Azure Functions

Lab 14

by

Andrea Hatch, Nishava Inc.

Deep Azure @McKesson

Overview

I. Set-up all Prerequisites
II. Create your First Function using Azure
III. Create your First Function using VS

Objective of Demos

- Demo 1: using the Azure portal
  - Create a Hello world function using only the Azure Portal
- Demo 2: using Visual Studio
  - Create a Hello world function using Visual Studio and then publishing to Azure

My Environment

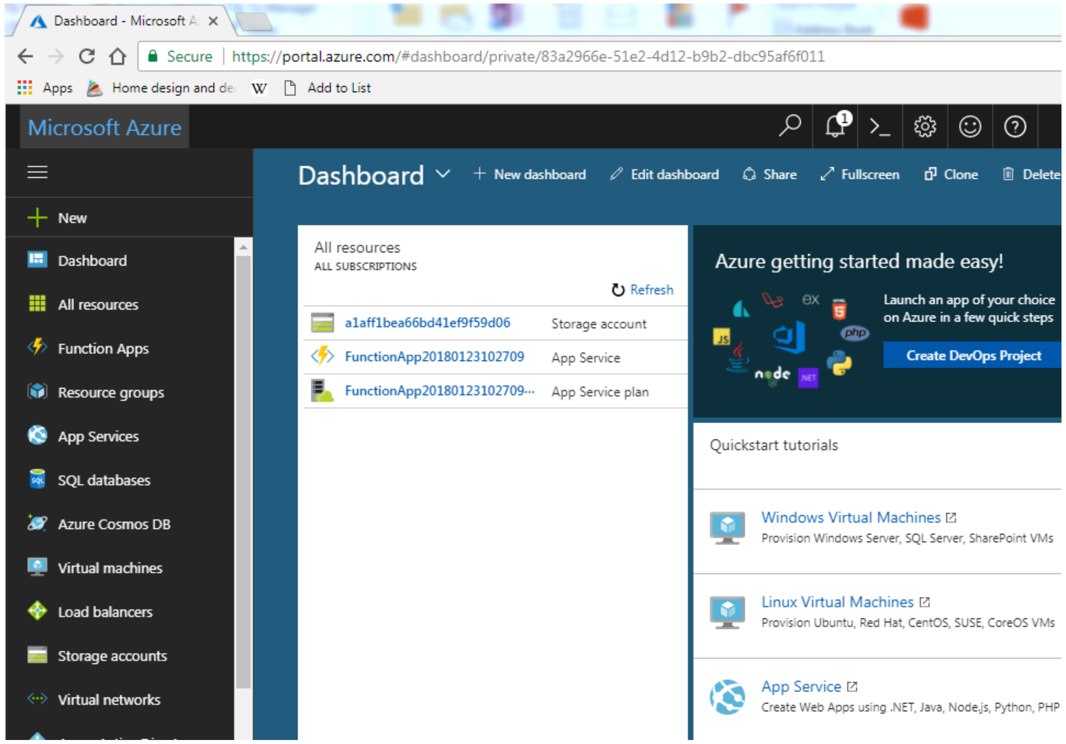
- Windows 7
- Visual Studio 2017 version 15.5
- Azure (free trial)

Visual Studio
Community 2017

I. Prerequisites

Prerequisites

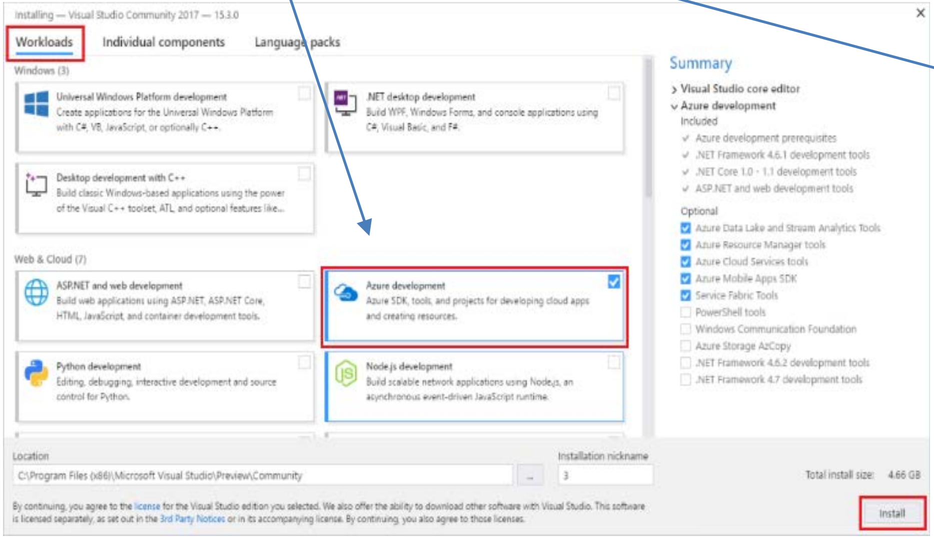
- You will need to have the following before working with Functions in Azure:
  - Azure account



a1aff1bea66bd41ef9f59d06	Storage account
FunctionApp20180123102709	App Service
FunctionApp20180123102709...	App Service plan

Prerequisites

- You need the following before working with Functions in VS:
  1. Visual Studio version 15.4 (released in November 2017) or higher
     - In VS select Tools -> Extensions and Updates -> Updates -> Product Updates
  2. Azure Development Workload
     - In VS select Tools -> Get Tools and Features -> Azure Development Workload
  3. Azure Functions and Web Job Tools
     - In VS select Tools-> Extensions and Updates-> Updates -> Visual Studio Marketplace -> Azure Functions and Web Job Tools



Windows (3)

Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, JavaScript, or optionally C++.

.NET desktop development
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F#.

Desktop development with C++
Build classic Windows-based applications using the power of the Visual C++ toolset, ATL, and optional features like...

Web & Cloud (7)

ASP.NET and web development
Build web applications using ASP.NET, ASP.NET Core, HTML, JavaScript, and container development tools.

Azure development ✓
Azure SDK, tools, and projects for developing cloud apps and creating resources.

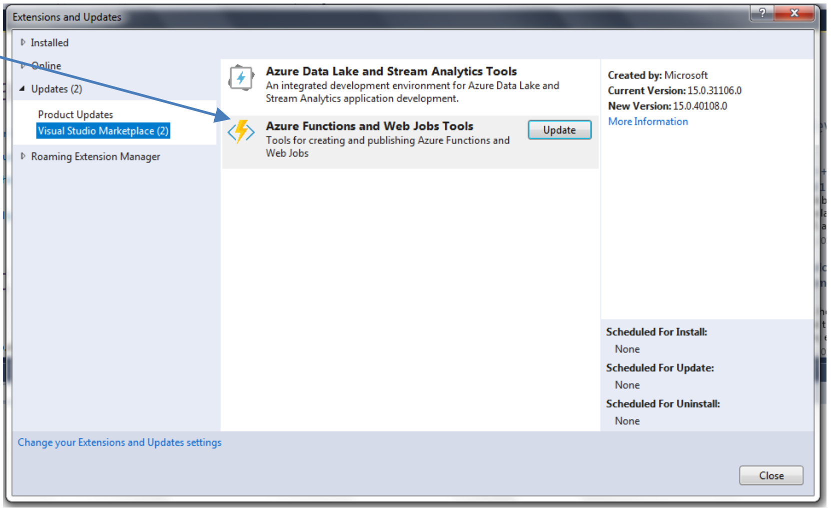
Python development
Editing, debugging, interactive development and source control for Python.

Node.js development
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Summary
- Visual Studio core editor
- Azure development
  - Included
    - ✓ Azure development prerequisites
    - ✓ .NET Framework 4.6.1 development tools
    - ✓ .NET Core 1.0 - 1.1 development tools
    - ✓ ASP.NET and web development tools
  - Optional
    - ☑ Azure Data Lake and Stream Analytics Tools
    - ☑ Azure Resource Manager tools
    - ☑ Azure Cloud Services tools
    - ☑ Azure Mobile Apps SDK
    - ☑ Service Fabric Tools
    - ☐ PowerShell tools
    - ☐ Windows Communication Foundation
    - ☐ Azure Storage AzCopy
    - ☐ .NET Framework 4.6.2 development tools
    - ☐ .NET Framework 4.7 development tools

Location: C:\Program Files (x86)\Microsoft Visual Studio\Preview\Community
Installation nickname: 3
Total install size: 4.66 GB

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.


Extensions and Updates
Installed
Online
Updates (2)
Product Updates
Visual Studio Marketplace (2)
Roaming Extension Manager
Update
More Information
Change your Extensions and Updates settings
Close

Azure Data Lake and Stream Analytics Tools
An integrated development environment for Azure Data Lake and Stream Analytics application development.

