



Microsoft Power Platform

Dev in a day

Lab 03 Custom connector for existing API/ December 2022

Table of Contents

Lab Scenario	1
Exercise 1 – Create Custom Connector	2
Task 1: Download open API definition and create solution.....	2
Task 2: Create custom connector	3
Task 3: Modify the definition	6
Task 4: Test connector	8
Exercise 2 – Add Custom Code.....	11
Task 1: Add code from resource folder.....	11
Task 2: Test custom code	15
Exercise 3 – Use Custom Connector	16
Task 1: Test connector from canvas app.....	16
Task 2: Test connector from flow (This task is optional)	20

Lab Scenario

Working as part of the PrioritZ fusion team you will be configuring a custom connector for an existing API. The team would like to add badging to the PrioritZ application to give credit to users when they have completed ranking an item. The team identified an existing API, but it doesn't have a Power Platform connector.

When you review the API, you see that it has four operations and uses API key authentication.

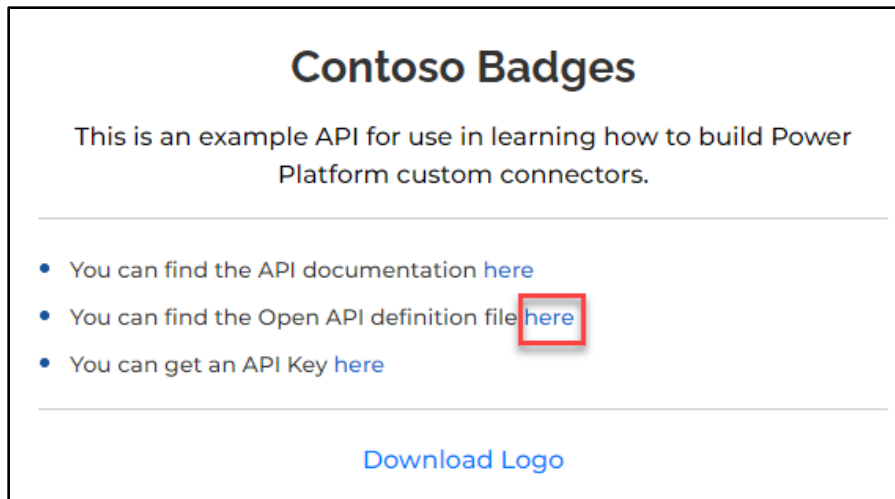
AddCredit	
POST	/AddCredit
GetRecipient	
GET	/GetRecipient
ListBadges	
GET	/ListBadges
ListRecipients	
GET	/ListRecipients

Exercise 1 – Create Custom Connector

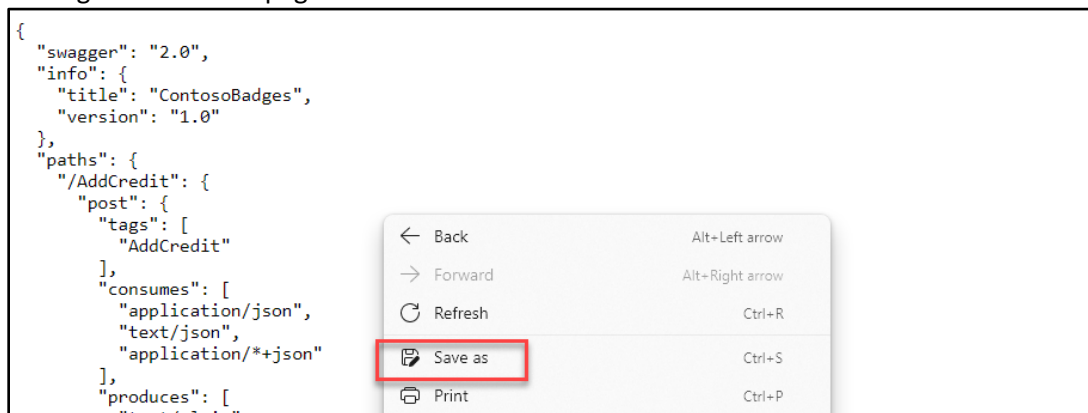
In this exercise, you will download open API definition, create a solution for Contoso Badges custom connector. Currently, custom connectors must be in a separate solution from the apps and flows that use them, and then create the custom connector.

Task 1: Download open API definition and create solution

1. Navigate to [Contoso Coffee Badges \(contosobadgestest.azurewebsites.net\)](https://contosobadgestest.azurewebsites.net)
2. Click on open the **Open API definition file** link.



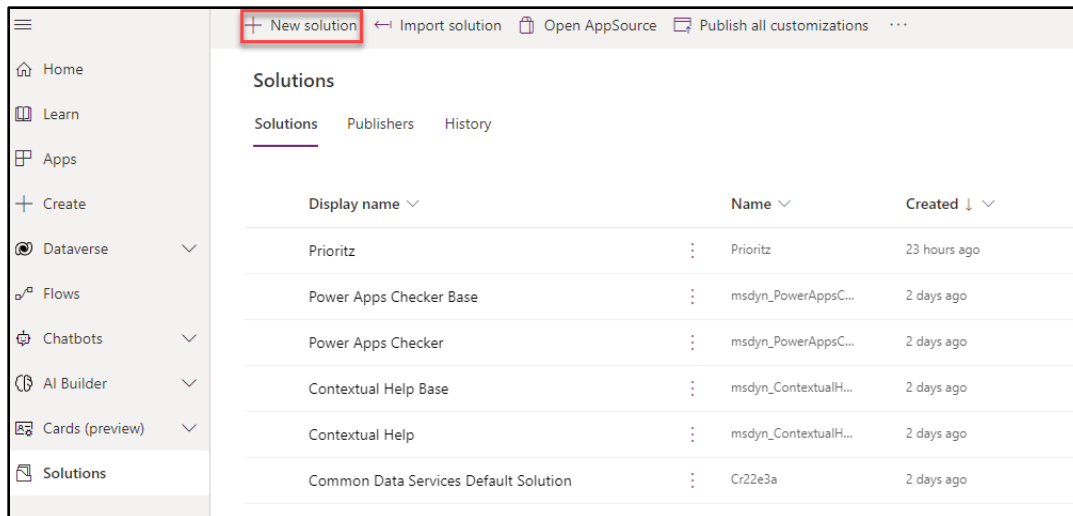
3. Do a quick review of the Open API definition.
4. Right click on the page and select **Save as**.



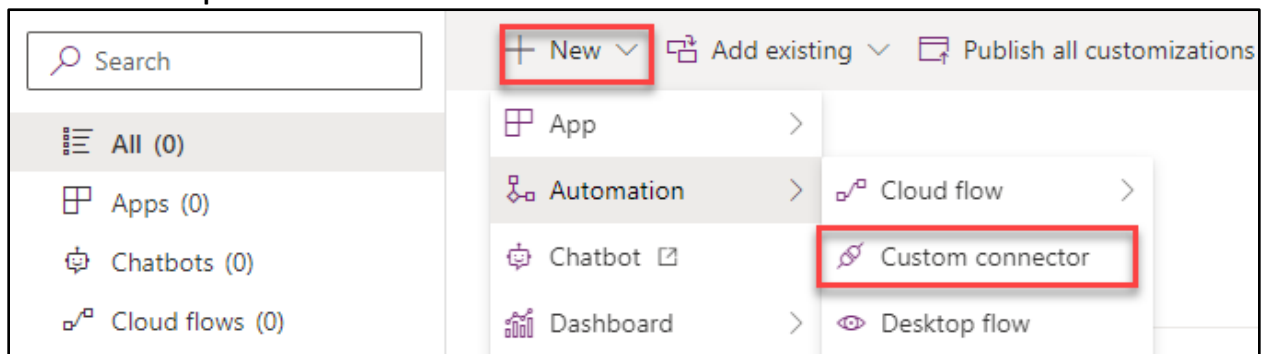
5. Save the **swagger.json** file on your local computer.

Task 2: Create custom connector

1. Navigate to [Power Apps maker portal](#) and make sure you are in your dev environment.
2. Select **Solutions** and click **+ New solution**.



3. Enter **Contoso Badges connector** for Display name, select **Contoso Coffee** for Publisher, and click **Create**.
4. Open the **Contoso Badges connector** solution you created.
5. Click **+ New | Automation** and select **Custom connector**.



6. Enter **Badges connector** for Connector name, **contosobadgestest.azurewebsites.net** for Host, and click **Create connector**.

