What is REST API ?

REST (Representational State Transfer) API is an architectural style that defines a set of constraints to be used for creating networked applications. It relies on HTTP protocol and provides methods for HTTP verbs, such as GET, POST, PUT, DELETE, etc.

The REST API relies on the concept of "resources", which are identified by URIs (Uniform Resource Identifiers) and can be accessed using standard HTTP methods.

REST API's have several advantages, including:

Statelessness: Each request from a client to a server must contain all the information needed to understand and process the request. The server cannot store any information about the client's state between requests.

Client-Server: The client-server model separates the user interface concerns from the data storage concerns, resulting in systems that are more scalable and maintainable.

Cacheable: The REST API can use the HTTP cache to cache server responses. This can improve the performance of the system by reducing the number of requests to the server.

Layered System: The architecture of a REST API allows it to be composed of multiple layers, each of which is independent of the others. This enables greater flexibility in the system design and allows for the implementation of additional features, such as load balancing and security.

There are numerous types of APIs, making it difficult for new developers to differentiate between each kind. In particular, Representational State Transfer (REST) is a software architectural style that developers apply to web APIs. REST APIs provide simple, uniform interfaces because they can be used to make data, content, algorithms, media, and other digital resources available through web URLs.

To understand how REST APIs work, it is critical to understand resources. A resource can be any information that could be named, such as a document or image, a collection of other resources, a non-virtual object, and more. Meanwhile, REST uses a resource identifier to recognize the specific resource involved in an interaction between components.