Workshop Creator Workflows

IHub - Gerador de Spokes

servicenow

Table of contents:

Integration Hub: Spoke Generator

Introduction

IntegrationHub is the central place on the platform for consuming, creating, and managing integrations on your ServiceNow instance. In the event that you are seeking to connect a workflow to an external system, but ServiceNow or ServiceNow technology partners do not offer an out-of-the-box integration (Spoke) for it, you can easily create your own. IntegrationHub provides numerous Action Steps to connect your workflow to external systems that use protocols such as REST, PowerShell, SSH, etc. In this lab, we will focus on REST integration.

Goal

In this lab, we will showcase the new 'Spoke Generator' capability. This feature enables you to generate new spokes and spoke actions for third-party APIs that adhere to the OpenAPI specification. You will be using a provided YAML file.

> NOTE
>
> A YAML file for OpenAPI specs is a user-friendly configuration file that defines the structure, endpoints, and details of a RESTful API. It makes it easier for developers to document and communicate the API's design. The spoke generator can automatically utilize this YAML file to create spoke actions.

This innovative feature, the "Spoke Generator," significantly reduces the time required to create a new integration.

Use Case

ACME Inc. currently uses a third-party system for managing visitor access and authorization to their buildings, which is overseen by a Security and Property Management company. ACME Inc. aims to enhance the experience for both visitors and employees. They utilize ServiceNow and intend to automate the process of verifying visitor access and printing badges through ServiceNow workflows. ACME Inc. has developed a modernized experience using the ServiceNow platform, allowing visitors to independently check in via kiosks located at the reception of each building.

However, there are no pre-built integrations available for the Visitor Access application, which is hosted and managed externally to the ServiceNow Platform. As ServiceNow developers, our task is to create this integration (Spoke) so that we can retrieve information from this remote system through a workflow triggered when a visitor checks in at the kiosk.

In this scenario, you had a meeting with the administrator of the third-party visitor access app who has supplied you with the following API documentation: API Documentation Link . You can click the link to gain an understanding of how this API functions.

This API is relatively straightforward. As you can observe, it consists of only one API method, 'checkUser,' and this method necessitates three inputs."

Field	type
firstname	string
lastname	string
dateofbirth	YYY-MM-DD

By providing the user information, the API will return a message indicating whether a user exists or not. If the user does exist, it will also provide additional information that you will need to utilize within a ServiceNow workflow. You can refer to the example of the response returned by the API for clarification


```
{
"code": "0",
"message": "User exists",
"user": {
   "guest_title": "Guest",
```

```
    "phone": "123-456-7890",
    "host_name": "Jane Smith",
    "host_id_number": "987654",
    "host_email": "jane@example.com",
    "guest_email": "john@example.com",
    "building_location": "Building A",
    "access_expiration": "2023-12-31"
         }
}
```

In the JSON response, you will receive a 'code' indicating success (0), a 'message' confirming the user's existence, and comprehensive 'user' information, which includes guest title, phone number, host name, host ID number, email addresses, building location, and access expiration date. If the user does not exist, you will receive a code 1. In our particular scenario, the user data obtained from the response will be utilized to print a badge for the visitor. (The badge printing process is addressed in another lab using our RPA Hub technology.)

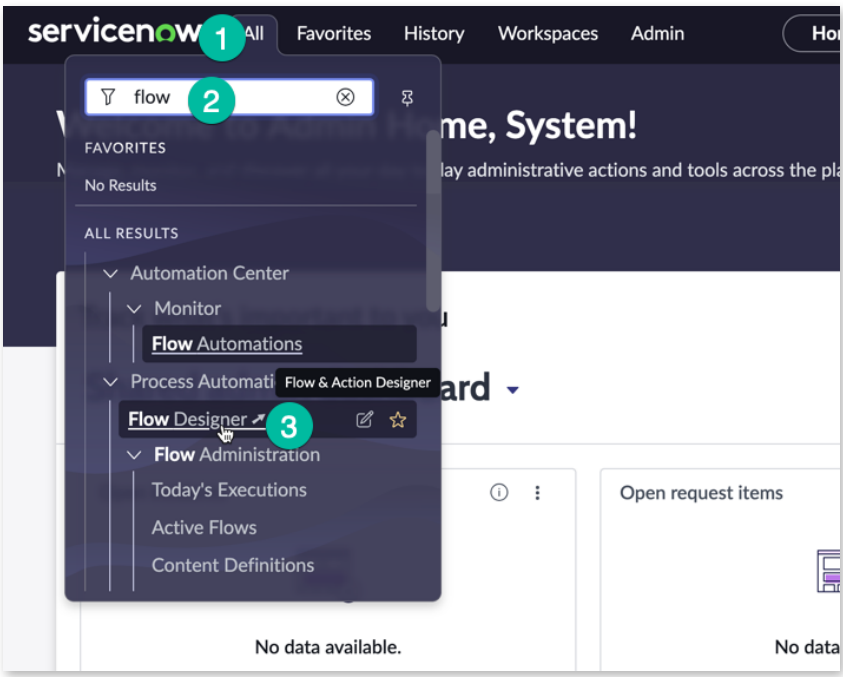
Create the Spoke

Overview

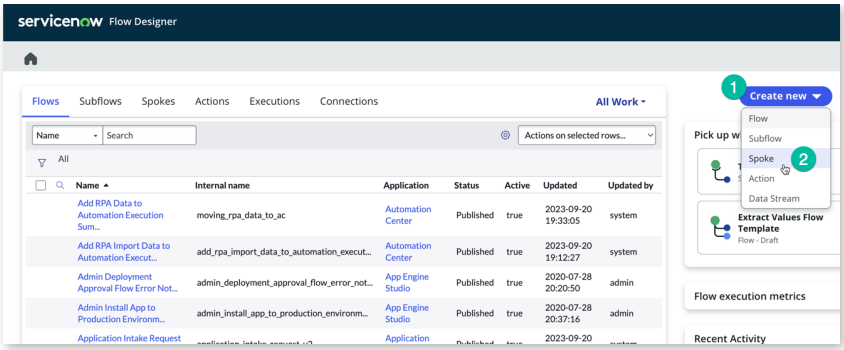
In this exercise, you will create a custom Integration Hub Spoke.

Instructions

1. Log in to your instance, then, on the main page, click **All** (1). Next, type **flow** (2) and click **Flow Designer** (3) to open the Flow Designer UI.

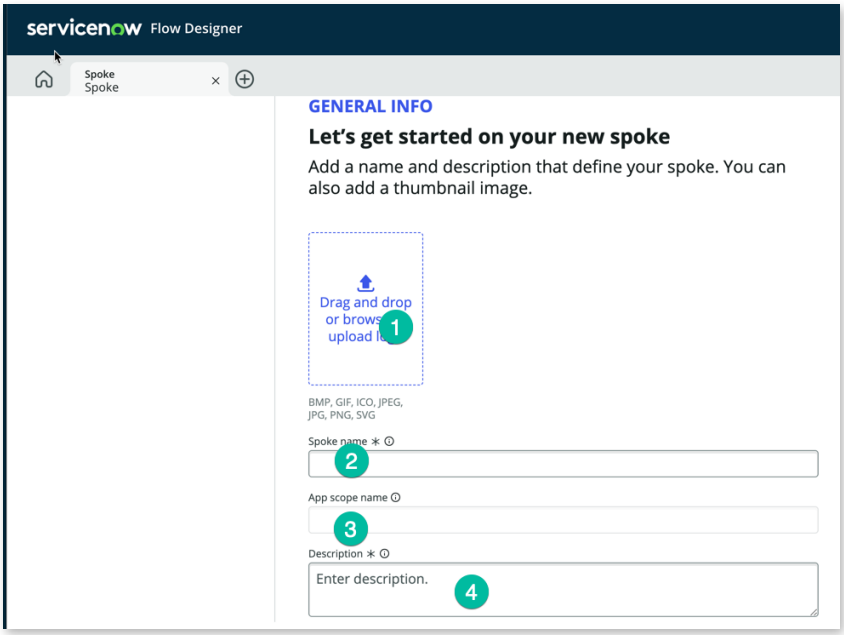


2. Once you are in the **Flow Designer** UI, to access the Spoke Generator, select **Create New** (1) (located on the right-hand side of the screen), and then click on **Spoke**.

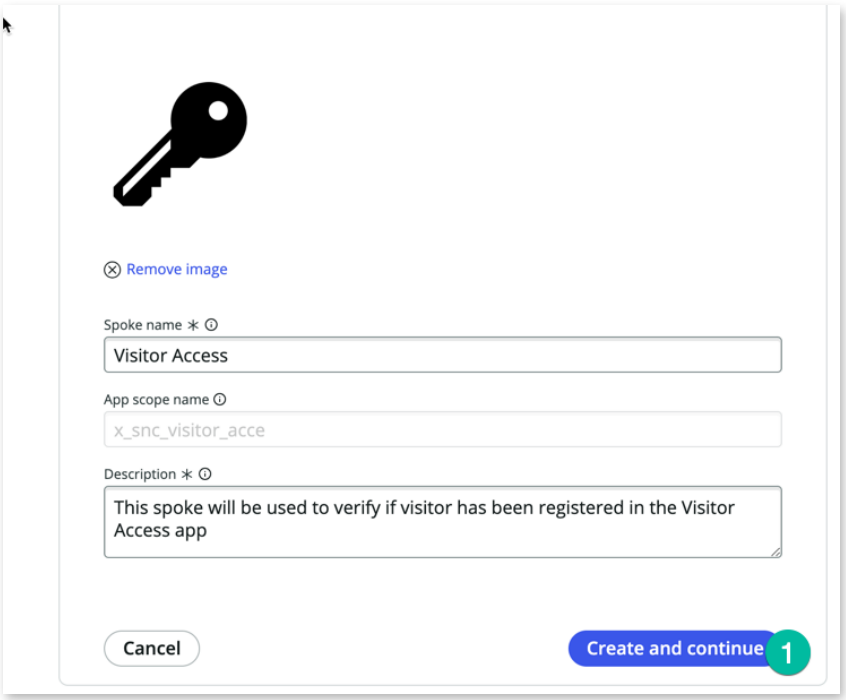


3. The Spoke Generator will request the following information from you: a thumbnail image (1) (you can upload a thumbnail image to serve as your Spoke logo; feel free to find a free image on Google Images and upload it here), a Spoke Name (2), and a Description (4).

Field	Value
Spoke Name	Visitor Access
App Scope Name	This field is generated automatically from the Spoke Name
Description	This spoke will be used to verify if visitor has been registered in the Visitor Access app

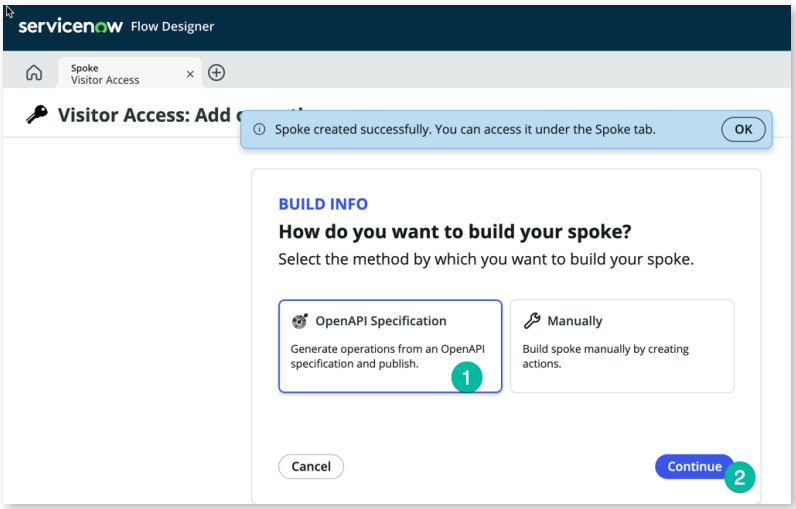


4. Once you have set the value, click Create and Continue (1) as shown below:

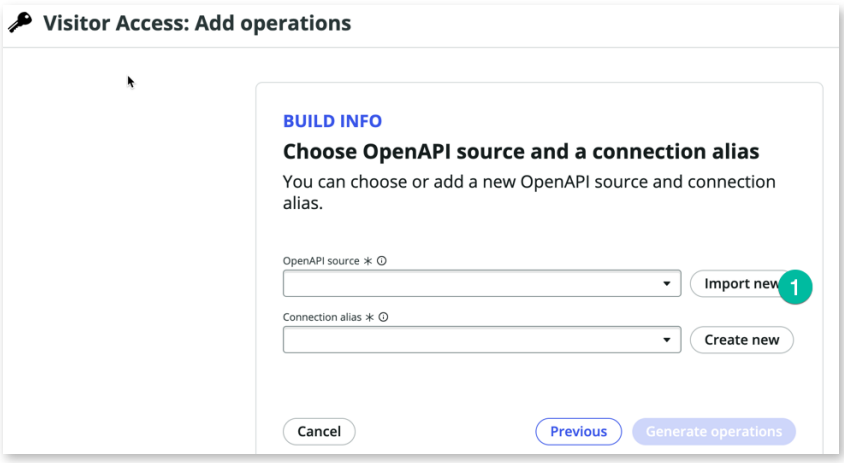


5. On the next screen, you will be prompted to select the method you wish to use for creating your new spoke. We intend to utilize the OpenAPI Specification method, as we have been furnished with the YAML file that describes the API and adheres to the OPENAPI Specification.

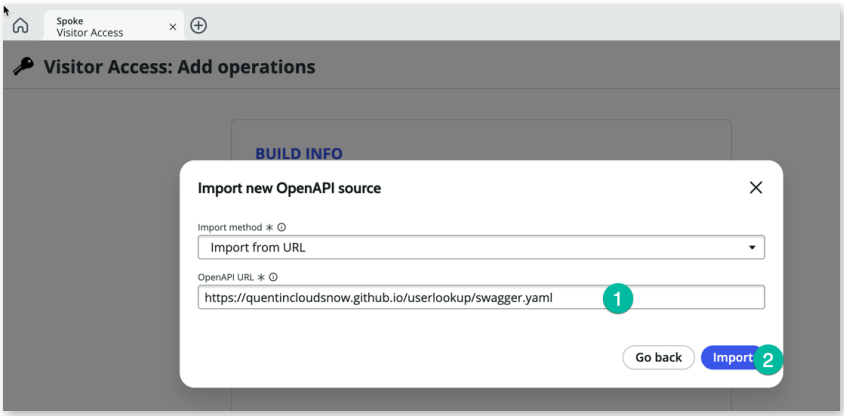
- Select **OpenAPI Specification** (1) then click **Continue** (2)



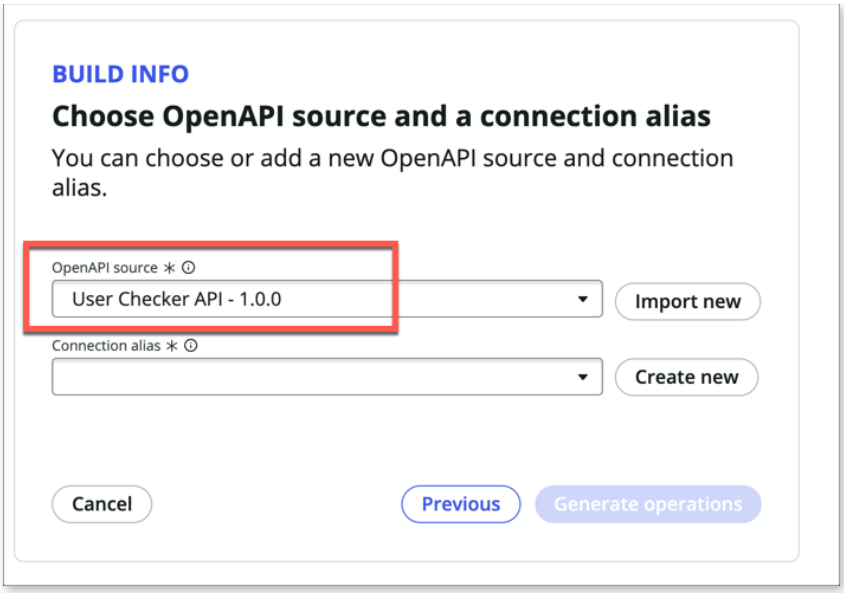
6. On the next screen, **Visitor Access: Add operations**, this is where you can provide the YAML file. Click **Import New** (1), and then provide the url to the YAML file, copy URL from the link here Swagger YAML File .



