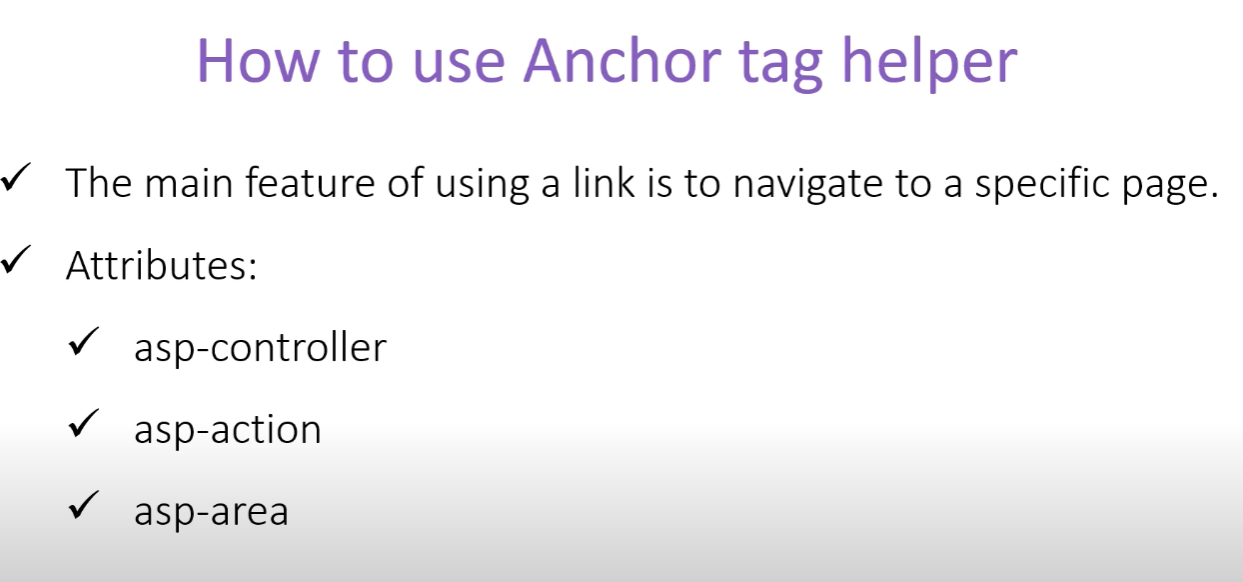
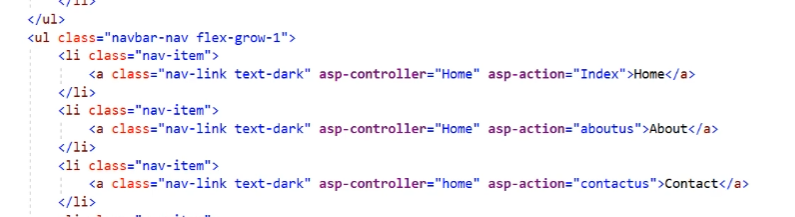


**Anchor Tag Helper**

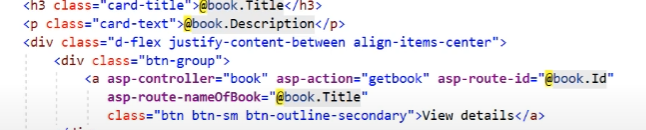
Using default href inside anchor tag works perfectly alright, but if there arise a scenario where we want to change the foldter structure (moving the whole folder inside another), we need to manually update all the links inside every href. For situation like this tag helpers fix this problem. This takes away all our work for deployment and stuff. We just need to change the MapControllerRoute method in case we make any change in directory.

