# AutoValidateAntiForgeryToken vs. ValidateAntiForgeryToken

13

I was trying to secure a post method against side scripting just now through providing an anti forgery token but noticed, in .Net Core there is another attribute named as `AutoAntiForgeryToken`. The XML comments, and the online search did not provide much info on this new attribute.

Any help and description of what the new attribute is, will be much appreciated.

asp.net asp.net-core asp.net-core-mvc .net-core

3

asked Oct 9 '16 at 19:10
Arrrr
1,180 1 16 36

## 2 Answers

21

From `AutoValidateAntiforgeryTokenAttribute` documentation:

> An attribute that causes validation of antiforgery tokens for all unsafe HTTP methods. An antiforgery token is required for HTTP methods other than GET, HEAD, OPTIONS, and TRACE. It can be applied at as a global filter to trigger validation of antiforgery tokens by default for an application.

`AutoValidateAntiforgeryTokenAttribute` allows to apply Anti-forgery token validation globally to all unsafe methods e.g. `POST`, `PUT`, `PATCH` and `DELETE`. Thus you don't need to add `[ValidateAntiForgeryToken]` attribute to each and every action that requires it.

To use it add the following code to your `ConfigureServices` method of `Startup` class

```
services.AddMvc(options =>
{
    options.Filters.Add(new AutoValidateAntiforgeryTokenAttribute());
});
```

edited Mar 20 '17 at 1:44
ChadT
**6,398** 2 35 54

answered Oct 10 '16 at 8:31
Andrei Mihalciuc
**1,330** 8 12

0

You may sometimes find yourself in need of marking multiple requests across a controller, while requiring some requests to not require anti-forgery, such as various GET based actions. There are several tools you can use to help the process become more convenient and comfortable for the user. The first is the AutoValidateAntiforgeryToken attribute. It behaves similarly to the ValidateAntiForgeryToken attribute; however, it will automatically ignore actions which are called with the methods: GET, HEAD, OPTIONS, and TRACE, which are designed for data retrieval. This allows you to quickly and easily add anti-forgery to all methods which can change data, without affecting methods for retrieving data.

The following code is an example of the AutoValidateAntiforgeryToken attribute:

```
[AutoValidateAntiforgeryToken]
public class AntiForgeController : Controller
{
    public IActionResult Index()
    {
        return View();
    }
    [HttpPost]
    public IActionResult Index(string userName)
    {
        return View("Index", userName);
    }
    [HttpDelete]
    public IActionResult RemoveUser(string userName)
    {
        string url = string.Format("~/RemovedUser/{0}", userName);
        return RedirectToAction("Account", "RemoveUser", "User");
    }
}
```

In this example, the normal Index action (GET) will work regardless of origin, while both the Index action with a POST method and the RemoveUser action which is a DELETE method will both require the client to utilize anti-forgery tokens.

answered May 24 at 10:04
Stefano Magistri