

# Apple Pay for PayU Mobile SDK

---

Apple Pay™ is a digital wallet which lets you make card payments in a simple and fast manner, without having to enter your card details every time. The card data is securely stored by Apple. Apple Pay is supported in the following countries: <https://www.apple.com/ios/feature-availability/#apple-pay>

## Requirements

### Hardware

Note that Apple Pay is supported only by following devices:

<https://support.apple.com/en-us/HT208531>

### Software

- Feature is supported for iOS 10+
- Currently PayU supports Visa and Mastercard

### PayU backend

POS with enabled Apple Pay payment method is required for both sandbox (testing) and production (real transactions) environment:

- Contact PayU support to enable Apple Pay payments on your POS for each environment: [itsupport@payu.pl](mailto:itsupport@payu.pl)

## Prerequisites

**Before implementing and using Apple Pay in the app a number of prerequisites must be completed:**

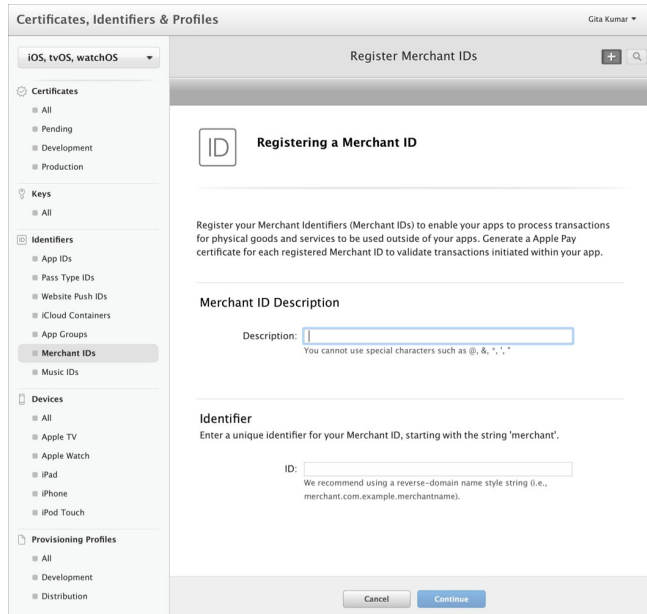
- Possesing active Apple Developer Account
- Creating a Merchant Identifier
- Creating an Apple Pay Payment Processing Certificate
- Enabling Apple Pay in XCode project

[https://developer.apple.com/documentation/passkit/apple\\_pay/setting\\_up\\_apple\\_pay\\_requirements](https://developer.apple.com/documentation/passkit/apple_pay/setting_up_apple_pay_requirements)

According to documentation available on 19.09.2018:

## Create a Merchant Identifier (Apple)

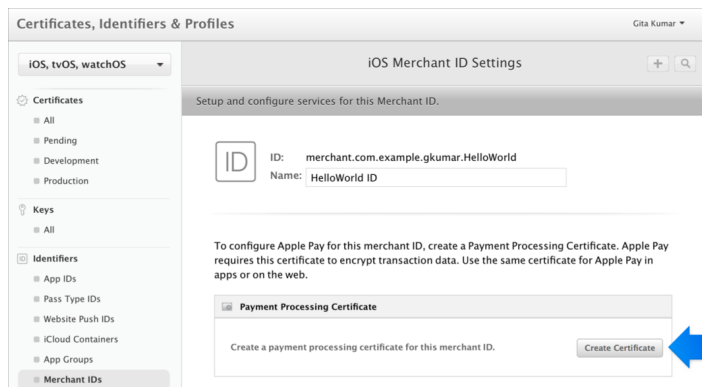
- Sign in to your Apple Developer Program (Required role: Team Agent or Admin)
- In "Certificates, Identifiers & Profiles", choose "iOS, tvOS, watchOS" from the pop-up menu on the left Under
- Identifiers, select Merchant IDs, then in the upper-right corner, click the Add button (+)
- Enter the merchant description and identifier name, then click Continue