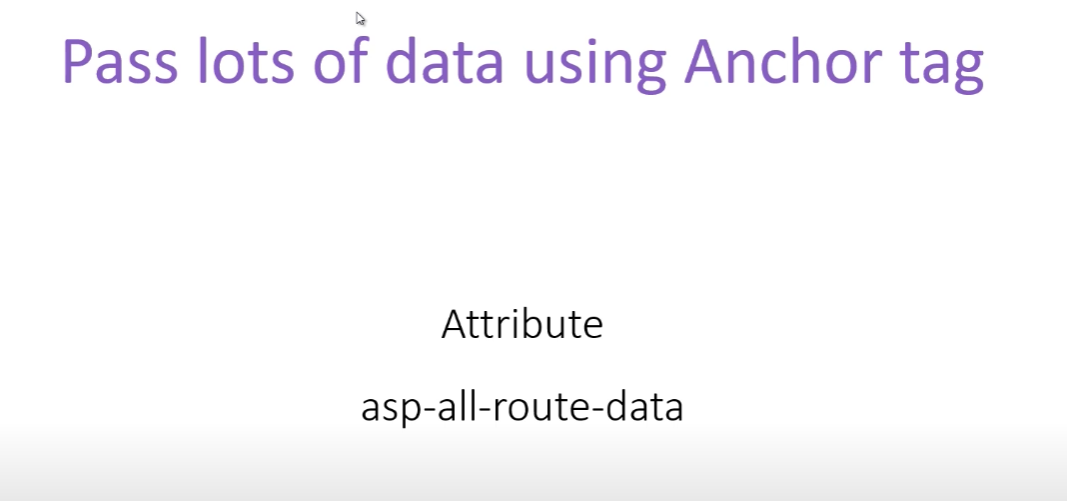


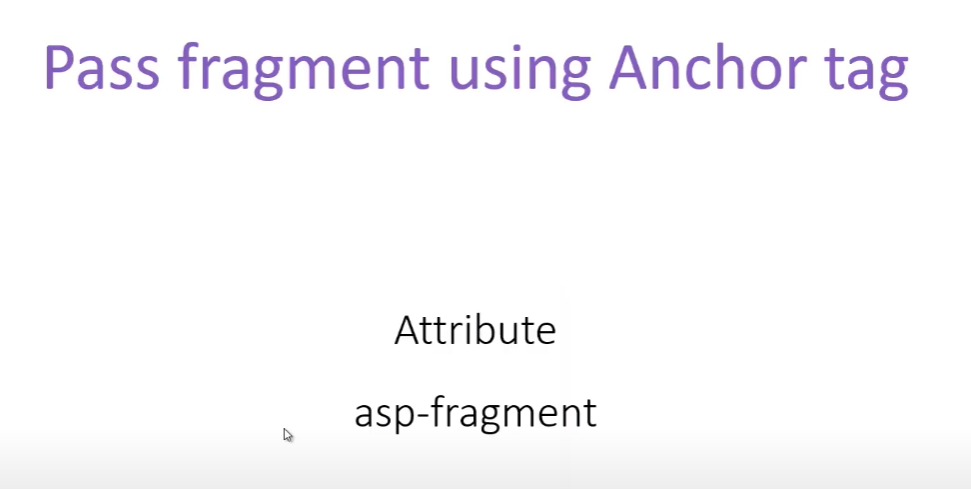
Inside the anchor tag, we need to define the controller as well as the action with taghelpers,

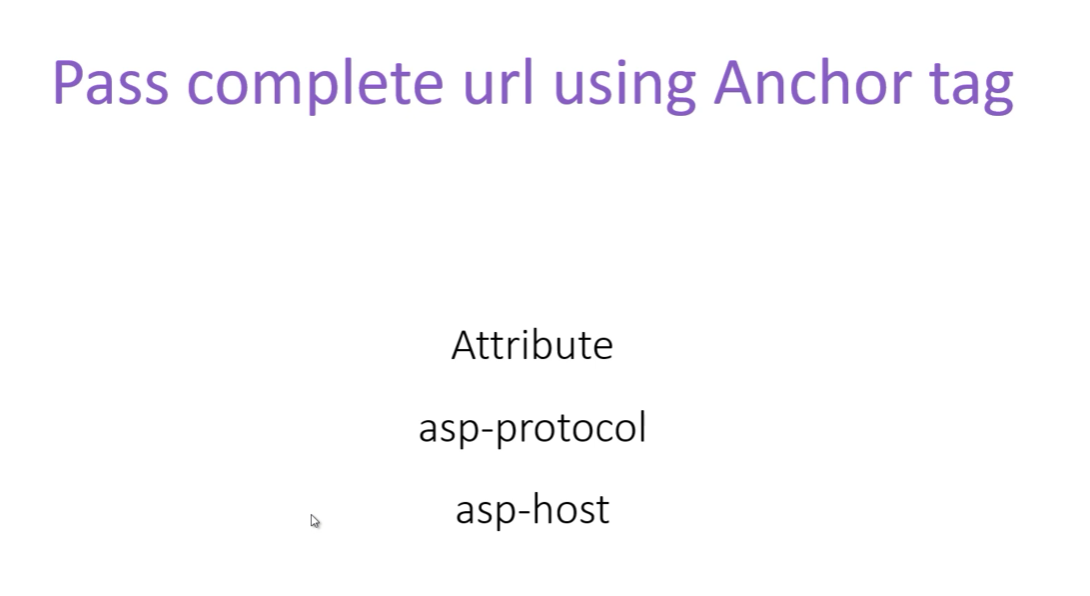


We can pass attribute as well with the routes (Like we used variables in href within {} tag).here is an example of passing id and nameofBook using taghelper

