校企合作项目记录表（跟岗学习专用）

记录时段	2019 年 11 月 04 日 —— 2019 年 11 月 09 日	
企业名称	广东元心科技有限公司	
记录类型	学习记录	
学生姓名	苏俊贤	指导老师签名：
记录内容（可另页附）	1. 同源策略 如果两个页面的协议，端口（如果有指定）和主机都相同，则两个页面具有相同的源，这就是同源的定义。 同源策略在 web 应用的安全模型中是一个重要概念。在这个策略下，web 浏览器允许第一个页面的脚本访问第二个页面里的数据，但是也只有在两个页面有相同的源时。源是由 URI，主机名，端口号组合而成的。这个策略可以阻止一个页面上的恶意脚本通过页面的 DOM 对象获得访问另一个页面上敏感信息的权限。 因为同源策略，浏览器限制了我们访问另一个源的资源，这时候我们就需要跨域去访问另一个源的资源了。 2. 跨域 跨域是指从一个域名的网页去请求另一个域名的资源。比如从 http://www.baidu.com/页面去请求 http://www.google.com 的资源。跨域的严格一点的定义是：只要 协议，域名，端口有任何一个的不同，就被当作是跨域。 3. CORS CORS 是一个 W3C 标准，全称是"跨域资源共享"，它是处理跨域问题的标准做法，它允许浏览器向跨源服务器，发出 XMLHttpRequest 请求，从而克服了 AJAX 只能同源使用的限制。 CORS 的请求分为两类：简单请求和非简单请求。只要同时满足以下两大条件就算是简单请求，反之就是非简单请求。 （1）请求方法是以下三种方法之一： ▪ HEAD ▪ GET ▪ POST （2）HTTP的头信息不超出以下几种字段： ▪ Accept ▪ Accept-Language ▪ Content-Language ▪ Last-Event-ID ▪ Content-Type：只限于三个值application/x-www-form-urlencoded、multipart/form-data、text/plain	

1） 简单请求

如果是简单请求的话，浏览器会直接发出 cors 请求，就是会在头部信息里添加一个 Origin 字段。Origin 字段来说明本次请求来自哪个源（协议 + 域名 + 端口）。服务器根据这个值，决定是否同意这次请求。

如果 Origin 指定的源，不在许可范围内，服务器会返回一个正常的 HTTP 回应。浏览器发现，这个回应的头信息没有包含 Access-Control-Allow-Origin 字段，就知道出错了，从而抛出一个错误，被 XMLHttpRequest 的 onerror 回调函数捕获。注意，这种错误无法通过状态码识别，因为 HTTP 回应的状态码有可能是 200。

如果 Origin 指定的域名在许可范围内，服务器返回的响应，会多出几个头信息字段。

```
Access-Control-Allow-Origin: http://api.bob.com
Access-Control-Allow-Credentials: true
Access-Control-Expose-Headers: FooBar
Content-Type: text/html; charset=utf-8
```

（1） Access-Control-Allow-Origin

该字段是必须的。它的值要么是请求时 Origin 字段的值，要么是一个*，表示接受任意域名的请求。

（2） Access-Control-Allow-Credentials

该字段可选。它的值是一个布尔值，表示是否允许发送 Cookie。默认情况下，Cookie 不包括在 CORS 请求之中。设为 true，即表示服务器明确许可，Cookie 可以包含在请求中，一起发给服务器。这个值也只能设为 true，如果服务器不要浏览器发送 Cookie，删除该字段即可。

（3） Access-Control-Expose-Headers

该字段可选。CORS 请求时，XMLHttpRequest 对象的 getResponseHeader()方法只能拿到 6 个基本字段：Cache-Control、Content-Language、Content-Type、Expires 、 Last-Modified 、 Pragma 。 如 果 想 拿 到 其 他 字 段 ， 就 必 须 在 Access-Control-Expose-Headers 里面指定。像上面图片的例子，getResponseHeader('FooBar')可以返回 FooBar 字段的值。

2） 非简单请求

非简单请求的 CORS 请求，会在正式通信之前，增加一次 HTTP 查询请求。浏览器先询问服务器，当前网页所在的域名是否在服务器的许可名单之中，以及可以使用哪些 HTTP 动词和头信息字段。只有得到肯定答复，浏览器才会发出正式的 XMLHttpRequest 请求，否则就报错。

