VBCS Training Lab- Budapest Integrating an External REST API

Introduction

In this hands-on-lab, we will create an extension to Oracle Sales Cloud that consumes the custom TwitterHandle property on the Contact object and displays a list of tweets for the contact.

# Hands on Lab Instructions

Creating a Web Application

1. Go to https://oic-vbdemo.uscom-east-1.oraclecloud.com/ic/builder/
2. Login in as angelo.santagata@oracle.com / ILoveKangaroo4242#
3. Click New button to create a new application
4. Name the application BudapestLab*MyName*
5. Click Finish

Registering the Contact Object

6. In the left sidebar, click the Service Connections icon
7. Click the ⊕ icon in the Services panel
8. Click Define By Specification
9. In the Service Specification tab, enter the following information:

Field	Value
Service ID	Contact
API Type	ADF Describe
Service Specification	https://cccn-test.fa.em2.oraclecloud.com/crmRestApi/resources/11.13.17.11/contacts/describe
Authentication Mechanism	Basic
Username	angelo.santagata@oracle.com
Password	ILoveKangaroo4242#

10. Click Next
11. Expand Contacts and select the GetContactsContactsID endpoint. Click Finish.
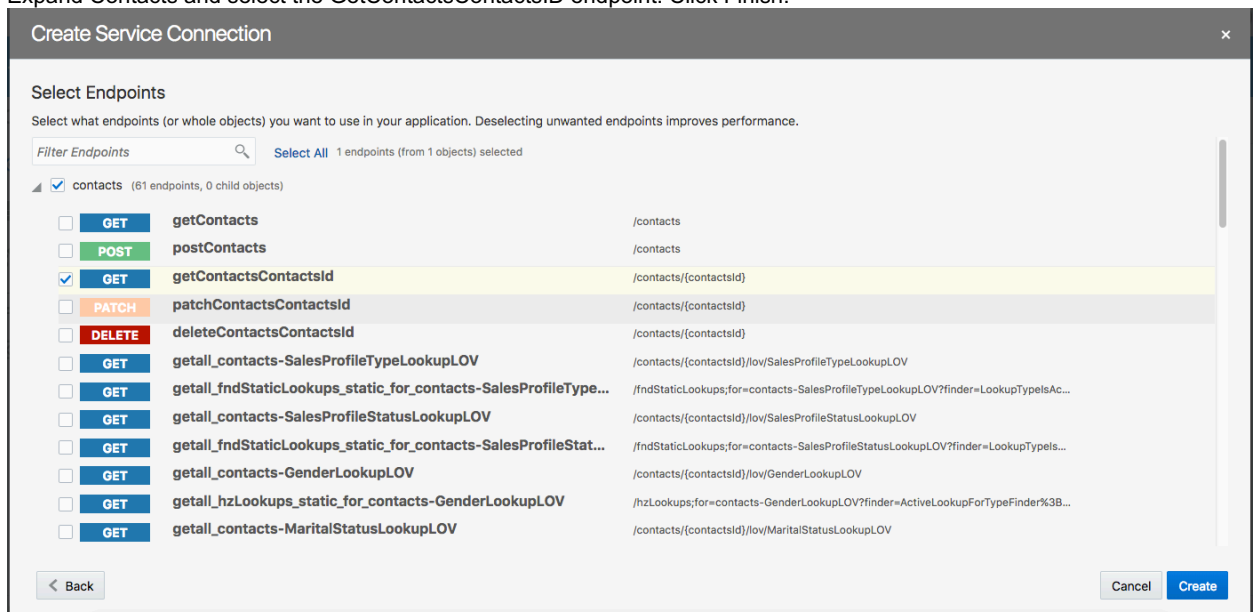
Registering the Twitter API

12. In the left sidebar, click the Service Connections icon

13. Click the icon in the Services panel

14. Click Define By Endpoint

15. In the Service tab, enter the following information:

Field	Value
Method	Get
URL	https://api.twitter.com/1.1/statuses/user_timeline.json
Action Hint	Get Many
Service Base URL	Leave default (https://api.twitter.com/1.1/statuses)
Service Name	Twitter
Service ID	Leave default (twitter)

16. In the Authentication tab, enter the following information:

Field	Value
Authentication Method	Client Credentials OAuth 2.0
Client Id	Q5GfmUalvQYVTAH0ZAfyd31HH
Secret	IB0UlQfLtxFyioOcqEHeGbFYPrHmXqxtIgzUctgfbskUaJ0ThV
Scope	Leave blank
Authorization URL	https://api.twitter.com/oauth2/token

17. Click Request > URL Parameters. Add two Dynamic Query Parameters:
   a. screen_name – String
   b. count - Number

Create Service Connection ✕

Method * ⓘ URL * Action Hint * ⓘ
GET ▼ https://api.twitter.com/1.1/statuses/user_timeline.json ▼

Service   Authentication   **Request**   Response   Test

Body   Headers   **URL Parameters**

Path Parameters ⓘ

No items to display.

