

# jQuery

---

注：jQuery由于用到Ajax，放在后端编程后讲解

一个JavaScript库，极大简化了JS编程。

jQuery库包含以下功能：

- HTML 元素选取
- HTML 元素操作
- CSS 操作
- HTML 事件函数
- JavaScript 特效和动画
- HTML DOM 遍历和修改
- AJAX
- Utilities

除了上述功能外，它之所以流行，是官方、非官方提供了海量的插件。

## 安装

下载jquery，解压后，一个开发用的，一个min是压缩后的部署用。

```
<script src="jquery.js"></script>
```

## 基本语法

```
$(selector).action()
```

- 美元符号定义 jQuery
- 选择符 (selector) “查询”和“查找” HTML 元素
- jQuery 的 action() 执行对元素的操作

文档就绪函数

为了防止文档加载完之前就允许jQuery代码，需要一个网页加载完的函数。

```
$(document).ready(function(){  
    /* jQuery functions go here */  
});
```

可以简写为下面形式

```
$(function(){  
    /* jQuery functions go here */  
});
```

## jQuery 元素选择器

jQuery 使用 CSS 选择器来选取 HTML 元素。

`$("p")` 选取 `<p>` 元素。

`$("p.intro")` 选取所有 `class="intro"` 的 `<p>` 元素。

`$("p#demo")` 选取所有 `id="demo"` 的 `<p>` 元素。

## jQuery 属性选择器

jQuery 使用 XPath 表达式来选择带有给定属性的元素。

`$("[href]")` 选取所有带有 `href` 属性的元素。

`$("[href='#']")` 选取所有带有 `href` 值等于 `"#"` 的元素。

`$("[href!='#']")` 选取所有带有 `href` 值不等于 `"#"` 的元素。

`$("[href$.jpg]")` 选取所有 `href` 值以 `".jpg"` 结尾的元素。

## jQuery 事件函数

为jQuery对象增加事件回调函数

```
<html>
<head>
  <title>test jquery</title>
  <meta charset="utf-8">
  <script src="jquery.js"></script>
  <script type="text/javascript">
    $(function(){
      var flag = true;
      $('button.testclick').click(function(event){
        console.log(event.target);
        $('#root').css('color', flag?'red':'blue');
        flag = !flag;
      });
    });
  </script>
</head>
<body>
  <div id='root'>welcome to www.magedu.com</div>
  <button class='testclick'>click me</button>
</body>
</html>
```

注意

```
$('#button.testclick').click() //调用click
$('#button.testclick').click(function(){}) // 设置click事件回调
```

## Ajax

jQuery对XMLHttpRequest组件的调用接口实现了封装，更加方便调用。

默认是异步请求。

GET请求

```

<html>

<head>
  <title>test jquery</title>
  <meta charset="utf-8">
  <script src="jquery.js"></script>
  <script type="text/javascript">
    $(function () {
      var flag = true;
      $('button.testclick').click(function (event) {
        console.log(event.target);
        console.log(this);
        $('#root').css('color', flag ? 'red' : 'blue');
        flag = !flag;
      });
      $('button.ajaxget').click(function(){
        $.get('http://127.0.0.1:8000/user/show?k1=1&k2=2&k2=3',
          function(resdata){
            console.log(resdata); //resdata返回就是response的正文内容
            console.log(resdata.t1);
          });
      });
    });
  </script>
</head>

<body>
  <div id='root'>welcome to www.magedu.com</div>
  <button class='testclick'>click me</button>
  <button class='ajaxget'>GET</button>
</body>

</html>

```

```

# python服务器端django view函数
# /user/show
#
def show(request:HttpRequest):
    print('-' * 30)
    print(1, request.GET)
    print('-' * 30)
    print(2, request.POST)
    print(3, request.body)
    res = JsonResponse({'t1':1000})
    #res = HttpResponse(b'show')
    res['Access-Control-Allow-Origin'] = '*'
    return res

```

Access-Control-Allow-Origin问题

服务器端响应头设置header = {'Access-Control-Allow-Origin': '\*'}

## POST请求

```
<html>

<head>
  <title>test jquery</title>
  <meta charset="utf-8">
  <script src="jquery.js"></script>
  <script type="text/javascript">
    $(function () {
      var flag = true;
      $('button.testclick').click(function (event) {
        console.log(event.target);
        console.log(this);
        $('#root').css('color', flag ? 'red' : 'blue');
        flag = !flag;
      });
      $('button.ajaxget').click(function(){
        $.get('http://127.0.0.1:8000/user/show?k1=1&k2=2&k2=3',
          function(resdata,status){
            console.log(resdata); //resdata返回就是response的正文内容
            console.log(resdata.t1);
            console.log(status);
          });
      });
      $('button.ajaxpost').click(function(){
        $.post('http://127.0.0.1:8000/user/show?k1=1&k2=2&k2=3',
          {'p1':100, 'p2':'abc'},
          function(resdata, status){
            console.log(resdata); //resdata返回就是response的正文内容
            console.log(status);
          });
      });
    });
  </script>
</head>

<body>
  <div id='root'>welcome to www.magedu.com</div>
  <button class='testclick'>click me</button>
  <button class='ajaxget'>GET</button>
  <button class='ajaxpost'>POST</button>
</body>

</html>
```