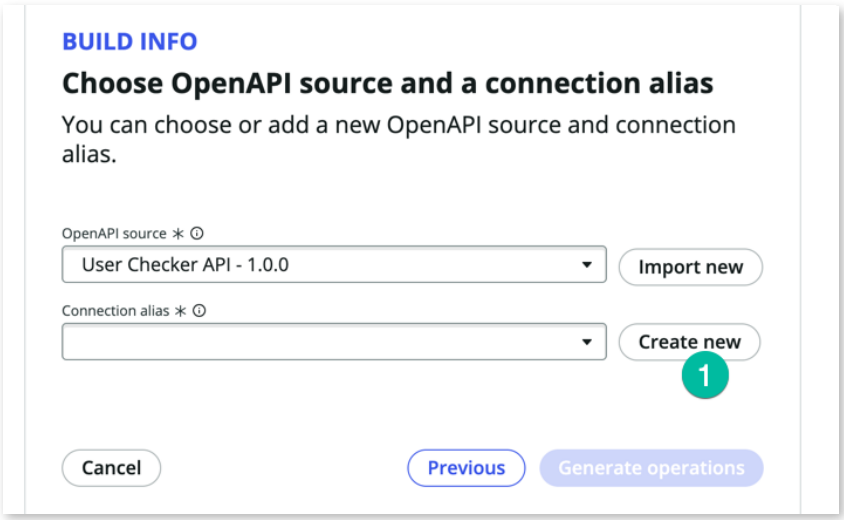
7. Paste the URL copied in your clipboard in the previous step, then paste it in the **OpenAPI URL** field (1), then click **Import** (2)



8. Once the import is done you should see something similar to this:

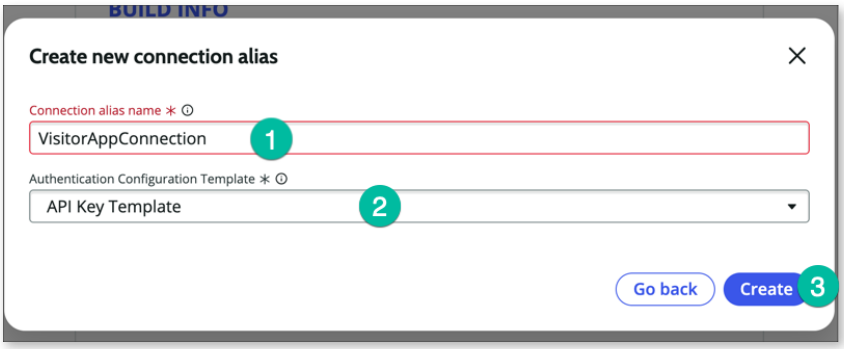


9. Then click on **Create New** next to the **Connection Alias field** (1)

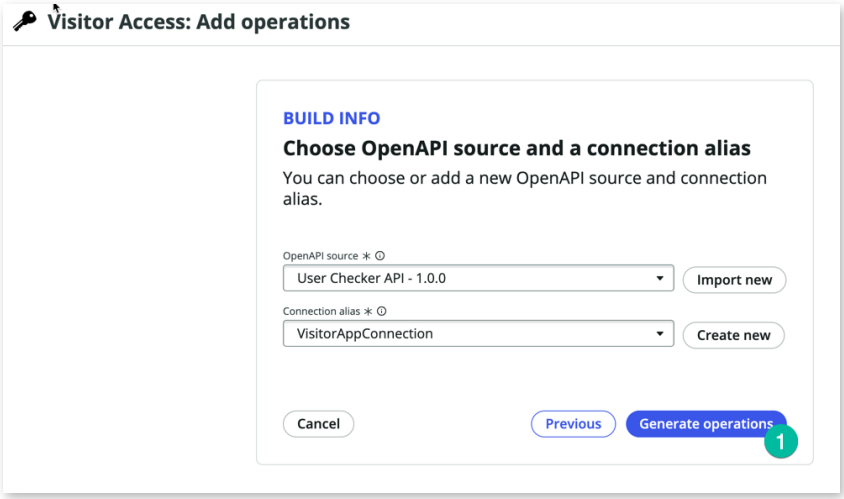


> NOTE
>
> In ServiceNow, a Connection Alias is a configuration setting used to establish and oversee connections with external systems. It functions as an abstraction layer for connecting to external systems and streamlines the integration process within ServiceNow workflows and other components. Normally, when connecting ServiceNow to an external system, you must configure the URL of the endpoint (the third-party system) and specify how to authenticate with it. This is accomplished through Connection and Credentials settings in ServiceNow. In practice, it is essential to engage in discussions with the administrator of the remote system and coordinate with the security team before initiating this configuration.

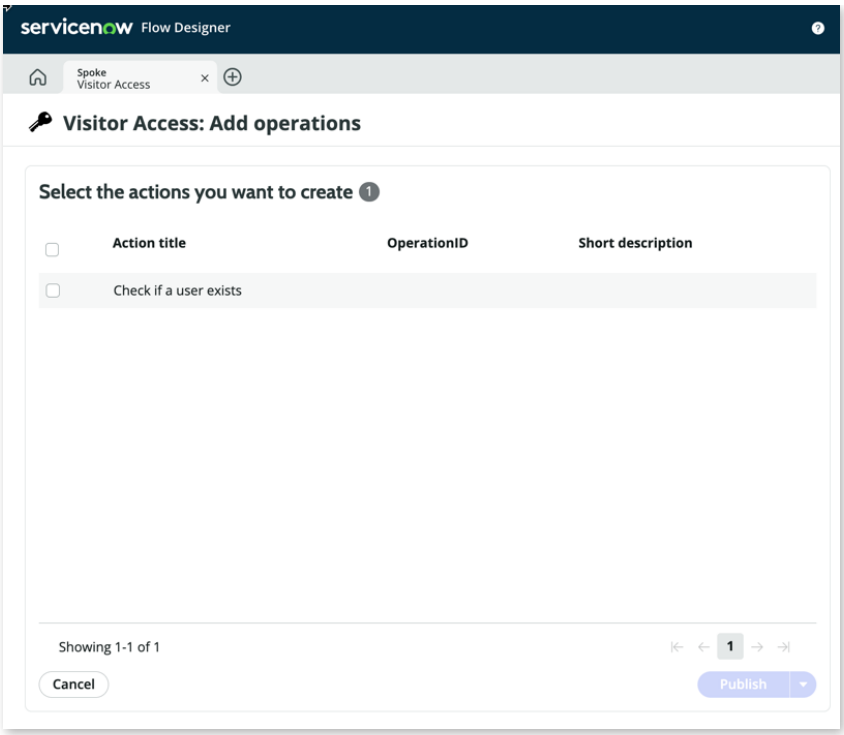
10. In the **Connection alias name** (1) type **VisitorAppConnection** and keep the **Authentication Configuration Template** with the default value **Api Key Template** (2), then click **Create** (3)



11. Click **Generate operation** (1)

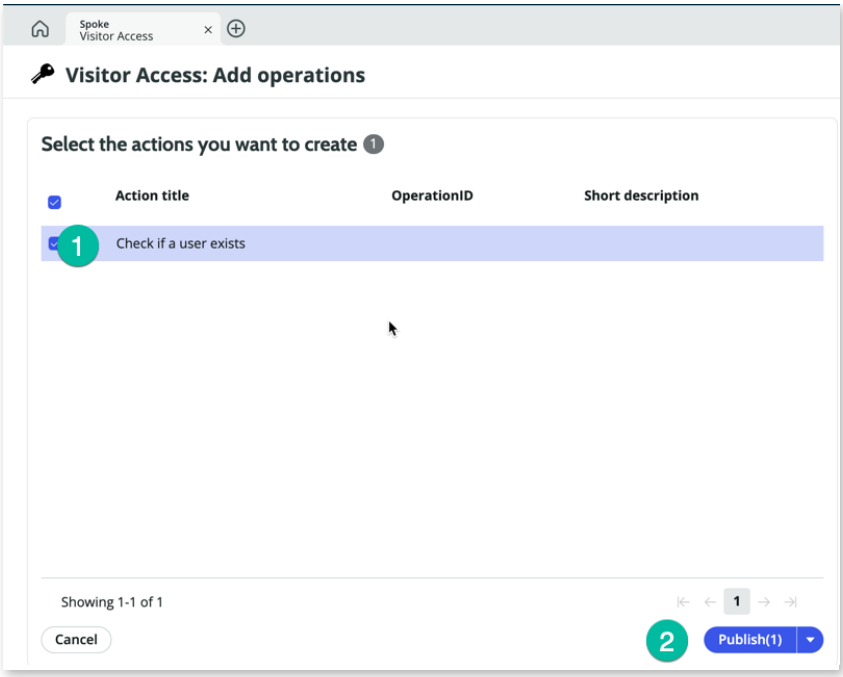


12. The system will then prompt you to select which Spoke Action you want to create as shown below

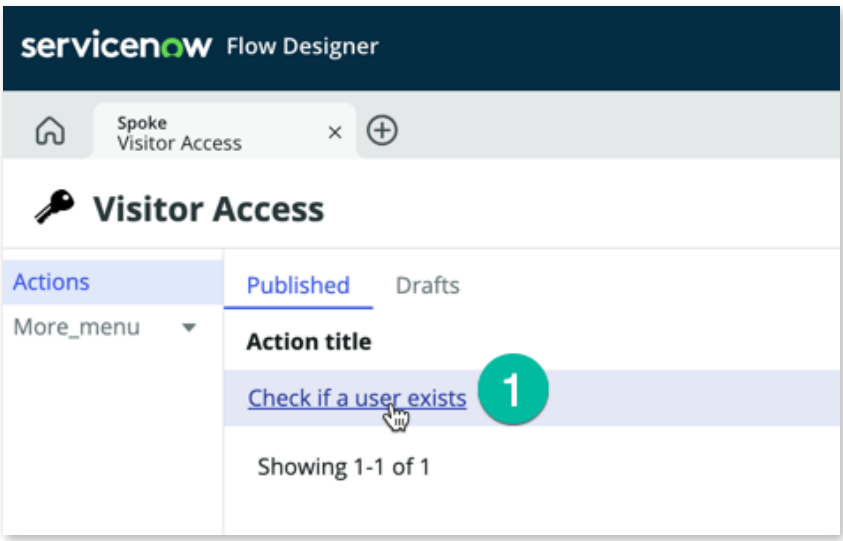


> **NOTE**
>
> For this lab, we are utilizing a very basic API designed explicitly for lab and educational purposes. This API consists of just one method, which is why only one action is visible. In real-world scenarios, most commercial applications you endeavor to integrate will have dozens or even hundreds of methods within their API. You will have the option to choose the methods you wish to utilize from ServiceNow and create Spoke Actions for them.

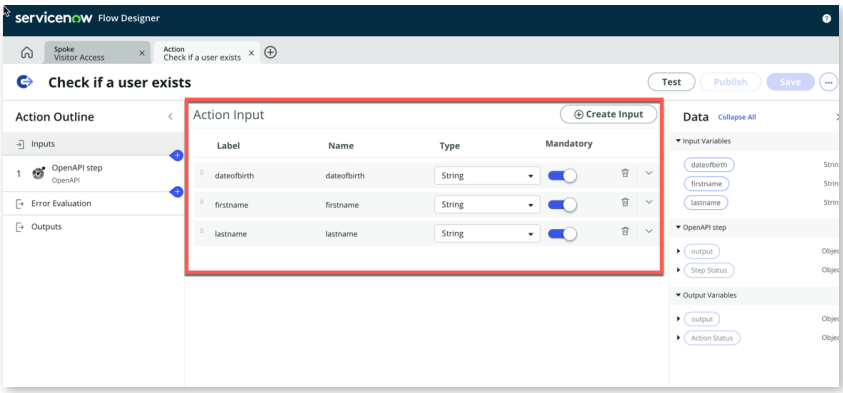
13. Select the **Check if a user exists** (1) Action then click **Publish** (2)



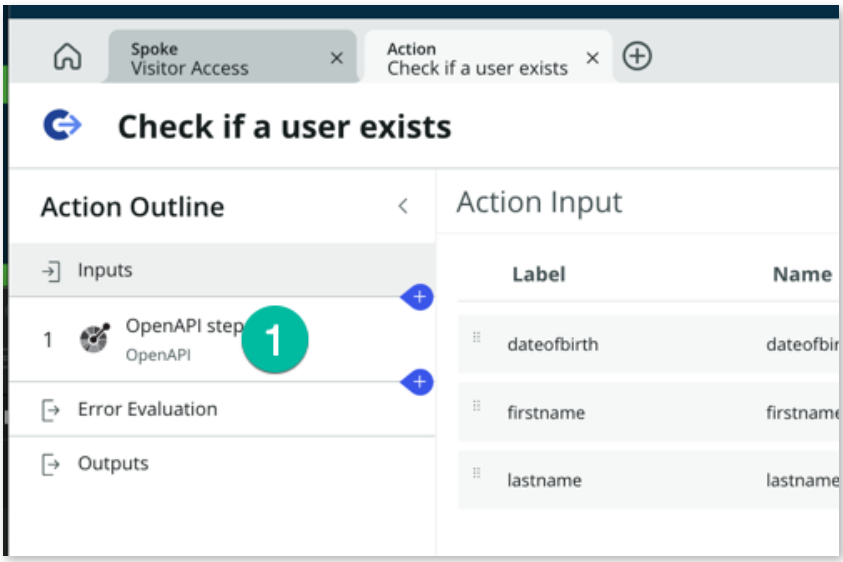
14. Click the newly created Spoke Action **Check if a user exists** (1) this will open the Action editor in Flow designer so we can inspect it



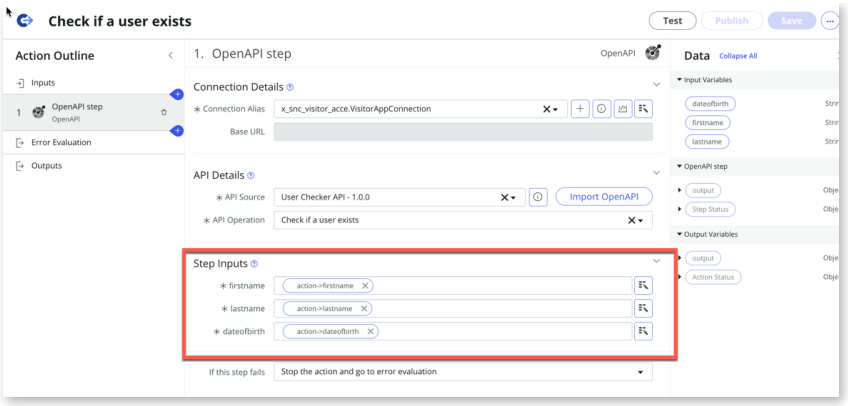
15. Notice the Action Input section, the Inputs for that Spoke action was created automatically



16. Click the OpenAPI Step (1)

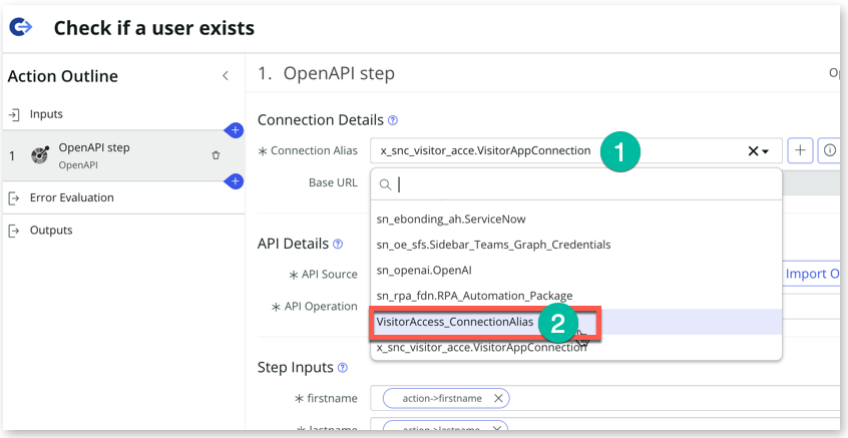


17. Notice the step inputs, they are using the Action inputs, so the values from the Action Inputs will be passed as parameters when the API Call is made to the external system.



