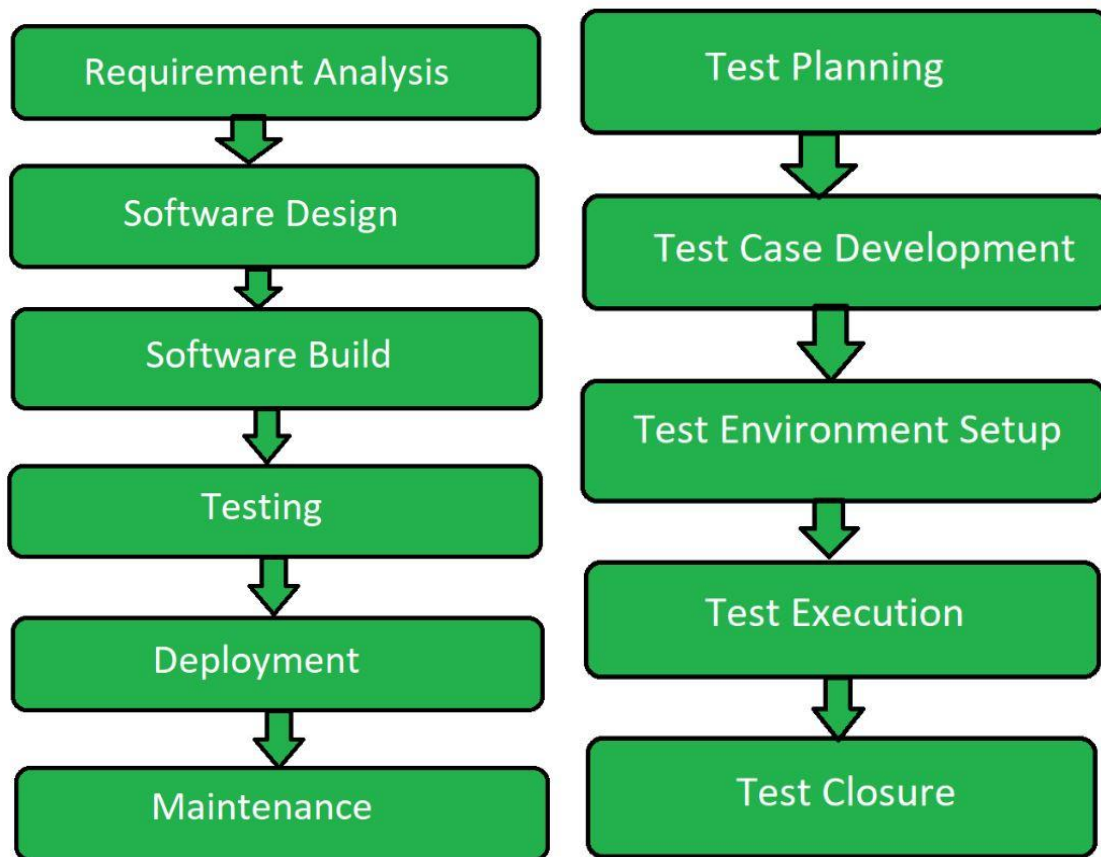


1)Difference between SDLC and STLC

Software Development Life Cycle (SDLC) is a sequence of different activities performed during the software development process.



Software Testing Life Cycle (STLC) is a sequence of different activities performed during the software testing process.

Difference between SDLC and STLC:

SDLC

SDLC is mainly related to software development.

Besides development other phases like testing is also included.

SDLC involves total six phases or steps.

In SDLC, more number of members (developers) are required for the whole process.

STLC

STLC is mainly related to software testing.

It focuses only on testing the software.

STLC involves only five phases or steps.

In STLC, less number of members (testers) are needed.

SDLC

In SDLC, development team makes the plans and designs based on the requirements.

Goal of SDLC is to complete successful development of software.

It helps in developing good quality software.

SDLC phases are completed before the STLC phases.

Post deployment support , enhancement , and update are to be included if necessary.

Creation of reusable software systems is the end result of SDLC.

