Dealing with Different Versions and Protecting the Documentation

Kevin Dockx
ARCHITECT

@KevinDockx https://www.kevindockx.com

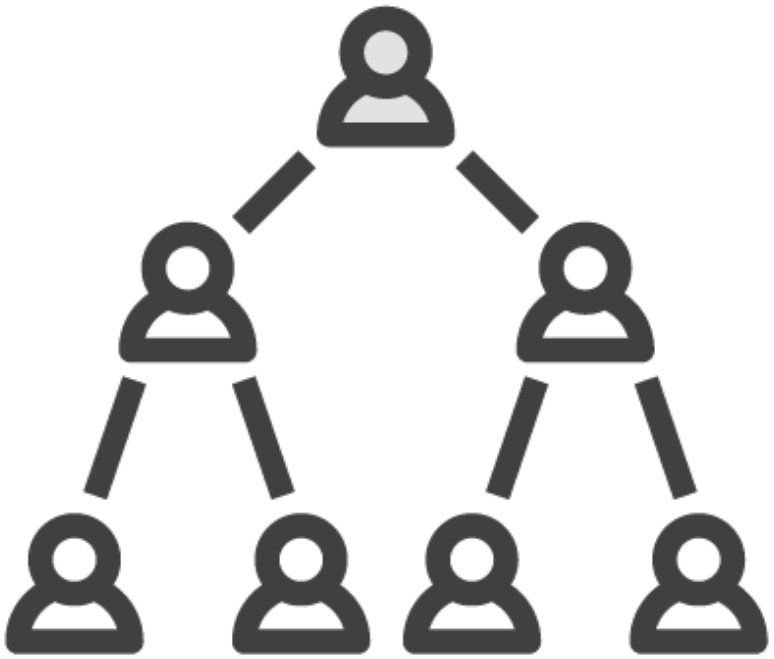
Coming Up

Working with multiple OpenAPI specifications

Creating an OpenAPI specification for each API version

Describing authentication in an OpenAPI specification

Working with Multiple OpenAPI Specifications

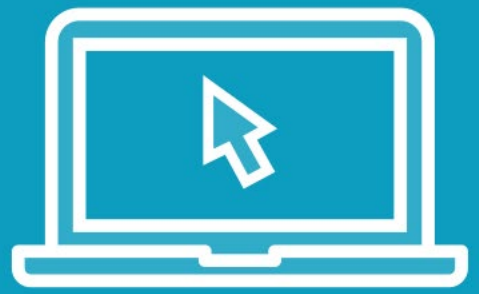


Use multiple OpenAPI specification for grouping

- For example: admins versus regular users

Use that principle to group specifications by API versions

Demo



Working with multiple OpenAPI specifications

Versioning with ASP.NET Core's Built-in Approach

As APIs evolve, different versions start to co-exist

- Different versioning strategies exist

Version the URI

- https://root/api/**v1**/authors
- https://root/api/**v2**/authors

Version the URI via query string parameter

- https://root/api/authors?**version=v1**

Version via custom request header

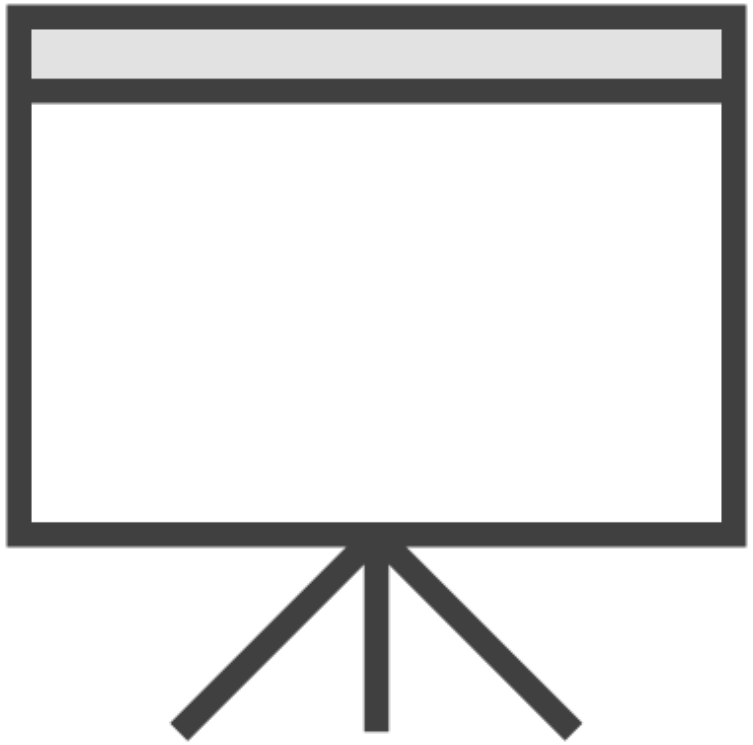
- X-version: “v1”

Version via Accept header

- Accept: “application/json;version=v1”

Version the media types

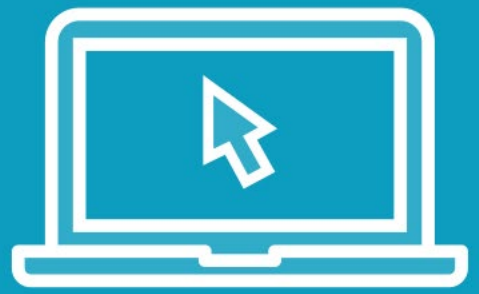
- Accept: “application/vnd.marvin.book.v1+json”



