

# Hướng dẫn tích hợp Viettalk vào ứng dụng

## Version 0.1



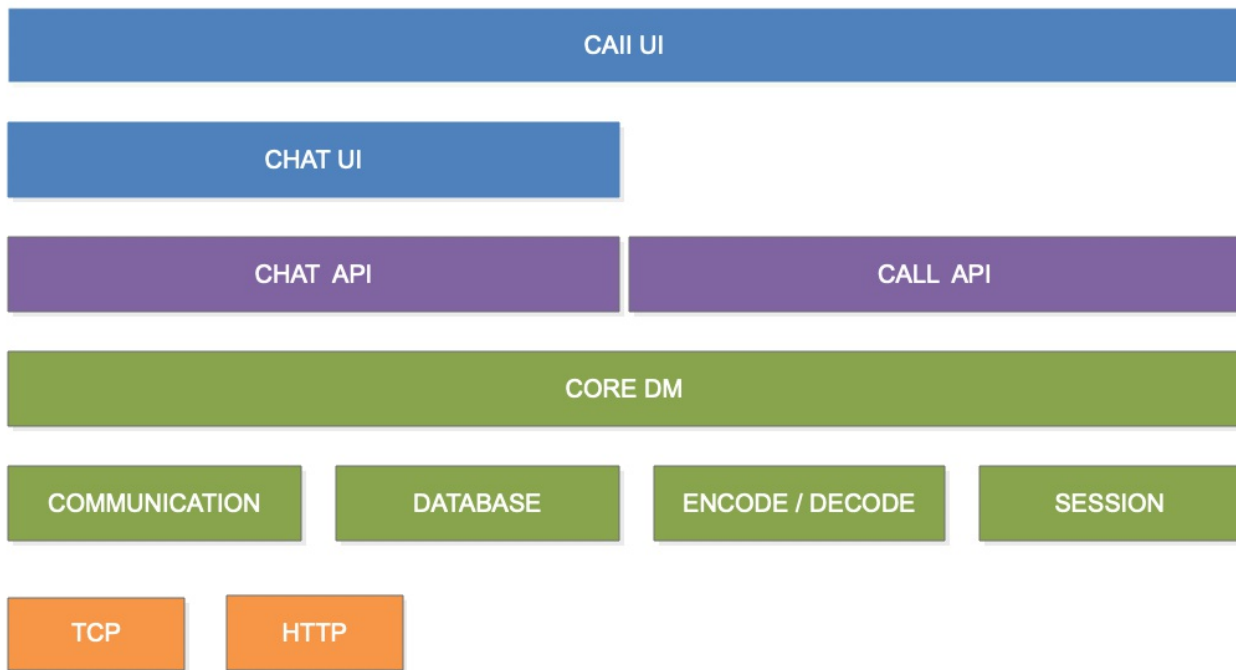
build passing

Tài liệu hướng dẫn tích hợp chức năng chat và gọi điện của **VIETTALK** vào ứng dụng.

## Chức năng chính!

- API kết nối đến hệ thống chat
- API kết nối tới hệ thống voip
- Giao diện danh sách cuộc hội thoại
- Giao diện hội thoại chat đơn và nhóm
- giao diện nhận và trả lời cuộc gọi đến
- giao diện gọi đi

## Kiến trúc chung



## SDK gồm có!

- **CoreDM.framework** ==> chứa kiến trúc database và giao tiếp kết nối tới server
- **ProtoModel.framework** ==> chứa định nghĩa model
- **VietTalkCore.framework** ==> Chứa api chat và báo hiệu
- **VTCallAPI.framework** ==> chứa những api Liên quan VOIP
- **ViettalkUI.framework** ==> chứa giao diện chat
- **ViettalkCallUI.framework** ==> chứa giao diện gọi điện

## ProtoModel.framework Dependency

Framework Name	Link	Version
protobuf-swift	<a href="https://github.com/alexeyxo/protobuf-swift">https://github.com/alexeyxo/protobuf-swift</a>	v4.0.6

	swift	
--	-------	--

## VietTalkCore.framework Dependency

Framework Name	Link	Version
CoreDM	cung cấp cùng thư viện	0.1
PhoneNumberKit	<a href="https://github.com/marmelroy/PhoneNumberKit">https://github.com/marmelroy/PhoneNumberKit</a>	2.6

## VTCallAPI.framework Dependency

Framework Name	Link	Version
VietTalkCore	cung cấp cùng thư viện	0.1
AudioToolBox	System framework	
AVFoundation	System framework	