STLC

In STLC, testing team(Test Lead or Test Architect) makes the plans and designs.

Goal of STLC is to complete successful testing of software.

It helps in making the software defects free.

STLC phases are performed after SDLC phases.

Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts.

A tested software system is the end result of STLC.

2) Software Engineering | Verification and Validation

Verification and Validation is the process of investigating that a software system satisfies specifications and standards and it fulfills the required purpose. **Barry Boehm** described verification and validation as the following:

Verification: *Are we building the product right?*

Validation: *Are we building the right product?*

Verification:

Verification is the process of checking that a software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we have.

Verification is **Static Testing**.

Activities involved in verification:

1. Inspections
2. Reviews
3. Walkthroughs
4. Desk-checking

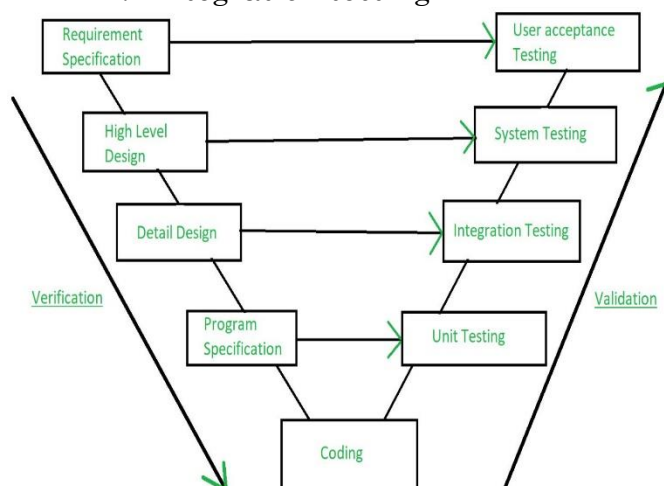
Validation:

Validation is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e. it checks what we are developing is the right product. it is validation of actual and expected product.

Validation is the **Dynamic Testing**.

Activities involved in validation:

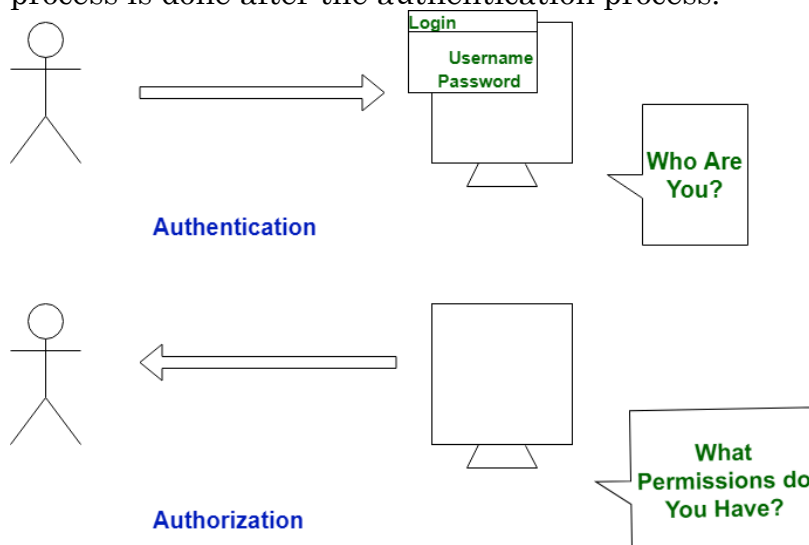
1. Black box testing
2. White box testing
3. Unit testing
4. Integration testing



Note: Verification is followed by Validation.

3) Difference between Authentication and Authorization

Both Authentication and Authorization are unit utilized in respect of knowledge security that permits the safety on an automatic data system. Each area unit terribly crucial topics usually related to the online as key items of its service infrastructure. However, each the terms area unit terribly completely different with altogether different ideas. whereas it's true that they're usually employed in an equivalent context with an equivalent tool, they're utterly distinct from one another. In authentication process, the identity of users are checked for providing the access to the system. While in authorization process, person's or user's authorities are checked for accessing the resources. Authentication is done before the authorization process, whereas authorization process is done after the authentication process.



let's see the difference between authentication and authorization:

S.NO	Authentication	Authorization
1.	In authentication process, the identity of users are checked for providing the access to the system.	While in authorization process, person's or user's authorities are checked for accessing the resources.
2.	In authentication process, users or persons are verified.	While in this process, users or persons are validated.
3.	It is done before the authorization process.	While this process is done after the authentication process.
4.	It needs usually user's login details.	While it needs user's privilege or security levels.