We'll first version our API

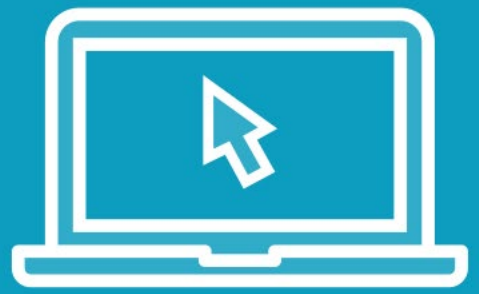
Afterwards, we'll match our OpenAPI specifications to our API versions

Demo



Versioning your API

Demo



Matching OpenAPI specifications to API versions

Protecting Your API

Most APIs require some form of authentication

- From that, we know that the OpenAPI specification should contain information on how to authenticate

HTTP authentication via Authorization header

- Basic, Bearer

API keys

- Headers, query string, cookies

OAuth2

OpenID Connect (discovery spec)

```
"securitySchemes":
   {"basicAuth":
      {"type": "http",
       "description": "Input your username and password to
                       access this API",
      "scheme": "basic"}}
…
"security": [{"basicAuth": []}]
```

Describing API Authentication

Use “securitySchemes” to define all schemes the API supports

Use “security” to apply specific schemes to the whole API or individual operations

HTTP authentication via Authorization header

- Basic, Bearer
- Scheme: **http**

API keys

- Headers, query string, cookies
- Scheme: **apiKey**

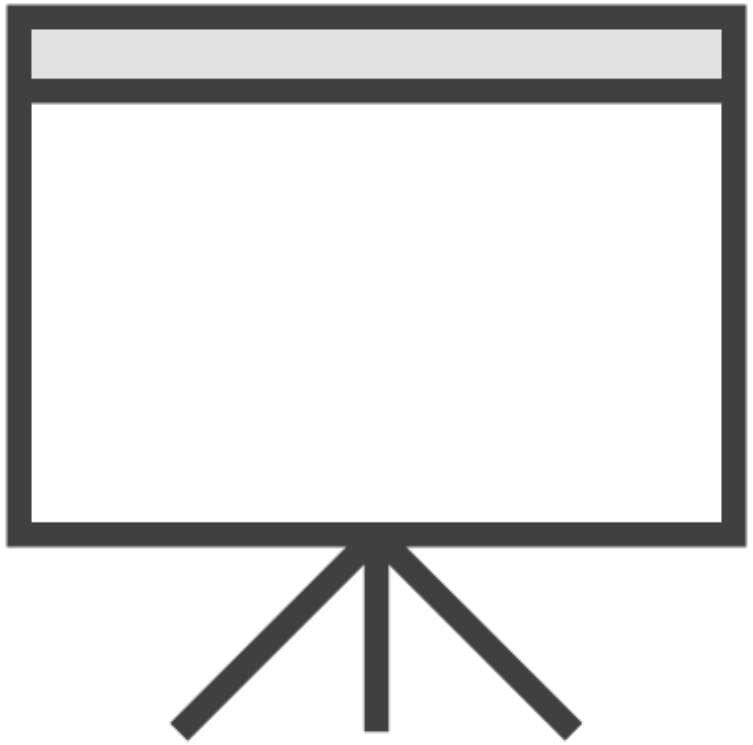
OAuth2

- Scheme: **oauth2**

OpenID Connect (discovery spec)

- Scheme: **openIdConnect**

Protecting Your API



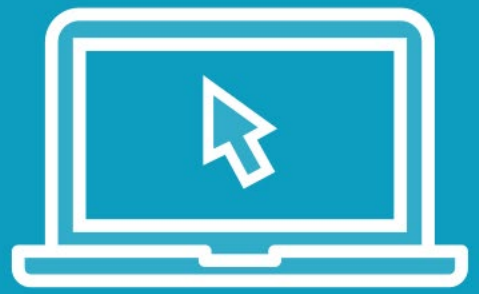
For the demo we'll use Basic Authentication

- Username/password are passed on each request

This isn't the best way to protect an API, but it's easy to setup and allows us to focus on the OpenAPI specification

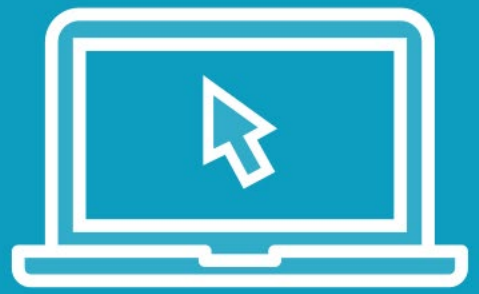
- Principles for other forms of authentication remain the same

Demo



Protecting your API

Demo



Adding authentication support to the OpenAPI specification

Summary

Group operations via the GroupName property on ApiExplorerSettings

Group operations by API version

- VersionedApiExplorer
- DocInclusionPredicate extension

Describe your APIs authentication via security & securitySchemes tags

- AddSecurityDefinition
- AddSecurityRequirement