Lab Manual- Performing the Blob Operation using VS Code

Prepared for: Training

Date: 18th Nov 2018

Prepared by: Aditi Shrivastava

Document Name: Lab Manual

Document Number BLOBLAB0721

Contributor:

Table of Contents

1 OBJECTIVE

Deploying your software becomes a lot easier after Docker where you don't have to worry about missing a system configuration or a prerequisite. In This Lab will cover the basics of installing Apache Webserv er on Ubuntu and run Image with Docker containers expose on port 80.

- Pull latest Ubuntu Image
- Create a container with Ubuntu Image and Expose it on port 80
- Update the Ubuntu Image
- Install Apache web server
- Start the Apache service
- Install Nano Editor and edit the default Index.html
- Access the web site from browser using docker IP

2 PRE-REQUISISTE

- Prior knowledge of Linux
- Prior knowledge of docker
- A local Computer with 4 CPU, 16 GB RAM, 200 GB disk space

3 Lab Scenario

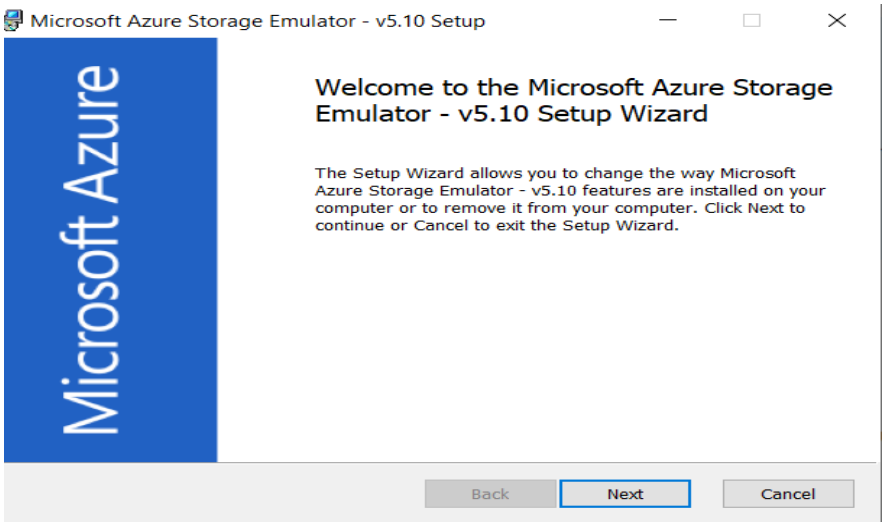
4 Setup the Enviornment

4.1 Step1 : Download and Install Storage Emulator

- Download the Azure Storage Emulator from below URL

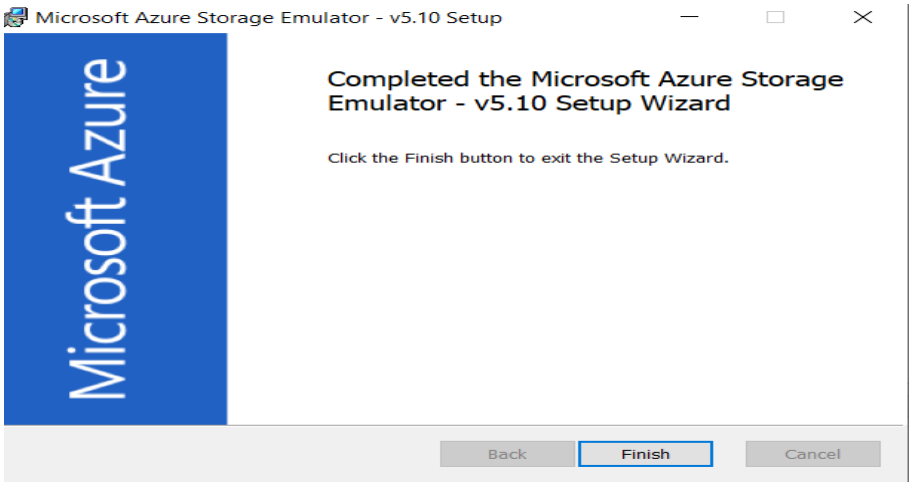
  https://go.microsoft.com/fwlink/?linkid=717179&clcid=0x409

