

## ***Give a Write up on the topic authorization & CORS***

Browsers and web applications belonging to the same origin as that of a server can access the resources easily. Whilst the others that has different origin uses Cross-Origin Resource Sharing mechanism. This ensures safety of the data. In here, additional HTTP headers are used to notify the server that the browser/web application from a different origin is trying to access the resources. i.e., a cross origin HTTP request is executed by the application if the resources belong to a different origin.

CORS work by adding new HTTP headers to let the servers describe the origins are permitted to read the information from the server. Generally, an Authorization Header provides authentication information on request and on receiving an unauthenticated request it will respond with HTTP 401 Unauthorized response. This in turn will trigger the browser to ask the user credentials. The browser repeats the same procedure but this time it adds up user credentials in the authorization header.

If an authorization header must be sent along with the request, then the browser belonging to the other origin must permit it first. But, in some cases, CORS HTTP response headers can grant access. This is because, they are the response headers and the application that handles the request should grant permission.

When performing a cross-origin request which includes authorization header, the server needs to respond with approval of the use of credentials. The browser that requests the access should set the request to include the credentials. The server that responds should set Access-Control-Allow-Credentials: true, thus the browser knows that the authentication is permitted and Access-Control-Allow-Origin should be set with \* which is a wildcard or the URL. However, in case of request with credentials, URL should be set.

In short, the browser does not worry about Access-Control-Allow-Origin header when the requests are from same domain. However, for request from different domain, it focuses on Access-Control headers and acts based on them. For instance, if the Access-Control-Allow-Origin is set with wildcard or Access-Control-Allow-Credentials: false, then it ignores the authentication headers.