

# Integrating PowerApps with External Systems



# Agenda

- Calling External Services using HTTP Actions
- Executing Child Flows from a Parent Flow
- Creating and Testing Custom Connector
- Configuring a Custom Connector to use OAuth



# HTTP Request and Response

- HTTP Request
  - URL
  - Verb (e.g. GET, POST, PUT, PATCH, etc)
  - Headers
  - Body
- HTTP Response
  - Status code
  - Headers
  - Body



# Working With REST-based Web Service

- Calling REST-based web service involve creating structured URLs
  - `http://subliminalsystems.com/api/Customers/?$select=LastName,FirstName,CustomerId`
- REST-based URL structured into several parts

<code>http://subliminalsystems.com/api/</code>	<code>Customers/</code>	?	<code>\$select=LastName,FirstName,CustomerId</code>
Web Service Base URL	Resource		Query String Parameters



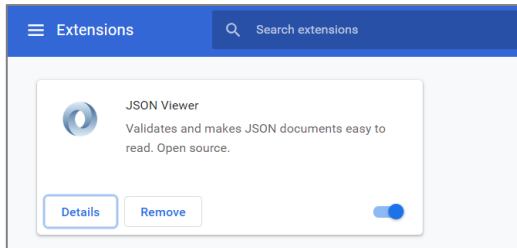
# OData Query String Parameters

- Many web services support OData
  - OData is a REST-based specification for CRUD operations
  - OData defines standard query string parameters
- Commonly-used OData query string parameters
  - **\$select** – allows you to select which properties are returned
  - **\$filter** – allows you to filter rows returned in result
  - **\$orderby** – allows you to sort rows in result
  - **\$top** – allows you to specify how many rows should be returned
  - **\$skip** – allows you to skip over rows that are not needed
  - **\$expand** – allows you to retrieve more data in one round trip

