

先上一个demo(01复习AJAX发起HTTP请求与获取响应)

操作:

js可以获取部分请求和响应

整个流程

## 先上一个demo(01复习AJAX发起HTTP请求与获取响应)

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>首页</title>
6 <!--01-复习AJAX发起HTTP请求与获取响应-->
7 </head>
8
9
10 <body>
11 <button id="myButton">点我</button>
12 <script src="main.js"></script>
13 </body>
14 </html>
15
16
```

main.js

```
1 myButton.addEventListener('click', (e) => {
2   let request = new XMLHttpRequest()
3   //请求第一部分
4   request.open('post', '/xxx') //配置request
5   request.setRequestHeader('frank', '18'); //这是自己设置的
6   //请求第二部分
7   request.setRequestHeader('Content-Type', 'x-www-form-urlencoded');
8   //请求第四部分
9   request.send('我偏要设置request第四部分')
10  request.onreadystatechange = () => {
11    if (request.readyState === 4) {
12      console.log('请求响应都完毕了')
13      console.log(request.status)
14      console.log(request.statusText)
15      //获取响应第一部分
```

```

16 if(request.status >= 200 && request.status <300){
17 console.log('说明请求成功')
18 //获取响应的第二部分
19 console.log(request.getResponseHeader('Content-Type'))
20 //获取请求第四部分
21 console.log(request.responseText)
22 let string = request.responseText
23 //把符合JSON语法的字符串转换成JS对应的值
24 let object = window.JSON.parse(string)
25 //JSON.parse 是浏览器提供的
26
27 }else if(request.status >= 400){
28 console.log('说明请求失败')
29 }
30 }
31 }
32 })
33
34

```

server.js:

```

1 var http = require('http')
2 var fs = require('fs')
3 var url = require('url')
4
5 var port = process.env.PORT || 8888;
6
7 var server = http.createServer(function(request, response){
8 var temp = url.parse(request.url, true)
9 var path = temp.pathname
10 var query = temp.query
11 var method = request.method
12
13 //从这里开始看，上面不要看
14
15 if(path === '/'){ //如果用户请求的是/路径
16 let string = fs.readFileSync('./index.html')
17 response.statusCode = 200
18 response.setHeader('Content-Type', 'text/html; charset=utf-8')
19 response.write(string)
20 response.end()
21 }else if(path === '/main.js'){
22 let string = fs.readFileSync('./main.js', 'utf-8')
23 response.statusCode = 200

```

```

24 response.setHeader('Content-Type','text/javascript;charset=utf-8')
25 response.write(string)
26 response.end()
27 }else if(path='/xxx'){
28 response.statusCode = 200
29 response.setHeader('Content-Type','text/xml')
30 response.setHeader('Access-Control-Allow-Origin','*')
31 //符合JSON语法的字符串
32 response.write(`
33 {
34 "note":{
35 "to":"小谷",
36 "from":"方方",
37 "heading":"打招呼",
38 "content":"hi"
39 }
40 }
41 `)
42 response.end()
43 }else{
44 response.statusCode = 404
45 response.setHeader('Content-Type','text/html;charset=utf-8')
46 response.write('呜呜呜')
47 response.end()
48 }
49
50 //代码结束，下面不要看
51 console.log(method+''+request.url)
52 })
53
54 server.listen(port)
55 console.log('监听' + port + '成功\n请在空中转体720度然后用电饭煲打开\nhttp:localhost:'+port)
56

```

**操作:**