Connector Name: **Badges connector**


1. General > 2. Security > 3. Definition > 4. Code (Preview) > 5. Test

Swagger Editor: ☐ Create connector: ☒ Cancel:

General information

Add an icon and short description to your custom connector. Your host and base URL will be automatically generated from the swagger file.

General information

 [Upload connector icon](#)
Supported file formats are PNG and JPG. (< 1MB)

[Upload](#)

Icon background color
A color to show behind the icon (e.g., '#007ee5')

Description
Give your custom connector a short description

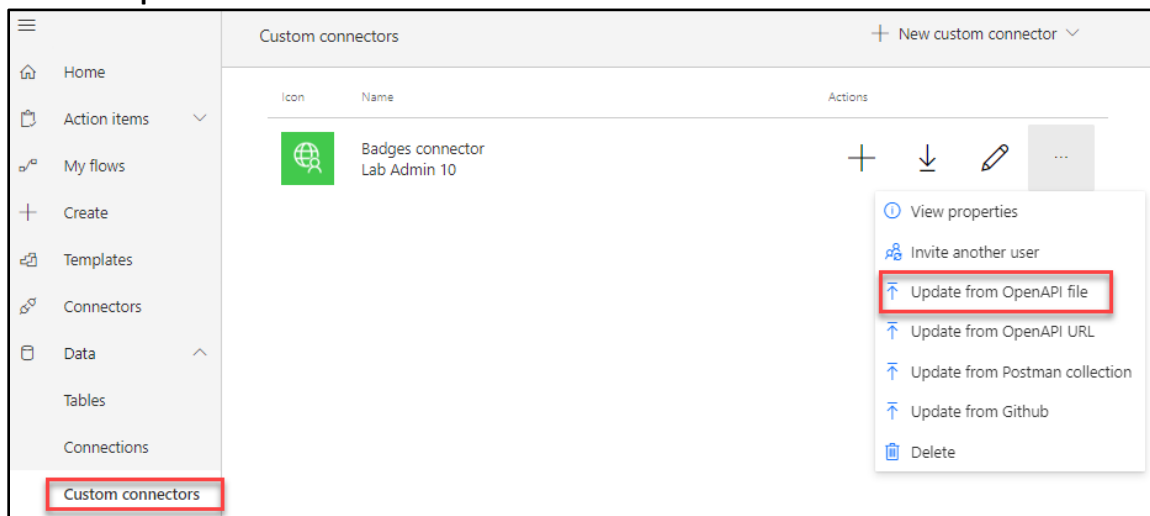
☐ Connect via on-premises data gateway [Learn more](#)

Scheme *
☒ HTTPS ☐ HTTP

Host *
contosobadgestest.azurewebsites.net

Base URL

7. Select **Custom connectors** from the sitemap.
8. Click on the ... **More actions** button of the custom connector you created and select **Update from Open API file**.



9. Click **Import**.
10. Select the **swagger.json** file you saved to your machine and click **Open**.
11. Click **Continue**.
12. Enter **Connector for badges** for Description, **contosobadgestest.azurewebsites.net** for Host, and advance to **Security**.

icon background color

#007ee5

Description

Connector for badges

☐ Connect via on-premises data gateway [Learn more](#)

Scheme *

☒ HTTPS ☐ HTTP

Host *

contosobadgestest.azurewebsites.net

Base URL

/

Security →

13. Review the security configuration and advance to **Definition**.
14. Do not navigate away from this page.

Task 3: Modify the definition

1. Select the **AddCredit** action.
2. Select **Important** for Visibility.

Actions (4)

Actions determine the operations that users can perform. Actions can be used to read, create, update or delete resources in the underlying connector.

☒ 1 AddCredit ...

☐ 2 GetRecipient ...

☐ 3 ListBadges ...

☐ 4 ListRecipients ...

[+ New action](#)

Triggers (0)

Triggers read data in from your connector. A trigger focuses on a particular event that happens, say a new Contact or Order

General

Summary [Learn more](#)

Add Credit

Description [Learn more](#)

Add Credit

Operation ID *

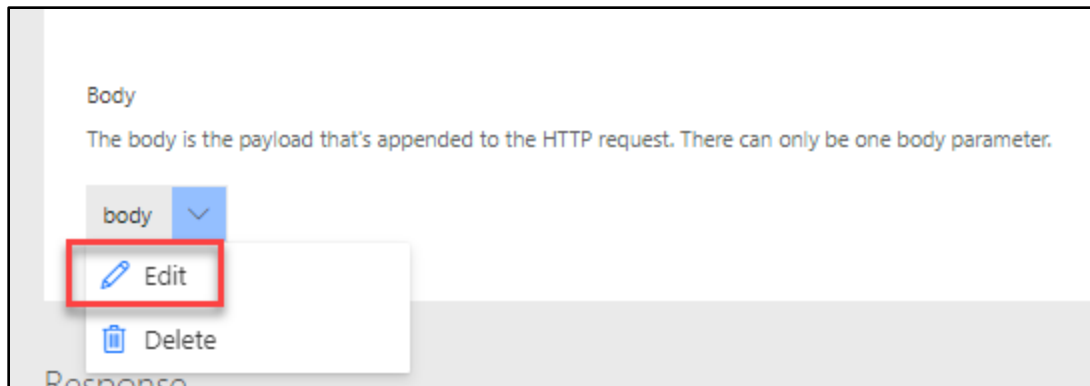
This is the unique string used to identify the operation.

AddCredit

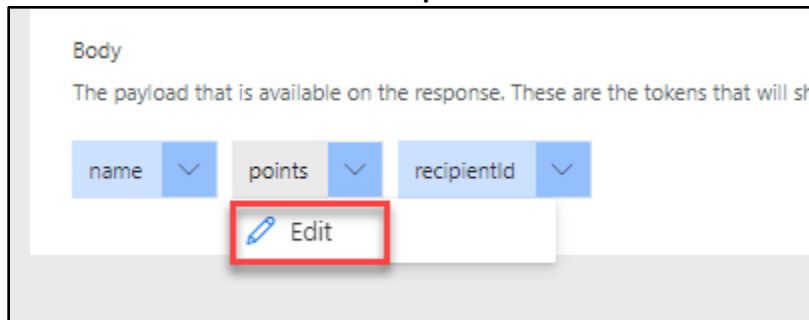
Visibility [Learn more](#)

☐ none ☐ advanced ☐ internal ☒ important

3. Scroll down to the **Request** section, click on the chevron button of the **body** and select **Edit**.



4. Click on the chevron button of **points** and select **Edit**.



5. Select **Yes** for Is required and click on the **← Back** button.

6. Click on the chevron button of **recipientid** and select **Edit**.
7. Select **Yes** for Is required and click on the **← Back** button.
8. Click on the chevron button of **name** and select **Edit**.
9. Select **Yes** for Is required and click on the **← Back** button.
10. Click on the **← Back** button again.

← Back

Parameter

Name *

body

Is required?

☐ Yes ☒ No

Visibility [Learn more](#)

☒ none ☐ advanced ☐ internal ☐ important

Location *

☐ Path ☐ Query ☐ Header ☒ Body

Body

The payload that is available on the response. These are the tokens that will show up as the outputs in designer.

