DAY30 跨域

2023年5月29日 11:43

跨域概述

1.获取第三方接口数据?

获取淘宝搜索引擎的数据(不是所有的网站都能找到这样的接口地址)

淘宝搜索引擎的接口是第三方的接口(不是本公司内部的接口)

```
ajax_promise({
  url: 'https://suggest.taobao.com/sug?k=1&area=c2c&q=apple&code=utf-8&ts=1685321917665'
}).then(res => {
  console.log(res);
})
```

上面的代码产生跨域错误的核心关键字：No 'Access-Control-Allow-Origin'

2.为什么会产生上面的跨域错误

因为浏览器为了自身的安全，设置了同源策略

3.什么叫同源策略

浏览器阻止从一个域名上面去获取另一个域名上面的数据(信息)

同源策略是浏览器最基本的安全策略。

4.那些情况会产生跨域

4.1.域名不同

4.2.协议不同产生跨域：http 和 https产生跨域

4.3.端口不同产生跨域

4.4.一级域名和二级域名：https://www.baidu.com/ -> https://news.baidu.com/

4.5.域名和域名对应的ip地址产生跨域

例如：

本地的域名 localhost

本地的ip 127.0.0.1

```
ajax_promise({
  url: 'http://localhost:8888/goods/list'
}).then(res => {
  console.log(res);
});
```

5.如何解决跨域

5.1.jsonp:前端的核心 - 是一种接口的格式

5.2.后端代理：后端获取第三方的接口，将数据传给前端

5.3.cors：后端将要跨域访问的域名添加到接口代码中

5.4.nginx：后端配置

jsonp

一.接口的常用格式

1.json格式

2.jsonp格式

3.xml格式 - 目前使用的非常少

二.jsonp

JSONP(json with padding):本意是利用json数据进行填充

1.script标签里面src不存在跨域，引入任意文件(包括第三方的文件)，并且使用里面的代码。

```
<!-- 引入第三方文件 -->
<script src="https://cdn.bootcdn.net/ajax/libs/jquery/3.6.4/jquery.js">
</script>
<script>
  console.log(jQuery);//jQuery就是上面引入js文件里面的对象
</script>

<!-- 引用本地的txt文件，里面包含js代码 -->
hello.txt 内容
const arr = ["zhangsan","lisi","wangwu"];

<script src="./hello.txt"></script>
<script>
  console.log(arr);// ['zhangsan', 'lisi', 'wangwu']
</script>

jsonp.txt 内容
fn(["zhangsan","lisi","wangwu"])

<script>
  function fn(a) {
    console.log(a);
  }
</script>
<script src="./jsonp.txt"></script>
```

注意：因为是两个script，必须先声明函数，再调用函数

```
<!-- 利用jsonp手段获取淘宝接口的数据 -->
<script>
  function taobao(a) {
```

```
    console.log(a);
  }
</script>
<script src="https://suggest.taobao.com/sug?k=1&area=c2c&q=衣服&code=utf-8&ts=
1685321917665&callback=taobao"></script>
```

JSONP的特点 - 非常重要

1.script标签里面src属性不存在跨域，可以引入任意文件(包括第三方的文件)，并且使用里面的代码。

2.JSONP仅支持get方式(将携带的大量参数拼接到地址栏后面)，这也是JSONP的弊端。

3.利用回调约到函数的名称

其他解决跨域方式

一.后端代理：

1.前端获取第三方接口数据，不是jsonp格式

2.让本公司的后端帮我们获取第三方的接口数据，后端不存在跨域(后端脱离浏览器)

3.让本公司的后端将数据做成接口，提供本公司的前端使用

```
// ajax_promise({
//   url: 'http://localhost/JS2303/week06/DAY30/code/demo.php'
// }).then(res => {
//   console.log(res);
// })
```

注意：看一下，你们的电脑无法预览。

二.cors：跨站资源共享

制作接口的时候，将需要跨域访问的域名添加到接口中。

只要在服务端设置Access-Control-Allow-Origin就可以实现跨域请求。

```
ajax_promise({
  url: 'http://localhost/JS2303/week06/DAY30/code/demo.php'
}).then(res => {
  console.log(res);
});
```

三.nginx：后端配置(反向代理)

后端打开配置文件，利用配置文件进行修改

通过代理的服务器实现获取数据，然后将数据返回给前端

提供视频

四. .....