



App Connect with RPA

Hands-on Lab

Version 1.1

21 September 2021

Authors:

Nigel Crowther – ncrowther@uk.ibm.com

Table of Contents

1. Introduction	3
2. Use case detail	4
3. Prerequisites	5
4. Lab Instructions.....	6
Create a Cloudant database.....	6
Provision an App Connect Instance	11
Import the RPA Bot API	13
Create an App Connect Flow.....	17
Fix Errors in the Flow.....	18
Run the App Connect Flow	25
View App Connect Logs.....	26
View Processing Result in Cloudant	28
Stop the flow.....	29
Advanced Exercise – Asynchronous invocation of RPA API	30
Conclusion.....	33

1. Introduction

In this hands-on lab you will build an App Connect flow to run an RPA bot. The bot will be executed through an API. The specification of the API is here:

<https://rpapi.eu-gb.mybluemix.net/>

When the App Connect flow is executed, it will read leads from Cloudant and insert them into a legacy sales application by invoking an RPA bot. See Figure 1.

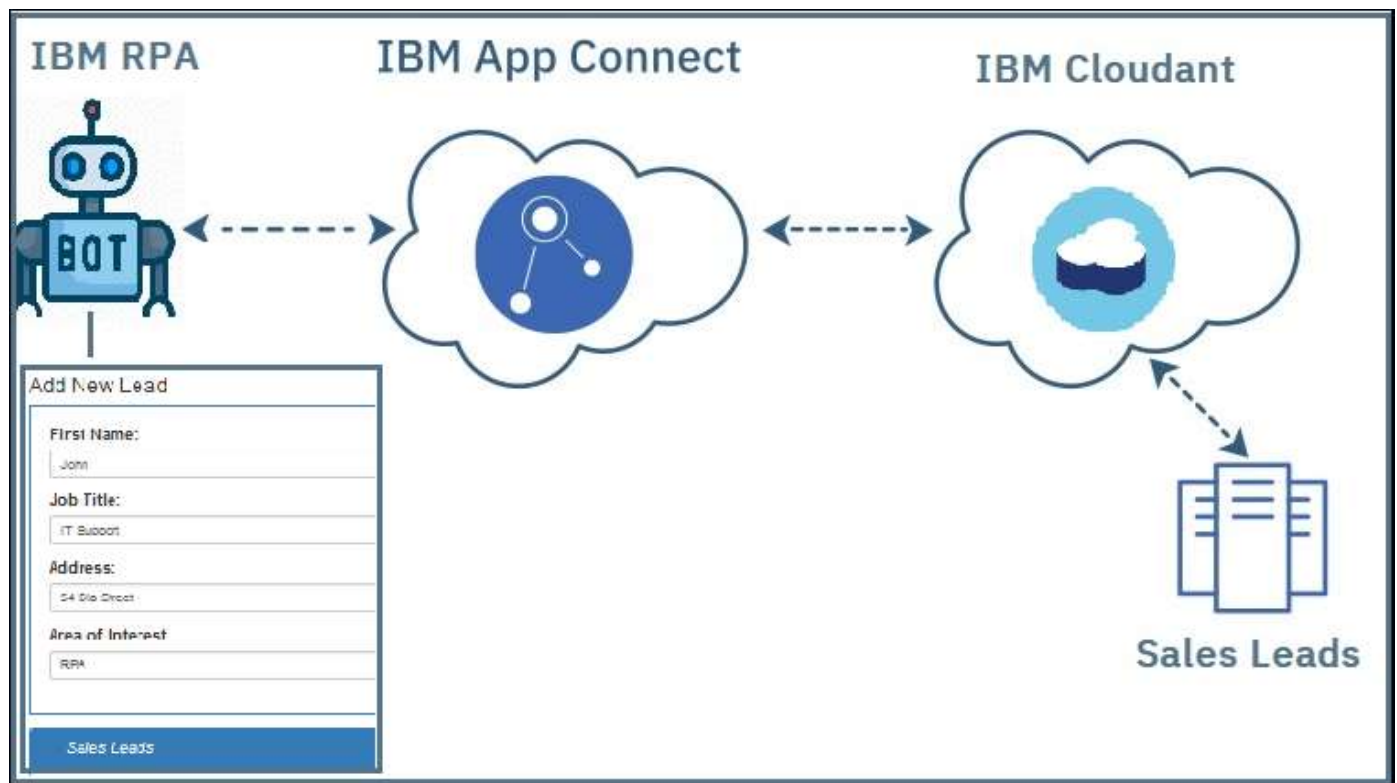


FIGURE 1. LAB ARCHITECTURE

2. Use case detail

Sales leads are currently captured by the sales team and entered to a Cloudant database. The sales leads are manually copy/pasted into the sales lead application via its legacy user interface. This task is error prone, and the sales team repeatedly ask if this can be automated, but automation has been too difficult. Until now ...

3. Prerequisites

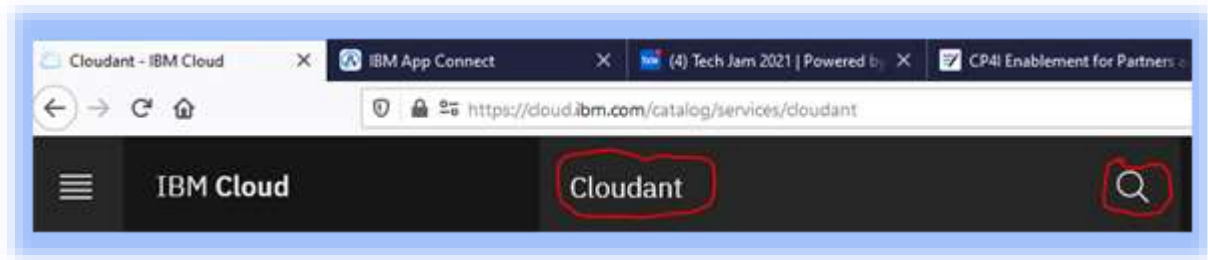
- An IBM Cloud Account. If you don't have one, visit: <https://cloud.ibm.com/registration>
- App Connect Lite (free)
- Cloudant Lite (free)

4. Lab Instructions

Create a Cloudant database

Log into IBM Cloud

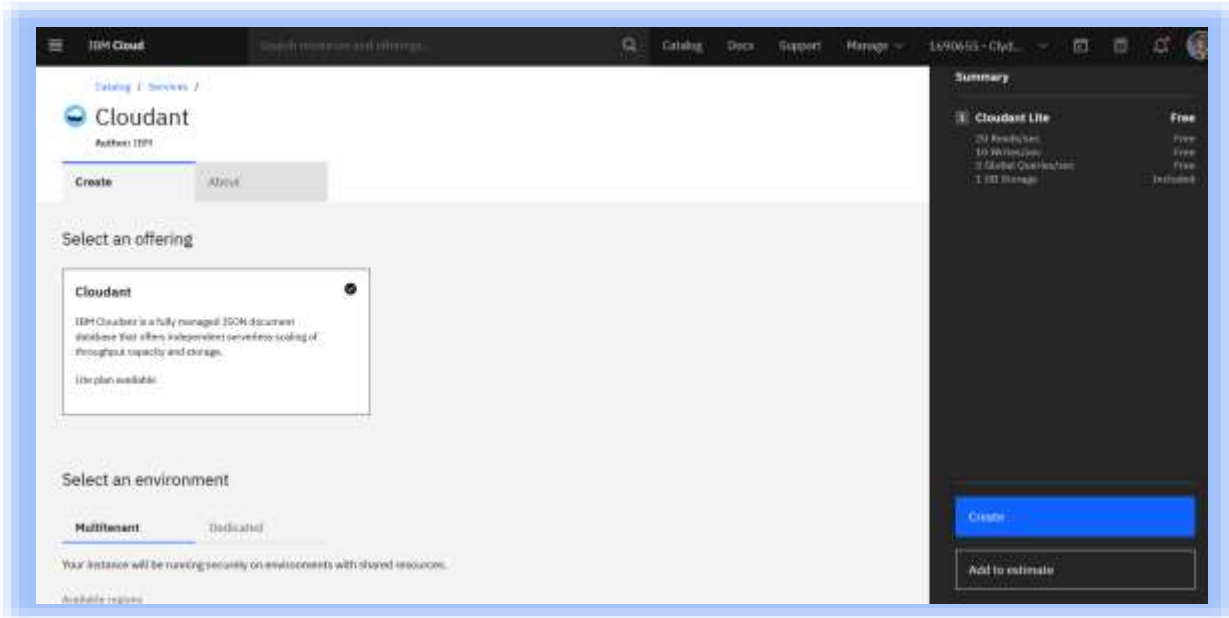
In the search bar, type Cloudant and then press the magnifying glass.



