Javascript与移动原生应用的交互

小鱼网 刘玮


21:13
39
厦门
爆料
五折卡
物尽其用
租房
吓死人，东孚某路口惊现诡异人偶代替交警路口执勤，大晚上要吓尿啊
网络
3小时前
历史上最大PE：看多尔衮怎样收购大明朝资产
头条
消息
服务
论坛
我的


21:13
39
服务
推荐服务
五折问卷
上门保养
五折卡
租房
热门服务
12周年
婚博会
掌上农场
预算审核
美食榜单
看房团
装修招标
小鱼车标
小鱼厨房
五折激活
抢婚团
生活服务
公交查询
违章查询
医保查询
头条
消息
服务
论坛
我的

问题来源

实现的成果

点击按钮之后自动在h5页面里是登录的

实现思路

- 点击去php页面的时候，参数里添加用户id
- 然后在php里接收这个用户id，并自动登录
- 于是还要做好加密、防伪、身份验证

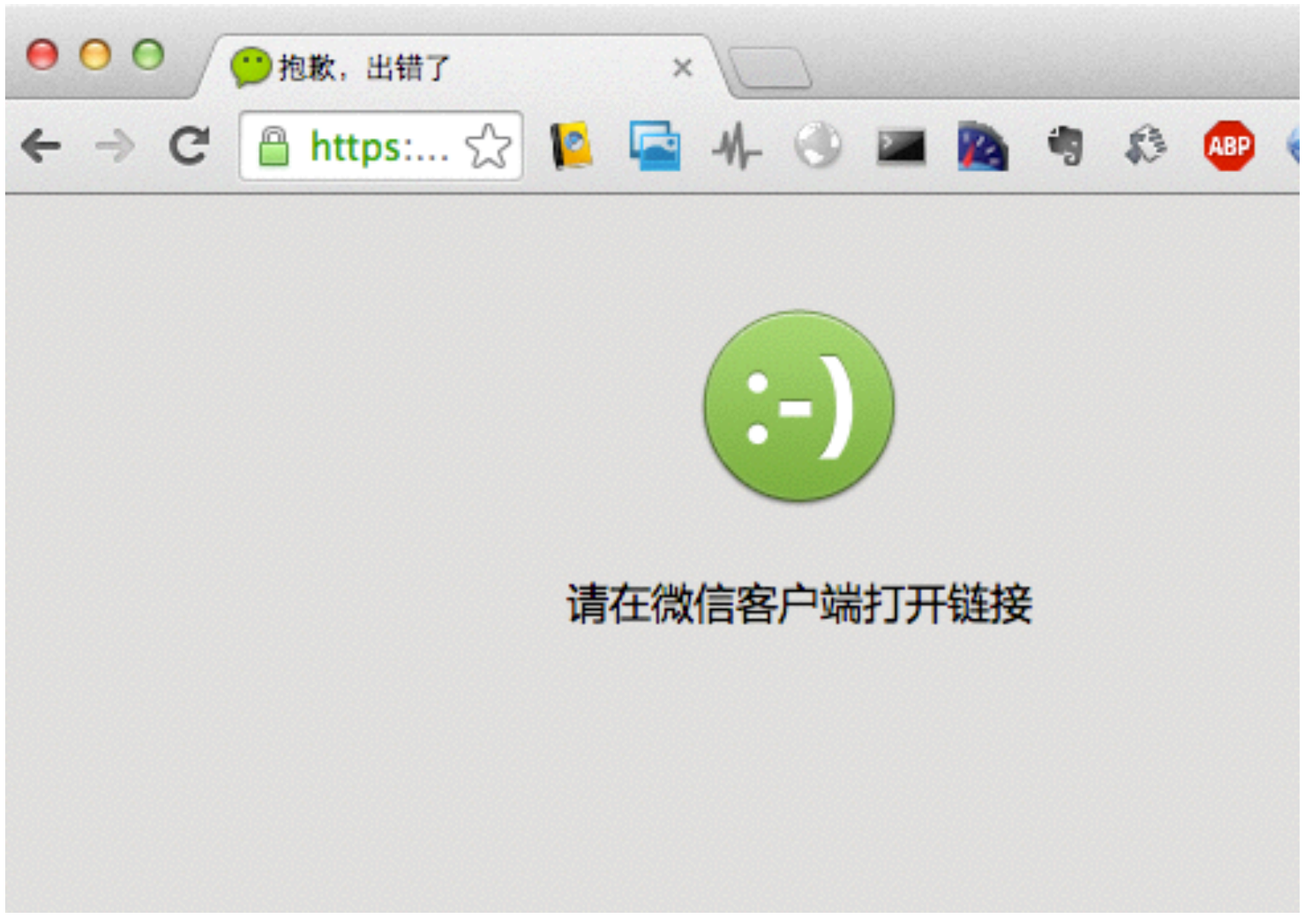
问题

- 把user信息暴露在网络上

解决方式

- 为WebView加上UserAgent标识
- String ua = webview.getSettings().getUserAgentString(); webview.getSettings().setUserAgentString(ua+"; MyUserAgent/“+appversion);
- 然后在PHP中添加对UserAgent的判断，防止从非客户端发出的请求。
- 不过还是不太靠谱，因为UserAgent还是可以被伪造。

效果



问题

如果外面的客户端没登录里面的php也不会在登录状态

然后可能出现在php里登录了，但在客户端没登录

产品页

实际是个HTML5

这里是原生

转到了原生登录页

这里是HTML5

解决方案

- Android
  WebView自带的JavascriptInterface

code

- mWebView.addJavascriptInterface(new Object() {
  public void clickOnAndroid(final int i) {
  mHandler.post(new Runnable() {
  public void run() {
  int j = i;
  j++;
  Toast.makeText(Test.this, "测试调用java" + String.valueOf(j),
  Toast.LENGTH_LONG).show();
  }
  });
  }
  }, “demo");

- b.c

How About iOS

- 一个神奇的开源项目 WebViewJavascriptBridge

- https://github.com/marcuswestin/WebViewJavascriptBridge

code

- self.bridge = [WebViewJavascriptBridge bridgeForWebView:webView handler:^(id data, WVJBResponseCallback responseCallback) {
    NSLog(@"Received message from javascript: %@", data);
    responseCallback(@"Right back atcha");
}];

- [self.bridge send:@"Well hello there"];
- [self.bridge send:[NSDictionary dictionaryWithObject:@"Foo" forKey:@"Bar"]];
- [self.bridge send:@"Give me a response, will you?" responseCallback:^(id responseData) {
  NSLog(@"ObjC got its response! %@“, responseData);
  }];

```
function connectWebViewJavascriptBridge(callback) {
    if (window.WebViewJavascriptBridge) {
        callback(WebViewJavascriptBridge)
