

# Build & Inspect API Requests

## Tool Used

- **Postman** (API testing tool)  
(Alternative: *Hopscotch – browser-based*)
- 

### 1 GET Request – Fetch Data from Public API

**URL:**

<https://jsonplaceholder.typicode.com/posts/1>

**Method:**

GET

#### What I did

- Opened Postman
- Selected **GET** method
- Entered the URL
- Clicked **Send**

#### Result

- The server returned a JSON response
- It contained post details like `userId`, `id`, `title`, and `body`
- Status code was **200 OK**, meaning the request was successful

The screenshot shows the Postman application interface. At the top, it displays an 'Overview' tab, the URL 'GET https://jsonplaceholder.', and a note 'No environment'. Below the URL, there's a search bar with 'https://jsonplaceholder.typicode.com/posts/1' and a 'Send' button. The main workspace shows a 'Params' tab selected, with a table for 'Query Params'. Under the 'Body' tab, the response is shown as JSON, indicating a 200 OK status with a response time of 167 ms and a size of 1.31 KB. The JSON response is as follows:

```
1 {  
2   "userId": 1,  
3   "id": 1,  
4   "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",  
5   "body": "quia et suscipit\\nsuscipit recusandae consequuntur expedita et cum\\nreprehenderit  
molestiae ut ut quas totam\\nnostrum rerum est autem sunt rem eveniet architecto"  
6 }
```

## 2 POST Request – Send Data to API

**URL:**

<https://jsonplaceholder.typicode.com/posts>

**Method:**

POST

**Headers:**

Content-Type: application/json

**Body (raw → JSON):**

```
{  
  "title": "My First API Post",
```

```
"body": "Learning APIs is fun!",  
"userId": 101  
}
```

## What I did

- Selected **POST** method in Postman
- Added the JSON body
- Clicked **Send**

## Result

- The server accepted the data
- It returned the same data along with a new **id**
- Status code was **201 Created**

Note: This API is a fake test API, so data is not actually saved on the server.

The screenshot shows the Postman interface with the following details:

- Request URL:** https://jsonplaceholder.typicode.com/posts
- Method:** POST
- Headers:** content-type=application/json
- Query Params:** content-type=application/json
- Body:** JSON

```
1 {  
2   "title": "My First API Post",  
3   "body": "Learning APIs is fun!",  
4   "userId": 101,  
5   "id": 101  
6 }
```
- Response:** 201 Created, 275 ms, 1.32 KB

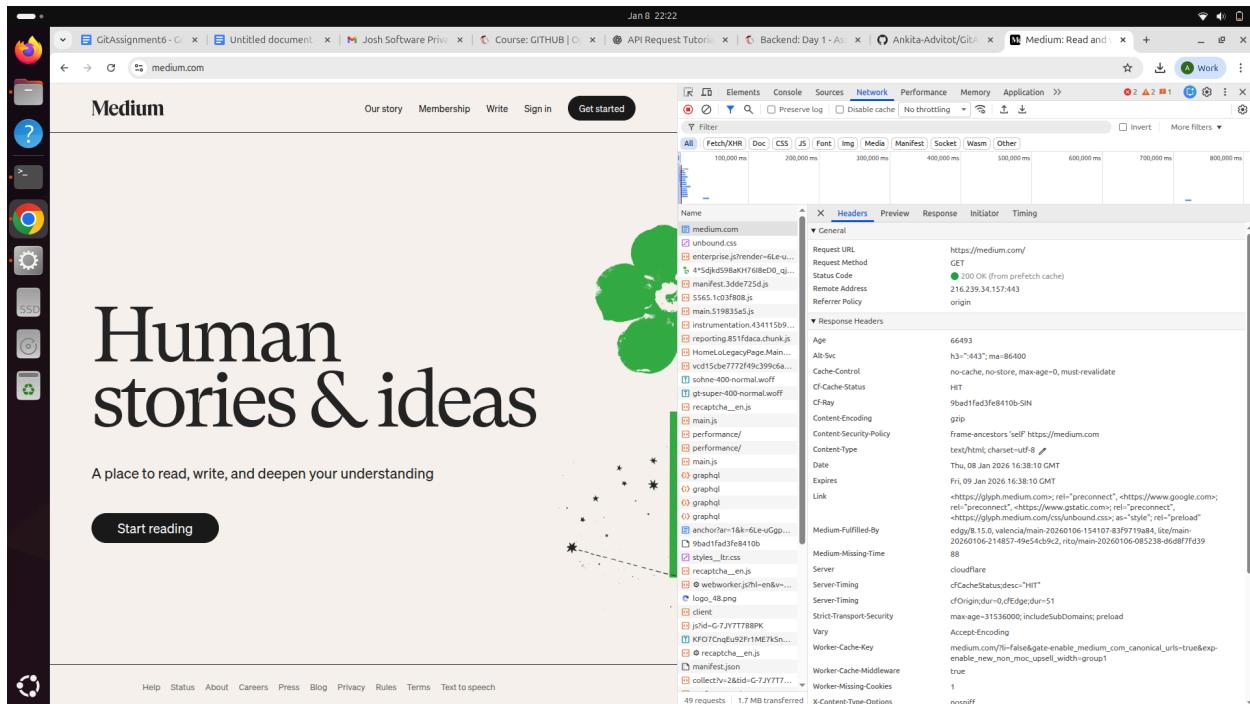
## ③ Inspecting Network Requests in Browser DevTools