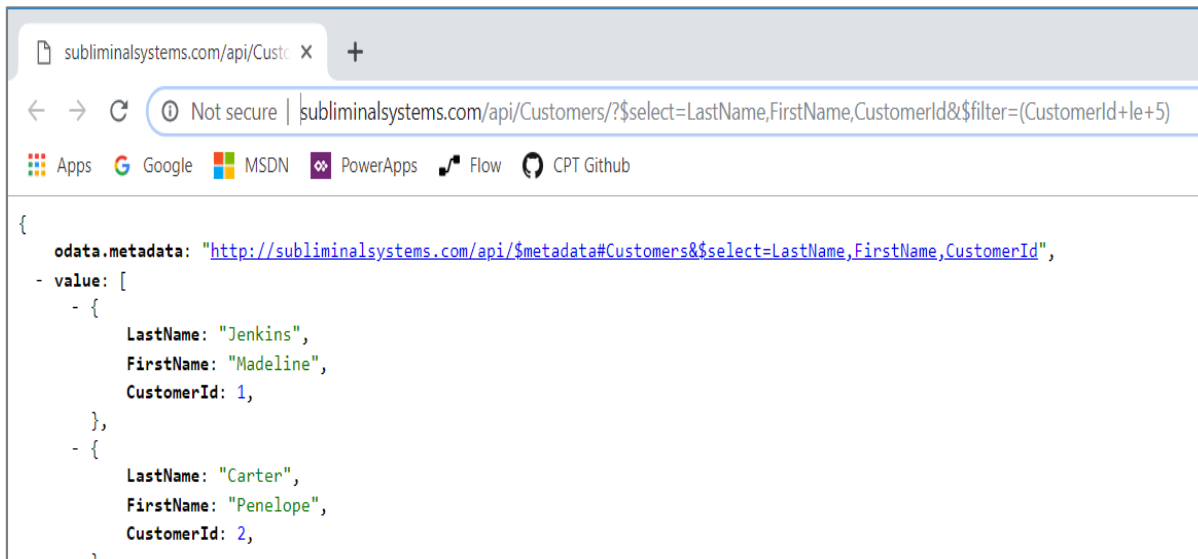


# Using the JSON Viewer Chrome Extension

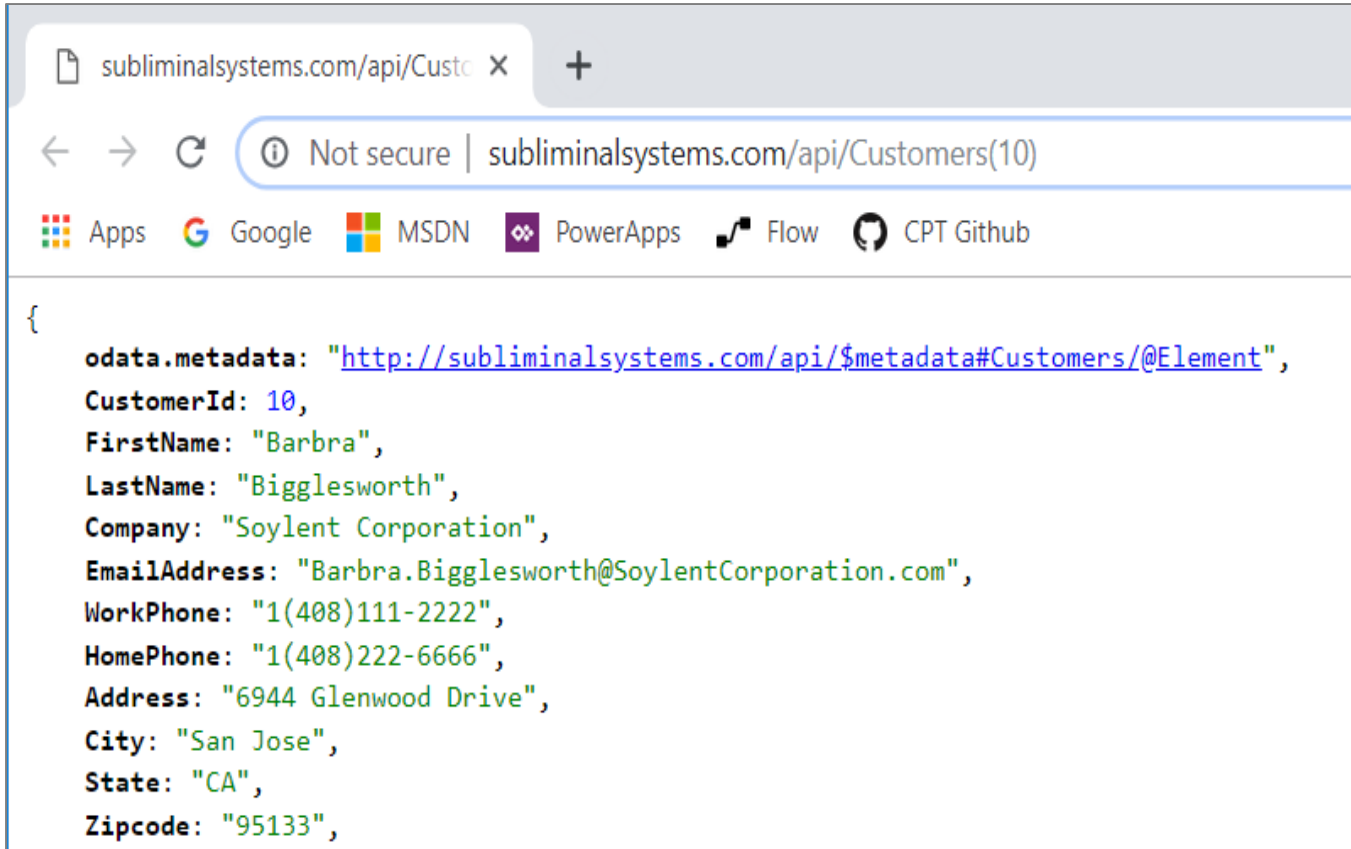
- Install the JSON Viewer extension



- Now you can inspect JSON in the browser



# Retrieving a Single OData Instance

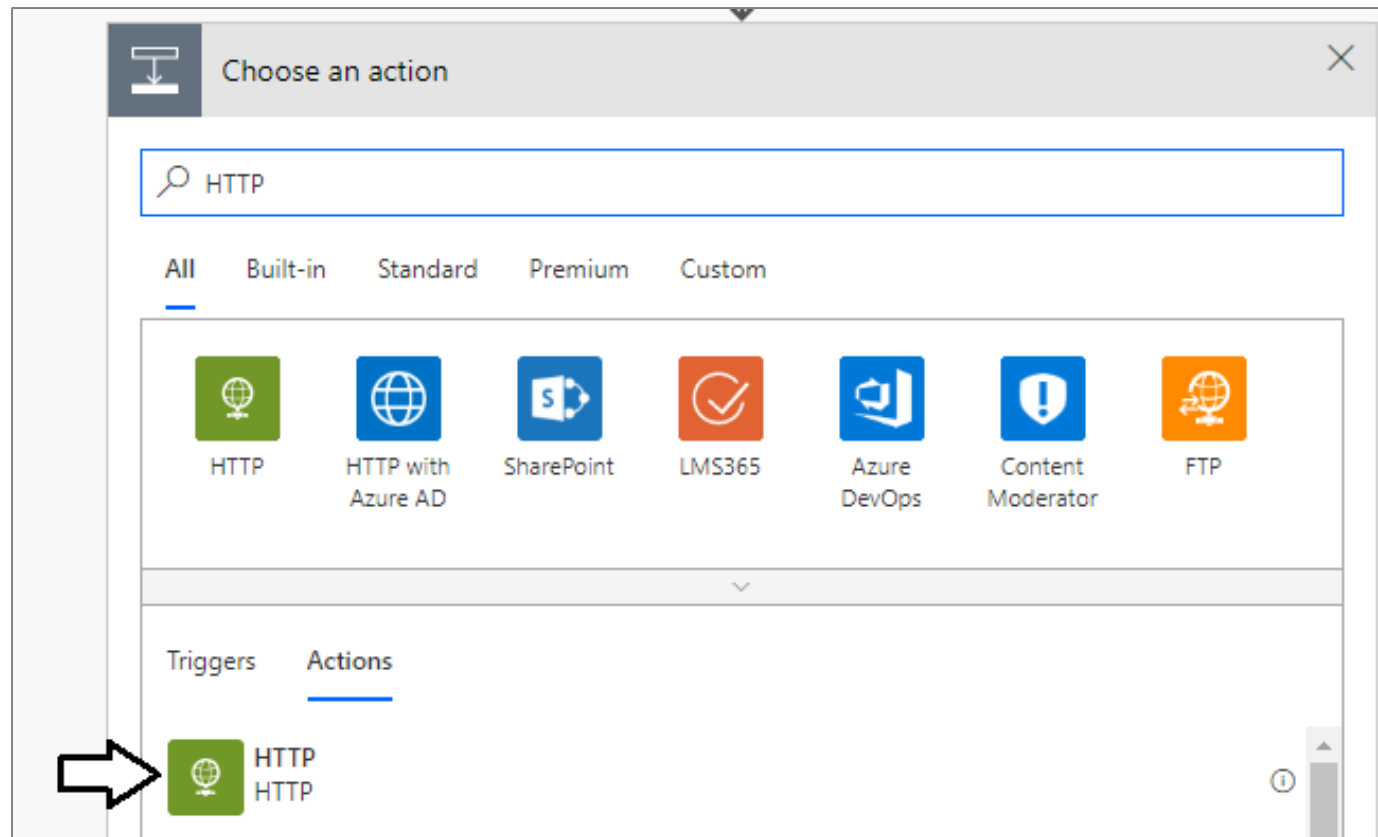


A screenshot of a web browser window. The address bar shows the URL `subliminalsystems.com/api/Custo` with a plus sign to the right. Below the address bar, there's a navigation bar with icons for Apps, Google, MSDN, PowerApps, Flow, and CPT Github. The main content area displays a JSON object representing an OData instance for a customer with ID 10.

```
{
  odata.metadata: "http://subliminalsystems.com/api/$metadata#Customers/@Element",
  CustomerId: 10,
  FirstName: "Barbra",
  LastName: "Bigglesworth",
  Company: "Soylent Corporation",
  EmailAddress: "Barbra.Bigglesworth@SoylentCorporation.com",
  WorkPhone: "1(408)111-2222",
  HomePhone: "1(408)222-6666",
  Address: "6944 Glenwood Drive",
  City: "San Jose",
  State: "CA",
  Zipcode: "95133",
}
```

