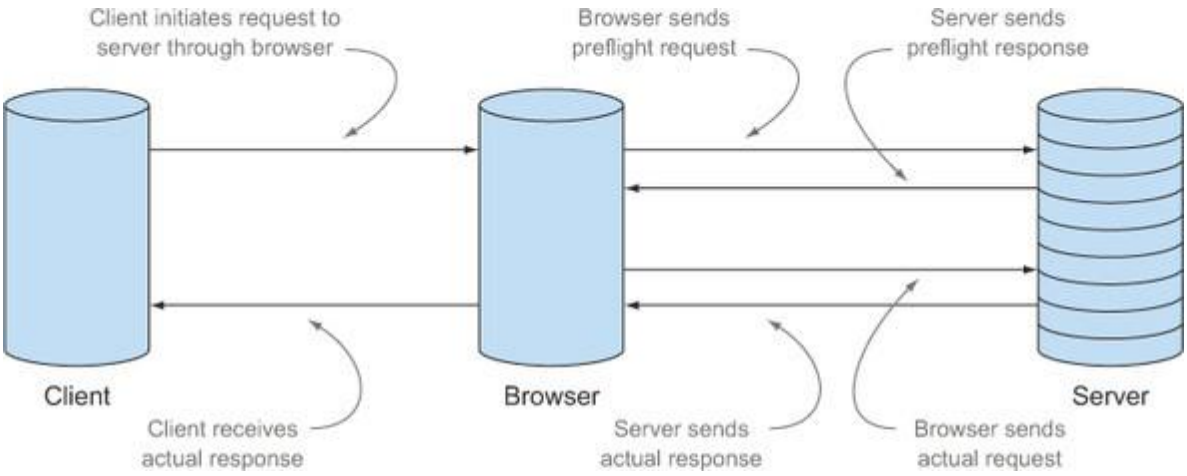
Exercise 5: Cross Origin Resource Sharing

A preflight request is a small cross-origin resource sharing (CORS) request that is send (to the server) by the browser before the actual request. It has information about used HTTP method and present HTTP headers. It can be said that the preflight request gives a chance to the server to analyse what the actual request will appear like, before it is made. The server then indicates whether or not a request is safe or return an error to the client indicating no actual request to be send.

It is an OPTIONS request, using three HTTP request headers such as:
Access-Control-Request-Method,
Access-Control-Request-Headers, and
The Origin header.


Client initiates request to server through browser
Browser sends preflight request
Server sends preflight response
Client
Browser
Server
Client receives actual response
Server sends actual response
Browser sends actual request

Lifecycle of a CORS preflight request

For example, a client might be asking a server if it would allow a DELETE request, before sending a DELETE request, by using a preflight request:

```
OPTIONS /resource/foo
Access-Control-Request-Method: DELETE
Access-Control-Request-Headers: origin, x-requested-with
Origin: https://foo.bar.org
```

If the server allows it, then it will respond to the preflight request with an Access-Control-Allow-Methods response header, which lists DELETE:

```
HTTP/1.1 204 No Content
Connection: keep-alive
Access-Control-Allow-Origin: https://foo.bar.org
Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE
Access-Control-Max-Age: 86400
```

A preflight request is used by the browser when :

- The client requests HTTP methods such as Put, Patch, Delete, Trace

- Client sets the content type request header with values except multipart/ form-data, text/ plain, application/x-www-form-urencoded
- Client sets additional request headers that do not include Accept, Content-language, Accept-language.

A preflight request is used by the browser because:

- Communicating with the old servers that do not understand CORS could be allowed and,
- Safeguarding against the potentially dangerous requests like Delete.