

基于 LINCMS 和 UNIAPP 的企业微信应用开发

龙滨

民航重庆空管分局

DOI:10.12238/acair.v2i4.10745

[摘 要] LINCMS是国内林间有风团队开发的一个开源CMS框架,包含适合于PC的基于VUE的管理前端和基于KOA的后端。由于其管理前端不适合于企业微信,笔者设计一种基于UNIAPP的前端同LINCMS后端进行安全通信的方法,并采用H5方式实现企业微信应用。该方法也可应用于其他架构的后端系统。

[关键词] LINCMS; UNIAPP; 企业微信; VUE; KOA; access token; 令牌; 安全

中图分类号: F271 文献标识码: A

Enterprise wechat application development based on LINCMS and UNIAPP

Bin Long

Chongqing Air Traffic Control Branch of CAAC

[Abstract] LINCMS is an open source CMS framework developed by the domestic forest wind team, including VUE based management front end suitable for PC and back end based on KOA. Because its management front-end is not suitable for enterprise wechat, the author designs a UNIAPP-based secure communication method between the front-end and the LINCMS back-end, and adopts H5 mode to realize the enterprise wechat application. The method can also be applied to the back-end systems of other architectures.

[Key words] LINCMS; UNIAPP; enterprise wechat; VUE; KOA; access token; token; security

LINCMS是国内林间有风团队开发的一个开源CMS框架,其后端采用基于KOA框架,前端采用VUE3框架,由于整体较为轻量、界面美观,容易二次开发。UNIAPP是基于VUE前端框架,编写一套代码可发布到多个平台,在国内应用非常广泛。由于我单位有相关管理平台接入企业微信的需求,笔者经长期研究,实现在LINCMS作为后端的基础上,采用UNIAPP企业微信手机前端的需求。

1 核心需求

企业微信APP的核心需求有三个。一是用户无感进行企业微信验证、无感登录到应用。

二是建立良好的数据验证机制,保证数据传输的安全性。三是保留LINCMS原设计基础,不对系统做大的改动。

2 企业微信开发方式选择

目前很多公司采用企业微信进行公司或企业内部管理,企业微信应用的开发包括内部应用开发和第三方应用开发,企业微信应用有小程序方式或者H5网页方式。由于我们计划采用内部应用开发,且单位有自己的服务器,因此考虑采用H5网页方式。小程序开发在用户接口方面同H5网页开发稍有区别,整体原理类似。

3 用户登录体系解决方案

由于LINCMS和企业微信各有自己的用户体系,如果不使用

LINCMS的用户体系,而全部使用企业微信的用户体系,对系统的改动难度太大,因为涉及到权限、组织机构等相应改动,不亚于重新开发一套系统,为此保留LINCMS原有系统的用户体系是非常必要的。

为此,我们考虑采用一种中间方式:

3.1利用企业微信账户体系进行无感用户登录的自动合法性验证,以代替LINCMS的用户名密码登录,简化用户登录流程。

3.2用户登录完成后,则利用LINCMS原有登录验证机制(JWT的access token)进行登录验证和数据包验证。由于用户不直接向企业微信收发数据,保证了企业微信应用的安全性,避免了企业微信的密钥和应用的密钥向用户端泄露的风险。

这里还有一个问题,如何实现企业微信用户同LINCMS用户的关联?我们可以将企业用户的id号和LINCMS的用户进行关联,也就是说,我们可以在LINCMS的user表中增加一个字段,如:wework_user_id,使得每个用户有一个唯一的同企业微信用户id对应的字段。这样,就可以建立起两个用户体系的关联。这样,我们在采用企业微信进行用户验证后,企业微信会返回一个user_id,根据这个user_id,我们就可以在LINCMS系统中获取用户的信息(代替LINCMS用户名、密码方式获取用户信息),并利用LINCMS原有的access token授权机制进行通信。

可见,我们只用企业微信的用户系统来进行用户验证,不读取企业微信的用户等信息,应用中需要用户、组织机构等信息则需要通过LINCMS自带的数据库来获取。同时,我们在用户客户端不存储同企业微信通信的相关信息(如user_ticket等),而是在后端进行存储记录,以保证数据的安全性。

4 用户登录的具体流程设计

参考企业微信流程,并考虑数据安全性,我们设计出UNIAPP客户端登录的流程如下:

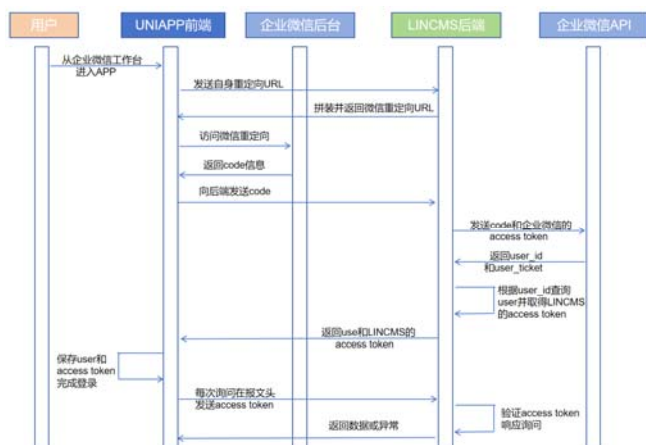


图1 UNIAPP用户端登录流程示意图

整个流程主要包括以下步骤:

(1) 用户从企业微信的工作台中点击APP,进入应用的登录页;

(2) 在应用的登录页中,首先拼装一个用户APP本身的重定向链接,然后用户先向自己的后端服务器发送一个重定向链接请求,后端将APP重定向链接再次拼装到向微信后台发送的重定向链接中(为什么不直接在客户端拼装该链接呢,这是因为该链接中包含企业的corp_id和该企业登录企业微信的access token信息,考虑信息安全,上述信息不应放到用户前端),后端服务器将该拼装后的重定向链接发送给用户客户端。

该登陆页生成如下app重定向链接:

<http://localhost:5173/#/pages/index/oauthCallback>

后端返回的微信重定向链接示例如下:

https://open.weixin.qq.com/connect/oauth2/authorize?appid=CORPID&redirect_uri=OAUTH_CALLBACK&response_type=code&scope=snsapi_base&state=STATE&agentid=AGENTID#wechat_redirect

其中企业id(即appid)、agentid(应用id)需要在后台提前设置。

(1) 用户登录页获取企业微信重定向链接后,执行在用户前端index页面的window.location.href=企业微信重定向链接这句代码,执行该语句后,由于是访问企业微信的https页面,企业微信服务器经内部处理,返回一个前面用户前端发出的app的重定向并提供了code参数。

企业微信返回的重定向地址示例如下:

<https://localhost:5173/#/pages/index/oauthCallback>

?code=CODE&state=STATE

可以看到,返回的重定向地址中,包含一个code信息。需要说明的是,企业微信通过该code,可以判断该用户的身份,比如可以获取用户的个人信息等(具体企业微信采用什么实现方法这一点,企业微信官方未予说明)。另外,STATE是用户端后端发送的自己的状态信息,我们可以设置为LOGIN,表示登录状态。

用户前端收到code后,就可以进行用户身份的验证。我们设置一个callback.vue页面,在该页面启动时,从微信返回的回调地址中获取code信息,并将该code发送给用户后端接口,用来获取用户信息和用户后端的access token,注意这个access token是lincms用于授权验证的access token。在获取到access token后,前端每次在向后端发送数据包时,会将access token放到文件头中进行发送,后端收到该数据包后,会access token进行验证,如果验证通过,则响应前端需要的数据,否则向前端返回验证错误信息。

在上述过程中,后端收到code后,需要向企业微信服务器发送验证信息,验证该用户的身份,企业微信服务器会返回一个包含user_id和user_ticket的信息,后端收到后,根据user_id,可以在用户数据表中根据该用户id获取到该用户的必要信息。另外,后台也要向前端返回一个access token信息。由于LINCMS采用JWT验证,其access token是通过userid信息来进行计算的。因此,我们可以根据该user_id来生成一个access token,并将其返回给前端。

后端获取code后,向企业微信后台发送code和该企业微信的access token,并获取到user_id和user_ticket。根据该user_id,LINCMS用户数据表中获得该用户信息,并返回给前端,需要说明的是,ctx.app.context.access_token是企业微信的access_token,而不是LINCMS应用的access token,LINCMS的access token是根据user_id调用jwt工具内部函数获取的。

企业微信的access token是企业微信同用户后端通信的许可凭据,通常7200秒会过期。由于我们需要企业id(corp_id)和agentSecret(秘钥)才能获取企业微信的许可令牌access token。为实现无感获取该令牌,我们每次后台在获取企业微信access token的时候,先将当前时间和记录的上次更新的access token时间对比,如果时间超时,则重新获取令牌,并记录令牌获取的时间。

