

[Home \(/\)](#)

[ASP.NET Core \(/asp-net-core\)](#)

[Tutorials \(/razor-pages/tutorial/bakery\)](#)

[Razor Page Files \(/razor-pages\)](#)

[Razor Syntax \(/razor-syntax\)](#)

[Page Models \(/razor-pages/pagemodel\)](#)

[Handler Methods \(/razor-pages/handler-methods\)](#)

[ViewData \(/razor-pages/viewdata\)](#)

[Action Results \(/razor-pages/action-results\)](#)

[Tag Helpers \(/razor-pages/tag-helpers/\)](#)

[View Components \(/razor-pages/view-components\)](#)

[Routing and URLs \(/razor-pages/routing\)](#)

[Startup \(/startup\)](#)

[Configuration \(/configuration\)](#)

[Middleware \(/middleware\)](#)

[Dependency Injection \(/advanced/dependency-injection\)](#)

[Working With Forms \(/razor-pages/forms\)](#)

[Validation \(/razor-pages/validation\)](#)

[Model Binding \(/razor-pages/model-binding\)](/razor-pages/model-binding)[State Management \(/razor-pages/state-management\)](/razor-pages/state-management)[Caching \(/razor-pages/caching\)](/razor-pages/caching)[Managing Security With ASP.NET Identity \(/identity\)](/identity)[Using AJAX \(/razor-pages/ajax\)](/razor-pages/ajax)[Working with JSON \(/web-api\)](/web-api)[Scaffolding \(/miscellaneous/scaffolding\)](/miscellaneous/scaffolding)[Publishing To IIS \(/publishing/publish-to-iis\)](/publishing/publish-to-iis)[Advanced \(/advanced\)](/advanced)[Table Of Contents \(/table-of-contents\)](/table-of-contents)

Action Results in Razor Pages

Action results in Razor Pages are commonly used as the return type of [handler methods \(/razor-pages/handler-methods\)](/razor-pages/handler-methods) and are responsible for generating responses and appropriate status codes. Action results implement either the abstract `Microsoft.AspNetCore.Mvc.ActionResult` class, or the `Microsoft.AspNetCore.Mvc.IActionResult` interface. ASP.NET Core includes more than three dozen `ActionResult` classes covering a wide range of needs, including but not limited to executing and returning the content of a Razor page, returning the content of a file, redirecting to another resource or simply returning a specific HTTP status code. It is also possible to author your own `ActionResult` classes if you need something bespoke.

The following code illustrates the use of the `RedirectToPageResult` class in a simple `onGet` handler method, which results in the user being redirected to the specified page:

```
public IActionResult OnGet()
{
    return new RedirectToPageResult("Index");
}
```

The `onGet` handler method has a return type of `IActionResult`, which means that the return type can be any class that implements the `IActionResult` interface somewhere in its inheritance hierarchy. Ultimately, this means any action result in the entire framework. Generally, it is a good idea to be as specific as possible with handler method return types, so this example should be refactored as follows:

```
public RedirectToPageResult OnGet()
{
    return new RedirectToPageResult("Index");
}
```

If your handler method returns more than one type of action result depending on conditions, you will need to broaden the return type of the method to `ActionResult` or `IActionResult` as this example demonstrates, which returns a `PageResult` if the `ModelState` is not valid, or a `RedirectToPageResult` (a different type) if `ModelState` is valid:

```
public IActionResult OnPost()
{
    if(!ModelState.IsValid)
    {
        return new PageResult();
    }
    // otherwise do some processing
    return new RedirectToPageResult("Index");
}
```

Many action results have associated helper methods defined on the Razor Pages `PageModel` class that negate the need to "new up" instances of `ActionResult` classes and thereby help simplify code. The `Page()` method returns a `PageResult`, and the `RedirectToPage` method returns a `RedirectToPageResult`, so the previous example can be simplified using those methods:

```
public IActionResult OnPost()
{
    if(!ModelState.IsValid)
    {
        return Page();
    }
    // otherwise do some processing
    return RedirectToPage("Index");
}
```

The following table lists `ActionResult` classes that are intended to be used in Razor Pages development, along with any HTTP status codes they return, and associated helper methods, if they exist.

ActionResult	Helper	HTTP Status Code	Description
ChallengeResult	Challenge	401	Used in authentication. You will return a <code>ChallengeResult</code> if the user has not authenticated successfully. It returns a 401 (Unauthorized) HTTP status code.
ContentResult	Content	200	Takes a string and returns it with a <code>text/plain</code> content-type header by default. Overloads enable you to specify the content-type to return other formats such as <code>text/html</code> or <code>application/json</code> , for example.
EmptyResult		200	This action result type can be used to denote that a server-side operation completed successfully where there is no return value.
FileContentResult	File	200	Returns a file from a byte array, stream or virtual path.
FileStreamResult		200	Returns a file from a stream
ForbidResult	Forbid	403	Used in authentication. The <code>Forbid</code> method returns a 403 (Forbidden) HTTP status code.

