

Aliyun Open Service iOS SDK 使用说明

baocai zhang

www.giser.net

1 简介

Aliyun Open Service iOS SDK 使用 Objective-c 实现了 Aliyun Open Service 提供的功能，主要 OSS 和 OTS。

OSS 功能包括 Bucket 的创建、删除、浏览，Object 的创建、删除浏览，多点上传以及 Object Group 等功能，关于 OSS 提供的服务请参考 OSS API 说明文档。

OTS 功能主要包括表和视图的创建、删除、查询，表数据的添加、更新、删除以及事务的开始、提交、中断等功能。关于 OTS 提供的服务请参考 OTS API 说明文档。

2 主要内容

Aliyun Open Service iOS SDK 主要包括源代码（BSD 协议开源）、说明文档、示例工程、在线帮助文档和本地帮助文档等。

在线帮助文档地址：<http://osssdkhelp.sinaapp.com/>

本 SDK 同样支持在 Mac OS X 系统中使用，并提供了相关工程。

3 主要依赖

本 SDK 依赖下列主要 lib 和 framework：

Foundation.framework

CFNetwork.framework

SystemConfiguration.framework

MobileCoreServices.framework

libz.dylib

4 第三方库

本 SDK 使用到的第三方类库包括以下部分：

Reachability

[ASIHttpRequest](#)

[GHKit](#)

[tbxml](#)

