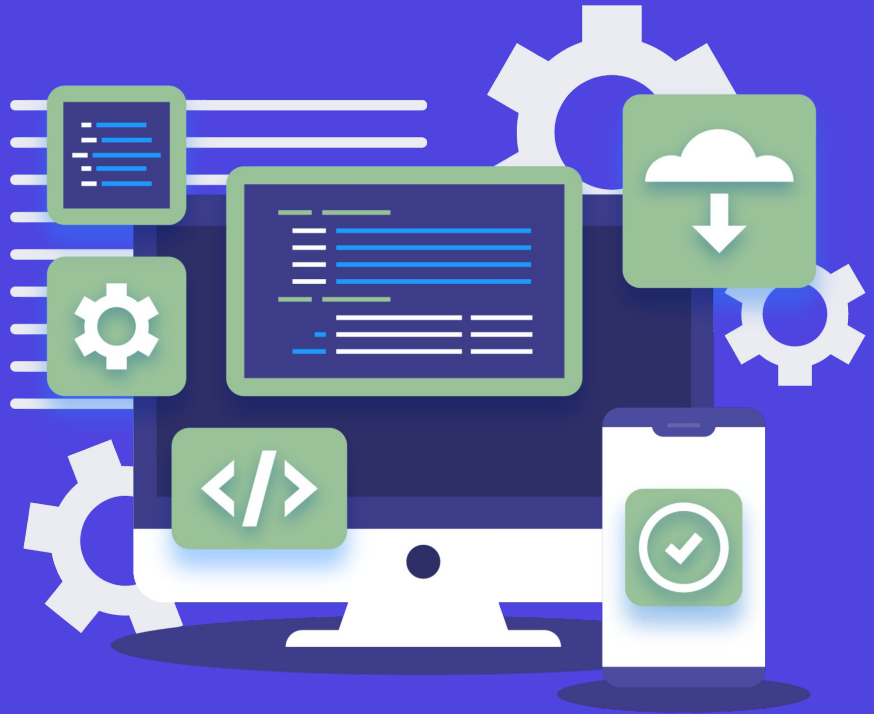


# Client-Server Architecture

**Relevel**  
by Unacademy



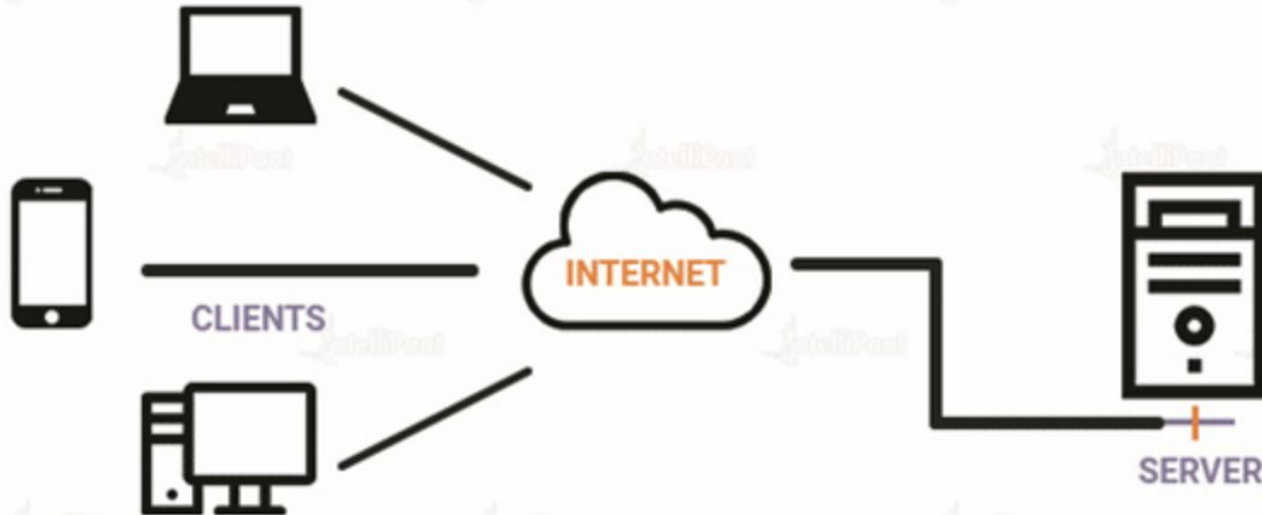
# Ever wondered How you are able to view live score on internet ?

