

wangjun5159的专栏

保持探索的心，多尝试，多动手



目录视图

摘要视图

RSS 订阅

个人资料



干净的句号

+ 加关注

发私信



访问：34412次

积分：2172

等级：BLOG > 5

排名：第12329名

原创：178篇

转载：0篇

译文：0篇

评论：21条

文章搜索



文章分类

- javase (18)
- mysql workbench (5)
- mysql (30)
- css (7)
- javascript (7)
- jquery (8)
- mongodb (4)
- mybatis (3)
- spring (8)
- postman (3)
- svn (2)
- swagger (1)
- http (7)
- jsp (6)
- nginx (1)
- tomcat (7)
- eclipse (8)
- html (1)
- apache (1)
- memcache (2)
- regular expression (1)
- json (2)
- maven (6)
- php (1)

原 Access control allow origin 简单请求和复杂请求

标签：跨域

2015-10-13 11:52 1665人阅读 评论(0) 收藏 举报

分类：http (6) javascript (6)

版权声明：本文为博主原创文章，未经博主允许不得转载。

错误信息：

XMLHttpRequest cannot load

http://web.image.myqcloud.com/photos/v2/10008653/bhpocket/0/?sign=4FcLKd5B8...
p4SkFVUEJtZ1omZT0xNDQ0NzExMDE5JnQ9MTQ0NDcwNzQxOSZyPTEzMdMyMDgzOTAmd'
'Access-Control-Allow-Origin' header is present on the requested resource. Origin
'http://localhost' is therefore not allowed access.

分析：

Access control allow origin直译过来就是"访问控制允许同源"，这是由于ajax跨域访问引起的。所谓跨域就是，在a.com域下，访问b.com域下的资源；出于安全的考虑，浏览器允许跨域写，而不允许跨域读，写就是上行，发送请求，send request,读就是下行，接受响应，receive response；理解了这两条规则，以下现象我们也就能理解了；

- 1、表单默认提交（get、post）、超链接访问域外的资源，这是允许的，因为在点击按钮/超链接时，浏览器地址已经变了，这就是一个普通的请求，不存在跨域；
- 2、ajax(借助xmlHttpRequest)跨域请求，这是被禁止的，因为ajax就是为了接受接受响应，这违背了，不允许跨域读的原则
- 3、jsonp属于跨域读且形式限制为GET方式，它利用了script标签的特性；这是允许的。因为浏览器把跨域读脚本，当作例外，类似的img、iframe的src都可以请求域外资源

解决方案：

方案1，给chrome浏览器添加参数,--disable-web-security

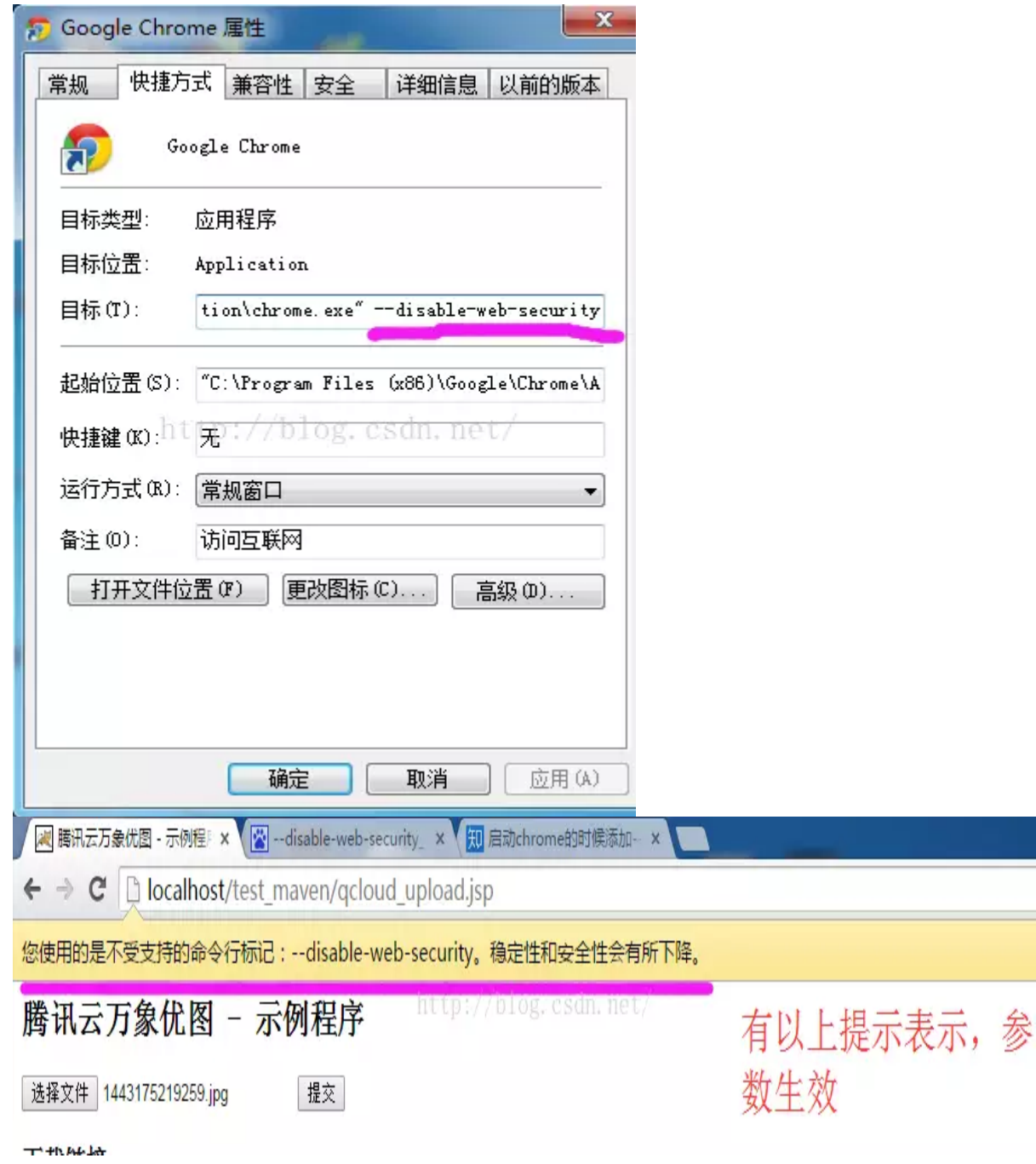
- log4j (1)
- wireshark (4)
- jasperreport (9)
- jenkins (4)
- dubbo (1)
- vim (2)
- linux (3)
- windows (1)
- java nio (2)
- jvm (3)
- charset encoding (6)
- design pattern (1)
- git (5)
- easyui (2)
- android (8)
- sublime (1)
- mycat (1)
- 计算机基础知识 (5)
- 职业感悟 (3)
- genymotion (1)
- opencsv (1)
- shiro (1)
- 微信 (1)

文章存档	
2016年07月	(9)
2016年06月	(11)
2016年05月	(40)
2016年04月	(28)
2016年02月	(12)
展开	

阅读排行	
postman中 form-data 、	(2550)
provisional headers are	(1866)
swagger整合spring mv	(1799)
Access control allow o	(1662)
postman 发送json请求	(927)
multipart/form-data与a	(555)
java 替换反斜杠 Unex	(546)
Tomcat favicon.ico 40	(519)
java对象转json, 数字	(496)
mysql workbench 保存	(470)

评论排行	
swagger整合spring mv	(14)
provisional headers are	(2)
postman中 form-data 、	(2)
mysql char 、varchar	(1)
mongodb 日期类型	(1)
mysql 为什么不能在同	(1)
spring 动态数据源不起	(0)
前台限制默认表单重复	(0)
前台限制ajax重复提交	(0)
java 字符集与编码 unic	(0)

- *Android RocooFix 热修复框架
- * android6.0源码分析之 Camera API2.0下的初始化流程分析



方案2

a, 非IE浏览器，利用xmlHttpRequest，在request header添加origin:本域名地址，通常不需要手动添加，浏览器会自动添加
IE浏览器，利用XDomainRequest发送请求，它会自动封装header中添加origin
b,response header添加access-control-allow-origin:*

跨域演示及代码

验证过程，首先访问http://本机ip:port/project_name/a.jsp,然后，a.jsp发送ajax请求，http://localhost:port/project_name/b.jsp，这样就产生了跨域的问题。

a.jsp

```
[html]
01. <%@ page language="java" contentType="text/html; charset=UTF-8"
02.     pageEncoding="UTF-8"%>
03. <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr">
04. <head>
05. <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
06. <title>跨域演示</title>
07. <script type="text/javascript" src="jquery-1.11.2.js"></script>
08. <script type="text/javascript" src="jquery.form.js"></script>
09. </head>
10.
11. <script type="text/javascript">
12. $(document).ready(function() {
13.
14.     $.getJSON('http://localhost/test_maven/b.jsp', function(data) {
15.         alert("request succeed");
16.     });
17. });
18. </script>
19. </body>
```

b.jsp

*Android_GestureDetector手势滑动使用

*Android MaterialList源码解析

*Android官方开发文档Training系列课程中文版：创建自定义View之View的创建

最新评论

swagger整合spring mvc教程
干净的句号: @qq_35399691:
你的这个问题是版本不一致的问题。swagger-springmvc-1.0...

swagger整合spring mvc教程
qq_35399691: 你好，我引入了swagger-annotations-1.5.9，swagger-models-1....

mongodb 日期类型
zhangleitre: 学习了

postman中 form-data、x-www
干净的句号: @tang745163962:对文件格式没有要求。

postman中 form-data、x-www
tang745163962: 刚有用到，这里，但是传文件内容格式有要求吗？

mysql char、varchar
丿木呈广予口贝: 很好的解释了char和varchar的区别，学习知识就应该从细节做起

mysql 为什么不能在同一个表
wyy0924: 赞！

swagger整合spring mvc教程
干净的句号: @i_Creator:
http://localhost:8080/hi/api-docs，这是js...

swagger整合spring mvc教程
i_Creator: 请问，Can't read swagger JSON from http://localhost:8...

provisional headers are showr
干净的句号: @qq_28158319:
找出后台程序耗费时间很长的代码，改正。

[html] 复制 清除 格式

```
01. <%@ page language="java" contentType="text/html; charset=UTF-8"
02.     pageEncoding="UTF-8"%>
03. <%
04.
05.     response.setHeader("Access-Control-Allow-Origin", "*");
06.     response.getWriter().write("hahahhaha");
07. %>
```

如果注释掉response.setHeader("Access-Control-Allow-Origin", "*");，那么就会出现access control allow origin错误；使用通配符*，允许所有跨域访问，所以跨域访问成功。但是，请注意使用通配符*，会允许来自任意域的跨域请求访问成功，这是比较危险的，所以在生产环境通常会做更精确控制；

简单请求

上述请求是简单请求，只要添加了Access-Control-Allow-Origin：*，就会访问成功，如果是复杂请求，我们需要更进一步的处理，才能成功。这里先解释一下什么是简单请求和复杂请求。

[html] 复制 清除 格式

```
01. Simple requests
02.
03. A simple cross-site request is one that meets all the following conditions:
04.
05. The only allowed methods are:
06. GET
07. HEAD
08. POST
09. Apart from the headers set automatically by the user agent (e.g. Connection, User-
    Agent, etc.), the only headers which are allowed to be manually set are:
10. Accept
11. Accept-Language
12. Content-Language
13. Content-Type
14. The only allowed values for the Content-Type header are:
15. application/x-www-form-urlencoded
16. multipart/form-data
17. text/plain
```

简单来说，符合上述条件的都是“简单请求”，除了简单请求就是“复杂请求”。

复杂请求

在正式post之前，浏览器会先发出一个options请求（也叫preflight），同时header带上origin还有Access-Control-Request-*:**之类的头，服务器响应会返回相应的access-control-allow-origin，如果匹配，那么浏览器就会发送正式post，否则就会出现上述错误。这也解答了，跨域访问时，我们明明发送的post请求，失败的话，查看chrome network会发现是options方法的原因。





根据上述过程，后台方法额外需要options方法，以下是测试样例。





这里的content-type不属于(application/x-www-form-urlencoded，multipart/form-data，text/plain)中的任何一种，所以是复杂请求。

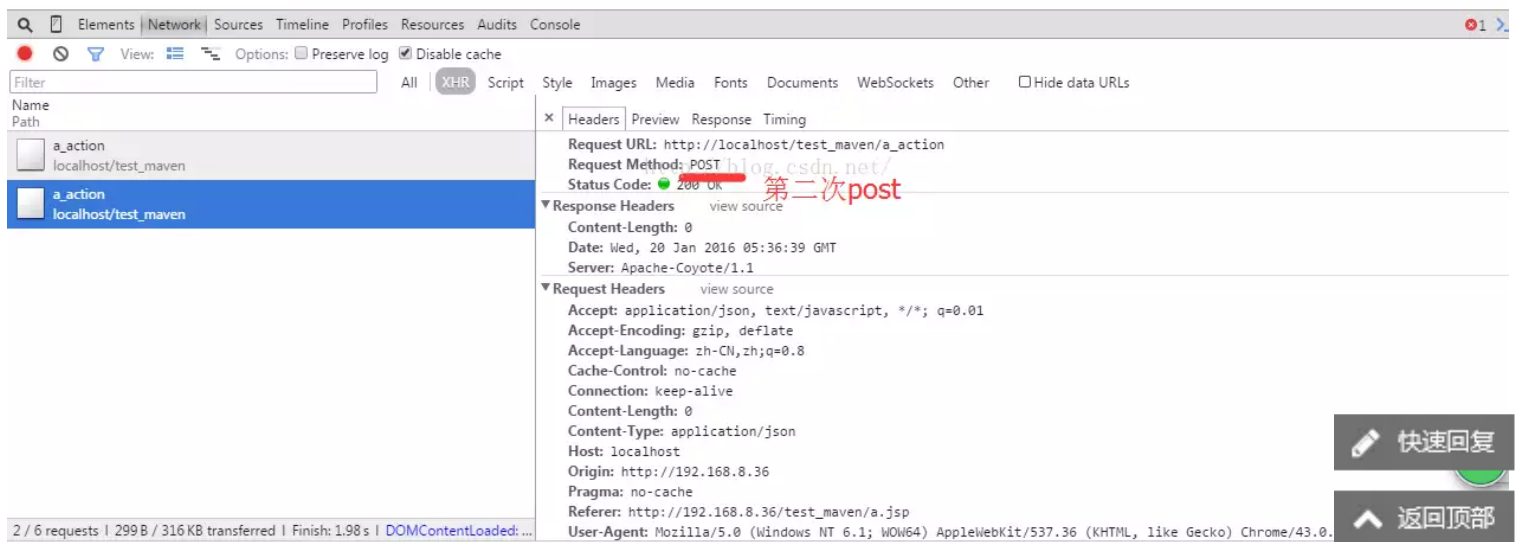
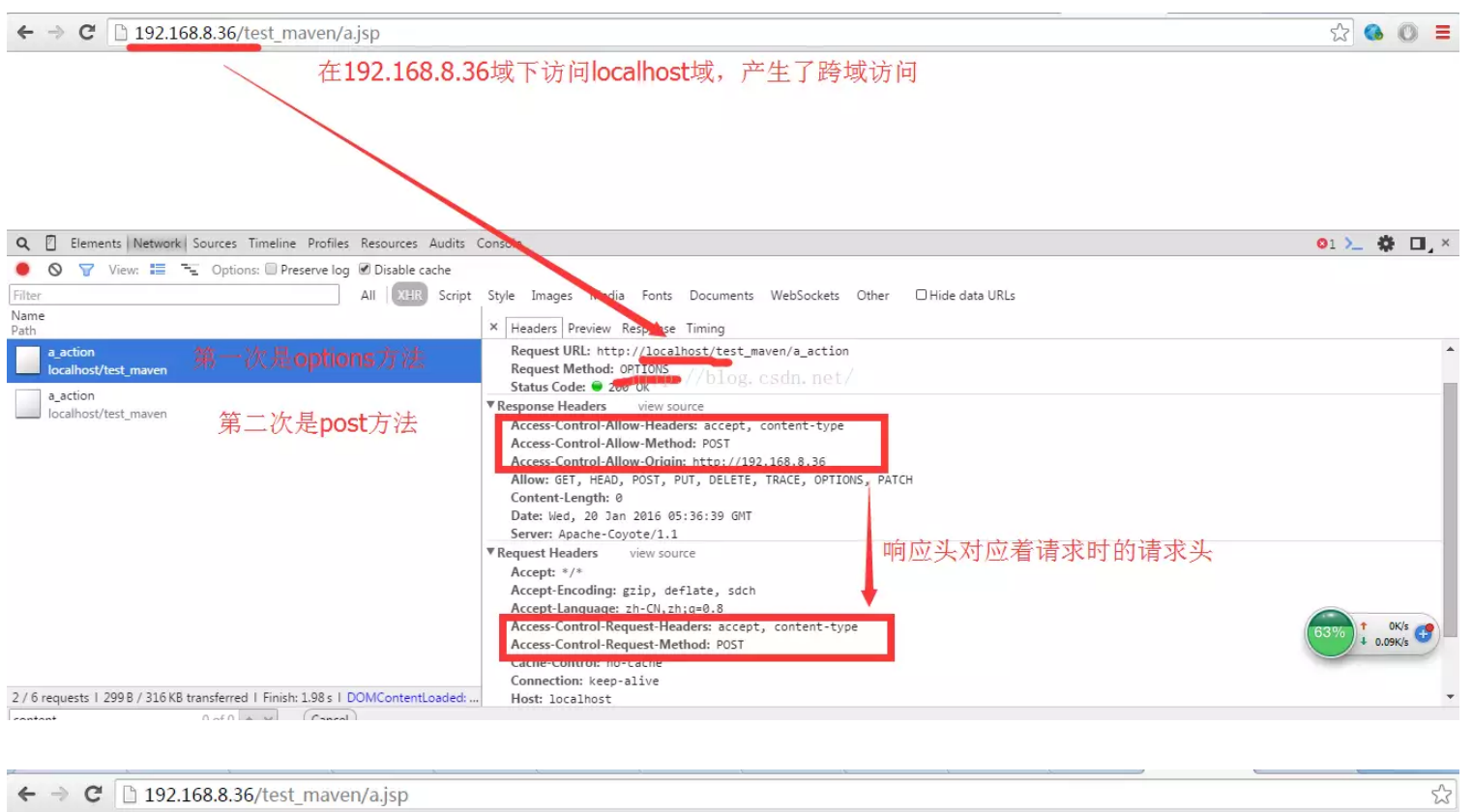
[html] 复制 清除 格式

```
01. $.ajax({
02.
03.     type: "post",
04.     url: "http://localhost/test_maven/a_action",
05.     contentType: "application/json",
06.     dataType: "json",
07.     success: function(){
08.         alert("request succeed");
09.     }
10. });
```



```
[html]      
01. @RequestMapping(value = "/a_action",  
02.     method=RequestMethod.OPTIONS)  
03. public void aActionOption(HttpServletRequest response ) throws IOException{  
04.  
05.     System.out.println("option execute.....");  
06.     response.setHeader("Access-Control-Allow-Headers", "accept, content-type");  
07.     response.setHeader("Access-Control-Allow-Method", "POST");  
08.     response.setHeader("Access-Control-Allow-Origin", "http://192.168.8.36");  
09. }
```

```
[html]      
01. @RequestMapping(value = "/a_action",method=RequestMethod.POST)  
02.     public void aAction(HttpServletRequest response ) throws IOException{  
03.  
04.     System.out.println("a_action execute.....");  
05. }
```



这个方案是利用了w3c的最新规范，所以该方案依赖浏览器的实现。

总结：

出现Access control allow origin错误，说明是跨域请求失败！浏览器发送请求成功，同时浏览器也接收到响应了，但是限制了XmlHttpRequest接收请求，不会让xmlhttprequest接受到响应，并且在js控制台报错。这也就是我们在网络控制台(Network)能看见http 状态码是200，但是在js控制台(Console)出现js错误的原因。

参考：

《白帽子讲web安全》 P28 、 P144 cross origin resource sharing的内容
跨域post请求实现方案小结--转
CORS



顶
1

踩
0

- 上一篇 maven 快照 更新策略
- 下一篇 1GB等于2的多少次方

我的同类文章

http (6)		javascript (6)	
• 微信内嵌浏览器sessionid丢...	2016-07-20 阅读 113	• https 建立连接过程	2016-05-26 阅读 57
• 登陆后 · 地址栏中有sessionid	2016-05-04 阅读 49	• http 301 、 302 、 304 、 400 、 ...	2016-04-25 阅读 242
• multipart/form-data与applica..	2015-11-04 阅读 552	• nginx 代理情况下 jsp reque...	2015-08-27 阅读 436

猜你在找

- ArcGIS for JavaScript

▪ JavaScript与XML

▪ 基于Servlet+JDBC+Bootstrap+MySQL+AJAX权限管理系统

▪ 如何解决ajax跨域问题

▪ Java之路

▪ HTML5 服务器推送事件Server-sent Events实战开发

▪ 快速掌握jQuery视频教程

▪ 跨域的几种方式

▪ 从此不求人:自主研发一套PHP前端开发框架

▪ 前端面试题目搜集

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点 · 不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

快速回复

返回顶部