PowerApps Ox Build Tools

Configure WebHook Task

Getting Started Guide

Getting Started Overview

This guide will demonstrate how to use the Configure WebHook Task to automatically set WebHook parameters in the Azure DevOps Release Pipeline.

This guide assumes you have some development experience with PowerApps / Dynamics 365 and Azure Function Apps.

Prerequisites

Please download and install Dynamics 365 Tools via nuget. This will include the Plugin Registration Tool.

Please follow the PowerApps Build Tools lab:

https://github.com/microsoft/PowerApps-Samples/blob/master/azure/build-tools/PowerApps_Build%20Tools_Lab.zip

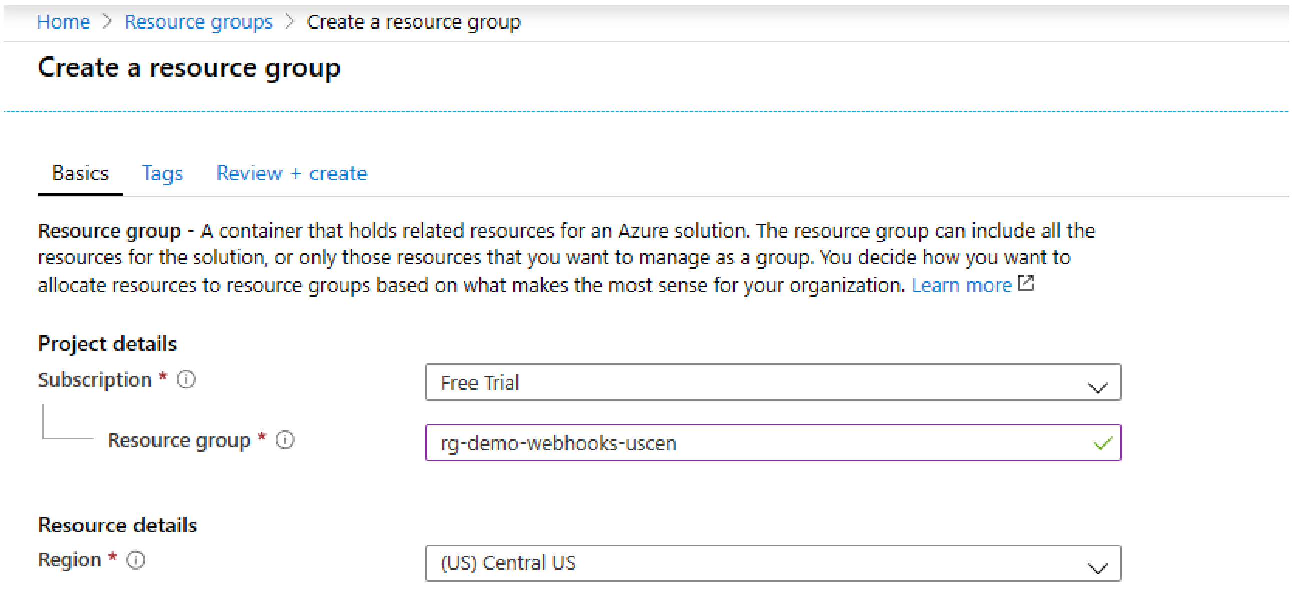
The lab completes the following tasks:
- creates 3 PowerApps CDS Environments: dev, build and prod
- creates simple PowerApp solution with 1 custom entity "Time Off Request" and a model driven app
- installs Microsoft PowerApps BuildTools into Azure DevOps
- creates Azure DevOps pipelines:
    - Create Export from Dev
    - Build Managed Solution
    - PROD Release