5 构建前端http请求工具request

在LINCMS的管理后台的前端,采用改进的axios工具同后端进行http通信,基本原理是对axios进行http请求和响应进行拦截。在http请求中,如果前端登录时已经向后端取得了合法令牌access_token,则每次都在请求报文头中加入access_token,发送给后端,后端收到前端的请求后,通过jwt工具对报文头中的access_token进行验证,如果验证合法,则向前端返回所需要的信息;如验证不合法,则返回异常报文。前端对后端返回的报文进行响应拦截,检查接收报文状态码,如果状态正常,则正常处置;如果状态异常,则在前端界面提示异常信息。

由于UNIAPP采用uni.request()函数进行通信,因此我们需要对其进行改造,利用uni.request()函数建立一个request工具,专门进行同后端的http通信。建立后,我们可以用request.get(url, option)等方法同后台进行通信。

下面建立request工具:

```
export default {
  config: {
    beforeRequest(originConfig = {}) { //请求拦截
      ...
      if(accessToken) { //如果已从后端获取令牌,则写入报头
        reqConfig.header={authorization:accessToken}
      }
      ...
    },
    responseRequest(res) { //响应拦截
      ...
    }
  },
  request(options) {
    return this.config.beforeRequest(options) //执行请求拦截
      .then(opt=>{return uni.request(opt)}) //用uni.request()发送请求
      .then(this.config.responseRequest) //响应拦截
      .then(data=>{return data}) //响应拦截后对数据进行处理
  },
  post(url, params=null, options={}) {
    ...
  },
  ...
}
```

通过上述代码,可以用类似下述方式发送post或其他请求:

```
request.post('https://localhost:5000/v1/wework/loginByWeworkCode', data)
```

其中, beforeRequest() 请求拦截的在系统已收到后端的合法令牌时,则将access token加入每个请求报头。另外,我们需要进行用于登录超时处置,如果UNIAPP前端用户一段时间未向后端发送请求(长时间不登录),则在请求中清除access_token,使得同后台不能成功通信。

在responseRequest() 函数中,我们需要根据拦截返回的

HTTP状态码进行检查。如果是正常信息(如: 状态码为2xx),则向前端返回拦截的数据;否则是后端返回了错误信息码,检查错误信息代码如果是指定代码(如10000等值),则表示access token令牌异常,则执行退出登录相关操作,在前端显示令牌异常信息,并将前端切换到登录页面。

通过上述处理,我们在前端实现了同后端进行通信的request方法。由于在用户成功登录后,都会在报头附带access token,后端通过对其进行校验,认为该报头是合法的,因此可以正常响应。如果未成功登录,就不会有access token,后端就会返回错误信息,前端收到后,将错误信息通过uni.showToast()进行在界面的显示。

6 对LINCMS的其他改造

另外,LINCMS本身是不带组织结构的。但在企业微信的应用中,往往是需要进行组织结构的,因此需要对其进行改造,在LINCMS中构建组织结构相应前后台。

7 对前端持久化存储的处置

我们可以参照LINCMS管理前端的设计,基于vuex-persist构建起持久化存储。在UNIAPP中,我们需要记录当前用户的个人信息、登录状态、访问权限等,都可用通过LINCMS管理前端的设计进行改造。

沿用LINCMS的权限管理机制也是非常重要的,这样可以在UNIAPP中限制个人的访问权限,防止超权限的访问,提升系统的安全性,同时也减少对系统的改动。

8 总结

LINCMS在系统设计上本身适合于基于PC的管理平台,设计上并没有针对微信或企业微信用户进行定制。为此,笔者提出了一种设计思路,可以实现采用UNIAPP前端同LINCMS后端进行无感登录访问。这种方法不仅对于LINCMS适用,同时也适用于其他的CMS系统,比如springboot后端等架构的系统。

[参考文献]

- [1]隋阳,丁睿,王汉青.运行核电厂安全生产标准化管理信息系统的研发[J].核动力工程,2018,(4):152-156.
- [2]李娟.基于Vue+Node的高职院校学生成绩管理系统设计与实现[J].现代信息科技,2023,7(09):115-117.
- [3]胡森,谢华,李俊.高校企业微信多身份难题的解决与实践[J].现代信息科技,2023(7):2.

作者简介:

龙滨(1976--),男,汉族,重庆市巫溪县人,大学本科(研究生学位),职称:高级工程师,研究方向:空中交通管理通信、导航、监视。