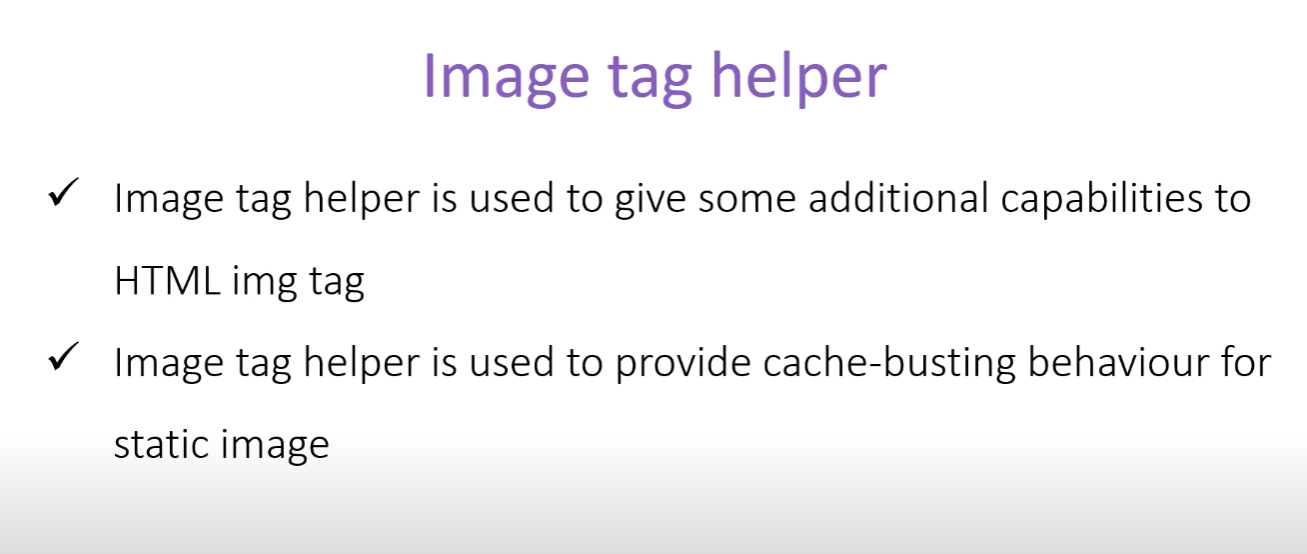






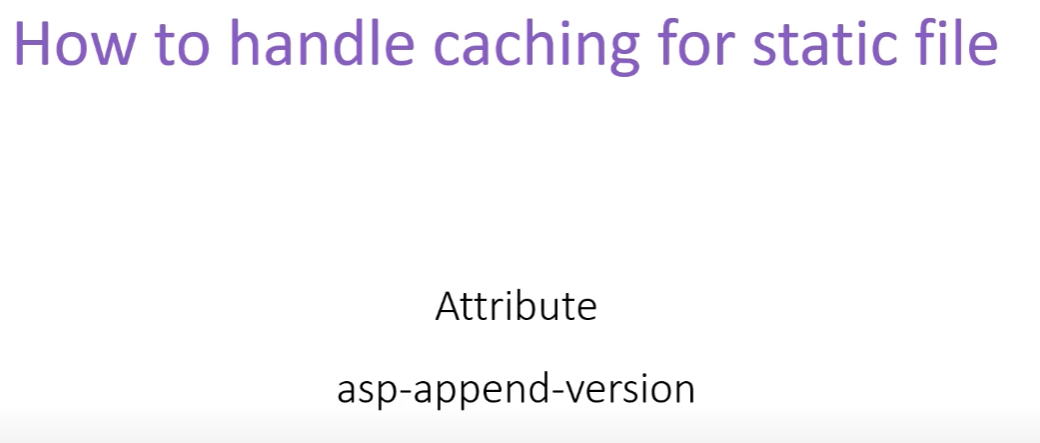


**Image Tag Helper**



When any particular image gets loaded in the browser, then browser cache the image. So when the next time the image is requested to be loaded, it will be loaded directly from the browser cache>

But this creates a small problem as well, since browser doesn’t request the server for image and get it from cache, if we change the image in the server side it wont get updated in the client side.

