Apple Pay - Frontend

Revision History

Version	Date	Content
1.0.0	2023.05.15	First Draft

Table of Contents

Apple Pay in Apps

If you want to know the latest SDK version and the difference between each version, please refer to Github Release Page: TapPay iOS Github

Support iOS 10 and later.

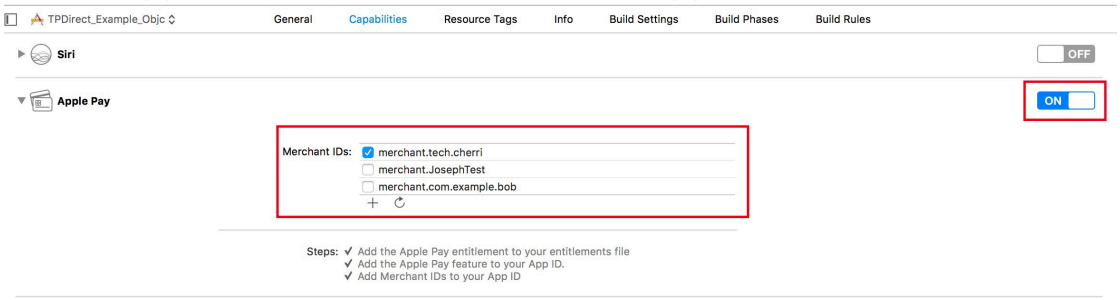
Environment

Before you get to coding, make sure you have done the following things:

1. Download and import TPDirect.framework into your project.

2. Import PassKit.framework into your project.

3. Enable Apple Pay in your Xcode and add Apple Merchant IDs.



4. Use TPDSetup to set up your environment.

5. Please open Advertising Identifier, to improve the accuracy of fraud detect.

TPDSetup

iOS-objC

```
+ (instancetype _Nonnull)setWithAppId:(int)appId
                              withAppKey:(NSString *_Nonnull)appKey
                          withServerType:(TPDServerType)serverType;
```

iOS-Swift

```
class func setWithAppKey(_ appKey: String, withAppId appId: Int32, with serverType:
TPDServerType) -> Self
```

iOS-objC

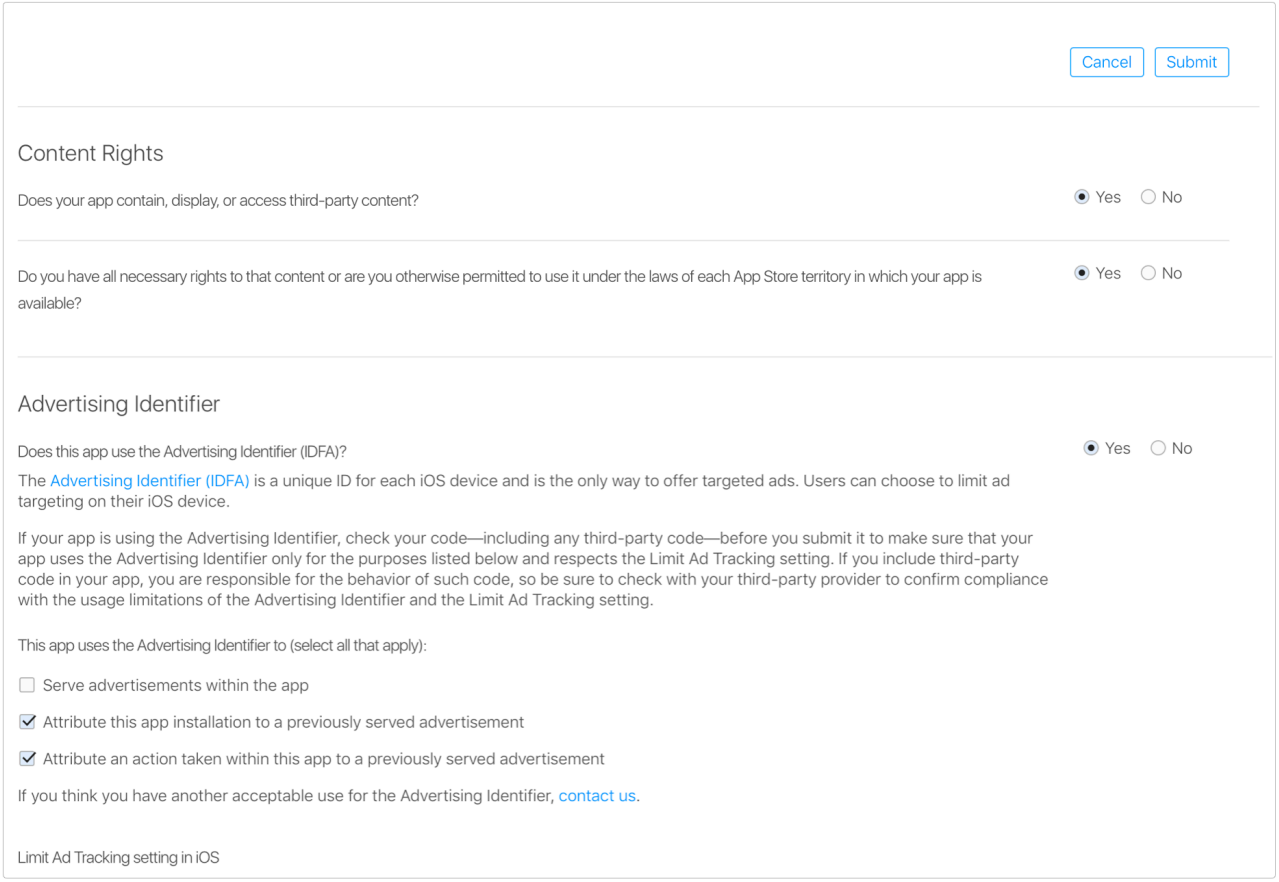
```
...

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions {
    // Override point for customization after application launch.
    [TPDSetup setWithAppId:"APP_ID" withAppKey:@"APP_KEY"
withServerType:TPDServer_SandBox];

    return YES;
}
```

