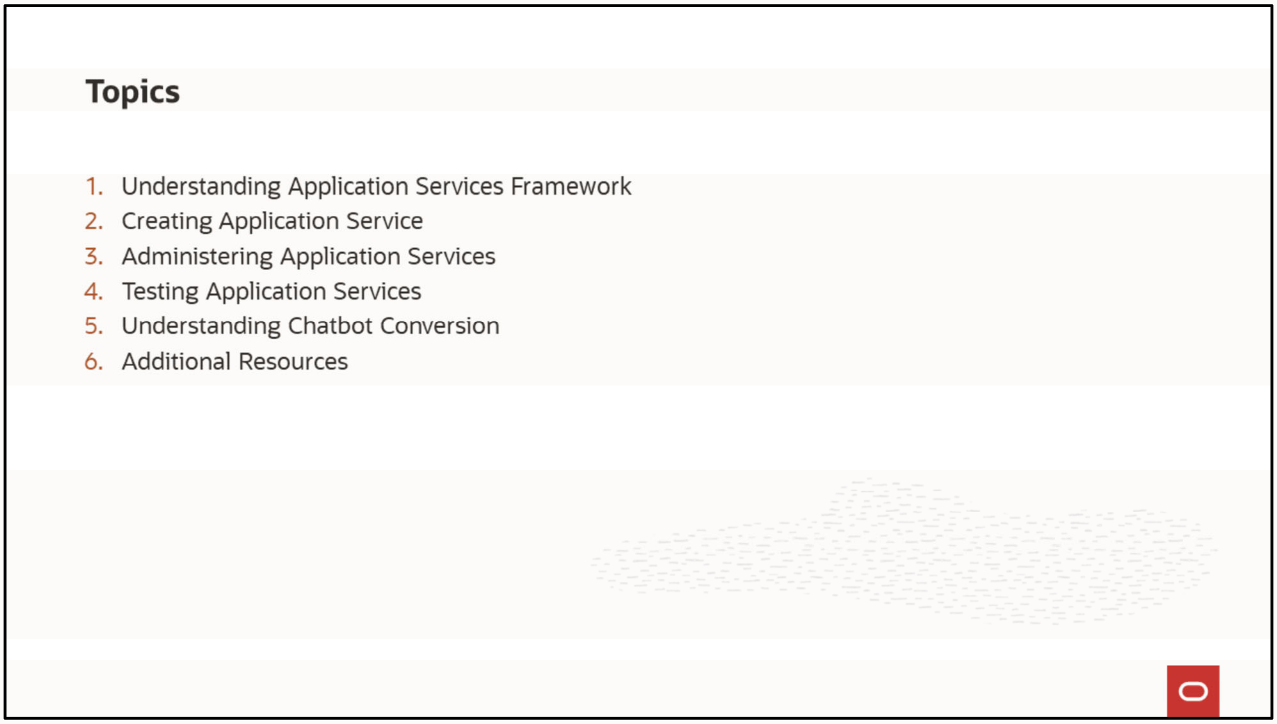
Spotlight on

Application Services Framework for PeopleTools 8.59

ORACLE

Application Services Framework Enhancements

Application Services Framework was introduced in PeopleTools 8.57 to provide easy to build and maintain REST services used only by PeopleSoft digital assistant skills. Prior to 8.59, these skills were referred to as chatbots. With PT 8.59 Application Services Framework has been enhanced to create application services that conform to Oracle REST standards and can be used anywhere Provider REST services are supported.



Topics

1. Understanding Application Services Framework
2. Creating Application Service
3. Administering Application Services
4. Testing Application Services
5. Understanding Chatbot Conversion
6. Additional Resources

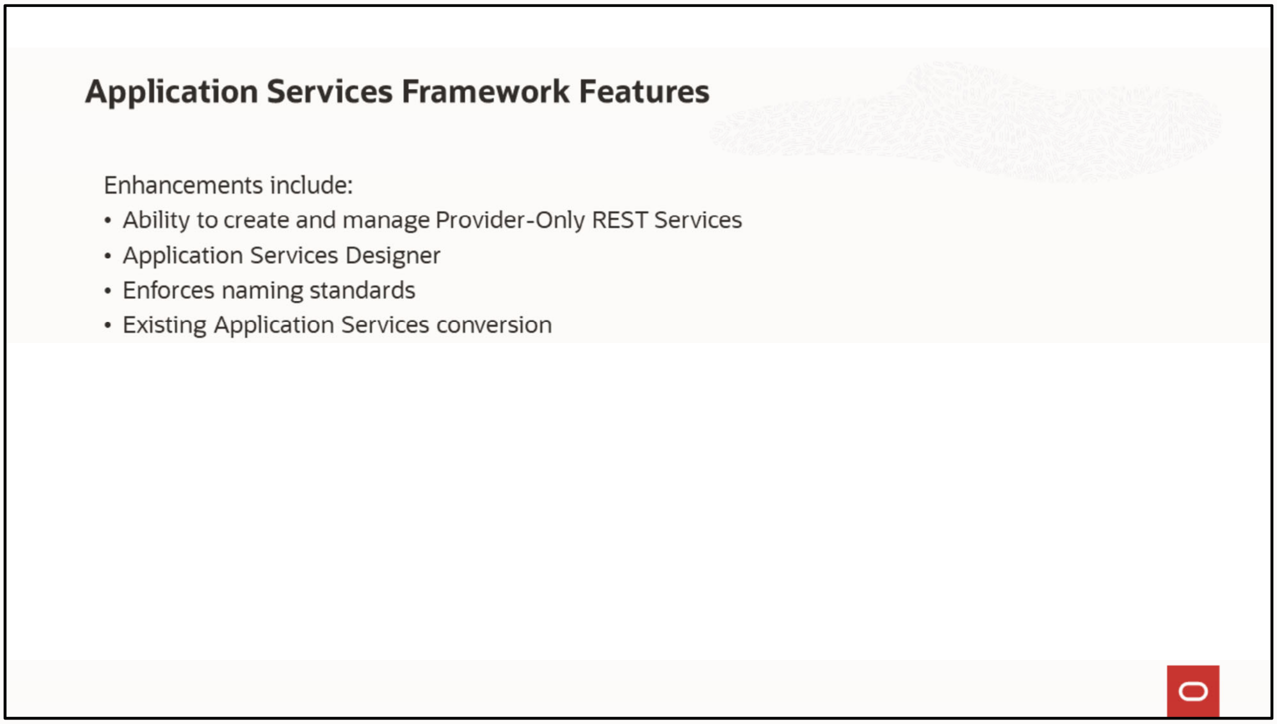
In this video, we will introduce the Application Services Framework enhancements and demonstrate how to:

- Create a new Application Service.
- Administer Application Services.
- Test Application Services.

And describe the chatbot conversion when upgrading to PeopleTools 8.59.

Understanding Application Service

With the Application Services Framework, you can create and manage Provider-only REST Services.
The Application Services Designer simplifies the process of maintaining REST services and enforces naming standards.
PeopleSoft-delivered Application Services including skills will be modified to meet the Oracle REST standard and delivered when the PeopleSoft Image for your application is available on PeopleTools 8.59.



Application Services Framework Features

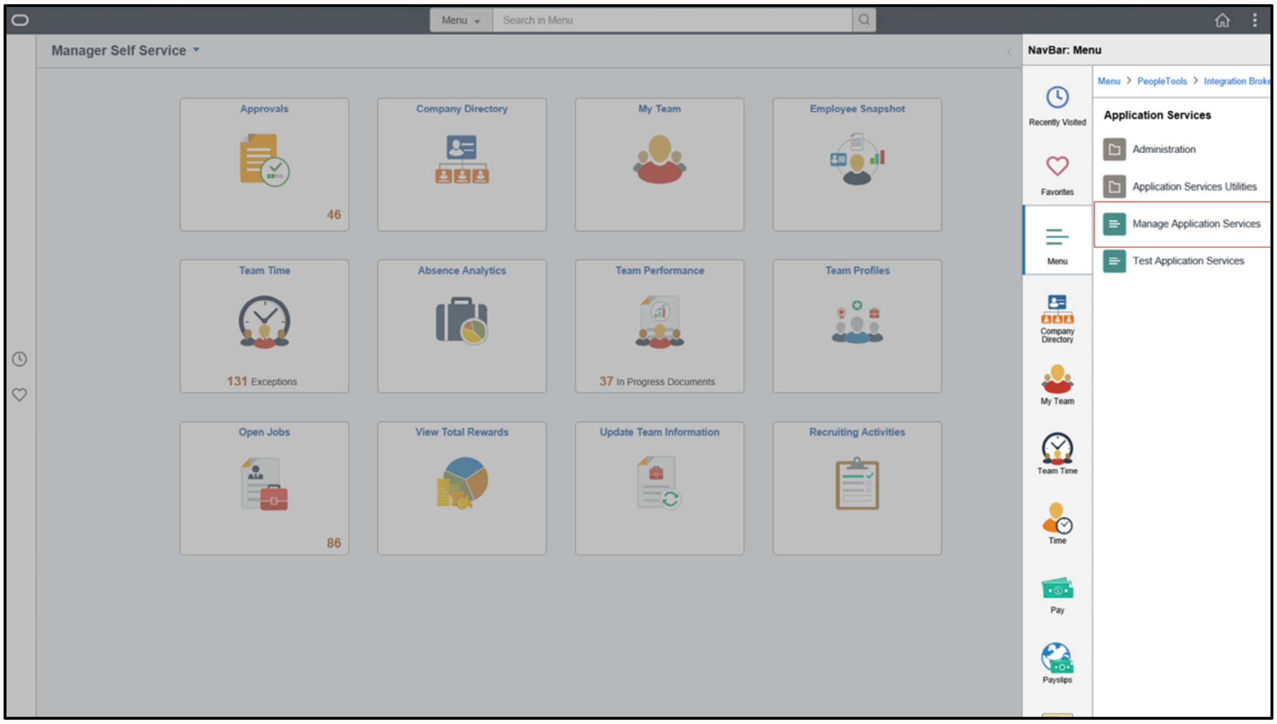
Enhancements include:

- Ability to create and manage Provider-Only REST Services
- Application Services Designer
- Enforces naming standards
- Existing Application Services conversion

With the Application Services Framework, you can create and manage Provider-only REST Services.
The Application Services Designer simplifies the process of maintaining REST services and enforces naming standards.
PeopleSoft-delivered Application Services including skills will be modified to meet the Oracle REST standard and delivered when the PeopleSoft Image for your application is available on PeopleTools 8.59.

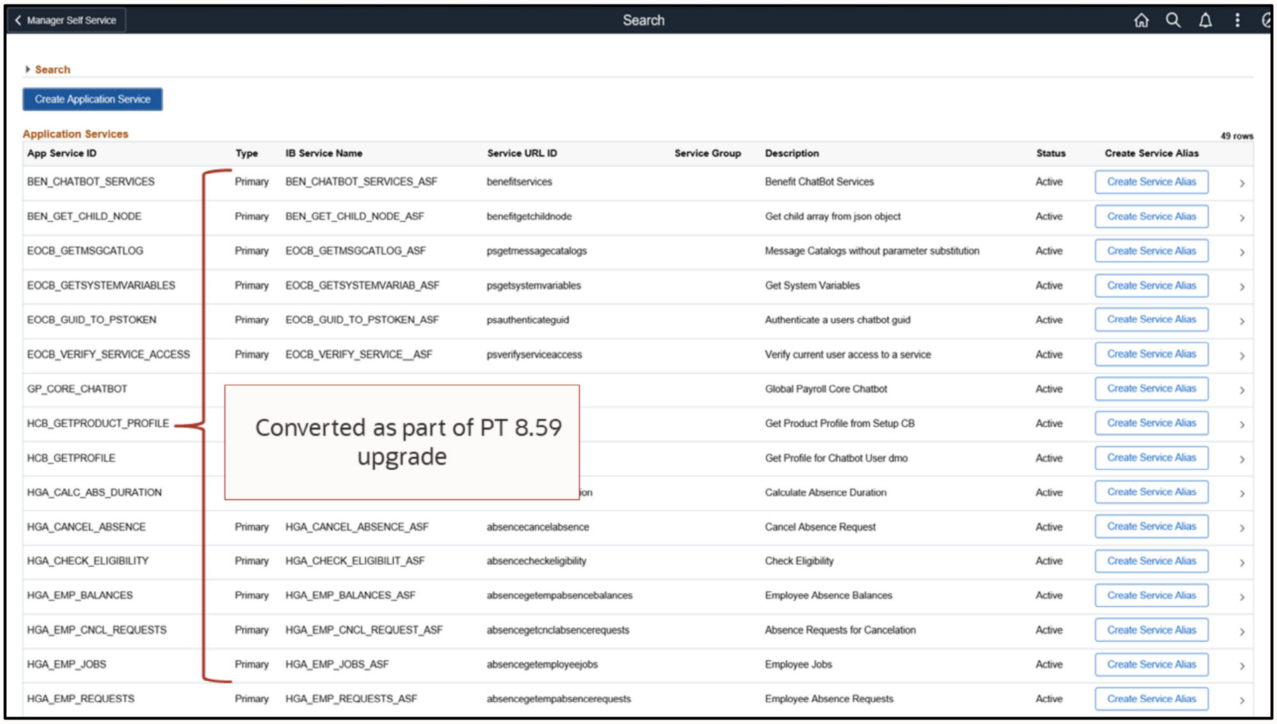
Creating Application Service

We will begin by creating a new Application Service.



To access Application Services from the menu, select PeopleTools, Integration Broker, Application Services.

Select Manage Application Services.



▸ Search

Create Application Service

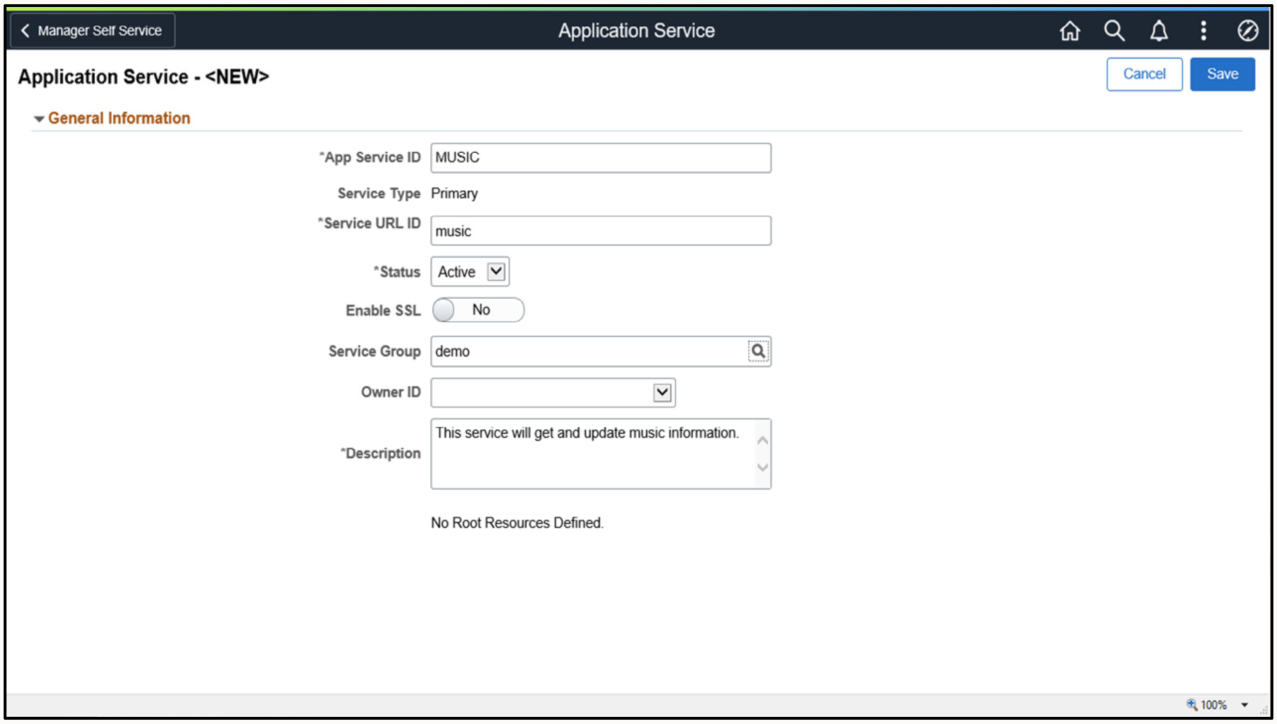
Application Services

49 rows

App Service ID	Type	IB Service Name	Service URL ID	Service Group	Description	Status	Create Service Alias	
BEN_CHATBOT_SERVICES	Primary	BEN_CHATBOT_SERVICES_ASF	benefitservices		Benefit ChatBot Services	Active	Create Service Alias	›
BEN_GET_CHILD_NODE	Primary	BEN_GET_CHILD_NODE_ASF	benefitgetchildnode		Get child array from json object	Active	Create Service Alias	›
EOCB_GETMSGCATLOG	Primary	EOCB_GETMSGCATLOG_ASF	psgetmessagecatalogs		Message Catalogs without parameter substitution	Active	Create Service Alias	›
EOCB_GETSYSTEMVARIABLES	Primary	EOCB_GETSYSTEMVARIAB_ASF	psgetsystemvariables		Get System Variables	Active	Create Service Alias	›
EOCB_GUID_TO_PSTOKEN	Primary	EOCB_GUID_TO_PSTOKEN_ASF	psauthenticateguid		Authenticate a users chatbot guid	Active	Create Service Alias	›
EOCB_VERIFY_SERVICE_ACCESS	Primary	EOCB_VERIFY_SERVICE__ASF	psverifyserviceaccess		Verify current user access to a service	Active	Create Service Alias	›
GP_CORE_CHATBOT					Global Payroll Core Chatbot	Active	Create Service Alias	›
HCB_GETPRODUCT_PROFILE					Get Product Profile from Setup CB	Active	Create Service Alias	›
HCB_GETPROFILE					Get Profile for Chatbot User dmo	Active	Create Service Alias	›
HGA_CALC_ABS_DURATION			ion		Calculate Absence Duration	Active	Create Service Alias	›
HGA_CANCEL_ABSENCE	Primary	HGA_CANCEL_ABSENCE_ASF	absencecancelabsence		Cancel Absence Request	Active	Create Service Alias	›
HGA_CHECK_ELIGIBILITY	Primary	HGA_CHECK_ELIGIBILIT_ASF	absencecheckeligibility		Check Eligibility	Active	Create Service Alias	›
HGA_EMP_BALANCES	Primary	HGA_EMP_BALANCES_ASF	absencegetempabsencebalances		Employee Absence Balances	Active	Create Service Alias	›
HGA_EMP_CNCL_REQUESTS	Primary	HGA_EMP_CNCL_REQUEST_ASF	absencegetcnclabsencerequests		Absence Requests for Cancelation	Active	Create Service Alias	›
HGA_EMP_JOBS	Primary	HGA_EMP_JOBS_ASF	absencegetemployeejobs		Employee Jobs	Active	Create Service Alias	›
HGA_EMP_REQUESTS	Primary	HGA_EMP_REQUESTS_ASF	absencegetempabsencerequests		Employee Absence Requests	Active	Create Service Alias	›

Converted as part of PT 8.59 upgrade

Prior to 8.59 Application Services were only used to create digital assistant skills, referred to as chatbots. As a part of the PeopleTools Upgrade to 8.59, these services are converted to the new format and will run in PeopleTools 8.59, however they will not be compliant as REST Web Services.
We will discuss the conversion later in this video.
To create a new Application Service, select Create Application Service.



Application Service - <NEW>

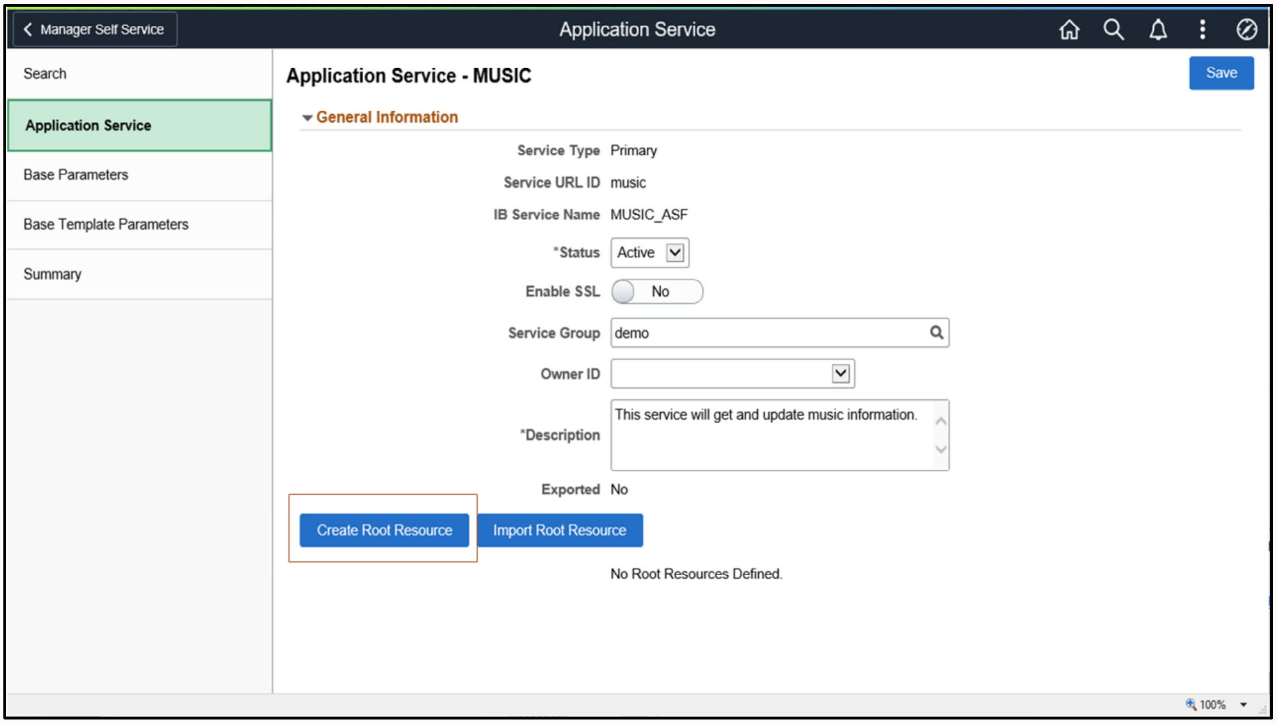
General Information

*App Service ID	MUSIC
Service Type	Primary
*Service URL ID	music
*Status	Active
Enable SSL	No
Service Group	demo
Owner ID	
*Description	This service will get and update music information.

No Root Resources Defined.

For this demo we will create an application service for music.
By default new Application Services are primary.
The service URLID name must be lowercase.
The default status is active.
You can enable SSL.
Optionally you can assign a service group and owner id.
We'll create a service group – demo.
Enter a description for the service and then click Save.

When you save the Application Service, the system will verify that the service ID meets the Oracle standards and that it is unique.



Application Service - MUSIC

Service Type Primary

Service URL ID music

IB Service Name MUSIC_ASF

*Status Active

Enable SSL No

Service Group demo

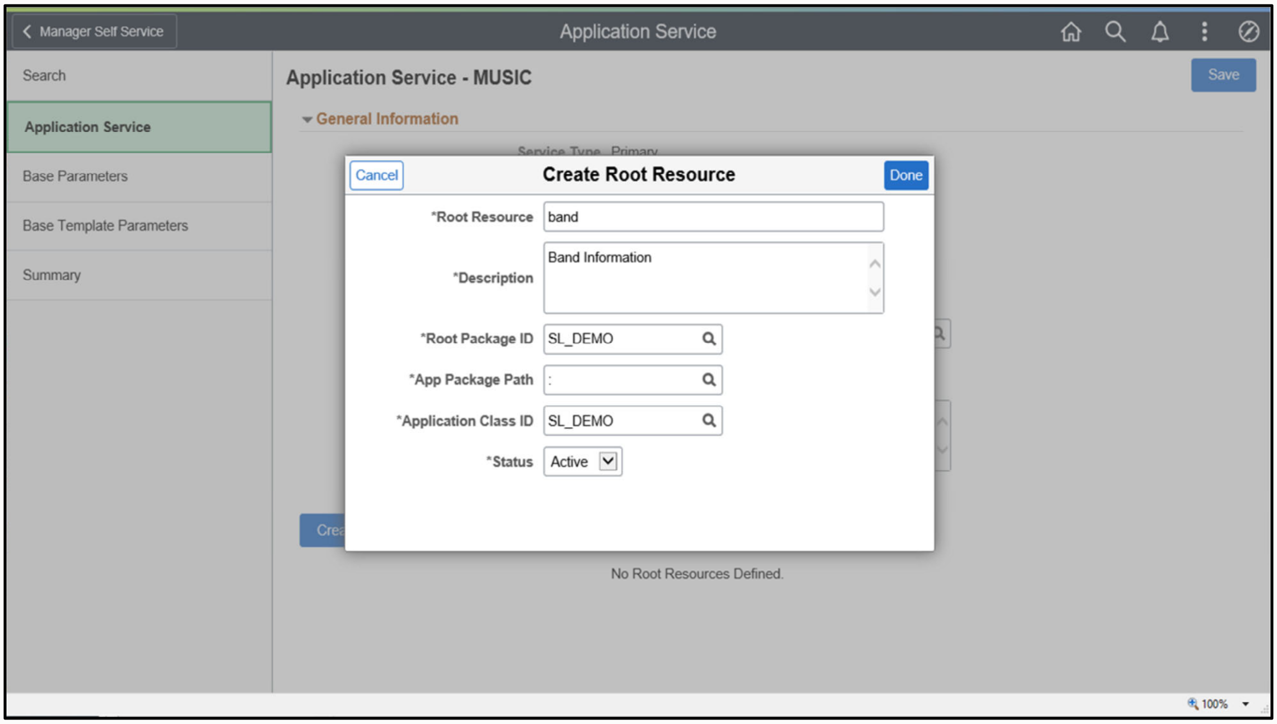
Owner ID

*Description This service will get and update music information.

Exported No

No Root Resources Defined.

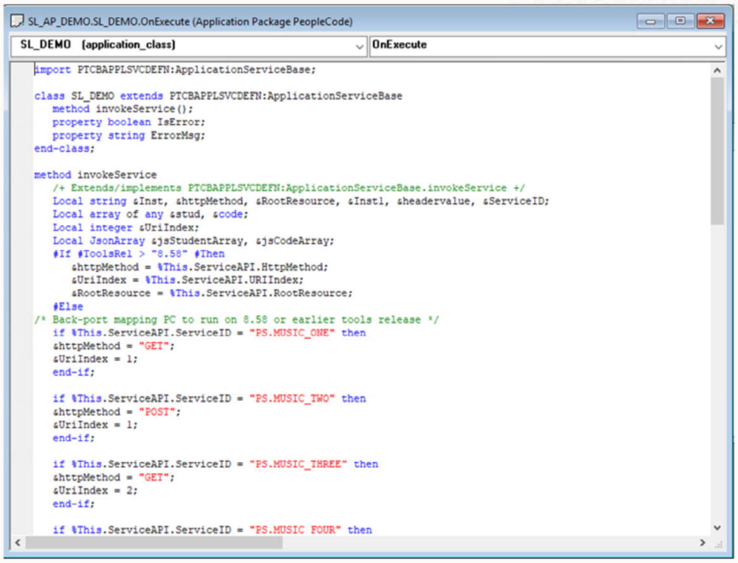
The save process generates the necessary IB metadata, updates the Service Name and enables the Create Root Resource and Import Root Resource buttons.
We will create a new root resource.



*Root Resource	band
*Description	Band Information
*Root Package ID	SL_DEMO
*App Package Path	:
*Application Class ID	SL_DEMO
*Status	Active

The root resource is the URL to the web service.
Enter a root resource name and description.
Select the application package that contains the application logic to execute.