- Install the Emulator by Clicking the MSI down loaded
  - Click Next



A. Click Next to Begin Install

B. Click Finish to Begin Install

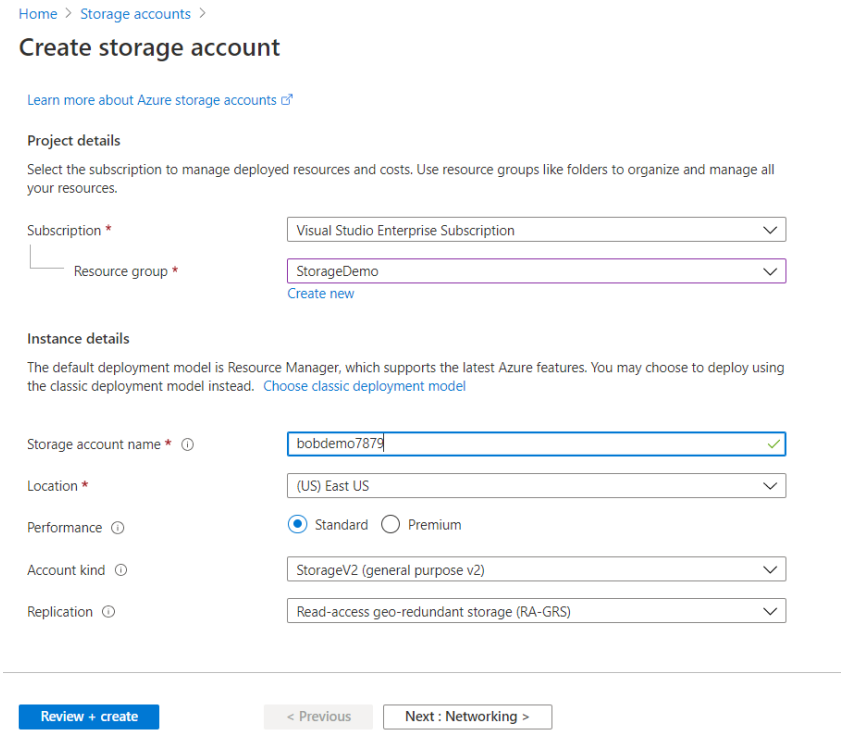


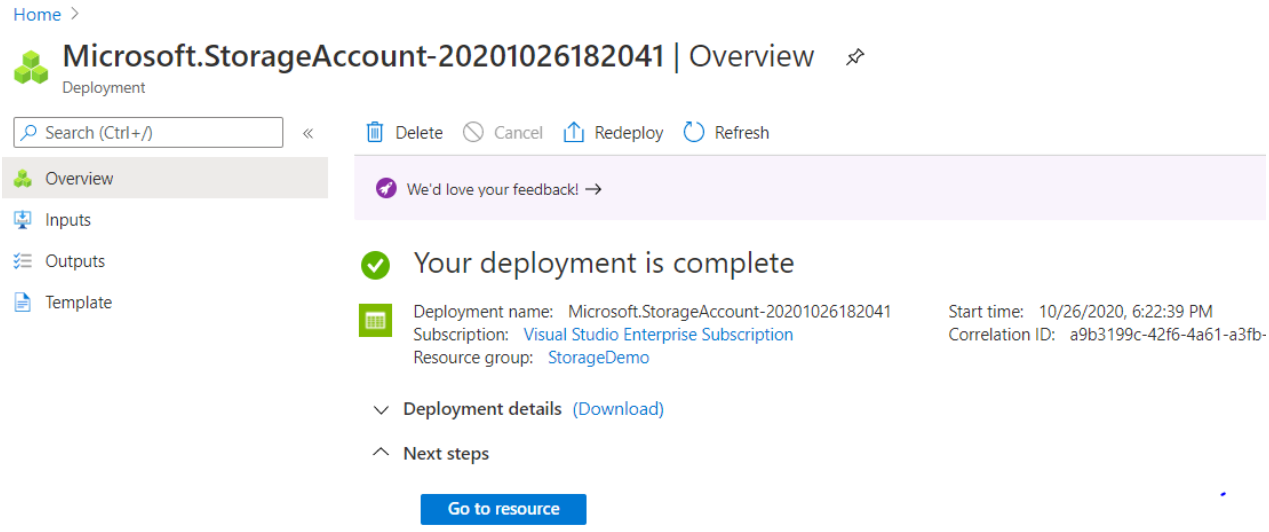
4.2 Step2 : Create an Folder and Clone the Code from Github

A. Open Command Prompt and Create a Folder ad then go inside it.

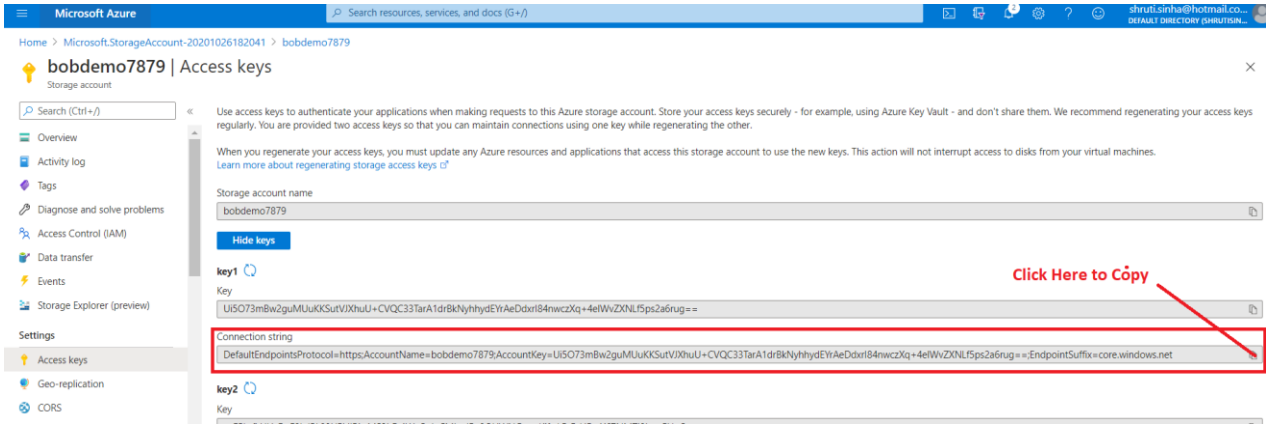
```
md blobtest
cd blobtest
```

4.3 Step3 : Create an a Storage Account and Set the Enviornment Variable





- In the **Settings** section of the storage account overview, select **Access keys**. Here, you can view your account access keys and the complete connection string for each key.



- Find the **Connection string** value under **key1**, and select the **Copy** button to copy the connection string. You will add the connection string value to an environment variable in the next step.

4.3.1 Configure your storage connection string

- To set the environment variable, open a **DOS** console window and Type below command

setx AZURE_STORAGE_CONNECTION_STRING

"DefaultEndpointsProtocol=https;AccountName=funcstorshruti;AccountKey=r2pI9EDNYbffYjeGeU5Ffg0asft9+cqYkJ5hjZ4OWpq5Kn5zjycxn0Alf74be/y3DXHFyizxH984QN7CTZoCoA==;EndpointSuffix=core.windows.net"

```
D:\blobtest\blobmvp\BlobQuickstartV12>setx AZURE_STORAGE_CONNECTION_STRING "DefaultEndpointsProtocol=https;AccountName=b
obdemo7879;AccountKey=Ui5O73mBw2guMUuKKSutVJXhuU+CVQC33TarA1drBkNyhhydEYrAeDdxrI84nwczXq+4elWvZXNLf5ps2a6rug==;EndpointS
uffix=core.windows.net"
```

- Now Close the DOS Console and Relaunch it again

4.4 Create a .NET Core application named BlobQuickstartV12.

- Now Create New Project use the `dotnet new` command to create a new console app with the name *BlobQuickstartV12*

  dotnet new console -n BlobQuickstartV12

```
PS D:\blobtest> cd .\blobmvp\
PS D:\blobtest\blobmvp> dotnet new console -n BlobQuickstartV12

Welcome to .NET Core 3.1!
---------------------
SDK Version: 3.1.403

Telemetry
---------
The .NET Core tools collect usage data in order to help us improve your experience.
t of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variable to '

Read more about .NET Core CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

----------------
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
```

- Switch to the newly created **BlobQuickstartV12** directory.

  `cd BlobQuickstartV12`

- In side the **BlobQuickstartV12** directory, create another directory called **data.** This is where the blob data files will be created and stored.

