Custom Connectors

Benedikt Bergmann

Power Platform Community Frankfurt

Benedikt Bergmann

- CEO @ CRMK Deutschland

- .NET Developer
- Angular/React
- Dynamics 365
- Power Platform

- benedikt@benediktbergmann.eu
- https://www.linkedin.com/in/benedikt-bergmann
- https://twitter.com/BergmannBene
- http://benediktbergmann.eu

https://bio.link/benediktbergmann

MVP Microsoft® Most Valuable Professional

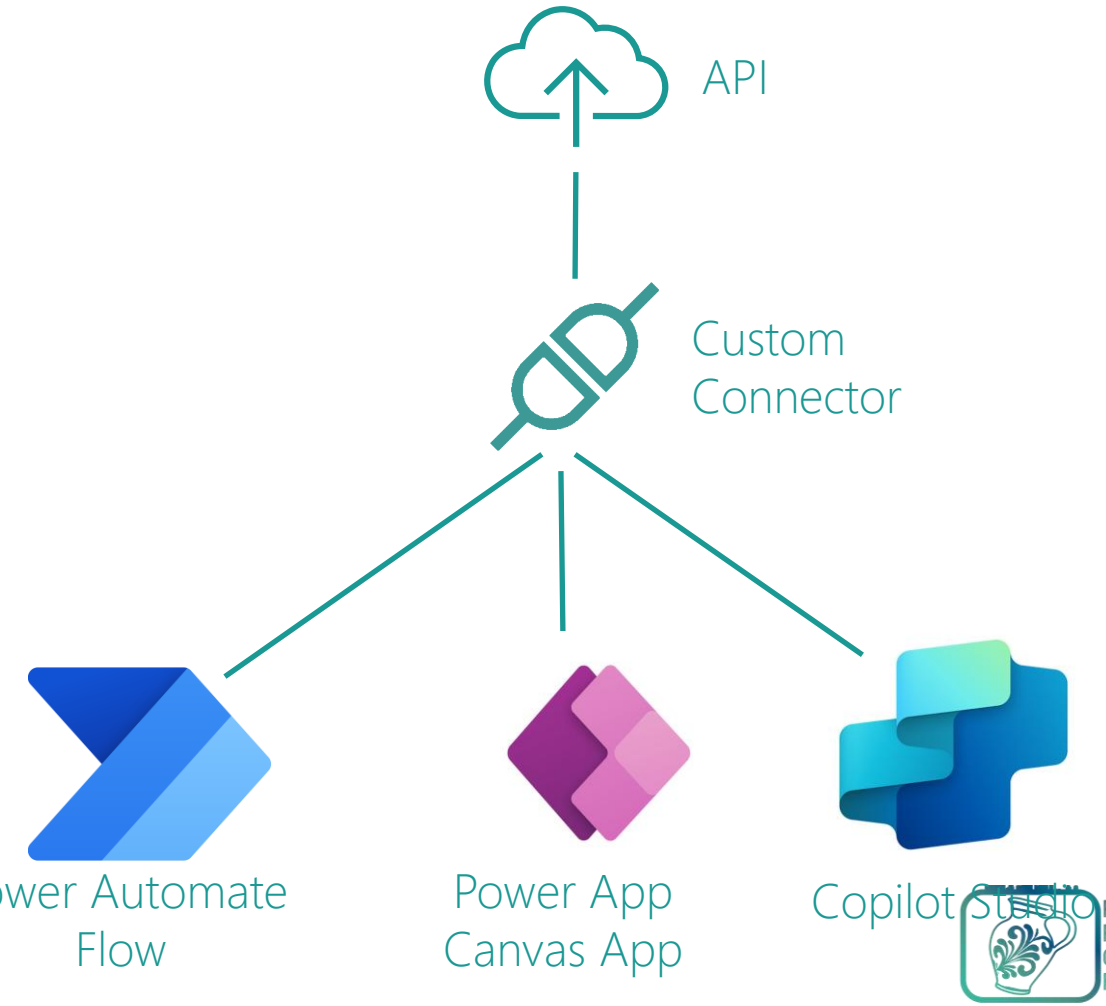
Agenda

- Theory
  - What & Why
  - Which
  - Components
  - Types
  - How
- Demo
- Limitations
- Resources

