



WEEX Conf

WEEX 在极客时间 App 中的实践

孙涛（大猫）

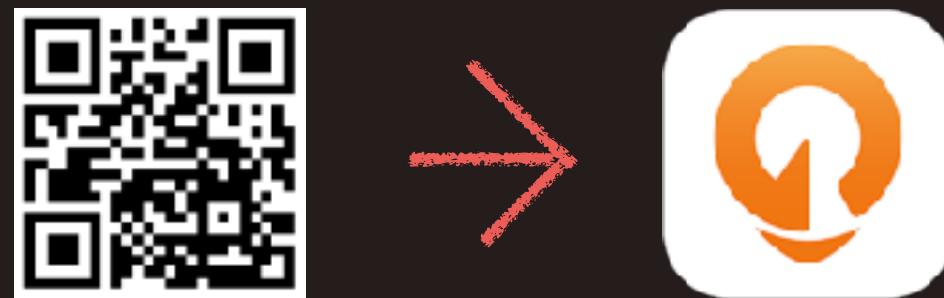
极客邦科技前端负责人

大纲





极客时间是什么？



极客时间是极客邦科技出品的一款 IT 内容知识服务 App,
包含专栏订阅、极客新闻、热点专题、直播、视频课和音频
等多形式的知识内容





视频直播、极客新闻、极客商城等



IT 专栏



音频播放、下载、不同用户播放进度存储，播放状态同步等



视频课程

Why?

为什么是 Weex



-
1. 性能优异
 2. 团队使用 Vue，上手简单
 3. 团队新、人数少，用 Weex 性价比高
 3. 大家都用，社区才能火起来
 4. 与极客邦的使命契合，助力中国开源技术，从自身产品做起





极客时间诞(xue)生(lei)记(shi)

1.5 人，5 个月，1 个 iOS App



“

我看了 Weex 的
demo，开发咱们这个
东西 So easy！

”

- 我自信满满

2017.5

iOS 版启动

开发人员最大的错觉：这个很好做！

2017.10.17

iOS 版在 Qcon 亮相

发布完以后我跟 iOS 开发 Mapple
两人相顾无言，惟有泪千行……



1.5 人， 40 天， 1 个安卓 App



“

幸好没有重新定义十一
月底啊！

”



-池老师老泪纵横地说

2017.10.19

安卓版启动

有一天到了公司同事们冲我笑得很意味深长，后来得知池老师发文说十一月底不发版就拉着我们跳油锅

2017.11.30

安卓版面世

幸好没有重新定义十一月底，也保住了老腊肉池老师和小鲜肉安卓小迪哥

多页面，

iOS 和 安卓各有一个纯原生页面

其余全是 Weex 实现

目前 Weex 页面共 44 个



iOS 仅有充值页为原生



安卓仅有音频播放页为原生

自定义组件和模块



开发和改造了二十几个 module / component

present

eventBus

navigatorExtend

pushSetting

storage

picker

audio

...

settings

statusBarControl

audioPersistance

modal

statistics

download

imagePicker

thirdParty

webview



WebView

解决黑线、闪代码、与 H5 通信等问题



H5 通知 Webview (iOS) , 参考 web 组件源码实现

```
- (void)notifyWebview:(NSDictionary *) data
{
    NSString *json = [WXUtility JSONString:data];
    NSString *code = [NSString stringWithFormat:@"(function(){var evt=null;var data=%@;if(typeof CustomEvent==='function'){evt=new CustomEvent('notify',{detail:data})}else{evt=document.createEvent('CustomEvent');evt.initCustomEvent('notify',true,true,data)}document.dispatchEvent(evt)}())", json];
    [_jsContext evaluateScript:code];
}
```



H5 接收 Webview 通知 (iOS / Android)

```
document.addEventListener('notify', event => {
  try {
    let data = event.detail.data
    switch (event.detail.type) {
      case 'audio:play':
        if (this.data.audio_md5 && (this.data.audio_md5 === data.audio_md5)) {
          this.playing = true
        }
        break
    }
  } catch (e) {
    console.error(e)
  }
})
```



Webview 调用 H5 (Android)

```
public void callJS(String functionName) {
    if(getWebView() == null)
        return;
    final String method = "javascript:" + functionName;
    getWebView().post(new Runnable() {
        @Override
        public void run() {
            getWebView().evaluateJavascript(method, new ValueCallback<String>() {
                @Override
                public void onReceiveValue(String s) {
                }
            });
        }
    });
}
```