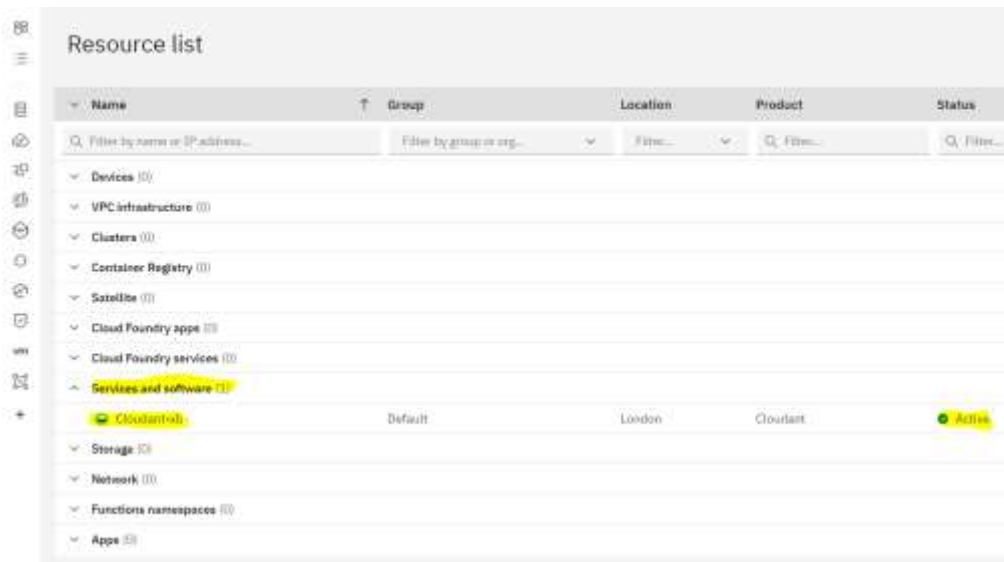
Select the Cloudant Service:



Select the **Cloudant Lite** service and then press “Create”



After a minute or two you should see the service has been provisioned under *Services and Software*. Make sure the service is in *Active* state.



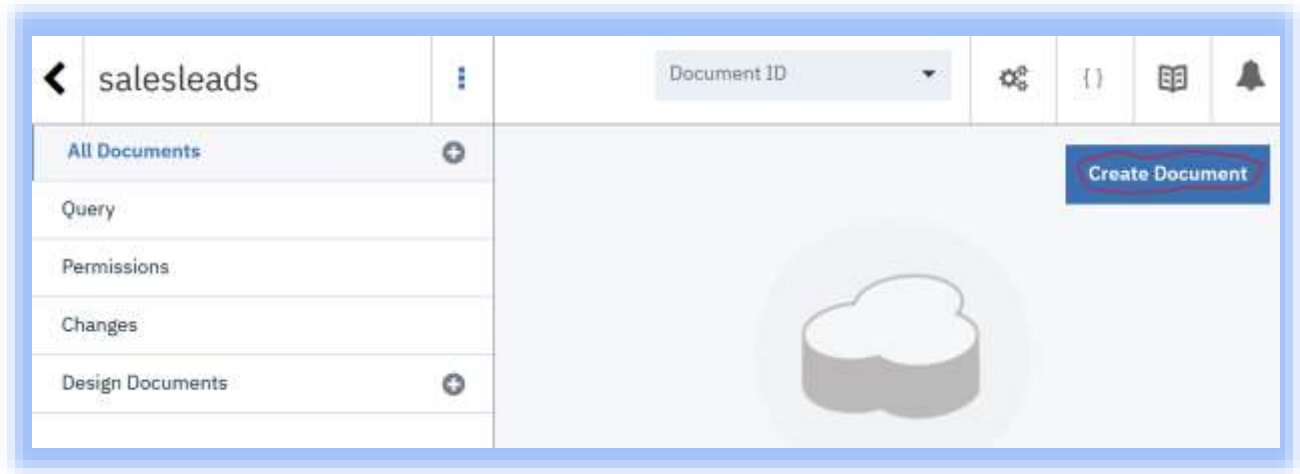
Click on your provisioned Cloudant instance and click *Launch Dashboard*. Select Databases and then *Create Database*:



Give the database the name of *salesleads* and make it non-partitioned. Then press Create.

A screenshot of the 'Create Database' modal form. The title 'Create Database' is at the top. Below it is a 'Database name' label and a text input field containing 'salesleads'. Underneath is a 'Partitioning' section with two radio buttons: 'Partitioned' and 'Non-partitioned', with 'Non-partitioned' being selected. Below the radio buttons is a link that says '> What is a Partitioned Database?'. At the bottom of the form are two buttons: 'Cancel' and 'Create'.

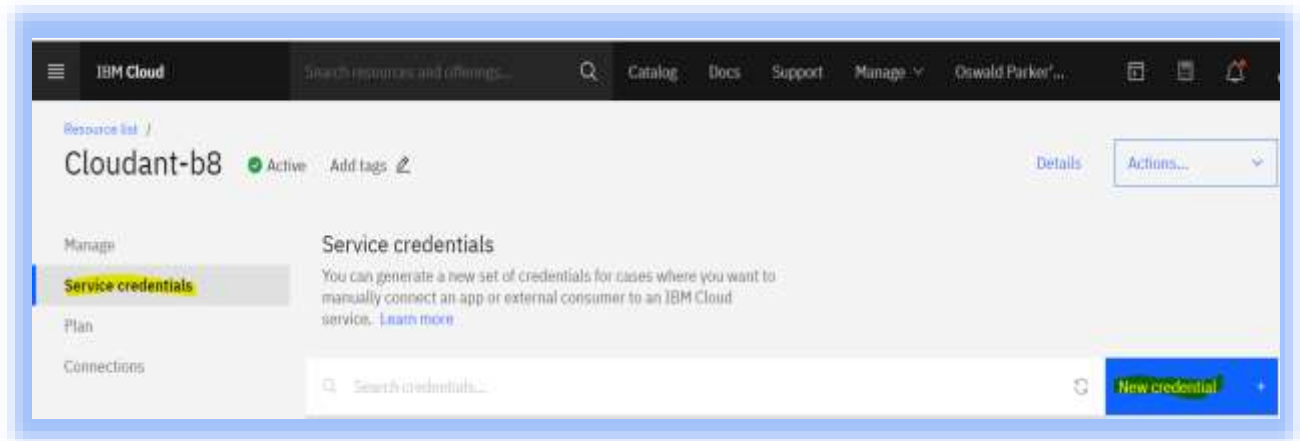
Select *Create Document*:



Paste the following JSON into the document and then press *Create Document*. You should see the message “*Document created successfully*”

```
{
  "_id": "28eb67f1e89deb5a058758433d36c446",
  "first_name": "Ned",
  "last_name": "Flanders",
  "job_title": "IT Support",
  "company": "IBM",
  "email": "ned@ibm.com",
  "phone": "87898977",
  "client_address": "101 Acasia Av",
  "client_city": "Springfield",
  "client_zipcode": "786786",
  "interest": "8",
  "followup": "Yes",
  "client_state": "North Carolina"
}
```

