API Versioning in Asp.Net Core 2.0

Aram Koukia Follow
Apr 14, 2018 · 3 min read

API Versioning is one of those never ending debates that happens anytime you want to build an API no matter how many APIs you have built before.

Some are a fan of accepting API versions in the query strings, some from the request headers and etc, and based on all these conversations in the older days of Asp.Net Web API, we used to go in and build some http handler or attribute or some routes to handle different versions of the API.

But with .Net Core this is much easier and you add API Versioning to any API with ONE LINE OF CODE!

This is one of the most beautiful implementations that has been done in .Net Core that solves a lot of problems.

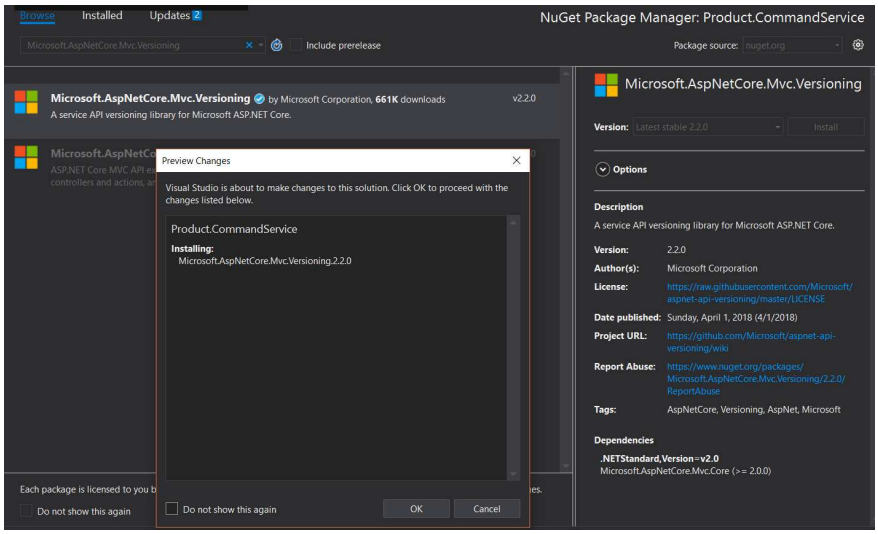
If you ever decide you want to use query strings instead of request headers, it would be a very trivial change!

Interested to see how it is done? Let's get into it then.

Adding Versioning to your .Net Core Web API:

To get started open your API.Net MVC Core project and install the following nuget package:

Microsoft.AspNetCore.Mvc.Versioning.



Then open your Startup.cs class and add the following line to the ConfigureServices method:

```
1  public void ConfigureServices(IServiceCollection services)
2  {
3      services.AddMvc();
4      services.AddApiVersioning(o => o.ApiVersionReader = new
5  }
```

You will need the following using statement:

using Microsoft.AspNetCore.Mvc.Versioning;

As you can see I am adding API versioning into the request header with the header name = “api-version”.

Now in the API and controller side we can specify which controller will be handling which version, like the following:

```
namespace Product.CommandService.Controllers.Product.V1
{
  [ApiVersion("1.0")]
  [Produces("application/json")]
  [Route("api/Product")]
  public class ProductController : Controller
  {
  }
}

namespace Product.CommandService.Controllers.Product.V2
{
  [ApiVersion("2.0")]
```

As you can see we have the same ProductController class, and one of them will handle the incoming requests with api-version=1.0 and the other one will handle the requests with api-version=2.0 in the request header.

