# Contents

[Project Overview 2](#_Toc172721491)

[Objective 2](#_Toc172721492)

[Tools and Technologies 2](#_Toc172721493)

[API Details 2](#_Toc172721494)

[Steps Undertaken 2](#_Toc172721495)

[Step 1: Create a Power App 2](#_Toc172721496)

[Step 2: Create a Custom Connector 2](#_Toc172721497)

[General Information 2](#_Toc172721498)

[Host and Base URL 3](#_Toc172721499)

[Security 4](#_Toc172721500)

[Define the Action 4](#_Toc172721501)

[Test the Connector 6](#_Toc172721502)

[Step 3: Configure the Power App 6](#_Toc172721503)

[Add Data Source 6](#_Toc172721504)

[Add Controls 6](#_Toc172721505)

[Step 4: Validate and Test 6](#_Toc172721506)

[Schema Validation 6](#_Toc172721507)

[Testing the App 6](#_Toc172721508)

[Conclusion 7](#_Toc172721509)

[Recommendations 7](#_Toc172721510)

[Appendices 7](#_Toc172721511)

[Appendix A: Full Power Apps Formula 7](#_Toc172721512)

[Dropdown On Select Property 7](#_Toc172721513)

[Text Field 8](#_Toc172721514)

[Appendix B: JSON Response Schema 8](#_Toc172721515)

Project Documentation: Integrating a REST API with Power Apps

# Project Overview

## Objective

The objective of this project is to create a Power App that calls a web service to fetch weather data using a REST API and displays the result within the app. This project demonstrates the integration of external web services within Power Apps, enhancing its functionality for business applications.

## Tools and Technologies

Microsoft Power Apps  
REST API  
Custom Connectors in Power Apps

## API Details

API Key: 598dca88bdd74b0abf903135242307  
Base URL: https://api.weatherapi.com/v1  
Endpoint: /current.json

# Steps Undertaken

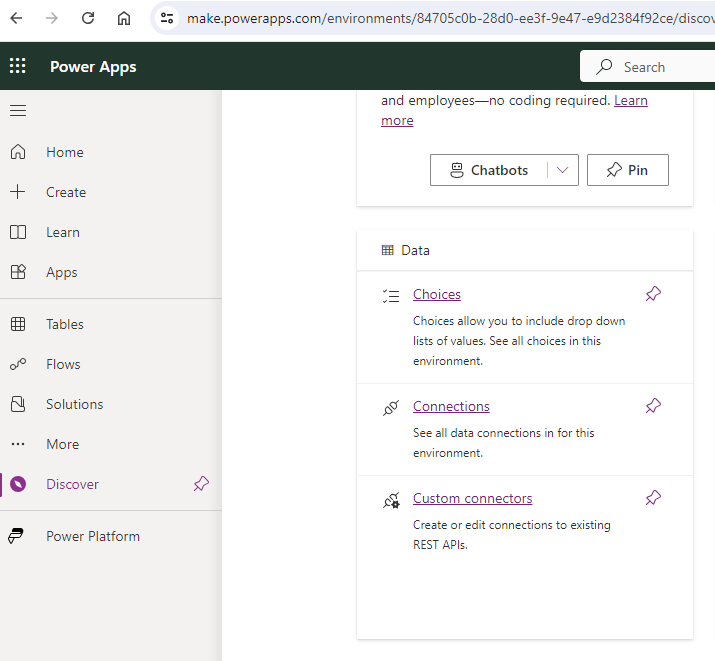
## Step 1: Create a Power App

1. Sign in to Power Apps:  
 - Navigate to Power Apps (https://make.powerapps.com/).  
 - Sign in with your Microsoft account.  
  
