What is CORS?

Jay Chow

Software Engineer, Cisco eCommerce, IT

Cisco, San Jose

9th June 2020

Agenda

1. Background of CORS
2. What is a security policy?
3. What is CORS? Why is it necessary?
4. How can CORS manage requests from external resources?
5. Access-Control-Allow-Origin
6. Reference

Background of CORS

- You may notice that webpages you visit make frequent requests to load assets like images, fonts, and more, from many different locations across the Internet
- If these requests for assets go unchecked, the security of your browser may be at risk
- For example, your browser may be subject to hijacking, or your browser might blindly download malicious code. As a result, many modern browsers follow *security policies* to mitigate such risks

What is a security policy

- Servers are used to host web pages, applications, images, fonts, and much more. When you use a web browser, you are likely attempting to access a distinct website, hosted on a server.
- Websites often request these hosted resources from different servers over the Internet
- Security policies on servers mitigate the risks associated with requesting assets hosted on different server.
- An example of a security policy is *same-origin*.
- The same-origin policy is very restrictive. Under this policy, a document (i.e., like a web page) hosted on server *A* can only interact with other documents that are also on server *A*. In short, the same-origin policy enforces that documents that interact with each other have the same *origin*.
- An origin is made up of the following three parts: the protocol, host, and port number. The details of these individual parts aren't necessary at the moment, but it is important to illustrate how the same-origin policy uses these parts.

What is a security policy(cont'd)

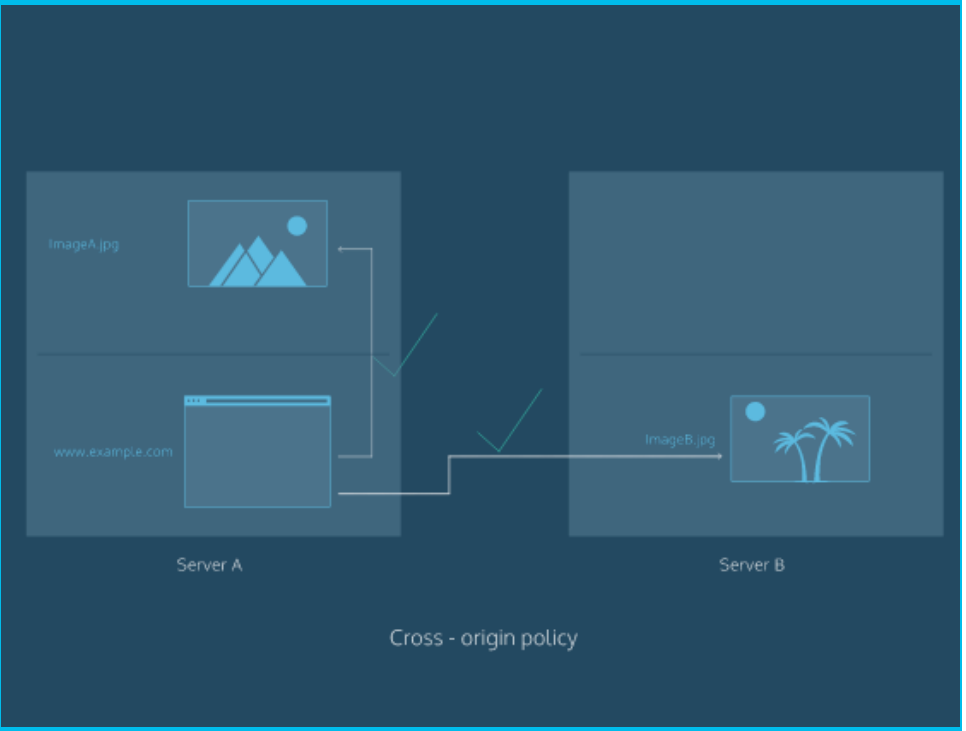
- Consider you are on this website - http://www.example.com/foo-bar.html
- Then from this url above, you navigate to another url http://www.example.com/hello-world.html

- This will be allowed as the protocol(HTTP), host(example.com) and port 80 of both urls match each other(note: 80 is default) - **same origin** policy applies here

- Note: if you try to go https://www.en.example.com/hello.html instead, this will not be allowed due to different protocol(HTTPS) and host(en.example.com) BUT this could happen with **CORS**

- Having a security policy like **same-origin** may be too restrictive

What is CORS? Why is it necessary?



A request for a resource (like an image or a font) outside of the origin is known as a *cross-origin* request

CORS (cross-origin resource sharing) manages cross-origin requests

Cross-origin requests, however, mean that servers must implement ways to handle requests from origins outside of their own. CORS allows servers to specify who (i.e., which origins) can access the assets on the server, among many other things

The CORS standard is needed because it allows servers to specify not just who can access its assets, but also *how* the assets can be accessed

Cross-origin requests are made using the standard HTTP request methods. Most servers will allow GET requests, meaning they will allow resources from external origins (say, a web page) to read their assets.

HTTP requests methods like PATCH, PUT, or DELETE, however, may be denied to prevent malicious behavior. For many servers, this is intentional. For example, it is likely that server A does not want servers B, C, or D to edit or delete its assets.

How does CORS manage requests from external resources?

- An HTTP header is a piece of information associated with a request or a response
- Headers are passed back and forth between your web browser (also referred to as a client) and a server when the web page you are on wants to use resources hosted on a different server
- Headers are used to describe requests and responses
- The CORS standard manages cross-origin requests by adding new HTTP headers to the standard list of headers. The following are the new HTTP headers added by the CORS standard:

1) **Access-Control-Allow-Origin** (will focus on ths)
2) Access-Control-Allow-Credentials
3) Access-Control-Allow-Headers
4) Access-Control-Allow-Methods
5) Access-Control-Expose-Headers
6) Access-Control-Max-Age
7) Access-Control-Request-Headers
8) Access-Control-Request-Method
9) Origin

Access-Control-Allow-Origin

- The Access-Control-Allow-Origin header allows servers to specify how their resources are shared with external domains

- When a GET request is made to access a resource on Server A, Server A will respond with a value for the Access-Control-Allow-Origin header

- Many times, this value will be *, meaning that Server A will share the requested resources with *any* domain on the Internet. Other times, the value of this header may be set to a particular domain (or list of domains), meaning that Server A will share its resources with that specific domain (or list of domains)

- The Access-Control-Allow-Origin header is critical to resource security.

References

https://www.codecademy.com/articles/what-is-cors