

OpenAPI overview

 cloud.google.com/api-gateway/docs/openapi-overview

API Gateway supports APIs that are described using the OpenAPI specification, version 2.0. Your API can be implemented using any publicly available REST framework such as Django or Jersey.

You describe your API in a **YAML** file referred to as an *OpenAPI document*. This page describes some of the benefits to using OpenAPI, shows a basic OpenAPI document, and provides additional information to help you get started with OpenAPI.

Note: For API Gateway, you use the same Open API syntax for defining your APIs as described in the Cloud Endpoints doc at OpenAPI overview.

Benefits

One of the primary benefits to using OpenAPI is for documentation; once you have an OpenAPI document that describes your API, it is easy to generate reference documentation for your API.

There other benefits to using OpenAPI. For example, you can:

- Generate client libraries in dozens of languages
- Generate server stubs
- Use projects to verify your conformance and to generate samples

Basic structure of an OpenAPI document

An OpenAPI document describes the surface of your REST API, and defines information such as:

- The name and description of the API
- The individual endpoints (paths) in the API
- How the callers are authenticated

If you are new to OpenAPI, take a look at the Swagger basic structure website, which provides a sample OpenAPI document (also referred to as a Swagger specification) and briefly explains each section of the file. The following example illustrates this basic structure:

```

swagger: "2.0"
info:
  title: API_ID optional-string
  description: "Get the name of an airport from its three-letter IATA code."
  version: "1.0.0"
host: DNS_NAME_OF_DEPLOYED_API
schemes:
  - "https"
paths:
  "/airportName":
    get:
      description: "Get the airport name for a given IATA code."
      operationId: "airportName"
      parameters:
        -
          name: iataCode
          in: query
          required: true
          type: string
      responses:
        200:
          description: "Success."
          schema:
            type: string
        400:
          description: "The IATA code is invalid or missing."

```

Note: For API Gateway, use of the `host` property in your API definition is optional. You can either omit the `host` property entirely from the API definition or set it to the DNS name of the deployed API. API providers often set it to the DNS name when sharing the OpenAPI Spec with their API consumers. Do not set the `host` property to **null** or an empty value, as this will result in an error.

In addition to the basic structure, use the `openapi.yaml` file to configure:

- The `title` field with the name of your API and an *optional-string* with a brief description.
- How to configure a path to use an API key
- Various security schemes for authentication
- OpenAPI extensions

Generating an OpenAPI document

Depending on what language you are using, you might be able to generate an OpenAPI document. In Java, there are open source projects for both Jersey and Spring that can generate an OpenAPI document from annotations. There is also a Maven plugin. For Python and Node developers, OpenAPI.Tools might be an interesting project, or awesome-openapi3.

The OpenAPI community is continually developing tools to help in the composition (and, for some languages, automatic generation) of OpenAPI documents. See the The OpenAPI Specification for more.

What's next

Note: Because you use the same Open API syntax for the API Gateway Beta as you used for Cloud Endpoints, these links refer to locations in the [Cloud Endpoints](#) documentation.
Rate and review