Query Parameters Add ▼

Dynamic

Name	Type	Default Value	Required	
screen_name	String ▼		☐	✕
count	Number ▼		☐	✕

‹ Back Cancel Create

18. Click Test. Enter SFGiants as the screen_name and click Send. The response body appears in the Body area below. Click the Copy to Response Body button.

Create Service Connection ✕

Method * ⓘ URL * Action Hint * ⓘ
GET ▼ https://api.twitter.com/1.1/statuses/user_timeline.json Retrieve Many ▼

Service   Authentication   Request   Response   **Test**

| query | screen_name | string | SFGiants | ☐ |
| query | count | integer | | ☐ |

Reset to Defaults

Send Status: 200 OK

▲ Response

Body Headers

```
[
  {
    "created_at": "Thu Feb 15 20:53:52 +0000 2018",
    "id": 964241370813780000,
    "id_str": "964241370813779971",
    "text": "BOCHY BALL takes you behind the scenes and shows how Bruce Bochy creates the conditions - chemistry, character, cul...
https://t.co/mDnEUHeIO1",
```

Copy to Response Body

‹ Back Cancel Create

19. Click Create.

Calling the Contact REST Endpoint

Here we will create a Type that will mirror the response payload of the Contact REST Endpoint. Then we will create a variable of that type and an Action Chain to populate it from a REST call.

20. Click the Main tab in the browser. If you closed it, you can open it by going to Web Apps > webapp1 > flows > main > main.

21. Click the Variables tab, then the Types tab.

22. Add a Type from Endpoint
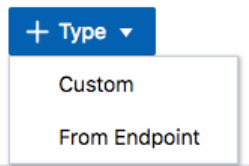
23. Select the GetContactsContactsId endpoint and click Next.
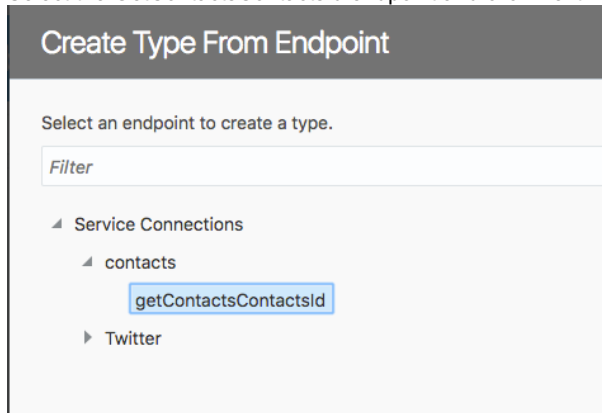
24. Select the PartyNumber, ContactName, and TwitterHandle_c fields and click Finish.

25. Open the **web app start page** by clicking the Main tab and double-clicking the start page icon.

26. Go to the Variables tab ⟨(x)⟩ and add a new variable called Contact of type getContactsContactsId.

27. Create a new String variable called ContactPartyNumber. Make it a required input parameter passed in the URL
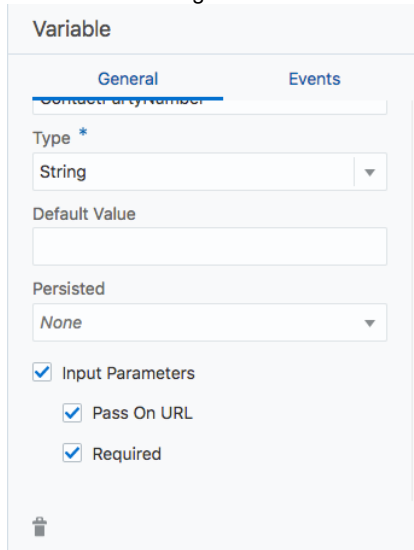

28. Give ContactPartyNumber a default value of 9. This is just to get some data in the designer. At runtime it'll get passed in.

29. Go to the Actions ⚑ tab and create a new action called populateContact.

30. Open the action chain and drag a Call REST Endpoint action into the chain. Set getContactsContactsId as the endpoint.