What & Why

- Wrapper around API
- Triggers and Actions for API

- Minimize complexity
- Increase productivity
- Better logging & Analytics



Power Platform Community Frankfurt

Which

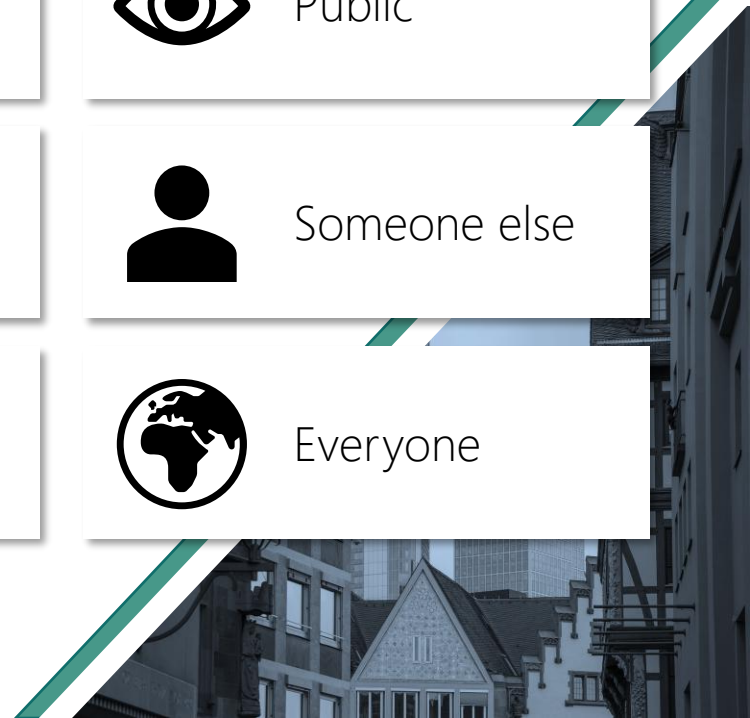
- Any Restful API
- Public APIs from third party services
  - Spotify
  - Yelp
  - …
- Public APIs you manage
  - Azure Functions
  - Azure Web Apps
  - Azure API Apps
  - And more
- Private APIs via On-Prem Data Gateway

Use-cases

- Legacy API
- OnPrem API
- Internal API
- No Connector (yet)

Custom connector types

Connector Type	Custom Connector	Certified Connector	Independent Publisher Connector
API Availability	Public or private	Public	Public
Service Owner	Connector developer or someone else	Connector developer	Someone else
Audience	Internal	Everyone	Everyone



Products

- Power Automate
- Power Apps
- Logic Apps
- Copilot Studio

Components

Info

Name, Logo, Description

Security

Authentication

Actions

Endpoints and their visibility, headers, definition, etc.

Triggers

Timed or event-driven webhook actions

References

Reusable input or output objects

Policies

Templates for modifying runtime behaviour

Power Platform Community Frankfurt

Configure API security

How a custom connector authenticates to the API

- No authentication
- Basic authentication – Username and password
- OAuth 2.0 – Generic OAuth 2.0 or prebuilt configurations for Azure AD, GitHub, Slack, etc.
- API key – For example an Azure Function