Azure Functions and Web Jobs Tools
Tools for creating and publishing Azure Functions and Web Jobs

Created by: Microsoft
Current Version: 15.0.31106.0
New Version: 15.0.40108.0

Scheduled For Install: None
Scheduled For Update: None
Scheduled For Uninstall: None

II. Working with Functions in Azure

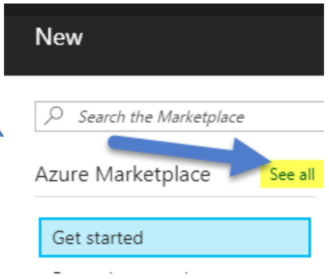
Azure Functions

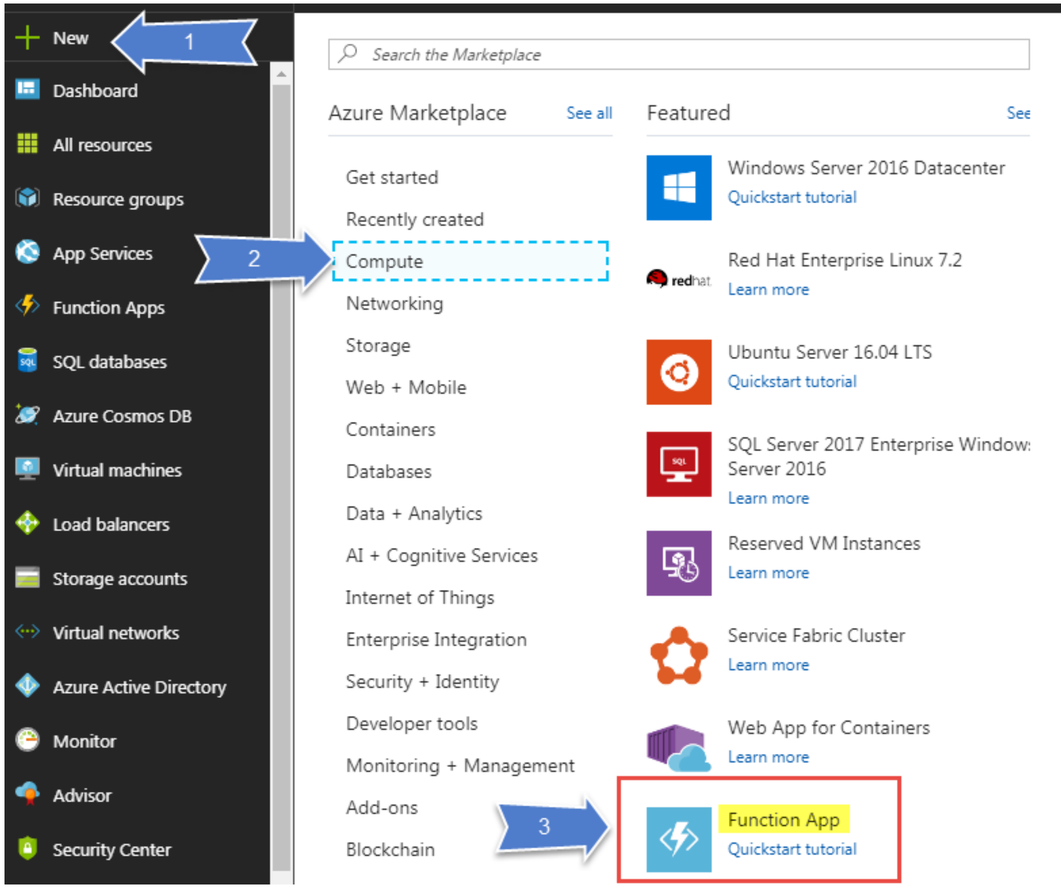
- Serverless architecture
- Event driven
- Used to focus on building your apps and not focus on provisioning and maintaining servers
- You can create functions in any language you choose
- Azure Functions allow you to do the following:
  - Timer-based processing
  - Azure Service event processing
  - SaaS event processing
  - Serverless mobile back ends
  - Real-time stream processing
  - Real-time bot messaging

Source: https://azure.microsoft.com/en-us/services/functions/

Creating a Function App

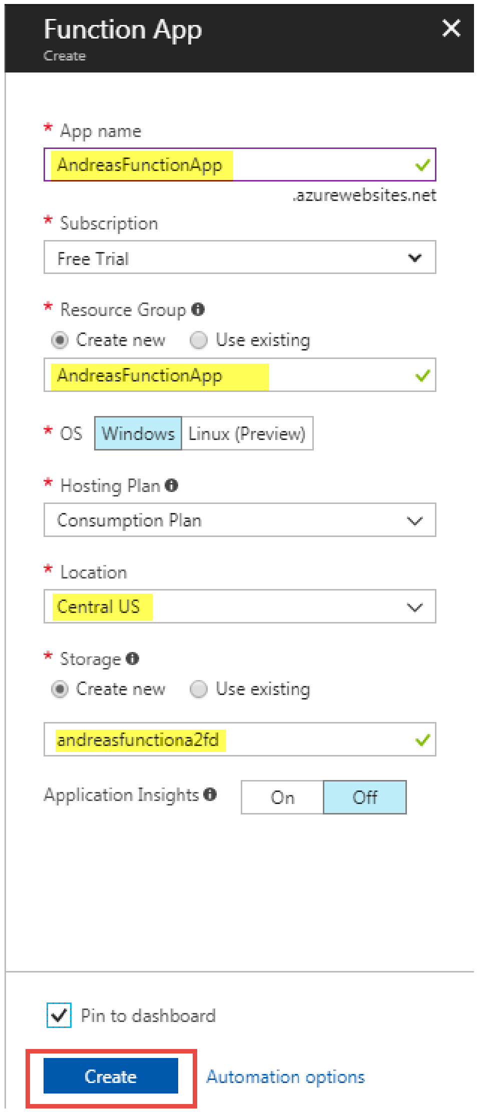
- You need to first create a function app to host your functions
- In Azure, select New -> Compute -> Function App
- Note: If you don't see compute next to Azure Marketplace select See all





Creating a Function App

- You will need to enter a unique name for your Function App
- Subscription: Keep default
- Resource Group: Create new
- Location: Select your location
- Storage Account: Create new
- Select Pin to dashboard
- Select Create



Function App
Create

* App name
AndreasFunctionApp ✓
.azurewebsites.net

* Subscription
Free Trial

* Resource Group
[✓ Create new] Use existing
AndreasFunctionApp ✓

* OS [✓ Windows] Linux (Preview)

* Hosting Plan
Consumption Plan

* Location
Central US

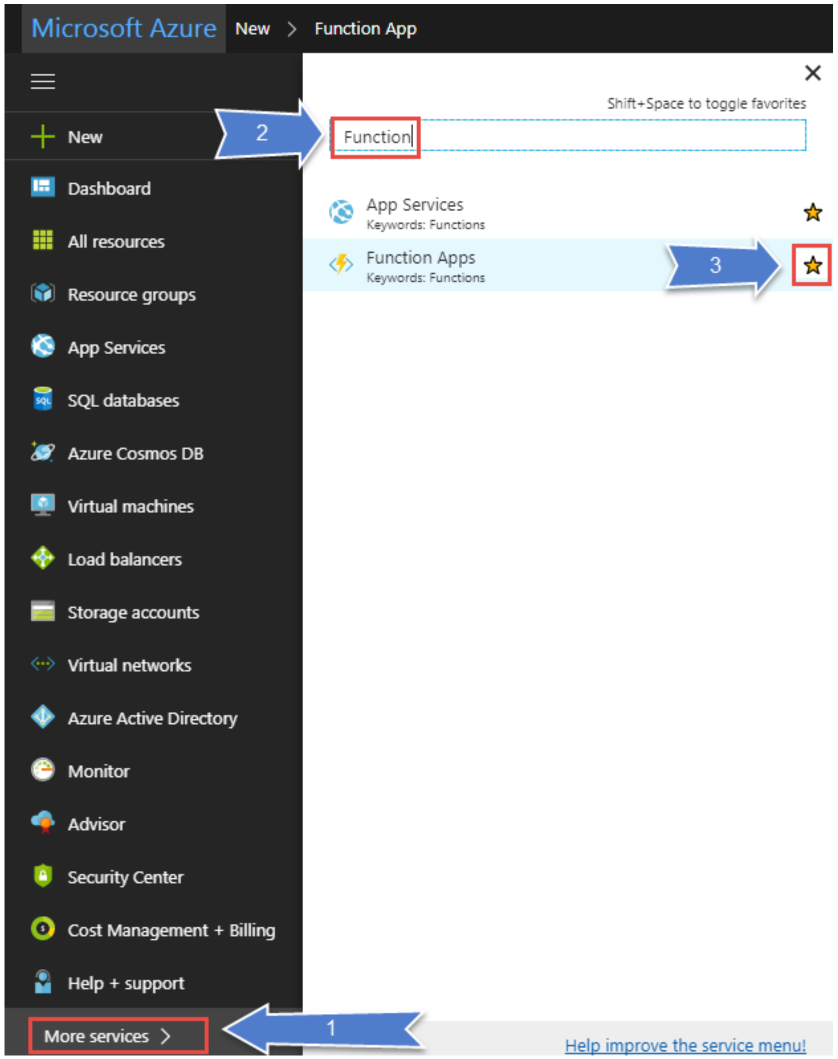
* Storage
[✓ Create new] Use existing
andreasfunctiona2fd ✓

Application Insights On [✓ Off]

