request中发送json数据用post方式发送Content-type用application/json;charset=utf-8方式发送的话，直接用springMVC的@RequestBody标签接收后面跟实体对象就行了，spring会帮你自动拼装成对象，如果Content-type设置成application/x-www-form-urlencoded;charset=utf-8就不能用spring的东西了，只能以常规的方式获取json串了。

前台代码：

$.ajax({

type:"post",

url:"",

contentType:"application/json;charset=UTF-8",

data:JSON.stringify({name:"jinshihui"}),

})

后台代码：

Jasmine测试框架

为什么会发生Ajax跨域

（1）浏览器限制

（2）跨域

（3）XHR请求

跨域问题是浏览器做的限制，后台是没有要求的。

JSONP去解决

被调用方解决--支持跨域

（1）服务器端实现

（2）Nginx配置

（3）Apache配置

工作中比较常见的简单请求：

方法为：GET HEAD POST

请求header里面：无自定义头

Content-Type：text/plain multipart/form-data application/x-www-form-urlencoded

常见的非简单请求：

put delete 方法的ajax请求

发送json格式的ajax请求

带自定义头的ajax请求

带cookie的跨域问题：

（1）设置Access-Control-Allow-Origin为具体的域名，不能是\*

（2）设置多个域名时，要设置发送方的域名，不能是被调用发的域名

https://blog.csdn.net/brave\_coder/article/details/77103899

# 处理跨域OPTIONS请求方式

跨域请求问题

前提

前后端分离的B/S架构系统

前后端独立开发，后端采用JBoss restEasy 框架搭建restful服务

后端代码开发完成并且部署在某一台测试机上

前端开发过程中，需要本地调试代码，并且后端服务不在本机上而是在测试机上的时候会出现浏览器跨域请求的问题

后端已经添加拦截器并且在拦截器中配置了allowedOrigins, allowedMethods, allowCredentials等参数

解决方式

浏览器在发送跨域请求时候，一般会先发送OPTIONS请求检验请求资源是否可以获取

后端服务需要定义处理options请求方法的接口

因此需要在rest api中定义options处理的接口, 该接口检测并响应@Path(“/{path:.\*}”)的所有请求路径

/\*\*

\* 跨域请求发送options请求时候进行处理

\*

\* @param requestMethods

\* @param requestHeaders

\* @return

\* @throws Exception

\*/

@OPTIONS

@Path("/{path:.\*}")

public Response handleCORSRequest(@HeaderParam("Access-Control-Request-Method") final String requestMethods,

@HeaderParam("Access-Control-Request-Headers") final String requestHeaders) throws Exception {

Response.ResponseBuilder builder = Response.ok();

if (requestHeaders != null)

builder.header("Access-Control-Allow-Headers", requestHeaders);

if (requestMethods != null)

builder.header("Access-Control-Allow-Methods", requestMethods);

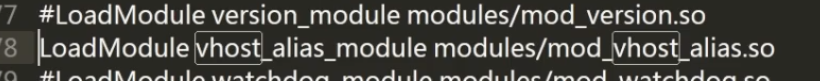
return builder.build();

}

# 被调用方—Apache解决方案

（1）配置虚拟主机

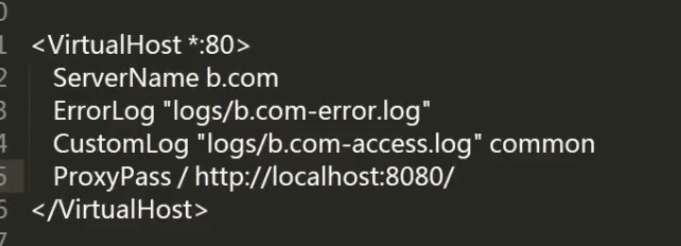
打开服务器下面的conf目录，打开httpd.conf文件，在文件中搜索vhost，把这行前面的注释去掉。



