iOS评论称赞集成指南

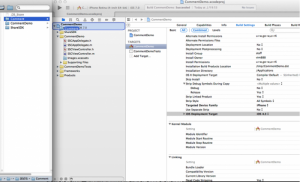
注意：在集成评论称赞SDK前，请先完成ShareSDK的集成步骤，详细请参考：《iOS快速集成指南》

将评论称赞SDK加入到工程中

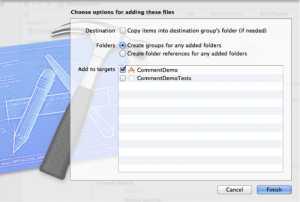
（如果是使用CocoaPods只需要在Podfile文件中添加

pod 'CommentSDK'即可）

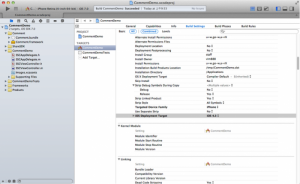
直接拖动Comment目录到工程中。如下图：

[](http://analytics.sharesdk.cn/docs/wp-content/uploads/2014/09/700px-ios_cmt_doc_001.png)

拖到工程中后，弹出以下对话框，勾选”Copy items into destination group’s folder(if needed)”，并点击“Finish“按钮, 如图：

[](http://analytics.sharesdk.cn/docs/wp-content/uploads/2014/09/700px-ios_cmt_doc_002.png)

导入后最终效果如下图：

[](http://analytics.sharesdk.cn/docs/wp-content/uploads/2014/09/700px-ios_cmt_doc_003.png)

显示评论列表

先导入评论称赞的头文件：

#import <Comment/Comment.h>

* 当前类为UIViewController类型及子类时，可以调用如下代码弹出评论列表。

[self presentCommentListViewControllerWithContentId:@"topic\_1" title:@"文章标题" animated:YES];

其中contentId为主题的标识，此标识在整个应用系统中必须要唯一，否则会导致多个主题评论列表混淆。title参数为主题标题。

* 当前的ViewController包含在UINavigationController中，还可以调用如下代码，push一个评论列表:

[self pushCommentListViewControllerWithContentId:@"topic\_1" title:@"标题" animated:YES];

* 当前类为非UIViewController类型时，可以调用如下代码先创建一个评论列表视图控制器：

SSCCommentListViewController \*vc = [Comment commentListViewPageWithContentId:@"topic\_1" title:@"标题"];

取得视图控制器后可以根据需要定制评论列表的显示。

最终显示效果如下图所示：

[](http://analytics.sharesdk.cn/docs/wp-content/uploads/2014/09/320px-ios_cmt_doc_004.png)

显示评论工具栏

先导入评论称赞头文件：

#import <Comment/Comment.h>

* 当前类为UIViewController类型及子类时，可以调用如下代码显示评论工具栏。

[self showCommentToolbarWithContentId:@"topic\_1" title:@"标题"];

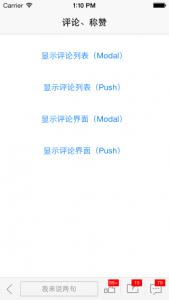
其中contentId为主题的标识，此标识在整个应用系统中必须要唯一，否则会导致多个主题评论列表混淆。title参数为主题标题。

* 当前类为非UIViewController类型时，可以调用如下代码创建工具栏：

SSCCommentToolbar \*toolbar = [Comment commentToolbarWithContentId:@"22" title:@"标题" frame:CGRectMake(0.0, 0.0, 320, 44)];

取得工具栏引用后，可以放入任意视图中进行显示。

最终显示效果如下图所示：

[](http://analytics.sharesdk.cn/docs/wp-content/uploads/2014/09/320px-ios_cmt_doc_006.png)

显示编辑评论视图

先导入评论称赞头文件：

#import <Comment/Comment.h>

* 当前类为UIViewController类型及子类时，可以调用如下代码弹出评论列表。
* [self presentCommentViewControllerWithContentId:@"123456"
* title:@"文章标题"
* comment:nil
* animated:YES
* result:^(SSCResponseState state, id<ISSCComment> comment, NSError \*error) {
* if (state == SSCResponseStateSuccess)
* {
* NSLog(@"评论成功!");
* }
* else if (state == SSCResponseStateFail)
* {
* NSLog(@"评论失败! %@", error);
* }

}];

其中contentId为主题的标识，此标识在整个应用系统中必须要唯一，否则会导致多个主题评论列表混淆。title参数为主题标题。comment参数为被评论的评论信息，如果传入nil则表示对主题进行评论。result就是评论返回的回调状态。

* 当前的ViewController包含在UINavigationController中，还可以调用如下代码，push一个评论编辑界面:
* [self pushCommentViewControllerWithContentId:@"123456"
* title:@"文章标题"
* comment:nil
* animated:YES
* result:^(SSCResponseState state, id<ISSCComment> comment, NSError \*error) {
* if (state == SSCResponseStateSuccess)
* {
* NSLog(@"评论成功!");
* }
* else if (state == SSCResponseStateFail)
* {
* NSLog(@"评论失败! %@", error);
* }

}];

参数意义与present方式一致。

* 当前类为非UIViewController类型时，可以调用如下代码先创建一个评论编辑视图控制器：
* SSCCommentViewController \*vc = [Comment commentViewWithContentId:@"123456"
* title:@"文章标题"
* comment:nil
* result:^(SSCResponseState state, id<ISSCComment> comment, NSError \*error) {
* if (state == SSCResponseStateSuccess)
* {
* NSLog(@"评论成功!");
* }
* else if (state == SSCResponseStateFail)
* {
* NSLog(@"评论失败! %@", error);
* }

}];

最终显示效果如下图所示：

[](http://analytics.sharesdk.cn/docs/wp-content/uploads/2014/09/320px-ios_cmt_doc_005.png)