The screenshot displays the Cricbuzz website interface. At the top, there's a navigation bar with links for Live Scores, Schedule, Archives, News, Series, Teams, Videos, Rankings, and More. Below this, a 'FEATURED MATCHES' section shows scores for AUS vs ENG (473-9 d & 45-1), CD vs OTG (180-4 (20)), CLS vs KDW (182-6 (20)), and GGD vs JKS (129-8 (20)). A 'LATEST NEWS' sidebar on the left contains headlines about quarantine issues, KL Rahul's appointment, and Gautam Gambhir's role. The main content area features a large article titled 'Australia dominant after bagging massive lead' with a photo of the Australian team celebrating. To the right, there's an advertisement for Airtel Broadband and a 'FEATURED VIDEOS' section with a video titled 'Cricbuzz Chatter: Australia take a lead of 282 on Day 3! Michael Hussey reviews'.

**Backbone of all the interactions on internet is a simple architecture!**

CLIENT-SERVER Architecture

Imagine Client as your interface and Server where the actual information resides



# Client



## Client

1. Requests Information
2. Depends upon server
3. Holds no additional resources

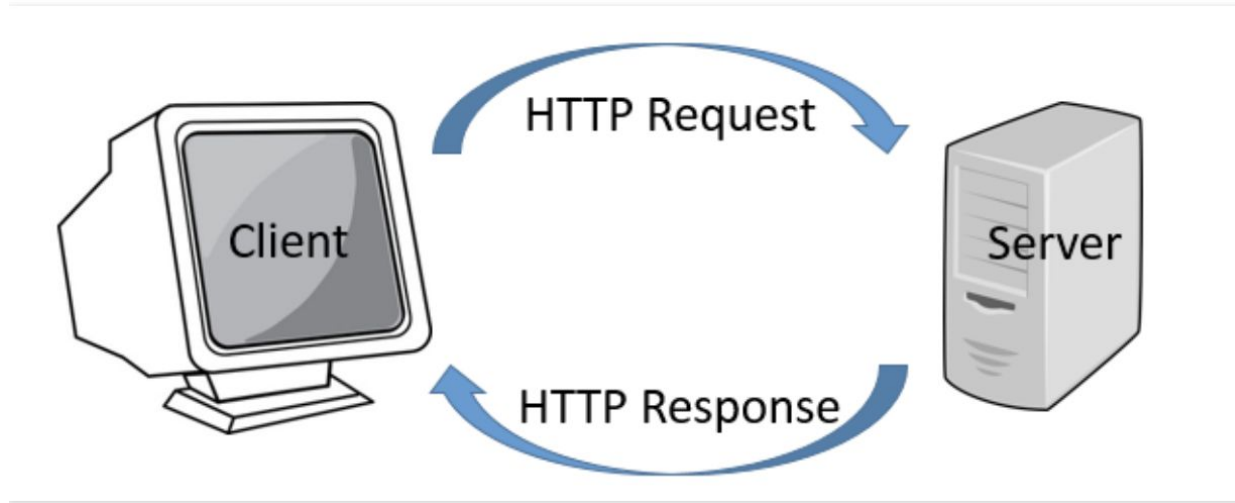
# Server

1. Responsible for Hosting and managing resources which compute results for client
2. Abstracts out the implementation
3. Responds to whatever client asks for



# How does the communication between Client and Server actually hap

HTTP to the rescue !!!!



# Types Of HTTP Requests

1. GET
2. POST
3. PUT
4. DELETE
5. HEAD
6. TRACE
7. OPTIONS
8. PATCH





# Sample HTTP Request

GET /test HTTP/1.1

User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)

Host: www.tutorialspoint.com

Accept-Language: en-us

Accept-Encoding: gzip, deflate

Connection: Keep-Alive



# HTTP Response

The screenshot shows the Chrome DevTools Network tab. The top toolbar includes filters for 'Fetch/XHR', 'JS', 'CSS', 'Img', 'Media', 'Font', 'Doc', 'WS', 'Wasm', 'Manifest', and 'Other'. The left sidebar shows a list of requests, with the selected request being '7relatedposts=1'. The main panel displays the 'Headers' tab for the selected request. The 'General' section shows the request URL, method (POST), status code (200), remote address, and referrer policy. The 'Response Headers' section lists various headers including access-control, cache-control, content-encoding, content-type, date, expires, host-header, referrer-policy, server, strict-transport-security, vary, x-ac, x-content-type-options, x-frame-options, x-hacker, and x-robots-tag. The 'Request Headers' section shows the authority, method, path, scheme, and accept headers.

