

Leveraging the Microsoft Graph API using Azure Logic Apps

James Corbould

Abhishek Kumar

Contents

Objective	3
Prerequisites	4
Steps.....	5
Login to OneDrive and create a new folder and Excel document	5
Login to Azure Portal and create new Logic App	7
Login to Graph Explorer and create Webhook	11
Modify Excel document to test webhook notification and email alert	13

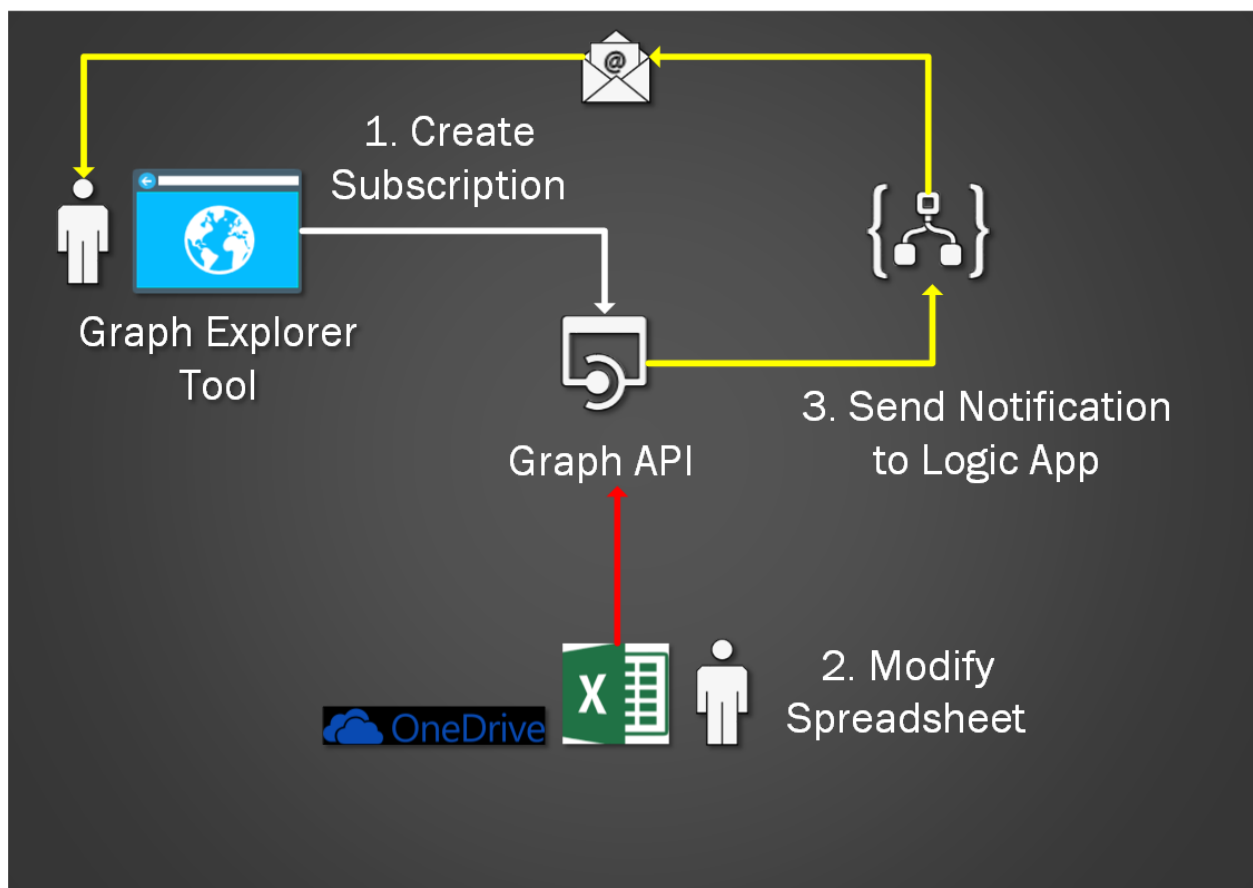
Objective

Receive an email notification when an Excel spreadsheet hosted in OneDrive is modified.

