iOS 平台 MakeysSDK 文档

编号：MAKEYS_IOS_SDK

版本：MAKEYS_IOS_SDK V1.0.0

修订记录：

时间	文档版本	修订人	备注
2018/05/29	1.0.0	周美丽	初稿

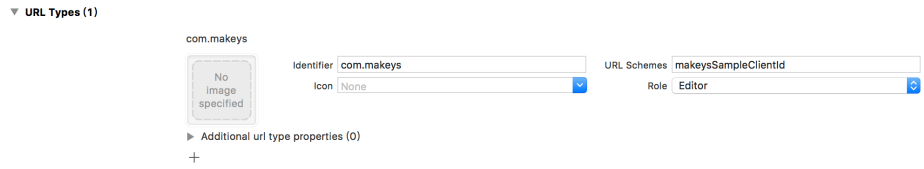
目录

MakeysSDK

一、SDK 接入设置

1、设置工程回调 URL Scheme

修改 info.plist 文件 URL types 项为自己的授权登录回调地址,”makeys[你的应用程序的 Appkey]”,例如:makeysSampleClientId

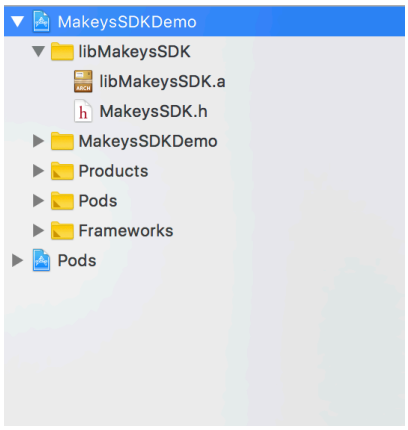


2、设置工程 Scheme 白名单

在项目的 info.plist 中添加 LSApplicationQueriesSchemes，类型为 Array。然后给它添加 makeys，类型为**字符串**类型；

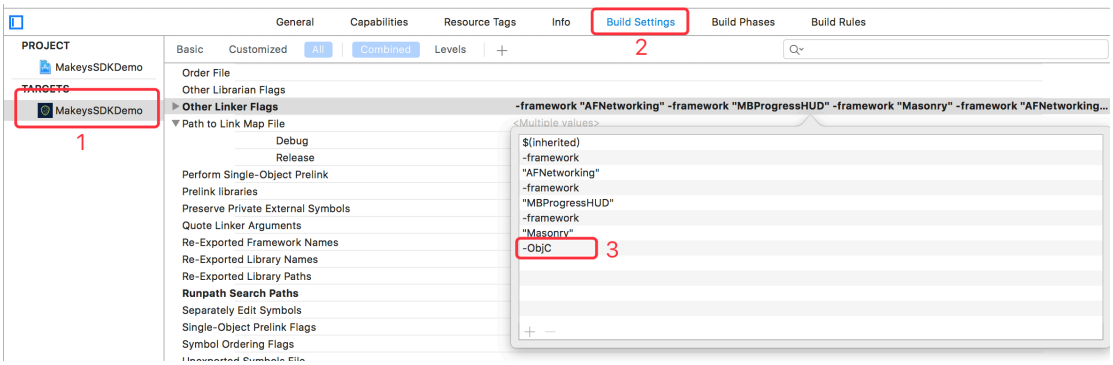
3、添加 SDK 文件到工程

将从 GitHub 上下载的 libMakeysSDK 文件夹添加至工程，其中包含 MakeysSDK.h 文件以及 libMakeysSDK.a，统共 2 个文件。



4、在工程中引入静态库之后，需要在编译时添加-objC 编译选项

避免静态库中类加载不全造成程序崩溃。方法：程序 Target->Buid Settings- >Linking 下 Other Linker Flags 项添加-ObjC。



5、定义应用授权登录所需的几个常量

AppKey:第三方应用申请的 appkey,用来身份鉴证、显示来源等;AppRedirectURL:应用回调页。对于 Mobile 客户端应用来说,是不存在 Server 的,故此处的应用回调页地址只要与应用回调页中的 url 地址保持一致就可以了,如图所示:

```
#define kAppKey         @"SampleClientId"
#define kRedirectURI    @"http://your_callback_uri"
```

6、注册 appkey(clientid)

程序启动时,在代码中向 MakeysSDK 注册你的 Appkey，可设置授权界面语言，默认跟随系统。

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Override point for customization after application launch.

    [MakeysSDK setLanguageType:MakeysSDKLanguageTypeChinese];

    [MakeysSDK registerApp:kAppKey];

    return YES;
}
```

7、重写 AppDelegate 的 handleOpenURL 和 openURL 方法

```
- (BOOL)application:(UIApplication *)app openURL:(NSURL *)url options:(NSDictionary<UIApplicationOpenURLOptionsKey,id> *)options
{
    return [MakeysSDK handleOpenURL:url delegate:self];
}

- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url sourceApplication:(NSString *)sourceApplication annotation:(id)annotation
{
    return [MakeysSDK handleOpenURL:url delegate:self];
}

- (BOOL)application:(UIApplication *)application handleOpenURL:(NSURL *)url
{
    return [MakeysSDK handleOpenURL:url delegate:self ];
}
```

二、应用场景代码示例

1、Makeys 客户端授权认证