  Mkdir data

```
PS D:\blobtest\blobmvp> cd BlobQuickstartV12
PS D:\blobtest\blobmvp\BlobQuickstartV12> mkdir data


    Directory: D:\blobtest\blobmvp\BlobQuickstartV12


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        26-10-2020     18:56                data
```

4.4.1 install the Azure Blob storage client library for .NET package

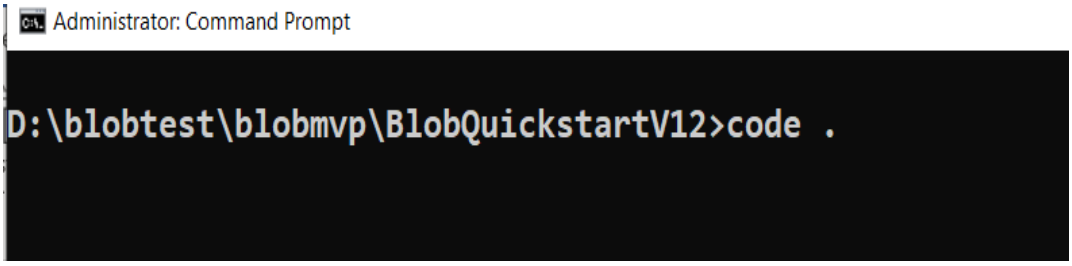
While still in the application directory, install the Azure Blob storage client library for .NET package by using the `dotnet add package` command.

dotnet add package Azure.Storage.Blobs

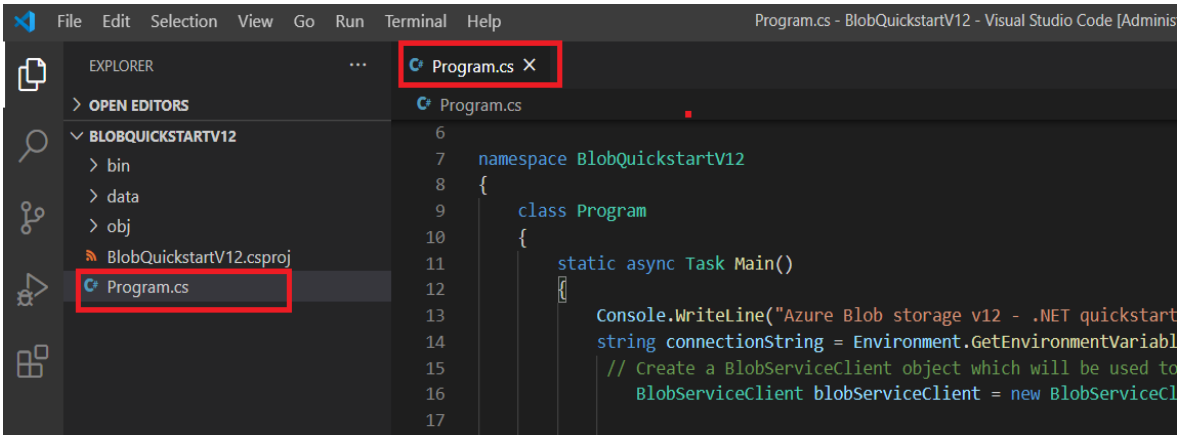
```
PS D:\blobtest\blobmvp\BlobQuickstartV12> dotnet add package Azure.Storage.Blobs
  Determining projects to restore...
  Writing C:\Users\TechPledge\AppData\Local\Temp\tmp15FC.tmp
info : Adding PackageReference for package 'Azure.Storage.Blobs' into project 'D:\blobtest\blobmvp\BlobQuickstartV12\BlobQuickstartV12.csproj'.
info : Restoring packages for D:\blobtest\blobmvp\BlobQuickstartV12\BlobQuickstartV12.csproj...
info :   GET https://api.nuget.org/v3-flatcontainer/azure.storage.blobs/index.json
info :   OK https://api.nuget.org/v3-flatcontainer/azure.storage.blobs/index.json 1158ms
info :   GET https://api.nuget.org/v3-flatcontainer/azure.storage.blobs/12.6.0/azure.storage.blobs.12.6.0.nupkg
info :   OK https://api.nuget.org/v3-flatcontainer/azure.storage.blobs/12.6.0/azure.storage.blobs.12.6.0.nupkg 1019ms
info :   GET https://api.nuget.org/v3-flatcontainer/azure.storage.common/index.json
info :   GET https://api.nuget.org/v3-flatcontainer/azure.core/index.json
info :   GET https://api.nuget.org/v3-flatcontainer/system.text.json/index.json
info :   OK https://api.nuget.org/v3-flatcontainer/azure.storage.common/index.json 1018ms
info :   GET https://api.nuget.org/v3-flatcontainer/azure.storage.common/12.5.2/azure.storage.common.12.5.2.nupkg
info :   OK https://api.nuget.org/v3-flatcontainer/azure.core/index.json 1037ms
info :   GET https://api.nuget.org/v3-flatcontainer/azure.core/1.4.1/azure.core.1.4.1.nupkg
info :   OK https://api.nuget.org/v3-flatcontainer/system.text.json/index.json 1060ms
```

4.4.2 Set up the app framework

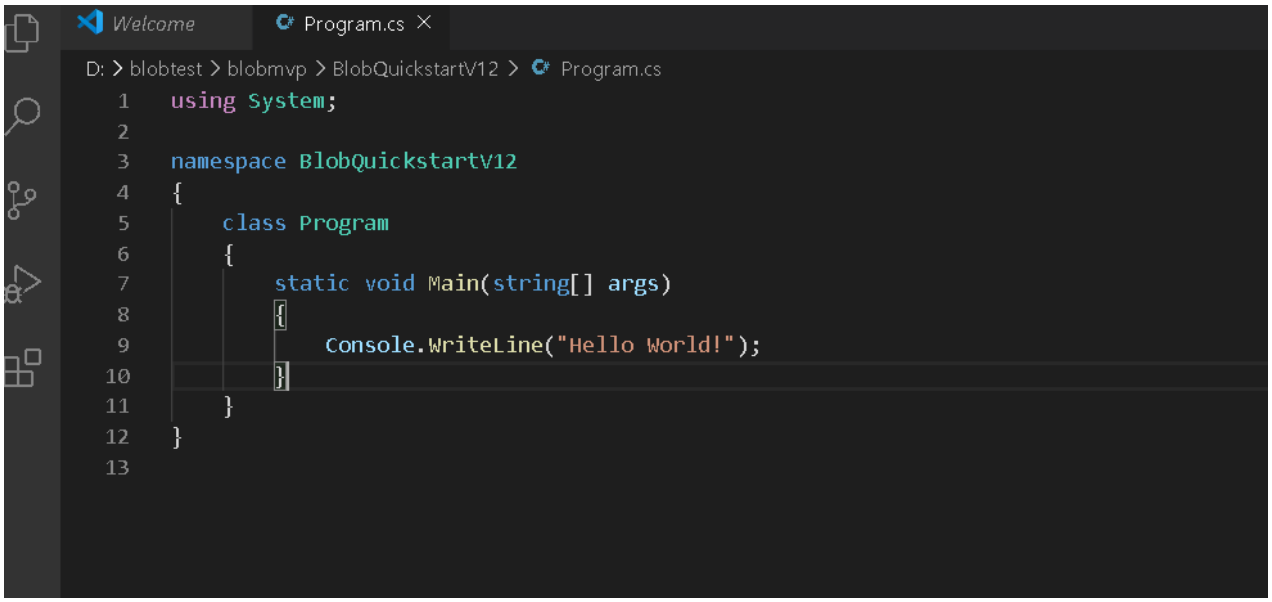
1. To Launch the Code Editor type ***Code .*** file in your editor Code .



