



SharePoint Development using Azure Functions and CSOM

Fabian G. Williams
Practice Director
Withum Digital

About the Speaker

Fabian Williams, MVP, MCSD, MCDBa, MCSE
Practice Director for Intelligent Process Automation
Withum Digital

-  www.fabiangwilliams.com
-  [@FabianWilliams](https://twitter.com/FabianWilliams)
-  linkedin.com/in/fabiangwilliams
-  fabian@adotob.com



What is this
Azure Function
you speak of?

How does
Azure Function
Work

Session
Ingredients -
High Level

Demo 1 –
Runtime
Experience

Deconstructing
Demo 1

Ideas of Azure
Functions
with/w-out
SharePoint

Demo 2 –
SharePoint Site
Provisioning
Azure Function

Questions and
Answers

Why do this session? Why does this matter? What Problem does it solve?

TL;DR;

So, this is a leave behind /hand out.... If you were not present when I delivered this session this will help frame why I feel this topic is important.

- SharePoint Farm solutions are in the past & even if you have an On Premises environment, you should be writing solutions that are portable i.e. can work both On Prem & In the Cloud – Azure Functions give you that
- There are other options other than writing code to run in the Web Browser aka JavaScript that are available to back end developers [like myself] who for whatever reason or another don't want to keep up with the cool kids ☺ -- Azure Functions give you that
- Azure Functions are multi faceted. What does that mean? I can use “THE SAME” Azure Function that I write for my SharePoint solution for my Web Solution, for my Mobile Solution, for whatever else... do you know why? Its event driven. I look out for an interaction, I respond to it. Input/Output

STOP



www.shutterstock.com · 31370068

<https://github.com/fabianwilliams/azurefunctionsandcsom>

<https://github.com/fabianwilliams/AzureFunctionsDemoware>

What is an Azure Function? How do you define it?

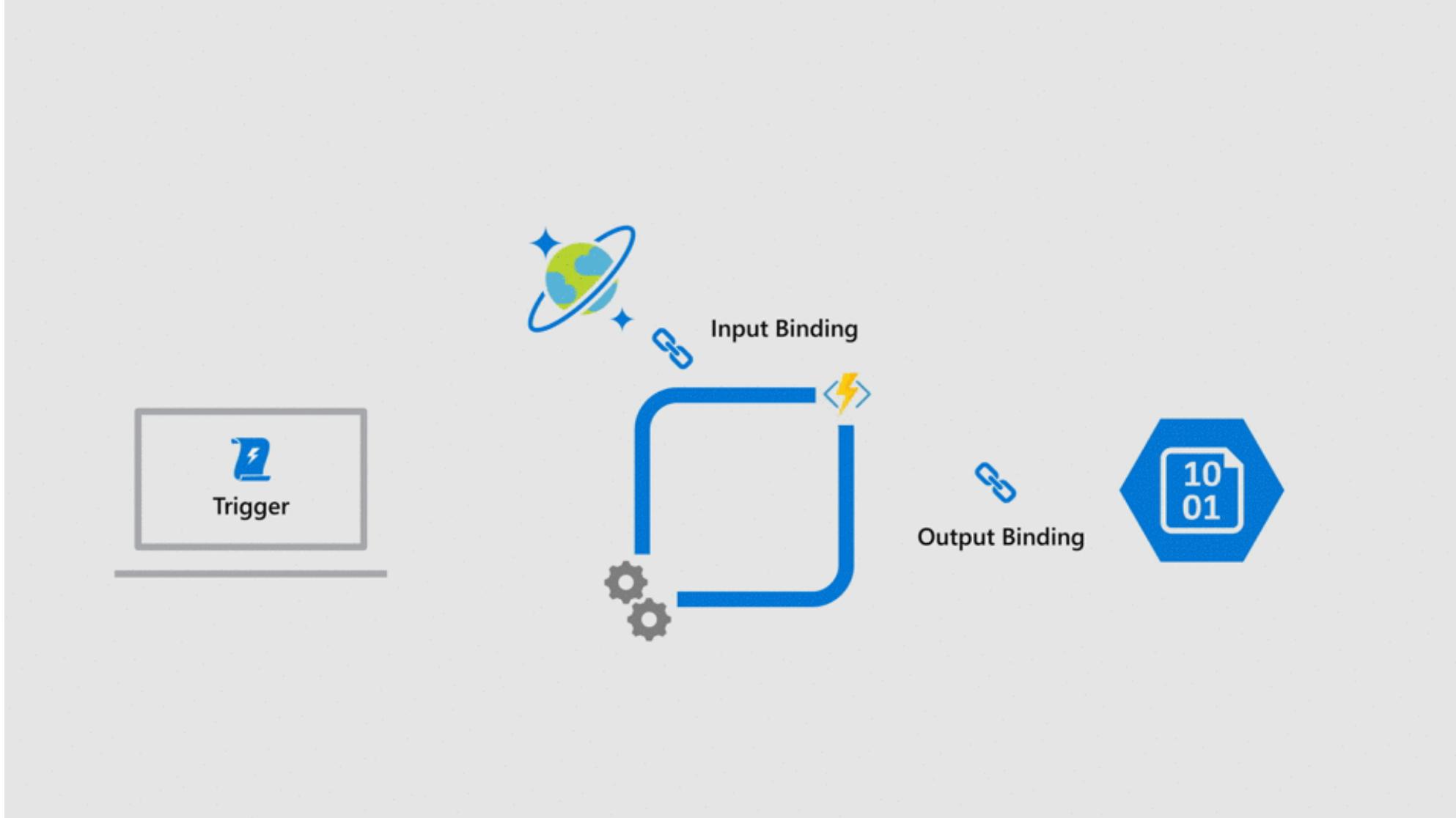


Azure Functions is a *serverless* compute service that enables you to run code on-demand without having to explicitly provision or manage infrastructure. Use Azure Functions to run a script or piece of code in response to a variety of events.

<https://docs.microsoft.com/en-us/azure/azure-functions>

Serverless is the abstraction of servers, it is computing in a cloud execution model where you dynamically manage resources and runtime execution rather than on premises capacity

In a nutshell...



Language Support in Visual Studio alone



Triggers & Bindings

Type	1.x	2.x1	Trigger	Input	Output
Blob storage	✓	✓	✓	✓	✓
Cosmos DB	✓	✓	✓	✓	✓
Event Grid	✓	✓	✓		
Event Hubs	✓	✓	✓		✓
HTTP & webhooks	✓	✓	✓		✓
IoT Hub	✓	✓	✓		✓
Microsoft Graph					
Excel tables		✓		✓	✓
Microsoft Graph					
OneDrive files		✓		✓	✓
Microsoft Graph					
Outlook email		✓			✓
Microsoft Graph					
events		✓	✓	✓	✓
Microsoft Graph					
Auth tokens		✓		✓	
Mobile Apps	✓			✓	✓
Notification Hub	✓				✓
Queue storage	✓	✓	✓		✓
SendGrid	✓	✓			✓
Service Bus	✓	✓	✓		✓
SignalR		✓		✓	✓
Table storage	✓	✓		✓	✓
Timer	✓	✓	✓		
Twilio	✓	✓			✓

How to find other Triggers

<https://www.serverlesslibrary.net/?type=functionapp>

Microsoft Azure

Filter by

Technology

- Functions 2.x
- Functions 1.x
- Logic Apps
- Blob Storage
- Storage Queue
- Cosmos DB
- Cognitive Services
- Azure Active Directory
- App Service
- Key Vault
- SQL Server
- Service Bus Queue
- Event Grid

Language

- JavaScript
- TypeScript
- Java
- C#
- C# Script
- F#
- Python
- PowerShell

Solution Area

- Web API
- Data Processing
- Integration
- Authentication
- Automation

Azure serverless community library

Search

Displaying 70 results

Sort By

[Read Excel From Blob Storage, Process and Add to Azure Queue.](#)
By: raahmed | 361 downloads | Created on: 1/29/2019 | 2 1
This application takes a specific type of csv file (detailed in the project README), cleans the data and then places the

Tags :

[Tug of War](#)
By: joescars | 228 downloads | Created on: 6/30/2017 | 3 1
This simple demo game is used to teach basic concepts around building Azure Functions with node.js.

