



MICROSOFT ARCHITECT

By Joydip Kanjilal, Columnist, InfoWorld
AUG 17, 2020 3:00 AM PDT

How to redirect a request in ASP.NET Core MVC

Take advantage of redirect action results in ASP.NET Core MVC to elegantly redirect a request to a specified URL

ASP.NET Core is a cross-platform, open source, lean, fast, and modular framework for building high-performance web applications. ASP.NET Core MVC applications enable you to redirect a request to a specified URL in several different ways. This article talks about how we can accomplish this with code examples wherever appropriate.

To work with the code examples provided in this article, you should have Visual Studio 2019 installed in your system. If you don't already have a copy, you can download Visual Studio 2019 [here](#).

[[Also on InfoWorld: How to refactor God objects in C#](#)]

Create an ASP.NET Core MVC project in Visual Studio

First off, let's create an ASP.NET Core project in Visual Studio 2019. Assuming Visual Studio 2019 is installed in your system, follow the steps outlined below to create a new ASP.NET Core project in Visual Studio.

[[Also on InfoWorld: Understanding Microsoft .NET 5](#)]

1. Launch the Visual Studio IDE.
2. Click on "Create new project."
3. In the "Create new project" window, select "ASP.NET Core Web Application" from the list of templates displayed.
4. Click Next.

5. In the “Configure your new project” window, specify the name and location for the new project.
6. Optionally check the “Place solution and project in the same directory” check box, depending on your preferences.
7. Click Create.
8. In the “Create a New ASP.NET Core Web Application” window shown next, select .NET Core as the runtime and ASP.NET Core 3.1 (or later) from the drop-down list at the top.
9. Select “Web Application (Model-View-Controller)” as the project template to create a new ASP.NET Core MVC application.
10. Ensure that the check boxes “Enable Docker Support” and “Configure for HTTPS” are unchecked as we won’t be using those features here.
11. Ensure that Authentication is set to “No Authentication” as we won’t be using authentication either.
12. Click Create.

Following these steps will create a new ASP.NET Core MVC project in Visual Studio 2019. We’ll use this project in the sections below to illustrate how we can redirect requests when working with action methods in ASP.NET Core 3.1.

Redirect action results in ASP.NET Core MVC

There are several types of action results in ASP.NET Core MVC such as `RedirectResult`, `RedirectToActionResult`, `RedirectToRouteResult`, and `LocalRedirectResult`. All of these classes extend the `ActionResult` class and the `IActionResult` and `IKepTempDataResult` interfaces and return Found (Http Status Code 302), Moved Permanently (Http Status Code 301), Temporary Redirect (Http Status Code 307), or Permanent Redirect (Http Status Code 308).

We’ll examine how we can work with each of these in this section.

Use `RedirectResult` in ASP.NET Core MVC

You can use any of the following methods to return a `RedirectResult`:

RECOMMENDED WHITEPAPERS



Advantages of using a Managed Service Provider

UNITED STATES ▼



How ComplyScore Checks Vendors for AWS Best Practices Compliance



2021 GigaOm Radar for Cloud Observability Distinguishes Splunk as the Market Leader

- Redirect – Http Status Code 302 Found (temporarily moved to the URL provided in the location header)
- RedirectPermanent – Http Status Code 301 Moved Permanently
- RedirectPermanentPreserveMethod – Http Status Code 308 Permanent Redirect
- RedirectPreserveMethod – Http Status Code 307 Temporary Redirect

The following lines of code show how you can use each of these methods.

```
Redirect("/Author/Index");
```

```
RedirectPermanent("/Author/Index");
```

```
RedirectPermanentPreserveMethod("/Author/Index");
```

```
RedirectPreserveMethod("/Author/Index");
```

Alternatively, you can return an instance of `RedirectResult` as shown in the code snippet given below.

```
public RedirectResult Index()  
{  
    return new RedirectResult(url: "/Author/Index", permanent: true,  
                              preserveMethod: true);  
}
```

Note that the `Redirect` method can be used to redirect a request to a specified URL. This method is available in the abstract base class called `ControllerBase`.

```
public RedirectToActionResult Index()
{
    return Redirect("https://google.com");
}
```

It should be noted that the controllers you create in ASP.NET Core MVC extend the Controller class. This class in turn extends the ControllerBase class and implements the IActionFilter, IFilterMetadata, IAsyncActionFilter, and IDisposable interfaces.

Use RedirectToActionResult in ASP.NET Core MVC

This action result can be used to redirect to the specified action and controller. If no controller is specified it redirects to the specified action within the current controller. You can use any of the following methods to redirect to the specified action and return an instance of RedirectToActionResult from your action method.

- **RedirectToAction** – Http Status Code 302 Found (temporarily moved to the URL provided in the location header)
- **RedirectToActionPermanent** – Http Status Code 301 Moved Permanently
- **RedirectToActionPermanentPreserveMethod** – Http Status Code 308 Permanent Redirect
- **RedirectToActionPreserveMethod** – Http Status Code 307 Temporary Redirect

The following code snippet illustrates how the RedirectToAction method can be used.

```
public RedirectToActionResult Index()
{
    return RedirectToAction(actionName: "Index", controllerName: "Author");
}
```

You can skip the controller name if you want to redirect the request to an action method in the current controller. The following code snippet shows how this can be achieved.

```
public RedirectToActionResult Index()
{
    return RedirectToAction(actionName: "Privacy");
}
```

Use RedirectToRouteResult in ASP.NET Core MVC

This is yet another action result that can be used to redirect the request to the specified route. You can use any of the following methods to return an instance of RedirectToRouteResult from your action method.