Sample Application Class



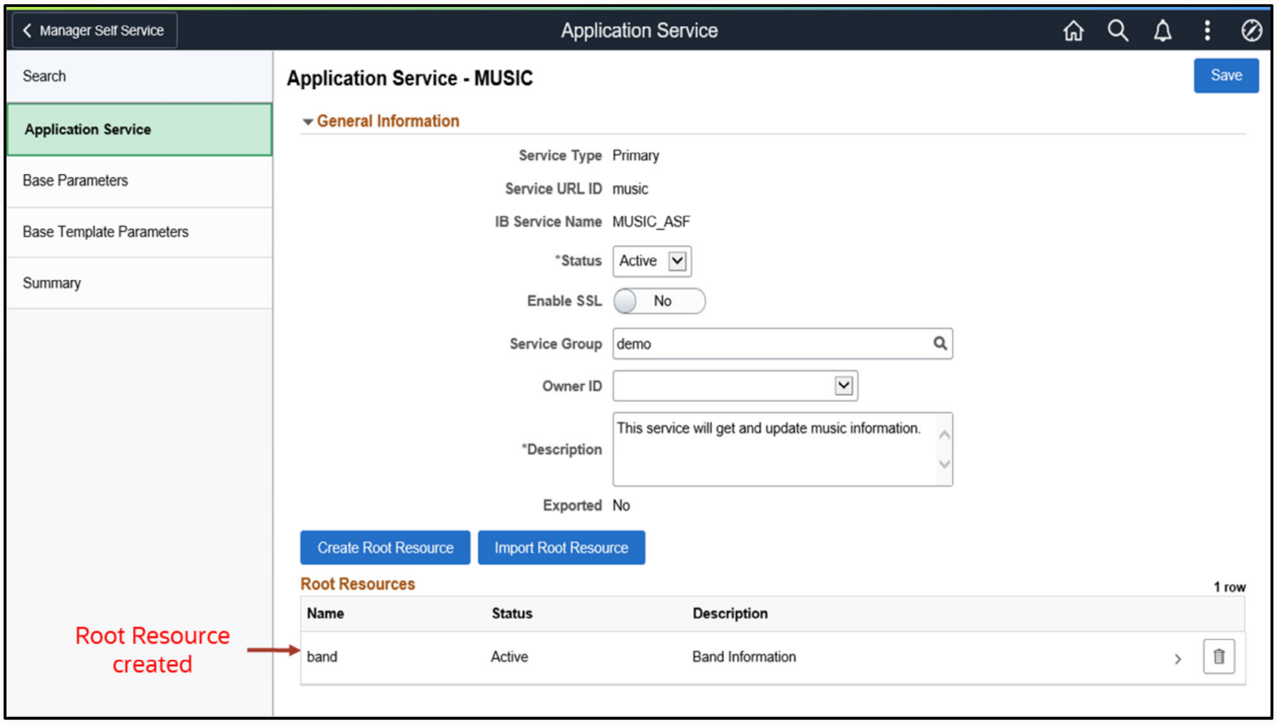
SL_AP_DEMO.SL_DEMO.OnExecute (Application Package PeopleCode)
SL_DEMO (application_class)
OnExecute

```
import PTCBAPPLSVCDEFN:ApplicationServiceBase;

class SL_DEMO extends PTCBAPPLSVCDEFN:ApplicationServiceBase
   method invokeService();
   property boolean IsError;
   property string ErrorMsg;
end-class;

method invokeService
   /+ Extends/implements PTCBAPPLSVCDEFN:ApplicationServiceBase.invokeService +/
   Local string &Inst, &httpMethod, &RootResource, &Instl, &headervalue, &ServiceID;
   Local array of any &stud, &code;
   Local integer &UriIndex;
   Local JsonArray &jsStudentArray, &jsCodeArray;
   #If #ToolsRel > "8.58" #Then
      &httpMethod = %This.ServiceAPI.HttpMethod;
      &UriIndex = %This.ServiceAPI.URIIndex;
      &RootResource = %This.ServiceAPI.RootResource;
   #Else
/* Back-port mapping PC to run on 8.58 or earlier tools release */
   if %This.ServiceAPI.ServiceID = "PS.MUSIC_ONE" then
   &httpMethod = "GET";
   &UriIndex = 1;
   end-if;

   if %This.ServiceAPI.ServiceID = "PS.MUSIC_TWO" then
   &httpMethod = "POST";
   &UriIndex = 1;
   end-if;

   if %This.ServiceAPI.ServiceID = "PS.MUSIC_THREE" then
   &httpMethod = "GET";
   &UriIndex = 2;
   end-if;

   if %This.ServiceAPI.ServiceID = "PS.MUSIC FOUR" then
```

The application class must extend the PCBAPPLSVCDEFN:ApplicationServiceBase application class and then implement the invokeService() method to define the application logic.

Use the %This.ServiceAPI property to read and write parameters and other Application Services metadata.

After you have entered the Application Package information, click Done for the root resource.



Application Service - MUSIC

General Information

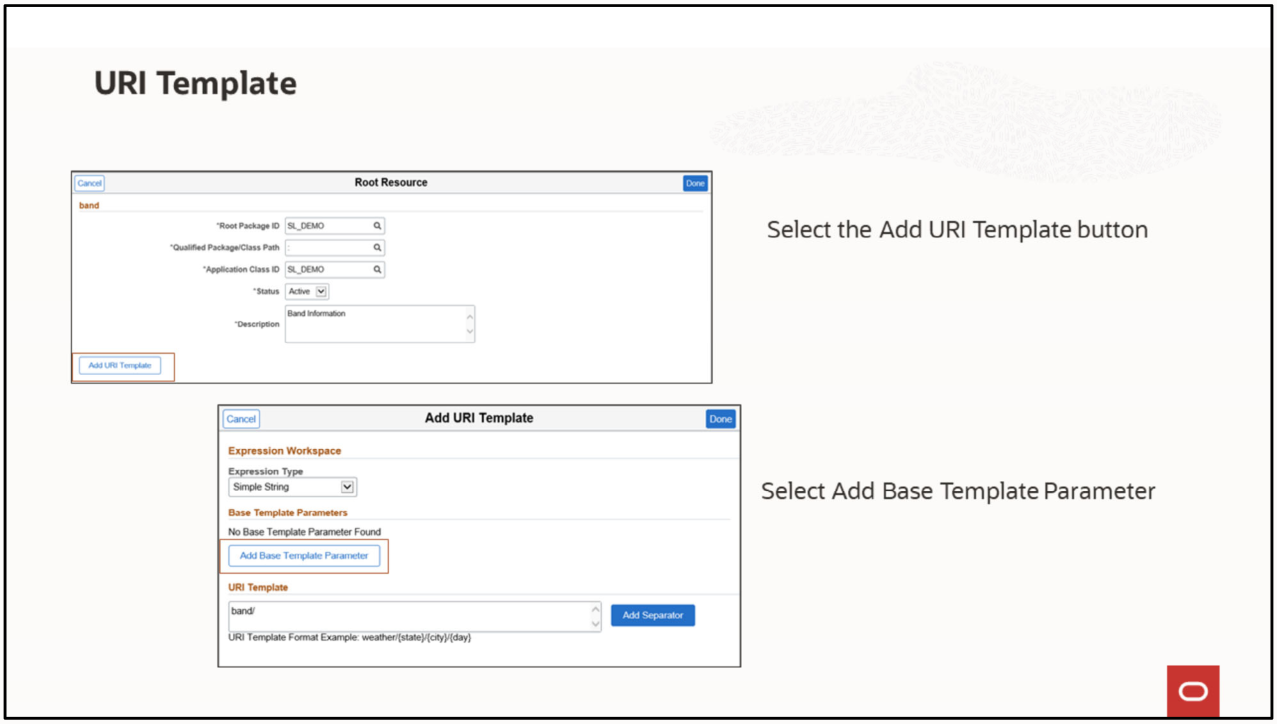
Service Type	Primary
Service URL ID	music
IB Service Name	MUSIC_ASF
*Status	Active
Enable SSL	No
Service Group	demo
Owner ID	
*Description	This service will get and update music information.
Exported	No

Root Resources 1 row

Name	Status	Description
band	Active	Band Information

Root Resource created

After creating the root resource, select the resource to add URI Templates.



URI Template

Root Resource

band

*Root Package ID: SL_DEMO

*Qualified Package/Class Path: :

*Application Class ID: SL_DEMO

*Status: Active

*Description: Band Information

Select the Add URI Template button

Add URI Template

Expression Workspace

Expression Type: Simple String

Base Template Parameters

No Base Template Parameter Found

URI Template: band/

URI Template Format Example: weather/{state}/{city}/{day}

Select Add Base Template Parameter

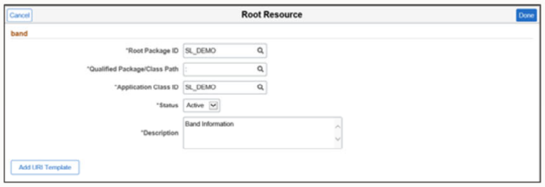
URI templates are strings that can be transformed into URIs after embedded variables are substituted.

Select the Add URI Template button.

By default, the expression type is Simple String, which is used for Provider services. Currently no base template parameters exist for this Application Service.

Select Add Base Template Parameter.

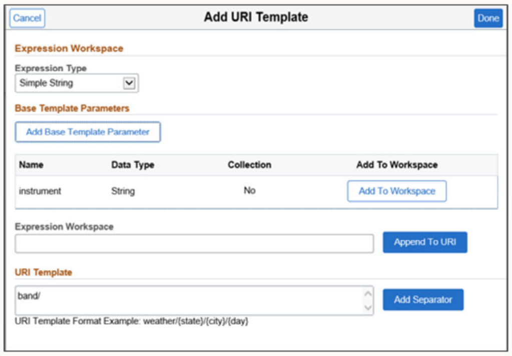
Add Base Parameter



band

*Root Package ID: SL_DEMO
*Qualified Package/Class Path: :
*Application Class ID: SL_DEMO
*Status: Active
*Description: Band Information

Add a parameter for instrument



Expression Workspace

Expression Type: Simple String

Base Template Parameters

Name	Data Type	Collection	Add To Workspace
instrument	String	No	Add To Workspace

Expression Workspace

URI Template

band/

URI Template Format Example: weather/{state}/{city}/{day}

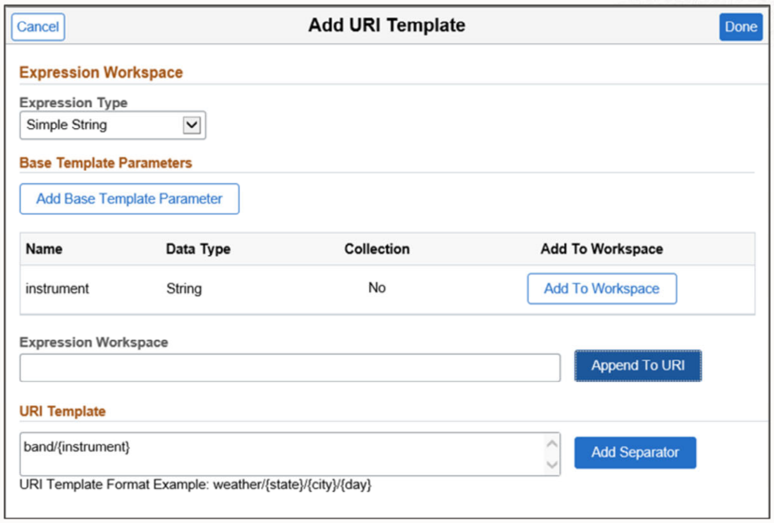
Parameter available

We will add a parameter for instrument.

The parameter name must adhere to Oracle's naming standard.
Select the data type and length if appropriate.
Select the collection checkbox if this parameter will be part of a collection of values.
Enter a description, then click Done to add the parameter.

Once defined the parameter is available to be used in all the URI templates for this application service.

Form URI



Add URI Template

Expression Workspace

Expression Type: Simple String

Base Template Parameters

Name	Data Type	Collection	Add To Workspace
instrument	String	No	Add To Workspace

Expression Workspace:

URI Template

band/{instrument}

URI Template Format Example: weather/{state}/{city}/{day}

Add to Workspace,

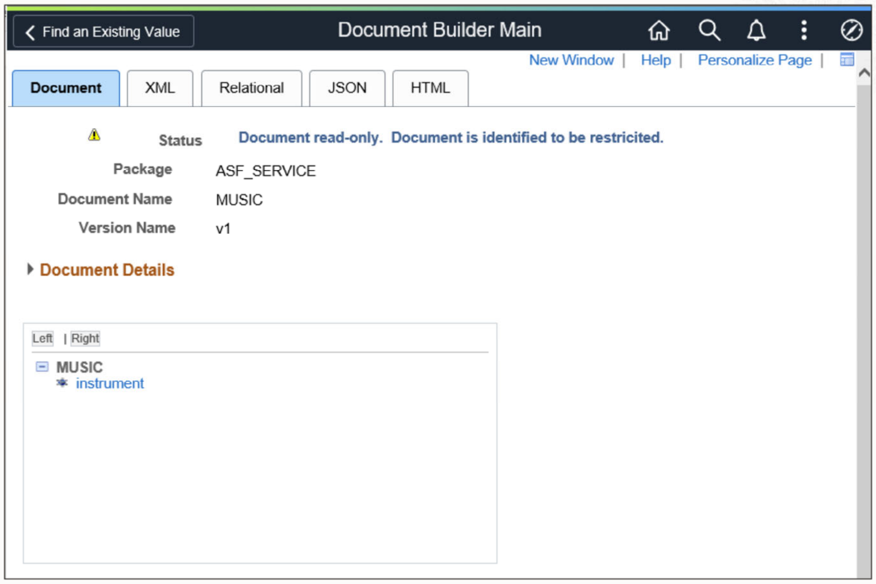
then Append to URI.

To form the URI, click the Add to workspace push button. This will display the parameter in brackets in the Expression workspace section.

You can then click Append to URI to add the variable to the URI template.

Select done when the URI template is complete.

Restricted Document



Status: Document read-only. Document is identified to be restricited.

Package: ASF_SERVICE

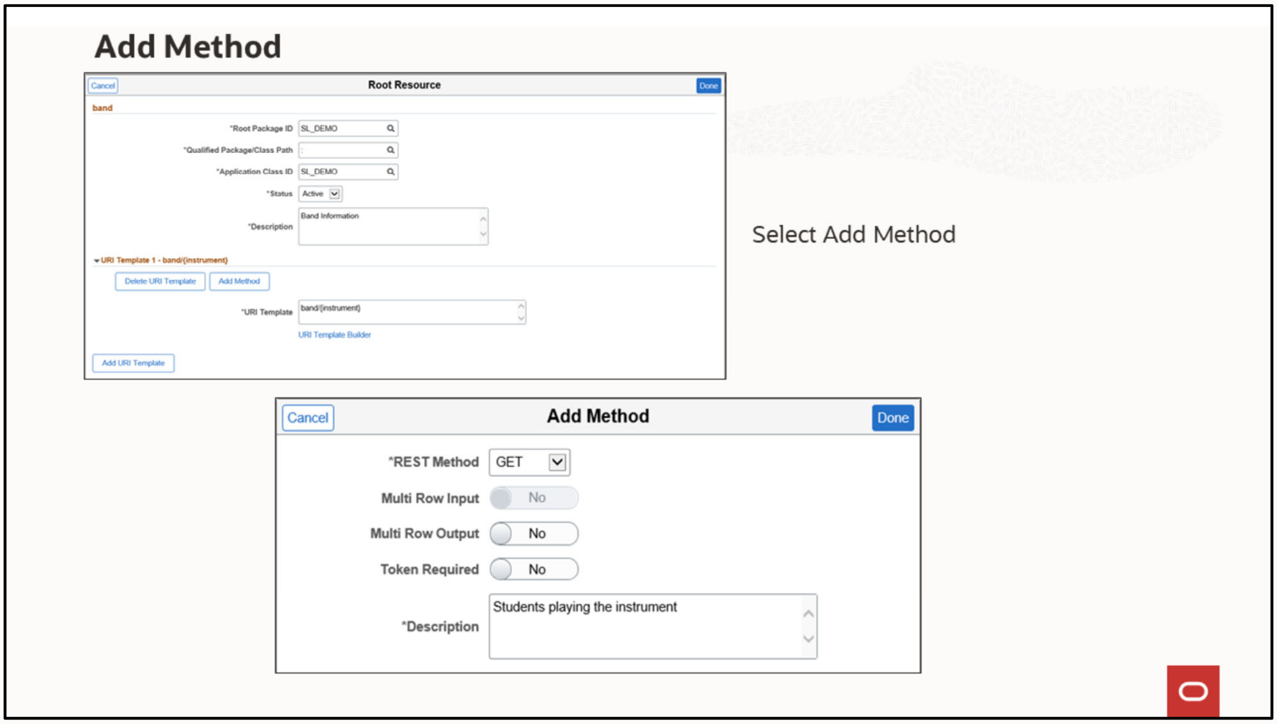
Document Name: MUSIC

Version Name: v1

MUSIC
- instrument

When the Application Service is saved, a restricted document is created defining the primitives used for value replacement within the URI template.

The document is created in the ASF_SERVICE document package with the App service ID.



Add Method

band

*Root Package ID SL_DEMO

*Qualified Package/Class Path :

*Application Class ID SL_DEMO

*Status Active

*Description Band Information

URI Template 1 - band/{instrument}

*URI Template band/{instrument}

*REST Method GET

Multi Row Input No

Multi Row Output No

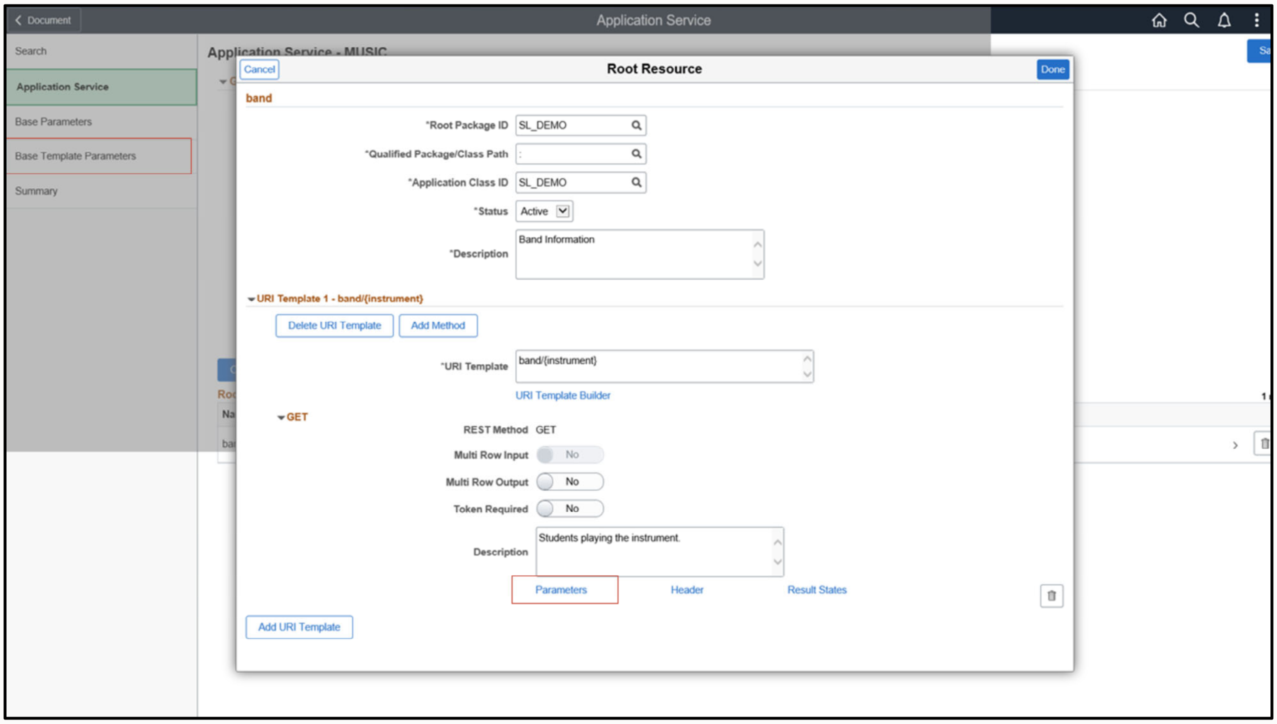
Token Required No

*Description Students playing the instrument

Select Add Method

Next we will add a method to the URI Template.
Click the Add Method button.

Select the REST method from the drop-down list.
For this demo, we will use GET. We want to get a list of all students for an instrument.
Indicate if there are multiple rows of input or output. Since this is a GET, multi row input is grayed out.
Indicate if a token is required. The default token type is oAuth2, the type can be changed when you set up security.
For this demo, a token is not required.
Enter the description, then click Done.



*Root Package ID: SL_DEMO

*Qualified Package/Class Path: :

*Application Class ID: SL_DEMO

*Status: Active

*Description: Band Information

*URI Template: band/{instrument}

REST Method: GET

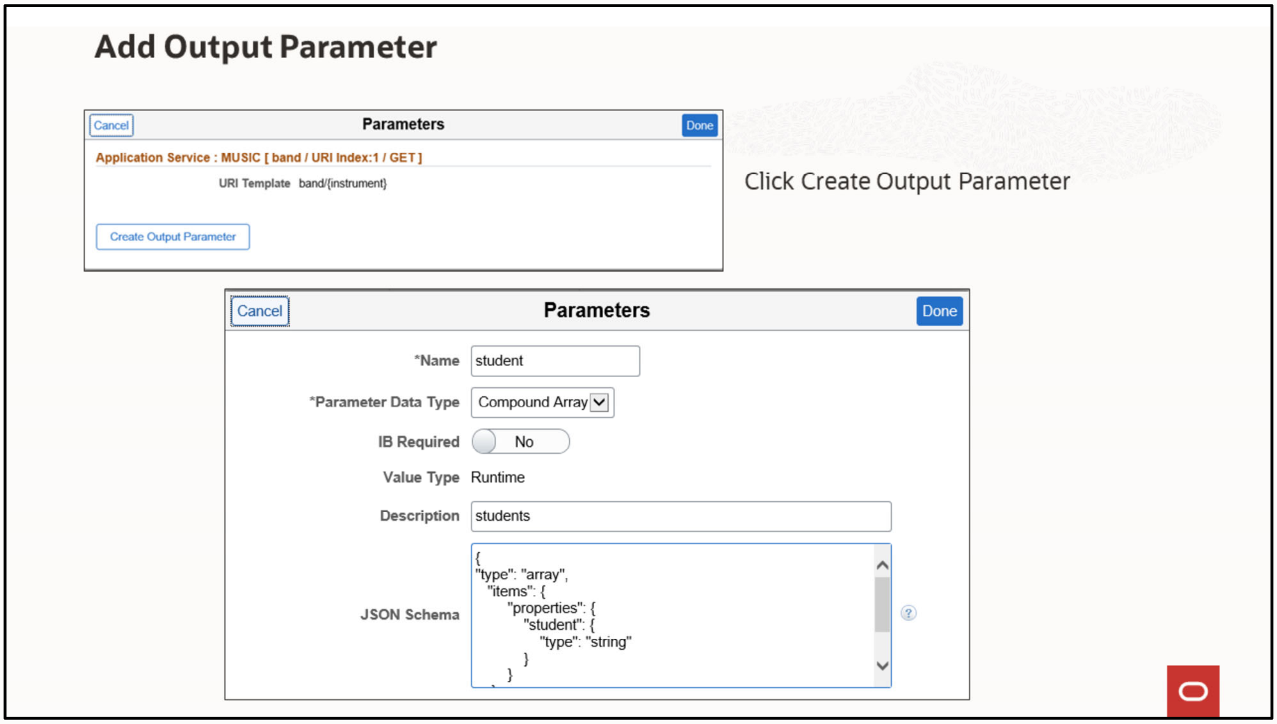
Multi Row Input: No

Multi Row Output: No

Token Required: No

Description: Students playing the instrument.

Click the Parameters link to add an output parameter.
Output parameters can be added directly from the parameters page for the application method or by selecting the Base Parameters menu item for the Application Service.
For REST methods other than GET, you can add input and output parameters.



Add Output Parameter

Application Service : MUSIC [band / URI Index:1 / GET]

URI Template band/{instrument}

Click Create Output Parameter

*Name	student
*Parameter Data Type	Compound Array
IB Required	No
Value Type	Runtime
Description	students

JSON Schema

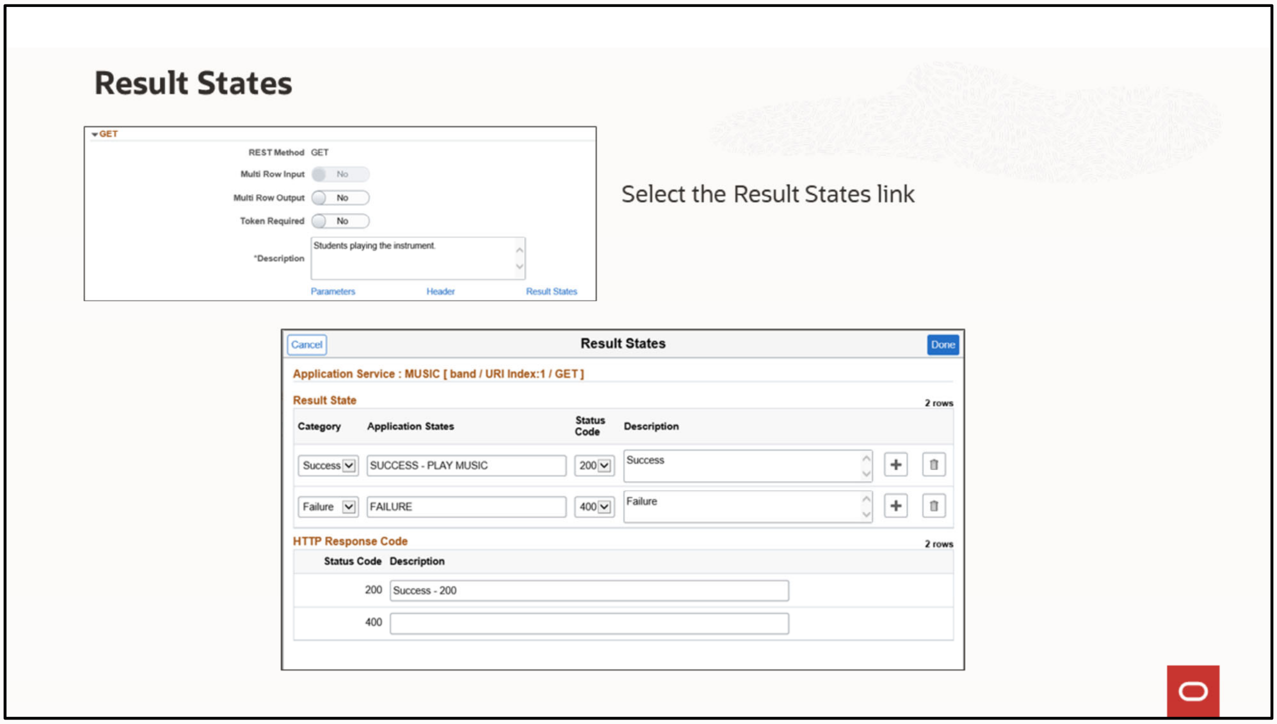
```
{
"type": "array",
  "items": {
     "properties": {
        "student": {
           "type": "string"
        }
     }
```

We will create an output parameter for student.
Click Create Output Parameter.

The parameter data type will be Compound Array as we want to return multiple students.
Indicate if IB is required and enter a description.
Since this is a compound array, we must supply the JSON schema for the parameter.

Click Done.
The output parameter was added.

Click Done again on the Parameters page.



Result States

GET

REST Method: GET

Multi Row Input: No

Multi Row Output: No

Token Required: No

*Description: Students playing the instrument.

Select the Result States link

Result States

Application Service : MUSIC [band / URI Index:1 / GET]

Result State — 2 rows

Category	Application States	Status Code	Description
Success	SUCCESS - PLAY MUSIC	200	Success
Failure	FAILURE	400	Failure

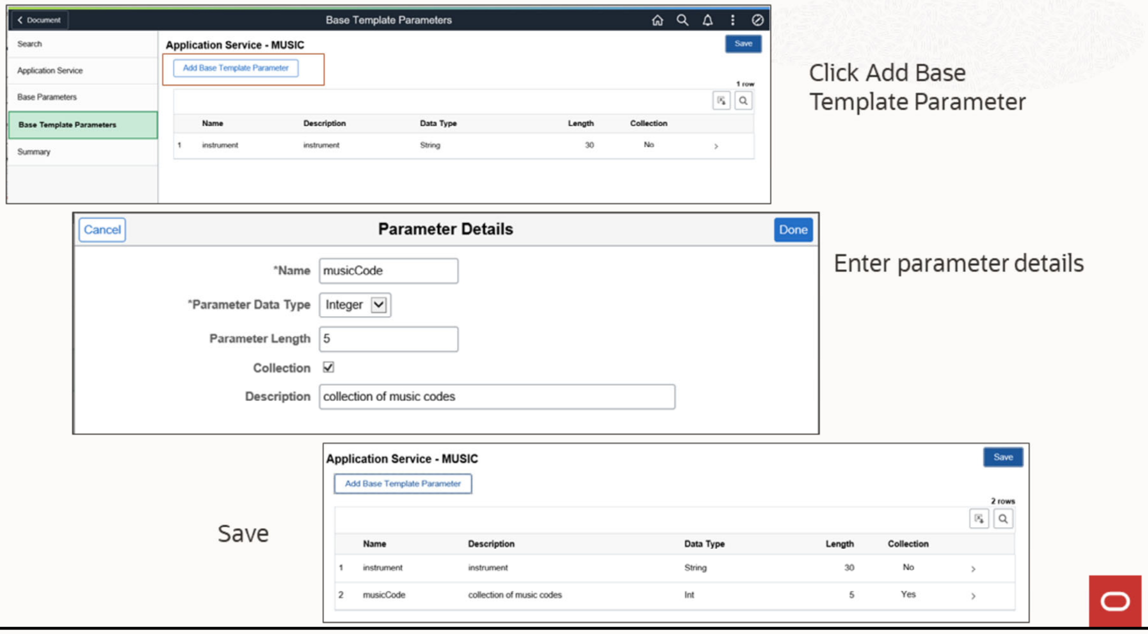
HTTP Response Code — 2 rows

Status Code	Description
200	Success - 200
400	

Select the Result States link to review the result states.

