



PUBLIC

Simplify development of business apps with a visual no-code experience

Workshop Steps

TABLE OF CONTENTS

1. DEFINE THE APP'S STRUCTURE.....	3
2. CREATE THE USER INTERFACE AND SET THE APP THEME	7
3. DATA SOURCE – SAP API BUSINESS HUB	13
4. USE THE API FOR THE DATA.....	16
5. ADD LOGIC TO SELECT WHICH BANK DETAILS ARE SHOWN	24
EXTRA 1 – ADD A FILTER TO THE BANKS LIST	28
EXTRA 2 – MAKE FILTER WORK IMMEDIATELY.....	32

www.sap.com/contactsap

© 2022 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

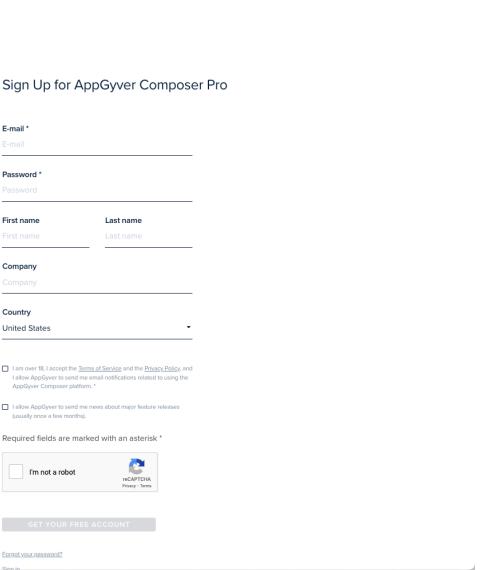
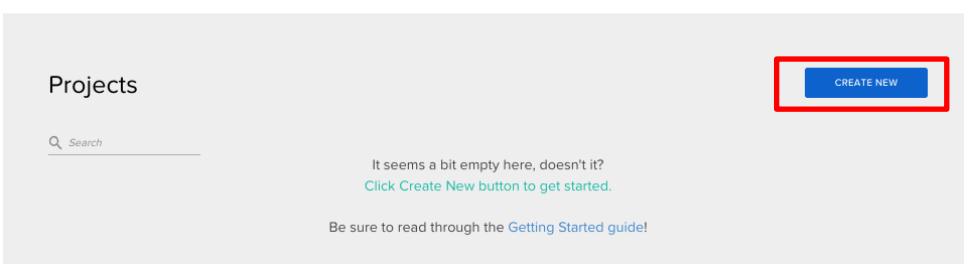
SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See www.sap.com/trademark for additional trademark information and notices.

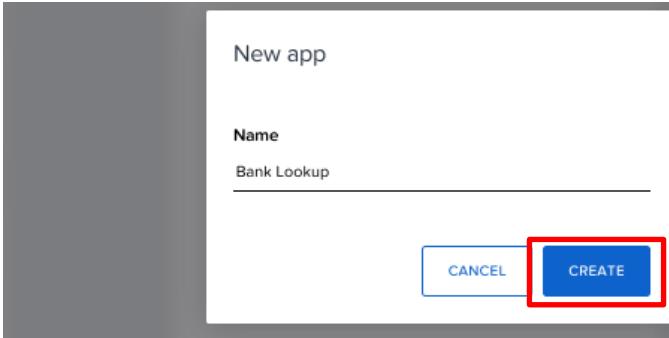
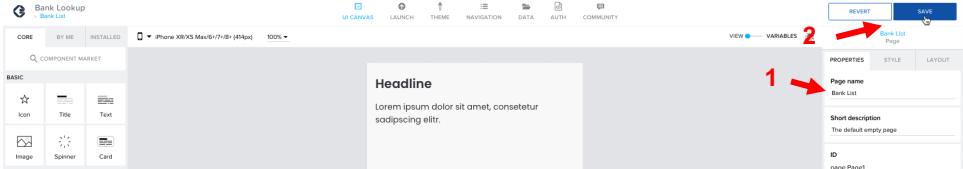
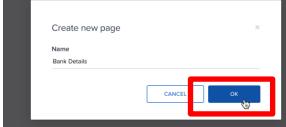
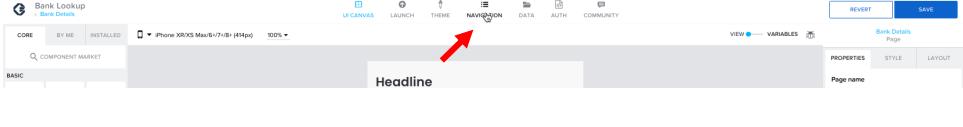
1. DEFINE THE APP'S STRUCTURE

In this section, you will setup your AppGyver account and begin creating a first app.

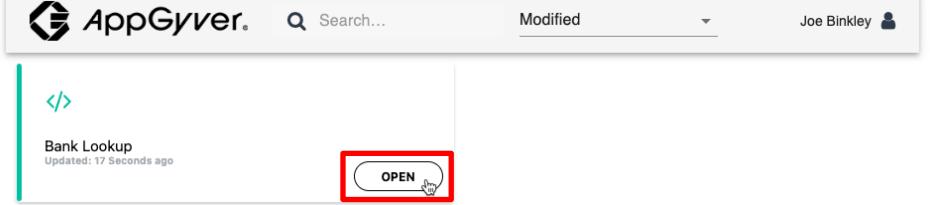
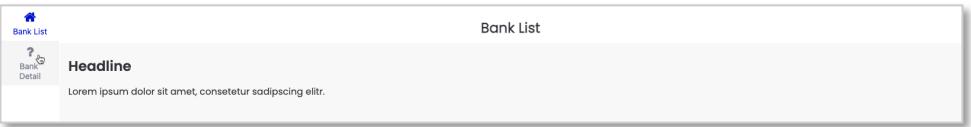
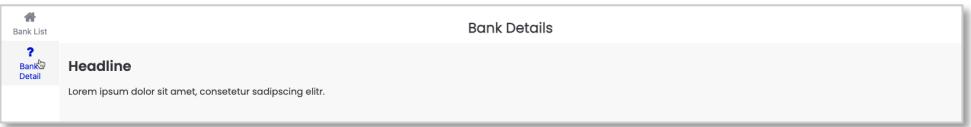
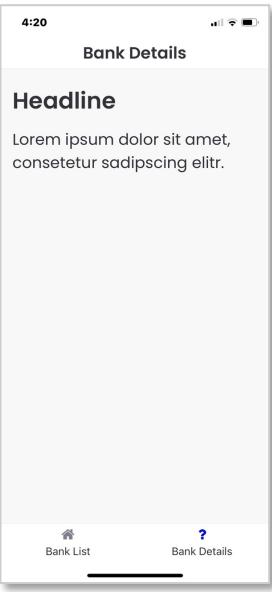
For simplicity, we'll use the community edition of AppGyver, but apart from sign-up instructions, these guidelines should work for the BTP SAP AppGyver edition.

The app will have two pages – a list page and a detail page.

	Actions	Screenshots
1	Please sign up for a free AppGyver community edition account at www.appgyver.com	 
2	Click "CREATE NEW"	

	Actions	Screenshots
3	Name it “Bank Lookup” and click CREATE	
4	1. Change the page name to: “Bank List” 2. Save your changes.	
5	Click on Bank List to go to the Pages view	
6	Click ADD NEW PAGE	
7	Name the page “Bank Details” and Click “OK”	
8	You should now be on the UI Canvas for the Bank Details Page. Click NAVIGATION to setup page navigation	
9	Click “Add Item” to add our detail page to the navigation	

	Actions	Screenshots
10	<ol style="list-style-type: none"> Set the Tab name to “Bank Details” Set the Page to: Bank Details 	<p>The screenshot shows the 'Navigation header bar' configuration. A red arrow labeled '1' points to the 'Tab name' field where 'Bank Details' is selected. Another red arrow labeled '2' points to the 'Page' dropdown menu which is currently set to 'Bank List'.</p>
11	<p>Let's also fix the “Empty Page” label.</p> <ol style="list-style-type: none"> Click on Empty page in the page list Change the Tab name to “Bank List” Change the Page to “Bank List” Click Save 	<p>The screenshot shows the 'Navigation header bar' configuration. A red arrow labeled '1' points to the 'Label' field where 'Empty Page' is selected. Another red arrow labeled '2' points to the 'Tab name' field where 'Empty Page' is selected. A third red arrow labeled '3' points to the 'Page' dropdown menu which is currently set to 'Bank List'. A fourth red arrow labeled '4' points to the 'SAVE' button at the top right.</p>
12	<p>Let's see a preview. Click “LAUNCH”</p>	<p>The screenshot shows the 'PREVIEW' section of the preview portal. A red arrow labeled '1' points to the 'OPEN APP PREVIEW PORTAL' button. A red dashed box labeled '2' highlights the 'Mobile Apps' section, which includes links to download from the App Store and Google Play, and a QR code for logging in. A red arrow labeled '3' points to the QR code itself.</p>
13	<p>1. Open the Preview Portal to see the web view of the App.</p> <p>Optional – Preview on your mobile device</p> <p>2. Get “SAP AppGyver Preview” from the Apple App Store or the Google Play store</p> <p>3. Click on the QR code box to reveal the QR code for your apps</p> <p>4. Scan the QR code in the preview app to see the app on your device</p>	<p>The screenshot shows the 'PREVIEW' section of the preview portal. A red arrow labeled '1' points to the 'OPEN APP PREVIEW PORTAL' button. A red dashed box labeled '2' highlights the 'Mobile Apps' section, which includes links to download from the App Store and Google Play, and a QR code for logging in. A red arrow labeled '3' points to the QR code itself.</p>