iOS-Swift

```
...

func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
  // Override point for customization after application launch.
  TPDSetup.setWithAppId("APP_ID", withAppKey: "APP_KEY", with:
TPDServerType.sandBox)

  return true
}
```

Name	Type	Usage
appID	int	Please refer to appid.
appKey	String	Please refer to appkey
serverType	TPDServerType	Use “TPDServerType.Sandbox” for sandbox environment, and “TPDServerType.Production” for production environment.

Please open Advertising Identifier, to improve the accuracy of fraud detect.



Codes

1. Import PassKit and TPDirect frameworks into your project.

iOS-objC

```
#import <PassKit/PassKit.h>
#import <TPDirect/TPDirect.h>
```

iOS-Swift

```
import PassKit
import TPDirect
```

2. Implement the following classes in order:
    1. TPDApplyPay
    2. TPDApplePayDelegate
    3. TPDTransactionResult
    4. PKPaymentButton
    5. TPDMerchant
    6. TPDConsumer
    7. TPDCart

TPDApplePay

iOS-objC

```
@interface ViewController () <TPDApplePayDelegate>
@property (nonatomic, strong) TPDApplePay *applePay;
@end
```

iOS-Swift

```
class ViewController: UIViewController {
    var applePay : TPDApplePay!
}
```

TPDApplePayDelegate

iOS-objC

```
@protocol TPDApplePayDelegate <NSObject>

@required
// Send To The Delegate After Receive Prime and prime expiry millis.
- (void)tpdApplePay:(TPDApplePay *)applePay didReceivePrime:(NSString *)prime
withExpiryMillis:(long)expiryMillis withCardInfo:(TPDCardInfo *)cardInfo
withMerchantReferenceInfo:(NSDictionary *)merchantReferenceInfo;

// Send To The Delegate After Apple Pay Payment Processing Succeeds.
- (void)tpdApplePay:(TPDApplePay *)applePay didSuccessPayment:
(TPDTransactionResult *)result;

// Send To The Delegate After Apple Pay Payment Processing Fails.
- (void)tpdApplePay:(TPDApplePay *)applePay didFailurePayment: (TPDTransactionResult
*)result;

@optional
// Send To The Delegate After Apple Pay Payment's Form Is Shown.
- (void)tpdApplePayDidStartPayment:(TPDApplePay *)applePay;

// Send To The Delegate After User Selects A Payment Method.
// You Can Change The PaymentItem Or Discount Here.
- (TPDCart *)tpdApplePay:(TPDApplePay *)applePay didSelectPaymentMethod:
(PKPaymentMethod *)paymentMethod cart:(TPDCart *)cart;

// Send To The Delegate After User Selects A Shipping Method.
// Set shippingMethods ==> TPDMerchant.shippingMethods.
- (void)tpdApplePay:(TPDApplePay *)applePay didSelectShippingMethod:
(PKShippingMethod *)shippingMethod;

// Send To The Delegate After User Authorizes The Payment.
// You Can Check Shipping Contact Here, Return YES If Authorized.
- (BOOL)tpdApplePay:(TPDApplePay *)applePay
canAuthorizePaymentWithShippingContact:(PKContact *)shippingContact;

// Send To The Delegate After User Cancels The Payment.
- (void)tpdApplePayDidCancelPayment:(TPDApplePay *)applePay;

// Send To The Delegate After Apple Pay Payment's Form Disappeared.
- (void)tpdApplePayDidFinishPayment:(TPDApplePay *)applePay;

@end
```