Instructions

Create Azure Function App

1. Log on to Azure Portal (https://portal.azure.com).

2. Create a "Resource Group". This allows you to specify the region.


Home > Resource groups > Create a resource group

Create a resource group

Basics | Tags | Review + create

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. Learn more

Project details

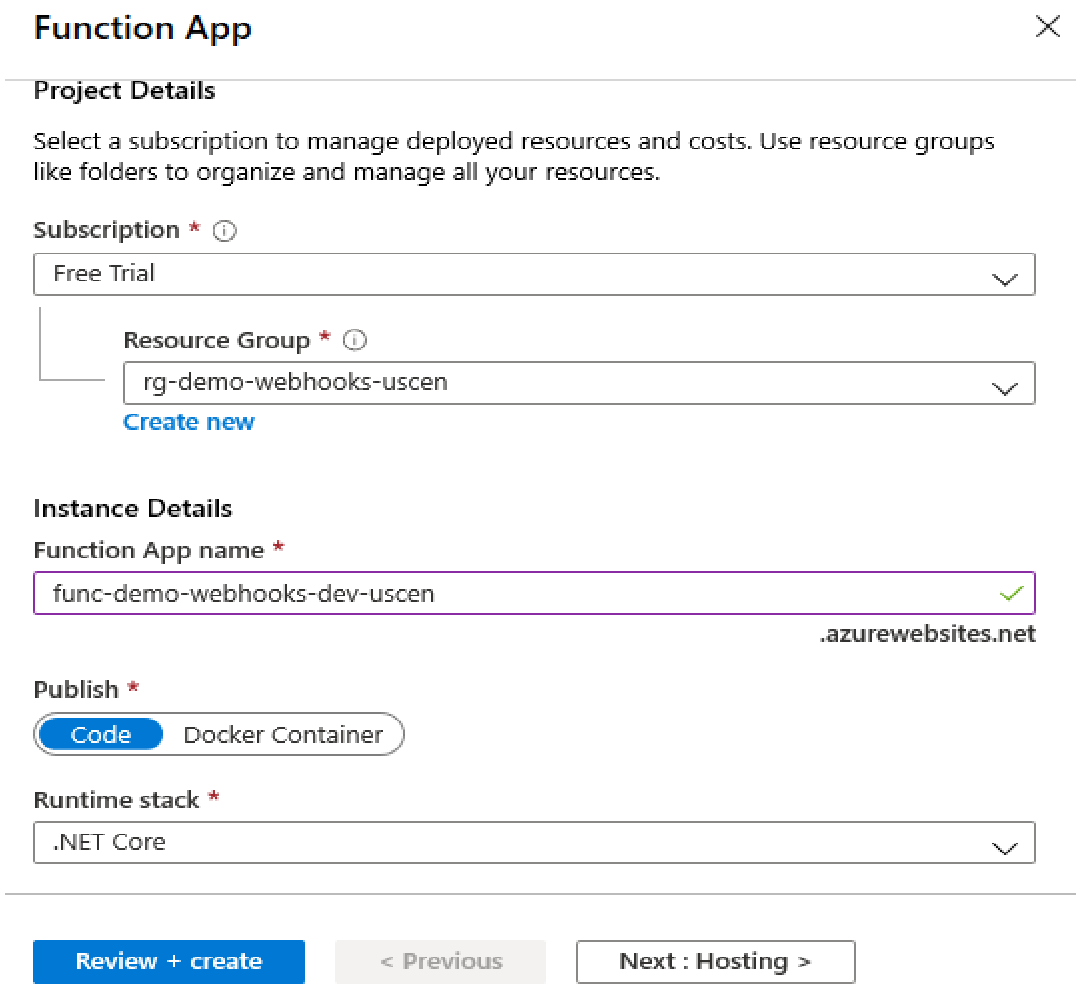
Subscription *	Free Trial
Resource group *	rg-demo-webhooks-uscen

Resource details

Region *	(US) Central US

3. Create two new “Function Apps”.

Create new Function App as shown below. Note the name has “dev” aligning it with the PowerApps dev environment.


Function App

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Free Trial
Resource Group *	rg-demo-webhooks-uscen

Create new

Instance Details

Function App name *	func-demo-webhooks-dev-uscen .azurewebsites.net
Publish *	[✓ Code] Docker Container
Runtime stack *	.NET Core

4. Under the “Function App”, add a new function of type “WebHook + API”. Choose Ïn-portal” development environment for this function.

