

Q&A

1. 如何自定义一键分享列表？

在三种分享方式的接口下都带有一个 `oneKeyShareList` 参数，此参数用于自定义显示一键分享的平台类型，如下加粗部分代码所示：

```
//构造分享内容
id<ISSContent> publishContent = [ShareSDK content:CONTENT
                                     defaultContent:@""
                                     image:[ShareSDK imagePath:imagePath]
                                     title:@"ShareSDK"
                                     url:@"http://www.sharesdk.cn"
                                     description:@"这是一条测试信息"
                                     mediaType:SSPublishContentMediaTypeNews];

//定制一键分享列表
NSArray *oneKeyShareList = [ShareSDK getShareListWithType:
                               ShareTypeSinaWeibo,
                               ShareTypeTencentWeibo,
                               nil];

//弹出分享菜单
[ShareSDK showShareActionSheet:nil
  shareList:nil
  content:publishContent
  statusBarTips:YES
  authOptions:authOptions
  shareOptions:shareOptions
  result:^(ShareType type, SSPublishContentState state, id<ISSStatusInfo>
statusInfo, id<ICMErrorInfo> error, BOOL end) {
    if (state == SSPublishContentStateSuccess)
    {
        NSLog(@"分享成功");
    }
    else if (state == SSPublishContentStateFail)
    {
        NSLog(@"分享失败, 错误码:%d, 错误描述:%@", [error errorCode], [error
errorDescription]);
    }
}
}];
```

如果不需要一键分享可以传入 `nil`，默认一键分享列表则可以传入 `[NSArray defaultOneKeyShareList]` 参数。

2. 微信和 QQ 的 AppKey 在哪里配置?

从微信和 QQ 开放平台申请下来 AppKey 后，首先导入微信或者 QQ 的 SDK 头文件：

```
//导入微信SDK头文件
#import "WXApi.h"
//导入QQSDK头文件
#import <QQApi/QQApi.h>
```

打开*AppDelegate.m(*代表你的工程名字)，然后在 didFinishLaunchingWithOptions 方法中写入注册应用的代码，如下：

```
//添加QQ应用
[ShareSDK connectQQWithAppId:@"QQ0F0A941E" qqApiCls:[QQApi class]];
//添加微信应用
[ShareSDK connectWeChatWithAppId:@"wx6dd7a9b94f3dd72a" wechatCls:[WXApi class]];
```

然后确保 openURL 方法中加入如下代码：

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

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

最后打开*-info.plist (*代表你的工程名字) 文件。添加用于处理微信和 QQ 的回调 URL Scheme。
如下图：

ISSUserInfo.h	Supported interface orientations (if available)	Array	(1 item)
NSArray+ShareSDK.h	URL types	Array	(1 item)
ShareConfig.h	Item 0 (Editor)	Dictionary	(3 items)
ShareSDK.h	Document Role	String	Editor
ShareType.h	URL Schemes	Array	(3 items)
ShareViewStyle.h	Item 0	String	sinaweibosso.2279784657
SSInheritValue.h	Item 1	String	wx6dd7a9b94f3dd72a
SSNotificationType.h	Item 2	String	QQ075BCD15
SSPublishContentMediaType.h	URL identifier	String	
UserParamType.h	Icon already includes gloss effects	Boolean	YES
AGShareSDKDemo	Status bar style	String	Opaque black style
Supporting Files			
AGShareSDKDemo-Info.plist			
InfoPlist.strings			
main.m			

3. 如何查看错误码?

所有带错误码返回的接口（包括分享、获取用户资料、关注用户等）都通过特定的 block 方法中的 **error** 进行返回。因此，在调用接口的时候指定 block 方法，然后根据返回结果来获取错误代码即可。下面加粗部分代码展示了如何捕获分享异常的错误码以及错误描述：

```
//弹出分享菜单
[ShareSDK showShareActionSheet:nil
 shareList:nil
 content:publishContent
 statusBarTips:YES
 authOptions:authOptions
 shareOptions:shareOptions
 result:^(ShareType type, SSPublishContentState state, id<ISSStatusInfo>
statusInfo, id<ICMErrorInfo> error, BOOL end) {
    if (state == SSPublishContentStateSuccess)
    {
        NSLog(@"分享成功");
    }
    else if (state == SSPublishContentStateFail)
    {
        NSLog(@"分享失败, 错误码:%d, 错误描述:%@", [error errorCode], [error
errorDescription]);
    }
}
}];
```

