of the departments at ACME	Show/Hide List Operations Expand Operations
emple	oyees : Lists all of the employees at ACME	Show/Hide List Operations Epand Operations
GET	/employees	Receive a list of the employees
POST	/employees	Creates a new employee
GET	/employees/{employeeNumber}	Retrieve the details for an employee
PUT	/employees/{employeeNumber}	Updates an existing employee
DELETE	/employees/{employeeNumber}	Deletes an existing employee
GET	/employees/{employeeNumber}/department	Retrieve the department for an employee
PUT	/employees/{employeeNumber}/department	Assign the department for the employee
GET	/employees/{EDLEVEL}/predictSalary	Retrieve the predicted salary for an employee

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

The input parameter is employeeNumber. Provide a suitable value, say 000010.

GET /employees	;/{employeeNumber}			Retrieve the details for an employee
Implementation No Retrieve the details fo	otes or an employee			
Response Class (Sta Model Model Schem	atus 200) a			
<pre>{ "DBResp": { "UserReturnCod "RowsAdded": @ "RowsUpdated": "RowsDeleted": "SQLCODE_Error "SQLSTATE_SQLS "SQL_Error_Mess "SQL_STATE_SQLS "SQL_Error_Mess "SQL_Error_Mess "SQL_STATE_SQLS "SQL_Error_Mess "SQL_STATE_SQLS "SQL_Error_Mess "SQL_STATE_SQLS "SQL_Error_Mess "SQL_STATE_SQLS "SQL_Error_Mess "SQL_STATE_SQLS "SQL_STATE_SQL "SQL_STATE_SQLS "SQLST</pre>	le": 0, ": 0, 0, 0, code": 0, itate": "string", sage": "string"			- - -
Response Content Typ	De application/json 💌			
Parameters	\frown			
Parameter Valu	Je	Description	Parameter Type	Data Type
employeeNumber			path	string
Response Message	S			
HTTP Status Code	Reason	Response Model		
404	The employee cannot be found			
500	Something wrong in Server			
Try it out!				

8. When you have provided an employeeNumber, click Try it out!

If successful, the returned data will look something like this. Note the database response information (user return code, number of rows returned), as well as the user data.

Note that you can copy the Request URL below, and paste directly into a browser.

```
Hide Response
 Try it out!
Request URL
 http://betaworks-esb10:7800/HR_Services/resources/employees/000010
Response Body
  {
    "DBResp": {
      "UserReturnCode": 0,
      "RowsRetrieved": 1
    },
    "Employee": [
      {
        "EMPNO": "000010",
        "FIRSTNME": "CHRISTINE",
        "MIDINIT": "I",
        "LASTNAME": "HAAS",
        "WORKDEPT": "A00",
        "PHONENO": "3978",
        "HIREDATE": "1995-01-01T00:00:00Z",
        "JOB": "PRES ",
        "EDLEVEL": 18,
        "SEX": "F",
        "BIRTHDATE": "1963-08-24T00:00:00+01:00",
        "SALARY": 152750,
        "DONUC". 1000
Response Code
 200
Response Headers
   "content-type": "application/json; charset=utf-8"
 }
```

9. Provide an employeeNumber that does not exist, for example 000012 (but make sure you use an employeeNumber that has 6 characters).

You will see the service has worked (UserReturnCode = 0), but no data has been found (RowsRetrieved = 0).

Parameters				
Parameter	Value	Description	Paramete	
EMPNO	000012		path	
Decrease Mag	20.000			
UTTR Status Code	Beacon	Baspapsa Madal		
	The endlower end the few	Response model		
404	The employee cannot be fou	nd		
500	Something wrong in Server			
Try it out!	de Response			
Request URL				
http://betaworks-esb10:7800/HRDB_RESTServices/resources/employees/000012				
Deenenee Deeh				
Response Body				
ł.				
"DBResp": {				
"UserRetu	rnCode": 0,			
"RowsRetr	ieved": 0			
},				
"Employee":				
J				
Response Code	e de la companya de l			
200				

10. Provide an employeeNumber that will generate several matches, for example "0020". This will use the "LIKE" operator in the SQL where clause.

Two records will be retrieved from the database. You will see the element UserReturnCode = 0, and the element RowsRetrieved = 2.

Use the slide bar to see the data of the two returned rows.

Parameters			
Parameter	Value	Description	Parameter
EMPNO	0020		path
Response Mes	sages		
HTTP Status Code	e Reason	Response Model	
404	The employee cannot be found		
500	Something wrong in Server		
H	de Response		
Request URL			
http://betawo	rks-esb10:7800/HR_Services/resource	es/employees/0020	
	_		
Response Body	/		
{	r		
UBResp : -	l unaCada": A		
"Bows Botu	aincode : 0,		
L	ieveu . 2		
J. "Employee"	. г		
{			
"EMPNO	": "000020".		
"FIRST	WME": "MICHAEL",		
"MIDIN	IT": "L",		
"LASTN	AME": "THOMPSON",		
"WORKDI	EPT": "B01",		
"PHONEI	NO": "3476",		
"HIRED/	ATE": "2003-10-10T00:00:00+01:00",		
"JOB":	"MANAGER ",		
"EDLEV	EL": 18,		
"SEX":	"M",		
"BIRTH	DATE": "1978-02-02T00:00:00Z",		
"SALAR	Y": 94250,		
Response Code	2		
200			

4. Appendix

4.1 Recreating the HRDB database and tables

The HRDB database, and the EMPLOYEE and DEPARTMENT tables have already been created on the supplied VMWare image. If you wish to recreate your own instance of this database, the command 1_Create_HRDB_database.cmd and 2_Create_HRDB_Tables.cmd are provided for this. These files are contained in the c:\student10\Create_HR_Database folder.

If used in conjunction with the workshop VMWare image, these commands must be run under the user "iibadmin". Appropriate database permissions are included in the scripts to GRANT access to the user iibuser.

When an IIB node is created, or recreated, the appropriate JDBC and security definitions must be created. Run the command files 3_Create_JDBC_for_HRDB and 4_Create_HRDB_SecurityId.cmd.

END OF LAB GUIDE