PayU S.A.
182, Grunwaldzka Str.
60-166 Poznań
Poland

Tel. +48 61 630 60 05
Email: pomoc@payu.pl
www.payu.pl

PayTouch iOS v2.1

Mobile Payments for PayU platform

PayU S.A.
182, Grunwaldzka Str.
60-166 Poznań
Poland

Tel. +48 61 630 60 05
Email: pomoc@payu.pl
www.payu.pl

PayTouch Integration

The PayTouch library makes it easy to integrate mobile payments in the iOS application. The SDK is available in form of a static library. It provides both network communication, security and user interface for the payment process. It takes care of the payment method management, processing payment and payment authorization (CVV, 3D Secure, etc.)

The main use case is to accept single payment with following payment methods:

- Credit/Debit cards (Visa, Mastercard, Maestro)
- PayU Express - http://express.payu.pl/
- Pay-By-Link – http://payu.pl/blog/pay-by-link-plac-bezposrednio-z-konta/

Requirements

- iOS 8 or later
- Server-side OAuth2 token retrieval – described in a separate document

SDK package contains

- Technical documentation
- Sample application with integrated SDK
- PayTouch SDK
  - Header files
  - Static library
    - **libPayU-iOS-SDK-Oneclick_Release** – This SDK version does not allow debuging due to security measures. It must be used only for release i.e. it will work in application installed from .ipa file.
  - Assets bundle

PayU S.A.
182, Grunwaldzka Str.
60-166 Poznań
Poland

Tel. +48 61 630 60 05
Email: pomoc@payu.pl
www.payu.pl

Importing the SDK

1. Drop the library folder into iOS application project.
2. Add **-all_load** to Build Settings > Linking > Other Linker Flags.

Credentials

For production environment you will have to retrieve OAuth2 access token from the PayU backend service via your backend application. This process must be conducted by your backend since direct storage of your **`client_id`** and **`client_secret`** in the application is **prohibited**.

Environment

You can define “PUEnvironment” string key in your application’s Info.plist file to switch SDK between “Production” and “Sandbox” environments. If the environment key is not defined or value is not valid then “Production” environment is used by default.

Key	Type	Value
▼ Information Property List	Dictionary	(18 items)
PUEnvironment	String	Sandbox

Localization

The SDK has support for Polish, English, German and Czech language. Language is selected automatically based on the system language setting.

Integration tutorial

PayU S.A.
182, Grunwaldzka Str.
60-166 Poznań
Poland

Tel. +48 61 630 60 05
Email: pomoc@payu.pl
www.payu.pl

PUPaymentService is the main class you will work with. Proper communication to and handling requests from this class is essential. This tutorial shows the simplest way of integrating the SDK into your application.

PUAuthorizationDataSource implementation

In order to use SDK in a production environment you have to create a class that implements PUAuthorizationDataSource protocol. This class will be used by PayTouch SDK, in order to obtain valid access token and application callback scheme. Your implementation will be called by payment service when specified data will be needed. Instance of this class should be passed to payment service as soon as possible.

- refreshTokenWithCompletionHandler:

This method will be called when merchant access token is not valid or is missing. Payment service will ask client application for OAuth credentials for current user. You should contact your server to obtain PayU access token for currently logged user. Example:

```
// Obj-C
- (Void)refreshTokenWithCompletionHandler:(void (^)(NSString *, NSError
*))completionHandler
{
    /* Fetch access token from your backend */

    completionHandler(/* access token */, nil); /* Invoke when token is retrieved */
    completionHandler(nil, /* error */); /* Invoke when there was an error */
}

// Swift
func refreshToken(completionHandler: @escaping (String?, Error?) -> Void) {
    /* Fetch access token from your backend */

    completionHandler(/* access token */, nil); /* Invoke when token is retrieved */
    completionHandler(nil, /* error */); /* Invoke when there was an error */
}
```

- applicationCallbackScheme;

With PayTouch it is possible to authorize payments using PayU or Bank applications.

```
// Obj-C
- (NSString *)applicationCallbackScheme
{
    return @"sample-app-x-callback";
}
```

PayU S.A.
182, Grunwaldzka Str.
60-166 Poznań
Poland

Tel. +48 61 630 60 05
Email: pomoc@payu.pl
www.payu.pl

```swift
// Swift
func applicationCallbackScheme() -> String {
    return "sample-app-x-callback"
}
```

In order to enable paying in external applications You have to provide payment service with Your application callback scheme. Example:
More about this integration in „*Paying using external applications*”.