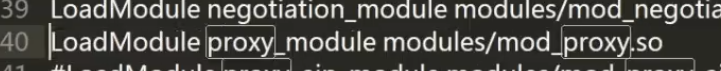
将下面这行的配置注释去掉 

然后在conf/extra/这个目录下面找到虚拟主机对应的配置文件

打开httpd-vhosts.conf这个文件，这个文件中的每个节点就是一个虚拟主机，新增下面的配置信息：



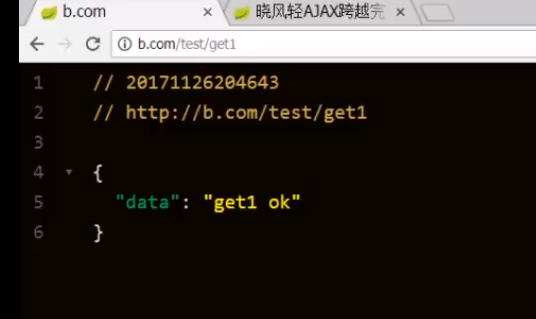
然后我们要在httpd.conf文件中打开proxy模块，把前面的注释去掉。



还要打开proxy\_http模块，把前面的注释去掉。

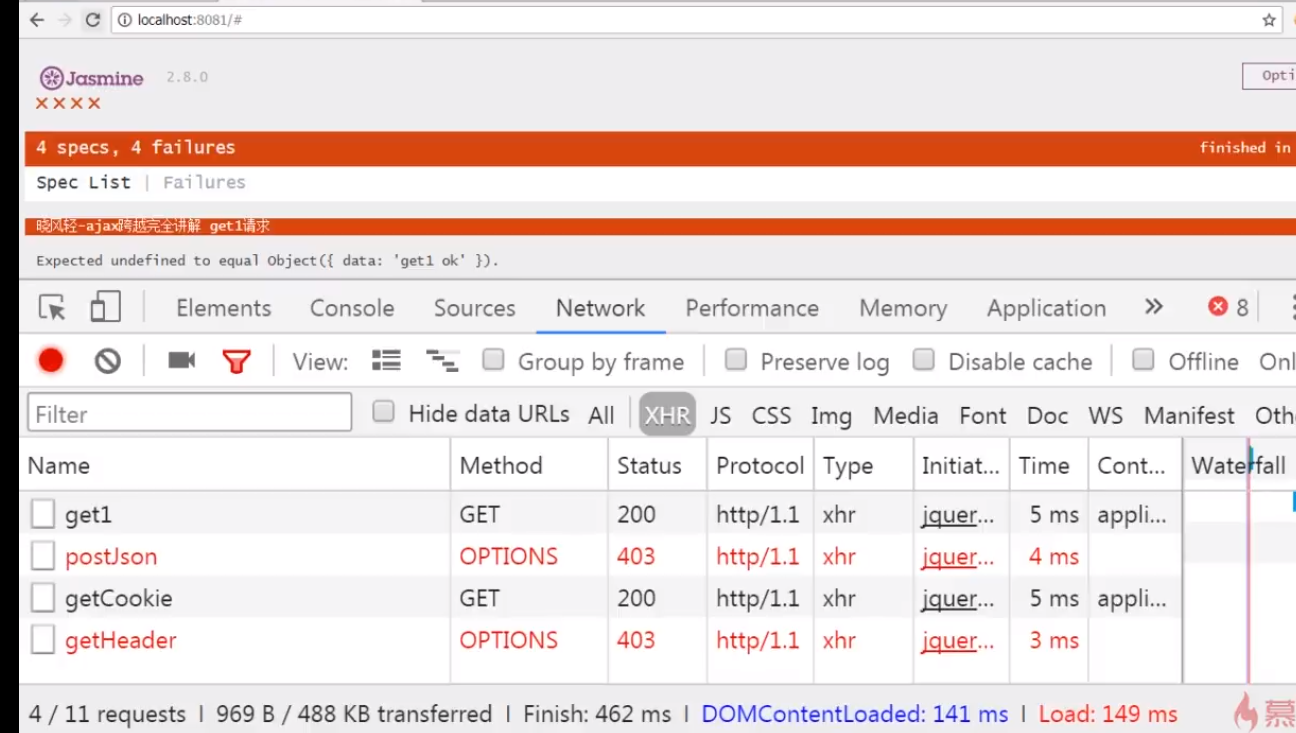


然后我们回到服务器的bin目录，双击httpd.exe文件，启动服务器。

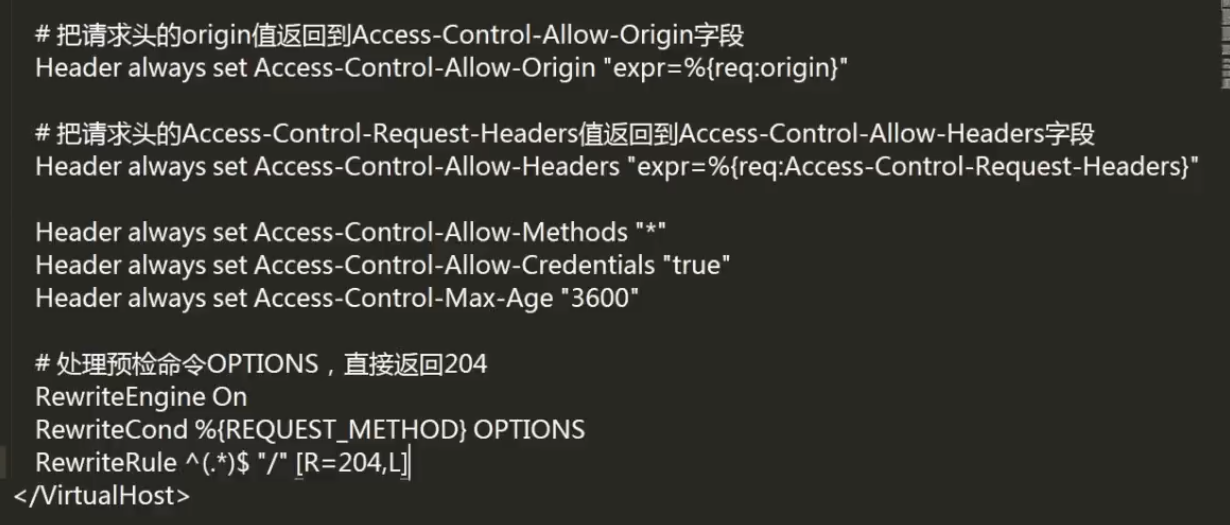


b.com可以正确访问，说明我们的虚拟主机配置成功。

然后我们访问测试用例：



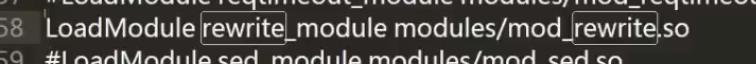
发现访问不成功，然后我们需要在httpd-vhosts.conf文件中增加响应头信息。



然后我们回到httpd.conf文件，把headers模块前面的注释去掉

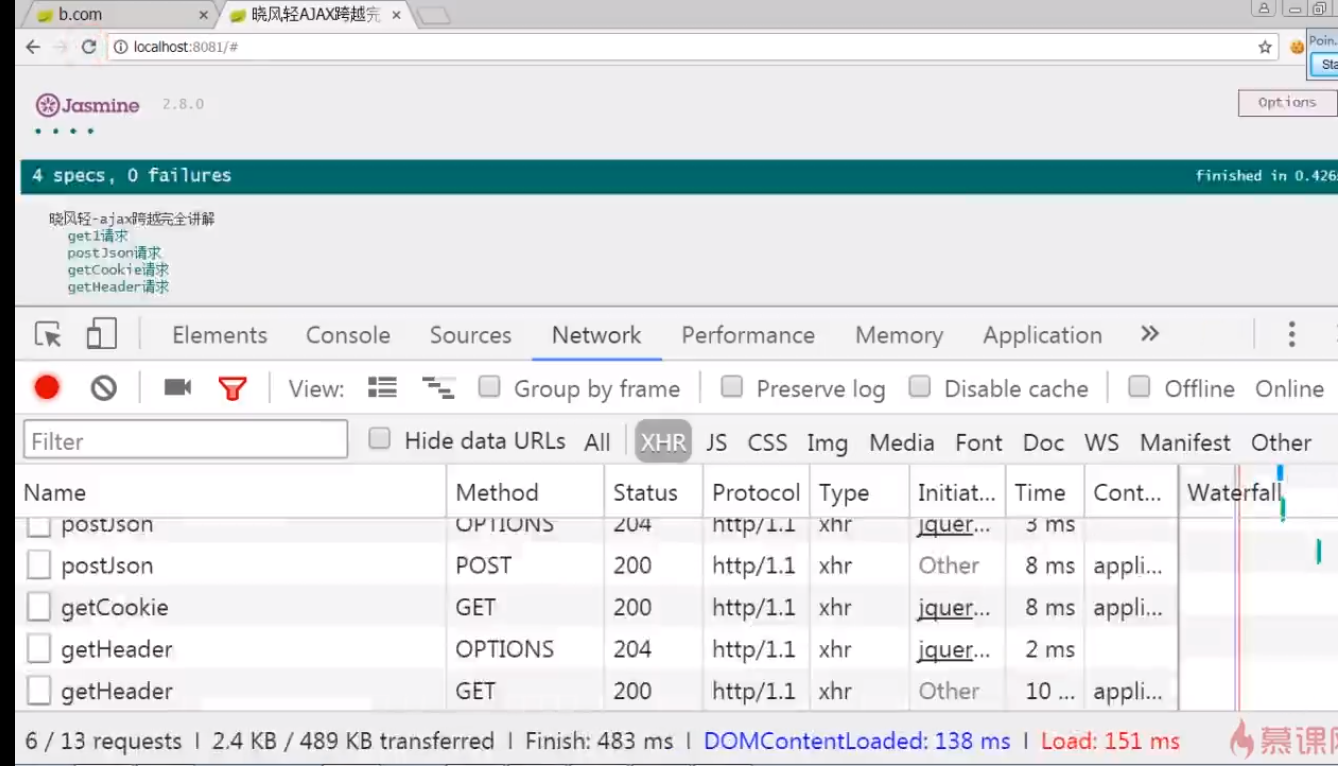


把rewrite模块前面的注释去掉。



然后把这两个配置文件都保存，然后再重启服务器。

再访问我们的测试用例：



说明经过Apache服务器之后，我们的跨域问题解决了。

涉及的配置文件：httpd.conf、httpd-vhosts.conf这两个文件