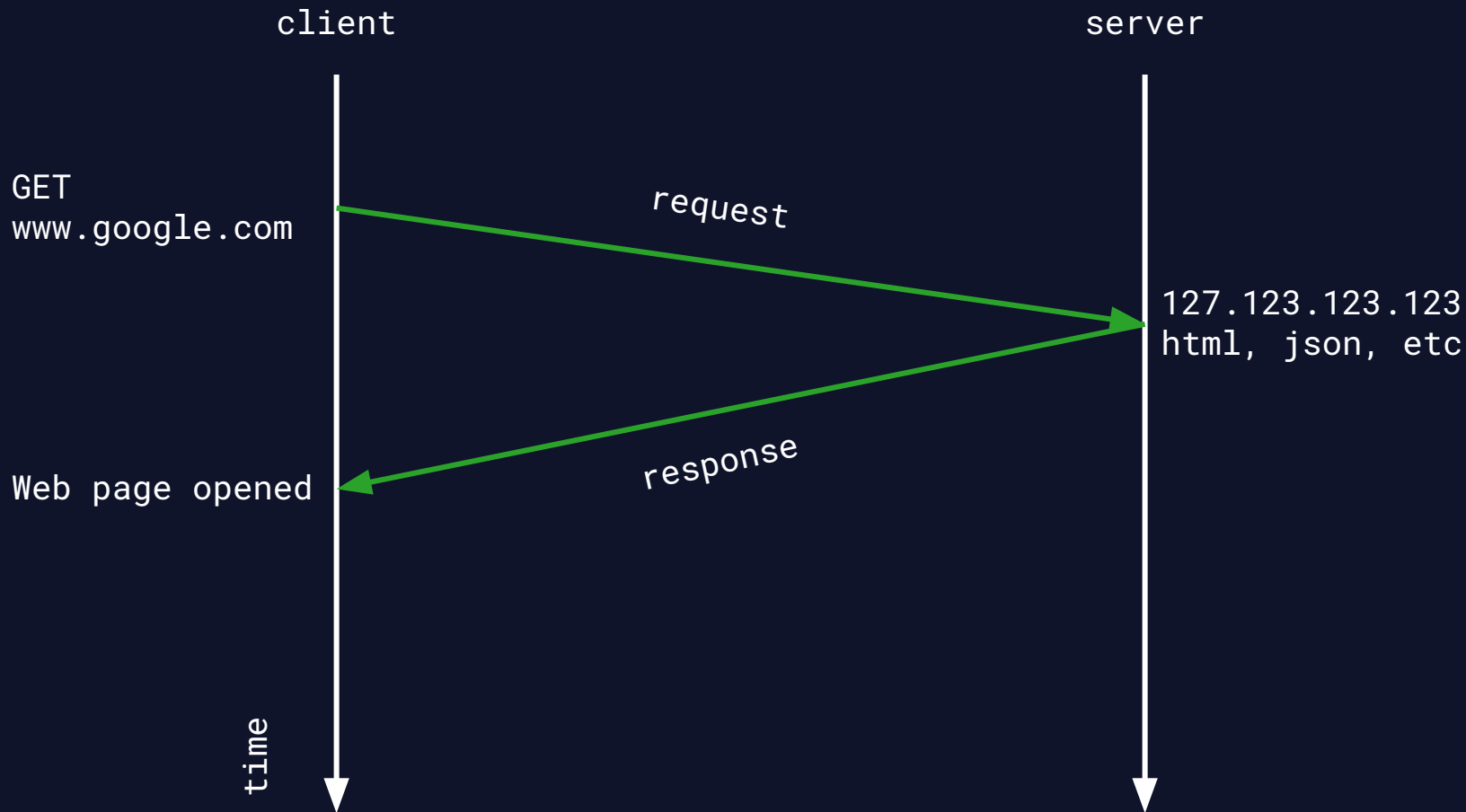# Demystifying HTTP

The Backbone of the Web

Request-Response Model

Clients (browser & insomnia) send requests to servers.

Servers process these requests and send back responses.

Everything you do online involves this back-and-forth.

Anatomy of http

Headers

HTTP headers are metadata associated with an HTTP request or response.

They provide additional information about the data being transmitted.

Headers are represented as key-value pairs.

Common headers

Request

User-Agent: Indicates the client making the request, typically a web browser or application.

Accept: Informs the server about the types of media (e.g., JSON, HTML) that the client can handle.

Authorization: Contains credentials or tokens for authentication.

Cookie: Contains client-specific data, such as session IDs.

Response

Content-Type: Specifies the media type of the response body (e.g., text/html, application/json).

Content-Length: Indicates the length of the response body in bytes.

Location: Used in redirections to specify the new URL.

Cache-Control: Controls how caching should be performed by clients and intermediaries.

Body

The HTTP body is the payload of the HTTP request or response. It contains the actual data being sent between the client and server. Here's more about HTTP bodies:

Common body

Request

Form Data: In HTML forms, data is sent in the request body in a URL-encoded or multipart format.

JSON Data: For APIs, data is often sent in JSON format in the request body.

Binary Data: File uploads and binary data are included in the request body in a binary format (e.g., image uploads).

Response

HTML for rendering web pages

JSON for API responses

files for download

HTTP Methods

GET: Retrieve data from a server.

POST: Send data to be processed by a server.

PUT: Update existing data on a server.

DELETE: Remove data from a server.

HTTP Status Codes

200 OK: Success!

404 Not Found: The requested resource isn't there.

500 Internal Server Error: Something went wrong on the server.

https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

Stateless protocol

Each request-response is independent.

Servers don't remember previous interactions with clients.