S.NO	Authentication	Authorization
5.	Authentication determines whether the person is user or not.	While it determines What permission do user have?
6.	Generally, transmit information through an ID Token.	Generally, transmit information through an Access Token.
7.	Example : Employees in a company are required to authenticate through the network before accessing their company email.	Example : After an employee successfully authenticates, the system determines what information the employees are allowed to access.

4) Understanding Test Management Process

- Last Updated : 20 Oct, 2021

Prerequisite: Software Testing

As like design and development, software testing is also one important part of SDLC (Software Development Life Cycle). During testing phase of software development, testing activities are managed well to complete the testing process smoothly and on time as well.

Test Management :

Test Management is a process where testing activities are managed to ensures high quality and high end testing of software application. This method consists of tracking, organization, controlling process, checks visibility of the testing process in order to deliver the high quality software application. It makes sure the software testing process run as expected.

Test Management Process :

It is a software process which manages start to end of the all software testing activities. This management process provides planning, controlling, tracking and monitoring facilities throughout the whole group cycle, these process includes several activities like test case design and test execution, test planning etc. It also gives initial plan and discipline specifications to the software testing process.

Responsibilities:

- Works in collaboration test analyst and technical test analyst to select and customizes the appropriate templates also establishes standards.
- Provides all facilities to keep the track and controls the testing throughout the project.
- Gives clear concept of understanding of testing activity of prior upcoming project and also post ones

Test management process has **two main part of test Management Process:**

Planning :

- Risk analysis
- Test Estimation
- Test planning

Execution :

- Testing Activity
- Issue Management
- Test report and evolution

Activity of test process :

1. **Test plan –**
Rough sketch are served in order to test plans to convey the process of testing. Gives clear vision about complete testing process.
2. **Test design –**
Test design affords the implementation process of testing.
3. **Test execution –**
It shows the actual system result against the expected result during test execution.
4. **Exit criteria –**
It gives the signal when to stop the test execution process.

5. Test reporting –

Test reporting picturizes test process and result for the particular testing cycle.

Tools :

Some commonly used tools are listed below-

1. qTest
2. Zephyr
3. Test Collab
4. XQual
5. TestRail
6. Testpad

Advantages :

- Reuses current test and compares the results with last trails .
- Prevents duplicate issues
- Enables conceptual graphical visualization regarding reports
- Reports errors via email
- Combines easily with automation tools and C
- Deals with tests accordingly to the requirements effortlessly
- Integrated quickly with slack
- Deals test based on cycle and sprints

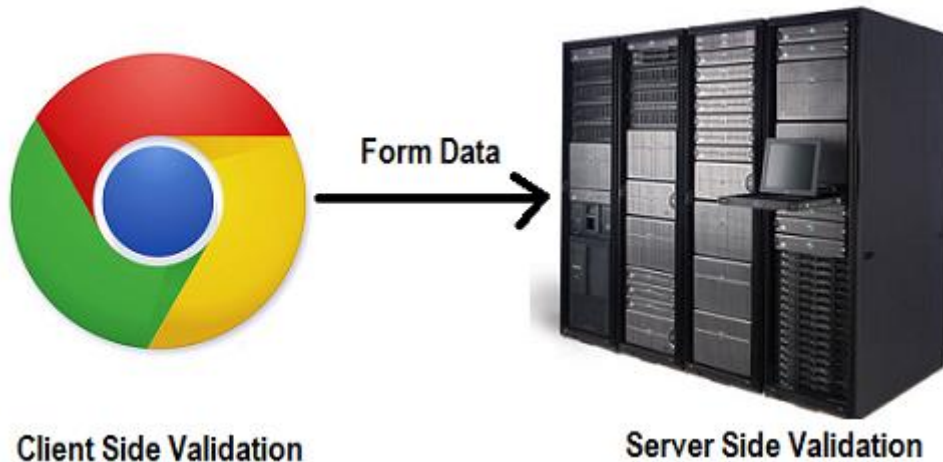
Disadvantages :