By default a single Result State will be created indicating success with a Status Code of 200. You can modify the status and add additional result states.
We will change the application state for success and add a category for failure.
We will use failure code 400.
After updating the result states, click Done.

Base Template Parameters



Application Service - MUSIC

1 row

	Name	Description	Data Type	Length	Collection
1	instrument	instrument	String	30	No

Click Add Base Template Parameter

Parameter Details

*Name	musicCode
*Parameter Data Type	Integer
Parameter Length	5
Collection	☑
Description	collection of music codes

Enter parameter details

Save

Application Service - MUSIC

2 rows

	Name	Description	Data Type	Length	Collection
1	instrument	instrument	String	30	No
2	musicCode	collection of music codes	Int	5	Yes

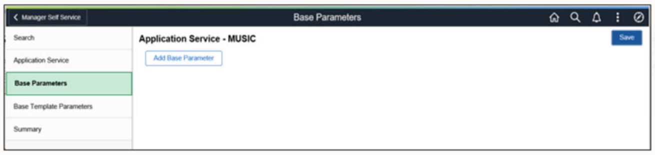
Base template parameters are also available from the left-panel.
Select this link to view the base template parameters and add additional parameters.
We will add a parameter for a collection of music codes.

Click Add Base Template Parameter.

Use camel case for the name.
The parameter type is Integer with a length of 5.
This will be a collection of music codes.
Enter the description and click Done.

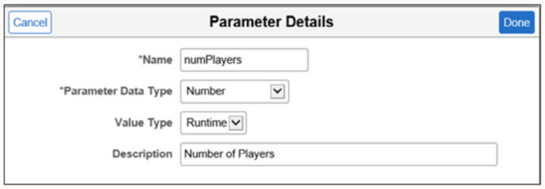
Save the Application Service and click OK.

Base Parameters



Application Service - MUSIC

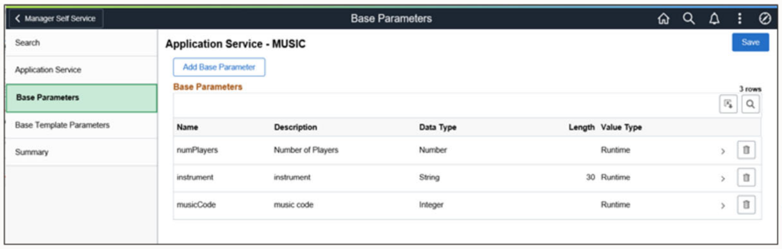
Select Add Base Parameter.



*Name	numPlayers
*Parameter Data Type	Number
Value Type	Runtime
Description	Number of Players

Enter parameter details.

List if base parameters.



Application Service - MUSIC

Base Parameters

3 rows

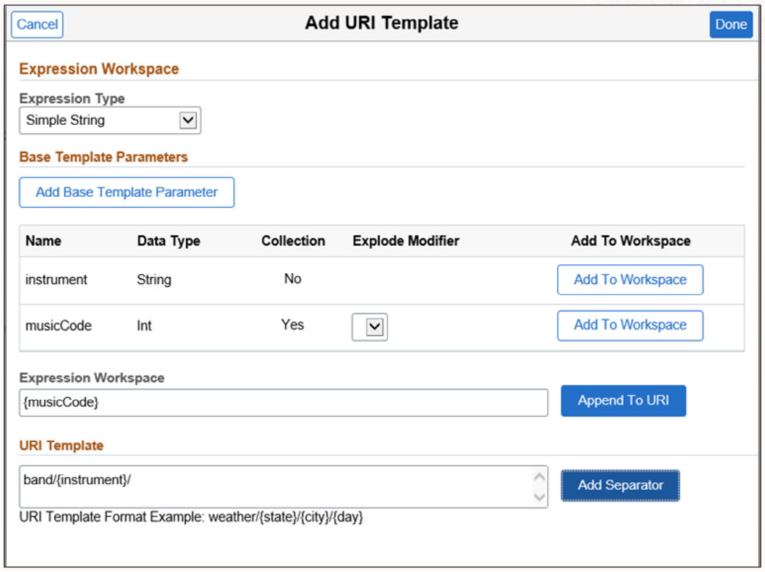
Name	Description	Data Type	Length	Value Type
numPlayers	Number of Players	Number		Runtime
instrument	instrument	String	30	Runtime
musicCode	music code	Integer		Runtime

Use the Base Parameters page to define a set of parameters that can be reused for any root resource as input or output parameters.
We will create three base parameters to use as output.
Select Add Base Parameter.

First we will create a parameter for number of players – the data type is number and we will enter the description, then click Done.
Next we will add a parameter for instrument. The data type is string with a length of 30. Enter the information and click Done.
We also need an output parameter for music code, which will be an integer. Enter the information and click Done.

The base parameters are now available to use as input or output for this Application Service.
Save the Application Service and then OK on the message that the save was successful.

Adding Another URI Template



Add URI Template

Expression Workspace

Expression Type: Simple String

Base Template Parameters

Name	Data Type	Collection	Explode Modifier	Add To Workspace
instrument	String	No		Add To Workspace
musicCode	Int	Yes		Add To Workspace

Expression Workspace: {musicCode}

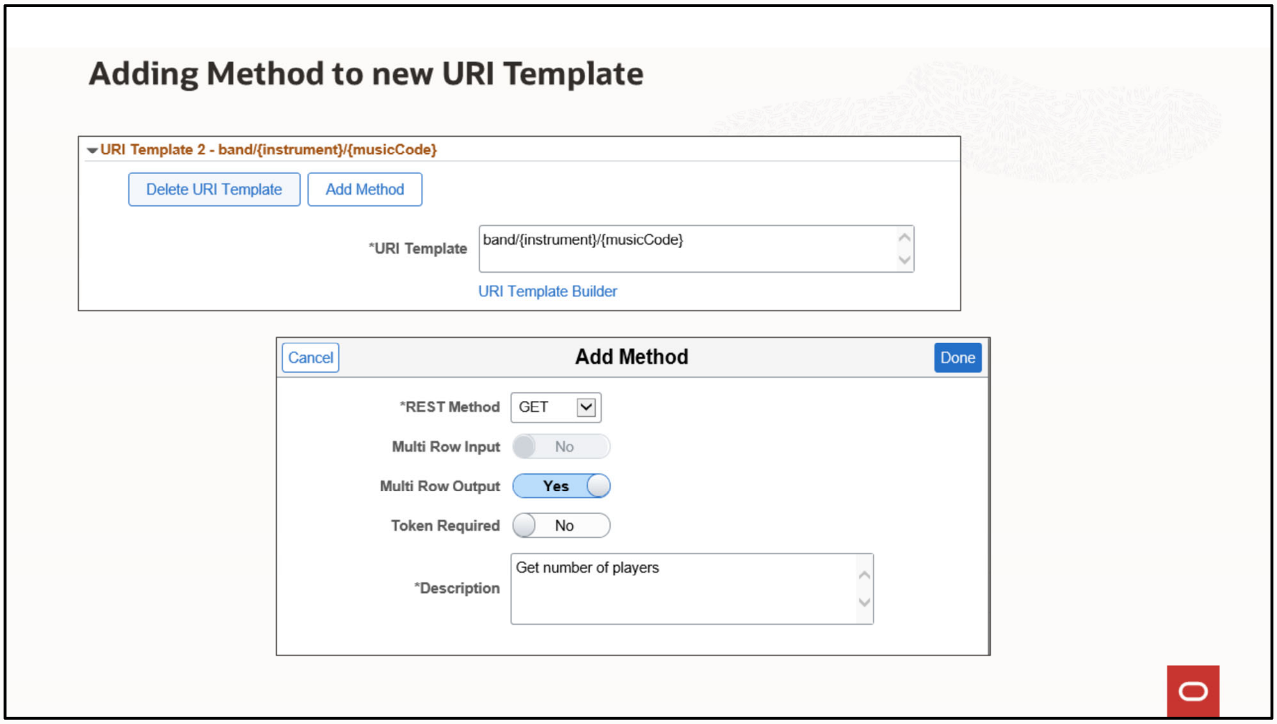
URI Template

band/{instrument}/

URI Template Format Example: weather/{state}/{city}/{day}

Now we will add another URI template.
Click on the root resource ***band***.
Select Add URI Template.

The existing base template parameters for this Application Service are listed.
For this template, we will add instrument and musicCode.
Click Add to Workspace for instrument, then Append to URI.
Add a Separator.
Add musicCode to the Workspace, then Append to URI.
We can see the URI template, click Done.



Adding Method to new URI Template

URI Template 2 - band/{instrument}/{musicCode}

*URI Template: band/{instrument}/{musicCode}

*REST Method: GET

Multi Row Input: No

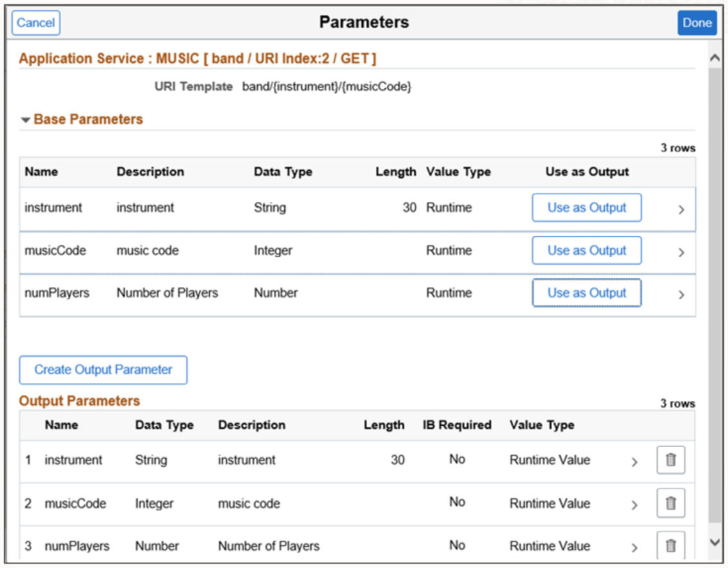
Multi Row Output: Yes

Token Required: No

*Description: Get number of players

We will add a GET method.
Click Add Method.
This method will have multi row output to return the number of players for an instrument and code.
We will not use a token.
Enter a description and click Done.

Adding Output Parameters



Application Service : MUSIC [band / URI Index:2 / GET]

URI Template band/{instrument}/{musicCode}

Base Parameters

3 rows

Name	Description	Data Type	Length	Value Type	Use as Output
instrument	instrument	String	30	Runtime	Use as Output
musicCode	music code	Integer		Runtime	Use as Output
numPlayers	Number of Players	Number		Runtime	Use as Output

Output Parameters

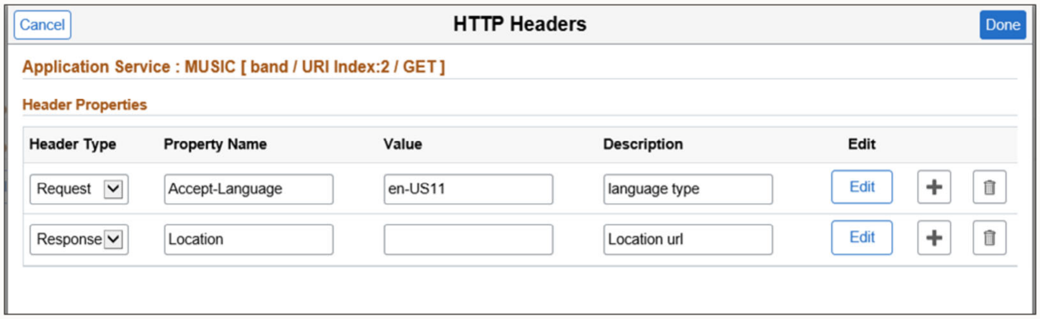
3 rows

	Name	Data Type	Description	Length	IB Required	Value Type
1	instrument	String	instrument	30	No	Runtime Value
2	musicCode	Integer	music code		No	Runtime Value
3	numPlayers	Number	Number of Players		No	Runtime Value

Now we will add the output parameters.
Select the Parameters link.
Expand Base parameters, note that the base parameters we added are now available.
Select Use as Output for each of these.

We will have 3 output parameters.
Click Done.

Adding Headers



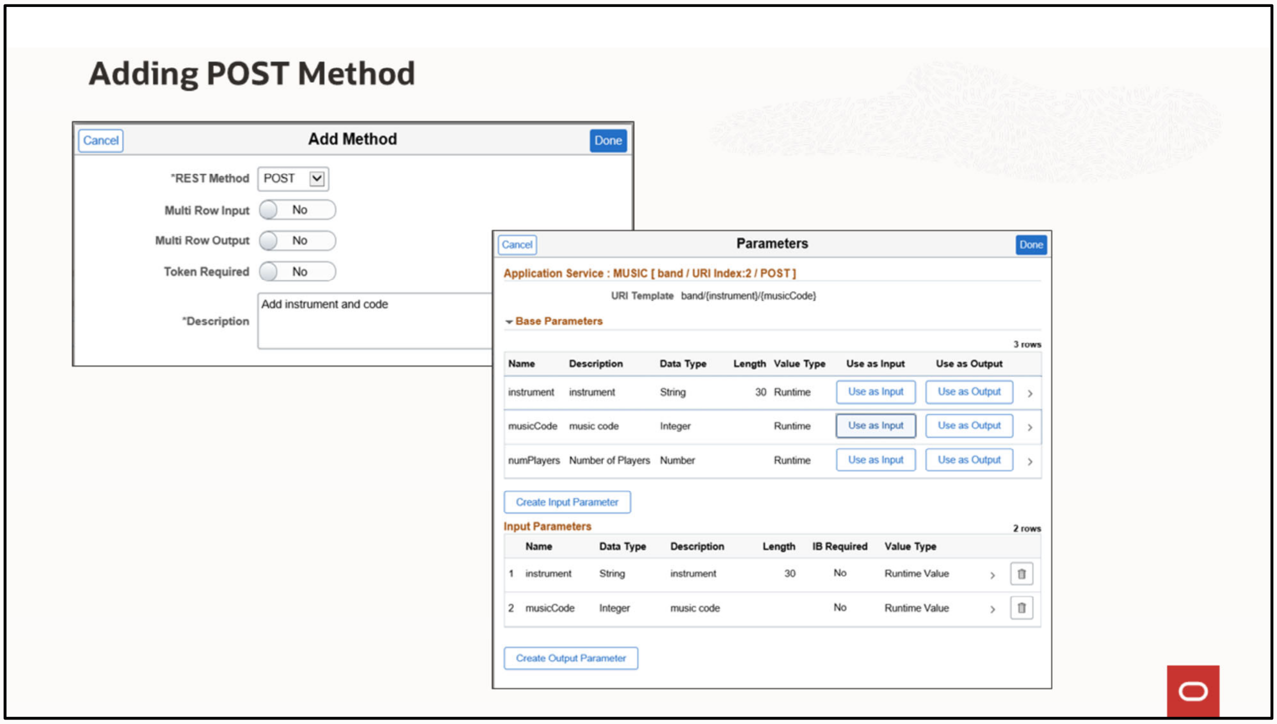
HTTP Headers

Application Service : MUSIC [band / URI Index:2 / GET]

Header Properties

Header Type	Property Name	Value	Description	Edit
Request	Accept-Language	en-US11	language type	Edit
Response	Location		Location url	Edit

We will also add a request and response header for this method.
Select the Header link.
We will add a request header for language.
Enter the property name, value and description.
Select the Add row icon to add the response header for location.
The header type is response.
Enter the property name and description.
Click Done.
Click Done again on the HTTP Headers page.



Adding POST Method

*REST Method: POST

Multi Row Input: No

Multi Row Output: No

Token Required: No

*Description: Add instrument and code

Application Service : MUSIC [band / URI Index:2 / POST]

URI Template band/{instrument}/{musicCode}

Base Parameters

3 rows

Name	Description	Data Type	Length	Value Type	Use as Input	Use as Output
instrument	instrument	String	30	Runtime	Use as Input	Use as Output
musicCode	music code	Integer		Runtime	Use as Input	Use as Output
numPlayers	Number of Players	Number		Runtime	Use as Input	Use as Output

Input Parameters

2 rows

	Name	Data Type	Description	Length	IB Required	Value Type
1	instrument	String	instrument	30	No	Runtime Value
2	musicCode	Integer	music code		No	Runtime Value

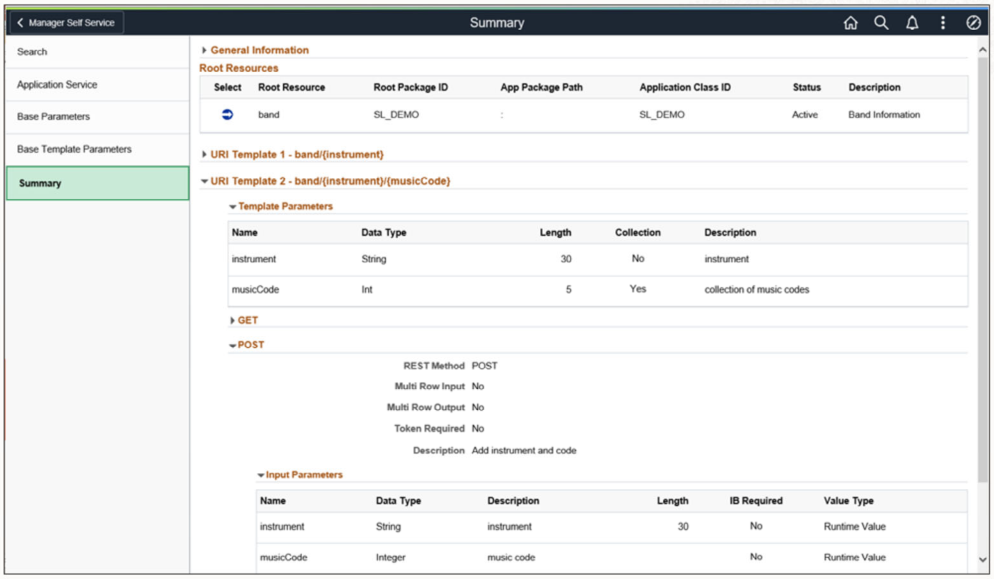
We will add one more method for POST.
Click Add Method.

The REST Method is POST.
This method does not have multi row input or output and does not require a token.
Add the description and click Done.

Select the Parameters link to add the input parameters.
Expand the Base Parameters.
Select the Use as Input Button for instrument and musicCode.
Click Done.
Click Done again on the Root Resource page.

Save the Application Service.
Click OK on the message that the save was successful.

Summary



General Information

Root Resources

Select	Root Resource	Root Package ID	App Package Path	Application Class ID	Status	Description
➲	band	SL_DEMO	:	SL_DEMO	Active	Band Information

URI Template 1 - band/{instrument}

URI Template 2 - band/{instrument}/{musicCode}

Template Parameters

Name	Data Type	Length	Collection	Description
instrument	String	30	No	instrument
musicCode	Int	5	Yes	collection of music codes

GET

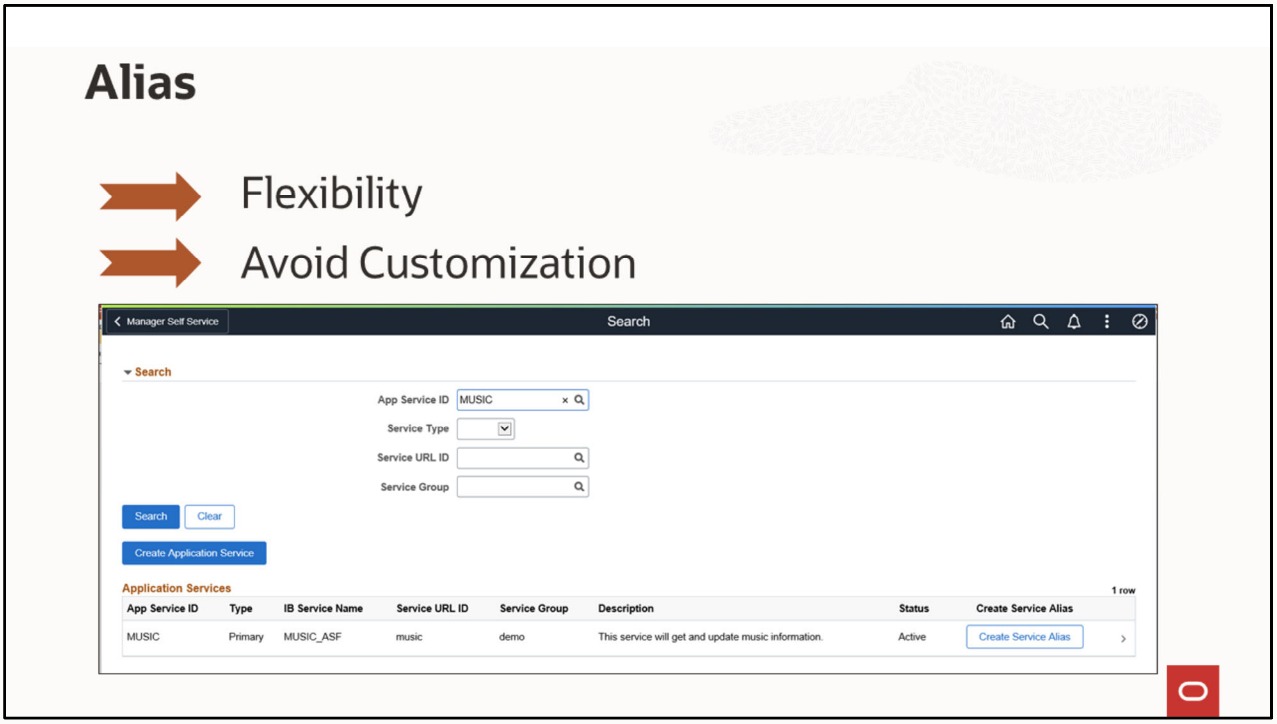
POST

REST Method POST
Multi Row Input No
Multi Row Output No
Token Required No
Description Add instrument and code

Input Parameters

Name	Data Type	Description	Length	IB Required	Value Type
instrument	String	instrument	30	No	Runtime Value
musicCode	Integer	music code		No	Runtime Value

Once you have entered the URI Templates, you can use the Summary link to review the Application Service.
This page displays all the metadata necessary to code the application class for the Application Service.



Alias

- Flexibility
- Avoid Customization

App Service ID: MUSIC
Service Type:
Service URL ID:
Service Group:

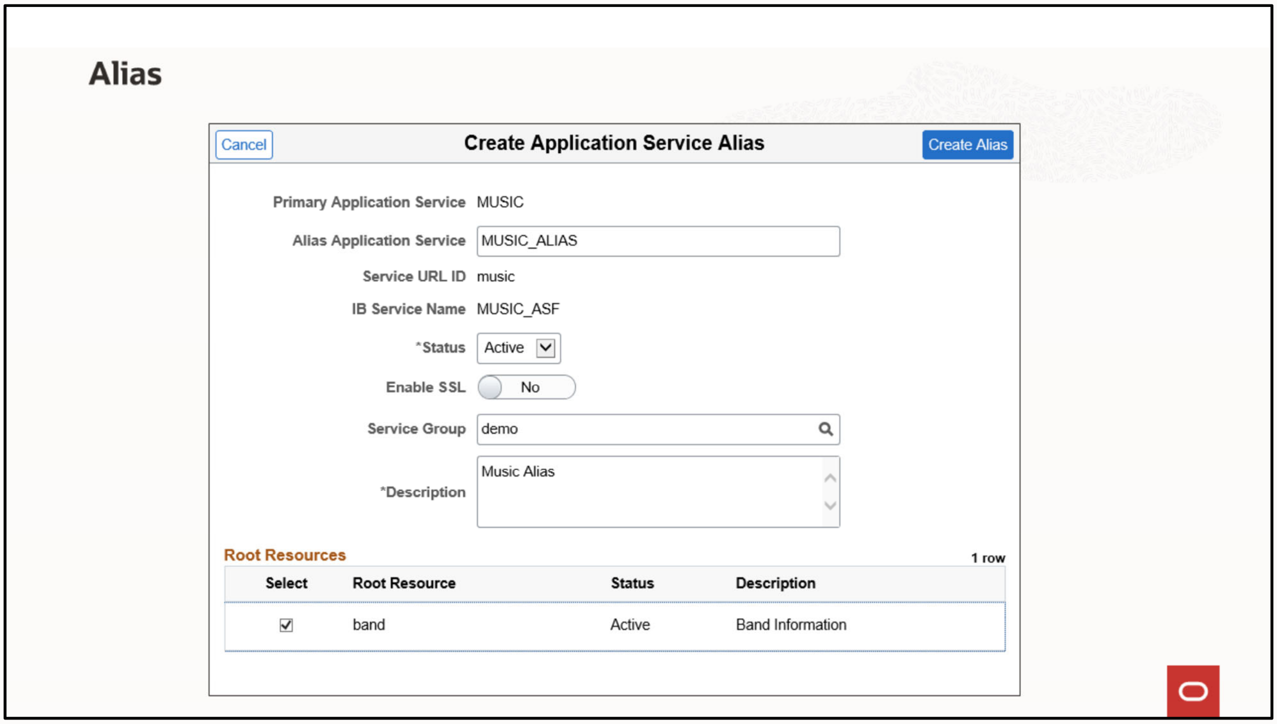
Application Services 1 row

App Service ID	Type	IB Service Name	Service URL ID	Service Group	Description	Status	Create Service Alias
MUSIC	Primary	MUSIC_ASF	music	demo	This service will get and update music information.	Active	Create Service Alias

The ability to create a Service Alias provides customers with the flexibility to change the application logic in a registered Application Service, thereby avoiding customization.

To create a service alias, search for the Application Service.
We will select Music.

Click the Create Service Alias button.



Alias

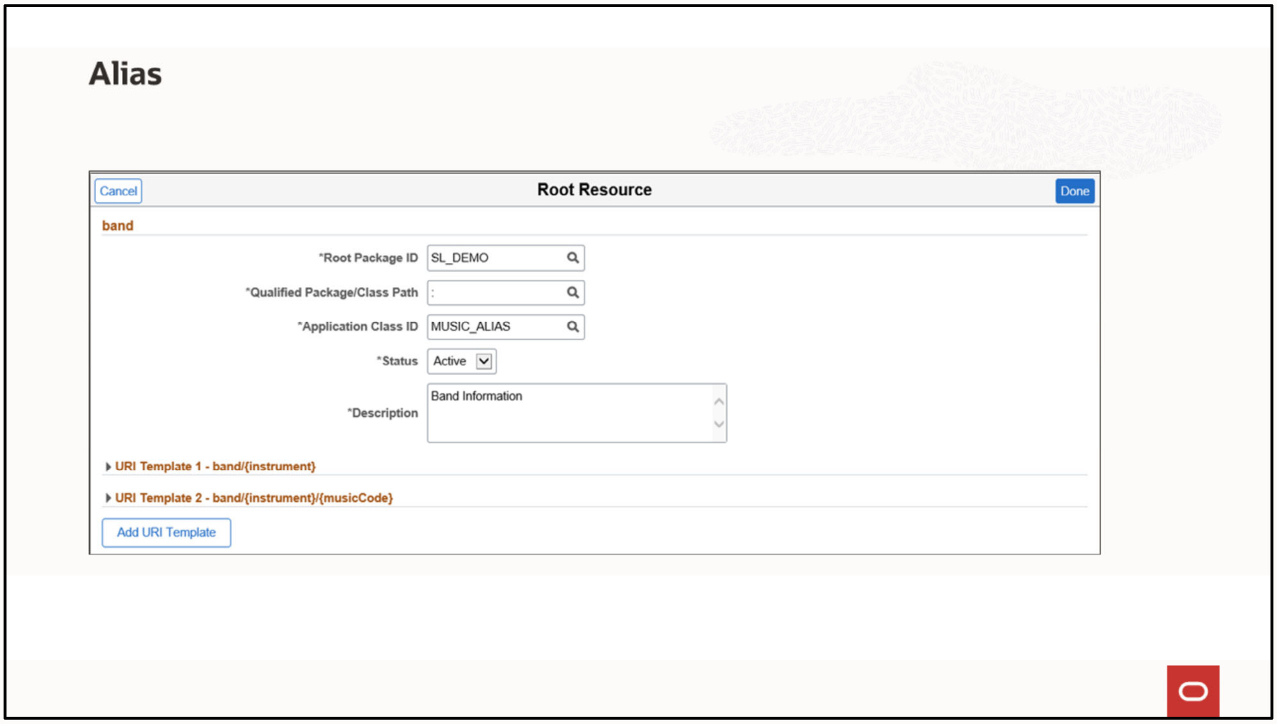
Primary Application Service	MUSIC
Alias Application Service	MUSIC_ALIAS
Service URL ID	music
IB Service Name	MUSIC_ASF
*Status	Active
Enable SSL	No
Service Group	demo
*Description	Music Alias

Root Resources — 1 row

Select	Root Resource	Status	Description
☑	band	Active	Band Information

Enter an Alias name.
We will add this to the service group demo and add a description.
Select the root resource or resources to alias.
Click Create Alias.
Click OK on the message that the Alias was created.

The Service URL ID remains the same for the Alias and the Primary Application Service from which it was created. The application logic is executed based on the Application Service that is active.



