

IOS SDK

- [Общие сведения](#)
- [Системные требования](#)
- [Описание фреймворка AssistMobile](#)
- [Порядок интеграции мобильного приложения интернет-магазина с фреймворком AssistMobile](#)
- [Примеры реализации оплаты с помощью фреймворка AssistMobile](#)
- [Интеграция с Apple Pay](#)

Общие сведения

Фреймворк AssistMobile позволяет разработать мобильное приложение интернет-магазина для проведения через АПК Ассист платежей банковской картой с мобильных устройств.

Пользователями фреймворка AssistMobile являются разработчики мобильного приложения интернет-магазина.

Настоящий документ содержит сведения по установке, настройке и использованию фреймворка AssistMobile, достаточные для его применения.

Системные требования

Для работы с AssistMobile необходимо использовать среду разработки iOS SDK версии **не ниже 8.4**.

Описание фреймворка AssistMobile

Фреймворк AssistMobile представляет собой файл *AssistMobile.framework*.

Для доступа к функционалу библиотеки нужно использовать объект класса **AssistPay**.

Класс содержит:

- инициализатор, принимающий поддерживающие протокол **AssistPayDelegate** объекты;
- метод **start**, принимающий объект класса **UIViewController** и объект **PayData** с данными об оплате;
- метод **getResult**, принимающий структуру с данными об оплате **PayData**.

Протокол **AssistPayDelegate** определяет метод **payFinished**, с помощью которого поддерживающий этот протокол объект будет получать результат оплаты. Метод **payFinished** будет вызван, если оплата завершена. Метод имеет следующие параметры:

- **bill** - номер транзакции в АПК Ассист в строковом формате;
- **status** - статус платежа;
- **message** - описание статуса.

Статус может иметь следующие значения:

- **Unknown** - статус неизвестен – вероятно, произошла непредвиденная ошибка и не удалось получить статус оплаты, через некоторое время следует вызвать метод **getResult** и дождаться возвращения статуса; если ошибка повторяется, то необходимо связаться со службой поддержки Assist;
- **InProgress** - оплата находится в обработке, через некоторое время следует повторно вызвать метод **getResult**, для получения результата;
- **Delayed** - платеж отложен, используется при включении функции отложенных платежей;
- **Approved** - платеж завершен успешно;
- **PartialApproved** - платеж завершен успешно на неполную сумму, используется при включении функции подтверждения платежа на неполную сумму;
- **PartialDelayed** - платеж частично отложен, используется при включении специальной функции;
- **Canceled** - платеж отменен;
- **PartialCanceled** - платеж частично отменен;
- **Declined** - оплата отклонена;
- **Timeout** - оплата завершена по таймауту;
- **Repeated** - в системе есть платеж с таким же номером заказа.

Метод **start** позволяет инициировать процесс оплаты, принимает параметры:

- **UIViewController** - контроллер вызывающего приложения;
- **PayData** - данные оплаты.

Метод **getResult** позволяет получить результат оплаты из АПК Ассист. Принимает параметр **PayData**.

- **PayData** - объект этого класса содержит данные платежа.

Данные платежа задаются свойствами объекта:

Название	Обязательное поле	Тип	Описание
merchantId	Да	String	Идентификатор предприятия в системе АПК Ассист
customerId	Да/нет	String	Идентификатор покупателя, используется при подключении OneClick
orderIdNumber	Да	String	Номер заказа
language	Нет	Language	Язык авторизационных страниц (RU, EN, BY или UK)
languageStr	Нет	String	Язык авторизационных страниц в строковом формате, используются значения из перечисления Language
orderAmount	Да	String	Сумма платежа в оригинальной валюте (например, 100.99).
orderComment	Нет	String	Комментарий
orderCurrency	Да/нет	Currency	Валюта (RUB, USD, EUR, BYR или UAH)
orderCurrencyStr	Нет	String	Валюта в строковом формате, используются значения из перечисления Currency
lastName	Нет	String	Фамилия покупателя
firstName	Нет	String	Имя покупателя
middleName	Нет	String	Отчество покупателя
email	Нет	String	E-mail адрес покупателя
address	Нет	String	Адрес покупателя

homePhone	Нет	String	Домашний телефон покупателя
workPhone	Нет	String	Рабочий телефон покупателя
mobilePhone	Нет	String	Мобильный телефон покупателя
fax	Нет	String	Факс покупателя
country	Нет	String	Страна покупателя
state	Нет	String	Код региона покупателя
city	Нет	String	Город покупателя
zip	Нет	String	Индекс предприятия связи покупателя
signature	Да/нет	String	Подпись. Формируется строка, сшитая из следующих параметров заказа: MerchantId;OrderNumber;OrderAmount;OrderCurrency , разделенных точкой с запятой. На базе этой строки алгоритмом MD5 формируется дайджест. Дайджест подписывается закрытым RSA ключом мерчанта. Длина ключа - 1024. Полученная байтовая последовательность является подписью магазина. Подпись передается нам в виде строки BASE64.