- RedirectToRoute – Http Status Code 302 Found (temporarily moved to the URL provided in the location header)
- RedirectToRoutePermanent – Http Status Code 301 Moved Permanently
- RedirectToRoutePermanentPreserveMethod – Http Status Code 308 Permanent Redirect
- RedirectToRoutePreserveMethod – Http Status Code 307 Temporary Redirect

The following code snippet shows how the RedirectToRoute method can be used.

```
public RedirectToRouteResult Index()
{
    return RedirectToRoute("author");
}
```

You can also specify the route value when redirecting as shown in the code snippet given below.

```
var routeValue = new RouteValueDictionary
(new { action = "View", controller = "Author" });
return RedirectToRoute(routeValue);
```

Use LocalRedirectResult in ASP.NET Core MVC

This action result is used when you want to redirect to a local URL. It throws an InvalidOperationException if you use an external URL with it. You can use any of the following methods to return an instance of LocalRedirectResult from your action method.

- LocalRedirect – Http Status Code 302 Found (temporarily moved to the URL provided in the location header)
- LocalRedirectPermanent – Http Status Code 301 Moved Permanently

- [LocalRedirectPermanentPreserveMethod – Http Status Code 308 Permanent Redirect](#)
- [LocalRedirectPreserveMethod – Http Status Code 307 Temporary Redirect](#)

Redirect to razor pages in ASP.NET Core MVC

Finally, note that you can even redirect to razor pages using the `RedirectToPage` method, specifying the target razor page to redirect the request to. The `RedirectToPage` method returns a `RedirectToPageResult` instance together with a HTTP Status Code 302.

If you have a page named `Author`, where you want the request to be redirected, you can use the following code snippet.

```
public IActionResult RedirectToAuthorPage()
{
    return RedirectToPage("Author");
}
```

How to do more in ASP.NET Core:

- [How to use attribute routing in ASP.NET Core](#)
- [How to pass parameters to action methods in ASP.NET Core MVC](#)
- [How to use API Analyzers in ASP.NET Core](#)
- [How to use route data tokens in ASP.NET Core](#)
- [How to use API versioning in ASP.NET Core](#)
- [How to use Data Transfer Objects in ASP.NET Core 3.1](#)
- [How to handle 404 errors in ASP.NET Core MVC](#)
- [How to use dependency injection in action filters in ASP.NET Core 3.1](#)
- [How to use the options pattern in ASP.NET Core](#)
- [How to use endpoint routing in ASP.NET Core 3.0 MVC](#)
- [How to export data to Excel in ASP.NET Core 3.0](#)
- [How to use `LoggerMessage` in ASP.NET Core 3.0](#)
- [How to send emails in ASP.NET Core](#)
- [How to log data to SQL Server in ASP.NET Core](#)

- How to schedule jobs using Quartz.NET in ASP.NET Core
- How to return data from ASP.NET Core Web API
- How to format response data in ASP.NET Core
- How to consume an ASP.NET Core Web API using RestSharp
- How to perform async operations using Dapper
- How to use feature flags in ASP.NET Core
- How to use the FromServices attribute in ASP.NET Core
- How to work with cookies in ASP.NET Core
- How to work with static files in ASP.NET Core
- How to use URL Rewriting Middleware in ASP.NET Core
- How to implement rate limiting in ASP.NET Core
- How to use Azure Application Insights in ASP.NET Core
- Using advanced NLog features in ASP.NET Core
- How to handle errors in ASP.NET Web API
- How to implement global exception handling in ASP.NET Core MVC
- How to handle null values in ASP.NET Core MVC
- Advanced versioning in ASP.NET Core Web API
- How to work with worker services in ASP.NET Core
- How to use the Data Protection API in ASP.NET Core
- How to use conditional middleware in ASP.NET Core
- How to work with session state in ASP.NET Core
- How to write efficient controllers in ASP.NET Core

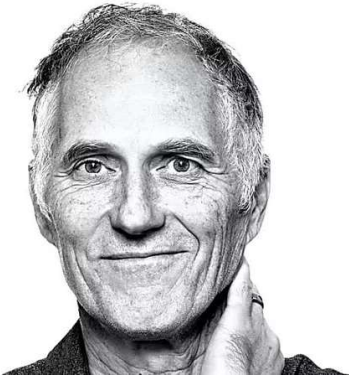
Joydip Kanjilal is a Microsoft MVP in ASP.Net, as well as a speaker and author of several books and articles. He has more than 20 years of experience in IT including more than 16 years in Microsoft .Net and related technologies.

Follow     

- Stay up to date with InfoWorld's newsletters for software developers, analysts, database programmers, and data scientists.
- Get expert insights from our member-only Insider articles.

YOU MAY ALSO LIKE

Recommended by



Tim O'Reilly: the golden age of the programmer is over



10 top-notch libraries for C++ programming



The decline of Heroku



The 24 highest paying developer roles in 2020



How to use SortedDictionary, SortedList, and SortedSet in C#



Are industry clouds an opportunity or a distraction?



What is a computational storage drive? Much-needed



6 reasons to switch to managed Kubernetes



Red Hat Enterprise Linux takes aim at edge



6 ways Alibaba Cloud challenges AWS, Azure, and GCP



The shifting market for PostgreSQL

SPONSORED LINKS

dtSearch® instantly searches terabytes of files, emails, databases, web data. See site for hundreds of reviews; enterprise & developer evaluations

Truly modern web app and API security thinking. It's a thing. See how.

Want lightning fast analytics? See why the Incorta data analytics platform is changing enterprise data forever.

2020 was a year of rapid progression of digital transformation for businesses. The following is a snapshot of the digital transformation advancements made across all facets of business.

DDoS extortion attacks are real. Don't Negotiate. Mitigate with NETSCOUT. Learn more.