```
$node server 8888
```

监听8888成功

请在空中转体720度然后用电饭煲打开

```
http://localhost:8888
```

```
GET/
```

```
GET/main.js
```

```
POST/xxx
```

```
GET/
```

```
GET/main.js
```

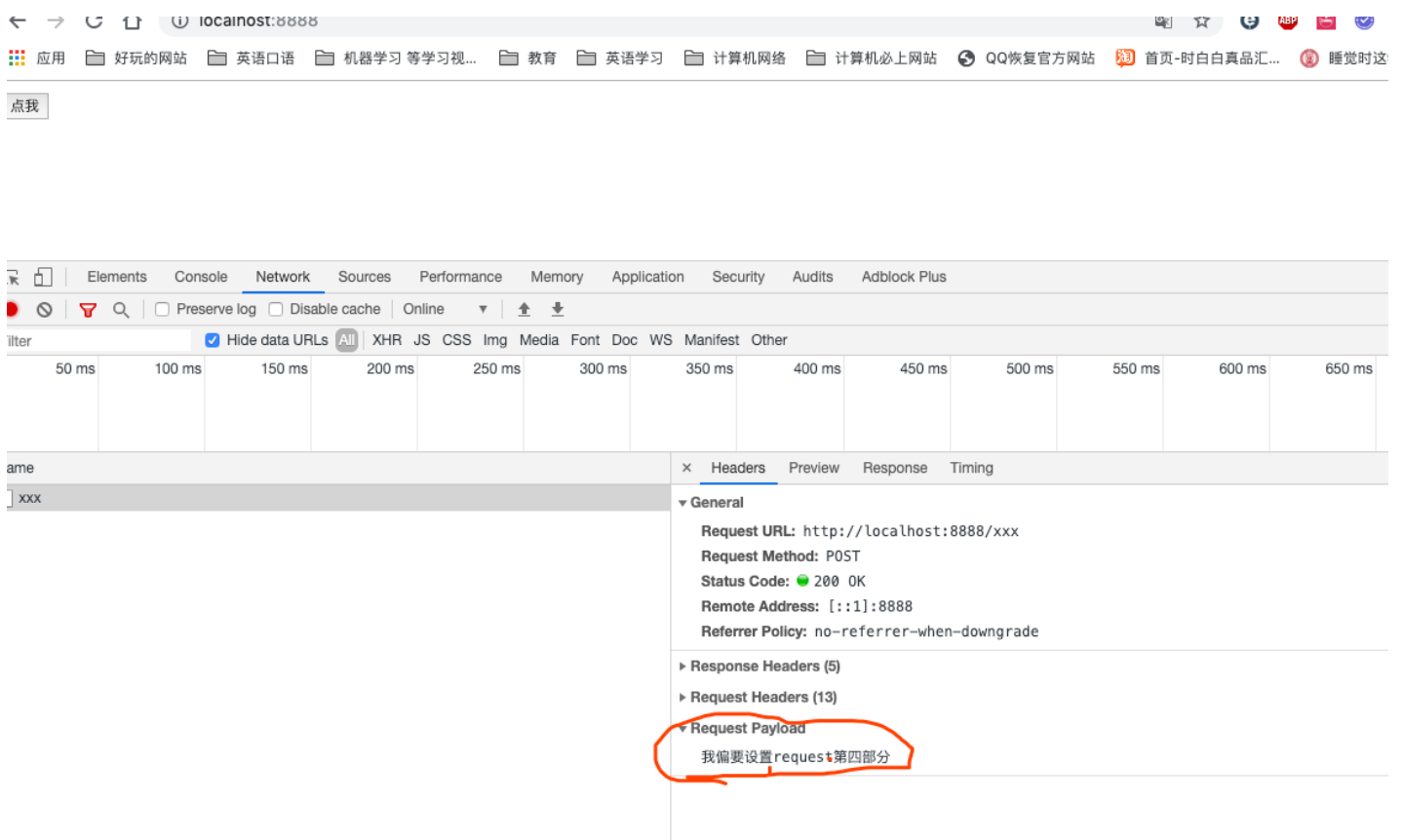
```
GET/favicon.ico
```

```
POST/xxx
```

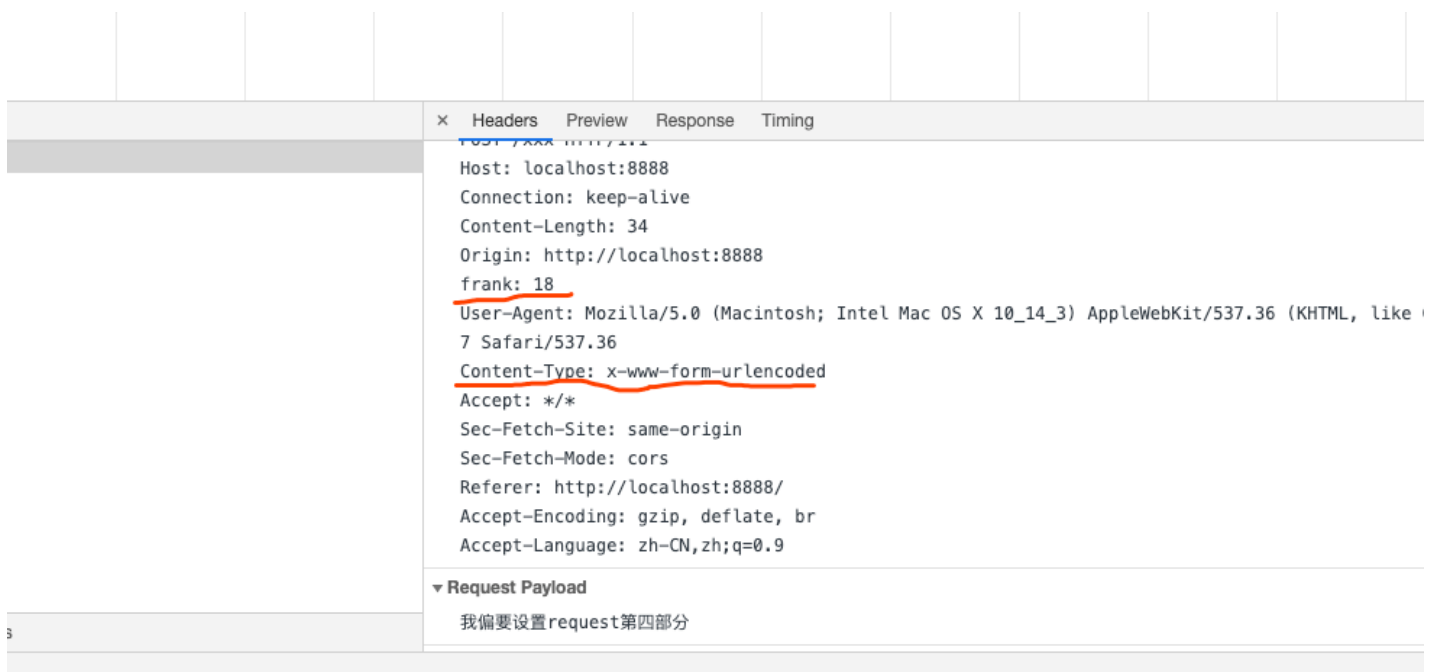
这里的8888端口是写在 `server.js` 里面的;