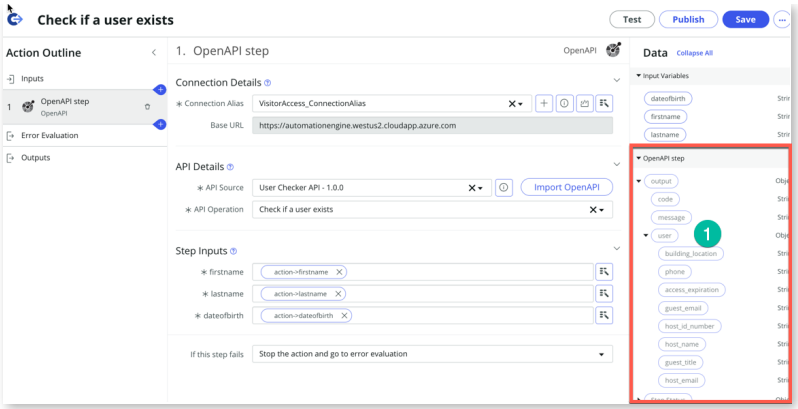
18. We need to update the Connection Alias in the Connection Details section. We have precreated in this lab instance a Connection record that point to the right API End point. We are going to use this one.

19. Click on the **Connection Alias field** (1) then select **VisitorAccess_ConnectionAlias** (2)



20. Notice the **Base URL** field was updated and displays the URL for the API End point. Those connection alias records are typically managed a by Security team or user with higher privileges.

- On the right hand-side notice the outputs available from the OpenAPI Step, expand the **user** (1) section as shown below:



Those are all the values that we can retrieve from the external app and use in a ServiceNow Workflow.

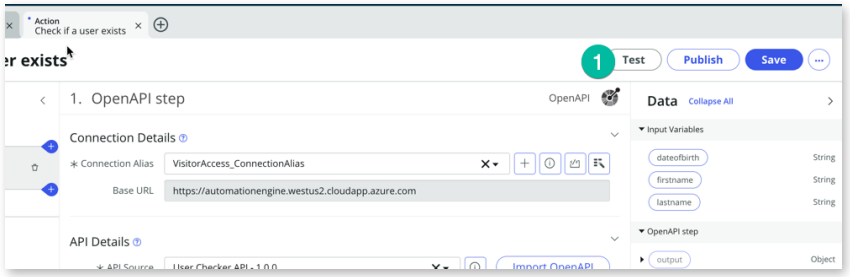
Test the Spoke

Overview

In this activity, you will test the new spoke.

Instructions

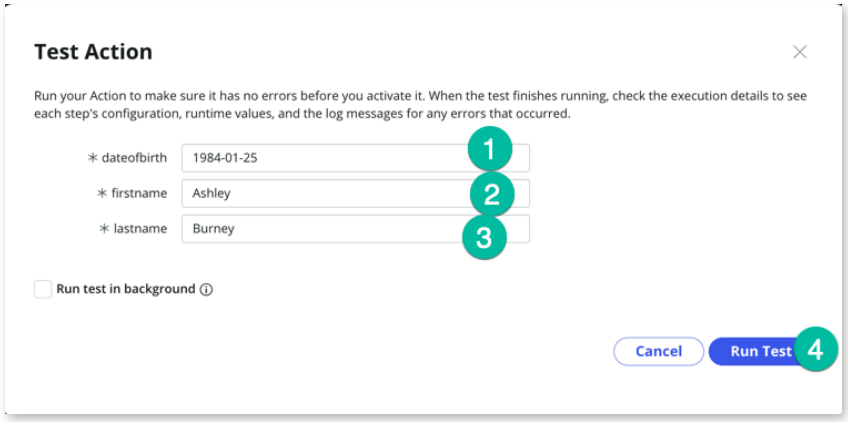
1. It's time to test that Spoke Action! Click on the **Test** button



2. You will be prompted to enter some user information, **dateofbirth** (1), **firstname** (2), **lastname** (3) please use those values below

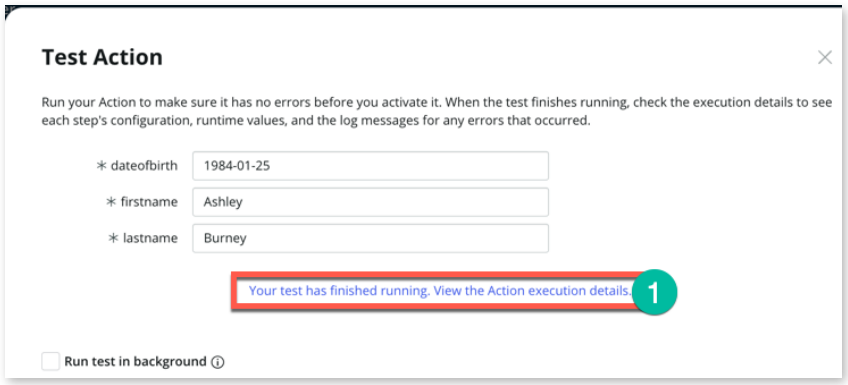
   - Then click **Run Test** (4)

Field	Value
dateofbirth	1984-01-25
firstname	Ashley
lastname	Burney

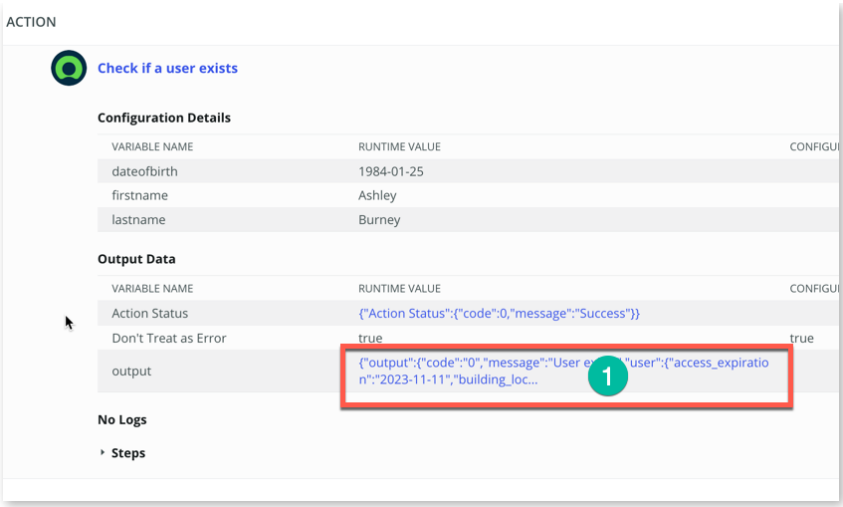


> NOTE
>
> Typically those values will be passed to the action via a Workflow. we are just testing the action manually right now.

3. Once the Action has been executed, click on **Your test has finished running. View the Action execution details** (1) to inspect the response we have received from the external system.

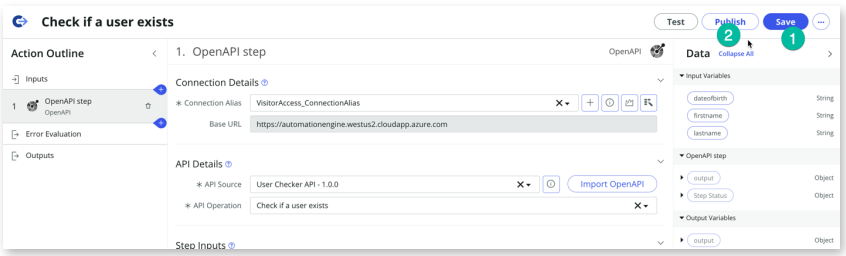


4. On the **Execution Details** page, scroll down until you see the **ouput Data** section and click on the output detail (1)



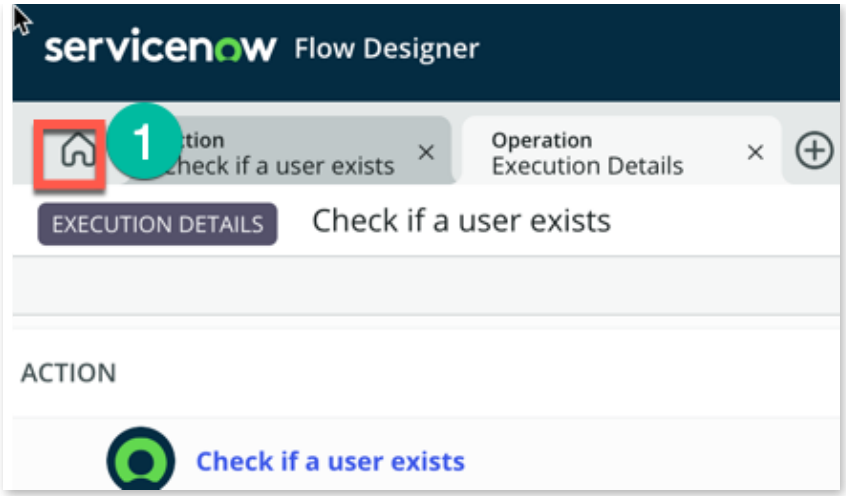
You should see a screen similar to this. Notice the response returned by the API. It contains the return code, message and additional user information. Our new Spoke Action works!

5. Before you proceed to the next step,on your new Action please make sure to click "Save" (1) and then "Publish" (2) to ensure that the Action is saved and includes the update we added to the flow (changing the Connection Alias).



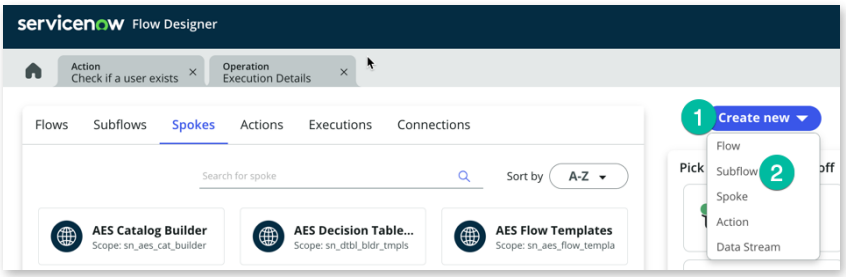
Now let's use that Spoke Action from a flow!

6. Click on the Home button to return to the main page of flow designer



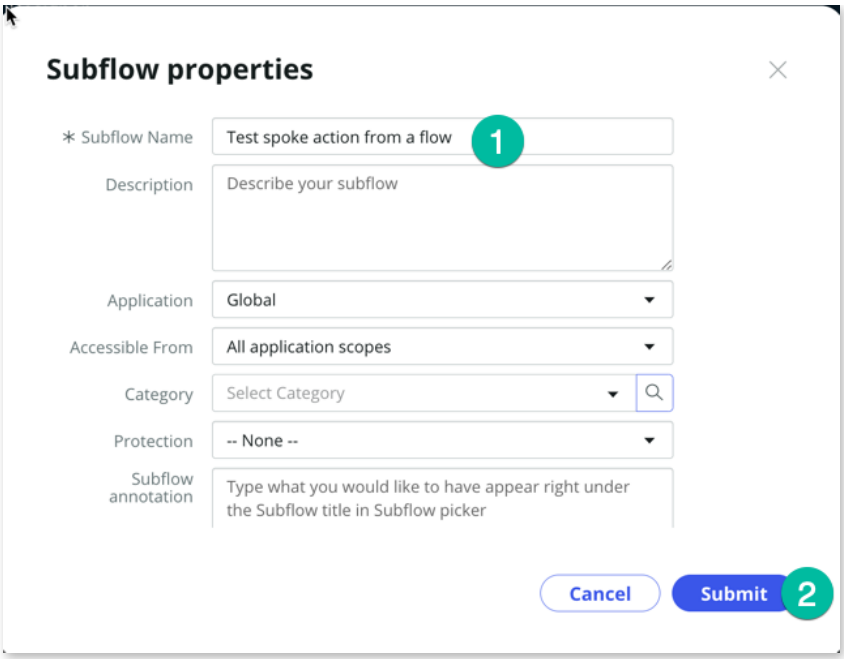
In the following steps, we just want to show a builder can consume/use the new Spoke Action that we have created.