Tags :

[Live stream Age and Emotion](#)
By: JimPaine | 103 downloads | Created on: 12/13/2018 | 4 0
An HTML5 page which streams your camera to the Face API for age and emotion, all via a V2 function

Tags :

[Start / Stop VMs on a schedule](#)
By: eamonnoreilly | 101 downloads | Created on: 4/30/2019 | 4 0
Start / Stop virtual machines in the specified resource group, subscription, or by tag on a schedule. You need to assign

Tags :

[Java: Triggers and Binding examples](#)
By: daniel-rocha | 99 downloads | Created on: 1/20/2019 | 1 0
20+ ready-to-use examples of triggers and bindings with Azure Java Functions

Tags :

What is this
Azure Function
you speak of?

How does
Azure Function
Work

Session
Ingredients -
High Level

Demo 1 –
Runtime
Experience

Deconstructing
Demo 1

Ideas of Azure
Functions
with/w-out
SharePoint

Demo 2 –
SharePoint Site
Provisioning
Azure Function

Questions and
Answers

What is an Azure Function? How do you define it?



- Write in the language you know: You know C#, do C#, you know JavaScript do your JavaScript, & more F#, Python, PHP, etc
- Consumption model or App Service Plan model
- Use your own libraries i.e. NuGet, NPM, or upload your own DLL
- Backed by OAuth providers such as AAD, and other Social IDP
- Integrate with other SaaS, Develop simply in the Azure IDE or use full SDLC with CI/CD DevOps

 HTTP trigger A function that will be run whenever it receives an HTTP request, responding based on data in the body or query string C# F# JavaScript	 Queue trigger A function that will be run whenever a message is added to a specified Azure Storage queue C# F# JavaScript	 Event Grid trigger A function that will be run whenever an event grid receives a new event C# JavaScript	 GitHub webhook A function that will be run whenever it receives a GitHub webhook C# F# JavaScript
 Timer trigger A function that will be run on a specified schedule C# F# JavaScript	 Blob trigger A function that will be run whenever a blob is added to a specified container C# F# JavaScript	 Face locator A function that processes images and outputs the bounding rectangle of faces using Cognitive Services C# F# JavaScript	 Application Insights Power BI A function that pushes real-time availability data from Application Insights to Power BI C#

1. **HTTP Trigger** - Trigger the execution of your code by using an HTTP request.
2. **Timer Trigger** - Batch tasks on a predefined schedule, think your old Timer Jobs from SharePoint On Premises days
3. **CosmosDB Trigger** - Act on 'documents' as they are added /updated to collections in Cosmos DB NoSQL database
4. **Blob Trigger** - Really a good catch all for text files as well as binary files (BLOB) ingestion and then react to them
5. **Queue Trigger** - IMO the workhorse of chaining events (functions) together
6. **EventGrid Trigger** - Really acts like Blob Trigger but... supports a subscription-based model for receiving events, which includes filtering.
7. **EventHub Trigger** - Respond to events delivered to an Azure Event Hub. Useful in IOT environments and instrumentation
8. **ServiceBus Queue Trigger** - Connect your code to other Azure services or on-premises services by listening to message queues.
9. **ServiceBus Topic Trigger** - Connect your code to other Azure services or on-premises services by subscribing to topics.

Bindings, what is a binding?

Binding to a function is a way of declaratively connecting another resource to the function; bindings may be connected as input bindings, output bindings, or both. Data from bindings is provided to the function as parameters.

- You can mix and match different bindings to suit your needs
- Bindings are optional and a function might have one or multiple input and/or output bindings
- There are Version 1 and Version 2 Azure Functions Runtime (more on this later) but Bindings are linked to the version of the runtime
- Version 2 however is far superior to Version 1 ~ duh
- As a best practice, secrets and connection strings should be managed using app settings
- When a function is running locally, app setting values come from the local.settings.json file
 - If you publish from Visual Studio to Azure you can set the Publishing Profile to update the App Settings in Azure Functions
 - If you CI/CD from Github or other ALM you may have to use Azure CLI or scripting to update the App Settings

```
{  
  [FunctionName("b_OcrQDataToParsedContact")]  
  0 references | Fabian Williams, 191 days ago | 1 author, 1 change  
  public static async Task ParseBusinessCardContactAsync([QueueTrigger("ocrboxed-bizcard", Connection = "IncomingBussCardConnString")]  
    OcrPayload myQueueItem, ILogger log,  
    [Table("scparsedinfo", Connection = "AzureWebJobsStorage")] IAsyncCollector<FullCardInfoTable> outputTable,  
    [Table("excepentities", Connection = "AzureWebJobsStorage")] IAsyncCollector<RawDocTableInfo> exoutputTable,  
    [Table("fullunkunk", Connection = "AzureWebJobsStorage")] IAsyncCollector<RawDocTableInfo> fukukoutputTable,  
    [Queue("scgreetnewcontact", Connection = "AzureWebJobsStorage")] IAsyncCollector<FullCardInfoTable> sendEmailQueue,  
    [Queue("scnotifyteamschan", Connection = "AzureWebJobsStorage")] IAsyncCollector<FullCardInfoTable> addToTeamsQueue,  
    [Queue("scnewgraphcontact", Connection = "AzureWebJobsStorage")] IAsyncCollector<FullCardInfoTable> addToGraphContactsQueue,  
    [Queue("scnewmailchimpsub", Connection = "AzureWebJobsStorage")] IAsyncCollector<FullCardInfoTable> addToMailChimpQueue,  
    [Queue("scnewsfdcprospect", Connection = "AzureWebJobsStorage")] IAsyncCollector<FullCardInfoTable> addToSFDCQueue)  
}
```

What is this
Azure Function
you speak of?

