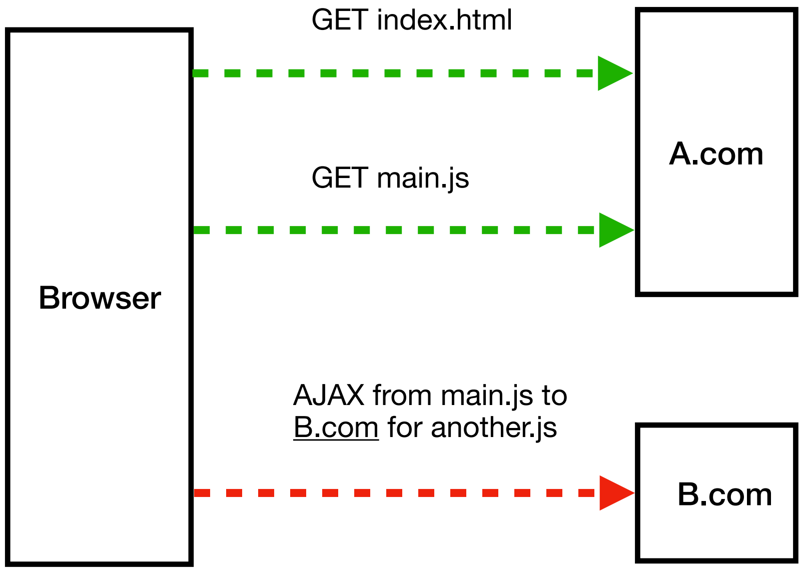
Cross-Origin Resource Sharing (CORS)

What is CORS?

Restrictions imposed by the ***server*** on the ***web browser*** regarding where resources (i.e., JavaScript, CSS, HTML, images, etc.) can be requested from.

By default, the ***initial*** server request sets the **origin** of the web browser for scripts. Any HTTP request to another origin (cross-origin) will be denied unless explicitly allowed by the **server** known as enabling CORS.



Why do we need CORS?

To allow external origin resources to be able to loaded, thus overriding the Same Origin Policy.

What is the Same Origin Policy?

The policy is to enforce allowing the execution of JavaScript from ONLY the same origin, otherwise malicious scripts can be *easily* executed without the user's knowledge such as through invisible iFrames, fake ads, links, etc.

What is considered the “Same Origin”?

- Scheme
- Hostname
    - Subdomain: **www**.csulb.edu
    - Domain: www.**csulb**.edu
    - Top-Level Domain: www.csulb.**edu**
- Port

http://www.csulb.edu/classes/index.html

Example	Same	Reason
http://www.csulb.edu/classes	Yes	Same origin
http://www.csulb.edu/classes/**another.html**	Yes	Same origin
http://www.csulb.edu/**v2**/classes/index.html	Yes	Same origin
http://www.csulb.edu/**faculty**/index.html	Yes	Same origin
http://www.csulb.edu:**80**/classes/index.html	Yes	Same origin
http://**user:password@**www.csulb.edu/classes/index.html	Yes	Same origin
http://www.csulb.edu/classes/index.html**?section=11**	Yes	Same origin
http://www.csulb.edu:**8080**/classes/index.html	No	Different port
http:www.csulb.edu/classes/index.html	No	Missing //
https://www.csulb.edu/classes/index.html	No	Different scheme
https://www.csulb.edu:**80**/classes/index.html	No	Different scheme
http://csulb.edu/classes/index.html	No	Missing subdomain
http://**v2**.www.csulb.edu/classes/index.html	No	New subdomain “v2”
http://**cecs**.csulb.edu/classes/index.html	No	Different subdomain

Why have a subdomains?

- Organizing website
- Controlling access to resources

- Network security

Subdomain	Potential Purpose
www.csulb.edu	Separating network between Internet and Intranet
cecs.csulb.edu	Grouping content by business needs (department level)
css.csulb.edu	Increased concurrent browser connection limits to host
m.csulb.edu	Server auto redirects for specific users (AB testing)

How is CORS implemented?

Browser

- Sends preflight request with specific HTTP Request Headers to ask if CORS is allowed. If ***allowed***, the actual HTTP request is sent to server.
  - Preflight only necessary for "complex" requests due to backwards compatibility with HTML Forms
  - GET is typically not preflighted
  - POST with JSON always need preflight

https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS#examples_of_access_control_scenarios

Server

- Server sends back HTTP Response Headers to preflight request consisting of allowed cross-origin
  - Can config CORS to restrict cookies to being sent

What are the CORS HTTP Headers?

Server HTTP Headers

- **Access-Control-Allow-Origin**: <origin> | *
  - Only one of this header to specify **a single origin** access
  - Should also include **Vary: Origin** to tell that server responses will vary based on **Origin** request header

- **Access-Control-Expose-Headers**: <header-name>, <header-name> | *
  - Allow JavaScript access to specific headers
- **Access-Control-Max-Age**: <seconds>
  - Specify how long preflight request can be cached
- **Access-Control-Allow-Credentials**: true
  - Specify if credentials (HTTP Cookies, HTTP Auth) can be sent in actual request
  - “Simple” GET requests will need the server to send the header as part of the response or else the browser will ignore response
  - Default is false
- **Access-Control-Allow-Methods**: <method>, <method> | *
  - Specify the HTTP method allowed in actual request
- **Access-Control-Allow-Headers**: <header-name>, <header-name> | *
  - Specify the HTTP header allowed in actual request

Browser HTTP Headers

- **Origin**: <origin>
  - The hostname from where the request is initiated
  - Value can be empty for some requests
  - Always sent in any CORS request
- **Access-Control-Request-Method**: <method>
  - For preflight CORS request to specify the HTTP method that will be used in actual request
- **Access-Control-Request-Headers**: <field-name>, <field-name> | *
  - For preflight CORS request to specify the HTTP headers that will be sent in actual request