H5 通知 Webview (Android)

```
wv.addJavascriptInterface(new NotifyInterface(), "name: "webviewJSInterface");
```

```
public class NotifyInterface extends Object {  
    @JavascriptInterface  
    public void notifyWeex(String msg) {  
        EventBus.getDefault().post(new NotifyWeexEvent(msg));  
    }  
}
```

```
@Subscribe  
public void onEventMainThread(NotifyWeexEvent event) {  
    final String msg = event.getMsg();  
    final Map<String, Object> params = new HashMap<>();  
    params.put("data", msg);  
  
    MainActivity.instance.runOnUiThread(new Runnable() {  
        @Override  
        public void run() {  
            fireEvent(type: "notify", params);  
        }  
    });  
}
```



H5 接收 Webview 通知 (Android)

```
  initEventDispatcher: function () {  
    if (utils.isInApp() && utils.isInAndroidApp()) {  
      window.geekTimeNotify = param => {  
        let data = typeof param === 'string' ? JSON.parse(param) : param;  
        let evt = new CustomEvent('notify', { detail: param });  
        document.dispatchEvent(evt);  
      };  
    };  
  },
```



EventBus

解决应用级数据通信问题

1. Native -> Weex (原生页面传输数据给Weex)
2. Weex -> Weex (Weex页面之间发送消息和数据)



Native -> Weex (原生页面传输数据给Weex)

实现应用级别的事件

与 GlobalEvent 实现区别是不通过 userWeexInstanceld 获取 userWeexInstance

```
- (void)nativeFireGlobalEvent:(NSNotification *)notification
{
    NSDictionary *userInfo = notification.object;
    for (WXModuleKeepAliveCallback callback in _eventCallback[notification.name]) {
        callback(userInfo, true);
    }
}
```

Native直接发送消息

```
NSString *savedBadgeJsonStr = [myBadgesDict GKT_jsonString];
[self.storageModule setItem:@"my_badges"
                      value:savedBadgeJsonStr?:@"
                      callback:nil];
[[NSNotificationCenter defaultCenter] postNotificationName:@"UpdateMyBadges"
                                              object:myBadgesDict];
```



Weex -> Weex (Weex页面之间发送消息和数据)

需要通过 Native 做中转，比如：登录页面 -> 专栏列表页

```
....// 保存用户信息
....saveLoginInfo: function (data) {
....  utils.saveLoginInfo(data).then(data => {
....    eventBus.fireNativeGlobalEvent('LoginSuccess', {
....      status: 'OK'
....    })
....    utils.rootDismiss('true')
....  })
....},
```

专栏列表页添加监听

```
mounted: function () {
  eventBus.addNativeEventListener('LoginSuccess', this.loginHandler)
},
```



Weex A -> Native -> Weex B

```
// Weex 给 Native 发送通知
@JSMethod
public void fireNativeGlobalEvent(String eventName, HashMap param){
    HashMap<String, HashMap> eventBusDataFromWeexMap = new HashMap<>();
    eventBusDataFromWeexMap.put(eventName, param);
    EventBus.getDefault().post(new DataSynEvent(eventBusDataFromWeexMap));
    HashMap event = new HashMap<String, Object>();
    event.put("result", "success");
    event.put("data", param);
    nativeFireGlobalEvent(eventName, event);
}

// Weex 添加监听
@JSMethod
public void addNativeEventListener(String eventName, JSCallback callback){
    HashMap<String, JSCallback> callbackHashMap = new HashMap<>();
    callbackHashMap.put(eventName, callback);
    jsCallbackList.add(callbackHashMap);
}
```



Native -> Weex

```
// Native 给 weex 执行回调
public static void nativeFireGlobalEvent (String eventName, HashMap event) {
    for(int i = 0; i < jsCallbackList.size(); i++) {
        if (jsCallbackList.get(i).containsKey(eventName)) {
            try {
                jsCallbackList.get(i).get(eventName).invokeAndKeepAlive(event);
            } catch (Throwable thr) {
                CrashReport.postCatchedException(thr);
            }
        }
    }
}
```



Video Component

比较复杂的视频播放器组件，倍速、清晰度、列表连续播放、权限判断、非 Wi-Fi 提醒



视频播放页 (iOS)

