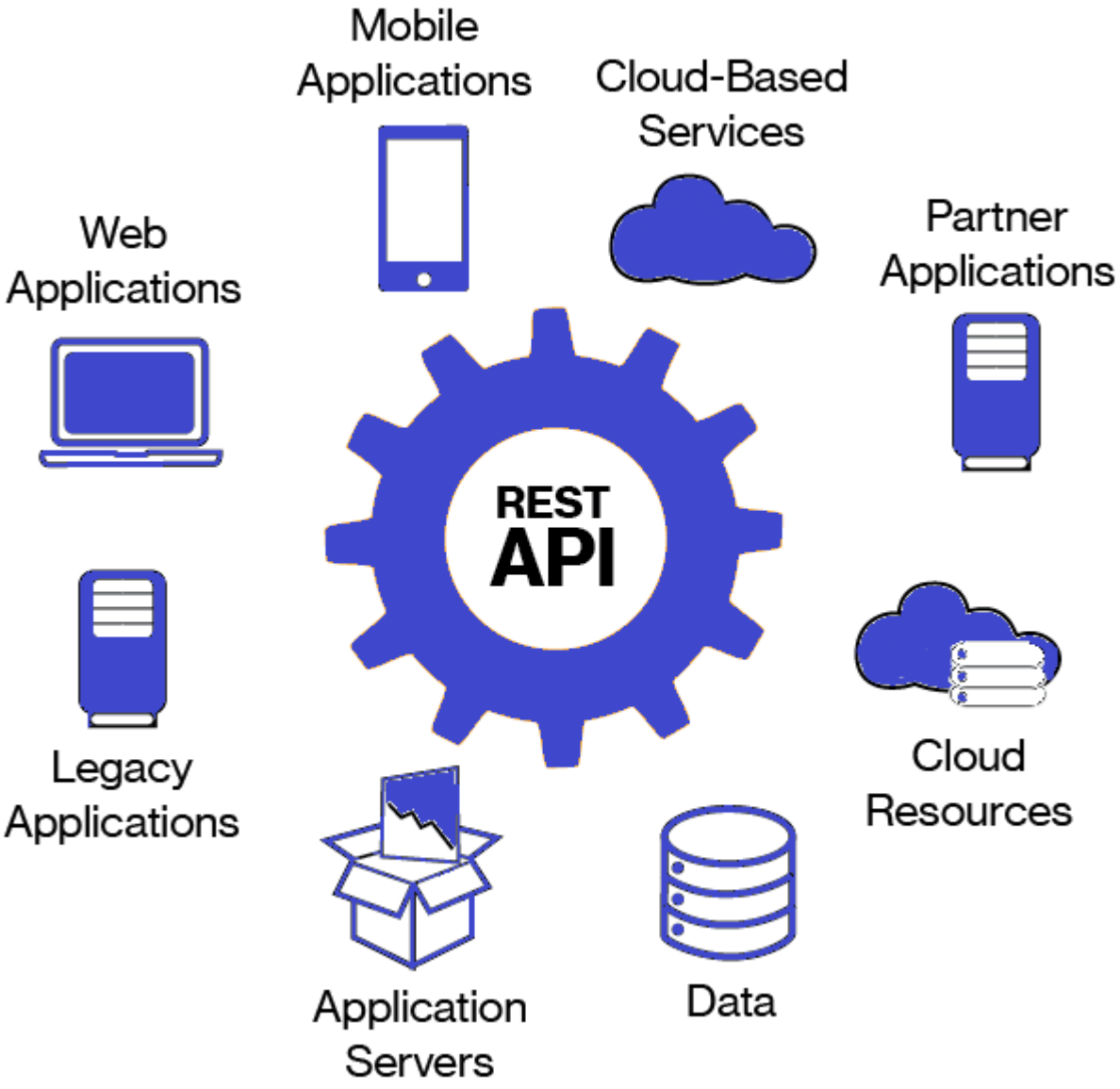

By

Herbert G.Sri Harsha

11.02.2021

Explaining Concepts

Understanding **REST API** concepts.

Accessing of web pages over the internet has increased a lot from the day it was invented, though we hardly know where the data for these web pages come from. A storage device called **Server** is responsible for sending the data.

One who communicates with the server in extracting the data is called the **Client.** This way of communication method is called the Client-Server model. Desktops, laptops, tablets, and smartphones are the devices that are involved in this model.

Why Rest API?

Let us start illustrating in detail about the REST API.

REST stands for **Representational State Transfer** communication protocol. It is lightweight, highly scalable, and maintainable architecture. It is commonly used to create APIs for web-based applications that specifically works with media components, files, or even objects on a hardware device. It deploys multiple standards, so it takes fewer resources and less bandwidth. For better understanding let's illustrate with a simple example, to book movie tickets in any of the apps we might have observed that the application needs a lot of input data, as the data present in the application is never static. The data contained in the app needs to update frequently from the server/webserver. Typically, when a client sends a **Request** through an API and the server **Response** back in the form of HTML which is not in an easily readable format, in spite we prefer data returned to be in a structured format that might be in JSON or XML.

- JSON is represented in the form of an object where the object values are returned to the user.
- XML is represented in the form of hierarchical data structure in which data can be returned.

This ticket booking process looks so simple but a lot of methods such as GET, POST, PUT, and DELETE are involved to get the required information from the server, but in some instances where data continuously keeps on updating and many methods are being called leads to data crash or complicated. To avoid such scenarios REST API is introduced.

How REST API works?

Basing upon the client request REST creates a state of an object and thereafter and sends the value of the object.

Features of REST API

The following are the main feature of REST API.

- Simpler than SOAP
- Proper Documentation
- Proper logging Error Messages

Principles of REST API

The following are the principles of REST API.

- Stateless: Requests sent from client to server contains all the information that is required to make the server understand.
- Client-Server: An uniform interface that separates the clients from the servers, this way of separating help in improving the user's interface portability across multiple platforms as well as scalability of the server components.
- Uniform Interface: To get the uniformity of the application rest has defined 4 interface constraints

    - ❖ Resource Identification
    - ❖ Source Manipulation using Representation
    - ❖ Self Descriptive Messages
    - ❖ Hypermedia.

- Cacheable: To perform better applications are often made as cacheable it is done by labeling the response from the server as cacheable or non-cacheable, it also helps in resue of steal data.
- Layered Systems: This architecture allows an application to be more stable by limiting the component behavior it enables load balancing and provides shared caches for promoting scalability. it also helps in enhancing the application's security as components in each layer can not interact beyond the next immediate layer.
- Code on Demand: It is an optional constant it simplifies the clients by creating a smart application that doesn't rely on its own code structure.

Methods of Rest API

We use web technologies that work on crude applications(creating a resource). A resource is what you want to do, use the HTTP methods to create a resource.

- Creating -> Post
- Reading -> GET
- Updating -> PUT
- Delete -> Delete