How does
Azure Function
Work

Session
Ingredients -
High Level

Demo 1 –
Runtime
Experience

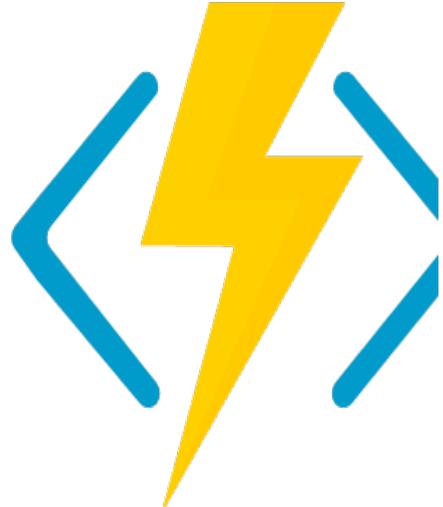
Deconstructing
Demo 1

Ideas of Azure
Functions
with/w-out
SharePoint

Demo 2 –
SharePoint Site
Provisioning
Azure Function

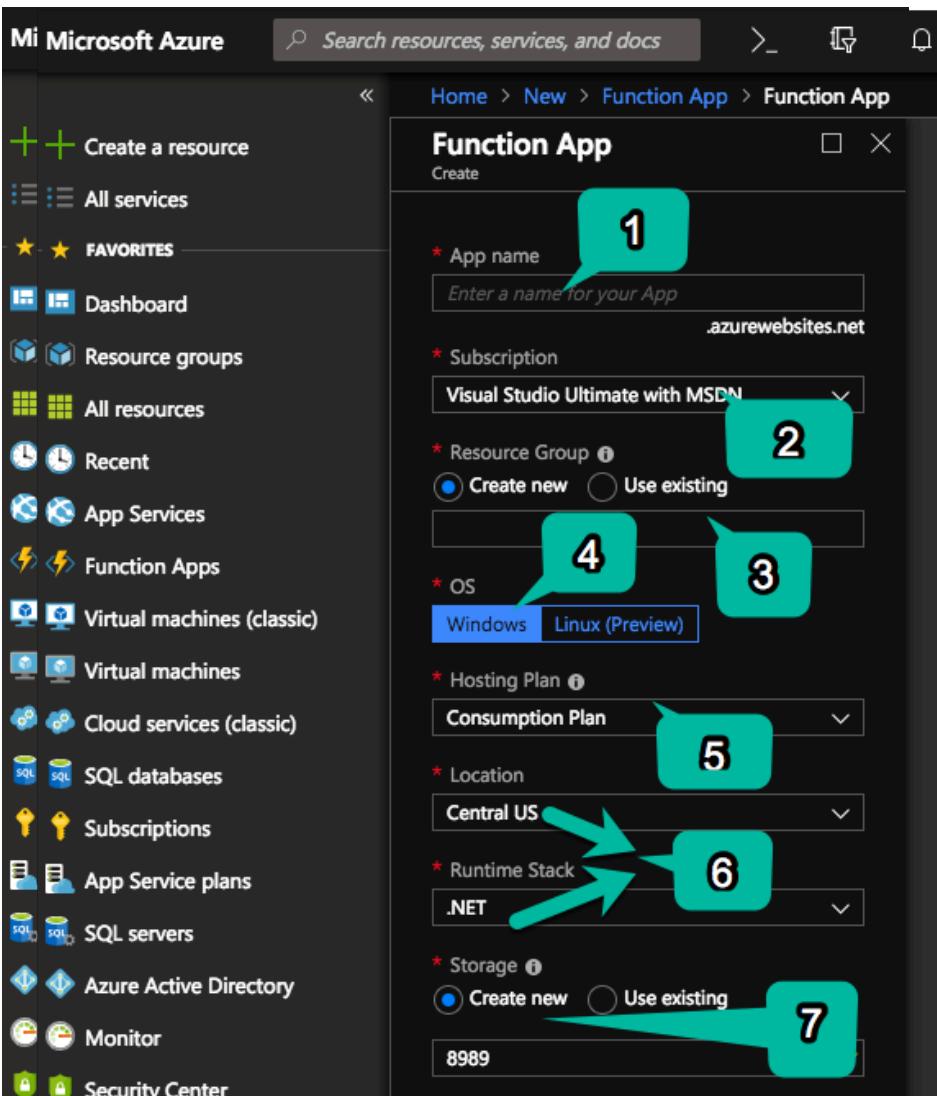
Questions and
Answers

What do you need to do this



1. Set up and create your Azure Function
2. Create a Certificate that you will use for the Authentication in Azure AD. This is the broker that will identify the Azure Function with SharePoint Online
3. Stub out your Azure Function App to get the full URI
4. Register your Application [the Azure Function] in Azure AD
5. Grant the permission to your Application to work within your SharePoint (or other Azure backed workloads) environment
6. Finish Code your solution however you would like to, in the Azure Function IDE, VS, VSCode etc.

What do you need to do this



- Provide a unique name which will become your endpoint URI for your function
- Choose the subscription you want to use
- Choose or create a Resource Group
- I haven't done a Linux OS one yet TBH
- This is the choice between FREE & azure will do its best to stay awake & responsive "or" you allocate resource you will pay for & it will be performant and ready
- Standard stuff
- The function is backed by a storage account so choose one or create one. I typically will have a resource group already set up with storage so I keep everything easy to find and named similar but unique

Making of the Cert

#From Kirk Evans Blog

<https://blogs.msdn.microsoft.com/kaevans/2016/08/12/using-powershell-with-certificates/>

```
$cert = New-SelfSignedCertificate -KeyExportPolicy Exportable `  
-Provider "Microsoft Strong Cryptographic Provider" `  
-Subject "CN=FabianSPOOfficeFiles" `  
-NotBefore (Get-Date) `  
-NotAfter (Get-Date).AddYears(2) `  
-CertStoreLocation "cert:\CurrentUser\My" `  
-KeyLength 2048
```

```
Export-Certificate -Type CERT -Cert $cert -FilePath "C:\1fabsCert\FabsWillyPrivateCertDemo1.cer"
```

```
$cred = Get-Credential
```

```
Export-PfxCertificate -Cert $cert -Password $cred.Password -FilePath  
"C:\1fabsCert\FabsWillyPrivateCertDemo1.pfx"
```

Read back information from the Cert

```
Export-Certificate -Type CERT -Cert $cert -FilePath "C:\1fabsCert\FabsWillyPrivateCertDemo1.cer"
```

```
$cred = Get-Credential
```

```
Export-PfxCertificate -Cert $cert -Password $cred.Password -FilePath  
"C:\1fabsCert\FabsWillyPrivateCertDemo1.pfx"
```

```
$fabswillycer = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2
```

```
$fabswillycer.Import("C:\1fabsCert\FabsWillyPrivateCertDemo1.cer")
```

```
$bin = $fabswillycer.GetRawCertData()
```

```
echo $bin
```

```
$base64Value = [System.Convert]::ToBase64String($bin)
```

```
echo $base64Value
```

```
$bin = $fabswillycer.GetCertHash()
```

```
$base64Thumbprint = [System.Convert]::ToBase64String($bin)
```

```
echo $base64Thumbprint
```

```
$keyid = [System.Guid]::NewGuid().ToString()
```

```
echo $keyid
```

```
$startDate = $($fabswillycer.NotAfter.ToString("s"))
```

```
echo $startDate
```

Uploading / Working with the Certificate

A few options: PowerShell, Portal, Using Copy to Output from your Visual Studio Project

The screenshot shows the Azure Functions portal interface. On the left, the navigation bar includes 'Dashboard', 'All subscriptions', 'Function Apps', 'affspv1fabs', 'ContactOnboarding', and 'csomfun'. Under 'csomfun', there are 'Functions' and several function names: 'a_DisplaySPOOfficeFiles', 'b_SubWebCreationUtil', 'c_CreateO365Contact', 'SPFestDC19Demo1Timer', and 'zz_KeepMeRunningLonger'. The 'SSL' link in the sidebar is highlighted with a green arrow.