## Steps

1. Open **Chrome Browser**
2. Press **F12** or **Right-click → Inspect**
3. Go to the **Network** tab
4. Open any website (e.g., [google.com](https://google.com))
5. Click on any request from the list

## Observed Details

- **URL:** The address where the request was sent
- **Method:** GET
- **Request Headers:** Information sent by browser (User-Agent, Accept, etc.)
- **Response Headers:** Information sent by server (Status, Content-Type, etc.)
- **Body:** No body for GET request



The screenshot shows a Mac desktop with a vertical dock on the left containing icons for Finder, Home, Mail, Safari, and others. A Firefox browser window is open, displaying the Medium homepage. The page features a large green illustration of a brain and the text "Human stories & ideas". Below the illustration is the tagline "A place to read, write, and deepen your understanding" and a "Start reading" button.

The browser's developer tools Network tab is active, showing a list of requests and responses. The "Headers" section is expanded, displaying detailed information about the request to `https://medium.com/`. Headers include:

- Request URL: `https://medium.com/`
- Request Method: GET
- Status Code: 200 OK (from prefetch cache)
- Remote Address: 216.239.34.157.443
- Referrer Policy: origin

Other headers listed include:

- Age: 66493
- All-Svc: h3=>443; ma=86400
- Cache-Control: no-cache, no-store, max-age=0, must-revalidate
- CF-Cache-Status: HIT
- CF-Ray: 9bad1fad3fe8410b-SIN
- Content-Encoding: gzip
- Content-Security-Policy: frame-ancestors 'self' https://medium.com
- Content-Type: text/html; charset=utf-8
- Date: Thu, 08 Jan 2026 16:38:10 GMT
- Expires: Fri, 09 Jan 2026 16:38:10 GMT
- Link: <<https://gliph.medium.com/>>;rel="preconnect",<<https://www.google.com/>>;rel="preconnect",<<https://www.gstatic.com/>>;rel="preconnect",<<https://gliph.medium.com/css/unbound.css>>;as="style";rel="preload"
- Medium-Fulfilled-By: edge/8.15.0, valid=main/main-3020106-151407-83971984, lite/main-2020106-214857-4ec42d42, r7z/main-20200106-983239-d5d87f1d98
- Medium-Missing-Time: 88
- Server: cloudflare
- Server-Timing: cfCacheStatus=0;cfEdgeDlr=51
- Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
- Vary: Accept-Encoding
- Worker-Cache-Key: medium.com/fi/false&gate\_enable\_medium\_com\_canonical\_urls=true&enable\_new\_non\_mic\_upsell\_width=group1
- Worker-Cache-Middleware: true
- Worker-Missing-Cookies: 1
- X-Content-Type-Options: nosniff

Below the Headers section, the "Timing" section is also visible, showing various network metrics.

medium.com

unbound.css

enterprise.js?render=6Le-u...

4\*Sdjkds98aKH76l8eD0\_qj...

manifest.3dde725d.js

5565.1c03f808.js

main.519835a5.js

instrumentation.434115b9...

reporting.851fdaca.chunk.js

HomeLoLegacyPage.Main...

vcd15cbe7772f49c399c6a...

sohne-400-normal.woff

gt-super-400-normal.woff

recaptcha\_en.js

main.js

performance/

performance/

main.js

graphql

graphql

graphql

graphql

anchor?ar=1&k=6Le-uGgp...