PUPaymentServiceDelegate implementation

Implement method presented below in order to enable showing of view controllers that are integral part of payment process like:

- user payment methods list
- authorization view controllers

- (void)paymentServiceDidRequestPresentingViewController:

You should present received view controllers modally. Example:

```objc
// Obj-C
- (void)paymentServiceDidRequestPresentingViewController:(UIViewController
*)viewController
{
    [self.navigationController presentViewController:viewController animated:YES
completion:nil];
}
```

```swift
// Swift
func paymentServiceDidRequestPresenting(_ viewController: UIViewController) {
    navigationController?.present(viewController, animated: true, completion: nil)
}
```

- (void)paymentServiceDidSelectPaymentMethod:

You can use this method to know when user selects a payment method or when change occurs that clears current selected payment method. Example:

```objc
// Obj-C
- (void)paymentServiceDidSelectPaymentMethod:(PUPaymentMethodDescription *)paymentMethod
{
    BOOL hasPaymentMethod = paymentMethod != nil;
    self.payButton.enabled = hasPaymentMethod;
}
```

```swift
// Swift
```

PayU S.A.
182, Grunwaldzka Str.
60-166 Poznań
Poland

Tel. +48 61 630 60 05
Email: pomoc@payu.pl
www.payu.pl

```
func paymentServiceDidSelectPaymentMethod(_ paymentMethod: PUPaymentMethodDescription?)
{
    let hasPaymentMethod = paymentMethod != nil
    payButton.isEnabled = hasPaymentMethod
}
```

PUPaymentService initialization

Create strong property that will hold PUPaymentService and will prevent its deallocation to make sure that payment flows won't be interrupted. Example:

Payment service configuration example:

PayU S.A.
182, Grunwaldzka Str.
60-166 Poznań
Poland

Tel. +48 61 630 60 05
Email: pomoc@payu.pl
www.payu.pl

```
// Obj-C
self.paymentService = [[PUPaymentService alloc] init];
self.paymentService.dataSource = /* Class that implements PUAuthorizationDataSource
protocol */
self.paymentService.delegate = /* Class that implements PUPaymentServiceDelegate;
protocol */

// Swift
paymentService = PUPaymentService()
paymentService.dataSource = /* Class that implements PUAuthorizationDataSource protocol
*/
paymentService.delegate = /* Class that implements PUPaymentServiceDelegate protocol */
```

Paying using external applications

Assuming that you have already provided your application callback scheme to payment service (described in "*PUAuthorizationDataSource implementation")* you only have to pass the URL that your application has received to payment service. Example:

```
// Obj-C
- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
sourceApplication:(NSString *)sourceApplication annotation:(id)annotation
{
    [self.paymentService handleOpenURL:url];

    return YES;
}

// Swift
func application(_ app: UIApplication, open url: URL, options:
[UIApplicationOpenURLOptionsKey: Any] = [:]) -> Bool {
    paymentService.handleOpen(url)
    return true
}
```

Payment method widget

Payment method widget is a subclass of UIView managed by payment service. Its purpose is to show selected payment method to the user and to inform payment service that user wants to select payment method from the list. You only have to get the widget from payment service using:

- paymentMethodWidgetWithFrame:

and present it in your view controller. Widget will resize horizontally but keep in mind that height is fixed to 85. Example:

PayU S.A.
182, Grunwaldzka Str.
60-166 Poznań
Poland

Tel. +48 61 630 60 05
Email: pomoc@payu.pl
www.payu.pl

```
// Obj-C
[self.widgetView addSubview:[self.paymentService
paymentMethodWidgetWithFrame:self.widgetView.bounds]];

// Swift
let widget = paymentService.paymentMethodWidget(withFrame: widgetContainer.bounds)
widgetContainer.addSubview(widget)
```

In this example You just add widget to view that You have designated to this purpose.