* name * points * recipientId

11. Advance to **Code**.
12. Review the code and advance to **Test**.
13. Click **Update connector** and wait for the connector to be updated

4. Code (Preview) > 5. Test

Swagger Editor

✓ Update connector X Close

Connections

Selected connection *

None

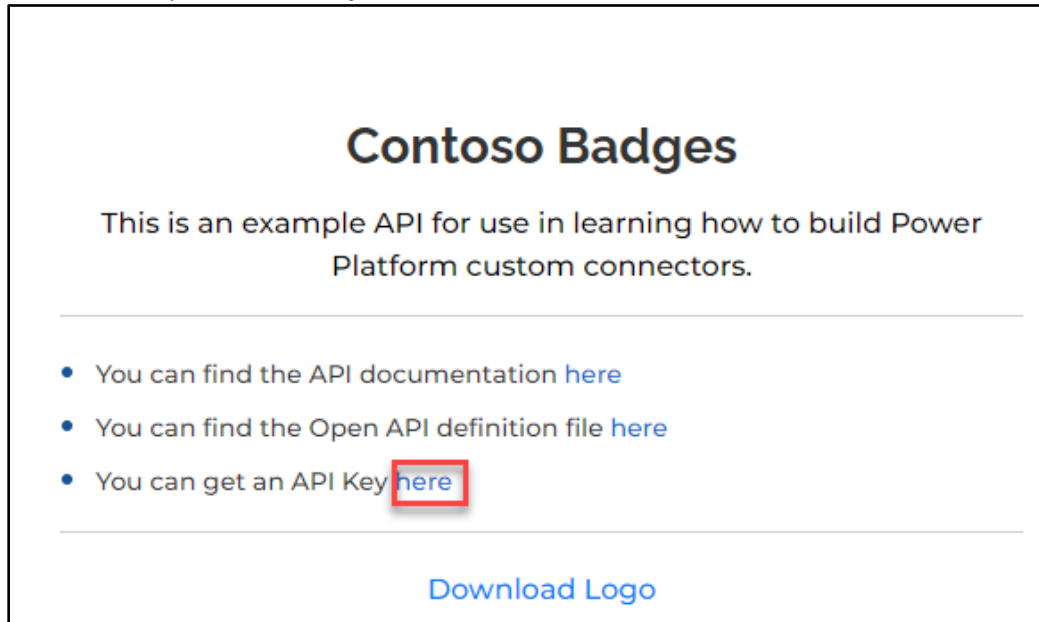
+ New connection

14. Do not navigate away from this page.

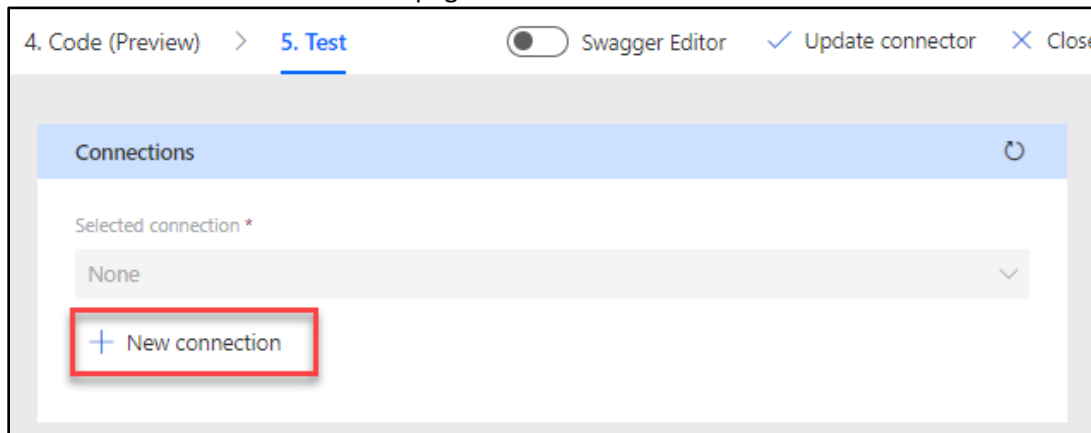
Task 4: Test connector

1. Open a new browser tab or window and navigate to [Contoso Coffee Badges \(contosobadgestest.azurewebsites.net\)](https://contosobadgestest.azurewebsites.net)

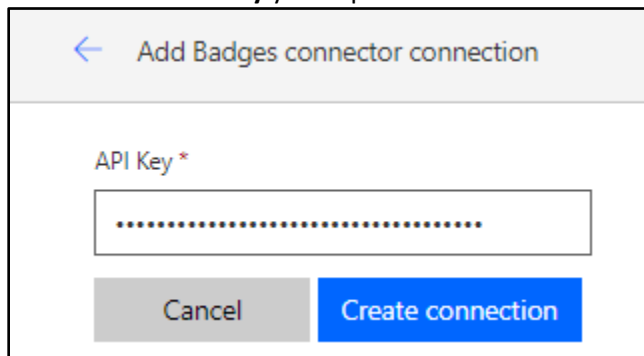
2. Click on open the **API Key** link



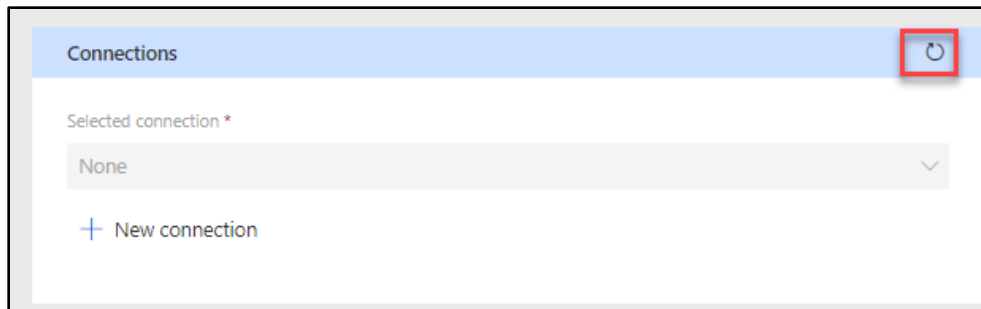
3. Copy the **API Key**. Save this key you will use it multiple times.
4. Go back to the connector test page and click **+ New Connection**.



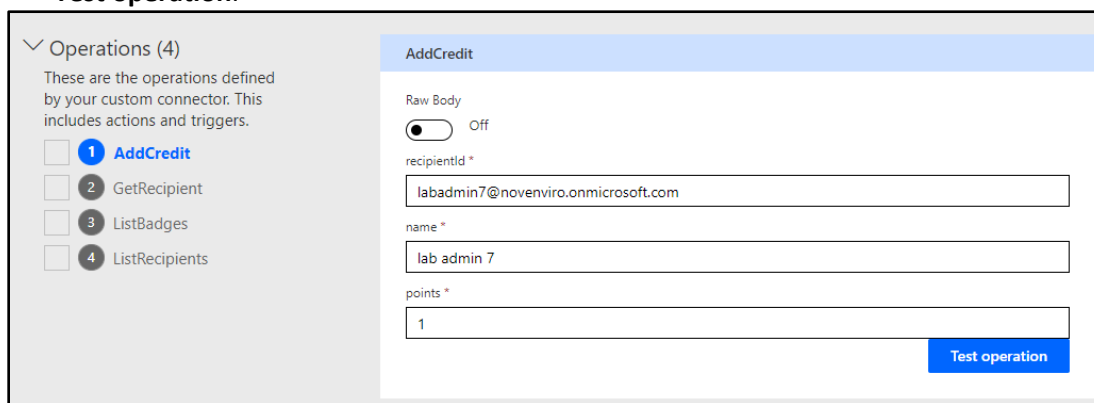
5. Paste the **API Key** you copied and click **Create connection**.



6. Click on the **Refresh** connections button.



7. The connection you created should get selected.
8. Go to the **AddCredit** operation.
9. Enter your email address for **recipientId**, enter your name for **name**, enter **1** for points, and click **Test operation**.



10. The test should succeed, and the response should look like the image below.



Exercise 2 – Add Custom Code

In this exercise, you will add a new operation to only return the current badge name and image URL. You will do this by using the custom code feature to reshape the response from the API.

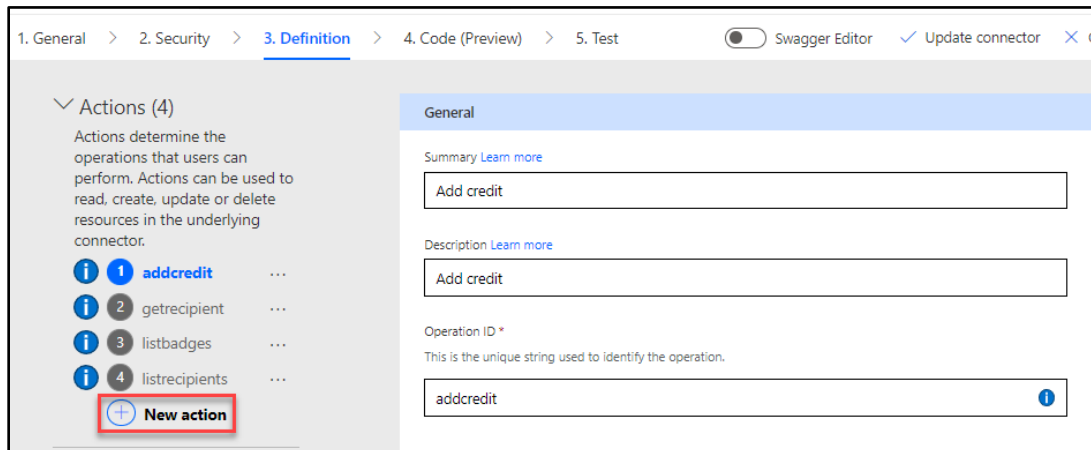
Note: If you already have the custom connector open, skip to step 4

Task 1: Add code from resource folder

1. Navigate to [Power Automate](#)
2. Expand **Data** and select **Custom connectors**.
3. Click the **Edit** button of the custom connector you created.