2. Open ***program.cs*** file for editing



3. You Got the Default Code



5 Create the Container in Azure Storage and Upload the Blob

5.1 Add this code inside the Main method:

```
using Azure.Storage.Blobs;
using Azure.Storage.Blobs.Models;
using System;
using System.IO;
using System.Threading.Tasks;

namespace BlobQuickstartV12
{
    class Program
    {
        static async Task Main()
        {
            Console.WriteLine("Azure Blob storage v12 - .NET quickstart sample\n");
            string connectionString = Environment.GetEnvironmentVariable("AZURE_STORAGE_CONNECTION_STRING");
            // Create a BlobServiceClient object which will be used to create a container client
                BlobServiceClient blobServiceClient = new BlobServiceClient(connectionString);

                //Create a unique name for the container
                string containerName = "quickstartblobs" + Guid.NewGuid().ToString();

            // Create the container and return a container client object
                BlobContainerClient containerClient = await blobServiceClient.CreateBlobContainerAsync(containerName);

                // Create a local file in the ./data/ directory for uploading and downloading
                    string localPath = "./data/";
                    string fileName = "quickstart" + Guid.NewGuid().ToString() + ".txt";
                    string localFilePath = Path.Combine(localPath, fileName);

                    // Write text to the file
                    await File.WriteAllTextAsync(localFilePath, "Hello, World!");

                    // Get a reference to a blob
                    BlobClient blobClient = containerClient.GetBlobClient(fileName);
```

```
            Console.WriteLine("Uploading to Blob storage as blob:\n\t {0}\n", blobClient.Uri);

            // Open the file and upload its data
            using FileStream uploadFileStream = File.OpenRead(localFilePath);
            await blobClient.UploadAsync(uploadFileStream, true);
            uploadFileStream.Close();
    }

    }

}
```

Program.cs

```
using Azure.Storage.Blobs;
using Azure.Storage.Blobs.Models;
using System;
using System.IO;
using System.Threading.Tasks;

namespace BlobQuickstartV12
{
    class Program
    {
        static async Task Main()
        {
            Console.WriteLine("Azure Blob storage v12 - .NET quickstart sample\n");
            string connectionString = Environment.GetEnvironmentVariable("AZURE_STORAGE_CONNECTION_STRING");
             // Create a BlobServiceClient object which will be used to create a container client
                BlobServiceClient blobServiceClient = new BlobServiceClient(connectionString);

                //Create a unique name for the container
                string containerName = "quickstartblobs" + Guid.NewGuid().ToString();

            // Create the container and return a container client object
                BlobContainerClient containerClient = await blobServiceClient.CreateBlobContainerAsync(containerName);

                // Create a local file in the ./data/ directory for uploading and downloading
                    string localPath = "./data/";
                    string fileName = "quickstart" + Guid.NewGuid().ToString() + ".txt";
                    string localFilePath = Path.Combine(localPath, fileName);

                    // Write text to the file
                    await File.WriteAllTextAsync(localFilePath, "Hello, World!");

                    // Get a reference to a blob
                    BlobClient blobClient = containerClient.GetBlobClient(fileName);

                    Console.WriteLine("Uploading to Blob storage as blob:\n\t {0}\n", blobClient.Uri);

                    // Open the file and upload its data
                    using FileStream uploadFileStream = File.OpenRead(localFilePath);
                    await blobClient.UploadAsync(uploadFileStream, true);
                    uploadFileStream.Close();
        }

        }

    }
```

- Now Save the File with File→Save

5.2 Now Build and Run The Code

- In Terminal Type below command

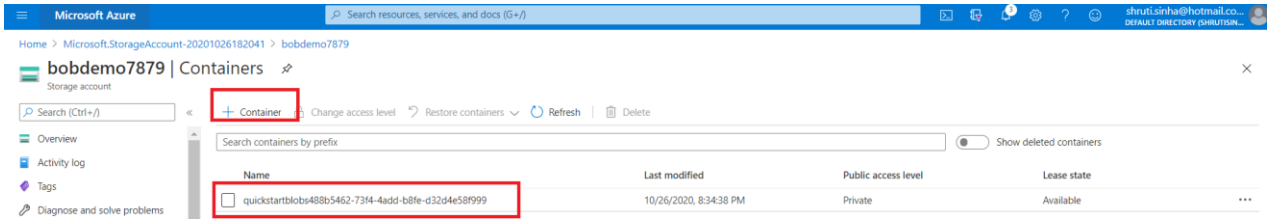
dotnet build

```
PS D:\blobtest\blobmvp\BlobQuickstartV12> dotnet build
Microsoft (R) Build Engine version 16.7.0+7fb82e5b2 for .NET
Copyright (C) Microsoft Corporation. All rights reserved.

  Determining projects to restore...
  All projects are up-to-date for restore.
  BlobQuickstartV12 -> D:\blobtest\blobmvp\BlobQuickstartV12\bin\Debug\netcoreapp3.1\BlobQuickstartV12.dll

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:01.23
```

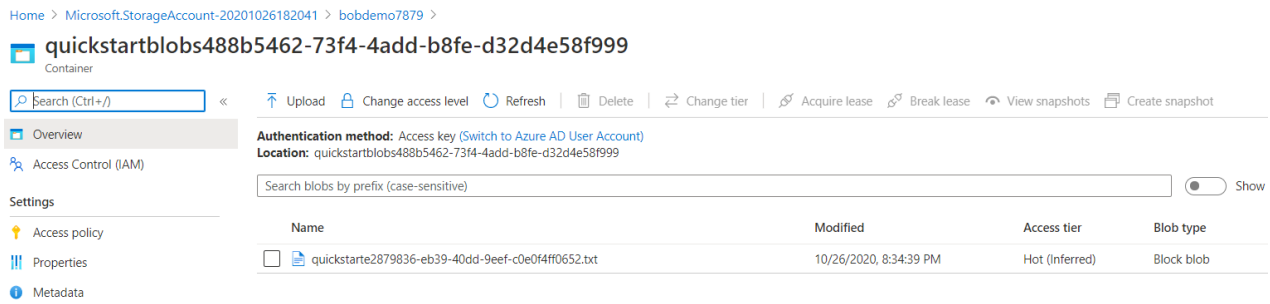
dotnet run

```
PS D:\blobtest\blobmvp\BlobQuickstartV12> dotnet run
Azure Blob storage v12 - .NET quickstart sample

Uploading to Blob storage as blob:
        https://bobdemo7879.blob.core.windows.net/quickstartblobse7ee4076-b280-4c53-8227-ea9592d6ad1b/quickstart38c26bbe-0b4a-49a8-8956-a6c4f5676c7f.txt
```