5 使用步骤

1) 将 Aliyun Open Service SDK 和第三方库的源代码或 framework 加入到你的应用工程中，如图 5-1、5-2 所示：

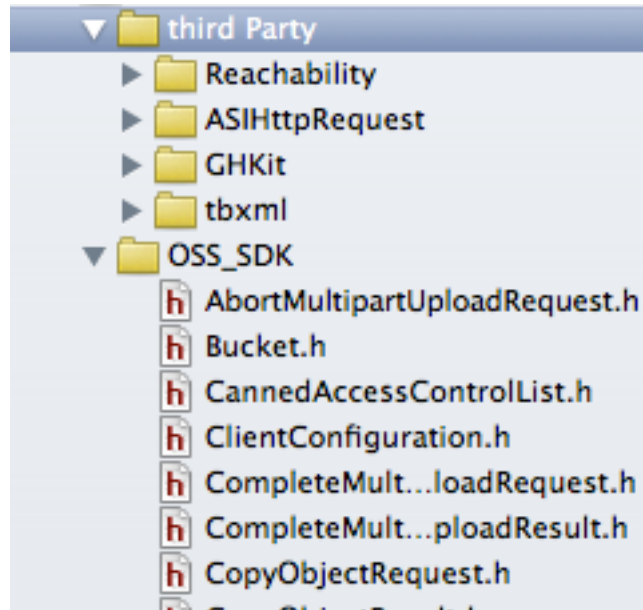


图 5-1 将源代码加入到工程中

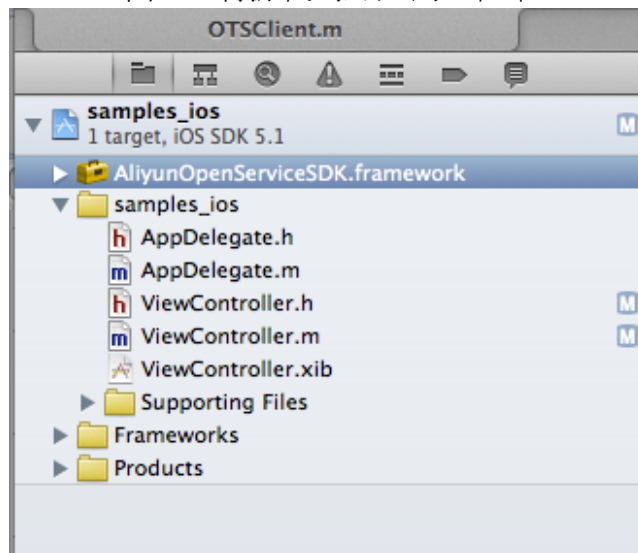


图 5-2 将 framework 加入到工程中

在开发中既可以使用源代码的方式也可以使用 framework 的方式来使用本 SDK。

2) 添加依赖库的引用，将 3 中提到的依赖添加到工程依赖中,依赖库列表如下：

Foundation.framework

CFNetwork.framework

SystemConfiguration.framework

MobileCoreServices.framework

libz.dylib

添加后如图 5-3 所示：

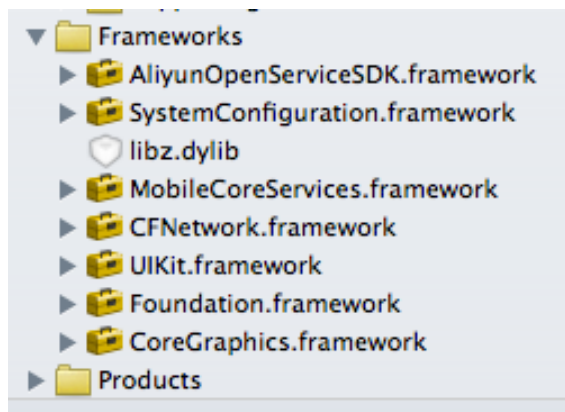


图 5-3 添加依赖库

3) 在 build setting 选项中将 Other linker Flags 设为-ObjC, 如图 5-4,

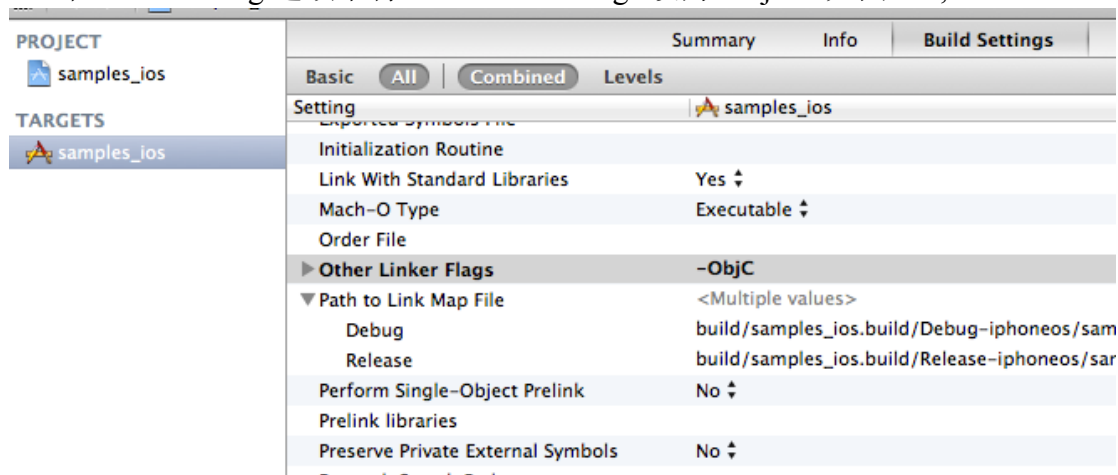


图 5-4 添加编译选项

4) 使用 Aliyun Open Service SDK 进行 OSS 对象操作

4.1 在需要使用 OSS 功能的文件中包含 OSS 头文件

```
#import <AliyunOpenServiceSDK/OSS.h>
```

4.2 声明 OSSClient 类

```
OSSClient * _client;
```

4.3 初始化 client 对象

```
_client = [[OSSClient alloc] initWithAccessId:accessid  
andAccessKey:accesskey];
```

其中 accessid 为 OSS 访问需要的 accessid, accesskey 为 accessid 对应的 key。

accessid 和 accesskey 的获取请参考 OSS 帮助文档。

4.4 设置 client 对象的代理, 用来响应 client 的请求结果

类实现 OSSClientDelegate 协议

```
@interface ViewController ()<OSSClientDelegate>
```

将 delegate 设为 self

```
_client.delegate = self;
```

4.5 使用 client 的方法进行 OSS 对象操作, 例如创建 bucket

```
[_client createBucket:@"barrycc11"];
```

4.6 实现 OSSClientDelegate 中关于 createBucket 的方法, 主要是创建成功和创建失败的方法, 用来获取请求的结果。

```
-(void)bucketCreateFinish:(OSSClient*) client  
result:(Bucket*) bucket
```

```
{
    NSLog(@"%@", bucket);
}
```

```
-(void)bucketCreateFailed:(OSSClient*) client
error:(OSSError*) error
{
    NSLog(@"%@", error.errorMessage);
}
```

其他 OSS 对象的操作同样使用上面的流程进行处理。

关于其他方法的使用请参考示例工程和帮助文档。

5) 使用 Aliyun Open Service SDK 进行 OTS 对象操作

5.1 在需要使用 OSS 功能的文件中包含 OSS 头文件

```
#import <AliyunOpenServiceSDK/OTS.h>
```

5.2 声明 OTSClient 类

```
OSSClient * _otsClient;
```

5.3 初始化 client 对象

```
_otsClient = [[OTSClient alloc] initWithAccessId:accessid
andAccessKey:accesskey];
```

其中 accessid 为 OTS 访问需要的 accessid, accesskey 为 accessid 对应的 key。

accessid 和 accesskey 的获取请参考 OTS 帮助文档。

5.4 设置 client 对象的代理, 用来响应 client 的请求结果

类实现 OTSClientDelegate 协议

```
@interface ViewController ()<OTSClientDelegate>
```

将 delegate 设为 self

```
_otsClient.delegate = self;
```

5.5 使用 client 的方法进行 OTS 对象操作, 例如创建 tableGroup

```
[_otsClient createTableGroup:@"bc3"
partitionKeyType:PartitionKeyType_STRING];
```

5.6 实现 OTSClientDelegate 中关于 createTableGroup 的方法, 主要是创建成功和创建失败的方法, 用来获取请求的结果。

```
-(void) OTSClientCreateTableGroupFailed:(OTSClient *)client error:(OTSError
*)error
{

}

-(void) OTSClientCreateTableGroupFinished:(OTSClient *)client
result:(CreateTableGroupResult *)result
{

}
```