This is achieved via the Graph API which detects when the spreadsheet has been changed. A webhook will be setup via the Graph Explorer tool that indicates the URL listening for changes: a webhook notification will be sent to the subscribing URL, which will be a Logic App endpoint. The Logic App will receive the notification JSON message and use the Outlook connector to send an email alert to the configured email address.

The email will alert the user that the Excel spreadsheet has been modified.

A summary of the flow can be viewed in the diagram below:



Prerequisites

- Azure account - You can [Open an Azure account for free](#) or [Activate Visual Studio subscriber benefits](#).
- Office 365 with OneDrive (a limited number of trial Office 365 licenses are available for use for the day, please see James or Abhishek)

Steps

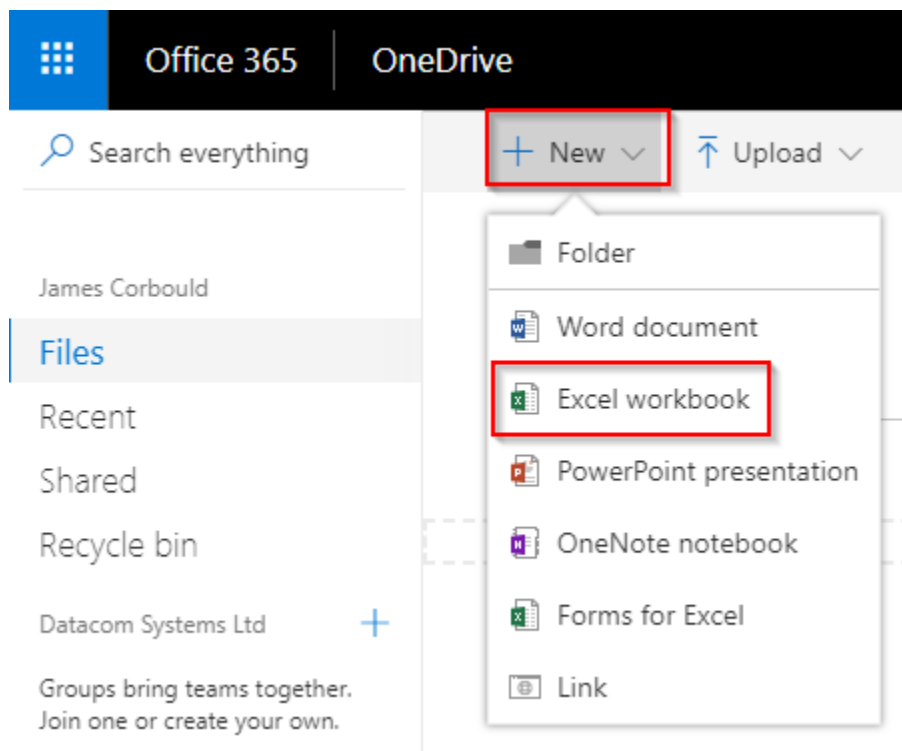
To build the solution in this lab you have to follow the steps described in this section. From a high level view the steps are:

1. Login to [OneDrive](#) using your O365 account and create a new folder and a new Excel document.
2. Login into the [Azure Portal](#) and create a new Logic App to receive webhook notifications and send an alert email.
3. Login to [Graph Explorer](#) using your O365 account and create a new webhook subscription, with the notification URL matching the URL of your newly created Logic App.