- Now Go to Portal and Check you have Blob Container Created



- Go Inside the Container and Check the Blob



6 List the blobs in a container

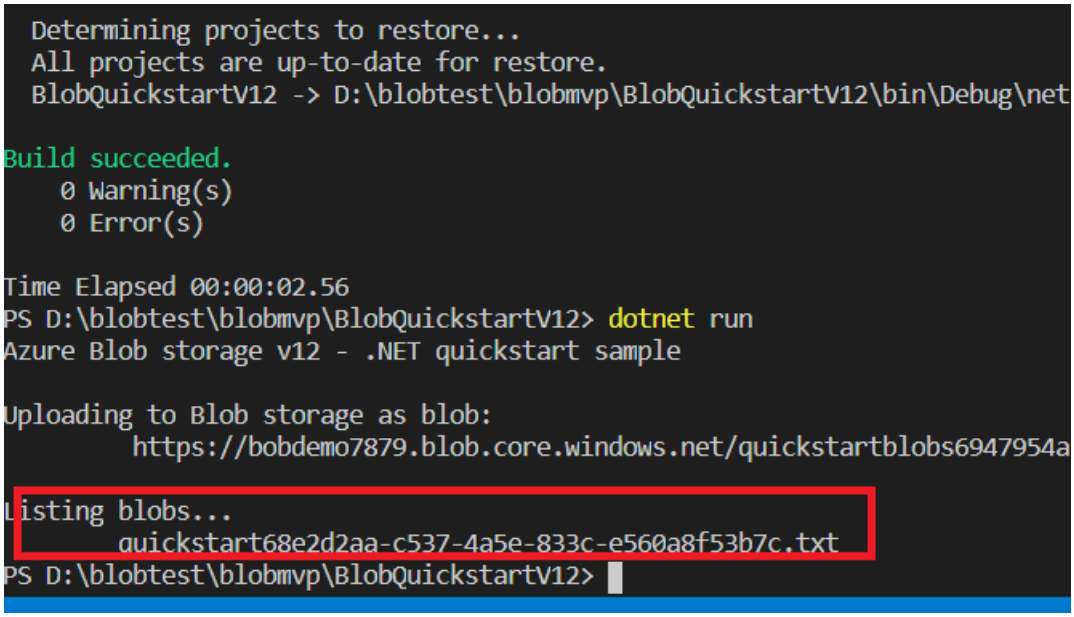
- Add this code to the end of the Main method:

```
Console.WriteLine("Listing blobs...");

// List all blobs in the container
await foreach (BlobItem blobItem in containerClient.GetBlobsAsync())
{
    Console.WriteLine("\t" + blobItem.Name);
}
```

- Now Save the File with File→Save
- Now again Build and Run the Code

  Dotnet Build

  Dotnet Run



7 Download blobs

- Add this code to the end of the Main method:

```
// Download the blob to a local file
// Append the string "DOWNLOADED" before the .txt extension
// so you can compare the files in the data directory
string downloadFilePath = localFilePath.Replace(".txt", "DOWNLOADED.txt");
```

```
Console.WriteLine("\nDownloading blob to\n\t{0}\n", downloadFilePath);

// Download the blob's contents and save it to a file
BlobDownloadInfo download = await blobClient.DownloadAsync();

using (FileStream downloadFileStream = File.OpenWrite(downloadFilePath))
{
   await download.Content.CopyToAsync(downloadFileStream);
   downloadFileStream.Close();
}
```

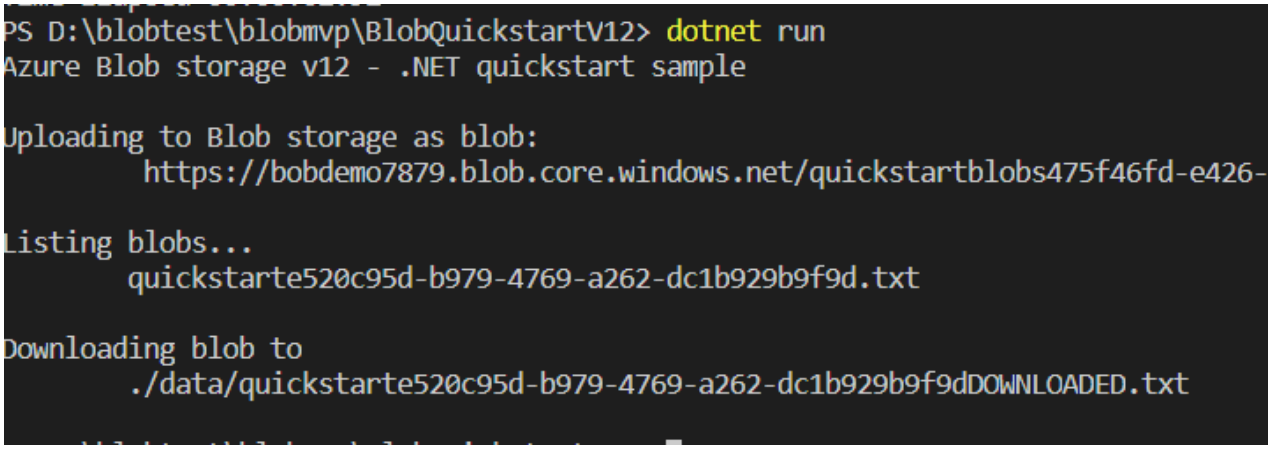
Program.cs

```
44                  await foreach (BlobItem blobItem in containerClient.GetBlobsAsync())
45                  {
46                      Console.WriteLine("\t" + blobItem.Name);
47                  }
48                      // Download the blob to a local file
49                  // Append the string "DOWNLOADED" before the .txt extension
50                  // so you can compare the files in the data directory
51                  string downloadFilePath = localFilePath.Replace(".txt", "DOWNLOADED.txt");
52
53                  Console.WriteLine("\nDownloading blob to\n\t{0}\n", downloadFilePath);
54
55                  // Download the blob's contents and save it to a file
56                  BlobDownloadInfo download = await blobClient.DownloadAsync();
57
58                  using (FileStream downloadFileStream = File.OpenWrite(downloadFilePath))
59                  {
60                      await download.Content.CopyToAsync(downloadFileStream);
61                      downloadFileStream.Close();
62                  }
```

