JavaScript里的网络请求需要在同一个域之下，比如http://localhost:80 在此域名下的二级或者三级域名可以发送网络请求，而域名变成http://localhost:8080 或者 https://www.baidu.com/之类就无法取得get或post请求的返回值(请求可以正确发送，但是返回值会被安全策略拦截)，一般情况下只能使用jsonp发起get请求，因为浏览器安全策略对于图片 js文件等资源也就是用scr发起的请求不会做跨域限制，这种方案鄙人实验过了，无法获取返回值，经过搜索终于找到了Tampermonkey脚本实现跨域通信的方法：使用GM\_开头的特权函数。

一般的Tampermonkey脚本头部如下：

// ==UserScript==

// @name 测试

// @namespace http://your.homepage/

// @version 0.1

// @description enter something useful

// @author You

// @match https://www.baidu.com/

// @grant none

// ==/UserScript==

注意倒数第二行：

@grant none

可以在这里声明要使用的GM\_特权函数，这些带GM\_开头的函数可以突破浏览器的一些常规的安全策略。想实现跨域访问可以使用 GM\_xmlhttpRequest 特权函数。

下面是修改后的文件头：

// ==UserScript==

// @name 测试

// @namespace http://tampermonkey.net/

// @version 0.1

// @description try to take over the world!

// @author You

// @match https://www.baidu.com/

// @grant GM\_xmlhttpRequest

// @grant GM\_download

// ==/UserScript==

GM\_download 是下载函数

这样就可以使用 GM\_xmlhttpRequest 特权函数突破常规安全策略实现跨域发送消息，下面是例子：

GM\_xmlhttpRequest({

method: "POST",

url: "http://localhost/cannot",

headers: {

"Content-Type": "application/x-www-form-urlencoded;charset=utf-8"

},

data:"content=erwer",

onload: function(response){

console.log("请求成功");

console.log(response.responseText);

},

onerror: function(response){

console.log("请求失败");

}

});

GM\_xmlhttpRequest 具体的各种参数都是什么含义请参看鄙人博客里的Tampermonkey中文文档。保存后启动Tampermonkey脚本会弹出一个警告界面，询问使用者是否允许跨域访问，是只允许一次还是一直允许。

Promise化版本：

//send数据函数

//参数1：url;参数2：请求类型get或post;参数3：post的body;

function runAsync(url,send\_type,data\_ry) {

var p = new Promise((resolve, reject)=> {

GM\_xmlhttpRequest({

method: send\_type,

url: url,

headers: {

"Content-Type": "application/x-www-form-urlencoded;charset=utf-8"

},

data:data\_ry,

onload: function(response){

//console.log("请求成功");

//console.log(response.responseText);

resolve(response.responseText);

},

onerror: function(response){

//console.log("请求失败");

reject("请求失败");

}

});

})

return p;

}

//调用↓

runAsync("http://localhost/cannot/toilet","POST","content=erwer").then((result)=> {return result;}).then(function(result){

console.log("第57行"+result);

});

参考资料：

https://www.cnblogs.com/onlyxx/p/5179459.html

————————————————

版权声明：本文为CSDN博主「l198738655」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/l198738655/article/details/106219547