Ways to create a new custom connector

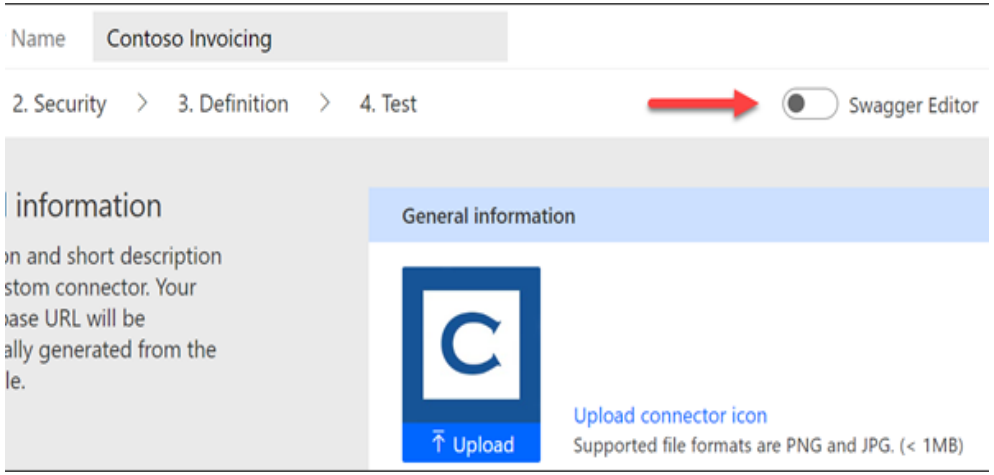
- Create from blank
- Create from Azure Service (Preview)
- Import an OpenAPI file
- Import an OpenAPI from URL
- Import a Postman collection
- Import from GitHub

OpenAPI extensions

Identify extensions by looking for

x-ms-<name> in the OpenAPI definition

Add extensions using the Swagger editor



Example extensions

- x-ms-capabilities
- x-ms-url-encoding
- x-ms-dynamic-values
- x-ms-dynamic-list
- x-ms-dynamic-schema
- x-ms-visibility

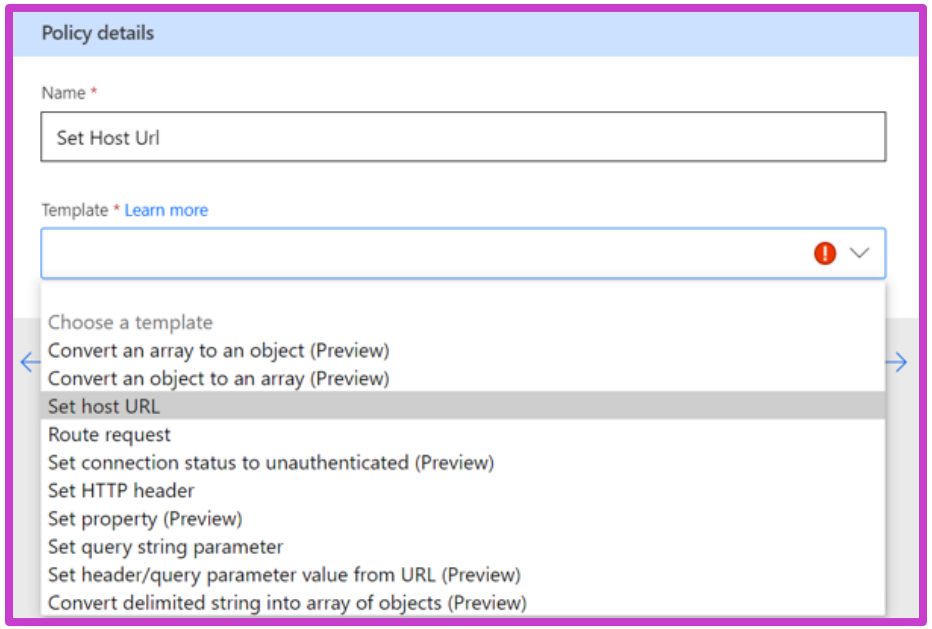
Power Platform Community Frankfurt

Policy templates

- Policies allow you to modify the behavior of a custom connector at runtime.
- You can use policies to perform data conversion, route requests, set parameter values, and more.
- You can configure policies directly in the custom connector API properties file before import, or you can do it from the maker portal in the custom connector designer by applying policy templates.