- Now Save the File with File→Save
- Now again Build and Run the Code

Dotnet Build

Dotnet Run



8 Delete the Container

- Add this code to the end of the Main method:

```
// Clean up
Console.Write("Press any key to begin clean up");
Console.ReadLine();

Console.WriteLine("Deleting blob container...");
await containerClient.DeleteAsync();

Console.WriteLine("Deleting the local source and downloaded files...");
File.Delete(localFilePath);
File.Delete(downloadFilePath);

Console.WriteLine("Done");
```

- Now Save the File with File→Save
- Now again Build and Run the Code

Dotnet Build

Dotnet Run

9 Work with Sample Application

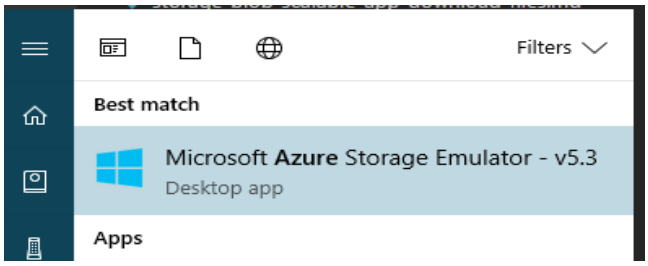
Decide on a name for the new container. The code below appends a GUID value to the container name to ensure that it is unique.

B. Now Type below Git URL to Clone (You Must have **git** installed in your PC from *https://git-scm.com/download/win*)

git clone https://github.com/ShrutiSinhaa/TutorialForStorage.git

```
D:\blobtest>git clone https://github.com/ShrutiSinhaa/TutorialForStorage.git
Cloning into 'TutorialForStorage'...
Receiving objects:  55% (118/213), 884.0
remote: Total 213 (delta 0), reused 0 (delta 0), pack-reused 213
Receiving objects: 100% (213/213), 6.32 MiB | 1.92 MiB/s, done.
Resolving deltas: 100% (68/68), done.
```

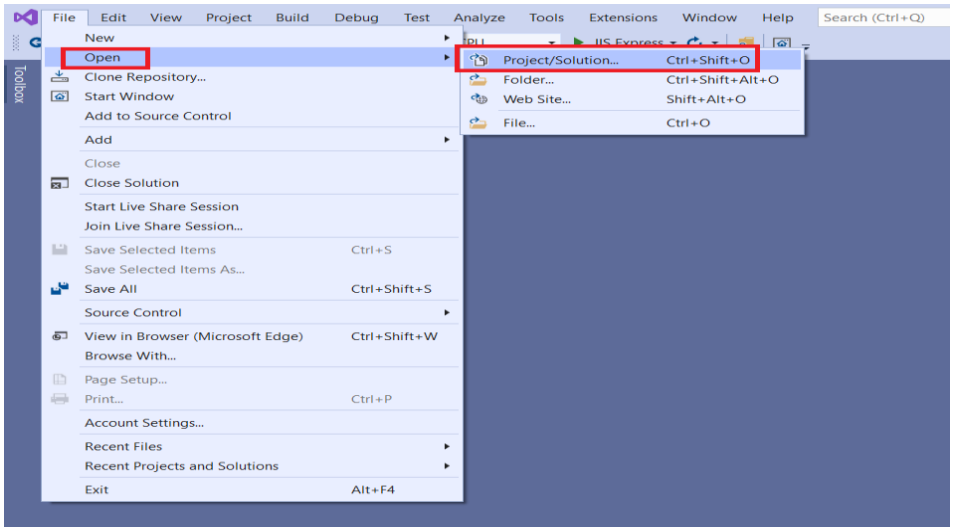
C. First, start the Azure Storage Emulator



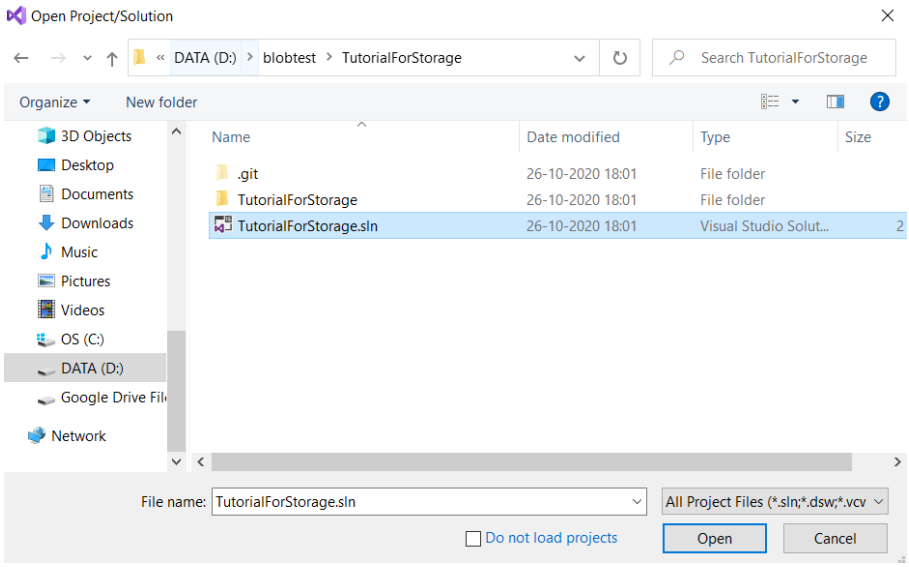
D. It will launch a command prompt

```
C:\Program Files (x86)\Microsoft SDKs\Azure\Storage Emulator>AzureStorageEmulator.exe start
Windows Azure Storage Emulator 5.10.0.0 command line tool
Autodetect requested. Autodetecting SQL Instance to use.
Looking for a LocalDB Installation.
Probing SQL Instance: '(localdb)\MSSQLLocalDB'.
Found a LocalDB Installation.
Probing SQL Instance: '(localdb)\MSSQLLocalDB'.
Found SQL Instance (localdb)\MSSQLLocalDB.
Creating database AzureStorageEmulatorDb510 on SQL instance '(localdb)\MSSQLLocalDB'.

Granting database access to user DESKTOP-BE5SIHJ\TechPledge.
Database access for user DESKTOP-BE5SIHJ\TechPledge was granted.

Initialization successful. The storage emulator is now ready for use.
The storage emulator was successfully started.

C:\Program Files (x86)\Microsoft SDKs\Azure\Storage Emulator>cmd /K AzureStorageEmulator.exe help
Windows Azure Storage Emulator 5.10.0.0 command line tool
Usage:
    AzureStorageEmulator.exe init            : Initialize the emulator database and configuration.
    AzureStorageEmulator.exe start           : Start the emulator.
    AzureStorageEmulator.exe stop            : Stop the emulator.
    AzureStorageEmulator.exe status          : Get current emulator status.
    AzureStorageEmulator.exe clear           : Delete all data in the emulator.
    AzureStorageEmulator.exe help [command]  : Show general or command-specific help.

See the following URL for more command line help: http://go.microsoft.com/fwlink/?LinkId=392235
```

