

DBS Checkout SDK for iOS

SDK состоит из библиотеки ядра (**DBSCheckoutCore**) и библиотеки UI форм (**DBSCheckoutSDK**) в виде отдельных xcframeworks.

Возможности SDK

- Прием платежей карт **Visa, Mastercard, Humo, Uzcard** и **МИР**
- Отображение деталей заказа
- Прохождение оплат с использованием **3DS**
- Поддержка узбекской, русской и английской локализации
- Поддержка оплаты в сумах, рублях и долларах
- Отслеживание событий успешного проведения оплаты или ошибки с кодом
- Добавление новой карты (данные передаются зашифрованные с помощью асимметричного шифрования) или использование добавленных ранее для проведения платежа
- Возможность кастомизации шрифтов и цветовой схемы (индивидуально для мерчанта)
- Поддержка приложений с циклом управления на **UIKit** и **SwiftUI**
- Поддержка светлой и темной цветовых схем

Требования и ограничения

Для работы **DBS Checkout SDK** необходима iOS версии 13.0 или выше. Библиотека UI форм **DBSCheckoutSDK** может быть использована только с ядром **DBSCheckoutCore**.

Подключение

Через XCode и xcframeworks

SDK можно подключить с помощью непосредственного добавления **DBSCheckoutCore.xcframework** и **DBSCheckoutSDK.xcframework** в проект. Для этого нужно скачать архивы **DBSCheckoutCore.xcframework.zip** и **DBSCheckoutSDK.xcframework.zip** из **Release** секции **gitHub** репозитория, распаковать их

и добавить ссылки на библиотеки в секцию **Frameworks, Libraries and Embedded Content**:

PROJECT

SampleApp

TARGETS

SampleUIKit+SPM

SampleAppSUI+XCF

Build 1

+

Deployment Info

iPhone Orientation

☒ Portrait

☐ Upside Down

☒ Landscape Left

☒ Landscape Right

iPad Orientation

☒ Portrait

☒ Upside Down

☒ Landscape Left

☒ Landscape Right

Status Bar Style

Default

Requires full screen

Supports multiple windows

App Icons and Launch Screen

App Icon

AppIcon

+

App Icons Source

☐ Include all app icon assets

Launch Screen File

▼

Supported Intents

Class Name

Authentication

Add intents eligible for in-app handling here

+

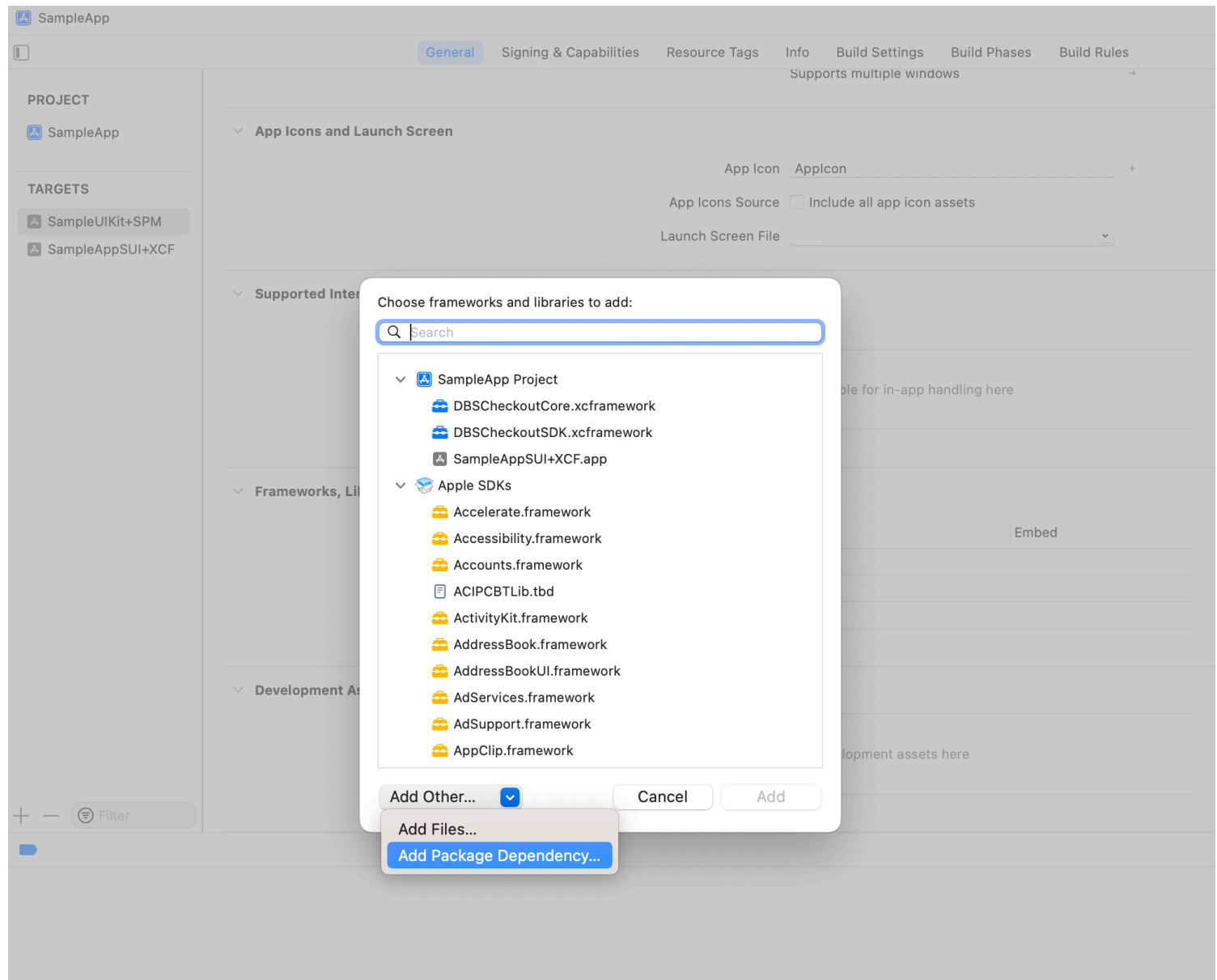
—

Frameworks, Libraries, and Embedded Content

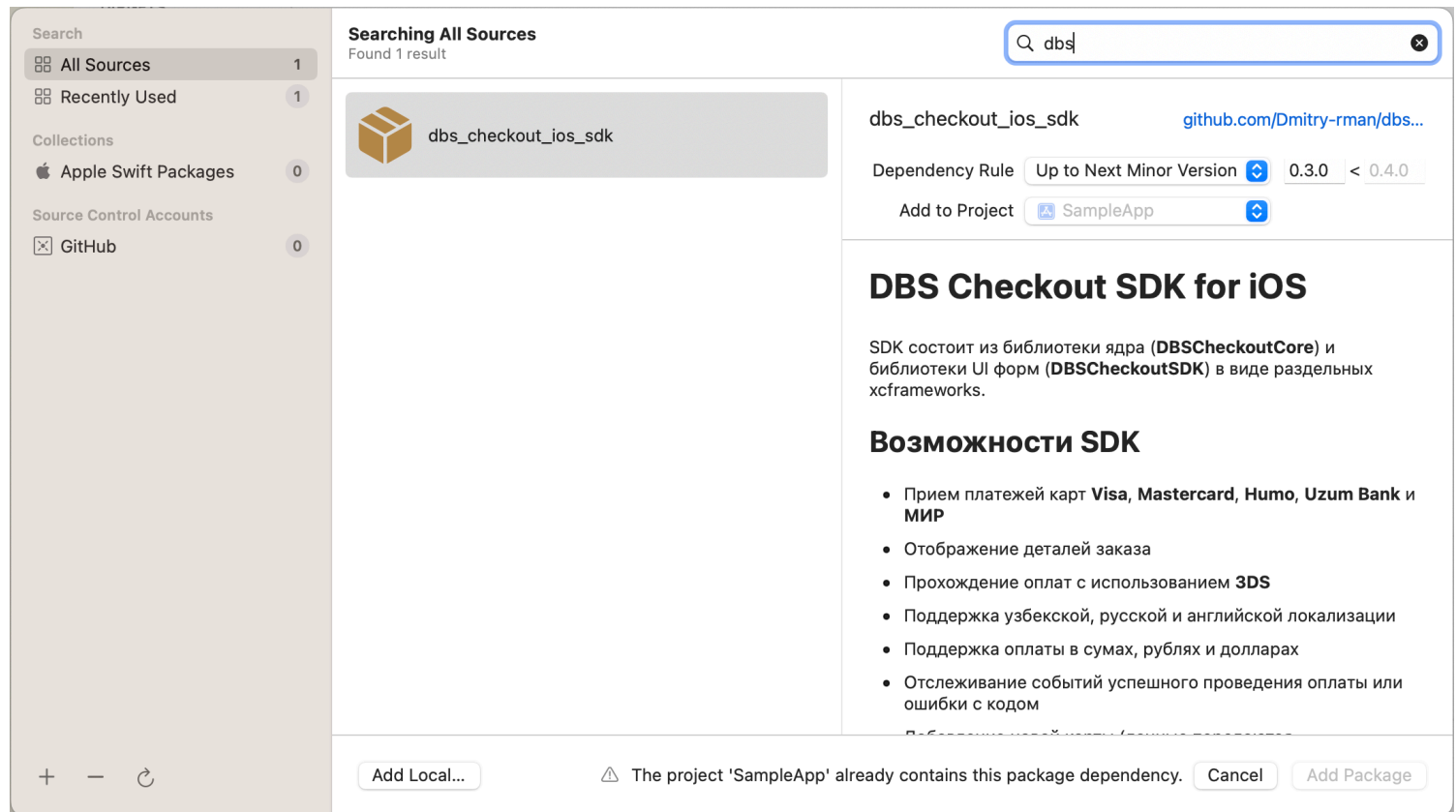
| Name | Embed |
|---|----------------|
|  DBSCheckoutCore.xcframework | Embed & Sign ⬆ |
|  DBSCheckoutSDK.xcframework | Embed & Sign ⬆ |

