

# **IBM Integration Bus**

Using Callable Flows

Featuring:

CallableInput and Callable Reply nodes CallableFlowInvoke node

April 2016 Hands-on lab built at product Version 10.0.0.4

1. INTRODUCTION	3
<ol> <li>Scenario Overview</li> <li>Outline of tasks</li> <li>Configure Integration Bus node to work with the second se</li></ol>	
2. DEVELOP THE APPLICATIONS	5
<ul> <li>2.1 IMPORT THE PROVIDED RESOURCES</li> <li>2.2 CREATE A NEW MESSAGE MAP</li> <li>2.3 CREATE THE CALLABLE MESSAGE FLOW</li> <li>2.4 CREATE THE CALLING MESSAGE FLOW</li> <li>2.5 DEPLOY THE SHARED LIBRARY AND CALLABLE APP</li> </ul>	
3. TEST THE CALLING APPLICATION	23
<ul><li>3.1 TEST WITH FLOW EXERCISER</li><li>3.2 TEST WITH SOAPUI</li></ul>	23 
END OF LAB GUIDE	

## 1. Introduction

IIB V10.0.0.4 introduces Callable Flows. This enables a message flow (or integration service, or REST API) to invoke a separate message flow in a call/return (blocked wait) programming model.

The calling and called message flows operate on the same message tree, so there is no requirement for any protocol-related components. For example, the message flows simply access message tree elements, and there is not requirement for any knowledge of network location of invoked components.

### 1.1 Scenario Overview

In this lab, you will make the necessary configurations in SQL Server and IIB to enable an IIB application (integration service) to connect to SQL Server, and retrieve some rows from a specific SQL Server table.

The integration service is provided for you, and is the solution of the integration service that was developed in Lab 1 (Create Integration Service) in this series of labs.

## 1.2 Outline of tasks

The tasks to complete in this lab are the following:

- 1. Import a partially-built shared library and applications
- 2. Complete the development of a new map and two message flows
- 3. Deploy and test.

### 1.3 Configure Integration Bus node to work with DB2

If you have already done Lab 1 in this series (create an Integration Service), you can skip straight to Develop the Applications on the next page.

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

1. Open an IIB Command 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.

### 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. If used in conjunction with the VM 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.

## 2. Develop the Applications

## 2.1 Import the provided resources

- 1. Login to Windows with the user **iibadmin**, password = **passw0rd**.
- 2. To ensure development artefacts do not overlap, create a new workspace with the name \workspace\_callable.

Workspace Launcher		l	×
Select a workspace			
IBM Integration Toolkit stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.			
Workspace: C:\Users\iibuser\IBM\IIBT10\workspace_callable	•	Browse	
Copy Settings			
?	ОК	Cancel	

3. Import the PI file

## c:\student10\CallableFlow\resources\ EmployeeService\_callable\_startingPoint.10.0.0.4.zip

Ensure all projects are selected, and click Finish.

🌐 Import Project Ir	terchange Contents		_ 🗆 ×
Import Projects Import Projects from a	zip file.		<u> </u>
From zip file: Project location root:	C:\student10\CallableFlow\resources\EmployeeService_ C:\Users\iibuser\IBM\IIBT10\workspace_callable	_callable_startingPoint.10.0.0.4.zip	Browse
German EmployeeSerr     German EmployeeSerr     German EmployeeSerr     German EmployeeSerr     German EmployeeSerr	ice_Callable ice_callingApp ice_interface_and_maps		
Select All Deselect	t All Select Referenced		
?	< Bac	k Next > Finish	Cancel

4. The project EmployeeService\_interface\_and\_maps is the project that you have used in earlier labs. It contains the schemas that are required for the employeeNumber and EmployeeResponse elements, and the submap that will be used to retrieve data from the HRDB database.

The applications **EmployeeService\_callable** and **EmployeeService\_callingApp** are predefined applications to save time during the course of this lab. You will extend the provided message flows in these two applications.