## ViettalkUI.framework Dependency

Framework Name	Link
SwiftLinkPreview	<a href="https://github.com/LeonardoCardoso/SwiftLinkPreview">https://github.com/LeonardoCardoso/SwiftLinkPreview</a>
SnapKit	<a href="https://github.com/SnapKit/SnapKit">https://github.com/SnapKit/SnapKit</a>
Kingfisher	<a href="https://github.com/onevcats/Kingfisher">https://github.com/onevcats/Kingfisher</a>
SwipeCellKit	<a href="https://github.com/SwipeCellKit/SwipeCellKit">https://github.com/SwipeCellKit/SwipeCellKit</a>
CLImageEditorCarthage	<a href="https://github.com/DungntVccorp/CLImageEditorCarthage">https://github.com/DungntVccorp/CLImageEditorCarthage</a>
Pulsator	<a href="https://github.com/shu223/Pulsator">https://github.com/shu223/Pulsator</a>

SKPhotoBrowser	<a href="https://github.com/suzuki-0000/SKPhotoBrowser">https://github.com/suzuki-0000/SKPhotoBrowser</a>
MBProgressHUD	<a href="https://github.com/jdg/MBProgressHUD">https://github.com/jdg/MBProgressHUD</a>
TOCropViewController	<a href="https://github.com/TimOliver/TOCropViewController">https://github.com/TimOliver/TOCropViewController</a>
THContactPickerCart	<a href="https://github.com/DungntVccorp/THContactPickerCart">https://github.com/DungntVccorp/THContactPickerCart</a>
Floaty	<a href="https://github.com/kciter/Floaty">https://github.com/kciter/Floaty</a>
DZNEmptyDataSet	<a href="https://github.com/dzenbot/DZNEmptyDataSet">https://github.com/dzenbot/DZNEmptyDataSet</a>
Google Map SDK	<a href="https://developers.google.com/maps/documentation-ios/sdk/intro">https://developers.google.com/maps/documentation-ios/sdk/intro</a>
Google Places SDK	<a href="https://developers.google.com/places/ios-sdk/">https://developers.google.com/places/ios-sdk/</a>

## ViettalkUI.framework Dependency

Framework Name	Link	Version
ViettalkUI	cung cấp cùng thư viện	0.1

## Yêu cầu hệ thống

- iOS 9.0+ / Mac OS X 10.14+
- Xcode 10.1+
- Swift 4.2+
- C++11
- armv7s arm64 arm64e
- iPhone 5s trở lên ( không hỗ trợ giao diện cho Ipad)
- không hỗ trợ IOS đã bị **Jailbreak**

# Cài Đặt

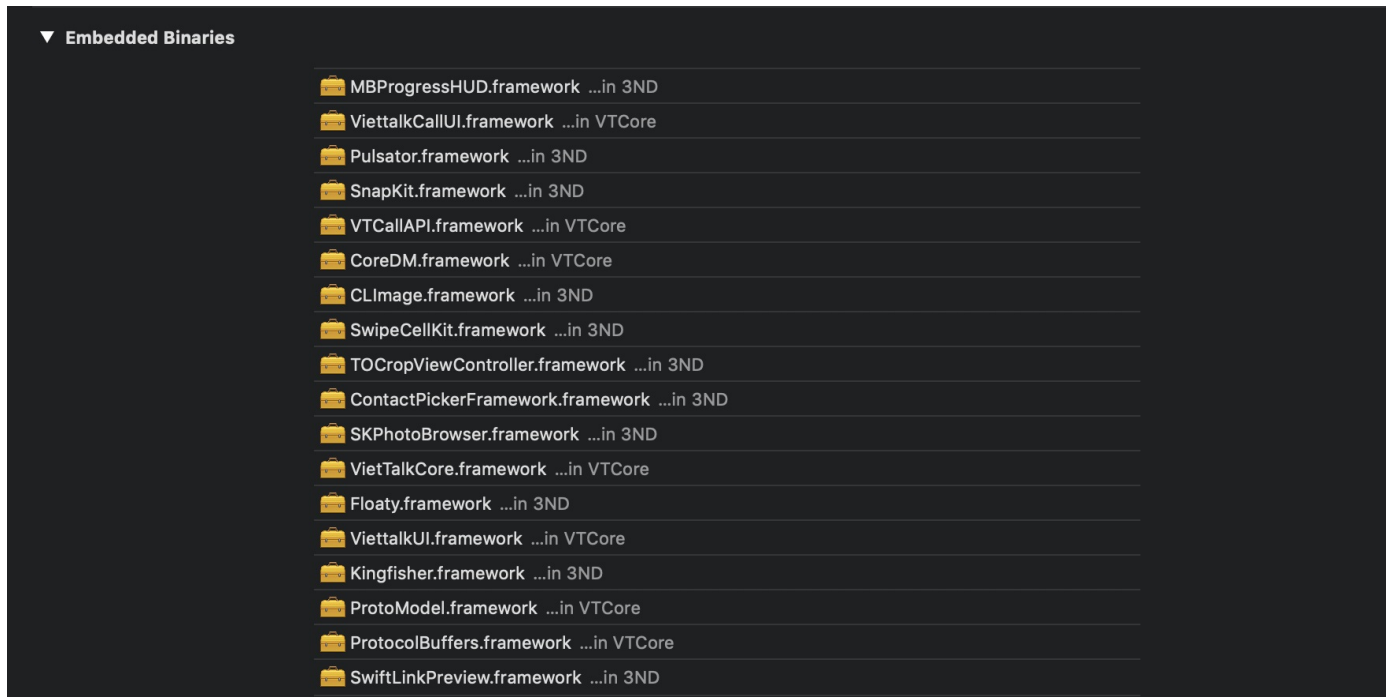
- Kéo thả những framework sau vào **Linked Frameworks And Libraries** trong **TARGETS** project

```
CoreDM.framework
ProtocolBuffers.framework
ProtoModel.framework
ViettalkCallUI.framework
VietTalkCore.framework
ViettalkUI.framework
VTCallAPI.framework
```