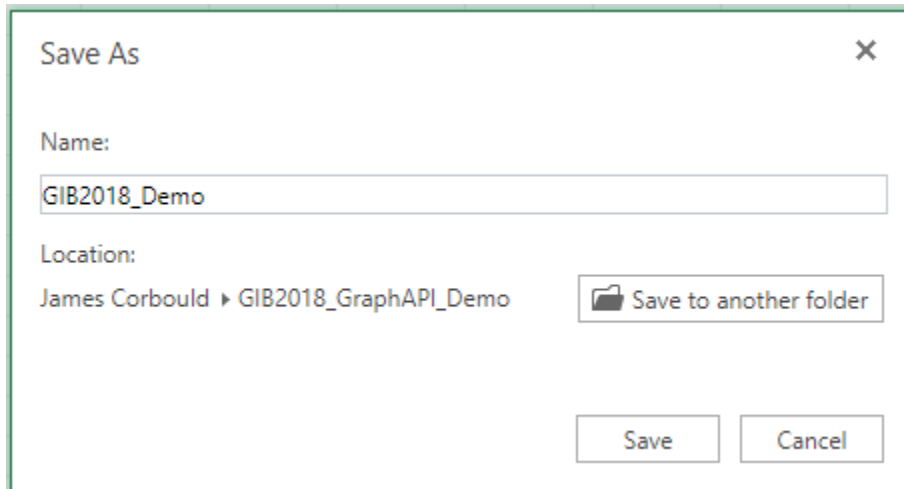
Login to OneDrive and create a new folder and Excel document

Navigate to [OneDrive](#) Live using your browser and sign in using your O365 account.

Create a new folder named `GIB2018_GraphAPI_Demo` in the directory root location and also create a new Excel spreadsheet named `GIB2018_Demo.xlsx`, as shown below:



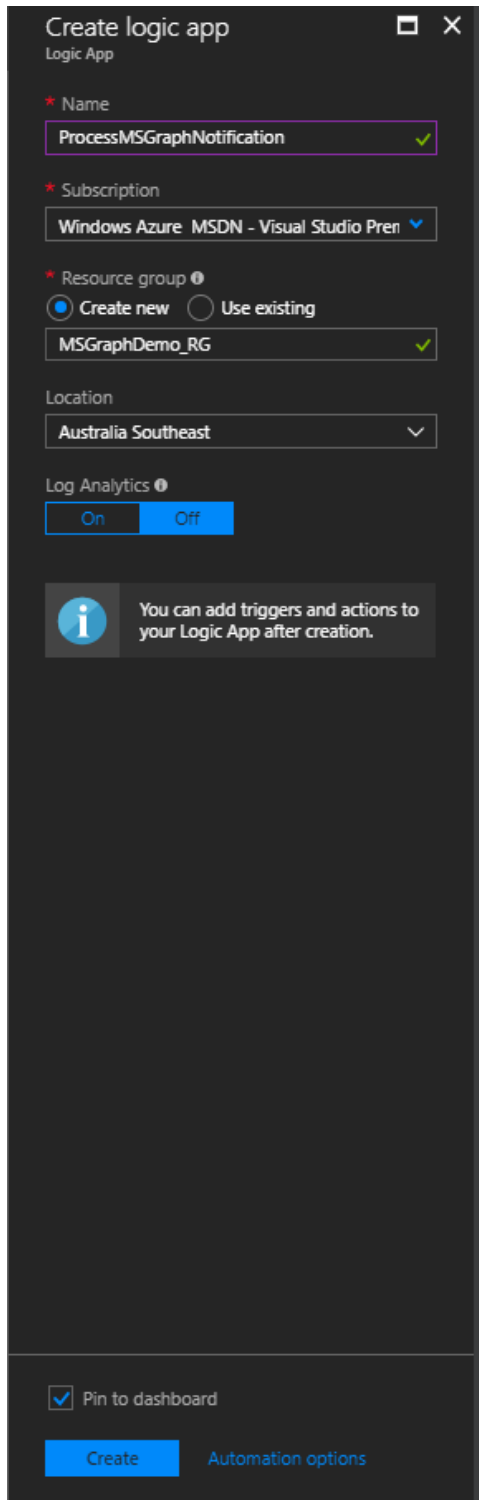
When selecting New → Excel workbook, a new Excel document will open in the browser. Click File → Save As, choosing the filename GIB2018_Demo.xlsx:



Login to Azure Portal and create new Logic App

Login to the [Azure Portal](#) using your Azure account.