🔚 Application Development 🙁 💐 Patterns Explorer	Ë	E	€Ę}	~ •
Application Development				
EmployeeService_Callable     Flows     G Referenced Libraries     EmployeeService_callingApp     BempServ_JSON_Calling_getEmployee.msgflow     Referenced Libraries     Other Resources     Schema Definitions     Schema Definitions     G getEmployee_WS.map     getEmployees_submap.map     Other Resources				

## 2.2 Create a new Message Map

1. In the EmployeeService\_interface\_and\_maps library, right-click the Maps folder, and select New, Message Map.

Employ	eeService_interface_and_map: nema Definitions GDL Definitions	S
	New	▶ 🗏 Babflow
	Сору	🖹 Message Model
	Paste	🚖 Message Map
	Delete	ESQL File
	Move	🔛 Broker Schema
	Rename	
	🚵 Import	J Example
	Z Export	😭 Other Ctrl+N
	Refresh	
	Run in New Workbench	

2. Name the new map getEmployee\_callable, and click Next.

🌐 New Message Map		
Specify a new message map file Select map type, container, name, and broker schema for the new	v map.	
Type of map that you want to create: Simple message map called by a message flow node Submap called by another map		
Container: EmployeeService_interface_and_maps		New
Map name: getEmployee_callable		
Map organization Use default broker schema Schema; (default broker schema)		<b>_</b>
Sack	Next > Finish	Cancel

- 3. The input to this map will be employeeNumber, which will be a single element in the IIB message tree. Similarly, the output from the map will be the complex element EmployeeResponse. No protocol-related elements will be required to be processed by this map.
  - Set the map input to DFDL and XML schemas, employeeNumber.
  - Set the map output to DFDL and XML schemas, EmployeeResponse.

Click Finish.

IBM Integration Bus V10 Workshop

🌐 New Message Map	<u>_</u> _×
Select map inputs and outputs Creates a map that can contain message inputs and outputs with the Proper and LocalEnvironment can be added to the map after creation.	iies folder. Optionally, database operations, message headers,
Filter map input names (? = any character, * = any String):	Filter map output names (? = any character, * = any String):
J Select map inputs	Select map outputs
EmployeeService_interface_and_maps     DFDL and XML Schemas     Body {http://schemas.xmlsoap.org/soap/envelope/}     Body {http://hrdb/ibadmin}     Ge DBResp {}     DBResp {}     Ge DBPARTMENT {http://hrdb/ibadmin}     Ge EMPLOYEE {http://hrdb/ibadmin}     Ge EmployeeResponse {http://hrdb/ibadmin}     Ge EmployeeResponse {http://hrdb/ibadmin}     Ge Envelope {http://schemas.xmlsoap.org/soap/envelope/}     Ge getEmployee {http://cmployeeService}     Ge getEmployeeResponse {http://EmployeeService}     Ge getEmployeeResponse {http://EmployeeService}     Ge Header {http://schemas.xmlsoap.org/soap/envelope/}     EmployeeResponse status://EmployeeService}     Ge Header {http://schemas.xmlsoap.org/soap/envelope/}	Construction of the second secon
Physical Location:	
Library: SemployeeService_interface_and_maps Path: Andb/iibadmin/HRDB.xsd Namespace: Antp://hrdb/iibadmin}	
•	< Back Next > Finish Cancel

4. The input and output assemblies will be created as shown.

📄 getEmployee_callable 🔀						
getEmployee_callable						
▼getEmployee_callable	🕸 🖓   🎼 🗳 🍕 📢	×  🖙 🏭 🐧 🐧 🏷	<b>I</b>	🗄 🏭 🚝 🛛 😂		
🖃 🚰 Message Assembly	employeeNumber			🖃 🔩 Message Assembly		EmployeeResponse
$\Rightarrow l \\ < Click to filter>$				$\Rightarrow$ <click filter="" to=""></click>		
		Overrides		🗄 📌 Properties	[01]	PropertiesType
🗄 📌 Properties	[01] PropertiesType	Move +		🗉 📌 EmployeeResponse	[11]	EmployeeResponseType
📌 employeeNumber	[11] employeeNumberType					

5. Connect the employeeNumber element to the EmployeeResponse element.

This will create a Local map transformation.

