

# Versioning Your API

---



**Shawn Wildermuth**

MICROSOFT MVP, INSTRUCTOR, AND FILMMAKER

@shawnwildermuth    wildermuth.com



# What Is API Versioning?

**Once you publish an API, it's set in stone**

- Users/Customers rely on the API
- But requirements will change
- Need a way to evolve the API
  - Don't break clients
- API Versioning isn't Product Versioning
  - Don't tie them together



# The Problem with API Versioning

## In .NET

- Is done with 'package' versions
  - E.g. Assemblies

## API versioning is harder

- Needs to support new and old users
- Side-by-side deployment isn't feasible
- Code base to support both versions



# API Versioning Schemes

**There are lots of ways to version an API**

- Not all of them are recommended
- Find a mechanism that works for you
- But remember:
  - 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

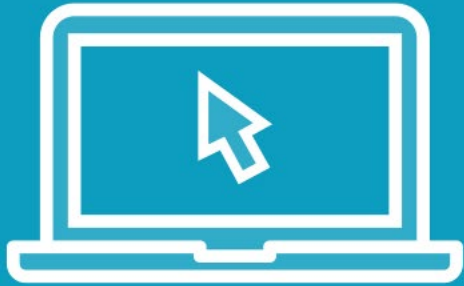
Content-Type: application/vnd.yourapp.camp.v1+json

Accept: application/vnd.yourapp.camp.v1+json





# Demo



## Introducing Versioning



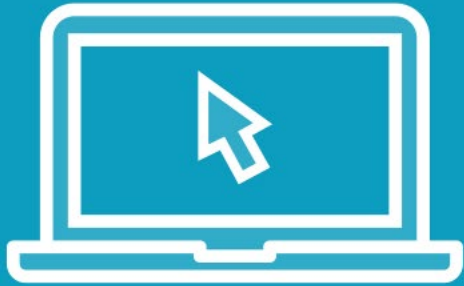
# Demo



## Versioning Actions



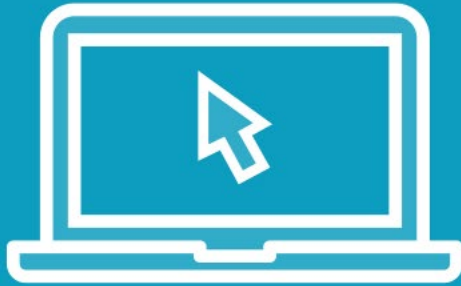
# Demo



## Versioning Controllers



# Demo



## Versioning with Headers



# Demo



## Using Multiple Versioning Methods



# Demo



## URL Versioning



# Demo



## Versioning Conventions



## Other Versioning Methods

### **The Versioning Library supports:**

- Versioning by Namespaces
- Versioning with Content Type
- Writing your own Readers
- Writing your own Resolvers





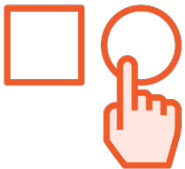
# What We've Learned



API Versioning is supported by Microsoft



You can version your URLs and your payloads



You can choose attribute or conventions for versioning configuration

