

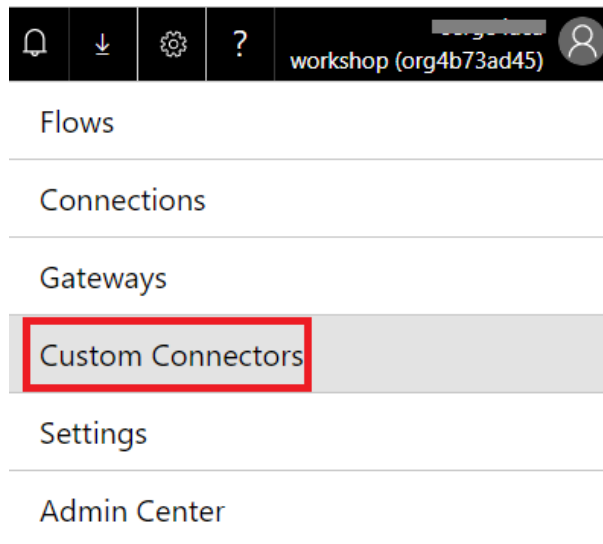
Lab 11. Create a Custom Connector

Author: Serge Luca aka "Doctor Flow"

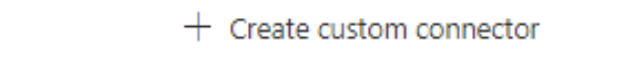
Prerequisites: creating and calling a custom connector requires a Premium connector.

Tasks:

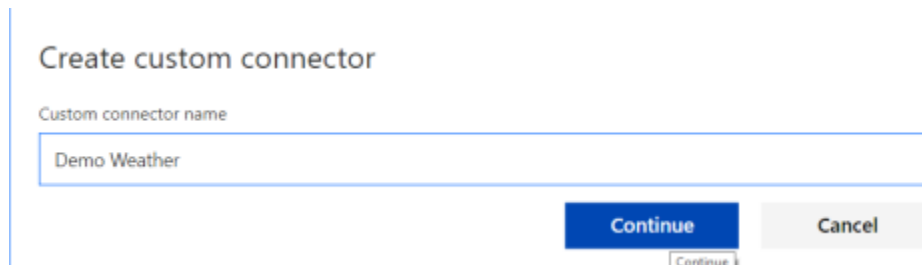
1. Before starting this lab, make sure your API key has been generated. You can reuse the key from the previous lab.
2. Go to the **Connector** menu and select **Custom Connectors**:



3. Click on **Create custom connector**:




4. Select **Create from blank**.
5. Name the connector "Demo Weather" and click Continue

A screenshot of the 'Create custom connector' form. The title 'Create custom connector' is at the top. Below it, the label 'Custom connector name' is followed by a text input field containing the text 'Demo Weather'. At the bottom right, there are two buttons: 'Continue' (in blue) and 'Cancel' (in grey). A small 'Continue' link is also visible below the 'Continue' button.

6. Click **Continue** and in the next window, provide the host (**api.weatherstack.com**) and a short description of what your connector does and also select http instead of https:

General information


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

Icon background color

Description

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

Scheme *

☐ HTTPS ☒ HTTP

Host *

Base URL

7. Click **Security** to move to the next screen.
8. The authentication type should be **API key**.
9. Since we want the Key parameter to be provided in the query string, create an API key with **key** as Parameter label and Parameter name; switch the parameter location to Query as illustrated in the following picture:

Authentication type

Choose what authentication is implemented by your API *

API Key



 Edit

API Key

Users will be required to provide the API Key when creating a connection

Parameter label *

key

Parameter name *

access_key

Parameter location *

Query



 Edit

10. Click **Definition**:

1. General > 2. Security > **3. Definition** > 4. Test

▼ Actions (0)

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

Actions

Start by adding an action or trigger on the left.

← Security

+ 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 being created and provides the relevant data so that users can take action on that event.

+ New trigger

▼ References (0)

References are reusable parameters used by both actions and triggers.

11. Click on **New action**:

12. Define the Action as follow:

1. General > 2. Security > **3. Definition** > 4. Test

✓ Create connector

▼ Actions (1)

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

! 1 GetWeather ...

