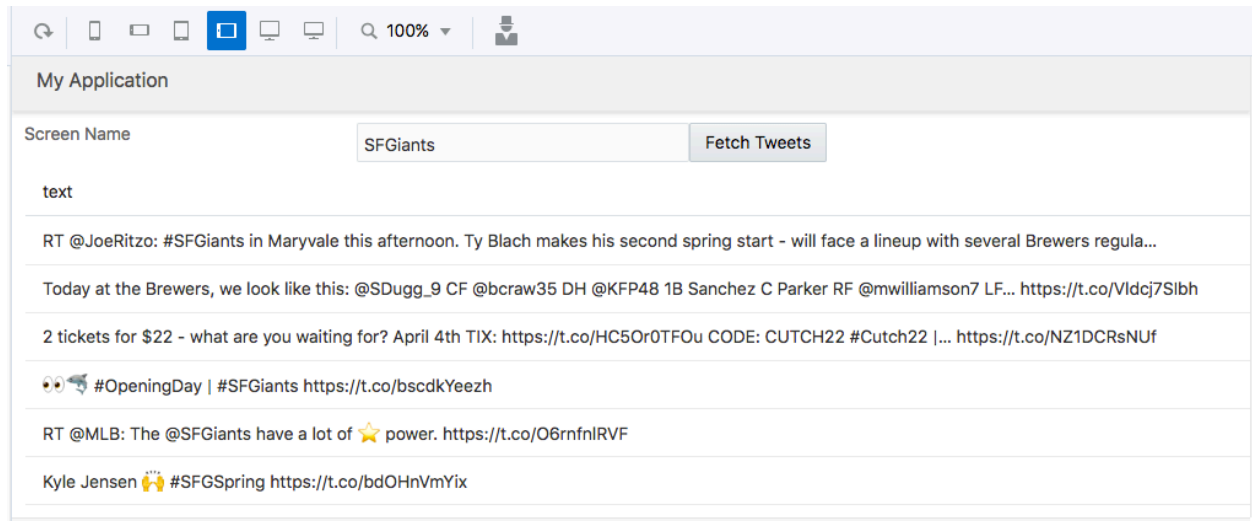


VBCS Training Lab 02

Integrating an External REST API

Introduction

In this hands-on-lab, we will create a new application that retrieves a list of tweets for a user.



The screenshot shows a web browser window with a search bar and a user profile icon. Below the browser window is a web application titled "My Application". The application has a "Screen Name" input field containing "SFGiants" and a "Fetch Tweets" button. Below the input field is a "text" label. The application displays a list of tweets from the user "SFGiants".

Screen Name: SFGiants Fetch Tweets

text

RT @JoeRitzo: #SFGiants in Maryvale this afternoon. Ty Blach makes his second spring start - will face a lineup with several Brewers regula...

Today at the Brewers, we look like this: @SDugg_9 CF @bcraw35 DH @KFP48 1B Sanchez C Parker RF @mwilliamson7 LF... <https://t.co/Vldcj7Slbh>

2 tickets for \$22 - what are you waiting for? April 4th TIX: <https://t.co/HC5Or0TFOu> CODE: CUTCH22 #Cutch22 |... <https://t.co/NZ1DCRsNUf>

👁️🗨️ #OpeningDay | #SFGiants <https://t.co/bscdkYeezh>

RT @MLB: The @SFGiants have a lot of 🌟 power. <https://t.co/O6rnfnlRVF>

Kyle Jensen 🙌🏼 #SFGSpring <https://t.co/bdOHnVmYix>

Hands on Lab Instructions

Creating a Web Application



1. Click New button to create a new application
2. Name the application VBCSLab02
3. Append your username to the Application ID to make it unique – i.e. VBCSLab02tMyName
4. Click Finish

Creating an Application in Twitter

Before you can connect to the Twitter APIs, you need to register for the developer program and create an app in Twitter.

5. Go to <https://developer.twitter.com/en/docs/basics/getting-started> and follow the instructions under Get started: **Build an app on Twitter**.