iOS-Swift

```
public protocol TPDApplePayDelegate : NSObjectProtocol {
  // Send To The Delegate After Receive Prime.
  public func tpdApplePay(_ applePay: TPDApplePay!, didReceivePrime prime: String!,
withExpiryMillis expiryMillis: Int, withCardInfo cardInfo: TPDCardInfo,
withMerchantReferenceInfo merchantReferenceInfo: [AnyHashable : Any]!)

  // Send To The Delegate After Apple Pay Payment Processing Succeeds.
  public func tpdApplePay(_ applePay: TPDApplePay!, didSuccessPayment result:
TPDTransactionResult!)

  // Send To The Delegate After Apple Pay Payment Processing Fails.
  public func tpdApplePay(_ applePay: TPDApplePay!, didFailurePayment result:
TPDTransactionResult!)

  // Send To The Delegate After Apple Pay Payment's Form Is Shown.
  optional public func tpdApplePayDidStartPayment(_ applePay: TPDApplePay!)

  // Send To The Delegate After User Selects A Payment Method.
  // You Can Change The PaymentItem Or Discount Here.
  @available(iOS 9.0, *)
  optional public func tpdApplePay(_ applePay: TPDApplePay!, didSelect paymentMethod:
PKPaymentMethod!, cart: TPDCart!) -> TPDCart!

  // Send To The Delegate After User Selects A Shipping Method.
  // Set shippingMethods ==> TPDMerchant.shippingMethods.
  @available(iOS 8.0, *)
  optional public func tpdApplePay(_ applePay: TPDApplePay!, didSelect shippingMethod:
PKShippingMethod!)

  // Send To The Delegate After User Authorizes The Payment.
  // You Can Check Shipping Contact Here, Return YES If Authorized.
  @available(iOS 9.0, *)
  optional public func tpdApplePay(_ applePay: TPDApplePay!,
canAuthorizePaymentWithShippingContact shippingContact: PKContact!) -> Bool

  // Send To The Delegate After User Cancels The Payment.
  optional public func tpdApplePayDidCancelPayment(_ applePay: TPDApplePay!)

  // Send To The Delegate After Apple Pay Payment's Form Disappeared.
  optional public func tpdApplePayDidFinishPayment(_ applePay: TPDApplePay!)
}
```

Name	Usage
didSuccessPayment	Called when a successful payment has been made.
didFailurePayment	Called when the payment failed.
tpdApplePayDidStartPayment	Called when the payment begins.
didSelectPaymentMethod	Called when a payment method is selected.
didSelectShippingMethod	Called when a shipping method is selected.
canAuthorizePaymentWithShippingContact	Check if the payment can be authorized with the given shipping contact.

Name	Usage
tpdApplePayDidCancelPayment	Called when the payment is cancelled.
tpdApplePayDidFinishPayment	Called when the payment is finished, but before the result has been determined.
didReceivePrime:withExpiryMillis:withCardInfo:withMerchantReferenceInfo	Called when you obtained the Prime and expiry millis from TapPay. You should pass this token back to your server, and call our Pay by Prime API to finish the transaction. Afterwards, call “showPaymentResult” to display the result.
showPaymentResult	Called after you finished the payment process on your server with Pay by Prime API.

TPDTransactionResult

iOS-objC

```
@interface TPDTransactionResult : NSObject
 // message, Report Message.
 @property (nonatomic, strong) NSString *message;

 // status, Result Code, '0' Means Success.
 @property (nonatomic, assign) NSInteger status;

 // amount
 @property (nonatomic, strong) NSDecimalNumber *amount;

 // paymentMehod
 @property (nonatomic, strong) PKPaymentMethod *paymentMethod;
 @end
```

iOS-Swift

```
class TPDTransactionResult : NSObject {
  // message, Report Message.
  var message: String!

  // status, Result Code, '0' Means Success.
  var status: Int

  // amount
  var amount: NSDecimalNumber!

  // paymentMehod
  var paymentMethod: PKPaymentMethod!
}
```

The returned object in didSuccessPayment and didFailurePayment, TPDTransactionResult, will contain the following values:

Name	Usage
amount	Amount for the transaction.
status	Status code for the transaction.
message	Error message for the transaction.
paymentMehod	Apple's PKPaymentMethod