**General**

- Request URL: `https://gavilan.blog/wp-admin/admin-ajax.php`
- Request Method: `POST`
- Status Code: `200`
- Remote Address: `192.0.78.25:443`
- Referrer Policy: `strict-origin-when-cross-origin`

**Response Headers**

- `access-control-allow-credentials: true`
- `access-control-allow-origin: https://gavilan.blog`
- `cache-control: no-cache, must-revalidate, max-age=0`
- `content-encoding: br`
- `content-type: text/html; charset=UTF-8`
- `date: Thu, 16 Dec 2021 07:56:38 GMT`
- `expires: Wed, 11 Jan 1984 05:00:00 GMT`
- `host-header: WordPress.com`
- `referrer-policy: strict-origin-when-cross-origin`
- `server: nginx`
- `strict-transport-security: max-age=31536000`
- `vary: Accept-Encoding`
- `vary: Origin`
- `x-ac: 3.bom_dca`
- `x-content-type-options: nosniff`
- `x-frame-options: SAMEORIGIN`
- `x-hacker: If you're reading this, you should visit automattic.com/jobs and apply to join the fun, mention this header.`
- `x-robots-tag: noindex`

**Request Headers**

- `:authority: gavilan.blog`
- `:method: POST`
- `:path: /wp-admin/admin-ajax.php`
- `:scheme: https`
- `accept: */*`

# HTTP Response codes

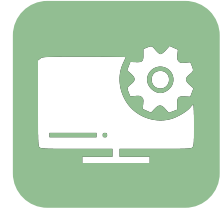
1. 100's
2. 200's
3. 300's
4. 400's
5. 500's



# GET VS POST METHOD

Discuss

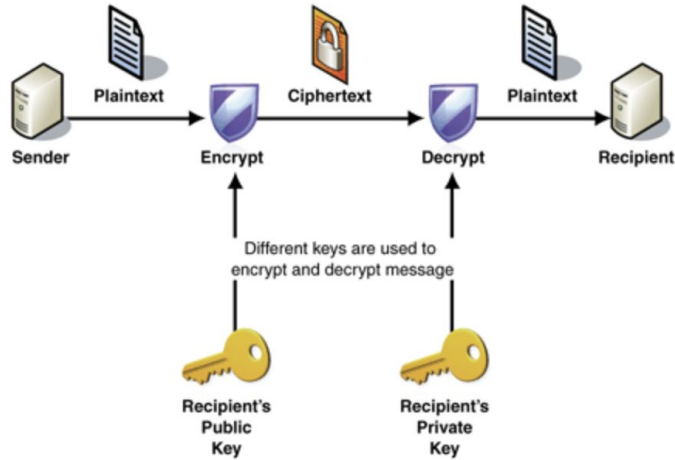
1. Reloading
2. Bookmarking
3. Caching
4. Security
5. Size



**Is HTTP Secure ?**

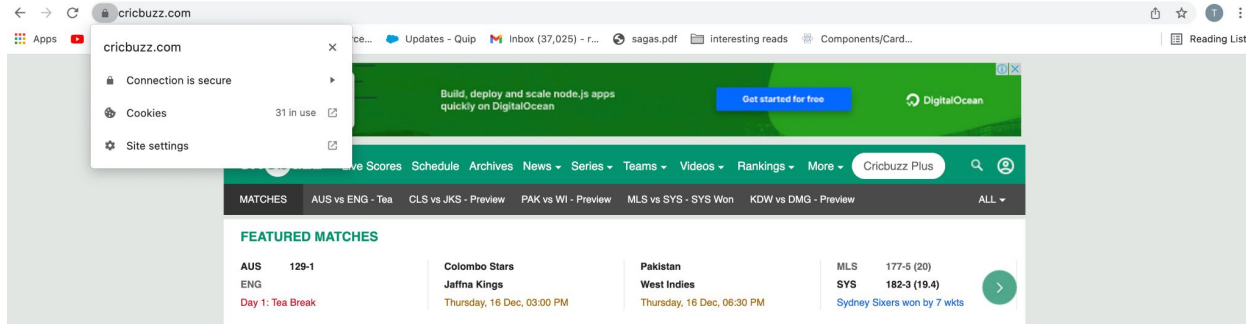
**Intro to HTTPS (Hypertext Transfer Protocol Secure)**

