Building Microservices

Andrew Stiefel of F5
Product Marketing Manager | January 19, 2023

APIs are the connective tissue of cloud-native applications – the means by which an application's component microservices communicate. As applications grow and scale, so does the number of microservices and APIs. While this is an unavoidable outcome in most cases, it creates significant challenges for the Platform Ops teams responsible for ensuring the reliability, scalability, and security of modern applications. We call this problem API sprawl and wrote about it in a previous blog post.

As a first attempt to solve API sprawl, an organization might try to use a top-down approach by implementing tools for automated API discovery and remediation. While this is effective in the near term, it often imposes an undue burden on the teams responsible for building and operating APIs and microservices. They either have to rework existing microservices and APIs to address security and compliance issues or go through an arduous review process to obtain the required approvals. This is why many large software organizations adopt a decentralized approach that uses adaptive governance to give developers the autonomy they need.

Rather than putting in last-minute safeguards, a bottom-up approach to the problem is more effective over the long term. The teams building and operating APIs for different microservices and applications are the first to be involved, and often begin by adopting an API-first approach to software development in your organization.

What Is API-First?

APIs have been around for decades. But they are no longer simply "application programming interfaces". At their heart APIs are developer interfaces. Like any user interface, APIs need planning, design, and testing. API-first is about acknowledging and prioritizing the importance of connectivity and simplicity across all the teams operating and using APIs. It prioritizes communication, reuseability, and functionality for API consumers, who are almost always developers.

There are many paths to API-first, but a design-led approach to software development is the end goal for most companies embarking on an API-first journey. In practice, this approach means API are completely defined before implementation. Work begins with designing and documenting how the API will function. The team relies on the resulting artifact, often referred to as the *API contract*, to inform how they implement the application's functionality.

*Explore design techniques to support an API-first approach to software development that is both durable and flexible in Chapter 1 of the eBook **Mastering API Architecture** from O'Reilly, compliments of NGINX.*

developers when they interact with an API. Learn why it's so important for Platform Ops teams to take the API developer experience into consideration.

- **Consistent governance and security –** Cloud and platform architects can organize the API ecosystem in a consistent way by incorporating security and governance rules during the API design phase. This avoids the costly reviews required when issues are discovered later in the software process.
- **Improved software quality –** Designing APIs first ensures security and compliance requirements are met early in the development process, well before the API is ready to be deployed to production. With less need to fix security flaws in production, your operations, quality, and security engineering teams have more time to work directly with the development teams to ensure quality and security standards are met in the design phase.
- **Faster time to market –** With fewer dependencies and a consistent framework for interservice communication, different teams can build and improve their services much more efficiently. A consistent, machine-readable API specification is one tool that can help developers and Platform Ops teams to work better together.

Overall, adopting an API-first approach can help a company build a more flexible, scalable, and secure microservices architecture.

How Adopting a Common API Specification Can Help

In the typical enterprise microservice and API landscape, there are more components in play than a Platform Ops team can keep track of day to day. Embracing and adopting a standard, machine-readable API specification helps teams understand, monitor, and make decisions about the APIs currently operating in their environments.

Adopting a common API specification can also help improve collaboration with stakeholders during the API design phase. By producing an API contract and formalizing it into a standard specification, you can ensure that all stakeholders are on the same page about how an API will work. It also makes it easier to share reusable definitions and capabilities across teams.

Today there are three common API specifications, each supporting most types of APIs:

- OpenAPI – JSON or YAML descriptions of all web APIs and webhooks
- AsyncAPI – JSON or YAML descriptions of event-driven APIs
- JSON Schema – JSON descriptions of the schema objects used for APIs

REST APIs make up the bulk of APIs in production today and the OpenAPI Specification is the standard way to write an API definition for a REST API. It provides a machine-readable contract that describes how a given API functions. The OpenAPI Specification is widely supported by a variety of API management and API gateway tools, including NGINX. The rest of this blog will focus on how you can use the OpenAPI Specification to accomplish a few important use cases.

