PowerApps Ox Build Tools

Configure Service Bus EndPoint Task

Getting Started Guide

Getting Started Overview

This guide will demonstrate how to use the Configure Service EndPoint Task to automatically set Service EndPoint SAS 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 download and install Azure Service Bus Explorer (via Chocolatey).

Please follow the PowerApps Build Tools lab:

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

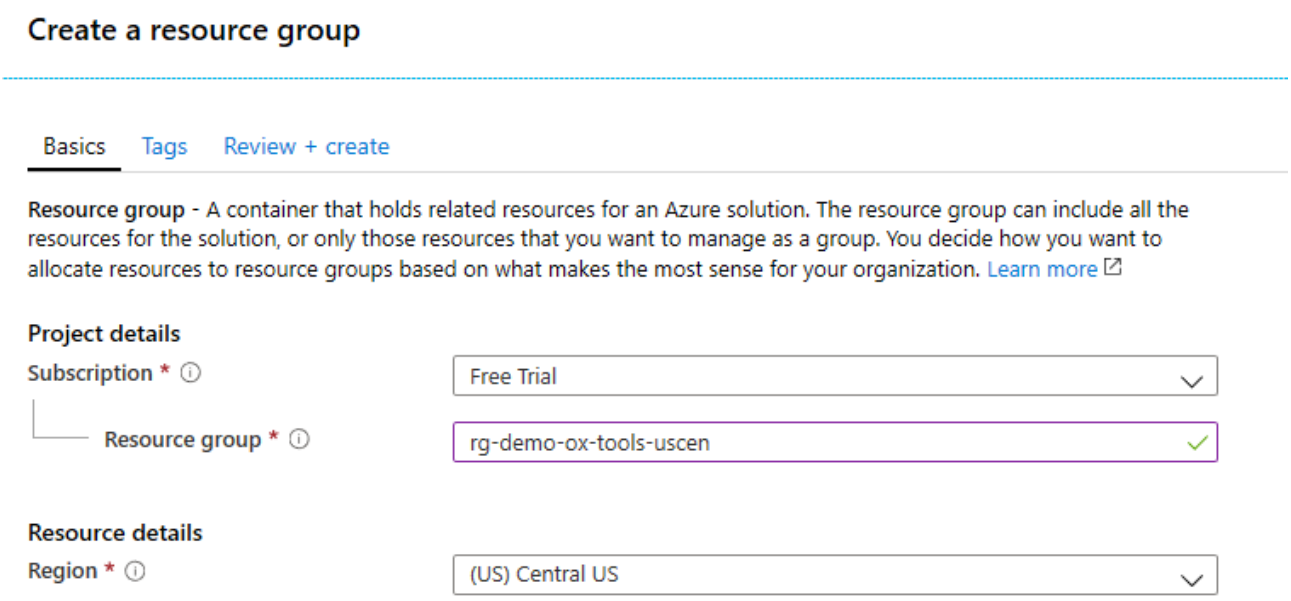
The lab completes the following tasks:
- creates 3 PowerApps CDS Environments: dev, build and prod
- creates simple PowerApp solution with 1 custom entity “Time Off Request” and a model driven app
- installs Microsoft PowerApps BuildTools into Azure DevOps
- creates Azure DevOps pipelines:
    - Create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

Instructions

Create Azure Service Bus

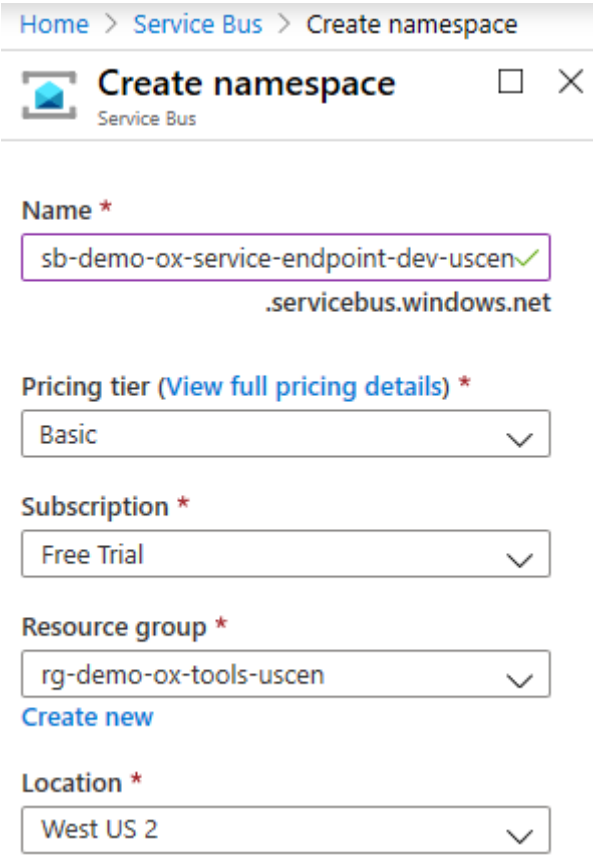
1. Log on to Azure Portal (https://portal.azure.com).