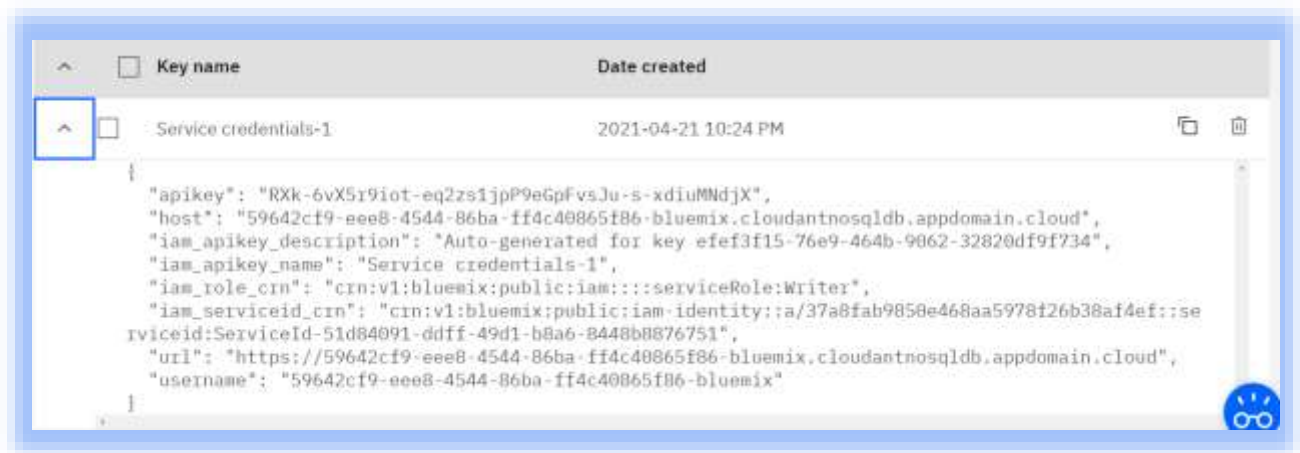
Back in IBM Cloud, click on your Cloudant instance and then click on Service Credentials. Select *New Credential*:



Give the credential name of *AppConnectCreds* and make the role *Writer*:

The screenshot shows a 'Create credential' dialog box. It has a title bar with a close button. The form contains two main fields: 'Name:' with a text input field containing 'AppConnectCreds', and 'Role:' with a dropdown menu showing 'Writer'. Below these fields is an 'Advanced options' button with a dropdown arrow. At the bottom of the dialog, there are two buttons: 'Cancel' and 'Add'.

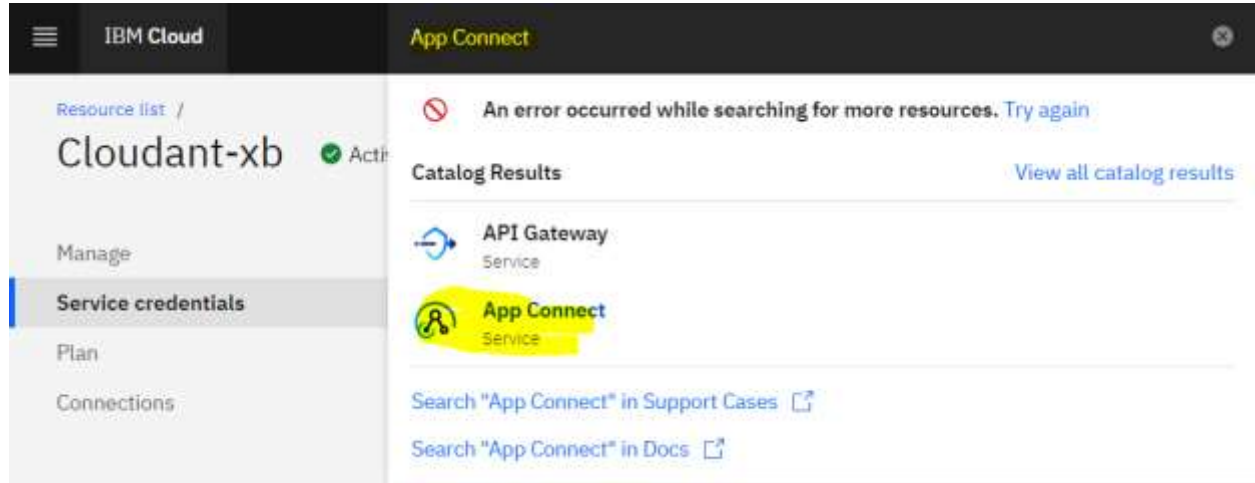
Press *Add*. Now expand the newly created service Credentials to view the details.



Copy these details into notepad so that you can use them in the next section.

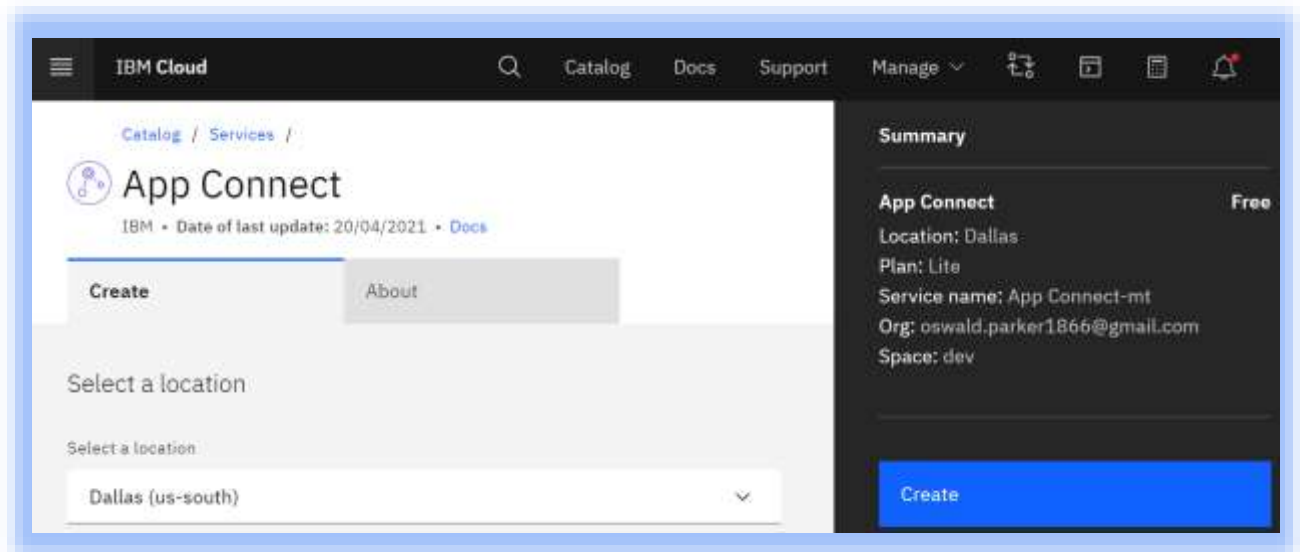
Provision an App Connect Instance

Still in IBM Cloud, in the search bar, type *App Connect*:

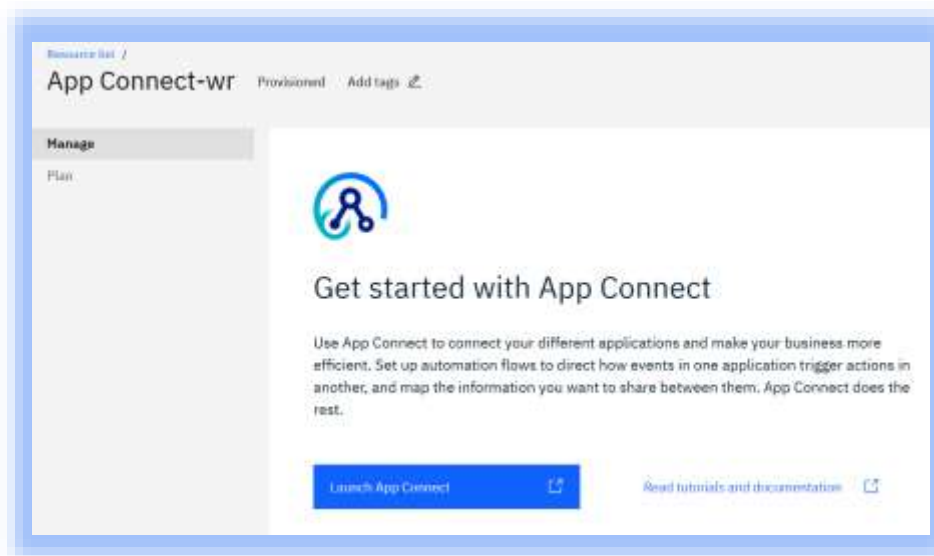


Select the *App Connect* service. Create an App Connect Lite (free plan) by pressing *Create*.