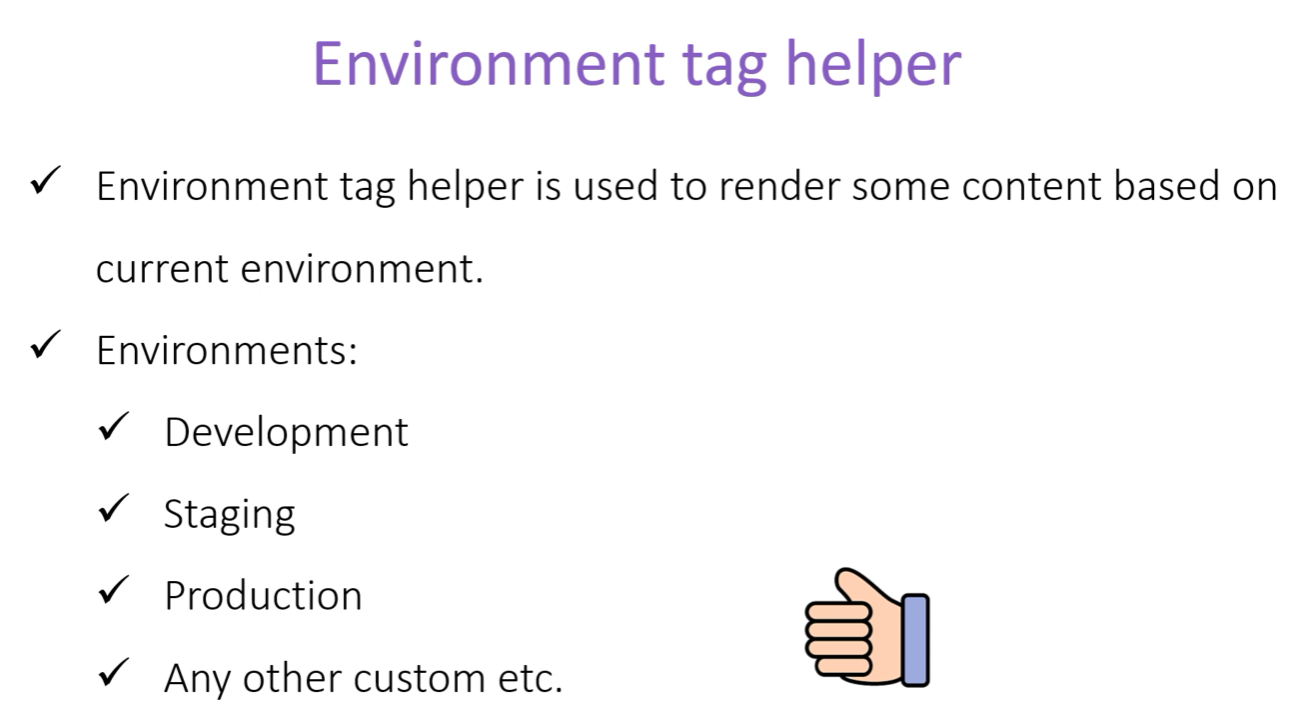




Once we use the asp-append-version to true, it will have a hashed version key append to the image. So, this solves our problem, cause each browser saves the image with a URL name. If an image URL is changed then the browser reload the image from server, otherwise it will reload from cache.

Another important thing is, image tag helpers only work if we are accessing the image from the server, not from any 3rd party hosting.

