26-第二十六章 Ajax封装 jsonp (二)

一、Ajax封装

```
    function request(){
            ajax({
                type:'POST',//请求类型
                url:'data.json',//请求路径
                asyn:true,//是否异步
                data:{//数据
                    username:'千寻',
                    password:'888'
                },
                dataType:'text',//请求数据类型
                success:function(response){//请求成功处理函数
                    console.log( response );
                   var response = JSON.parse(response);
                    console.log( response[2].name );
                },
                error:function(status){//请求失败处理函数
                    console.log(status);

                }

            })

        }

//ajax 封装
        function ajax(obj){
            var type = obj.type||'GET',//请求类型
            url = obj.url,//url处理
            asny = obj.asny!==true,//异步处理
            data = '',//预设写入数据
            dataType = obj.dataType||'json',//请求类型
            success = obj.success,//回调函数
            error = obj.error;//错误处理函数
            for(key in obj.data){//数据处理
                data += key+'='+obj.data[key]+'&';
            }
            data = encodeURI(data);
            //console.log(data);
            var xhr=new XMLHttpRequest();
            //console.log(xhr);
            if(window.XMLHttpRequest){
                xhr = new XMLHttpRequest();
            }else{
                try{
                   xhr = new ActiveXObject('Msxml2.XMLHTTP.6.0');
                }catch(e){
                   xhr = new ActiveXObject('Msxml2.XMLHTTP.3.0');
                }
```

```
            }
            //如果是GET请求方式,设置数据
            if(type.toUpperCase() == 'GET'){
                var date = new Date().getTime();//添加一个时间用来处理，GET方式缓存的问题
                //console.log(date);
                url = url+'?'+data+date;
                data =null;
            }else if(dataType.toUpperCase() == 'XML'){
                data =null;
            }
            xhr.open(type,url,asny);//请求架设
            xhr.setRequestHeader('content-Type','application/x-www-form-urlencoded');//设置请求头信息
            //console.log(data);
            xhr.send(data);//发送数据
            xhr.onreadystatechange= function(){//监听请求
                //readyState 为XMLHttpRequest对象的状态
                //status 为服务器返回的http状态码

                if(xhr.readyState== 4 && xhr.status==200){
                    var response;
                    if(dataType === 'json' || dataType === 'text'){
                        if(dataType === 'json'){
                            //解析json数据
                            response = JSON.parse(xhr.responseText);
                        }else{
                            //普通数据
                             response = xhr.responseText;
                        }

                    }else{
                        response = xhr.responseXML;
                    }
                    success&&success(response);
                }else{
                    error&&error(xhr.status);
                }
            }

        }
```

二、jsonp

jsonp,大家已经知道json是什么了，那么p是什么？p是填充的意思,json的填充

> jsonp返回的数据形式是 `callback({age:30,name:'二狗'})` 而不是 `{age:30,name:'二狗'}` 是用括号包裹,前面的名称就是‘ 填充 ’，也就是jsonp的p.
>
> - 为什么前面有一段字母呢？因为script接受的数据如果是一个json没办法保存，而调用一个函数,而{}数据作为实参传递

jsonp前端代码看起来像这样

```
1.      <script>
2.          function callback(data){ //定义接收数据的处理函数
3.              console.log( data);
4.          }
5.      </script>
6.      <script src='./8-1jsonp.php?jsonp=callback'></script>
7.      //我需要后台返回一个callback({数据})这样的一个值，实质就是调用上面的函
    数
8.
```

> jsonp中8-1jsonp.php后台中的代码

```
1.  <?php
2.      $val = $_GET['jsonp'];//获取jsonp的值：callback,
3.      $arr = array(
4.          "name"=>"千寻",
5.          "age"=>50
6.      );//php中的json
7.      echo $val."(".json_encode($arr).")";//JSON对象进行转字符串处理
8.      //就是返回 callback({name:'千寻', age:50})
9.
10. ?>
```