

# SeelKit 集成文档

## 接入指南

### 创建应用，获取 Appkey

集成【SeelKit】SDK 之前, 您首先需要到 【SeelKit】官网注册并且添加新应用, 获得 AppKey。

## 集成流程

### Cocoapods 自动集成（推荐）

#### 更新 pod 环境

在终端执行 pod setup 命令，拉取最新 pod 库时间较长。

#### 集成 SDK

在项目根目录的 Podfile 的文件中添加 pod 'SeelKit'

```
target 'SeelKitDemo' do

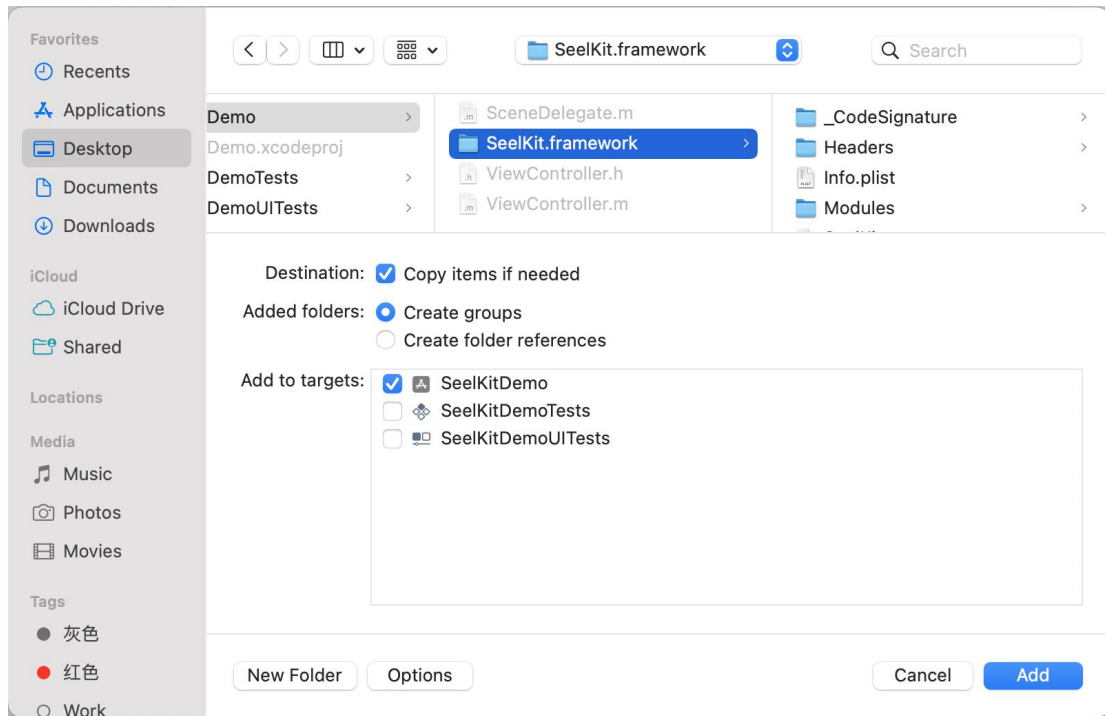
  pod 'SeelKit'

end
```

## 手动集成

#### 工程配置

- 1.在官网中下载 SeelKitSDK
- 2.XcodeFile —> Add Files to "Your Project", 在弹出 Panel 选中所下载 SDK —>Add。(注: 选中 “Copy items if needed” )



## SDK 使用

### 初始化

#### 1. 接口函数

```
/// 初始化sdk  
/// @param appKey 开发者在官网申请到的appKey  
- (void)startWithAppKey:(NSString *)appKey;
```

#### 2. 示例代码

```
#import <SeelKit/SeelKit.h>  
  
- (BOOL)application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {  
    [[SKManager shared] startWithAppKey:@"your appKey"];  
    return YES;  
}
```

# 添加 SeelView

## 1.接口函数

```
/// 将 SeelView 添加到你想要展示的 View 上
/// @param superView superView
/// @param pointX x
/// @param pointY y
/// @param viewWidth width
- (void)addToSuperView:(UIView *)superView pointX:(CGFloat)pointX
pointY:(CGFloat)pointY viewWidth:(CGFloat)viewWidth;
```

## 2.示例代码

```
- (SeelView *)seelView
{
    if (!_seelView) {
        _seelView = [SeelView new];
        _seelView.backgroundColor = [UIColor whiteColor];
        _seelView.delegate = self;///如需监听 Switch 开关，请设置代理，实现代理方法
        [_seelView addToSuperView:self pointX:16.f pointY:0.f
viewWidth:kScreenWidth - 32.f];
    }
    return _seelView;
}
```

## 3.实现 SeelViewDelegate

```
#pragma mark --- SeelViewDelegate
- (void)onSwitchChecked
{
    NSLog(@"打开");
}
- (void)onSwitchUnchecked
{
    NSLog(@"关闭");
}
```