In the center, the 'Networking' section shows the 'SSL' configuration for the 'csomfun' function. It lists 'Bindings', 'Private Certificates (.pfx)', and 'Public Certificates (.cer)'. The 'Private Certificates (.pfx)' tab is selected. Below it, a 'Private Certificate' section provides instructions on how to use private certificates for SSL bindings. It includes links for 'Import App Service Certificate', 'Upload Certificate', and 'Import KeyVault Certificate'.

On the right, a code editor displays a C# snippet for handling certificates:

```
1  using Microsoft.IdentityModel.Clients.ActiveDirectory;
2  using Microsoft.SharePoint.Client;
3  using System.Security.Cryptography.X509Certificates;
4
5  public static class csomHelper {
6
7      private static string ClientId = "obs-ured-for-demo-dc82469b62f7";
8      private static string Cert = "FabianWilliamsPrivateCert.pfx"; // Fill in name of your cert
9      private static string CertPassword = "REDACTED"; // TODO: Explore more secure place for this
10     private static string Authority = "https://login.windows.net/fabswilly.onmicrosoft.com/";
11     private static string Resource = "https://fabswilly.sharepoint.com/";
12
13     public async static Task<ClientContext> GetClientContext(string siteUrl)
14     {
15         var authenticationContext = new AuthenticationContext(Authority, false);
16
17         // TODO: Substitute your Azure function name for GetDocUrl2 below:
18         var certPath = Path.Combine(Environment.GetEnvironmentVariable("HOME"), "site\\wwwroot\\shared\\", Cert);
19         var cert = new X509Certificate2(System.IO.File.ReadAllBytes(certPath),
20             CertPassword,
21             X509KeyStorageFlags.Exportable |
22             X509KeyStorageFlags.MachineKeySet |
23             X509KeyStorageFlags.PersistKeySet);
24
25         ClientContext clientContext = new ClientContext(siteUrl);
26         clientContext.AuthenticationMode = ClientAuthenticationMode.Windows;
27         clientContext.ClientCertificates.Add(cert);
28
29         return clientContext;
30     }
31 }
```

A green arrow points to the final part of the code snippet where the certificate path and password are defined.

A Quick Aside on Managing The Certificate Process

Public Side – Azure AD

Home > JahMekYan Enterprises - App registrations > AzureFunctCsomCreateWeb - Certificates & secrets

AzureFunctCsomCreateWeb - Certificates & secrets

Overview Quickstart Manage Branding Authentication Certificates & secrets API permissions Expose an API Owners Manifest Support + Troubleshooting Troubleshooting New support request

1

Credentials enable applications to identify themselves to the authentication service receiving tokens at a web addressable location (using an HTTPS scheme). For a of assurance, we recommend using a certificate (instead of a client secret) as a

2

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

3

Upload certificate THUMPRINT START DATE EXPIRES FE9BD69582C675F4CF0BDE7... 9/21/2018 9/21/2020

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

4

New client secret DESCRIPTION EXPIRES VALUE

No client secrets have been created for this application.

Certificates & secrets

Upload certificate

Upload a certificate (public key) with one of the following file types: .cer, .pem, .crt

Select a file

Add Cancel

Dashboard > csomfun

csomfun

Function Apps

- affspv1fabs
- ContactOnboarding
- csomfun**
- Functions
- a_DisplaySPOOfficeFiles
- b_SubWebCreationUtil
- c_CreateO365Contact
- SPFestDC19Demo1Timer
- zz_KeepMeRunningLonger

Networking

- Networking
- SSL**
- Custom domains
- Authentication / Authorization
- Identity
- Push notifications

Monitoring

Dashboard > csomfun > SSL

SSL

csomfun

Refresh FAQs Delete bindings

Bindings Private Certificates (.pfx) Public Certificates (.cer)

5

PFX Private Certificate

Private certificates (.pfx) can be used for SSL bindings and can be loaded to the certificate store for your app to consume. To understand how to load the certificates for your app to consume click the learn more link. Uploaded certificates are not available for manual download from the Azure Management Portal, they can only be used by your app hosted on App Service after the required App Settings are set properly or used for SSL. Learn more

Import App Service Certificate Upload Certificate Import KeyVault Certificate

Private Certificates

Status Filter All Healthy Warning Expired

HEALTH ...	HOSTNAME	EXPIRATION...	THUMPRINT
He...	DisplaySPOOfficeFiles	9/21/2...	FE9BD69582C675F4CF0BDE7...

Register your Application (callback to Azure Function App)

The screenshot illustrates the steps to register a new application in the Microsoft Azure portal:

- Step 1:** In the left sidebar, under the "Azure Active Directory" section, the "App registrations" option is selected (indicated by a green speech bubble with the number 1).
- Step 2:** In the main content area, the "App registrations" page is displayed. The "Manage" section shows options like "Users", "Groups", "Organizational relationships", "Roles and administrators", "Enterprise applications", "Devices", and "App registrations". The "App registrations" option is highlighted (indicated by a green speech bubble with the number 2).
- Step 3:** A green speech bubble with the number 3 points to the "New application registration" button at the top of the "JahMekYan Enterprises - App registrations" page.

The "Create" dialog box is open on the right side of the screen, showing the configuration for a new application:

- Name:** DemoWareOne
- Application type:** Web app / API
- Sign-on URL:** kz8e1Op7HoGf4bMLB0O0fqZChdrUiZLN ... (with a green checkmark)

A blue "Create" button is visible at the bottom of the dialog.

Set the Required Permissions on the App

Microsoft Azure

Search resources, services, and docs

Home > JahMekYan Enterprises - App registrations > AzureFunctCsm1 > Settings > Required permissions

AzureFunctCsm1
Registered app

Settings Manifest Delete

Display name AzureFunctCsm1

Application type Web app / API

Home page https://ewebsites.net/api/a...
ewebsites.net/api/a...

