

ajax

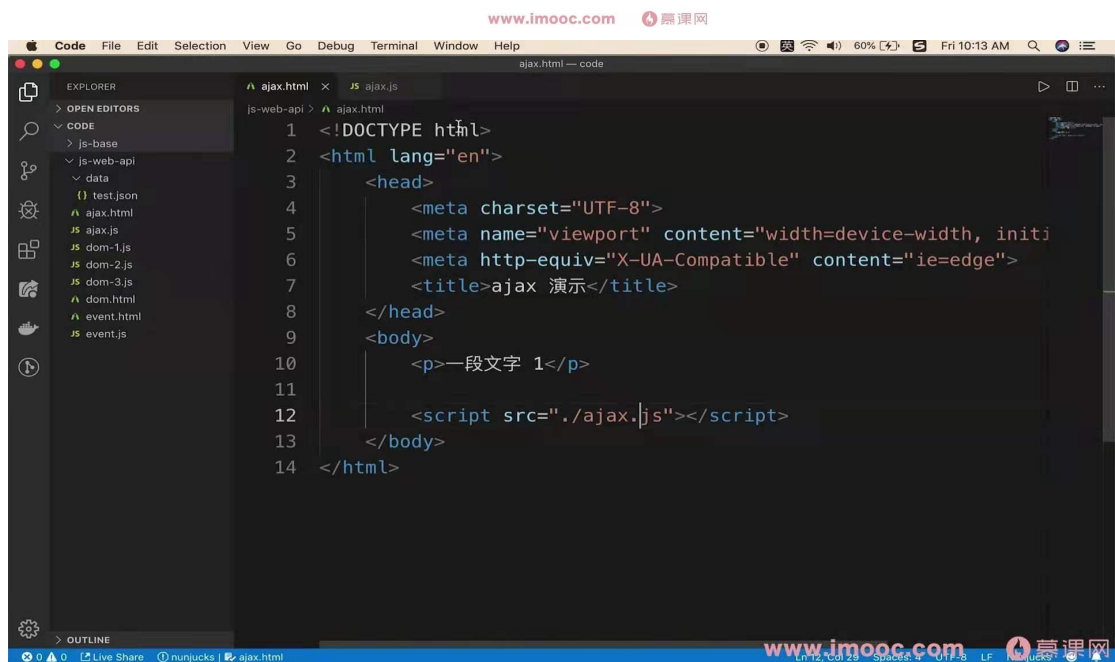
- ◆ 题目
- ◆ 知识点
- ◆ 解答

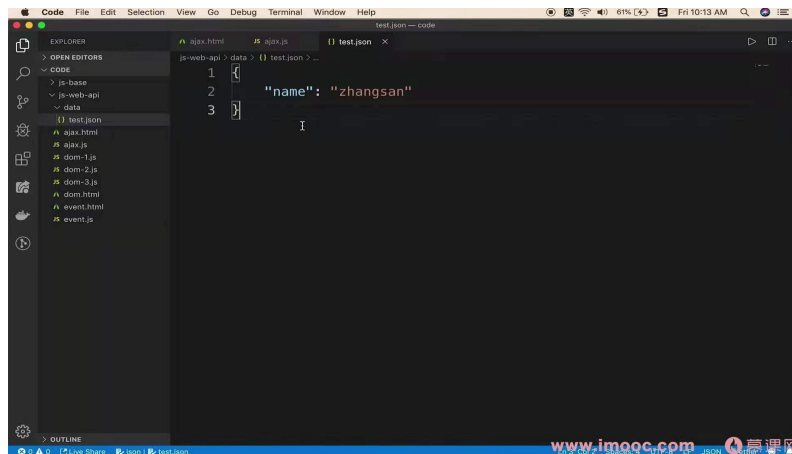
题目

- ◆ 手写一个简易的 ajax
- ◆ 跨域的常用实现方式

知识点

- ◆ XMLHttpRequest
- ◆ 状态码
- ◆ 跨域：同源策略，跨域解决方案

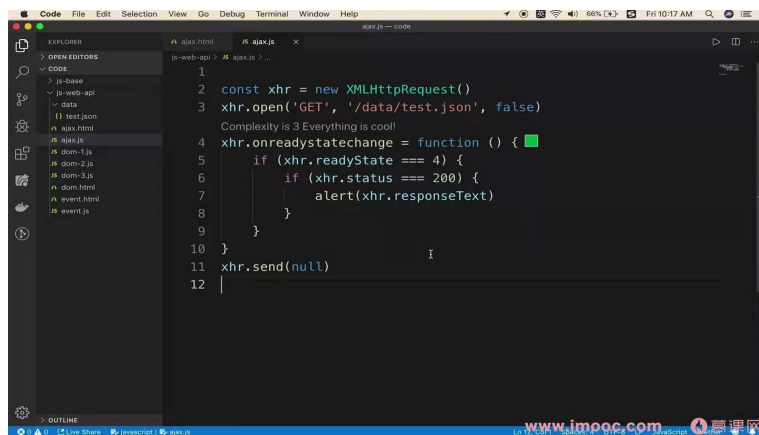




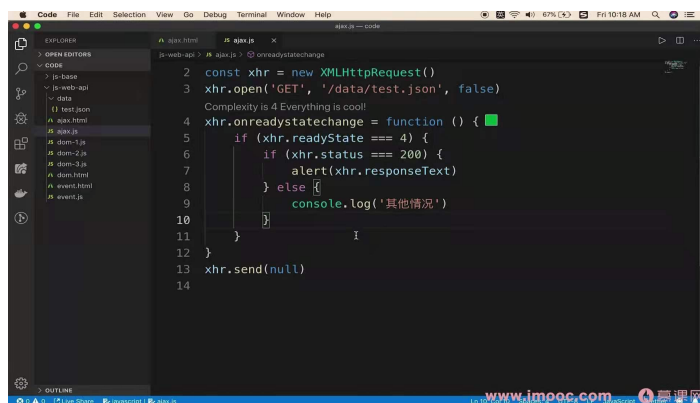
XMLHttpRequest

```
// get 请求  
const xhr = new XMLHttpRequest()  
xhr.open("GET", "/api", false)  
xhr.onreadystatechange = function () {  
  // 这里的函数异步执行, 可参考之前 JS 基础中的异步模块  
  if (xhr.readyState == 4) {  
    if (xhr.status == 200) {  
      alert(xhr.responseText)  
    }  
  }  
}  
xhr.send(null)
```