error 参数是一个实现了 ICMErrorInfo 协议的对象，其包含三个属性：

一个是 **errorLevel**，用于指定错误级别；有三种级别分别是 **CLErrorLevelAPI**，**CLErrorLevelHTTP** 以及 **CLErrorLevelNetwork**。其中 **CLErrorLevelAPI** 为调用 API 时返回错误，该错误码与各个平台返回错误码相同，详细可以参考各个平台的错误码描述。**CLErrorLevelHTTP** 为 HTTP 请求错误，错误码为 HTTP 返回状态码，**CLErrorLevelNetwork** 则表示网络错误。

第二个是 **errorCode**，包含了指定的错误代码。

第三个是 **errorDescription**，包含了对应错误码的描述信息。

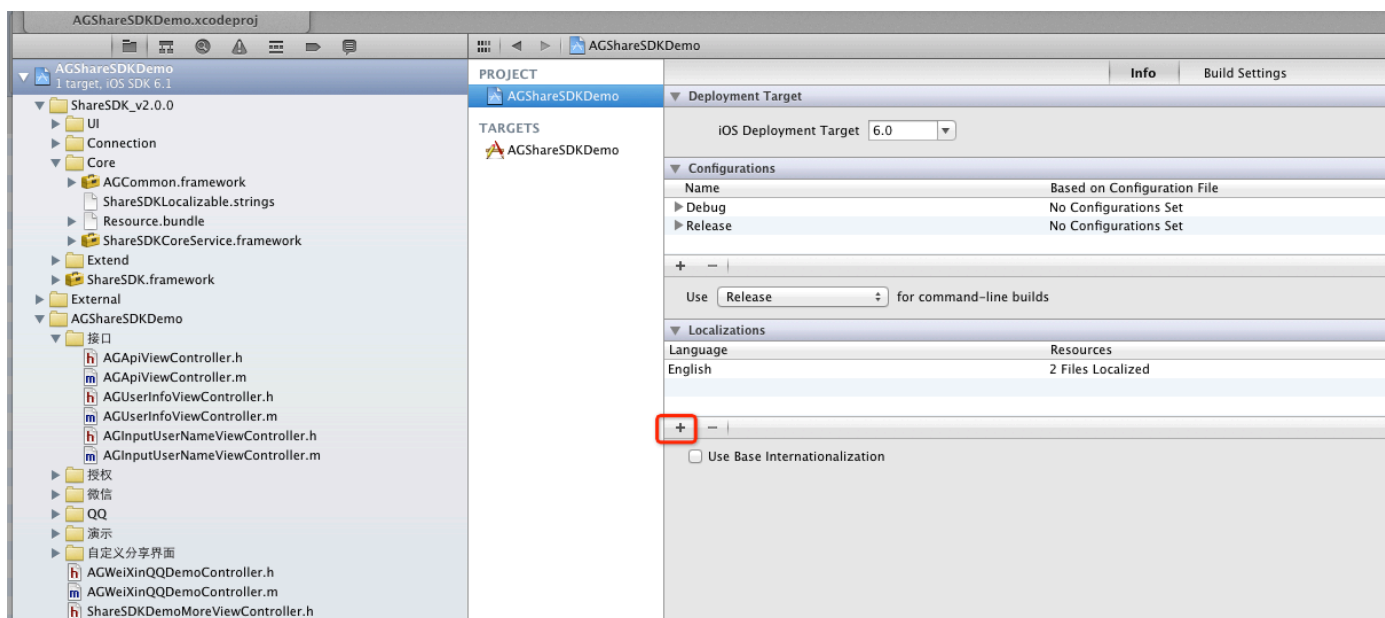
4. 如何获得成功或失败的信息？

要想知道一个接口的返回状态是否成功，可以通过每个带有 **result** 参数的接口传递一个 **block** 方法。然后由 **result** 返回的参数中判断是否成功，以及失败的错误信息。**result** 的 **block** 方法定义可 SDK 中的 **ShareSDKEventHandlerDef.h** 头文件，下面加粗部分代码展示了，如何判断分享是否成功，如果不成功则打印错误日志：

