How to Create a HTML Mashup in SAP Cloud for Customer to Consume a Google Map API
# Content

1 Introduction.................................................................................................................4
1.1 Business Scenario...................................................................................................4
1.2 Prerequisites............................................................................................................4
2 Create PDI Admin/Development User ..................................................................5
3 Create Solution..........................................................................................................8
4 Test the external API..................................................................................................20
1 Introduction

1.1 Business Scenario

External APIs are in most of the cases an important part of a Customers business, being used for lots of reasons. This document shows how to consume the Google Maps API in Cloud for Customer, showing the map of an address retrieved from the system inside an Individual Customer.

1.2 Prerequisites

**SAP Cloud for Customer**
Initial setup and configuration was already performed in tenant as it is mentioned in the SAP Cloud for Customer Administrator Guide

**SAP Cloud Application Studio**
<latest version and should match with the version of SAP Cloud for Customer tenant>

**SDK Know-how**
Basic SDK knowledge recommended.

**Internet Explorer browser**
For test purposes. The Mashup only works in the Internet Explorer browser. Version 9 or higher + is required.
Create PDI Admin/Development User

In order to develop a solution for Cloud for Customer you must have an user with permission to PDI Partner Development. Follow these steps to obtain a PDI user:

1. Connect to the SAP Cloud for Customer system using an Internet browser, open the Business Configuration tab and select Implementation Projects.

2. For the First Implementation Project, choose Open Activity List.

3. Under Prepare tab, choose Define Administrators for Project Team.

4. Choose Create Service Agents link.
5. Enter the First Name and Last Name and choose Save.


7. Close the Opened pages.

You must define administrators.


9. Choose Go to see all the Business Users.

10. Select the user created before and choose Edit → Attributes.
11. Maintain the User ID, password values and choose Save.


13. Under the tab, Work Center and View Assignment, make the below assignment by selecting the check box. Choose Save.
3  Create Solution

The next step is creating a PDI solution that will consume the external API. To do so, follow these steps:

1. Log into the system tenant with the username and password created in step 2. For this access Administration → Log On.

2. Access Administration → Create Solution and create a new PDI solution. Enter a description for your solution and click ok.
3. Extend the Customer BO.

4. In the extended BO, create an element in the Common node to store the address value. Save and activate the XBO.

```java
import AP.COMMON.SDT;
import AP.FO.BusinessPartner.Global;

    node AddressInformation {
    }

    node Common {
        [Label("Address")] element FE_address : LANGUAGEINDEPENDENT_EXTENDED_Text;
    }

    node CurrentEmployeeResponsible {
    }
}
```

5. Create script to copy the value from Individual Customer to the created extension field.
   - Right click in the XBO and select “Create Script Files”.
• Select Root node’s event BeforeSave and click OK.

- Insert the following code to copy the value from the Individual Customer to the extension field, when the extension field is empty. If the field already has a value, it is not changed.

```javascript
import ABSL;

var formattedAddress = this.CurrentDefaultAddressInformation.Address.FormattedAddress.GetFirst();
if (formattedAddress.IsSet()) {
    var formattedPostalAddress = formattedAddress.FormattedPostalAddress;
    var common = this.Common.GetFirst();
    if (common.IsSet() && common.FE_address.IsInitial()) {
        common.FE_address = formattedPostalAddress.FirstLineDescription + ", " + formattedPostalAddress.SecondLineDescription + ", " + formattedPostalAddress.ThirdLineDescription;
    }
}
```

6. Enhance the screens of the XBO. The screen where the map will appear is called COD_SEOD_IND_ACCOUNT_TI.TI.xuicomponent.
   - Right click the XBO and select “Enhance Screen”

- Select the COD_SEOD_IND_ACCOUNT_TI screen and press ok.
7. Create a new HTML mashup and name it AddressMashup.

- The admin mode must be enabled to create this item. In case the admin is not already enabled, click “Yes” in the pop-up that appears.

- A system window containing the HTML mashup configuration fields appears. Configure the fields as follows:
  - Mashup Category: Productivity & Tools
  - Port Binding: URL Navigation
  - Mashup Name: <your mashup name>
  - Mashup Description: <any or none mashup description>
  - Status: Active
  - Type: URL
  - URL: https://maps.google.com/

  Additionally, two parameters must be added to the URL. Click the “Add” button and add the following information:
  - Parameter 1:
    - Name: output
    - Constant: embed
    - Mandatory: yes
  - Parameter 2:
    - Name: q
    - Parameter Binding: URL
    - Mandatory: yes
Click the “Preview” button and the Google Map must appear:

AddressMashup

Save and Close

At the end, the screen must be exactly like this:

NEW HTML MASHUP

General Information

Mashup Category: Productivity & Tools
Port Binding: URL Navigation
Mashup Name: AddressMashup
Description:
Status: Active Inactive

Configuration Information

Type: HTML Code URL
URL: https://maps.google.com/
Height: 327

<table>
<thead>
<tr>
<th>Name</th>
<th>Constant</th>
<th>Parameter Binding</th>
<th>Mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>output</td>
<td>embed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>q</td>
<td>URL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Save and Close the configuration screen.
8. Create a new Port Type Package (PTP).