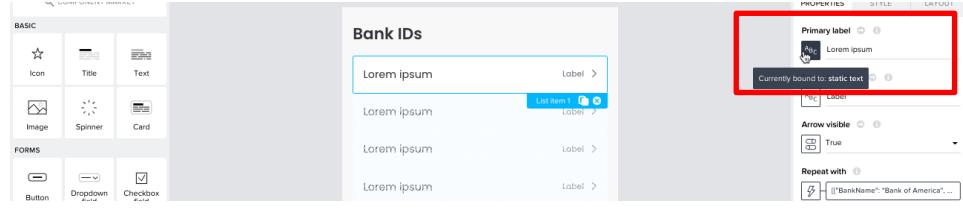
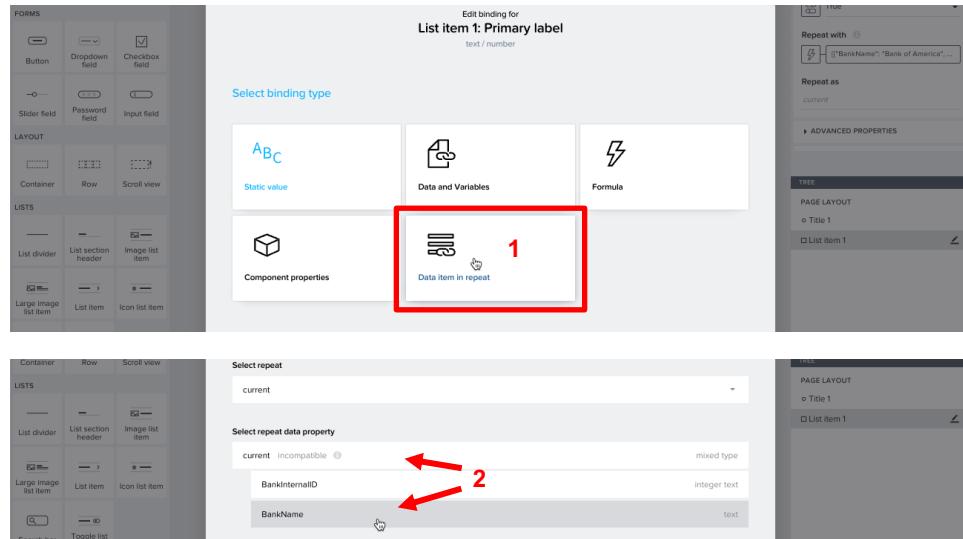
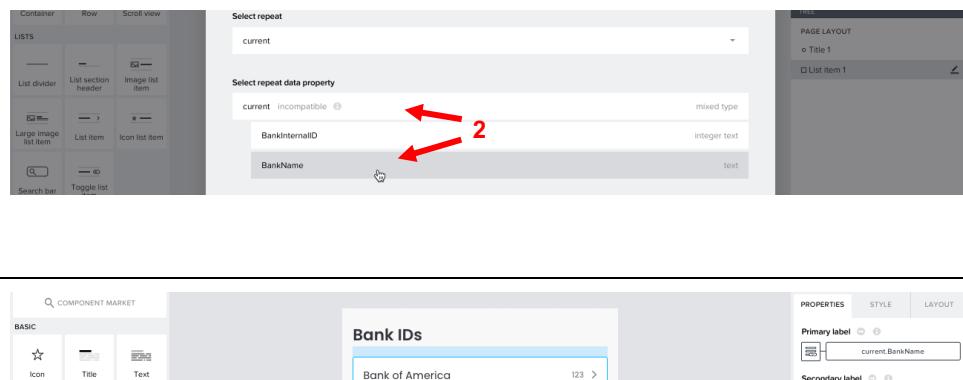
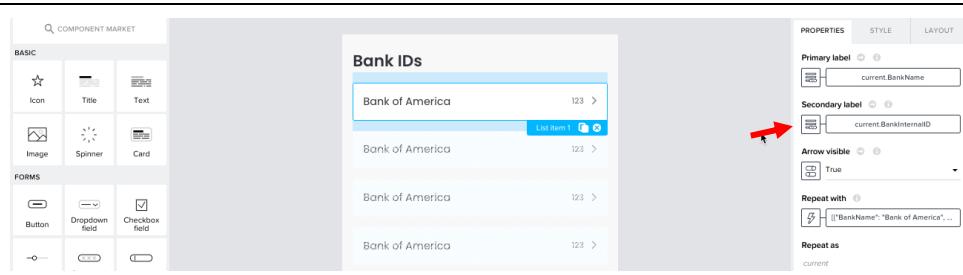
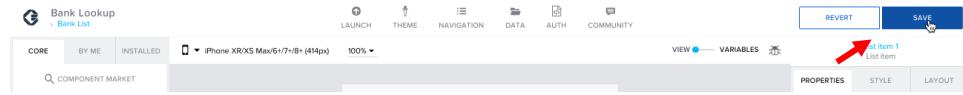
	Actions	Screenshots
14	<p>The preview will open in a new tab.</p> <p>Click OPEN to see the initial view of the app</p>	
15	<p>Previews</p> <p>Leave a tab open to the preview and/or your preview app open in your mobile device.</p> <p>As we step through the exercise, return to the preview and notice that it updates automatically after each save.</p>	<p>In a browser:</p>  <p>After clicking Bank Detail:</p>  <p>On an iPhone:</p> 

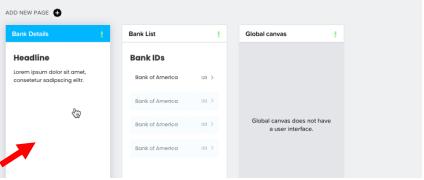
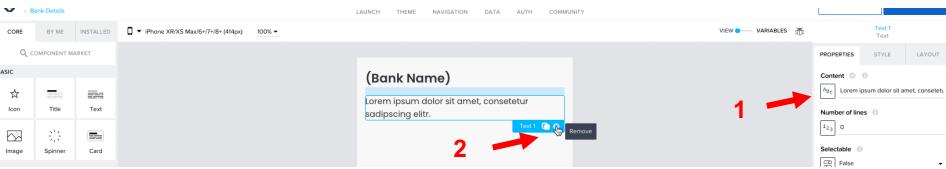
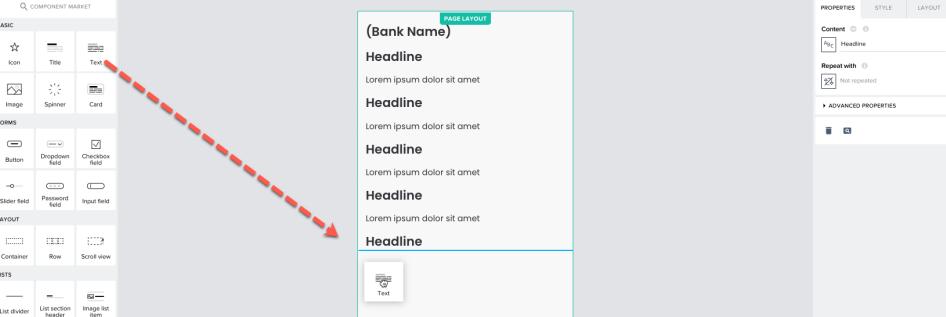
2. CREATE THE USER INTERFACE AND SET THE APP THEME

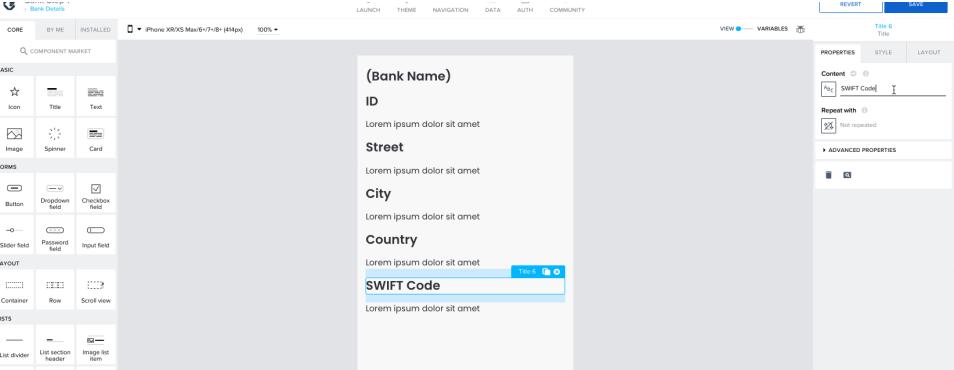
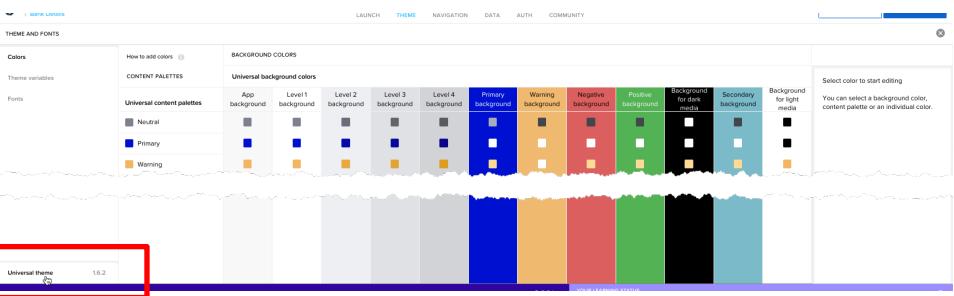
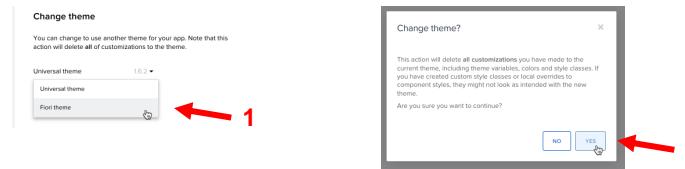
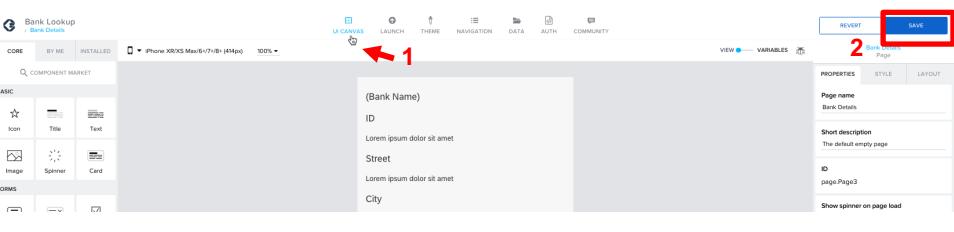
In this section, you will add the fields to show a list of banks using temporary data and layout the view for bank details.

