< Previous

Next >

ScriptBundle in ASP.NET MVC

We have learned how bundling technique works in ASP.NET MVC. Here, we will learn how to create a bundle of multiple JavaScript files in one http request.

ASP.NET MVC API includes ScriptBundle class that does JavaScript minification and bundling.

Open App_Start\BundleConfig.cs file in the MVC folders. The BundleConfig.cs file is created by MVC framework by default. You should write your all bundling code in the BundleConfig.RegisterBundles() method. (you can create your own custom class instead of using BundleConfig class, but it is recommended to follow standard practice.) The following code shows a portion of the RegisterBundles method.

Example: BundleConfig.RegisterBundle()

```
using System.Web;
using System.Web.Optimization;

public class BundleConfig
{
    public static void RegisterBundles(BundleCollection bundles)
    {
        // create an object of ScriptBundle and
        // specify bundle name (as virtual path) as constructor parameter
        ScriptBundle scriptBndl = new ScriptBundle("~/bundles/bootstrap");


        //use Include() method to add all the script files with their paths
        scriptBndl.Include(
                            "~/Scripts/bootstrap.js",
                            "~/Scripts/respond.js"
                          );


        //Add the bundle into BundleCollection
        bundles.Add(scriptBndl);

        BundleTable.EnableOptimizations = true;
    }
}
```

In the above example, we have created a bundle of two JavaScript files, bootstrap.js and respond.js using ScriptBundle for demo purposes.

1. First of all create an instance of ScriptBundle class by specifing the bundle name as a constructor parameter. This bundle name is a virtual path starting with ~/. You can give anything in virtual path but it's recommended to give a path that will be easy to identify as a bundle. Here, we have given "~/bundles/bootstrap" path, so that we can easily identify that this bundle includes bootstrap related files.
2. Use Include method to add one or more JS files into a bundle with its relative path after root path using ~ sign.
3. Final, add the bundle into BundleCollection instance, which is specified as a parameter in RegisterBundle() method.

4. Last, `BundleTable.EnableOptimizations = true` enables bundling and minification in debug mode. If you set it to false then it will not do bundling and minification.

You can also use IncludeDirectory method of bundle class to add all the files under particular directory as shown below.

ScriptBundle Example:

```
public static void RegisterBundles(BundleCollection bundles)
{
    bundles.Add(new ScriptBundle("~/bundles/scripts").IncludeDirectory("~/Scripts/","*.js",true));
}
```

Thus, you can create a bundle of JavaScript files using ScriptBundle. MVC framework invokes BundleConfig.RegisterBundle() method from the Application_Start event in Global.asax.cs file, so that it can add all the bundles into BundleCollection at the starting of an application.

Example: Invoke RegisterBundle() in Application_Start event

```
public class MvcApplication : System.Web.HttpApplication
{
    protected void Application_Start()
    {
        BundleConfig.RegisterBundles(BundleTable.Bundles);
    }
}
```

Using Wildcards

Sometime third party script files includes versions in a name of script file. So it is not advisable to changes the code whenever you upgrade the version of script file. With the use of wildcards, you don't have to specify a version of a script file. It automatically includes files with the version available.

For example, Jquery files includes the version in a name. So you can use {version} wildcard to pickup a version based on available version.

Example: Wildcard with bundle

```
public class BundleConfig
{
    public static void RegisterBundles(BundleCollection bundles)
    {
        bundles.Add(new ScriptBundle("~/bundles/jquery")
               .Include( "~/Scripts/jquery-{version}.js"));
    }
}
```

Now, it will pick up jquery file added in a project. If you have included jquery-1.7.1.js then it will render this file and when you upgrade jquery file to jquery-1.10.2.js then it will automatically render 1.10 version file without changing or compiling code.

Using CDN

You can also use Content Delivery Network to load script files. For example, you can load jquery library from CDN as shown below.

Example: Load files from CDN

```
public class BundleConfig
{
    public static void RegisterBundles(BundleCollection bundles)
    {
        var cdnPath = "http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.1.min.js";

        bundles.Add(new ScriptBundle("~/bundles/jquery", cdnPath)
               .Include( "~/Scripts/jquery-{version}.js"));
    }
}
```

In the above code, jquery will be requested from the CDN while in release mode and in the debug mode, jquery library will be loaded from a local source. Please note that you should have a fallback mechanism to deal with a CDN request failure.

Now, let's see how to use the bundle into a razor view.