✓ Pin to dashboard

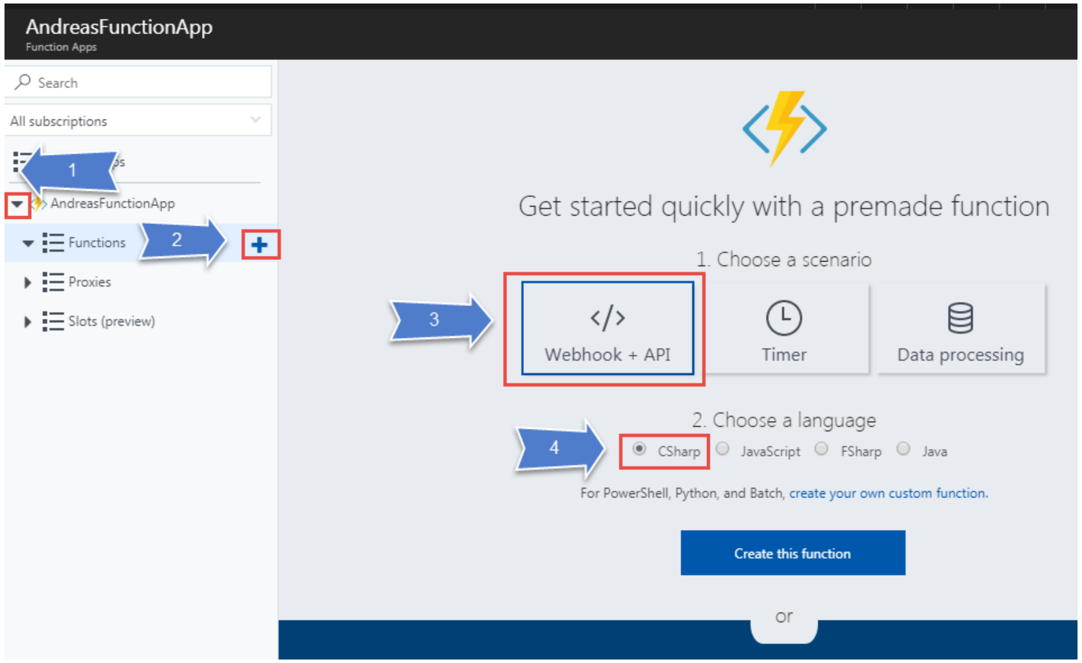
Add your Function App to your Favorites

- You can add your Function App to your favorites so that you can have easier access to it when you are not on your dashboard
- Select More Services
- Type in the search Functions
- Next to Function Apps select the star



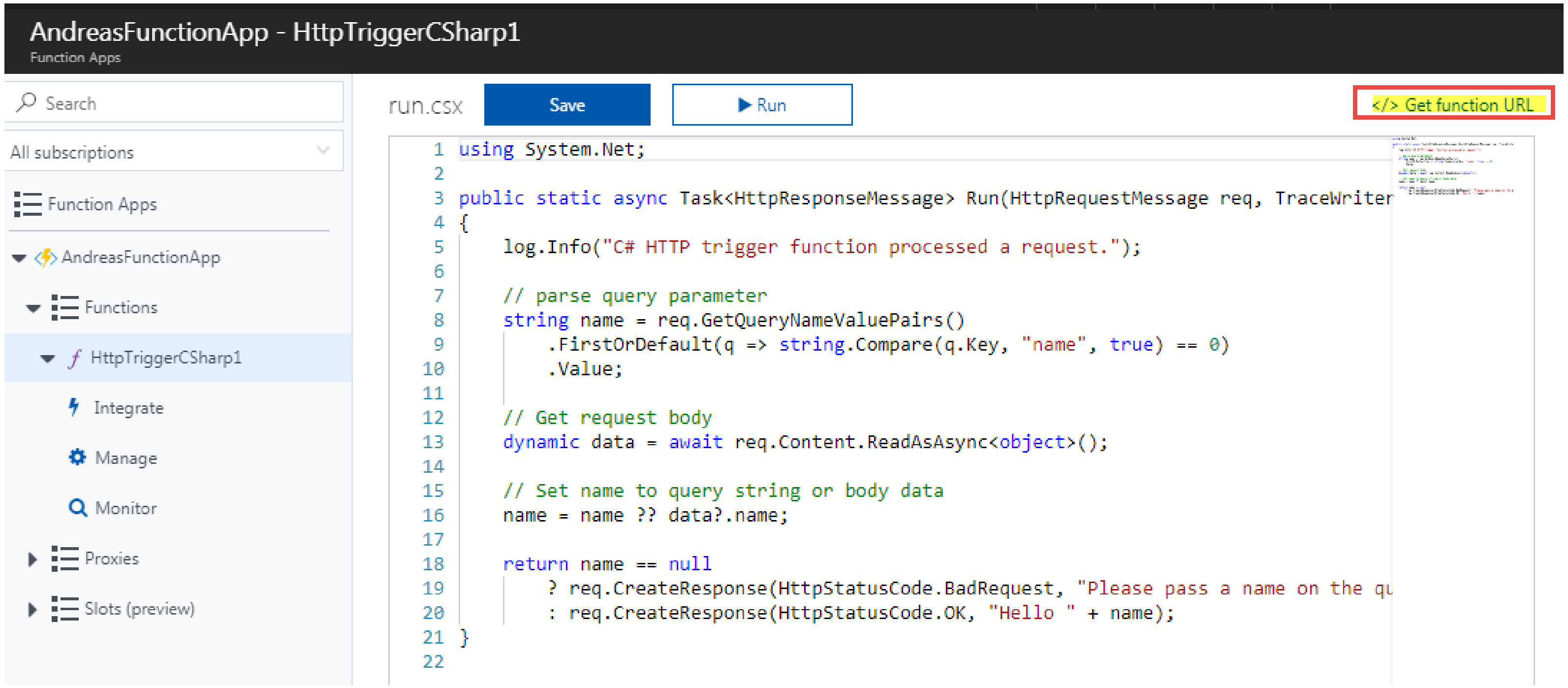
Create an HTTP triggered Function

- Go into your new Function app and expand it to see Functions
- Select the + next to Functions
- Select WebHook + API
- Choose a language for your function
- Select Create this function



Testing your Function

- Once your function is created you can test your function by selecting the </> Get Function URL in the top right

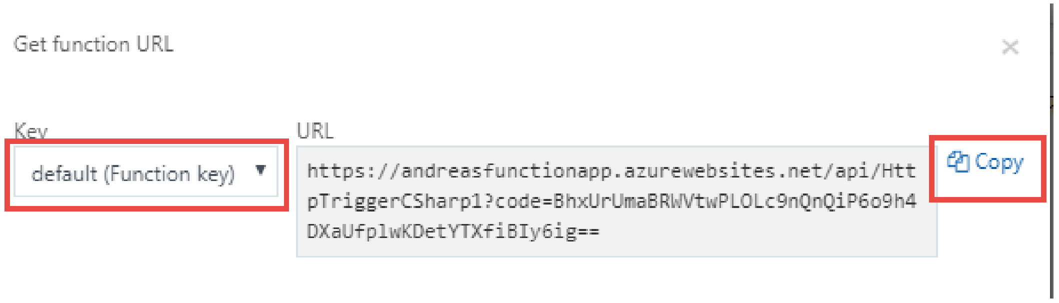


```
using System.Net;

public static async Task<HttpResponseMessage> Run(HttpRequestMessage req, TraceWriter
{
    log.Info("C# HTTP trigger function processed a request.");

    // parse query parameter
    string name = req.GetQueryNameValuePairs()
        .FirstOrDefault(q => string.Compare(q.Key, "name", true) == 0)
        .Value;

    // Get request body
    dynamic data = await req.Content.ReadAsAsync<object>();

    // Set name to query string or body data
    name = name ?? data?.name;

    return name == null
        ? req.CreateResponse(HttpStatusCode.BadRequest, "Please pass a name on the qu
        : req.CreateResponse(HttpStatusCode.OK, "Hello " + name);
}

```

&name=<yourname>
&name=<yourname>

Testing your Function

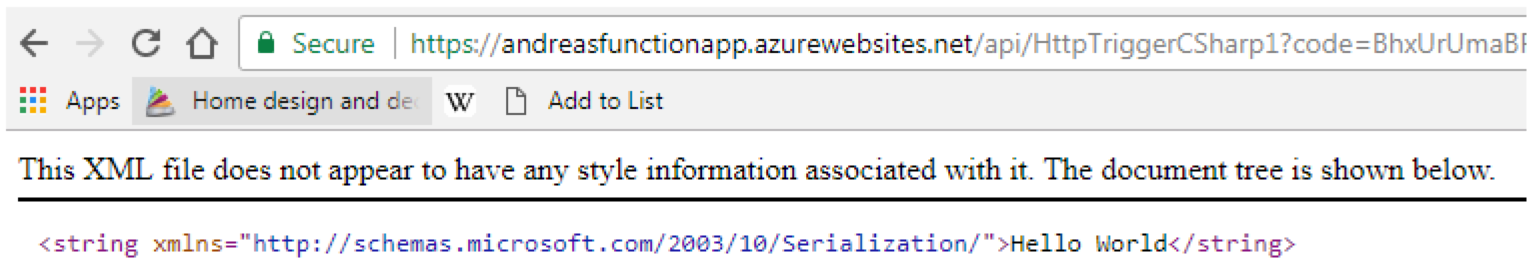
- Make sure that your Key is set to default and then select to copy the URL String



default (Function key)

```
https://andreasfunctionapp.azurewebsites.net/api/HttpTriggerCSharp1?code=BhxUrUmaBRWVtwPLOLc9nQnQiP6o9h4DXaUfplwKDetYTXfiBIy6ig==
```