+ 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 being created and provides the relevant data so that users can

General

Summary [Learn more](#)

Get weather

Description [Learn more](#)

Get weather in a specific city

Operation ID *

This is the unique string used to identify the operation.

GetWeather

Visibility [Learn more](#)

☒ none ☐ advanced ☐ internal ☐ important

13. In a new browser tab, type a weather request to make sure it works fine and also to generate sample data that we will reuse in our connector definition (don't forget to pass your key as a parameter, as well as the city):

← → ↻ 🏠 ⓘ Not secure | api.weatherstack.com/current?access_key=96380665c7c7dc1b14b87493ee253a0d&query=paris

```
[{"request":{"type":"City","query":"Paris, France","language":"en","unit":"m"},"location":{"name":"Paris","country":"France","region":"Ile-de-France","lat":48.8566,"localtime_epoch":1569442080,"utc_offset":"+2.0"},"current":{"observation_time":"06:08 PM","temperature":18,"weather_code":116,"weather_icons":["https://assets.weatherstack.com/images/wsymbols01_png_64/wsymbol_0004_black_low_cloud.png"],"weather_descriptions":["Partly cloudy"],"wind_speed":24,"wind_degree":240,"wind_dir":"WSW","pressure":1008,"precip":0.6,"humidity":73,"cloudcover":75,"feelslike":18,"uv_index":0,"visibility":10000}]
```

14. Keep this tab open and go back to the connector definition.
15. Down below in **Request**, click **Import from sample**

General

Summary [Learn more](#)

Get weather

Description [Learn more](#)

Get weather in a specific city

Operation ID *

This is the unique string used to identify the operation.

GetWeather

Visibility [Learn more](#)

☒ none ☐ advanced ☐ internal ☐ important

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

16. Pass your query string, set the verb to **Get** and click **import**

Import from sample



Verb *

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

URL *

`http://api.weatherstack.com/current?access_key=96380665c7c7dc1b14`

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

17. 2 parameters will be visible in a Query: **access_key, query**.
18. The access_key will be registered in the connector, in such way that there is no need to pass the key for each query; therefore we can delete it:

Request

+ Import from sample

Verb *

The verb describes the operations available on a single path.

GET

URL *

This is the request URL.

https://api.weatherstack.com/current

Path

Path is used together with Path Templating, where the parameter value is actually part of the operation's URL.

Query

Query parameters are appended to the URL. For example, in /items?id=####, the query parameter is id.

access_key ...

query ...

Edit

Delete

s that are part of the request.

19. Edit the **query** parameter and fill in the **Description** as "Fill in the City Name", **Summary** as "city name" and make the fields required:

Parameter

Name *

City

Description [Learn more](#)

Fill in the city name

Summary [Learn more](#)

City name

Default value

Is required?

☒ Yes ☐ No

Visibility [Learn more](#)

☒ none ☐ advanced ☐ internal ☐ important

Location *

☐ Path ☒ Query ☐ Header ☐ Body

Type

string

Format

Dropdown type [Learn more](#)

☒ Disabled ☐ Static ☐ Dynamic

20. Click on **Back**, scroll to **Response**, click on **default** to import another sample; the scroll bar is in the middle of the screen as illustrated in the picture:

Request

+ Import from sample

Verb *

The verb describes the operations available on a single path.

GET

URL *

This is the request URL.

http://api.weatherstack.com/current

Path

Path is used together with Path Templating, where the parameter value is actually part of the operation's URL.

Query

Query parameters are appended to the URL. For example, in /items?id=####, the query parameter is id.

* City ...

Headers

These are custom headers that are part of the request.

Body

The body is the payload that's appended to the HTTP request. There can only be one body parameter.

Response

It defines the shape of response returned by the underlying connector when making the request.

default default

21. In the next window, click on **Import from sample**:

← Back

Response + Import from sample

* Name
default

Headers
These are custom headers that are part of the response.

References Used
The following are the references used by this entity.

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

* key-body-output ...

Validation
This helps you identify potential issues with this response.