9bad1fad3fe8410b

styles\_ltr.css

recaptcha\_en.js

webworker.js?hl=en&v=...

logo\_48.png

client

js?id=G-7JY7T788PK

KFO7CnqEu92Fr1ME7kSn...

recaptcha\_en.js

manifest.json

collect?v=2&tid=G-7JY7T7...

49 requests | 1.7 MB transferred

**Headers**

Request URL: https://medium.com/

Request Method: GET

Status Code: 200 OK (from prefetch cache)

Remote Address: 216.239.34.157:443

Referrer Policy: origin

**Response Headers**

Age	66493
Alt-Svc	h3=":443"; ma=86400
Cache-Control	no-cache, no-store, max-age=0, must-revalidate
Cf-Cache-Status	HIT
Cf-Ray	9bad1fad3fe8410b-SIN
Content-Encoding	gzip
Content-Security-Policy	frame-ancestors 'self' https://medium.com
Content-Type	text/html; charset=utf-8
Date	Thu, 08 Jan 2026 16:38:10 GMT
Expires	Fri, 09 Jan 2026 16:38:10 GMT
Link	<https://glyph.medium.com>; rel="preconnect", <https://www.google.com>; rel="preconnect", <https://www.gstatic.com>; rel="preconnect", <https://glyph.medium.com/css/unbound.css>; as="style"; rel="preload" edgy/8.15.0, valencia/main-20260106-154107-83f9719a84, lite/main-20260106-214857-49e54cb9c2, rito/main-20260106-085238-d6d8f7fd39
Medium-Fulfilled-By	88
Medium-Missing-Time	88
Server	cloudflare
Server-Timing	cfCacheStatus;desc="HIT"
Server-Timing	cfOrigin;dur=0,cfEdge;dur=51
Strict-Transport-Security	max-age=31536000; includeSubDomains; preload
Vary	Accept-Encoding
Worker-Cache-Key	medium.com/?li=false&gate-enable_medium_com_canonical_urls=true&exp-enable_new_non_moc_upsell_width=group1
Worker-Cache-Middleware	true
Worker-Missing-Cookies	1
X-Content-Type-Options	nosniff

▼ Request Headers	
<b>⚠ Provisional headers are shown. <a href="#">Learn more</a></b>	
Referer	<a href="https://www.google.com/">https://www.google.com/</a>
Sec-Ch-Ua	"Google Chrome";v="143", "Chromium";v="143", "Not A(Brand");v="24"
Sec-Ch-Ua-Mobile	?0
Sec-Ch-Ua-Platform	"Linux"
Upgrade-Insecure-Requests	1
User-Agent	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36

## 4 Request → Response Explanation (5–8 lines)

When a request is made, the browser or Postman sends a request to the server with a URL and method (GET or POST).

The server receives this request and processes it.

If the request is valid, the server sends back a response.

The response contains a status code, headers, and sometimes a body.

GET requests fetch data from the server.

POST requests send data to the server.

This process is called the **request-response cycle**.

# Request Headers (Sent by the browser / Postman → Server)

## Meaning:

Request headers are **extra information** sent along with the **request** to tell the server **who is asking and how**.

Think of it like an **envelope on a letter**.

## Simple example:

When you open a website, your browser tells the server:

- “I am using Chrome”
- “I can understand JSON”
- “I am a desktop browser”

## Common Request Headers (easy meaning)

Header	What it means
User-Agent	Which browser or app is making the request
Accept	What type of response the client wants (JSON, HTML)
Content-Type	Format of data being sent (mostly JSON in APIs)
Authorization	Login/token details (if required)

### 📌 Example line you may see in DevTools:

User-Agent: Mozilla/5.0

→ Means: *Request is coming from a browser*

---

# Response Headers (Sent by Server → Browser / Postman)

### **Meaning:**

Response headers are **extra information** the server sends back to explain **what it is returning**.

Think of it like a **reply note attached to the letter**.

### **Simple example:**

The server replies:

- “Request was successful”
- “Here is JSON data”
- “The data size is this much”

### **Common Response Headers (easy meaning)**

Header	What it means
Status	Result of request (200 = success)
Content-Type	Type of response data (JSON / HTML)
Content-Length	Size of response
Date	When the response was sent

#### **Example line you may see:**

Content-Type: application/json

→ Means: *Server is sending JSON data*

---

## **Very Short One-Line Summary (for exams / docs)**

- **Request Headers:** Information sent by the client to the server
  - **Response Headers:** Information sent by the server back to the client
-