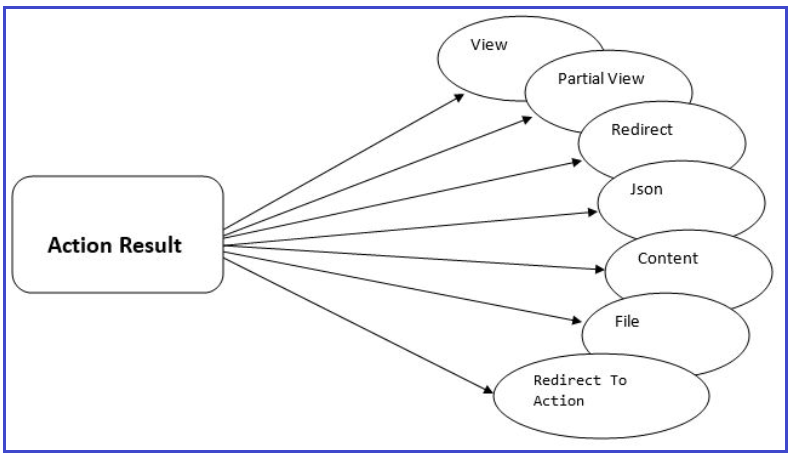
## What is the Action Method in ASP.NET MVC?

All the public methods inside a Controller which respond to the URL is known as Action Method. When creating an Action Method we must follow these rules.

1. The action method must be public.
2. It cannot be overloaded
3. It cannot be a static method
4. ActionResult is a base class of all the result type action methods.

##### What is the Action Result in MVC?

Action Result is a result of action methods or return types of action methods. The action result is an abstract class. It is the base class for all types of action results.



**Types of Action Results**

There are different types of action results in ASP.NET MVC. Each result has a different type of result format to view page. ActionResult is the base class of all the result type action method. The following are the Result types that an action method can return in MVC.

1. **ViewResult –** Represents HTML and markup.
2. **EmptyResult** – Represents no result.
3. **RedirectResult**– Represents a redirection to a new URL.
4. **JsonResult**– Represents a JavaScript Object Notation result that can be used in an AJAX application.
5. **JavaScriptResult**– Represents a JavaScript script.
6. **ContentResult**– Represents a text result.
7. **FileContentResult**– Represents a downloadable file (with the binary content).
8. **FilePathResult**– Represents a downloadable file (with a path).
9. **FileStreamResult**– Represents a downloadable file (with a file stream).

These results are categorized into three sections:

1. **Content-returning**
2. **Redirection**
3. **Status**.

**Content-returning:**

These ActionResults are responsible for returning content to the browser or calling script. The examples are as follows

1. ViewResult
2. PartialViewResult
3. FileResult
4. ContentResult
5. EmptyResult
6. JsonResult
7. JavaScriptResult

**Redirection:**

These ActionResults are responsible for redirecting to other URLs or actions. The examples are as follows

1. RedirectResult
2. RedirectToRouteResult
3. RedirectToActionResult

**Status:**

These ActionResults are responsible for returning status codes to the browser for it to use. The examples are as follows

1. HttpStatusCodeResult
2. HttpUnauthorizedResult
3. HttpNotFoundResult

## ****Redirect, RedirectToRoute and RedirectToAction in MVC****

##### ****Redirect Result in MVC****

**If you wanted to redirect to a URL, then you can use RedirectResult, like this:**

public class HomeController : Controller

{

public RedirectResult Index()

{

return Redirect("https://Test.com");

}

}

That works great for redirecting to outside sites from the current application, but not so much for redirecting to other pages within the same application. For that, we can use RedirectToRouteResult.

Redirect result is returning the result to a specific URL. It is rendered to the page by URL. If we give the wrong URL, it will show 404-page errors.

##### ****RedirectToRoute Result in MVC****

RedirectToRouteResult is used whenever we need to go from one action to another.

**public** **class** HomeController : Controller

**{**

**public** RedirectToRouteResult Index**()**

**{**

**return** RedirectToRoute**(**new **{** controller = "Home", action = "About" **})**;

**}**

**}**

That’s not very friendly though. There’s a better way, an overload of this helper called RedirectToAction:

##### ****RedirectToAction Result in MVC****

The RedirectToAction Result is returning the result to a specified controller and action method. Controller name is optional in RedirectToAction method. If not mentioned, Controller name redirects to a mentioned action method in the current Controller. Suppose action name is not available but mentioned in the current controller, then it will show a 404 error page.