📄 *getEmployee_callable 🔀				
📄 getEmployee_callable				
▼getEmployee_callable	해 1억   🐔 🗳 🕸 14	×  🗧 🎝 🐧 🐧 🗎 🔳	in i	
🖃 📲 Message Assembly	employeeNumber		🖃 🚰 Message Assembly	EmployeeResponse
<pre></pre>			$\Rightarrow$ <click filter="" to=""></click>	
		Overrides	🕞 🗄 📌 Properties	[01] PropertiesType
🗉 📌 Properties	[01] PropertiesType		🕨 🗄 EmployeeResponse	[11] EmployeeResponseType
📌 employeeNumber	[11] employeeNumberType	Local map 🔻 💆		
-				

6. Change the Local map transformation to a Submap (click the blue down-arrow).

av 👔 Local map 🗸	<b>*</b>
	<b></b>
	⊡
	E Custom Transforms
	⊡···/───────────────────────────────

7. The map editor will detect that the Submap transformation requires the name of the submap file to be specified. In the Properties of the Submap transform, on the General tab, click the Browse button.

The list of available submaps will be shown. Only one submap will be available; this is the submap that has been used in several of the labs in this series, so select **getEmployee\_submap** and click OK.

🔒 getEmployee_ca	allable					
<b>▼getEmployee_</b>	callable	해 1회   삶 삶 14	×  🖗 🛱 🕅 🕅	0 🔁   🔳 🌆	1 🛅 🏭 📰	
🖃 🕍 Message A	ssembly	employeeNumber	]		□ 🔓 Message Assembly	EmployeeResponse
⇒ <click filter<="" th="" to=""><th>r&gt;</th><th></th><th></th><th></th><th>⇒i <click filter="" to=""></click></th><th></th></click>	r>				⇒i <click filter="" to=""></click>	
			Overndes		🗈 📌 Properties	[01] PropertiesType
🕀 📌 Propertie	es	[01] PropertiesType	Canal Assign		🗄 📌 EmployeeResponse	[11] EmployeeResponseType
📌 employe	eNumber	[11] employeeNumberType	<sup>1</sup> Submap	•	Submap Selection	<u>_     ×</u>
			-		Choose a resource:	
					Pattern (? = any characte	er, * = any string):
					Matching Resources:	
					getEmployees_subm	ap.map
					_	
🗖 Properties 🔀	Probler	ms 🗄 Outline 🧔 Tasks 🏢 D	eployment Log			
Transform - Sut	man					
Transform out						
General	File:	getEmployee_callable.map		Brow	se	
Cardinality	Map: 1			<b>•</b>		
Condition					In Folders:	
Sort					EmployeeService_int	erface_and_maps
Order						
Documentation						
					Show only applicable r	naps
					0	OK Creat
					U	OK Cancel

8. The error on the Submap transform will have been resolved, and the General tab will show the name of the submap file and mapname that will be invoked by the submap transform.

🔒 getEmployee_c	allable						
▼getEmployee_	_callable	ब्यू 🔍 🕷 🗞 🖏 🕼	×  🖙 🏦 🕅 🕅 1	) 🖩			
🖃 🔠 Message /	Assembly	employeeNumber			🖃 🔩 Message Assembly		EmployeeResponse
⇒ <click filte<="" td="" to=""><td>er&gt;</td><td></td><td></td><td></td><td><pre>Click to filter&gt;</pre></td><td></td><td></td></click>	er>				<pre>Click to filter&gt;</pre>		
🗉 📌 Properti	ies	[01] PropertiesType	Overrides  Move   Assian		e ⊕ ∰ Properties	[01] [11]	PropertiesType EmployeeResponseType
📌 employe	eeNumber	[11] employeeNumberType	Submap 👻	<u> </u>	, ,		
∢							
Properties 🔀	📳 Proble	ems 🗄 Outline 🖉 Tasks 🏢 De	eployment Log				e 1
Transform - Su	Ibmap						
General	File:	getEmployees_submap.map		Bro	New		
Cardinality	Map:	getEmployees_submap		•			
Variables							
Condition							
Sort							

Save (Ctrl-S) and close the map.