Validation ✓

Validation succeeded.

22. ...and in the next window, paste the JSON result grabbed from the browser:

✕ Cancel

+ Import from sample

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 response.

Body

```
{
  "request": {
    "type": "City",
    "query": "Paris, France",
    "language": "en",
    "unit": "m",
    "location": {
      "name": "Paris",
      "country": "France",
      "region": "Île-de-France",
      "lat": "48.867",
      "lon": "2.333",
      "timezone_id": "Europe/Paris",
      "localtime": "2019-09-25 19:37",
      "localtime_epoch": 1569440220,
      "utc_offset": "2.0",
      "current": {
        "observation_time": "05:37 PM",
        "temperature": 18,
        "weather_code": 116,
        "weather_icons": [
          "https://assets.weatherstack.com/images/wsymbols01_png_64/wsymbol_0002_sunny_intervals.png"
        ],
        "weather_descri"
      }
    }
  }
}
```

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

Import Close

23. Click **Import**.

24. Click **Create Connector**

4. Test

✓ Create connector

✕ Cancel

← Back

Response

+ Import from sample

Name *

default

Headers

These are custom headers that are part of the response.

References Used

The following are the references used by this entity

Body

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

cloudcover

feelslike

humidity

is_day

observation_time

precip

pressure

temperature

uv_index

visibility

weather_code

current.weather_de...

current.weather_ic...

wind_degree

wind_dir

wind_speed

country

lat

localtime

localtime_epoch

lon

name

region

timezone_id

utc_offset

language

query

type

unit

25. You can now test the connector (click Test):

1. General
>
2. Security
>
3. Definition
>
4. Test

✓ Update conn

Test operation

Test a specified operation of this custom connector using the selected connection. You must update the custom connector in order to test recent changes.

Connections

Selected connection *

None

+ New connection

Operations (1)

These are the operations defined by your custom connector. This includes actions and triggers.

1

GetWeather

GetWeather

q *

Test operation

26. However, you need to create a New connection; click on **New connection**, provide your key, and click again on Create connection:

← Add Demo Weather connection


key *

.....|

Cancel Create connection

27. You will be redirected to the previous page, where you can click the refresh icon in order to display the new connection:

Test operation
Test a specified operation of this custom connector using the selected connection. You must update the custom connector in order to test recent changes.

Connections 

Selected connection *

Demo Weather (Created at 2019-09-25T19:08:53.5179402Z) ▾

+ New connection

✓ **Operations (1)**
These are the operations defined by your custom connector. This includes actions and triggers.

1 GetWeather

GetWeather

query

london

Test operation

28. Provide a city name, like London, click **Test Operation** and you should get the corresponding weather

Operations (1)

These are the operations defined by your custom connector. This includes actions and triggers.

1 GetWeather

GetWeather

query

london

Test operation

RequestResponse

Status

(200)

Headers

```
{
  "cf-ray": "51bf6f2a4b559d24-AMS",
  "content-encoding": "gzip",
  "content-type": "application/json; Charset=UTF-8",
  "date": "Wed, 25 Sep 2019 19:34:13 GMT",
  "x-amplayer-transaction-id": "6d9b80e7-ee72-434d-b951-9a3327de4245"
}
```

Body

```
{
  "request": {
    "type": "City",
    "query": "London, United Kingdom",
    "language": "en",
    "unit": "m"
  },
  "location": {
    "name": "London",
    "country": "United Kingdom",
    "region": "City of London, Greater London",
    "lat": "51.517",
    "lon": "-0.106",
    "timezone_id": "Europe/London"
  }
}
```

29. Click close, and you will notice that your connector has been created:

Custom connectors

+ New custom connector

Name
<div> <div> <div>Demo Weather</div> <div>serge luca</div> </div> <div> <div>+</div> <div>↓</div> <div>✎</div> <div>...</div> </div> </div>