2. Create a “Resource Group”. (If you prefer, can reuse existing Resource Group). This allows you to specify the region.

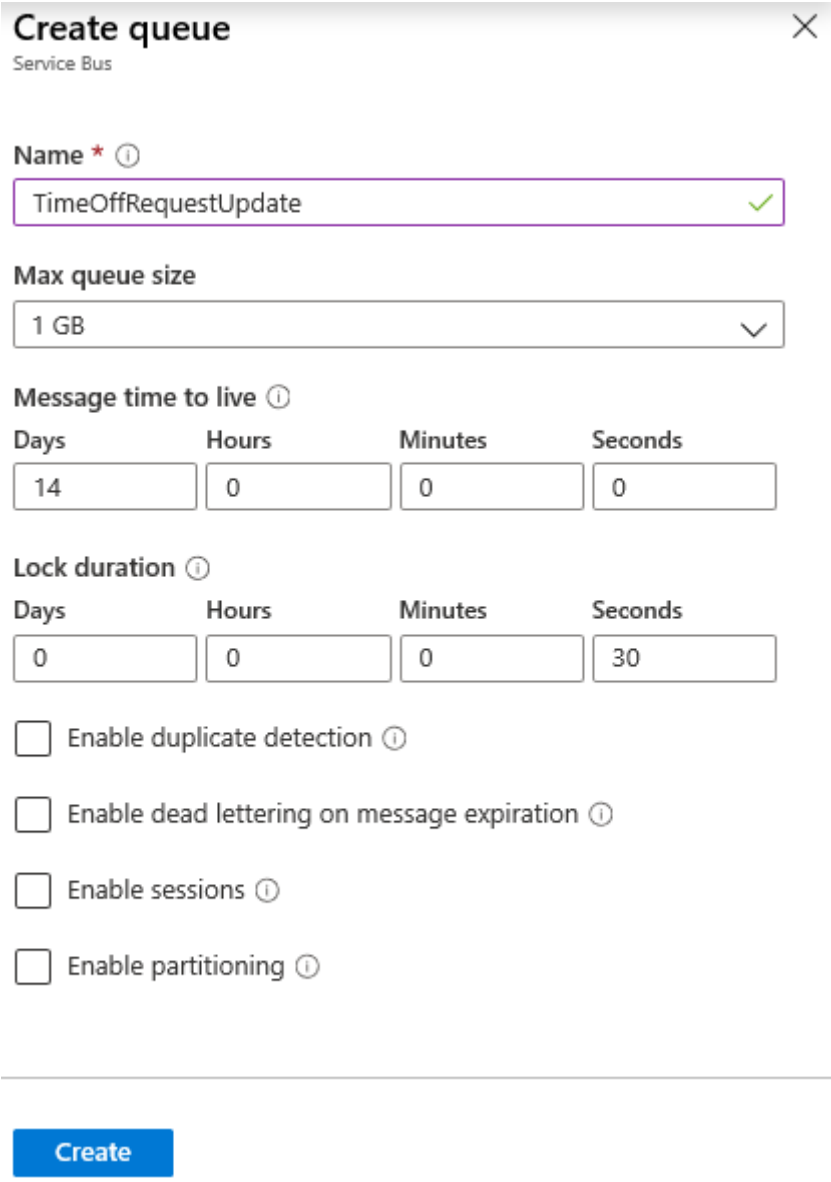


3. Create two new Azure Service Busses / Queues, for dev and prod environments.

In the Azure Portal, create a new Service Bus as shown below. Note name has “dev” to indicate dev environment.