如果请求方法是 PUT 或 DELETE，或者 Content-Type 字段的类型是 application/json 的话，浏览器就会发现这是一个非简单请求，就自动发送一个查询请求，要求服务器确认可以这样请求，下面是查询请求的头部信息：

```
OPTIONS /cors HTTP/1.1
Origin: http://api.bob.com
Access-Control-Request-Method: PUT
Access-Control-Request-Headers: X-Custom-Header
Host: api.alice.com
Accept-Language: en-US
Connection: keep-alive
User-Agent: Mozilla/5.0...
```

查询请求的 method 是 OPTIONS，表示这个请求是来询问的。头信息里面，关键字段是 Origin，表示请求来自哪个源。

除了 Origin 字段，查询请求的头信息包括两个特殊字段。

（1） Access-Control-Request-Method

该字段是必须的，用来列出正常请求所用到的 HHTTP method，上面的例子是 PUT。

（2） Access-Control-Request-Headers

该字段是一个逗号分隔的字符串，指定浏览器 CORS 请求会额外发送的头信息字段。

服务器收到查询请求后，检查了 Origin、Access-Control-Request-Method 和 Access-Control-Request-Headers 字段以后，确认允许跨源请求，就可以做出回应。

```
HTTP/1.1 200 OK
Date: Mon, 01 Dec 2008 01:15:39 GMT
Server: Apache/2.0.61 (Unix)
Access-Control-Allow-Origin: http://api.bob.com
Access-Control-Allow-Methods: GET, POST, PUT
Access-Control-Allow-Headers: X-Custom-Header
Content-Type: text/html; charset=utf-8
Content-Encoding: gzip
Content-Length: 0
Keep-Alive: timeout=2, max=100
Connection: Keep-Alive
Content-Type: text/plain
```

上面的 HTTP 回应中，关键的是 Access-Control-Allow-Origin 字段，表示 http://api.bob.com 可以请求数据。该字段也可以设为星号，表示同意任意跨源请求。

（如果浏览器否定了查询请求，会返回一个正常的 HTTP 回应，但是没有任何 CORS 相关的头信息字段。

```
XMLHttpRequest cannot load http://api.alice.com.
Origin http://api.bob.com is not allowed by Access-Control-Allow-Origin.
```

如果服务器肯定了查询请求，就会返回以下的 CORS 相关字段：

```
Access-Control-Allow-Methods: GET, POST, PUT
Access-Control-Allow-Headers: X-Custom-Header
Access-Control-Allow-Credentials: true
Access-Control-Max-Age: 1728000
```

（1） Access-Control-Allow-Methods
该字段必需，它的值是逗号分隔的一个字符串，表明服务器支持的所有跨域请求的方法。注意，返回的是所有支持的方法，而不单是浏览器请求的那个方法。这可以避免浏览器发送多次查询请求。
（2） Access-Control-Allow-Headers
如果浏览器请求包括 Access-Control-Request-Headers 字段，则 Access-Control-Allow-Headers 字段是必需的。它也是一个逗号分隔的字符串，表明服务器支持的所有头信息字段，不限于浏览器在"预检"中请求的字段。
（3） Access-Control-Allow-Credentials
该字段与简单请求的一致。
（4） Access-Control-Max-Age
该字段可选，用来指定本次预检请求的有效期，单位为秒。上面结果中，有效期是 20 天（1728000 秒），即允许缓存该条回应 1728000 秒（即 20 天），在此期间，不用发出另一条查询请求。

一旦服务器通过了"预检"请求，以后每次浏览器正常的 CORS 请求，就都跟简单请求一样，会有一个 Origin 头信息字段。服务器的回应，也都会有一个 Access-Control-Allow-Origin 头信息字段。
下面是浏览器发送的 HTTP 头部信息：

```
PUT /cors HTTP/1.1
Origin: http://api.bob.com
Host: api.alice.com
X-Custom-Header: value
Accept-Language: en-US
Connection: keep-alive
User-Agent: Mozilla/5.0...
```

Origin 字段是浏览器自动添加的。
服务器接收到请求就会回应以下信息：

```
Access-Control-Allow-Origin: http://api.bob.com
Content-Type: text/html; charset=utf-8
```

这样就实现了跨域请求。