Порядок интеграции мобильного приложения интернет-магазина с фреймворком AssistMobile

Для интеграции мобильного приложения интернет-магазина с фреймворком AssistMobile необходимо выполнить следующие действия:

- скачать актуальную версию *AssistMobile.framework*;
- добавить фреймворк в разрабатываемый проект, перетаскив фреймворк в Xcode в дерево проекта;
- установить в свойствах проекта *Build Settings* -> *Build Options* -> *Embedded Content Contains Swift Code* значение «Yes»;
- добавить *AssistMobile.framework* в свойствах *General*->*Embedded Binaries*;
- добавить ключ *NSLocationWhenInUseUsageDescription* в файл *Info.plist*, в *Information Property List*, тип *String*, значение - строка "Позволить отправить данные геолокации в АПК Ассист".

Примеры реализации оплаты с помощью фреймворка AssistMobile

Пример кода Swift

```

import UIKit
import AssistMobile

class ViewController: UIViewController, AssistPayDelegate {
    @IBOutlet weak var result: UILabel!
    var data = PayData()

    @IBAction func startPay(sender: UIButton) {

        data.merchantId = "123456"
        data.orderNumber = "test_payment_01"
        data.orderAmount = "100.05"
        data.orderComment = "This is a test!"
        data.orderCurrency = .RUB
        data.lastname = "Ivanov"
        data.firstname = "Ivan"
        data.middlename = "Ivanovich"
        data.email = "i3@mail.ru"
        data.mobilePhone = "+79210000000"
        data.address = "Nevskiy prospekt, 1"
        data.country = "Russian Federation"
        data.state = "Saint-Petersburg"
        data.city = "Saint-Petersburg"

        let pay = AssistPay(delegate: self)
        pay.start(self, withData: data)
    }

    func payFinished(bill: String, status: PaymentStatus, message: String?) {
        let msg = message ?? ""
        result.text = "Finished: bill = \(bill), status = \(status.rawValue), message = \(msg)"
    }
}

```

Пример кода на Objective C

ViewController.h

```

#import <UIKit/UIKit.h>
#import "AssistMobile/AssistMobile.h"

@interface ViewController : UIViewController<AssistPayDelegate>
@property (weak, nonatomic) IBOutlet UILabel *result;
- (IBAction)pay:(UIButton *)sender;

@end

```

ViewController.m

```

#import "ViewController.h"
@interface ViewController ()

@property (strong, nonatomic) PayData* data;
@end

@implementation ViewController
- (IBAction)pay:(UIButton *)sender {
    if (!_data) {
        _data = [[PayData alloc] init];
    }

    _data.orderAmount = @"100.05";
    _data.orderNumber = @"test_payment_002";
    _data.merchantId = @"679471";

    AssistPay* assistPay = [[AssistPay alloc] initWithDelegate: self];
    [assistPay start:self withData: _data];
}

- (void)payFinished:(NSString * __nonnull)bill status:(NSString * __nonnull)status message:(NSString *
__nullable)message
{
    NSString* res = @"status: ";
    res = [res stringByAppendingString: status];
    res = [res stringByAppendingString: @"", billnumber: "];
    res = [res stringByAppendingString: bill];
    res = [res stringByAppendingString: @"", message: "];
    res = [res stringByAppendingString: message];
    _result.text = res;
}

@end

```

Интеграция с Apple Pay

1. Выполнить шаги из раздела 4 «Порядок интеграции мобильного приложения интернет-магазина с фреймворком AssistMobile».
2. Скомпилировать AssistMobile.framework из исходников.
3. Добавить фреймворк в свой проект.
4. Включить опцию Build Settings -> Build Options -> Embedded Content Contains Swift Code в YES в настройках проекта.
5. Добавить AssisyMobile.framework в настройку General->Embedded Binaries.
6. В ключ NSLocationWhenInUseUsageDescription в настройках Info.plist -> Information Property List, добавляем параметр типа String со значением 'Permit to send geolocation data to Assist.

Пример кода Swift

```

import UIKit
import AssistMobile

class ViewController: UIViewController, AssistPayDelegate {
    @IBOutlet weak var result: UILabel!
    var data = PayData()

    @available(iOS 10.0, *)
    @IBAction func payWithApplePay(_ sender: UIButton) {
        data = PayData()
        pay = AssistPay(delegate: self)
        data.merchantId = "your merchant id in assist"
        var apmid = "merchant.id.in.apple"
        data.login = "your account login in assist"
        data.password = "your account password in assist"
        data.orderNumber = "order number"
        data.orderComment = "comment"
        data.orderAmount = "100.05"
        data.orderCurrency = .RUB
        data.lastname = "Ivanov"
        data.firstname = "Ivan"
        data.email = "ivan@mailhost.ru"

        AssistLinks.currentHost = "https://payments.assist.ru"

        pay!.startWithApplePay(self, withData: data, applePayMerchantId: apmid)
    }

    func payFinished(bill: String, status: PaymentStatus, message: String?) {
        let msg = message ?? ""
        result.text = "Finished: bill = \(bill), status = \(status.rawValue), message = \(msg)"
    }
}

```

SDK и пример приложения доступны для скачивания по следующей ссылке:

<https://github.com/assist-group/assist-mcommerce-sdk-ios>

[Наверх](#)