Versioning and Documenting Your API

Kevin Dockx
Architect

@KevinDockx | www.kevindockx.com

Coming Up

Versioning your API

- Approaches in ASP.NET Core
- Implementation

Coming Up

Documenting your API with OpenAPI (Swagger)

- Incorporating XML comments
- Describing response types and status codes
- Incorporating API authentication

Versioning Your API

As APIs evolve, different versions start to co-exist

- Different versioning strategies exist

Versioning Your API

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`

Versioning Your API

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"
```

Versioning Your API

NuGet package

- https://www.nuget.org/packages/Asp.Versioning.Mvc/

Demo

Supporting versioning

Demo

Versioning your API

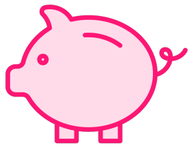
Demo

Supporting versioned routes

Documenting Your API with OpenAPI / Swagger

Public APIs need documentation, but so do in-company APIs

Documentation leads to knowledge leads to adoption



Clear documentation saves time and money

OpenAPI Specification

An OpenAPI specification describes the capabilities of your API, and how to interact with it. It's standardized, and in JSON or YAML format.

Documenting Your API with OpenAPI / Swagger

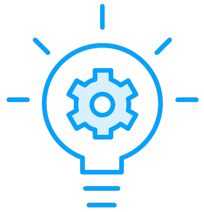
Tools and components can use this specification to generate something from, like a documentation UI

Documenting Your API with OpenAPI / Swagger

Two requirements

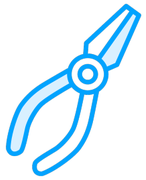
- A tool or component that generates the specification by inspecting our API
- A tool or component that generates a documentation UI from that specification

Clearing up the Terminology Confusion



OpenAPI specification and Swagger specification are the same thing

Swagger is a set of open-source built around that OpenAPI specification



Swashbuckle.AspNetCore helps with working with OpenAPI in ASP.NET Core

Documenting Your API with OpenAPI / Swagger

Swashbuckle.AspNetCore

- Generates an OpenAPI specification from your API
- Wraps swagger-ui and provides an embedded version of it
- https://github.com/domaindrivendev/Swashbuckle.AspNetCore

Demo

Adding Swagger support to a project

Demo

Incorporating XML comments on actions

Demo

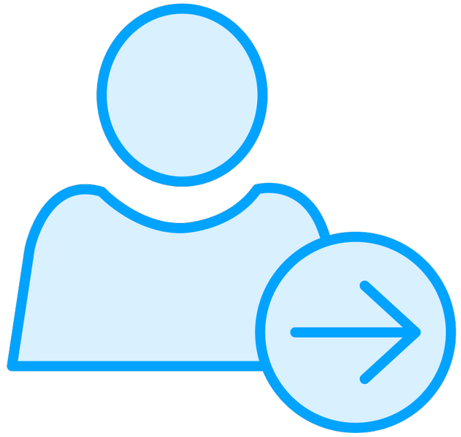
Describing response types and status codes

Demo

Supporting different documentation versions

Documenting API Authentication

Documentation for your API should describe how to authenticate with it, if applicable



Allow user-friendly interaction with an API that requires authentication via swagger-ui

Documenting API Authentication

HTTP authentication schemes (bearer, basic, ...)

Security scheme type: **http**

API keys

Security scheme type: **apiKey**

OAuth2

Security scheme type: **oauth2**

OpenID Connect

Security scheme type: **openIdConnect**

Demo

Adding authentication support to your documentation

Summary

As APIs evolve, different versions start to co-exist

- Avoid breaking consumers that use an older version

Common versioning strategies

- URI versioning
- Custom headers
- Versioned media types

Summary

API documentation is important for both public and private APIs

An OpenAPI specification is a standardized description of your API from which a documentation UI can be generated

Summary

Improve your documentation by

- Using `ActionResult<T>`
- Incorporating XML comments
- Describing response types and status codes
- Incorporating authentication

Be as specific as possible!

Up Next:

Testing and Deploying Your API