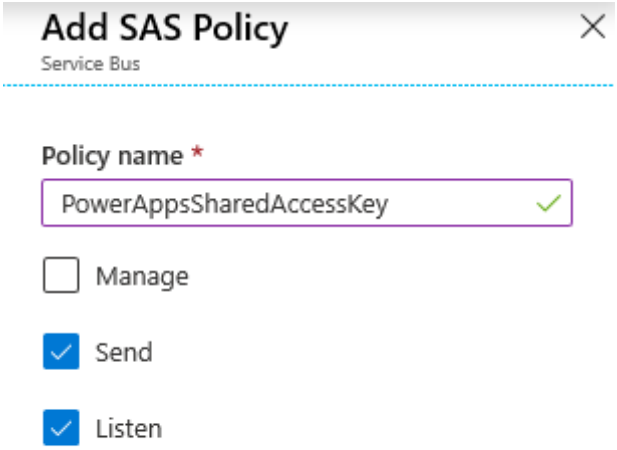


4. Open the Service Bus and create a new Queue (under Entities). The name will be “TimeOffRequestUpdate”. Example below.

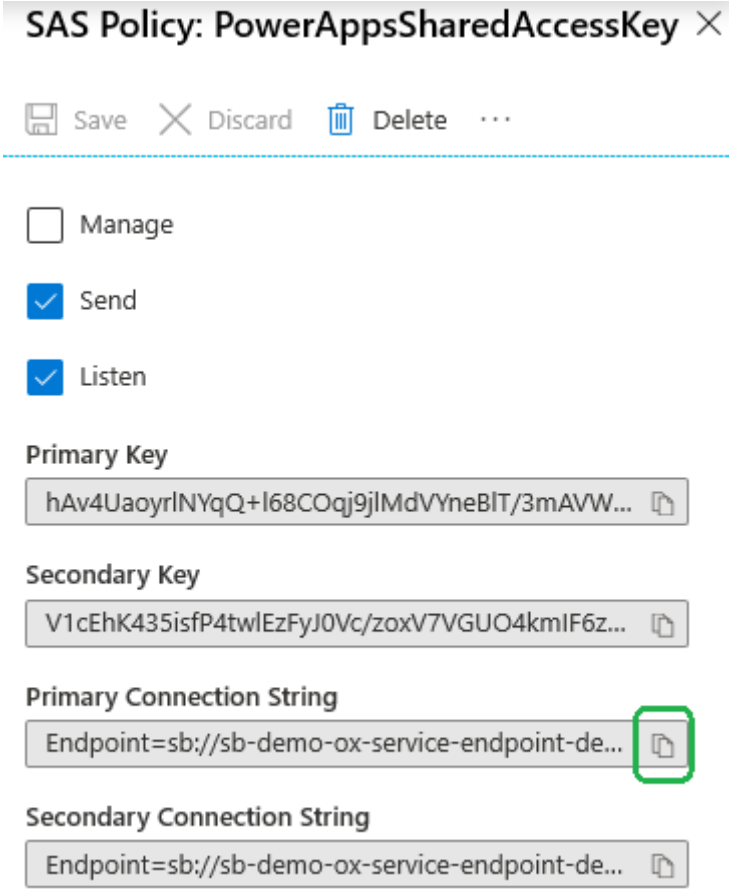


5. From the Service Bus, create a new “Shared access policy” (under Security).

Ensure Send and Listen are checked.



