


✓ 100 XP 

# Interact with Azure Content Delivery Networks by using .NET

3 minutes

You can use the Azure CDN Library for .NET to automate creation and management of CDN profiles and endpoints. Install the `Microsoft.Azure.Management.Cdn` directly from the Visual Studio Package Manager console or with the .NET CLI.

In this unit, you see code examples illustrating common actions.

## Create a CDN client

The following example shows creating a client by using the `CdnManagementClient` class.

C#

```
static void Main(string[] args)
{
    // Create CDN client
    CdnManagementClient cdn = new CdnManagementClient(new
    TokenCredentials(authResult.AccessToken))
        { SubscriptionId = subscriptionId };
}
```

## List CDN profiles and endpoints

The following method lists all the profiles and endpoints in our resource group. If it finds a match for the profile and endpoint names specified in our constants it notes it for later so we don't try to create duplicates.

C#

```
private static void ListProfilesAndEndpoints(CdnManagementClient cdn)
{
    // List all the CDN profiles in this resource group
```

```

var profileList = cdn.Profiles.ListByResourceGroup(resourceGroupName);
foreach (Profile p in profileList)
{
    Console.WriteLine("CDN profile {0}", p.Name);
    if (p.Name.Equals(profileName, StringComparison.OrdinalIgnoreCase))
    {
        // Hey, that's the name of the CDN profile we want to create!
        profileAlreadyExists = true;
    }

    //List all the CDN endpoints on this CDN profile
    Console.WriteLine("Endpoints:");
    var endpointList = cdn.Endpoints.ListByProfile(p.Name, resourceGroupName);
    foreach (Endpoint e in endpointList)
    {
        Console.WriteLine("-{0} ({1})", e.Name, e.HostName);
        if (e.Name.Equals(endpointName, StringComparison.OrdinalIgnoreCase))
        {
            // The unique endpoint name already exists.
            endpointAlreadyExists = true;
        }
    }
    Console.WriteLine();
}
}

```

## Create CDN profiles and endpoints

The following example shows creating an Azure CDN profile.

C#

```
private static void CreateCdnProfile(CdnManagementClient cdn)
{
    if (profileAlreadyExists)
    {
        Console.WriteLine("Profile {0} already exists.", profileName);
    }
    else
    {
        Console.WriteLine("Creating profile {0}.", profileName);
        ProfileCreateParameters profileParms =
            new ProfileCreateParameters() { Location = resourceLocation, Sku = new
            Sku(SkuName.StandardVerizon) };
        cdn.Profiles.Create(profileName, profileParms, resourceGroupName);
    }
}
```

Once the profile is created, we create an endpoint.

C#

```
private static void CreateCdnEndpoint(CdnManagementClient cdn)
{
    if (endpointAlreadyExists)
    {
        Console.WriteLine("Profile {0} already exists.", profileName);
    }
    else
    {
        Console.WriteLine("Creating endpoint {0} on profile {1}.", endpointName,
        profileName);
        EndpointCreateParameters endpointParms =
            new EndpointCreateParameters()
            {
                Origins = new List<DeepCreatedOrigin>() { new
                DeepCreatedOrigin("Contoso", "www.contoso.com") },
                IsHttpAllowed = true,
                IsHttpsAllowed = true,
                Location = resourceLocation
            };
        cdn.Endpoints.Create(endpointName, endpointParms, profileName,
        resourceGroupName);
    }
}
```

## Purge an endpoint

A common task that we might want to perform is purging the content in our endpoint.

C#

```
private static void PromptPurgeCdnEndpoint(CdnManagementClient cdn)
{
    if (PromptUser(String.Format("Purge CDN endpoint {0}?", endpointName)))
    {
        Console.WriteLine("Purging endpoint. Please wait...");
        cdn.Endpoints.PurgeContent(resourceGroupName, profileName, endpointName,
new List<string>() { "/"*"});
        Console.WriteLine("Done.");
        Console.WriteLine();
    }
}
```

---

## Next unit: Knowledge check

[Continue >](#)

---