ī

## 2.3 Create the Callable Message Flow

1. Expand the application EmployeeService\_Callable, and open the message flow getEmployee.

The flow has been created for you, but has not been populated with any nodes.

Drop three nodes onto the flow editor, and connect as shown:

- CallableInput from the Callable Flow folder
- Mapping node named it getEmployee\_callable
- CallableReply from the Callable Flow folder

🖽 *getEmployee.msgflow 🔀	
👌 🔮 Palette	Flow Exerciser: 🔍 🖼 况 🔍
Ci Database	a
🥪 File	
🚌 Email	
🖓 TCPIP	CallableReply
CORBA	
🔄 Business Decisions	
🗟 CICS	getEmployee cauable
🕞 IMS	
💭 Validation	
Can Security	
🔞 Timer	
😂 Callable Flow 🛛 👳	
CallableInput	
CallableReply	
🖈 CallableFlowInvoke 🗨	

2. Highlight the CallableInput node and on the Basic Properties, set the Endpoint Name to **EmpServ\_get**.

(	
🖽 *getEmployee.msgflow 🔀	
👌 😳 Palette	Flow Exerciser: 🔍 🚰 🚧 🤞
Ci Database	
🐼 File	
腕 Email	
CPIP	
CORBA	(CallableInput)
Business Decisions	
CICS	
🖬 IMS 💌	
Graph User Defined Properties	
	1
🔲 Properties 🔀 🔝 Proble	ms 🗄 Outline 🖉 Tasks 🏢 Deploy
CallableInput Node P	roperties - CallableInput
Description	
Basic Endpoint Name	e* EmpServ_get
Monitoring	

3. Highlight the Mapping node getEmployee\_callable.

In the properties of the map node, use the Browse button to set the name of the map file to **getEmployee\_callable** in the Shared Library. (This is the map that you created earlier).

Ci Database Database File Email CORBA CORBA Dusiness Decisions CICS Cing IMS Graph User Defined Property	Callab	J leInput getEmploy	Cal ee_callable	ableReply
Properties 💥 📳 P	roblems 🗄 Outline	Tasks E Deployment Log		₫ ~ □ □
	opennes gerrinp			
Description				
Basic Mapping	routine* {default}:g	etEmployee_getEmployee_callable		Browse
Validation Transact	ion* Automatic			
Monitoring				
	•	Data Transformation Map Selec	tion	
	Filt	ter names (? = any character, * = an	y String):	
	Se	lect a Data Transformation Map		
		{default}:getEmployee callable in }	Shared Library EmployeeService	e interface
	-5	{default}:getEmployee_vvs in share	ed Library EmployeeService_in	terface_and
		d		
		cation		
		EmployeeService_interface_and	_maps	
		-		
	(	?	ОК	Cancel

4. The flow is now complete, so save and close the flow.

			~
C) Database	<b>_</b>		
🗟 File			
🚮 Email			
🖓 TCPIP			
CORBA		CallableInput	
E Business Dec	isions		
CICS		getEmployee_callable	
	<b>_</b>		
Graph User Def	fined Properties		
Properties 2	🛛 🔝 Problen	ns 🗄 Outline 🖉 Tasks 🥅 Deployment Log 🛃 🛃	~
🖶 Mapping	Node Proper	ties - getEmployee_callable	
Description			
Basic	Manaina routin	* (default) cotting and a collable in Charad Library Employee Carvice interface and many	Browse
Validation	Mapping rouun	<ul> <li>{ueraulty:getemployee_callable in shared Library Employeeservice_interface_and_maps</li> </ul>	Drowsen
Monitoring	Transaction*	Automatic	•
nonicoling			

Using Callable Flows

Provided by IBM BetaWorks

## 2.4 Create the Calling Message Flow

1. Expand the EmployeeService\_callingApp application, and open the EmpServ\_JSON\_Calling\_getEmployee message flow.

The flow does not have any nodes at this point. Drop the following nodes onto the flow:

- HTTP Input
- CallableFlowInvoke
- HTTP Reply

📴 getEmployee.msgflow	🔠 *EmpServ_JSO	N_Calling_getE	mployee	.msgflow 🖇	×)	
CPIP	Flow Exerciser:	• # V		€		
CORBA	<b>//</b>					
C IMS	( <u>HTTP Input</u> )					HTTP Reply
Co Timer C→ Callable Flow ↔				Callable	(2) leFlowInvoke	
CallableFlowInvoke						

- 2. Set the following node properties:
  - HTTP Input
    - Basic, Path suffix URL: /empServClient\_Calling\_getEmployee
    - Input Message Parsing, Message Domain: JSON
  - CallableFlowInvoke
    - Target Application: EmployeeService\_Callable
    - Target Endpoint Name: EmpServ\_get

3. Drop a new mapping node onto the flow, and name as extractEmployee.



- 4. Double-click to open the map, and set the input and output as follows:
  - Input: IBM supplied message models, JSON object
  - Output: DFDL and XML schemas, employeeNumber

Click Finish.

🌐 New Message Map	
Select map inputs and outputs Creates a map that can contain message inputs and outputs with the Properties fol LocalEnvironment can be added to the map after creation.	der. Optionally, database operations, message headers, and
Filter map input names (? = any character, * = any String):	Filter map output names (? = any character, * = any String):
J Select map inputs	Select map outputs
A EmployeeService_callingApp     A EmployeeService_interface_and_maps     B Supplied message models     B SOAP_Domain message model supplied by IBM)     R SOAP_Domain_Msg (SOAP domain message model supplied by IBM)     R SOAP_Domain_Msg (SOAP domain message model supplied by IBM)     R SOAP_Domain_Msg (SOAP domain message model supplied by IBM)     R SOAP_DOMAIN_MSG (SOAP domain message model supplied by IBM)     R SOAP_DOMAIN_MSG (SOAP domain message model supplied by IBM)     R SOAP_DOMAIN_MSG (SOAP domain message model supplied by IBM)     R SOAP_DOMAIN_MSG (SOAP domain message model supplied by IBM)     R SOAP_DOMAIN_MSG (SOAP domain message model supplied by IBM)     SOAP_SAPPLICATION SAPPLICATION SA	A     ComplexeService_interface_and_maps     DFDL and XML Schemas     DFDL and XML Schemas     DFDL and XML Schemas.xmlsoap.org/soap/envelope/}     Government     DeResp 0     Government     DeRATMENT {http://hrdb/ibadmin}     Government     Government     Government     DerARTMENT {http://hrdb/ibadmin}     Government     Governm
Physical Location:	
Library: Path: Namespace: AmployeeNumber.xsd	
3	< Back Next > Finish Cancel

5. In the map editor, expand the JSON input message assembly.

Right-click the "any" element, and select "Add User-Defined".

ر چ 🖃	ISON	[11]	JSONMsgType	
е	e Padding		string	
⊟ よ	choice of cast items	[11]		
1	뢷 Data	[11]	anyType	
	e Data	[11]	JSONObject	
	20 may	Fo #1		-0
	Undo			Ľ
	Redo			L .
	Revert			L .
	Cut		Ctrl+X	L
	Сору		Ctrl+C	L .
	Paste		Ctrl+V	L
	Delete			
	🕼 Add output		Ctrl+Shift+N, O	
	Sector Add environment n	napping		
	Open Information I	Popup	Ctrl_shift_I	
	Open Declaration	ropup	F3	L .
	🞦 Cast			
	🛃 Add User-Defined		Ctrl+Shift+C	

6. Set the name of the new element to **empNumber**. The type can be left as string.

This enables the map to parse an incoming JSON message containing a single element named empNumber.