6. Open the newly created Share Access Policy and copy the Primary Connection String.

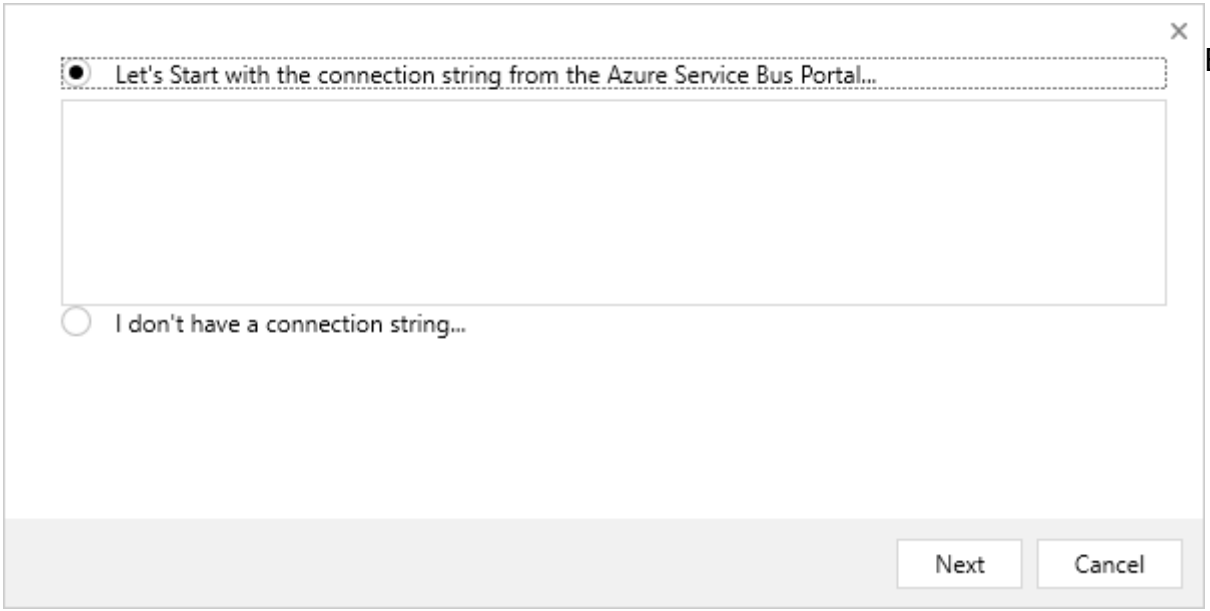


7. Repeat steps 3-6 to create “prod” Service Bus / Queue. Copy the “prod” Primary Connection String as well.

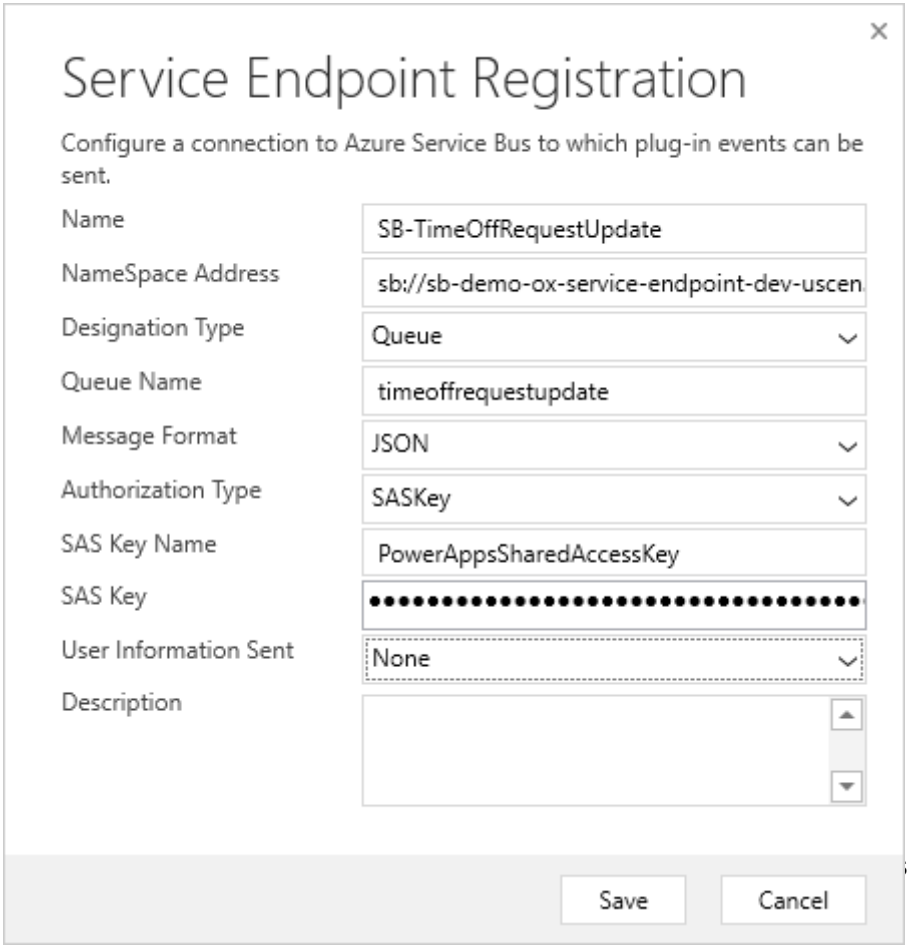
Create PowerApps Service EndPoint

1. Register the Service EndPoint in the PowerApps environment. Open the Plugin Registration Tool → Connect to DEV environment → Register New Service Endpoint.

