

# 跨域请求

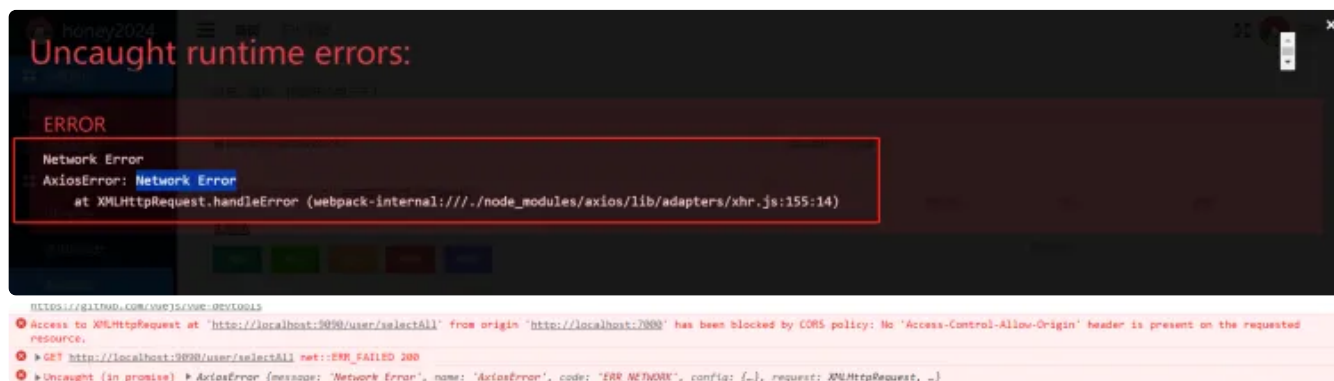
方案 1

方案 2 (推荐)

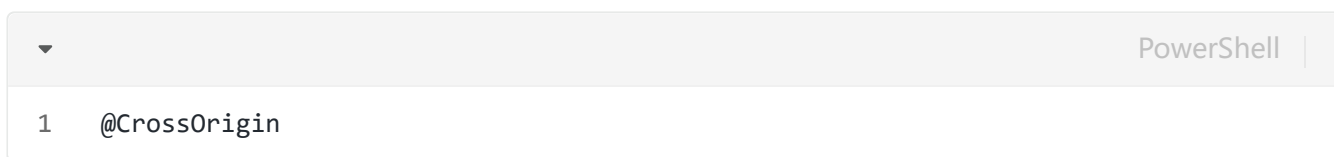
SpringBoot 解决跨域问题 CorsConfig

## 方案 1

这个错误是因为前后端的端口不一致，导致了跨域请求



在你的 UserController 的上面加个小注解

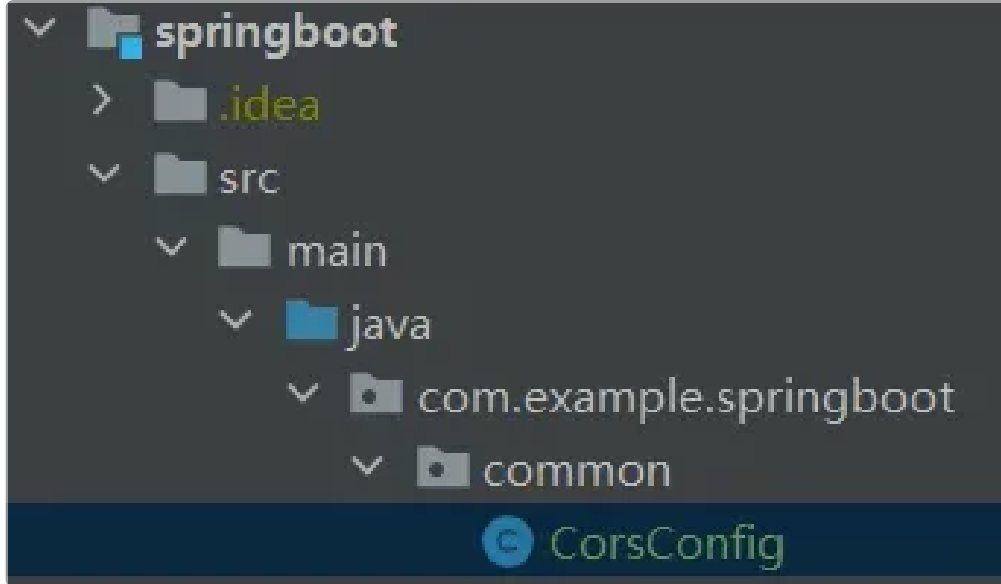


但是这个操作比较麻烦的地方在于每一个 controller 层都需要添加一个这个注释。

## 方案 2 (推荐)

我们通过一个配置文件解决这个问题：

## SpringBoot 解决跨域问题 CorsConfig



Java

```
1  import org.springframework.context.annotation.Bean;
2  import org.springframework.context.annotation.Configuration;
3  import org.springframework.web.cors.CorsConfiguration;
4  import org.springframework.web.cors.UrlBasedCorsConfigurationSource;
5  import org.springframework.web.filter.CorsFilter;
6
7  @Configuration
8  public class CorsConfig {
9
10     // 当前跨域请求最大有效时长。这里默认1天
11     private static final long MAX_AGE = 24 * 60 * 60;
12
13     @Bean
14     public CorsFilter corsFilter() {
15         UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSou
16         rce();
17         CorsConfiguration corsConfiguration = new CorsConfiguration();
18         corsConfiguration.addAllowedOrigin("*"); // 1 设置访问源地址
19         corsConfiguration.addAllowedHeader("*"); // 2 设置访问源请求头
20         corsConfiguration.addAllowedMethod("*"); // 3 设置访问源请求方法
21         corsConfiguration.setMaxAge(MAX_AGE);
22         source.registerCorsConfiguration("/**", corsConfiguration); // 4 对接口配
23         置跨域设置
24         return new CorsFilter(source);
25     }
26 }
```