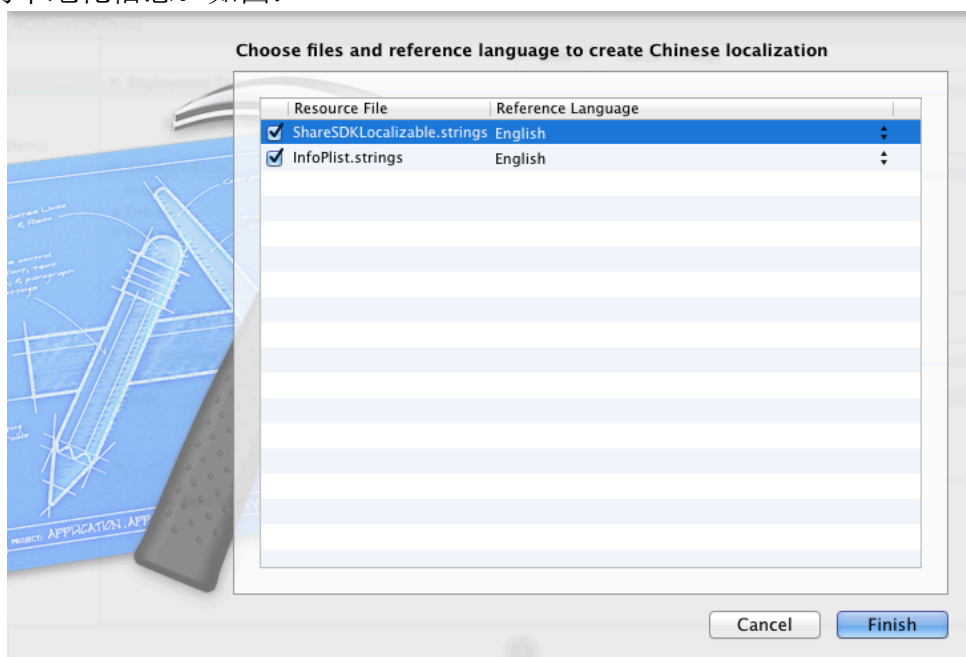
```
//弹出分享菜单
[ShareSDK showShareActionSheet:nil
    shareList:nil
    content:publishContent
    statusBarTips:YES
    authOptions:authOptions
    shareOptions:shareOptions
    result:^(ShareType type, SSPublishContentState state, id<ISSStatusInfo>
statusInfo, id<ICMErrorInfo> error, BOOL end) {
    if (state == SSPublishContentStateSuccess)
    {
        NSLog(@"分享成功");
    }
    else if (state == SSPublishContentStateFail)
    {
        NSLog(@"分享失败, 错误码:%d, 错误描述:%@", [error errorCode], [error
errorDescription]);
    }
}
}];
```