Alias

band

*Root Package ID	SL_DEMO
*Qualified Package/Class Path	:
*Application Class ID	MUSIC_ALIAS
*Status	Active
*Description	Band Information

URI Template 1 - band/{instrument}

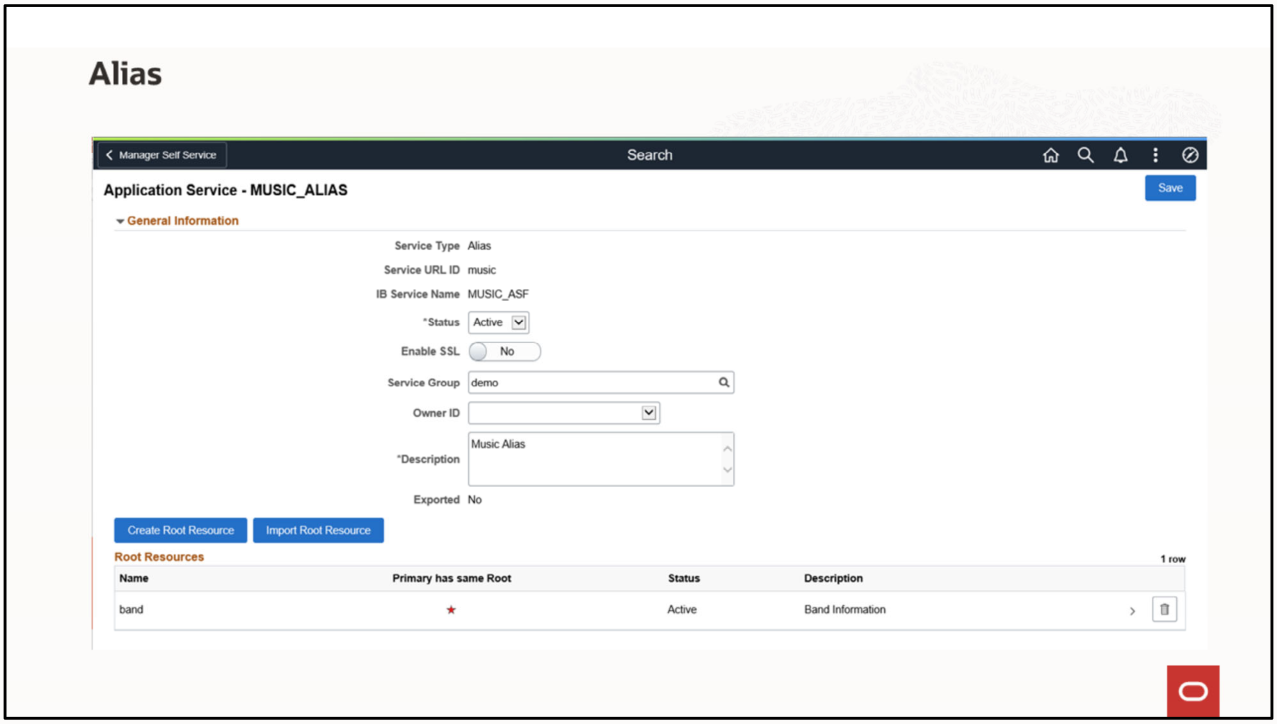
URI Template 2 - band/{instrument}/{musicCode}

A new application class must be created for the alias which contains the application logic for the Application Service. If the primary Application Service is modified in a delivered update, it will not affect the alias application service.

To add the application logic for the Alias, select the root resource, in this case band.

Enter the root package id, class path and application class for the alias.

Click Done.



Alias

Application Service - MUSIC_ALIAS

General Information

Service Type	Alias
Service URL ID	music
IB Service Name	MUSIC_ASF
*Status	Active
Enable SSL	No
Service Group	demo
Owner ID	
*Description	Music Alias
Exported	No

Root Resources

1 row

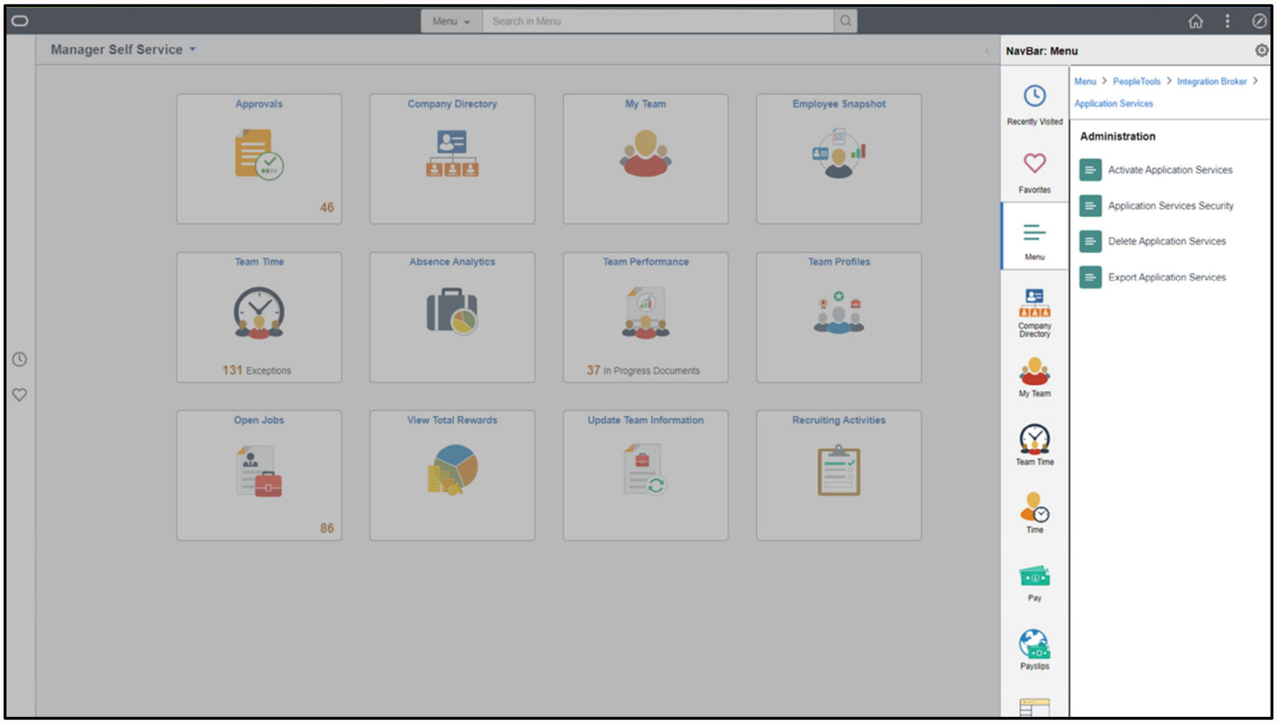
Name	Primary has same Root	Status	Description
band	*	Active	Band Information

Make any other modification necessary for the alias, such as adding additional root resource or changing parameters. For this demo, we will save the Application Service without any further changes.

Save the Application Service and click OK on the message.

Administering Application Service

Next we will look at administering Application Services.

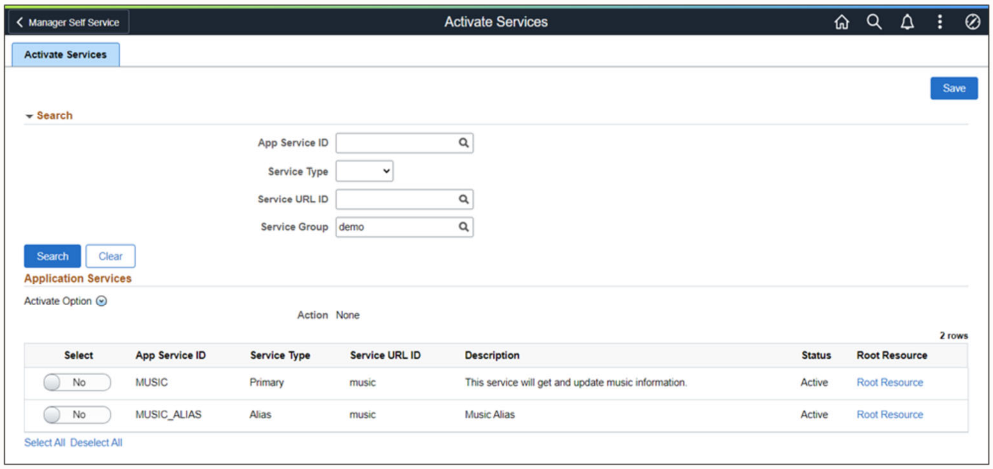


From the menu select PeopleTools, Integration Broker, Application Services, Administration.

Administration includes activating a service, setting security, deleting a service and exporting a service. We will look at each of these.

Activate Services

Menu > PeopleTools> Integration Broker> Application Services> Administration> Activate Application Services



App Service ID

Service Type

Service URL ID

Service Group: demo

Application Services

Activate Option

Action None

2 rows

Select	App Service ID	Service Type	Service URL ID	Description	Status	Root Resource
No	MUSIC	Primary	music	This service will get and update music information.	Active	Root Resource
No	MUSIC_ALIAS	Alias	music	Music Alias	Active	Root Resource

Select All Deselect All

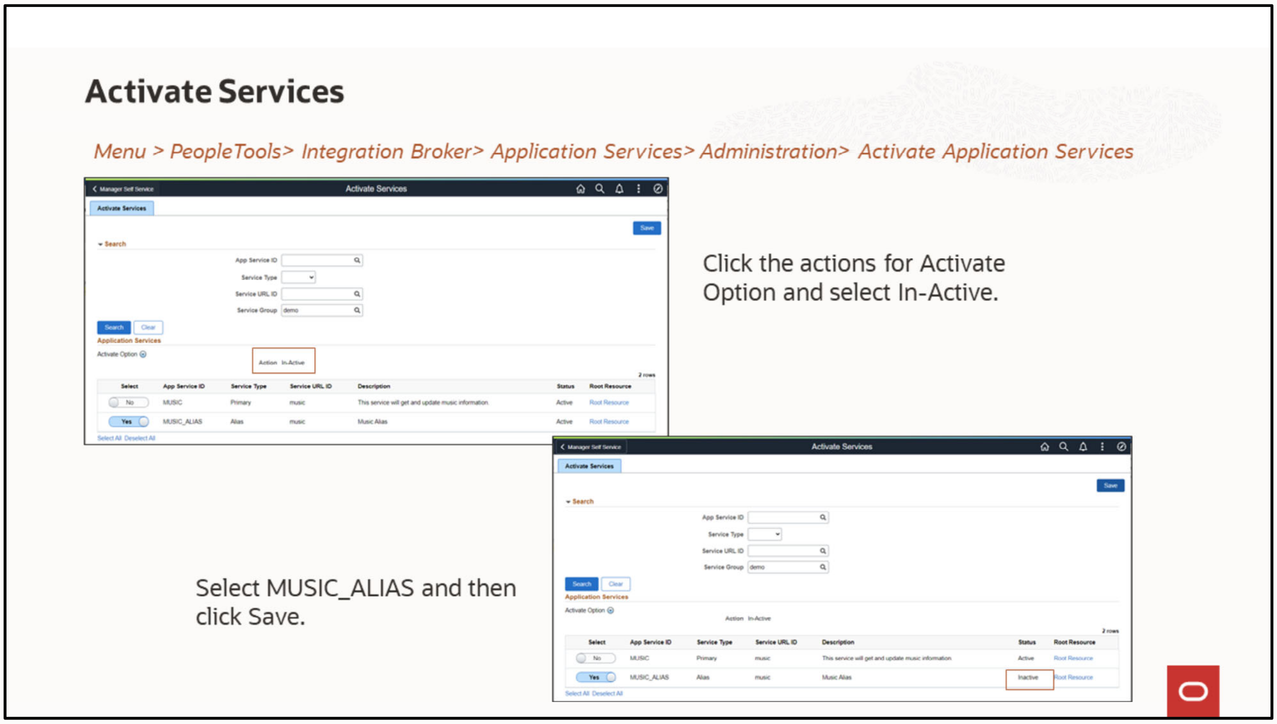
To activate or inactivate a service, select Activate Application Services.

Use Search to find the Application Service, we will search by the service group. Click Search.

An Alias Root Resource will always have priority over its associated Primary if both are Active.

If the Alias is not active, the primary Application Service is used.

If the Alias Root Resource is not active, the primary Application Service root resource is used.



Activate Services

Menu > PeopleTools> Integration Broker> Application Services> Administration> Activate Application Services

App Service ID:
Service Type:
Service URL ID:
Service Group: demo

Activate Option

Action In-Active

2 rows

Select	App Service ID	Service Type	Service URL ID	Description	Status	Root Resource
No	MUSIC	Primary	music	This service will get and update music information.	Active	Root Resource
Yes	MUSIC_ALIAS	Alias	music	Music Alias	Active	Root Resource

Click the actions for Activate Option and select In-Active.

Select MUSIC_ALIAS and then click Save.

App Service ID:
Service Type:
Service URL ID:
Service Group: demo

Activate Option

Action In-Active

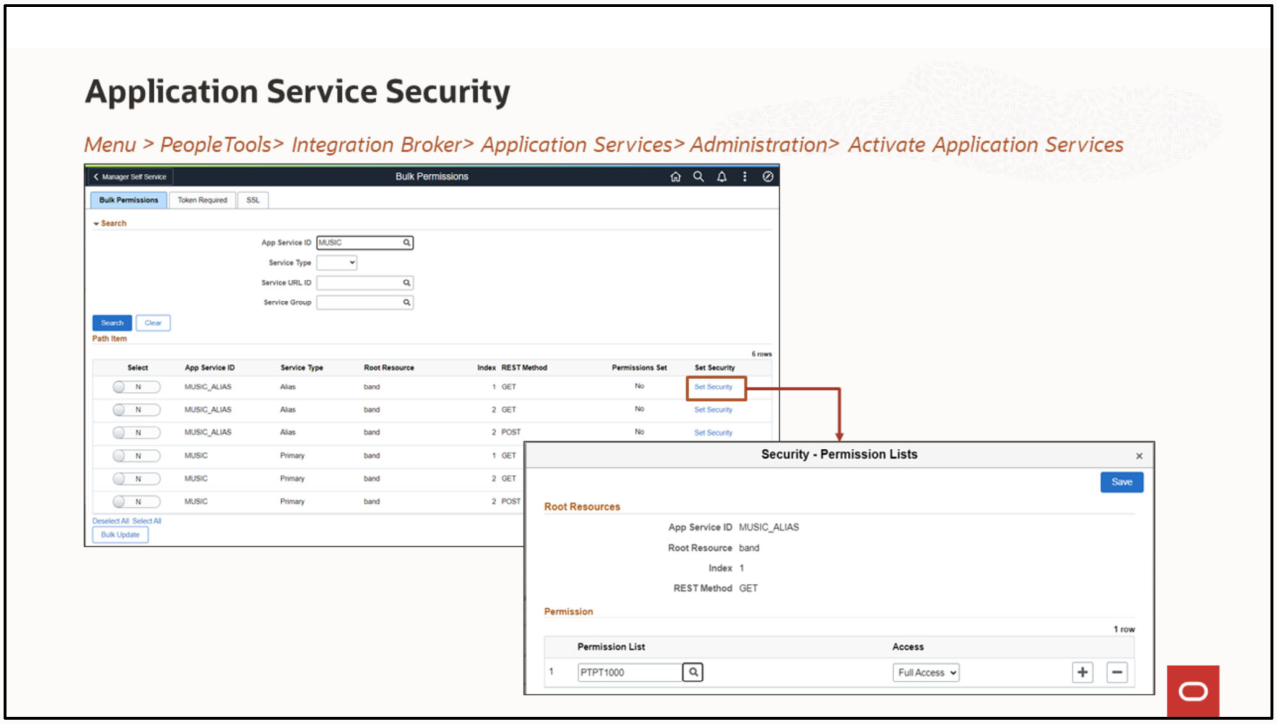
2 rows

Select	App Service ID	Service Type	Service URL ID	Description	Status	Root Resource
No	MUSIC	Primary	music	This service will get and update music information.	Active	Root Resource
Yes	MUSIC_ALIAS	Alias	music	Music Alias	Inactive	Root Resource

For this demo we will inactivate the Alias.

Click the actions for Activate Option and select In-Active.
Notice that the Action now shows In-Active.
Select MUSIC_ALIAS and then click Save.
Click OK on the message that the save was successful.

We now see that MUSIC_ALIAS is inactive.



Application Service Security

Menu > PeopleTools> Integration Broker> Application Services> Administration> Activate Application Services

App Service ID: MUSIC
Service Type:
Service URL ID:
Service Group:

6 rows

Select	App Service ID	Service Type	Root Resource	Index	REST Method	Permissions Set	Set Security
N	MUSIC_ALIAS	Alias	band	1	GET	No	Set Security
N	MUSIC_ALIAS	Alias	band	2	GET	No	Set Security
N	MUSIC_ALIAS	Alias	band	2	POST	No	Set Security
N	MUSIC	Primary	band	1	GET		
N	MUSIC	Primary	band	2	GET		
N	MUSIC	Primary	band	2	POST		

Security - Permission Lists

Root Resources

App Service ID: MUSIC_ALIAS
Root Resource: band
Index: 1
REST Method: GET

Permission

1 row

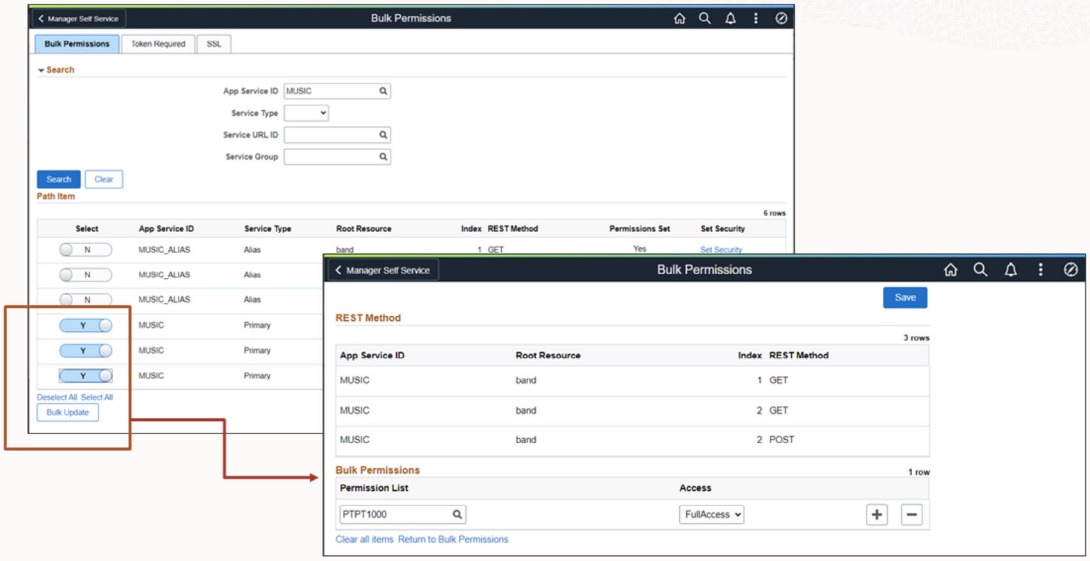
	Permission List	Access
1	PTPT1000	Full Access

Next we will look at Application Service Security.
Return to the menu and select Application Services Security.
Use Search to select the Application Service. We will search by service group.

Use the Set Security link to set security for a specific root resource, index and method.

Enter the permission list and access, then Save.
Click OK.
We can see that the security is only set for that row.

Application Service Security - Bulk Update



App Service ID: MUSIC
Service Type:
Service URL ID:
Service Group:

Path Item

6 rows

Select	App Service ID	Service Type	Root Resource	Index	REST Method	Permissions Set	Set Security
N	MUSIC_ALIAS	Alias	band	1	GET	Yes	Set Security
N	MUSIC_ALIAS	Alias					
N	MUSIC_ALIAS	Alias					
Y	MUSIC	Primary					
Y	MUSIC	Primary					
Y	MUSIC	Primary					

REST Method

3 rows

App Service ID	Root Resource	Index	REST Method
MUSIC	band	1	GET
MUSIC	band	2	GET
MUSIC	band	2	POST

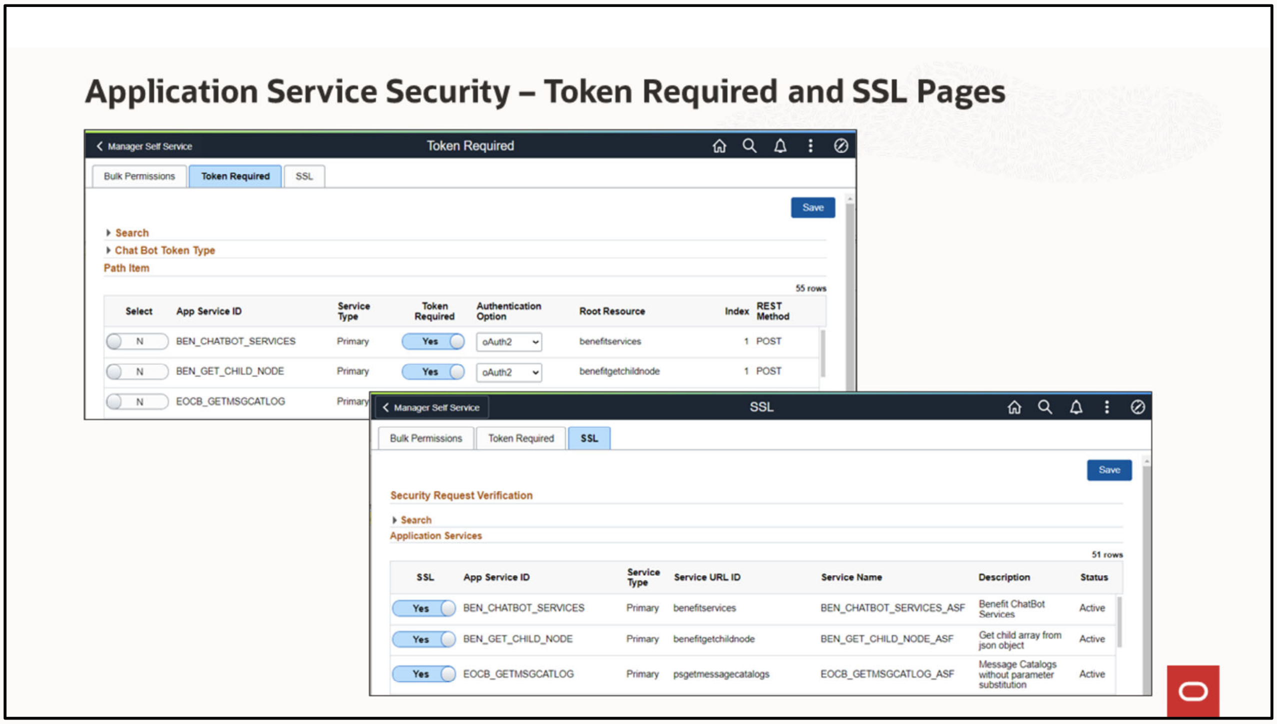
Bulk Permissions

1 row

Permission List	Access
PTPT1000	FullAccess

Use the Bulk Update feature to apply security for multiple root resources and rest methods.
You can use the Select All button or individually select the Application services.
We will select the MUSIC Application Services, then click Bulk Update.

Select the Permission list or lists and access level, then save.
Click OK.
Then click the Return to Bulk Permissions link.



Application Service Security – Token Required and SSL Pages

▸ Search
▸ Chat Bot Token Type
Path Item

55 rows

Select	App Service ID	Service Type	Token Required	Authentication Option	Root Resource	Index	REST Method
N	BEN_CHATBOT_SERVICES	Primary	Yes	oAuth2	benefitservices	1	POST
N	BEN_GET_CHILD_NODE	Primary	Yes	oAuth2	benefitgetchildnode	1	POST
N	EOCB_GETMSGCATLOG	Primary					

Security Request Verification
▸ Search
Application Services

51 rows

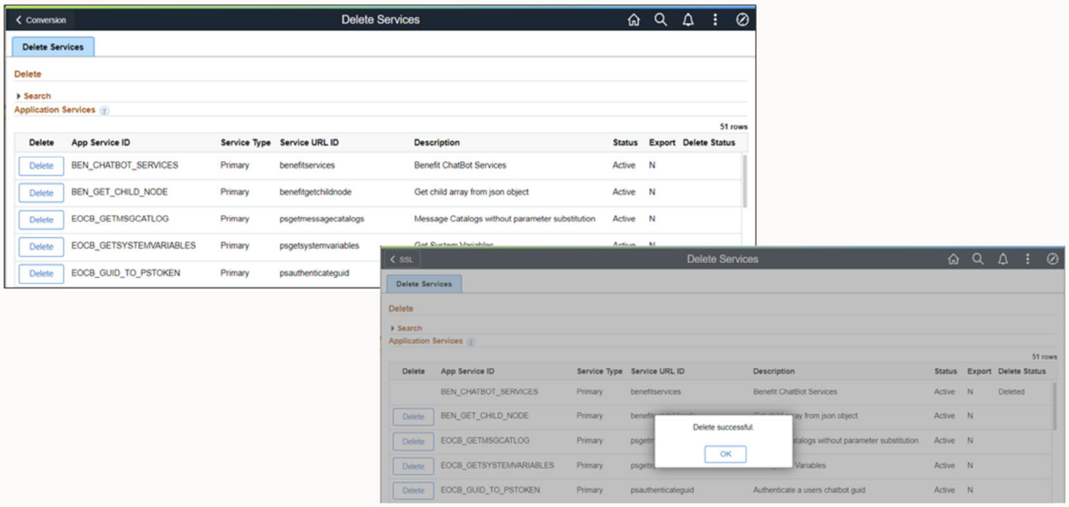
SSL	App Service ID	Service Type	Service URL ID	Service Name	Description	Status
Yes	BEN_CHATBOT_SERVICES	Primary	benefitservices	BEN_CHATBOT_SERVICES_ASF	Benefit ChatBot Services	Active
Yes	BEN_GET_CHILD_NODE	Primary	benefitgetchildnode	BEN_GET_CHILD_NODE_ASF	Get child array from json object	Active
Yes	EOCB_GETMSGCATLOG	Primary	psgetmessagecatalogs	EOCB_GETMSGCATLOG_ASF	Message Catalogs without parameter substitution	Active

Use the Token Required tab to set the authentication option where token is required. The default is oAuth2. You can change this to Basic Auth.

Use the SSL tab to set SSL to indicate that SSL is required for the Application Service.

Delete Application Service

Menu > PeopleTools> Integration Broker> Application Services> Administration> Delete Application Services



Delete

Search

Application Services

51 rows

Delete	App Service ID	Service Type	Service URL ID	Description	Status	Export	Delete Status
Delete	BEN_CHATBOT_SERVICES	Primary	benefitservices	Benefit ChatBot Services	Active	N	
Delete	BEN_GET_CHILD_NODE	Primary	benefitgetchildnode	Get child array from json object	Active	N	
Delete	EOCB_GETMSGCATLOG	Primary	psgetmessagecatalogs	Message Catalogs without parameter substitution	Active	N	
Delete	EOCB_GETSYSTEMVARIABLES	Primary	psgetsystemvariables	Get System Variables	Active	N	
Delete	EOCB_GUID_TO_PSTOKEN	Primary	psauthenticateguid				

Delete

Search

Application Services

51 rows

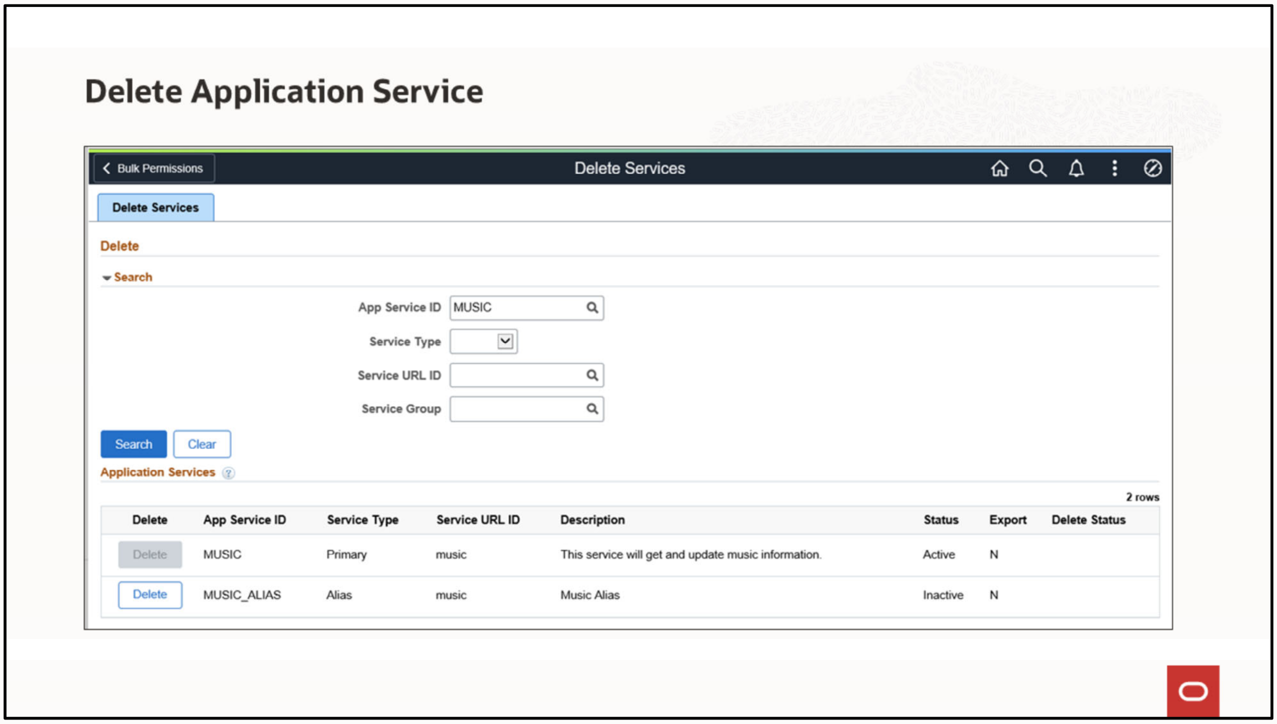
Delete	App Service ID	Service Type	Service URL ID	Description	Status	Export	Delete Status
	BEN_CHATBOT_SERVICES	Primary	benefitservices	Benefit ChatBot Services	Active	N	Deleted
Delete	BEN_GET_CHILD_NODE	Primary	benefi	ay from json object	Active	N	
Delete	EOCB_GETMSGCATLOG	Primary	psgetm	talogs without parameter substitution	Active	N	
Delete	EOCB_GETSYSTEMVARIABLES	Primary	psgets	Variables	Active	N	
Delete	EOCB_GUID_TO_PSTOKEN	Primary	psauthenticateguid	Authenticate a users chatbot guid	Active	N	

Delete successful.