**public** **class** HomeController : Controller

**{**

**public** ActionResult Index**()**

**{**

**return** RedirectToAction**(**"Login", "Account"**)**;

**}**

**}**

## ****Status Result in ASP.NET MVC Application****

The Status Action Results will return status codes to the browser for it to use. As part of the Status Result, we will discuss the following three things.

1. HttpStatusCodeResult
2. HttpUnauthorizedResult
3. HttpNotFoundResult

#### ****HttpStatusCodeResult:****

HttpStatusCodeResult returns an HTTP status code to the browser, along with a custom message to be displayed. Let’s modify the Home Controller to understand this concept in MVC.

**<**span style="font-family: arial, helvetica, sans-serif;"**>public** **class** HomeController : Controller

**{**

**public** HttpStatusCodeResult UnauthorizedStatusCode**()**

**{**

**return** new HttpStatusCodeResult**(**HttpStatusCode.Unauthorized, "You are not authorized to access this controller action."**)**;

**}**

**public** HttpStatusCodeResult BadGateway**()**

**{**

**return** new HttpStatusCodeResult**(**HttpStatusCode.BadGateway, "I have no idea what this error means."**)**;

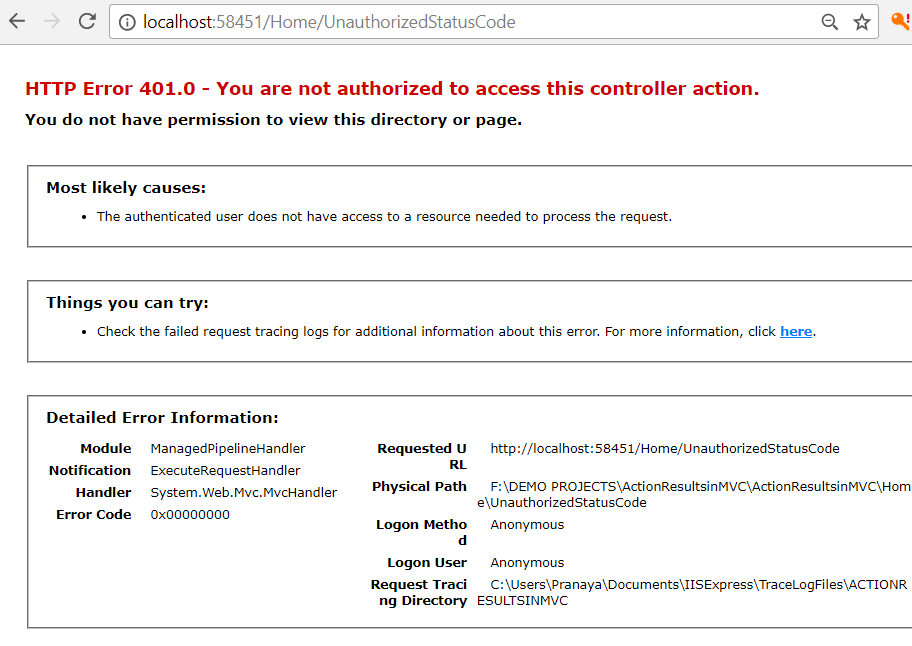
**}**

**}**

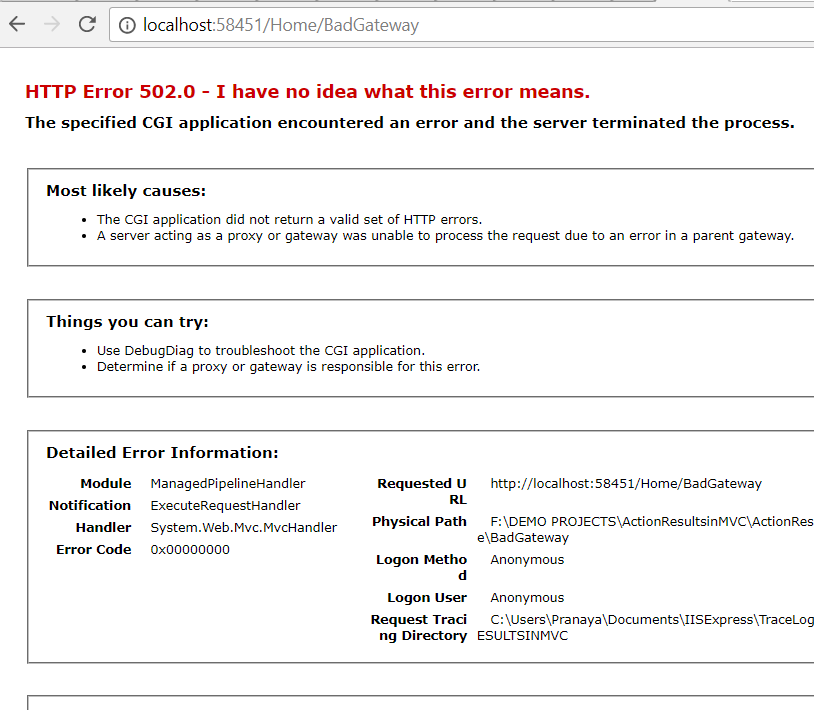