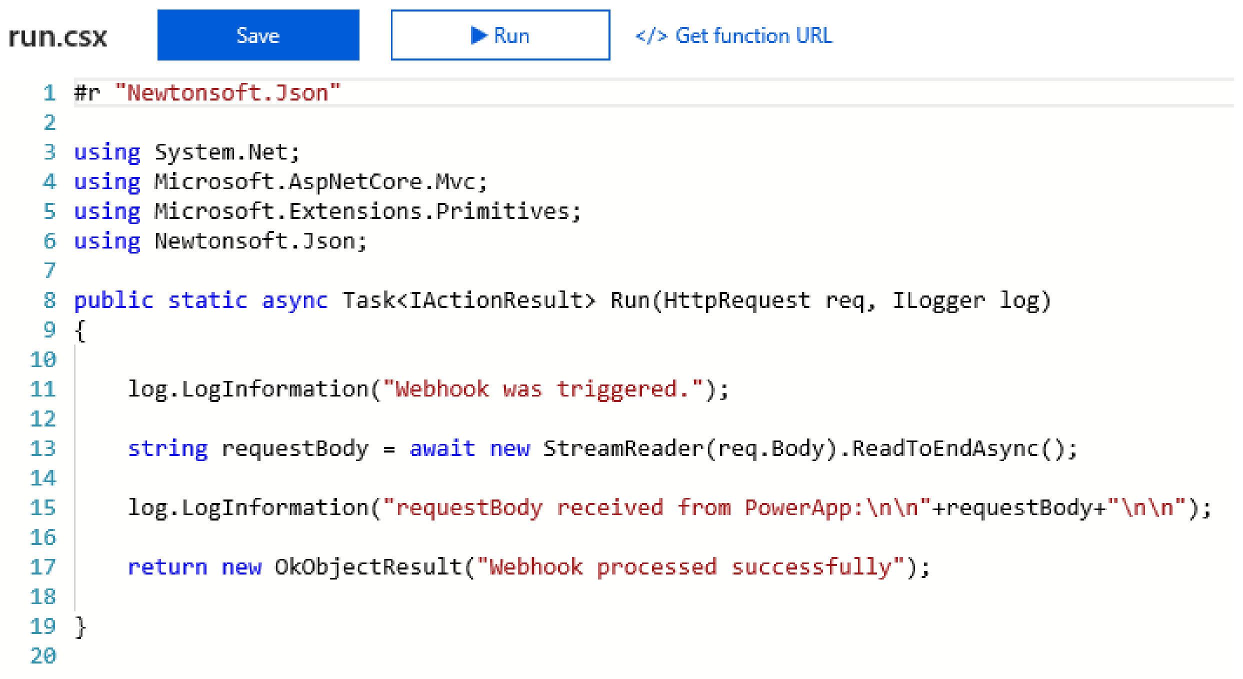
5. The function is created with default name HttpTrigger1. This should be changed.

To change it, click on Function App → Platform Features → Console.

Commands:
```
ls
rename HttpTrigger1 TimeOffRequestCreate
```

Then restart the Function App.

6. Change the default C# code as shown below. This is a demo Function App that does nothing with the JSON except dump it to the log.

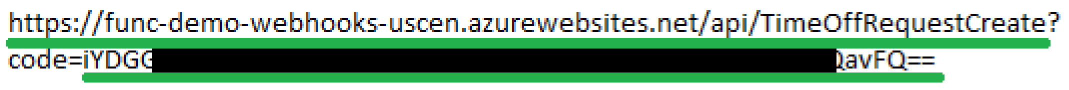


```
#r "Newtonsoft.Json"

using System.Net;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Primitives;
using Newtonsoft.Json;

public static async Task<IActionResult> Run(HttpRequest req, ILogger log)
{

    log.LogInformation("Webhook was triggered.");

    string requestBody = await new StreamReader(req.Body).ReadToEndAsync();

    log.LogInformation("requestBody received from PowerApp:\n\n"+requestBody+"\n\n");

    return new OkObjectResult("Webhook processed successfully");

}

```

7. Test the function using the Azure. Go to “Test” tab for the function, put a JSON in the Request Body. Then click “Run”. The JSON should appear in the logs.

8. Take note of the Function URL for PowerApp configuration. Click on “Get Function URL”.

The Function URL has the URL and a Code:



https://func-demo-webhooks-uscen.azurewebsites.net/api/TimeOffRequestCreate?
code=iYDGG[redacted]QavFQ==