To delete an Application service, select Delete Application Services from the menu.

When you delete an application service, all of the associated metadata is also deleted.
Only one application service can be deleted at a time.

Click the Delete button for the Application Service ID to delete.
Click Yes to confirm the delete. Then click OK on the delete successful message.



Delete Application Service

Delete Services

Delete

Search

App Service ID	MUSIC
Service Type	
Service URL ID	
Service Group	

Application Services

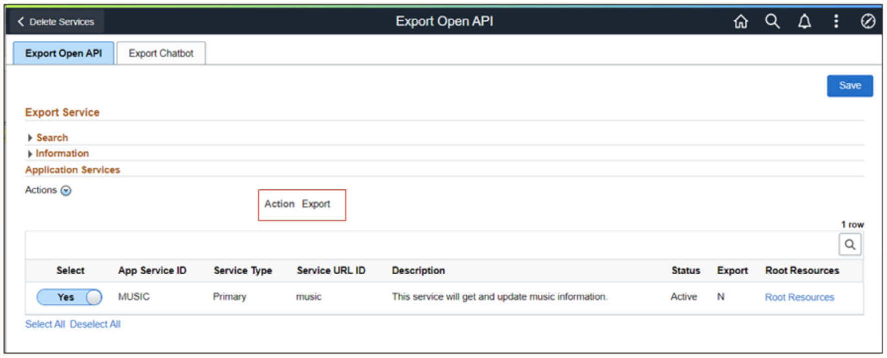
2 rows

Delete	App Service ID	Service Type	Service URL ID	Description	Status	Export	Delete Status
Delete	MUSIC	Primary	music	This service will get and update music information.	Active	N	
Delete	MUSIC_ALIAS	Alias	music	Music Alias	Inactive	N	

Let’s search for the service group demo.
Notice that the delete button is grayed out for music.
An application service defined as primary can only be deleted if the associated Alias has been deleted.
The delete button will also be grayed out if the Application Service has been exported. In order to delete the Application Service, you must first delete the export of the Application Service.

Export Application Services

Menu > PeopleTools> Integration Broker> Application Services> Administration> Export Application Services



Action Export

1 row

Select	App Service ID	Service Type	Service URL ID	Description	Status	Export	Root Resources
Yes	MUSIC	Primary	music	This service will get and update music information.	Active	N	Root Resources

Only the root resources with a status of active can be exported.

Next we will look at exporting an Application Service. Return to the menu and select Export Application Service.
First we will search by service group.

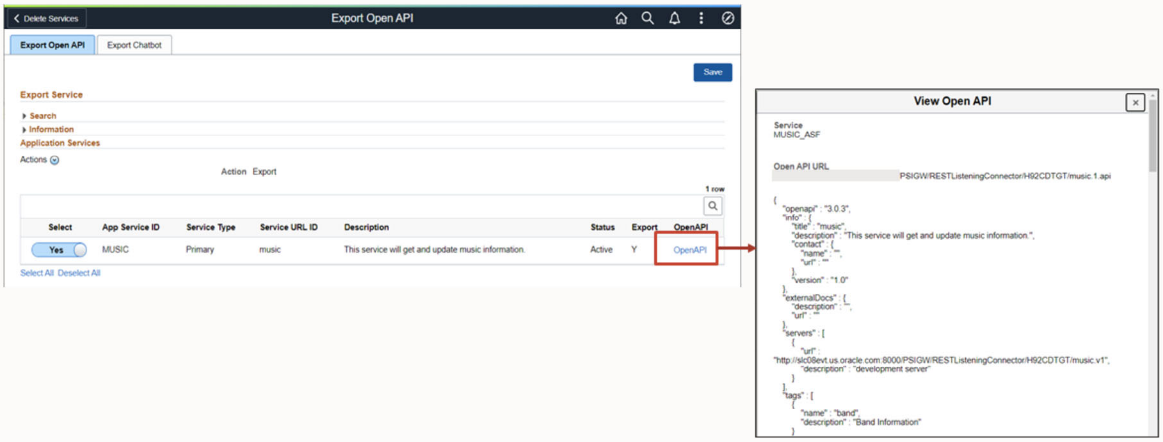
Only the root resources with a status of active can be exported.

The Application Service must be exported to create the REST API web service. Exporting the Application Service will create an Open API document describing the Service.

To export the service, select Export in the actions. Select Yes for the service and save. Click OK on the message that the Application Service was exported.

Export Application Services

Menu > PeopleTools> Integration Broker> Application Services> Administration> Export Application Services



Export Service

- Search
- Information

Application Services

Actions

Action Export

1 row

Select	App Service ID	Service Type	Service URL ID	Description	Status	Export	OpenAPI
Yes	MUSIC	Primary	music	This service will get and update music information.	Active	Y	OpenAPI

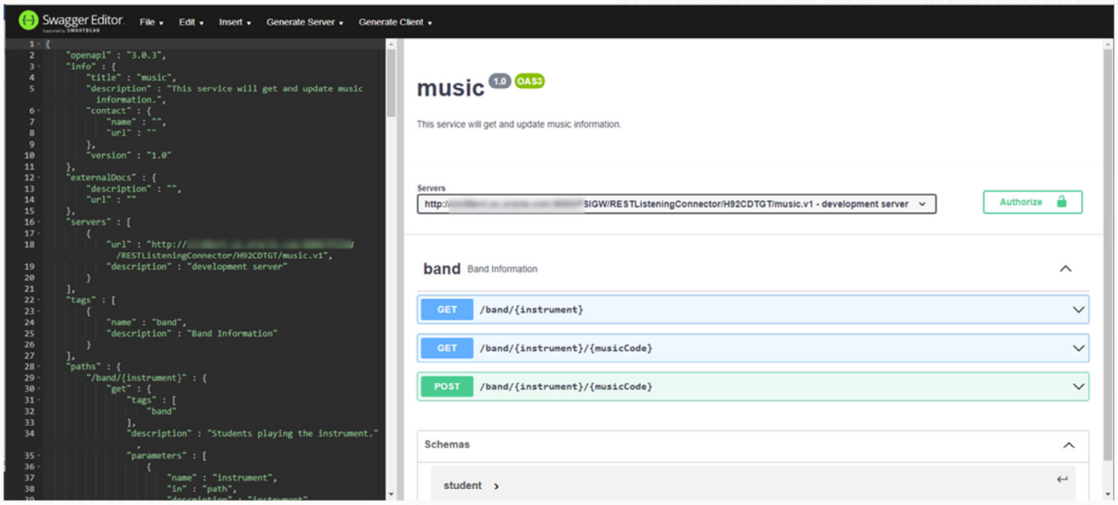
Service
MUSIC_ASF

Open API URL
[redacted]PSIGW/RESTListeningConnector/H92CDTGT/music.1.api

```
{
    "openapi" : "3.0.3",
    "info" : {
        "title" : "music",
        "description" : "This service will get and update music information.",
        "contact" : {
            "name" : "",
            "url" : ""
        },
        "version" : "1.0"
    },
    "externalDocs" : {
        "description" : "",
        "url" : ""
    },
    "servers" : [
        {
            "url" :
"http://slc08evt.us.oracle.com:8000/PSIGW/RESTListeningConnector/H92CDTGT/music.v1",
            "description" : "development server"
        }
    ],
    "tags" : [
        {
            "name" : "band",
            "description" : "Band Information"
        }
```

For Exported Services an OpenAPI link will be displayed.
When selected the OpenAPI document will be displayed in JSON format.

OpenAPI in Swagger



```
{
    "openapi" : "3.0.3",
    "info" : {
        "title" : "music",
        "description" : "This service will get and update music
          information.",
        "contact" : {
            "name" : "",
            "url" : ""
        },
        "version" : "1.0"
    },
    "externalDocs" : {
        "description" : "",
        "url" : ""
    },
    "servers" : [
        {
            "url" : "http://[redacted]
              /RESTListeningConnector/H92CDTGT/music.v1",
            "description" : "development server"
        }
    ],
    "tags" : [
        {
            "name" : "band",
            "description" : "Band Information"
        }
    ],
    "paths" : {
        "/band/{instrument}" : {
            "get" : {
                "tags" : [
                    "band"
                ],
                "description" : "Students playing the instrument."
                  ,
                "parameters" : [
                    {
                        "name" : "instrument",
                        "in" : "path",
                        "description" : "instrument"
```

music 1.0 OAS3

This service will get and update music information.

Servers

http://[redacted]SIGW/RESTListeningConnector/H92CDTGT/music.v1 - development server

band Band Information

GET /band/{instrument}

GET /band/{instrument}/{musicCode}

POST /band/{instrument}/{musicCode}

Schemas

student

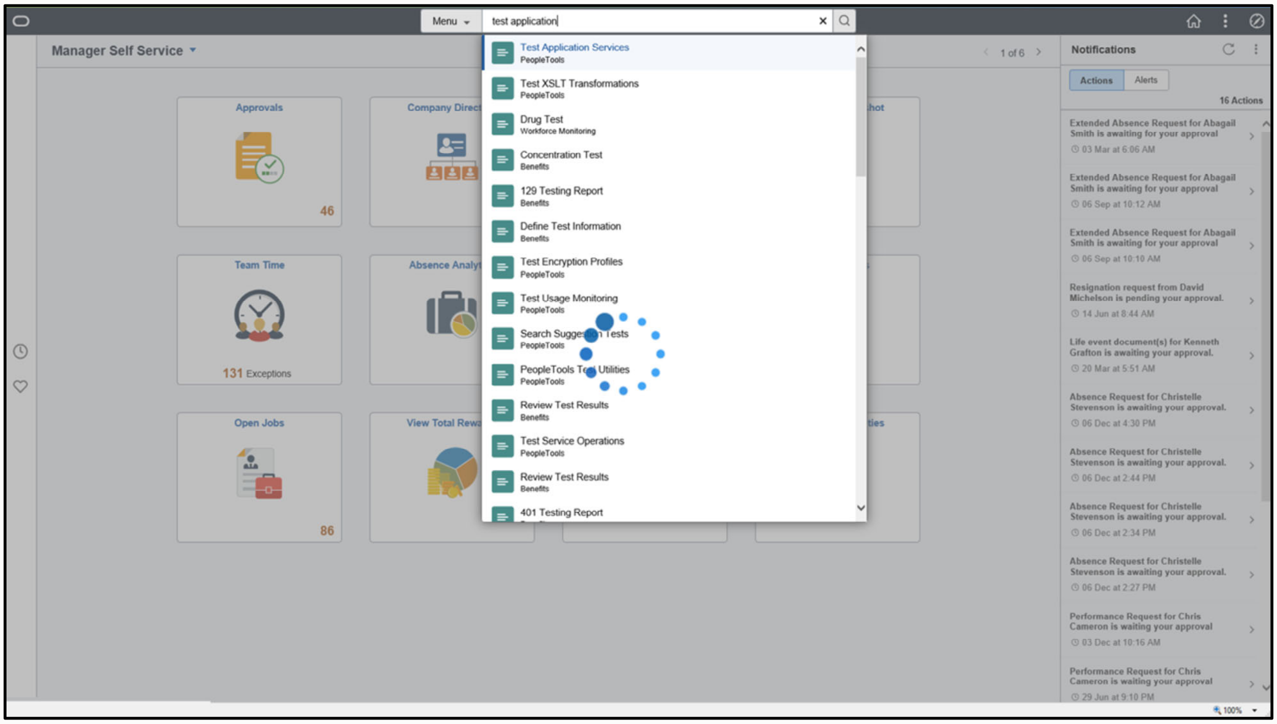
The JSON can be copied and loaded into any OpenAPI Editor to view the document in a human readable form.

This is an example of the OpenAPI in Swagger.

Testing Application Service

The Application Service Tester utility enables you to create a URI template with variables to test your Application Service.

The Application Service Tester provides the ability to test both standard and ChatBot Application Services .



Extended Absence Request for Abagail Smith is awaiting for your approval
03 Mar at 6:06 AM

Extended Absence Request for Abagail Smith is awaiting for your approval
06 Sep at 10:12 AM

Extended Absence Request for Abagail Smith is awaiting for your approval
06 Sep at 10:10 AM

Resignation request from David Michelson is pending your approval.
14 Jun at 8:44 AM

Life event document(s) for Kenneth Grafton is awaiting your approval.
20 Mar at 5:51 AM

Absence Request for Christelle Stevenson is awaiting your approval.
06 Dec at 4:30 PM

Absence Request for Christelle Stevenson is awaiting your approval.
06 Dec at 2:44 PM

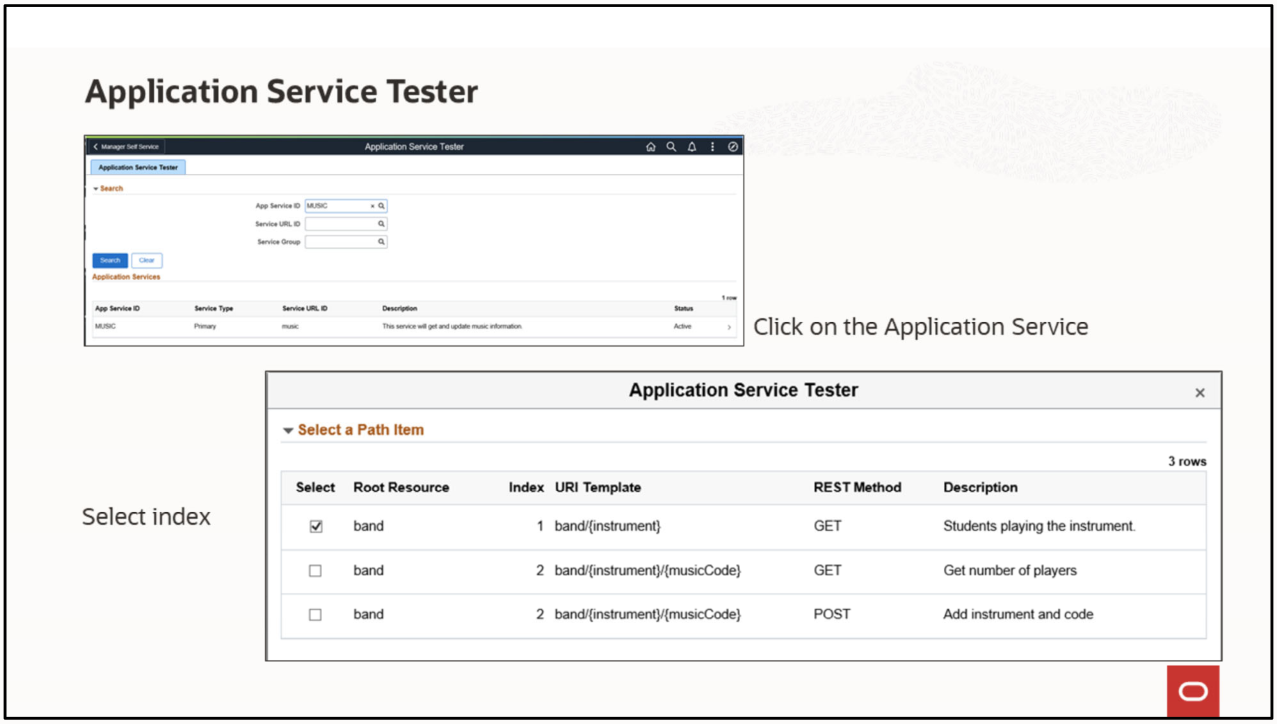
Absence Request for Christelle Stevenson is awaiting your approval.
06 Dec at 2:34 PM

Absence Request for Christelle Stevenson is awaiting your approval.
06 Dec at 2:27 PM

Performance Request for Chris Cameron is waiting your approval
03 Dec at 10:16 AM

Performance Request for Chris Cameron is waiting your approval
29 Jun at 9:10 PM

From the homepage, you can search for Test Application Services.



Application Service Tester

App Service ID: MUSIC

Service URL ID:

Service Group:

1 row

App Service ID	Service Type	Service URL ID	Description	Status
MUSIC	Primary	music	This service will get and update music information.	Active

Click on the Application Service

Application Service Tester

3 rows

Select	Root Resource	Index	URI Template	REST Method	Description
☑	band	1	band/{instrument}	GET	Students playing the instrument.
□	band	2	band/{instrument}/{musicCode}	GET	Get number of players
□	band	2	band/{instrument}/{musicCode}	POST	Add instrument and code

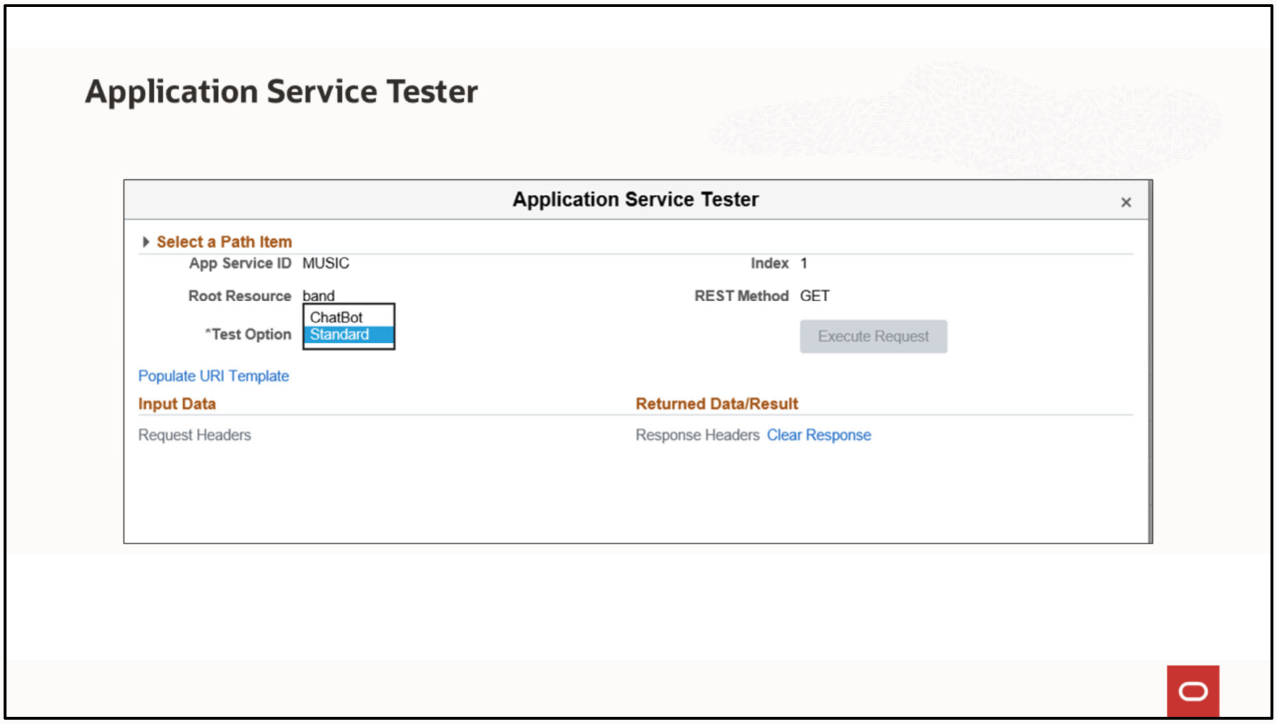
Select index

Use Search to select the application service to test.

We will test Music.

When you click on the Application Service to test, the indexes for the service are displayed.

We will select the first index.



Application Service Tester

Application Service Tester

Select a Path Item

App Service ID MUSIC

Index 1

Root Resource band

REST Method GET

*Test Option ChatBot / Standard

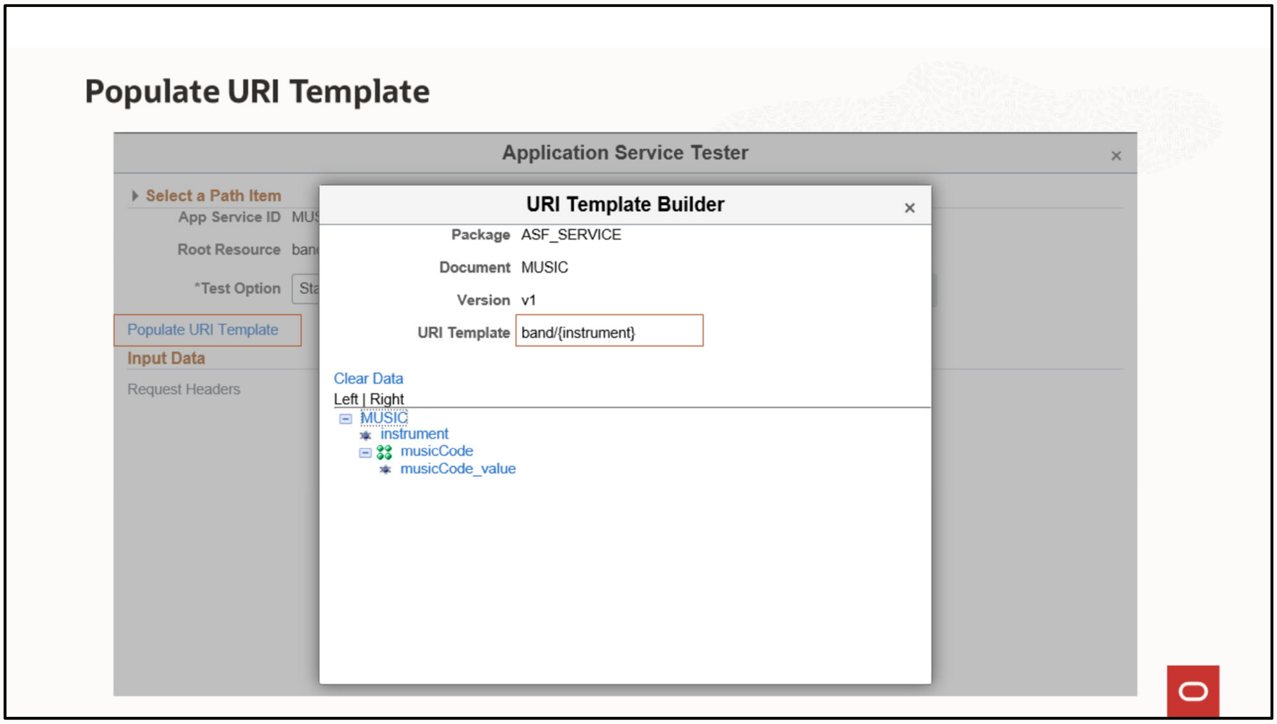
Input Data

Request Headers

Returned Data/Result

Response Headers Clear Response

By default, the test option is standard.
A Chatbot test option is also available. Use this option to test converted Chatbots or new skills created on the new Application Service framework.



Populate URI Template

Application Service Tester

URI Template Builder

Package ASF_SERVICE

Document MUSIC

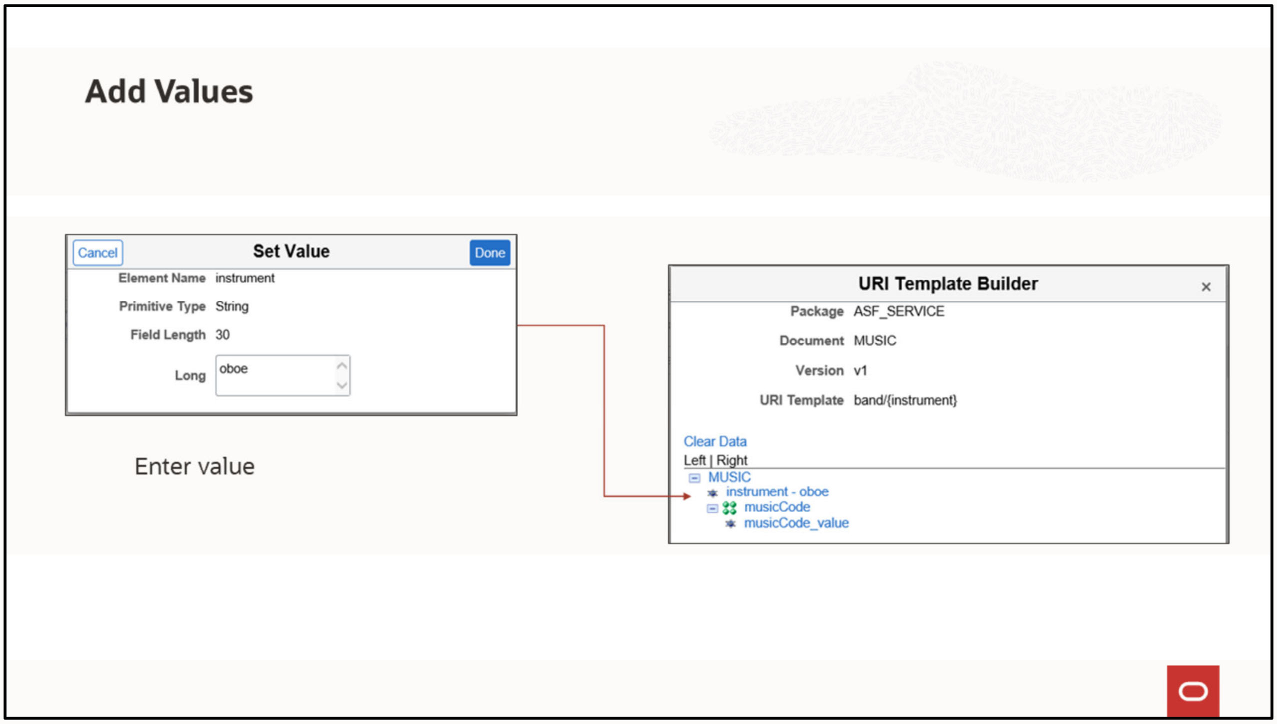
Version v1

URI Template band/{instrument}

Music is a standard Application Service.

Select the Populate URI Template link to enter the variables for your test.

Notice that the URI Template format is displayed. We need to enter a value for instrument.



Add Values

Set Value

Element Name instruments

Primitive Type String

Field Length 30

Long oboe

Enter value

URI Template Builder

Package ASF_SERVICE

Document MUSIC

Version v1

URI Template band/{instrument}

Clear Data

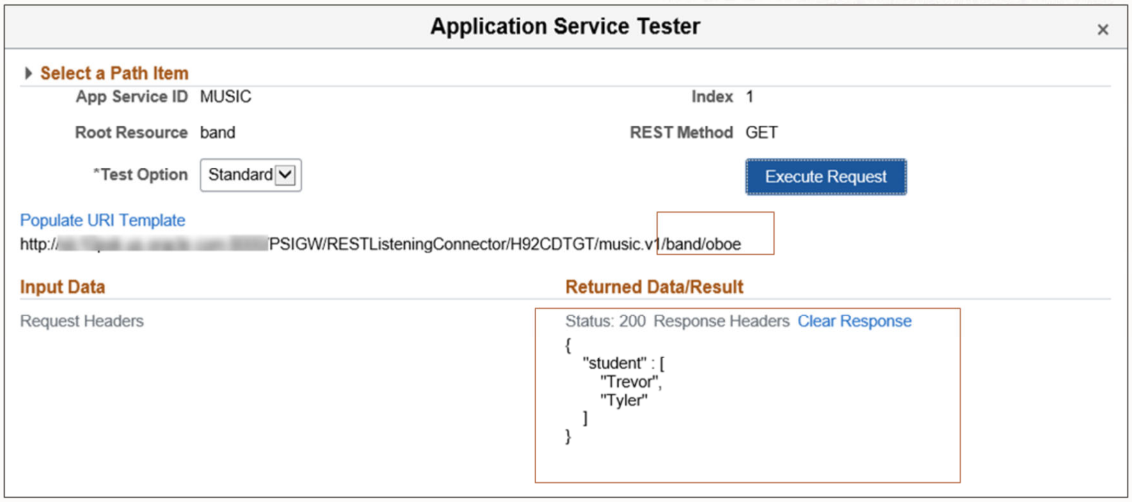
Left | Right

- MUSIC
  - instrument - oboe
  - musicCode
    - musicCode_value

Click the instrument link.
Enter a value and click Done.