4. Select the **Definition** tab and click **New action**.



5. Enter **Get current badge** for Summary, **Get current badge** for Description, and **getcurrentbadge** for Operation ID.

Actions (5)

Actions determine the operations that users can perform. Actions can be used to read, create, update or delete resources in the underlying connector.

- 1 addcredit ...
- 2 getrecipient ...
- 3 listbadges ...
- 4 listrecipients ...
- 5 **getcurrentba...** ...

[+ New action](#)

Triggers (0)

Triggers read data in from your...

General

Summary [Learn more](#)

Get current badge

Description [Learn more](#)

Get current badge

Operation ID *

This is the unique string used to identify the operation.

getcurrentbadge

Visibility [Learn more](#)

☒ none ☐ advanced ☐ internal ☐ important

6. Scroll down to the **Request** section and click **+ Import from sample**.

Request

It defines the pre-requirements needed in order to make a request. Describes a single operation parameter. A unique parameter is defined by a combination of a name and location.

[+ Import from sample](#)

Response

7. Select **Get** for Verb, enter <https://contosobadgestest.azurewebsites.net/getcurrentbadge?id={id}> for URL, and click **Import**.

Verb *

☒ GET ☐ DELETE ☐ POST ☐ PUT ☐ HEAD ☐ OPTIONS

☐ PATCH

URL *

https://contosobadgestest.azurewebsites.net/getcurrentbadge?id={id}

This is the request URL.

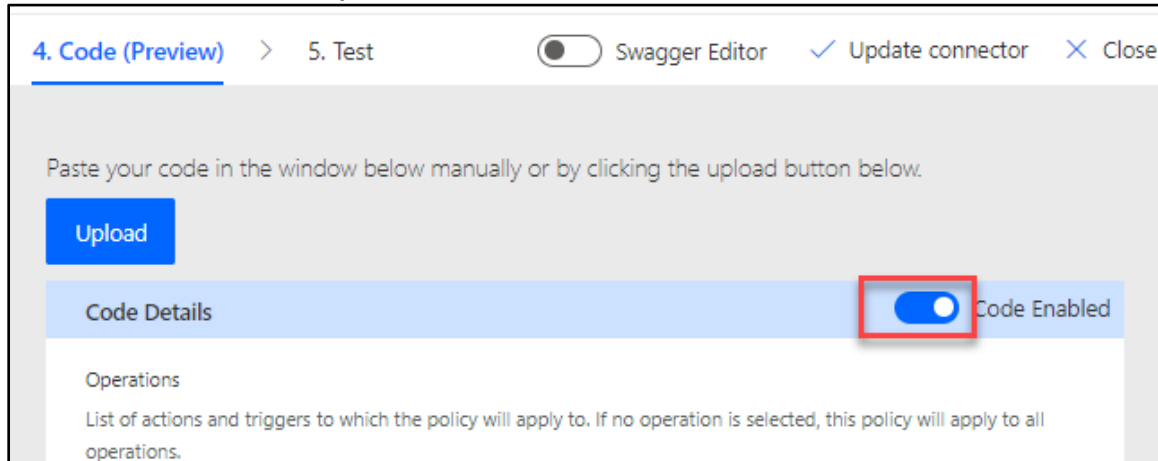
Headers

Headers separated by a new line, e.g.:
Content-Type application/json
Accept application/json

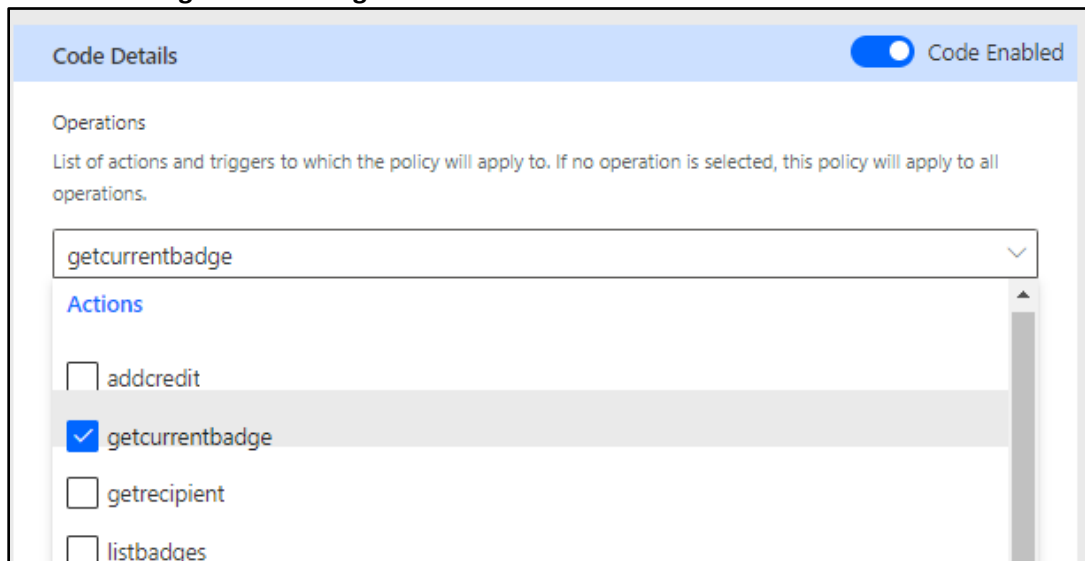
These are custom headers that are part of the request.

Import **Close**

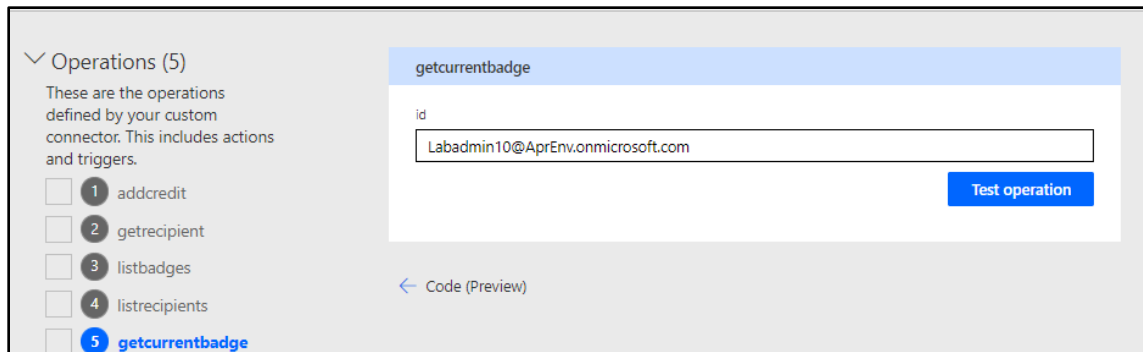
8. Click **Update connector** and wait for the connector to be updated.
9. Select the **Code** tab.
10. Enable **Code** and click **Upload**.