Required permissions

Add + Add Grant permissions

API	APPLICATION PERMI...	DELEGATED PERMI...
Office 365 SharePoint Online	2	0
Windows Azure Active Directory	0	1

Settings Required permissions

Enterprises - App registrations > AzureFunctCsm1 > Settings > Required permissions > Enable Access

Enable Access Office 365 SharePoint Online

Save Delete

APPLICATION PERMISSIONS

REQUIRES ADMIN
2

Read user profiles Yes

Read and write user profiles Yes

Read and write managed metadata Yes

Read managed metadata Yes

Read and write items and lists in all site collections Yes

Have full control of all site collections Yes

Read items in all site collections Yes

Read and write items in all site collections Yes

DELEGATED PERMISSIONS

REQUIRES ADMIN
No

Read user profiles Yes

Read and write user profiles Yes

Read and write user files No

Read user files No

Have full control of all site collections Yes

Read and write items and lists in all site collections No

This screenshot shows the 'Required permissions' section for an Azure app registration named 'AzureFunctCsm1'. On the left, there's a sidebar with various icons. The main area shows the app's details: Display name 'AzureFunctCsm1', Application type 'Web app / API', and Home page 'https://ewebsites.net/api/a...'. Below this is a table for 'Required permissions' with two rows: 'Office 365 SharePoint Online' (2 application, 0 delegated) and 'Windows Azure Active Directory' (0 application, 1 delegated). To the right, under 'Settings' and 'Required permissions', is a detailed view for 'Enable Access' on 'Office 365 SharePoint Online'. It lists various permissions under 'APPLICATION PERMISSIONS' (e.g., Read user profiles, Have full control of all site collections) and 'DELEGATED PERMISSIONS' (e.g., Read user files, Have full control of all site collections). Most permissions require admin consent ('Yes'), except for one in each category which requires no admin consent ('No').

csomfun - b_SubWebCreationUtil

Function Apps

 Search

All subscriptions

Function Apps

▶ <⚡ affspv1fabs

▼ csomfun

Functions

4 Integrate

 Manage

Q Monitor

▼ f b SubWebCreationUtil

Integrate

 Manage

Q Monitor

▶ Proxies

▶ Slots (preview)

▶  fabhw

▶ fabSFIAIgnite2017K2

▶ fabsfunctalpha

run.csx

Save

Cance

csomHelper.csx

What is this
Azure Function
you speak of?

How does
Azure Function
Work

Session
Ingredients -
High Level

Demo 1 –
Runtime
Experience

Deconstructing
Demo 1

Ideas of Azure
Functions
with/w-out
SharePoint

Demo 2 –
SharePoint Site
Provisioning
Azure Function

Questions and
Answers

DEMO

1. Azure Functions CSOM to return and manipulate
Excel Online Data

Azure Functions v1 with ADAL and Certificate Authentication

Demo 1 Leave Behind

The screenshot shows the Postman application interface. On the left, there's a sidebar titled "Collections" containing several items: "AzureFunctions and CSOM" (2 requests), "CreateSharePointWeb" (1 request), "DocumentDB copy" (16 requests), "FabianPlayPen" (1 request), "FDA1" (9 requests), "Postman Echo" (21 requests), and "WithumPEAssurance" (20 requests). The main workspace is titled "My Workspace" and has a search bar and various icons at the top. A collection named "Find SPO File via WOPI" is selected. Below it, a POST request is configured with the URL https://csmofun.azurewebsites.net/api/a_DisplaySPOOfficeFile. The "Body" tab is active, showing a JSON payload:

```
1 {  
2   "docLibandPath": "/Shared%20Documents/",  
3   "siteUrl": "https://fabswilly.sharepoint.com"  
4 }
```

The "Body" tab also includes options for "form-data", "x-www-form-urlencoded", "raw", and "binary". The "Headers" tab shows 10 headers. At the bottom, the response status is listed as "Status: 200 OK Time: 4349 ms". The "Body" tab has sub-options for "Pretty", "Raw", "Preview", and "JSON".

Demo 1 Leave Behind

The screenshot shows a Microsoft Edge browser window with three tabs open:

- App registrations - Microsoft Azure (active tab)
- Site Contents
- Site Contents

The address bar shows a SharePoint URL: [https://fabswilly.sharepoint.com/:x/r/_layouts/15/Doc.aspx?sourcedoc={c0c...](https://fabswilly.sharepoint.com/:x/r/_layouts/15/Doc.aspx?sourcedoc={c0c...})

The main content area displays an Excel Online spreadsheet titled "SharegateMetalogixComparison". The spreadsheet has three columns labeled A, B, and C, and three rows labeled 1, 2, and 3.

A	B	C
1 Topic	Metalogix	Sharegate
2	Diagnostic Manager	Standard Migration
3	Content Matrix	Nintex Migration

The first row (Topic) is highlighted in blue. The second row (Diagnostic Manager) is white. The third row (Content Matrix) is also blue.

What is this
Azure Function
you speak of?

How does
Azure Function
Work

Session
Ingredients -
High Level

Demo 1 –
Runtime
Experience

Deconstructing
Demo 1

Ideas of Azure
Functions
with/w-out
SharePoint

Demo 2 –
SharePoint Site
Provisioning
Azure Function

Questions and
Answers

Questions you may have

- What is `csomHelper.csx` and why do you need it?
 - Answer in session
- What are the use cases for a function like this?
 - Think mobile app or web app something where SPO is an external resource
- Oversimplified?
 - Yes. But it illustrates the point of making CSOM Calls..
The next demo we will be creating a Web Site so that's a bit more meaningful & we will use a complex object

What is this
Azure Function
you speak of?

How does
Azure Function
Work

Session
Ingredients -
High Level

Demo 1 –
Runtime
Experience

Deconstructing
Demo 1

Ideas of Azure
Functions
with/w-out
SharePoint

Demo 2 –
SharePoint Site
Provisioning
Azure Function

Questions and
Answers

What else can I do? What is available to me?

- Lets discuss Advanced Kudu Options and the other Areas I have not even touched yet
- Lets discuss Version 2 & 3 of Azure Functions and using proper Development Tooling such in Visual Studio (you can also use VS Code)

What is this
Azure Function
you speak of?

How does
Azure Function
Work

Session
Ingredients -
High Level

Demo 1 –
Runtime
Experience

Deconstructing
Demo 1

Ideas of Azure
Functions
with/w-out
SharePoint

Demo 2 –
SharePoint Site
Provisioning
Azure Function

Questions and
Answers

Demo 2 – Leave behind

The screenshot shows the Postman application interface. The top navigation bar includes File, Edit, View, Window, Help, and a status bar indicating 100% battery, Thu Oct 18 16 53 58. The main header says "Postman" with a "My Workspace" dropdown and various icons.

The left sidebar lists collections: "History" (selected), "Collections" (highlighted in red), "AzureFunctions and CSOM" (2 requests), "CreateSharePointWeb" (1 request), "DocumentDB copy" (16 requests), "FabianPlayPen" (1 request), "FDA1" (9 requests), "Postman Echo" (21 requests), and "WithumPEAssurance" (20 requests).

The central workspace displays a collection named "Create SPO Site via AzFunct CSOM". A POST request is configured with the URL https://csomfun.azurewebsites.net/api/b_SubWebCreationUtil?code=TI3ss. The "Body" tab is selected, showing JSON (application/json) content:

```
1 {  
2   "scUrl" : "https://fabswilly.sharepoint.com",  
3   "title" : "PreSPSBMore Demo Code",  
4   "description" : "fabtester SPSBmore1 from Chrome Postman ",  
5   "url" : "prespsbmore1",  
6   "template" : "STS#0"  
7 }
```

The response section shows a successful 200 OK status with a time of 9529 ms and a size of 182 B. The "Pretty" tab is selected in the response view.

Demo 2 – Leave behind

The screenshot shows a SharePoint site named "JahMekYan Enterprises Team Site". The left navigation bar includes links for Home, Notebook, Documents, Pages, Site contents (which is selected), and Recycle bin. The main content area displays the "Site Contents" page for the "Subsites" tab. It lists two items:

Name	Description	Views	Created
PreSPSBMore Demo Code	fabtester SPSbmore1 from Chrome Postman		21 minutes ago
SPC 19 Pre Demo Code	fabtester1 from Chrome Postman		26 days ago

Two large green arrows point to the "Name" and "Description" columns of the first row.

