
Microsoft Power Platform Connectors

Welcome to the Microsoft Power Platform Connectors open source repository. This repository contains custom connectors, certified connectors, and related tools to facilitate connector development for Azure Logic Apps, Microsoft Power Apps, and Microsoft Power Automate.

Custom Connectors

The `custom-connectors` folder contains fully functional connector samples which can be deployed to the Power Platform for extension and use. If you are looking to publish a connector to the Power Platform, please explore Certified Connectors and Independent Publisher Connectors.

Certified Connectors

The `certified-connectors` folder contains certified connectors which are built by partners who own the end service of their connector. These connectors are deployed and available out-of-box within the Power Platform for use. One requirement of our connector certification program is that new certified connectors be open sourced for community contributions. The `certified-connectors` folder is managed by the Microsoft Connector Certification Team to ensure that within the `master` branch, the connector version is identical to that deployed in the Power Platform. The `dev` branch is maintained by the connector owner and the Microsoft Connector Certification Team to allow community development of the connector prior to certification and deployment of a version.

Independent Publisher Connectors

The `independent-publisher-connectors` folder contains connectors that are submitted by publishers (MVPs, developers, and companies) that do not own the underlying service behind their connector. These connectors are deployed and available out-of-box within the Power Platform as premium connectors. Anyone can submit a new connector to this folder, add functionality to connectors in this folder, and resolve issues related to the connectors in this folder. The folder is managed by the Independent Publisher Connector Community, which includes Independent Publishers and Project Coordinators. The master branch is maintained by the Microsoft Connector Certification Team, who ensures that the connector version is identical to that deployed in the Power Platform. The dev branch is maintained by the connector maintainer(s) and the Microsoft Connector Certification Team to allow community development of the connector prior to certification and deployment of a version. Click [here](#) to view the Independent Publisher Connector Manifesto.

Contributing

This project welcomes contributions and suggestions. Most contributions require you to agree to a Contributor License Agreement (CLA), which declares that you have the right to, and actually do, grant us the rights to use your contribution. For details, visit <https://cla.microsoft.com>.

Files to Include

Please submit the following files: An Open API 2.0 swagger definition, an API properties file, and a README.md.

API Definition (Swagger) File

The API definition, also known as the swagger, describes the API for the custom connector using the OpenAPI specification.

For further details, see the `apiDefinition.swagger.json` JSON schema.

API Properties File

The API properties file contains some properties for the custom connector. These properties are not part of the API definition. It contains information such as the brand color, authentication information, etc. A typical API properties file looks like the following:

```
1 {
2   "properties": {
3     "capabilities": [],
4     "connectionParameters": {
5       "api_key": {
6         "type": "securestring",
7         "uiDefinition": {
8           "constraints": {
9             "clearText": false,
10            "required": "true",
11            "tabIndex": 2
12          },
13          "description": "The KEY for this API",
14          "displayName": "KEY",
15          "tooltip": "Provide your KEY"
16        }
17      }
18    },
19    "iconBrandColor": "#007EE6",
```

```
20     "policyTemplateInstances": [  
21     {  
22         "title": "MyPolicy",  
23         "templateId": "setqueryparameter",  
24         "parameters": {  
25             "x-ms-apimTemplateParameter.name": "queryParameterName",  
26             "x-ms-apimTemplateParameter.value": "queryParameterValue",  
27             "x-ms-apimTemplateParameter.existsAction": "override"  
28         }  
29     }  
30 ]  
31 }  
32 }
```

More information on each of the properties are given below:

- **properties**: The container for the information.
- **connectionParameters**: Defines the connection parameter for the service.
- **iconBrandColor**: The icon brand color in HTML hex code for the custom connector. Independent Publisher connectors must set the color to "#da3b01".
- **capabilities**: Describes the capabilities for the connector, e.g. cloud only, on-prem gateway etc.
- **policyTemplateInstances**: An optional list of policy template instances and values used in the custom connector.

For further details, see the apiProperties.json JSON schema.

README.md

README.md file for your connector includes a description for your connector, any prerequisite a developer or contributor may need to build your connector. It includes instructions on how to use your connector and api, how to get credentials, supported operations, known issues and limitations, etc. This file is meant to be a standalone guide for deploying and using your connector by other users and developers. A good example is the Azure Key Vault custom connector. A readme.md template for Certified Connectors and Independent Publisher Connectors is also included for your reference. If you are submitting an Independent Publisher connector that requires OAuth, please make sure to explain how to create the OAuth app. The Microsoft Certification Team will use those instructions to create the app, so please make sure they are detailed and accurate.