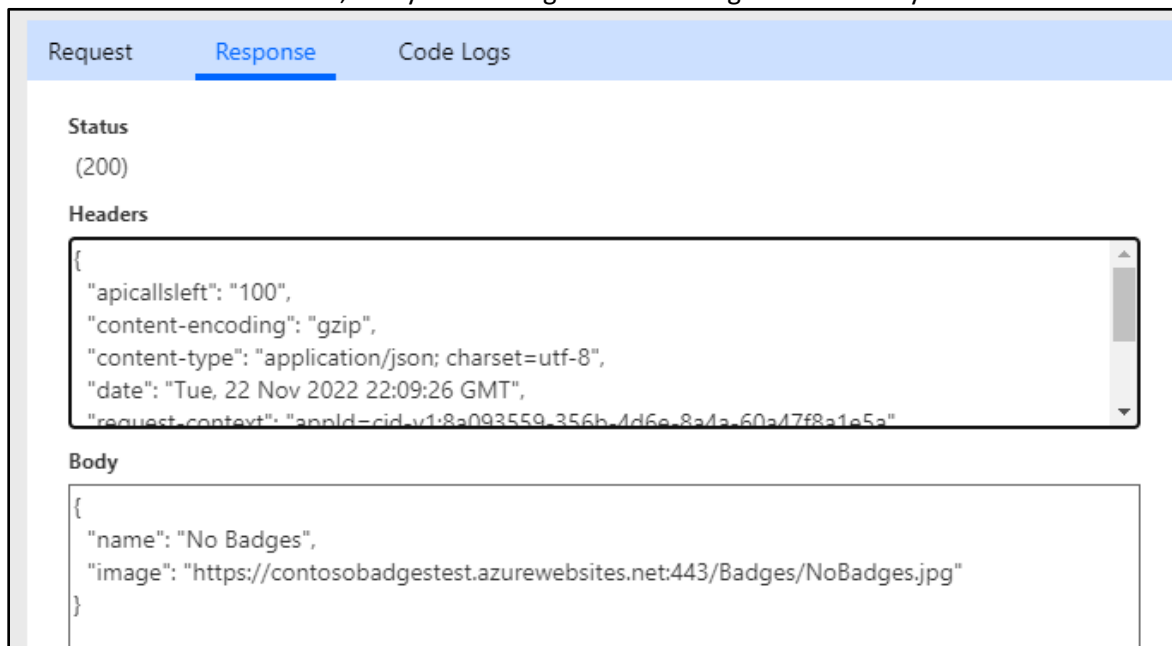
11. Select the **CustomConnectorCode.csx** file located in the lab resources folder and click **Open**.
12. Select the **getcurrentbadge** action.



13. Review the code you just added.
14. Click **Update connector** and wait for the connector to be updated.
15. Advance to **Test**.
16. Select the **getcurrentbadge** action.
17. Provide your email address as id and click **Test operation**.



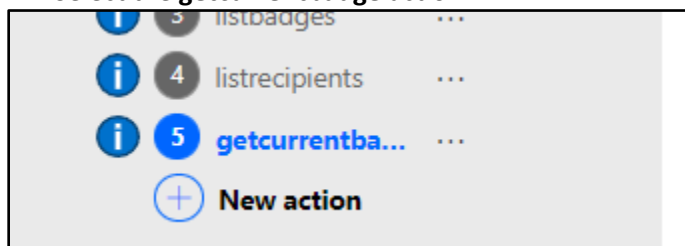
18. The test should succeed, and you should get current badge for the user you created.



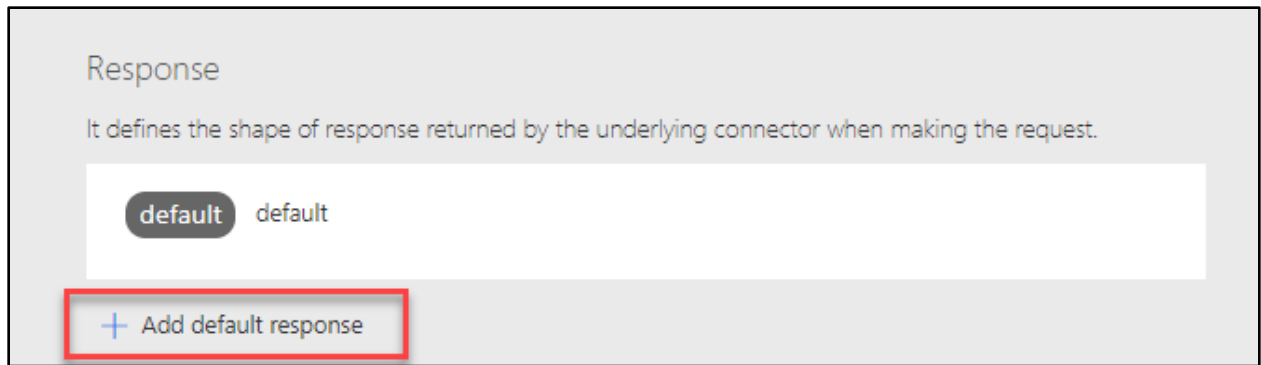
19. Copy the Response **Body** JSON.

20. Select the Definition tab.

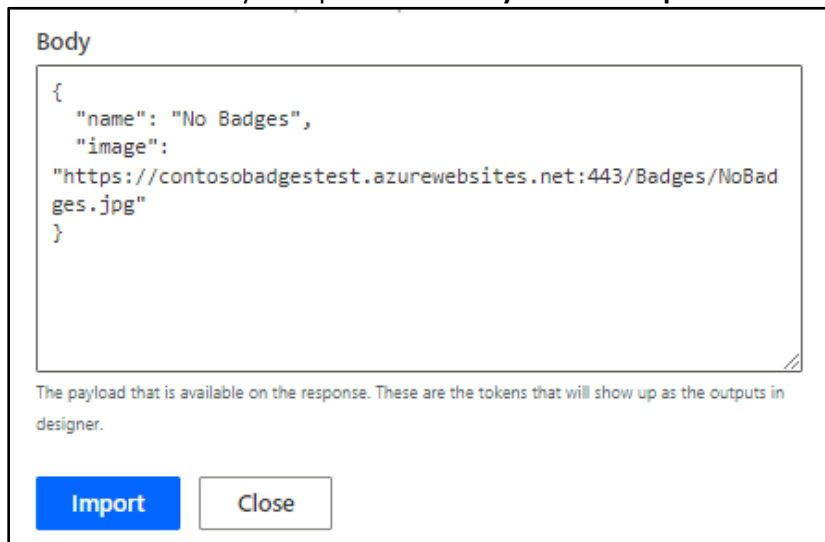
21. Select the **getcurrentbadge** action.



22. Scroll down to the **Response** section and click **+ Add default response**.



23. Paste the JSON you copied in the **Body** and click **Import**.



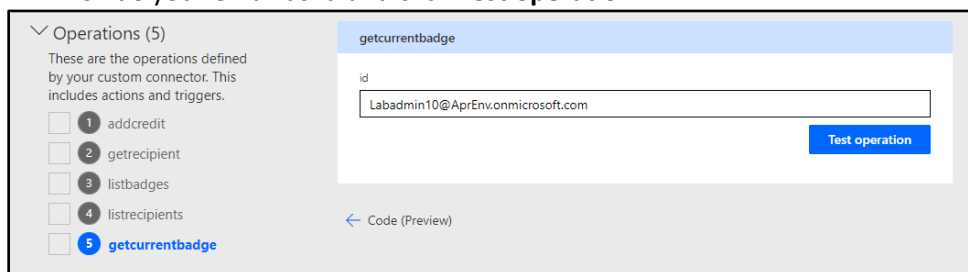
24. Click **Update connector** and wait for the connector to be updated.

25. Do not navigate away from this page.

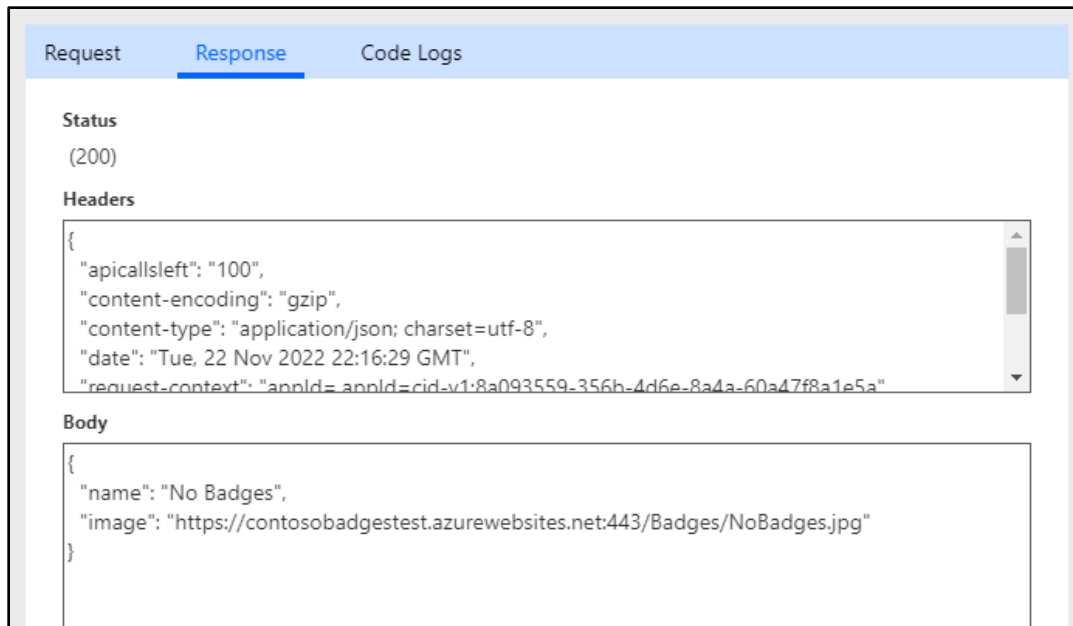
Task 2: Test custom code

In this task, you will test your custom code.