Ignoring API Versioning for some endpoints

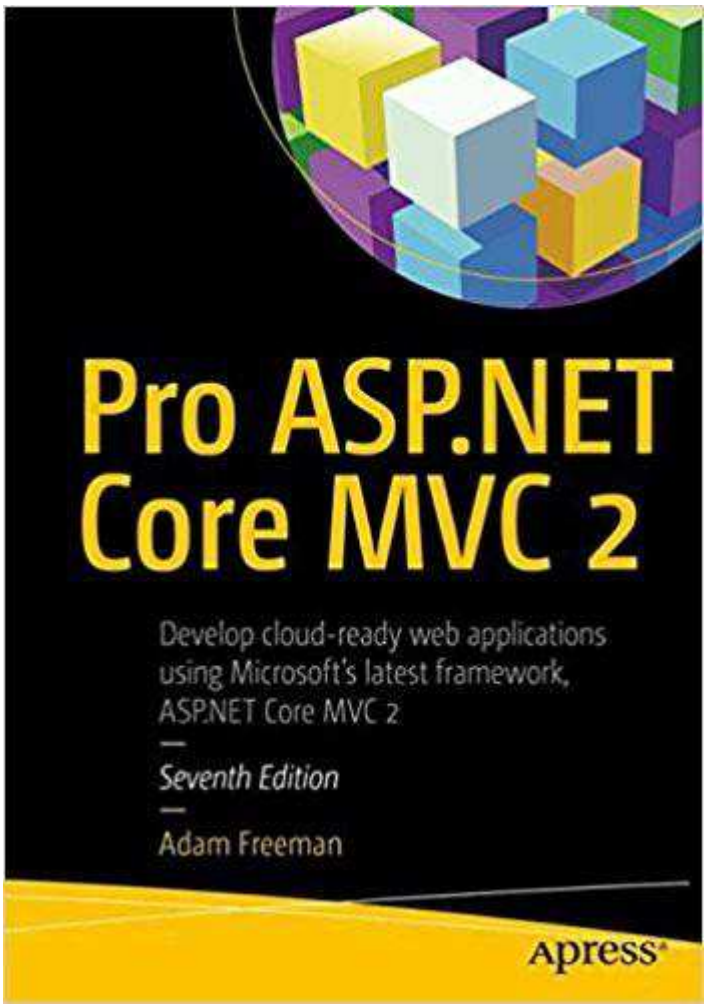
If you have some APIs that you don't want any versioning for them and you will only have one version of them, you can opt out using [ApiVersionNeutral] attribute like the following:

```
[ApiVersionNeutral]
[Route("api/optout")]
public class OptOutControler : Controller
{
        [HttpGet]
```

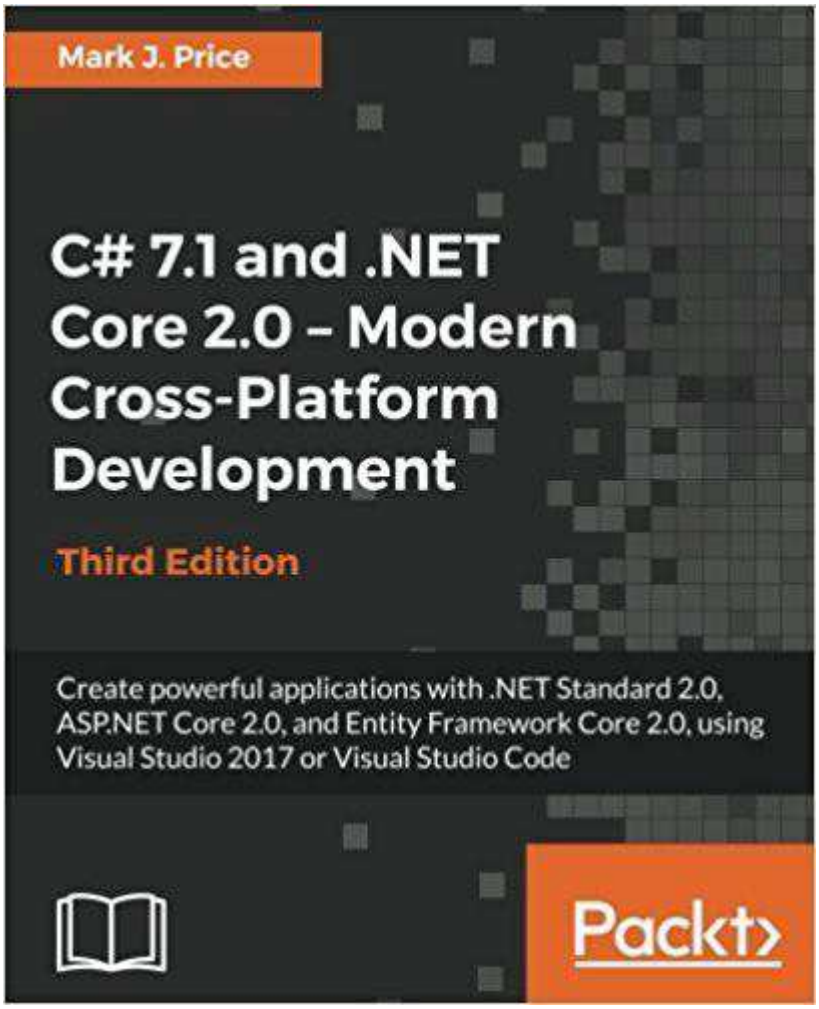
Now you can go ahead and add versioning to all of your existing APIs.