DEMO

2. Auto Provisioning a SharePoint Site using Azure Functions

Use Case is a Tweet, a Slack Message, a MS Flow, tons of reason you want outside influence to manipulate internal protected resources

What is this
Azure Function
you speak of?

How does
Azure Function
Work

Session
Ingredients -
High Level

Demo 1 –
Runtime
Experience

Deconstructing
Demo 1

Ideas of Azure
Functions
with/w-out
SharePoint

Demo 2 –
SharePoint Site
Provisioning
Azure Function

Questions and
Answers

Demo Leave Behind & Explanatory Notes

The screenshot shows a Microsoft Visual Studio interface with the following components:

- Code Editor:** Displays C# code for a project named "affun". The code includes logic for handling Azure Functions triggers (QueueTrigger, FunctionName) to process contact queue items and send emails via SendGrid.
- Solution Explorer:** Shows the project structure. Under "Dependencies", it lists NuGet packages such as DeveloperForce.Force, Microsoft.Azure.WebJobs.Extensions.CosmosDB, Microsoft.Azure.WebJobs.Extensions.SendGrid, Microsoft.Azure.WebJobs.Extensions.Storage, Microsoft.Extensions.Configuration, Microsoft.Extensions.Configuration.FileExtensions, Microsoft.Extensions.Configuration.Json, Microsoft.Graph, Microsoft.Identity.Client, and Microsoft.NET.Sdk.Functions. It also shows local files like .gitignore, host.json, and local.settings.json.
- SQL API Results:** A table titled "FabianSessions" with columns "id" and "/id". It contains two rows of data:

	id	/id
1	338efa3c...	338efa3c...
2	9f278681...	9f278681...

Below the table, there are links for "Scale & Settings", "Stored Procedures", "User Defined Functions", and "Triggers".
- Properties View:** Shows the properties for the selected "FabianSessions" table.

The code in the editor is as follows:

```
20 0 references | 0 changes | 0 authors, 0 changes
21 public static class OnboardContact
22 {
23     0 references | 0 changes | 0 authors, 0 changes
24     public static object email { get; private set; }
25     [FunctionName("EmailNewContact")]
26     0 references | 0 changes | 0 authors, 0 changes
27     public static void EmailNewContact(
28         [QueueTrigger("fabsqueue-sendgridemail", Connection = "AzureWebJobsStorage")] Contact myQueueItem,
29         TraceWriter log,
30         [SendGrid()] out SendGridMessage message)
31     {
32         string emailBody = File.ReadAllText(@"D:\home\site\wwwroot\emailBody.txt");
33         string b64vCard = Environment.GetEnvironmentVariable("Fab64EncodedVCard");
34         message = new SendGridMessage();
35         message.AddTo(myQueueItem.EmailAddress);
36         message.AddContent("text/html", emailBody);
37         message.SetFrom(new EmailAddress("fabian@adotob.com"));
38         message.SetSubject($"Great connecting with you {myQueueItem.FirstName} from Fabian Williams ");
39         message.AddAttachment(@"FabianWilliams_PersonalvCard.vcf", b64vCard);
40     }
41
42     private static Lazy<HttpClient> HttpClient = new Lazy<HttpClient>(() => new HttpClient());
43
44     [FunctionName("ProcessMSTeamsQueue")]
45     0 references | 0 changes | 0 authors, 0 changes
46     public static async Task ProcessMSTeamsQueue(
47         [QueueTrigger("fabsqueue-msteamscard")] string myQueueItem,
48         ILogger log)
49     {
50         var data = JsonConvert.DeserializeObject<Contact>(myQueueItem);
51
52         string WebhookUrl = Environment.GetEnvironmentVariable("FWorldNewContactTeamWebHook");
53         log.LogInformation("Sending to Microsoft Teams Channel Now");
54         var teamsResult = await HttpClient.Value.PostAsync(WebhookUrl,
55             new StringContent($"{{\"@type\": \"MessageCard\", \"@context\": \"http://schema.org/extensions\"}}"));
56
57         teamsResult.EnsureSuccessStatusCode();
58         log.LogInformation($"Result is {teamsResult.StatusCode}");
59         log.LogInformation($"C# Fabian New Contact Queue Function completed... ({myQueueItem})");
60
61     [FunctionName("CreateNewContact")]
62     [Display(Name = "Create New Contact", Description = "Function creates a New Contact Individual in a Contact Collection")]
63     0 references | 0 changes | 0 authors, 0 changes
64     public static async Task<IActionResult> CreateNewContact([HttpTrigger(AuthorizationLevel.Function, "post", Route = "contact")] HttpRequest req,
65         [CosmosDB(
66             databaseName: "FabianSessions",
67             collectionName: "Contacts",
68             ConnectionStringSetting = "FabSessionCosmosDBConn")] IAsyncCollector<Contact> createSessionOut,
69         TraceWriter log,
70         [Queue("fabsqueue-msteamscard", Connection = "AzureWebJobsStorage")] IAsyncCollector<Contact> outputQueue,
71         [Queue("fabsqueue-sendgridemail", Connection = "AzureWebJobsStorage")] IAsyncCollector<Contact> outputEmailQueue)
```

Demo Leave Behind & Explanatory Notes

Flow

New Contact Intake

Home Approvals My flows Templates Connectors Data Learn

New Contact Intake Add a description

CONNECTIONS See all >

- Microsoft Forms fabian@fabswillyo...
- FabianOnboardC... FabianOnboardCo...