1. Select the **Test** tab.
2. Select the connection you created earlier.
3. Go to the **Operations** section and select the **getcurrentbadge** operation.
4. Provide your email as **id** and click **Test operation**.



5. The operation should succeed, and the response **Body** should look like the image below.



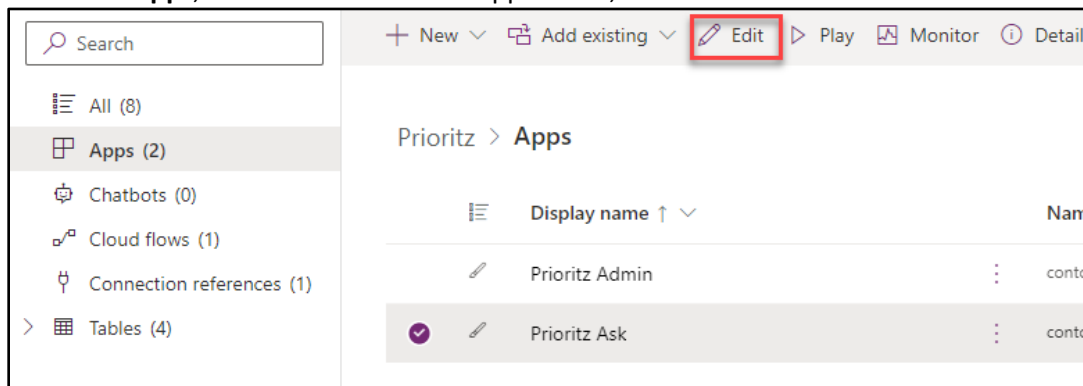
Exercise 3 – Use Custom Connector

In this exercise, you will test the custom connector you created using a flow and a canvas application.

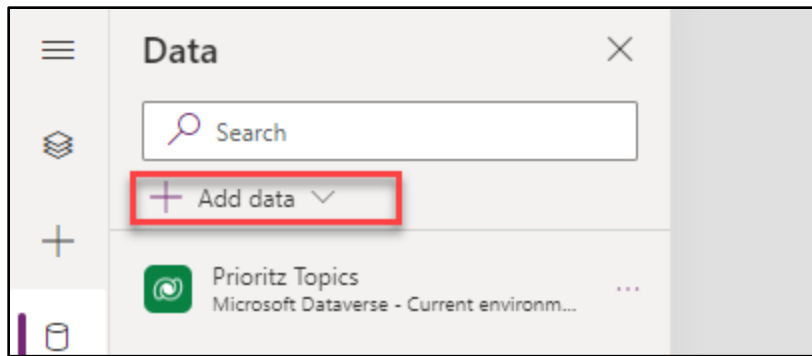
Task 1: Test connector from canvas app

In this task, you will use the custom connector you created to show the user's current badge on the PrioritZ Ask canvas application.

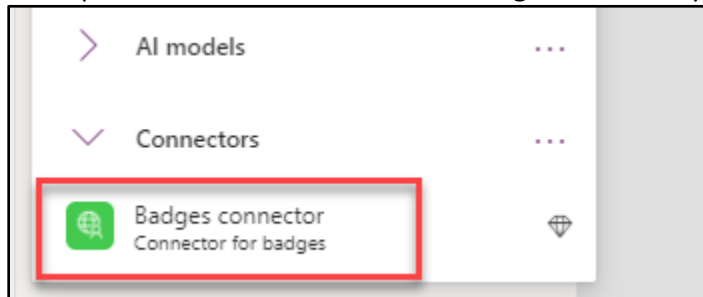
1. Navigate to [Power Apps maker portal](#) and make sure you are in your dev environment.
2. Expand **Solutions** and open the **PrioritZ** solution.
3. Select **Apps**, select the **PrioritZ Ask** application, and click **Edit**.



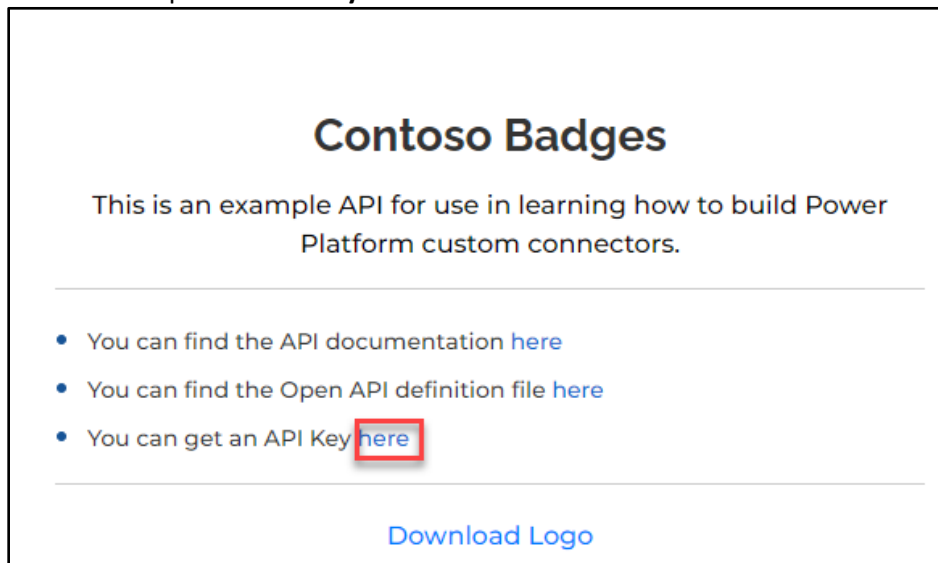
4. Select **Data** from the left and click **+ Add data**.



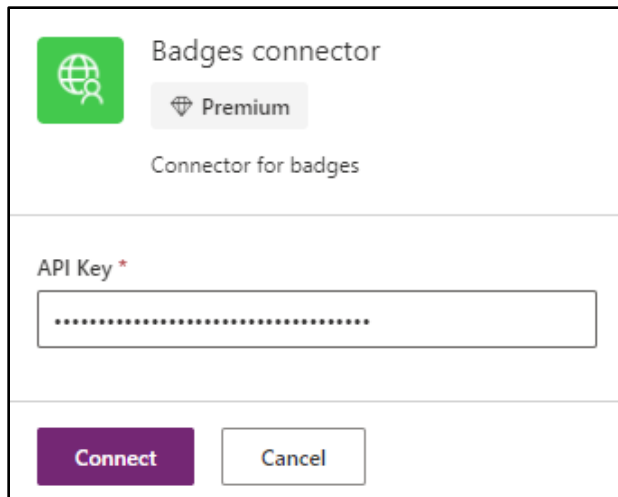
5. Expand **Connectors** and select the **Badges connector** you created.