Video Page

Video Component +
Webview Component

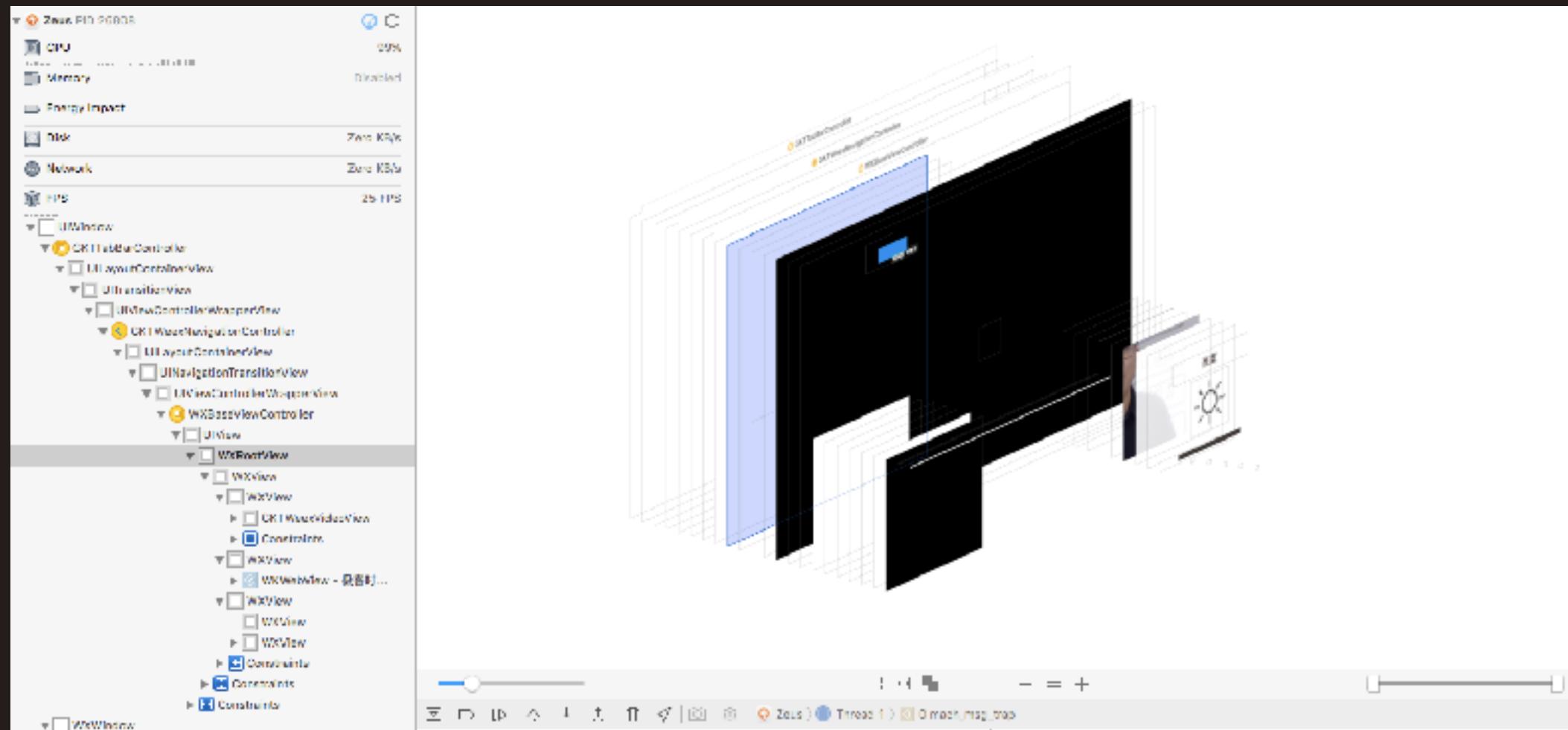
Weex Video Component

视频课页面



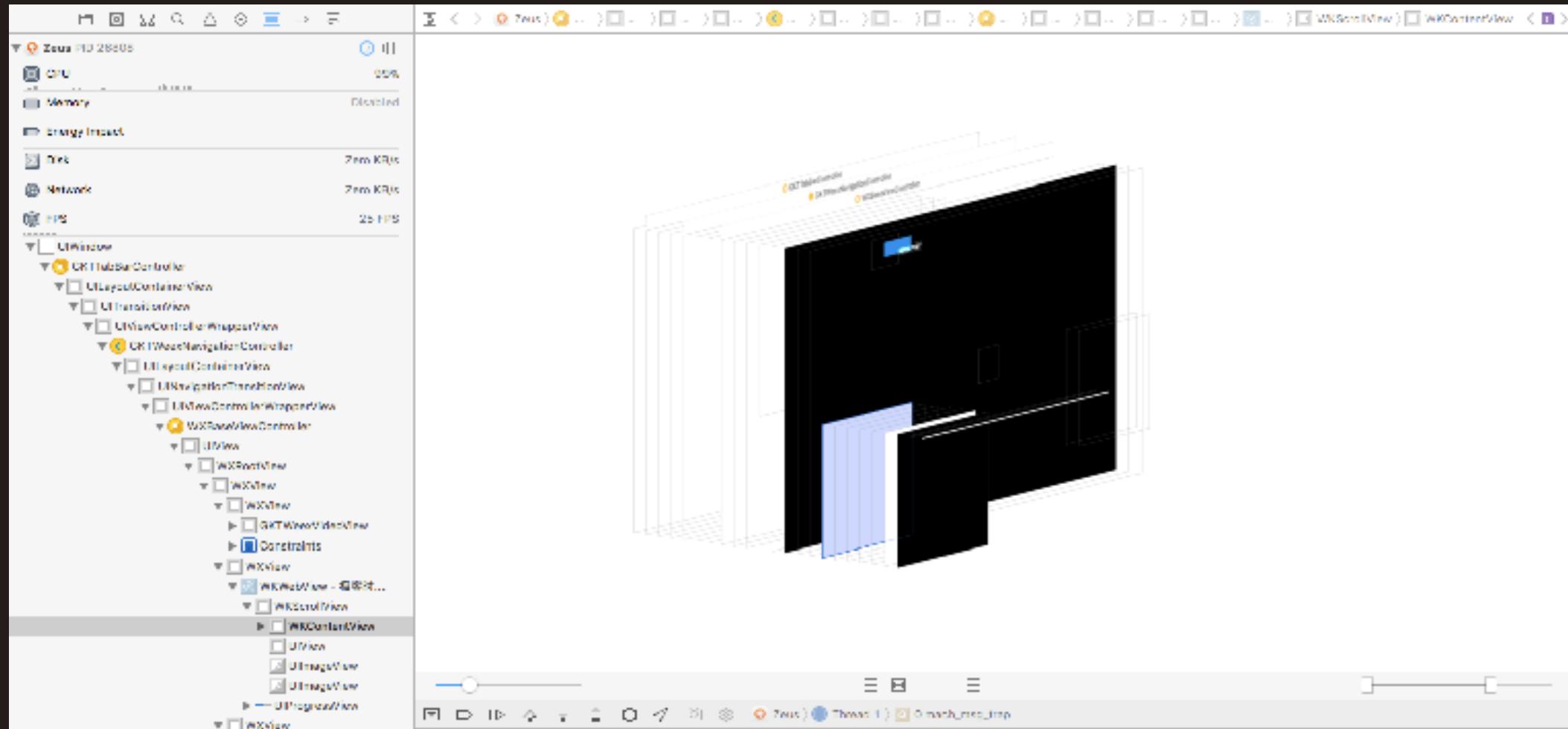
横屏时需要适配 Video 的 css 以及 Video 父容器的 frame

视图组件的父视图横屏时 frame 没改变



视频页面其他组件横屏时也需要适配

视频页面 Webview 组件视图横屏时需要隐藏



控制是否自动旋转、页面支持的方向功能

Native 端添加 WXBaseViewController (GKTLandScape)

```
- (BOOL)shouldAutorotate
{
    Ivar ivar = class_getInstanceVariable([self class], "_sourceURL");
    id sourceUrl = object_getIvar(self, ivar);
    if ([sourceUrl isKindOfClass:[NSURL class]]) {
        NSString *pageViewName = [self getPageViewNameFromSourceUrl:sourceUrl];
        if ([pageViewName isEqualToString:@"course_intro"] ||
            [pageViewName isEqualToString:@"course_detail"]){
            return !GKTVVideoBrightnessViewShared.isLockScreen; // 未来功能
        } else {
            return NO;
        }
    }

    return NO;
}
```



3 段小插曲



插曲 1

App 上线前贸然更新 SDK，半夜求助 Weex 团队



而且同样的问题还求助了两次!