- Review the settings, then click Register
- Click Done

## Create an Apple Pay Payment Processing Certificate

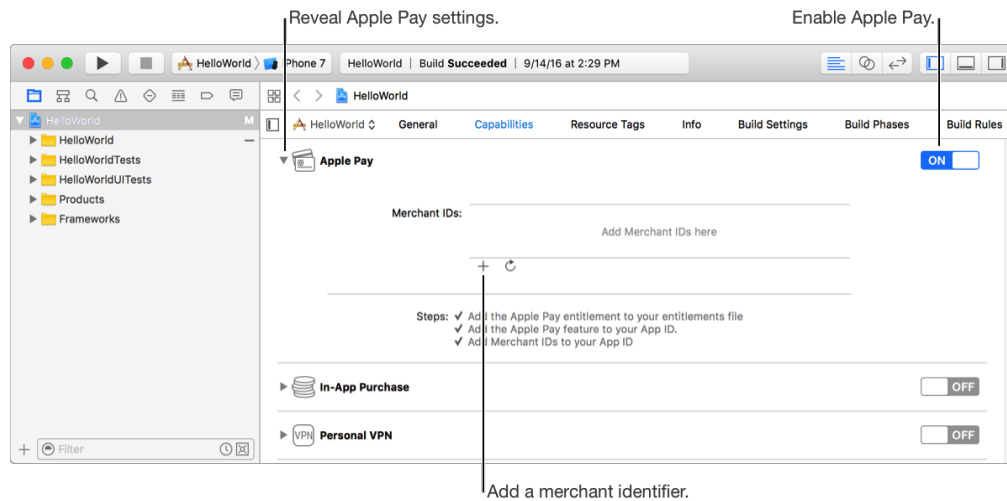
- Sign in to your Apple Developer Program
- Select "Certificates, Identifiers & Profiles"
- Select Identifiers / Merchant IDs
- Select the Merchant ID that you have created, and select Edit
- In the Apple Pay Payment Processing Certificate section, select Create Certificate



- On the next screen select Continue
- Download the CSR file from the PayU site <https://www.payu.pl/> and upload it to the site, selecting Choose File and then Continue Download
- the generated certificate
- Send the downloaded certificate (the file apple\_pay.cer) to the PayU IT support department (tech@payu.pl)

## Enable Apple Pay in XCode project

- In the project editor, select the target and click Capabilities
- In the Apple Pay section, click the switch to turn it from OFF to ON.



## Sandbox

The Apple Pay payment method is also available in a sandbox environment. In the integration process, we suggest creating an independent Merchant ID (with a name ending “.test”, for example) together with a set of certificates. For this purpose, download the file CSR Sandbox from the PayU site <https://www.payu.pl/>

Because Apple Pay is not the default payment method, please contact the PayU IT support department (tech@payu.pl) after registering in the sandbox environment, but before beginning integration using the Apple Pay payment method.

