HTTP Byte: Debrief Pointers

Milestone 1

Suppose Chrome versions below 80.0 don't support GIF images. We need our server to return a corresponding PNG image if any unsupported browser asks for the GIF image. How would the server know the Chrome version from which the request was made?

Browsers sends a **User-Agent** request header along with HTTP requests to denote the software it's using. If you check the request headers sent to the Flipkar server on visiting https://www.flipkart.com/, you'll see something like this

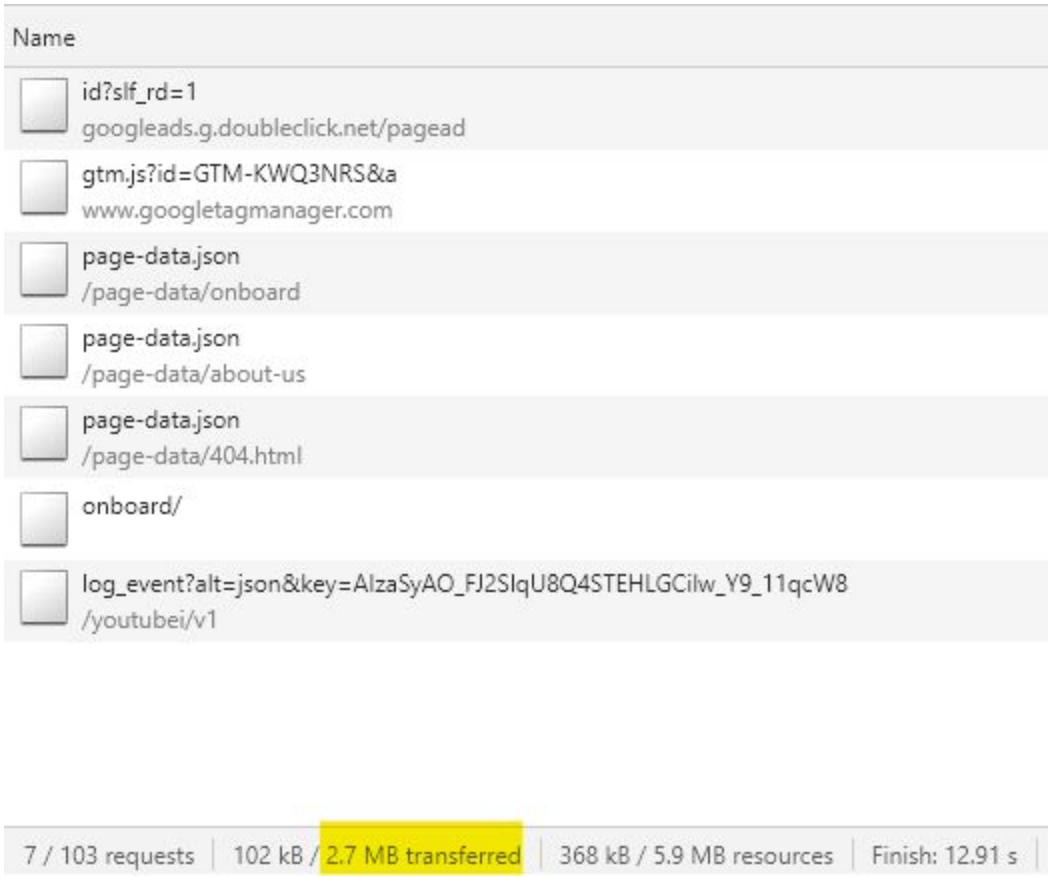
```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36
```

So the Chrome version here is 84.0 and the server can parse this to check if the browser supports GIF.