Note: the region for the App Connect instance needs to reside in the same region as the Cloudant database.

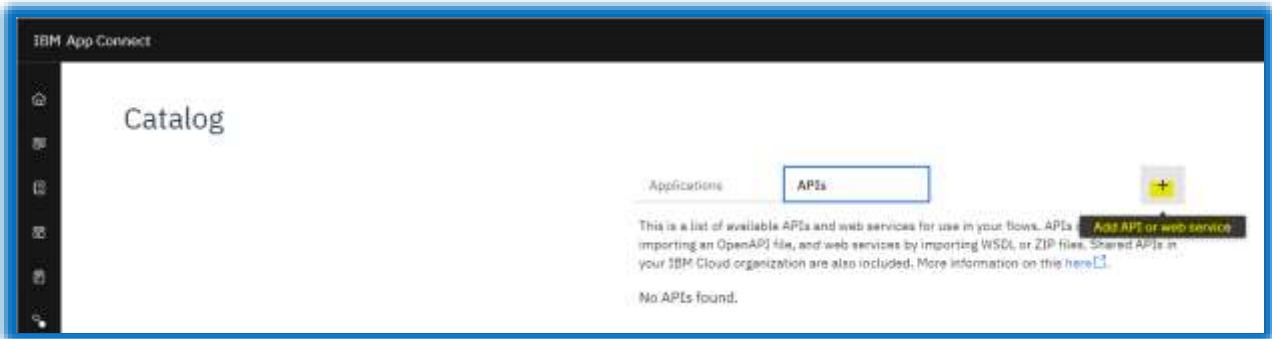


After a few minutes your App Connect instance will be provisioned. Select *Launch App Connect*.

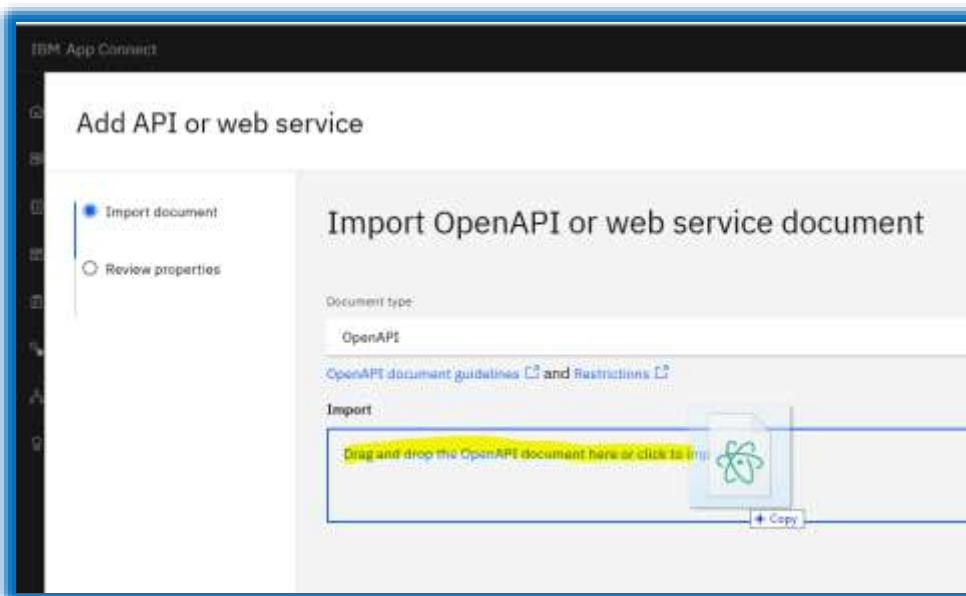


Import the RPA Bot API

On the left-hand side, select *Catalog*  and then select the *APIs* tab and then press the + button



Download https://rpapi.eu-gb.mybluemix.net/RPA_API.yaml to a local file. Drag it to the drop area:



Once dragged, you should see this:

The screenshot shows a dialog box titled "Add API or web service". On the left, there are two tabs: "Import document" (selected) and "Review properties". The main area is titled "Import OpenAPI or web service document". It contains a "Document type" dropdown menu set to "OpenAPI". Below this is a link: "OpenAPI document guidelines and Restrictions". Under the "Import" section, there is a text input field containing "RPA_API.yaml", which is highlighted in yellow.

Press *Next*. Keep the name *RPA API*. Press *Add API*. The API should be created without errors.

This screenshot is similar to the first one, but it shows the "RPA_API.yaml" text in the "Import" field with a small drop-down arrow on its right side. At the bottom of the dialog, there are three buttons: "Cancel", "Next" (disabled), and "Add" (active).

Select the drop-down arrow on the newly created *RPA API* and press *Connect*

Applications


APIs

+

This is a list of available APIs and web services for use in your flows. APIs can be added by importing an OpenAPI file, and web services by importing WSDL or ZIP files. Shared APIs in your IBM Cloud organization are also included. More information on this [here](#).

Q

Search APIs



RPA API

Not connected

Imported

^

This API presents an authenticated interface to invoke bot scripts both synchronously and asynchronously. Authentication is enforced through Basic Auth (username/password). Authentication is managed through a nodejs server script which has to be de

Connect

RPA API actions

The action is the task you want your flow to complete.

▼ Rpa api

runAsyncScript
script

runSyncScript
script

Enter a username of *rpaserviceaccount* and a password of *RPA4Ever!*



RPA API
Not connected



Imported

Connect to RPA API

User name: (optional)

rpaserviceaccount

If using basic authentication, specify a user name that is authorized to access RPA API.

Password: (optional)

.....



If using basic authentication, specify a password for the user name.

API key: (optional)



Specify the API key that you use to access RPA API.

Allow self-signed certificates: (optional)

Set to true or false. Set to true if you want to be able to accept self-signed certificates that are trusted and used only in a non-production environment. The default is false.

Override the base path for the API: (optional)

Specify an override value for the base path in the OpenAPI definition.

Override the host name and port of the RPA API server: (optional)

If you chose to connect via a private network, then you must set this and you must specify the host name and port in the format <http or https>://<host name>:<port>. Otherwise, this field is optional.

Network name:

Specifies the name of the network that App Connect will use to access your system. Required only if connecting to a system in a private network.

<Not selected (optional)>



Cancel

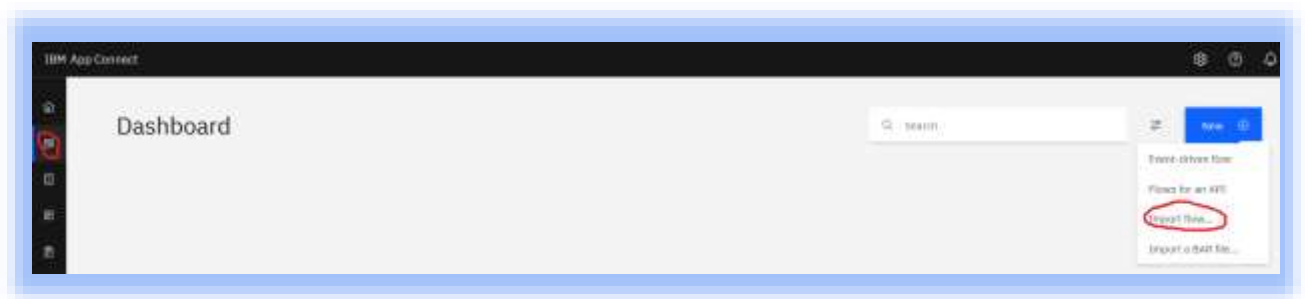
Connect

Press *Connect* again.