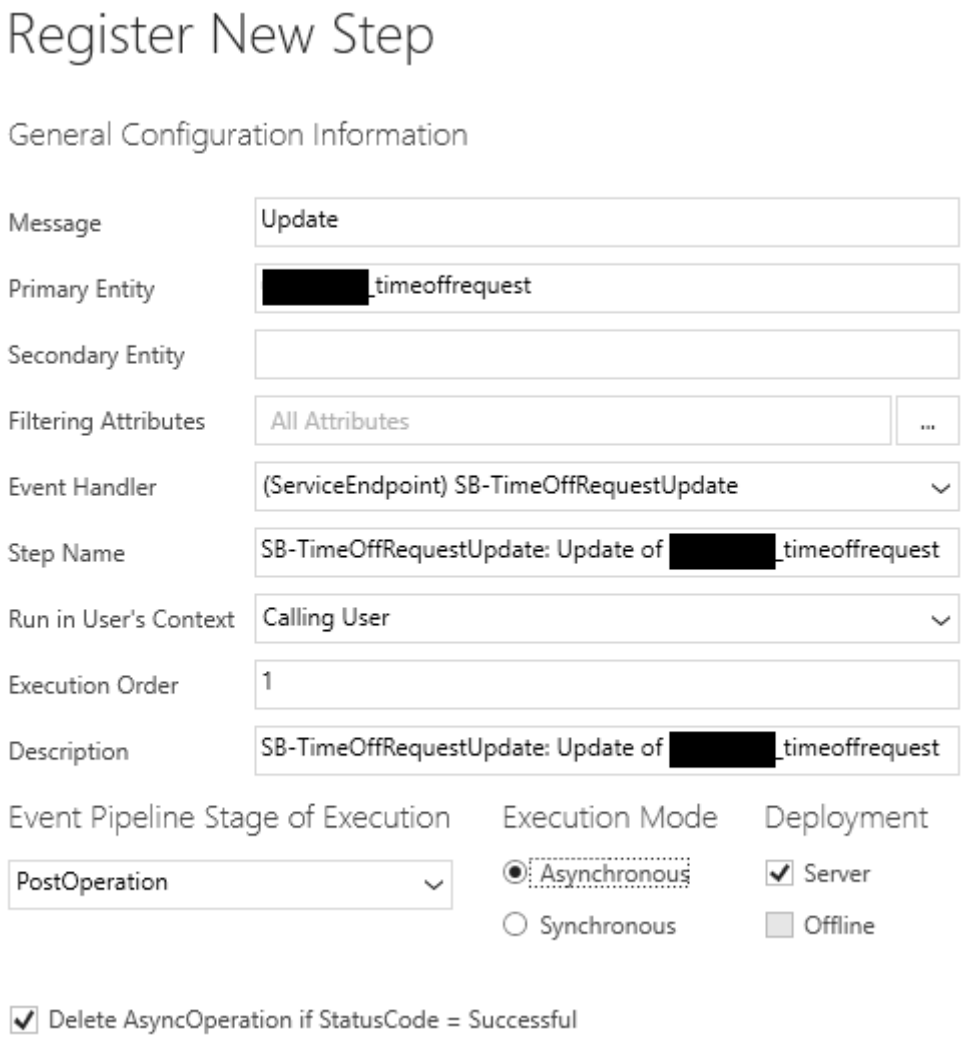
Enter the Primary Connection String from previous step.



Enter information as shown below. Ensure Message Format is JSON.



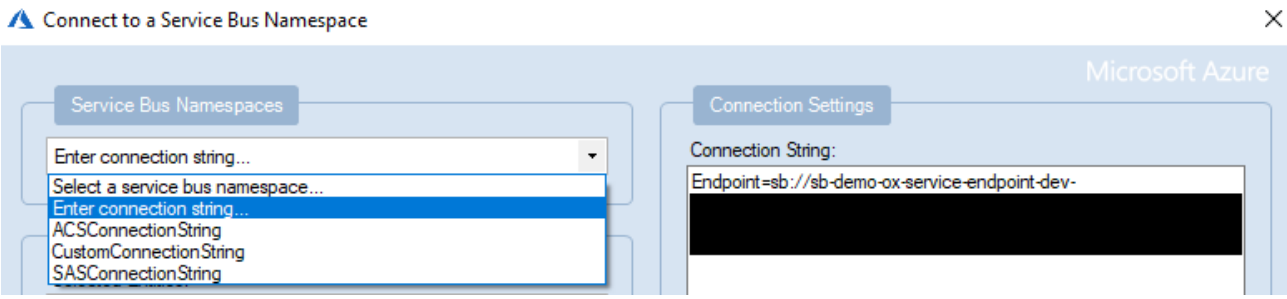
2. Register a New Step on the Service EndPoint to activate it. Ensure Execution Mode is Asynchronous.