[11]	JSONMsgType
[01]	string
[11]	
[11]	anyType
[11]	JSONObject
[0*]	
[11]	
[11]	string
	[11] [01] [11] [11] [11] [0*] [11]

7. Connect the new element **empNumber** to the output **employeeNumber**. This will create a Move transform.

EmpServ_JSON_Calling_getEmploy	ee_extractEmployee			
EmpServ_JSON_Calling_getEmp	ployee_extractEmployee	🕸 🗐 🕷 🗳 🕸	× ₽₽	II 🕯 🖼
🖃 📩 Message Assembly	JSON		🖃 🖧 Message Assembly	emp
<pre>→i</pre> <click filter="" to=""></click>			$\rightarrow i$ <click filter="" to=""></click>	
		Overrides	🗄 🗄 📌 Properties	[01] Prop
🗄 📌 Properties	[01] PropertiesType		🖉 employeeNumber	[11] emp
🖃 📌 JSON	[11] JSONMsgType			
e Padding	[01] string			
🖃 📇 choice of cast items	[11]			
🖳 Data	[11] anyType			
🖃 ųe Data	[11] JSONObject			
🖃 📇 choice of cast items	[0*]			
🖺 any	[11]			
ម៉ីខ្លាំ empNumber	[11] string	Move 🗸		
		_		

- 8. This is all that is required for this map, so save and close the map.
- 9. Drop a second map onto the flow, named **buildJSONOutput**.

//B			HTTP Reply
	extractEmployee	CallableFlowInvoke	build JSONOutput

- 10. Open the second map, and set the input and output as follows:
  - Input: DFDL and XML schemas, EmployeeResponse
  - Output: IBM supplied message models, **JSON object**

Click Finish.

🌐 New Message Map	
Select map inputs and outputs Creates a map that can contain message inputs and outputs with the Properties f LocalEnvironment can be added to the map after creation.	older. Optionally, database operations, message headers, and
Filter map input names (? = any character, * = any String):	Filter map output names (? = any character, * = any String):
J Select map inputs	Select map outputs
EmployeeService_callingApp    EmployeeService_interface_and_maps	
?	< Back Next > Finish Cancel

11. In the output message assembly, expand the JSON output, and right-click the "**any**" element.

### Select Cast.

	🖃 📇 Message Assembly	,		1SON		
	⇒ <click filter="" to=""></click>			5501		
	🗈 📌 Properties		[01]	PropertiesType		
	🖃 📌 JSON		[11]	JSONMsgType		
	e Padding		[01]	string		
	🖃 🖧 choice of cast	items	[11]			
	<mark>멸</mark> Data		[11]	anyType		
	🖃 🗓 Data		[11]	JSONObject		
$\sim$	ny 🐔 The test test test test test test test te	(	Undo			
		F	Redo Revert			
		(	Cut		Ctrl+	×.
		0	Сору		Ctrl+	C
		F	Paste		Ctrl+	-V
		[	Delete			
		ا 🛟 ا 🚑	Add output Add environ	ment mapping	Ctrl+	Shift+N, O
		i (	Open Inforn Open Declar	nation Popup ration	Ctrl+ F3	Shift+I
		<u>1</u>	Cast			
		e <sup>8</sup> /	Add User-De	efined	Ctrl+	Shift+C
		2-1	Add Connec	tion		

12. In the Type Selection window, select EmployeeResponse, and click OK.

	Type Selection
	Choose a type (? = any character, * = any string):
	Matching types:
	€ Body € DBResp
	© DEPARTMENT
	e employee
(	
	e Envelope
	e Fault
	e getEmployeeResponse
	e Header
	Qualifier:
	ttp://hrdb/iibadmin (EmployeeService_interface_a
	OK Cancel

13. In the map editor, connect **EmployeeResponse** to **EmployeeResponse**. This will create a Move transform.

This will transform the EmployeeResponse element into the same element, but wrapped in a JSON output message.

Save and close the map.

