

آموزش MVC Core – Azure Blob Storage In ASP.NET Core

2.0

Problem

نحوه استفاده از ذخیره سازی Azure Blob در ASP.NET Core.

Solution

یک class library ایجاد کنید و WindowsAzure.Storage - NuGet package را اضافه کنید.

یک کلاس برای تنظیم کردن تنظیمات اضافه کنید:

```
public class AzureBlobSettings
{
    public AzureBlobSettings(string storageAccount,
        string storageKey,
        string containerName)
    {
        if (string.IsNullOrEmpty(storageAccount))
            throw new ArgumentNullException("StorageAccount");

        if (string.IsNullOrEmpty(storageKey))
            throw new ArgumentNullException("StorageKey");

        if (string.IsNullOrEmpty(containerName))
            throw new ArgumentNullException("ContainerName");

        this.StorageAccount = storageAccount;
        this.StorageKey = storageKey;
        this.ContainerName = containerName;
    }

    public string StorageAccount { get; }
    public string StorageKey { get; }
    public string ContainerName { get; }
}
```

یک کلاس برای تنظیم کردن blob item اضافه کنید:

```
public class AzureBlobItem
{
    public AzureBlobItem(IListBlobItem item)
    {
        this.Item = item;
    }

    public IListBlobItem Item { get; }

    public bool IsBlockBlob => Item.GetType() == typeof(CloudBlockBlob);
}
```

```

public bool IsPageBlob => Item.GetType() == typeof(CloudPageBlob);
public bool IsDirectory => Item.GetType() == typeof(CloudBlobDirectory);

public string BlobName => IsBlockBlob ? ((CloudBlockBlob)Item).Name :
    IsPageBlob ? ((CloudPageBlob)Item).Name :
    IsDirectory ? ((CloudBlobDirectory)Item).Prefix :
    "";

public string Folder => BlobName.Contains("/") ?
    BlobName.Substring(0, BlobName.LastIndexOf("/")) : "";

public string Name => BlobName.Contains("/") ?
    BlobName.Substring(BlobName.LastIndexOf("/") + 1) : BlobName;
}

```

private helper یک کلاس برای دسترسی به ذخیره سازی اضافه کنید. برای دسترسی به ذخیره سازی یک **methods** را اضافه کنید.

```

private async Task<CloudBlobContainer> GetContainerAsync()
{
    //Account
    CloudStorageAccount storageAccount = new CloudStorageAccount(
        new StorageCredentials(settings.StorageAccount, settings.StorageKey), false);

    //Client
    CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();

    //Container
    CloudBlobContainer blobContainer =
blobClient.GetContainerReference(settings.ContainerName);
    await blobContainer.CreateIfNotExistsAsync();

    return blobContainer;
}

private async Task<CloudBlockBlob> GetBlockBlobAsync(string blobName)
{
    //Container
    CloudBlobContainer blobContainer = await GetContainerAsync();

    //Blob
    CloudBlockBlob blockBlob = blobContainer.GetBlockBlobReference(blobName);

    return blockBlob;
}

private async Task<List<AzureBlobItem>> GetBlobListAsync(
bool useFlatListing = true)
{
    //Container
    CloudBlobContainer blobContainer = await GetContainerAsync();

```

```

//List
var list = new List<AzureBlobItem>();
BlobContinuationToken token = null;
do
{
    BlobResultSegment resultSegment =
        await blobContainer.ListBlobsSegmentedAsync("", useFlatListing,
new BlobListingDetails(), null, token, null, null);
    token = resultSegment.ContinuationToken;

    foreach (IListBlobItem item in resultSegment.Results)
    {
        list.Add(new AzureBlobItem(item));
    }
} while (token != null);

return list.OrderBy(i => i.Folder).ThenBy(i => i.Name).ToList();
}

```

می‌توانیم اکنون برای upload و download (موارد blob items) از public methods ها استفاده کنیم:

```

public async Task UploadAsync(string blobName, string filePath)
{
    //Blob
    CloudBlockBlob blockBlob = await GetBlockBlobAsync(blobName);

    //Upload
    using (var fileStream = System.IO.File.Open(filePath, FileMode.Open))
    {
        fileStream.Position = 0;
        await blockBlob.UploadFromStreamAsync(fileStream);
    }
}

public async Task UploadAsync(string blobName, Stream stream)
{
    //Blob
    CloudBlockBlob blockBlob = await GetBlockBlobAsync(blobName);

    //Upload
    stream.Position = 0;
    await blockBlob.UploadFromStreamAsync(stream);
}

public async Task<MemoryStream> DownloadAsync(string blobName)
{
    //Blob
    CloudBlockBlob blockBlob = await GetBlockBlobAsync(blobName);

    //Download

```

```

using (var stream = new MemoryStream())
{
    await blockBlob.DownloadToStreamAsync(stream);
    return stream;
}
}

public async Task DownloadAsync(string blobName, string path)
{
    //Blob
    CloudBlockBlob blockBlob = await GetBlockBlobAsync(blobName);

    //Download
    await blockBlob.DownloadToFileAsync(path, FileMode.Create);
}

```

متدی برای گرفتن لیستی از blob items را اضافه می کنیم:

```

public async Task<List<AzureBlobItem>> ListAsync()
{
    return await GetBlobListAsync();
}

public async Task<List<string>> ListFoldersAsync()
{
    var list = await GetBlobListAsync();
    return list.Where(i => !string.IsNullOrEmpty(i.Folder))
        .Select(i => i.Folder)
        .Distinct()
        .OrderBy(i => i)
        .ToList();
}

```

تزریق(Inject) و استفاده از storage helper :

```

public class HomeController : Controller
{
    private readonly IAzureBlobStorage blobStorage;

    public HomeController(IAzureBlobStorage blobStorage)
    {
        this.blobStorage = blobStorage;
    }
}

```

Note

این کدها دارای یک کنترلر با اکشن هایی برای listing ، downloading و uploading است.

در ASP.NET Core Web Application ، configure services :

```

public void ConfigureServices(
    IServiceCollection services)

```

```
{
  services.AddScoped<IAzureBlobStorage>(factory =>
  {
    return new AzureBlobStorage(new AzureBlobSettings(
      storageAccount: Configuration["Blob_StorageAccount"],
      storageKey: Configuration["Blob_StorageKey"],
      containerName: Configuration["Blob_ContainerName"]));
  });

  services.AddMvc();
}
```

Discussion

کد نمونه نیاز به تنظیم یک Azure account ، blob storage account و container دارد.

شما می توانید Source Code از لینک زیر دانلود کنید:

<https://github.com/TahirNaushad/Fiver.Azure.Blob>