# Encryption at work!!!



Example of asymmetric encryption system

# Is your current web page Secure ?



# REST (Representational state transfer)

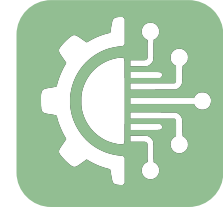
1. Client-Server architecture
2. Cacheability
3. Layered-System
4. Stateless
5. Uniform Interface



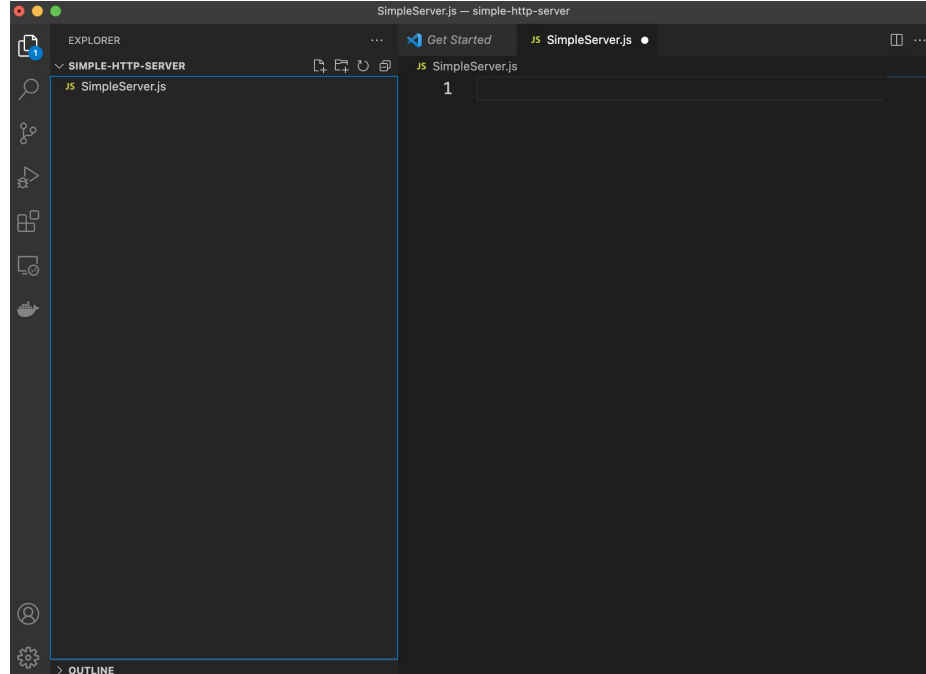
# Lets Build our own HTTP Server Now!!!

## Pre-Req

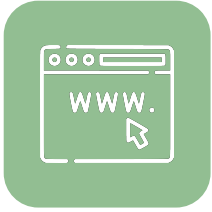
- Code editor
- Git (Optional)
- Node js installed



## Set up Root directory



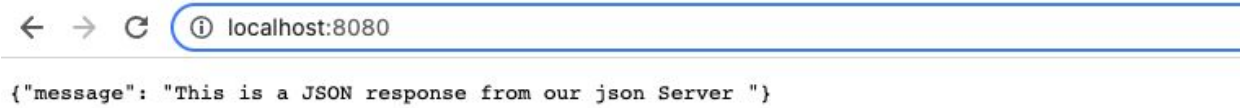
# Create HTTP Server



← → ↻ ⓘ localhost:8080

Our first http server

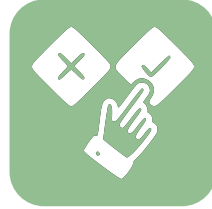
## Create HTTP Server with json response0



## Create HTTP Server with CSV response

details

name	rollNumber	school
Tushar raina	34	Thapar



**MCQ'S**

# Homework!

1. Let us change the port to 8448
2. Let us return error 400 error code



**The End!!!**