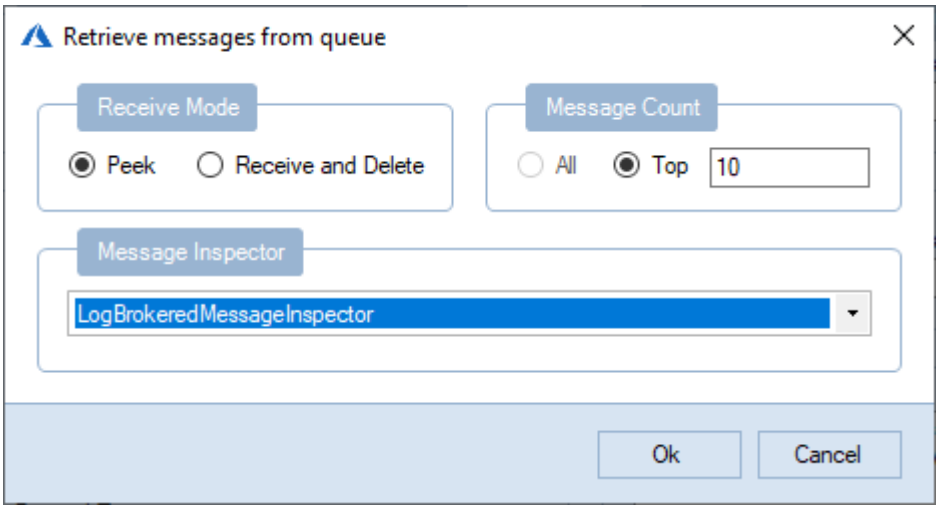
3. Test the new Service EndPoint. Log in to the PowerApps DEV environment and update a Time Off Request record.

4. Start the Service Bus Explorer. Connect to a Service Bus Namespace.

Do not use the PowerApps Connection String from previous steps. Use the default RootManageSharedAccessKey that was created automatically and has the “Manage” claim.



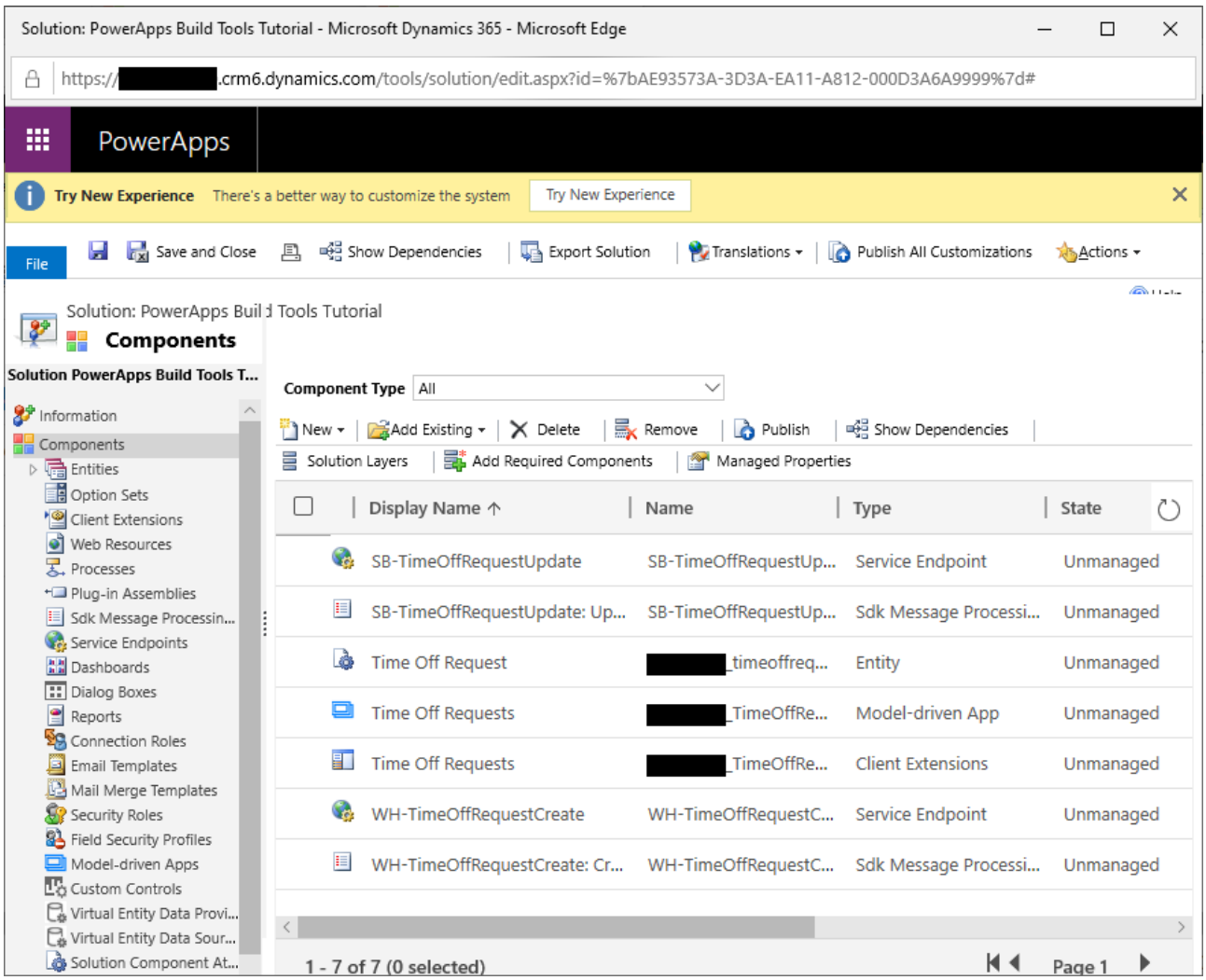
5. Click on the “timeoffrequestupdate” queue → Messages.