27. Click on **Create New** (1) then **Subflow** (2)

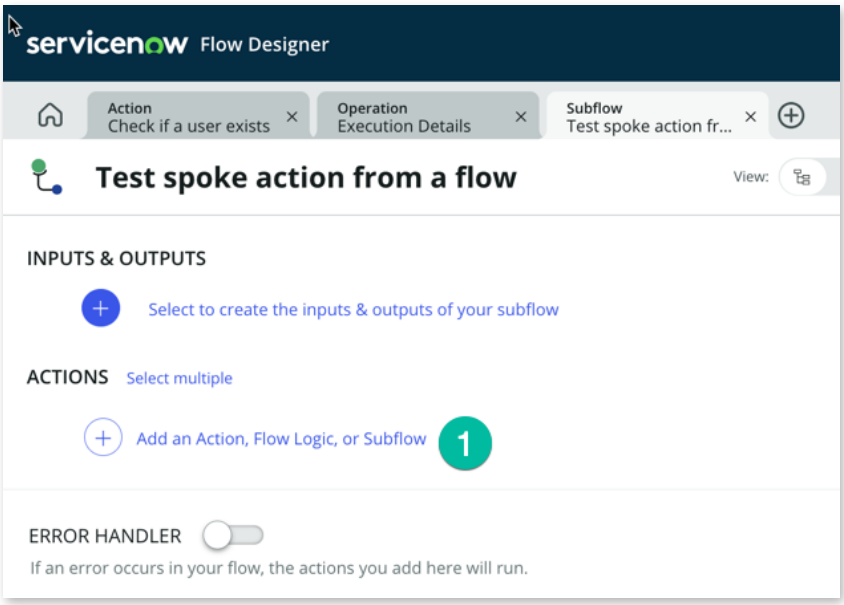


28. Enter a **Subflow name** (put the name you want) and click **submit** (leave all other field with default values)

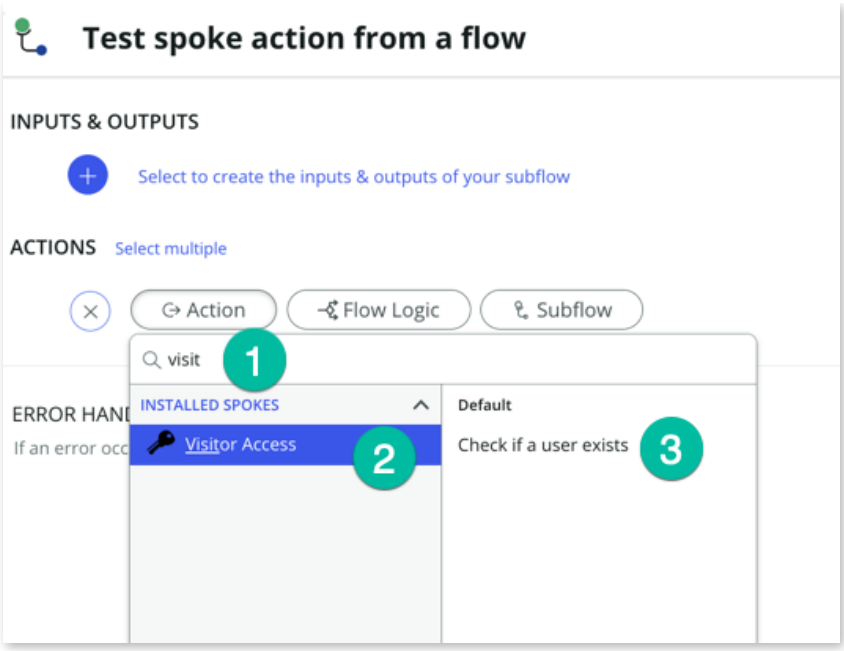
> NOTE
>
> Typically a builder would create a new flow or subflow in his own Application Scope, but for a quick test in a lab instance it doesn't matter we can save it in the Global scope.



29. A new tab will open for your new subflow, under **ACTIONS** Click **Add an Action, Flow Logic or Subflow** (1)



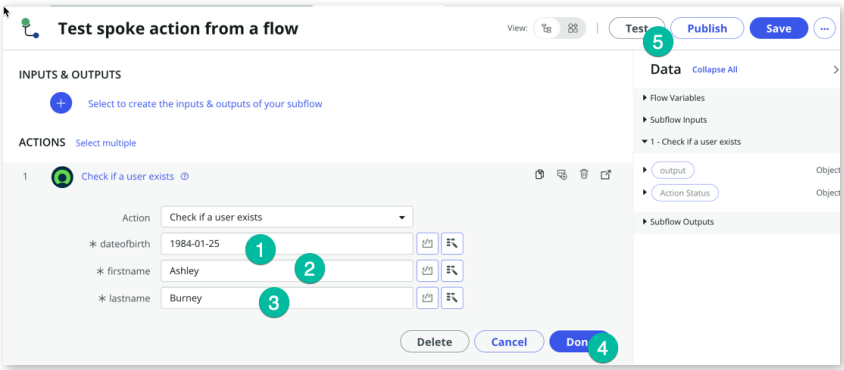
30. Select Action then type **Visit**, this should display your new spoke **Visitor Access** (2), click on it then click the action **Check if a useer exists** (3)



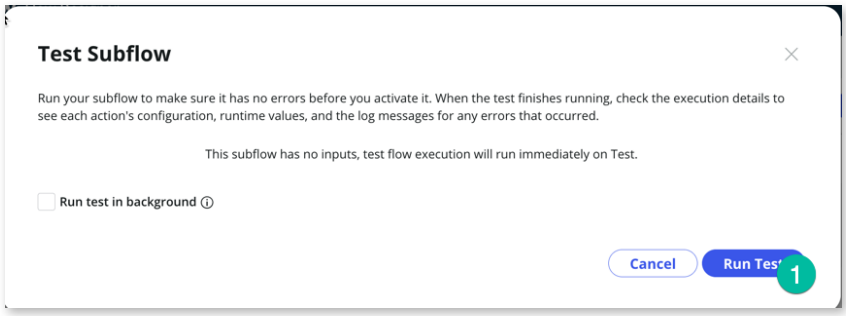
31. Now we can pass the value to the action, here we are going to hardcode values just for test purposes, typically we would lookup a record in ServiceNow and pass values from that record to the action. We will cover that in detail in the optional section of the lab.

Field	Value
dateofbirth	1984-01-25
firstname	Ashley
lastname	Burney

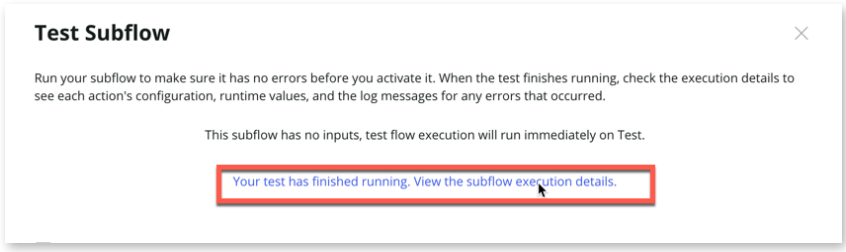
32. Using the values from the table above, set the **dateofbirth** (1). **firstname** (2). **lastname** (3) then click **Done** (4) and click the **Test** button.



Then click **Run Test** (1)



33. Click **Your Test has finished running, View the subflow exectution details**



Recap

In this lab, we've learned how to create a new Spoke using the Spoke Generator, allowing us to integrate ServiceNow with an external application featuring a usable API. In the optional section of this lab, we will delve into more advanced concepts of Flow Designer/Ihub and demonstrate how to employ the value retrieved from the Spoke action to update a record in ServiceNow.

> **NOTE**
> With the Spoke Generator, you no longer need to manually configure the REST Step and JSON Parser Step; the Spoke action generated by the Spoke Generator handles this automatically for you.

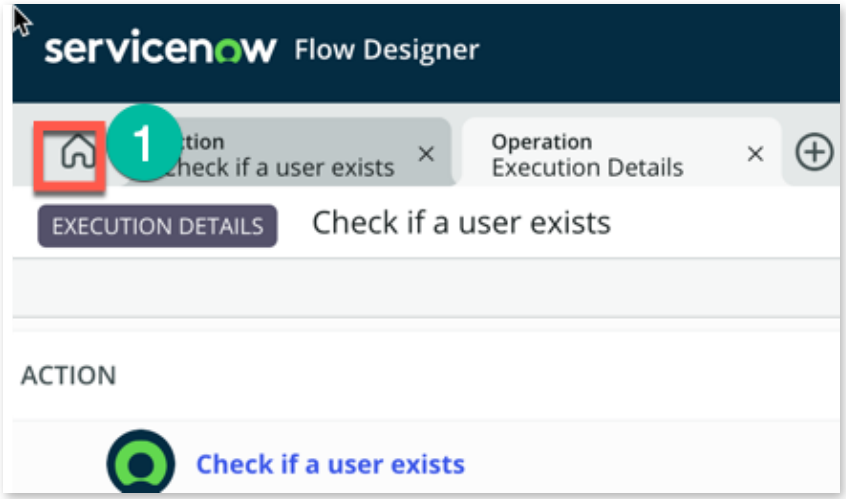
Use the Spoke in a Flow

Overview

In this activity, you will use the new Spoke in a Flow.

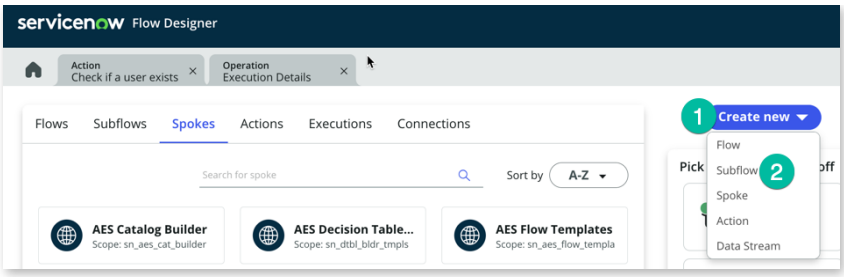
Instructions

1. Click on the Home button to return to the main page of flow designer



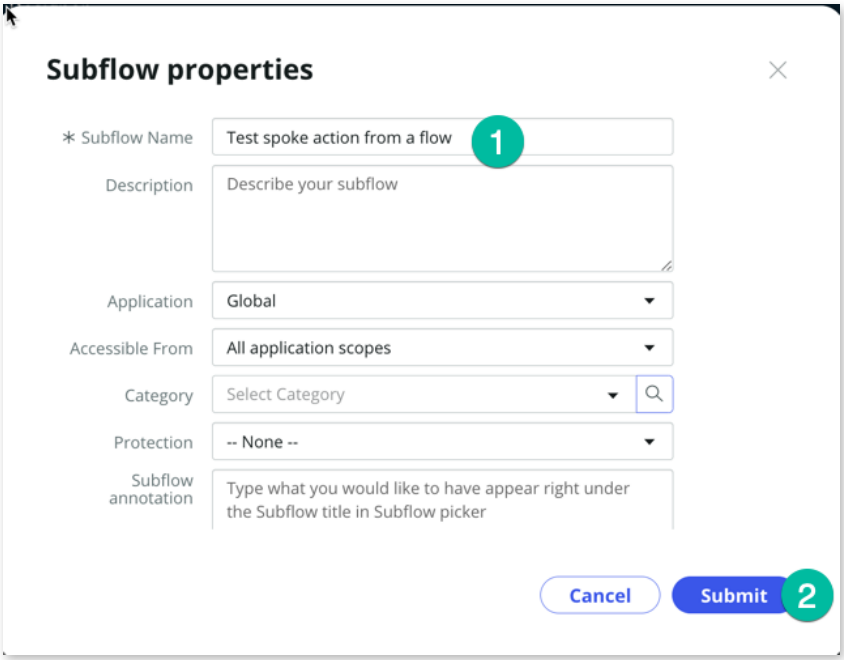
In the following steps, we just want to show a builder can consume/use the new Spoke Action that we have created.

2. Click on **Create New** (1) then **Subflow** (2)

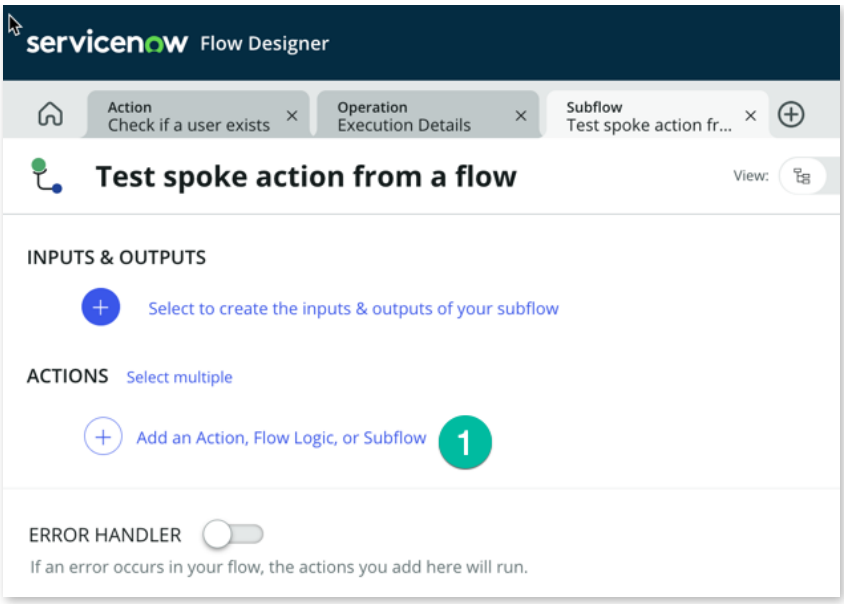


3. Enter a **Subflow name** (put the name you want) and click **submit** (leave all other field with default values)

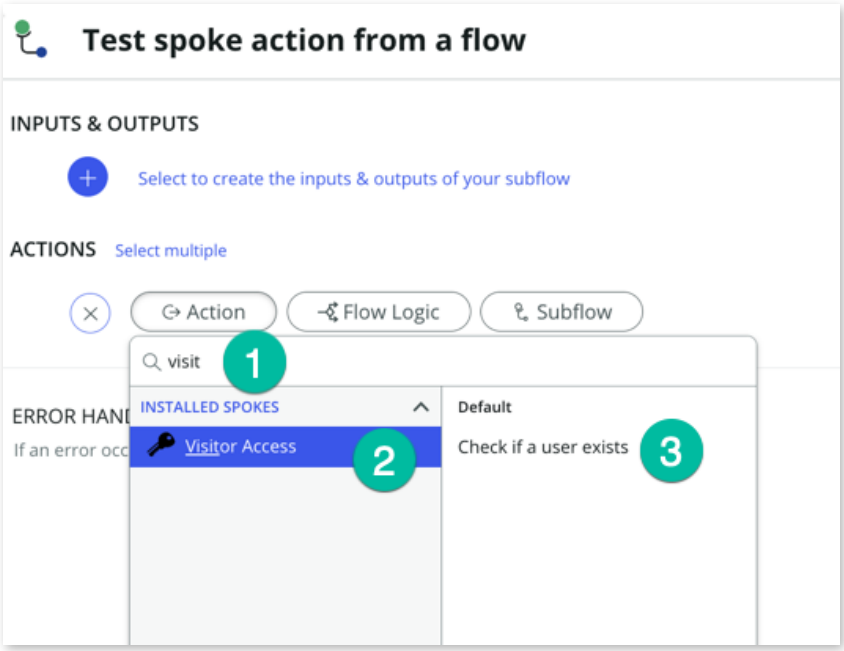
> NOTE
>
> Typically a builder would create a new flow or subflow in his own Application Scope, but for a quick test in a lab instance it doesn't matter we can save it in the Global scope.



