**HTTP Methods**

* GET Method:

A GET request is used to retrieve a data.This the default method.If you are not specifying the method it will take GET as a request method.This the common method.

Query string i.e., name/value pairs is sent in the URL of a GET request.

For eg:/test/demo.php?name1=value1&name2=value2

GET Request should never be used when dealing with sensitive data.

* POST Method:

POST is used to send some data to a server.for example:file update

It is used to create a resource.

If you are dealing with sensitive data then use POST method.

Data sent to the server with POST is stored in the request body of http request.

For example:HTTP POST http://www.appdomain.com/users/123/accounts

* PUT method:

PUT method is used to request the server to store the included body at the location specified by given url.PUT is used to update the existing resource.

For example:HTTP PUT http://www.google.com/users/123

* DELETE Method:

DELETE method is used to request a server to delete a file from the location given by specified url.It is idempotent.If you delete a resource it’s removed from collection of resources.Repeatedly calling delete on same resource will not change the outtcome because it is already removed from collection of resource it will give the status as (404)file not found.

For Example:HTTP DELETE http://www.google.com/users/123

* HEAD Method:

HEAD method functions similar to the GET method.Transfers the header line and section only but without the response body.

* CONNECT Method:

The CONNECT method is used by the client to establish a network connection to a web server over HTTP.

For example:CONNECT [www.google.com](http://www.google.com) http/1.1

* OPTIONS Method:

OPTIONS method is used by the client to find out the HTTP methods and other options supported by a web server. The client can use a URL for the OPTIONS method, or an asterisk (\*) to refer to the entire server.It describes the communication option for the resource.

For Example:OPTIONS \* http/1.1

* TRACE Method:

TRACE method is used to echo the contents of a request back to the requester which can be used for debugging purpose at the time of development.

For example:TRACE /http/1.1

**HTTP status codes:**

**1xx:In this status is informational**

It means request has been received and process is continuing.

1) 100:continue

Only a part of the request has been received by the server, but as long as it has not been rejected, the client should continue with the request.

2) 101:Switching Protocol

Server switches protocol.

**2xx:status is successful**

1) 200:OK

The request is OK.

2) 201:Created

The request is complete,the new resource is created.

3) 202:Accepted

The request is accepted for processing, but the processing is not complete.

4) 203:Non Authoritative Information

The information in a entity header is from local or third party not from original server.

5) 204:No content

A status code and a header are given in the response, but there is no entity-body in the reply.

**3xx:Redirection**

It means further action must be taken in order to complete the request.

1) 300:Choices

A link list. The user can select a link and go to that location. Maximum five addresses .

2) 307:Temporary redirect

The requested page has moved temporarily to a new url.

**4xx:Client Error**

It means request contains incorrect syntax.

1) 400:Bad request

The server did not understand the request.

2) 403:Forbidden

The client does not have resource to access this permission.

3) 404:Not Found

The resource could not be found at this time.It might be deleted or does not exist yet.

4) 411:Content Length

The "Content-Length" is not defined. The server will not accept the request without it.

5) 408:Request Timeout

The request took longer than the server was prepared to wait.

5xx:Server Error

1) 500:Internal Server Error

The request was not completed. The server met an unexpected condition.

2) 503:Service Unavailable

The request was not completed. The server is temporarily down.

**How Rest works?**

Rest is a REpresentational state transfer.REst are the rules by which we decide an application structure.It is an language independent architectural style for developing web services. Rest based applications can be written in any language be it Java,kotlin, .net,Javascript. HTTP is the protocol used in rest.The 4 methods GET,PUT,POST,DELETE are used in rest based web services.Rest works entirely on a client-server architecture.Web exchange and the communication can be done by using the json formats including XML,HTML,etc.Client send the request to the server.IN between there is DNS resolver which is used to find the IP related to that request.then that IP will send to the server.According to that,server will give the response to client.Communication between client-server must be stateless such that each request comes from client to server must contain the necessary information about the request,cannot take the advantage of stored context stored on the server.

In order to improve the network efficiency,add cache constraints to form the client-cache-stateless-server style.A cache acts as mediator between client and server.If the response are considered cacheable,use that response for the later requests that are equivalent.

There is a uniform interface between components.It is a layered system like there will be hierarchical layers.