```
{
    MakeysAuthorizeRequest *request = [MakeysAuthorizeRequest request];
    request.state = @"Verification";
    request.scope = @"user_info";
    request.redirectURI = kRedirectURI;
    [MakeysSDK sendRequest:request];
}
```

调用sendRequest 的方法后会跳转到Makeys程序。如果当前Makeys客户端没有账号,则进入登录界面;如果当前Makeys客户端已经有账户,则进入授权登录界面,选择要向第三方授权的账户。当授权完成后会回调给第三方应用程序,第三方实现MakeysSDKDelegate 的 didReceiveMakeysResponse：responseStatusCode：方式监听此次请求的 response.

此中 state 内容为用户自定义(可不填写)，MakeysSDK 回调 Response 中会通过 requestState 包含原 request.state 中的所有数据，用于数据传输过程中校验相关的上下文环境数据;scope 内容参照开放平台权限列表，可不填写。

2、从第三方应用向 MakeysSDK 发送请求

代码示例如:

```
{
    MakeysAuthorizeRequest *request = [MakeysAuthorizeRequest request];
    request.state = @"Verification";
    request.scope = @"user_info";
    request.redirectURI = kRedirectURI;
    [MakeysSDK sendRequest:request];
}
```

同上此中 state 内容为用户自定义(可不填写)，MakeysSDK 回调 Response 中会通过 requestState 包含原 request.state 中的所有数据，用于数据传输过程中校验相关的上下文环境数据;scope 内容参照开放平台权限列表，可不填写。

3、第三方应用收到 MakeysSDK 回调

代码示例如:

```objc
- (void)didReceiveMakeysResponse:(MakeysBaseResponse *)response responseStatusCode:(MakeysSDKResponseStatusCode)responseStatusCode {

    if (responseStatusCode == MakeysSDKResponseStatusCodeSuccess) {
        MakeysSuccessedResponse *authorizeResponse = (MakeysSuccessedResponse *)response;
        NSString *requestState = authorizeResponse.requestState;
        NSString *accessToken = authorizeResponse.accessToken;

        NSMutableDictionary *parameters = [NSMutableDictionary dictionary];
        int num = (arc4random() % 10000000000);
        NSString *randomNumber = [NSString stringWithFormat:@"%.10d", num];
        parameters[@"state"] = requestState;
        parameters[@"redirectUri"] = kRedirectURI;
        parameters[@"code"] = accessToken;
        parameters[@"nonce"] = randomNumber;
        UINavigationController *rootNav = (UINavigationController *)self.window.rootViewController;
        AuthorizeLoginViewController *rootVC = rootNav.viewControllers.firstObject;
        [rootVC loginByAuthCode:parameters];
    }
    else if (responseStatusCode == MakeysSDKResponseStatusCodeAuthDeny) {
        MakeysFailedResponse *authorizeResponse = (MakeysFailedResponse *)response;
        NSString *errorCode = authorizeResponse.errorCode;
        NSString *errorCodeDescription = authorizeResponse.errorCodeDescription;
        NSString *message = [NSString stringWithFormat:@"response.errorCode:%@\nresponse.errorCodeDescription:%@",errorCode,errorCodeDescription];
        UIAlertController *alertController = [UIAlertController alertControllerWithTitle:@"授权失败" message:message preferredStyle:UIAlertControllerStyleAlert];
        UIAlertAction *actionCancel = [UIAlertAction actionWithTitle:@"确定"
                                                               style:UIAlertActionStyleCancel
                                                             handler:nil];
        [alertController addAction:actionCancel];
        [self.window.rootViewController presentViewController:alertController animated:YES completion:nil];
    }
    else if (responseStatusCode == MakeysSDKResponseStatusCodeUserCancel) {
        UIAlertController *alertController = [UIAlertController alertControllerWithTitle:@"用户点击取消" message:nil preferredStyle:UIAlertControllerStyleAlert];
        UIAlertAction *actionCancel = [UIAlertAction actionWithTitle:@"确定"
                                                               style:UIAlertActionStyleCancel
                                                             handler:nil];
        [alertController addAction:actionCancel];
        [self.window.rootViewController presentViewController:alertController animated:YES completion:nil];
    }
    else if (responseStatusCode == MakeysSDKResponseStatusCodeUserCancelInstall) {
        UIAlertController *alertController = [UIAlertController alertControllerWithTitle:@"用户取消下载" message:nil preferredStyle:UIAlertControllerStyleAlert];
        UIAlertAction *actionCancel = [UIAlertAction actionWithTitle:@"确定"
                                                               style:UIAlertActionStyleCancel
                                                             handler:nil];
        [alertController addAction:actionCancel];
        [self.window.rootViewController presentViewController:alertController animated:YES completion:nil];
    }
}
```

MakeysSDK 回调 responseStatusCode 为 MakeysSDKResponseStatusCodeSuccess（授权成功）、MakeysSDKResponseStatusCodeUserCancel（用户取消）、MakeysSDKResponseStatusCodeAuthDeny（授权失败）、MakeysSDKResponseStatusCodeUserCancelInstall（用户取消安装 Makeys 客户端）的枚举，可根据对应枚举做相关的数据处理。

当 responseStatusCode 返回为 MakeysSDKResponseStatusCodeSuccess 时，回调的 response 中包含 accessToken、requestState。accessToken 为访问相关信息的认证口令； requestState 包含原 request.state 中的所有数据，用于数据传输过程中校验相关的上下文环境数据。

当 responseStatusCode 返回为 MakeysSDKResponseStatusCodeAuthDeny 时，回调 Response 中包含 errorCode、errorCodeDescription。授权失败时 MakeysSDK 会通过 errorCode 返回相关响应状态码，通过 errorCodeDescription 返回状态码描述；具体状态码说明，请访问开发文档->OAuth2.0->开发相关资源->返回状态码说明。

当 responseStatusCode 返回为 MakeysSDKResponseStatusCodeUserCancel、MakeysSDKResponseStatusCodeUserCancelInstall 时，回调 Response 中不包含相关信息。

具体调用示例参见demo