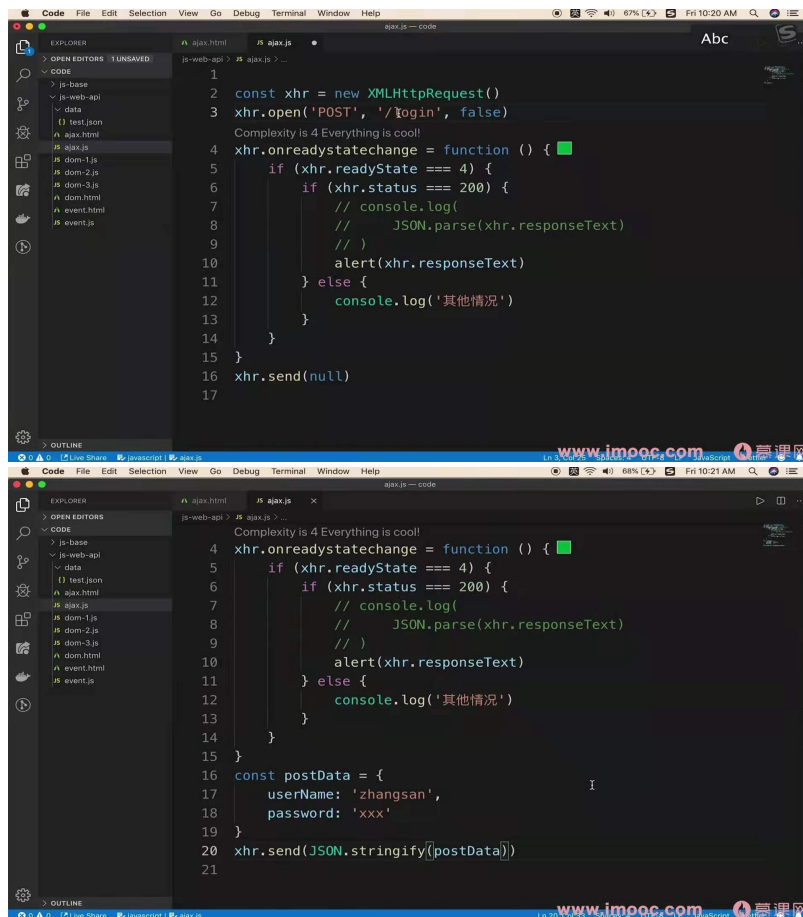
www.imoooc.com 慕课网



www.imoooc.com 慕课网



www.imoooc.com 慕课网



xhr.readyState

- 0 - (未初始化) 还没有调用send()方法
- 1 - (载入) 已调用send()方法, 正在发送请求
- 2 - (载入完成) send()方法执行完成, 已经接收到全部响应内容
- 3 - (交互) 正在解析响应内容
- 4 - (完成) 响应内容解析完成, 可以在客户端调用



xhr.status

- 2xx - 表示成功处理请求, 如 200
- 3xx - 需要重定向, 浏览器直接跳转, 如 301 302 304
- 4xx - 客户端请求错误, 如 404 403
- 5xx - 服务器端错误

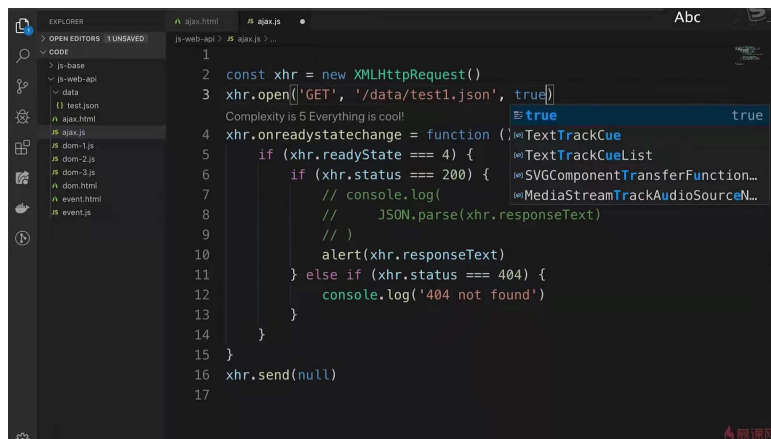


XMLHttpRequest

```
// post 请求
const xhr = new XMLHttpRequest()
xhr.open("POST", "/login", true)
xhr.onreadystatechange = function () {
  // 这里的函数异步执行, 可参考之前 JS 基础中的异步模块
  if (xhr.readyState === 4) {
    if (xhr.status === 200) {
      alert(xhr.responseText)
    }
  }
}

const postData = {
  userName: 'zhangsang',
  password: 'xxx'
}

xhr.send(JSON.stringify(postData))
```



跨域

- ◆ 什么是跨域 (同源策略)
- ◆ JSONP
- ◆ CORS (服务端支持)



同源策略

- ◆ ajax 请求时, 浏览器要求当前网页和 server 必须同源 (安全)
- ◆ 同源: 协议、域名、端口, 三者必须一致
- ◆ 前端: <http://a.com:8080/> ; server: <https://b.com/api/xxx>



加载图片 css js 可无视同源策略

- ◆ ``
- ◆ `<link href=跨域的css地址 />`
- ◆ `<script src=跨域的js地址 ></script>`



加载图片 css js 可无视同源策略

- ◆ `` 可用于统计打点，可使用第三方统计服务
- ◆ `<link />` `<script>` 可使用 CDN ，CDN 一般都是外域
- ◆ `<script>` 可实现 JSONP



