REST APIs

When creating REST APIs, you can leverage the OpenAPI-Specification (OAI) (originally known as the Swagger Specification) to describe them:

> The OpenAPI Specification (OAS) defines a standard, programming language-agnostic interface description for HTTP APIs, which allows both humans and computers to discover and understand the capabilities of a service without requiring access to source code, additional documentation, or inspection of network traffic. When properly defined via OpenAPI, a consumer can understand and interact with the remote service with a minimal amount of implementation logic.
>
> Use cases for machine-readable API definition documents include, but are not limited to: interactive documentation; code generation for documentation, clients, and servers; and automation of test cases. OpenAPI documents describe an APIs services and are represented in either YAML or JSON formats. These documents may either be produced and served statically or be generated dynamically from an application.

There are implementations available for many languages like C#, including low-level tooling, editors, user interfaces, code generators, etc. Here you can find a list of known tooling for the different languages: OpenAPI-Specification/IMPLEMENTATIONS.md.

Using Microsoft TypeSpec

While the OpenAPI-Specification (OAI) is a popular method for defining and documenting RESTful APIs, there are other languages available that can simplify and expedite the documentation process. Microsoft TypeSpec is one such language that allows for the description of cloud service APIs and the generation of API description languages, client and service code, documentation, and other assets.

Microsoft TypeSpec is a highly extensible language that offers a set of core primitives that can describe API shapes common among REST, OpenAPI, GraphQL, gRPC, and other protocols. This makes it a versatile option for developers who need to work with a range of different API styles and technologies.

Microsoft TypeSpec is a widely adopted tool within Azure teams, particularly for generating OpenAPI Specifications in complex and interconnected APIs that span multiple teams. To ensure consistency across different parts of the API, teams commonly leverage shared libraries which contain reusable patterns. This makes easier to follow best practices rather than deviating from them. By promoting highly regular API designs that adhere to best practices by construction, TypeSpec can help improve the quality and consistency of APIs developed within an organization.

Resources

- ASP.NET Core web API documentation with Swagger / OpenAPI.
- Microsoft TypeSpec.
- Design Patterns - REST API Guidance

Last update: August 22, 2024