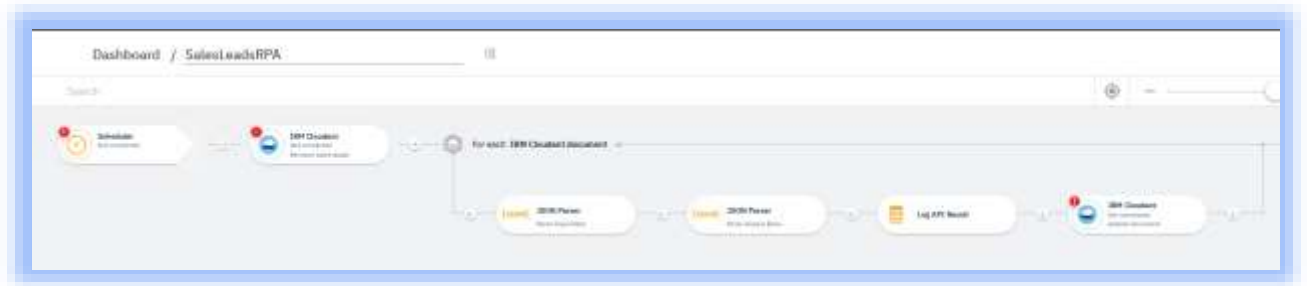
An API Account should now have been created. We will use this in the next step.

Create an App Connect Flow

On the left-hand side, select *Dashboard*  and then on the right-hand side, press *New->Import Flow...*



Download <https://rpapi.eu-gb.mybluemix.net/SalesLeadsRPA.yaml> to a local file. Drag this file to the drop area and press *Import*. The API Flow should be imported. You should see several errors that we will rectify in the next step.



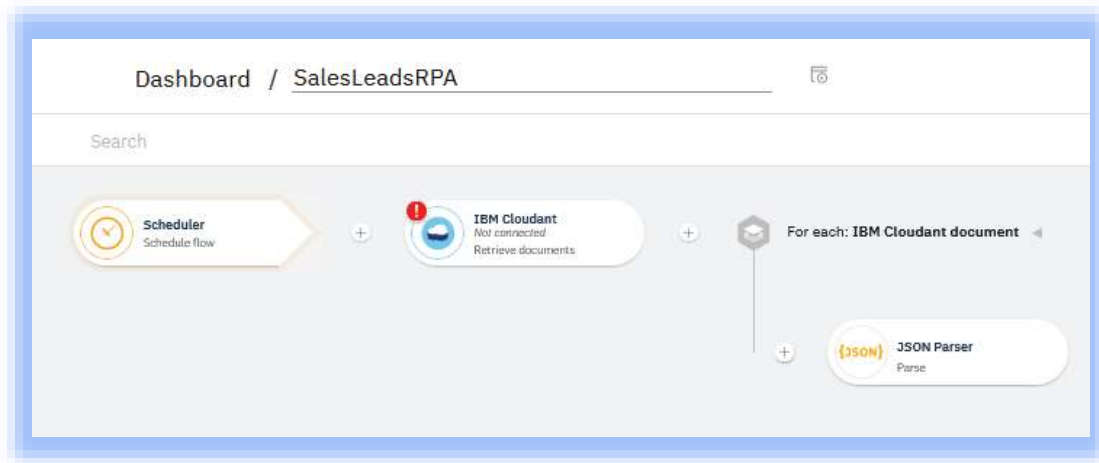
Fix Errors in the Flow

Click on the Scheduler inside the flow. Click the down arrow and add a new scheduler account and then use the default schedule below:

The 'Schedule flow' dialog box is shown with the following configuration:

- *Select schedule type: Repeating interval
- Run every: 23 Hour
- on: ☐ Mon ☐ Tue ☐ Wed ☐ Thu ☐ Fri ☐ Sat ☒ Sun
- Timezone: UTC
- ☒ Also run the flow when it's first switched on.

The first error should disappear:



Click on the *IBM Cloudant* node. Click on *connect*. Enter the credentials of the Cloudant instance you created earlier. Note: *password* is a legacy field and is not required with newer versions of Cloudant.

Retrieve documents

Connect to IBM Cloudant

Cloudant user name: (optional)

59642cf9-eee8-4544-86ba-ff4c40865f86-bluemix

If the service instance uses legacy credentials for authentication, provide your user name.

Cloudant password: (optional)

|

If the service instance uses legacy credentials for authentication, provide the associated password.

Cloudant host: (optional)

59642cf9-eee8-4544-86ba-ff4c40865f86-bluemix.cloudantnosqldb.appdomain.cloud

If the service instance supports Identity and Access Management (IAM) authentication, provide the host name of the instance.

Cloudant API key: (optional)

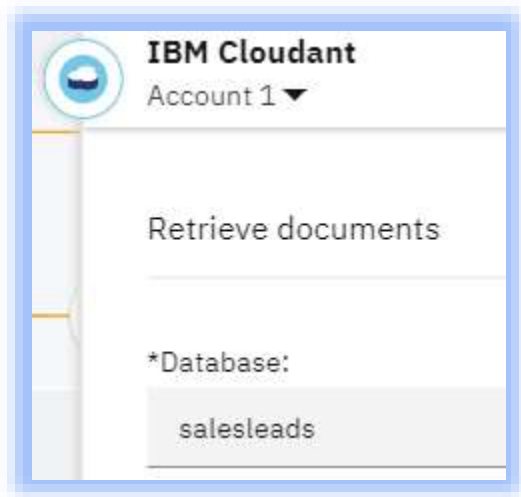
.....

If the service instance supports IAM authentication, provide your API key.

Cancel

Connect

If you have more than one Cloudant database, select *salesleads*

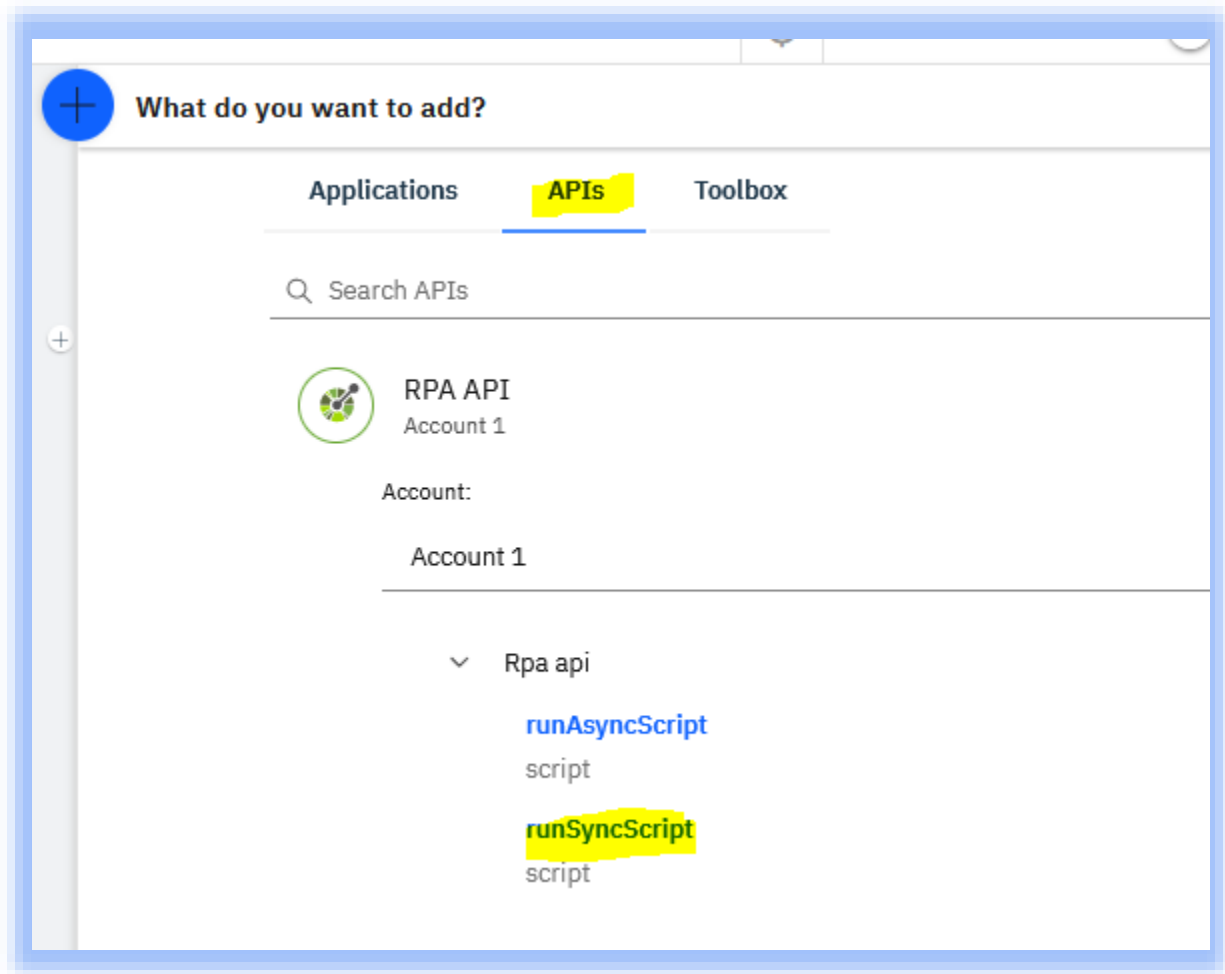


The Cloudant database should now have no errors.