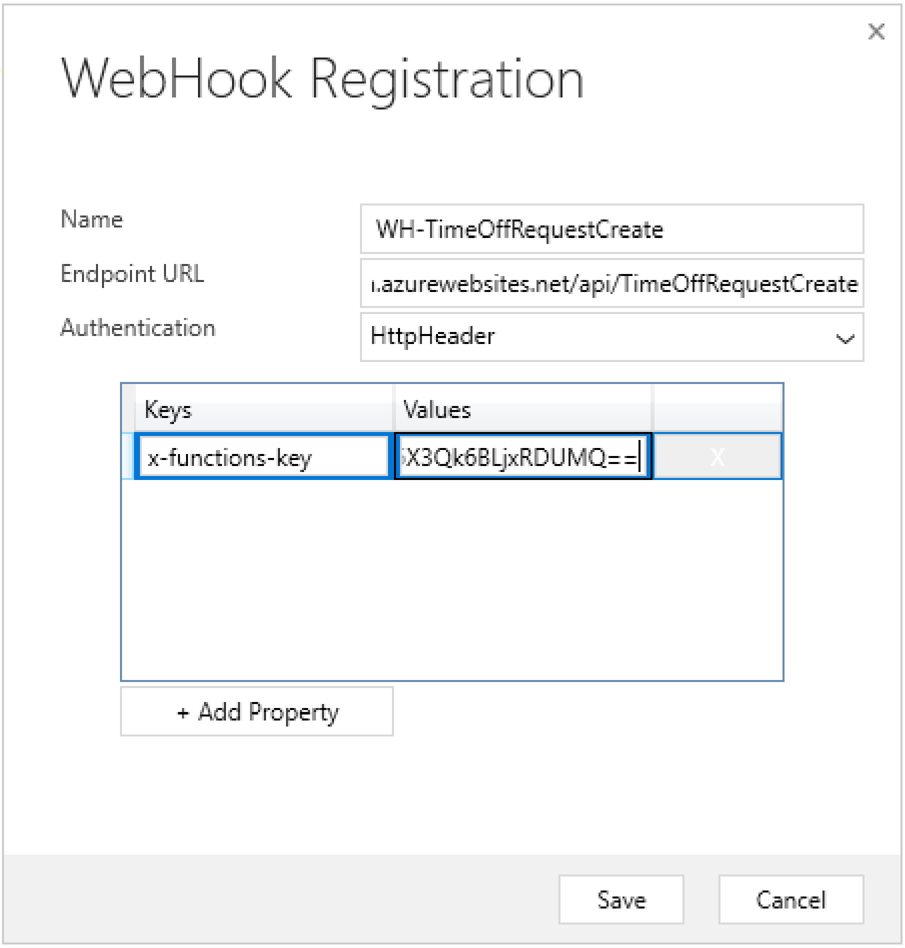
9. Repeat steps 3 – 8 to create “prod” Function App (aligning with the PowerApps environment).