ActionResult	Helper	HTTP Status Code	Description
LocalRedirectResult	LocalRedirect	302	Redirects the user to the location within the current site specified by the relative url ¹ .
	LocalRedirectPermanent	301	
	LocalRedirectPreserveMethod	307	
	LocalRedirectPreserveMethodPermanent	308	
NotFoundResult	NotFound	404	Returns an HTTP 404 (Not Found) status code indicating that the requested resource could not be found.
PageResult	Page	200	Will process and return the result of the current page.
PartialResult	Partial ³	200	Returns a <u>Partial Page (/razor-pages/partial-pages)</u>
PhysicalFileResult	PhysicalFile	200	Returns a file from the specified physical path.
RedirectResult	Redirect	301	Redirects the user to the URL specified, with an HTTP status code dependent on the options specified. The default is a temporary redirect (302) ¹
	RedirectPermanent	302	
	RedirectPreserveMethod	307	
	RedirectPermanentPreserveMethod	308	
RedirectToPageResult	RedirectToPagePermanent	301	Redirects the user to the specified page. ^{1,2}
	RedirectToPage	302	
	RedirectToPagePreserveMethod	307	
	RedirectToPagePreserveMethodPermanent	308	
SignInResult			Part of Identity. Represents the result of a sign-in operation.
SignOutResult			Part of Identity. Represents the result of a sign-out operation.
StatusCodeResult			Enables you to generate a response with your chosen HTTP Status code
UnauthorizedResult		401	

ActionResult	Helper	HTTP Status Code	Description
ViewComponentResult		200	Returns the result of executing a <u>ViewComponent (/razor-pages/view-components)</u>

Notes

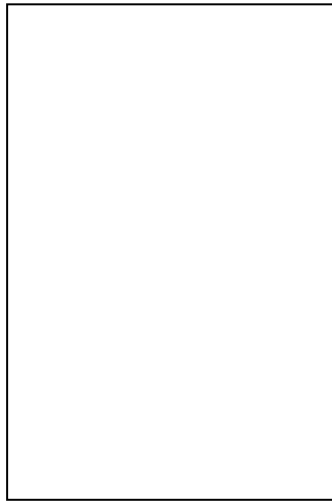
1. The Redirect action results include options to specify that the redirect should be permanent and/or the HTTP verb should be preserved across redirects. Temporary redirects are denoted by HTTP status codes 302 and 307. Permanent redirects are denoted by status codes 301 and 308. Permanent redirect instructions are usually honoured by search engines and other indexing technologies, which will replace the existing entry in their index with the new location specified in the `location` header that action result generates.

If you do not opt to preserve the HTTP method that the original request used, the redirected request may (and often will) use GET regardless of the method used for the original request. If you want to ensure that the original HTTP method is preserved for the subsequent request (and potentially any form values), you should use one of the options that includes `PreserveMethod` in their name, or specify `true` for the appropriate parameter in the more generic `RedirectResult` method.

2. The value passed in to the `RedirectToPageResult` methods is relative to the current page, unless it is prefixed with a forward slash (`/`) in which case it is relative to the *Pages* folder.

3. The `Partial` helper method is available from ASP.NET Core 2.2 onwards.

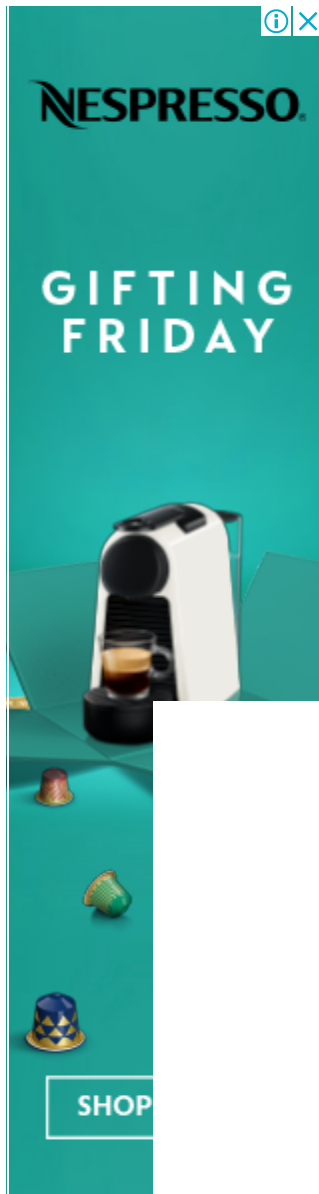
Last updated: 24/10/2020 07:02:14



On this page

Action Results in Razor Pages

Notes



Latest Updates

Caching in Razor Pages (<https://www.learnrazorpages.com/razor-pages/caching>)

Middleware in Razor Pages (<https://www.learnrazorpages.com/middleware>)

Using Cookies in Razor Pages (<https://www.learnrazorpages.com/razor-pages/cookies>)

The Razor _Layout.cshtml file (<https://www.learnrazorpages.com/razor-pages/files/layout>)

An Introduction To ASP.NET Core Razor Pages (<https://www.learnrazorpages.com/>)

Publishing and deploying a Razor Pages application to IIS on Windows (<https://www.learnrazorpages.com/publishing/publish-to-iis>)

© 2020 - Mike Brind.

All rights reserved.

Contact me at Outlook.com