

Ajax ∞

跨域和JSONP

- 同源:如果两个页面的协议 域名和端口都相同,则两个页面具有相同的源,反之,则是跨域
- 同源策略:是浏览器提供的一个安全功能.例如:浏览器规定,A网站的JavaScript,不允许和非同源的网站C之间,进行资源的交互
注意:浏览器允许发起跨域请求,但是,跨域请求回来的数据,会被浏览器拦截,无法被页面获取
- 实现跨域数据请求 两种方法:
- JSONP: 出现的早,兼容性好(兼容低版本IE)。是前端程序员为了解决跨域问题,被迫想出来的一种临时解决方案。缺点是只支持 GET 请求, 不支持 POST 请求。
- CORS: 出现的较晚,它是 W3C 标准,属于跨域 Ajax 请求的根本解决方案。支持 GET 和 POST 请求。缺点是不兼容某些低版本的浏览器。
- JSONP实现原理 可以通过查询字符串的方式 告诉外部服务器 应该调用哪个函数 注意:JSONP并没有用到XMLHttpRequest请求
- 因此,JSONP的实现原理,就是通过 <script> 标签的 src 属性,请求跨域的数据接口,并通过函数调用的形式,接收跨域接口响应回来的数据。
- 使用jquery发起JSONP请求
 - ```
$.ajax({
 url:'http://ajax.frontend.itheima.net:3006/api/jsonp?name=zs&age=20',
 dataType:'jsonp',
 // 发送到服务端的参数名称 默认值为callback
 jsonp:'callback',
 // 自定义的回调函数名称,默认值为jQueryxxx格式
 jsonpCallback:'abc',
 success:function(res){
 console.log(res);
 }
})
```
  - 默认情况下,使用jquery发起jsonp请求,会自动携带一个callback=jQueryxxx参数  
jQueryxxx是随机生成的回调函数名称

防抖策略

- 是当事件被触发后,延迟n秒后再执行回调,如果在这n秒内事件又被触发,则重新计时
- 用户在输入框中连续输入一串字符时,可以通过防抖策略,只在输入完后,彩之星查询的请求,可以有效减少请求次数
- ```
// 1.定义延时器的id
var timer=null
// 2.定义防抖函数
function debounceSearch(kw){
  timer=setTimeout(function(){
    //获取用户输入的数据
    getSuggestList(kw)
  },500)
}
3.// 清空timer
clearTimeout(timer)
```
- 缓存
 - ```
// 定义缓存区域
var cachObj={}

// 1.获取到用户输入的内容 当做键
var k=$('#ipt').val().trim()
// 2.需要将数据作为值,进行缓存
cachObj[k]=res
```
  - 基本思想:当用户在搜索框输入一个值之后 会出来对应的建议项 将这些建议项先放在缓存区域  
再追加一个值 会出来一个新的建议项  
如果之后删掉了追加的值 则不会重复发起请求 而是将缓存区域的值显示出来

节流  
减少一段时间内事件的触发频率 防止事件被无限制触发

- 节流应用场景
- 鼠标连续不断地触发某事件（如点击），只在单位时间内只触发一次；
- 懒加载时要监听计算滚动条的位置，但不必每次滑动都触发，可以降低计算的频率，而不必去浪费 CPU 资源；
- 节流阀为空，表示可以执行下次操作；不为空，表示不能执行下次操作。
- 当前操作执行完，必须将节流阀重置为空，表示可以执行下次操作了。
- 每次执行操作前，必须先判断节流阀是否为空。