Creating a Fork

To contribute to this open source repository, start by creating a fork on this repository. To do so, select the “fork” button in the upper right corner, and create your own copy of the repository. Next, sync your fork with the remote repository and clone your forked repository to your local machine.

```
git clone https://github.com/YOUR-USERNAME/PowerPlatformConnectors.git
```

Check your remote URL.

```
git remote -v
```

```
1 > origin https://github.com/YOUR_USERNAME/PowerPlatformConnectors.git
    (fetch)
2 > origin https://github.com/YOUR_USERNAME/PowerPlatformConnectors.git
    (push)
```

Add an upstream repository for your clone.

```
git remote add upstream https://github.com/microsoft/PowerPlatformConnectors.git
```

Verify the upstream links.

```
git remote -v
```

```
1 > origin https://github.com/YOUR_USERNAME/PowerPlatformConnectors.git
    (fetch)
2 > origin https://github.com/YOUR_USERNAME/PowerPlatformConnectors.git
    (push)
3 > upstream https://github.com/microsoft/PowerPlatformConnectors.git (
    fetch)
4 > upstream https://github.com/microsoft/PowerPlatformConnectors.git (
    push)
```

To keep your fork up to date with this repository’s updates, run these commands:

```
git fetch upstream
```

```
git checkout master
```

```
git merge upstream/master
```

You are now ready to develop your connector in your own branch.

Submitting to the Open Source Repository

Contributions to the open source repository are made through pull requests. Prior to submitting a pull request, ensure that 1) you have thoroughly tested the connector 2) you have provided response schemas unless the responses are dynamic, and 3) that your pull request does not contain any sensitive or specific information, for example client ids or client secrets. Any sensitive values can be replaced with fake or dummy values for the purposes of submission as long as it is clearly indicated. Also, ensure that the `readme.md` of the connector is updated with the latest information, or created for new connector submissions. An example of a clear, structured, `readme.md` can be found for the Azure Key Vault connector. A `readme.md` template for Certified Connectors and Independent Publisher Connectors is also included for your reference. Put the `readme.md` in the same directory as the other connector files. Add tags indicating which connector type you are submitting. Connector type name should match the folder name you are submitting to: `custom-connector`, `certified-connector`, or `independent-publisher-connector`.

Certified Connectors For new connectors which will be submitted for certification, create a directory under the `certified-connectors` directory, place the connector files in the sub-folder, and submit a pull request to the `dev` branch. Ensure that a clear, structured, `readme.md` is included.

Add a tag by selecting the labels option to “certified-connector”

Updates to certified connectors must first be made through a pull request to the `dev` branch for review by the connector owner.

Once a pull request has been merged to the `dev` branch, the connector owner can submit the connector for certification through the Connector certification tab in ISV Studio. Once certified, the Microsoft Certification team will handle merging the updates from `dev` to `master`.

Updates to an existing custom connector can be made through a simple pull request to the `dev` branch to update the custom connector files.

Independent Publisher Connectors Follow the same instructions as above on submitting for certification, create a directory under the “independent-publisher-connectors” directory and place the connector files in the sub-folder. The `"iconBrandColor"` in the API properties file must be set to `"#da3b01"`. Set your pull request title to “Connector Name (Independent Publisher).” Paste in screenshots from the Test operations section and 3 unique operations (actions/triggers) working within a Flow. This can be in one flow or part of multiple flows. For each one of those flows, I have pasted in screenshots of the Flow succeeding. Add a tag by selecting the labels option to “independent-publisher-connector.” If the connector uses OAuth, I have provided detailed steps on how to create an app in the `readme.md`.

Custom Connectors Follow the same instructions on submitting for certification, create a directory under the custom-connectors directory and place the connector files in the sub-folder. Add a tag by selecting the labels option to “custom-connector”.

Tooling and Validation

CLA When a pull request is submitted, a CLA-bot will automatically determine whether you need to provide a CLA and annotate the PR appropriately. Simply follow the instructions provided by the bot to ensure your pull request can be properly reviewed. You will only need to do this once across all repos using our CLA.

This project has adopted the Microsoft Open Source Code of Conduct. For more information see the Code of Conduct FAQ or contact opencode@microsoft.com with any additional questions or comments.

Swagger Validation A submitted pull request will also be validated against our Swagger Validator tool, which checks the connector files to ensure it is a proper Swagger file and adheres to our connector requirements and guidelines. Any errors or warnings will be added to the PR for both the submitter and the reviewer to understand. We do not accept pull requests with outstanding unresolved Swagger Validator issues.

Breaking Change Detector Another validation which runs on a submitted pull request is the breaking changes validator. This is to catch any inadvertent, non-backwards-compatible (i.e. breaking) changes which may break a current user experience, for example, deleting a published operation. The Breaking Change Detector compares the previous version of the Swagger with the new submission and raises awareness of any breaking change. The submitter and reviewer must both acknowledge any breaking changes submitted and ensure that no end users are inadvertently negatively affected.

Legal Notices

Microsoft and any contributors grant you a license to the Microsoft documentation and other content in this repository under the Creative Commons Attribution 4.0 International Public License, see the LICENSE file, and grant you a license to any code in the repository under the MIT License, see the LICENSE-CODE file.

Microsoft, Windows, Microsoft Azure and/or other Microsoft products and services referenced in the documentation may be either trademarks or registered trademarks of Microsoft in the United

States and/or other countries. The licenses for this project do not grant you rights to use any Microsoft names, logos, or trademarks. Microsoft's general trademark guidelines can be found at <http://go.microsoft.com/fwlink/?LinkID=254653>.

Privacy information can be found at <https://privacy.microsoft.com/en-us/>

Microsoft and any contributors reserve all others rights, whether under their respective copyrights, patents, or trademarks, whether by implication, estoppel or otherwise.