iOS-objC

```
 - (void)tpdApplePayDidStartPayment:(TPDApplePay *)applePay {
  NSLog(@"=======================================================");
  NSLog(@"Apple Pay On Start");
  NSLog(@"======================================================= \n\n");
}

- (void)tpdApplePay:(TPDApplePay *)applePay didSuccessPayment:
(TPDTransactionResult *)result {
  NSLog(@"=======================================================");
  NSLog(@"Apple Pay Did Success ==> Amount : %@", [result.amount stringValue]);
  NSLog(@"shippingContact.name : %@ %@",
applePay.consumer.shippingContact.name.givenName,
applePay.consumer.shippingContact.name.familyName);
  NSLog(@"shippingContact.emailAddress : %@",
applePay.consumer.shippingContact.emailAddress);
  NSLog(@"shippingContact.phoneNumber : %@",
applePay.consumer.shippingContact.phoneNumber.stringValue);
  NSLog(@"======================================================= \n\n");
}

- (void)tpdApplePay:(TPDApplePay *)applePay didFailurePayment: (TPDTransactionResult
*)result {
  NSLog(@"=======================================================");
  NSLog(@"Apple Pay Did Failure ==> Message : %@, ErrorCode : %ld", result.message,
(long)result.status);
  NSLog(@"======================================================= \n\n");
}

- (void)tpdApplePayDidCancelPayment:(TPDApplePay *)applePay {
  NSLog(@"=======================================================");
  NSLog(@"Apple Pay Did Cancel");
  NSLog(@"======================================================= \n\n");
}

- (void)tpdApplePayDidFinishPayment:(TPDApplePay *)applePay {
  NSLog(@"=======================================================");
  NSLog(@"Apple Pay Did Finish");
  NSLog(@"======================================================= \n\n");
}

- (void)tpdApplePay:(TPDApplePay *)applePay didSelectShippingMethod:
(PKShippingMethod *)shippingMethod {
  NSLog(@"=======================================================");
  NSLog(@"======> didSelectShippingMethod: ");
  NSLog(@"Shipping Method.identifier : %@", shippingMethod.identifier);
  NSLog(@"Shipping Method.detail : %@", shippingMethod.detail);
```

```
  NSLog(@"========================================================= \n\n");
}

- (TPDCart *)tpdApplePay:(TPDApplePay *)applePay didSelectPaymentMethod:
(PKPaymentMethod *)paymentMethod cart:(TPDCart *)cart {
  NSLog(@"=========================================================");
  NSLog(@"======> didSelectPaymentMethod: ");
  NSLog(@"========================================================= \n\n");
  if (paymentMethod.type == PKPaymentMethodTypeDebit) {
    [self.cart addPaymentItem:[TPDPaymentItem paymentItemWithItemName:@"Discount"
                                                           withAmount:[NSDecimalNumber
decimalNumberWithString:@"-1.00"]]];
  }
  return self.cart;
}

- (BOOL)tpdApplePay:(TPDApplePay *)applePay
canAuthorizePaymentWithShippingContact:(PKContact *)shippingContact {
  NSLog(@"=========================================================");
  NSLog(@"======> canAuthorizePaymentWithShippingContact ");
  NSLog(@"shippingContact.name : %@ %@",
applePay.consumer.shippingContact.name.givenName,
applePay.consumer.shippingContact.name.familyName);
  NSLog(@"shippingContact.emailAddress : %@", shippingContact.emailAddress);
  NSLog(@"shippingContact.phoneNumber : %@",
shippingContact.phoneNumber.stringValue);
  NSLog(@"========================================================= \n\n");
  return YES;
}

- (void)tpdApplePay:(TPDApplePay *)applePay didReceivePrime:(NSString *)prime
withExpiryMillis:(long)expiryMillis withCardInfo:(TPDCardInfo *)cardInfo
withMerchantReferenceInfo:(NSDictionary *)merchantReferenceInfo {

  // 1. Send Your 'Prime' To Your Server, And Handle Payment With Result
  // ...
  NSLog(@"=========================================================");
  NSLog(@"======> didReceivePrime ");
  NSLog(@"Prime : %@", prime);
  NSLog(@"apple pay %lu", expiryMillis);
  NSLog(@"totalAmount : %@",applePay.cart.totalAmount);
  NSLog(@"Client  IP : %@",applePay.consumer.clientIP);
  NSLog(@"shippingContact.name : %@ %@",
applePay.consumer.shippingContact.name.givenName,
applePay.consumer.shippingContact.name.familyName);
  NSLog(@"shippingContact.emailAddress : %@",
applePay.consumer.shippingContact.emailAddress);
  NSLog(@"shippingContact.phoneNumber : %@",
applePay.consumer.shippingContact.phoneNumber.stringValue);
  PKPaymentMethod * paymentMethod = applePay.consumer.paymentMethod;
  NSLog(@"Type : %ld", (long)paymentMethod.type);
  NSLog(@"Network : %@", paymentMethod.network);
  NSLog(@"Display Name : %@", paymentMethod.displayName);
  NSLog(@"========================================================= \n\n");

  // 2. If Payment Success, applePay.
  BOOL paymentResult = YES;
  [applePay showPaymentResult:paymentResult];
}
```

