

Understanding REST

An architectural style for creating networked applications is called REST. It is predicated on a set of guidelines that specify how programs ought to communicate with one another.

Applications that are RESTful are scalable, dependable, and user-friendly.

The key principles of REST are:

Resources: The central idea of REST is resources. Anything that has a URI (Uniform Resource Identifier) assigned to it is considered a resource. Resources can be anything, including data structures, documents, and photos.

uniform interface: A uniform interface is used to manipulate all resources. This indicates that all resources are manipulated using the same set of operations (such as GET, POST, PUT, and DELETE).

Statelessness: Applications that use REST are stateless. This indicates that every request a client sends to a server is complete and contains all the data the server requires to handle the request. In between requests, the server does not save any state.

RESTful applications are layered in a layered system. This indicates that the application is split up into several layers, each of which is in charge of a particular task. This facilitates the design, development, and maintenance of the application.

Applications that use RESTful techniques can be cached. This implies that proxies and web browsers, among other intermediaries, are able to cache responses. This may enhance the application's performance.