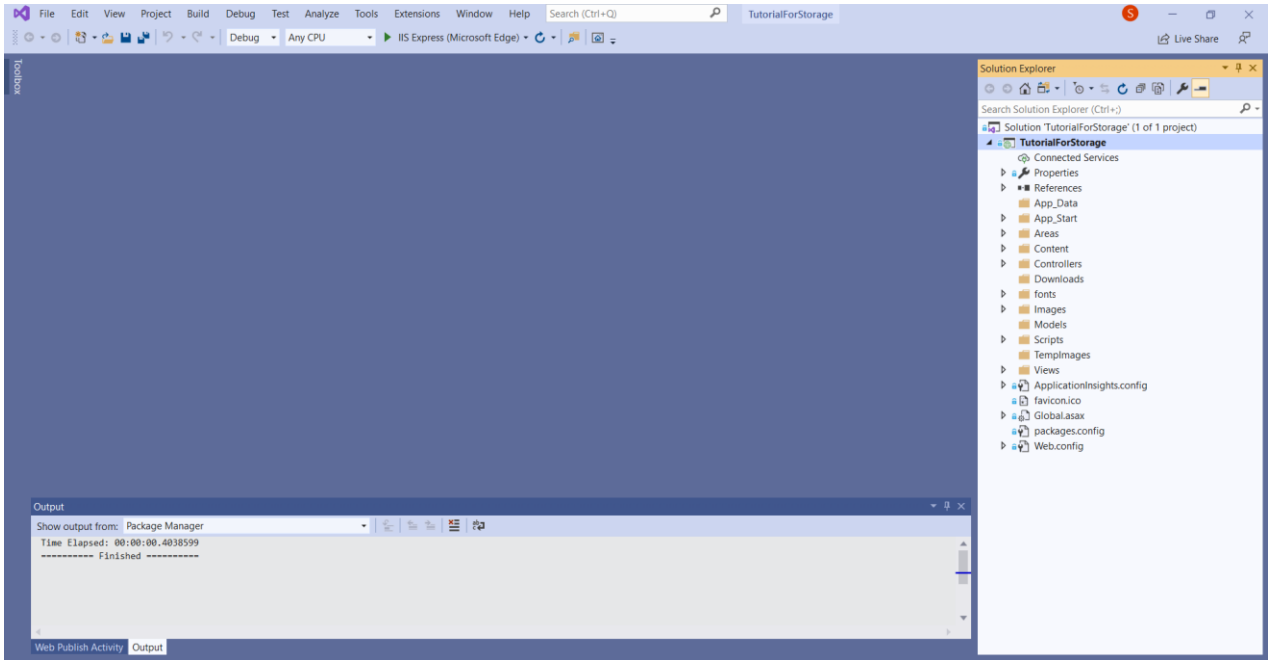
E. It will launch a command prompt



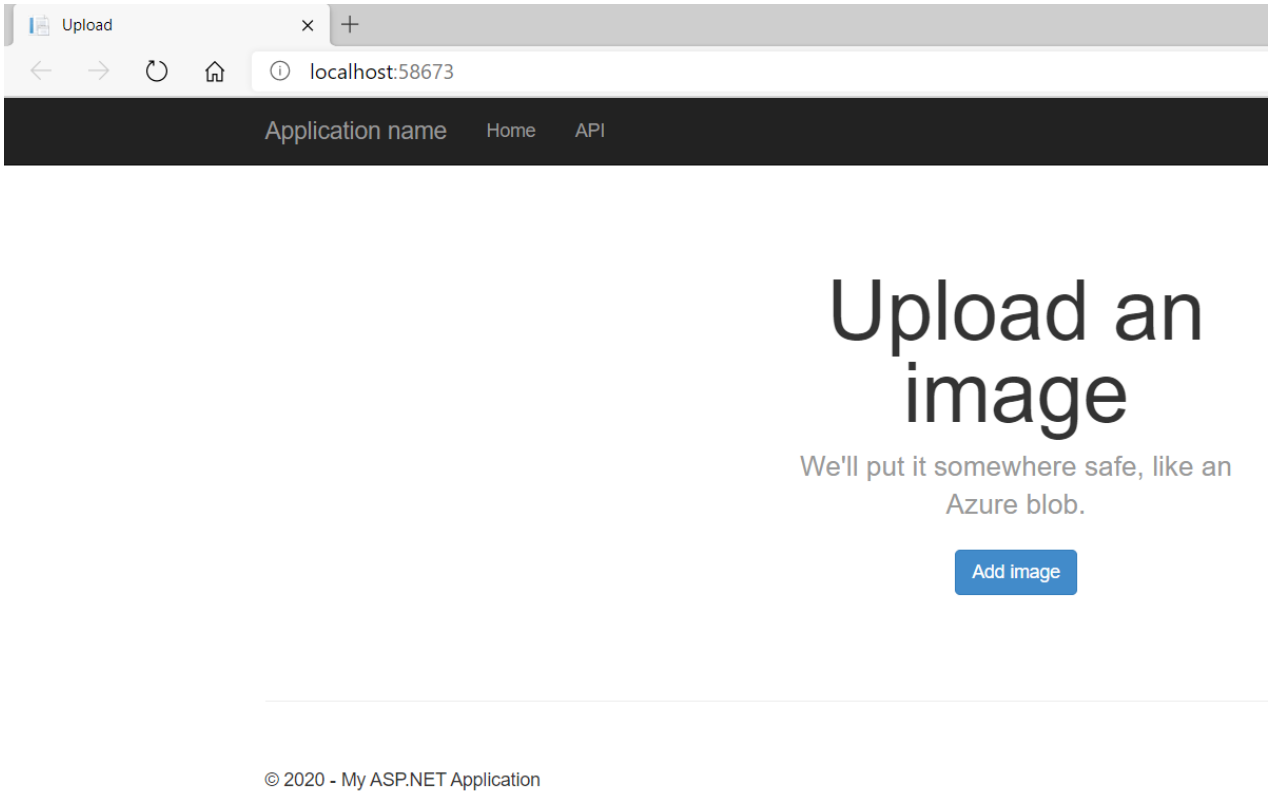
F. It will launch a command prompt



Start it by pressing **f5** in Visual Studio



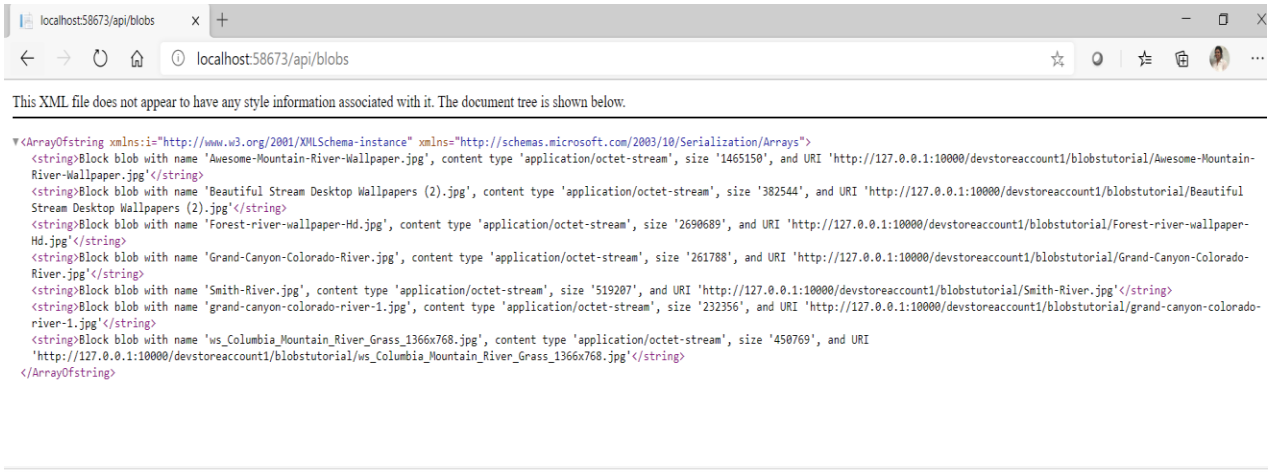
This will launch the web app on `http://localhost:58673/` so that you can interact with it

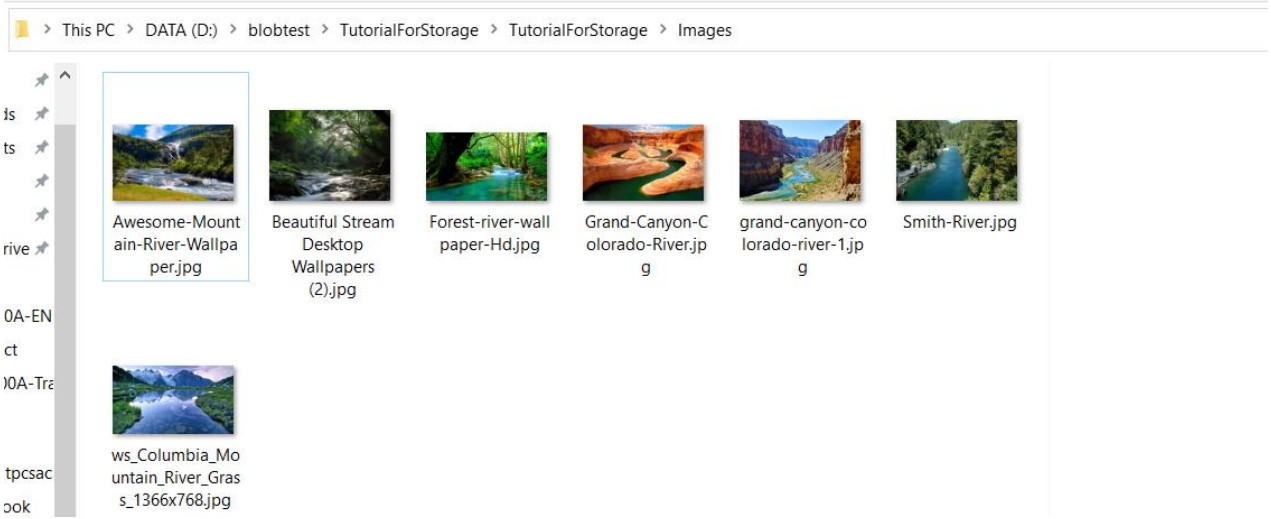


Next, enter the following in your browser to test that your GET API is working:

`http://localhost:58673/api/blobs`

```
It will display the list of all the Blobs store
```


localhost:58673/api/blobs
localhost:58673/api/blobs
This XML file does not appear to have any style information associated with it. The document tree is shown below.
<ArrayOfstring xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
<string>Block blob with name 'Awesome-Mountain-River-Wallpaper.jpg', content type 'application/octet-stream', size '1465150', and URI 'http://127.0.0.1:10000/devstoreaccount1/blobstutorial/Awesome-Mountain-River-Wallpaper.jpg'</string>
<string>Block blob with name 'Beautiful Stream Desktop Wallpapers (2).jpg', content type 'application/octet-stream', size '382544', and URI 'http://127.0.0.1:10000/devstoreaccount1/blobstutorial/Beautiful Stream Desktop Wallpapers (2).jpg'</string>