```
myButton.addEventListener('click', (e) => {
  let request = new XMLHttpRequest()
  //请求第一部分
  request.open('post', '/xxx') //配置request, 这里是Post的时候, 第四部分才能在chrome里面显示
  request.setRequestHeader('frank', '18'); //这是自己设置的
  //请求第二部分
  request.setRequestHeader('Content-Type', 'x-www-form-urlencoded');
  //请求第四部分
  request.send('我偏要设置request第四部分')
  request.onreadystatechange = () => {
    if (request.readyState === 4) {
      console.log('请求响应都完毕了')
      console.log(request.status)
      console.log(request.statusText)
      //获取响应第一部分
      if (request.status >= 200 && request.status < 300) {
        console.log('说明请求成功')
        //获取响应的第二部分
        console.log(request.getResponseHeader('Content-Type'))
        var console: Console
        console.log(request.responseText)
        let string = request.responseText

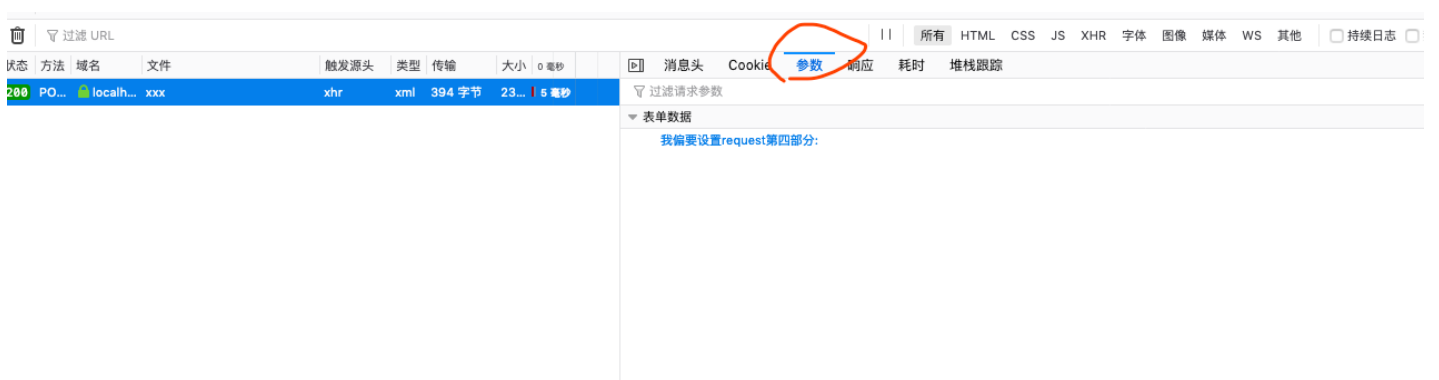
        //把符合JSON语法的字符串转换成JS对应的值
        let object = window.JSON.parse(string)
```



下面这个也是自己设置的,可以参加上面的代码



火狐浏览器里面是在参数里面:



js可以获取部分请求和响应

8 1. JS 可以设置任意请求 header 吗  
9 第一部分 request.open('get', '/xxx')  
10 第二部分 request.setHeader('content-type', 'x-www-form-urlencoded')  
11 第四部分 request.send('a=1&b=2')  
12 2. JS 可以获取任意响应 header 吗?  
13 第一部分 request.status / request.statusText  
14 第二部分 request.getResponseHeader() / request.getAllResponseHeaders()  
15 第四部分 request.responseText

## 整个流程