Payment

Payment is the last point of the process. When you invoke **- submitPaymentRequest: completionHandler:** the user will be directed through the payment process. In the best case scenario the payment will be finished with no interaction. However keep in mind that the user may be asked to confirm payment using some verification method (CVV, 3DS, etc.). All the authorization view controllers are provided by PayU and you are not required to do any action beside implementing ***- paymentServiceDidRequestPresentingViewController*** delegate method.

To submit payment you have to create payment request representing your transaction:

```
// Obj-C
PUPaymentRequest *paymentRequest = [[PUPaymentRequest alloc] init];
paymentRequest.extOrderId = @"unique_payment_identifier_from_your_system";
paymentRequest.paymentDescription =@"Order description that will be displayed to the
user";
paymentRequest.amount = [NSDecimalNumber decimalNumberWithMantissa:500 exponent:0
isNegative:NO];
paymentRequest.currency = @"PLN";
paymentRequest.notifyURL = [NSURL URLWithString:[@"https://your.url.com/notify/"
stringByAppendingString:paymentRequest.extOrderId]];

// Swift
let paymentRequest = PUPaymentRequest()
paymentRequest.extOrderId = "Unique payment identifier in your system"
paymentRequest.paymentDescription = "Order description that will be displayed to the
user"
paymentRequest.amount = NSDecimalNumber(decimal: Decimal(5.00))
paymentRequest.currency = "PLN"
paymentRequest.notifyURL = URL(string:
"https://api.merchantbackend.com/notify/\(paymentRequest.extOrderId)")!
```

Then you just pass payment request to method responsibile for submitting payments and handle completion handler code block. Example:

PayU S.A.
182, Grunwaldzka Str.
60-166 Poznań
Poland

Tel. +48 61 630 60 05
Email: pomoc@payu.pl
www.payu.pl

```objc
// Obj-C
[self.paymentService submitPaymentRequest:paymentRequest
                        completionHandler:^(PUPaymentRequestResult result) {
                            switch (result.status) {
                                case PUPaymentRequestStatusSuccess:
                                    /* Payment submission was successful */
                                    break;
                                case PUPaymentRequestStatusRetry:
                                    /* User decided to change payment method. You
should enable the user to submit payment once again */
                                    break;
                                case PUPaymentRequestStatusFailure:
                                    /* Payment submission failed */
                                    break;
                            }
                        }];

// Swift
paymentService.submitPaymentRequest(paymentRequest) { [weak self] result in
     switch result.status {
     case .success:
       /* Payment submission was successful */
     case .retry:
       /* User decided to change payment method. You should enable the user to submit
payment once again */
     case .failure:
       /* Payment submission failed */
     }
}
```

When payment submission failed there are two specific cases you should consider:

- [**PUErrorCodePaymentCancelledByUser**, **PUErrorCodeExteralPaymentCancelledByUser**] means that user cancelled the payment
- **PUErrorCodeExteralPaymentInterruptedByUser** means that something unexpected has happend but it doesn't mean that the transaction has failed. Probably it has failed but you should inform the user to check payment status to make sure.

Sending order is asynchronous. PayTouch informs only about status of payment submission. Every single payment must be confirmed server side.

Logout user from application

When user decides to logout from your application, PayU SDK must be notified about it, in order to clear sensitive user data. This method **must** be called after successful user logout to avoid unexpected SDK behaviour.

PayU S.A.
182, Grunwaldzka Str.
60-166 Poznań
Poland

Tel. +48 61 630 60 05
Email: pomoc@payu.pl
www.payu.pl

- clearUserContext;

```
// Obj-C
- (void)logOutPressed:(id)sender
{
    [self.paymentService clearUserContext];
}

// Swift
func logOutPressed(_ sender: Any) {
    paymentService.clearUserContext()
}
```