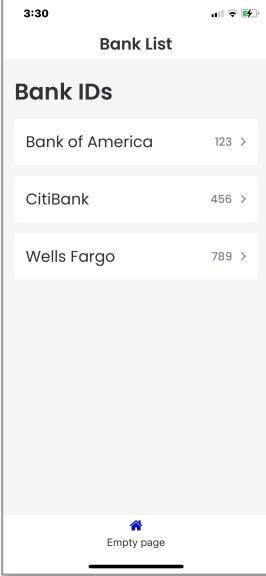
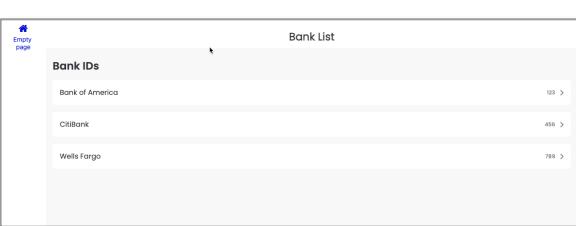
	Actions	Screenshots
1	<ol style="list-style-type: none"> 1. Return to the AppGyver UI Canvas view to work on the user interface of your app. 2. Make sure you are on the Bank List page 	
2	<ol style="list-style-type: none"> 1. Click on the "Headline" object 2. update the content to "Bank IDs" 	
3	Remove the text block under the Bank IDs by clicking on the blue "X"	
4	Drag a "List Item" control onto the canvas under the Bank IDs heading	

	Actions	Screenshots
5	<ol style="list-style-type: none"> Click on the new list item to adjust its properties. Click on the icon under “Repeat with” to setup the data to use. 	
6	<p>Temporarily, we'll use a short “hard coded” list of banks. We'll do this by providing a formula.</p> <ol style="list-style-type: none"> Click Formula Click Create formula 	
7	<ol style="list-style-type: none"> Enter the text provided in the formula area. <i>(It is recommended to copy this from the provided text file)</i> Click SAVE 	<p>[{"BankName": "Bank of America", "BankInternalID": "123"}, {"BankName": "CitiBank", "BankInternalID": "456"}, {"BankName": "Wells Fargo", "BankInternalID": "789"}]</p>

	Actions	Screenshots
8	We'll now set the Primary and Secondary labels based on the data. Click on the icon under the Primary label.	
9	1. Choose “Data item in repeat” 2. Click “current” and then “BankName” 3. Save	 
10	Repeat the steps above to set the Secondary label to “BankInternalID”	
11	Save the app updates	

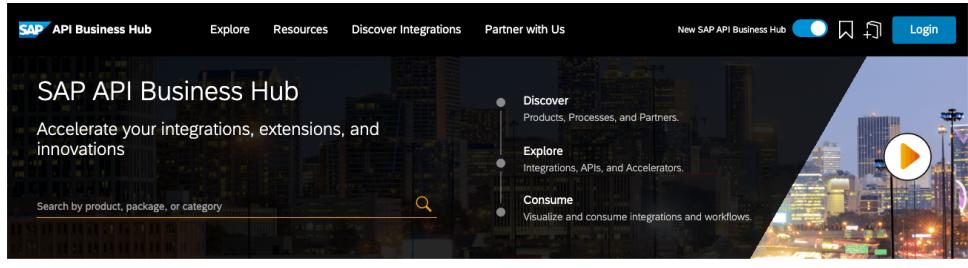
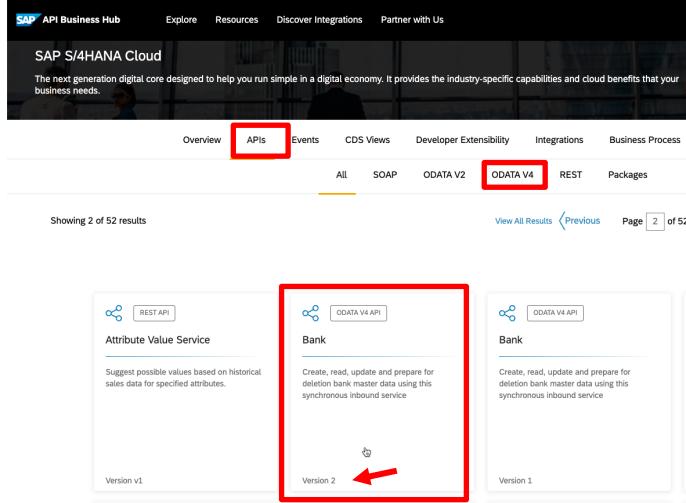
	Actions	Screenshots
12	Navigate to the list of pages	
13	Select the bank details page	
14	<ol style="list-style-type: none"> Change the Content of the top Title component from "Headline" to "(Bank Name)" Delete the text box under it. 	
15	<p>Let's add the detail elements to the UI.</p> <p>Drag 5 pairs of a Title component followed by a Text component to the page</p>	

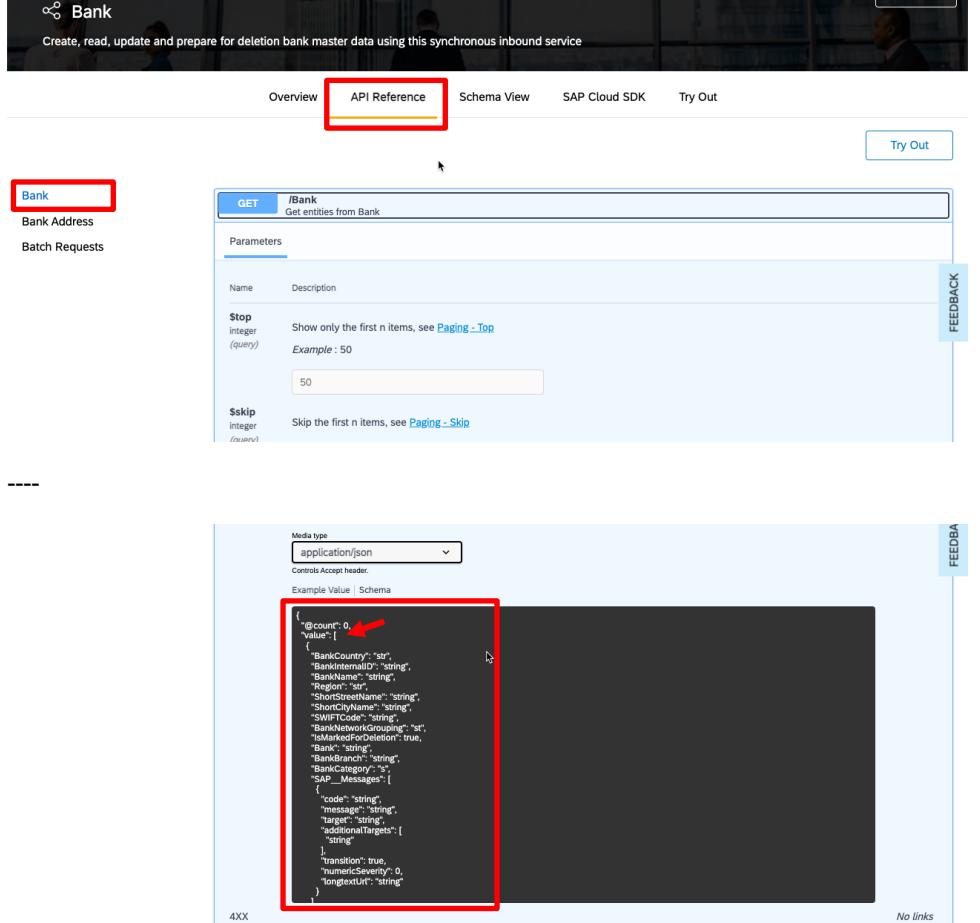
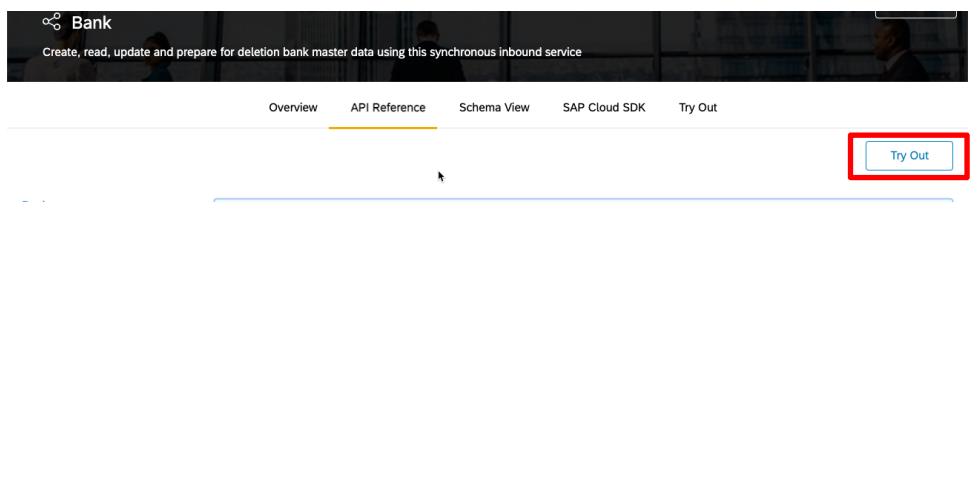
	Actions	Screenshots
16	Change the content of the Title components to: ID Street City Country SWIFT Code	
17	Let's also change the App to use the Fiori theme Click on THEME	
18	At the bottom left of the page, click on the current theme name. (It should say "Universal Theme")	
19	1. Select the Fiori Theme 2. Confirm the theme change	
20	1. Click the UI CANVAS icon at the top to return to the app canvas view The UI should now look like Fiori 2. Click SAVE	

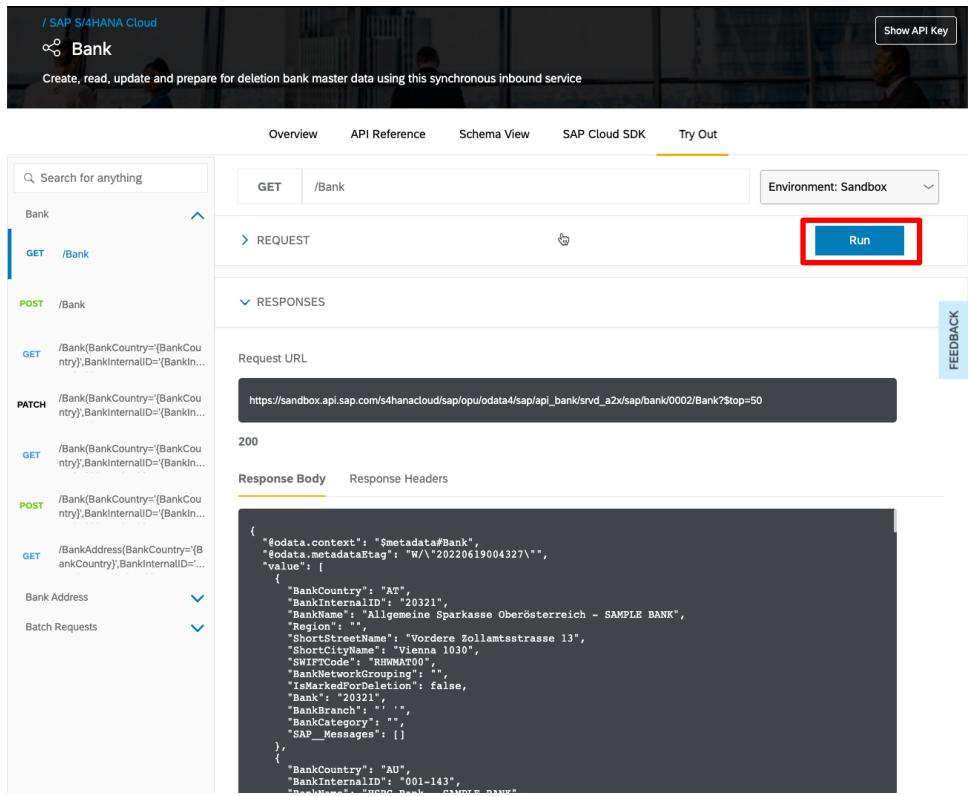
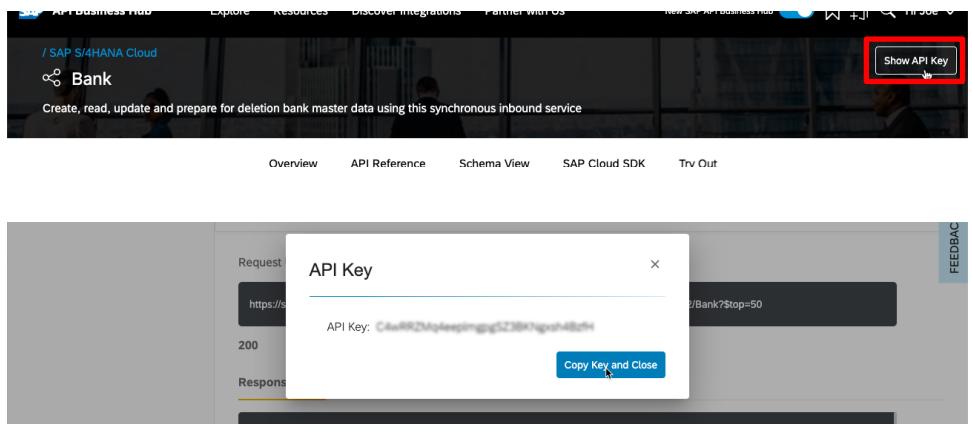
	Actions	Screenshots
21	<p>Preview the App:</p> <p>It should “refresh” automatically in your preview App or preview browser tab</p>	 