6. Click **+ Add a connection**.
7. Open a new browser tab or window and navigate to [Contoso Coffee Badges \(contosobadgestest.azurewebsites.net\)](https://contosobadgestest.azurewebsites.net)
8. Click on open the **API Key** link

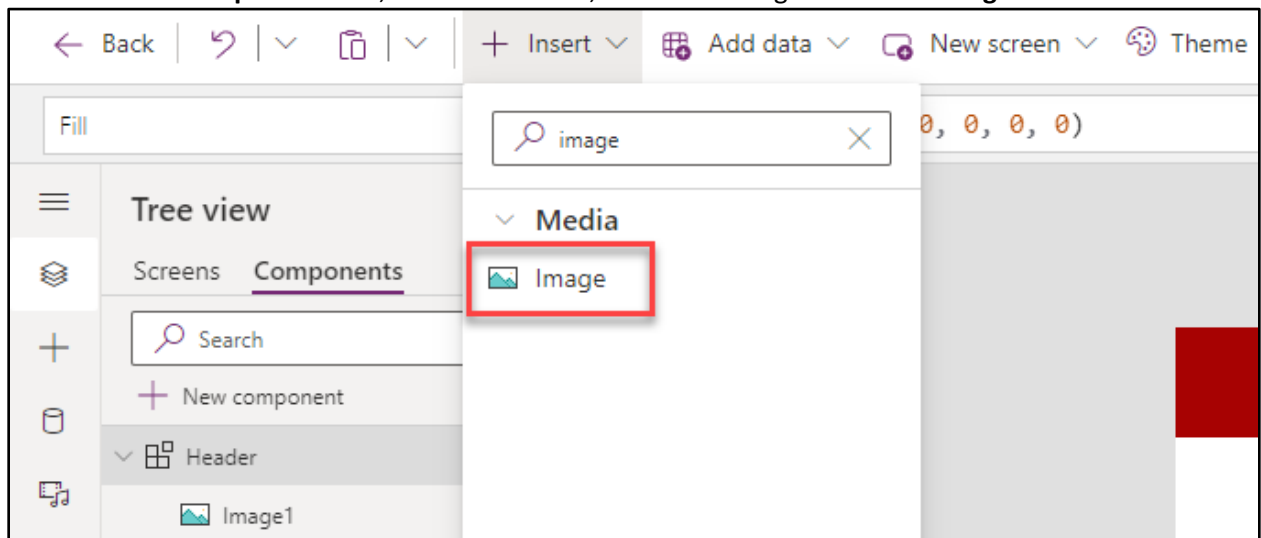


9. Copy the **API Key**. Keep this API Key in a notepad, you will need it again.
10. Go back to the app designer, paste the **API Key** you copied, and click **Connect**.

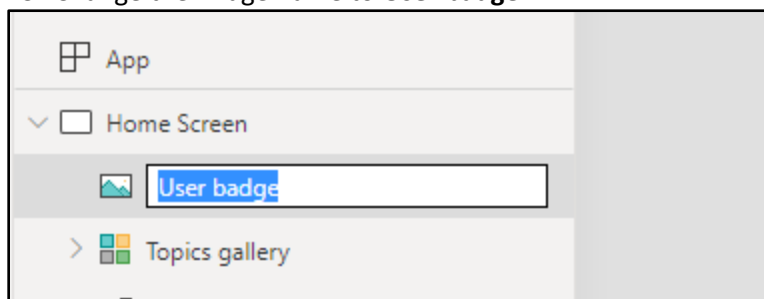


11. Select the **Tree view**.

12. Select the **Components** tab, click **+ Insert** tab, search for image and select **Image**.

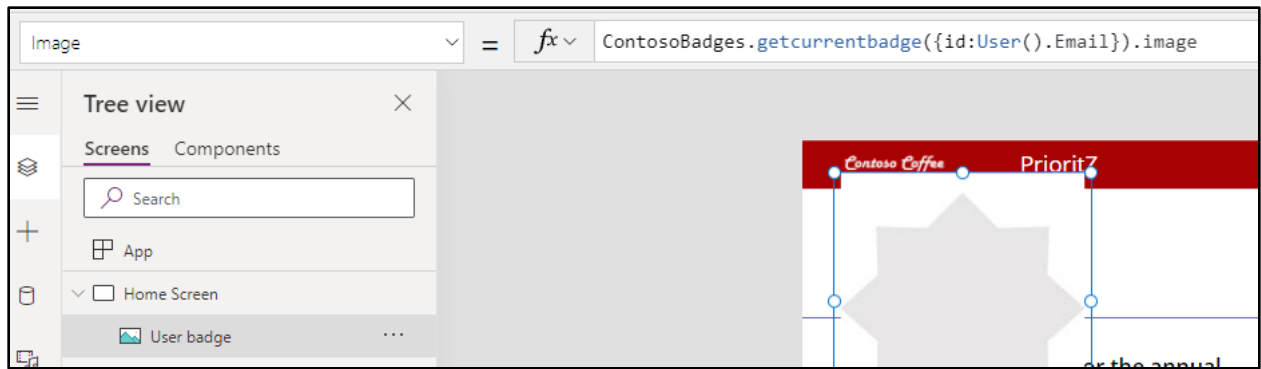


13. Change the Image name to **User badge**.



14. Set the User badge **Image** value to the formula below.

```
ContosoBadges.getcurrentbadge({id:User().Email}).image
```

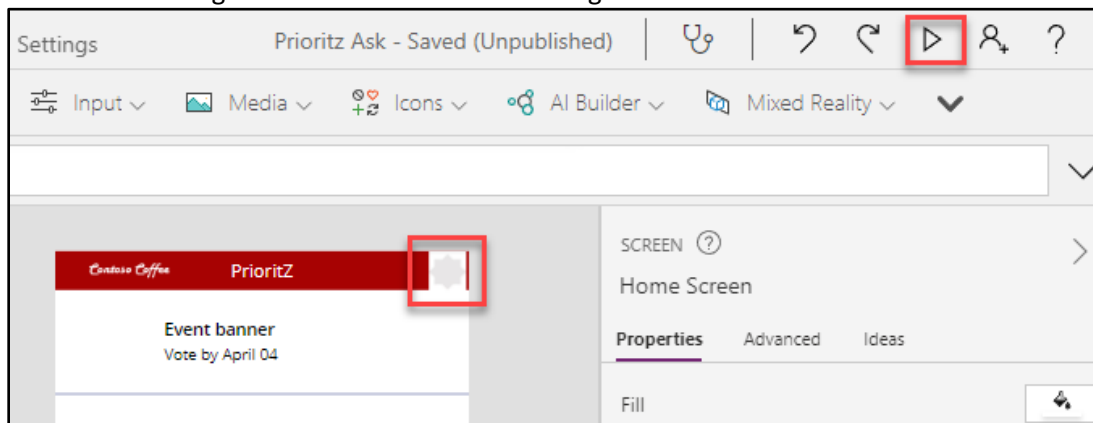


15. Set the Tooltip value of the User badge to the formula below.

`ContosoBadges.getcurrentbadge({id:User().Email}).name`

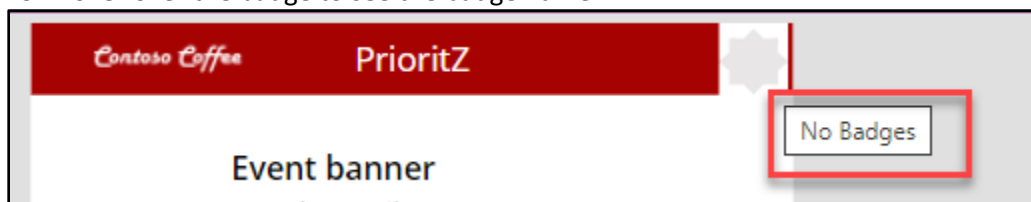
16. Make the image smaller and move it to top right corner of the screen.

17. The User badge should now look like the image below.



18. Select **Screens** tab in the Tree view. Click **Play** button.

19. Hover over the badge to see the badge name.

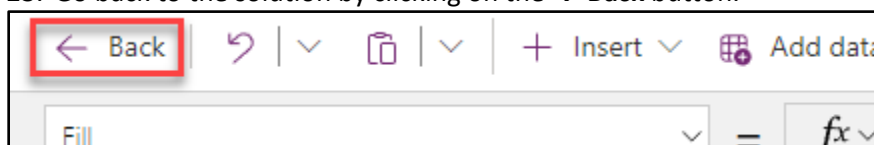


20. Close the preview.

21. Click **Publish**.

22. Select **Publish this version**.

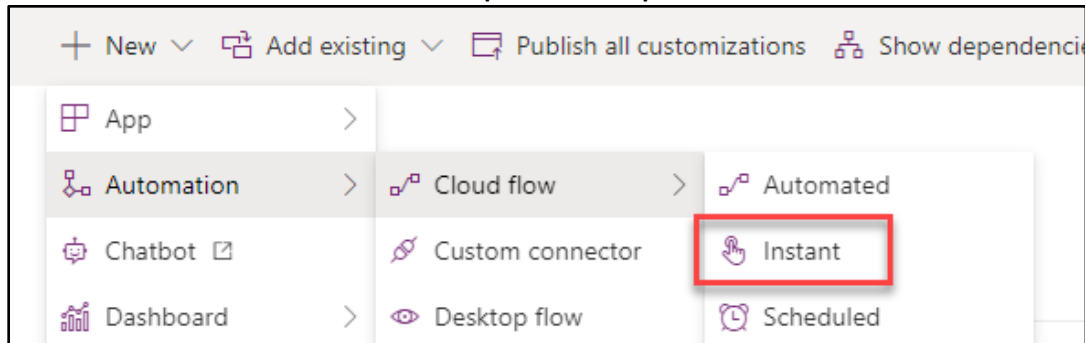
23. Go back to the solution by clicking on the **← Back** button.