- Paste the URL string into your web browser and add the following to the end: &name=World
  - This will display Hello World in your browser



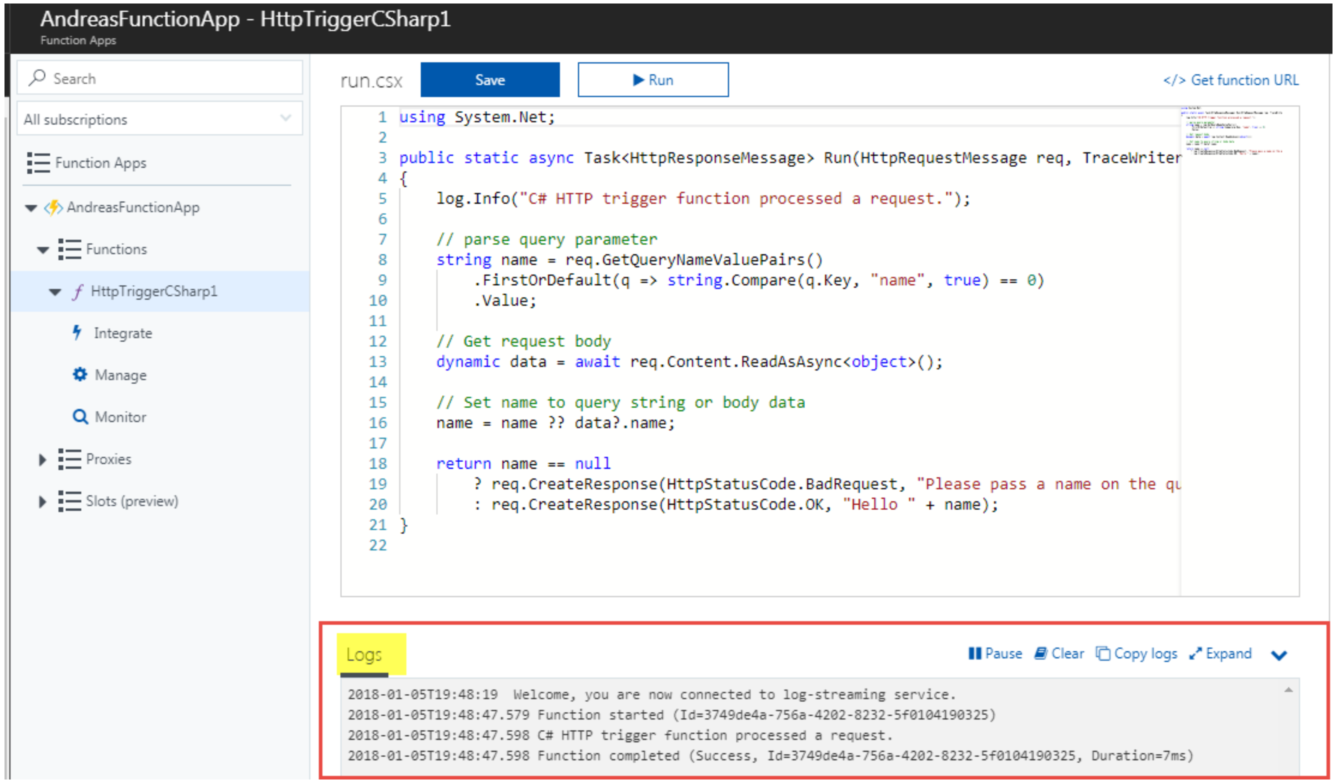
https://andreasfunctionapp.azurewebsites.net/api/HttpTriggerCSharp1?code=BhxUrUmaBI

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Hello World</string>
```

Function Logs

- Logs are created every time that you run your function
- You can view your logs in Azure at the bottom of your function screen



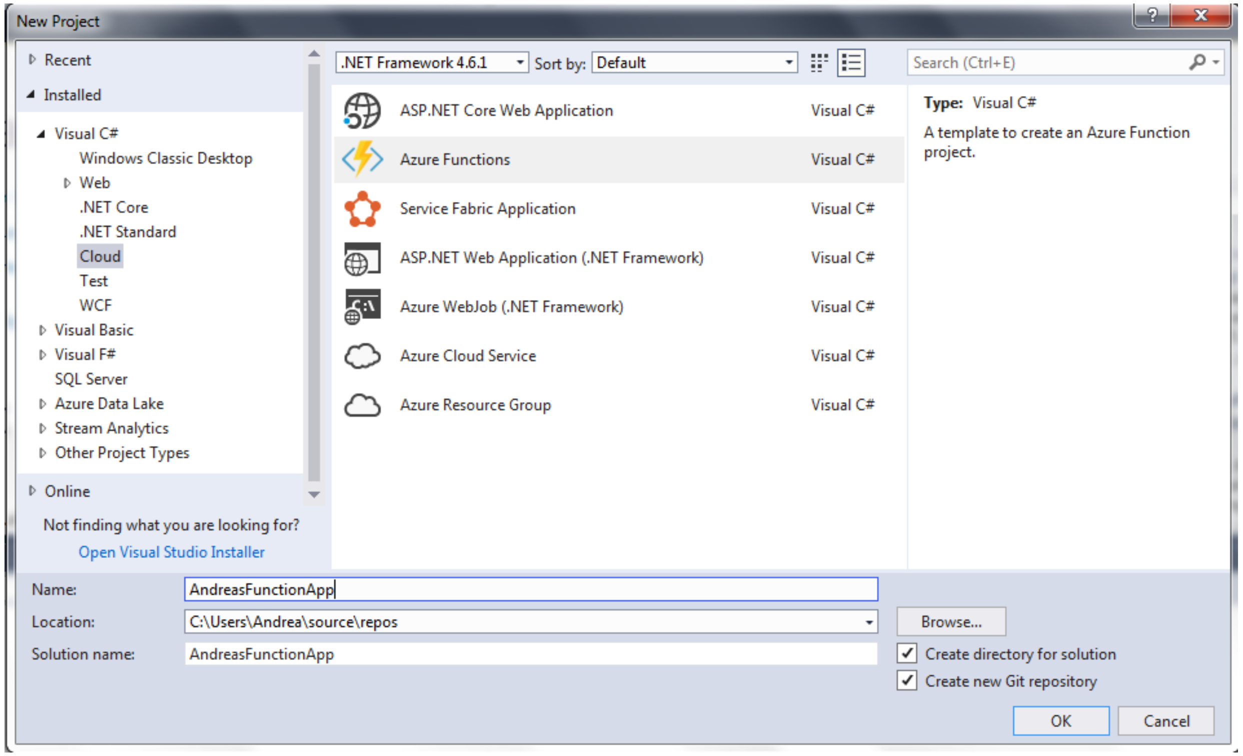
```
using System.Net;

public static async Task<HttpResponseMessage> Run(HttpRequestMessage req, TraceWriter
{
    log.Info("C# HTTP trigger function processed a request.");

    // parse query parameter
    string name = req.GetQueryNameValuePairs()
        .FirstOrDefault(q => string.Compare(q.Key, "name", true) == 0)
        .Value;

    // Get request body
    dynamic data = await req.Content.ReadAsAsync<object>();

    // Set name to query string or body data
    name = name ?? data?.name;

    return name == null
        ? req.CreateResponse(HttpStatusCode.BadRequest, "Please pass a name on the qu
        : req.CreateResponse(HttpStatusCode.OK, "Hello " + name);
}
```

