https://www.cnblogs.com/lgjlife/p/10445483.html

[**解决Ajax请求时无法重定向的问题**](https://www.cnblogs.com/lgjlife/p/10445483.html)

今天发现，当使用Ajax请求时，如果后台进行重定向到其他页面时是无法成功的，只能在浏览器地址栏输入才能够实现重定向。

Ajax默认就是不支持重定向的，它是局部刷新，不重新加载页面。

需要实现的功能是，后台网关拦截请求，看请求中是否存在token.如果不存在就跳转到登录页面。因为大多数请求都是使用Ajax.一开始发现无法进行重定向，每次都是返回到Ajax的结果处理函数。最终的解决办法如下，需要后台和前端进行处理。

**后台：**

[复制代码](javascript:void(0);)

/\*\*

\*功能描述

\* @author lgj

\* @Description 重定向工具类

\* @date 2/27/19

\*/

@Slf4j

public class RedirecUtil {

/\*\*

\*功能描述

\* @author lgj

\* @Description 重定向

\* @date 2/27/19

\* @param:

\* @return:

\*

\*/

public static void redirect(RequestContext ctx, String redirectUrl){

try{

//如果是Ajax请求

if("XMLHttpRequest".equals(ctx.getRequest().getHeader("X-Requested-With"))){

log.debug("ajax redirect");

sendRedirect(ctx.getResponse(),redirectUrl);

}

//如果是浏览器地址栏请求

else {

log.debug("normal redirect ");

ctx.getResponse().sendRedirect(redirectUrl);

}

}

catch(Exception ex){

ex.printStackTrace();

}

}

/\*\*

\*功能描述

\* @author lgj

\* @Description Ajax请求时重定向处理

\* @date 2/27/19

\* @param:

\* @return:

\*

\*/

private static void sendRedirect(HttpServletResponse response, String redirectUrl){

try {  
　　　　　　　//这里并不是设置跳转页面，而是将重定向的地址发给前端，让前端执行重定向

//设置跳转地址

response.setHeader("redirectUrl", redirectUrl);

//设置跳转使能

response.setHeader("enableRedirect","true");

response.flushBuffer();

} catch (IOException ex) {

log.error("Could not redirect to: " + redirectUrl, ex);

}

}

}

[复制代码](javascript:void(0);)

**前端处理**

***第一种方式：使用Ajax的complete方法***

[复制代码](javascript:void(0);)

$.ajax({

type: "post",

url: "/auth/token/check",

success: function(data,status){

console.log("/token/check 返回 status : "+status)

},  
 //请求完成调用

(XHR, TS){

console.log("complete");

var url = XHR.getResponseHeader("redirectUrl");

console.log("redirectUrl = " + url);

var enable = XHR.getResponseHeader("enaleRedirect");

console.log("enableRedirect = " + enable);

if((enable == "true") && (url != "")){   
  
 var win = window;   
   
 　　　　　　　　 while(win != win.top){  
 　　　　　　win = win.top;  
 　　　　　　　　}  
 　　　　　　　　 win.location.href = url;   
　　　　　　　　　　}   
　　　　　　},   
　　　　});  
 })

[复制代码](javascript:void(0);)

但是上面有个问题就是，每个ajax都需要编写 comlete 方法，代码复用率低。

***第二种方法 ： 使用全局的complete方法***

*ajax请求*

[复制代码](javascript:void(0);)

$("#NON-TOKEN").click(function () {

$.ajax({

type: "post",

url: "/auth/token/check",

success: function(data,status){

console.log("/token/check 返回 status : "+status)

},

});

[复制代码](javascript:void(0);)

全局处理

注意这参数*(event, xhr, settings)*不能少，否则会报错。

//每一个Ajax 请求完成之后都会执行。  
$(document).ajaxComplete(function (event, xhr, settings) {

console.log("ajaxComplete ")

redirectHandle(xhr);

})

[复制代码](javascript:void(0);)

function redirectHandle(xhr) {

　　//获取后台返回的参数

var url = xhr.getResponseHeader("redirectUrl");

console.log("redirectUrl = " + url);

var enable = xhr.getResponseHeader("enableRedirect");

if((enable == "true") && (url != "")){

var win = window;

while(win != win.top){

win = win.top;

}

win.location.href = url;

}  
}

[复制代码](javascript:void(0);)

**也可以将上述前端代码放入一个js文件中，在需要进行重定向的时候引入即可，便可以实现更高的代码复用率。**

**redirect.js**

[复制代码](javascript:void(0);)

function redirectHandle(xhr) {

var url = xhr.getResponseHeader("redirectUrl");

console.log("redirectUrl = " + url);

var enable = xhr.getResponseHeader("enableRedirect");

if((enable == "true") && (url != "")){

var win = window;

while(win != win.top){

win = win.top;

}

win.location.href = url;

}

}

$(function () {

$(document).ajaxComplete(function (event, xhr, settings) {

console.log("ajaxComplete adffdafadsaf")

redirectHandle(xhr);

})

})

[复制代码](javascript:void(0);)

引入js文件，src根据据实际情况设置。

<script src="/common/redirect.js"></script>

=======本博客代码一般存放在GitHub,使用过程中代码链接路径可能会发生变化，如无法跳转，请联系我进行修改！=============   
[推荐:《Java常用技术和书籍推荐》](https://github.com/lgjlife/Java-Book) 

如果，您认为阅读这篇博客让您有些收获，不妨点击一下右下角的推荐按钮。   
如果，您希望更容易地发现我的新博客，不妨关注一下。因为，我的写作热情也离不开您的肯定支持。   
感谢您的阅读，如果您对我的博客所讲述的内容有兴趣，请继续关注我的后续博客。