



链滴

优雅的解决跨域问题

作者: [zouchanglin](#)

原文链接: <https://ld246.com/article/1580189110400>

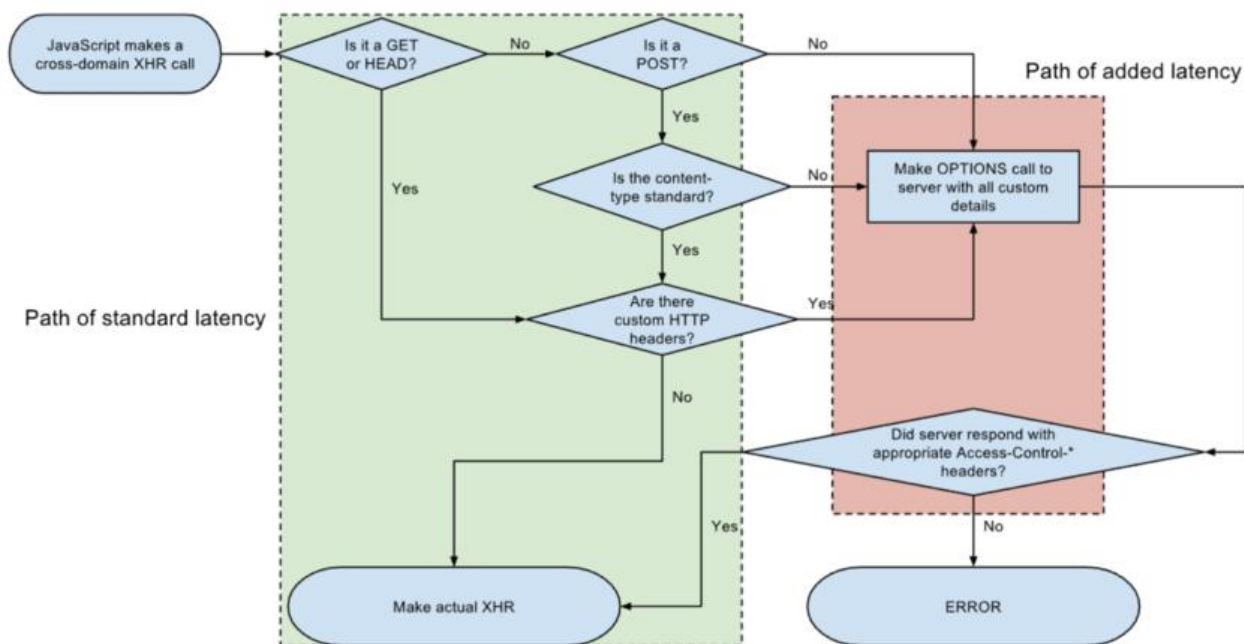
来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



0、浏览器跨域问题

看到浏览器报Access-Control-Allow-Origin毫无疑问了，看来是跨域问题



CORS是一种访问机制，英文全称是Cross-Origin Resource Sharing，即我们常说的跨域资源共享，过在服务器端设置响应头，把发起跨域的原始域名添加到Access-Control-Allow-Origin 即可

浏览器端：发送AJAX请求前需设置通信对象XHR的withCredentials 属性为true。

服务器端：设置Access-Control-Allow-Credentials为true。

两个条件缺一不可，否则即使服务器同意发送Cookie！

Access-Control-Allow-Methods 表示允许哪些跨域请求的提交方式。（例如GET/POST）

Access-Control-Allow-Headers 表示跨域请求的头部的允许范围。

Access-Control-Expose-Headers 表示允许暴露哪些头部信息给客户端。

Access-Control-Max-Age 表示预检请求的最大缓存时间。

1、返回新的CorsFilter

这是一种全局跨域的方式

在任意配置类，返回一个新的CorsFilter Bean，并添加映射路径和具体的CORS配置信息。

```
package edu.xpu.cors.config;
```

```
@Configuration
```

```
public class GlobalCorsConfig {
```

```
    @Bean
```

```
    public CorsFilter corsFilter() {
```

```
        //1.添加CORS配置信息
```

```
        CorsConfiguration config = new CorsConfiguration();
```

```
        //放行哪些原始域
```

```
        config.addAllowedOrigin("*");
```

```
        //是否发送Cookie信息
```

```
        config.setAllowCredentials(true);
```

```
        //放行哪些原始域(请求方式)
```

```
        config.addAllowedMethod("*");
```

```
        //放行哪些原始域(头部信息)
```

```
        config.addAllowedHeader("*");
```

```
        //暴露哪些头部信息（因为跨域访问默认不能获取全部头部信息）
```

```
        config.addExposedHeader("*");
```

```
        //2.添加映射路径
```

```
        UrlBasedCorsConfigurationSource configSource = new UrlBasedCorsConfigurationSource
```

```
    );
```

```
        configSource.registerCorsConfiguration("/**", config);
```

```
        //3.返回新的CorsFilter.
```

```
        return new CorsFilter(configSource);
```

```
    }
```

```
}
```

2、重写WebMvcConfigurer

在任意配置类，返回一个新的WebMvcConfigurer Bean，并重写其提供的跨域请求处理的接口，目的是添加映射路径和具体的CORS配置信息。

```
package edu.xpu.cors.config;
```

```
@Configuration
```

```
public class GlobalCorsConfig {
```

```
    @Bean
```

```
public WebMvcConfigurer corsConfigurer() {  
    return new WebMvcConfigurer() {  
        @Override  
        //重写父类提供的跨域请求处理的接口  
        public void addCorsMappings(CorsRegistry registry) {  
            //添加映射路径  
            registry.addMapping("/**")  
                //放行哪些原始域  
                .allowedOrigins("**")  
                //是否发送Cookie信息  
                .allowCredentials(true)  
                //放行哪些原始域(请求方式)  
                .allowedMethods("GET", "POST", "PUT", "DELETE")  
                //放行哪些原始域(头部信息)  
                .allowedHeaders("**")  
                //暴露哪些头部信息（因为跨域访问默认不能获取全部头部信息）  
                .exposedHeaders("Header1", "Header2");  
        }  
    };  
}
```