The OpenAPI Specification is an open source format for defining APIs in either JSON or YAML. You can include a wide range of API characteristics, as illustrated by the following simple API example. Here a simple HTTP `GET` request returns a list of items on an imaginary grocery list.

```
responses:
  '200':
    description: Successfully returned a list
    content:
      schema:
        type: array
        items:
          type: object
          properties:
             item_name:
                type: string
```

Definitions that follow the OpenAPI Specification are both human- and machine-readable. This means there is a single source of truth that documents how each API functions, which is especially important in organizations with many teams building and operating APIs. Of course, to manage, govern, and secure APIs at scale you need to make sure that the rest of the tools in your API platform – API gateways, developer portals, and advanced security – also support the OpenAPI Specification.

Dive deeper into how to design REST APIs using the OpenAPI Specification in Chapter 1 of Mastering API Architecture.

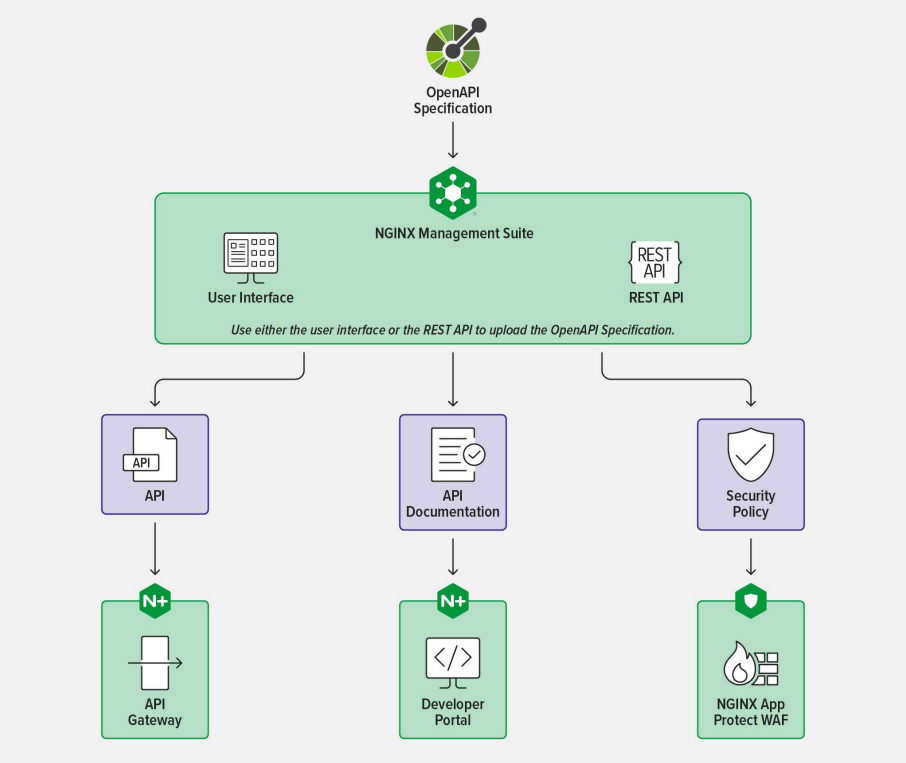
Benefits of Adopting a Common API Specification

Using a common API specification, such as the OpenAPI Specification, has several benefits:

- **Improved interoperability –** A common, machine-readable specification means different systems and clients can consume and use the API contract. This makes it easier for Platform Ops teams to integrate, manage, and monitor complex architectures.
- **Consistent documentation –** The API contract is documented in a standard format, including the endpoints, request and response formats, and other relevant details. Many systems can use the contract to generate comprehensive documentation, providing clarity and making it easier for developers to understand how to use the API.
- **Better testing –** API specifications can be used to automatically generate and run tests, which can help ensure the API implementation adheres to the contract and is working as expected. This can help identify issues with an API before it is published to production.
- **Improved security –** Advanced security tools can use the OpenAPI Specification to analyze API traffic and user behavior. They can apply positive security by verifying that API requests comply with the methods, endpoints, and parameters supported by the API endpoint. Non-conforming traffic is blocked by default, reducing the number of calls your microservices have to process.
- **Easier evolution –** API specifications can help facilitate the evolution of the API contract and application itself over time by providing a clear and standard way to document and communicate changes in both machine- and human-readable formats. When coupled with proper versioning practices, this helps minimize the impacts of API changes on API consumers and ensures that an API remains backward compatible.

Overall, using a common API specification can help to improve the interoperability, documentation, testing, security, and gradual evolution of an API.

- Generate API documentation for the developer portal
- Apply positive security to protect API endpoints



Use the OpenAPI Specification to publish an API to the API gateway and documentation to the developer portal, and to set security policies for the WAF via CI/CD pipelines or the user interface

Publish APIs to the API Gateway

API Connectivity Manager uses the OpenAPI Specification to streamline API publication and management. API developers can publish APIs to the API gateway using either the NGINX Management Suite user interface or the fully declarative REST API. APIs are added to the gateway as API proxies, which contain all the ingress, backend, and routing configurations the API gateway needs to direct incoming API requests to the backend microservice. You can use the REST API to deploy and manage APIs as code by creating simple CI/CD automation scripts with tools like Ansible.

For complete instructions on using the OpenAPI Specification to publish an API, see the API Connectivity Manager documentation.

Generate API Documentation for the Developer Portal

Maintaining documentation is often a headache for API teams. But out-of-date documentation on developer portals is also a major symptom of API sprawl. API Connectivity Manager uses the OpenAPI Specification to automatically generate documentation and publish it to the developer portal, saving API

functionality will be available later in 2023. You can, however, use Instance Manager (another NGINX Management Suite module) and the OpenAPI Specification to write custom policies for your WAF. For additional information, see the documentation for NGINX App Protect WAF and Instance Manager.

Learn more about API security and threat modeling, and how to apply authentication and authorization at the API gateway in Chapter 7 of Mastering API Architecture.

Summary

An API-first approach to building microservices and applications can benefit your organization in many ways. Aligning teams around the OpenAPI Specification (or another common API specification that is both human- and machine-readable) helps enable collaboration, communication, and operations across teams.

Modern applications operate in complex, cloud-native environments. Adopting tools that enable an API-first approach to operating APIs is a critical step towards realizing your API-first strategy. With NGINX you can use the OpenAPI Specification to manage your APIs at scale across distributed teams and environments.

Start a 30-day free trial of NGINX Management Suite, which includes access to API Connectivity Manager, NGINX Plus as an API gateway, and NGINX App Protect to secure your APIs.

NGINX Management Suite, API connectivity, API Connectivity Manager, API strategy, API-first

k

Managing Kubernetes Traffic with F5 NGINX: A Practical Guide

Learn how to manage Kubernetes traffic with F5 NGINX Ingress Controller and F5 NGINX Service Mesh and solve the complex challenges of running Kubernetes in production.

DOWNLOAD NOW

About The Author

Andrew Stiefel

Product Marketing Manager

More Blogs By Andrew Stiefel →

About F5 NGINX

F5, Inc. is the company behind NGINX, the popular open source project. We offer a suite of technologies for developing and delivering modern applications. Together with F5, our combined solution bridges the gap between NetOps and DevOps, with multi-cloud application services that span from code to customer.

Learn more at **nginx.com** or join the conversation by following **@nginx** on Twitter.

k

Try Out NGINX Plus on Your Turf

Get Started

Ask Us a Question

Contact Sales

Secure And Deliver Extraordinary Digital Experiences

F5 NGINX's portfolio of automation, security, performance, and insight capabilities empowers our customers to create, secure, and operate adaptive applications that reduce costs, improve operations, and better protect users.

k