    } else {
        document.addEventListener('WebViewJavascriptBridgeReady', function() {
            callback(WebViewJavascriptBridge)
        }, false)
    }
}

connectWebViewJavascriptBridge(function(bridge) {

    var callbackButton = document.getElementById('buttons').appendChild(document.createElement('button'))
    callbackButton.innerHTML = 'Fire testObjcCallback'
    callbackButton.onclick = function(e) {
        e.preventDefault()
        log('JS calling handler "testObjcCallback"')
        bridge.callHandler('testObjcCallback', {'foo': 'bar'}, function(response) {
            log('JS got response', response)
        })
    }
})
```

遗憾的是

- 在JS端这特么是两套玩法啊。
- 但这个明显难不过我们的工程湿同学。

合二为一

最后只用一个js包实现统一的调用

```
/**
 * 打开登录页
 */
function openLoginPage(){
    openapk.callHandler('openLoginPage',{a:1},function(n){
    });
}

/**
 * 打开个人中心
 * @param uid
 */
function openUserPage(uid){
    openapk.callHandler('openUserPage',{uid:uid},function(n){
    });
}

/**
 * 打开消息对话框
 * @param uid
 * @param username
 */
function openMessagePage(uid,username){
    openapk.callHandler('openMessagePage',{uid:uid,username:username},function(n){
    });
}
```

其它框架


NativeScript
Build truly native apps with JavaScript
Develop iOS, Android and Windows Phone apps from a single code base
GET STARTED
VIEW IN GITHUB


React Native
Docs
Support
GitHub
React
React Native
A FRAMEWORK FOR BUILDING NATIVE APPS USING REACT
React Native enables you to build world-class application experiences on native platforms using a consistent developer experience based on JavaScript and React. The focus of React Native is on developer efficiency across all the platforms you care about - learn once, write anywhere. Facebook uses React Native in multiple production apps and will continue investing in React Native.
Get started with React Native

万能的程序猿万岁~