

05 springboot 解决跨域访问问题

笔记本： 课堂笔记-房屋租赁系统

创建时间： 2020/9/7 22:49

更新时间： 2020/9/7 23:10

作者： yishuihan1104@163.com

1、问题描述

1、对于前后端分离的项目来说，如果前端项目与后端项目部署在两个不同的域下，那么势必会引起跨域问题的出现。针对跨域问题，我们可能第一个想到的解决方案就是jsonp，并且以前处理跨域问题我基本也是这么处理。但是jsonp方式也同样有不足，不管是对于前端还是后端来说，写法与我们平常的ajax写法不同，同样后端也需要作出相应的更改。并且，jsonp方式只能通过get请求方式来传递参数，当然也还有其它的不足之处，jQuery ajax方式以jsonp类型发起跨域请求，其原理跟<script>脚本请求一样，因此使用jsonp时也只能使用GET方式发起跨域请求。跨域请求需要服务端配合，设置callback，才能完成跨域请求。针对于此，我并没有急着使用jsonp的方式来解决跨域问题，去网上找寻其它方式，也就是本文主要所要讲的，在springboot中通过cors协议解决跨域问题。

2、Cors协议H5中的新特性：Cross-Origin Resource Sharing（跨域资源共享）

通过它，我们的开发者（主要指后端开发者）可以决定资源是否能被跨域访问。cors是一个w3c标准，它允许浏览器（目前ie8以下还不能被支持）像我们不同源的服务器发出xmlHttpRequest请求，我们可以继续使用ajax进行请求访问。

3、springboot中解决方案

3.1 方式1 创建一个filter解决跨域

```
package com.yanzhen.cors;

import org.springframework.stereotype.Component;
```

```

import javax.servlet.*;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

@Component
public class SimpleCORSFilter implements Filter {

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {

    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response,
        FilterChain chain) throws IOException, ServletException {
        HttpServletResponse response2= (HttpServletResponse) response;
        response2.setHeader("Access-Control-Allow-Origin","*");

        response2.setHeader("Access-Control-Allow-Methods", "POST, GET,
OPTIONS, DELETE, HEAD");

        response2.setHeader("Access-Control-Max-Age", "3600");

        response2.setHeader("Access-Control-Allow-Headers", "access-control-
allow-origin, authority, content-type, version-info, X-Requested-With");

        chain.doFilter(request, response);

    }

    @Override
    public void destroy() {

    }

}

```

3.2 方式2 基于WebMvcConfigurerAdapter配置加入Cors的跨域

```

package com.yanzhen.cors;

import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.CorsRegistry;
import
org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

/**
 * 创建全局跨域配置类
 */
@Configuration
public class CorsConfig extends WebMvcConfigurerAdapter {

    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping("/*")
            .allowedOrigins("*")
            .allowCredentials(true)
            .allowedMethods("GET", "POST", "PUT", "DELETE")
            .maxAge(3600);
    }
}

```

如果你想做到更细致也可以使用@CrossOrigin这个注解在controller类中使用。

这样就可以指定该controller中所有方法都能处理来自http:19.168.1.10:8080中的请求。

第一种Filter的方案也支持springmvc。

第二种常用于springboot。