On next screen the update message will be displayed in JSON format.

5. Add the new service endpoint to the PowerApps Solution.

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



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 service endpoint has been deployed to Prod. Obviously at this time, it does not have it's settings correctly configured.

Deploy PowerApps Ox Build Tools – Configure Service Bus EndPoint

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

2. Go to the Project → Pipelines → Library.

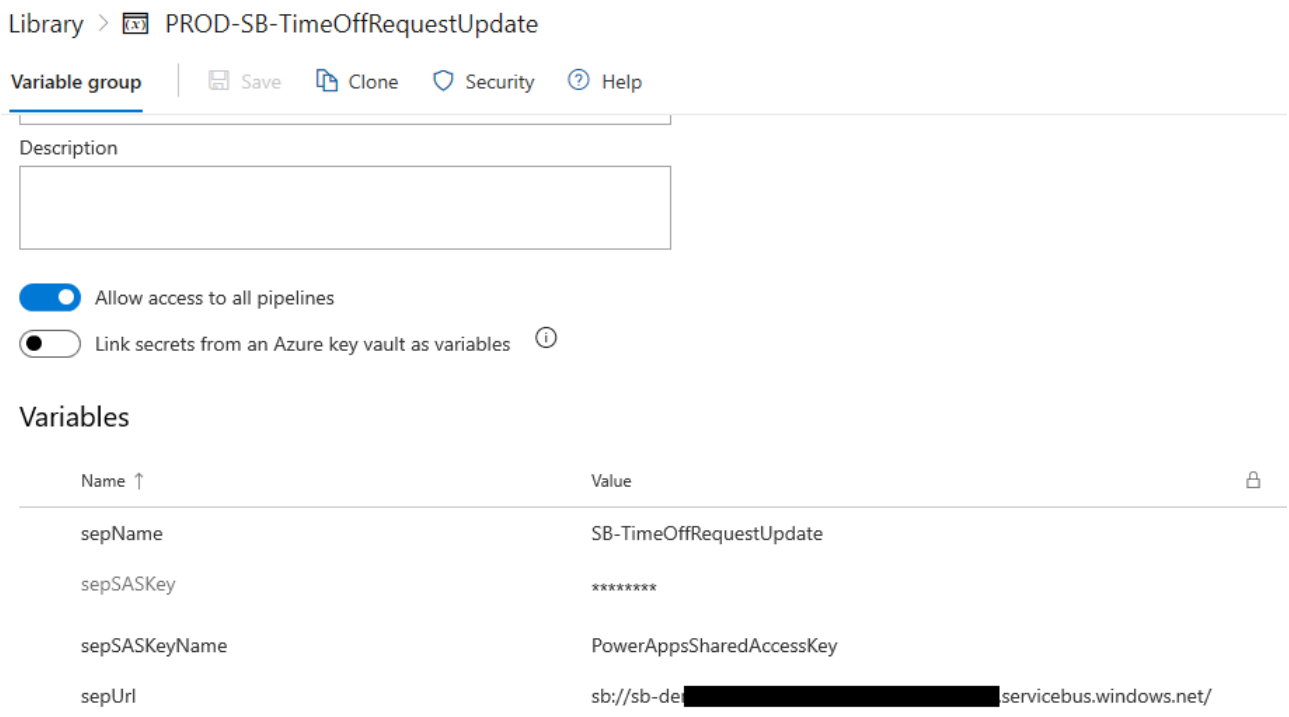
3. Create a variable group “PROD-SB-TimeOffRequestUpdate”.