## 4.可自定义内容

```
/// titleStr
@property (nonatomic, copy) NSString *titleStr;

/// titleFont
@property (nonatomic, strong) UIFont *titleFont;

/// titleColor
@property (nonatomic, strong) UIColor *titleColor;
```

```

/// describeAttributedText
@property (nonatomic, copy) NSAttributedString *describeAttributedText;

/// describeFont
@property (nonatomic, strong) UIFont *describeFont;

/// describeColor
@property (nonatomic, strong) UIColor *describeColor;

/// switchBackgroundColor default #DCDCDC
@property (nonatomic, strong) UIColor *switchBackgroundColor;

/// switchOnTintColor default #C8C5FF
@property (nonatomic, strong) UIColor *switchOnTintColor;

/// switchThumbTintColor default #645AFF
@property (nonatomic, strong) UIColor *switchThumbTintColor;

/// iconImage
@property (nonatomic, strong) UIImage *iconImage;

/// priceFont
@property (nonatomic, strong) UIFont *priceFont;

/// priceColor
@property (nonatomic, strong) UIColor *priceColor;

```

## 获取保单报价

### 1.接口函数

```

/// 获取报价
/// @param params NSDictionary 具体传什么参数，需与后端开发人员沟通
/// @param completion success:是否成功 price: 保单价格 error: error
- (void)creatQuotesParams:(NSDictionary *)params completion:(void (^)(BOOL success,NSString *price, NSError*))completion;

```

### 2.示例代码

```

[[SKManager shared] creatQuotesParams: @{
    @"quote_type":@"bra-sidelineSwap",
    @"quote_info":@{

```

```

        @"cart_id":@"cart_12345",
        @"request_timestamp_in_mills":@12322333423,
        @"items":@[
            @{
                @"item_id":@"5821444812",
                @"quantity":@1,
                @"price":@34.33,
                @"original_price":@34.33,
                @"variant_id":@"23123",
                @"product_id":@"12344",
                @"title":@"title_1",
                @"description":@"description_1",
                @"color":@"red",
                @"size":@"s",
                @"brand_id":@"21",
                @"brand_name":@"brand_1",
                @"category":@"category_1"
            }
        ],
        @"customer_info":@{
            @"session_id":@"347d24413737713f326719ad450263ba"
        }
    }
}

completion:^(BOOL success, NSString * _Nonnull price, NSError * error) {
    if (success) {
        NSLog(@"%@", price);
    } else {
        NSLog(@"%@", error);
    }
}
}];

```

## 创建保单

### 1.接口函数

```

/// 创建保单
/// @param params NSDictionary 具体传什么参数，需与后端开发人员沟通
/// @param completion success: 是否成功 policies: 成功后返回保单内容 error: error
- (void)creatPoliciesParams:(NSDictionary *)params completion:(void (^)(BOOL success,NSDictionary *policies,NSError *error))completion;

```

### 2.示例代码

```

[[SKManager shared] creatPoliciesParams: @{

```

```

        @"quote_id": @"2022659951102665",
        @"policy_type": @"sra-garmentory",
        @"order_info": @{
            @"order_id": @"1231659942324234",
            @"order_ts": @"2022-05-05T02:52:29.150000Z",
            @"policy_tax": @4.99,
            @"item_tax": @[
                @{
                    @"item_id": @"2222",
                    @"tax": @4.22
                }
            ]
        }
    } completion:^(BOOL success, NSDictionary * _Nonnull policies, NSError *
_Nonnull error) {
        if (success) {
            NSLog(@"%@", policies);
        } else {
            NSLog(@"%@", error);
        }
    }
};

```

## 自定义监听

### 1.接口函数

```

/// 监听
/// @param event_type 监听类型
/// @param event_data 监听数据
- (void)listenerWithEvent_type:(NSString *)event_type event_data:(NSDictionary
*)event_data;

```

### 2.事例代码

```

[[SKManager shared] listenerWithEvent_type:@"Log" event_data:@{
    @"test" : @"test"
}];

```

## Swift 集成

## iOS OC 和 Swift 混编文档

Swift 调用 Objective-C 需要建立一个桥接头文件进行交互

## 新建桥接头文件

- 1.该文件在第一次创建 Swift 文件时会提示创建该文件，选择创建。
  - 2.该文件可以手动创建，选择 Header File 文件创建
- 桥接命名格式为 工程名-Bridging-Header.h

```
SeelKitDemo > SeelKitDemo > h SeelKitDemo-Bridging-Header > No Selection
1 //
2 // SeelKitDemo-Bridging-Header.h
3 // SeelKitDemo
4 //
5 // Created by CP on 2022/11/8.
6 //
7
8 #ifndef SeelKitDemo_Bridging_Header_h
9 #define SeelKitDemo_Bridging_Header_h
10
11 #import <SeelKit/SeelKit.h>
12
13 #endif /* SeelKitDemo_Bridging_Header_h */
14
```

## 配置信息

将桥接文件路径拷到 Objective-C Bridging Header 里  
设置 Precompile Bridging Header 为 Yes

## 导入 SDK 头文件

在工程名-Bridging-Header.h 中加入 SDK 的头文件

```
#ifndef SeelKitDemo_Bridging_Header_h
#define SeelKitDemo_Bridging_Header_h

#import <SeelKit/SeelKit.h>

#endif /* SeelKitDemo_Bridging_Header_h */
```