```
2018-01-05T19:48:19  Welcome, you are now connected to log-streaming service.
2018-01-05T19:48:47.579 Function started (Id=3749de4a-756a-4202-8232-5f0104190325)
2018-01-05T19:48:47.598 C# HTTP trigger function processed a request.
2018-01-05T19:48:47.598 Function completed (Success, Id=3749de4a-756a-4202-8232-5f0104190325, Duration=7ms)
```

II. Working with Functions in Visual Studio

Create a Function App

- In VS, Select File-> New-> Project
- Under Installed -> Visual C# -> Cloud -> Azure Functions
- Add a name for your project
- Select Ok



New Project
Recent
Installed
Visual C#
Windows Classic Desktop
Web
.NET Core
.NET Standard
Cloud
Test
WCF
Visual Basic
Visual F#
SQL Server
Azure Data Lake
Stream Analytics
Other Project Types
Online
Not finding what you are looking for?
Open Visual Studio Installer
.NET Framework 4.6.1
Sort by: Default
Search (Ctrl+E)
ASP.NET Core Web Application Visual C#
Azure Functions Visual C#
Service Fabric Application Visual C#
ASP.NET Web Application (.NET Framework) Visual C#
Azure WebJob (.NET Framework) Visual C#
Azure Cloud Service Visual C#
Azure Resource Group Visual C#
Browse...
OK
Cancel

Type: Visual C#

A template to create an Azure Function project.

Name: AndreasFunctionApp

Location: C:\Users\Andrea\source\repos

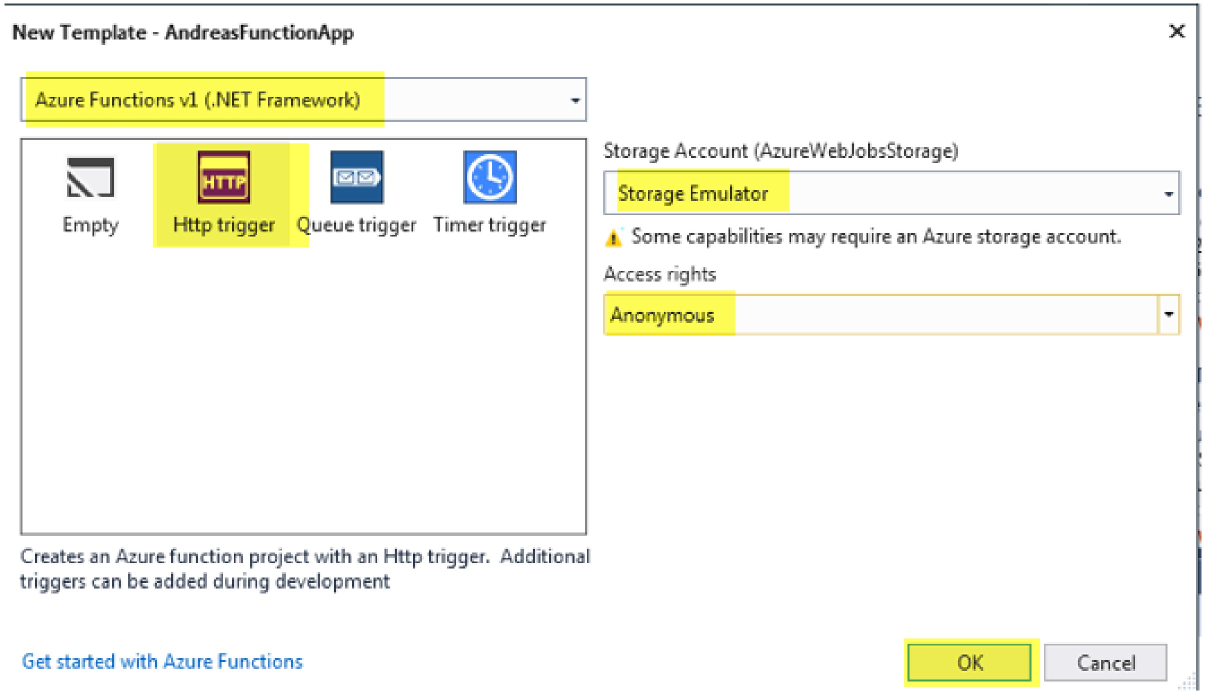
Solution name: AndreasFunctionApp

✓ Create directory for solution

✓ Create new Git repository

Create a Function App

- Select Azure Functions v1
- Select HTTP trigger
- In the Storage Account select Storage Emulator
- For the Access rights select Anonymous
- Select OK to create your function project and HTTP triggered function


New Template - AndreasFunctionApp
Empty
Http trigger
Queue trigger
Timer trigger
Get started with Azure Functions
OK
Cancel

Azure Functions v1 (.NET Framework)

Creates an Azure function project with an Http trigger. Additional triggers can be added during development

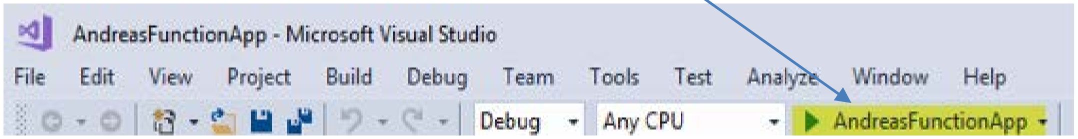
Storage Account (AzureWebJobsStorage): Storage Emulator

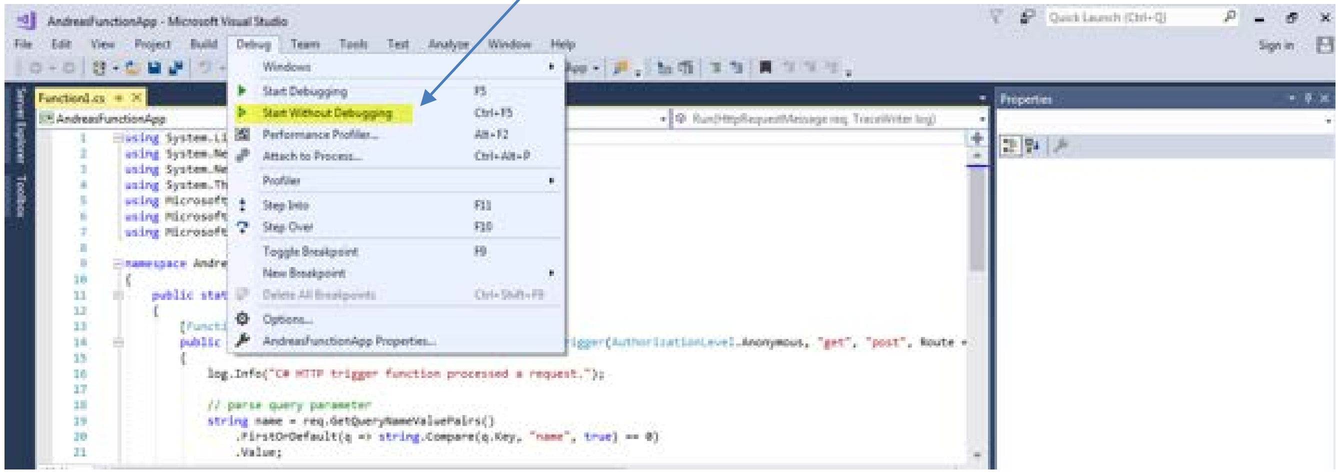
Some capabilities may require an Azure storage account.

Access rights: Anonymous

Test your Function Locally

- Select F5, Run button or Tools -> Start Without Debugging in order to start testing locally as we have done in other labs

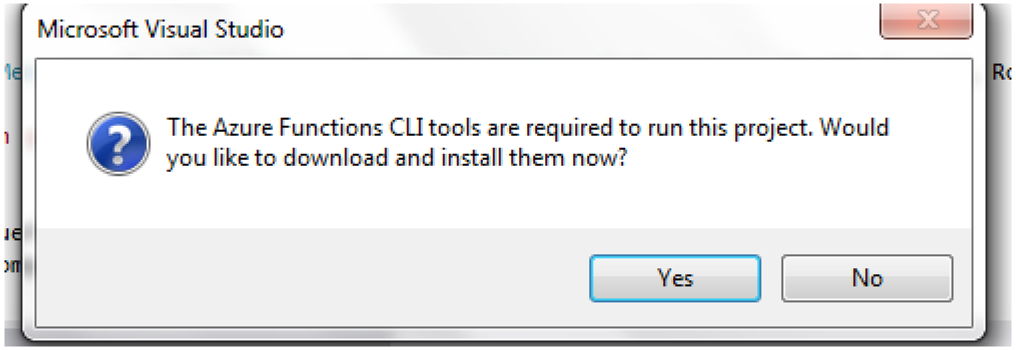




```
using System.Li
using System.Ne
using System.Ne
using System.Th
using Microsoft
using Microsoft
using Microsoft

namespace Andre
{
    public stat
    {
        [Functi
        public                                  rigger(AuthorizationLevel.Anonymous, "get", "post", Route =
        {
            log.Info("C# HTTP trigger function processed a request.");

            // parse query parameter
            string name = req.GetQueryNameValuePairs()
                .FirstOrDefault(q => string.Compare(q.Key, "name", true) == 0)
                .Value;
```

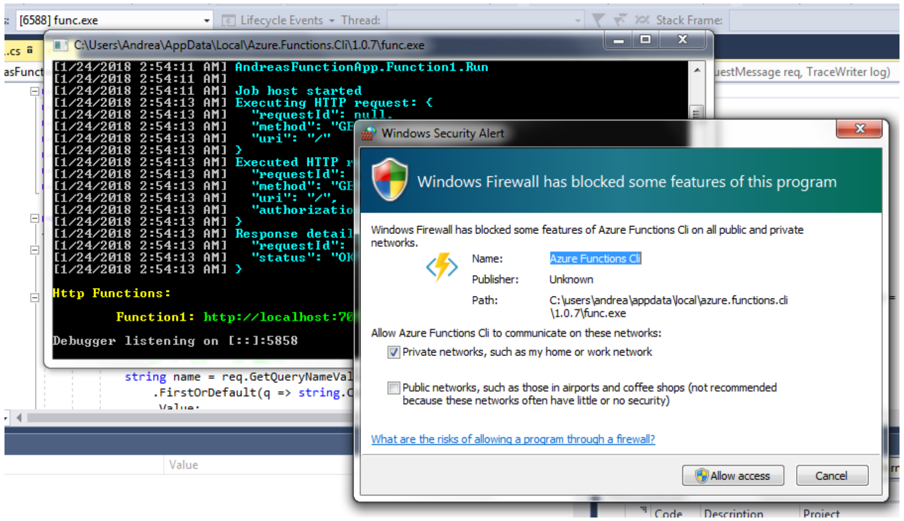
Test your Function Locally

- Some pop ups that you may receive and must accept:
  - Request for VS to download and install Azure Functions Core (CLI) tool
  - Firewall exception so that the tools can handle HTTP requests



Microsoft Visual Studio

The Azure Functions CLI tools are required to run this project. Would you like to download and install them now?



C:\Users\Andrea\AppData\Local\Azure.Functions.Cli\1.0.7\func.exe