4. A new tab will open for your new subflow, under **ACTIONS** Click **Add an Action, Flow Logic or Subflow** (1)



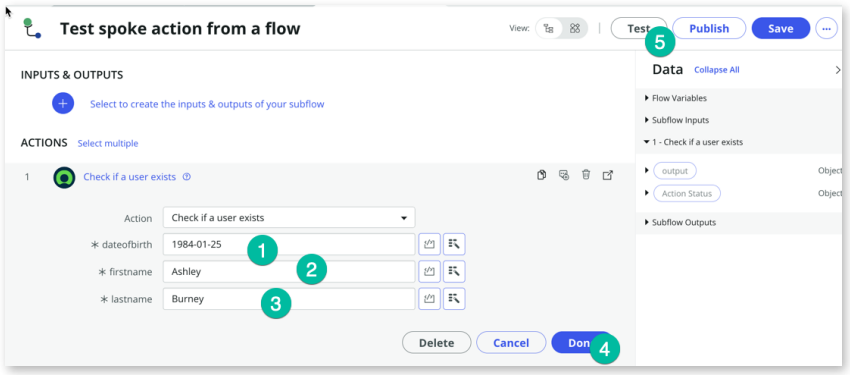
5. Select Action then type **Visit**, this should display your new spoke **Visitor Access** (2), click on it then click the action **Check if a useer exists** (3)



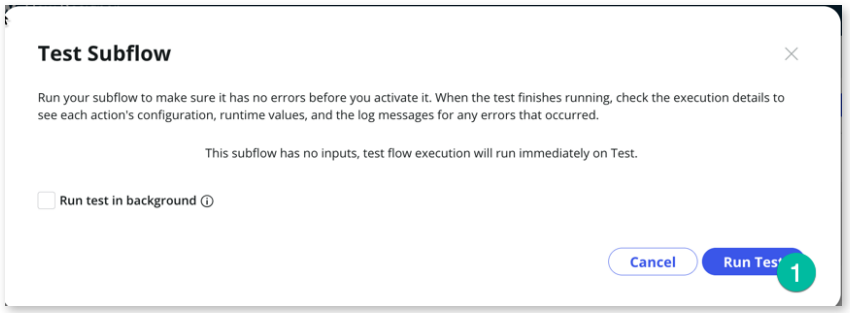
6. Now we can pass the value to the action, here we are going to hardcode values just for test purposes, typically we would lookup a record in ServiceNow and pass values from that record to the action. We will cover that in detail in the optional section of the lab.

Field	Value
dateofbirth	1984-01-25
firstname	Ashley
lastname	Burney

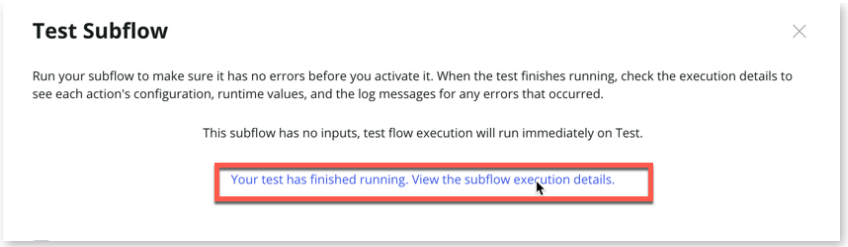
Using the values from the table above, set the **dateofbirth** (1). **firstname** (2). **lastname** (3) then click **Done** (4) and click the **Test** button.



7. Then click **Run Test** (1)



8. Click **Your Test has finished running, View the subflow exectution details**



Recap

In this lab, we've learned how to create a new Spoke using the Spoke Generator, allowing us to integrate ServiceNow with an external application featuring a usable API. In the optional section of this lab, we will delve into more advanced concepts of Flow Designer/Ihub and demonstrate how to employ the value retrieved from the Spoke action to update a record in ServiceNow.

> **NOTE**
> With the Spoke Generator, you no longer need to manually configure the REST Step and JSON Parser Step; the Spoke action generated by the Spoke Generator handles this automatically for you.

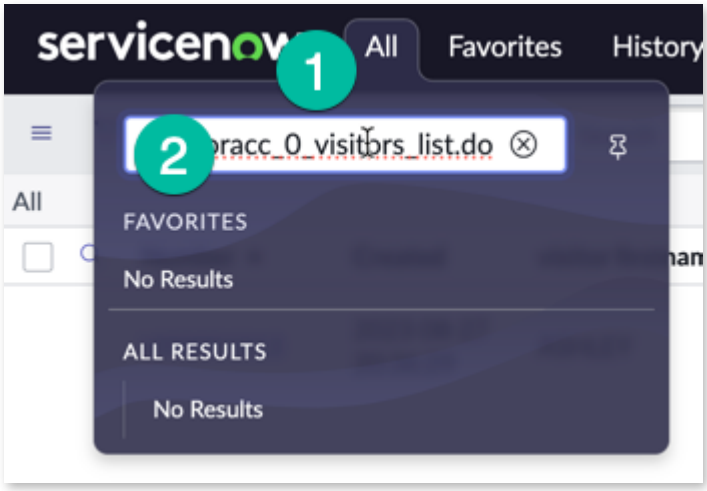
Optional Lab: Section 1

Overview

Handling integrations is often not as straightforward as this lab may suggest. In the previous section of the lab, the data was sent to the API in the same format as it was in ServiceNow. However, in this scenario, we are sending data to an API that expects the data to be in a specific format. In this optional lab, we will explore how to apply data transformation in Flow Designer before sending the data to the external API.

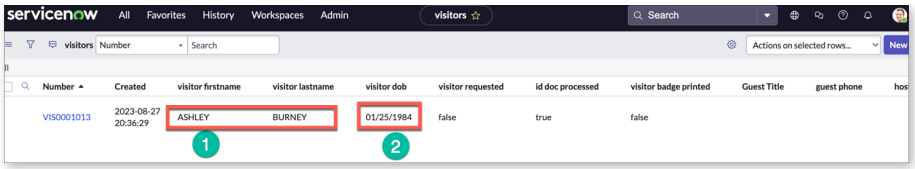
Instructions

1. On your instance, navigate to the **All** menu (1) and in the filter navigator (2), type "x_snc_visitoracc_0_visitors_list.do" and press enter to open that custom table.



In this use case, a custom table is utilized by a ServiceNow app developed by ACME Inc. When a visitor checks in at the reception of their building, they can scan their government ID. Using our DocIntel capability, a new record is generated in that table for the visitor. Information such as the First Name, Last Name, and Date of Birth of the visitor is then extracted from their government ID and stored in that table. In the following section, we will concentrate on creating the workflow that will take this data, transform it, and utilize it with the new spoke we created earlier in this lab.

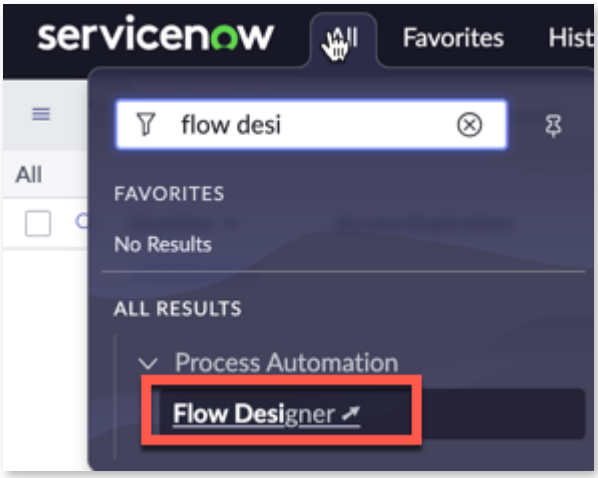
Please note the following details: Visitor First Name and Visitor Last Name values (1) are in uppercase. Additionally, observe the Date of Birth (2), which is in the MM/DD/YYYY format.



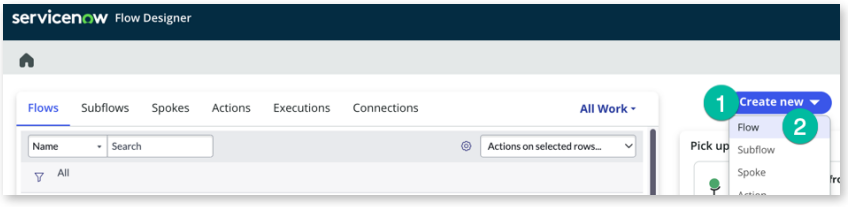
The external API for which we previously created a new spoke action won't recognize those values in that format. The API expects the format to be Firstname, Lastname, and a date in the YYYY-MM-DD format.

Now, let's create the workflow that will query the API to verify whether the user has the necessary authorization to access the building. We will utilize a Data Transform and our newly created Spoke Action for this purpose.

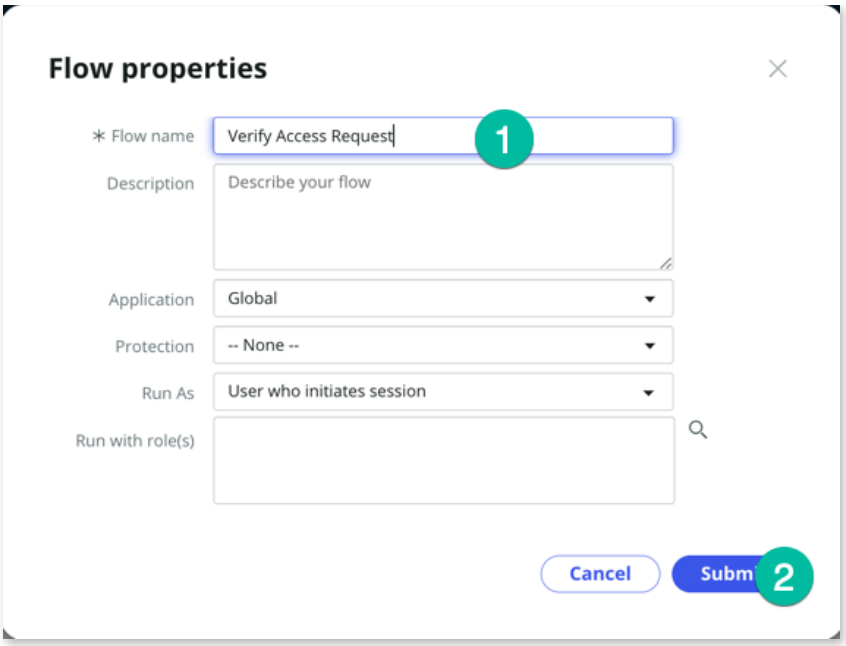
2. Open Flow Designer



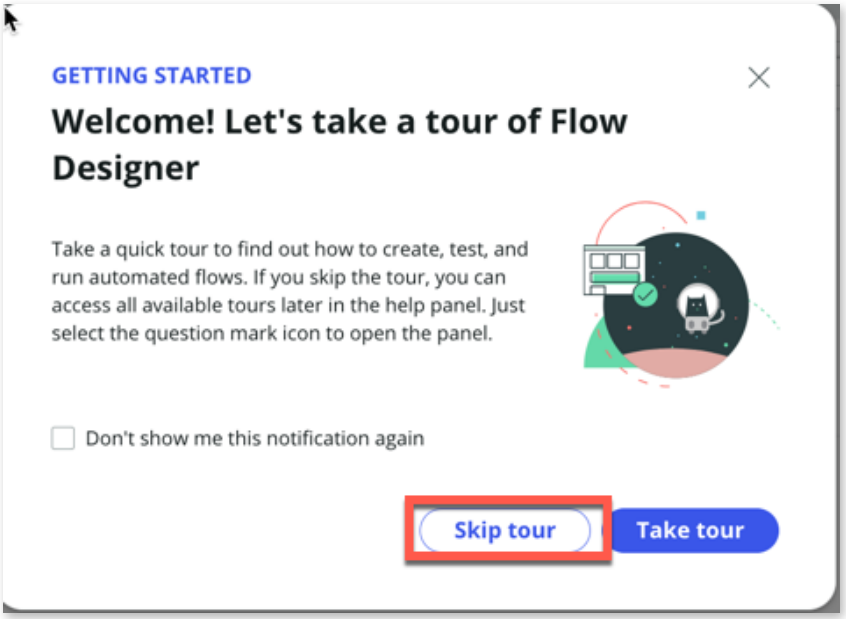
3. Click **Create new** (1) then **Flow** (2)



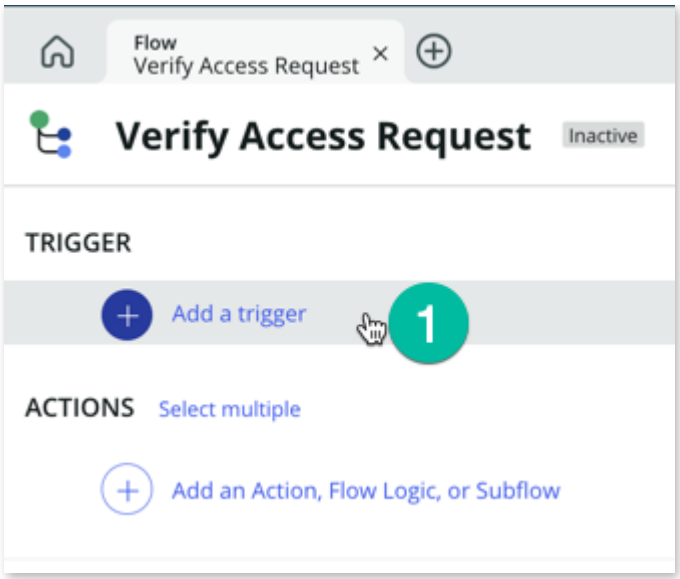
4. Type the value **Verify Access Request** on the **Flow Name** field (1) then click **Submit** (2)



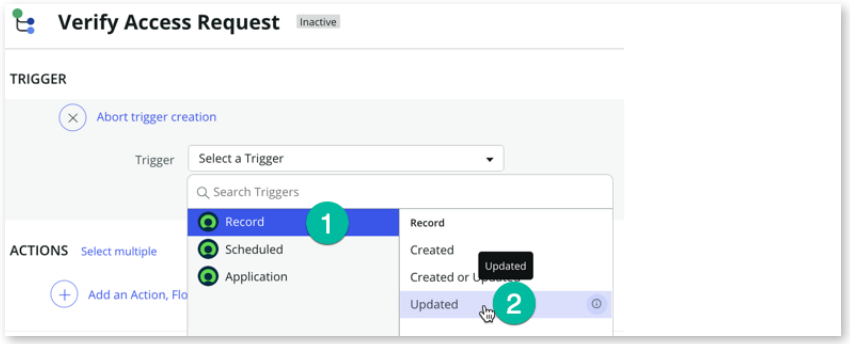
5. If you get this message, click **Skip tour**



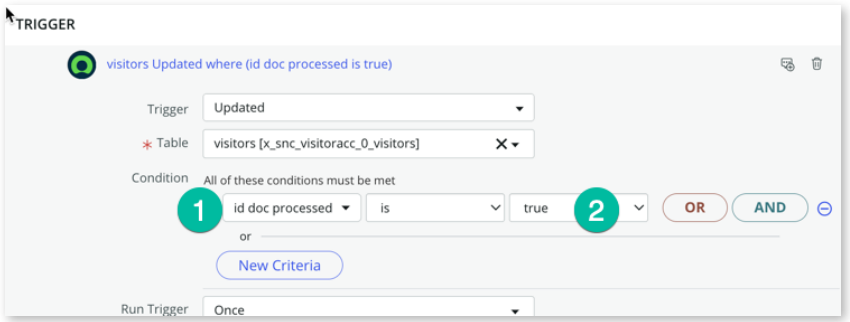
6. Click **Add a trigger** (1) to set the condition to trigger that flow.