EmpServ_JSON_Calling_get	Employee	_buildJSONOutput						
EmpServ_JSON_Calling_g	etEmplo	yee_buildJSONOutput	pi 🗊		×   🖛 🛱	🏷 🏷 🏷   🖩 🏣 😫		
□ 🚰 Message Assembly		EmployeeResponse				🖃 🔩 Message Assembly	JSON	
⇒ <click filter="" to=""></click>						Click to filter>		
				Overrides		🗉 📌 Properties	[01] PropertiesTyp	e
				Assister - 10		🖃 🐙 JSON	[11] JSONMsgType	
🗈 📌 Properties	[01]	PropertiesType				e Padding	[01] string	
						🖃 🖧 choice of cast items	[11]	
						🖳 Data	[11] anyType	
	[1 1]	EmployeePersonceType		Move 👻		🖃 🗓 Data	[11] JSONObject	
a ga employeercesponse	[1.1.4]	Employeercesponserype				🖃 🖧 choice of cast items	[0*]	
						any 🖉	[11]	
						🗉 🗓 EmployeeResponse	[11] EmployeeResp	onseType

- 14. Highlight the CallableFlowInvoke node, and set the following properties:
  - Target Application: EmployeeService\_Callable
  - Target Endpoint Name: EmpServ\_get

🖽 getEmploye	e.msgflow	Serv_JSON_Calling_getEmployee.msgflow 🕅						
A Palette     A Palette	MQ							
HTTP     HTTP     HTTP     Service:     SCA     WebSphere /     GR Routing     NET	Adapters	Input						
Graph User Defined Properties								
Callabl	eFlowInvoke Node P	roperties - CallableFlowInvoke						
Description	Target Application*	EmployeeService Callable						
Monitorina	Target Application							
	rarget Enupoint Name*							
	Request timeout (sec)	120						
	Call Preference	Prefer local calls						

15. Finally, connect the nodes as shown, then save the message flow.

CallableFlowInvoke
--------------------

### 2.5 Deploy the shared library and callable application

1. To make sure there are no conflicts with earlier scenarios, remove all currently deployed artefacts in the TESTNODE node.

Right-click the default server, and select Delete, All Flows And Resources.



2. Deploy the shared library **EmployeeService\_interface\_and\_maps**.

Deploy the application **EmployeeService\_Callable**.

🔚 Application Development 🙁 💐 Patterns Explorer						
Application Development						
EmployeeService_Callable      EmployeeService_callingApp      Flows      Maps      Referenced Ubraries      Other Resources      EmployeeService_interface_and_maps      Independent Resources						
器 Integration Nodes 🕱 🖁 Integration Registries 🛭 😪 Data Proj						
□····       Integration Nodes         □····       TESTIODE_iibuser         □····       Gefault         □····       EmployeeService_Callable         □····       ImployeeService_interface_and_maps						

## 3. Test the Calling Application

## 3.1 Test with Flow Exerciser

1. Open the message flow EmpServ\_JSON\_Calling\_getEmployee.

Click the red button to invoke the Flow Exerciser.

F	Flow Exerciser:	■ ¼ ] €	Q	
	HTTP Input	extractEmployee	CallableFlowInvoke	HTTP Reply build JSONOutput

2. Name the new message "Employee 000010".

Set the message payload to

```
{"empNumber":"000010"}
```

Click Send.

Send Message Send Message Create or select a message to send to information.	the flow. Click the message category header (e.g. Input Messages) for more          Name       Employee 000010         Name       Input Location:         Input Location:       HTTP Input         Message Details       Edit, type, or import a message.         Edit, type, or import a message.       Import from file         ["empNumber":"000010"}       Import from file         Show in hexadecimal viewer (Read Only)       Export Source	
?	A	pply Revert

3. When the flow has executed, highlight the "Received HTTP reply" message.

The returned data should show the row returned from the HRDB database. Since the response will be in JSON format, this dialogue will not be able to format the data fully.

Progress Information	<u>_ 0 ×</u>
E Invoke Message Flow (Employee 000010)	
Message flows deployment successfully completed	
En Starting	
Received HTTP reply message for "HTTP Input"	
Stopped	
<pre>{"EmployeeResponse":{"out":"http:///hrdb/viibadmin","DBResp": {"UserReturnCode":0,"RowsRetrieved":1,"RowsAdded":0,"RowsUpdated":0,"RowsDelet Code_ErrorCode":0,"SQLState_SQLState":{},"SQL_Error_Message":{}},"EMPLOYEE": {"EMPNO":"000010","FIRSTNME":"CHRISTINE","MIDINIT":"I","LASTNAME":"HAAS","WOR 00","PHONENO":"3978","HIREDATE":"1995-01-01","JOB":"PRES ","EDLEVEL":18,"SEX":"F","BIRTHDATE":"1963-08- 24","SALARY":152750,"BONUS":1000,"COMM":4220}}}</pre>	ed":0,"SQL KDEPT":"A
	Close

## 3.2 Test with SOAPUI

1. Open SOAPUI from the Start menu.

Open the project EmployeeService - CallableFlow, fully expand and open the request "Employees matching 0020".

