Using a Web Service as External Service
## Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Represents</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example Text</strong></td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><strong>&lt;Example text&gt;</strong></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>

## Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>🚨</td>
<td>Caution</td>
</tr>
<tr>
<td>📑</td>
<td>Example</td>
</tr>
<tr>
<td>📚</td>
<td>Note</td>
</tr>
<tr>
<td>🥀</td>
<td>Recommendation</td>
</tr>
<tr>
<td>☀️</td>
<td>Syntax</td>
</tr>
</tbody>
</table>
Contents
Introduction ............................................................................................................................. 2
Requirements and Dependencies .......................................................................................... 2
   Applicable Releases ........................................................................................................... 2
   Disclaimer ......................................................................................................................... 2
Create the Development Component Project ...................................................................... 2
Import of Web Services as external service ......................................................................... 8
Create Entity Service as Web Service wrapper ................................................................ 12
Configure external service (Runtime Configuration) .......................................................... 20
Test external service ........................................................................................................... 24
Introduction
The following tutorial shows you how to use external functionality (Web Service) in your CAF application. A Web Service will be imported as an external service and then wrapped by an entity service.

Requirements and Dependencies
Before you start with this tutorial you should have installed the following Software:

- SAP Web Application Server Java 7.0
- SAP NetWeaver Developer Studio 7.0

This tutorial is based on the following How-to Guides:

- Create Entity Service with maintenance UI

Additionally you should have configured CAF as described in the How-to Guide “Installation and Configuration Guide” and you should have deployed and initialized the EDM web service. (Under https://bis.wdf.sap.corp/twiki/bin/view/Sapinternal/NetWeaverEducationDataModel download the file “IntroductionToTheEducationDataModel.doc”. Follow the instructions in paragraph 4 to install and initialize the EDM web service. Follow the instructions in paragraph 5 to test the web service.)

Applicable Releases
This tutorial is compatible with the following releases:

- SAP NetWeaver ’04s
- SAP Composite Application Framework (CAF) 7.0

Disclaimer
Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Create the Development Component Project
1) Open the NetWeaver Development Studio. Switch to the CAF Perspective by selecting “Composite Application Services” as perspective from the menu Window -> Open Perspective -> Other...
   An icon will appear in the left pane.
2) Create a new Development Component Project by navigating File -> New -> Project. The New Project window will open.

3) Select Development Component -> Development Components Project and click Next.
4) Select Local Development -> MyComponent and click Next.

![Image of Local Development and MyComponent selection](image)

This tutorial does not discuss development using NetWeaver JDI. If you want to use the JDI, please refer the [http://help.sap.com](http://help.sap.com) for developing components using remote configurations.

5) In the next screen, Enter/Select the following.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>user_example (use only small letters)</td>
</tr>
<tr>
<td>Caption</td>
<td>User Example Application</td>
</tr>
<tr>
<td>Domain</td>
<td>SAP xApps</td>
</tr>
<tr>
<td>Type</td>
<td>Composite Application Services</td>
</tr>
</tbody>
</table>
Click on *Next* twice.

A screen will display the list of projects that will be created as a part of this Development Component Project.
Click Finish.

The creation of the CAF project is completed.
Import of Web Services as external service

1) Right-click on the **External Services** node in the **Service Explorer** and select **Import**.

2) Select **Web Service** and click **Next** button.
3) Select Local File System or URL as the backend system from which to import the Web Service metadata. Click Next.

5) Now you can find the external service created in the Service Explorer and open it with a double-click.
6) Switch to the *Operations* tab and select the function *getAllEmployees* for example. On its right you can enter a description of the function. Below you find a list of all input and output parameters.
7) Save the Metadata.

**Create Entity Service as Web Service wrapper**

To use this external service, we have to wrap it with an entity service. How to do this is described in the following pages.
1) Right-click on the **Entity Services** node and select **New**.
2) In the popup, enter the entity service’s name (e.g. User) and select Finish.

3) Now we have to create the necessary attributes for the entity service. To do this, switch to the Attributes tab, right-click on the User node and select Create Attribute.
4) In the popup enter the following data and click Finish.

5) Repeat steps 3 and 4 for the following attributes.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Description</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>pers_no</td>
<td>com.sap.caf.core.long</td>
<td>Employee number</td>
<td>Yes</td>
</tr>
<tr>
<td>firstName</td>
<td>com.sap.caf.core.longText</td>
<td>First Name</td>
<td></td>
</tr>
<tr>
<td>lastName</td>
<td>com.sap.caf.core.longText</td>
<td>Last Name</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>com.sap.caf.core.shortText</td>
<td>User Id</td>
<td></td>
</tr>
</tbody>
</table>
6) Switch to the *Operations* tab and click on the *Add…* button on the right.
7) In the popup enter the following data.

![New Operation Window](image1)

- **Operation name**: getUserDetailById
- **Description**: Gets all user details by its Id

![FindBy Attribute listing](image2)

8) Switch to the **Persistency** tab and uncheck **Local Persistency**. This is necessary to be able to map this operation to external services.
9) Switch to the **Datasource** tab and click on the previously created operation. On the right side, click on ![image] to map the operation to an external service.

<table>
<thead>
<tr>
<th>Entity Service Operation</th>
<th>External Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td></td>
</tr>
<tr>
<td>update</td>
<td></td>
</tr>
<tr>
<td>delete</td>
<td></td>
</tr>
<tr>
<td>getUserDataById</td>
<td></td>
</tr>
</tbody>
</table>

10) In the popup select the external service and click **OK**.

![External Service]

11) Now you can define the parameter mapping using Drag&Drop. The following mappings have to be carried out:

<table>
<thead>
<tr>
<th>Entity Service Parameter</th>
<th>External Operation Parameter</th>
</tr>
</thead>
</table>
12) Switch to the Permissions tab and uncheck Permission checks enabled.

13) Save the metadata, build the project and deploy it.
Configure external service (Runtime Configuration)

2) Now click on the *External Service Configuration* link.

3) Click on the *Service Registry* link on the left. In the service registry you can define and maintain the backend connections for your external services.

4) Now select the external service we have created and click the *Open Definition* button.
5) Now you should see the following screen. Here, select the root node of the external operation (the web service itself).

And at the bottom you should find:

6) Now select the function node of the external operation.

at the bottom you should find:

Test external service

External services are tested using the ‘wrapping’ entity services. This means they are tested like all other entity services.

Nevertheless, in the next steps you can find a description of how to test it.

   Click on Test tools.

2) Select the link Service Browser.
3) In the Service Browser navigate to the `getUserDetailById` operation.

4) Enter a valid user id (for example “25”) and click on `Execute query`. The result should look like this.