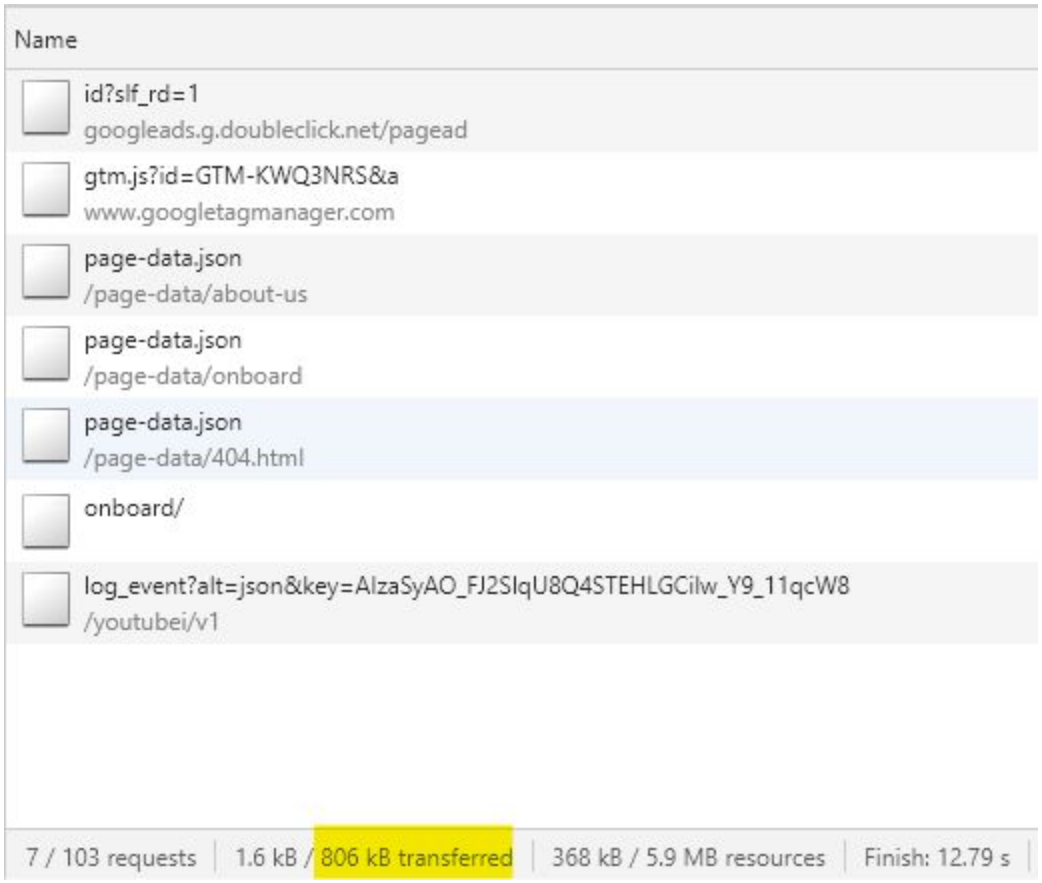
Curious why the header sent by Chrome is having keywords like **Mozilla**, **Safari** etc? They are for historical & compatibility reasons. See here & here

Open a browser tab in Incognito. Visit https://crio.do/ after opening the Networks tab in DevTools. Observe the size of data transferred. Open a new tab and do the same. Is there a difference in the size of data transferred now? Inspect the request & response headers in both situations to find out what's happening.

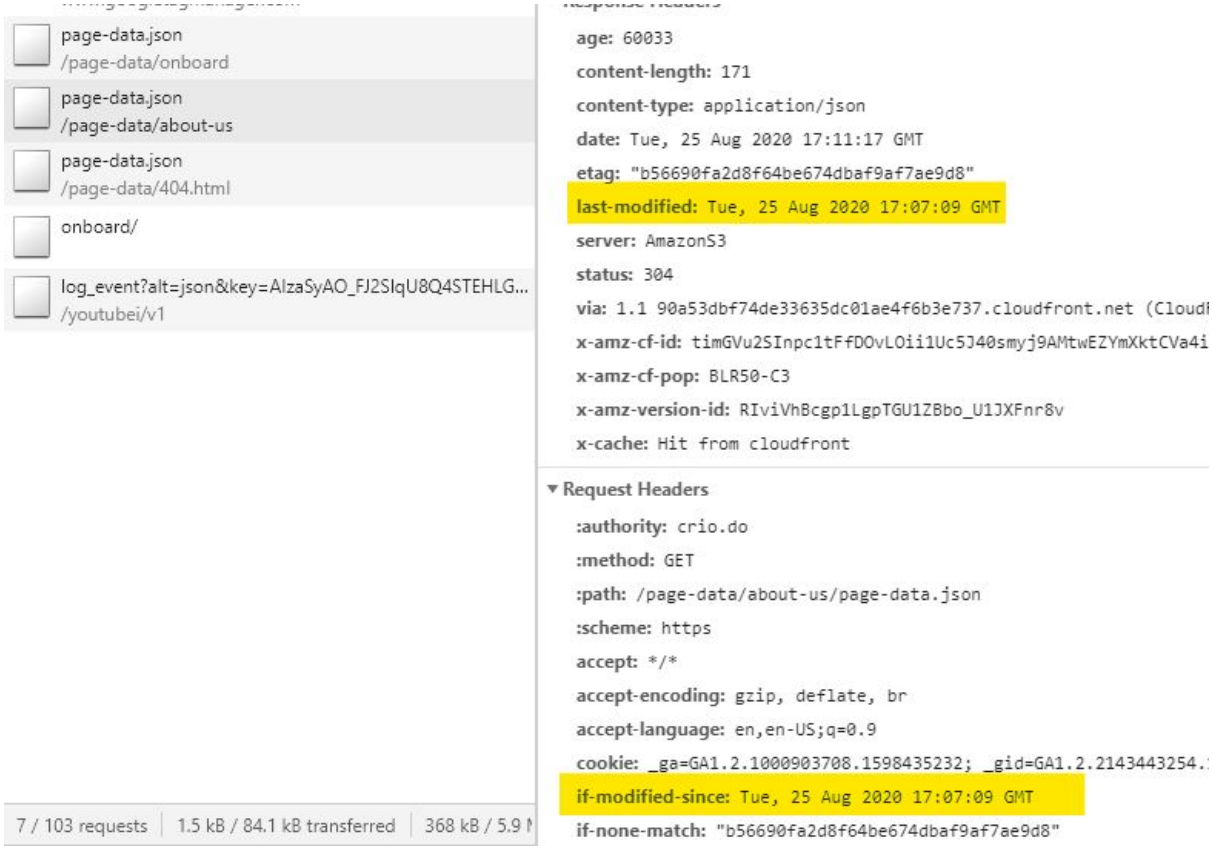
You'll be able to see in the bottom of the pane the size of data transferred to load the website. This as shown here is 2.7MB the first time.