5. 如何为 ShareSDK 实现本地化 UI 及提示信息？

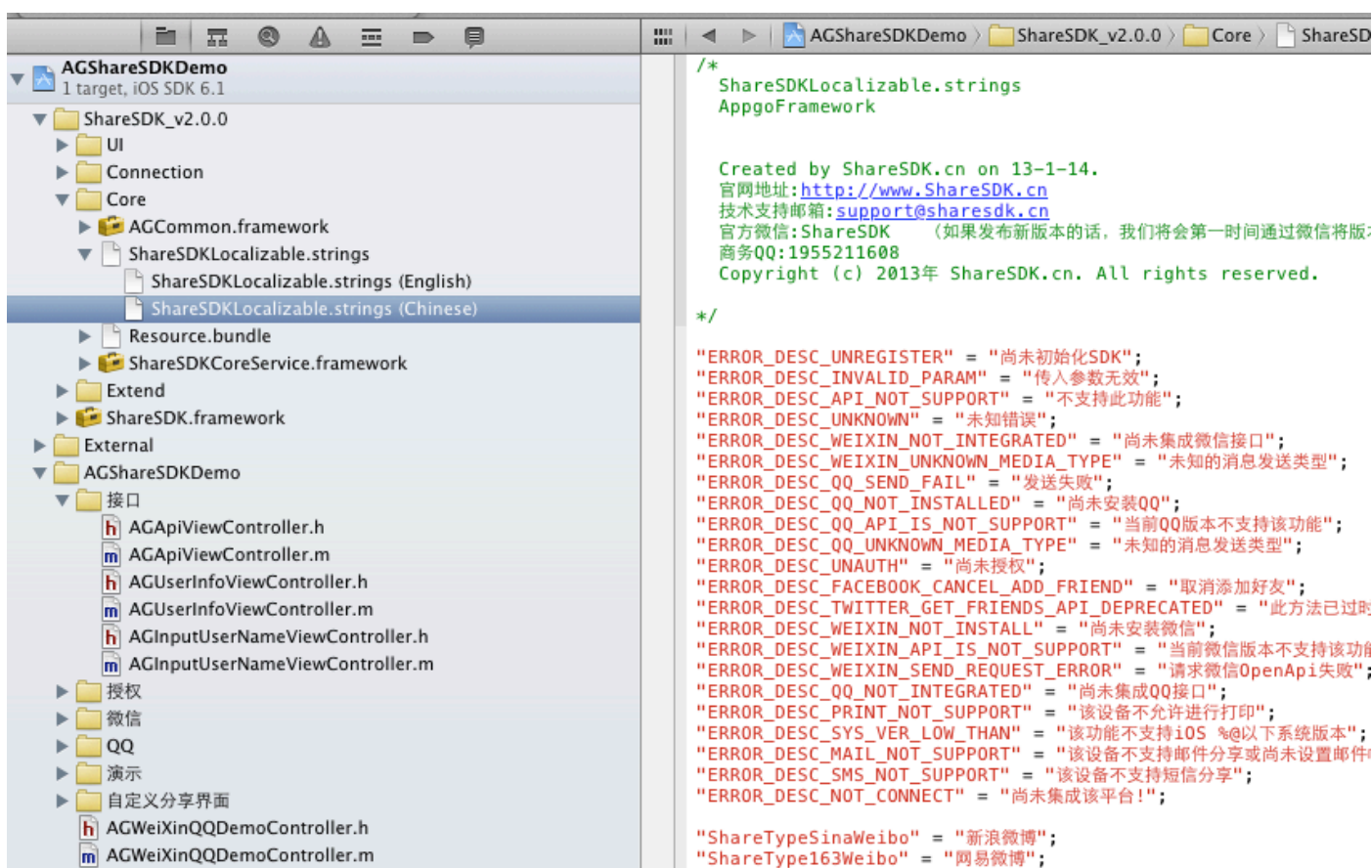
进入项目设置界面，在 **Info** 分页下的 **Localizations** 中点击“+”号新增自己需要的语言。如图：



再弹出的界面上选中 ShareSDKLocalizable.strings 文件，点击 Finish 按钮即可为 SDK 新增对应语言的本地化信息。如图：



然后点击 ShareSDKLocalizable.strings 文件左边的小三角形按钮展开分支，选择需要编辑的对应语言版本即可完成本地化操作。如图：



6. 如何自定义分享菜单中的平台列表？

在 showShareActionSheet 方法中的第二参数 shareList 用于指定分享菜单中需要显示那些分享平台。如果传入 nil 则表示使用默认的分​​享平台列表，如果不需要某些平台时可以使用 ShareSDK

中的 `getShareListWithType` 方法来构造一个符合自己需求的分享列表。下面加粗部分代码演示了只显示新浪微博和腾讯微博的分享列表：

```
NSString *imagePath = [[NSBundle mainBundle] pathForResource:IMAGE_NAME ofType:IMAGE_EXT];

//构造分享内容
id<ISSContent> publishContent = [ShareSDK content:CONTENT
                                     defaultContent:@" "
                                     image:[ShareSDK imagePath:imagePath]
                                     title:@"ShareSDK"
                                     url:@"http://www.sharesdk.cn"
                                     description:@"这是一条测试信息"
                                     mediaType:SSPublishContentMediaTypeNews];

NSArray *shareList = [ShareSDK getShareListWithType:
                        ShareTypeSinaWeibo,
                        ShareTypeTencentWeibo,
                        nil];

//弹出分享菜单
[ShareSDK showShareActionSheet:container
                        shareList:shareList
                        content:publishContent
                        statusBarTips:YES
                        authOptions:authOptions
                        shareOptions:shareOptions
                        result:^(ShareType type, SSPublishContentState state, id<ISSStatusInfo>
statusInfo, id<ICMErrorInfo> error, BOOL end) {
    if (state == SSPublishContentStateSuccess)
    {
        NSLog(@"分享成功");
    }
    else if (state == SSPublishContentStateFail)
    {
        NSLog(@"分享失败, 错误码:%d, 错误描述:%@", [error errorCode], [error
errorDescription]);
    }
}];
```