iOS-Swift

```
        extension ViewController :TPDApplePayDelegate {

  func tpdApplePayDidStartPayment(_ applePay: TPDApplePay!) {
    print("=====================================================")
    print("Apple Pay On Start")
    print("=====================================================\n\n")
  }

  func tpdApplePay(_ applePay: TPDApplePay!, didSuccessPayment result:
TPDTransactionResult!) {
    print("=====================================================")
    print("Apple Pay Did Success ==> Amount : \(result.amount.stringValue)")
    print("shippingContact.name : \
(applePay.consumer.shippingContact?.name?.givenName) \
( applePay.consumer.shippingContact?.name?.familyName)")
    print("shippingContact.emailAddress : \
(applePay.consumer.shippingContact?.emailAddress)")
    print("shippingContact.phoneNumber : \
(applePay.consumer.shippingContact?.phoneNumber?.stringValue)")
    print("Shipping Method.identifier : \(applePay.cart.shippingMethod.identifier)")
    print("Shipping Method.detail : \(applePay.cart.shippingMethod.detail)")
    print("=====================================================\n\n")
  }

  func tpdApplePay(_ applePay: TPDApplePay!, didFailurePayment result:
TPDTransactionResult!) {
    print("=====================================================")
    print("Apple Pay Did Failure ==> Message : \(result.message), ErrorCode : \
(result.status)")
    print("=====================================================\n\n")
  }

  func tpdApplePayDidCancelPayment(_ applePay: TPDApplePay!) {
    print("=====================================================")
    print("Apple Pay Did Cancel")
    print("=====================================================\n\n")
  }

  func tpdApplePayDidFinishPayment(_ applePay: TPDApplePay!) {
    print("=====================================================")
    print("Apple Pay Did Finish")
    print("=====================================================\n\n")
  }

  func tpdApplePay(_ applePay: TPDApplePay!, didSelect shippingMethod:
PKShippingMethod!) {
    print("=====================================================")
    print("======> didSelectShippingMethod: ")
    print("Shipping Method.identifier : \(shippingMethod.identifier)")
    print("Shipping Method.detail : \(shippingMethod.detail)")
    print("=====================================================\n\n")
  }

  func tpdApplePay(_ applePay: TPDApplePay!, didSelect paymentMethod:
PKPaymentMethod!, cart: TPDCart!) -> TPDCart! {
    print("=====================================================");
    print("======> didSelectPaymentMethod: ");
```

```
    print("=====================================================\n\n");
    if paymentMethod.type == .debit {
      self.cart.add(TPDPaymentItem(itemName: "Discount", withAmount:
NSDecimalNumber(string: "-1.00")))
    }
    return self.cart;
  }

  func tpdApplePay(_ applePay: TPDApplePay!,
canAuthorizePaymentWithShippingContact shippingContact: PKContact!) -> Bool {
    print("=====================================================")
    print("======> canAuthorizePaymentWithShippingContact ")
    print("shippingContact.name : \
(applePay.consumer.shippingContact?.name?.givenName) \
( applePay.consumer.shippingContact?.name?.familyName)")
    print("shippingContact.emailAddress : \(shippingContact.emailAddress)")
    print("shippingContact.phoneNumber : \(shippingContact.phoneNumber?.stringValue)")
    print("=====================================================\n\n")
    return true;
  }

  tpdApplePay(_ applePay: TPDApplePay!, didReceivePrime prime: String!, withExpiryMillis
expiryMillis: Int, withCardInfo cardInfo: TPDCardInfo, withMerchantReferenceInfo
merchantReferenceInfo: [AnyHashable : Any]!) {

    // 1. Send Your Prime To Your Server, And Handle Payment With Result
    // ...
    print("=====================================================");
    print("======> didReceivePrime");
    print("Prime : \(prime!)");
    print("Expiry Millis : \(expiryMillis)");
    print("total Amount :   \(applePay.cart.totalAmount!)")
    print("Client IP : \(applePay.consumer.clientIP!)")
    print("shippingContact.name : \
(applePay.consumer.shippingContact?.name?.givenName) \
(applePay.consumer.shippingContact?.name?.familyName)");
    print("shippingContact.emailAddress : \
(applePay.consumer.shippingContact?.emailAddress)");
    print("shippingContact.phoneNumber : \
(applePay.consumer.shippingContact?.phoneNumber?.stringValue)");
    let paymentMethod = self.consumer.paymentMethod!
    print("type : \(paymentMethod.type.rawValue)")
    print("Network : \(paymentMethod.network!.rawValue)")
    print("Display Name : \(paymentMethod.displayName!)")
    print("===================================================== \n\n");

    // 2. If Payment Success, applePay.
    let paymentResult = true;
    applePay.showPaymentResult(paymentResult)
    }
}
```

Name	Type	Usage
prime	String	prime token used in Pay by Prime
expiryMillis	Int	prime expired millis
cardInfo	TPDCardInfo	Card information For the format, please refer to the table shown below
merchantReferenceInfo	JSONObject	If the merchant uses the co-branded card management in the TapPay portal, and the transaction card number meets the setting, this parameter will be returned. Set in TapPay Portal, must limit 20 character and half alphanumeric.

cardInfo format

Name	Type (Max)	Usage
bincode	String(6)	First six digits of the card
lastFour	String(4)	Last four digits of the card
issuer	String	Card issuer
issuerZhTw	String	Card issuer chinese name
bankId	String	Bank identifier
funding	int	Card usage -1 = Unknown 0 = credit card 1 = debit card 2 = prepaid card
cardType	int	Card type -1 = Unknown 1 = VISA 2 = MasterCard 3 = JCB 4 = Union Pay 5 = AMEX
level	String	Card level
country	String	Country of card issuer
countryCode	String	Country code of card issuer

merchantReferenceInfo format

Name	Type	Usage
affiliateCodes	Array	Affiliated codes set by the merchant in the Co-brand card management area of the TapPay portal.