Select Customer as the business object for the PTP:

Open the created PTP and configure the port types as follows:

- Enable the edit mode in UI Designer.
- Select the root node from PTP and click on the plus sign to add a new Port Type.
- Right click on the created Port Type and rename it to Google_Maps_URL.
- Select the created Port Type and click on the plus sign to add a new Parameter.
- Right click on the Parameter and rename it to URL.
- Right click again on the Parameter and select Set as Key.

Save and activate the PTP. The Port Type Package must look like this:
9. Create an Embedded Component to display the address field and the Google Map.

- Open the Embedded Component and enable edit mode.
- Open the DataModel tab:
  - Right-Click in “Root” and select “Add data structure”. Rename it to “Customer”.
  - Add the address field to “Customer” structure:
    2. BOModel: Customer
    3. Open the Common node and select the extension field created in step 4.
    4. Drag and drop the address field in the Customer structure.
How to consume an external API in C4C

Right-click in “Root” and select “Add data structure”. Rename it to “Inport”.

Right-click in “Inport” and select “Add data field”. Rename it to “AccountUUIDIn”.

Add the address field to the EC:
Open View → BO Browser/Data Model and switch to “Data Model” tab:
Select the extension field added previously to the Customer structure.

Drag and drop the address field in the EC:
- On the top bar, click on the Add FlexLayoutRow button.

- Add the Google Maps mashup to the EC:
  - Open View ➔ Configuration Explorer.
  - Navigate to the following path:
    - `<your solution>_MAIN ➔ SRC ➔ Mashups ➔ Pipes`
  - Drag and drop the element to the EC.

- Adjust the EC labels:
  - Open View ➔ Property Explorer.
  - Select the section group, and change the Field Group Header. Change it to “Enter address to display on map”.

- Save and activate the EC.

10. Configure the EC controller.

10.1. Configure an Outport for the EC:
  - Open the Embedded Component in UI Designer and select the Controller tab.
  - Right-click in Outports folder and select “Add Outport”.

  The following configuration needs to be done:
  - PortType Package: `<your_solution>_MAIN ➔ SRC ➔ PTP_Address (created on step 8)
  - PortType Reference: Google_Maps_URL
  - The URL parameter must already appear.
  - Add a new parameter with the following configuration:
    - Parameter Name: Address
    - Parameter Binding: `<the extension field created in the previous step, inside Customer structure`
    - Select the Key checkbox

The Outport must look like this:

10.2. Configure an Inport:
• In the EC Controller tab, right-click on Inports folder and select “Add Inport”.

![Add Inport](image)

• Click in the Add Parameter button, and add the following parameter:
  - Parameter Name: AccountUUID
  - Parameter Binding: <select the AccountUUIDIn created in the previous step>

The inport should look like this:

![Import Configuration: AccountUUIDIn](image)

• Open View → Property Explorer.
• In the Events → OnFire, select “New Event Handler”.
• Name it “ReadAccount” and click on “Add Operation” button. Configure as follows:
  - Type: BOOperation
  - Name: Operation
  - BOOperation Type: Read
  - BOModel: Customer
  - Click on the “Add Parameter” button and configure it:
    - Name: Parameter
    - Type: KeyNavigation
    - Bind: <select the AccountUUIDIn created before>
    - BindList: <leave it empty>
    - Path: <select the Customer structure created before>

The Event Handler should look like this:

![ReadAccount: Operations](image)

10.3. Configure the Mashup Component:
• In the EC Controller tab, expand Mashup Components and select the Mashup added before to the EC.
• In navigations tab, right-click in navigations and select “Add Navigation”.

• Configure the Navigation as follows:
  ▪ OutPort: <the OutPort created in step 10.1>
  ▪ InPort: URL_Navigation_Info_In
  ▪ Bind the OutPort Parameter “Address” to the InPort “URL”

The Navigation must look like this:

11. Add the EC to the Individual Customer TI.
   • Open COD_SEOD_IND_ACCOUNT_TI in UI Designer.
   • Open View → Extensibility Explorer.
   • Select the first anchor “Undefined” and click on “Add View with Embedded Component”.

© 2014 SAP SE. All rights reserved.
- Set the tab title as "MAP LOCATION", and select the created embedded component under the path `<your_solution>_MAIN → SRC → EC_Map_Address`

A new Tab will be created. Please provide a title for the new Tab.

**Tab Title:** MAP LOCATION

**Embedded Component:** `YABONB14Y_MAIN/SRC/EC_Map_Address.EComponent`

- Click on "Bind" and configure as follows:
  - Outport: `PublicOutportECIndCustomerRoot`
  - Inport: `AccountUUIDIn`
  - In Outport Parameters select `AccountUUID`, and select `AccountUUID` in Inport Parameters as well. After that, click on Bind.
  - Click Ok.

- Apply, save and activate the TI.
4 Test the external API

1. Access Cloud for Customer system.
   * Please note that the Mashup only works in the Internet Explorer browser.

   SAP Cloud for Customer

   ![Login page](image)

2. Go to Individual Customers, located in Customers → Individual Customers.

3. Access a customer and go to the created tab “Map Location”.

   ![Customer page](image)

4. Enter an address in the field and wait while the location is shown in the Google Maps API.