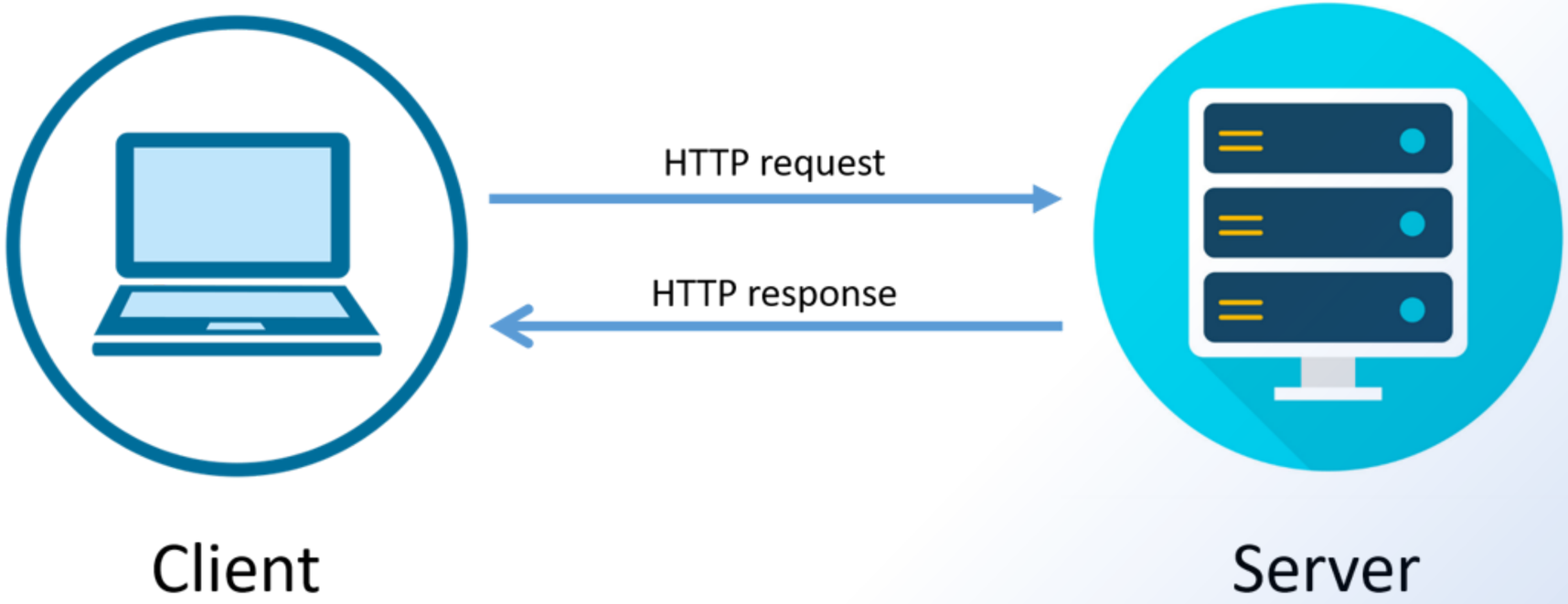
Request-Response Model



POSTMAN Http Client

Postman is an HTTP Client application, used to test request-response communication.

Postman is widely used for API testing and generating documentation

- Quickly and easily send REST, SOAP, and GraphQL requests directly within Postman.
- Generate and publish beautiful, machine-readable API documentation.
- Checking performance and response times at scheduled intervals.
- Communicate the expected behavior of an API by simulating endpoints and their responses

Http Request Segments:

HTTP Request is the first step to initiate web request/response communication. Every request is a combination of request header, body and request URL.

Request Area	Standard Data Type
Body	Simple String, JSON, Download, Redirect, XML
Header	Key Pair Value
URL Parameter	String

Request Compare GET vs. POST

Key Points	GET	POST
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be	Never
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions

Request Compare GET vs. POST

Key Points	GET	POST
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL. Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

When use GET() When POST()

- **GET** is used to request something from server with less amount of data to pass.
- When nothing should change on the server because of your action.
- When request only retrieves data from a web server by specifying parameters
- Get method only carries request url & header not request body.

- **POST** should be used when the server state changes due to that action.
- When request needs its body, to pass large amount of data.
- When want to upload documents , images , video from client to server

Http Request Throttling

Throttle Request refers to a process in which a user is allowed to hit the application maximum time in per second or per minute. Throttling is also known as request rate limiting.

- Essential component of Internet security, as DoS attacks can tank a server with unlimited requests.
- Rate limiting also helps make your API scalable by avoid unexpected spikes in traffic, causing severe lag time.

Http Response Segments

Http response is the final step of request-response communication. Every response is a combination of response header, body and cookies.

Response Area	Standard Data Type
Body	Simple String, JSON, Download, Redirect, XML
Header	Key Pair Value
Cookies	Key Pair Value

HTTP Response status messages

Code	Meaning	Description
200	OK	The request is OK (this is the standard response for successful HTTP requests)
201	Created	The request has been fulfilled, and a new resource is created
202	Accepted	The request has been accepted for processing, but the processing has not been completed
203	Non-Authoritative Information	The request has been successfully processed, but is returning information that may be from another source
204	No Content	The request has been successfully processed, but is not returning any content
205	Reset Content	The request has been successfully processed, but is not returning any content, and requires that the requester reset the document view

HTTP Response status messages

Code	Meaning	Description
206	Partial Content	The server is delivering only part of the resource due to a range header sent by the client
400	Bad Request	The request cannot be fulfilled due to bad syntax
401	Unauthorized	The request was a legal request, but the server is refusing to respond to it.
403	Forbidden	The request was a legal request, but the server is refusing to respond to it
404	Not Found	The requested page could not be found but may be available again in the future
405	Method Not Allowed	A request was made of a page using a request method not supported by that page

HTTP Response status messages

Code	Meaning	Description
408	Request Timeout	Request Timeout
500	Internal Server Error	A generic error message, given when no more specific message is suitable
502	Bad Gateway	The server was acting as a gateway or proxy and received an invalid response from the upstream server
503	Service Unavailable	The server is currently unavailable (overloaded or down)