<string>Block blob with name 'Forest-river-wallpaper-Hd.jpg', content type 'application/octet-stream', size '2690689', and URI 'http://127.0.0.1:10000/devstoreaccount1/blobstutorial/Forest-river-wallpaper-Hd.jpg'</string>
<string>Block blob with name 'Grand-Canyon-Colorado-River.jpg', content type 'application/octet-stream', size '261788', and URI 'http://127.0.0.1:10000/devstoreaccount1/blobstutorial/Grand-Canyon-Colorado-River.jpg'</string>
<string>Block blob with name 'Smith-River.jpg', content type 'application/octet-stream', size '519207', and URI 'http://127.0.0.1:10000/devstoreaccount1/blobstutorial/Smith-River.jpg'</string>
<string>Block blob with name 'grand-canyon-colorado-river-1.jpg', content type 'application/octet-stream', size '232356', and URI 'http://127.0.0.1:10000/devstoreaccount1/blobstutorial/grand-canyon-colorado-river-1.jpg'</string>
<string>Block blob with name 'ws_Columbia_Mountain_River_Grass_1366x768.jpg', content type 'application/octet-stream', size '450769', and URI 'http://127.0.0.1:10000/devstoreaccount1/blobstutorial/ws_Columbia_Mountain_River_Grass_1366x768.jpg'</string>
</ArrayOfstring>


This PC
DATA (D:)
blobtest
TutorialForStorage
TutorialForStorage
Images
Awesome-Mountain-River-Wallpaper.jpg
Beautiful Stream Desktop Wallpapers (2).jpg
Forest-river-wallpaper-Hd.jpg
Grand-Canyon-Colorado-River.jpg
grand-canyon-colorado-river-1.jpg
Smith-River.jpg
ws_Columbia_Mountain_River_Grass_1366x768.jpg