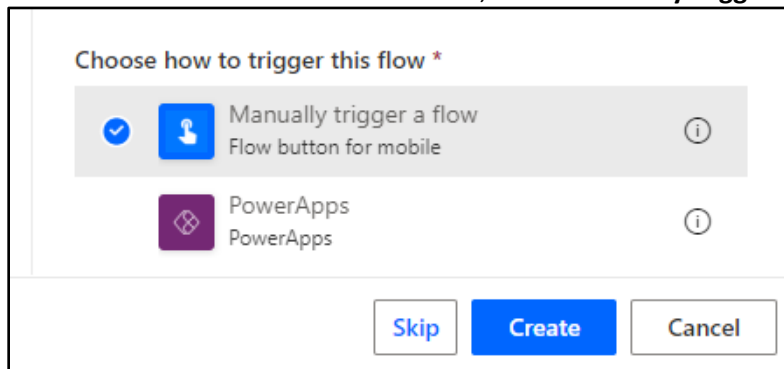
24. Do not navigate away from this page.

Task 2: Test connector from flow (*this task is optional*)

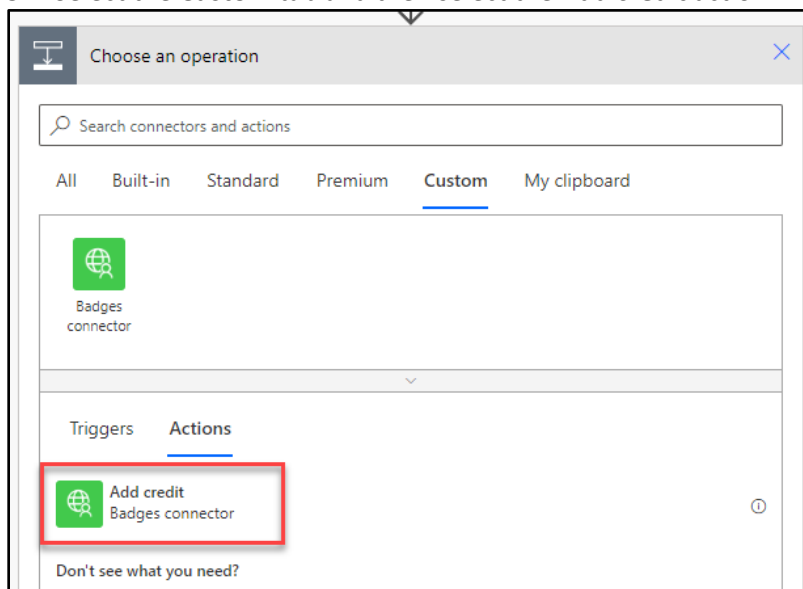
1. Make sure you are still in the **PrioritZ** solution.
2. Click **+ New** and select **Automation | Cloud flow | Instant**.



3. Enter **Test add credit** for flow name, select **Manually trigger a flow**, and click **Create**.



4. Click **+ New step**.
5. Select the **Custom** tab and then select the **Add credit** action.



6. Enter **Test connection**, paste the **API Key** you copied earlier, and click **Create**.

Badges connector

* Connection name: Test connection

* API Key: [Masked]

Create

7. Click on the **recipientId** field, go to the Dynamic content pane, and select **User email**.

Add credit

recipientId: User email x

name: [Empty]

points: [Empty]

+ New step Save

Dynamic content pane:

- user
- Manually trigger a flow
- User name: The display name of the user who triggered the flow.
- User email: The email address of the user who triggered the flow.**

8. Click on the **name** field, go to the Dynamic content pane, and select **User name**.

9. Enter **1** for points and click **Save**. Wait for the flow to be saved.

Manually trigger a flow

Add credit

recipientId: User email x

name: User name x

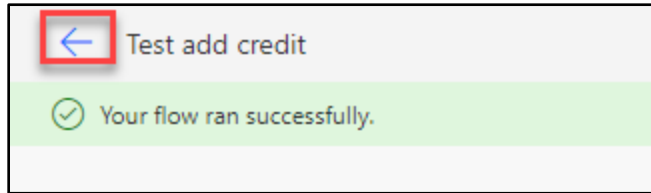
points: 1

+ New step Save

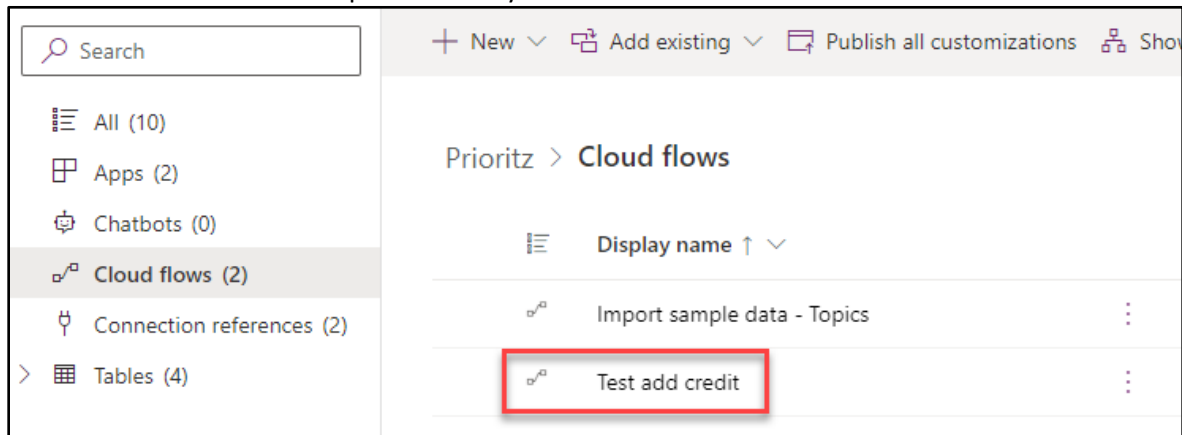
10. Click **Test**.

11. Select **Manually** and click **Test** again.

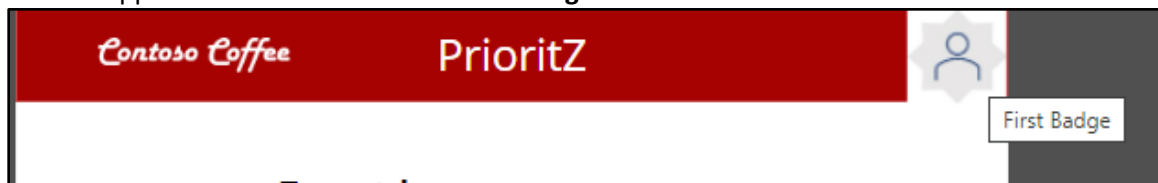
12. Click **Continue**.
13. Click **Run flow**.
14. Click **Done**.
15. The flow run should succeed.
16. Click on the ← back button.



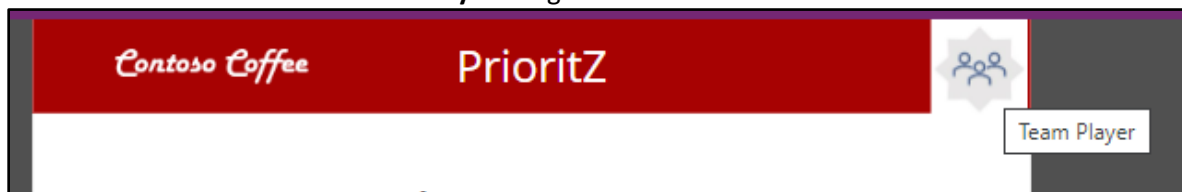
17. Select **Cloud flows** and open the flow you created.



18. Start a new browser window and navigate to [Power Apps maker portal](#).
19. Make sure you are the correct environment.
20. Select **Apps** and launch the **Prioritz Ask** application.
21. The application should now show **First Badge**.




22. Go back to flow and run it couple more times.
23. Go back to the **Prioritz Ask** application and refresh the page.
24. You should now see the **Team Player** badge.



25. Go to the flow and run it two more times.
26. Go back to the **Prioritz Ask** application and refresh the page.
27. You should now see the **Champ** badge

Contoso Coffee

PrioritZ



Champ