Versioning Your API

Shawn Wildermuth

@shawnwildermuth | wilderminds.com

This Module

Versioning Your API

What Is API Versioning?

- Once you publish an API, it's set in stone
  - Users/Customers rely on the API not changing
  - But requirements will change
  - Need a way to evolve the API without breaking existing clients
  - API Versioning isn't Product Versioning
    - Don't tie them together

The Problem with API Versioning

- In typical projects
  - Versioning is accomplished with different versions of the package
    - E.g. Assemblies
- API Versioning is harder
  - Your API needs to support both new and old users
  - Side-by-side deployment isn't feasible most of the time
  - Need to support both versions in same code base

API Versioning Schemes

- There are lots of ways to version an API
  - Not all of them are recommended
  - If you find a mechanism that works for you, that’s all that matters
  - But remember that you’re serving your clients, not yourselves

Versioning in the URI

```
// URI Path
https://foo.org/api/v2/Customers


// Query String
https://foo.org/api/Customers?v=2.0
```

Versioning with Headers

```
GET /api/camps HTTP/1.1
Host: localhost:44388
Content-Type: application/json
X-Version: 2.0
```

Versioning with Accept Header

```
GET /api/camps HTTP/1.1
Host: localhost:44388
Content-Type: application/json
Accept: application/json;version=2.0
```

Versioning with Content Type

```
GET /api/camps HTTP/1.1
Host: localhost:44388
Content-Type: application/vnd.yourapp.camp.v1+json
Accept: application/vnd.yourapp.camp.v1+json
```

Demo

Adding ApiVersioning

Demo

Using Versioning Attributes

Demo

Adding a Versioned Controller

Demo

Versioning the Payload

Demo

Customizing Versioning Method

Demo

Using Versioning Conventions

What We've Learned

Versioning Your API

ASP.NET Core has versioning support built-in

You can version your URLs and your Payloads

You can choose attribute or conventions for config