30. You can now create an instant Flow from blank to use this custom connector

Build an instant flow



Triggered manually from any device, easy-to-share instant flows

Flow name

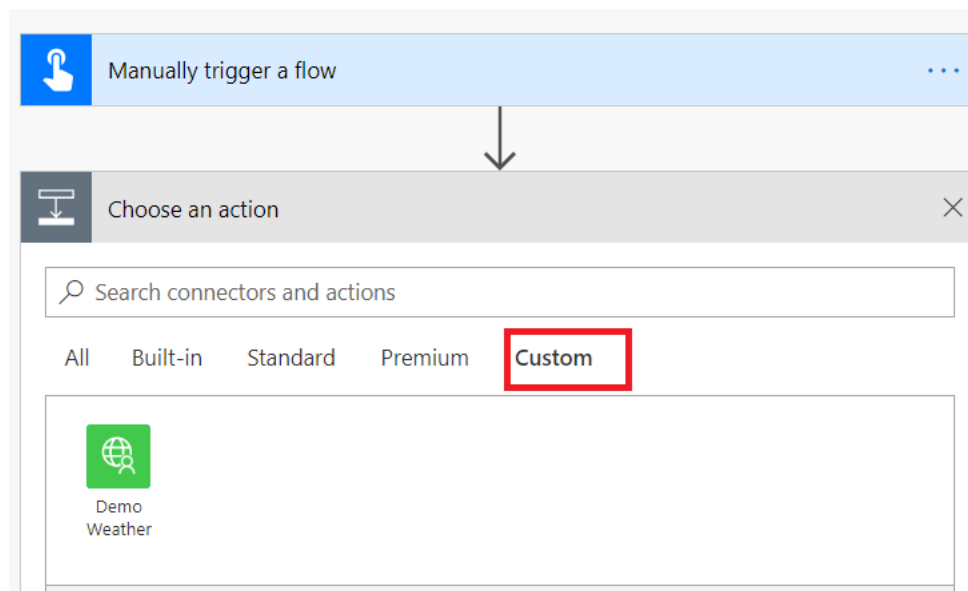
weather flow

Choose how to trigger this flow *

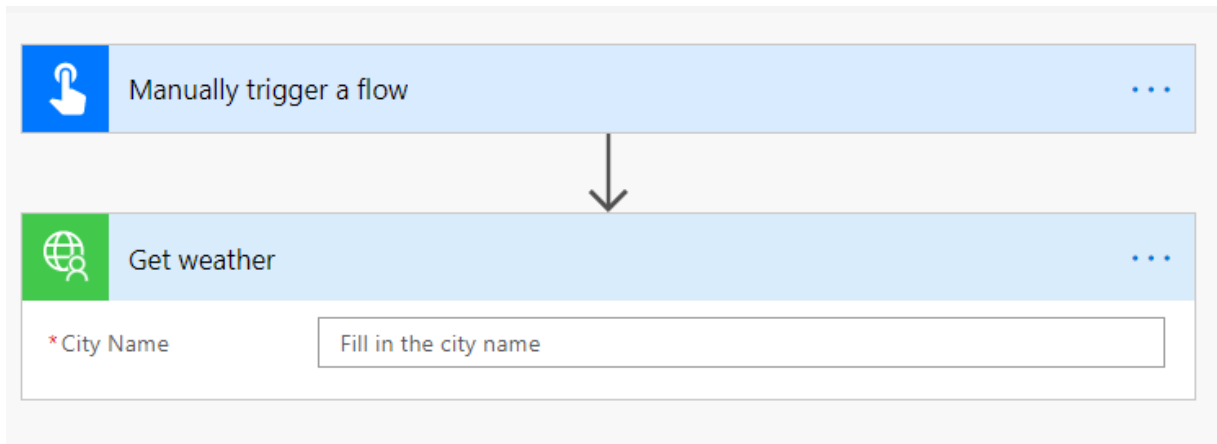
☒ From Microsoft Flow