7. Select **Record** (1) then click **Updated** (2).



8. Select the **visitors** (1) table from the list then click **Add Filters** (2) Set the condition as shown below

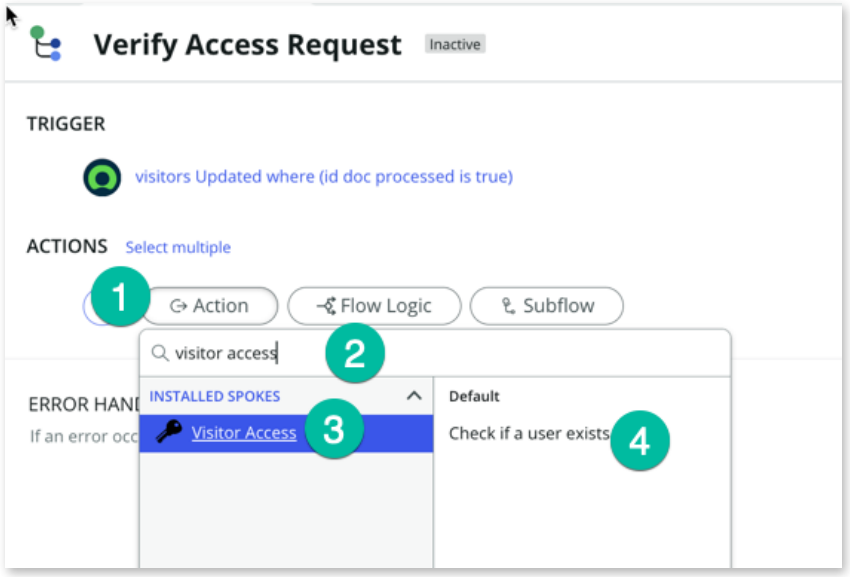


NOTE

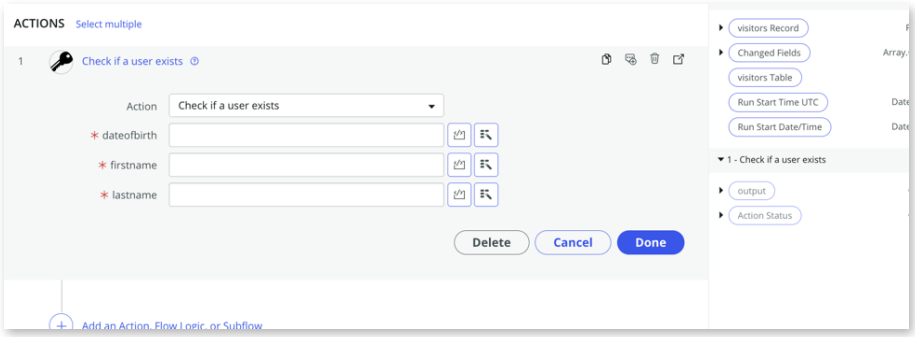
> With the condition we are setting above, the flow will be triggered as soon as DocIntel extracts data from the visitor's government ID and updates the visitor record associated with that visitor.

Now we need to add to our flow the spoke action that we have created earlier in this lab.

9. Under the **Actions** section, click **Action** (1), type **visitor access** (2) then select **Visitor Access** from the **INSTALLED SPOKES** list (3) then click **Check if a use exists** (4)

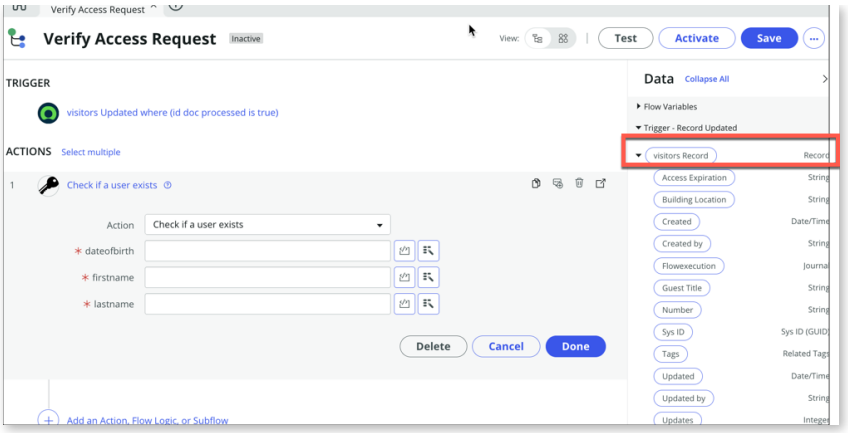


You should see a screen as shown below



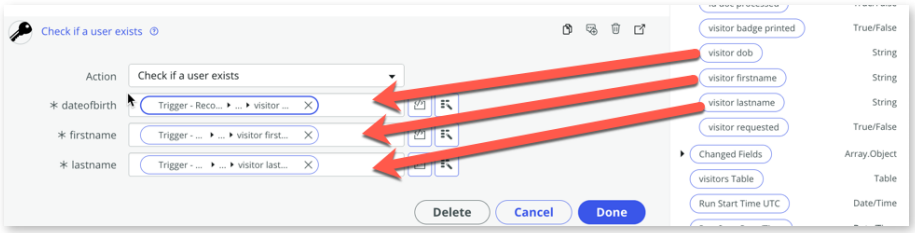
We need to pass the data from the record to that Action step.

10. Expand the Visitors Record secrtion in the data pill as shown below:



11. Then scroll down to see fields we need, **visitor dob**, **visitor lastname**, **visitor firstname**,

12. Grab those data fields and drop them to our action inputs as shown below:

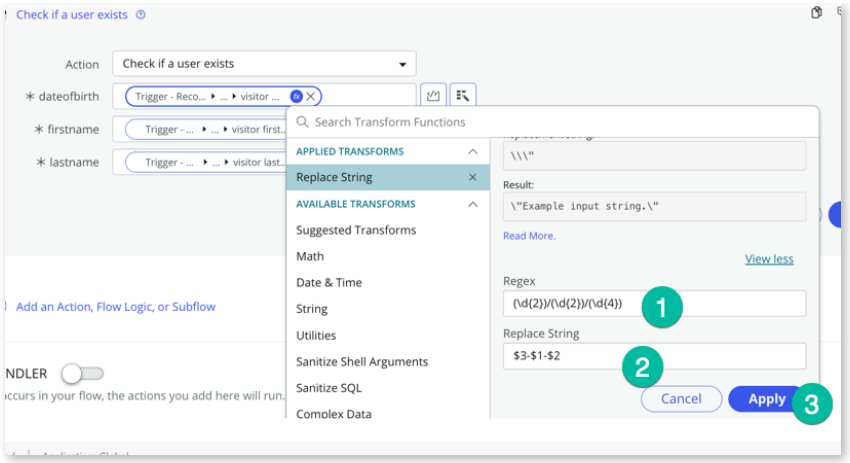


Remember, the external API won't accept the format of that data. We need to apply some data transformation to send the data in the expected format. In Flow Designer, we can use 'Transforms' to dynamically modify the data in fields.

Let's begin with the formatting of the date of birth. Remember that in the ServiceNow record, the date format is MM/DD/YYYY, but we need to send it to the API in the format YYYY-MM-DD.

13. If you click the value in the **dateofbirth** field, it will show a **fx** icon, click on it, then type **Replace** (2), then select **Replace String** (3), we that Transforms we can use a simple regular expression to match the string to replace. Alt text](<./images///2023-09-21_13-08-32 (2).png>)

14. On the **Regex** field (1) type this value **(\d{2})/(\d{2})/(\d{4})** and on the **Replace String** field (2) type this value **$3-$1-$2**



ADDITIONAL READING IF INTERESTED TO LEARN MORE ABOUT REGULAR EXPRESSION, OTHERWISE SKIP THIS.

The regex pattern `(\d{2})/(\d{2})/(\d{4})` is used to match and capture date strings in the format MM/DD/YYYY.

Here's what each part of the pattern does:

1. `(\d{2})`: This part captures two digits (0-9) and encloses them in parentheses to create a capturing group. It's used to match the month portion of the date (MM). `\d` represents any digit, and `{2}` specifies that exactly two digits must be matched.

2. `/`: This part matches the forward slash character (/) literally. It's used to separate the month, day, and year portions of the date.

3. `(\d{2})`: Similar to the first part, this captures two digits (0-9) to match the day portion of the date (DD).

4. `/`: Another forward slash to separate the day and year.

5. `(\d{4})`: This captures four digits (0-9) to match the year portion of the date (YYYY).

So, when you apply this regex pattern to a string, it will capture date strings in the format MM/DD/YYYY and store the month, day, and year portions as separate capturing groups, allowing you to extract and work with these components individually.

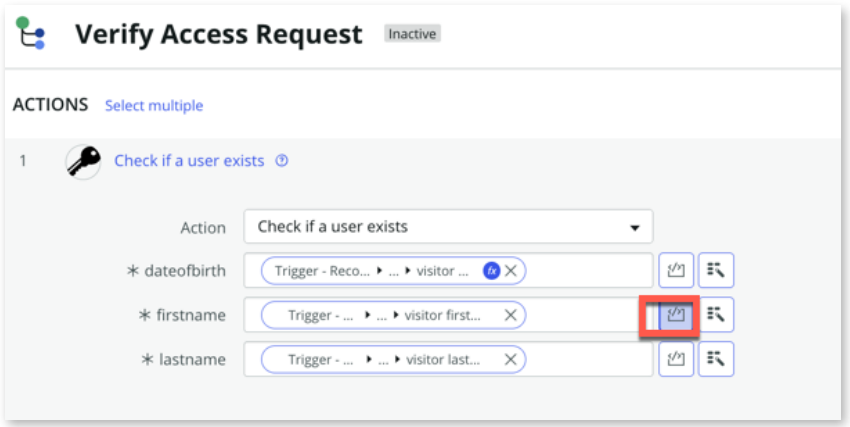
In the replace string field we have typed **$3-$1-$2**

$3 refers to the contents of capturing group 3, which is the year "2023." $1 refers to the contents of capturing group 1, which is the month "12." $2 refers to the contents of capturing group 2, which is

the day "31."
This allows us to format the date as desired, although there might be other ways to achieve this as well.

Now, let's perform some data transformation for the First Name and Last Name fields. They arrive in uppercase from ServiceNow, but the external API expects these values with only the first letter of the first name and the first letter of the last name in uppercase. Therefore, we need to transform the data to make it compatible. We could potentially use existing Transforms to accomplish this, but we want to introduce another method for more advanced data transformation. You can use the inline script feature on the field to transform the data.

15. Beside the firstname field, click that Toggle Scripting icon as shown below



add this piece of code:

```
// Access the value you want to transform
var inputString = fd_data.trigger.current.visitor_firstname;
// Replace 'your_field_name' with the actual field name

// Check if the inputString is not empty
if (inputString) {
   // Convert the string to lowercase, except the first
character
   var firstChar = inputString.charAt(0);
   var restOfString = inputString.slice(1).toLowerCase();
   var transformedString = firstChar + restOfString;

   // Return the transformed value
   return transformedString;
} else {
   // If the input is empty, return it as-is
```

```
    return inputString;
}
```

16. On the **lastname** field add this piece of code

```
// Access the value you want to transform
var inputString = fd_data.trigger.current.visitor_lastname;
// Replace 'your_field_name' with the actual field name

// Check if the inputString is not empty
if (inputString) {
    // Convert the string to lowercase, except the first
character
    var firstChar = inputString.charAt(0);
    var restOfString = inputString.slice(1).toLowerCase();
    var transformedString = firstChar + restOfString;

    // Return the transformed value
    return transformedString;
} else {
    // If the input is empty, return it as-is
    return inputString;
}
```

In summary, this script capitalizes the first letter of the input string while converting the rest of the string to lowercase. If the input is empty, it returns an empty string. This transformation ensures that the format matches the expectations of the external API endpoint."

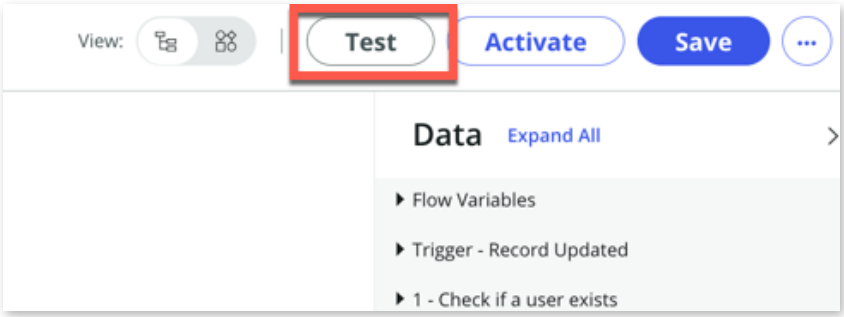
Optional Lab: Section 2

Overview

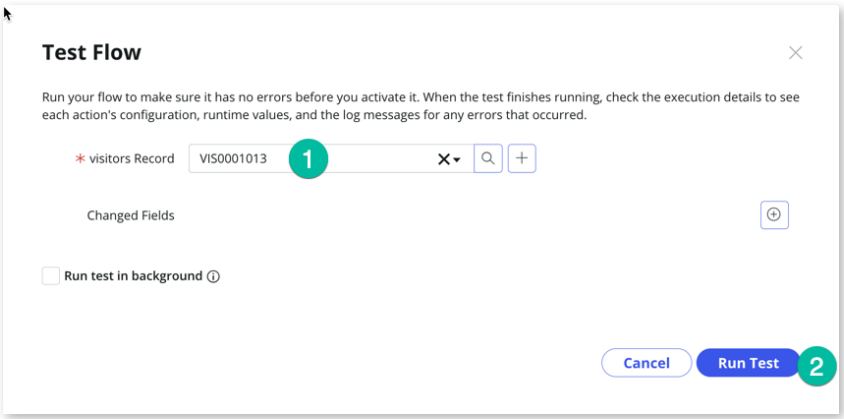
It is time to test if our Transforms work.