注意观察请求头

POST请求提交JSON

```

<html>

<head>
  <title>test jquery</title>
  <meta charset="utf-8">
  <script src="jquery.js"></script>
  <script type="text/javascript">
    $(function () {
      var flag = true;
      $('button.testclick').click(function (event) {
        console.log(event.target);
        console.log(this);
        $('#root').css('color', flag ? 'red' : 'blue');
        flag = !flag;
      });
      $('button.ajaxget').click(function () {
        $.get('http://127.0.0.1:8000/user/show?k1=1&k2=2&k2=3',
          function (resdata, status) {
            console.log(resdata); //resdata返回就是response的正文内容
            console.log(resdata.t1);
            console.log(status);
          });
      });
      $('button.ajaxpost').click(function () {
        $.post('http://127.0.0.1:8000/user/show?k1=1&k2=2&k2=3',
          { 'p1': 100, 'p2': 'abc' },
          function (resdata, status) {
            console.log(resdata); //resdata返回就是response的正文内容
            console.log(status);
          });
      });
      $('button.ajaxheader').click(function () {
        $.ajax({
          method: 'POST',
          //headers:{"Content-type":"application/json"},
          contentType:"application/json",
          url: 'http://127.0.0.1:8000/user/show?k1=1&k2=2&k2=3',
          data: { 'p1': 100, 'p2': 'abc' },
          beforeSend:function(request){
            request.setRequestHeader('Content-Type', 'application/json');
          },
          success: function (resdata, status) {
            console.log(resdata); //resdata返回就是response的正文内容
            console.log(status);
          }
        });
      });
    });
  </script>
</head>

<body>
  <div id='root'>welcome to www.magedu.com</div>

```

```
<button class='testclick'>click me</button>
<button class='ajaxget'>GET</button>
<button class='ajaxpost'>POST</button>
<button class="ajaxheader">Ajax Header</button>
</body>

</html>
```

直接点击Ajax Header按钮，在浏览器控制台出现下图错误

```
✖ XMLHttpRequest cannot load http://127.0.0.1:8000/user/show?k1=1&k2=2&k2=3. Request
header field Content-Type is not allowed by Access-Control-Allow-Headers in preflight
response. t.html:1
```

在jQuery官方有这么一句

**contentType** (default: 'application/x-www-form-urlencoded; charset=UTF-8')  
Type: [Boolean](#) or [String](#)  
When sending data to the server, use this content type. Default is "application/x-www-form-urlencoded; charset=UTF-8", which is fine for most cases. If you explicitly pass in a content-type to `$.ajax()`, then it is always sent to the server (even if no data is sent). As of jQuery 1.6 you can pass `false` to tell jQuery to not set any content type header. **Note:** The W3C XMLHttpRequest specification dictates that the charset is always UTF-8; specifying another charset will not force the browser to change the encoding. **Note:** For cross-domain requests, setting the content type to anything other than `application/x-www-form-urlencoded`, `multipart/form-data`, or `text/plain` will trigger the browser to send a preflight OPTIONS request to the server.

也就是说跨域访问如果设置不是这3种Content-Type，那么会触发preflight OPTIONS请求（预检请求）。

请求头如下

#### Request Headers [view parsed](#)

```
OPTIONS /user/show?k1=1&k2=2&k2=3 HTTP/1.1
Host: 127.0.0.1:8000
Connection: keep-alive
Access-Control-Request-Method: POST
Origin: null
X-DevTools-Emulate-Network-Conditions-Client-Id: 9e499c66-4f5f-48c2-8f1a
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Maxthon/5.1.0.4000 Chrome/55.0.2883.75 Safari/537.36
Access-Control-Request-Headers: content-type
Accept: */*
DNT: 1
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN
```

正常请求头如下

## Request Headers [view parsed](#)

POST /user/show?k1=1&k2=2&k2=3 HTTP/1.1

Host: 127.0.0.1:8000

Connection: keep-alive

Content-Length: 13

Accept: \*/\*

Origin: http://127.0.0.1:8000

X-DevTools-Emulate-Network-Conditions-Client-Id: 07b6ac00-3246-4901-bff152

X-Requested-With: XMLHttpRequest

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; zh-CN) AppleWebKit/537.36 (KHTML, like Gecko) Version/5.0.1 Safari/537.36

Content-Type: application/json

DNT: 1

Referer: http://127.0.0.1:8000/static/t.html

Accept-Encoding: gzip, deflate

Accept-Language: zh-CN

Cookie: sessionid=nh2ijpghb04edu802y9yh5j6226dxpxm; csrftoken=Uk5DtUQEv

### 解决方案

使用网站来访问网页即可。

Django的静态文件配置，如下

1、settings中INSTALLED\_APPS确保有django.contrib.staticfiles

2、settings中定义静态路径，STATIC\_URL = '/static/'

3、settings中定义

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR, "static"),

]

## 习题

Ajax方式提交GET方法，从服务器查回数据，使用表格显示，要求使用JS对表格进行动态追加。