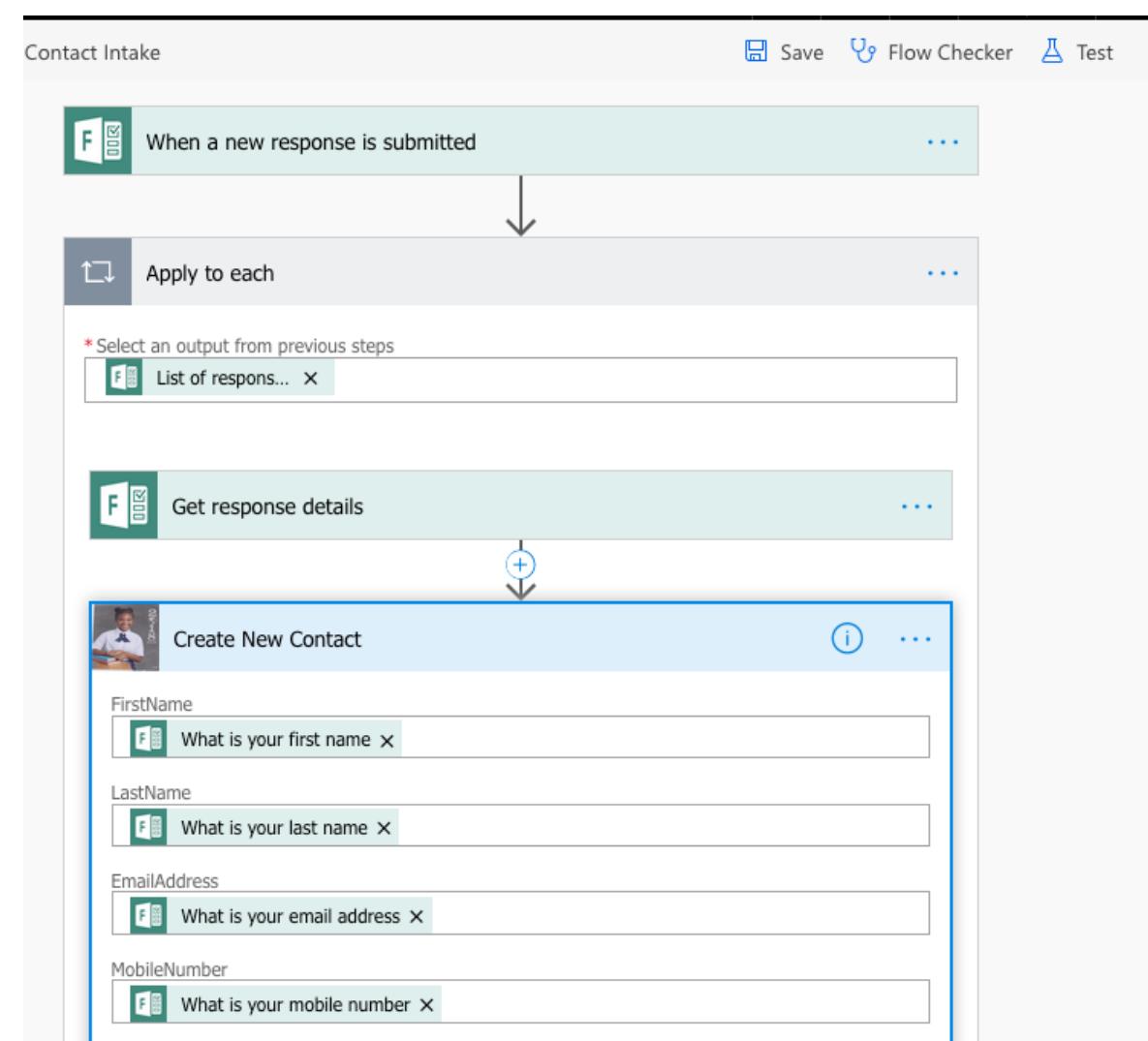
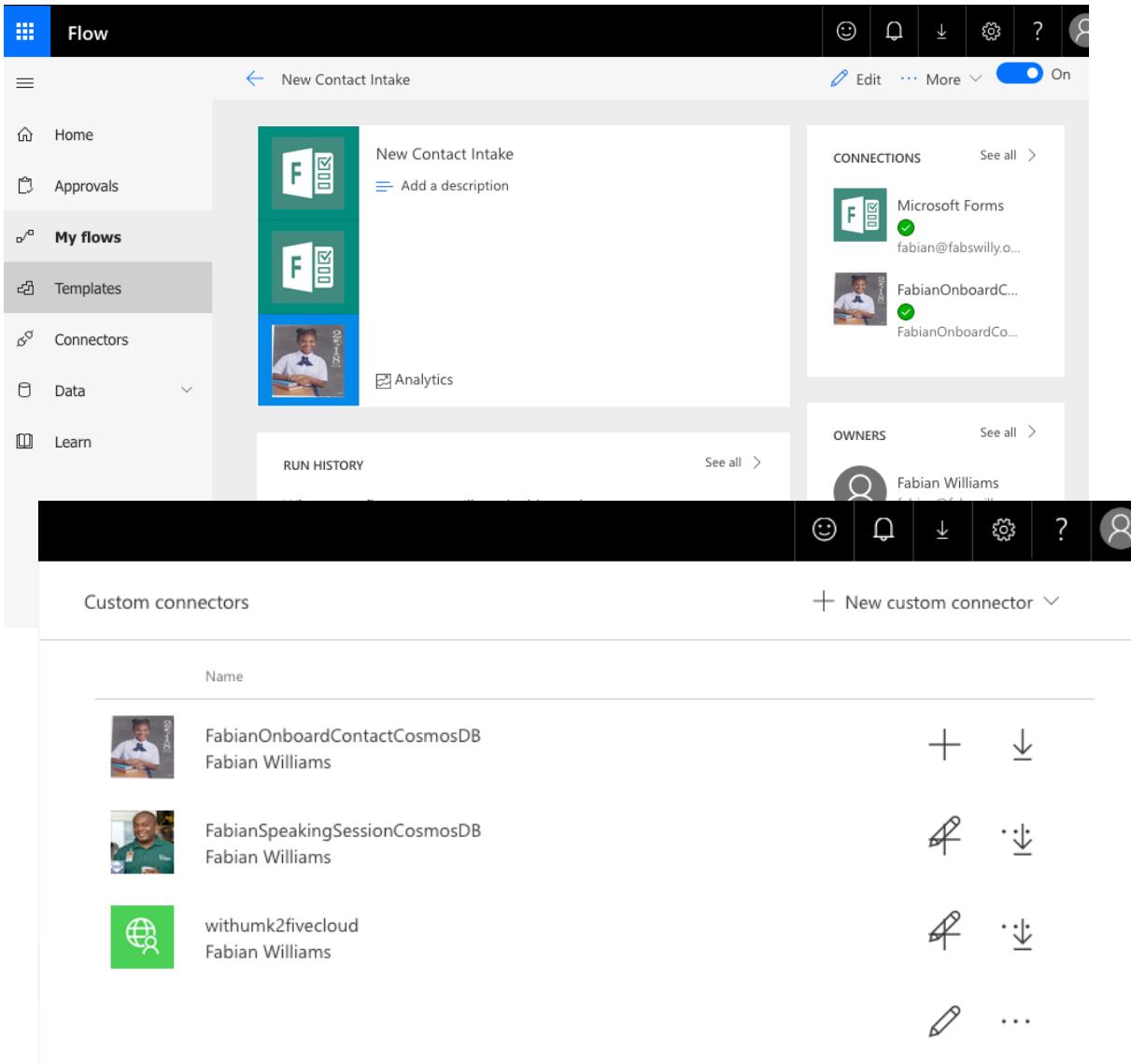
OWNERS See all >

- Fabian Williams

RUN HISTORY See all >

Custom connectors + New custom connector

Name
FabianOnboardContactCosmosDB Fabian Williams
FabianSpeakingSessionCosmosDB Fabian Williams
withumk2fivecloud Fabian Williams



DEMO

3. Creating an Azure Functions and using Open Definition Files (aka Swagger Def Files) and Consuming in Microsoft Flow

This Demo uses Helper Class and Methods from Mikael Svenson to decorate your Azure Functions in order to make them available and meaningful when importing custom connectors into Microsoft Flow

What is this
Azure Function
you speak of?

How does
Azure Function
Work

Session
Ingredients -
High Level

Demo 1 –
Runtime
Experience

Deconstructing
Demo 1

Ideas of Azure
Functions
with/w-out
SharePoint

Demo 2 –
SharePoint Site
Provisioning
Azure Function

Demos 3 –
Azure Function
V2 with Flow

Demo4 – Azure
Functions V2
w/ SFDC &
Mailchimp

Demo 5 Azure
Functions V2
w/ Graph API &
MSAL

Leave Behind,
Notes. How to
do what I just
did!

