

Disclaimer

Работа приложения с API CloudPayments показана в демонстрационных целях. В вашем приложении, запросы должны идти на ваш сервер, а с него на API CloudPayments. Там же стоит хранить данные для подключения (PublicId и ApiSecret). Описание работы с API CloudPayments представлено на странице <http://cloudpayments.ru/Docs/Api#payWithCrypto>

Описание работы приложения с SDK CloudPayments

SDK CloudPayments (CloudPaymentsAPI.framework) позволяет

- проводить проверку карточного номера на корректность

```
[CPService isCardNumberValid: cardNumberString];
```

- определять тип платежной системы

```
[CPService cardTypeFromCardNumber:cardNumberString];
```

- шифровать карточные данные и создавать криптограмму для отправки на сервер

```
CPService *_apiService = [[CPService alloc] init];
NSString *cryptogramPacket = [_apiService makeCardCryptogramPacket:self.cardNumberString
                                                                    andExpDate:self.cardExpirationDateString
                                                                    andCVV:self.cardCVVString
                                                                    andStorePublicID:_apiPublicID];
```

Пример использования SDK и API CloudPayments дан в файле `CPViewController`

Демо-приложение представляет из себя форму для ввода карточных данных и обработчик запросов к API CloudPayments.

Перед началом оплаты необходимо определить переменные (их значения можно взять из личного кабинета):

```
NSString *_apiPublicID = @"pk_000000000000000000000000";
NSString *_apiSecret = @"00000000000000000000000000000000";
```

а также

```
NSString *_termURL = @"http://your_url.com";
```

Параметр `_termURL` необходим для проведения 3DS авторизации платежа. Это должен быть ваш собственный адрес.

Подробнее: <http://cloudpayments.ru/Docs/Api#3ds>

После этого необходимо инициализировать SDK CloudPayments:

```
CPService *_apiService = [[CPService alloc] init];
```

В дальнейшем `_apiService` используется для создания пакета криптограммы.

Проведение оплаты

Проведение оплаты описано в методе `makePaymentAction`. Пояснения к описанию метода.

1. В демо-приложении словарь `paramsDictionary` содержит только обязательные параметры для запроса. Список всех возможных параметров представлен <http://cloudpayments.ru/Docs/Api#payWithCrypto>
2. Метод для проведения 3DS-авторизации
`-(void) make3DSPaymentWithAcsURLString: (NSString *) acsURLString andPaReqString: (NSString *) paR`
3. Метод для проведения окончания 3DS-авторизации

```
-(void) complete3DSPaymentWithPaResString: (NSString *) paResString andTransactionIdString: (NSStr
```

Ключевые моменты

1. Библиотека поставляется в виде .framework, который скомпилирован для трех текущих архитектур процессора armv7, armv7s, arm64 и i385, x86_64. Таким образом тестировать можно в iPhone Simulator. Библиотека может работать только в версиях iOS 6.0+. iOS 8 также поддерживается.
2. В демо-проекте для сетевого взаимодействия используется библиотека AFNetworking (см. <https://github.com/AFNetworking/AFNetworking>). Все права на код этой библиотеки принадлежат авторам библиотеки.
3. В демо-проекте для показа экранных уведомлений используется библиотека SVProgressHUD (см. <https://github.com/TransitApp/SVProgressHUD>). Все права на код этой библиотеки принадлежат авторам библиотеки.