What is REST API in simple terms?

A RESTful API is an architectural style for an application program interface (API) that uses HTTP requests to access and use data. That data can be used to GET, PUT, POST and DELETE data types, which refers to the reading, updating, creating and deleting of operations concerning resources

REST API (also called a “RESTful” API) is a specific type of API that follows these guidelines. REST stands for Representational State Transfer. This means that when a client requests a resource using a REST API, the server transfers back the current state of the resource in a standardized representation

For example, a REST API would use a GET request to retrieve a record, a POST request to create one, a PUT request to update a record, and a DELETE request to delete one. All HTTP methods can be used in API calls. A well-designed REST API is similar to a website running in a web browser with built-in HTTP functionality.

Difference between API and REST API?

API is basically a set of rules that developers create on the server-side to enable programs to communicate with each other. And REST determines how the API will look and work and what architectural pattern developers will follow to build it.

What does JSON stand for?

JavaScript Object Notation

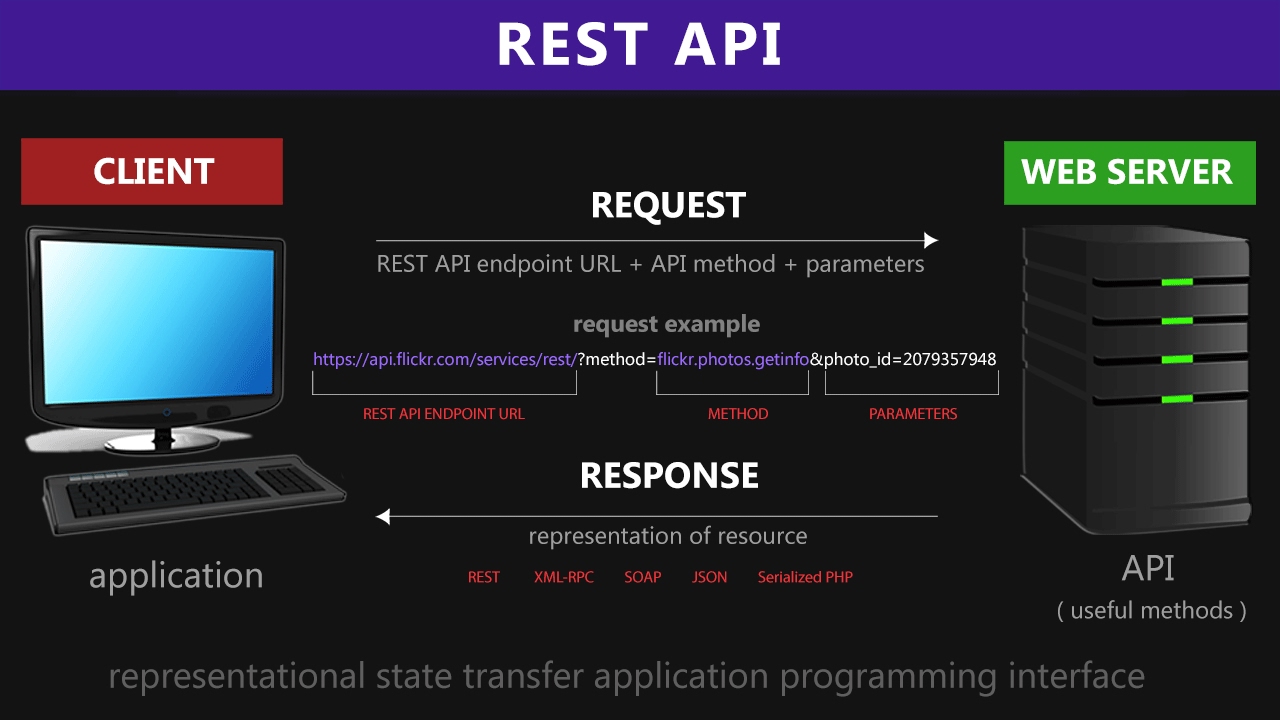
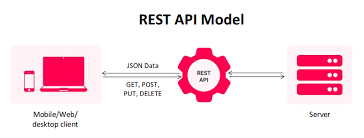
JavaScript Object Notation, more commonly known by the acronym JSON, is an open data interchange format that is both human and machine-readable.

JSON is used in REST API

With the increasing popularity of REST, the lightweight and human-readable JSON format has also quickly gained traction, as it's so well suited for quick data exchange. JSON stands for JavaScript Object Notation. It's an easy-to-parse and lightweight data-interchange format.

JSON is not an API but a data format webservices and programs use to communicate to each other.

JSON and XML are the two standards for sending and receiving data in REST APIs. Web programming languages such as Python, JavaScript, Ruby on Rails, and Java all have tools for parsing and working with XML and JSON.



Serializers allow complex data such as querysets and model instances to be converted to native Python datatypes that can then be easily rendered into JSON, XML or other content types. Serializers also provide deserialization, allowing parsed data to be converted back into complex types, after first validating the incoming data. ( for more details : <https://www.django-rest-framework.org/api-guide/serializers/> )