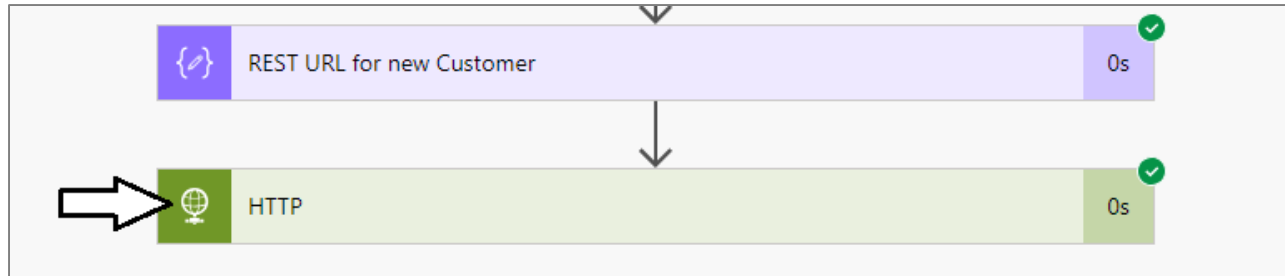


# Using the HTTP Action





# Inspecting the body of the HTTP Response

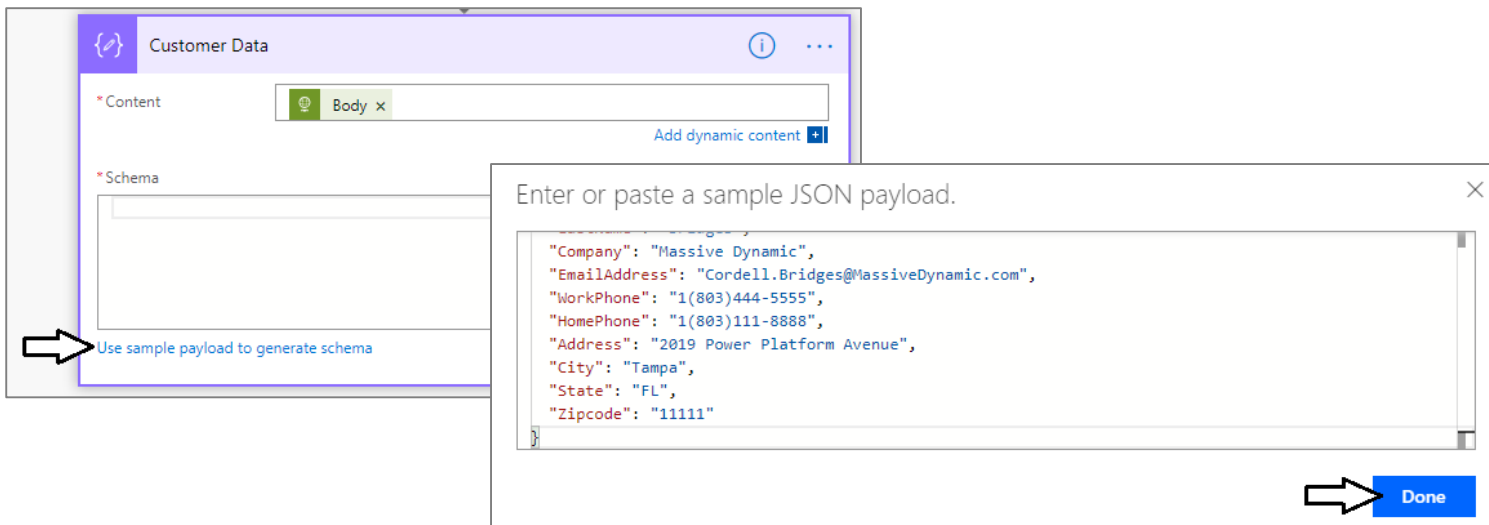
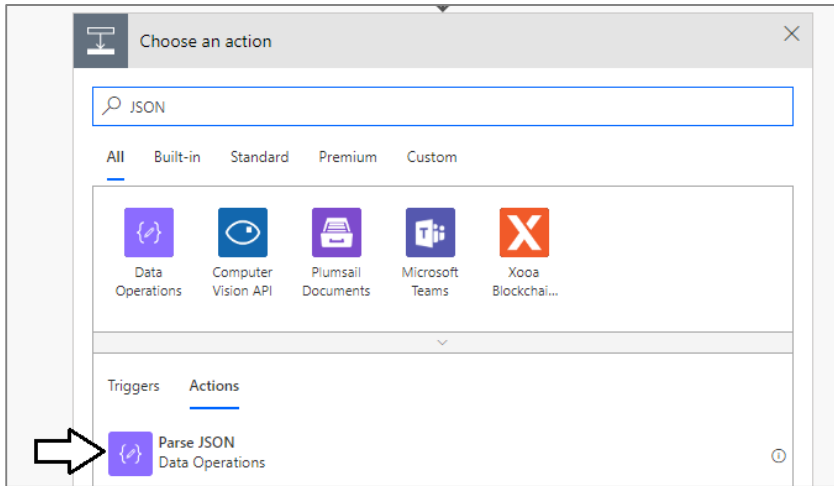


A screenshot of an HTTP response body. A white arrow points from the left towards the word "Body". The response is displayed in a text area with a scrollbar. The JSON data is as follows:

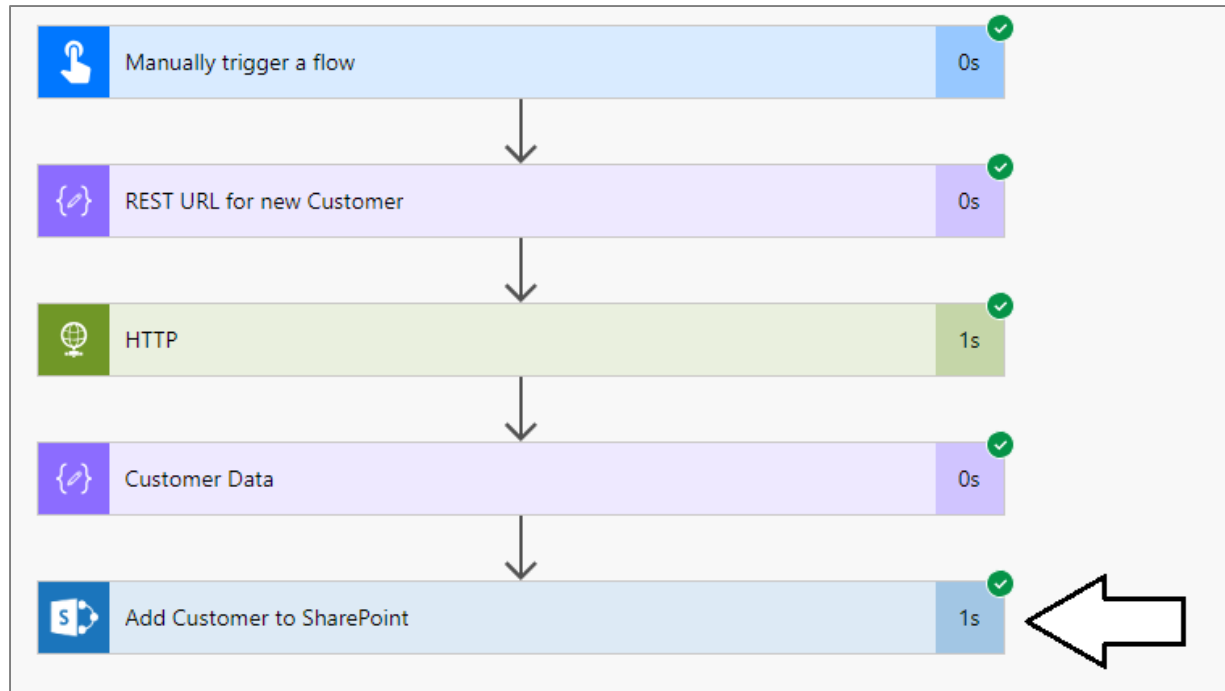
```
{  
  "odata.metadata": "http://subliminalsystems.com/api/\$metadata#  
  "CustomerId": 47,  
  "FirstName": "Cordell",  
  "LastName": "Bridges",  
  "Company": "Massive Dynamic",  
  "EmailAddress": "Cordell.Bridges@MassiveDynamic.com",  
  "WorkPhone": "1(800)444-5555"
```