Microsoft Flow

☐ From PowerApps
PowerApps

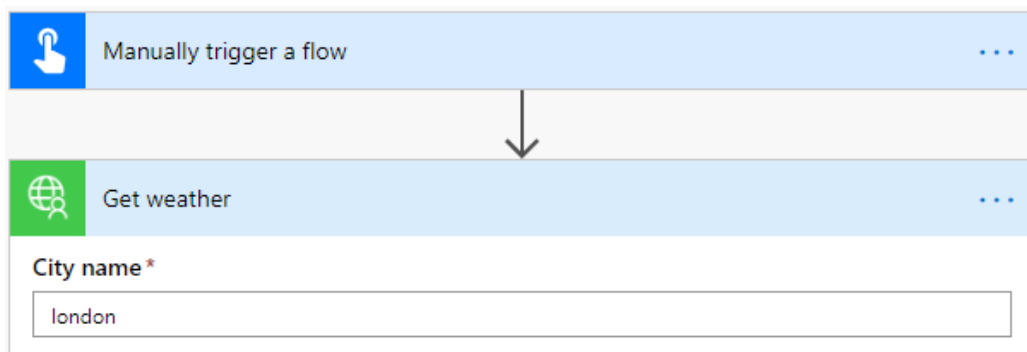
31. Add an "action" from the Custom category; you should find you Demo Weather custom connector:



32. Select the **Get Weather** action



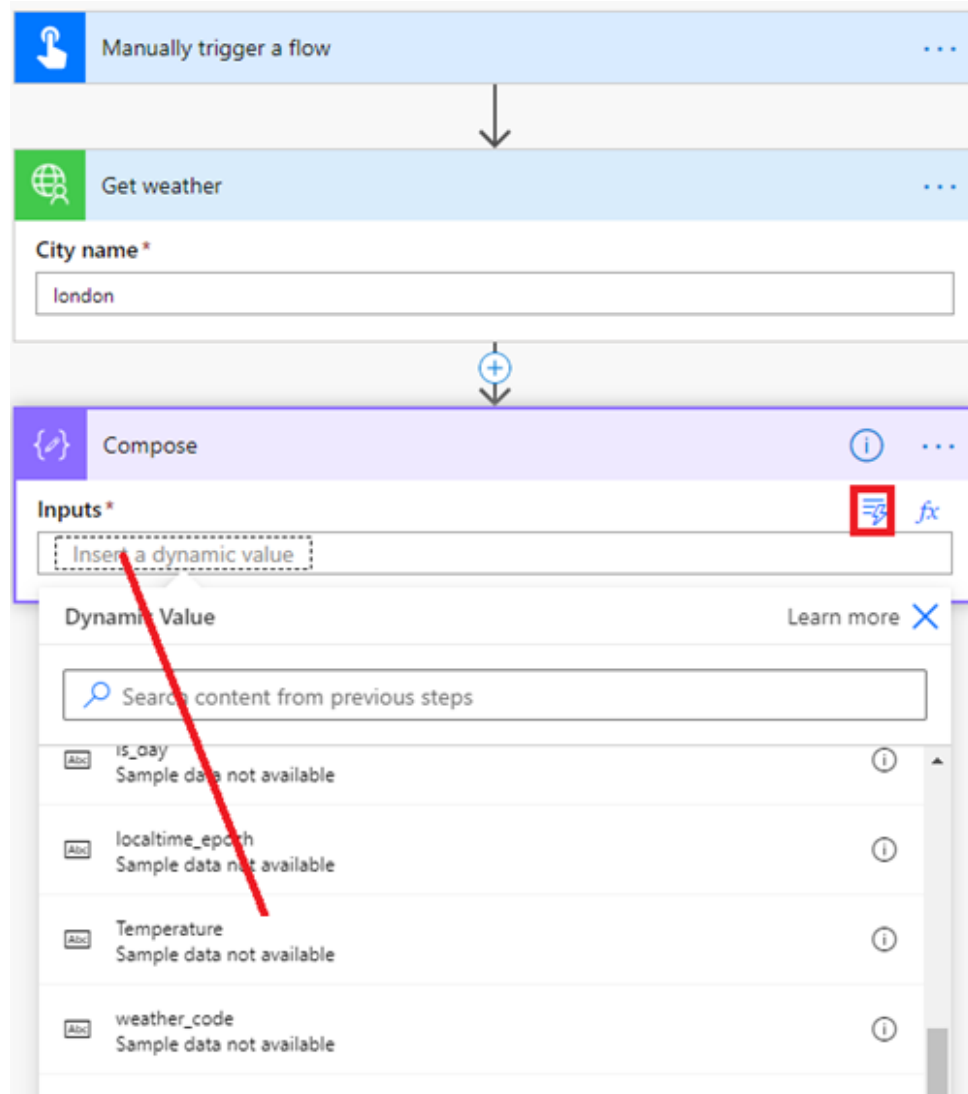
33. Provide London as the city name and save your Flow:



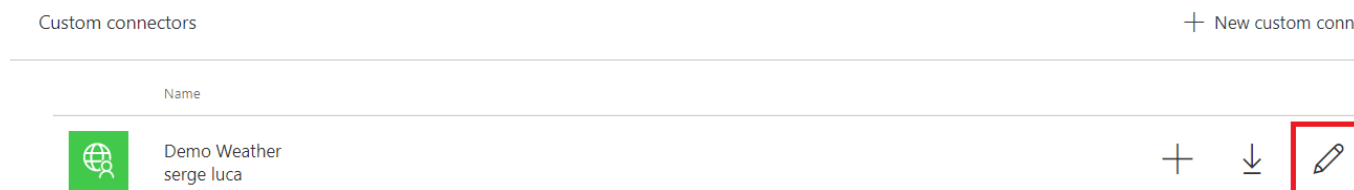
34. Run the Flow and check the output of "Get Weather":

timezone_id	Europe/London
localtime	2020-05-12 15:57
localtime_epoch	1589299020
utc_offset	1.0
observation_time	02:57 PM
Temperature	14
weather_code	113
weather_icons	"https://assets.weatherstack.com/images/wsymbols01_png_64/wsymbol

Add a **Compose** and store the **temperature** value:



35. As you can see, using a custom connector is more convenient than using the HTTP action. However we can make it even easier to use for power users: for instance, instead of having to use the name **Temperature**, we can modify the custom connector definition with **temperature in Celcius**.
36. Go back to your custom connector definition and click the Edit icon:



37. Go to Definition and click Default in Response:

1. General > 2. Security > **3. Definition** > 4. Test

+ New trigger

✓ **References (0)**
References are reusable parameters used by both actions and triggers.

✓ **Policies (0)**
Policies are used to change the behavior of actions and triggers through configuration. You can use one or more policies from a set of predefined templates.

+ New policy

Request

It defines the pre-requirements needed in order to make a request. A unique parameter is defined by a combination of the verb and the URL.

Request

Verb *
The verb describes the operations available on a single resource.

GET

URL *
This is the request URL.

`http://api.weatherstack.com/current`

Path
Path is used together with Path Templating, where the path is a template and the path parameters are the values that are substituted into the template.

Query
Query parameters are appended to the URL. For example, `http://api.weatherstack.com/current?units=metric`.

query ...

Headers
These are custom headers that are part of the request.

Body
The body is the payload that's appended to the HTTP request.

Response

It defines the shape of response returned by the request.

default default

38. Edit temperature:

Response + Import from sample

Name *

default

Headers

These are custom headers that are part of the response.

References Used

The following are the references used by this entity

Body

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

cloudcover ...

feelslike ...

humidity ...

is_day ...

observation_time ...

precip ...

pressure ...

temperature ...

uv_index ...

visibility ...

weather_code ...

current.weather_de... ...

current.weather_ic... ...

wind_degree ...

wind_dir ...

wind_speed ...

country ...

lat ...

localtime ...

localtime_epoch ...

lon ...

name ...

region ...

timezone_id ...

utc_offset ...

language ...

query ...

type ...

unit ...

39. Change the Title and the description:

Schema Property

Title

Temperature in celcius

Description [Learn more](#)

temperature in celcius in the selected city

Visibility [Learn more](#)

☒ none
 ☐ advanced
 ☐ internal
 ☐

Type

integer ▼

4. Test

✓ Update connector

← Back

Schema Property

Title

Temperature in celcius

Description [Learn more](#)

temperature in celcius in the selected city

40. Update your Flow; you should see the new property label:

