



《手写OS操作系统》小班二期招生，全程直播授课，大牛带你掌握硬核技术！

点此查看

首页 > 慕课教程 > Go工程师体系课全新版 > 6. 浏览器的跨域请求问题

全部开发者教程

观锁

5. 基于redis实现分布式锁

6. redlock详解

7. 常见的分布式锁实现方案

第14周 订单和购物车微服务

1. 表结构设计和proto文件定义

2. 支付宝支付的相关链接

3. 用户操作 – 表结构设计

4. 用户服务 – proto定义

5. service代码

6. userop-web代码

第16周 elasticsearch实现搜索微服务

1. 什么是elasticsearch

2. elasticsearch安装



bobby · 更新于 2022-11-08

← 上一节 5. jwt集成gin 7. 图形验证码 下一节 →

## 浏览器在什么情况下会发起options预检请求？

在非简单请求且跨域的情况下，浏览器会发起options预检请求。

Preflighted Requests是CORS中一种透明服务器验证机制。预检请求首先需要向另外一个域名的资源发送一个 HTTP OPTIONS 请求头，其目的就是为了判断实际发送的请求是否是安全的。

下面的情况需要进行预检

### 关于简单请求和复杂请求：

#### 1 简单请求

简单请求需满足以下两个条件

1. 请求方法是以下三种方法之一：

- HEAD
- GET
- POST

2. HTTP 的头信息不超出以下几种字段

- Accept
- Accept-Language
- Content-Language
- Last-Event-ID
- Content-Type: 只限于 (application/x-www-form-urlencoded、multipart/form-data、text/plain)

#### 2 复杂请求

非简单请求即是复杂请求

常见的复杂请求有：

1. 请求方法为 PUT 或 DELETE
2. Content-Type 字段类型为 application/json
3. 添加额外的http header 比如access\_token

在跨域的情况下，非简单请求会先发起一次空body的OPTIONS请求，称为"预检"请求，用于向服务器请求权限信息，等预检请求被成功响应后，才发起真正的http请求。

✎ 意见反馈

♡ 收藏教程

🔖 标记书签

浏览器的预检请求结果可以通过设置Access-Control-Max-Age进行缓存

<> 代码块

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <script src="http://libs.baidu.com/jquery/2.0.0/jquery.min.js"></script>
7  </head>
8  <body>
9      <button type="button" id="query">请求数据</button>
10     <div id="content" style="background-color: aquamarine; width: 300px;height: 50px">
11 </div>
12 </body>
13 <script type="text/javascript">
14     $("#query").click(function () {
15         $.ajax(
16             {
17                 url:"http://127.0.0.1:8021/u/v1/user/list",
18                 dataType: "json",
19                 type: "get",
20                 beforeSend: function(request) {
21                     request.setRequestHeader("x-token", "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTUuMi4wLjEiLCJ0eXciOiJ1cm9vdGVudCJ9");
22                 },
23                 success: function (result) {
24                     console.log(result.data);
25                     $("#content").text(result.data)
26                 },
27                 error: function (data) {
28                     alert("请求出错")
29                 }
30             }
31         );
32     });
33 </script>
34 </html>
```