# Parsing JSON



# Adding a New Item to a SharePoint List





**DEMO**

## **Calling an External Web Service with the HTTP Action**

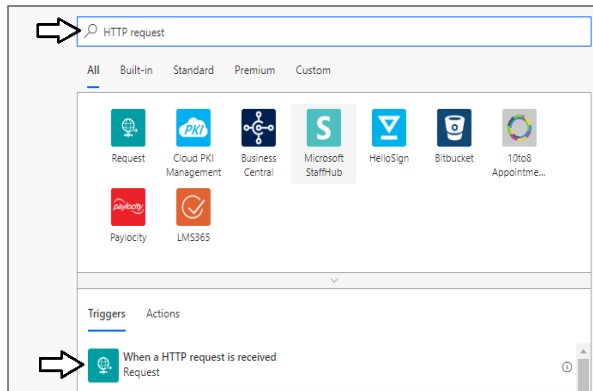
# Agenda

- ✓ Calling External Services using HTTP Actions
- Executing Child Flows from a Parent Flow
- Creating and Testing Custom Connector
- Configuring a Custom Connector to use OAuth

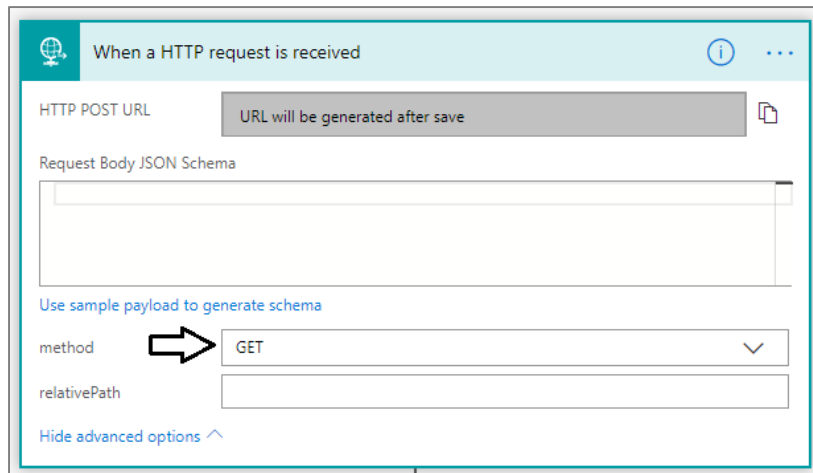


# Creating the Child Flow

- Child flow creating with HTTP trigger



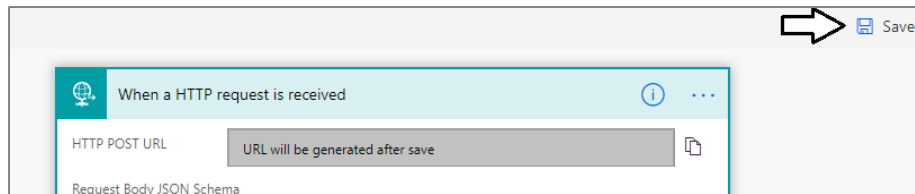
- Select an HTTP operation (e.g. GET, POST, PUT, DELET, PATCH, etc.)



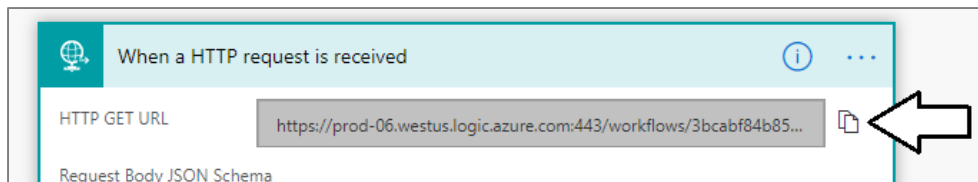


# Discovering the URL for an HTTP Trigger

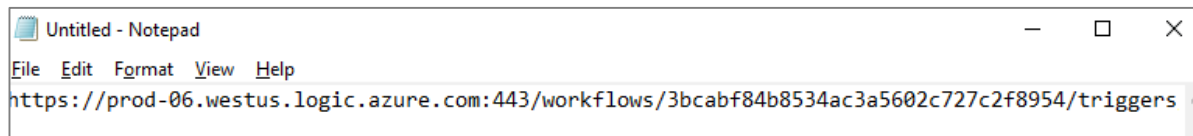
- HTTP POST URL is not generated until you first save the flow



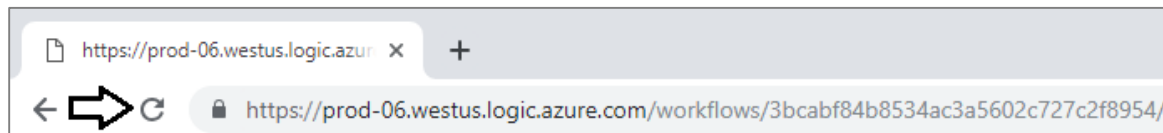
- After saving the flow you can copy the HTTP POST URL