```
[1/24/2018 2:54:11 AM] AndreasFunctionApp.Function1.Run
[1/24/2018 2:54:11 AM]
[1/24/2018 2:54:11 AM] Job host started
[1/24/2018 2:54:13 AM] Executing HTTP request: {
[1/24/2018 2:54:13 AM]   "requestId": null,
[1/24/2018 2:54:13 AM]   "method": "GE
[1/24/2018 2:54:13 AM]   "uri": "/"
[1/24/2018 2:54:13 AM] }
[1/24/2018 2:54:13 AM] Executed HTTP r
[1/24/2018 2:54:13 AM]   "requestId":
[1/24/2018 2:54:13 AM]   "method": "GE
[1/24/2018 2:54:13 AM]   "uri": "/",
[1/24/2018 2:54:13 AM]   "authorizatio
[1/24/2018 2:54:13 AM] }
[1/24/2018 2:54:13 AM] Response detail
[1/24/2018 2:54:13 AM]   "requestId":
[1/24/2018 2:54:13 AM]   "status": "OK
[1/24/2018 2:54:13 AM] }

Http Functions:

        Function1: http://localhost:70

Debugger listening on [::]:5858
```

```
string name = req.GetQueryNameVal
    .FirstOrDefault(q => string.C
```

Windows Security Alert

Windows Firewall has blocked some features of this program

Windows Firewall has blocked some features of Azure Functions Cli on all public and private networks.

Name:	Azure Functions Cli
Publisher:	Unknown
Path:	C:\users\andrea\appdata\local\azure.functions.cli \1.0.7\func.exe

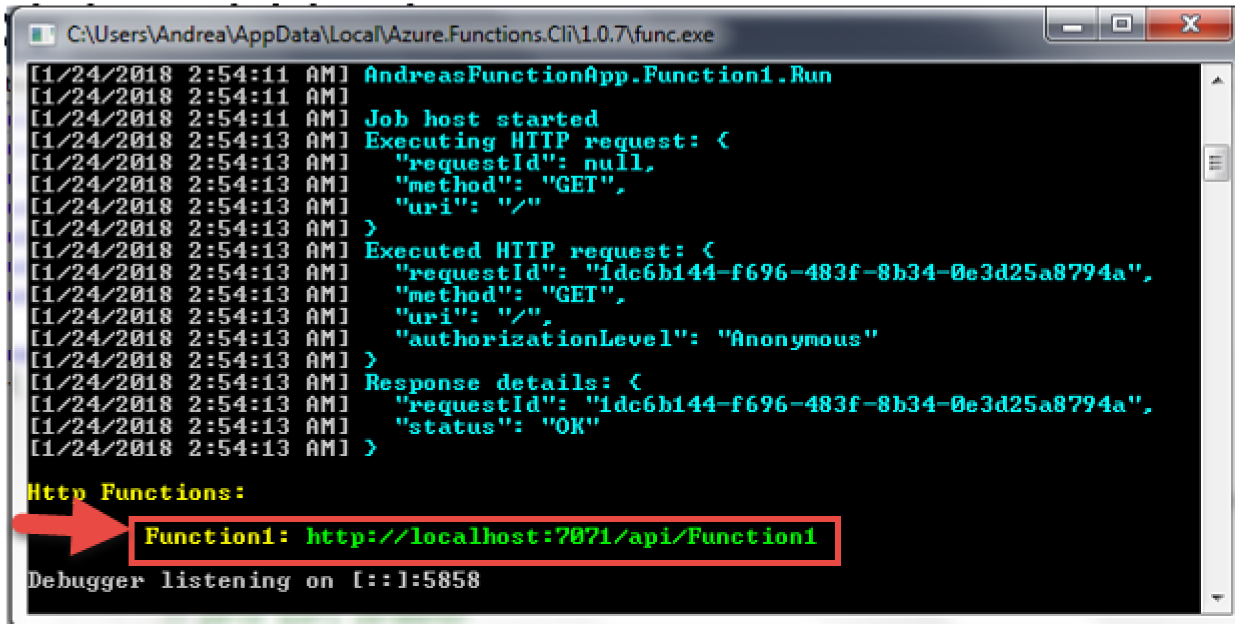
Allow Azure Functions Cli to communicate on these networks:

- [✓] Private networks, such as my home or work network
- [ ] Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

What are the risks of allowing a program through a firewall?

Test your Function Locally

- Your url for your function will pop up in your CLI
- Copy this URL

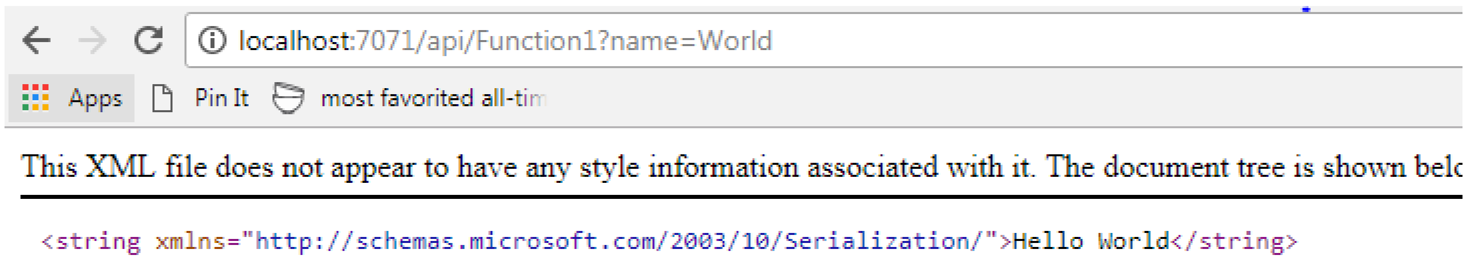


C:\Users\Andrea\AppData\Local\Azure.Functions.Cli\1.0.7\func.exe

```
[1/24/2018 2:54:11 AM] AndreasFunctionApp.Function1.Run
[1/24/2018 2:54:11 AM]
[1/24/2018 2:54:11 AM] Job host started
[1/24/2018 2:54:13 AM] Executing HTTP request: {
[1/24/2018 2:54:13 AM]   "requestId": null,
[1/24/2018 2:54:13 AM]   "method": "GET",
[1/24/2018 2:54:13 AM]   "uri": "/"
[1/24/2018 2:54:13 AM] }
[1/24/2018 2:54:13 AM] Executed HTTP request: {
[1/24/2018 2:54:13 AM]   "requestId": "1dc6b144-f696-483f-8b34-0e3d25a8794a",
[1/24/2018 2:54:13 AM]   "method": "GET",
[1/24/2018 2:54:13 AM]   "uri": "/",
[1/24/2018 2:54:13 AM]   "authorizationLevel": "Anonymous"
[1/24/2018 2:54:13 AM] }
[1/24/2018 2:54:13 AM] Response details: {
[1/24/2018 2:54:13 AM]   "requestId": "1dc6b144-f696-483f-8b34-0e3d25a8794a",
[1/24/2018 2:54:13 AM]   "status": "OK"
[1/24/2018 2:54:13 AM] }

Http Functions:

        Function1: http://localhost:7071/api/Function1

Debugger listening on [::]:5858
```

Test your Function Locally

- Paste your URL into your browser and add the following (?name=World) so that you can say “hello world”
- Example: my url is: http://localhost:7071/api/Function1
- So I would paste the following into my browser: http://localhost:7071/api/Function1?name=World

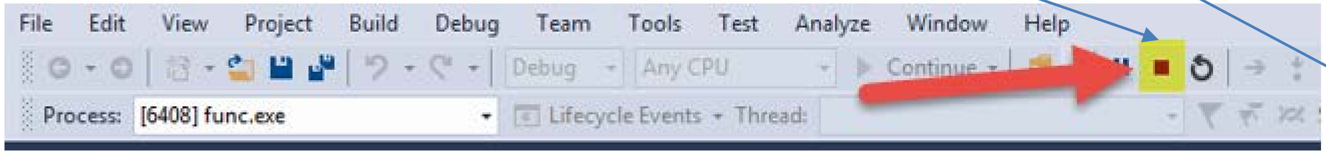


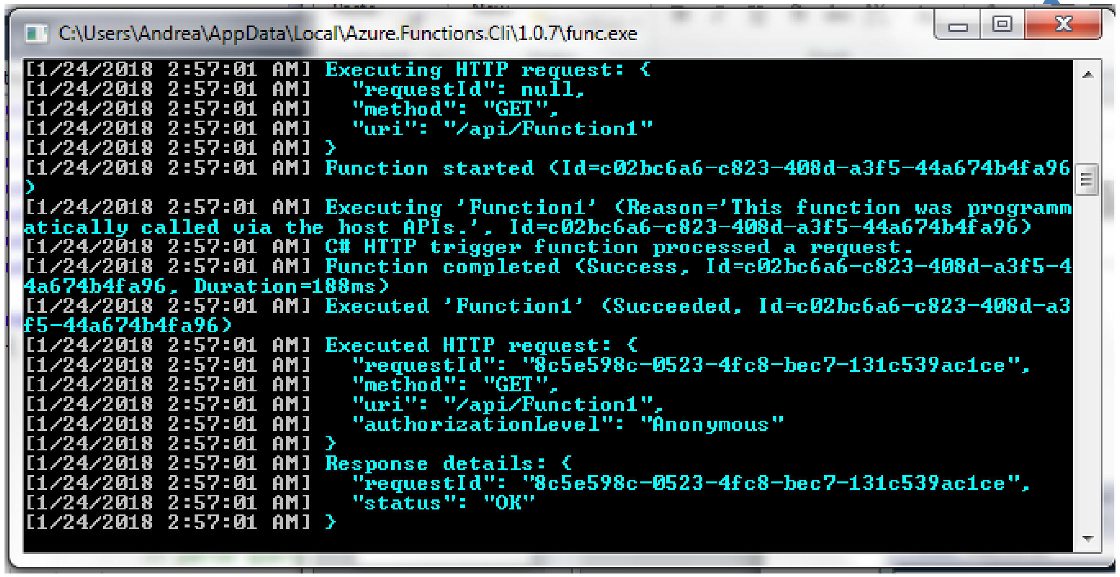
This XML file does not appear to have any style information associated with it. The document tree is shown belo

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Hello World</string>
```