Instructions

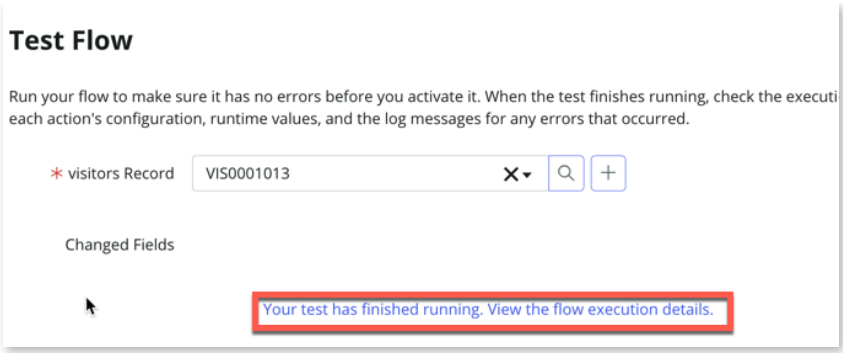
1. Click Test to try your flow



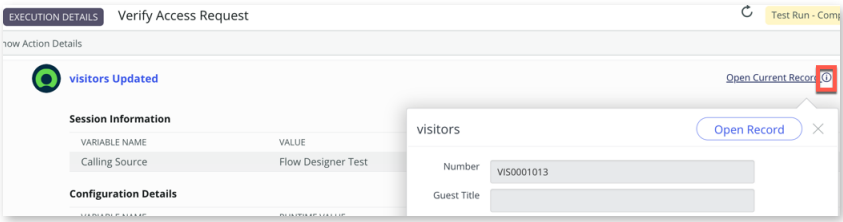
2. Pick a record from the list (1) then click **Run Test** (2)



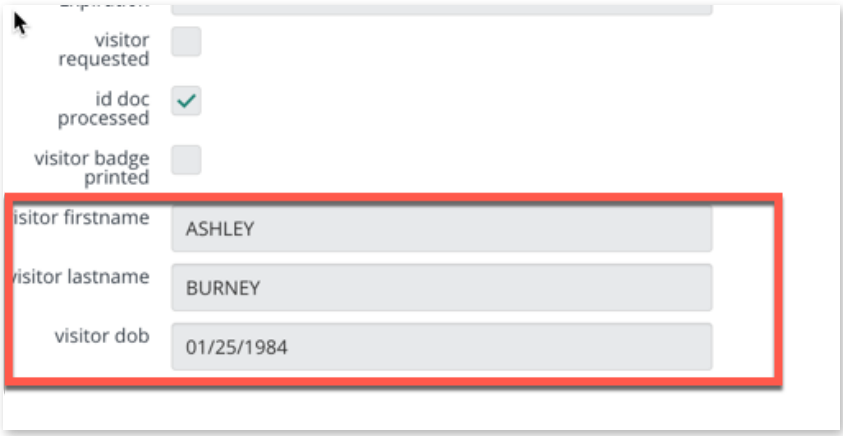
3. Click on **"Your test has finished running. View the flow execution details"**



4. In the **TRIGGER** section, on the right-hand side, single click on the **Open Current record** to inspect the values of that record

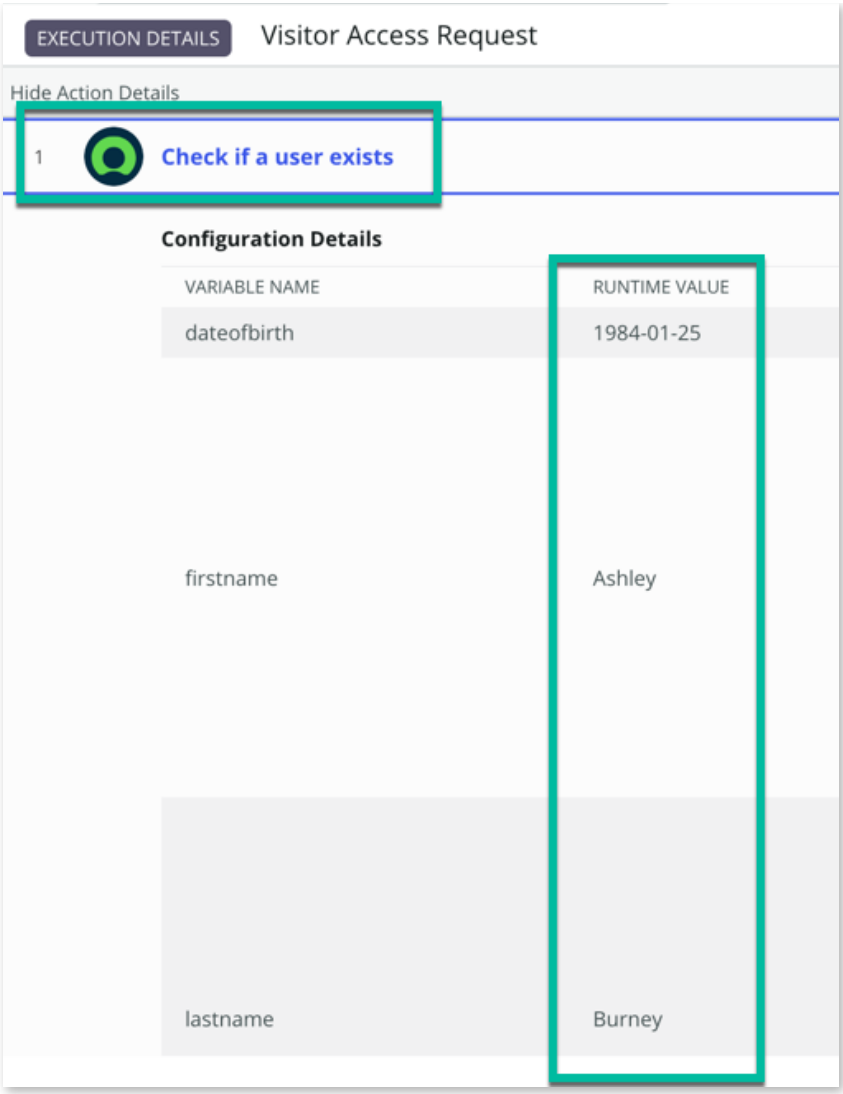


5. Scroll down to see the values for visitor firstname, visitor lasname, visitor dob



6. Notice the value are all stored in uppercase in ServiceNow, and the date of birth is not in the format that the API End point wants.Let Now see if our transforms have worked.

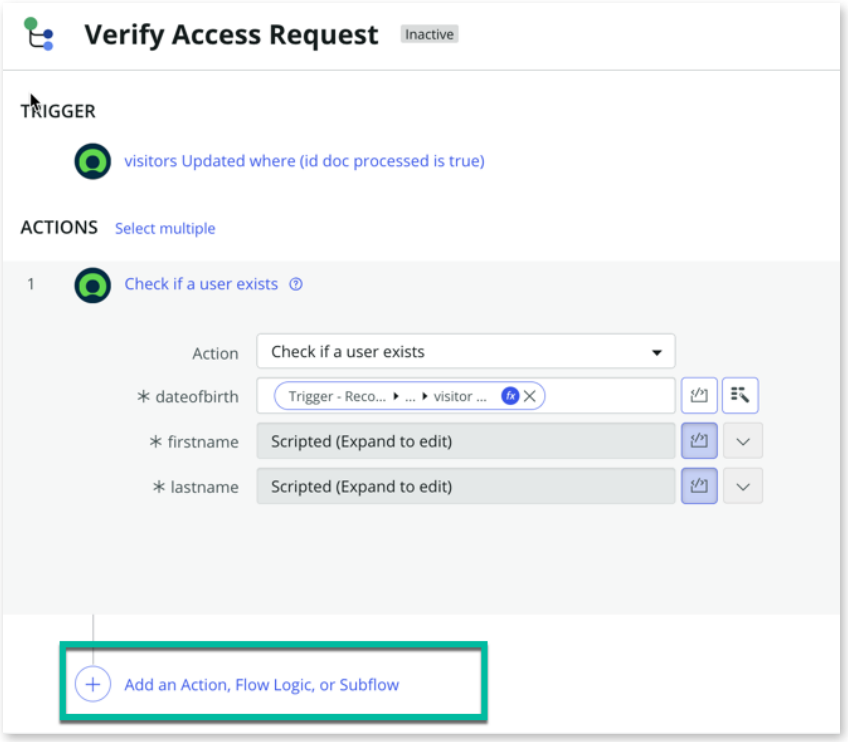
7. Click on the **Check if a user exists** (1) step to check if the values were transformed correctly:



8. If you continue to scroll down, you should be able to see the output that contains the response from the API End point

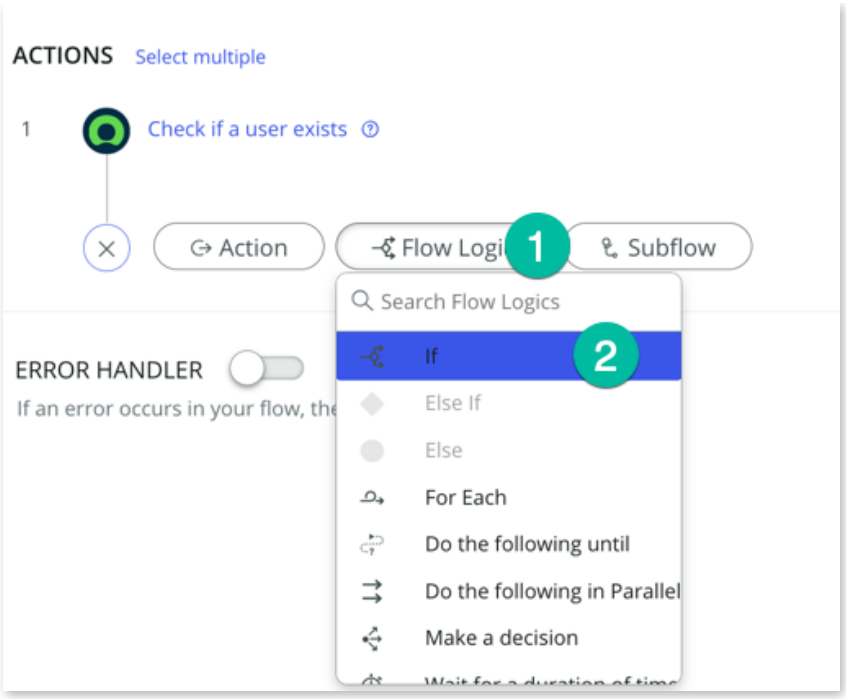
9. We are going to add steps to our flow to update the source record in ServiceNow with the values received from the external system (API endpoint). Please return to the 'Verify Access Request' flow.

10. Then click **Add and Action, Flow Logic, or Subflow**

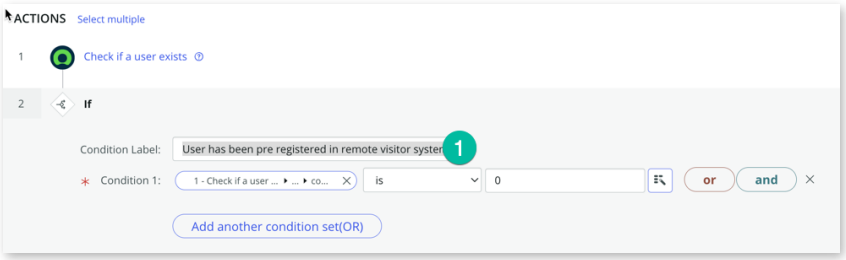


11. We want to incorporate logic into the flow so that if the API confirms the user's existence (indicating that the visitor has been authorized and registered in the external visitor access app), we can retrieve the output from the API and update the record with information about the user's authorizations.

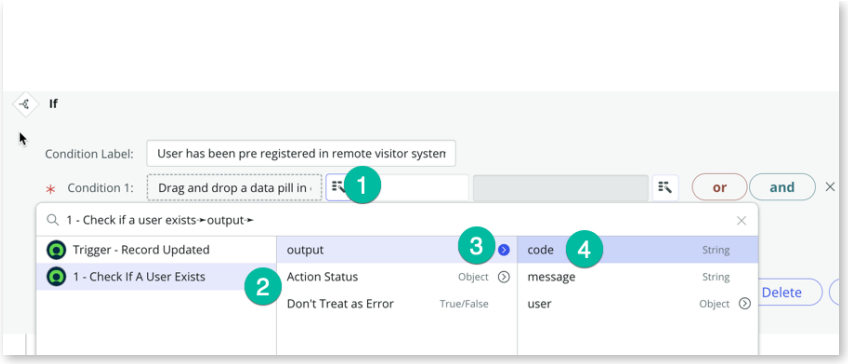
12. Add a **IF** statement by clicking **Flow Logic** (1), then **If** (2)



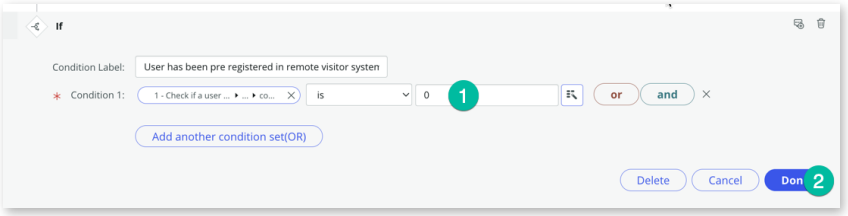
13. In the **Condition Label** field (1), copy and paste this text : "User has been pre registered in remote visitor system" this is just to make it easier for someone who read the flow to understand the logic then in the Condition.



14. In the Condition Section, Click the **Data pill** icon (1) then select **1 - Check if A user Exists** (2), **output** (3), then **code** (4)



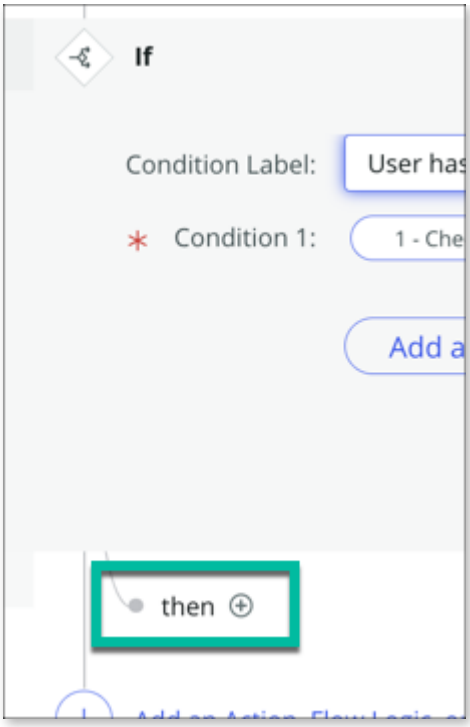
15. type the value **0** (1) as shown and click **Done** (2)



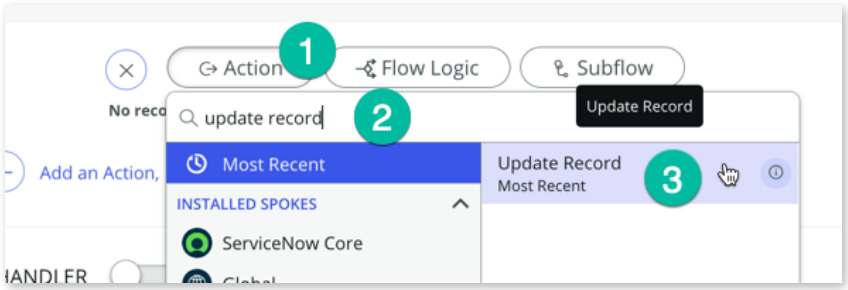
We know that when the API return code 0 means that the user have been found in the remote system

Now the the condition is set correctly we need to add a step to update the user record in servicenow when we meet this condition.

