**Representational State Transfer (REST) Web Services**

* It’s an “architectural style” of client-server application, centered around the “transfer” of “representations” of “resources” through requests and response.
* In the REST architectural style, data and functionality (i.e Web Services Methods are considered as resources and are accessed using Uniform Resource Identifiers (URIs), typically hyperlinks on the Web.
* The representation of that resource might be
* an XML document
* a JSON File
* a Simple Text
* an Image file
* an HTML page, etc.,
* A client application might
* retrieve a particular representation
* modify the resource by updating its data or
* Delete the resource entirely.
* The REST architectural style is designed to use a stateless communication protocol, typically HTTP
* The following principles encourage RESTful applications to be simple, lightweight, and fast

1. Resource Identification through URI:

Resources in RESTful web services are identified by URI’s

1. **Uniform Interface:**

Resources are manipulated using a fixed set of 4 operations

1. create
2. read
3. update
4. delete

These operations can be perform using below HTTP Metods respectively

1. PUT (creates a new resource)
2. GET (retrieves the current state of a resource in some representation)
3. POST (transfers a new state onto a resource OR Update the existing resource)
4. DELETE (Delete an existing resource)
5. Self-descriptive messages:

Resources are decouple from their representation so that their content can be accessed in a variety of formats, such as HTML, XML, Plain text, PDF, JPEG, JSON, and other formats.

* Hence RESTful web services are loosely coupled, lightweight web services they are well suited for creating APIs for clients spread across the internet.