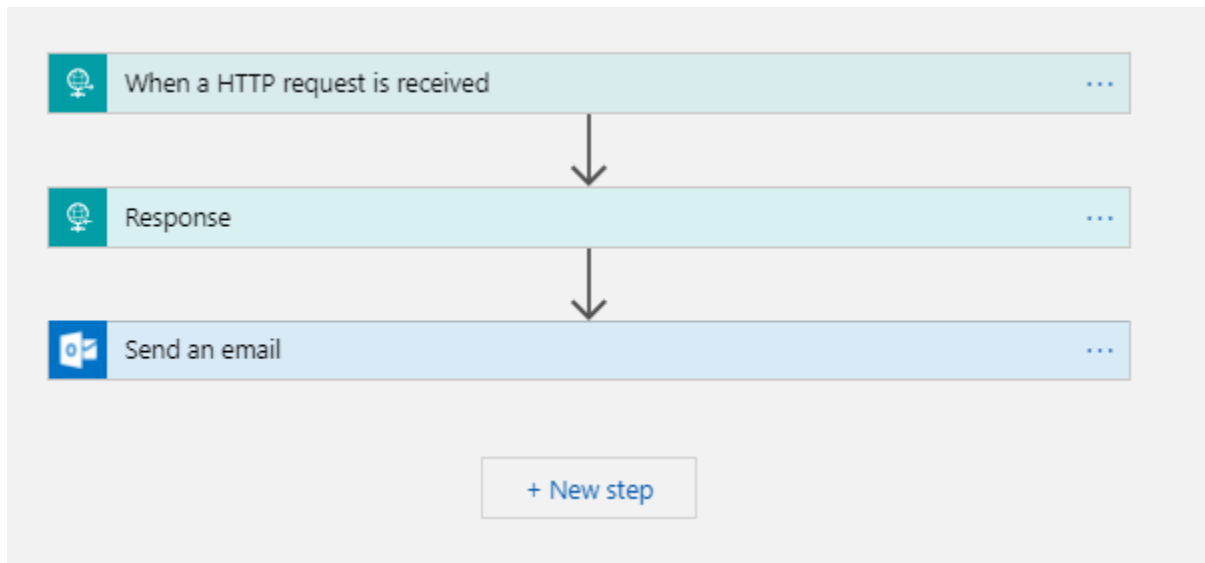
Create a new Logic App called `ProcessMSGraphNotification`:



The screenshot shows the 'Create logic app' form in the Azure Portal. The form is titled 'Create logic app' with a subtitle 'Logic App'. It contains the following fields and options:

- Name:** A text input field containing 'ProcessMSGraphNotification' with a green checkmark icon to its right.
- Subscription:** A dropdown menu showing 'Windows Azure MSDN - Visual Studio Pre' with a blue downward arrow.
- Resource group:** A section with two radio buttons: 'Create new' (selected) and 'Use existing'. Below the radio buttons is a text input field containing 'MSGraphDemo_RG' with a green checkmark icon to its right.
- Location:** A dropdown menu showing 'Australia Southeast' with a blue downward arrow.
- Log Analytics:** A section with two buttons: 'On' (highlighted in blue) and 'Off'.
- Information box:** A grey box with a blue 'i' icon and the text: 'You can add triggers and actions to your Logic App after creation.'
- Pin to dashboard:** A checkbox that is checked, with the text 'Pin to dashboard' next to it.
- Buttons:** At the bottom, there is a blue 'Create' button and a link 'Automation options'.

