同站 和 同源 你理解清楚了么？

原创 ConardLi code秘密花园 2020-06-28

同站（`same-site`）和同源（`same-origin`）经常在页面跳转、`fetch()`请求、`cookie`、打开弹出窗口、嵌入式资源和`iframe`等场景中被提到，但是有相当一部分同学的理解是错误的。

源（Origin）

`Origin`是协议(例如`HTTP`或`HTTPS`)、主机名和端口的组合。例如，给定一个 URL `https://www.example.com:443/foo`，它的`Origin`就是`https://www.example.com:443`。

同源(same-origin) 和跨域(cross-origin)

具有相同协议，主机名和端口的组合的网站被视为`相同来源`。其他所有内容均视为`跨域`。

Origin A	Origin B	Explanation of whether Origin A and B are "same-origin" or "cross-origin"
https://www.example.com:443	https://www.evil.com:443	cross-origin: different domains
	https://example.com:443	cross-origin: different subdomains
	https://login.example.com:443	cross-origin: different subdomains
	http://www.example.com:443	cross-origin: different schemes
	https://www.example.com:80	cross-origin: different ports
	https://www.example.com:443	same-origin: exact match
	https://www.example.com	same-origin: implicit port number (443) matches

站（Site）

像`.com`和`.org`这样的顶级域名(`tld`)会在根区域数据库中被列出。在上面的示例中，`site`是`TLD`和它前面的部分域的组合。例如，给定一个URL `https://www.example.com:443/foo`，`site`就是`example.com`。

然而，对于`.co.jp`或`.github`这样的域名。仅仅使用`.jp`或`.io`的`TLD`是不够细粒度的。而且也没有办法通过算法确定特定`TLD`的可注册域名级别。这就是创建"有效顶级域名"列表的原因。它们在公共后缀列表中定义。`etld`列表在`publicsuffix.org/list`上维护。

整个站点命名为`eTLD + 1`。例如，假定`URL`为`https://my-project.github.io`，则`eTLD`为`.github.io`，而`eTLD + 1`为`my-project.github.io`，这被视为`site`。换句话说，`eTLD+1`是有效的`TLD`紧接其之前的域的一部分。

同站(same-site) 和 跨站(cross-site)

具有相同`eTLD+1`的网站被视为"同站"。具有不同`eTLD+1`的网站是"跨站"。

Origin A	Origin B	Explanation of whether Origin A and B are "same-site" or "cross-site"
https://www.example.com:443	https://www.evil.com:443	cross-site: different domains
	https://login.example.com:443	same-site: different subdomains don't matter
	http://www.example.com:443	same-site: different schemes don't matter
	https://www.example.com:80	same-site: different ports don't matter
	https://www.example.com:443	same-site: exact match
	https://www.example.com	same-site: ports don't matter

schemeful same-site

尽管"同站"忽略了协议("无协议的同站")，但在某些情况下，必须严格区分协议，以防止`HTTP`被用作弱通道。在这些情况下，一些文档将"同站"更明确地称为`schemeful same-site`。在这种情况下，`http://www.example.com`和`https://www.example.com`被认为是跨站点的，因为协议不匹配。

Origin A	Origin B	Explanation of whether Origin A and B are "schemeful same-site"
https://www.example.com:443	https://www.evil.com:443	cross-site: different domains
	https://login.example.com:443	schemeful same-site: different subdomains don't matter
	http://www.example.com:443	cross-site: different schemes
	https://www.example.com:80	schemeful same-site: different ports don't matter
	https://www.example.com:443	schemeful same-site: exact match
	https://www.example.com	schemeful same-site: ports don't matter

如何检查请求是否为"同站"，"同源"，或"跨站"

`Chrome`发送请求时会附带一个`Sec-Fetch-Site HTTP Header`。截至2020年4月，还没有其他浏览器支持`Sec-Fetch-Site`，这个`HTTP Header`将有以下值之一：

- cross-site
- same-site
- same-origin
- none

通过检查`Sec-Fetch-Site`的值，您可以确定请求是"同站"，"同源"还是"跨站"。