Test your Function Locally

- Finally select the Stop button or exit out of your CLI to stop debugging so that you can publish your project to Azure

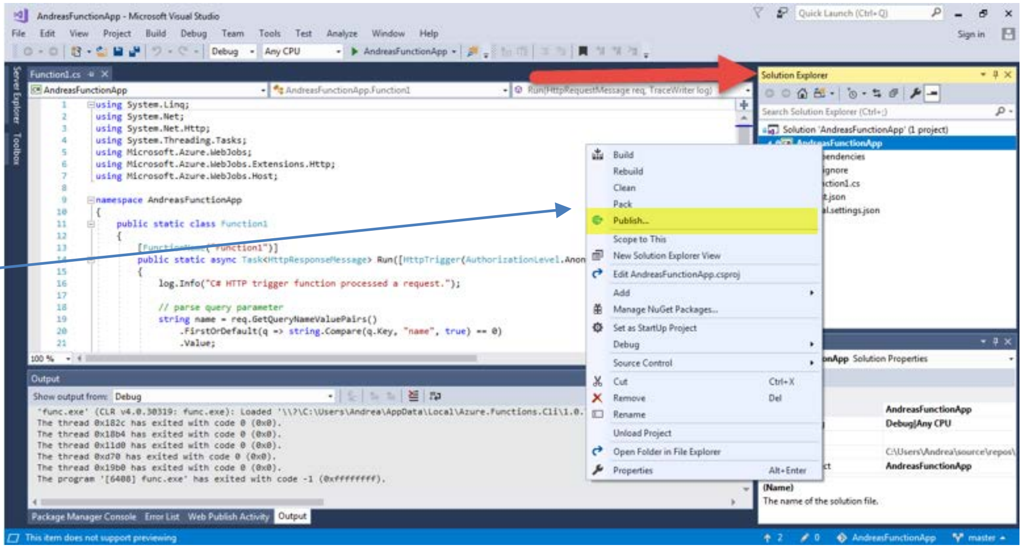



C:\Users\Andrea\AppData\Local\Azure.Functions.Cli\1.0.7\func.exe

```
[1/24/2018 2:57:01 AM] Executing HTTP request: {
[1/24/2018 2:57:01 AM]   "requestId": null,
[1/24/2018 2:57:01 AM]   "method": "GET",
[1/24/2018 2:57:01 AM]   "uri": "/api/Function1"
[1/24/2018 2:57:01 AM] }
[1/24/2018 2:57:01 AM] Function started (Id=c02bc6a6-c823-408d-a3f5-44a674b4fa96
)
[1/24/2018 2:57:01 AM] Executing 'Function1' (Reason='This function was programm
atically called via the host APIs.', Id=c02bc6a6-c823-408d-a3f5-44a674b4fa96)
[1/24/2018 2:57:01 AM] C# HTTP trigger function processed a request.
[1/24/2018 2:57:01 AM] Function completed (Success, Id=c02bc6a6-c823-408d-a3f5-4
4a674b4fa96, Duration=188ms)
[1/24/2018 2:57:01 AM] Executed 'Function1' (Succeeded, Id=c02bc6a6-c823-408d-a3
f5-44a674b4fa96)
[1/24/2018 2:57:01 AM] Executed HTTP request: {
[1/24/2018 2:57:01 AM]   "requestId": "8c5e598c-0523-4fc8-bec7-131c539ac1ce",
[1/24/2018 2:57:01 AM]   "method": "GET",
[1/24/2018 2:57:01 AM]   "uri": "/api/Function1",
[1/24/2018 2:57:01 AM]   "authorizationLevel": "Anonymous"
[1/24/2018 2:57:01 AM] }
[1/24/2018 2:57:01 AM] Response details: {
[1/24/2018 2:57:01 AM]   "requestId": "8c5e598c-0523-4fc8-bec7-131c539ac1ce",
[1/24/2018 2:57:01 AM]   "status": "OK"
[1/24/2018 2:57:01 AM] }
```

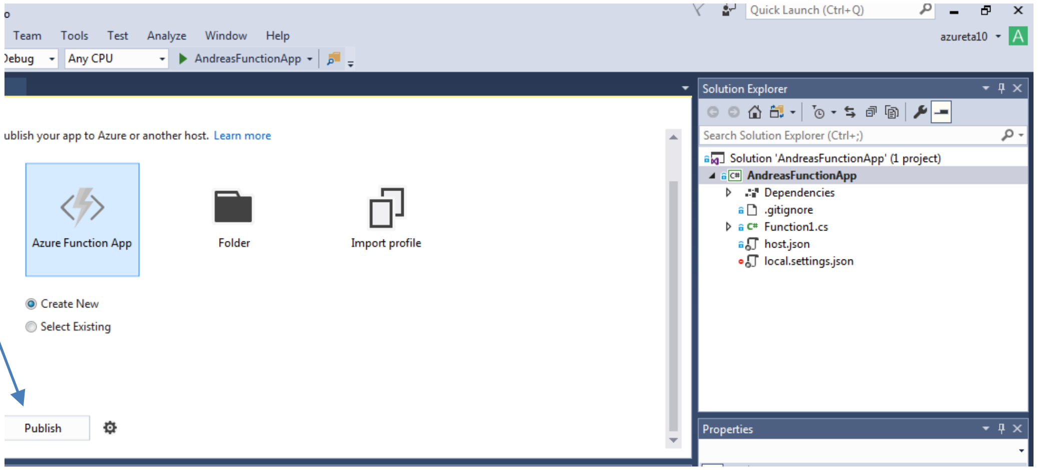
Publish to Azure

- In your solution explorer (on the right), right click on your project and click Publish
- Select Create New and then click Publish
- Note: Make sure that you have connected your VS account to your Azure account


AndreasFunctionApp - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug Any CPU AndreasFunctionApp
Function1.cs
Solution Explorer
Solution 'AndreasFunctionApp' (1 project)
AndreasFunctionApp
Build
Rebuild
Clean
Pack
Publish...
Scope to This
New Solution Explorer View
Edit AndreasFunctionApp.csproj
Add
Manage NuGet Packages...
Set as StartUp Project
Debug
Source Control
Cut Ctrl+X
Remove Del
Rename
Unload Project
Open Folder in File Explorer
Properties Alt+Enter
Package Manager Console
Error List
Web Publish Activity
Output
This item does not support previewing

```
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Threading.Tasks;
using Microsoft.Azure.WebJobs;
using Microsoft.Azure.WebJobs.Extensions.Http;
using Microsoft.Azure.WebJobs.Host;

namespace AndreasFunctionApp
{
    public static class Function1
    {
        [FunctionName("Function1")]
        public static async Task<HttpResponseMessage> Run([HttpTrigger(AuthorizationLevel.Anon
        {
            log.Info("C# HTTP trigger function processed a request.");

            // parse query parameter
            string name = req.GetQueryNameValuePairs()
                .FirstOrDefault(q => string.Compare(q.Key, "name", true) == 0)
                .Value;
```