When reloaded, the amount of data transferred got reduced to 806KB.



Though your data transfer could be different, there will be reduction in data transferred anyhow. This is due to using the **if-modified-since** in the HTTP request header and the **last-modified** response header the server sends. These values are used by the server to determine if to send any resource again. See [here]

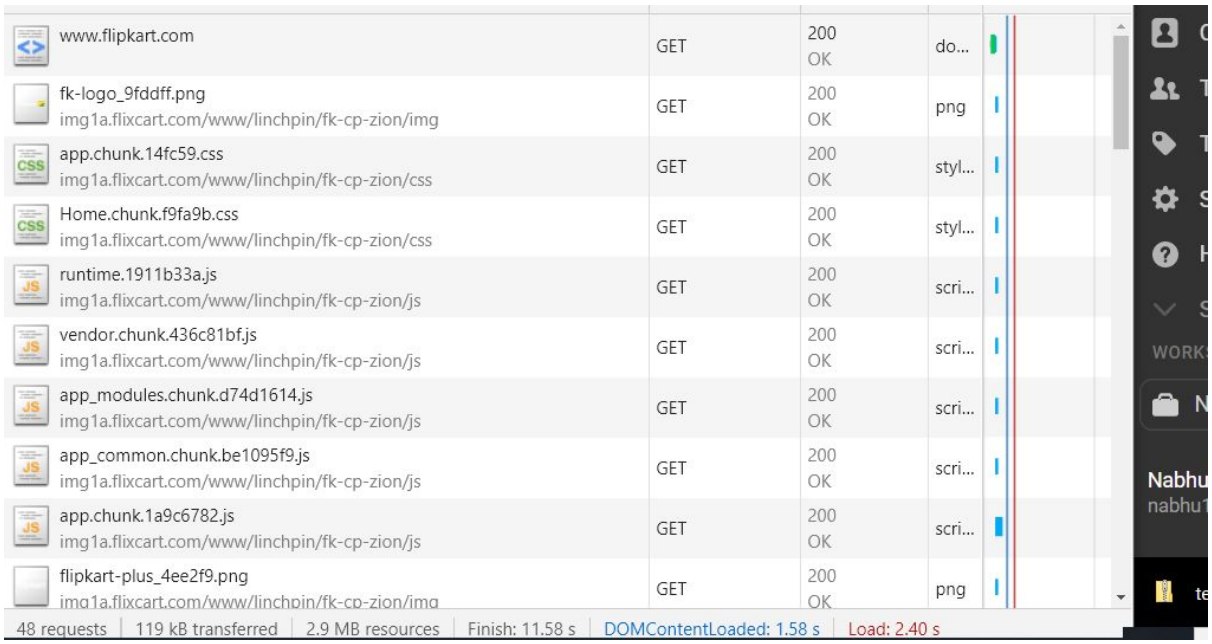


Depending on the website there'll be a considerable difference or no difference between the data transferred when visiting a website for the first time & subsequent visits. To make websites load faster, websites make use of caching by which resources like images are saved by the browser when it visits a website. HTTP provides headers for the server to specify how this has to happen.

Refer this

We looked at how requesting for a HTML file inturn creates a new HTTP request to fetch resources like scripts & images within it. Visit a couple of websites & inspect the resources loaded. Is there any order in which the resources are loaded? Does HTTP mandate this?