7. 豆瓣社区授权时页面出现“invalid_request:not_trial_user:xxxxxx Error 400”报错，如果解决？

由于所创建的 App 尚未通过审核，因此在授权用户不是测试用户的情况下进行授权就会出现如下报错页面：



解决方法是打开豆瓣开发者页面中的“我的应用”进入需要设置测试用户的 App，选择“测试用户”项，在右边界面的输入框填入报错界面中的 id 后，点击“添加”按钮添加测试用户即可。

8、人人网分享失败，出现“**Error Code:300; Error Description: Operation failed for 所有文件都处理失败**”解决方法：

出现此问题的原因很有可能是上传的图片不符合人人网的规定，可以检查图片信息是否符合规定，其规定如下：

- 如果不上传到指定的相册，即aid参数值为0，那么将默认上传到手机相册
- 每个相册上传照片的大小限制为120张。
- 目前支持的文件类型有：image/bmp, image/png, image/gif, image/jpeg, image/jpg
- 上传文件的大小不能超过8M
- 确保图片宽、高均大于50像素，并且宽高比大于1/3
- 计算 sig 的时候不包含所上传的文件

9、QQ 空间分享失败，出现“**Error Code:100031;Error Description:app has no privilege to use this api**”

出现此问题的原因有两种，第一种是你的应用设置的使用范围不正确，应该选择“移动应用”。如图：

* 使用范围 ☐ 网站 ☒ 移动应用

基本信息

* 网站/应用名称

第二种情况是你的应用没有申请 `upload_pic` 接口权限，这种情况是纯文字信息可以分享，但是带有图片的信息是无法分享的。请到 <http://wiki.opensns.qq.com/wiki/【QQ登录】API文档> 里面进行接口的权限申请。如图所示：

分类	功能说明	接口	是否需要申请接口权限
空间	获取用户在QQ空间的个人资料	<code>get_user_info</code> (仅支持网站调用)	不需要
		<code>get_simple_userinfo</code> (仅支持移动应用调用)	不需要
	发表一条说说到QQ空间	<code>add_topic</code>	需要，现在就申请
	发表一篇日志到QQ空间	<code>add_one_blog</code>	需要，现在就申请
	创建一个QQ空间相册	<code>add_album</code>	需要，现在就申请
	上传一张照片到QQ空间相册	<code>upload_pic</code>	需要，现在就申请
	获取用户QQ空间相册列表	<code>list_album</code>	需要，现在就申请
	获取安装了移动应用的好友信息	<code>get_app_friends</code> (仅供托管的移动应用调用)	托管后可申请
	同步分享到QQ空间、腾讯微博	<code>add_share</code>	不需要
	验证是否认证空间粉丝	<code>check_page_fans</code>	不需要
	发表一条微博信息到腾讯微博	<code>add_t</code>	不需要

10、如何解决带 gif 动画图片的信息分享成功后变为静态图片问题？

对于发送 gif 图片时，请使用下面方法构造图片附件对象：

```
+ (id<ISSCAttachment>)imageWithPath:(NSString *)path;

+ (id<ISSCAttachment>)imageWithData:(NSData *)data
    fileName:(NSString *)fileName
    mimeType:(NSString *)mimeType;
```

11、如何解决在分享视图点击分享按钮后界面处于无响应问题？

出现这种问题是由于没有设置 Key Window 的 `rootViewController` 属性导致的。因此解决的方法

就是打开 AppDelegate.m 文件，在 didFinishLaunchingWithOptions 方法中把 window 的 rootViewController 属性设置为应用的根视图控制器即可，如下：

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    .....

    self.window = [[[UIWindow alloc] initWithFrame:[UIScreen mainScreen] bounds]] autorelease];

    //主视图
    UIViewController *apiVC = [[[AGApiController alloc] init] autorelease];
    apiVC.title = @"接口";
    UINavigationController *navApiVC = [[[UINavigationController alloc]
initWithRootViewController:apiVC] autorelease];

    //左视图
    AGLeftSideViewController *leftVC = [[[AGLeftSideViewController alloc] init] autorelease];

    IIViewDeckController *vc = [[[IIViewDeckController alloc] initWithCenterViewController:navApiVC
leftViewController:leftVC] autorelease];
    vc.leftSize = self.window.width - (320 - 44.0);
    self.viewController = vc;

    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    return YES;
}
```