. . .

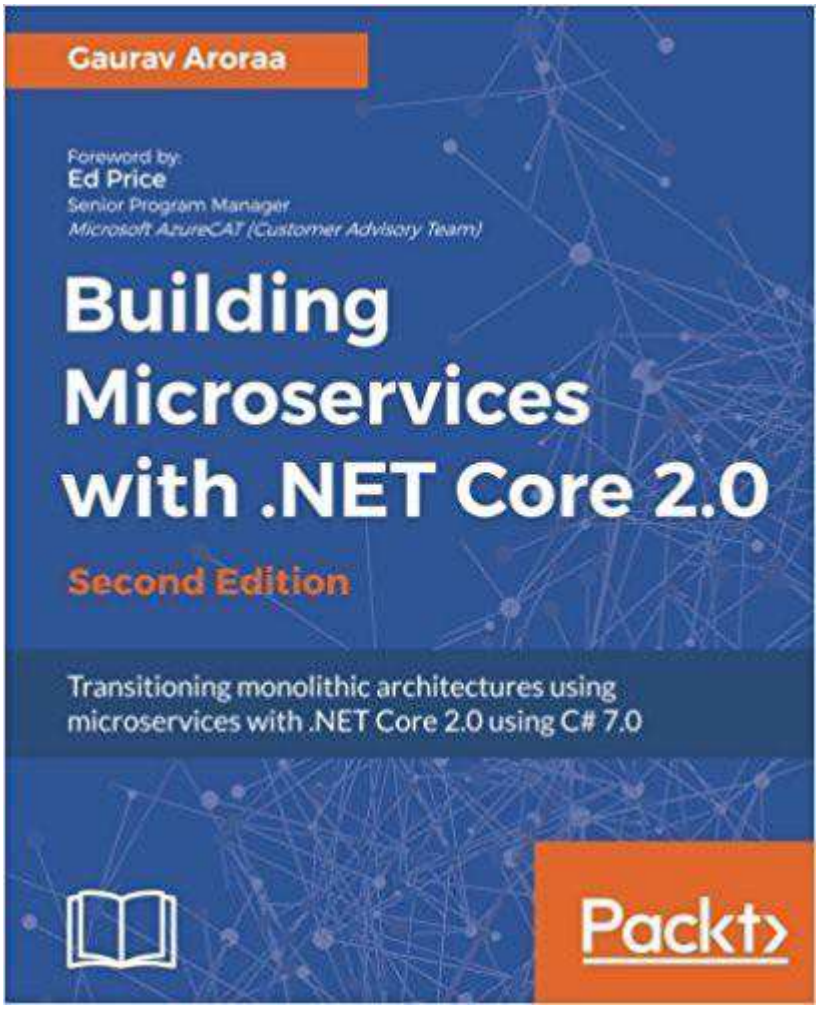
Want to learn more about .Net Core? Here are some useful books for you:



https://amzn.to/2Hzi7Z2



https://amzn.to/2JJL4Ce



https://amzn.to/2JOCgLw

Cheers!