Include ScriptBundle in Razor View

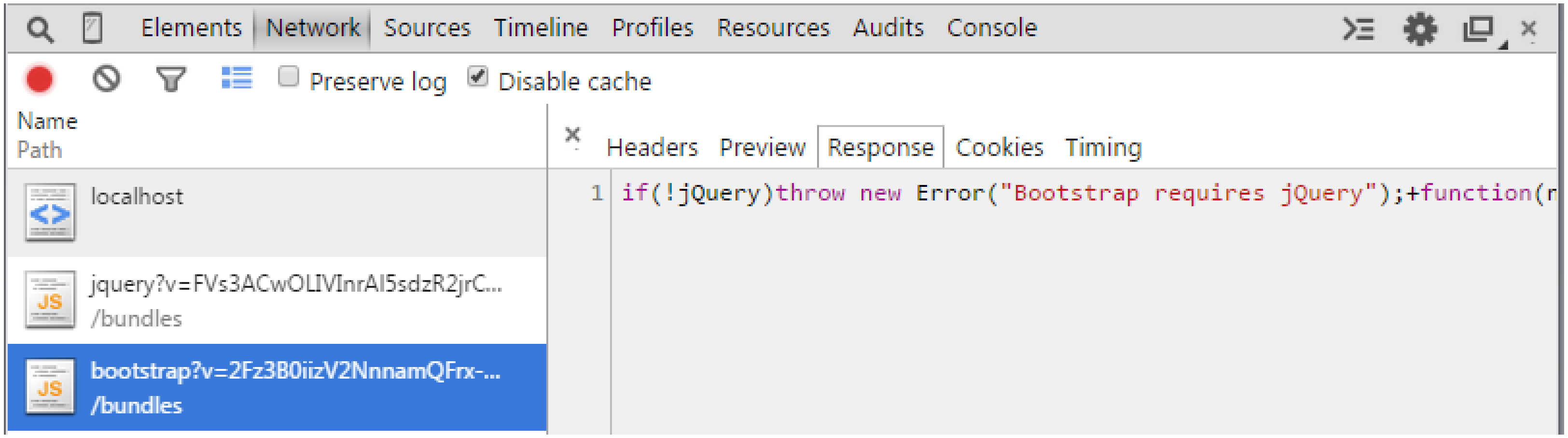
We have create a script bundle above. Now, we will learn how to include bundle into razor view.

The script bundles can be included using static Scripts class. Use Scripts.Render() method to include specified script bundle at runtime.

Example: Scripts.Render()

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title</title>
    @Scripts.Render("~/bundles/bootstrap")
</head>
<body>
    @*html code removed for clarity *@
</body>
</html>
```

Now, if you run the above example then you will find two script files is combined, minified and loaded in a single request. Please make sure that you have set debug = false in web.config `<compilation debug="false" targetFramework="4.5"/>`



☐ Preserve log ☑ Disable cache

localhost

jquery?v=FVs3ACwOLIVInrAI5sdzR2jrC...
/bundles

bootstrap?v=2Fz3B0iizV2NnnamQFrx-...
/bundles

```
if(!jQuery)throw new Error("Bootstrap requires jQuery");+function(r
```

Load Bundle in Browser

As you can see in the above figure that bootstrap bundle is loaded in a single request. It has also combined and minified two JS files for bootstrap.

Points to Remember :

1) Bundling and Minification minimize static script or css files loading time therby minimize page loading time.

2) ScriptBundle does minification of JavaScript files.

3) Create script or css bundles in BundleConfig class included in App_Start folder.

4) Use wildcard {version} to render available version files at runtime.

5) Use Scripts.Render("bundle name") method to include script bundle in a razor view.

f ▶ Share 🐦 ▶ Tweet
in ▶ Share ☎ ▶ Whatsapp

< Previous

Next >

TUTORIALSTEACHER.COM

TutorialsTeacher.com is optimized for learning web technologies step by step. Examples might be simplified to improve reading and basic understanding. While using this site, you agree to have read and accepted our terms of use and privacy policy.

✉ feedback@tutorialsteacher.com

TUTORIALS

- ASP.NET Core
- ASP.NET MVC
- IoC
- AngularJS 1
- Node.js
- D3.js

- Web API
- C#
- LINQ
- Entity Framework
- JavaScript
- jQuery
- Sass
- Https

E-MAIL LIST

Subscribe to TutorialsTeacher email list and get latest updates, tips & tricks on C#, .Net, JavaScript, jQuery, AngularJS, Node.js to your inbox.

Email address GO

We respect your privacy.

HOME PRIVACY POLICY TERMS OF USE ADVERTISE WITH US