- Copy and paste the URL into Notepad

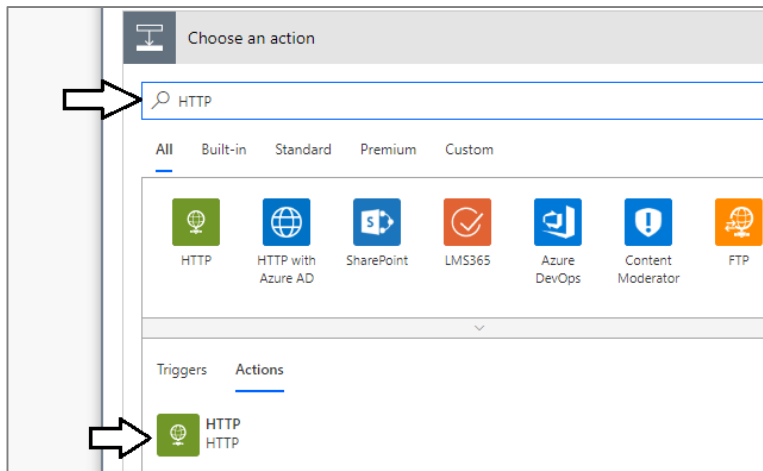


- Try executing the flow by adding the URL into address bar in browser

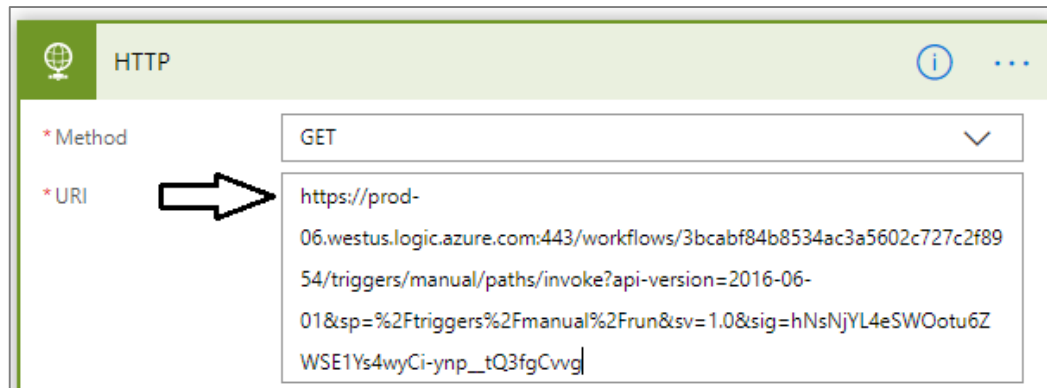


# Parent Flow

- Parent flow uses HTTP action to execute child flow



- Set Method parameter to Get and paste in URL for child flow







**DEMO**

**Calling a Child Flow from a Parent Flow**

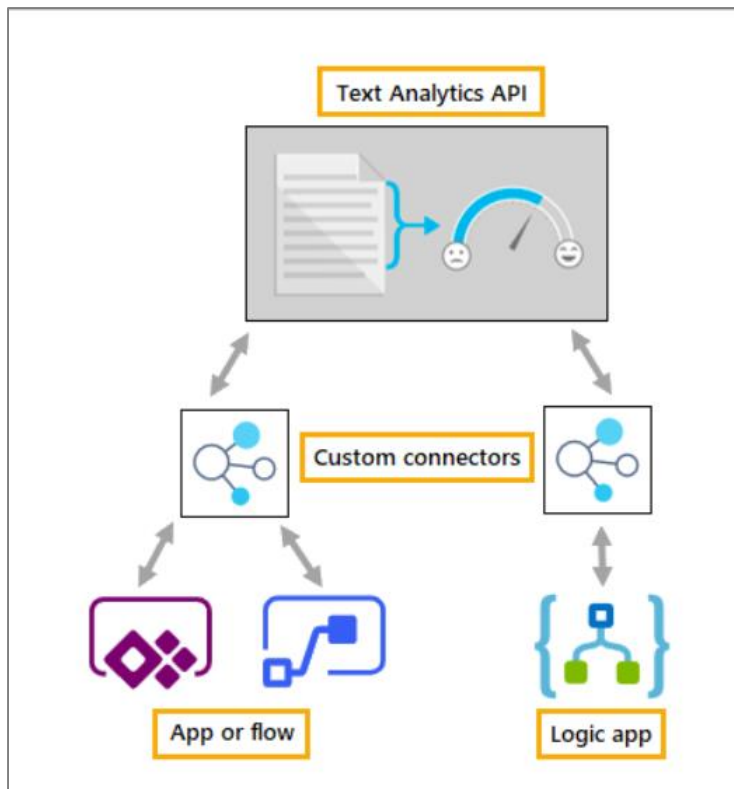
# Agenda

- ✓ Calling External Services using HTTP Actions
- ✓ Executing Child Flows from a Parent Flow
- Creating and Testing Custom Connector
- Configuring a Custom Connector to use OAuth

