1 前端请求，后台设置允许跨域

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

@Component
public class CorsFilter implements Filter {

    final static org.slf4j.Logger logger = org.slf4j.LoggerFactory.getLogger(CorsFilter.class);



    public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain) throws IOExcep
        HttpServletResponse response = (HttpServletResponse) res;

        HttpServletRequest reqs = (HttpServletRequest) req;

        response.setHeader("Access-Control-Allow-Origin",reqs.getHeader("Origin"));
        response.setHeader("Access-Control-Allow-Credentials", "true");
        response.setHeader("Access-Control-Allow-Methods", "POST, GET, OPTIONS, DELETE");
        response.setHeader("Access-Control-Max-Age", "3600");
        response.setHeader("Access-Control-Allow-Headers", "x-requested-with");

        chain.doFilter(req, res);
    }
    public void init(FilterConfig filterConfig) {}
    public void destroy() {}
}
```

注意：

```
response.setHeader("Access-Control-Allow-Headers", "x-requested-with");
```

在与前端的对接中，他在header里面带入了其他信息，结果还是跨域

前请求

```
Access-Control-Request-Headers: authorization
```

响应头

```
Access-Control-Allow-Headers: X-Requested-With
```

出现异常

```
Request header field Content-Type is not allowed by Access-Control-Allow-Headers in preflight respon
```

解决方式：

```
response.setHeader("Access-Control-Allow-Headers", "x-requested-with,authorization");
```

在后端代码加入 请求头里面的方式即可

2 前端设置的token后台取不到

前端请求头

```

响应

Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: X-Requested-With,Authorization
Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE
Access-Control-Allow-Origin: http://localhost:8080
Access-Control-Max-Age: 3600
Connection: close
Content-Language: en
Content-Length: 820
Content-Type: text/html;charset=utf-8
Date: Wed, 27 Mar 2019 00:54:09 GMT

请求

Access-Control-Request-Headers: authorization
Access-Control-Request-Method: GET
Origin: http://localhost:8080
Referer: http://localhost:8080/
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
```

采取的下面一种方式解决，不用CorsFilter

第二种方式

```
@Configuration
public class CorsConfig implements WebMvcConfigurer {
    @Override
    public void addCorsMappings(CorsRegistry registry) {
        //设置允许跨域的路径
        registry.addMapping("/**")
                //设置允许跨域请求的域名
                .allowedOrigins("*")
                //是否允许证书 不再默认开启
                 .allowCredentials(true)
                 //设置允许的方法
                 .allowedMethods("*")
                 .allowedHeaders("*")
                 //跨域允许时间
                 .maxAge(3600);
    }
}
```