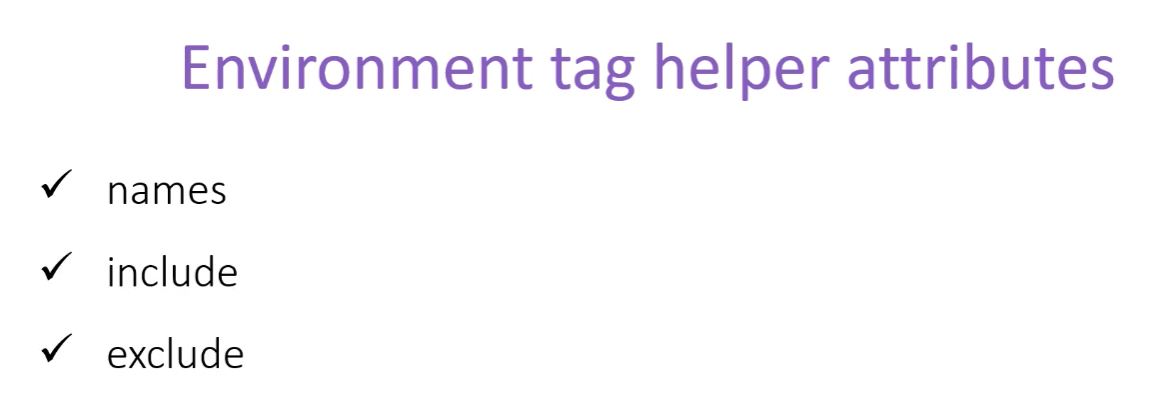
**Environment Tag Helper**



We have minified version of bootstrap as well as non-minified version. They both have their advantage and disadvantage. While minified version is smaller in size, we can’t debug with this.

So, to solve this problem, we can use specify when to load which file. For example, for production we can use the minified version, and for development we can use non-minified.

It has three attributes,

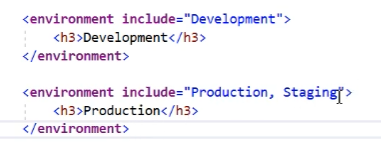


To show two different data depending on the environment, we can do the following,

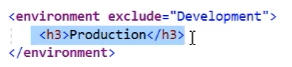


We can also put multiple environment variable names if we want to do same thing for multiple environments.

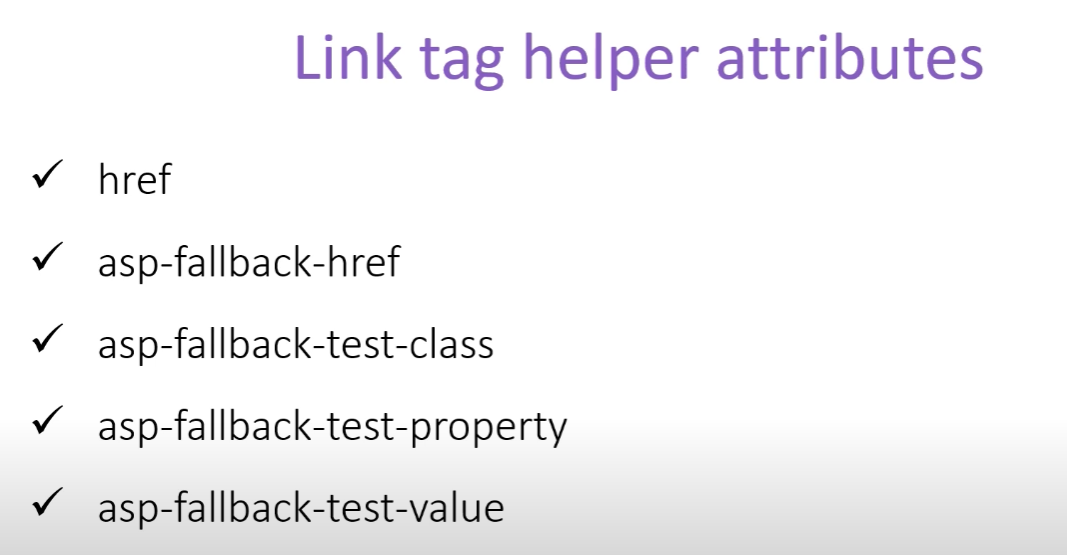
If we replace name with include it will work same,

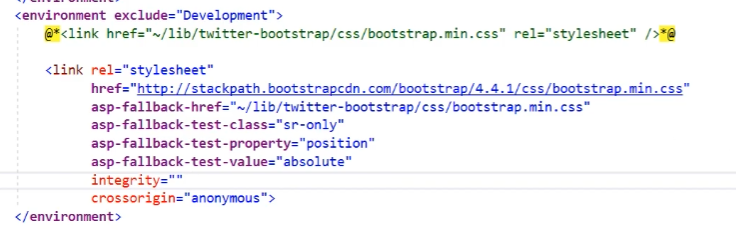


Exclude is the opposite of the include, it will work all environment other than the one specified.