跨域

- ◆ 所有的跨域，都必须经过 server 端允许和配合
- ◆ 未经 server 端允许就实现跨域，说明浏览器有漏洞，危险信号



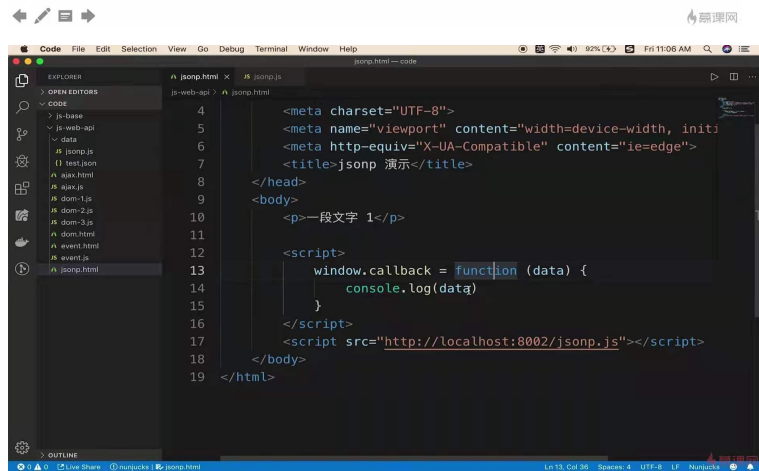
JSONP

- ◆ 访问 <https://imooc.com/> ，服务端一定返回一个 html 文件吗？
- ◆ 服务器可以任意动态拼接数据返回，只要符合 html 格式要求
- ◆ 同理于 `<script src= "https://imooc.com/getData.js" >`



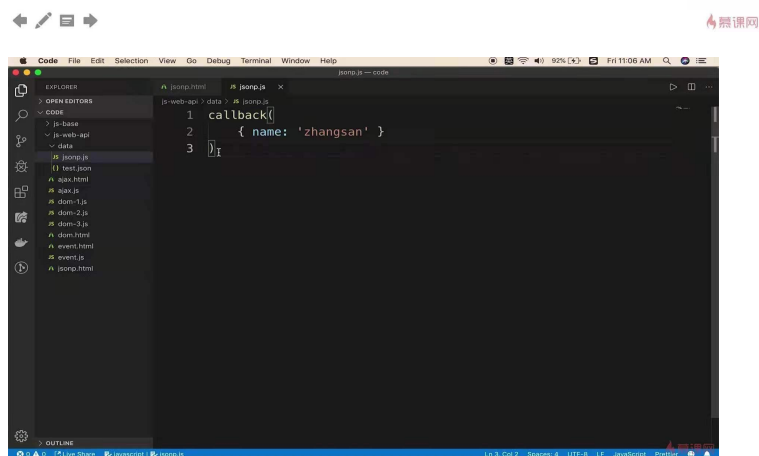
JSONP

- ◆ <script> 可绕过跨域限制
- ◆ 服务器可以任意动态拼接数据返回
- ◆ 所以，<script>就可以获得跨域的数据，只要服务端愿意返回



JSONP

```
<script>
window.callback = function (data) {
  // 这是我们跨域得到信息
  console.log(data)
}
</script>
<script src="https://kimooc.com/getData.js"></script>
<!-- 将返回 callback({ x: 100, y: 200 }) -->
```



jQuery 实现 jsonp

```
$.ajax({  
  url: 'http://localhost:8882/x-origin.json',  
  dataType: 'jsonp',  
  jsonpCallback: 'callback',  
  success: function (data) {  
    console.log(data)  
  }  
})
```



慕课网

CORS – 服务器设置 http header

```
// 第二个参数填写允许跨域的域名称，不建议直接写 "*"
response.setHeader("Access-Control-Allow-Origin", "http://localhost:8011");
response.setHeader("Access-Control-Allow-Headers", "X-Requested-With");
response.setHeader("Access-Control-Allow-Methods", "PUT,POST,GET,DELETE,OPTIONS");

// 接收跨域的cookie
response.setHeader("Access-Control-Allow-Credentials", "true");
```



慕课网