3. DATA SOURCE – SAP API BUSINESS HUB

In this section, you will look at the S/4 HANA Cloud API, and get an API Key, so you can bind your app to data in a sandbox instance of S/4 HANA Cloud.

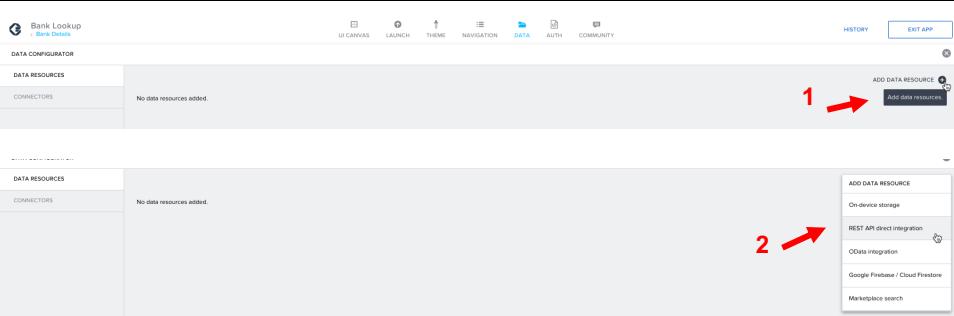
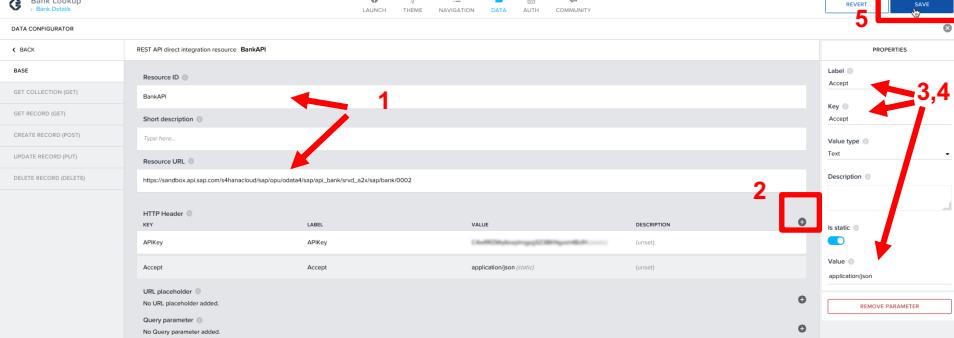
Actions	Screenshots and Explanation
1 Open the SAP API Business Hub at api.sap.com And click the S/4 HANA Cloud tile.	 <p>The screenshot shows the SAP API Business Hub landing page. At the top, there are navigation links: Explore, Resources, Discover Integrations, and Partner with Us. A toggle switch for 'New SAP API Business Hub' is on, along with a 'Logout' button. Below the header is a banner with the text 'SAP API Business Hub' and 'Accelerate your integrations, extensions, and innovations'. To the right is a city skyline image. On the left, there are three circular buttons labeled 'Discover', 'Explore', and 'Consume'. The 'Discover' button is active. Below the banner is a search bar with the placeholder 'Search by product, package, or category'. Underneath the search bar are several navigation tabs: Products (which is selected and highlighted in yellow), Business Processes, Categories, Industries, Partners, and More. In the center, there's a section titled 'Choose a Product to Explore' with the sub-instruction 'See the various resources that each product has to offer'. It lists three products: 'S/4HANA Cloud' (highlighted with a red box), 'S/4HANA', and 'Customer Experience'. Each product card includes a brief description, a list of icons, and a 'View All Products' link.</p>
2 Navigate to the API tab Scroll through the APIs to the Bank API, which we will use to read bank master data records. Click the ODATA V4 API button to look at the Bank API. (We are using version 2.)	 <p>The screenshot shows the SAP S/4HANA Cloud API list page. The top navigation bar includes 'Explore', 'Resources', 'Discover Integrations', and 'Partner with Us'. Below the bar, it says 'SAP S/4HANA Cloud' with a sub-instruction about the next generation digital core. The main content area has tabs: Overview, APIs (which is selected and highlighted in red), Events, CDS Views, Developer Extensibility, Integrations, and Business Process. Under the APIs tab, there are sub-tabs: All, SOAP, ODATA V2 (which is highlighted in red), REST, and Packages. It shows 'Showing 2 of 52 results' and a 'View All Results' link. There are two API cards: 'Attribute Value Service' (under REST API) and 'Bank' (under ODATA V4 API). The 'Bank' card is highlighted with a red box and has an arrow pointing to its 'Version 2' entry. Other details shown include 'Create, read, update and prepare for deletion bank master data using this synchronous inbound service' and 'Version 1'.</p>

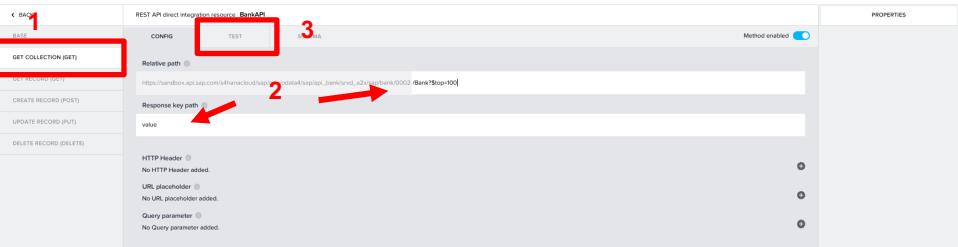
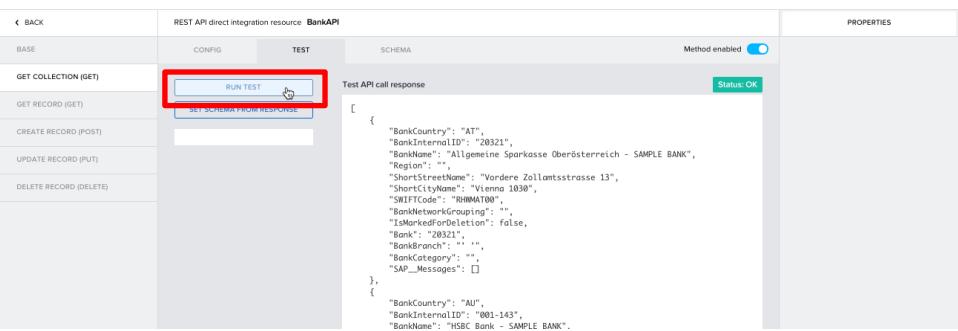
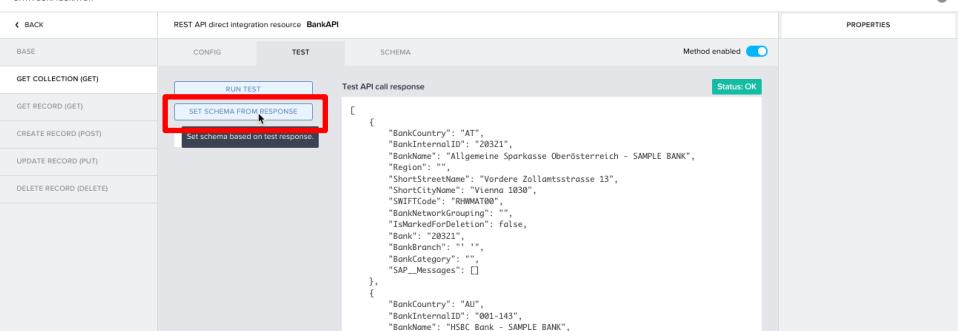
Actions	Screenshots and Explanation
3 Take a few minutes to explore the API reference.	<p>We will use the GET request for the "/Bank" API</p> <p>Notice that the data is returned in json format, with the "value" field containing a list of individual bank records.</p>  <pre data-bbox="796 1030 1008 1305"> { "@count": 0, "value": [{ "BankCountry": "string", "BankInternalID": "string", "BankName": "string", "Region": "string", "ShortStreetName": "string", "ShortStreetNumber": "string", "SWIFTCode": "string", "BankNetworkGrouping": "string", "IsMarkedForDeletion": true, "BankAddress": "string", "BankBranch": "string", "BankCategory": "string", "SAP_Messages": [{ "code": "string", "message": "string", "target": "string", "additionalTargets": ["string"], "transition": true, "numericSeverity": 0, "longTextUrl": "string" }] }] } </pre>
4 Use the Try Out button to send a request to the Sandbox S/4 HANA instance we will be using.	<p>You will need to login. (You may need to create an SAP account.)</p> <p>If you don't have an account and don't want to register for now, you can skip this section and use the API key provided in the text file provided for this workshop.</p> 