31. Click the -> button next to Input Parameters and map ContactPartyId to contactsId.
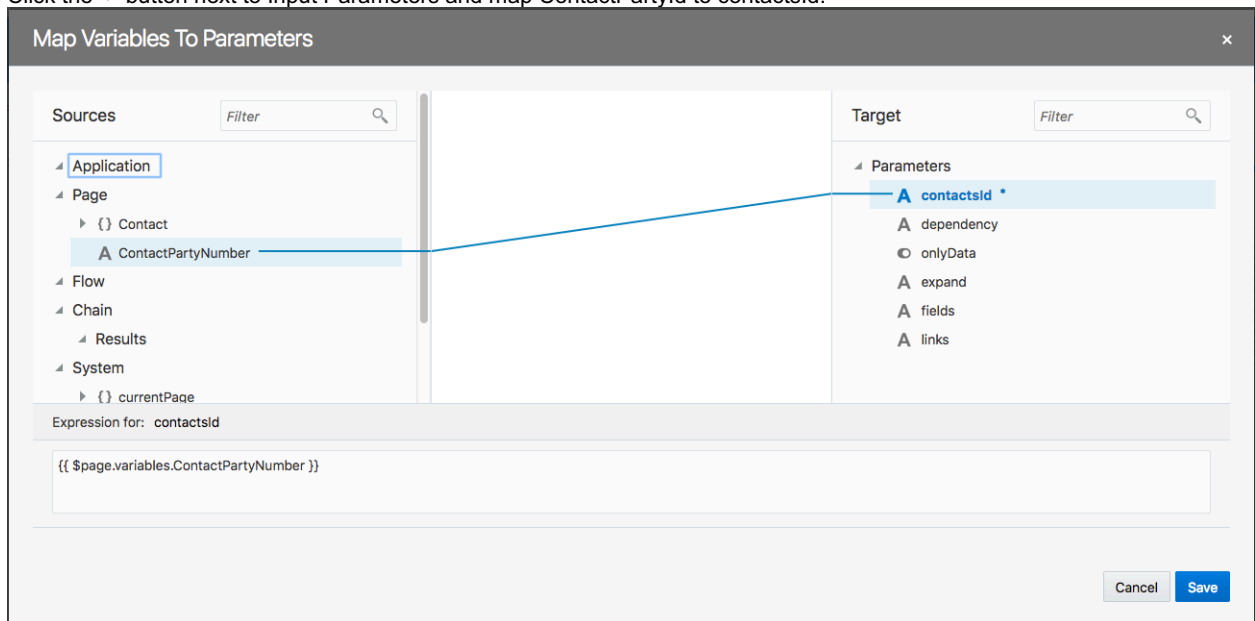
32. Drag an Assign Variables action onto the chain after the Call REST Endpoint action. Click the -> button next to Variables to open the Mapping dialog.

33. Select Chain > Results > callRestEndpoint1 > body and drag it over to Page > Contact
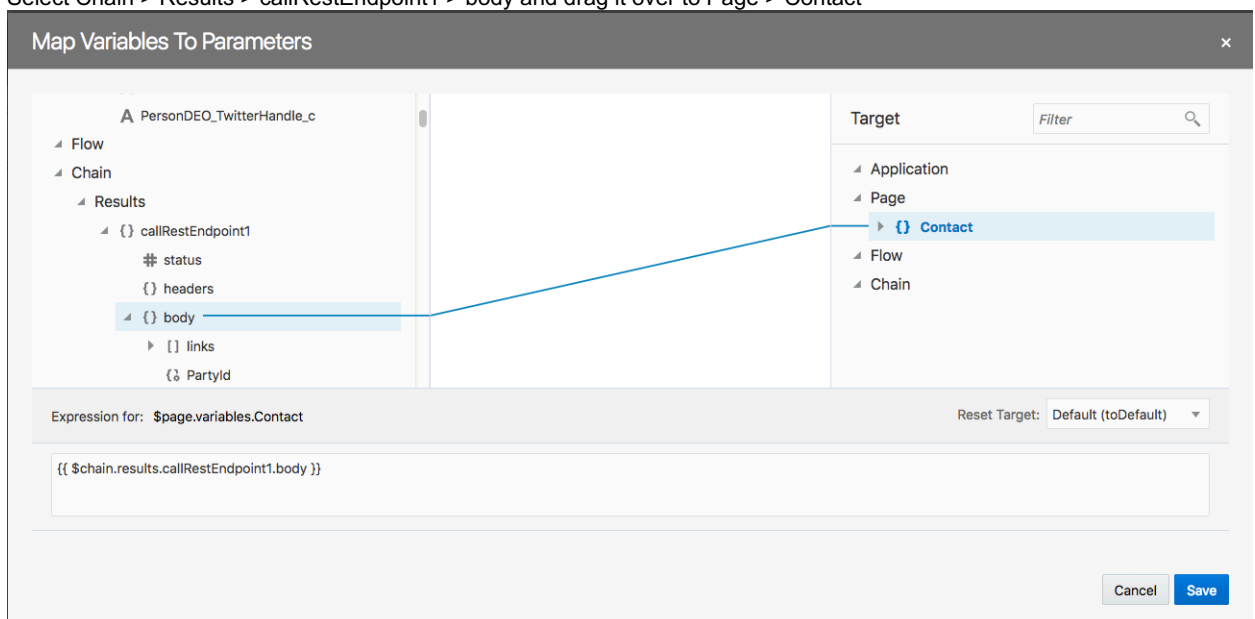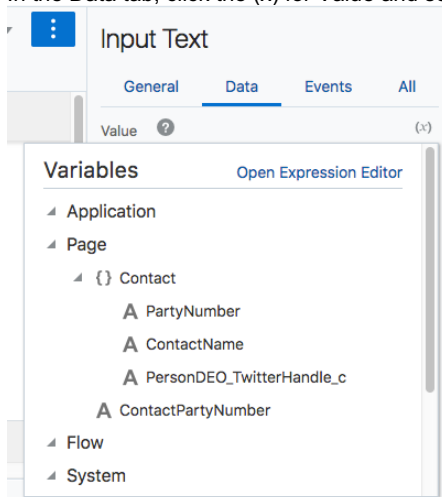