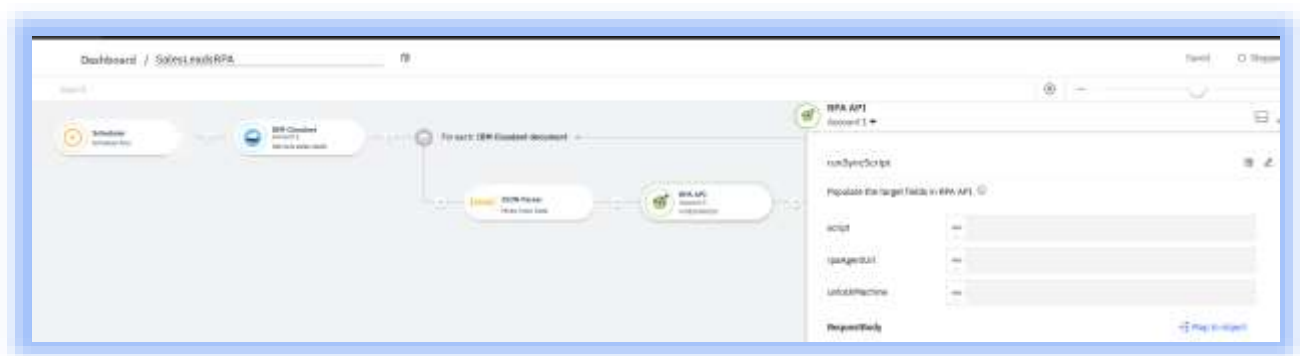
Back in the flow, click the '+' symbol in between the two JSON Parser nodes.



Select APIs and then select the RPA API and then *runSyncScript*



You should see the following:




Set *script* to `sales_lead_automation_API`

Set *RPAAgentUrl* to `LOOPBACK`

Set *unlockMachine* to `false`

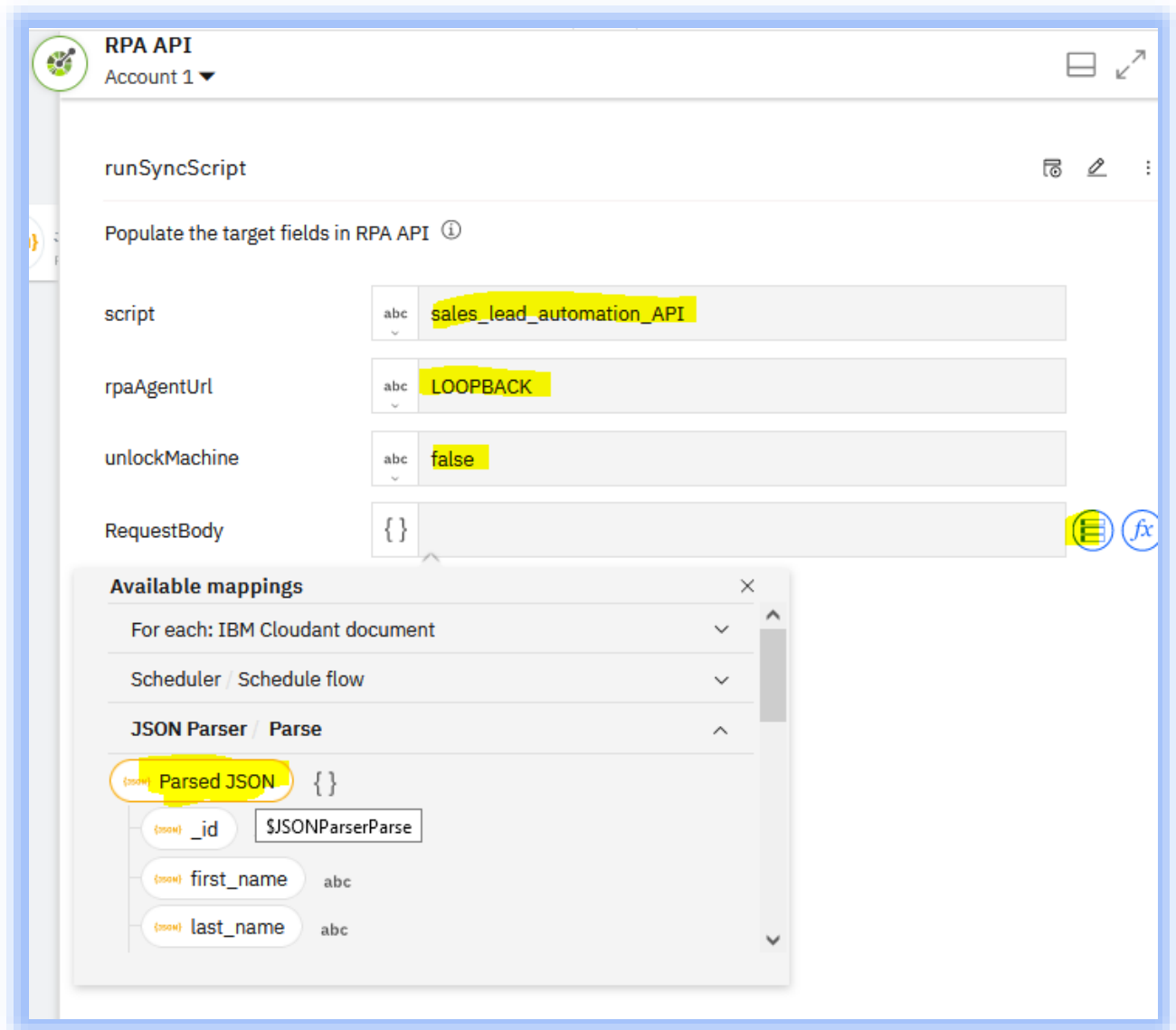
Click Map to Object:

 Map to object

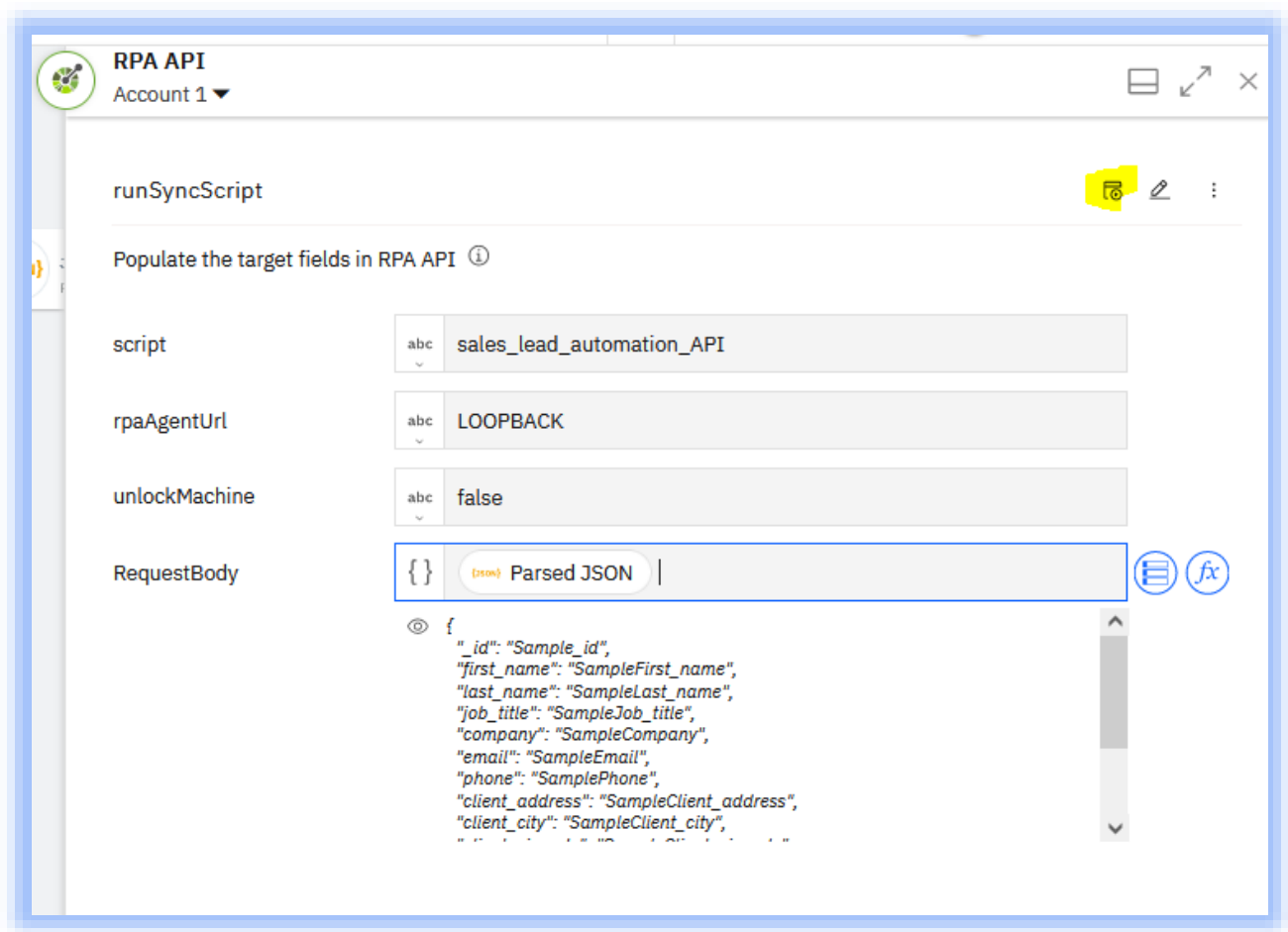
Click inside the *RequestBody* field and then click *Insert a mapping*:



Select JSON Parser->*Parsed JSON*:



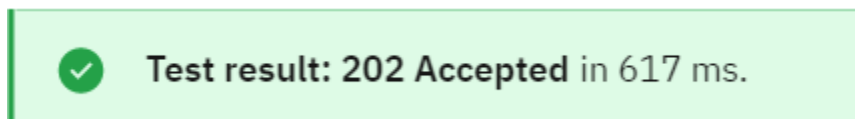
You should now see the following:



To test this node, press the *play* button and then press *continue*:



After a moment you should see:



This verifies that the API is working.

Note that for this lab we are using a virtual RPA agent called LOOPBACK. To see the API invoked against this RPA server view this video:

https://youtu.be/1K_Zja-okQg

Map API Response

Open the second JSON Parser node (Parse Output Data):



Delete the existing contents. Press Insert a mapping.



Insert the output from the RPA API node you created in the previous step:

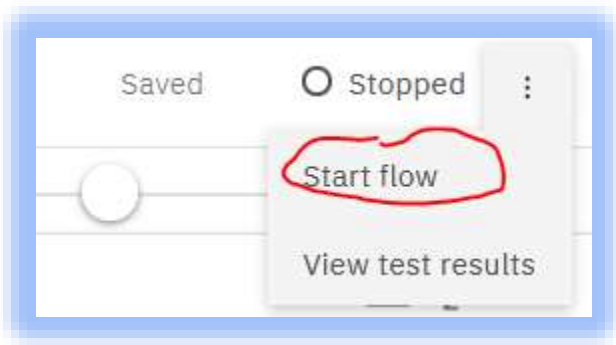


Expand *Output Schema* then hit *Generate Schema*.

Run the App Connect Flow

All the errors in the flow have now been fixed. We can run the entire flow.

Choose *Start flow* under the vertical ellipsis at the top right of the flow




The flow will run. It will perform the following:

1. Read a document from Cloudant
2. Invoke the RPA bot using the document as bot parameters
3. The bot fills in a web form using the passed parameters.
4. Once executed, the bot passes back the results where they are written to Cloudant
5. If there are several Cloudant documents, the process repeats from step 1

Note: If you want to make further edits to the flow, you must stop the flow first.

View App Connect Logs

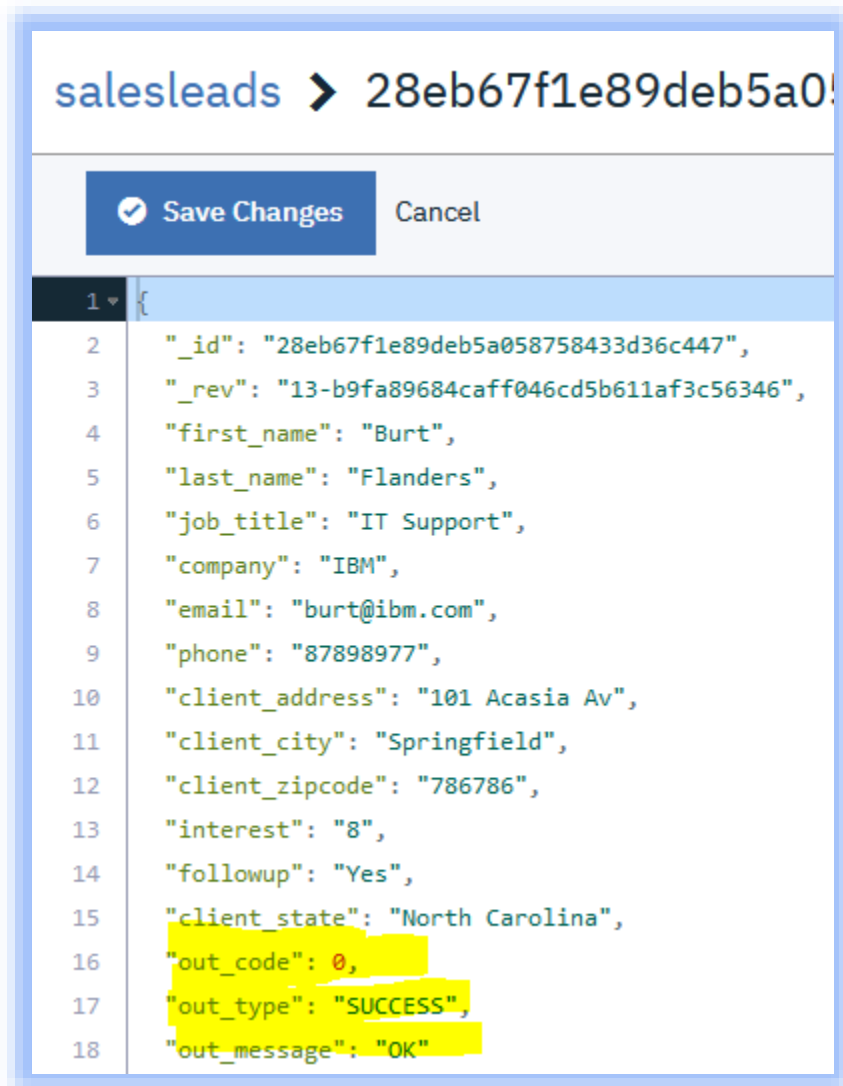
To view the result, select the Logs  icon. After a few minutes you should see `Flow SalesLeadsRPA completed successfully` in the log

Logs (9)


	Event time (UTC)	Message
▼	2021-09-21 10:45:53.093	Flow SalesLeadsRPA completed successfully.
▼	2021-09-21 10:45:53.003	Processing complete
▼	2021-09-21 10:45:52.441	{"out_code":0,"out_type":"SUCCESS","out_message":"OK"}

View Processing Result in Cloudant

To view the processing results in Cloudant, go back to your Cloudant database and open the document you created earlier. The processing results from the bot have been added to the Cloudant document. This indicates that the bot completed and has added three output fields.



Stop the flow

Go back into App Connect *Dashboard* , Within the SalesLeadsRPA vertical ellipses menu, select *Stop Flow*.