- Its not cost effective
- Doesn't support cloud based application
- No mobile app support

5) Client side validation and server side validation

Client side validation Vs server side validation

There is a usual question that which type of validations is better or best? Server side validation or Client side Validation ?



Validations can be performed on the server side or on the client side (web browser). The user input validation take place on the Server Side during a post back session is called Server Side Validation and the user input validation take place on the Client Side (web browser) is called Client Side Validation. Client Side Validation does not require a postback. If the user request requires server resources to validate the user input, you should use Server Side Validation. If the user request does not require any server resources to validate the input , you can use Client Side Validation.

Server Side Validation

In the Server Side Validation, the input submitted by the user is being sent to the server and validated using one of server side scripting languages such as ASP.Net, PHP etc. After the validation process on the Server Side, the feedback is sent back to the client by a new dynamically generated web page. It is better to validate user input on Server Side because you can protect against the malicious users, who can easily bypass your Client Side scripting language and submit dangerous input to the server.

Client Side Validation

In the Client Side Validation you can provide a better user experience by responding quickly at the browser level. When you perform a Client Side Validation, all the user inputs validated in the user's browser itself. Client Side validation does not require a round trip to the server, so the network traffic which will help your server perform better. This type of validation is done on the browser side using script languages such as JavaScript, VBScript or HTML5 attributes.

For example, if the user enter an invalid email format, you can show an error message immediately before the user move to the next field, so the user can correct every field before they submit the form.

Mostly the Client Side Validation depends on the JavaScript Language, so if users turn JavaScript off, it can easily bypass and submit dangerous input to the server . So the Client Side Validation can not protect your application from malicious attacks on your server resources and databases.

As both the validation methods have their own significances, it is recommended that the Server side validation is more **SECURE!**

6) HTTP request-response flow: for dummies

Hi meows! In this tutorial we'll be understanding the flow of client-server architecture, format of the request and response message as well as HTTP methods and response status.

Types of network architecture:

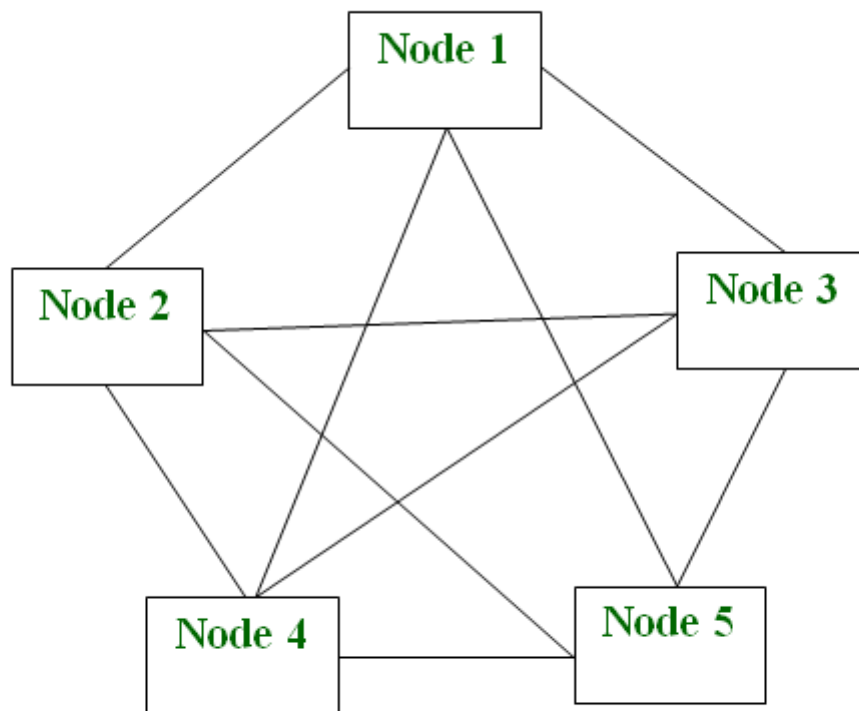
Network architecture essentially describes the design of a computer network. There are different ways in which computers are connected in a network.