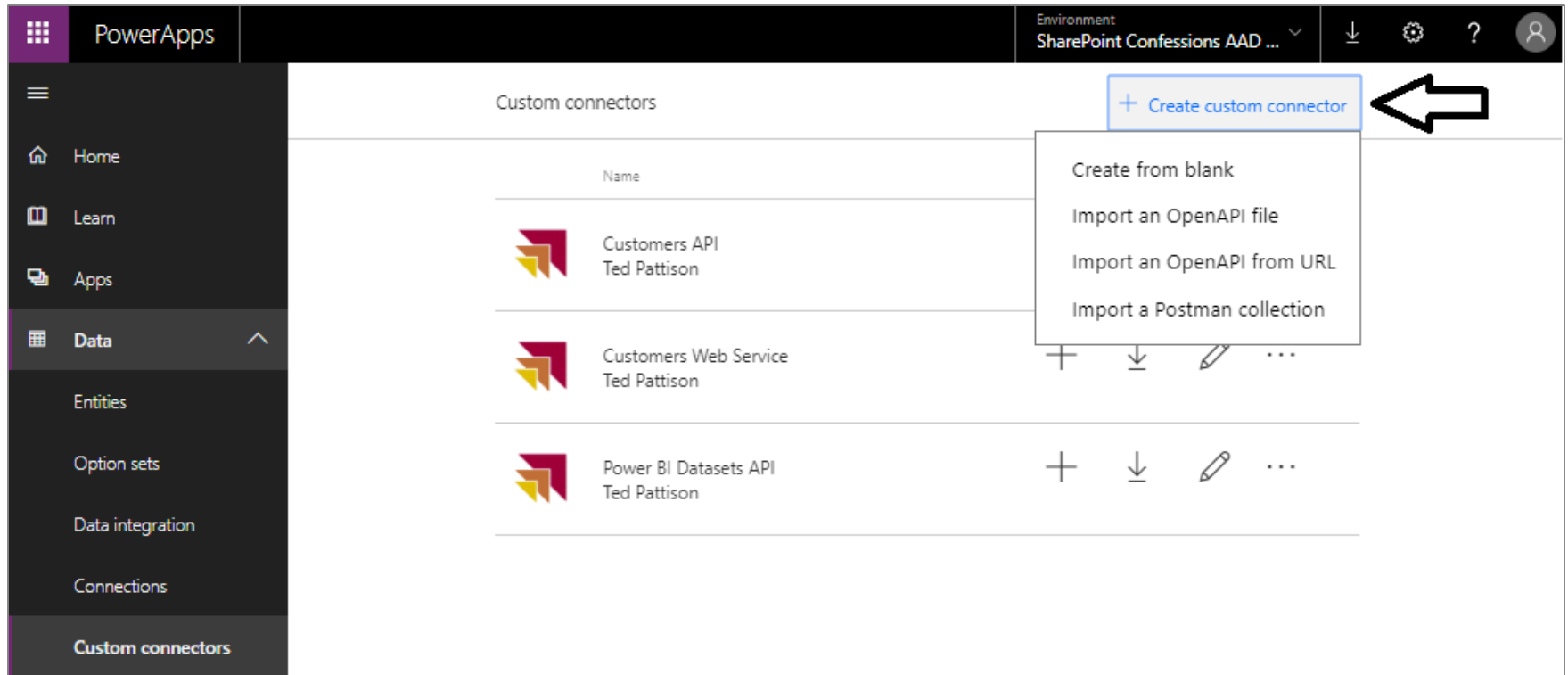


# Standard Connectors vs Custom Connectors

- PowerApps Supports two types of connectors
  - Standard connectors supplied out-of-box and vetted by Microsoft
  - Custom connectors created by organizations for their own use



# Creating a New Custom Connector



The screenshot shows the PowerApps interface. The top navigation bar includes the PowerApps logo, the environment name "SharePoint Confessions AAD ...", and icons for download, settings, help, and user profile. The left sidebar contains a menu with "Home", "Learn", "Apps", "Data", "Entities", "Option sets", "Data integration", "Connections", and "Custom connectors". The "Custom connectors" section is active. The main area is titled "Custom connectors" and features a table with three existing connectors: "Customers API", "Customers Web Service", and "Power BI Datasets API", all by "Ted Pattison". A "+ Create custom connector" button is located at the top right of the table, with a large black arrow pointing to it. A dropdown menu is open from this button, showing four options: "Create from blank", "Import an OpenAPI file", "Import an OpenAPI from URL", and "Import a Postman collection".

PowerApps

Environment  
SharePoint Confessions AAD ...

Custom connectors

+ Create custom connector

Create from blank  
Import an OpenAPI file  
Import an OpenAPI from URL  
Import a Postman collection

Name	
Customers API Ted Pattison	
Customers Web Service Ted Pattison	
Power BI Datasets API Ted Pattison	



# Defining Requests

Request + Import from sample

\* Verb

The verb describes the operations available on a single path.

GET

\* URL

This is the request URL.

http://subliminalsystems.com/api/Customers/

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.

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.

Request + Import from sample

\* Verb

The verb describes the operations available on a single path.

GET

\* URL

This is the request URL.

http://subliminalsystems.com/api/Customers({CustomerId})

Path

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

\* CustomerId ...

Query

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

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.





# Defining the Response

Connector Name

Customers Web Service

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

✓ Update connector ✕ Close

▼ Actions (2)

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

1 GetCustomers ...

2 **GetCustomer** ...

+ New action

▼ References (0)

References are reusable parameters used by both actions and triggers.

← 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.

Address ... BirthDate ... City ... Company ... CustomerId ...

EmailAddress ... FirstName ... FirstPurchaseDate ... Gender ...

HomePhone ... LastName ... LastPurchaseDate ... State ... WorkPhone ...

Zipcode ... odata.metadata ...

Validation

This helps you identify potential issues with this response.

Validation

✓

Validation succeeded.

← Security Test →



# OpenAPI Specification and Swagger

- OpenAPI specification (OAS)
  - Community-driven open specification
  - Defines standard interface description for REST APIs
  - allows humans and computers to discover web service API
  - OpenAPI Specification removes guesswork in calling a service.
- API description defined using JSON

```
{ CustomerFactoryAPI.swagger.json x
1  {
2    "swagger": "2.0",
3    "info": {
4      "title": "CustomerFactoryApi",
5      "description": "Sample custom connector providing sample customer data",
6      "version": "1.0"
7    },
8    "host": "cptlabs.azurewebsites.net",
9    "basePath": "/api/",
10   "schemes": [ "https" ],
11   "consumes": [],
12   "produces": [],
13   "paths": {
14     "/NextCustomer": {
15       "get": {
16         "responses": {
17           "default": {
18             "description": "default",
19             "schema": {
20               "type": "object",
21               "properties": { "firstName": { "type": "string", "description": "firstName" },
22                             "lastName": { "type": "string", "description": "lastName" },
23                             "company": { "type": "string", "description": "company" },
24                             "emailAddress": { "type": "string", "description": "emailAddress" },
```