## 3rd Framework

```
CLImage.framework
ContactPickerFramework.framework
CropViewController.framework
Kingfisher.framework
MBProgressHUD.framework
Pulsator.framework
SKPhotoBrowser.framework
SnapKit.framework
SwiftLinkPreview.framework
SwipeCellKit.framework
TOCropViewController.framework
Floaty.framework
```

## DZNEEmptyDataSet.framework



- Nếu sử dụng cho **Objective C** Project

Always Embed Swift Standard Libraries = "YES"



ở trong **Build Settings**

- Cấu hình File Info.plist

```
viettalk_is_dev_mode = "YES / NO"
viettalk_token = "token_duoc_cung_cap_theo_Bundle_identifi
er"
NSCameraUsageDescription = "str_description"
NSLocationWhenInUseUsageDescription = "str_description"
NSMicrophoneUsageDescription = "str_description"
```

```
NSPhotoLibraryUsageDescription = "str_description"
```

viettalk_token	⌵	String	1234567890
viettalk_is_dev_mode	⌵	Boolean	YES

chú ý token được cung cấp theo ID của ứng dụng

- Cài đặt Google Map và Google Map Place theo hướng dẫn

## GOOGLE MAP

## GOOGLE MAP PLACES

# Cấu Hình Sử dụng trên Project Objective C

- ở class AppDelegate import

```
@import GoogleMaps;  
@import GooglePlaces;  
@import VietTalkCore;  
@import ViettalkUI;  
@import ViettalkCallUI;  
#import <VTCallAPI/ViettalkCallAPI.h>
```

- Trong file AppDelegate.m thêm code ở trong hàm **didFinishLaunchingWithOptions**

```
[GMSServices provideAPIKey:@"map_key"];
[GMSPPlacesClient provideAPIKey:@"map_key"];
[[ViettalkAPI shared] setViettalkAPIDelegateWithDelegat
e:self];
```

