Authenticating Using WeChat OAuth With the Ayla User Service

05/18/2017

Introduction

Ayla's Mobile SDK for iOS release 5.6 has support for WeChat OAuth. WeChat login for third-party applications operates similarly to the OAuth 2.0 standard. A known limitation is that only a single application is supported for WeChat OAuth.

Before implementing WeChat login in your app, you'll need to register for a developer account on their developer portal, have your app approved, and obtain the corresponding App ID and App Secret. Login functionality will be available after your application has been approved.

Note: Currently, only the original login method is allowed for WeChat login via mobile application. Therefore users must install the WeChat App in their iOS device.

Enabling WeChat OAuth within Aura

Basic WeChat OAuth support as discussed below is included with Aura, though it is disabled by default. Enabling only requires defining the environment variable INCLUDE_WECHAT_OAUTH when running `pod install`. With this variable defined, Cocoapods will pull down and integrate the WechatOpenSDK into Aura, and the relevant application UI will be revealed and activated.

Note for US Developers: Enabling WeChat OAuth outside of China may require manually specifying the WechatOpenSDK file path in the Podfile. Contact Ayla Customer Success for further information.

Integrating WeChat OAuth into your application

1. Configure on Ayla Dashboard

1. Sign in to the Ayla Dashboard with `OEM:Admin` or higher access account

2. Navigate to `OEM Profile` | `OAuth Credentials`, and create a Wechat OAuth credential. Fill the Client ID and Client Secret fields with Wechat App ID and App secret you see in Wechat developer portal.

2. Install Wechat SDK

Add the following line in your `Podfile`, then run `pod install`

```
pod 'WechatOpenSDK'
```

If you are using swift, also need to add the following line into you Project-Bridging-Header.h:

```
#import "WXApi.h"
```

Should you encounter the error: *“[!] Unable to find a specification for `WechatOpenSDK`”* try ‘pod repo update’ and / or ‘pod clean’ followed by ‘pod update’

3. Integrating

1. Add a new URL scheme in project Info.plist file. Set the value with your Wechat App ID.

2. Since Wechat OAuth requires the Wechat App be installed, checking to ensure it is installed is the best practice. To perform the test, a new LSApplicationQueriesSchemes is required for Wechat. Just add the following in project Info.plist:

```
<key>LSApplicationQueriesSchemes</key>
  <array>
     <string>weixin</string>
  </array>
```

3. Register Wechat App in didFinishLaunchingWithOptions function of AppDelegate.

```
// Register Wechat App
WXApi.registerApp(AuraOptions.WechatAppId)
```

4. Implement WXApiDelegate, which should look similar to:

```
// MARK: - WXApiDelegate
extension AppDelegate: WXApiDelegate {
    func onReq(_ req: BaseReq!) {

    }

    func onResp(_ resp: BaseResp!) {
        let userInfo: [String: Any] = ["auth_resp": resp]
        NotificationCenter.default.post(name: NSNotification.Name(rawValue: AuraNotifications.WechatOAuthResponse), object: nil, userInfo: userInfo)
    }
}
```

5. Override handleOpenURL and openURL functions:

```
// Wechat OAuth
if WXApi.handleOpen(url, delegate: self) {
    return true
}
```

6. Next, call Wechat sign-in API (usually from your sign in page). Here is the sample code from Aura:

```
@IBAction func wechatLoginPressed(_ sender: Any) {
    if WXApi.isWXAppInstalled() {
        let authReq = SendAuthReq()
        authReq.scope = "snsapi_userinfo"
        authReq.state = "Aura"
        WXApi.send(authReq)
    } else {
        UIAlertController.alert(nil, message: "Wechat App must be i
nstalled for Wechat sign in.", buttonTitle: "OK", fromController: s
elf)
    }
}
```

7. Finally, the previous step will be redirected to the Wechat App. Once the user permits Wechat sign-in, control will be redirected back to your App. At this time pass the auth code to the delegate function which was set in step 4. With the auth code, you can sign into the Ayla User Service. Here is sample code from Aura:

```
let authProvider = AylaWechatOAuthProvider(authCode: authCode)
let loginManager = AylaNetworks.shared().loginManager
loginManager.login(with: authProvider, sessionName: AuraSessionOneN
ame, success: { [unowned self](_, sessionManager) -> Void in
    self.performSegue(withIdentifier: self.segueIdToMain, sender: s
essionManager)
}, failure: { [unowned self] (error) -> Void in
    self.presentError(error)
})
```