This will be a very simple Logic App:



Configure the HTTP request connector as follows:

When a HTTP request is received

HTTP POST URL: `https://prod-09.australiasoutheast.logic.azure.com:443/workflows/c1f...`

Request Body JSON Schema

```
{
  "properties": {
    "headers": {
      "properties": {
        "Connection": {
          "type": "string"
        },
        "Content-Length": {
          "type": "string"
        }
      }
    }
  }
}
```

[Use sample payload to generate schema](#)

[Show advanced options](#) ▾

Cut and paste the following for the request body JSON schema:

```
{
  "properties": {
    "headers": {
      "properties": {
        "Connection": {
          "type": "string"
        },
        "Content-Length": {
          "type": "string"
        },
        "Content-Type": {
          "type": "string"
        },
        "Host": {
          "type": "string"
        }
      },
      "type": "object"
    },
    "queries": {
      "properties": {
        "validationToken": {
          "type": "string"
        }
      },
      "type": "object"
    }
  },
  "type": "object"
}
```

Configure the HTTP response connector as follows:

The screenshot shows the 'Response' connector configuration in a Logic App. The 'Status Code' is set to 200. The 'Body' field contains a dynamic content token 'validationToken' with a close button. Below the body field is a 'Response Body JSON Schema' section with a red squiggly line indicating an error. A link 'Use sample payload to generate schema' is at the bottom.

Check in the Logic App Code View that validationToken corresponds to the following dynamic content and change it in the Code View if it is different:

```
"@triggerOutputs()?['queries']?['validationToken']"
```

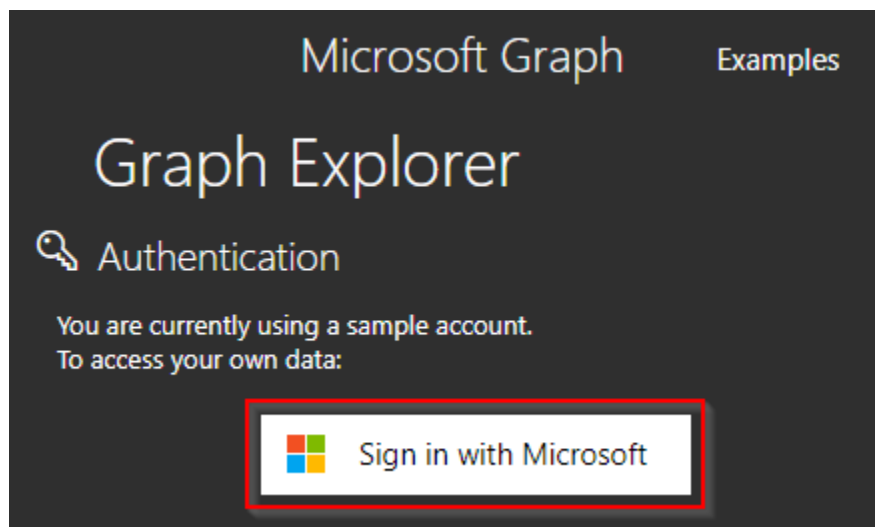
```
"definition": {
  "$schema": "https://schema.management.azure.com/providers/Microsoft.Logic/schemas/2016-09-01/workflowdefinition.json#",
  "actions": {
    "Response": {
      "inputs": {
        "body": "@triggerOutputs()?['queries']?['validationToken']",
        "statusCode": 200
      },
      "kind": "Http",
      "runAfter": {},
      "type": "Response"
    }
  }
}
```

(Note: the validationToken is used to verify to the Graph API that the notification URL exists. It is provided in the first call from the Graph API to the notification URL and returned back by the Logic App).

Configure the Outlook connector:

Login to Graph Explorer and create Webhook

Navigate to the [Graph Explorer tool](#) and login using your O365 account.