1. Peer to peer architecture

In peer to peer architecture computers are directly linked to each other and can share data with each other. In more technical terms, The peer to peer computing architecture **contains nodes that are equal participants in data sharing**.

Drawbacks

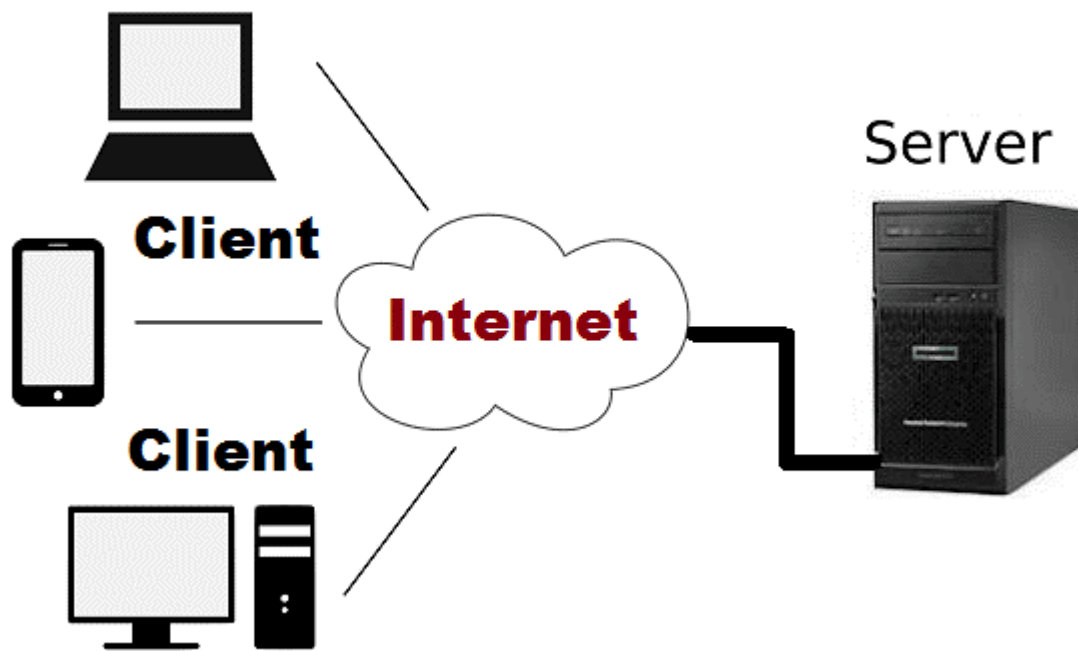
- we cannot backup data because there is no server
- only feasible for say; 10–20 computers



P2P Architecture

2. Client server architecture / Request-response architecture

In client-server architecture, client makes a request to the server, and server hosts / delivers and manages the response, and returns it to the client. In easier terms, client requests for resources (typing the URL in browser) the server, fetches resource from database and returns the response.



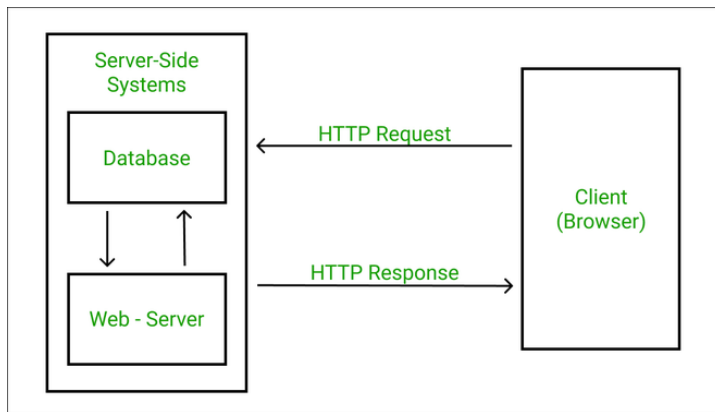
Advantages

- data can be easily backed up in the server
- increases speed of resource sharing

Flow of an HTTP Request / Response flow

working of websites:

To reiterate, this is how the web flow looks like when frontend and backend technologies are integrated in the client-server architecture.