Use cases for policy templates

- Route request
  - Routes requests to a specified endpoint on the same service
- Set Host URL
  - Route calls to different endpoints
  - Use expressions to access runtime values
- Set HTTP header
  - Override or add information to a request or response
  - Enable CORS for API Management connectors
- Set Query String Parameter
  - Adds or updates value of request query string parameter



Power Platform Community Frankfurt

DON’T HARDCODE – Use policies – Variables on different levels

Expression	Description
@headers('headerName')	Allows access to request headers. headerName is the name of the header for which value will be returned.
@queryParameters('queryParameterName')	Allows access to query parameters in request. queryParameterName is the name of the query parameter for which value will be returned.
@connectionParameters('connectionParameterName')	Allows access to request's connection parameters. connectionParameterName is the name of the property defined in connection parameters for which value will be returned.

Environment variables can use the following syntax in custom connector policies:

```
@environmentVariables("environmentVariableName")
```

Example

```
@environmentVariables("cr49f_SharePointSiteURL")
```

- **Query Parameter** -> setting per action ("per flow")
- **Connection Parameter** -> setting per connection ("per user")
- **Environment Variable** -> setting per environment ("global")

Implement custom code

To implement custom code, you'll need to create a class named Script, which must inherit from the abstract base class **ScriptBase**.

```
public class Script : ScriptBase
{
    public override
Task<HttpResponseMessage>
ExecuteAsync()
    {
        // Your code here
    }
}
```

Power Platform Community Frankfurt

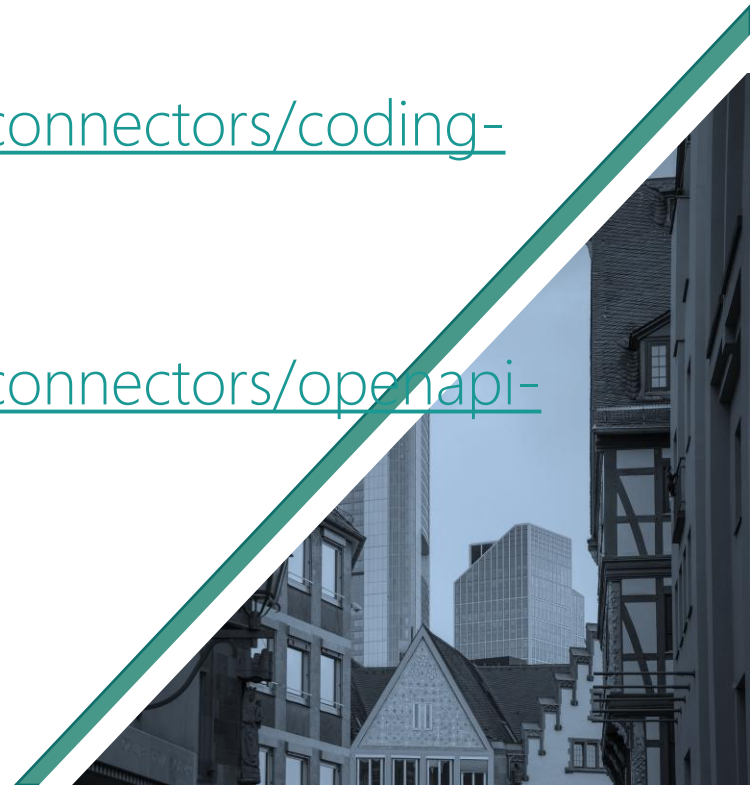
DEMO

Limitations

- Max size - 1 MB
- Amount of custom connectors
- Deployment
- Other authentications

Resources

- Power Platform Connectors CLI (Paconn)
  - https://docs.microsoft.com/en-us/connectors/custom-connectors/paconn-cli
- Coding standard
  - https://docs.microsoft.com/en-us/connectors/custom-connectors/coding-standards
- OpenAPI Extensions
  - https://docs.microsoft.com/en-us/connectors/custom-connectors/openapi-extensions



Danke

https://bio.link/benediktbergmann