In this video, we will discuss creating **CacheProfiles**. We will be using the example that we started in [Part 73](http://csharp-video-tutorials.blogspot.com/2013/08/part-73-outputcache-attribute-in-mvc.html). Please watch [Part 73](http://csharp-video-tutorials.blogspot.com/2013/08/part-73-outputcache-attribute-in-mvc.html) before proceeding.   
  
   
  
To cache the data returned by **Index**() action method, for 60 seconds, we would use **[OutputCache]** attribute as shown below.  
[OutputCache(Duration=60)]  
public ActionResult Index()  
{  
    return View(db.Employees.ToList());  
}  
  
   
  
In the example above, the **OutputCache**settings are specified in code. The disadvantage of this approcah is that  
1. If you have to apply the same cache settings, to several methods, then the code needs to be duplicated.  
2. Later, if we have to change these cache settings, then we need to change them at several places. Maintaining the code becomes complicated. Also, changing the application code requires build and re-deployment.  
  
**To overcome these disadvantages**, cache settings can be specified in web.config file using cache profiles.  
**Step 1:** Specify cache settings in web.config using cache profiles  
<system.web>  
  <caching>  
    <outputCacheSettings>  
      <outputCacheProfiles>  
        <clear/>  
        <add name="1MinuteCache" duration="60" varyByParam="none"/>              
      </outputCacheProfiles>  
    </outputCacheSettings>  
  </caching>  
</system.web>  
  
**Step 2:** Reference the cache profile in application code  
[OutputCache(CacheProfile = "1MinuteCache")]  
public ActionResult Index()  
{  
    return View(db.Employees.ToList());  
}  
  
**The cache settings are now read from one central location i.e from the web.config file. The advantage of using cache profiles is that**  
**1.** You have one place to change the cache settings. Mantainability is much easier.

**2.** Since the changes are done in web.config, we need not build and redeploy the application.  
  
**Using Cache profiles with child action methods**  
[ChildActionOnly]  
[OutputCache(CacheProfile = "1MinuteCache")]  
public string GetEmployeeCount()  
{  
    return "Employee Count = " + db.Employees.Count().ToString()   
        + "@ " + DateTime.Now.ToString();  
}  
  
When Cache profiles are used with child action methods, you will get an error - Duration must be a positive number.  
  
There colud be several ways to make cache profiles work with child action methods. The following is the approach, that I am aware of. Please feel free to leave a comment, if you know of a better way of doing this.

Create a**custom OutputCache attribute**, that loads the cache settings from the cache profile in web.config.   
**Step 1:** Right click on the project name in solution explorer, and add a folder with name = Common  
  
**Setp 2:** Right click on **"Common"** folder and add a class file, with name = PartialCacheAttribute.cs   
  
**Step 3:** Copy and paste the following code. Notice that, I have named the custom **OutputCache**attribute as **PartialCacheAttribute**.  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.Mvc;  
using System.Web.Configuration;  
  
namespace MVCDemo.Common  
{  
    public class PartialCacheAttribute : OutputCacheAttribute  
    {  
        public PartialCacheAttribute(string cacheProfileName)  
        {  
            OutputCacheSettingsSection cacheSettings =                 (OutputCacheSettingsSection)WebConfigurationManager.GetSection("system.web/caching/outputCacheSettings");  
            OutputCacheProfile cacheProfile = cacheSettings.OutputCacheProfiles[cacheProfileName];  
            Duration = cacheProfile.Duration;  
            VaryByParam = cacheProfile.VaryByParam;  
            VaryByCustom = cacheProfile.VaryByCustom;  
        }  
    }  
}  
  
**Step 4:** Use PartialCacheAttribute on the child action method, and pass it the name of the cache profile in web.config. Please note that, PartialCacheAttribute is in MVCDemo.Common namespace.  
[ChildActionOnly]  
[PartialCache("1MinuteCache")]  
public string GetEmployeeCount()  
{  
    return "Employee Count = " + db.Employees.Count().ToString()   
        + "@ " + DateTime.Now.ToString();  
}