When you close the URI Template Builder window, the URL is displayed with the variables.

Execute Request



Application Service Tester

▸ Select a Path Item

App Service ID MUSIC

Index 1

Root Resource band

REST Method GET

*Test Option Standard

Populate URI Template

http://[redacted]PSIGW/RESTListeningConnector/H92CDTGT/music.v1/band/oboe

Input Data

Request Headers

Returned Data/Result

Status: 200 Response Headers Clear Response

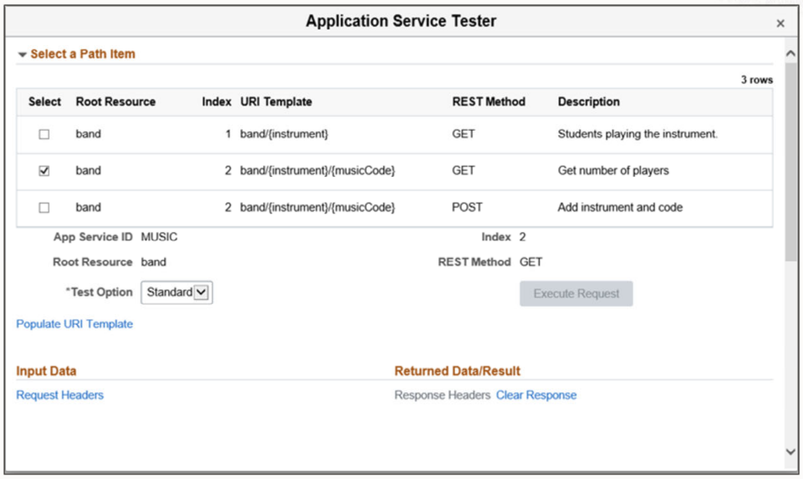
```
{
   "student" : [
       "Trevor",
       "Tyler"
   ]
}
```

When you close the URI Template Builder window, the URL is displayed with the variables.

Click Execute Request to test the Application Service.

The results will be displayed.

Test Another Path



Application Service Tester

▾ Select a Path Item

3 rows

Select	Root Resource	Index	URI Template	REST Method	Description
□	band	1	band/{instrument}	GET	Students playing the instrument.
☑	band	2	band/{instrument}/{musicCode}	GET	Get number of players
□	band	2	band/{instrument}/{musicCode}	POST	Add instrument and code

App Service ID MUSIC

Index 2

Root Resource band

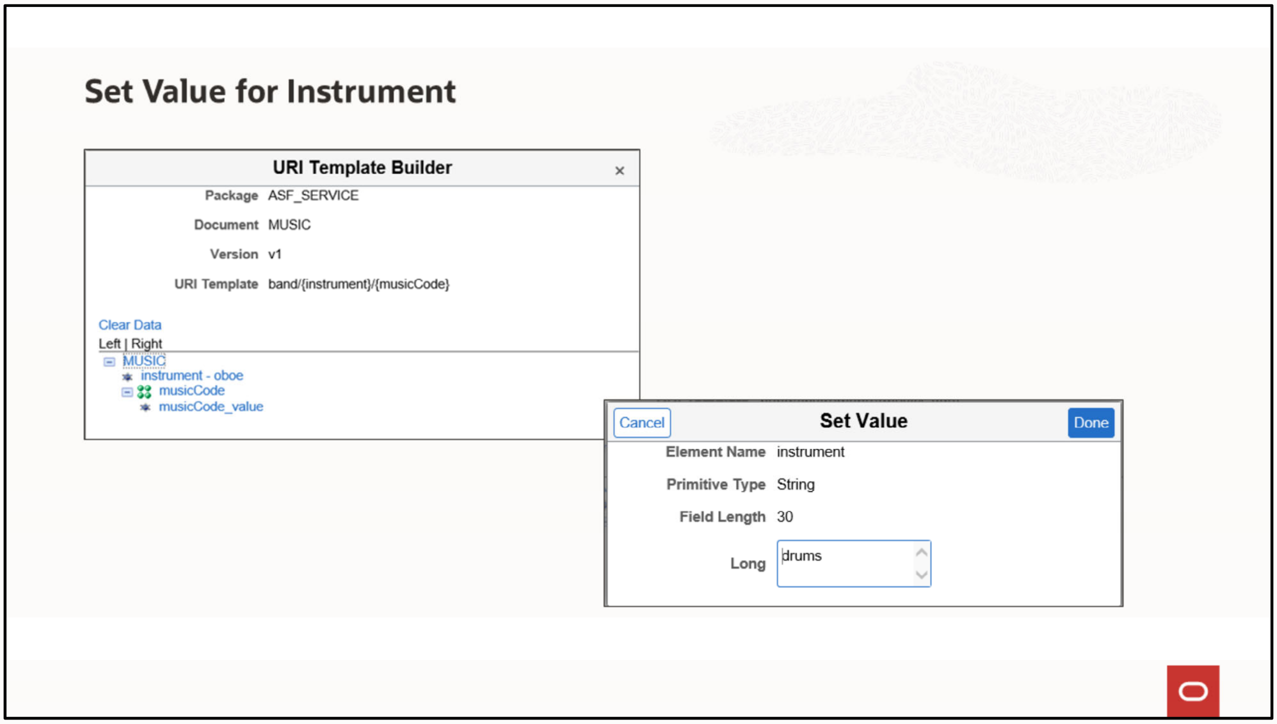
REST Method GET

*Test Option Standard

To test another path, expand the Select a Path Item section.

Select the path to test.

This time we will select index 2.



Set Value for Instrument

URI Template Builder

Package ASF_SERVICE

Document MUSIC

Version v1

URI Template band/{instrument}/{musicCode}

Set Value

Element Name instrument

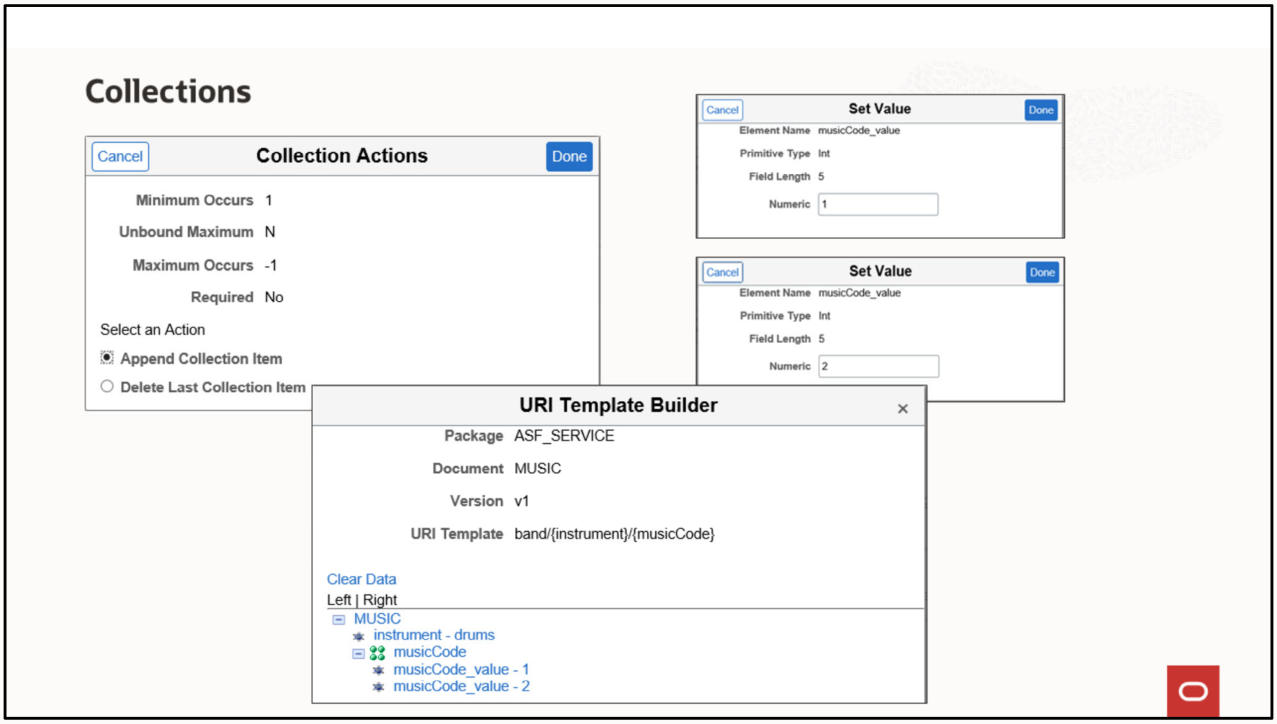
Primitive Type String

Field Length 30

Long drums

Again we need to populate the URI. Select the Populate URI Template link.
This URL includes both instrument and a collection of music codes.
The value for instrument is populated from our previous test. Click instrument to change the value.

First we will enter the instrument value.
We will enter a new value and click done.



Collections

Cancel **Collection Actions** Done

Minimum Occurs 1

Unbound Maximum N

Maximum Occurs -1

Required No

Select an Action

● Append Collection Item

○ Delete Last Collection Item

Cancel **Set Value** Done

Element Name musicCode_value

Primitive Type Int

Field Length 5

Numeric 1

Cancel **Set Value** Done

Element Name musicCode_value

Primitive Type Int

Field Length 5

Numeric 2

URI Template Builder ×

Package ASF_SERVICE

Document MUSIC

Version v1

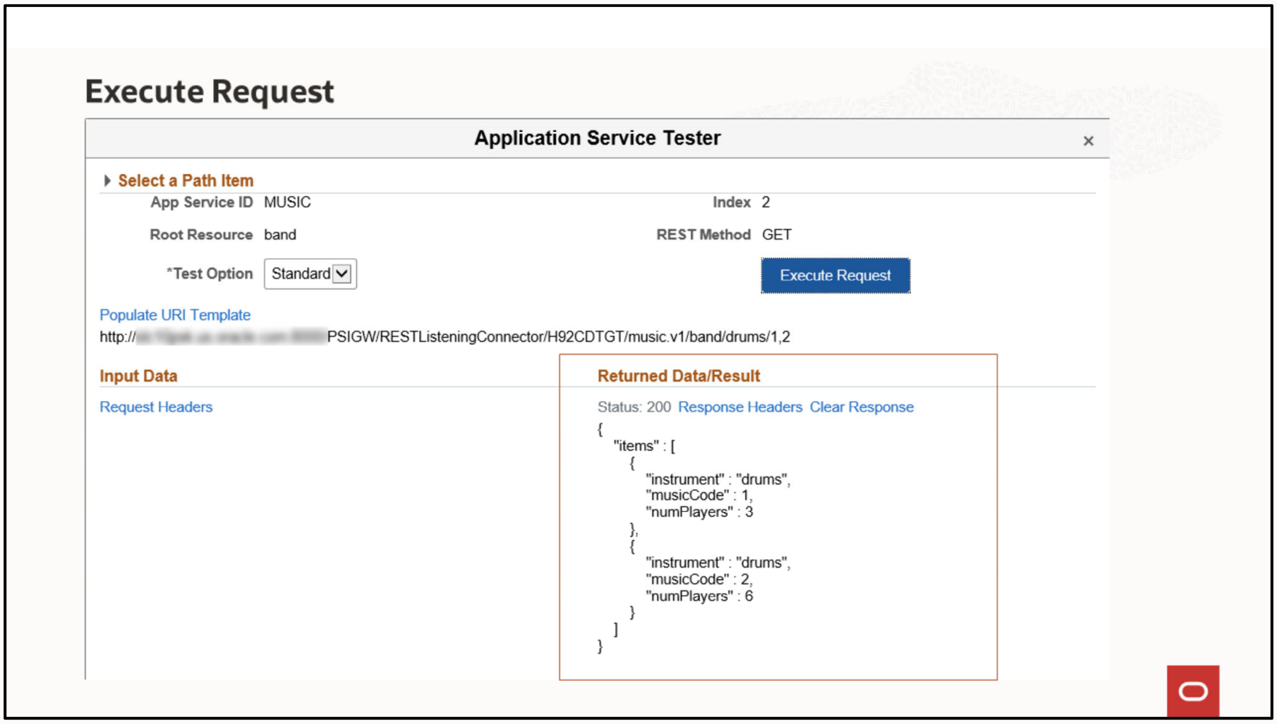
URI Template band/{instrument}/{musicCode}

Clear Data

Left | Right

- MUSIC
  - instrument - drums
  - musicCode
    - musicCode_value - 1
    - musicCode_value - 2

Click on musicCode to select the collection actions.
For collections, you can append a collection item or delete the last collection item.
 We want to append a collection item.
Click Done.
First we will enter a value for the first music code.
Enter a value and click Done.
Select the second music code, enter a value and click Done



Execute Request

App Service ID MUSIC

Index 2

Root Resource band

REST Method GET

*Test Option Standard

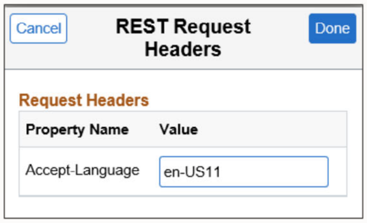
http://[redacted]PSIGW/RESTListeningConnector/H92CDTGT/music.v1/band/drums/1,2

Status: 200

```
{
    "items" : [
        {
            "instrument" : "drums",
            "musicCode" : 1,
            "numPlayers" : 3
        },
        {
            "instrument" : "drums",
            "musicCode" : 2,
            "numPlayers" : 6
        }
    ]
}
```

When we close the URI Template Builder, we see the URL with the variables.
Again we can execute the request.
The response shows 2 items.

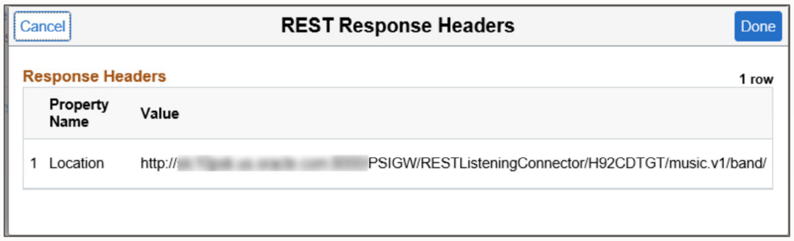
Headers



Request Headers

Property Name	Value
Accept-Language	en-US11

Request Header



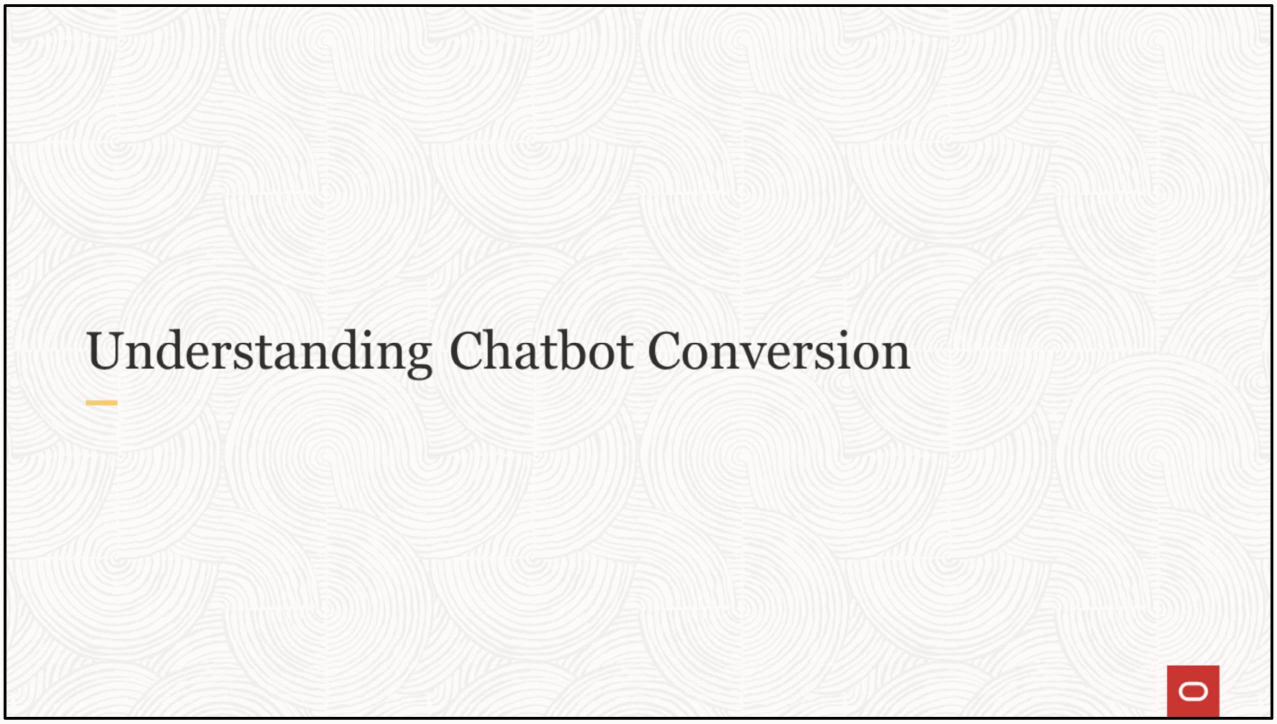
Response Headers 1 row

	Property Name	Value
1	Location	http://[redacted]PSIGW/RESTListeningConnector/H92CDTGT/music.v1/band/

Response Header

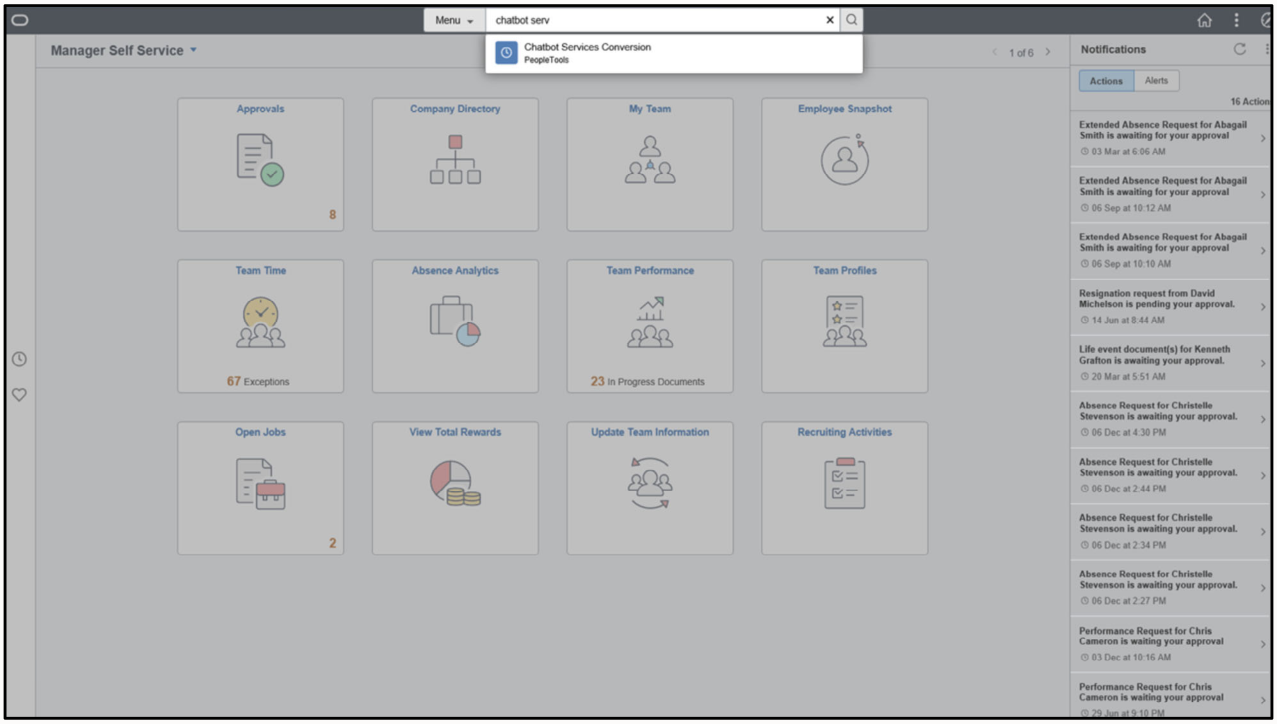
For this method, we created request and response headers.
Select the Request Header link to view the request header.

Select the Response Headers link to view the response header and then click Done.



Understanding Chatbot Conversion

Oracle Digital Assistant Skills created in a PeopleTools release prior to 8.59 were referred to as chatbots, these chatbots require conversion to comply with all Oracle REST standards. This conversion is performed as part of the upgrade to PeopleTools 8.59.



Extended Absence Request for Abagail Smith is awaiting for your approval
03 Mar at 6:06 AM

Extended Absence Request for Abagail Smith is awaiting for your approval
06 Sep at 10:12 AM

Extended Absence Request for Abagail Smith is awaiting for your approval
06 Sep at 10:10 AM

Resignation request from David Michelson is pending your approval.
14 Jun at 8:44 AM

Life event document(s) for Kenneth Grafton is awaiting your approval.
20 Mar at 5:51 AM

Absence Request for Christelle Stevenson is awaiting your approval.
06 Dec at 4:30 PM

Absence Request for Christelle Stevenson is awaiting your approval.
06 Dec at 2:44 PM

Absence Request for Christelle Stevenson is awaiting your approval.
06 Dec at 2:34 PM

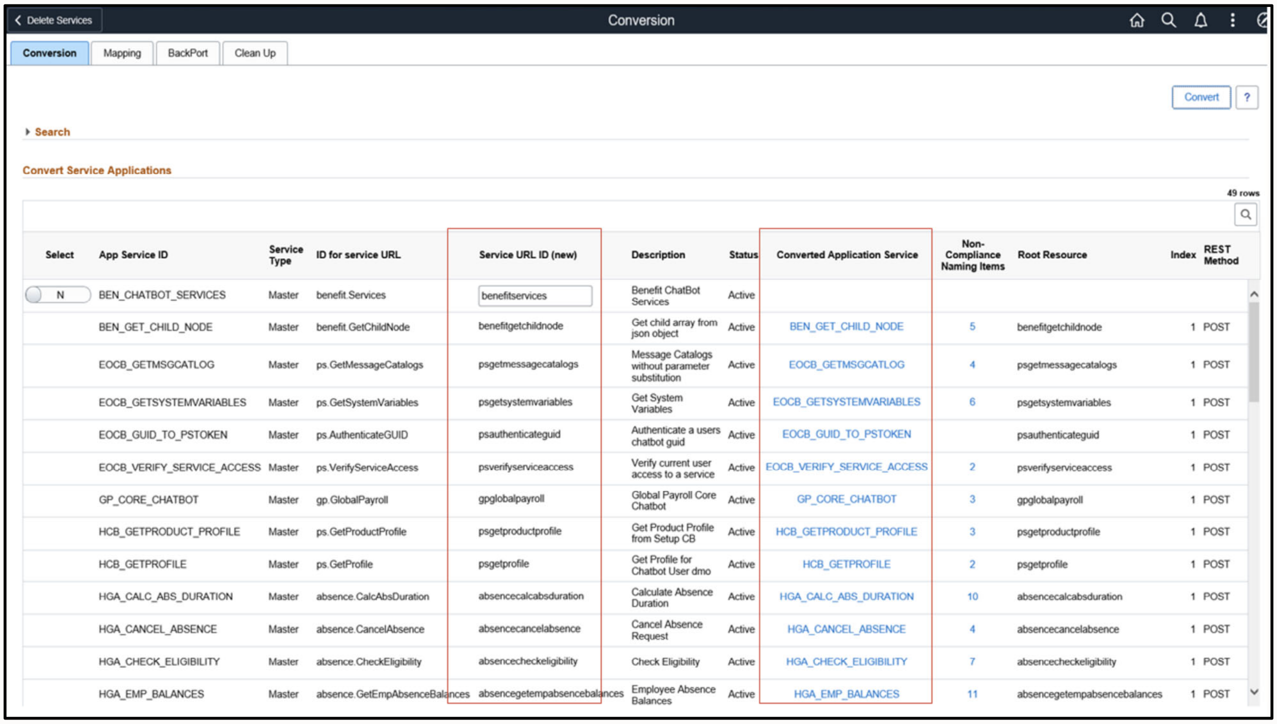
Absence Request for Christelle Stevenson is awaiting your approval.
06 Dec at 2:27 PM

Performance Request for Chris Cameron is waiting your approval
03 Dec at 10:16 AM

Performance Request for Chris Cameron is waiting your approval
29 Jun at 9:10 PM

A chatbot conversion utility is available and can be used if necessary, to run the conversion again.

Enter Chatbot Services Conversion in the Search bar to access the utility.



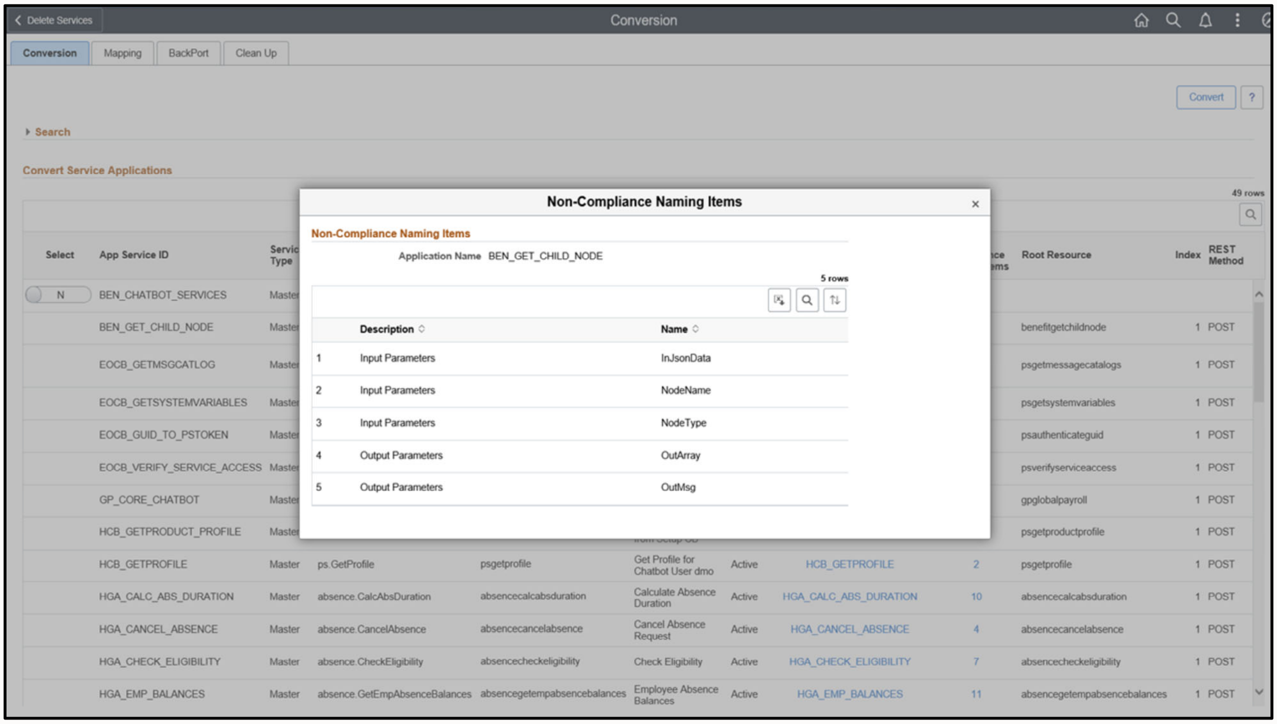
Convert Service Applications

49 rows

Select	App Service ID	Service Type	ID for service URL	Service URL ID (new)	Description	Status	Converted Application Service	Non-Compliance Naming Items	Root Resource	Index	REST Method
N	BEN_CHATBOT_SERVICES	Master	benefit.Services	benefitservices	Benefit ChatBot Services	Active					
	BEN_GET_CHILD_NODE	Master	benefit.GetChildNode	benefitgetchildnode	Get child array from json object	Active	BEN_GET_CHILD_NODE	5	benefitgetchildnode	1	POST
	EOCB_GETMSGCATLOG	Master	ps.GetMessageCatalogs	psgetmessagecatalogs	Message Catalogs without parameter substitution	Active	EOCB_GETMSGCATLOG	4	psgetmessagecatalogs	1	POST
	EOCB_GETSYSTEMVARIABLES	Master	ps.GetSystemVariables	psgetsystemvariables	Get System Variables	Active	EOCB_GETSYSTEMVARIABLES	6	psgetsystemvariables	1	POST
	EOCB_GUID_TO_PSTOKEN	Master	ps.AuthenticateGUID	psauthenticateguid	Authenticate a users chatbot guid	Active	EOCB_GUID_TO_PSTOKEN		psauthenticateguid	1	POST
	EOCB_VERIFY_SERVICE_ACCESS	Master	ps.VerifyServiceAccess	psverifyserviceaccess	Verify current user access to a service	Active	EOCB_VERIFY_SERVICE_ACCESS	2	psverifyserviceaccess	1	POST
	GP_CORE_CHATBOT	Master	gp.GlobalPayroll	gpglobalpayroll	Global Payroll Core Chatbot	Active	GP_CORE_CHATBOT	3	gpglobalpayroll	1	POST
	HCB_GETPRODUCT_PROFILE	Master	ps.GetProductProfile	psgetproductprofile	Get Product Profile from Setup CB	Active	HCB_GETPRODUCT_PROFILE	3	psgetproductprofile	1	POST
	HCB_GETPROFILE	Master	ps.GetProfile	psgetprofile	Get Profile for Chatbot User dmo	Active	HCB_GETPROFILE	2	psgetprofile	1	POST
	HGA_CALC_ABS_DURATION	Master	absence.CalcAbsDuration	absencecalcabsduration	Calculate Absence Duration	Active	HGA_CALC_ABS_DURATION	10	absencecalcabsduration	1	POST
	HGA_CANCEL_ABSENCE	Master	absence.CancelAbsence	absencecancelabsence	Cancel Absence Request	Active	HGA_CANCEL_ABSENCE	4	absencecancelabsence	1	POST
	HGA_CHECK_ELIGIBILITY	Master	absence.CheckEligibility	absencecheckeligibility	Check Eligibility	Active	HGA_CHECK_ELIGIBILITY	7	absencecheckeligibility	1	POST
	HGA_EMP_BALANCES	Master	absence.GetEmpAbsenceBalances	absencegetempabsencebalances	Employee Absence Balances	Active	HGA_EMP_BALANCES	11	absencegetempabsencebalances	1	POST

The conversion page displays converted *ID for service URL (new)* which conforms to the Oracle REST standard.