Create PowerApps Web Hook

1. Create a new web hook in the dev environment. Open the Plugin Registration Tool and connect to the dev environment.

2. Click “Register” → “Register New Web Hook”.

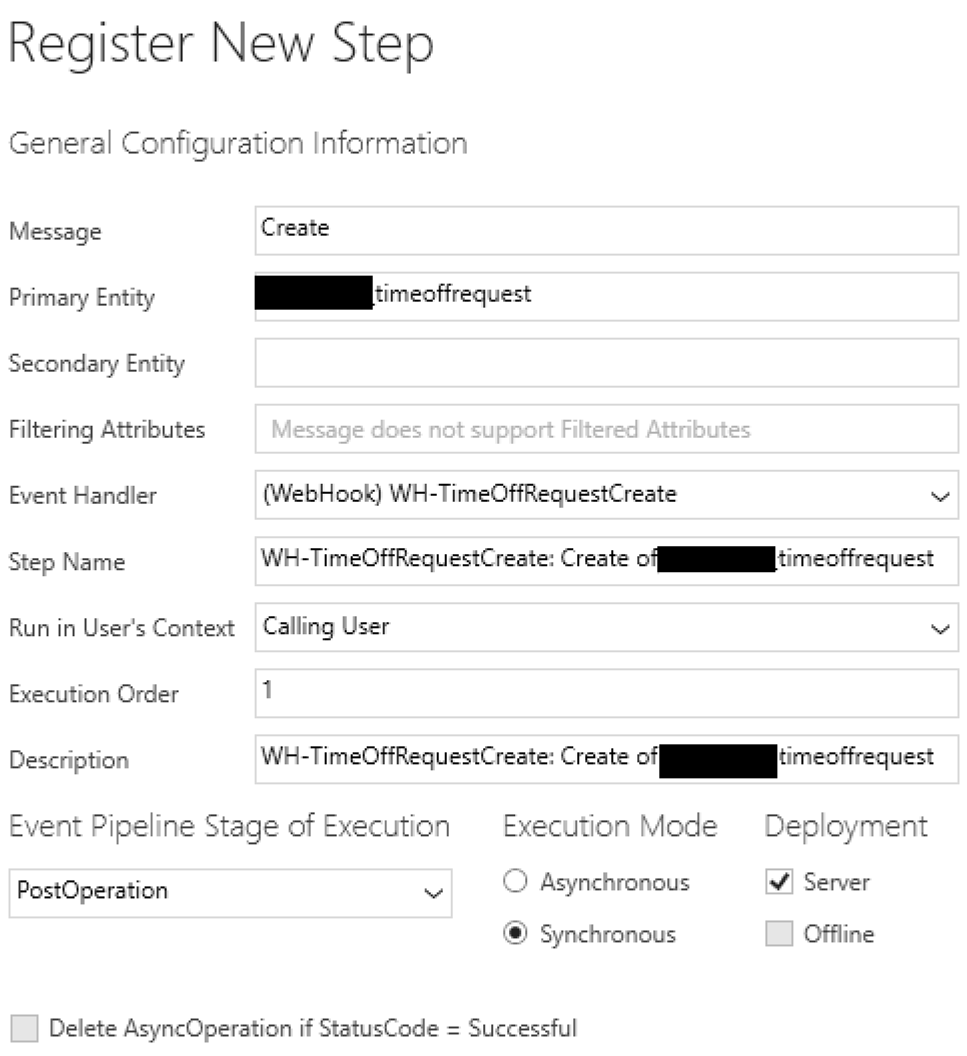
Enter the information as shown below.
- Endpoint URL is the dev Azure Function URL.
- Click “Add Property”. Key = “x-functions-key”. Value = Code from dev Azure Function.
- Click “Save”.


WebHook Registration

Name	WH-TimeOffRequestCreate
Endpoint URL	ı.azurewebsites.net/api/TimeOffRequestCreate
Authentication	HttpHeader

Keys	Values
x-functions-key	;X3Qk6BLjxRDUMQ==

3. Register a Step under the Web Hook so it is triggered when record is created.


Register New Step

General Configuration Information

Message	Create
Primary Entity	[redacted]timeoffrequest
Secondary Entity	
Filtering Attributes	Message does not support Filtered Attributes
Event Handler	(WebHook) WH-TimeOffRequestCreate
Step Name	WH-TimeOffRequestCreate: Create of [redacted]timeoffrequest
Run in User's Context	Calling User
Execution Order	1
Description	WH-TimeOffRequestCreate: Create of [redacted]timeoffrequest

Event Pipeline Stage of Execution: PostOperation

Execution Mode: ○ Asynchronous [✓ ● Synchronous]