Though it might seem CSS & JavaScript files are preferred over images, HTTP doesn't favor any particular type of files to be loaded first. Further HTTP requests to fetch required resources for a page are made asynchronously meaning that each request is made independently without waiting for other requests to complete. As images are mostly of larger size than other resources, these requests get completed the last.Order in which resources are requested can also depend on their relative ordering in the HTML file.



Milestone 2

Is it possible to send form data using a GET request? Why or why not?

Yes, it's possible though not recommended. Usually, form data contains fields that are sensitive like passwords and using GET requests for submitting these means your password will be out in the open along with the request URL. See here on how to send data using GET.

Are there any limitations in using a GET request to send data to the server?

Data in a GET request is sent as part of the URL and this has a limit of 2048 characters.
What is the URL character limit for get requests?
Can HTTP POST be limitless?

Milestone 3

When you try to access a resource that requires logging in, like LinkedIn feed, https://www.linkedin.com/feed, you get redirected to the login screen. That should be a 301, right? Can you verify.

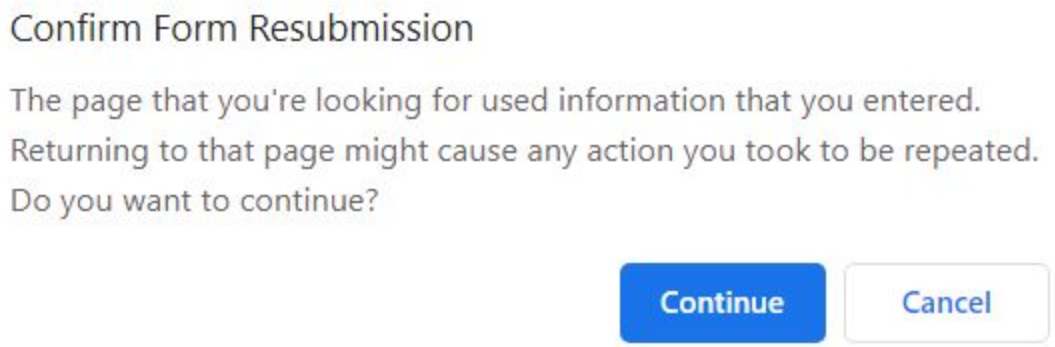
You'll be able to see, that was a 302 instead of 301. While 301 denotes a permanent redirect, 302 says the requested resource is temporarily unavailable. Here, 302 is used as the requested

resource was found, there just is another page to go through (Login page) before it can be accessed. More here

Name	Method	Status	Type
feed https://www.linkedin.com/feed	GET	302	
login?session_redirect=https%3A%2F%2... /uas	GET	303	
cold-join?session_redirect=https%3A%2... www.linkedin.com/signup	GET	200	document
%2Fregistration-frontend%2Fstylesheets... static-exp1.licdn.com/sc/p/com.linkedin....	GET	200	stylesheet
f4f3xa9bwiwmo5s7ttbnm1ro6 static-exp1.licdn.com/sc/h	GET	200	script

Refer this for more info on 301 vs 302

One day or another, you'd have come across the below pop-up when trying to reload a web page containing a form. Why does this happen? Is there any way to avoid this happening?



See here

Find out example situations that result in a 4xx or 5xx response code.

- We can get a 4xx status code if
  - Syntax of the HTTP request is wrong
  - Username or password provided is invalid
- We can get a 5xx status code if
  - Server is down :(
  - Server is overloaded with requests

Refer - 4xx - 5xx
For 4xx, try these

Quickly get your API token from here try the below steps out (any other API requiring authentication should also work). You can use this online curl utility.

Check the response code returned using DevTools for each of these cases

- 401 (Not Authorized) - Try without giving the API token -
  curl -X DELETE
  https://api.tiingo.com/tiingo/daily/AAPL/prices?startDate=2019-01-02&endDate=2019-12-12
- 404 (Not Found) - Try visiting some invalid resource - change **tiingo** to **tiigo**
  curl -X DELETE
  https://api.tiingo.com/tiigo/daily/AAPL/prices?startDate=2019-01-02&endDate=2019-12-12&token=**<your-api-token-here>**
- 405 (Method Not Allowed) - Try using DELETE -
  curl -X DELETE
  https://api.tiingo.com/tiingo/daily/AAPL/prices?startDate=2019-01-02&endDate=2019-12-12&token=**<your-api-token-here>**

5xx are server side errors and hence will be hard to replicate from user side

Milestone 4

Postman has a cool feature presenting us with commands to perform requests using cURL, Java, Python & multiple other languages. Find out how to do that.

After entering the url inside Postman, click on **code** to open the Code Snippets window