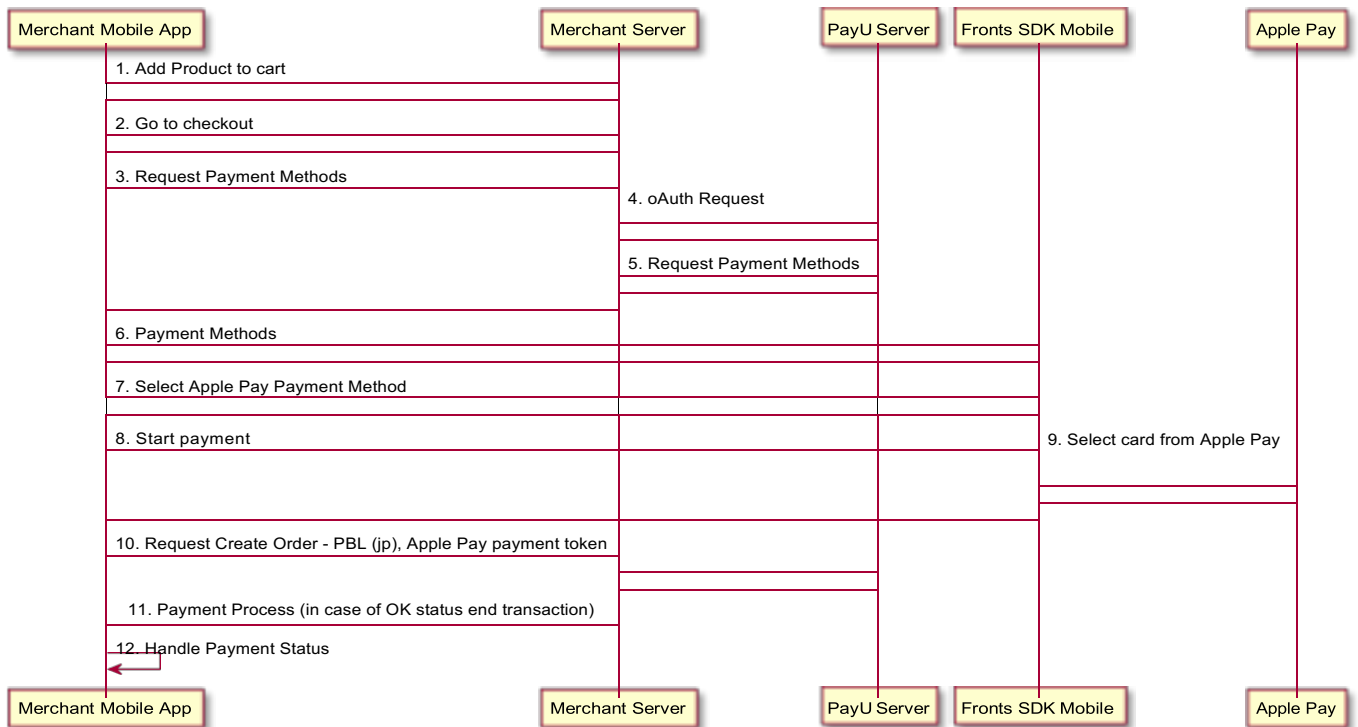
Before beginning to perform tests in the sandbox, please also read the Apple testing instructions <https://developer.apple.com/apple-pay/sandbox-testing/>

Note that unfortunately Poland is not one of the regions currently supported by Apple for sandbox tests. We therefore recommend that you change the region on the device from which tests will be carried out to one of the supported countries, such as the United Kingdom. This will make it possible to use test MasterCard or Visa cards. We recommend using the card numbers:

- 5204 2477 5000 1471 MasterCard
- 4761 1200 1000 0492 Visa

since these are configured in the PayU sandbox to enable payments to be completed successfully

## Apple Pay flow example



1. User adds items to basket in merchant app, Merchant app reacts to the add item action performed by user
2. User goes to checkout page in merchant app, Merchant app reacts
3. List of available payment methods is requested
4. Merchant server sends oauth request to PayU server, PayU server sends oauth response to merchant server  
<https://payu21.docs.apiary.io/#reference/api-endpoints/oauth-api-endpoint/get-access-token>
5. Merchant server requests a list of payment methods from PayU server, PayU server sends list of payment methods to merchant sever, Merchant server sends list of payment methods to merchant app
6. Merchant app sends list of payment methods to PayU SDK, PayU SDK displays available payment methods
7. User can select one of available payment methods. In this flow it will be Apple Pay
8. User taps button which initializes payment in merchant app
9. User chooses card from Apple Pay, PayU SDK starts Apple Pay transaction with passed transaction details, PayU SDK gets Apple Pay payment token after successful authorization by Apple, PayU SDK notifies merchant app, passing it the payment token, Merchant app sends Apple Pay payment token to Merchant server
10. Merchant creates an order using received Apple Pay payment token in PBL payment with value "jp", PayU server sends response with a status to merchant server
11. Payment is being processed - if status is OK transaction can be finished
12. Merchant app handles received payment status accordingly

## Introduction

Main purpose of this module is to enable new payment method - Apple Pay handling in PayU SDK, for iOS devices. Steps for implementation:

- Creating *PUApplePayHandler* object to handle Apple Pay transaction and pass the payment token to merchant mobile app
- Creating *PUApplePayTransaction* object and passing 'transaction details' (Apple merchant id, currency, country, user email address, description, total amount) to it
- Calling *authorizeTransaction* method (with created *PUApplePayTransaction* object and *UIParent* as arguments) on created *PUApplePayHandler*
- Receiving callback Apple Pay payment token to be used in payment process on merchant backed - order create request (if Apple Pay payment token was generated successfully) <https://payu21.docs.apiary.io/#reference/api-endpoints/paymethods-api-endpoint/create-a-new-order>
- Sending Order Create Request with Apple Pay payment token from merchant backend to PayU backend