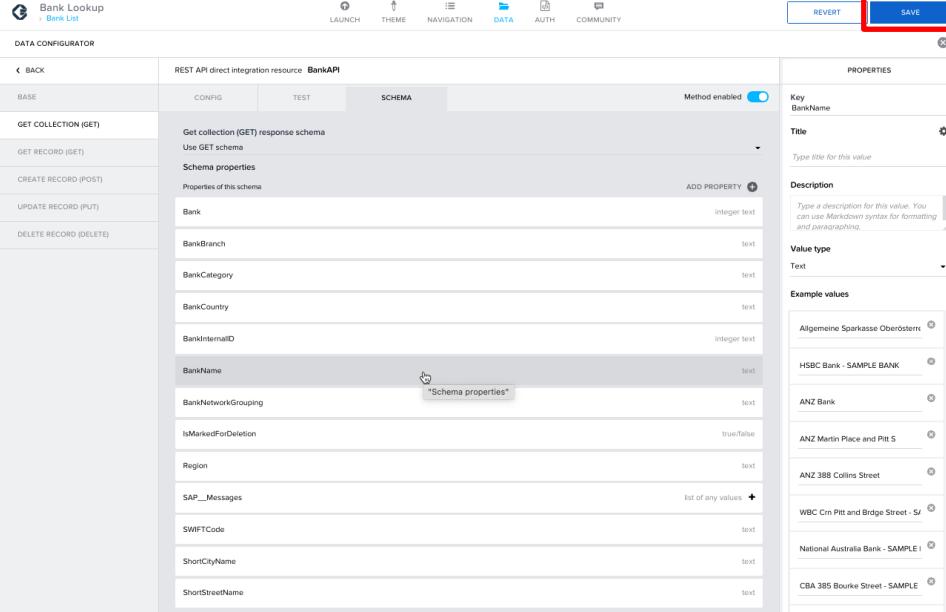
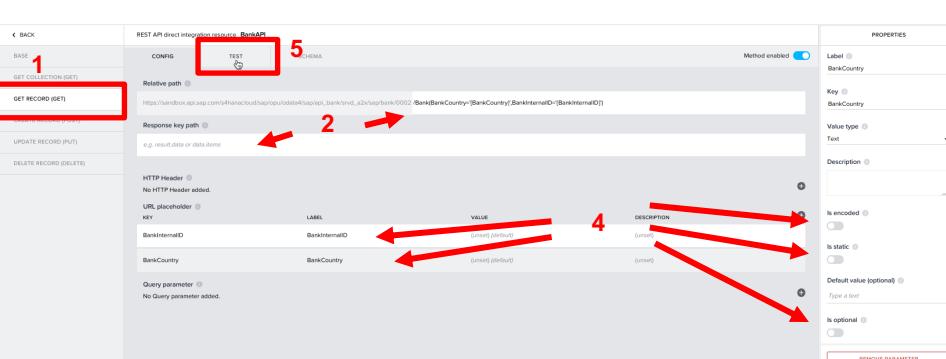
Actions	Screenshots and Explanation
5 Use the Run button to send a request to the sandbox S/4 HANA Cloud instance we will be using.	 <p>The screenshot shows the SAP API Business Hub interface. The top navigation bar includes 'API Business Hub', 'Explore', 'Resources', 'Discover Integrations', and 'Partner with us'. Below the navigation is a search bar and a 'Show API Key' button. The main area is titled '/SAP S/4HANA Cloud' and 'Bank'. It says 'Create, read, update and prepare for deletion bank master data using this synchronous inbound service'. The 'Try Out' tab is selected. On the left, there's a sidebar with various API endpoints like GET /Bank, POST /Bank, etc. The right side shows the 'REQUEST' section with a URL 'https://sandbox.api.sap.com/s4hanacloud/sap/opu/odata4/sap/api_bank/srvd_a2x/sap/bank/0002/Bank?\$top=50'. The 'RESPONSES' section shows a 200 status code and a JSON response body. The response body contains details about a bank, including its name, address, and various identifiers.</p>
6 Note the URL for the API And, to limit to just the first or "top" 50 rows	<p>https://sandbox.api.sap.com/s4hanacloud/sap/opu/odata4/sap/api_bank/srvd_a2x/sap/bank/0002/Bank?\$top=50</p>
7 To use the API in the App, we'll need the API Key, by pressing Show API Key. When the dialog pops up, copy the API key and save in a text (or notepad) file.	 <p>The screenshot shows the SAP API Business Hub interface. The top navigation bar includes 'API Business Hub', 'Explore', 'Resources', 'Discover Integrations', and 'Partner with us'. Below the navigation is a search bar and a 'Show API Key' button. The main area is titled '/SAP S/4HANA Cloud' and 'Bank'. It says 'Create, read, update and prepare for deletion bank master data using this synchronous inbound service'. The 'Try Out' tab is selected. A 'Request' panel shows a URL and a 200 status code. A 'Responses' panel shows a JSON response. A 'Copy Key and Close' button is visible in the 'API Key' dialog.</p>
8 Temporary API Key	<p>If you are unable to login to the API Business Hub, use this key to complete the remaining steps of the exercise:</p> <p>GGxAStBf39qA8MU4gDuhxWqVGILGjo80</p>

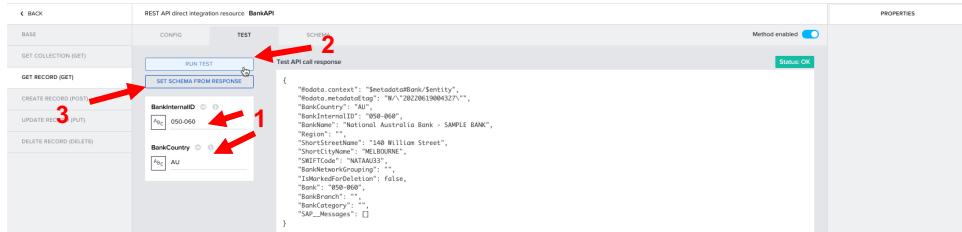
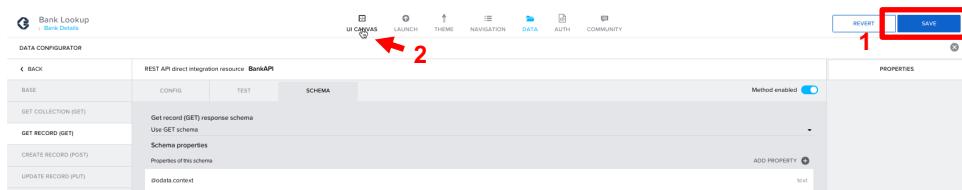
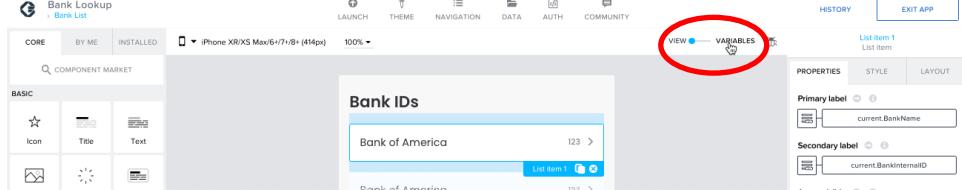
4. USE THE API FOR THE DATA

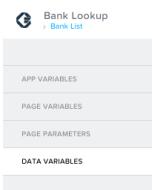
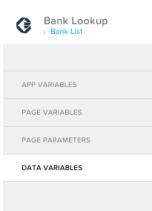
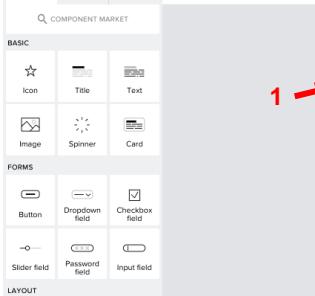
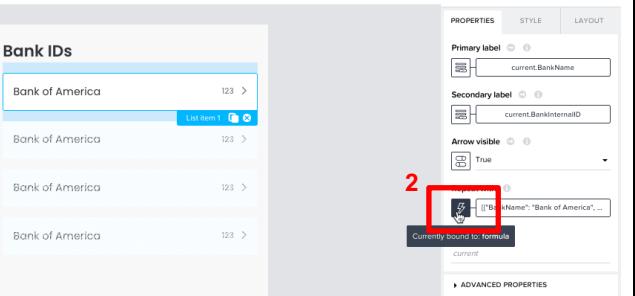
In this section, you will replace the temporary data with data from the sandbox instance of S/4 HANA Cloud using the Banks API.

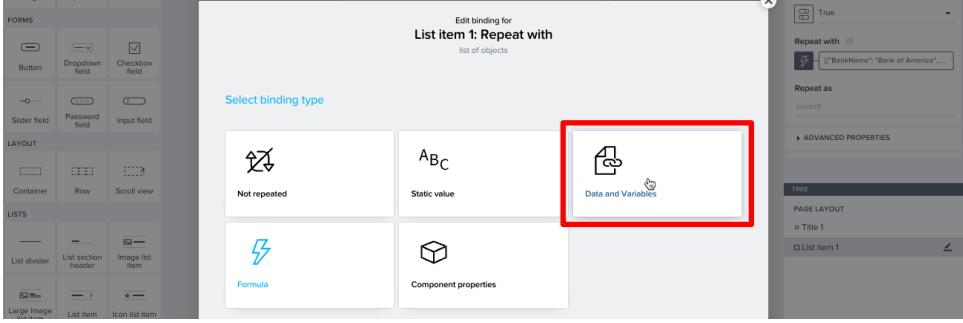
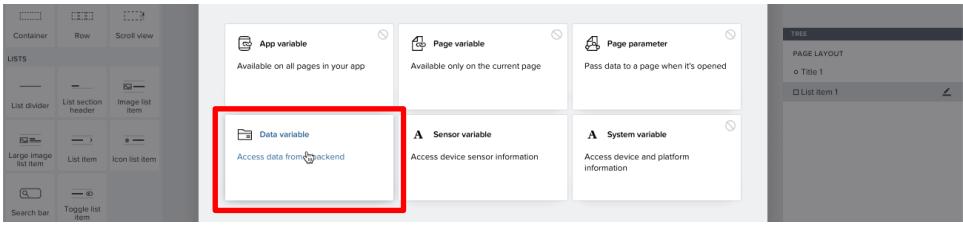
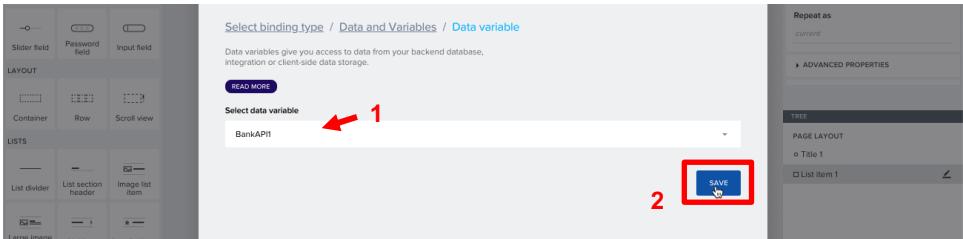
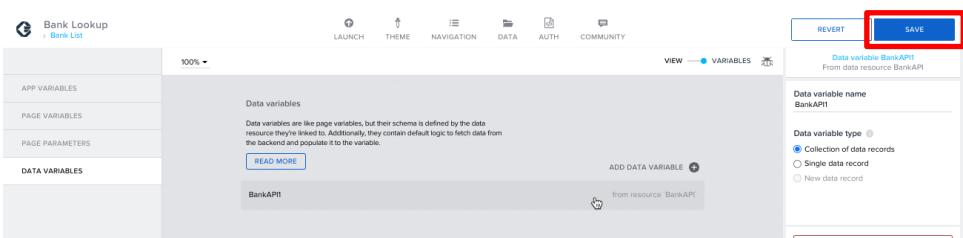
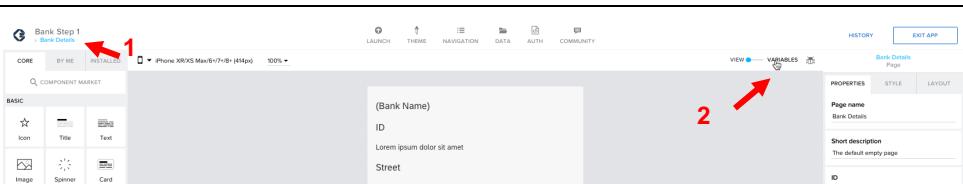
	Actions	Screenshots and Explanation
Note : Most errors in this workshop happen in these steps, so pay close attention to the instructions. Copying strings from PDF files can include invalid characters or extra spaces- all of the texts can be copied from the provided text file. Also ensure the capitalization follows exactly as specified.		
1	Back in AppGyver, Click on DATA to setup a data connection to the S/4 HANA Bank API.	
2	1. Click: ADD DATA RESOURCE 2. Select REST API direct integration	
3	<p>Configure the BASE DATA CONFIGURATION</p> <ol style="list-style-type: none"> Set the Resource ID and URL as shown. It is recommended that you copy the URL from the text file. Add two HTTP Headers Set the first HTTP Header to the Label, Key, and Value provided for API Key. Set the second HTTP Header to the Label, 	 <p>Resource ID: BankAPI Resource URL: https://sandbox.api.sap.com/s4hanacloud/sap/opu/odata4/sap/api_bank/srvd_a2x/sap/bank/0002</p>

