

HTTP

CORS: 跨域资源共享

浏览器同源策略

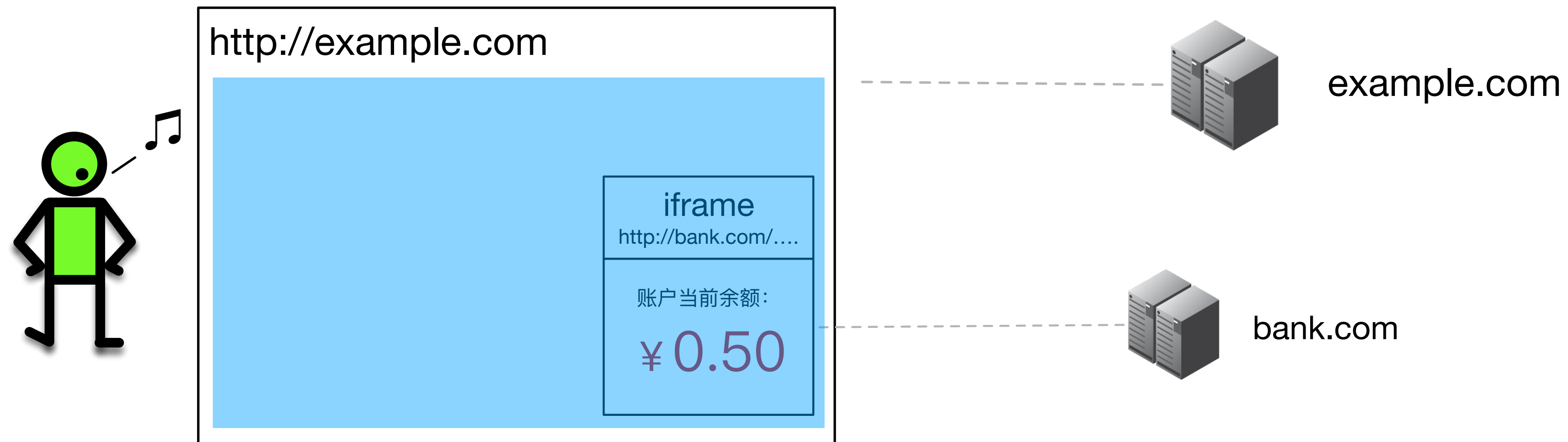
同源标准

- 协议相同
- 域名相同
- 端口相同

源网址URL	被调用URL	同源/跨域
http://s1.example.com/	https://s1.example.com/users/2/	跨域
http://s1.example.com/	http://s2.example.com/users/2/	跨域
http://s1.example.com/	http://s1.example.com:8081/users/2/	跨域
http://s1.example.com/usercenter/friends.html	http://s1.example.com/users/2/	同源

浏览器限制跨域的原因

预防部分CSRF攻击



JSONP

- JSONP: JSON with Padding
- 利用<script>标签，返回javascript

s1.example.com 中页面添加script标签:

```
<script type="text/javascript"
      src="http://s2.example.com/RetrieveUser?UserId=123&jsonp=parseResponse">
</script>
```

s2.example.com 服务器返回:

```
parseResponse({ "Name": "小明", "Id": 1823, "Rank": 7 })
```

CORS

- 跨域资源共享 Cross-Origin Resource Sharing
- 解决ajax同源策略问题，允许向外部服务器发出XMLHttpRequest
- 前端javascript处理跨域和同源ajax完全相同
- 新版浏览器支持（IE需10以上）

请求的分类

- 简单请求
- 非简单请求



- 请求方式一下三种之一
 - HEAD
 - GET
 - POST
- HTTP头信息在以下范围内
 - Accept
 - Accept-Language
 - Content-Language
 - Last-Event-ID
 - Content-Type仅限以下三个
 - application/x-www-form-urlencoded
 - multipart/form-data
 - text/plain

请求的处理（简单请求）

直接发送请求，在请求头中增加Origin

Origin: <http://s1.example.com:8080>

服务器如果允许跨域访问，需回应增加几

[Access-Control-Allow-Origin: http://s1.example.com:8080](http://s1.example.com:8080)

Access-Control-Allow-Credentials: true

Access-Control-Expose-Headers: X-My-Header

XMLHttpRequest对象的getResponseHeader()方法只能拿到一些最基本的响应头，Cache-Control、Content-Language、Content-Type、Expires、Last-Modified、Pragma，如果要访问其他头，则需要服务器设置Access-Control-Expose-Headers响应头



请

求)

使用OPTIONS

OPTIONS /cors

Origin: http://s1.e

Access-Control-I

Access-Control-I

服务器如果允

Access-Control-A

Access-Control-A

Access-Control-A

Access-Control-A

Access-Control-M

预检请求完成

客户端

服务器

预检请求

Preflight Request

正式请求

Main Request

OPTIONS /api/createuser HTTP/1.1
Origin: http://s1.example.com:8080
Access-Control-Request-Method: POST
Access-Control-Request-Headers: Content-Type, X-My-Header

HTTP/1.1 200 OK
Access-Control-Allow-Origin: http://s1.example.com:8080
Access-Control-Allow-Methods: POST, GET, PUT, OPTIONS
Access-Control-Allow-Headers: X-My-Header, Content-Type

POST /api/createuser HTTP/1.1
X-My-Header: xxx
Content-Type: text/json; charset=UTF-8
Origin: http://s1.example.com:8080
Access-Control-Request-Method: POST
Access-Control-Request-Headers: X-My-Header, Content-Type

HTTP/1.1 200 OK
Access-Control-Allow-Origin: http://s1.example.com:8080

其它解决方法

- 反向代理服务器（Nginx等）
- 后端程序接口代理

跨域的前端交互

s1.example.com上网页，需要和s2.example.com共享cookie

- Cookie

```
cookie.setDomain(".example.com");
```

example.com页面通过iframe显示子域内容，可在iframe内设置

```
document.domain = "example.com";
```

- 前端HTML5 `windows.postMessage`
- 前端iframe，利用url hash变化或window.title变化