Reference the PROD Primary Connection String that was copied earlier.
The 4 variables are: sepName, sepUrl, sepSASKeyName and sepSASKey

sepName	SB-TimeOffRequestUpdate (same for all environments)
sepUrl	URL from the PROD Primary Connection String
sepSASKeyName	SharedAccessKeyName from the PROD Primary Connection String
sepSASKey	SharedAccessKey from the PROD Primary Connection String

Please note that sepUrl includes the “sb://” protocol identifier, and “sepSASKey” should be a secret variable.

The Variable Group should look like below.

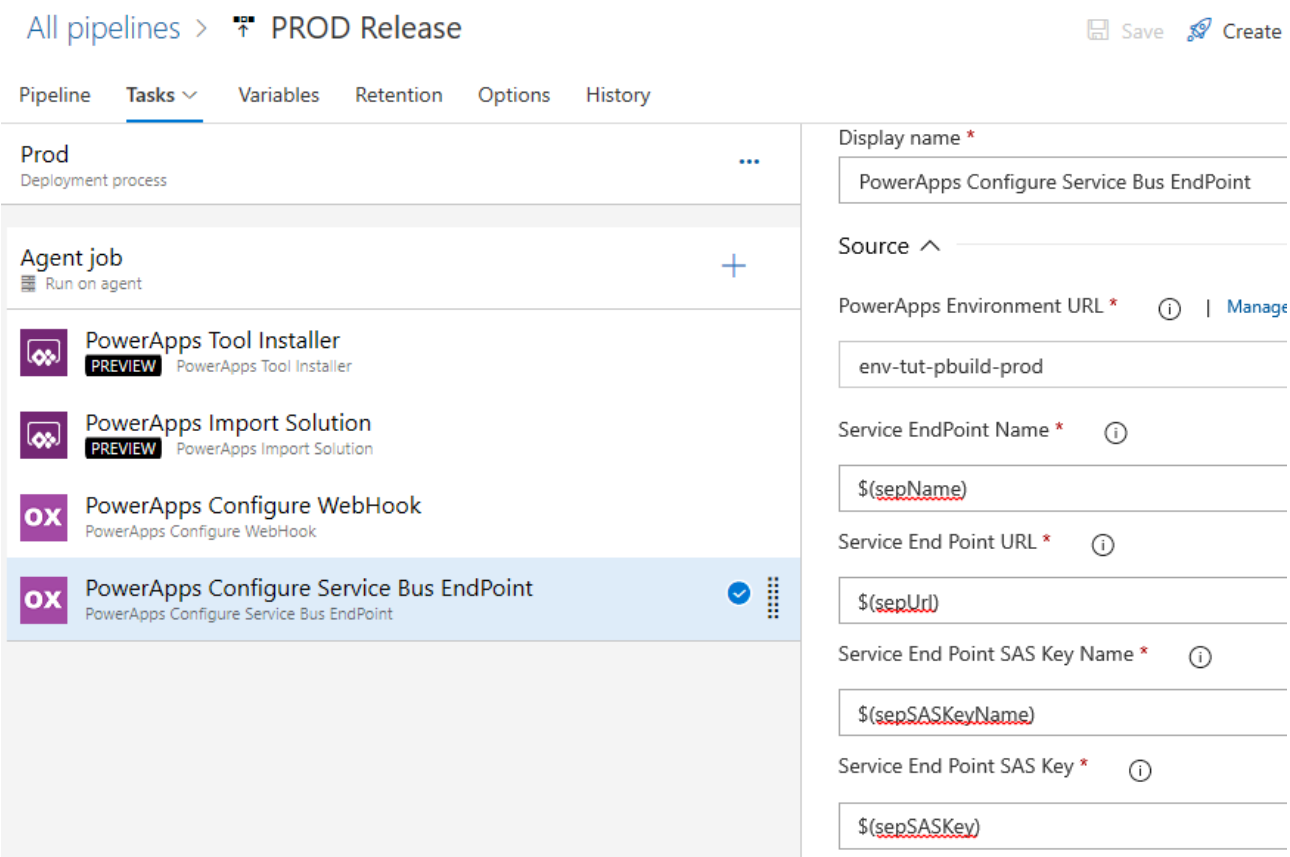


4. Edit the PROD Release Pipeline.

Link the variable group “PROD-SB-TimeOffRequestUpdate”.

Add the task “PowerApps Configure Service Bus EndPoint”.

Configure the parameters as shown below and save.



5. Create a release and ensure it completes successfully.

6. Verify the deployment.

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

Open Service Bus Explorer and connect to the PROD service bus using the RootManageSharedAccessKey Primary Connection String. The message will be queued and can be opened and verified.

Congratulations: The Service EndPoint was configured via the Azure DevOps Release Pipeline.