In this example, the chatbots that were converted as part of the PeopleTools 8.59 upgrade have a converted Application service name.



Convert Service Applications

49 rows

Select	App Service ID	Servic Type						ce ems	Root Resource	Index	REST Method
N	BEN_CHATBOT_SERVICES	Master									
	BEN_GET_CHILD_NODE	Master							benefitgetchildnode	1	POST
	EOCB_GETMSGCATLOG	Master							psgetmessagecatalogs	1	POST
	EOCB_GETSYSTEMVARIABLES	Master							psgetsystemvariables	1	POST
	EOCB_GUID_TO_PSTOKEN	Master							psauthenticateguid	1	POST
	EOCB_VERIFY_SERVICE_ACCESS	Master							psverifyserviceaccess	1	POST
	GP_CORE_CHATBOT	Master							gpglobalpayroll	1	POST
	HCB_GETPRODUCT_PROFILE	Master							psgetproductprofile	1	POST
	HCB_GETPROFILE	Master	ps.GetProfile	psgetprofile	Get Profile for Chatbot User dmo	Active	HCB_GETPROFILE	2	psgetprofile	1	POST
	HGA_CALC_ABS_DURATION	Master	absence.CalcAbsDuration	absencecalcabsduration	Calculate Absence Duration	Active	HGA_CALC_ABS_DURATION	10	absencecalcabsduration	1	POST
	HGA_CANCEL_ABSENCE	Master	absence.CancelAbsence	absencecancelabsence	Cancel Absence Request	Active	HGA_CANCEL_ABSENCE	4	absencecancelabsence	1	POST
	HGA_CHECK_ELIGIBILITY	Master	absence.CheckEligibility	absencecheckeligibility	Check Eligibility	Active	HGA_CHECK_ELIGIBILITY	7	absencecheckeligibility	1	POST
	HGA_EMP_BALANCES	Master	absence.GetEmpAbsenceBalances	absencegetempabsencebalances	Employee Absence Balances	Active	HGA_EMP_BALANCES	11	absencegetempabsencebalances	1	POST

Non-Compliance Naming Items

Non-Compliance Naming Items

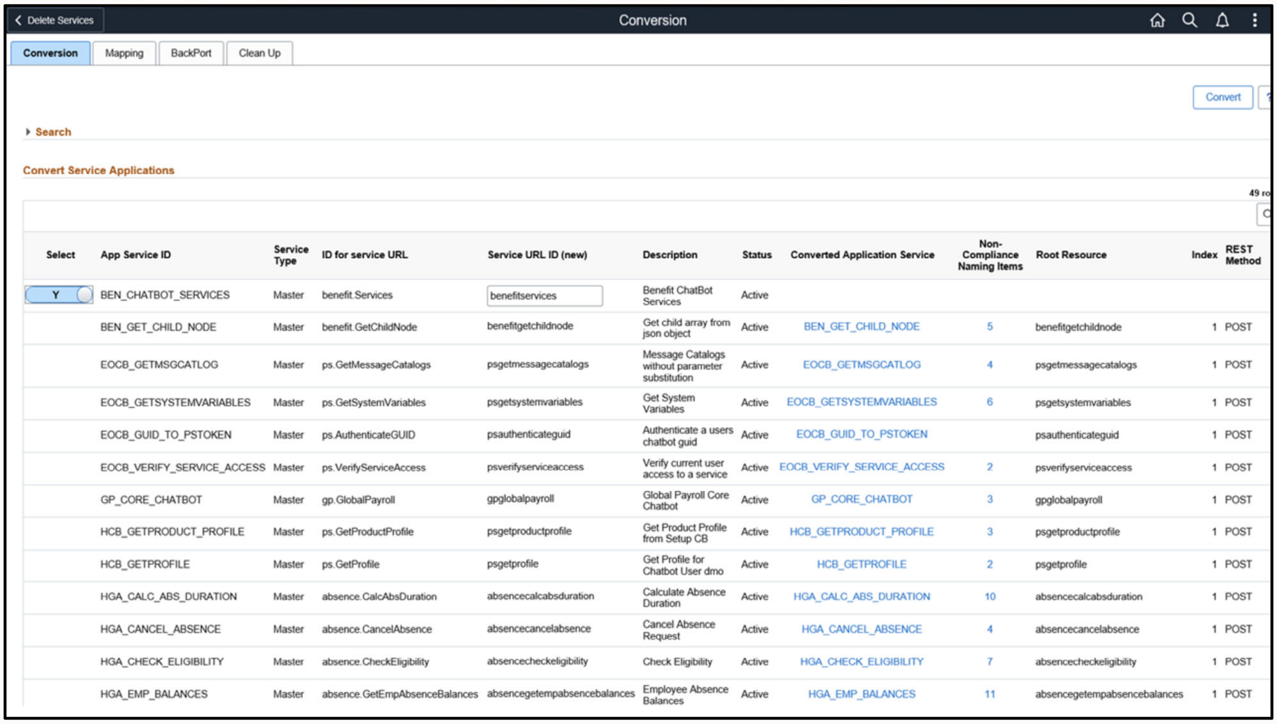
Application Name BEN_GET_CHILD_NODE

5 rows

	Description	Name
1	Input Parameters	InJsonData
2	Input Parameters	NodeName
3	Input Parameters	NodeType
4	Output Parameters	OutArray
5	Output Parameters	OutMsg

Click the number to view the non-compliant naming items.

Close the window.

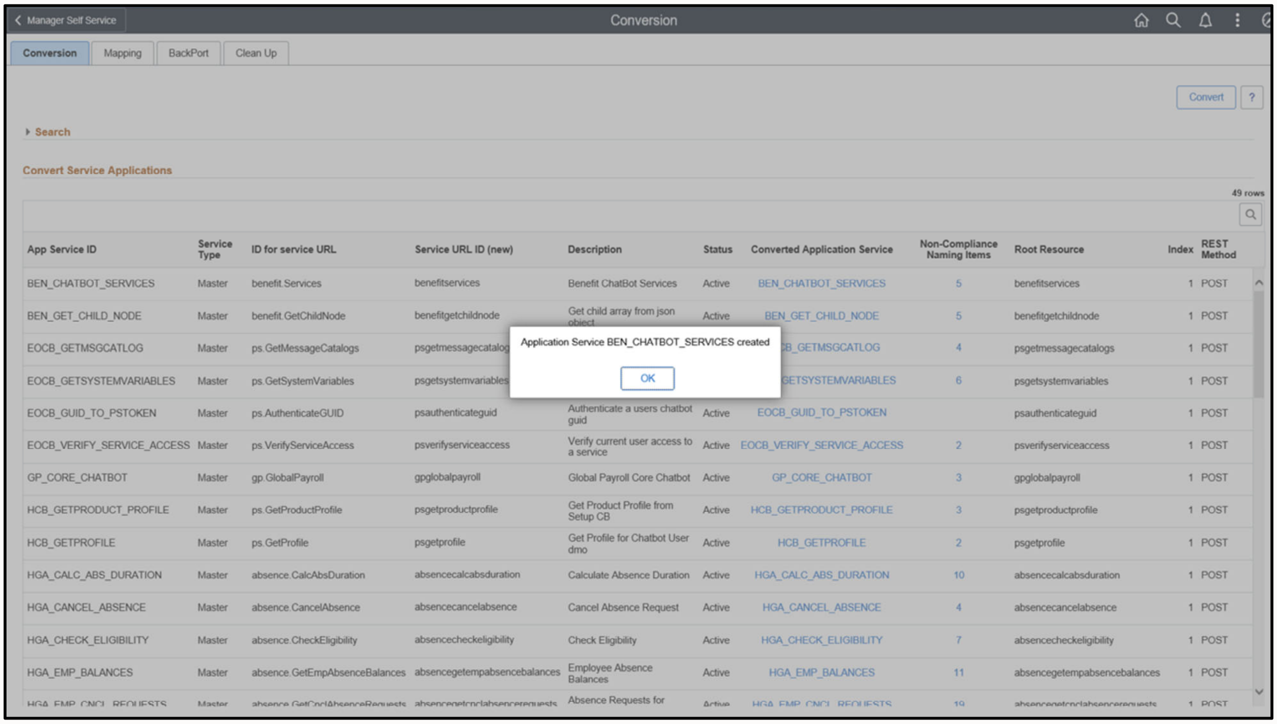


Convert Service Applications

Select	App Service ID	Service Type	ID for service URL	Service URL ID (new)	Description	Status	Converted Application Service	Non-Compliance Naming Items	Root Resource	Index	REST Method
Y	BEN_CHATBOT_SERVICES	Master	benefit.Services	benefitservices	Benefit ChatBot Services	Active					
	BEN_GET_CHILD_NODE	Master	benefit.GetChildNode	benefitgetchildnode	Get child array from json object	Active	BEN_GET_CHILD_NODE	5	benefitgetchildnode	1	POST
	EOCB_GETMSGCATLOG	Master	ps.GetMessageCatalogs	psgetmessagecatalogs	Message Catalogs without parameter substitution	Active	EOCB_GETMSGCATLOG	4	psgetmessagecatalogs	1	POST
	EOCB_GETSYSTEMVARIABLES	Master	ps.GetSystemVariables	psgetsystemvariables	Get System Variables	Active	EOCB_GETSYSTEMVARIABLES	6	psgetsystemvariables	1	POST
	EOCB_GUID_TO_PSTOKEN	Master	ps.AuthenticateGUID	psauthenticateguid	Authenticate a users chatbot guid	Active	EOCB_GUID_TO_PSTOKEN		psauthenticateguid	1	POST
	EOCB_VERIFY_SERVICE_ACCESS	Master	ps.VerifyServiceAccess	psverifyserviceaccess	Verify current user access to a service	Active	EOCB_VERIFY_SERVICE_ACCESS	2	psverifyserviceaccess	1	POST
	GP_CORE_CHATBOT	Master	gp.GlobalPayroll	gpglobalpayroll	Global Payroll Core Chatbot	Active	GP_CORE_CHATBOT	3	gpglobalpayroll	1	POST
	HCB_GETPRODUCT_PROFILE	Master	ps.GetProductProfile	psgetproductprofile	Get Product Profile from Setup CB	Active	HCB_GETPRODUCT_PROFILE	3	psgetproductprofile	1	POST
	HCB_GETPROFILE	Master	ps.GetProfile	psgetprofile	Get Profile for Chatbot User dmo	Active	HCB_GETPROFILE	2	psgetprofile	1	POST
	HGA_CALC_ABS_DURATION	Master	absence.CalcAbsDuration	absencecalcabsduration	Calculate Absence Duration	Active	HGA_CALC_ABS_DURATION	10	absencecalcabsduration	1	POST
	HGA_CANCEL_ABSENCE	Master	absence.CancelAbsence	absencecancelabsence	Cancel Absence Request	Active	HGA_CANCEL_ABSENCE	4	absencecancelabsence	1	POST
	HGA_CHECK_ELIGIBILITY	Master	absence.CheckEligibility	absencecheckeligibility	Check Eligibility	Active	HGA_CHECK_ELIGIBILITY	7	absencecheckeligibility	1	POST
	HGA_EMP_BALANCES	Master	absence.GetEmpAbsenceBalances	absencegetempabsencebalances	Employee Absence Balances	Active	HGA_EMP_BALANCES	11	absencegetempabsencebalances	1	POST

Any existing chatbots that were not converted during the upgrade or were deleted can be manually converted by selecting the Application Service.

Optionally, you can update the value in the Service URL ID (new) edit box.



Convert Service Applications

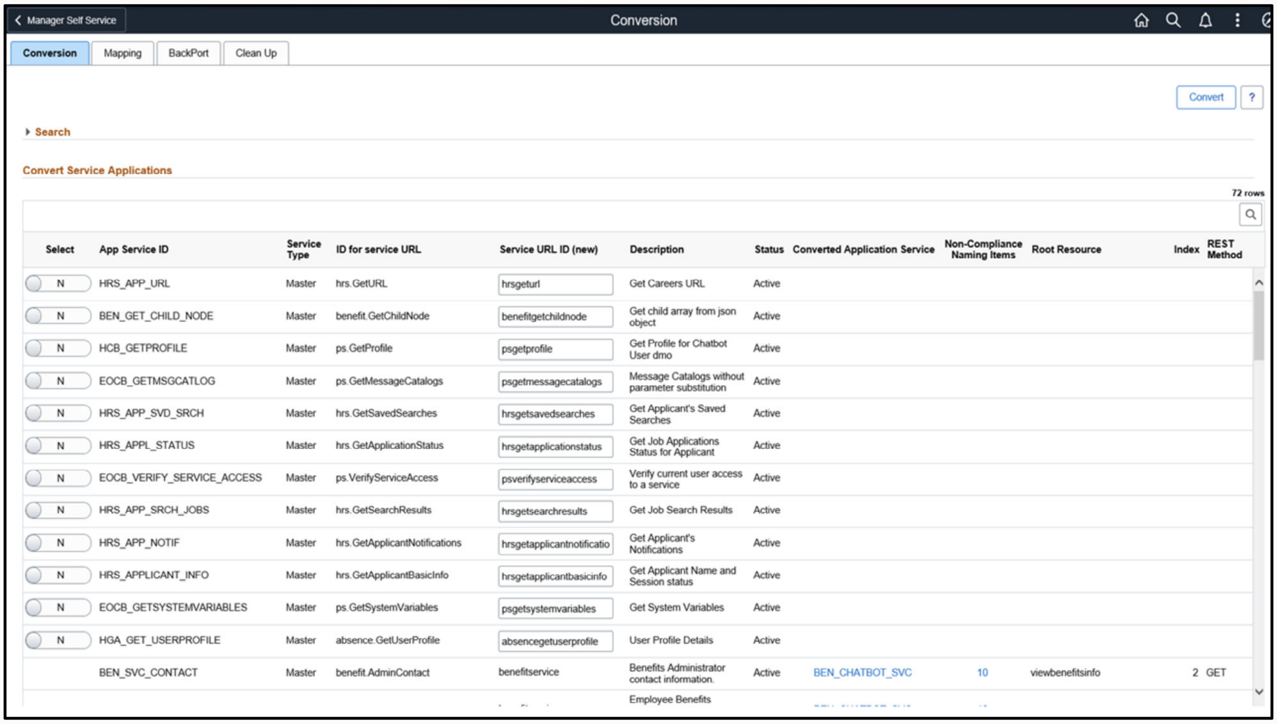
49 rows

App Service ID	Service Type	ID for service URL	Service URL ID (new)	Description	Status	Converted Application Service	Non-Compliance Naming Items	Root Resource	Index	REST Method
BEN_CHATBOT_SERVICES	Master	benefit.Services	benefitservices	Benefit ChatBot Services	Active	BEN_CHATBOT_SERVICES	5	benefitservices	1	POST
BEN_GET_CHILD_NODE	Master	benefit.GetChildNode	benefitgetchildnode	Get child array from json object	Active	BEN_GET_CHILD_NODE	5	benefitgetchildnode	1	POST
EOCB_GETMSGCATLOG	Master	ps.GetMessageCatalogs	psgetmessagecatalog			CB_GETMSGCATLOG	4	psgetmessagecatalogs	1	POST
EOCB_GETSYSTEMVARIABLES	Master	ps.GetSystemVariables	psgetsystemvariables			GETSYSTEMVARIABLES	6	psgetsystemvariables	1	POST
EOCB_GUID_TO_PSTOKEN	Master	ps.AuthenticateGUID	psauthenticateguid	Authenticate a users chatbot guid	Active	EOCB_GUID_TO_PSTOKEN		psauthenticateguid	1	POST
EOCB_VERIFY_SERVICE_ACCESS	Master	ps.VerifyServiceAccess	psverifyserviceaccess	Verify current user access to a service	Active	EOCB_VERIFY_SERVICE_ACCESS	2	psverifyserviceaccess	1	POST
GP_CORE_CHATBOT	Master	gp.GlobalPayroll	gpglobalpayroll	Global Payroll Core Chatbot	Active	GP_CORE_CHATBOT	3	gpglobalpayroll	1	POST
HCB_GETPRODUCT_PROFILE	Master	ps.GetProductProfile	psgetproductprofile	Get Product Profile from Setup CB	Active	HCB_GETPRODUCT_PROFILE	3	psgetproductprofile	1	POST
HCB_GETPROFILE	Master	ps.GetProfile	psgetprofile	Get Profile for Chatbot User dmo	Active	HCB_GETPROFILE	2	psgetprofile	1	POST
HGA_CALC_ABS_DURATION	Master	absence.CalcAbsDuration	absencecalcabsduration	Calculate Absence Duration	Active	HGA_CALC_ABS_DURATION	10	absencecalcabsduration	1	POST
HGA_CANCEL_ABSENCE	Master	absence.CancelAbsence	absencecancelabsence	Cancel Absence Request	Active	HGA_CANCEL_ABSENCE	4	absencecancelabsence	1	POST
HGA_CHECK_ELIGIBILITY	Master	absence.CheckEligibility	absencecheckeligibility	Check Eligibility	Active	HGA_CHECK_ELIGIBILITY	7	absencecheckeligibility	1	POST
HGA_EMP_BALANCES	Master	absence.GetEmpAbsenceBalances	absencegetempabsencebalances	Employee Absence Balances	Active	HGA_EMP_BALANCES	11	absencegetempabsencebalances	1	POST
HGA_EMP_CNCL_REQUESTS	Master	absence.GetCnclAbsenceRequests	absencegetcnclabsencerequests	Absence Requests for	Active	HGA_EMP_CNCL_REQUESTS	19	absencegetcnclabsencerequests	1	POST

Application Service BEN_CHATBOT_SERVICES created

When you select Convert, the conversion page will copy the data out of the old Application Service tables and build a new Application Service in the new framework.

Click OK on the message that the service was created.



Search

Convert Service Applications

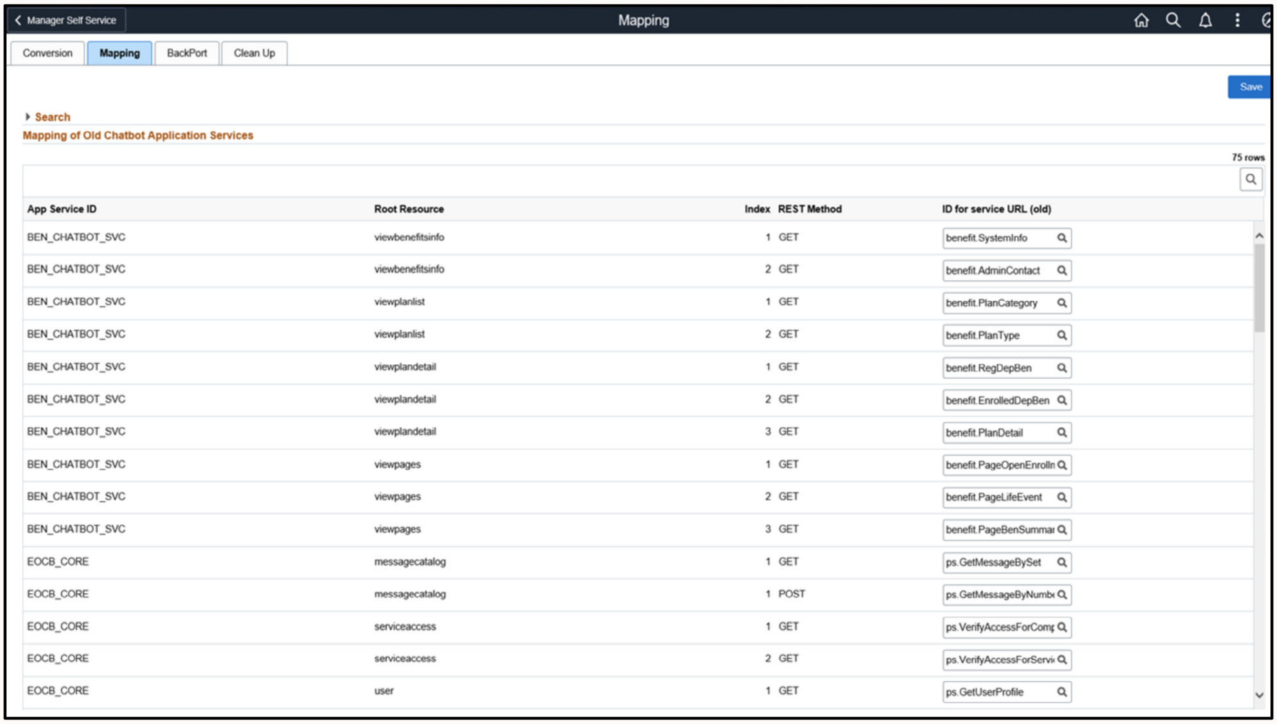
72 rows

Select	App Service ID	Service Type	ID for service URL	Service URL ID (new)	Description	Status	Converted Application Service	Non-Compliance Naming Items	Root Resource	Index	REST Method
N	HRS_APP_URL	Master	hrs.GetURL	hrsgeturl	Get Careers URL	Active					
N	BEN_GET_CHILD_NODE	Master	benefit.GetChildNode	benefitgetchildnode	Get child array from json object	Active					
N	HCB_GETPROFILE	Master	ps.GetProfile	psgetprofile	Get Profile for Chatbot User dmo	Active					
N	EOCB_GETMSGCATLOG	Master	ps.GetMessageCatalogs	psgetmessagecatalogs	Message Catalogs without parameter substitution	Active					
N	HRS_APP_SVD_SRCH	Master	hrs.GetSavedSearches	hrsgetsavedsearches	Get Applicant's Saved Searches	Active					
N	HRS_APPL_STATUS	Master	hrs.GetApplicationStatus	hrsgetapplicationstatus	Get Job Applications Status for Applicant	Active					
N	EOCB_VERIFY_SERVICE_ACCESS	Master	ps.VerifyServiceAccess	psverifyserviceaccess	Verify current user access to a service	Active					
N	HRS_APP_SRCH_JOBS	Master	hrs.GetSearchResults	hrsgetsearchresults	Get Job Search Results	Active					
N	HRS_APP_NOTIF	Master	hrs.GetApplicantNotifications	hrsgetapplicantnotificatio	Get Applicant's Notifications	Active					
N	HRS_APPLICANT_INFO	Master	hrs.GetApplicantBasicInfo	hrsgetapplicantbasicinfo	Get Applicant Name and Session status	Active					
N	EOCB_GETSYSTEMVARIABLES	Master	ps.GetSystemVariables	psgetsystemvariables	Get System Variables	Active					
N	HGA_GET_USERPROFILE	Master	absence.GetUserProfile	absencegetuserprofile	User Profile Details	Active					
	BEN_SVC_CONTACT	Master	benefit.AdminContact	benefitservice	Benefits Administrator contact information.	Active	BEN_CHATBOT_SVC	10	viewbenefitsinfo	2	GET
					Employee Benefits						

This is an example of the Chatbot Conversion utility on a PeopleSoft Update Image.

PeopleSoft applications will update the delivered Digital Assistant Skills as REST compliant Application Services. The updated Application Services will be delivered in the PeopleSoft Update Image for the application.

Since the customer's target database may be on a previous release, it is necessary to map the modified Application Service to the original old Application Service.



Mapping of Old Chatbot Application Services

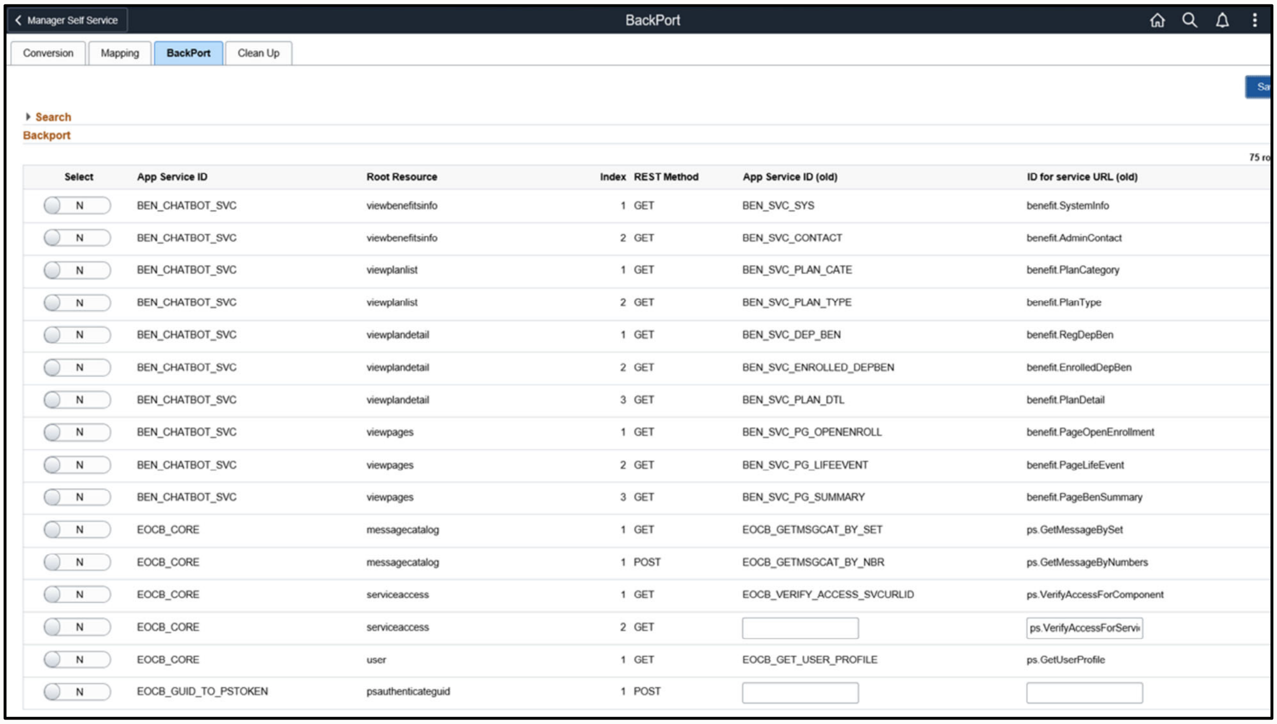
75 rows

App Service ID	Root Resource	Index	REST Method	ID for service URL (old)
BEN_CHATBOT_SVC	viewbenefitsinfo	1	GET	benefit.SystemInfo
BEN_CHATBOT_SVC	viewbenefitsinfo	2	GET	benefit.AdminContact
BEN_CHATBOT_SVC	viewplanlist	1	GET	benefit.PlanCategory
BEN_CHATBOT_SVC	viewplanlist	2	GET	benefit.PlanType
BEN_CHATBOT_SVC	viewplandetail	1	GET	benefit.RegDepBen
BEN_CHATBOT_SVC	viewplandetail	2	GET	benefit.EnrolledDepBen
BEN_CHATBOT_SVC	viewplandetail	3	GET	benefit.PlanDetail
BEN_CHATBOT_SVC	viewpages	1	GET	benefit.PageOpenEnrolln
BEN_CHATBOT_SVC	viewpages	2	GET	benefit.PageLifeEvent
BEN_CHATBOT_SVC	viewpages	3	GET	benefit.PageBenSummar
EOCB_CORE	messagecatalog	1	GET	ps.GetMessageBySet
EOCB_CORE	messagecatalog	1	POST	ps.GetMessageByNumb
EOCB_CORE	serviceaccess	1	GET	ps.VerifyAccessForComp
EOCB_CORE	serviceaccess	2	GET	ps.VerifyAccessForServi
EOCB_CORE	user	1	GET	ps.GetUserProfile

Select the Mapping page on the PUM source to view the mapping.

This mapping is required for proper backporting and to generate a properly defined chatbot discovery document.