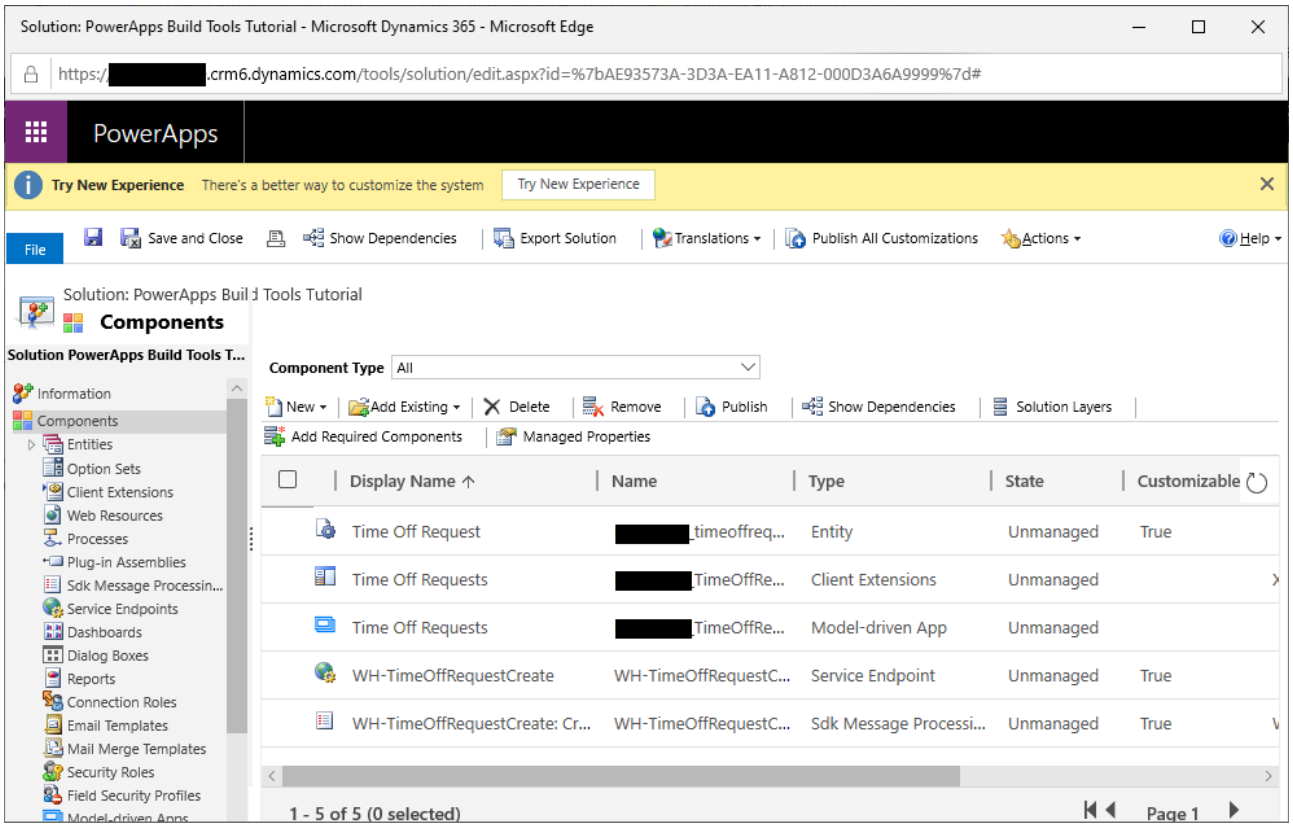
Deployment: [✓ Server] ☐ Offline

☐ Delete AsyncOperation if StatusCode = Successful

4. Test the web hook. Create a new “Time Off Request” in the PowerApp. Wait 5 minutes. Go to the Function App → Monitor. The logs should show successful invocation of the Function App.

5. Add the new web hook to the PowerApps Solution.

Open the solution → Sdk Message Processing → Add Existing
Select “Yes, include required components”.
The Components view of the Solution should look similar to below.


Solution: PowerApps Build Tools Tutorial - Microsoft Dynamics 365 - Microsoft Edge

https://[redacted].crm6.dynamics.com/tools/solution/edit.aspx?id=%7bAE93573A-3D3A-EA11-A812-000D3A6A9999%7d#

Component Type: All

Display Name ↑	Name	Type	State	Customizable
Time Off Request	[redacted]_timeoffreq...	Entity	Unmanaged	True
Time Off Requests	[redacted]TimeOffRe...	Client Extensions	Unmanaged	
Time Off Requests	[redacted]TimeOffRe...	Model-driven App	Unmanaged	
WH-TimeOffRequestCreate	WH-TimeOffRequestC...	Service Endpoint	Unmanaged	True
WH-TimeOffRequestCreate: Cr...	WH-TimeOffRequestC...	Sdk Message Processi...	Unmanaged	True

1 - 5 of 5 (0 selected) Page 1

6. Deploy the solution to build and prod environments using Azure DevOps.

Create Export from Dev
Build Managed Solution
PROD Release

7. Log on to the Prod PowerApps environment, open the solution, and confirm the web hook has been deployed to Prod. Obviously at this time, it does not have it's settings correctly configured.

Deploy PowerApps Ox Build Tools – Configure WebHook

1. In Azure DevOps, go to "Browse Marketplace" and install the "PowerApps Ox Build Tools" extension.

2. Go to the Project → Pipelines → Library.

3. Create a variable group "PROD-WH-TimeOffRequestCreate".

The 3 variables are: webHookName, webHookUrl and webHookKeyValues.