Because orders with the Apple Pay payment method will be billed as standard card payments, you may wish to consider integrating the transaction data download service. In this way you can distinguish cards entered via the PayU form from cards obtained from Apple Pay [http://developers.payu.com/en/restapi.html#trx\\_data\\_retrieve](http://developers.payu.com/en/restapi.html#trx_data_retrieve)

## Implementation overview

This module consists of:

### ***PUApplePayHandler***

**Handles authorization of Apple Pay payment**

Has one method:

- (void)authorizeTransaction:(*PUApplePayTransaction* \*)transaction withUIparent:(*UIViewController* \*)UIParent;

### ***PUApplePayHandlerDelegate***

**Protocol that should be implemented by *PUApplePayHandler* delegate to receive Apple Pay transaction status**

Protocol has 2 methods that have to be implemented:

- (void)paymentTransactionCanceledByUser:(*PUApplePayTransaction* \*)transaction; Transaction

cancellation handling should be implemented in this method

- (void)paymentTransaction:(*PUApplePayTransaction* \*)transaction result:(*NSString* \*)result;

Positive authorization handling should be implemented in this method (payment token is passed as argument)

## ***PUApplePayTransaction***

### **Contains Apple Pay transaction data**

Has following fields:

- initWithMerchantIdentifier:(NSString \*)merchantIdentifier - Apple merchant ID
- currencyCode:(PUCurrencyCode)currencyCode - Code of used currency, example:

```
typedef NS_ENUM(NSUInteger, PUCurrencyCode) {  
    PUCurrencyCodePLN,  
    PUCurrencyCodeCZK,  
    PUCurrencyCodeEUR,  
};
```

countryCode:(PUCountryCode)countryCode - Code of country, example:

```
typedef NS_ENUM(NSUInteger, PUCountryCode) {  
    PUCountryCodePL,  
    PUCountryCodeCZ,  
};
```

- contactEmailAddress:(NSString \*)contactEmailAddress - User's email address
- paymanetItemDescription:(NSString \*)paymanetItemDescription - Displayed transaction description
- amount:(NSDecimalNumber \*)amount; - Total amount to be paid

To use *PUApplePayHandler*:

- implement *PUApplePayHandlerDelegate* protocol
- create *PUApplePayTransaction* containing transaction details
- create *PUApplePayHandler* object
- set delegate on *PUApplePayHandler* object
- call method *authorizeTransaction* on *PUApplePayHandler* object, pass *PUApplePayTransaction* object and *UIparent* as arguments

**For more information see implementation example**

## Implementation example

In: YourClass

Create a method that will handle initiating Apple Pay transaction. Method should be invoked after tapping payment confirmation button ("Pay" button) in your app. Example:

Method invocation:

```
//Set payAction method call on payButton tap
payButton.addTarget(self, action: #selector(payAction(_:)), for: .touchUpInside)
```

Method implementation:

```
@objc func payAction(_ sender: Any) {
    //Your code to make sure that selected payment method is of Apple Pay type
    if let selectedPayment = self.paymentWidget_lastSelectedPaymentMethod, selectedPayment.isKind(of:
PUApplePay.self) {
        //Create a PUApplePayTransaction object and pass transaction details to it
        let applePayTransaction = PUApplePayTransaction(
            //Pass your Apple merchant identifier here
            merchantIdentifier: "your.apple.merchant.id",
            //Pass code for used currency. Available codes: PLN, CZK, EUR
            currencyCode: PUCurrencyCode.your_currency_code,
            //Pass code for country. Codes: countryCodePL, countryCodeCZ, countryCodeDE
            countryCode: PUCountryCode.your_country_code,
            //Pass email address of user placing the order
            contactEmailAddress: "user@email.address",
            //Pass description to be displayed on Apple Pay view
            paymentItemDescription: "Item description",
            //Pass the total amount of the payment as a string, eg. 55,50 PLN would be "55.5"
            amount: NSDecimalNumber(string: "55.5"))

        //Call a constructor of PUApplePayHandler to create an object
        self.applePayHandler = PUApplePayHandler()
        //Set delegate on PUApplePayHandler object (delegate implementation example below)
        self.applePayHandler.delegate = self
        //Call authorizeTransaction method on PUApplePayHandler object, pass PUApplePayTransaction object and
        UIParent as arguments
        applePayHandler.authorizeTransaction(applePayTransaction, withUIParent: self)
    }
}
```

