**Designing REST APIs: Part 2**

*Posted On September 24, 2019*

The client makes the request, Rest APIs receives the request, gathers and parses the data, returns the data and response header to client.

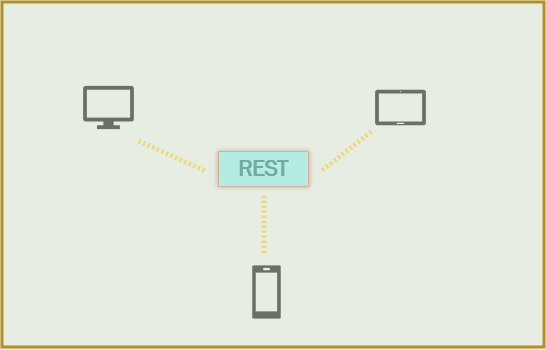
In REST API, REST is Representational State Transfer and API is Application Programming Interface.

“REST refers to a group of design constraints that bring about efficient, reliable and scalable systems.”

REST is data architechture and design methodology which receives set of standard methods called as verbs and returns standardized structure data typically JSON or XML. An API is a collection of tools used to access and work with REST resources through URI and Verbs including GET, PUT, POST and DELETE.

**The 6 Constraints of REST**

1. Client-server architechture. The client manages user interface concerns while the server manages data storage concerns.  
   There is a complete separation between the content and its presentation & interaction.
2. Statelessness. No client context or information like state can be stored on the server between requests.
3. Cacheability. All REST responses must be clearly marked as cacheable or not cacheable.
4. Layered system. The client cannot know, and shouldn’t care, whether it’s connected directly to the server or an intermediary. This ensures scalability and also helps in security.
5. Code on demand. Servers are allowed to transfer executable code like JavaScript and compiled components to clients.
6. Uniform Interface. This is categorised in 4 more constraints 6.1- Resource identification in request: The URI request must specify what resource it is looking for and what format the response should use. 6.2- Resource manipulation through representations: It means, once a client has a representation of a resource, it can modify or delete the resource. 6.3- Self-descriptive messages:Each representation must describe its own data format. If you are receiving JSON then it should have its media type set to JSON.Without this information data cannot be reliably parsed. 6.4- Hypermedia as the engine of application state: Once a client has access to a REST service, it should be able to discover all available resources and methods through the hyperlinks provided. The return representation should include hyperlinks to all resources and methods available.

If Web API meets this 6 constraints, it can be considered as RESTful APIs. Thus when the Rest Service run on the web over HTTP to give us access to web resource that meets 6 constraints, we call it RESTful API. 

Client consume the REST API by creating and sending requests and receiving and parsing responses.