

Vue2.0 中使用 jsonp 解决跨域

一、同源策略

要理解跨域，要先了解一下“同源策略”。同源是指，域名，协议，端口相同。“同源策略”，简单的说就是基于安全考虑，当前域不能访问其他域的东西。

URL	是否同源	说明
http://www.a.com https://www.a.com	不同源	协议不同
http://www.a.com http://www.b.com	不同源	域名不同
http://www.a.com:8080 http://www.a.com:1000	不同源	端口号不同
http://www.a.com/a.js http://www.a.com/b.js	同源	协议、域名、端口号都相同
http://www.a.com/test1/a.js http://www.a.com/test2/b.js	同源	同一域名下，不同文件夹

JSONP 由两部分组成：回调函数和数据。回调函数是当响应到来时应该在页面中调用的函数。回调函数的名字一般是在请求中指定的。而数据就是传入回调函数中的 JSON 数据。

二、参数介绍：

```
jsonp(url, opts, fn)
```

url (String) 服务器端数据接口地址

opts (Object) 一般只需关注 param 即可

(1) param (String) 默认是 callback,这是与后端约定的参数名称,也可以随便定义,只要前后端统一

(2) timeout (Number) 请求超时时间,默认是 6000ms

(3) prefix (String) callback 值的前缀,默认是 _jp

(4) name (String) 指定全局注册的回调方法名,一般不会用到,因为默认是 prefix+自增数字

fn 回调方法,用 es6 Promise

三、跨域处理

1、 首先安装这个库,执行命令:

```
npm install --save jsonp
```

2、 具体用法

```
import originJSONP from 'jsonp' //建一个叫jsonp.js 的文件,先引入 jsonp 库文件;
```

对外暴露方法 url:请求服务器地址, data:参数

```
export default function jsonp(url,data,option) {  
  url += (url.indexOf('?')<0?'':'&')+param(data)  
  return new Promise(function(resolve,reject) {  
    originJSONP(url,option, function(err,data) {  
      if(!err){  
        resolve(data)  
      }else{  
        reject(err)  
      }  
    })  
  })  
}
```

```

    }

    })

  })

}

```

定义一个将 Object 的参数处理成为 url 挂载参数的形式的函数

```

export function param(data) {

  let url = ""

  for (var k in data) {

    let value = data[k] !== undefined ? data[k] : ""

    url += '&' + k + '=' + encodeURIComponent(value)

  }

  return url ? url.substring(1) : ""

} //到这里 这个 Promise 就封装好了，接下来就具体调用来获取数据

```

3、建一个 js 文件，名字自己定义

import jsonp from '../jsonp.js' //引入前面封装的 jsonp.js(路径可根据具体的情况进行书写)

定义前后端统一参数

```

const options = {

  param:'Callback'

};

```

4、获取数据

```
export function getDataFunc() {  
  
  const url = '// 可以根据需求，进行 url 地址的设置'  
  
  const data = {  
  
    //根据具体的需求，进行定义参数  
  
  }  
  
  return jsonp(url,data,options)  
  
}
```

封装好的 getDataFunc() 在需要的 vue 文件中引用并使用。