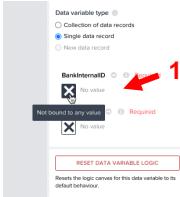
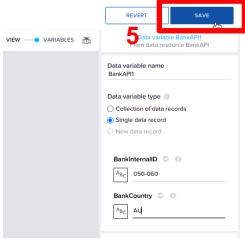
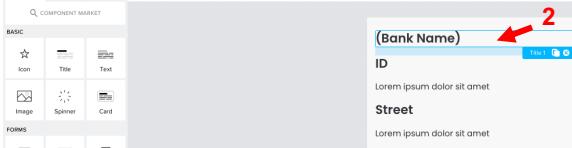
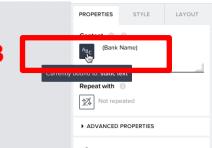
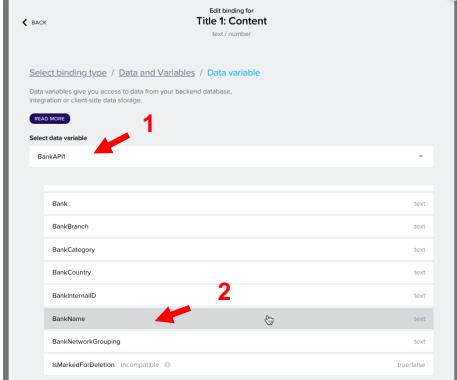
Actions	Screenshots and Explanation
<p>Key, and Value provided for Accept</p> <p>5. Click SAVE</p>	<p>HTTP Header (1) fields: 3 Label: APIKey Key: APIKey Value: <i>Use the API key copied from the previous step.</i></p> <p>HTTP Header (2) fields: 4 Label: Accept Key: Accept Value: application/json</p>
<p>4. 1. Click to Configure GET COLLECTION (GET) 2. Set the Relative path and the Response key path as shown. 3. Click TEST to confirm that the API works.</p>	 <p>Relative path: <i>/Bank?\$top=100</i></p> <p>Response key path: <i>value</i></p>
<p>5. Run the Test</p>	 <p>Test API call response</p> <pre>{ "BankCountry": "AT", "BankInternalID": "20321", "BankName": "Allgemeine Sparkasse Ober\u00f6sterreich - SAMPLE BANK", "Region": "Wien", "ShortStreetName": "Vordere Zollamtsstrasse 13", "ShortCityName": "Vienna 1030", "SWIFTCode": "RWMAT100", "BankNetworkGrouping": "", "ISMarkedForDeletion": false, "Bank": "20321", "BankBranch": "...", "BankCategory": "...", "SAP_Messages": [] }, { "BankCountry": "AU", "BankInternalID": "001-143", "BankName": "HSBC Bank - SAMPLE BANK", "Region": "..." }</pre>
<p>6. Click SET SCHEMA FROM RESPONSE so that it can be used by the app.</p>	 <p>Test API call response</p> <pre>{ "BankCountry": "AT", "BankInternalID": "20321", "BankName": "Allgemeine Sparkasse Ober\u00f6sterreich - SAMPLE BANK", "Region": "Wien", "ShortStreetName": "Vordere Zollamtsstrasse 13", "ShortCityName": "Vienna 1030", "SWIFTCode": "RWMAT100", "BankNetworkGrouping": "", "ISMarkedForDeletion": false, "Bank": "20321", "BankBranch": "...", "BankCategory": "...", "SAP_Messages": [] }, { "BankCountry": "AU", "BankInternalID": "001-143", "BankName": "HSBC Bank - SAMPLE BANK", "Region": "..." }</pre>

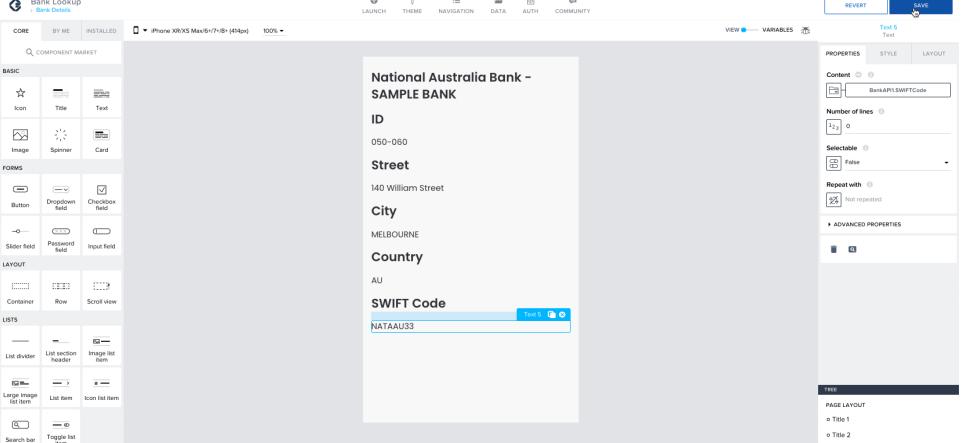
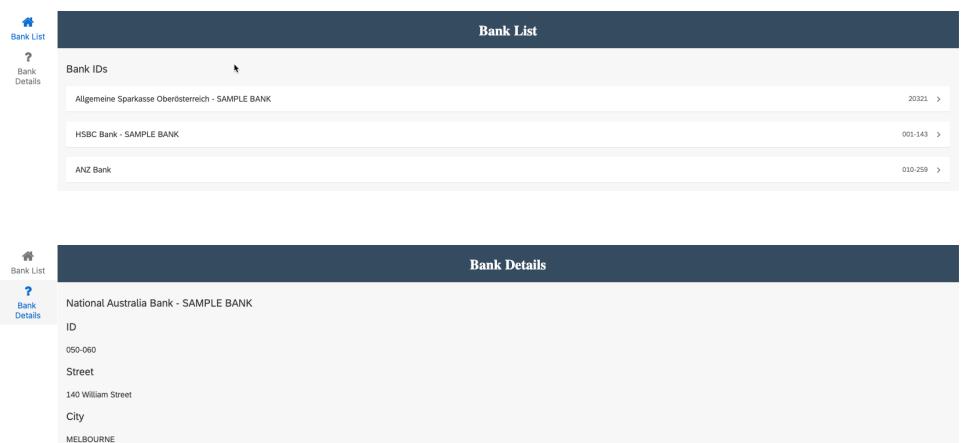
	Actions	Screenshots and Explanation		
7	<p>Take a minute to look at the data returned.</p> <p>Click SAVE</p>			
8	<ol style="list-style-type: none"> Click to Configure GET RECORD (GET) Set the Relative path and the Response key path as shown. (The response key path is empty) Remove the default URL placeholder for "id" Add and configure two URL placeholders as shown Click TEST to confirm that the API works <p><i>Note: Be careful about capitalization for label and key! BankInternalID and BankInternalId are not the same.</i></p>	 <p>Relative path: <code>/Bank(BankCountry='{BankCountry}',BankInternalID='{BankInternalID}')</code> </p> <p>Response key path: <code><leave empty></code> </p> <table border="1"> <tr> <td>URL Placeholder(1): Label: BankInternalID Key: BankInternalID Set IsEncoded to OFF Set IsStatic to OFF Set IsOptional to OFF </td> <td>URL Placeholder(2): Label: BankCountry Key: BankCountry Set IsEncoded to OFF Set IsStatic to OFF Set IsOptional to OFF </td> </tr> </table>	URL Placeholder(1): Label: BankInternalID Key: BankInternalID Set IsEncoded to OFF Set IsStatic to OFF Set IsOptional to OFF	URL Placeholder(2): Label: BankCountry Key: BankCountry Set IsEncoded to OFF Set IsStatic to OFF Set IsOptional to OFF
URL Placeholder(1): Label: BankInternalID Key: BankInternalID Set IsEncoded to OFF Set IsStatic to OFF Set IsOptional to OFF	URL Placeholder(2): Label: BankCountry Key: BankCountry Set IsEncoded to OFF Set IsStatic to OFF Set IsOptional to OFF			

	Actions	Screenshots and Explanation
9	<p>Test the API by</p> <ol style="list-style-type: none"> Setting the test values for BankInternalID and BankCountry as shown Click RUN TEST to confirm it works Click SET SCHEMA FROM RESPONSE 	 <p>BankInternalID: 050-060 BankCountry: AU</p>
10	<ol style="list-style-type: none"> Click SAVE Click the UI CANVAS icon to return to editing the App page 	
11	<p>On the Bank List page</p> <p>Click VARIABLES to add the data source just established as a data variable for this page</p>	

	Actions	Screenshots and Explanation
12	<p>Follow these steps to add the BankAPI as a DATA VARIABLE.</p> <p>(Note that the variable is set to return a collection of records.)</p> <ol style="list-style-type: none"> 1. Select DATA Variables 2. Click ADD DATA VARIABLE 3. Select BankAPI 4. Click SAVE <p>Return to the Bank List page VIEW</p>	    
13	<p>Now, let's replace the temporary data with the data provided by the API.</p> <ol style="list-style-type: none"> 1. Select the list item component 2. Click on Icon button under Repeat with 	 