Create a webhook subscription using the beta API endpoint using HTTP POST:

```
https://graph.microsoft.com/beta/subscriptions
```

POST

beta

https://graph.microsoft.com/beta/subscriptions

Request Body Request Headers

```

{
  "changeType": "updated",
  "notificationUrl": "https://prod-09.australiasoutheast.logic.azure.com:443/workflows/c1f06efe07",
  "resource": "me/drive/root",
  "expirationDateTime": "2018-03-25T00:00:00.000Z",
  "clientState": "SecretClientState"
}

```

✓ Success - Status Code 201 4004ms

Response Preview Response Headers

```

{
  "@odata.context": "https://graph.microsoft.com/beta/$metadata#subscriptions/$entity",
  "id": "3b757144-4699-43c6-a712-8472f8cd467a",
  "resource": "me/drive/root",
  "changeType": "updated",
  "clientState": "SecretClientState",
  "notificationUrl": "https://prod-09.australiasoutheast.logic.azure.com:443/workflows/c1f06efe",
  "expirationDateTime": "2018-03-25T00:00:00Z"
}

```

Specify the request body like the example below in Graph Explorer, using the Logic App HTTP POST URL for the JSON notificationUrl parameter:

```

{
  "changeType": "updated",
  "notificationUrl": {Logic App HTTP POST URL here},
  "resource": "me/drive/root",
  "expirationDateTime": "2018-03-25T00:00:00.000Z",
  "clientState": "SecretClientState"
}

```

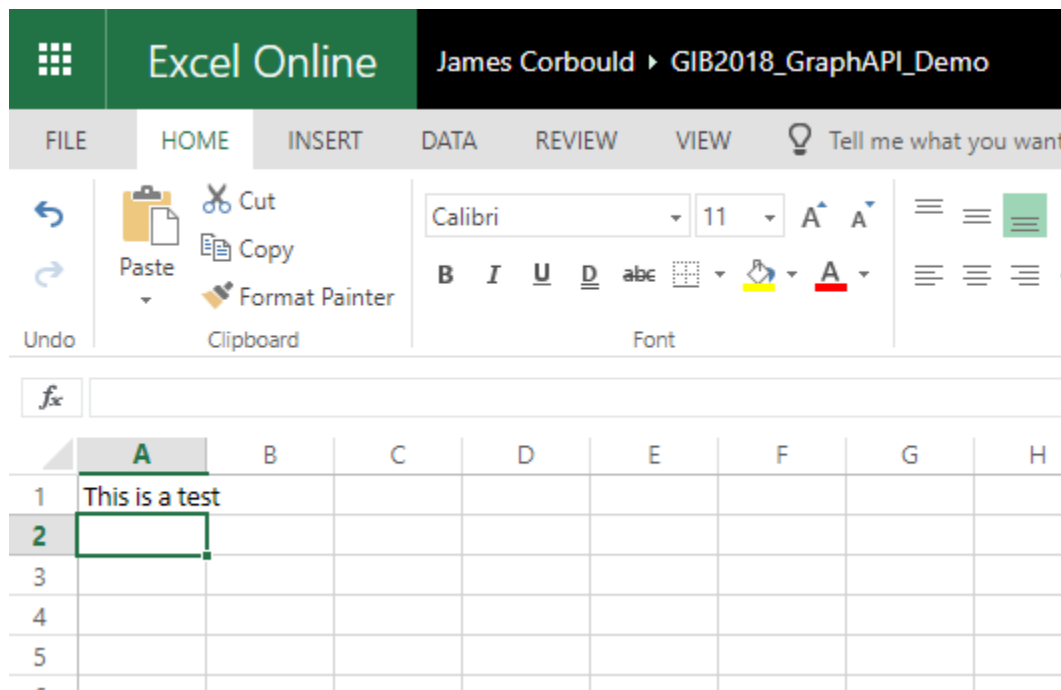
Note:

- Changes to any documents in the root of OneDrive will cause the webhook notification to fire.
- Set the value for `expirationDateTime` to a future date and time, such that it will be last the duration of this demo.

Check that HTTP status code 201 `Created` is returned, indicating that the webhook notification has been successfully created in MS Graph.

Modify Excel document to test webhook notification and email alert

To test the webhook notification, modify the spreadsheet located on OneDrive.



The Graph API should send a notification to the Logic App URL and an email should be received in Outlook:

OneDrive Excel spreadsheet has been modified



Mr Logic App

Today, 11:44 PM

James Corbould

This message was sent with low importance.

Please check the changes and approve.