PKPaymentButton

iOS-objC

```
@interface ViewController () <TPDApplePayDelegate>
@property (nonatomic, strong) PKPaymentButton *applePayButton;
@end
```

iOS-Swift

```
class ViewController: UIViewController {
    var applePayButton : PKPaymentButton!
}
```

You should check if the card or the device supports Apple Pay before setting up the payment button.

iOS-objC

```
/**
[TPDApplePay canMakePaymentsUsingNetworks:(NSArray<PKPaymentNetwork> *)]
  Check support cart network.

  TPDApplePay setupWthMerchant:(TPDMerchant *) withConsumer:(TPDConsumer *)
withCart:(TPDCart *) withDelegate:(id)
  Use TPDMerchant, TPDConsumer, TPDCart to initial Apple Pay Instance.

  startPayment
  Start Apple Pay Payment Sheet
*/

- (void)paymentButtonSetting {

// Check if the card can use Apple Pay.
    if ([TPDApplePay canMakePaymentsUsingNetworks:self.merchant.supportedNetworks])
{
        self.applePayButton = [PKPaymentButton
buttonWithType:PKPaymentButtonTypeBuy style:PKPaymentButtonStyleBlack];
    }else{
        self.applePayButton = [PKPaymentButton
buttonWithType:PKPaymentButtonTypeSetUp style:PKPaymentButtonStyleBlack];
    }

    [self.view addSubview:self.applePayButton];
    self.applePayButton.center = self.view.center;

    [self.applePayButton addTarget:self action:@selector(didClickButton:)
forControlEvents:UIControlEventTouchUpInside];
}

- (void)didClickButton:(PKPaymentButton *)sender {
    self.applePay = [TPDApplePay setupWthMerchant:self.merchant
withConsumer:self.consumer withCart:self.cart withDelegate:self];
    [self.applePay startPayment];
}
```

iOS-Swift

```
/**
  TPDApplePay.canMakePayments(usingNetworks: [PKPaymentNetwork]!)
  Check support cart network.

  TPDApplePay.setupWthMerchant(TPDMerchant!, with: TPDConsumer!, with: TPDCart!,
withDelegate: Any!)
  Use TPDMerchant, TPDConsumer, TPDCart to initial Apple Pay Instance.

  startPayment
  Start Apple Pay Payment Sheet
*/

func paymentButtonSetting() {

    // Check if the card can use Apple Pay.
    if (TPDApplePay.canMakePayments(usingNetworks: self.merchant.supportedNetworks))
{
        applePayButton = PKPaymentButton.init(paymentButtonType: .buy,
paymentButtonStyle: .black)
    } else {
        applePayButton = PKPaymentButton.init(paymentButtonType: .setUp,
paymentButtonStyle: .black)
    }

    view.addSubview(applePayButton)
    applePayButton.center = view.center

    applePayButton.addTarget(self, action:
#selector(ViewController.didClickBuyButton(sender:)), for: .touchUpInside)

}

@objc func didClickBuyButton(sender:PKPaymentButton) {

    applePay = TPDApplePay.setupWthMerchant(merchant, with: consumer, with: cart,
withDelegate: self)

    applePay.startPayment()

}
```

TPDMerchant

iOS-objC

```
@interface ViewController () <TPDApplePayDelegate>
@property (nonatomic, strong) TPDMerchant *merchant;
@end
```

iOS-Swift

```
class ViewController: UIViewController {
    var merchant : TPDMerchant!
}
```

iOS-objC

```
self.merchant = [TPDMerchant new];
self.merchant.merchantName              = @"TapPay!";
self.merchant.merchantCapability        = PKMerchantCapability3DS;
self.merchant.applePayMerchantIdentifier = @"merchant.apple.pay";
self.merchant.countryCode               = @"TW";
self.merchant.currencyCode              = @"TWD";
self.merchant.supportedNetworks         = @[PKPaymentNetworkAmex,
PKPaymentNetworkMasterCard, PKPaymentNetworkVisa, PKPaymentNetworkJCB];
```

iOS-Swift

```
merchant = TPDMerchant()
merchant.merchantName              = "TapPay!";
merchant.merchantCapability        = .capability3DS;
merchant.applePayMerchantIdentifier = "merchant.apple.pay";
merchant.countryCode               = "TW";
merchant.currencyCode              = "TWD";
merchant.supportedNetworks         = [.amex, .masterCard, .visa, .JCB]
```