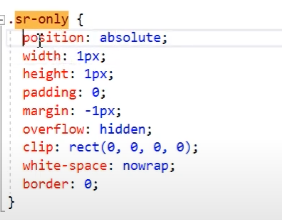
**Link Tag Helper**





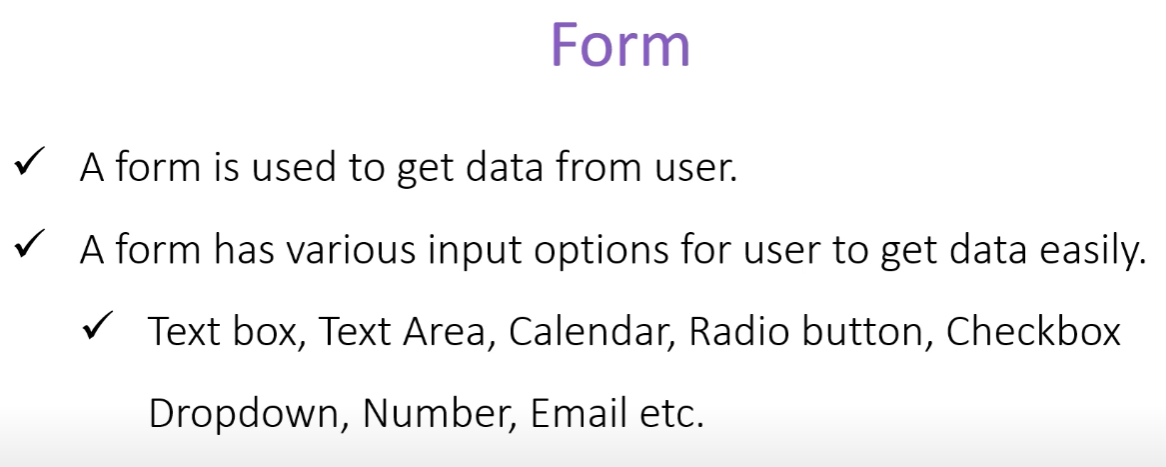
Fallback-href is used when the href content is not available for some reason. If the href content is available, it wont be used.

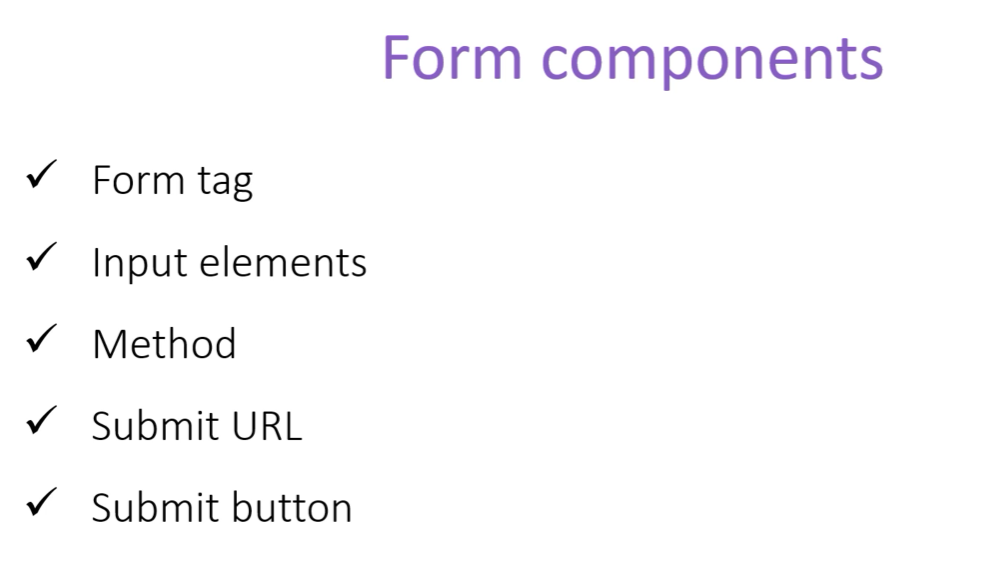
The rest of the properties are used to test, for example a bootstrap class here



If the mentioned property from the tag-helpers matches the cdn version, then it will load from the cdn. Otherwise it will load from the local version.

**Form Tag Helper**





Form **method** : **delete** is when we want to delete something, **get** is when we want to get some data(for example searching),**post** is when we want to add something new and if we want to update then we can use **put**.

When we create a form, asp.net automatically creates a request verification token for us. This is used for security purpose.

To use a form tag helpers, we need to use the @model directive and specify which model we are using . Then we can use asp-for tag helper in the input/label and asp.net will automatically assign the input field value to the model property when we submit the form. But if the data is not same name, it will not be mapped at all.

To handle the post request in the action method, we have to add the [HttpPost] decorator here. By default the action methods are configured to handle the get requests. We can change this by using [HtppPut/delete] etc with another action method with same name.

In case we don’t use a same name action or controller , we can specify them in the form tag, using the asp-action and asp-controller.