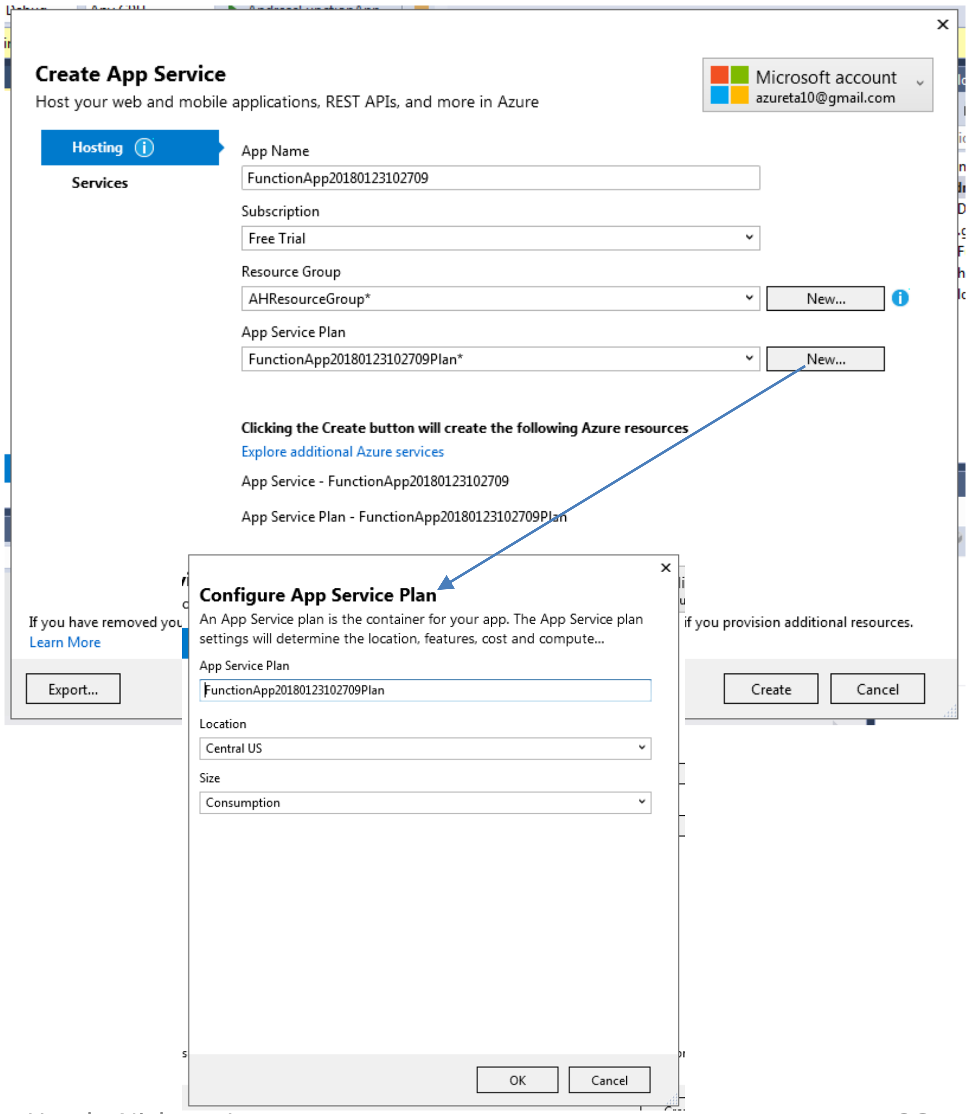
```
'func.exe' (CLR v4.0.30319: func.exe): Loaded '\\?\C:\Users\Andrea\AppData\Local\Azure.Functions.Cli\1.0.
The thread 0x182c has exited with code 0 (0x0).
The thread 0x18b4 has exited with code 0 (0x0).
The thread 0x11d0 has exited with code 0 (0x0).
The thread 0xd70 has exited with code 0 (0x0).
The thread 0x19b0 has exited with code 0 (0x0).
The program '[6408] func.exe' has exited with code -1 (0xffffffff).
```



ublish your app to Azure or another host. Learn more

Publish to Azure

- In the Create App Service dialog that appears, fill in the following details:
- Keep the unique App Name
- Keep the Subscription the default
- Select a Resource Group or create a new one
- Select new for your App Service Plan and then choose Consumption under size and choose your location
- Select Create


Create App Service

Host your web and mobile applications, REST APIs, and more in Azure

Microsoft account
azureta10@gmail.com

Hosting
Services

App Name: FunctionApp20180123102709

Subscription: Free Trial

Resource Group: AHResourceGroup*

App Service Plan: FunctionApp20180123102709Plan*

Clicking the Create button will create the following Azure resources

Explore additional Azure services

App Service - FunctionApp20180123102709

App Service Plan - FunctionApp20180123102709Plan

If you have removed you … if you provision additional resources.

Learn More

Configure App Service Plan

An App Service plan is the container for your app. The App Service plan settings will determine the location, features, cost and compute...

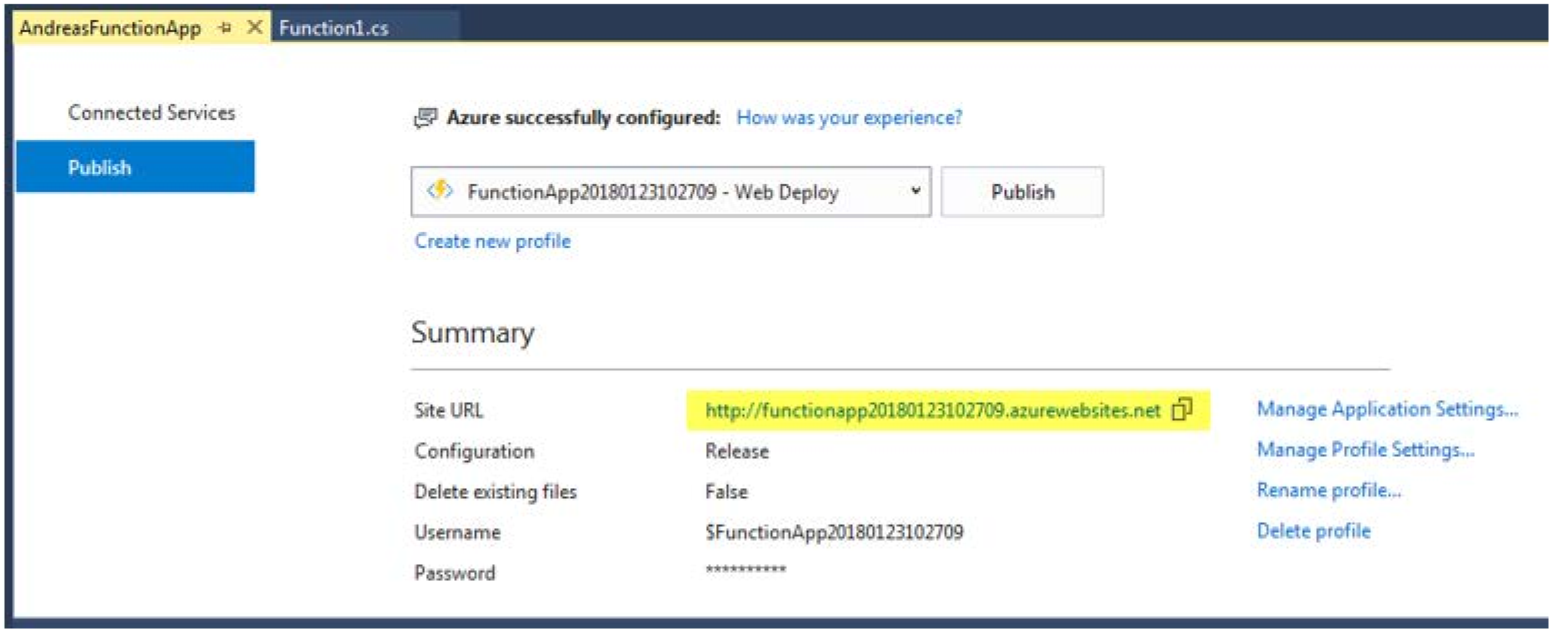
App Service Plan: FunctionApp20180123102709Plan

Location: Central US

Size: Consumption

Publish to Azure

- Your function app will now be ready to be viewed and in your Publish Summary you can see your Site URL
- Make a note of your URL so that you can open a web browser and again view your app



Azure successfully configured:

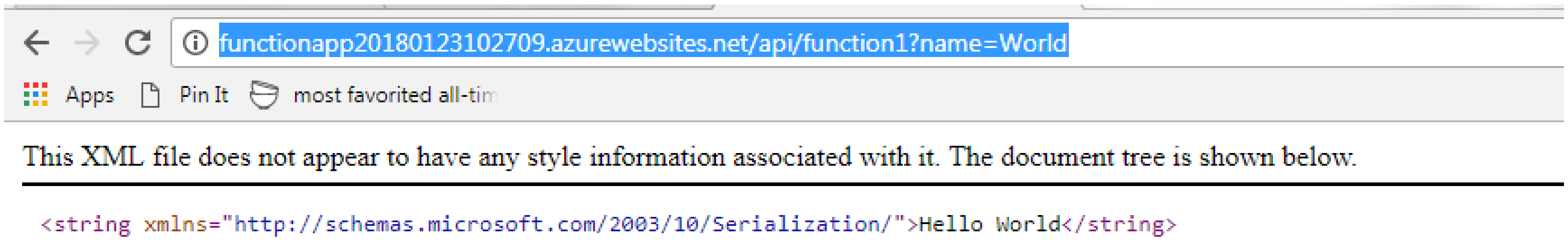
FunctionApp20180123102709 - Web Deploy

Summary

Site URL	http://functionapp20180123102709.azurewebsites.net
Configuration	Release
Delete existing files	False
Username	$FunctionApp20180123102709
Password	**********

Publish to Azure

- Paste in your url where you had originally said localhost and again add ?name=World
- Note: You will need your URL plus /api/Function Name
  - Example: functionapp20180123102709.azurewebsites.net/api/function1?name=World


functionapp20180123102709.azurewebsites.net/api/function1?name=World
Apps
Pin It
most favorited all-tim

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Hello World</string>
```

Summary

- Demonstrated steps to:
- Create a Function App in Azure Portal
- Create a Function App in Visual Studio
- Publish a VS Function App in Azure