headers

Important HTTP Headers – A Backend Developer's Guide

1. 📦 General Headers

Header	Purpose	Example	Notes
`Host`	Specifies the domain name of the server	`Host: example.com`	Required for virtual hosting
`User-Agent`	Identifies the client software	`User-Agent: Mozilla/5.0 ...`	Used for logging, analytics, or conditional content
`Accept`	Tells the server what media types are acceptable	`Accept: application/json`	Server may respond with `406 Not Acceptable` if unsupported
`Content-Type`	Indicates the media type of the request body	`Content-Type: application/json`	Must match the actual request body format
`Content-Length`	The size of the body in bytes	`Content-Length: 348`	Auto-calculated by most clients
`Authorization`	Used for passing credentials	`Authorization: Bearer <token>`	Common in JWT-based APIs

---

2. 🔁 Caching Headers

Header	Purpose	Example	Notes
`Cache-Control`	Specifies caching policies	`Cache-Control: no-cache, no-store`	Critical for dynamic content

`Pragma`	Legacy HTTP/1.0 cache control	`Pragma: no-cache`	Works with `Cache-Control` for compatibility
`Expires`	Date/time when the response is considered stale	`Expires: 0`	Use `Cache-Control` instead when possible
`ETag`	Unique identifier for resource version	`ETag: "abc123"`	Works with `If-None-Match` for conditional GET
`Last-Modified`	Timestamp of the last modification	`Last-Modified: Tue, 03 Jun 2025 08:00:00 GMT`	Used with `If-Modified-Since`

3. 🔐 Security Headers

Header	Purpose	Example	Notes
`X-Content-Type-Options`	Prevents MIME type sniffing	`X-Content-Type-Options: nosniff`	Prevents execution of malicious files
`X-Frame-Options`	Controls if site can be embedded in an iframe	`X-Frame-Options: DENY`	Helps prevent clickjacking
`X-XSS-Protection`	Enables XSS filtering in some browsers	`X-XSS-Protection: 1; mode=block`	Modern browsers have deprecated it
`Strict-Transport-Security`	Enforces HTTPS	`Strict-Transport-Security: max-age=31536000; includeSubDomains`	Only works on HTTPS responses
`Content-Security-Policy`	Restricts resources the browser can load	`Content-Security-Policy: default-src 'self'`	Helps prevent XSS and data injection

4. 🌐 CORS Headers (Cross-Origin Resource Sharing)

Header	Purpose	Example	Notes
`Access-Control-Allow-Origin`	Specifies who can access the resource	`Access-Control-Allow-Origin: *`	Use specific domain in production
`Access-Control-Allow-Methods`	Allowed HTTP methods	`GET, POST, PUT, DELETE`	Sent in preflight responses

`Access-Control-Allow-Headers`	Allowed custom headers	`Authorization, Content-Type`	Must match frontend request headers
`Access-Control-Allow-Credentials`	Allows cookies to be sent	`Access-Control-Allow-Credentials: true`	Works only with specific origin, not `*`

5. 🎯 Response-Specific Headers

Header	Purpose	Example	Notes
`Location`	Used with `3xx` redirects	`Location: https://newsite.com`	For `301`, `302`, `303`, `307`, `308`
`Set-Cookie`	Sends cookies to the client	`Set-Cookie: sessionId=abc123; HttpOnly`	Secure, SameSite, and Max-Age important
`Content-Disposition`	Specifies content handling (download, inline)	`Content-Disposition: attachment; filename="file.pdf"`	Used for file downloads
`Retry-After`	Informs client when to retry a request	`Retry-After: 120`	Used with `503 Service Unavailable`

6. 📋 Custom Headers

- You can define your own headers using the `X-` convention, e.g.:
    - `X-Request-ID`: for tracking requests
    - `X-Powered-By`: to indicate the backend tech (can be removed for security)