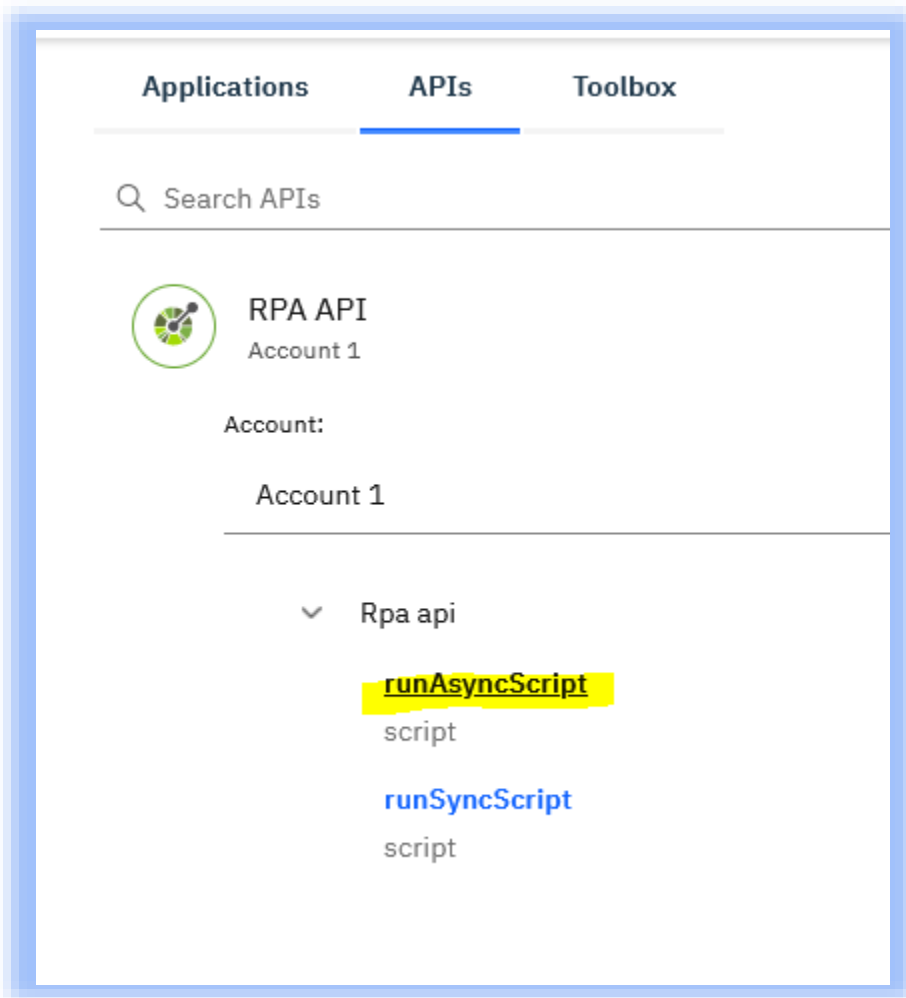
Advanced Exercise – Asynchronous invocation of RPA API

The API specifies both synchronous and asynchronous invocation: <https://rpapi.eu-gb.mybluemix.net>

In this advanced exercise you will change the API invocation to run asynchronously.

Open the App connect flow you created earlier. Stop the flow if it is already running. Select the RPA API node and delete it.

Recreate the RPA API node, but this time select *runAsyncScript*. See below:



Enter the following details:

Mode: INVOKE
Script: sales_lead_automation_API
RPAAgentUrl: LOOPBACK
UnlockMachine: false

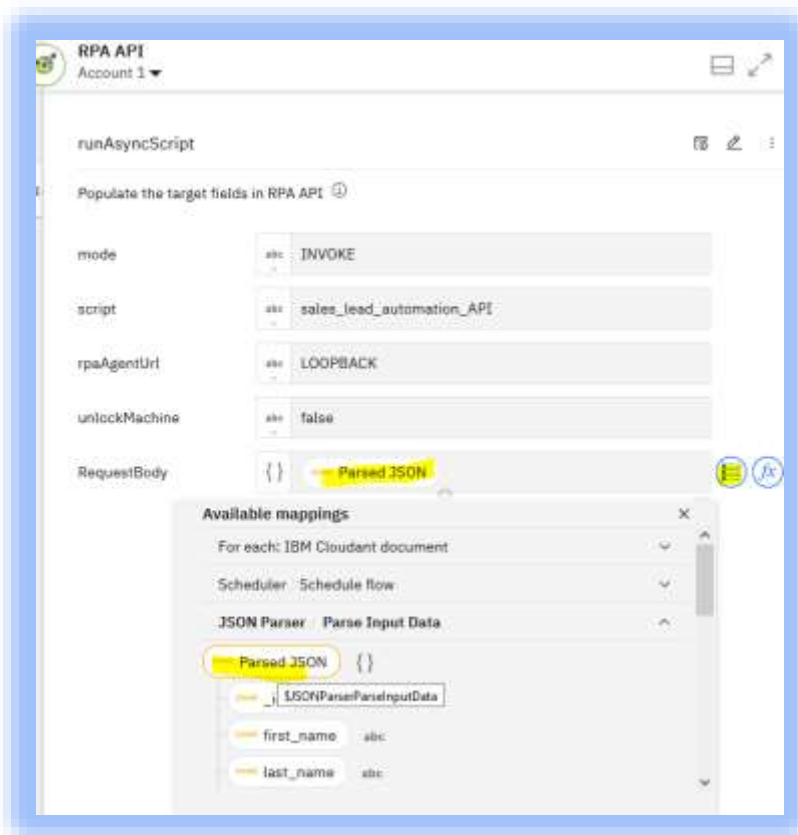
Click *Map to Object*:

 Map to object

Click inside the *RequestBody* field and then click *Insert a mapping*:



Select JSON Parser->*Parsed JSON*:



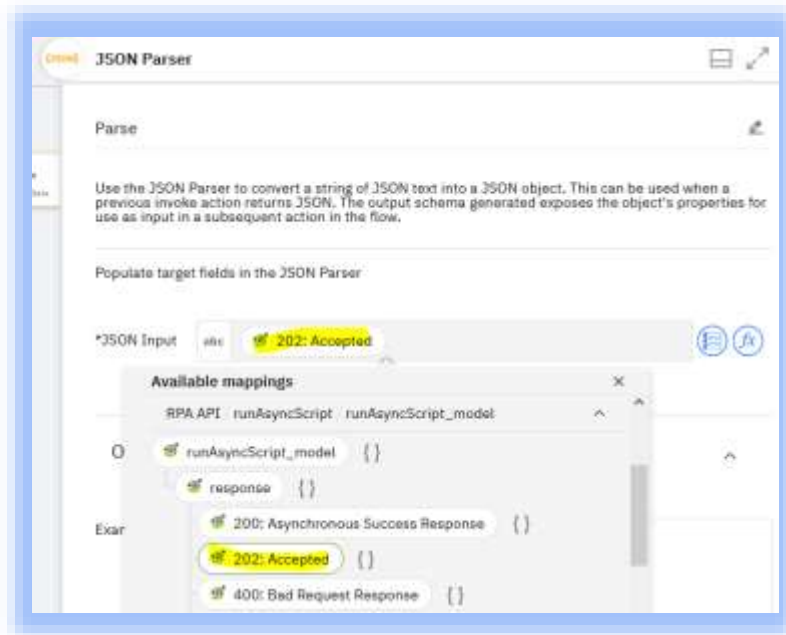
Now change the API output mapping. Open the second JSON Parser node (Parse Output Data):



Delete the existing contents. Press Insert a mapping.



Insert the output from the RPA API node you created in the previous step:



Expand *Output Schema* then hit *Generate Schema*.

Run the flow.

To view the result, select the Logs  icon. After a few minutes you should see Flow SalesLeadsRPA completed successfully in the log

View the results in the database. You should see that the results are PENDING.

What are the advantages and disadvantages of using asynchronous calls over synchronous?

Conclusion

In this lab, you built an App Connect flow which connected a Cloudant database to an RPA bot. The RPA bot was connected through an API. The bot took data from Cloudant and inserted it into a legacy user interface.

Congratulations. You have completed the RPA with App Connect lab!

THIS COMPLETES THIS HANDS-ON LAB