Name	Values
merchantName	Display merchant name on Payment Sheet.
merchantCapability	The payment capabilities supported by the merchant, please refer to Apple Pay merchantCapabilities.
applePayMerchantIdentifier	The Apple Merchant ID.
countryCode	The abbreviation code for the country following the ISO 3166-1 alpha-2 format.
currencyCode	The abbreviation code for the country currency following the ISO 4217 format.
supportedNetworks	Set the credit card network you want to support

supportedNetworks card network supported version

Card network	Version
American Express	iOS 8.0+ watchOS 3.0+
MasterCard	iOS 8.0+ watchOS 3.0+
Visa	iOS 8.0+ watchOS 3.0+
JCB	iOS 10.0+ watchOS 3.1+

TPDConsumer

iOS-objC

```
@interface ViewController () <TPDApplePayDelegate>
@property (nonatomic, strong) TPDConsumer *consumer;
@end
```

iOS-Swift

```
class ViewController: UIViewController {
    var consumer : TPDConsumer!
}
```

iOS-objC

```
self.consumer = [TPDConsumer new];
self.consumer.requiredShippingAddressFields  = PKAddressFieldEmail |
PKAddressFieldName | PKAddressFieldPhone;
self.consumer.requiredBillingAddressFields   = PKAddressFieldPostalAddress;
```

iOS-Swift

```
consumer = TPDConsumer()
consumer.requiredShippingAddressFields  = [.email, .name, .phone]
consumer.requiredBillingAddressFields   = [.postalAddress]
```

Name	Content
requiredShippingAddressFields	Required shipping address information field. Default no fields required.
requiredBillingAddressFields	Required billing address information field. Default no fields required.

After the Apple Pay authentication is successful, you can get the card information in the didReceivePrime delegate, TPDConsumer.paymentMethod
More info refer to apple pay docs PKPaymentMethod.

Name	Content
type	Card type 0 : unknown 1 : debit 2 : credit 3 : prepaid 4 : store
network	Card network
displayName	Card network and card last four number

TPDCart

iOS-objC

```
@interface ViewController () <TPDApplePayDelegate>
@property (nonatomic, strong) TPDCart *cart;
@end
```

iOS-Swift

```
class ViewController: UIViewController {
    var cart     : TPDCart!
}
```

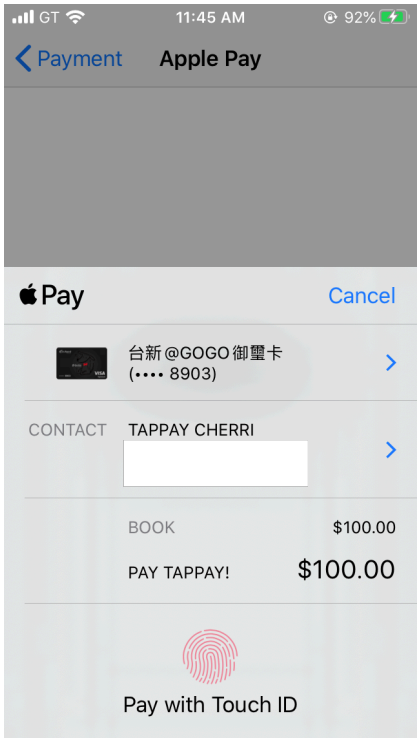
iOS-objC

```
/**
  [cart addPaymentItem:(TPDPaymentItem *)]
  Add payment item to item list.
*/
self.cart = [TPDCart new];

TPDPaymentItem *book = [TPDPaymentItem paymentItemWithItemName:@"Book"
                                                    withAmount:[NSDecimalNumber
decimalNumberWithString:@"100.00"]];
[self.cart addPaymentItem:book];

self.cart.shippingType   = PKShippingTypeShipping;
```

iOS-Swift

```
/**
  cart.add(TPDPaymentItem!)
  Add payment item to item list.
*/

cart = TPDCart()
let book = TPDPaymentItem(itemName:"Book",
                          withAmount: NSDecimalNumber(string: "100.00"))
cart.add(book)
cart.shippingType   = .shipping;
```

You can use TPDPaymentItem to add new items to TPDCart.
Use paymentItemWithItemName to set up to display the amount of items.
The sum of amount of items in TPDCart must be greater than zero, otherwise the program will return error code: 88016