1. Client requests for website from the web server.
2. Web server does processing using a dynamic language (PHP/ Python/ Node Js) and returns the response to client in the form of html, css and JS.

How Does the Server process the HTTP Request?

We run an application/web server (like Apache / Nginx) that's responsible for handling these incoming HTTP requests.

An application/web server accepts and fulfills requests from clients for dynamic content (content that needs processing at the server side).

These application/ web servers keep listening for incoming requests and respond to those requests.

There must be multiple processes running on a server, so how does our request reach Apache?

Just like any device, there might be multiple processes — that's where network ports are helpful. **So processes like Apache occupy network ports and listen to those ports.** A port can't be used by multiple processes.

HTTP protocol has some predefined rules shared universally. One of those is the default port for HTTP is 80, HTTPS is 443

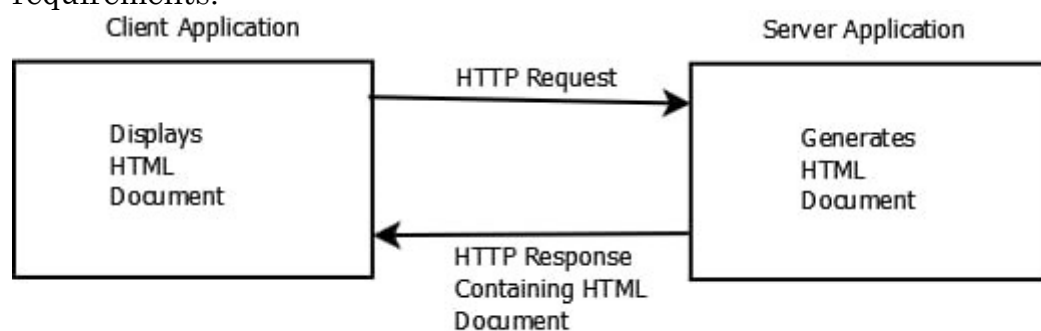
3. CSS & Js will be embedded in HTML bundle will be send to your browser at client-side.

HTTP messages are how data is exchanged between a server and a client. There are two types of messages:

1. requests sent by the client to trigger an action on the server.
2. responses, the answer from the server.

How does client-server communicate?

Client-server, refers to a communication model that links several devices through a network. The client machine/ application, in this context, makes a service requests to the server machine/application, which is responsible for meeting these requirements.



Client requests a resource (HTML document) from the server, which generates the html document and provides a response to the client. In simpler terms, server provides the client (if available), whatever it requested for.

HTTP Request

- Client requests data through typing the URL (https://facebook.com) in browser [AKA **Request message**] — HTTP Request