Questions and
Answers

Demo Leave Behind & Explanatory Notes

The screenshot displays a developer's workspace with several windows open:

- Solution Explorer:** Shows the project structure for "affun". The "3_MailChimpSFDC" folder contains the file "MailChimpSubscriber.cs". Other folders include "Dependencies", "shared", "Utils", ".gitignore", "host.json", and "local".
- Code Editor:** Displays the C# code for the "MailChimpSubscriber" function. The code uses Azure Functions and MailChimp API to subscribe users to a list.
- API Response:** A screenshot of the MailChimp API playground interface. It shows a GET request to "lists/0e7.../members". The response table lists three members with their email addresses and "Subreso" status.

```
15     [FunctionName("MailChimpSubscriber")]
16     public static void Run([QueueTrigger("scnewmailchimpsub", Connection = "AzureWebJobsStorage")]
17                             FullCardInfoTable myQueueItem,
18                             ILogger log)
19     {
20         log.LogInformation($"C# Queue trigger function processed: {myQueueItem}");
21
22         var method = Environment.GetEnvironmentVariable("CartWiselyMailChimpListId");
23         var key = Environment.GetEnvironmentVariable("CartWiselyMailChimpKey");
24
25         var fullname = myQueueItem.Info.Name.ToString();
26         var tmpFullName = fullname.Split(new char[] { ' ' });
27         var fn = tmpFullName[0];
28         var ln = tmpFullName[1];
29
30         MailChimpSub subscribeRequest = new MailChimpSub()
31         {
32             EmailAddress = myQueueItem.Info.Email,
33             Status = "subscribed",
34             MergeFields = new MergeFields()
35             {
36                 Fname = fn,
37                 Lname = ln
38             }
39         };
40
41         string payload = JsonConvert.SerializeObject(subscribeRequest);
42         log.LogInformation($"Working on Payload: {payload}");
43         var endpoint = String.Format("https://{}{0}.api.mailchimp.com/3.0/{1}", "us20", method);
44         byte[] dataStream = Encoding.UTF8.GetBytes(payload);
45         var responseText = string.Empty;
46         WebRequest request = HttpWebRequest.Create(endpoint);
47         WebResponse response = null;
48         try
49         {
50             request.ContentType = "application/json";
51             SetBasicAuthHeader(request, "anystring", key); // BASIC AUTH
52             request.Method = "POST";
53             request.ContentLength = dataStream.Length;
54             Stream newstream = request.GetRequestStream();
55
56             newstream.Write(dataStream, 0, dataStream.Length);
57             newstream.Close();
58
59             response = request.GetResponse();
60
61             // get the result
62             using (StreamReader reader = new StreamReader(response.GetResponseStream()))
63             {
64                 JsonSerializer json = new JsonSerializer();
65                 JObject content = JObject.Parse(reader.ReadToEnd());
66
67                 responseText = reader.ReadToEnd();
68             }
69         }
70     }
```

DEMO

4. Using Azure Functions to Create a Mail Chimp Subscriber or Salesforce Lead

...

What is this
Azure Function
you speak of?

How does
Azure Function
Work

Session
Ingredients -
High Level

Demo 1 –
Runtime
Experience

Deconstructing
Demo 1

Ideas of Azure
Functions
with/w-out
SharePoint

Demo 2 –
SharePoint Site
Provisioning
Azure Function

Demos 3 –
Azure Function
V2 with Flow

Demo4 – Azure
Functions V2
w/ SFDC &
Mailchimp

Demo 5 Azure
Functions V2
w/ Graph API &
MSAL

Leave Behind,
Notes. How to
do what I just
did!

Questions and
Answers

Demo Leave Behind & Explanatory Notes

A screenshot of the Visual Studio IDE showing a C# code editor and Solution Explorer. The code editor displays a file named `CreateContactviaCertAuth.cs` containing logic to create a new contact in Microsoft Graph using certificate authentication. The Solution Explorer shows a project named `affun` with various files including `AuthHandler.cs`, `AuthHandler.cs`, `Models.cs`, and `MsalAuthenticationProvider.cs`.

```
private static GraphServiceClient _graphServiceClient;

[FunctionName("CreateContactviaCertAuth")]
public static async Task AddNewOffice365Contact([QueueTrigger("scnewgraphcon")]
    FullCardInfoTable myQueueItem,
    ILogger log)

{
    var config = LoadAppSettings();
    if (null == config)...}

    GraphServiceClient graphClient = GetAuthenticatedGraphClient(config);
    try
    {
        if (graphClient != null)
        {
            var fullname = myQueueItem.Info.Name.ToString();
            var tmpFullName = fullname.Split(new char[] { ' ' });
            var fn = tmpFullName[0];
            var ln = tmpFullName[1];
            string newLeadRaw = JsonConvert.SerializeObject(myQueueItem);

            var businessPhonesList = new List<String>();
            businessPhonesList.Add(myQueueItem.Info.Phone?.ToString());
            var emailAddresses = new EmailAddress
            {
                Address = myQueueItem.Info.Email?.ToString(),
                Name = myQueueItem.Info.Name?.ToString(),
            };
            var bizAddresses = new PhysicalAddress
            {
                City = "ToDo Item",
                CountryOrRegion = "US",
                PostalCode = "0000",
                State = "Md",
                Street = "ToDoItem"
            };
            var emailAddressesList = new List<EmailAddress>();
            emailAddressesList.Add(emailAddresses);

            var contact = new Contact
            {
                CompanyName = myQueueItem.Info.Company?.ToString(),
                //BusinessAddress = bizAddresses,
                JobTitle = myQueueItem.Info.Title?.ToString(),
                //Manager = "Dan Holme",
                //NickName = "Bill",
                GivenName = fn,
                Surname = ln,
                PersonalNotes = newLeadRaw,
                EmailAddresses = emailAddressesList,
                BusinessPhones = businessPhonesList,
            };
            //graphClient.Me.Contacts
            var NewlyCreatedContact = await graphClient.Users["65926cde-1721-48d0-8f3e-1a1a1a1a1a1a"].Request()
        }
    }
```

A screenshot of the Microsoft 365 Contacts page in a web browser. A contact named "Rod Donovan" (represented by a red circular icon with "RD") has been created. The contact information includes an email address (`rdonovan@affun.com`) and a work phone number (`+1 202 555-5525`). The "Contact" tab is selected, showing the contact's details.

Contact information:

- Email: rdonovan@affun.com
- Work phone: +1 202 555-5525

Notes:

```
{"PartitionKey":"SmartCards","RowKey":"6b2eccb3-0865-4540-8722bcfbfa5da0","CardOwnerEmailId":"fabian@affun.com","Info":{"C":0,"Title":null,"Address":105 Avenue, Suite 404,"CityState":08837,"Email":rdonovan@affun.com,"Website":www.affun.com,"Latitude":38.8425,"Longitude":-77.2555,"Fax":null,"Cell":1234567890,"Total":null,"Category":4}}
```

DEMO

5. Creating a Contact in Office 365 using the Microsoft Graph API and Azure Functions

Use Case is imaging you have contact information and you want to ensure that it is captured in the corporate system, using this function ensures that it will