2. Create a New Canvas App:  
 - Click on “Create” and choose “Canvas app from blank.”  
 - Provide a name for your app and select the layout (Tablet or Phone).  
 - Example: \*\*Name:\*\* sampleapitest

## Step 2: Create a Custom Connector

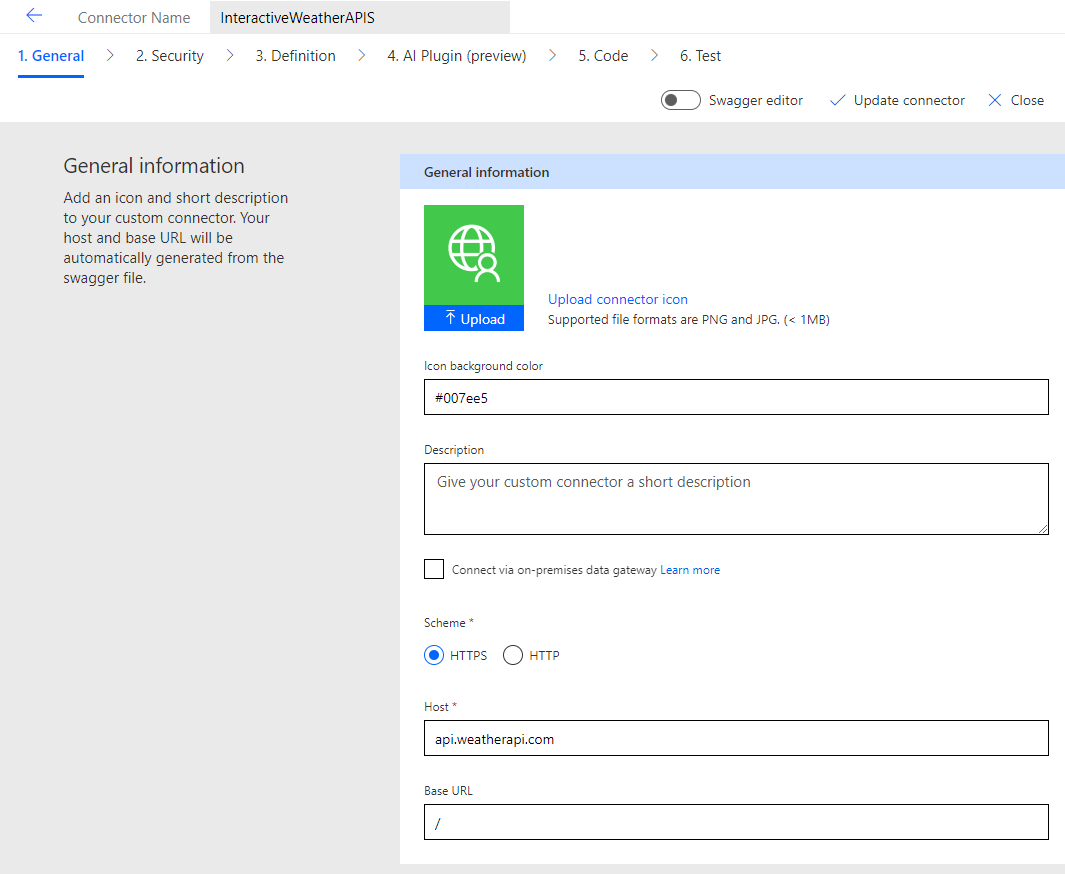
### General Information

Connector Name: InteractiveWeatherAPIS



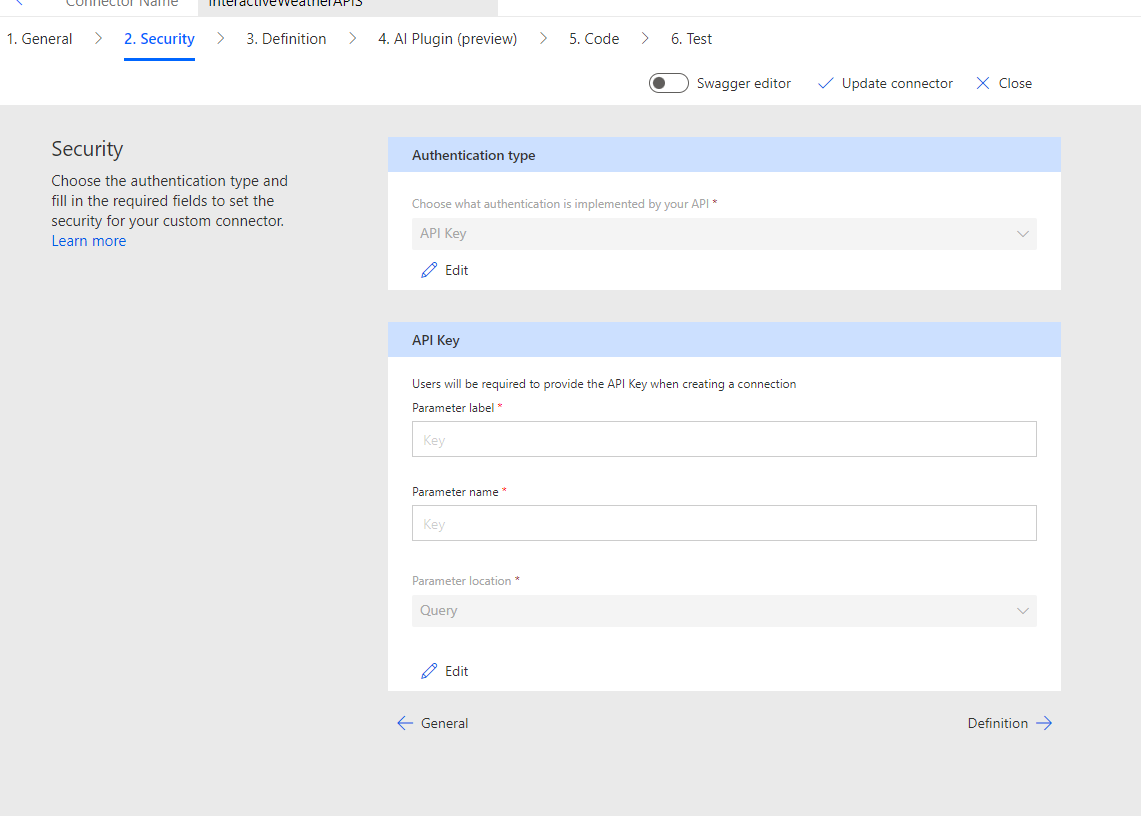
### Host and Base URL

Host: api.weatherapi.com  
Base URL: /v1



### Security

Authentication Type: API Key  
Parameter Label: key  
Parameter Name: key  
Location: Query



### Define the Action

Action Name: GetWeatherData  
Summary: Fetch weather data for a specific location  
Request URL: /current.json  
HTTP Method: GET  
Query Parameters:  
 - Name: q  
 - Location: Query  
 - Required: Yes

A screenshot of a computer

Description automatically generated A screenshot of a computer screen

Description automatically generated

Response:

Select the default one and add the response

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

### Test the Connector

Tested the connector with valid parameters (q and key) to ensure proper functioning.

## Step 3: Configure the Power App

### Add Data Source

1. Add the Custom Connector:  
 - In your Canvas app, click on Data in the left pane.  
 - Click on + Add data.  
 - Find and add your custom connector (InteractiveWeatherAPIS).

### Add Controls

1. Dropdown Control (Dropdown1):  
 Text property: Items: ["London", "New York", "Sydney"]  
 On Select Property:  
 Set(temperature, InteractiveWeatherAPIS.GetWeatherData({q: Dropdown1.SelectedText.Value}).current.temp\_f)  
  