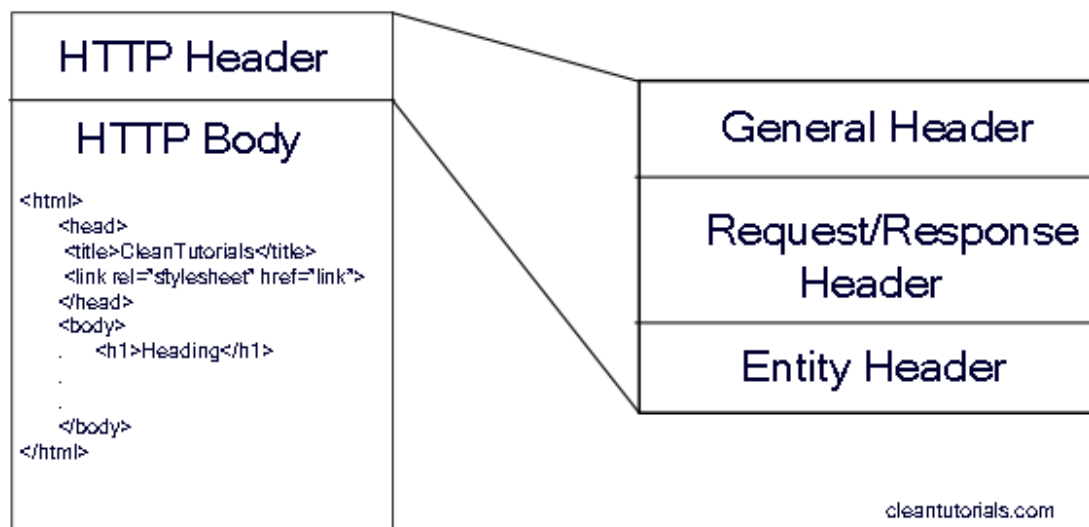
HTTP Response

- Request goes to that URL server (Facebook server) and returns the request(fb page) [AKA **Response message**] — HTTP Response

Format of request and response message:

This is the format in which request message is sent from client > server over the network.

HTTP Request/response



HTTP Request

- **General header** (contains information about the date and time client requested the request message)

- **Request/response header** (contains URL/data we want to send and HTTP method we choose to send the data with, authentication(secret key), content type of the data we choose to send, content size)
- **Entity header** (contains information about the html code (code length, programming language used))

Request HTTP Body

- Request HTTP Body contains information about HTML, CSS, JS that were sending to the server

This is the format in which response message is sent from server -> client (it contains some additional information)

HTTP Response

Response HTTP header

- general header (contains information about the date and time client requested the request message)
- request/response header (contains URL/data we want to send and HTTP method we choose to send the data with, authentication(secret key), content type of the data it is returning, content size, **what is the status of the API call** (success or failure?), redirect URL (if your redirecting to another webpage)
- entity header (contains information about the html code (code length, programming language used))

Response HTTP body

- Response HTTP body contains the content we requested for (basically everything shown on the DOM exists in the HTTP body)

But, how does the client communicate with the server?

HTTP methods

HTTP methods are request methods used by clients to connect with the server

There are some basic operations, that the user might want to perform on the website: AKA CRUD

C - user might want to, Create data on the server
R - user might want to, Read data from the server
U - user might want to, Update data on the server
D - user might want to, Delete data from the server

This is a list of some basic (most important) HTTP methods, However, we'll be discussing all 10 HTTP Method's in a bit.

1. Create data - Post Method
2. Read data- Get Method
3. Update data- Put Method
4. Delete data- Delete Method

HTTP Methods are very important for API testing

HTTP request methods are used for client to connect with the server.

When we make a network request, we specify what we want the server to do with the data using HTTP methods. We can inspect these network requests we make from the developer tools.

HTTP REQUEST METHODS:

When we are browsing the web, our browser is constantly sending requests to a server and the server, in turn, returns a response in a specific format or performs an action according to what we ask it to do.

HTTP request methods is an indication about what desired action is expected from the server.

These methods are placed after the API URL, each of these methods indicate a specific action from the server

1. GET

— This is the most common request of all. Through this request we ask for the representation of a resource: which can be an html, xml, json file, etc.

its used to get data from server [default request] i.e Get Method is asking for some data from the backend.

- used to fetch data from the server/particular URL [Read]

2. PUT

— its used to PUT data to the server. **If the resource already exists, it can be updated using PUT method.** If it doesn't exist, it can be created using PUT method.

- used to **update** complete data/resource at the server/particular URL [Update]

3. POST

— The **POST method** is used when we want to create a resource. When using POST, the data goes in the body of the request and not in the URI. its used to add/ POST data to the server. Post Method is you sending data to backend and telling it to do something with that data.

- used to **create new resource** at this URL

Difference between PUT and POST? Both are used for inserting data to the server right? well, almost... but they're very different.