## In: YourClass+Delegates

Delegate implementation:

```
//Extend your class with PUAApplePayHandlerDelegate protocol
extension YourClass: PUAApplePayHandlerDelegate {
    //This will be called if user cancels the transaction while its being processed
    //eg. tapped home button during authorization
    func paymentTransactionCanceled(byUser transaction: PUAApplePayTransaction!) {
        //Your code to handle transaction cancellation
    }
    //This will be called after successful authorization
    func paymentTransaction(_ transaction: PUAApplePayTransaction!, result: String!) {
        //Your code to handle successful authorization
        //Handling result string which contains payment token
        //so you can use it to create an order (value for "authorizationCode" key)
    }
}
```

## JSON object for Apple Pay order

After receiving Apple Pay payment token an order using Apple Pay as a payment method can be created on merchant server. This is very similar to a standard PBL payment, the only difference is setting the right values for "value" and "authorizationCode" of "payMethod" JSON:

- "value": "jp" - use "jp" value for Apple Pay method (Make sure your POS is configured correctly, see Requirements section) "authorizationCode":
- "payment\_token\_from\_callback" - use string received in above callback as the value for this key

```
{
  "customerIp": "customer_ip",
  "merchantPosId": "your_pos_id", //POS ID of merchant creating the order
  "description": "transaction_description", //Description of the transaction
  "currencyCode": "PLN", //Code for used currency
  "totalAmount": "5500", //Total amount of the order in the smallest unit of currency - 55,50 PLN would be 5500
  "products": [ //Array of objects representing ordered products
    {
      "name": "product_name", //Name of the product
      "unitPrice": "5500", //Unit price of the product in the smallest unit of currency - 55,50 PLN would be 5500
      "quantity": "1" //Ordered quantity of the product
    }
  ],
  "buyer": {
    "email": "user@email.address", //email address of user placing the order
    "firstName": "user_name", //name of user placing the order
    "lastName": "user_last_name", //last name of user placing the order
    "language": "pl" //language of the email message to be received by user
  },
  "payMethods": {
    "payMethod": {
      "type": "PBL", //Payment type. FOR APPLE PAY USE "PBL"
      "value": "jp", //Specifies target bank etc. FOR APPLE PAY USE "jp"
      "authorizationCode": "payment_token_from_callback" //FOR APPLE PAY USE STRING (APPLE PAY PAYMENT TOKEN)
    }
  }
}
```



# Links

---

<https://www.payu.pl/>

[http://developers.payu.com/en/mobile\\_sdk.html](http://developers.payu.com/en/mobile_sdk.html)

[https://drive.google.com/drive/folders/1BvDT6gUs2uD0VYMc\\_7InUfG0rWk8WBae](https://drive.google.com/drive/folders/1BvDT6gUs2uD0VYMc_7InUfG0rWk8WBae)

[http://developers.payu.com/pl/restapi.html#trx\\_data\\_retrieve](http://developers.payu.com/pl/restapi.html#trx_data_retrieve) <https://payu21.docs.apiary.io/#reference/api-endpoints/paymethods-api-endpoint/create-a-new-order>

<https://developer.apple.com/>

<https://www.apple.com/ios/feature-availability/#apple-pay>

<https://support.apple.com/en-us/HT208531>

[https://developer.apple.com/documentation/passkit/apple\\_pay/setting\\_up\\_apple\\_pay\\_requirements](https://developer.apple.com/documentation/passkit/apple_pay/setting_up_apple_pay_requirements)

[https://developer.apple.com/documentation/passkit/apple\\_pay](https://developer.apple.com/documentation/passkit/apple_pay)

<https://developer.apple.com/apple-pay/sandbox-testing/>