**Environment tag helper in ASP.NET Core** with an example.  
  
  
**Requirement**

* We are using **Bootstrap**in our ASP.NET Core Application
* For ease of debugging, on our local development machine (i.e on the **Development Environment**) we want the application to load non-minified bootstrap css file (bootstrap.css)
* On **Staging**, **Production**or any other environment **except Development environment**we want the application to load minified bootstrap css file (bootstrap.min.css) from a **CDN**(Content Delivery Network) for better performance.
* However, if the **CDN is down**or for some reason, our application is not able to reach the CDN, we want our application to fallback and load the minified bootstrap file (bootstrap.min.css) from our own application web server.

We can achieve this very easily using ASP.NET Core <environment> tag helper. Before we understand the <environment> tag helper, first let's understand how we set the name of the application environment.  
  
**How to set Application Environment**  
  
Use the ***ASPNETCORE\_ENVIRONMENT*** variable to set the application environment. On our local development machine we usually set this environment variable in ***launchsettings.json*** file. On a staging or production environment it is set in the operating system. 

**ASP.NET Core Environment Tag Helper**  
  
***Environment tag helper*** supports rendering different content depending on the application environment. The application environment name is set using using ***ASPNETCORE\_ENVIRONMENT*** variable.  
  
This example loads the non-minified bootstrap css file, if the application environment is "Development"

In Layout.cshtml file we inclue the below code

<**environment** **include**="Development">

    <link href="~/lib/bootstrap/css/bootstrap.css" rel="stylesheet" />

</**environment**>

This example loads the minified bootstrap css file from the CDN (Content Delivery Network), if the application environment is "Staging" or "Production".

<**environment** **include**="Staging,Production">

    <link rel="stylesheet"

            href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"

            integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"

            crossorigin="anonymous">

</**environment**>

***"include"*** attribute accepts a single hosting environment name or a comma-separated list of hosting environment names. On the ***<environment>*** tag helper, we also have ***"exclude"*** attribute. The content of the ***<environment>*** tag is rendered when the hosting environment doesn't match an environment listed in the ***exclude*** attribute value.  
  
This example loads the minified bootstrap css file from the CDN (Content Delivery Network), if the application environment IS NOT "Development". 

<**environment** **exclude**="Development">

    <link rel="stylesheet"

            href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"

            integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"

            crossorigin="anonymous">

</**environment**>

The ***"integrity"*** attribute on the <link> element is used for **Subresource Integrity check**. Subresource Integrity (SRI for short), is a security feature that allows a browser to check if the file being retrieved has been maliciously altered. When the browser downloads the file, it recalculates the hash and compares it against the "integrity" attribute hash value. If the hash values match, the browser allows the file to be downloaded otherwise it is blocked.  
  
**What if the CDN is down**  
  
If the CDN is down or for some reason, our application is not able to reach the CDN, we want our application to fallback and load the minified bootstrap file (bootstrap.min.css) from our own application web server. Consider the following example

<**environment** **include**="Development">

    <link href="~/lib/bootstrap/css/bootstrap.css" rel="stylesheet" />

</**environment**>

<**environment** **exclude**="Development">

    <**link** rel="stylesheet"

            integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"

            crossorigin="anonymous"

**href**="https://sstackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"

**asp-fallback-href**="~/lib/bootstrap/css/bootstrap.min.css"

**asp-fallback-test-class**="sr-only" **asp-fallback-test-property**="position"

**asp-fallback-test-value**="absolute"

**asp-suppress-fallback-integrity**="true" />

</**environment**>

If the application environment is "Development", non-minified bootstrap css file (bootstrap.css) is loaded from our application web server  
  
If the application environment IS NOT "Development", minified bootstrap css file (bootstrap.min.css) is loaded from the CDN  
  
A fallback source is specified using asp-fallback-href attribute. This means, if the CDN is down, our application fallsback and load the minified bootstrap file (bootstrap.min.css) from our own application web server.  
  
The following 3 attributes and their associated values are used to check if the CDN is down  
***asp-fallback-test-class="sr-only"***  
***asp-fallback-test-property="position"***  
***asp-fallback-test-value="absolute"***  
  
Obviously, there is some processing involved to calculate hash and compare it with the integrity attribute hash value. For most applications, fallback source is their own server. You have the option to turn off integrity check for the files downloaded from the fallback source by setting ***asp-suppress-fallback-integrity***attribute to ***true***.