ActionVerbs

In this section, you will learn about the ActionVerbs selectors attribute.

The ActionVerbs selector is used when you want to control the selection of an action method based on a Http request method. For example, you can define two different action methods with the same name but one action method responds to an HTTP Get request and another action method responds to an HTTP Post request.

MVC framework supports different ActionVerbs, such as HttpGet, HttpPost, HttpPut, HttpDelete, HttpOptions & HttpPatch. You can apply these attributes to action method to indicate the kind of Http request the action method supports. If you do not apply any attribute then it considers it a GET request by default.

The following figure illustrates the HttpGET and HttpPOST action verbs.

```
http://localhost/Student/Edit/1
                    public ActionResult Edit(int Id)
    HttpGET          {

                          var std = students.Where(s => s.StudentId == Id).FirstOrDefault();

                          return View(std);
                      }
http://localhost/Student/Edit [HttpPost]
    HttpPOST          public ActionResult Edit(Student std)
                      {
                          //update database here..

                          return RedirectToAction("Index");
                      }
```

ActionVerbs

The following table lists the usage of http methods:

Http method	Usage
GET	To retrieve the information from the server. Parameters will be appended in the query string.
POST	To create a new resource.
PUT	To update an existing resource.
HEAD	Identical to GET except that server do not return message body.
OPTIONS	OPTIONS method represents a request for information about the communication options supported by web server.
DELETE	To delete an existing resource.
PATCH	To full or partial update the resource.

Visit W3.org ☐ for more information on Http Methods.

The following example shows different action methods supports different ActionVerbs:

Example: ActionVerbs

```
public class StudentController : Controller
{
    public ActionResult Index()
    {
        return View();
    }

    [HttpPost]
    public ActionResult PostAction()
    {
        return View("Index");
    }


    [HttpPut]
    public ActionResult PutAction()
    {
        return View("Index");
    }

    [HttpDelete]
    public ActionResult DeleteAction()
    {
        return View("Index");
    }

    [HttpHead]
    public ActionResult HeadAction()
```

```csharp
    {
        return View("Index");
    }


    [HttpOptions]
    public ActionResult OptionsAction()
    {
        return View("Index");
    }


    [HttpPatch]
    public ActionResult PatchAction()
    {
        return View("Index");
    }
}
```

You can also apply multiple http verbs using AcceptVerbs attribute. GetAndPostAction method supports both, GET and POST ActionVerbs in the following example:
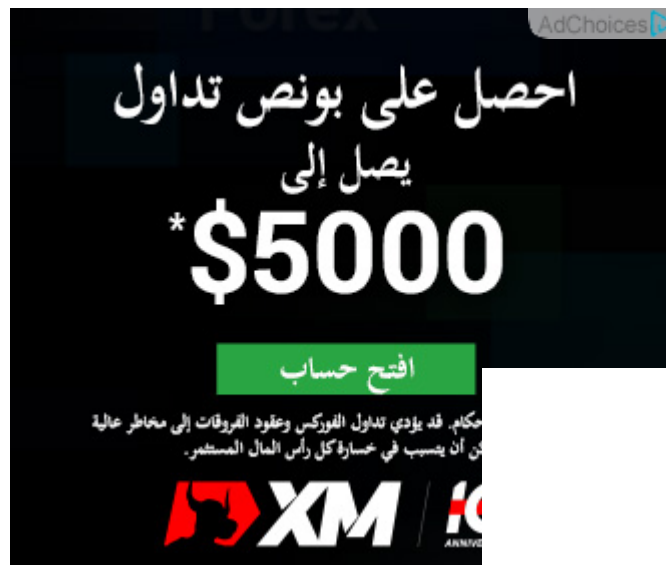
Example: AcceptVerbs

```csharp
[AcceptVerbs(HttpVerbs.Post | HttpVerbs.Get)]
public ActionResult GetAndPostAction()
{
    return RedirectToAction("Index");
}
```

💡 Points to Remember :

1) ActionVerbs are another Action Selectors which selects an action method based on request methods e.g POST, GET, PUT etc.

2) Multiple action methods can have same name with different action verbs. Method overloading rules are applicable.

3) Multiple action verbs can be applied to a single action method using AcceptVerbs attribute.

f ▶   Share   **y** ▶   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

> Web API

> C#

> LINQ

> Entity Framework

> AngularJS 1

> Node.js

> D3.js

> 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