

iOS Customer Service SDK Access Documentation

Imprint

Version	modify the content	Time
1.3.4	The customer service system supports sending video	2022.01.24
1.3.3	Add customer service multi-language setting interface	2021.12.07
1.3.2	Optimize local notification of customer service messages	2021.10.27
1.3.1	New version of customer service system	2021.05.18

一、SDK structure

Supported platforms: iPod Touch, iPhone, iPad. System requirements: iOS8.0+,

Support Framework: armv7 arm64

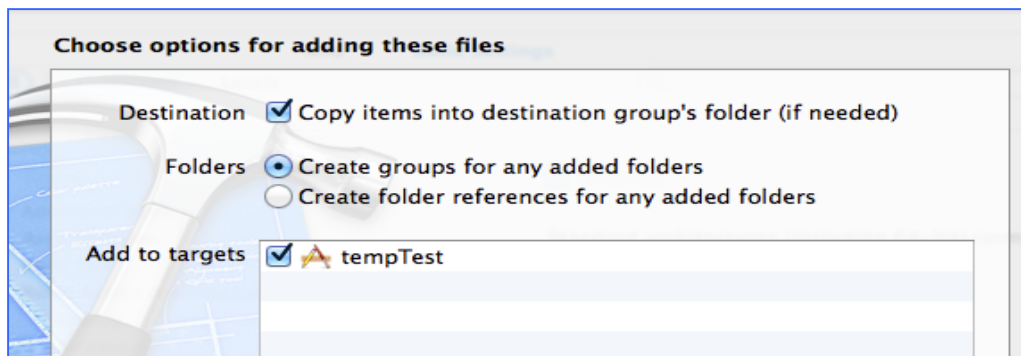
Environment requirements: xcode13.0+

二、Environment construction

Add SDK

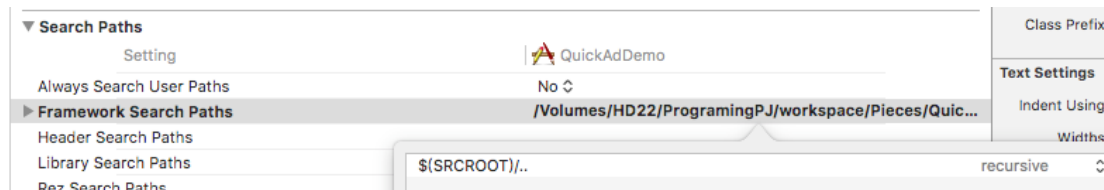
Drag and drop the SDK files CsSDK.frameworkt and CsRes.bundle into the project;

choose the correct target



Configure –ObjC in Build Settings->Linking->Other Linking Flags

If the framework header file is not found when compiling, you need to manually configure the framework search path, such as the relative path configuration in the demo



add system framework: WebKit.framework, JavaScriptCore.framework.

SDK permission application

Set Read Album and Use Camera permission description in Info.plist

NSCameraUsageDescription : Need to use camera to send picture information

to customer service

NSPhotoLibraryUsageDescription : Need to use photo album to send picture

information to customer service

NSPhotoLibraryAddUsageDescription : Need to use photo album to send

picture information to customer service

Precautions:

The customer service system only supports vertical screen, so it is necessary to

allow the window of the application to support vertical screen, which needs to

be set in AppDelegate: Implement or modify (this method has been added for

the project) the following system callback method and return to

omnidirectional : UIInterfaceOrientationMaskAll

```
- (UIInterfaceOrientationMask)application:(UIApplication
*)application supportedInterfaceOrientationsForWindow:(nullable
UIWindow *)window
{
    return UIInterfaceOrientationMaskAll;
}
```

If it is a project exported by Unity, you need to make the same settings as above in `UnityAppController.mm`

三、Interface Description

Set the corresponding language and AppId of the customer service (required)

```
/**
 *Customer service system: Set the appid and customer service language. This
 interface can be called multiple times to set multiple appids to support different
 languages. One appid corresponds to one language. The SDK will automatically match
 the current language of the device to choose which appid to use. At least you
 need to set a language
 *@param language appid corresponds to the customer service language, (Simplified
 Chinese@"zh-cn", Chinese Traditional Hong Kong@"zh-hk", Chinese Traditional
 Taiwan@"zh-tw", German@"de", English@"en-us", French@"fr", Japanese@"ja-jp",
 Korean@"kor", Russian@"ru", Thai@"th")
 *@param appid Customer service appid parameter, the parameter is applied in
 the customer service SDK background
 */
- (void)setLanguage:(NSString *)language appid:(NSString *)appid;
```

Open the customer service interface (required)

```
// Open customer service, upload parameters, please check CSParamModel.h
- (void)enterServiesystem:(CSParamModel *)param;
```

Local notification to receive new customer service messages (required)

```
/**
 * Receive local notifications when new customer service messages are available
 * call in - (void)application:(UIApplication *)application
 didReceiveLocalNotification:(nonnull UILocalNotification *)notification;
 * @param application application
 * @param notification notification
 */
- (void)application:(UIApplication *)application
 didReceiveLocalNotification:(nonnull UILocalNotification *)notification;
```

Sign up for Apple Remote Notifications (optional)

```
/**
 * Register Apple Remote Notification APNS for remote notification push messages,
 if there is no remote push, you can not call
 * call in - (BOOL)application:(UIApplication *)application
 didFinishLaunchingWithOptions:(NSDictionary *)launchOptions;
 */
- (void)registerForRemoteNotifications;
```

Get Apple Remote Push DeviceToken (optional)

```
/**
 * Get the deviceToken returned by Apple for receiving remote push notifications,
 if there is no remote push, you can not call
 * call in - (void)application:(UIApplication *)application
 didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken;
```

```
*@param deviceToken :deviceToken returned by apple
*/
- (void)didRegisterForRemoteNotificationsWithDeviceToken:(NSData
*)deviceToken;
```

Open or hide the floating window (optional)

```
// Show customer service floating menu
- (void)showMenu;
// Hide customer service floating menu
- (void)hideMenu;
```

Refer to the code in the demo and the interface description in the framework header file.

CSParamModel.h

Name	meaning	note
uId	User id	Required
userName	Can be set with uId	Required
userNickNameId	userNickNameId	Required
userNickName	Can be set with userNickNameId	Required
userNickNameNote	Comments for incoming nickname	Required
vipLevel	Game user Vip level	Required
userBalance	User balance	Required