**DEMO**

## **Creating a Simple Custom Connector**

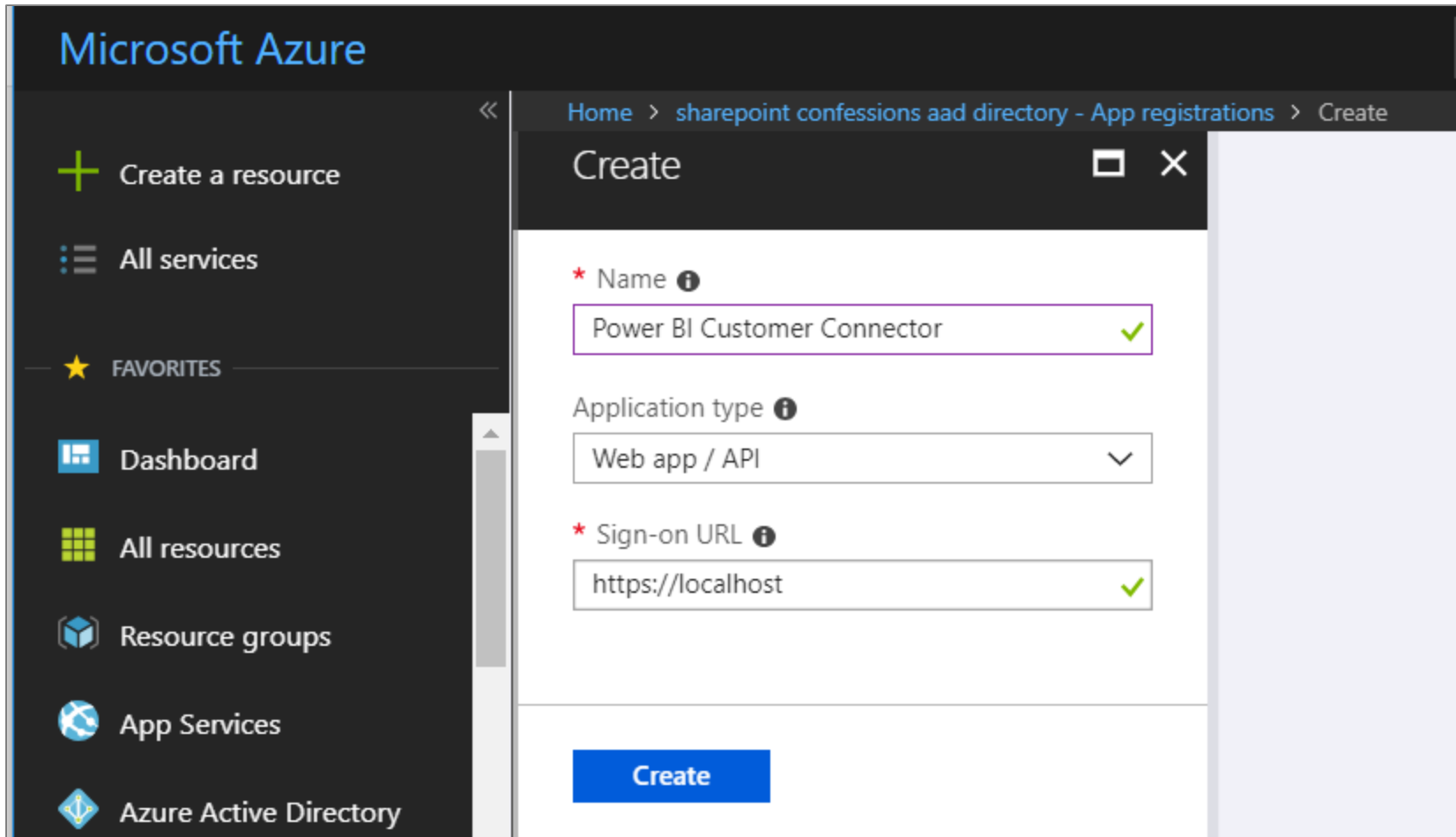


# Agenda

- ✓ Calling External Services using HTTP Actions
- ✓ Executing Child Flows from a Parent Flow
- ✓ Creating and Testing Custom Connector
- Configuring a Custom Connector to use OAuth



# Creating an Azure AD Application



The screenshot shows the Microsoft Azure portal interface. On the left is a dark sidebar with navigation options: 'Create a resource', 'All services', 'FAVORITES', 'Dashboard', 'All resources', 'Resource groups', 'App Services', and 'Azure Active Directory'. The main content area has a breadcrumb trail: 'Home > sharepoint confessions aad directory - App registrations > Create'. Below the breadcrumb is a 'Create' dialog box with a title bar containing a maximize icon and a close 'X' icon. The dialog contains three required fields, each marked with a red asterisk and an information icon: 'Name' with the value 'Power BI Customer Connector' and a green checkmark; 'Application type' with a dropdown menu set to 'Web app / API'; and 'Sign-on URL' with the value 'https://localhost' and a green checkmark. At the bottom of the dialog is a blue 'Create' button.

Microsoft Azure

Home > sharepoint confessions aad directory - App registrations > Create

Create

\* Name ⓘ  
Power BI Customer Connector ✓

Application type ⓘ  
Web app / API ▼

\* Sign-on URL ⓘ  
https://localhost ✓

Create



# Application ID (aka Client ID)

Home > sharepoint confessions aad directory - App registrations > Power BI Customer Connector

## Power BI Customer Connector

Registered app

Settings Manifest Delete

Display name	Application ID
Power BI Customer Connector	0d956d63-2b3a-4a6b-8b21-5b43db48cc77
Application type	Object ID
Web app / API	07aee325-93c9-44bb-a770-71773f948223
Home page	Managed application in local directory
<a href="https://localhost">https://localhost</a>	<a href="#">Power BI Customer Connector</a>

⌵

Click to copy



# Configuring Permissions

Home > sharepoint confessions aad directory - App registrations > Power BI Customer Connector > Settings

## Power BI Customer Connector

Registered app

⚙ Settings ✎ Manifest 🗑 Delete

Display name	Application ID
Power BI Customer Connector	0d956d63-2b3a-4a6b-8b21-5b43db48cc77
Application type	Object ID
Web app / API	07aee325-93c9-44bb-a770-71773f948223
Home page	Managed application in local directory
<a href="https://localhost">https://localhost</a>	<a href="#">Power BI Customer Connector</a>

⌵

### Settings


🔍 Filter settings

#### GENERAL

- 📄 Properties >
- 🔗 Reply URLs >
- 👤 Owners >

#### API ACCESS

- 🔑 Required permissions >
- 🔑 Keys >



# Select the Power BI Service for the API

BI Customer Connector > Settings > Required permissions > Add API access > Select an API

Add API access

1

Select an API

Power BI Service

>

2

Select permissions

>

Select an API

Search for other applications with Service Principal name

Windows Azure Active Directory

Office 365 Exchange Online

Microsoft Graph

Office 365 SharePoint Online

Skype for Business Online

Office 365 Yammer

Dynamics CRM Online

Power BI Service


Microsoft Rights Management Services

Microsoft Intune API

Windows Azure Service Management API

Done

Select



# Select the Required Permissions

Enable Access

☐ DELEGATED PERMISSIONS

↑↓

REQUIRES ADMIN

<input type="checkbox"/>	Read and write all dataflows	No
<input type="checkbox"/>	View all dataflows	No
<input type="checkbox"/>	Read and write all content in tenant	Yes
<input checked="" type="checkbox"/>	Read and Write all Reports	No
<input type="checkbox"/>	View users Groups	No
<input type="checkbox"/>	View all Groups	No
<input type="checkbox"/>	View all Reports (preview)	No
<input type="checkbox"/>	Create content (preview)	No
<input type="checkbox"/>	View content properties (preview)	No
<input checked="" type="checkbox"/>	Read and Write all Datasets	No
<input type="checkbox"/>	View all Datasets	No

Select



# Generating a Key Secret

onfessions aad directory - App registrations > Power BI Customer Connector > Settings > Keys

### Settings

Filter settings

GENERAL

- Properties >
- Reply URLs >
- Owners >

API ACCESS

- Required permissions >
- Keys >**

### Keys

Save Discard Upload Public Key

⚠ Copy the key value. You won't be able to retrieve after you leave this blade.