34. Click Save

35. Go to the Settings tab of the Start Page ⚙

36. In the Events panel on the left, click Add New Event. Select vbEnter and click Next. Select populateContact and click Finish.

Create the UI

37. Drop an Input Text component in the page.

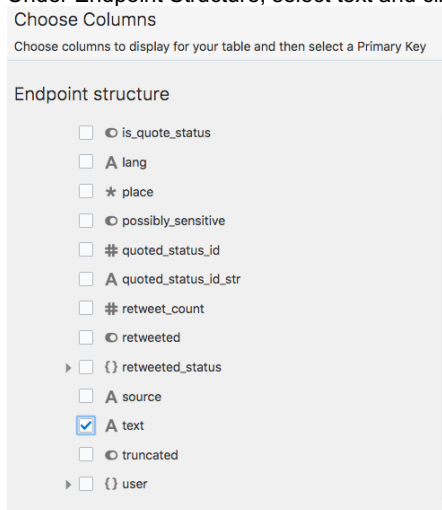38. Set the text in the label to Twitter Handle. Make it read-only.

39. In the Data tab, click the (x) for Value and select Contact > PersonDEO_TwitterHandle_c.

Input Text

General	Data	Events	All

Value ⓘ   (x)

Variables   Open Expression Editor

▲ Application
▲ Page
   ▲ {} Contact
       A PartyNumber
       A ContactName
   A PersonDEO_TwitterHandle_c
   A ContactPartyNumber
▲ Flow
▲ System

40. Drop a table under the Input Text component. In the Properties Inspector, click Add Data.

41. Select Service Connections > Twitter > getUser_timelineJson and click Next.

42. Under Endpoint Structure, select text and click Next.

Choose Columns

Choose columns to display for your table and then select a Primary Key

Endpoint structure

☐ ⊙ is_quote_status
☐ A lang
☐ ✳ place
☐ ⊙ possibly_sensitive
☐ # quoted_status_id
☐ A quoted_status_id_str
☐ # retweet_count
☐ ⊙ retweeted
▸ ☐ {} retweeted_status
☐ A source
☑ A text
☐ ⊙ truncated
▸ ☐ {} user

43. Map Page > Contact > PersonDEO_TwitterHandle_c to getUser_timelineJson > uriParameters > screen_name.
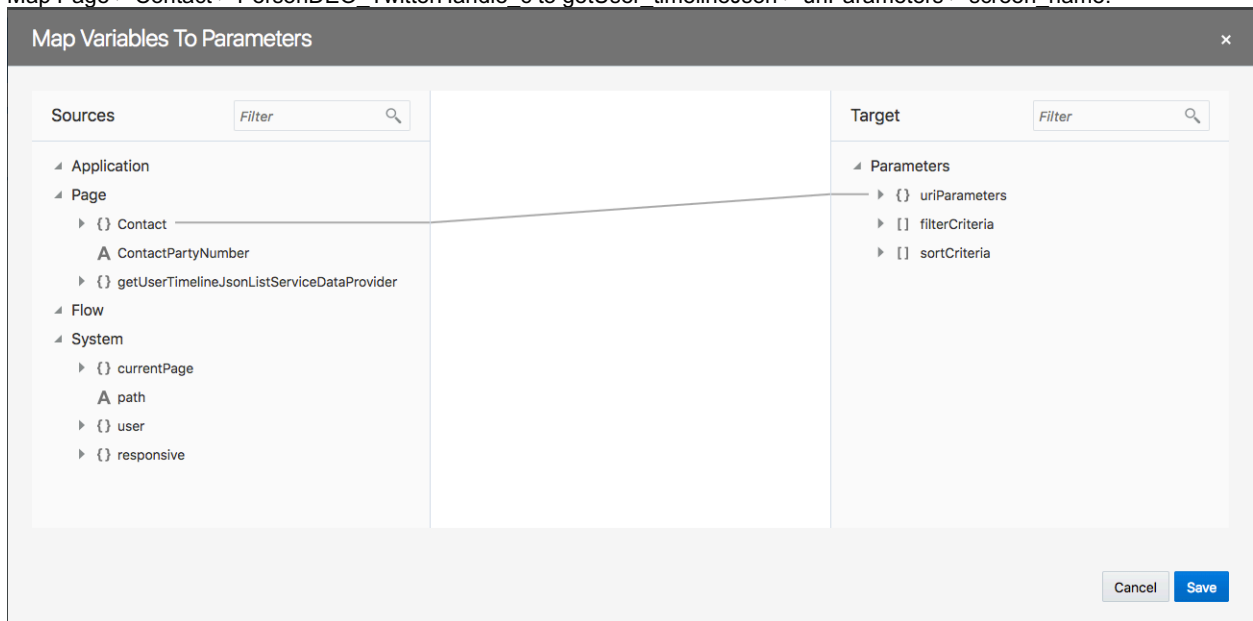
44. Click Finish.
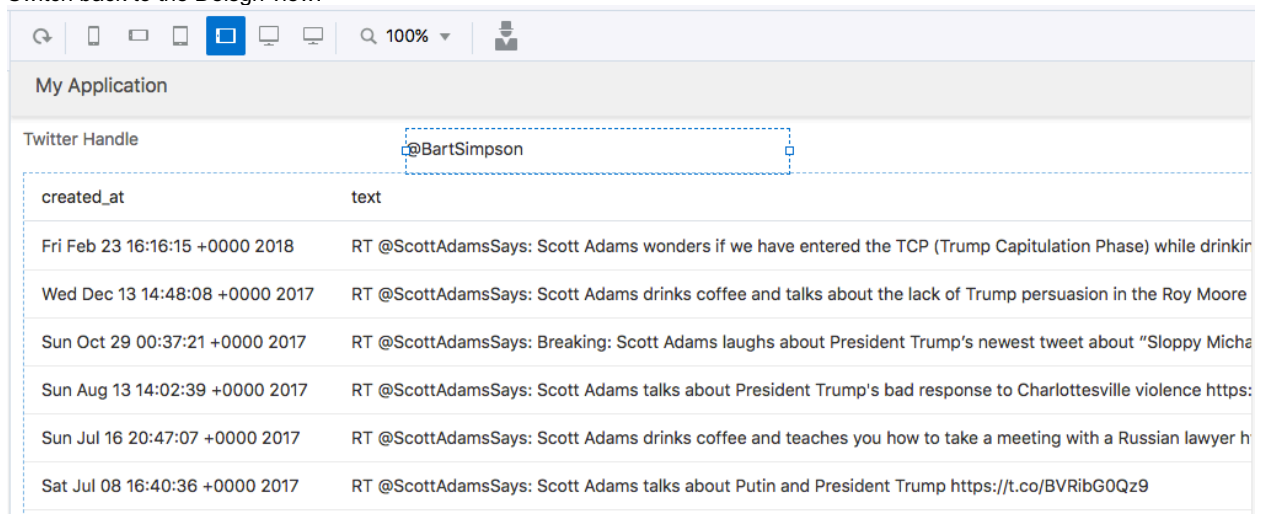
45. Back in the Design tab for the start page, click Code.

In line 9 above the <div class="oj-flex"> for the twitter table, enter a new line and enter <oj-bind-if test="[[$page.variables.Contact.PersonDEO_TwitterHandle_c != null]]">

46. Ad an ending </oj-bind-if> tag at the bottom of the page

47. Switch back to the Deisgn view.

Conclusion

Oracle Visual Builder Cloud Service is one of the fastest ways to develop and deploy an application to nearly any platform. Whether you are a seasoned application developer or have no technical background at all, VBCS gives you the ability to work with and share data more efficiently than ever before.

Be sure to visit the Oracle VBCS web site to get the latest details about this revolutionary application development tool and many other Oracle Cloud offerings.

https://cloud.oracle.com/visual-builder