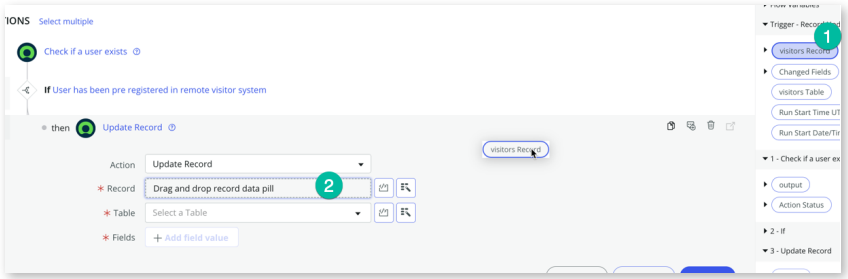
16. Click the **then** section as shown



17. Then Click **Action** (1), type **update record**, then click **Update record**

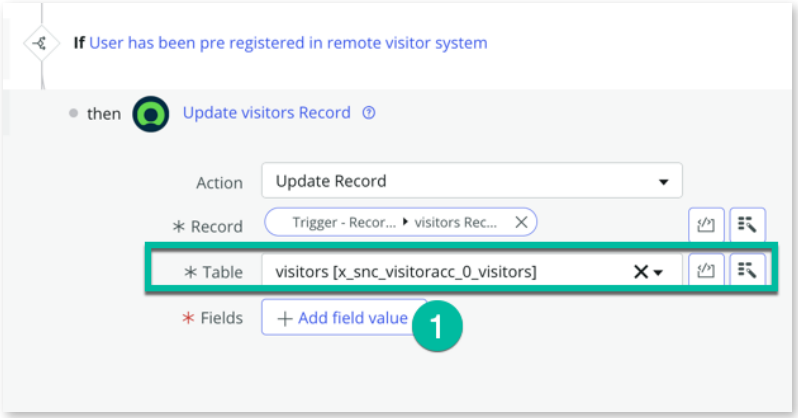


18. Drag the **Visitors Record** (1) from the data pill on the right hand side, then drop into the Record field of the **Update Record** step (2) as shown



19. This should set the right Table on the table field automatically as shown below.

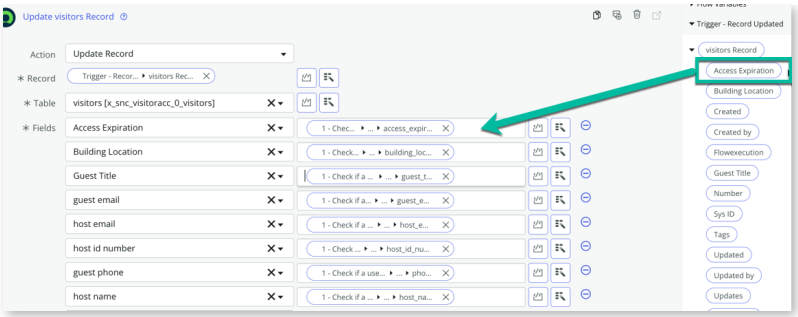
- Click **Add Field value** (1)



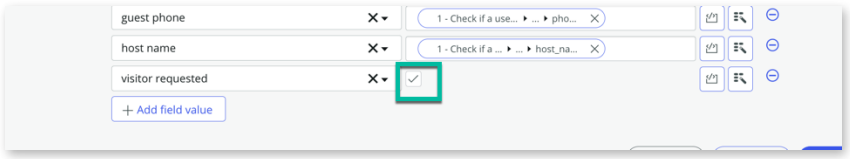
- then add those fields

Field
Access Expiration
Building Location
Guest Title
guest email
host email
host id number
Guest phone
Host name
visitor requested

- Drag each values from the data pill on the right hand side (Trigger - Record Updated - Visitor Record) to the corresponding field on the record



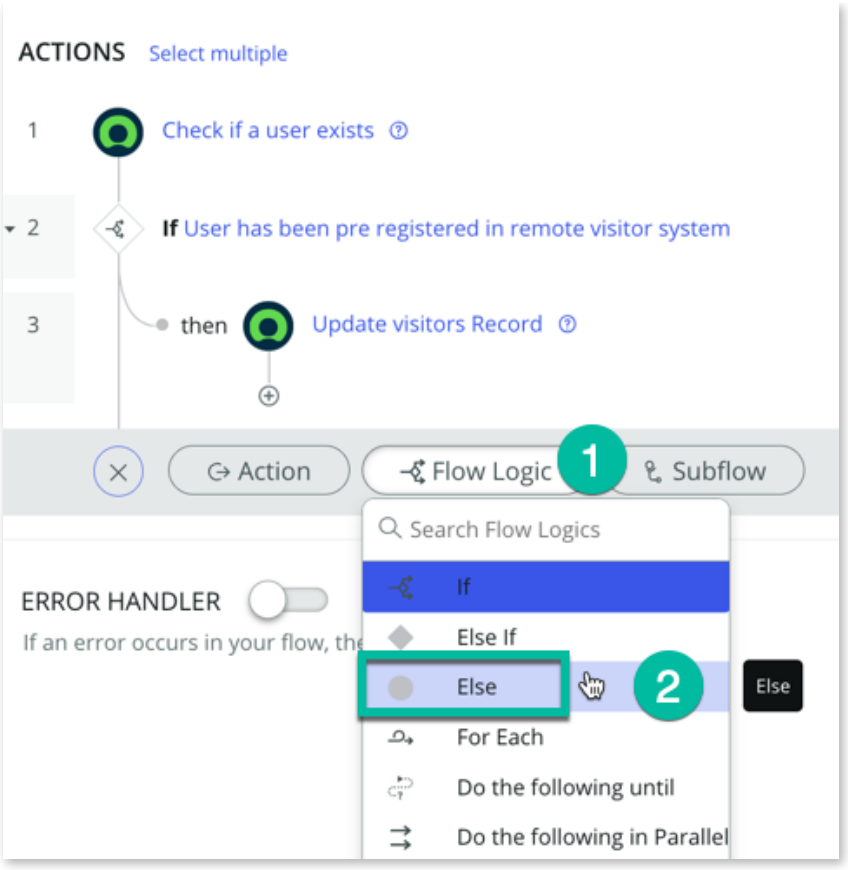
20. The last field, **visitor requesed** is a true/false type, you just need to click the check box as shown



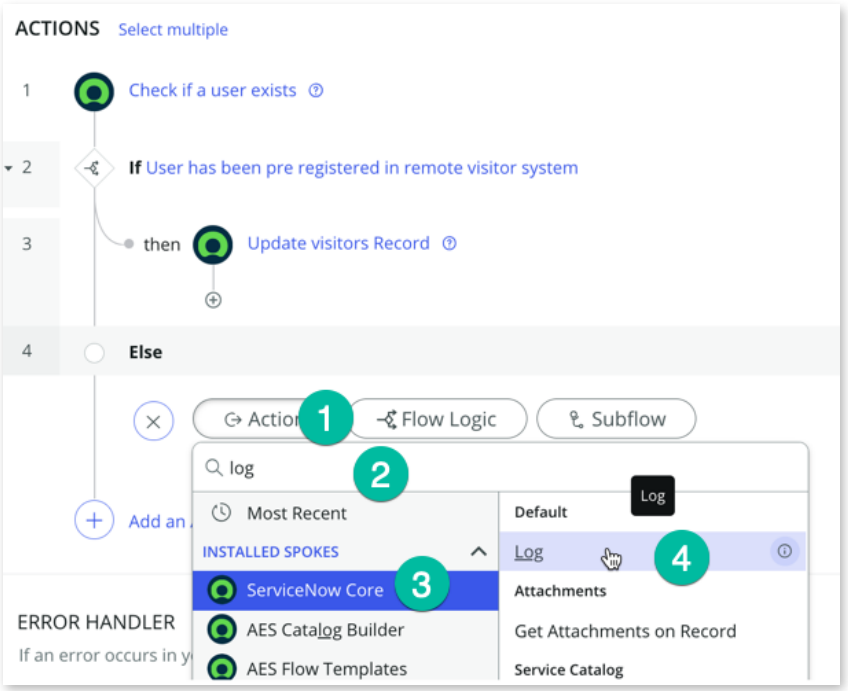
With the step we just added, if the user is found in the Visitor Access system, we then take all the information returned from the API and update the record in ServiceNow with those.

In the case the user is not found in the Visitor Access system, we just want to write a log. let just add that step.

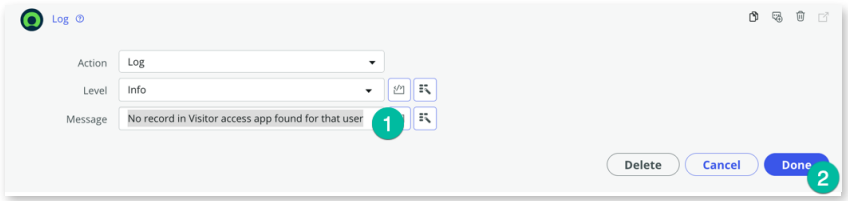
21. Add a Else statement to your flow as shown, click **Add an Action, Flow Logic, or Subflow**, then select **Flow logic** (1), **Else** (2)



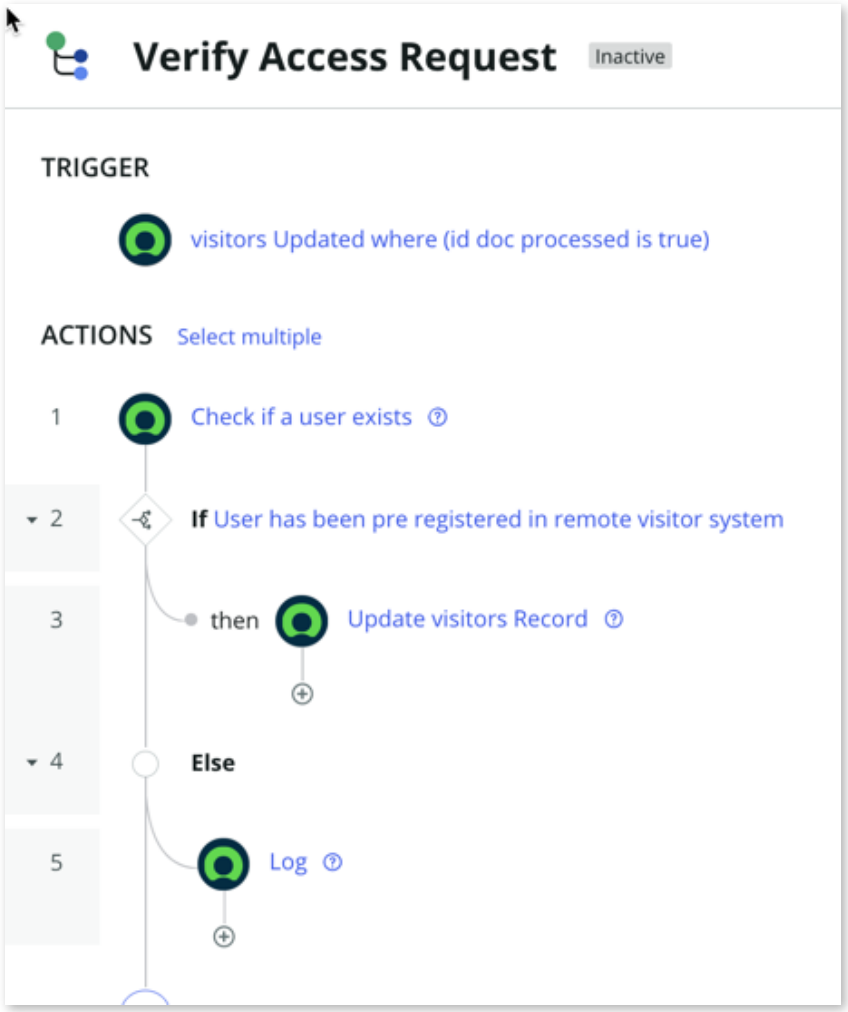
22. Add the **Log** step by click on **Action** (1), then type **log** (2), then click **ServiceNow Core** (3) then click **Log** (4)



23. On the **Field** (1), copy and past this value : "**No record in Visitor access app found for that user**" then click **Done** (2)



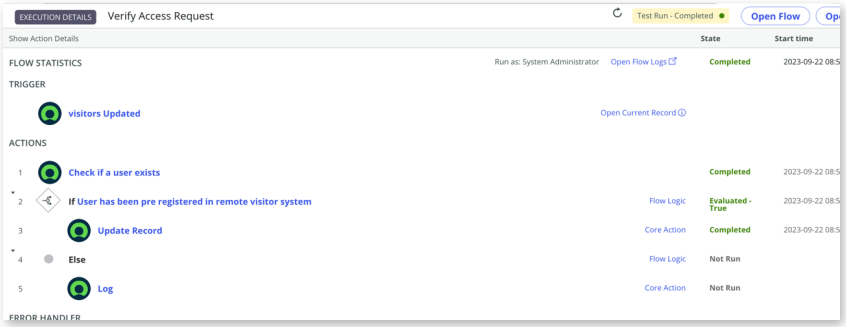
24. We are done building the flow, it should look like this:



25. It's time to test it, click the Test button, select a **visitor record** (2) then click **Run Test** (2)

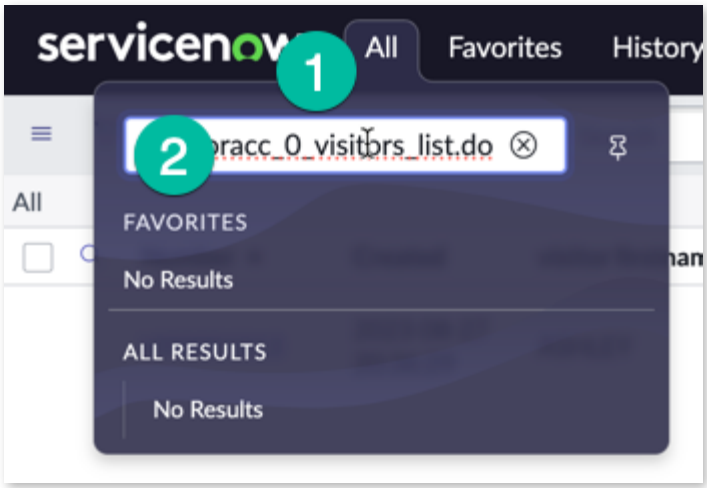
26. Open the execution detail of the flow to see if has updated the record with the values coming from the API

27. If the step Update Record shows **Completed** it means it the record was successfully updated!

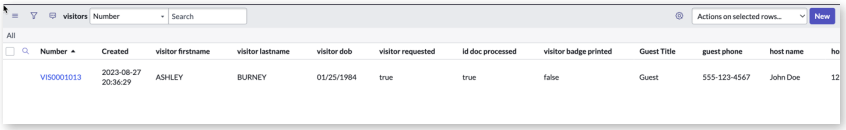


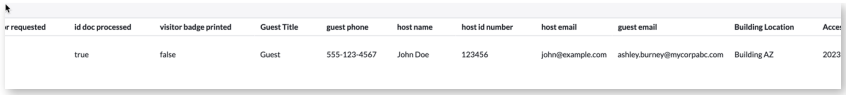
Feel free to go on the custom visitors table to verify that the record was update with the values from the API:

28. click on the **All** menu (1) and in the filter navigator (2) type the x_snc_visitoracc_0_visitors_list.do and press enter to open that custom table.



29. All the field on the visitor record you have used for your test should now be updated





r requested	id doc processed	visitor badge printed	Guest Title	guest phone	host name	host id number	host email	guest email	Building Location	Acce
	true	false	Guest	555-123-4567	John Doe	123456	john@example.com	ashley.burney@mycorpabc.com	Building AZ	2023

Recap

Congratulations, this marks the end of the optional section. You have learned how to transform data before sending it via a custom spoke that we have built!