Actions		Screenshots and Explanation
14	Select Data and Variables	
15	Choose Data variable	
16	1. Select BankAPI 2. Click SAVE to confirm	
17	Click SAVE to save the current state of the App	
18	Now, we'll link up data for the Bank Details page. 1. In UI CANVAS, switch to the Bank Details page 2. Click Variables	

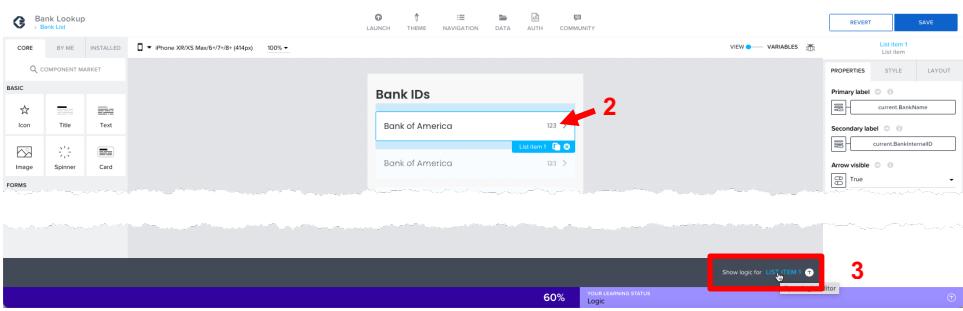
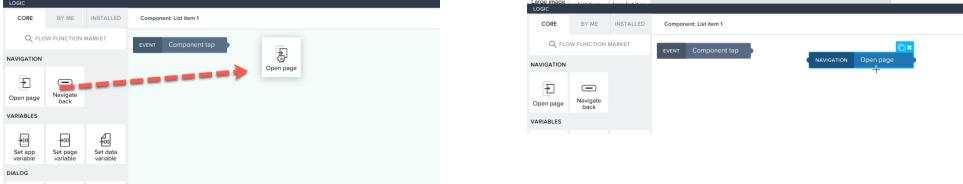
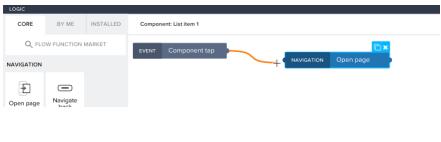
	Actions	Screenshots and Explanation
19	<ol style="list-style-type: none"> Select: DATA Variables Click: ADD DATA VARIABLE Select: BankAPI Select: Single Data Record 	 <p>1</p> <p>2</p> <p>3</p> <p>4</p> <p>5</p>
20	<p>For now, we'll set the two properties to static values.</p> <ol style="list-style-type: none"> Click the "X" icon under BankInternalID Select "Static text" Enter "050-060" Use the same procedure to set BankCountry to "AU" Click SAVE 	 <p>1</p>  <p>2</p> <p>3</p> <p>4</p> <p>5</p>
21	<ol style="list-style-type: none"> Switch back to the VIEW for the Bank Details page Click on Title field at the top of the page <p>Click the Icon under Content</p>	 <p>2</p>  <p>3</p>
22	<ol style="list-style-type: none"> Navigate to the BankAPI (Data and Variables, then Data Variable, then BankAPI) Select: BankName SAVE 	 <p>1</p> <p>2</p>

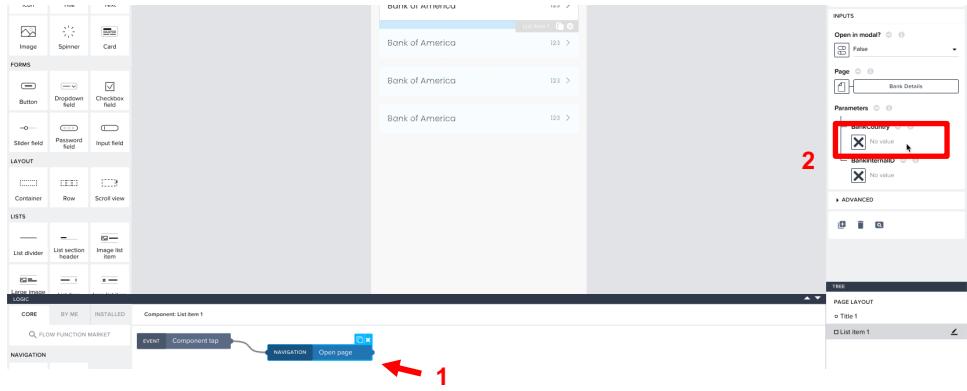
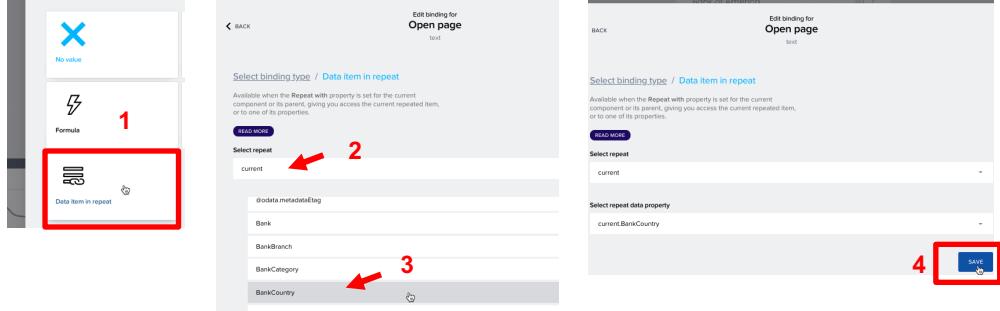
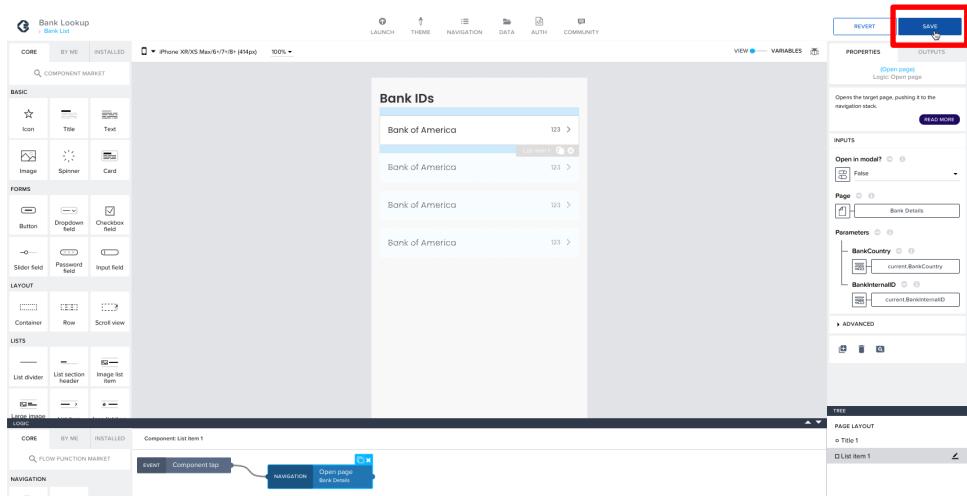
	Actions	Screenshots and Explanation
23	<ol style="list-style-type: none"> 1. Repeat the process above for the five labeled text fields using the data elements indicated 2. SAVE 	 <p>ID: BankInternalID</p> <p>Street: ShortStreetName</p> <p>City: ShortCityName</p> <p>Country: BankCountry</p> <p>SWIFT Code: SWIFTCode</p>
24	<ol style="list-style-type: none"> 3. The Preview should now show data from the database 	

5. ADD LOGIC TO SELECT WHICH BANK DETAILS ARE SHOWN

In this section, you will show the details page when you click on a bank from the list. To do this, we need to add two parameters to the Bank Details page and add an action to the Bank List page that opens the Details page.

	Actions	Screenshots and Explanation
1	<ol style="list-style-type: none"> 1. Navigate to the VIEW for the Bank Details Page 2. Click to switch to VARIABLES 	
2	<ol style="list-style-type: none"> 1. Click on PAGE PARAMETERS 2. Add a page parameter called BankCountry 3. Add a page parameter called BankInternalID 4. Click SAVE 	
3	<p>Now we need to set the data variable for the page to use the page parameters instead of static values</p> <ol style="list-style-type: none"> 1. Click on DATA VARIABLES 2. Click on BankAPI 3. Click the icon next to BankInternalID 4. Click Data and Variables, then Page Parameters, and BankInternalID 5. Save 6. Repeat the same 	

	Actions	Screenshots and Explanation
	procedure for BankCountry	
4	<ol style="list-style-type: none"> 1. Navigate to the Bank List page 2. Click on the List item 3. At the bottom of the canvas, click on : Show Logic for BANK LIST 	
5	On the logic panel, drag an Open Page component to the canvas	
6	<ol style="list-style-type: none"> 1. Point and hold the mouse down at the right bump on the Event component 2. Drag to the left bump of the Open page component 3. Release to connect the two components. 	

Actions	Screenshots and Explanation
<p>7 Opening the Bank Detail page requires setting two page parameters: BankCountry and BankInternalID.</p> <ol style="list-style-type: none"> Click the Open Page logic element Click on the X icon next to BankCountry 	
<p>8</p> <ol style="list-style-type: none"> Click Data item in repeat Click current Click BankCountry Click SAVE Repeat the same process for the BankInternalID page parameter 	
<p>9 Click SAVE</p>	

	Actions	Screenshots and Explanation
10	In preview, clicking on any Bank list item should now show a detail page with the appropriate data	
	<p>Congratulations, you have now created an App that retrieves and shows data from S/4HANA Cloud.</p> <p>If you have time, the following two sections go a little deeper and expose more of the logic you might use in a productive app.</p>	