- Trong file AppDelegate.m thực thi hàm delegate của **ViettalkCallAPIDelegate** và **ViettalkAPIDelegate**

```
-(void)onReceiveMessageWithForm:(NSData *)ChatID :(NSData
*)messageID{

}

-(void)onReceiveState:(NSInteger)state error:(NSString *)e
rror{
    if(state == ViettalkApiStateAuthen_error){
        /// RETRY AUTHEN
        if([error isEqualToString:@""] == NO){
            UIAlertController *alr = [UIAlertController al
ertControllerWithTitle:nil message:error preferredStyle:UI
AlertControllerStyleAlert];

            [alr addAction:[UIAlertAction actionWithTitle:
@"Đóng" style:UIAlertActionStyleDefault handler:nil]];

            [alr addAction:[UIAlertAction actionWithTitle:
@"Thử lại" style:UIAlertActionStyleDefault handler:^(UIAle
rtAction * _Nonnull action) {

                }]];
        }
    }
}
```



```

        [self.window.rootViewController presentViewController:alr animated:YES completion:nil];

    }

    }else if(state == ViettalkApiStateAuthen_success){

        /// CẤU HÌNH CALL NẾU SỬ DỤNG CALL TRONG ỨNG DỤNG
        NSDictionary *data_login = [[ViettalkAPI shared] get_user_login];

        NSString *sipDomain = data_login[@"sipDomain"];
        NSString *sipPassword = data_login[@"sipPassword"];
        ;
        NSString *sipProxy = data_login[@"sipProxy"];
        NSString *sipUserName = data_login[@"sipUserName"];
        ;

        [[ViettalkCallAPI sharedInstance] setDelegate:self];

        [[ViettalkCallAPI sharedInstance] setupAccount:sipUserName password:sipPassword domain:sipDomain proxy:sipProxy];


        UIAlertController *alr = [UIAlertController alertControllerWithTitle:nil message:@"Đăng nhập hệ thống chat t hành công" preferredStyle:UIAlertControllerStyleAlert];

        [alr addAction:[UIAlertAction actionWithTitle:@"Đóng" style:UIAlertActionStyleDefault handler:nil]];

        [self.window.rootViewController presentViewController:alr animated:YES completion:nil];
    }
}

```

```

    }else if(state == ViettalkApiStateAuthen_failure){

    }

}

-(void)onReceiveNewIncomingCall:(NSDictionary *)newCall{

    if ([[ViettalkUI shared] canMakeCall] == YES) {
        [[ViettalkUI shared] showIncommingCall:self.window
.rootViewController call_info:newCall];
    }
}

```

- Trong file AppDelegate.m bổ sung sử dụng trạng thái của ứng dụng

```

- (void)applicationWillResignActive:(UIApplication *)application {
    [[ViettalkAPI shared] applicationWillResignActive:application];
}

- (void)applicationDidEnterBackground:(UIApplication *)application {
    [[ViettalkAPI shared] applicationDidEnterBackground:application];
}

- (void)applicationWillEnterForeground:(UIApplication *)ap

```

```

plication {
    [[ViettalkAPI shared] applicationWillEnterForeground:a
pplication];
}
- (void)applicationDidBecomeActive:(UIApplication *)applic
ation {
    [[ViettalkAPI shared] applicationDidBecomeActive:appli
cation];
}
- (void)applicationWillTerminate:(UIApplication *)applicat
ion {
    [[ViettalkAPI shared] applicationWillTerminate:applica
tion];
}

```

## Sử dụng trên Objective C

- Để Hiển thị danh sách cuộc hội thoại cũ

```

if([[ViettalkUI shared] canShowChatHistory] == YES){
    [[ViettalkUI shared] showChatHistory:view_controll
er isPush:NO];
}

```

- Để hiển thị chi tiết một cuộc hội thoại hoặc tạo mới một cuộc hội thoại

```

if([[ViettalkUI shared] canShowMessage] == YES){

    [[ViettalkUI shared] showChatWithTel:self isPush:FALSE
tel:phone_number_string_84xxxxx nameDisplay:name_display_
string];
}

```

- Để tạo 1 group chat mới

```

[[ViettalkAPI shared] createChatWorkSpaceWithProfiles:new_
profile_list isGroup:isG onSuccess:^(Chat *result_create_c
hat) {

    [[ViettalkUI shared] showChat:self isPush:FALS
E chat:result_create_chat];

    } onError:^(id er) {
        NSLog(@"%@ ",er);
    } onTimeout:^(id timeout) {
        NSLog(@"TIME OUT");
    }];
}

```

- Để thực hiện call đến 1 số

```

if([[ViettalkUI shared] canMakeCall] == YES){

    [[ViettalkUI shared] showMakeCall:self tel:phone_n
umber_string_84xxxxx name:name_display_string];

}

```

# Sử dụng trên Project Swift Project

## Todos

- ☒ Demo trên objective C Project
- ☐ Demo trên swift project
- ☐ Hướng dẫn sử dụng và tài liệu API
- ☐ Hỗ trợ giao diện iPad
- ☐ Hỗ trợ push notification
- ☐ Hỗ trợ wake up call

## License

VIETTALK COPYRIGHT 2019 @ VIVAS - VNPT TECHNOLOGY