第一次：OS Weex SDK 0.15 升级到 0.16.0

第二次：iOS Weex SDK 和 安卓 Weex SDK **双双** 升级到 0.17.0



插曲 2

旋转屏的问题，这次无论如何也要自己解决





安卓版视频横屏后再返回
Weex 页面尺寸不能重新计算
导致布局错乱



去掉页面中的 Weex 组件，
Webview 整个铺满，用 H5
模拟组件

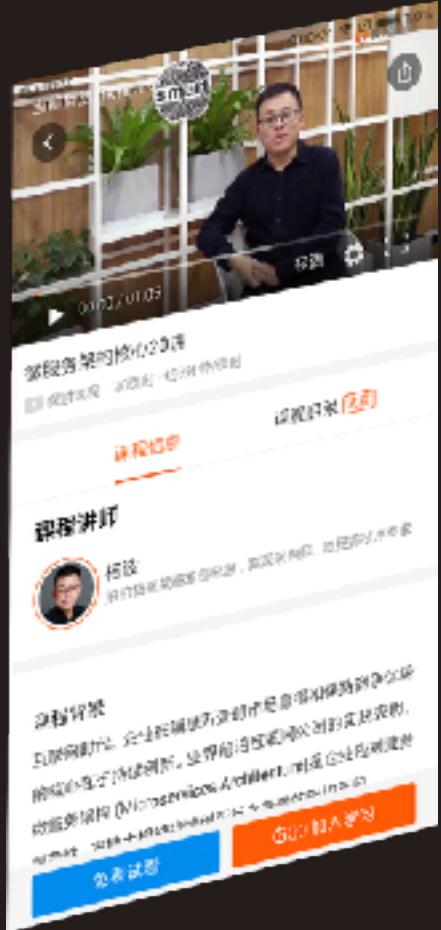


插曲3

安卓版 Weex 页面不能触发 viewappear 和 viewdisappear?

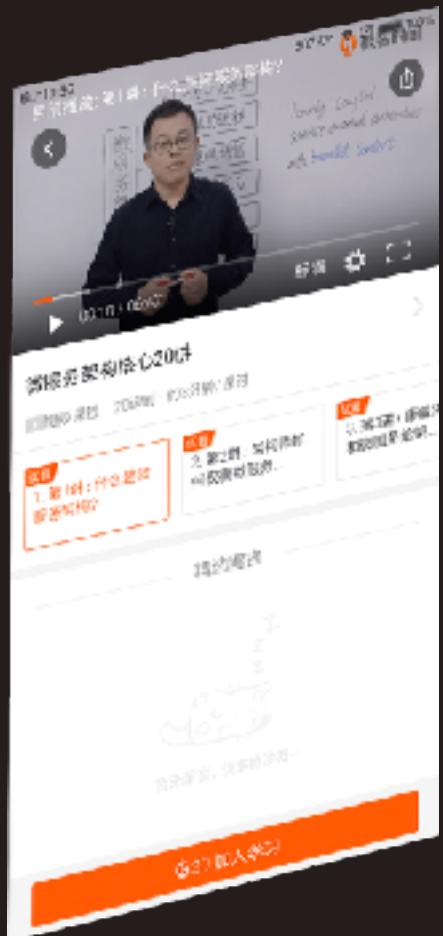
求助!





Webview A 绑定了登录通知

+



Webview B 也绑定了登录通知

未登录时

加入学习



调起了两次登录

x2

安卓需要在 Activity 手动执行一下 mWXSDKInstance 的 onActivityResume 等

```
@Override  
protected void onResume() {  
    MobclickAgent.onResume(context: this);  
    mWXSDKInstance.onActivityResume();  
    super.onResume();  
}
```

```
@Override  
protected void onPause() {  
    MobclickAgent.onPause(context: this);  
    mWXSDKInstance.onActivityPause();  
    overridePendingTransition(R.anim.back_enter, R.anim.back_exit);  
    super.onPause();  
}
```



一点感想

1. 文档先行
2. 配备好相应的原生开发，通读相关源码
3. 过于复杂的功能开发前要权衡、三思



没有事先搞好文档，中途开发了些别的东西，回到项目后看不懂了
安卓开发没有及时配备齐全导致进度耽误，后期两端开发不同步
特别复杂的功能建议用原生实现，否则可能逻辑会十分复杂或者有坑



对 Weex 的期待

1. 社区能更强大起来
2. 文档资料完善起来
3. 多组织交流活动



感谢一起战斗到吐血的兄弟们

希望 Weex 越来越完善，让更多的兄弟姐妹们早下班



Mapple

iOS 工程师，快 30
了，还没有女朋友



小迪

安卓工程师，还年
轻，但也没有女朋
友



旭明

Weex 工程师，有
女朋友，但架不住
老加班

Thank you!





WEEX

<https://weex.apache.org>