🔶 SoapUI 5.0.0						
<u>File T</u> ools <u>D</u> esktop <u>H</u> elp						
🖲 🕲 🗟 🔷 💥 🐻						
EmployeeService_PrebuiltWorkspace						
DepartmentService_REST						
EmployeeService						
🖶 🖳 EmployeeService - CallableFlow						
ia Intp://localhost:7800						
🚊 🖳 Get Employee (EmpServClient_Calling_getEmployee) [/						
ian <sup>e</sup> s <sup>a</sup> getEmployee						
Employee 000010						
ST Employees matching 0020						
🖶 🖳 EmployeeService - MessageFlowSecurity (BasicAuth)						
🗄 🖳 EmployeeService - MessageFlowSecurity (JSONClient, BasicAut						
🗄 🖳 🗄 EmployeeService - MessageFlowSecurity (SSL Test Harness)						
🗄 🖷 🔁 EmployeeService - MessageFlowSecurity (SSL, X. 509)						
👜 🖳 EmployeeService_JSONClient - Async						
B Enderstein teologische deutschen						

2. Click the green arrow to send the message.

This message ("0020") tells the submap to retrieve all employees which match this number. Two employee records will be retrieved.

Click the JSON tab to see the output in JSON format.

RE	Employees matching 0020					₫₫	X
	Method Endpoint Resource			Parameters		~	
	POST   http://localhost:7800	-	/empSer	vClient_Calling_getEmployee		+ 14	C
G.	+ X URD 50	6					
Ines		X	1日(	"EmployeeResponse": {			
Rec	Name Value Style Level	Z	2	"out": "http://hrdb/llbadmin",			
A S		ыN	4	"HeerPeturpCode": 0			
Ra		Ę	5	"RowsRetrieved": 2.			
		É	6	"RowsAdded": 0,			
		2	7	"RowsUpdated": 0,			
	×¥	Ra	8	"RowsDeleted": 0,			
	Required: Sets if parameter is required	-	9	"SQLCode_ErrorCode": 0,			
			10	"SQLState_SQLState": {},			
	туре:		11	"SQL_Error_Message": {}			
	Options:		13	PMDLOVER"-			
			14日	I I			
	▼		15	"EMPNO": "000020",			
			16	"FIRSTNME": "MICHAEL",			205
			17	"MIDINIT": "L",			
			18	"LASTNAME": "THOMPSON",			222
	{"empNumber":"0020"}		19	"WORKDEPT": "B01",			
			20	"PHONENO": "3476",			
			22	"JOR"- "MANAGED "			
			23	"EDLEVEL": 18			
			24	"SEX": "M",			
			25	"BIRTHDATE": "1978-02-02",			
			26	"SALARY": 94250,			
			27	"BONUS": 800,			
			28	"COMM": 3300			
			29	1.			
			30 []	1 "FMBNO"- "000200"			
			32	"FIRSTNME": "DAVID"			
			33	"MIDINIT": " ",			
			34	"LASTNAME": "BROWN",			
			35	"WORKDEPT": "D11",			
			36	"PHONENO": "4501",			
			37	"HIREDATE": "2002-03-03",			
			38	"JOB": "DESIGNER",			
			39	"EDLEVEL": 16,			-
			6				
	Hea Attach Represen JMS JMS Pro	ł	Headers (	5) Attachments (0) SSL Info Representations	(1) Schema (conflicts) JMS (0		
resp	onse time: 1302ms (709 bytes)					1:1	

## END OF LAB GUIDE

Provided by IBM BetaWorks