#### Passwords

DESCRIPTION	EXPIRES	VALUE
Key1	8/14/2019	ejVnaC0zTLRcc3ErbAhgchilb3CTC5SVyKgbvS9HLPk=

#### Public Keys



# Data Required to Create Custom Connector

CustomConnectorAzureAD.txt - Notepad

File Edit Format View Help

Application ID (aka Client ID)

0d956d63-2b3a-4a6b-8b21-5b43db48cc77

Secret Key

ejVnaC0zTLRcc3ErbAhgchi1b3CTC5SVyKgbvS9HLPk=






# Creating the New Custom Connector

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

✓ Update connector ✕ Close

### 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.



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

↑ Upload

Icon background color

Description

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

Scheme  
☒ HTTPS ☐ HTTP

\* Host

Base URL

Security →



# Configuring the Client Id and Client Secret

← Connector Name Power BI Datasets API

1. General > **2. Security** > 3. Definition > 4. Test ✓ Create connector ✗ Cancel


## Security

Choose the authentication type and fill in the required fields to set the security for your custom connector.  
[Learn more](#)

### Authentication type

Choose what authentication is implemented by your API \*

OAuth 2.0

 Edit

### OAuth 2.0

Identity Provider

Azure Active Directory

Client id \*

0d956d63-2b3a-4a6b-8b21-5b43db48cc77

Client secret \*

.....



# What About the Redirect URL?

The image shows two side-by-side screenshots of an OAuth 2.0 configuration interface. The left screenshot shows a form with a 'Redirect URL' field containing the text 'Save the custom connector to generate the redirect URL'. A red arrow points from this field to the right screenshot. The right screenshot shows the same form, but the 'Redirect URL' field now contains the URL 'https://msmanaged-na.consent.azure-apim.net/redirect'. Another red arrow points from the 'Redirect URL' field in the left screenshot to this field in the right screenshot. Both forms have the same other fields: Identity Provider (Azure Active Directory), Client id (0d956d63-2b3a-4a6b-8b21-5b43db48cc77), Client secret (masked), Login URL (https://login.windows.net), Tenant ID (common), Resource URL (https://analysis.windows.net/powerbi/api), and Scope (scope).

**OAuth 2.0**

Identity Provider  
Azure Active Directory

Client id \*  
0d956d63-2b3a-4a6b-8b21-5b43db48cc77

Client secret \*  
\*\*\*\*\*

Login URL  
https://login.windows.net

Tenant ID  
common

Resource URL \*  
https://analysis.windows.net/powerbi/api

Scope  
scope

Redirect URL  
Save the custom connector to generate the redirect URL

Edit

**OAuth 2.0**

Identity Provider  
Azure Active Directory

Client id \*  
0d956d63-2b3a-4a6b-8b21-5b43db48cc77

Client secret \*  
\*\*\*\*\*

Login URL  
https://login.windows.net

Tenant ID  
common

Resource URL \*  
https://analysis.windows.net/powerbi/api

Scope  
Scope

Redirect URL  
https://msmanaged-na.consent.azure-apim.net/redirect

Edit



# Adding the Redirect URL back in Azure AD






Scope

Scope

Redirect URL

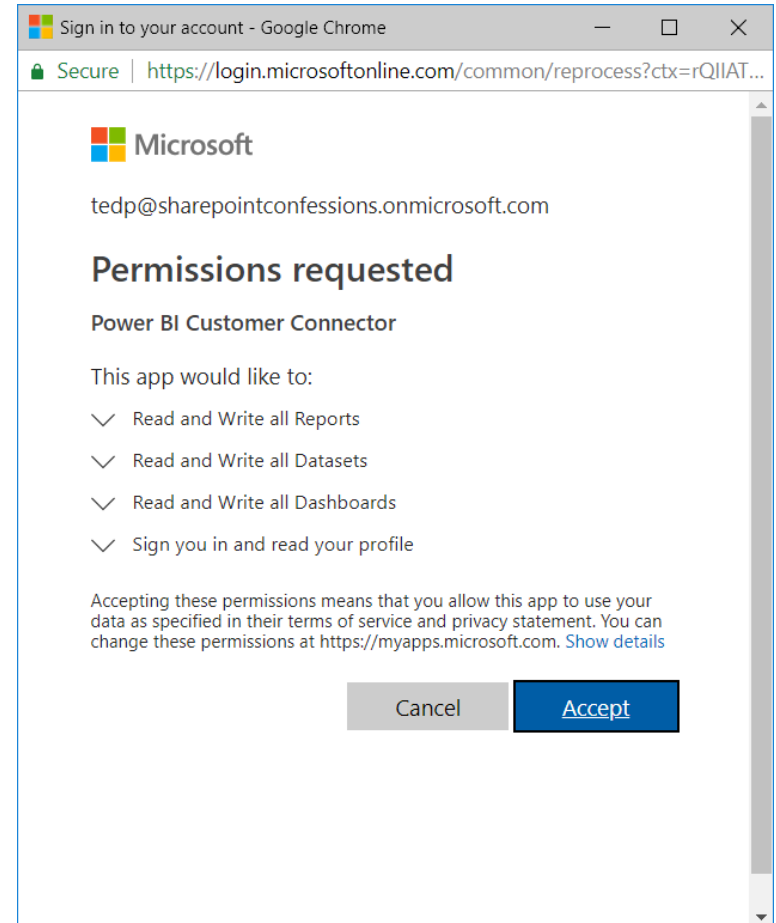
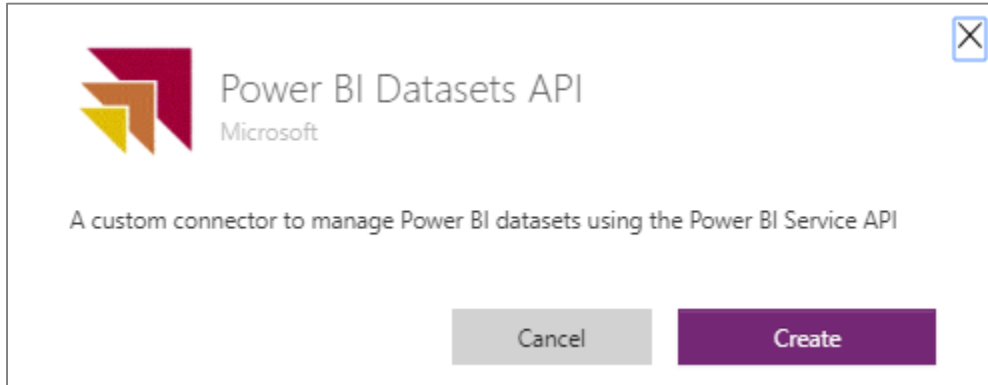
<https://msmanaged-na.consent.azure-apim.net/redirect>

Click to copy

Settings	×	Reply URLs
<input type="text" value="Filter settings"/>		 Save  Discard
<b>GENERAL</b>		<a href="https://msmanaged-na.consent.azure-apim.net/redirect">https://msmanaged-na.consent.azure-apim.net/redirect</a>
 Properties >		<input type="text"/>
 Reply URLs >		
 Owners >		



# Creating a New Connection



# Summary

- ✓ Calling External Services using HTTP Actions
- ✓ Executing Child Flows from a Parent Flow
- ✓ Creating and Testing Custom Connector
- ✓ Configuring a Custom Connector to use OAuth