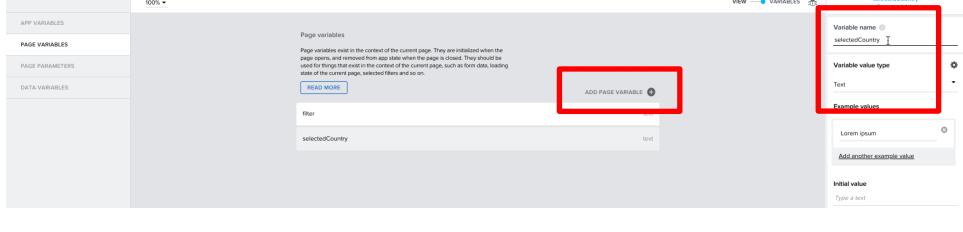
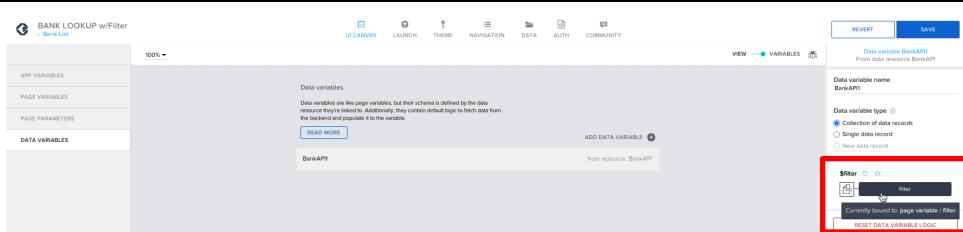
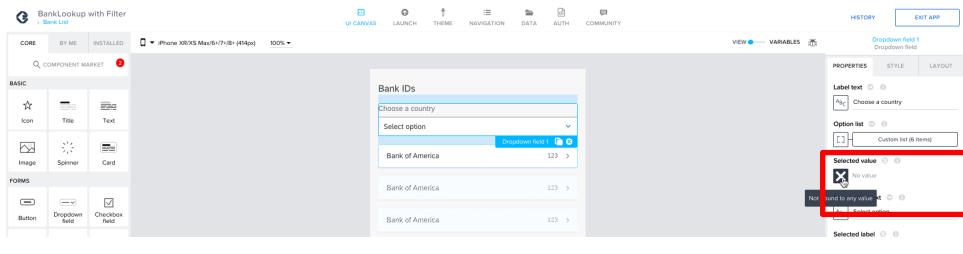
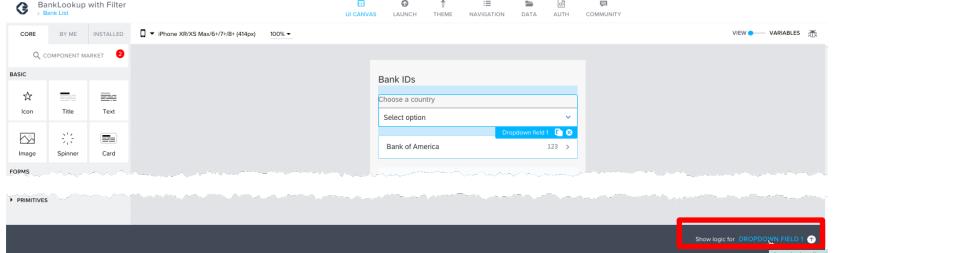
EXTRA 1 – ADD A FILTER TO THE BANKS LIST

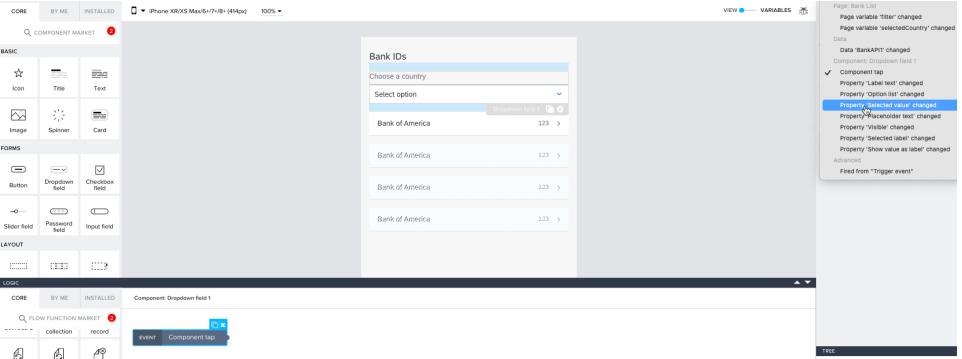
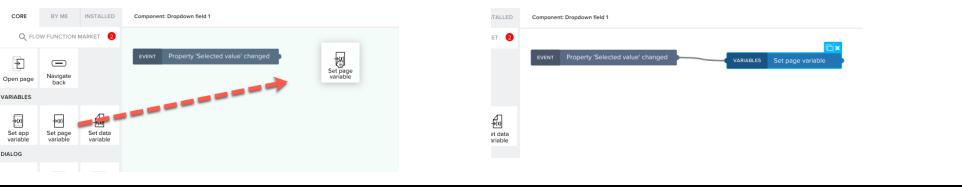
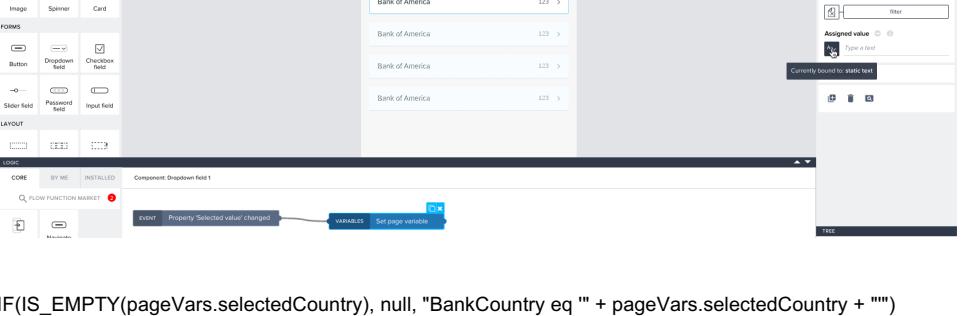
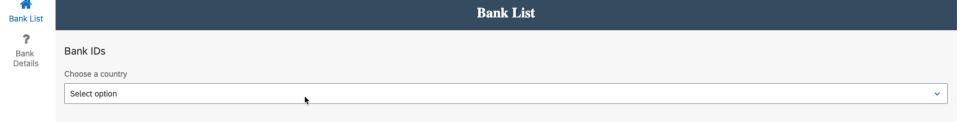
In this section, you will add a drop-down to filter the bank list by country and modify the data connection to use the filter.

Tip: Exit the APP and duplicate, naming the new app “Bank Lookup with Filter”

	Actions	Screenshots and Explanation
1	In the UI Canvas on the BankList page, drag a Dropdown field component to under the Bank IDs heading	
2	1. Set the Label text to “Choose a country” 2. Click on the Icon under options list add some values	
3	Populate this with a list of values. 1. Choose List of Values 2. Add each of the six label/value pairs shown 3. SAVE	
4	Preview the app to see the dropdown	

	Actions	Screenshots and Explanation
5	We will now adjust the API settings to incorporate an optional filter value	
6	<ol style="list-style-type: none"> Click on the DATA canvas Select the BankAPI Select GET COLLECTION Add a query parameter with the settings shown 	<p>Label: \$filter Key: \$filter Set IsStatic to OFF Set IsOptional to ON</p>
7	<p>You can test the filter using this text:</p> <p>BankCountry eq 'US'</p> <p>The results should show just the banks in the United States.</p> <p>Click SAVE.</p>	<pre> { "BankCountry": "US", "BankInternalID": "811000398", "BankName": "Bank of America - NY", "Region": "North America", "ShortCityName": "Church Street", "ShortCityName": "Palo Alto", "SAP_Messages": [], "BankNetworkGrouping": null, "IsMarkedForDeletion": false, "BankCategory": "Retail Banking", "BankBranch": null, "BankCategory": null, "SAP_Messages": [] }, { "BankCountry": "US", "BankInternalID": "811000018", "BankName": "Wells Fargo", "Region": "North America" } </pre>
8	Next we'll add Page variables to be used with the API settings to incorporate the filter value	

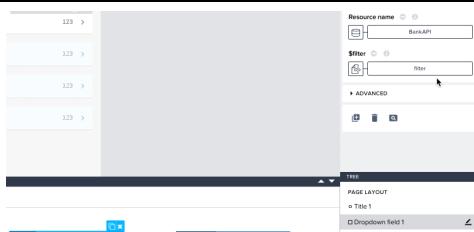
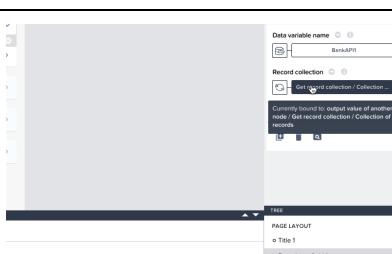
	Actions	Screenshots and Explanation
9	<ol style="list-style-type: none"> From the UI Canvas, Click on VARIABLES Select PAGE VARIABLES Add the two page variables with a value type of "Text": <ul style="list-style-type: none"> - filter - selectedCountry 	
10	<ol style="list-style-type: none"> Click on DATA VARIABLES Update the BankAPI data variable to use the Page Variable "filter" as the value for the \$filter query parameter. 	
11	<ol style="list-style-type: none"> Return to the UI canvas view for Bank List page. Select the Dropdown field Click the icon under Selected value Bind it to the Page Variable named selectedCountry 	
12	Open the logic editor for the Dropdown Field	

	Actions	Screenshots and Explanation
13	Change the trigger event source to: Property “Selected value” changed	
14	Drag a Set Page Variable component to the canvas and connect it to the Event	
15	For variable name, choose “filter” Set the value formula shown This will either set filter to “null”, meaning “don’t filter” Or to something like: BankCountry eq ‘DE’	
16	SAVE and preview the updated app	
17	The list should be empty until a filter is selected. You may also notice a few seconds of “lag” between selected a country and the data updating.	

EXTRA 2 – MAKE FILTER WORK IMMEDIATELY

In this section, you will update the app to update the display data as soon as a country filter is chosen.

	Actions	Screenshots and Explanation
1	<p>Let's start by looking at the logic for the BankAPI data variable.</p> <ol style="list-style-type: none"> Click on the UI Canvas Make sure the Bank List page is selected Click VARIABLES Select DATA VARIABLES Select BankAPI If needed, Click to Show logic 	
2	<p>The logic shows:</p> <ol style="list-style-type: none"> The page is mounted An attempt is made to get the record collection from the server If it's successful, the data variable is set to the collection of records retrieved. If there's an error, it's ignored After a delay of 5000 milliseconds (5 seconds) another attempt is made to update the data (step 2). <p>Since the data is being updated every 5 seconds, it may take 5 seconds for a change in the filters to take effect. Because of this, the update sometimes seems slow.</p>	
3	<p>Let's fix this by adding steps 2 and 3 above to the logic that executes when the drop-down selected value changes</p>	
4	<p>Click back to the UI Canvas and select the Dropdown field</p> <p>In the logic panel add two components</p> <ul style="list-style-type: none"> Get record collection Set data variable <p>Wire them up in the order shown</p>	

	Actions	Screenshots and Explanation
5	For the Get Record Collection component, set: Resource: BankAPI \$filter: the page variable called "filter"	
6	For the Set data variable component, set: Data variable name: BankAPI Record collection: Use the value from the previous step (Get record collection/collection of records)	
7	SAVE and Preview	The updates should now be immediate.
8	Bonus Challenge: Can you setup a default filter, so that some data shows up right away when the page opens before the user chooses a country?	