REST APIs

Representational state transfer (REST) is a fussy term for the simple process of exchanging data over the internet without changing its state. An application programming interface (API) is a platform for requesting and returning information. A REST API, then, is a flexible framework because the objects within it are free agents. They can serve any function and respond to a range of methods for finding, reading, changing, or deleting them. Stateless systems are easier to test, run, maintain, and update than their more rigid counterparts.

REST architecture centers on three principles:

1. **Uses HTTP Protocol.** Makes requests and responses using existing HTTP protocol.
2. **Remains Stateless.** Remains stateless. Client requests carry all the necessary data. The server need not change its state in order to store the information necessary to respond to such requests.
3. **Consumes Server Resources.** Relies on endpoints, which are URIs representing server resources. A resource is any scrap of info that can be named, from research results to cat videos.

The OpenAPI Specification

Think of all the building codes a contractor must follow when constructing a home. In the United States, countertops are always 48 inches above the floor. Electrical outlets are always 120 volts. Water drainage pipes must slope ¼ inch per foot. Without standards like these, electrical systems would fail, bathrooms and kitchens would flood, cabinets wouldn't fit, and you'd have a hard time using your Keurig. Likewise, API engineers follow certain rules (see table). Our API is based on rules spelled out in the OpenAPI (OAS) 3.0 specification.

A specification is just what it sounds like—a technical blueprint or template for building something. The OpenAPI specification (OAS) standardizes API archietcture. The OAS spec was donated to the public OpenAPI Initiative by its developer, Swagger. As a result, engineers often call it the "Swagger spec." This spec describes the API's services and capabilities in a human- and machine-readable format. Instead of specifying lightswitch heights and firewall placement, it covers version numbers, data structures, syntax, and many other attributes.

Selected Specs Covered in OpenAPI 3.0

Item	Description	Example(s)
Symbols and their meaning	Curly braces, for instance, indicate that the value within is a replaceable path parameter.	GET /offer/{offerId}
Document structure	A single document or a series of related sections.	openapi.yaml or openapi.json
Data types	A set of categories outlined in the JSON Schema Spec.	string boolean
Data format	The standardized way in which information is to be presented.	float (for the data type "number")
Status codes	A set of values defined in the IANA Status Code Registry.	404 Not Found
Versioning	A way of labeling API releases. Smart versioning can minimize the effect of dependencies. Version numbers are incremented using the *major.minor.patch* pattern proposed in Semantic Versioning 2.0.0.	4.0.1
Naming conventions	Standards governing case and case sensitivity	Camel case: termsOfService