Через добавление зависимости SPM

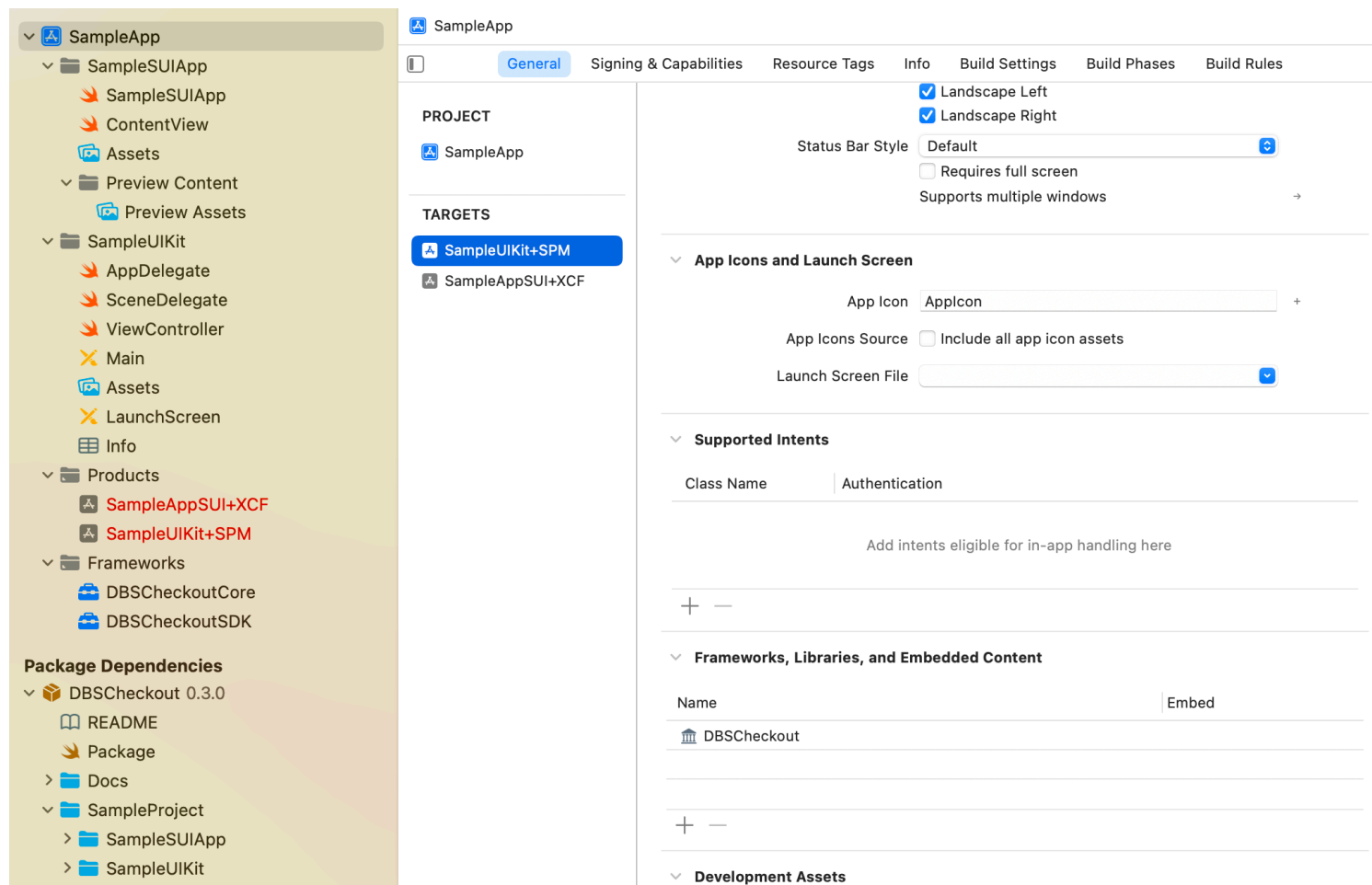
Для того, чтобы подключить в проект **DBS Checkout SDK** через Swift Package Manager в секции **Frameworks, Libraries and Embedded Content** нужно нажать на "+" и выбрать **Add Package Dependency**:



Далее вбить в строку поиска "https://github.com/Dmitry-rman/dbs_checkout_ios_sdk", выбрать его из списка результатов и добавить в нужный проект:



В результате в левом окне списка файлов появится секция **Package Dependencies** с номером совместимой версии, которая была указана:



P.S.: Вы всегда можете обновить SPM пакет до совместимого, щелкнув на него правой кнопкой мыши и выбрав пункт "Update Package".

Подготовка к работе

Вначале вам понадобится получить от платежного сервиса идентификатор платежа (**paymentId**). Подробнее об этом можно узнать в документации к [DBS Checkout](#), в личном кабинете или через почту checkout@inplattech.ru.

Для возможности работы с тестовым сервером **kube.inplatdev.ru** необходимо добавить в Info.plist свойство "App Transport Security Settings" со следующими значениями:

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoadsInWebContent</key>
  <true/>
  <key>NSAllowsArbitraryLoads</key>
  <false/>
  <key>NSExceptionDomains</key>
```

```
<dict>
  <key>kube.inplatdev.ru</key>
  <dict>
    <key>NSIncludesSubdomains</key>
    <true/>
    <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
    <true/>
    <key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
    <true/>
  </dict>
</dict>
</dict>
```

Начало работы

Для начала работы нужно создать конфигурацию **DBSCheckoutCoreSdkConfiguration**, используя следующие обязательные параметры:

- **orderId**: Id заказа платежа, полученного ранее от платежного сервиса.
- **environment**: режим работы SDK. Варианты **debug**(дев стенд), **merch**(бета-режим) и **prod**(релизный режим).

Дополнительные (опциональные) параметры:

- **language**(опциональный, по-умолчанию **ru**): язык интерфейса платежных форм. Варианты **ru**(русский), **uz**(узбекский), **en**(английский).
- **colorScheme**(опциональный, по-умолчанию **systemDefined**): цветовая схема UI. Варианты **light**(светлая тема), **dark**(темная тема), **systemDefined**(определяема операционной системой).

Для вызова платежной формы оплаты в **UIKit** используется метод **presentCheckoutView**, который вызывается у сконфигурированного экземпляра **sdk**. Метод принимает следующие параметры:

- **on** - ссылка на экземпляр **UIViewController**, с которого происходит показ.
- **dismissButtonType**: тип кнопки для закрытия формы. Варианты **close** (в виде крестика), **back** (в виде стрелки назад).
- **presentationType**: тип показа формы. Варианты **sheet**(пока в виде панели), **fullscreen**(показ на весь экран), **slide**(показ на весь экран, экран "приезжает" справа по аналогии с **SFSafariViewController** и может быть закрыт свайпом назад).

- **completionHandler**: замыкание, которое срабатывает после закрытия платежной формы и возвращает результат с типом-перечисление **PaymentCompletionResult**:

```
/// Тип результата формы оплаты
public enum PaymentCompletionResult {
    case success // Успешная оплата
    case closed(isFailed: Bool) // Форма закрыта. isFailed – была ошибка опла
```

Для создания View платежной формы в **SwiftUI** можно использовать метод **createCheckoutView** сконфигурированного экземпляра sdk. Метод имеет 2 параметра **dismissButtonType** и **presentationType** по аналогии с предыдущим методом. Вызов формы привязки карты осуществляется схожими методами **presentBindCardView** для **UIKit** и **createBindCardView** (создание формы) в случае **SwiftUI**.

Пример создания экземпляра SDK

Пример использования в UIKit приложении

```
///language по-умолчанию русский язык
let language = DBSCheckoutCoreSdkLanguage.ru

///ID заказа платежа, подлежащего оплате
let orderID = "order_id"

///Среда, используемая в SDK
let environment = DBSCheckoutCoreSdkEnvironment.merch

///Создать конфигурацию и экземпляр SDK
let configuration = DBSCheckoutCoreSdkConfiguration(orderId: orderID,
                                                    environment: environ
                                                    language: languag
                                                    colorScheme: .sys

let sdk = DBSCheckoutSdk(configuration: configuration)

///Показать флоу оплаты, где self – это экземпляр UIViewController
sdk.presentCheckoutView(on: self,
                        dismissButtonType: .close,
                        presentationType: .sheet,
                        completionHandler: { result in
```

```
        })

        //...

        ///Показать форму привязки карты
        sdk.presentBindCardView(on: self,
                                dismissButtonType: .close,
                                presentationType: .sheet,
                                completionHandler: { result in

        })
```

Пример использования в SwiftUI приложении

```
struct ContentView: View {

    @State var checkoutType: CheckoutType?

    enum CheckoutType: Int, Identifiable {

        var id: Int { self.rawValue }

        case bindCard
        case pay
    }

    var body: some View {

        VStack(spacing: 24){

            Button("Оплатить картой"){
                checkoutType = .pay
            }
            Button("Привязать карту"){
                checkoutType = .bindCard
            }
        }
        .padding()
        .sheet(item: $checkoutType) { type in
            createSDKView(type)
        }
    }

    @ViewBuilder
```



```
private func createSDKView(_ type: CheckoutType) -> some View {

    ///language по-умолчанию русский язык
    let language = DBSCheckoutCoreSdkLanguage.ru

    ///ID заказа платежа, подлежащего оплате
    let orderID = "order_id"

    ///Среда, используемая в SDK
    let environment = DBSCheckoutCoreSdkEnvironment.merch

    ///Создать конфигурацию и экземпляр SDK
    let configuration = DBSCheckoutCoreSdkConfiguration(orderID: orderID,
                                                         environment: environ
                                                         language: languag
                                                         colorScheme: .sys

    let sdk = DBSCheckoutSdk(configuration: configuration)

    switch type {

    case .pay:
        sdk.createCheckoutView(dismissButtonType: DBSCheckoutSdk.DismissBut
                             presentationType: DBSCheckoutSdk.Prese
                             completionHandler: { result in

        })
    case .bindCard:
        sdk.createBindCardView(dismissButtonType: DBSCheckoutSdk.DismissBut
                              presentationType: DBSCheckoutSdk.Prese
                              completionHandler: { result in

        })
    }
}
}
```

Поддержка

По всем возникающим вопросам, доработкам и предложениям обращаться на почту checkout@inplattech.ru