**<**/span**>**

##### ****Notice that there is no helper method for this ActionResult.****

The HttpStatusCode enumeration contains all HTTP status codes (so that you don’t have to remember what 402 or 307 mean). These are useful in exception-driven scenarios where you have custom error pages defined. So when we navigate to **Home/UnauthorizedStatusCode**, it will display the following error page



Similarly, when we navigate to **Home/BadGateway**, it will display the following error page



#### ****HttpUnauthorizedResult****

Returning an HttpUnauthorizedResult is the same as returning HttpStatusCodeResult with HttpStatusCode.Unauthorized, it’s just more readable:

**<**span style="font-family: arial, helvetica, sans-serif;"**>public** **class** HomeController : Controller

**{**

**public** HttpStatusCodeResult UnauthorizedResult**()**

**{**

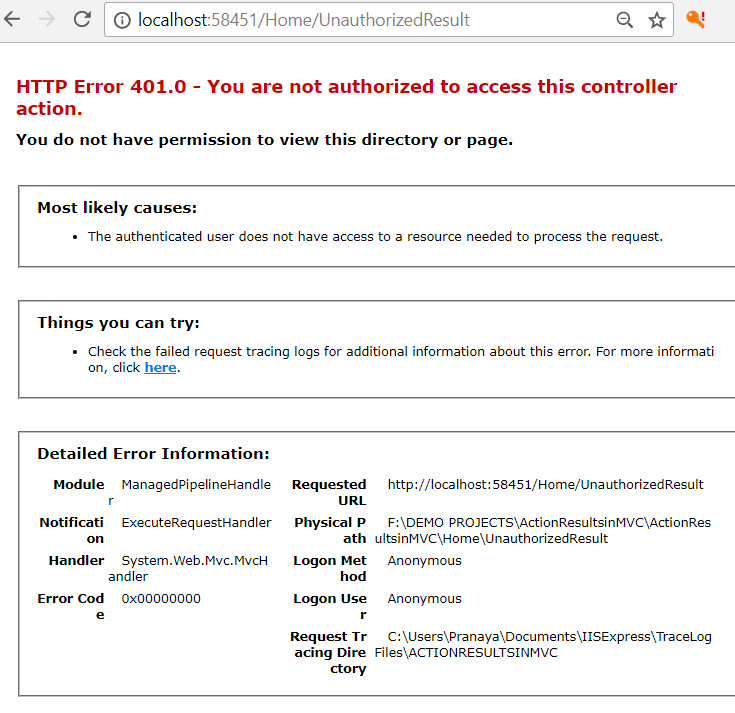
**return** new HttpUnauthorizedResult**(**"You are not authorized to access this controller action."**)**;

**}**

**}**

**<**/span**>**

When we navigate to **Home/UnauthorizedResult**, then it will display the following error page



#### ****HttpNotFoundResult****

This is also an overload of HttpStatusCodeResult, but unlike HttpUnauthorizedResult, it actually does have a helper method:

**<**span style="font-family: arial, helvetica, sans-serif;"**>public** **class** HomeController : Controller

**{**

**public** HttpNotFoundResult NotFound**()**

**{**

**return** HttpNotFound**(**"We didn't find that action, sorry!"**)**;

**}**

**}**

**<**/span**>**

When we navigate to **Home/NotFound**, then it will display the following error page

