jsonP跨域0525

跨域请求的几种方法：

1.高版本浏览器XMLHttpRequest配合后端设置header权限；

2.服务器代理；

3.ifram的src;

4.flash

5.jsonp

JSONP:

全局要放一个跟数据对应的函数；

静态的资源都是同步，

创建script标签并且插入这是异步操作；

JsonP的实现原理

通过创建script标签动态添加到页面（因为script标签会解析js代码）；

相当于说调用全局的函数，将数据传入到实参中（说明数据必须是函数名字加（）的数据才能使用jsonp）,全局的函数可以直接拿到实参的的数据从而实现了跨域请求；

```
var os = document.creatElement('script');
os.src = '.....网址'
```

2种删除方式，直接创建就立马删除，是不影响请求的，因为，当你创建script标签的那一刻起，*就已经向服务器发送了一次请求*，这次请求是异步操作，响应的时间跟服务器有关，当服务器运行完返回的时候，自动会调用fn1(传入数据)，只要页面中有个fn1函数，就会执行并且成功接收数据。

JSONP的小封装：

```
jsonp({
            url:'https://sp0.baidu.com/5a1Fazu8AA54nxGko9WTAnF6hhy/su',
            data:{
                wd: user.value,
                json:1
            },
            callback:'cb',
            success:function(data){
                var html = '';
                data.s.forEach((e,i)=>{
                    if(i > 3){
                        return;
                    }
                    html += '<li>'+e+'</li>';
                });
                ul.innerHTML = html;
            }
        });

function Jsonp(json){
    //防止函数名字重名加了时间戳加上随机数；
    var fnName = 'fn' + +new Date() + Math.random();
    //因为函数不能有小数点，所以把小数点替换成空；
    fnName = fnName.replace('.' , '');
    //存[ key1 = value1,key2 = value2]
```

```
    var  arr = [];
    for(var attr in json.data){
       arr.push(attr + '=' + json.data[attr]);
    }
    json.data = arr.join('&')//xx=xxx&xxx=xxxx
    var oS = document.creatElement('script');
    //url + xx=xxx&xx=xxxx&cn = fn2666225544
    oS.src = json.url + '?'+json.data+'&'+json.callback+'='+fnName;
    document.getElementsByTagName('head')[0].appendChild(oS);
        oS.remove();
        //当服务器请求成功之后会调用下面这个函数并且把数据传到data上
        //通过服务器调这个函数，就执行json.success把数据传出去
        window[fnName] = function (data){
            json.success(data);
        }
}
```