PUT	POST
Replacing existing resource or Creating if resource is not exist <i>http://www.example.com/customer/{id}</i> <i>http://www.example.com/customer/123/orders/456</i> Identifier is chosen by the client	Creating new resources (and subordinate resources) <i>http://www.example.com/customer/</i> <i>http://www.example.com/customer/123/orders</i> Identifier is returned by server
Idempotent i.e. if you PUT a resource twice, it has no effect. Ex: Do it as many times as you want, the result will be same. $x=1$;	POST is neither safe nor idempotent. It is therefore recommended for non-idempotent resource requests. Ex: $x++$;
Works as specific	Works as abstractive
If you create or update a resource using PUT and then make that same call again, the resource is still there and still has the same state as it did with the first call.	Making two identical POST requests will most-likely result in two resources containing the same information.

7) What is the project life cycle?

The project life cycle includes the steps required for project managers to successfully manage a project from start to finish. There are 5 phases to the project life cycle (also called the 5 process groups)—initiating, planning, executing, monitoring/controlling, and closing. Each of these project phases represents a group of interrelated processes that must take place.

The 5 process groups of project management

Initiating phase

The initiating phase of the project life cycle consists of just two separate processes: the project charter and stakeholder register. The point of this phase is to determine the vision for your project, document what you hope to accomplish, and secure approvals from a sanctioning stakeholder. The key components of the project charter include:

- Business case
- Project scope
- Deliverables
- Objectives
- Resources needed
- Milestone plan and timeline
- Cost estimate
- Risks and issues
- Dependencies

When you take the time to establish a clear and cohesive vision, think through who should ideally be involved in bringing the project to life, and secure the resources you'll need up front, you give your project a strong start that sets the stage for everything that comes next.

Planning phase

The planning phase process group is where you build the project infrastructure that will enable you to achieve your goal within your predetermined time and budget constraints, starting with a project management plan, project scope, work breakdown structure and more—and wrapping up with qualitative and quantitative risk analyses and risk responses. This is your detailed roadmap—

your blueprint for success. When you reach the end of this phase of the life cycle, everyone on your team will not only understand the vision of the project, they'll also understand precisely what they need to do to reach the finish line on time and within budget.

Whitepaper: The Complete Guide to Planning Creative Projects

Ebook: 3 Strategies to Plan Successful Marketing Projects

Executing phase

The executing phase is where the rubber hits the road—where most of the budget is allocated and most of the project deliverables are produced. You take your project plan and put it into action, whether that takes weeks, months, or even years. Villanova University defines the goal of this phase as, “managing teams effectively while orchestrating timeline expectations and reaching benchmark goals.” The executing phase often includes team development, stakeholder engagement, and quality assurance activities, either on a formal or informal basis.

Monitoring and controlling phase

The monitoring and controlling phase involves keeping an eye on the actual progress of the project against your plan and taking corrective action where necessary. No amount of perfect planning will exempt you from the need to be constantly vigilant with tracking and reporting. You know what they say about the best-laid plans, after all.

Blog: Helping Your Team Manage & Track Time

Blog: 4 Simple Ways to Keep Creative Projects On-Track

Closing phase

The closing phase is the final phase of the project life cycle includes just one solitary process, and it's more than simply checking off the project as done. It's essential to formally close the project and secure a sign-off or approval from the customer, stakeholders, and/or project sponsor. This process might include:

- Delivering the project
- Hosting a post-mortem meeting

- Archiving project records
- Celebrating or acknowledging the achievement
- Officially disbanding or releasing the team

The importance of this final step of the project life cycle can't be overstated, especially as more organizations are adopting the Hollywood model of work, where temporary teams come together around a specific project, and then disband and regroup for another project, much the way film crews operate. Every film production ends with a "wrap party," and so should every major work project.

8) How do define Success

I can Define my Success is based on my Hard work and Effect on a Particular work I can Done.

9)what are your Strengths and Weakness

My Strength is Learning New Technologies

I can Trust any One either the Person is Good or Bad

10)Have you faced any challenging Situations and how did you overcome it

I can face many Financial and Critical Situations to become this Stage

I can Balancing my Educational Background and Work for Part time to Lead it.