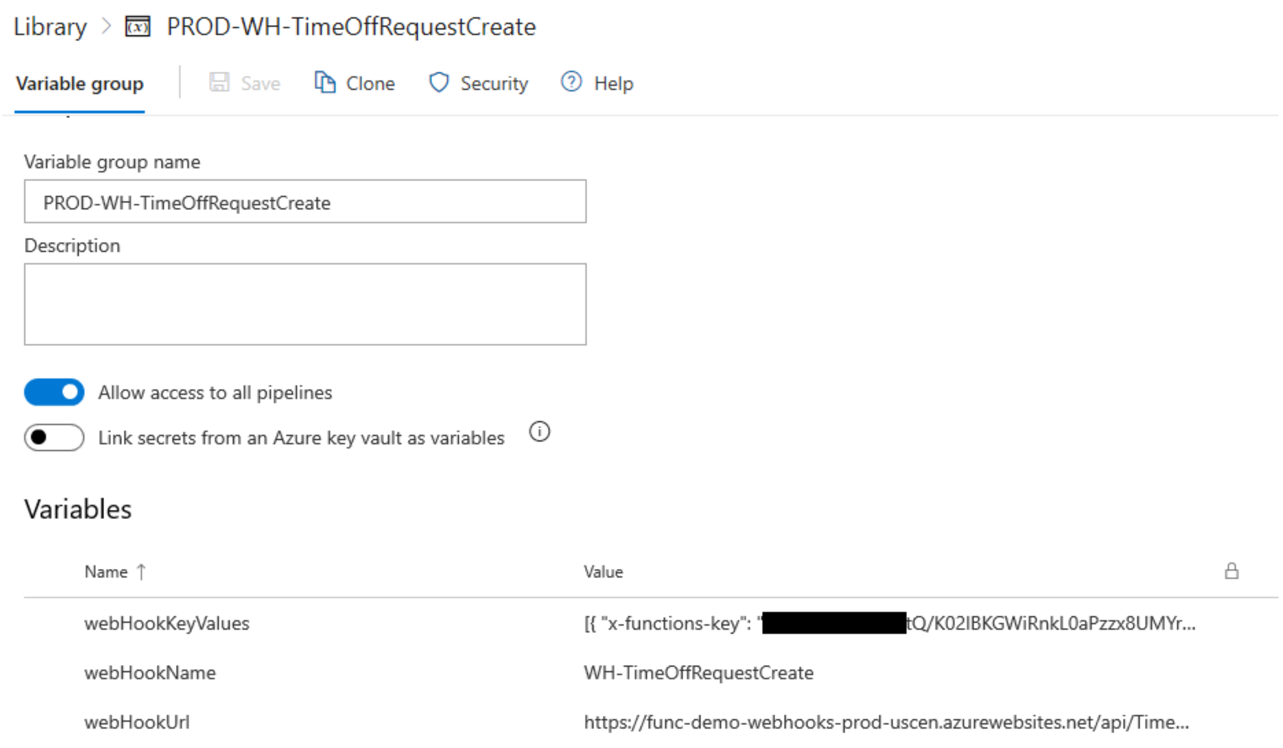
webHookUrl is the PROD Function App URL.

webHookKeyValues is a JSON string that has key value pairs. It requires key of "x-functions-key" with value of the PROD Function App Code.
NOTE: webHookKeyValues should be a secret variable in Azure DevOps. In this guide it has not been hidden.

Example:

```
[{ "x-functions-key": "XXXXXXXXXXXXkKtQ/K021BKGWiRnkL0aPzzx8UMYrmiRV3be3uwLYw==" }]
```

The Variable Group should look like below.



Variable group name: PROD-WH-TimeOffRequestCreate

Description:

Allow access to all pipelines: On

Link secrets from an Azure key vault as variables: Off

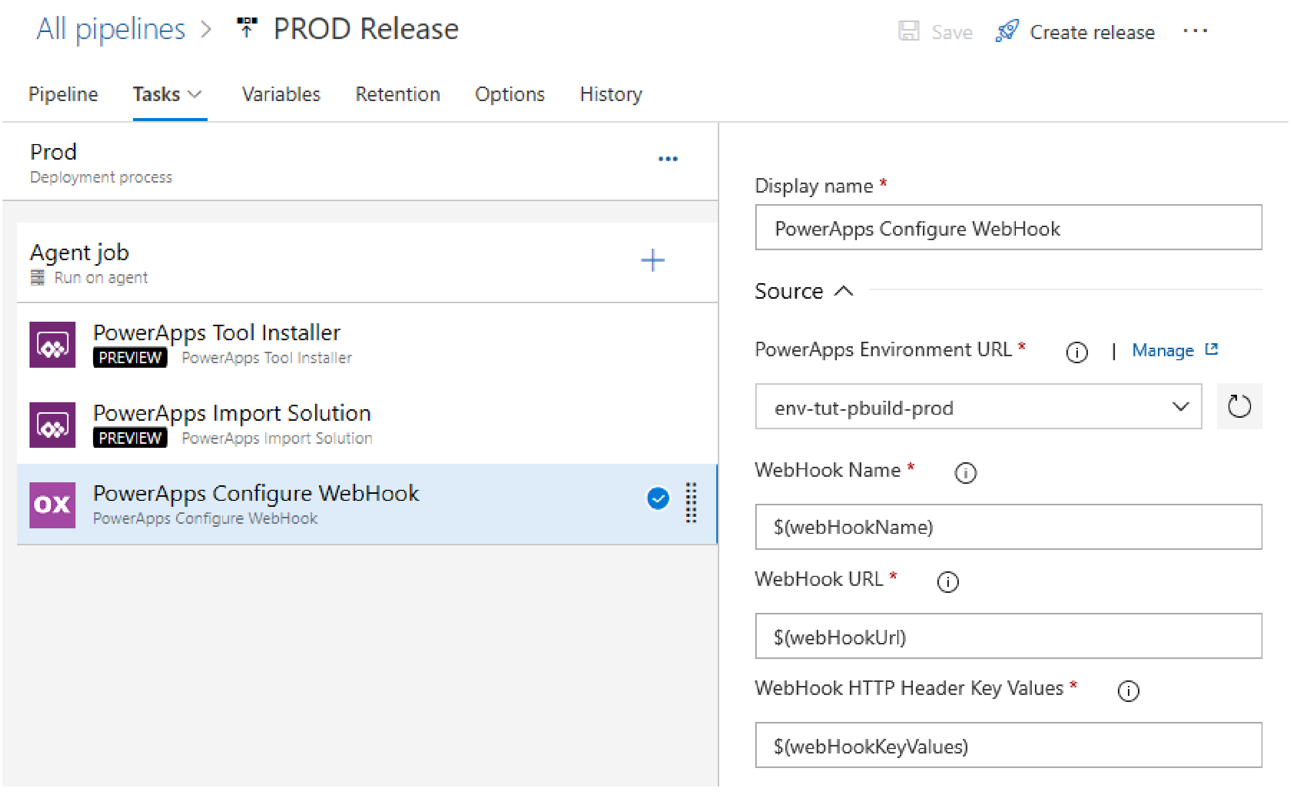
Variables

Name	Value
webHookKeyValues	[{ "x-functions-key": "[redacted]tQ/K02IBKGWiRnkL0aPzzx8UMYr...
webHookName	WH-TimeOffRequestCreate
webHookUrl	https://func-demo-webhooks-prod-uscen.azurewebsites.net/api/Time...

4. Edit the PROD Release Pipeline.

Add the task "PowerApps Configure WebHook".

Configure the parameters as shown below and save.


All pipelines > PROD Release

Save | Create release | ···

Pipeline | Tasks | Variables | Retention | Options | History

Prod
Deployment process

Agent job
Run on agent

PowerApps Tool Installer
PREVIEW PowerApps Tool Installer

PowerApps Import Solution
PREVIEW PowerApps Import Solution

PowerApps Configure WebHook
PowerApps Configure WebHook

Display name *: PowerApps Configure WebHook

Source

PowerApps Environment URL * | Manage: env-tut-pbuild-prod

WebHook Name *: $(webHookName)

WebHook URL *: $(webHookUrl)

WebHook HTTP Header Key Values *: $(webHookKeyValues)

5. Create a release and ensure it completes successfully.

6. Verify the deployment.

Log on to the PROD PowerApps environment. Create a new “Time Off Request” record.

Wait 5 minutes. Go to the PROD Function App → Monitor. The logs should show successful invocation of the Function App.

Congratulations: The Web Hook was configured via the Azure DevOps Release Pipeline.