This mapping is also used to ensure proper conversion of customer's Alias Application Services and to remove *orphaned* new Application Services.

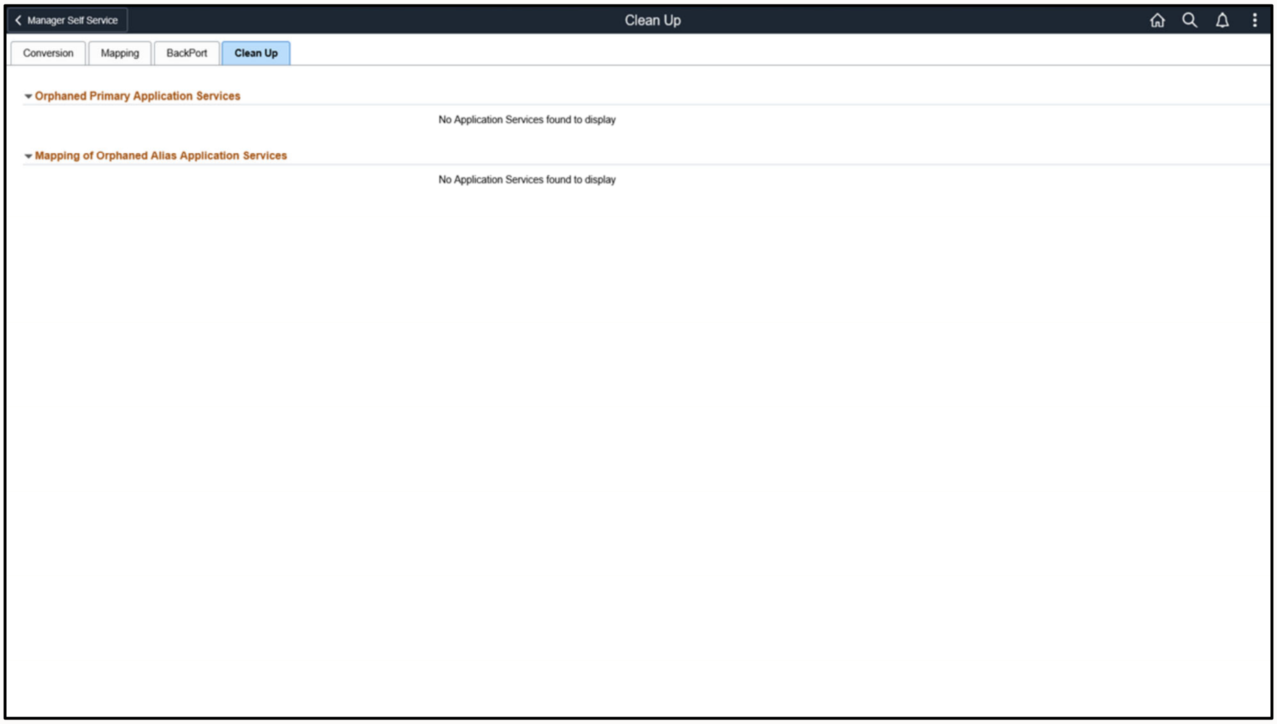


Backport

75 ro

Select	App Service ID	Root Resource	Index	REST Method	App Service ID (old)	ID for service URL (old)
N	BEN_CHATBOT_SVC	viewbenefitsinfo	1	GET	BEN_SVC_SYS	benefit.SystemInfo
N	BEN_CHATBOT_SVC	viewbenefitsinfo	2	GET	BEN_SVC_CONTACT	benefit.AdminContact
N	BEN_CHATBOT_SVC	viewplanlist	1	GET	BEN_SVC_PLAN_CATE	benefit.PlanCategory
N	BEN_CHATBOT_SVC	viewplanlist	2	GET	BEN_SVC_PLAN_TYPE	benefit.PlanType
N	BEN_CHATBOT_SVC	viewplandetail	1	GET	BEN_SVC_DEP_BEN	benefit.RegDepBen
N	BEN_CHATBOT_SVC	viewplandetail	2	GET	BEN_SVC_ENROLLED_DEPBEN	benefit.EnrolledDepBen
N	BEN_CHATBOT_SVC	viewplandetail	3	GET	BEN_SVC_PLAN_DTL	benefit.PlanDetail
N	BEN_CHATBOT_SVC	viewpages	1	GET	BEN_SVC_PG_OPENENROLL	benefit.PageOpenEnrollment
N	BEN_CHATBOT_SVC	viewpages	2	GET	BEN_SVC_PG_LIFEEVENT	benefit.PageLifeEvent
N	BEN_CHATBOT_SVC	viewpages	3	GET	BEN_SVC_PG_SUMMARY	benefit.PageBenSummary
N	EOCB_CORE	messagecatalog	1	GET	EOCB_GETMSGCAT_BY_SET	ps.GetMessageBySet
N	EOCB_CORE	messagecatalog	1	POST	EOCB_GETMSGCAT_BY_NBR	ps.GetMessageByNumbers
N	EOCB_CORE	serviceaccess	1	GET	EOCB_VERIFY_ACCESS_SVCURLID	ps.VerifyAccessForComponent
N	EOCB_CORE	serviceaccess	2	GET		ps.VerifyAccessForServi
N	EOCB_CORE	user	1	GET	EOCB_GET_USER_PROFILE	ps.GetUserProfile
N	EOCB_GUID_TO_PSTOKEN	psauthenticateguid	1	POST		

The BackPort page is used for Application Services that need to be backported to an earlier PeopleTools release (8.58 or 8.57).



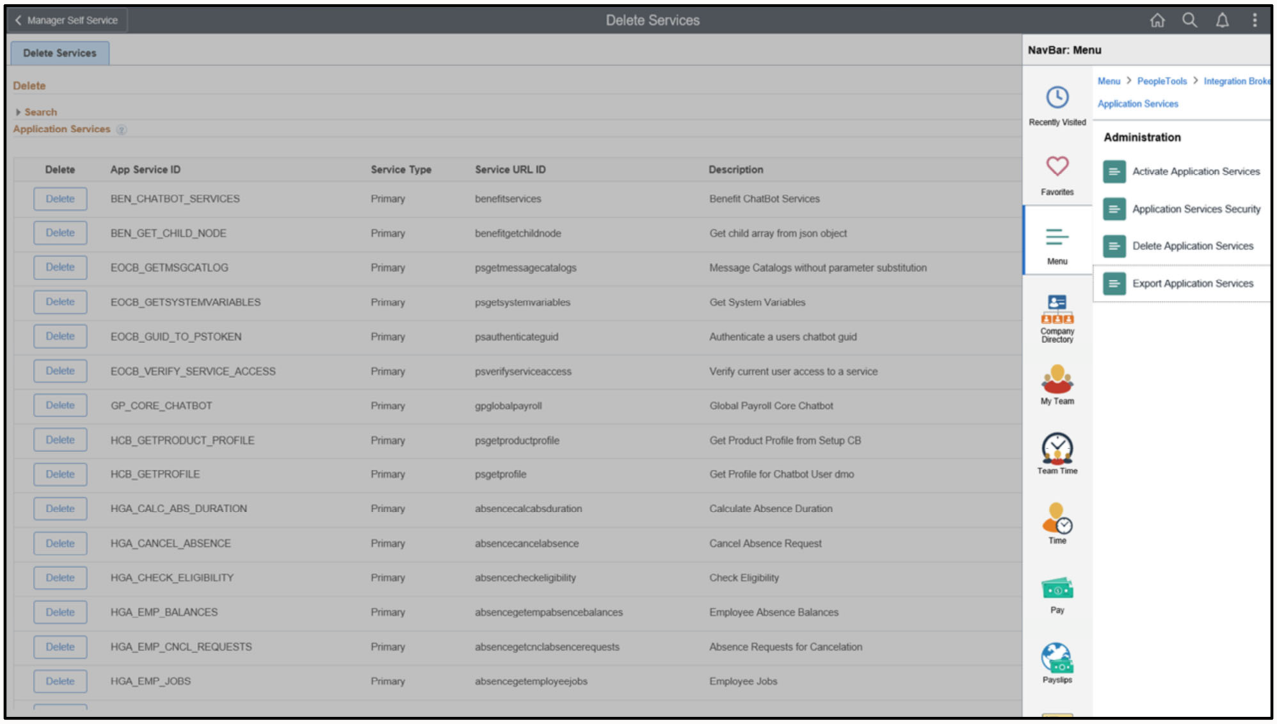
Orphaned Primary Application Services

No Application Services found to display

Mapping of Orphaned Alias Application Services

No Application Services found to display

The clean-up page will list any orphaned Primary Application Services and Mapping of Orphaned Alias Application Services.



Delete

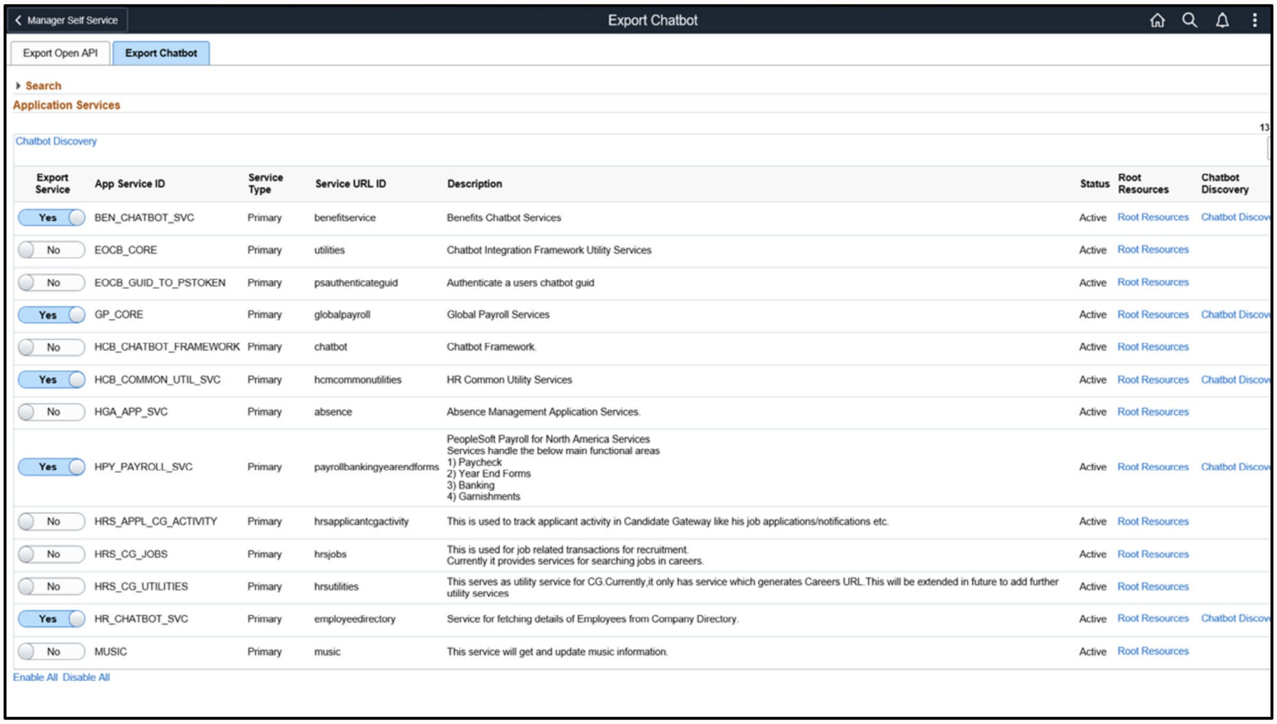
Search

Application Services

Delete	App Service ID	Service Type	Service URL ID	Description
Delete	BEN_CHATBOT_SERVICES	Primary	benefitservices	Benefit ChatBot Services
Delete	BEN_GET_CHILD_NODE	Primary	benefitgetchildnode	Get child array from json object
Delete	EOCB_GETMSGCATLOG	Primary	psgetmessagecatalogs	Message Catalogs without parameter substitution
Delete	EOCB_GETSYSTEMVARIABLES	Primary	psgetsystemvariables	Get System Variables
Delete	EOCB_GUID_TO_PSTOKEN	Primary	psauthenticateguid	Authenticate a users chatbot guid
Delete	EOCB_VERIFY_SERVICE_ACCESS	Primary	psverifyserviceaccess	Verify current user access to a service
Delete	GP_CORE_CHATBOT	Primary	gpglobalpayroll	Global Payroll Core Chatbot
Delete	HCB_GETPRODUCT_PROFILE	Primary	psgetproductprofile	Get Product Profile from Setup CB
Delete	HCB_GETPROFILE	Primary	psgetprofile	Get Profile for Chatbot User dmo
Delete	HGA_CALC_ABS_DURATION	Primary	absencecalcabsduration	Calculate Absence Duration
Delete	HGA_CANCEL_ABSENCE	Primary	absencecancelabsence	Cancel Absence Request
Delete	HGA_CHECK_ELIGIBILITY	Primary	absencecheckeligibility	Check Eligibility
Delete	HGA_EMP_BALANCES	Primary	absencegetempabsencebalances	Employee Absence Balances
Delete	HGA_EMP_CNCL_REQUESTS	Primary	absencegetcnclabsencerequests	Absence Requests for Cancelation
Delete	HGA_EMP_JOBS	Primary	absencegetemployeejobs	Employee Jobs

In order for Oracle Digital Assistant (ODA) to make call backs into PeopleSoft, the skill must be created in ODA.

To access the Export Chatbot page from the menu, select PeopleTools, Integration Broker, Administration, Export Application Service.



Application Services

Chatbot Discovery

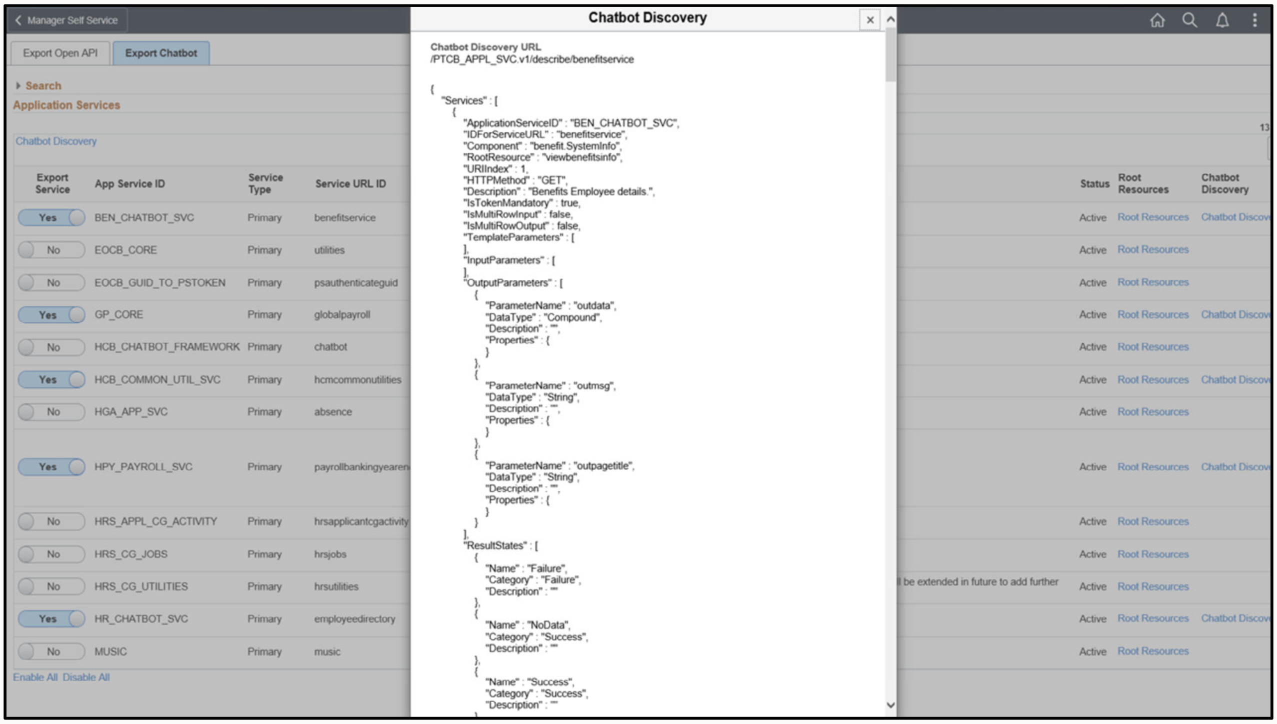
13

Export Service	App Service ID	Service Type	Service URL ID	Description	Status	Root Resources	Chatbot Discovery
Yes	BEN_CHATBOT_SVC	Primary	benefitservice	Benefits Chatbot Services	Active	Root Resources	Chatbot Discov
No	EOCB_CORE	Primary	utilities	Chatbot Integration Framework Utility Services	Active	Root Resources	
No	EOCB_GUID_TO_PSTOKEN	Primary	psauthenticateguid	Authenticate a users chatbot guid	Active	Root Resources	
Yes	GP_CORE	Primary	globalpayroll	Global Payroll Services	Active	Root Resources	Chatbot Discov
No	HCB_CHATBOT_FRAMEWORK	Primary	chatbot	Chatbot Framework.	Active	Root Resources	
Yes	HCB_COMMON_UTIL_SVC	Primary	hcmcommonutilities	HR Common Utility Services	Active	Root Resources	Chatbot Discov
No	HGA_APP_SVC	Primary	absence	Absence Management Application Services.	Active	Root Resources	
Yes	HPY_PAYROLL_SVC	Primary	payrollbankingyearendforms	PeopleSoft Payroll for North America Services Services handle the below main functional areas 1) Paycheck 2) Year End Forms 3) Banking 4) Garnishments	Active	Root Resources	Chatbot Discov
No	HRS_APPL_CG_ACTIVITY	Primary	hrsapplicantcgactivity	This is used to track applicant activity in Candidate Gateway like his job applications/notifications etc.	Active	Root Resources	
No	HRS_CG_JOBS	Primary	hrsjobs	This is used for job related transactions for recruitment. Currently it provides services for searching jobs in careers.	Active	Root Resources	
No	HRS_CG_UTILITIES	Primary	hrsutilities	This serves as utility service for CG.Currently,it only has service which generates Careers URL.This will be extended in future to add further utility services	Active	Root Resources	
Yes	HR_CHATBOT_SVC	Primary	employeedirectory	Service for fetching details of Employees from Company Directory.	Active	Root Resources	Chatbot Discov
No	MUSIC	Primary	music	This service will get and update music information.	Active	Root Resources	

Enable All Disable All

Select the Export Chatbot page to generate custom components for ODA usage from the Application Service metadata.

When you select YES in the Export Service column for the Application service, a Chatbot Discovery document is created.



Export Service	App Service ID	Service Type	Service URL ID		Status	Root Resources	Chatbot Discovery
Yes	BEN_CHATBOT_SVC	Primary	benefitservice		Active	Root Resources	Chatbot Discov
No	EOCB_CORE	Primary	utilities		Active	Root Resources	
No	EOCB_GUID_TO_PSTOKEN	Primary	psauthenticateguid		Active	Root Resources	
Yes	GP_CORE	Primary	globalpayroll		Active	Root Resources	Chatbot Discov
No	HCB_CHATBOT_FRAMEWORK	Primary	chatbot		Active	Root Resources	
Yes	HCB_COMMON_UTIL_SVC	Primary	hcmcommonutilities		Active	Root Resources	Chatbot Discov
No	HGA_APP_SVC	Primary	absence		Active	Root Resources	
Yes	HPY_PAYROLL_SVC	Primary	payrollbankingyearen		Active	Root Resources	Chatbot Discov
No	HRS_APPL_CG_ACTIVITY	Primary	hrsapplicantcgactivity		Active	Root Resources	
No	HRS_CG_JOBS	Primary	hrsjobs		Active	Root Resources	
No	HRS_CG_UTILITIES	Primary	hrsutilities	ll be extended in future to add further	Active	Root Resources	
Yes	HR_CHATBOT_SVC	Primary	employeedirectory		Active	Root Resources	Chatbot Discov
No	MUSIC	Primary	music		Active	Root Resources	

Enable All Disable All

Chatbot Discovery

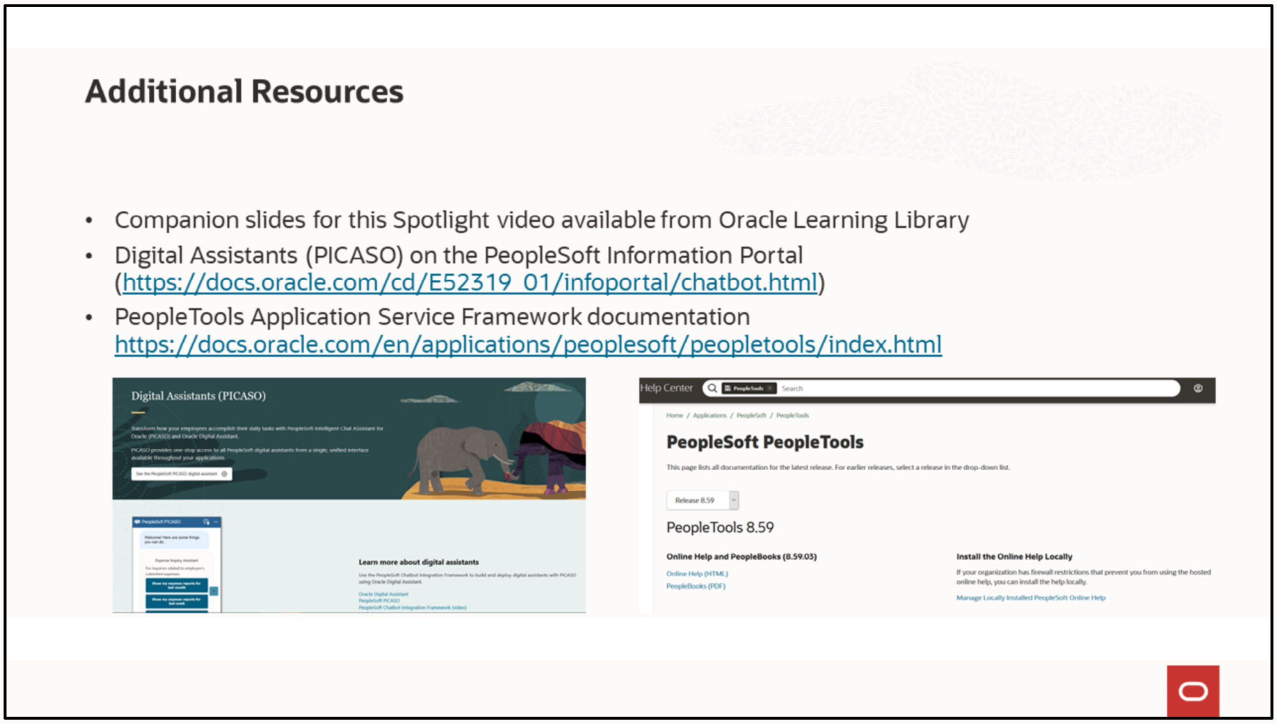
Chatbot Discovery URL
/PTCB_APPL_SVC.v1/describe/benefitservice

```
{
   "Services" : [
      {
         "ApplicationServiceID" : "BEN_CHATBOT_SVC",
         "IDForServiceURL" : "benefitservice",
         "Component" : "benefit.SystemInfo",
         "RootResource" : "viewbenefitsinfo",
         "URIIndex" : 1,
         "HTTPMethod" : "GET",
         "Description" : "Benefits Employee details.",
         "IsTokenMandatory" : true,
         "IsMultiRowInput" : false,
         "IsMultiRowOutput" : false,
         "TemplateParameters" : [
         ],
         "InputParameters" : [
         ],
         "OutputParameters" : [
            {
               "ParameterName" : "outdata",
               "DataType" : "Compound",
               "Description" : "",
               "Properties" : {
               }
            },
            {
               "ParameterName" : "outmsg",
               "DataType" : "String",
               "Description" : "",
               "Properties" : {
               }
            },
            {
               "ParameterName" : "outpagetitle",
               "DataType" : "String",
               "Description" : "",
               "Properties" : {
               }
            }
         ],
         "ResultStates" : [
            {
               "Name" : "Failure",
               "Category" : "Failure",
               "Description" : ""
            },
            {
               "Name" : "NoData",
               "Category" : "Success",
               "Description" : ""
            },
            {
               "Name" : "Success",
               "Category" : "Success",
               "Description" : ""
```

Click the link to view it.

Additional Resources

For more information on Application Services Framework



Additional Resources

- Companion slides for this Spotlight video available from Oracle Learning Library
- Digital Assistants (PICASO) on the PeopleSoft Information Portal (https://docs.oracle.com/cd/E52319_01/infoportal/chatbot.html)
- PeopleTools Application Service Framework documentation https://docs.oracle.com/en/applications/peoplesoft/peopletools/index.html

Digital Assistants (PICASO)

Transform how your employees accomplish their daily tasks with PeopleSoft Intelligent Chat Assistant for Oracle (PICASO) and Oracle Digital Assistant.

PICASO provides one-stop access to all PeopleSoft digital assistants from a single, unified interface available throughout your applications.

See the PeopleSoft PICASO digital assistant

PeopleSoft PICASO

Welcome! Here are some things you can do

Expense Inquiry Assistant

For inquiries related to employee's submitted expenses

Show my expense reports for last month

Show my expense reports for last week

Learn more about digital assistants

Use the PeopleSoft Chatbot Integration Framework to build and deploy digital assistants with PICASO using Oracle Digital Assistant.

Oracle Digital Assistant
PeopleSoft PICASO
PeopleSoft Chatbot Integration Framework (video)

Help Center | PeopleTools | Search

Home / Applications / PeopleSoft / PeopleTools

PeopleSoft PeopleTools

This page lists all documentation for the latest release. For earlier releases, select a release in the drop-down list.

Release 8.59

PeopleTools 8.59

Online Help and PeopleBooks (8.59.03)

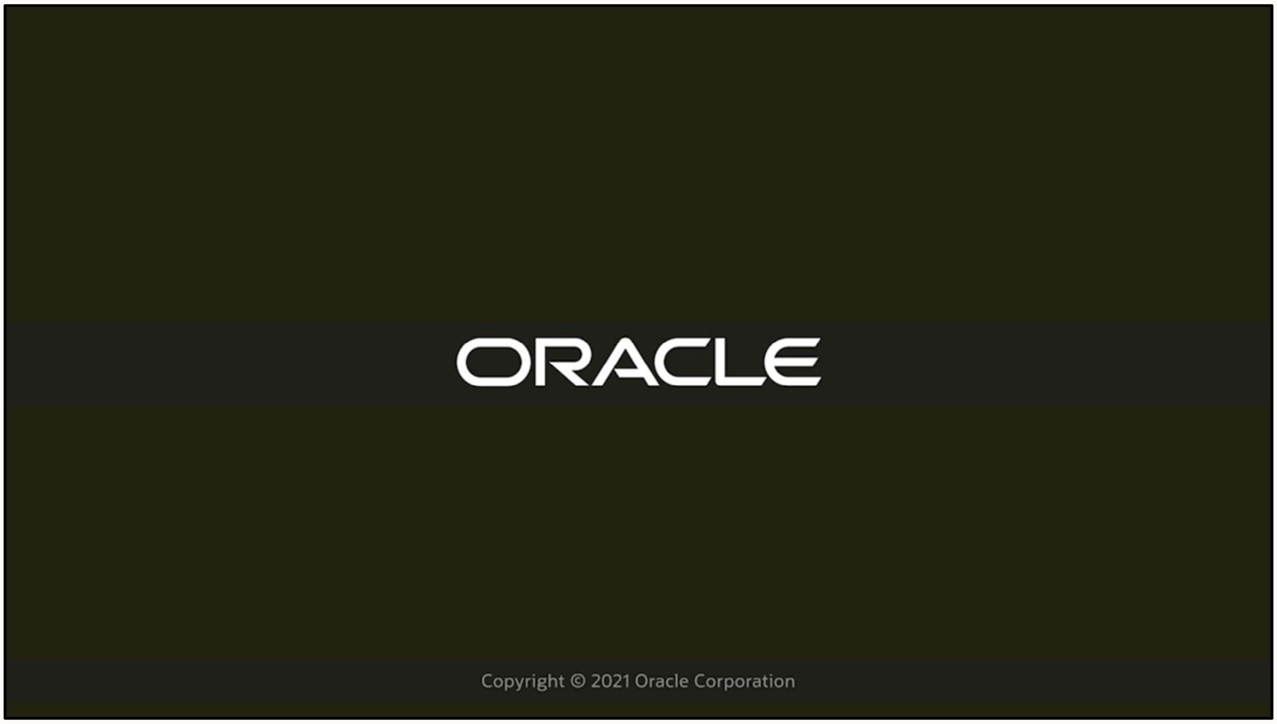
Online Help (HTML)
PeopleBooks (PDF)

Install the Online Help Locally

If your organization has firewall restrictions that prevent you from using the hosted online help, you can install the help locally.

Manage Locally Installed PeopleSoft Online Help

You can download the Companion Slides, with images and notes from this video, from the Oracle Learning Library.
From the PeopleSoft Information Portal, you can find information about PeopleSoft PICASO, digital assistants, and skills.
And refer to the PeopleTools Application Service Framework documentation from the PeopleTools Help Center.



ORACLE