2. Text field:  
 Text Property: temperature

## Step 4: Validate and Test

### Schema Validation

Updated the response schema in the custom connector to match the actual data types returned by the API:  
Changed properties from "integer" to "number" where necessary (e.g., temp\_c, feelslike\_c, etc.).

### Testing the App

Ran the app, selected a city from the dropdown, and the temperature was correctly fetched and displayed in the text field

A computer screen with a white screen

Description automatically generated

# Conclusion

The Power App was successfully configured to call a REST API using a custom connector. The application dynamically fetches and displays weather data based on the selected location from a dropdown menu. The project demonstrates the integration of external web services within Power Apps, enhancing its functionality for business applications.

## Recommendations

Error Handling: Implement additional error handling to manage API call failures and display appropriate messages to users.  
User Interface: Enhance the UI for better user experience, including loading indicators during API calls.  
Scalability: Consider adding more parameters and options for detailed weather data retrieval.

# Appendices

## Appendix A: Full Power Apps Formula

### Dropdown On Select Property

Set(temperature, InteractiveWeatherAPIS.GetWeatherData({q:Dropdown1.SelectedText.Value}).current.temp\_f)

Text property: Items: ["London", "New York", "Sydney"]

### Text Field

Text Property: temperature

## Appendix B: JSON Response Schema

{  
 "current": {  
 "temp\_c": 16,  
 "temp\_f": 60.8  
 }  
}