Registering the Twitter API

6. In the left sidebar, click the Service Connections icon 
7. Click the  icon in the Services panel
8. Click Define By Endpoint
9. 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)

10. In the Authentication tab, enter the following information. Go to <https://apps.twitter.com> to get this information:

Field	Value
Authentication Method	Client Credentials OAuth 2.0
Client Id	The Consumer Key (API Key) from your app
Secret	Consumer Secret (API Secret)
Scope	Leave blank
Authorization URL	https://api.twitter.com/oauth2/token

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

Authentication Mechanism

Client Credentials OAuth 2.0

Client Id ^{*}

Q5GfmUalvQYVTAH0ZAFyd31HH

Secret ^{*}

Secret

Scope

IB0UIQfLtxFyioOcqEHeGbFYPrHmXqxtlgzUctgfbskUaJ0ThV

Authorization URL ^{*}

https://api.twitter.com/oauth2/token

11. 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

Retrieve Many

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		<input type="checkbox"/> X
count	Integer		<input type="checkbox"/> X

Static

12. 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: GET URL: https://api.twitter.com/1.1/statuses/user_timeline.json Action Hint: 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


13. Click Create.

Create the UI

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

15. Drop an Input Text component in the page.

16. Set the text in the label to Screen Name.

17. Click the Variables button  and create a new String variable called screenNameTemp.

Variable

General Events

Id * screenName

Type * String

Default Value SFGiants

Persisted None

☐ Input Parameters

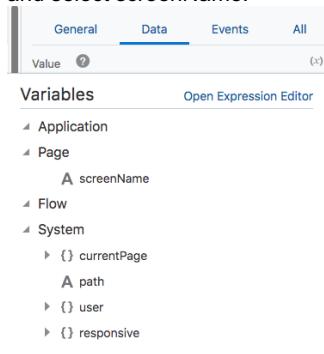
☐ Pass On URL

☐ Required

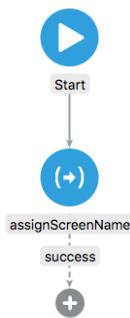
18. Create another String variable called screenname.

19. Click the Designer button .

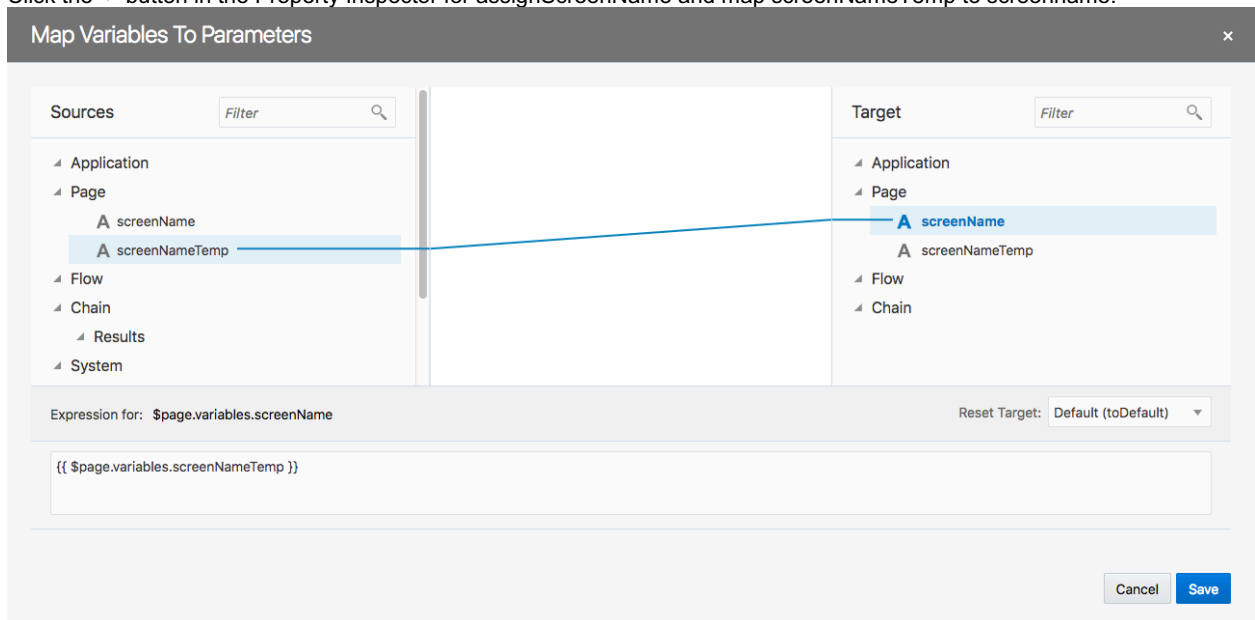
20. Select the Input Text component. In the Properties Inspector, click Data. Click the Variables button above the Value field and select screenName.



21. Drop a button next to the input text and set its display text to Fetch Tweets.
22. With the button selected, open the Events tab of the Properties Inspector. Click New Event > click quickstart
23. In the ButtonClickAction editor, drag an Assign Variables action under Start. Rename the action AssignScreenName.



24. Click the -> button in the Property Inspector for assignScreenName and map screenNameTemp to screenname.



25. Drop a table under the Input Text component. In the Properties Inspector, click Add Data.
26. Select Service Connections > Twitter > getUser_timelineJson and click Next.

27. 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
- ☐ lang
- ☐ place
- ☐ possibly_sensitive
- ☐ quoted_status_id
- ☐ quoted_status_id_str
- ☐ retweet_count
- ☐ retweeted
- ☐ retweeted_status
- ☐ source
- ☒ text
- ☐ truncated
- ☐ user

28. Map Page > screenName to getUser_timelineJson > uriParameters > screen_name.

Add Data

Cancel Select Endpoint Choose Columns Define Query Next Finish

Define Query Parameters

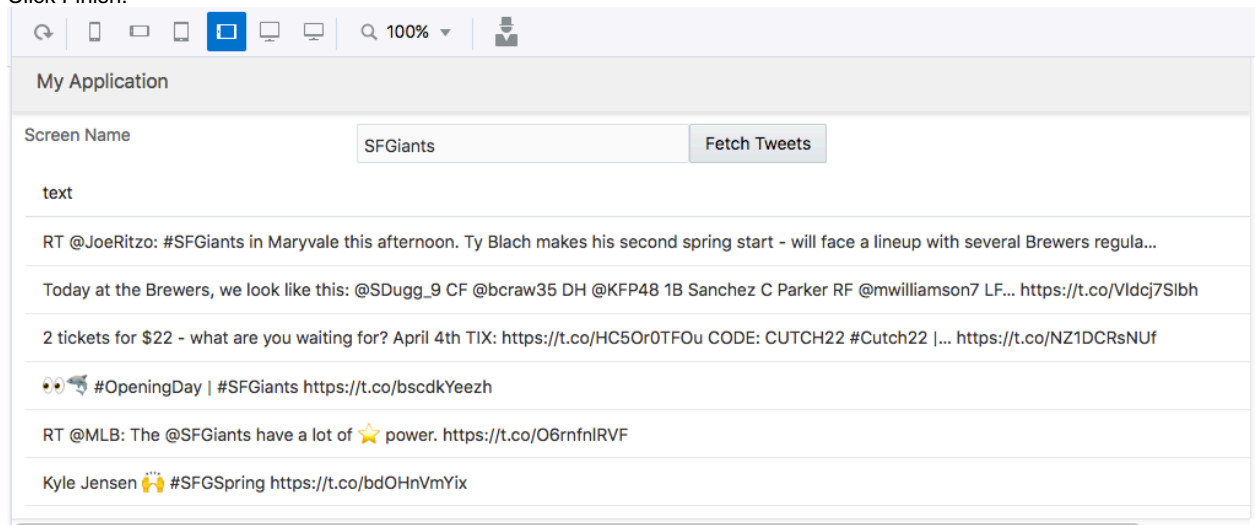
Select defined query parameters to display for your table.

Sources	Target
<ul style="list-style-type: none">Application<ul style="list-style-type: none">currentPagepathuserresponsivePage<ul style="list-style-type: none">screenNamescreenNameTemp	<ul style="list-style-type: none">getUser_timelineJson<ul style="list-style-type: none">uriParameters<ul style="list-style-type: none">screen_namecountheadersfilterCriteria

Expression for: uriParameters.screen_name

{{ \$page.variables.screenName }}

29. Click Finish.



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>