Deferred Payment

iOS-objC

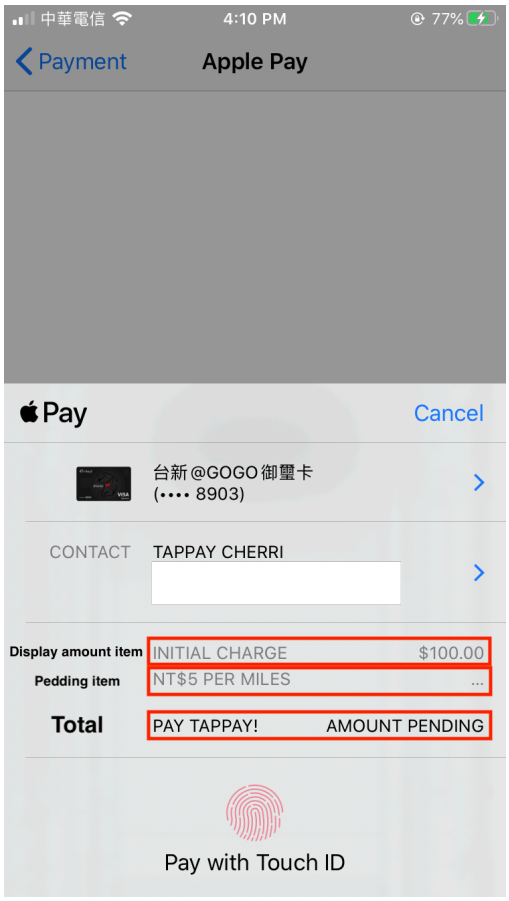
```
self.cart = [TPDCart new];
self.cart.isAmountPending = YES;
self.cart.isShowTotalamount = NO;

TPDPaymentItem *initialCharge = [TPDPaymentItem paymentItemWithItemName:@"Initial
Charge"
                withAmount:[NSDecimalNumber decimalNumberWithString:@"100.00"]];
[self.cart addPaymentItem:initialCharge];

TPDPaymentItem *improvementSurcharge  = [TPDPaymentItem
pendingPaymentItemWithItemName:@"NT$5 per miles"];
[self.cart addPaymentItem:improvementSurcharge];
```

iOS-Swift

```
cart = TPDCart()
cart.isAmountPending = true
cart.isShowTotalAmount = false

let initialCharge = TPDPaymentItem(itemName: "Initial Charge", withAmount:
NSDecimalNumber(string: "100.00"))
cart.add(initialCharge)

let improvementSurcharge = TPDPaymentItem.pendingPaymentItem(withItemName:
"NT$5 per miles")
cart.add(improvementSurcharge)
```

Please contact TapPay Customer Service (support@cherri.tech) to activate Apple Pay deferred payment.
Once the function has been activated, the prime could live for 30 days and payment amount is set by Pay by Prime, not by SDK.
The property isAmountPending of TPDCart could determine whether this transaction enables deferred payment or not. Default is false
The property isShowTotalAmount of TPDCart could determine that total amount of the payment sheet shows the sum of each payment item’s amount or AMOUNT PENDING. Default is false
If the property pendingPaymentItemWithItemName has been set in TPDCart, the payment item’s amount of the payment sheet will show 『・・・』, which means actual payment amount is unknown.

Use deferred payment, please note the following rules



All items in TPDCart have specific amount.

isAmountPending	isShowTotalAmount	Total field	Prime expire time
TRUE	TRUE	Display Total Amount	30days
TRUE	FALSE	Display Amount Pending	30days
FALSE	TRUE	Display Total Amount	90sec
FALSE	FALSE	Not Support, refer to error code: 88013	X

All items in TPDCart are amount pending.

isAmountPending	isShowTotalAmount	Total field	Prime expire time
TRUE	TRUE	Not Support, refer to error code:88014	X
TRUE	FALSE	Display Amount Pending	30days
FALSE	TRUE	Not Support, refer to error code: 88015	X
FALSE	FALSE	Not Support, refer to error code: 88013	X

The items in TPDCart have specific amount and amount pending.

isAmountPending	isShowTotalAmount	Total field	Prime expire time
TRUE	TRUE	Display Total Amount	30days
TRUE	FALSE	Display Amount Pending	30days
FALSE	TRUE	Not Support, refer to error code: 88015	X
FALSE	FALSE	Not Support, refer to error code: 88013	X

Notice

Due to total amount should be set above 0 in iOS 12 and version lower.
For compatibility, iOS SDK v2.6.0 and version higher will set total amount of 1 to Apple Pay in following situation.

1. The items of TPDCart are amount pending items and total amount is set to show 『AMOUNT PENDING』(self.cart.isAmountPending = true).
2. The total amount of items of TPDCart is 0 and total amount is set to show 『AMOUNT PENDING』(self.cart.isAmountPending = true)

Example

If you have any questions, please refer our Apple Pay Example.

Apple Pay on the Web

If you want to know the latest SDK version and the difference between each version, please refer to Github Release Page: TapPay Web Github

Apple Pay JavaScript API is only available on iOS device running iOS 10 or later or Mac running macOS Sierra (10.12) or later.

If you enable Apple Pay on the Web Merchant Application Service, please fill in merchant identifier which generated by TapPay (You can find it from Portal > Payment Methods > Apple Pay On The Web)

Requirements

In Apple Pay on the Web, you must meet the following requirements:

1. All pages that include Apple Pay must be served over HTTPS , In production and sandbox. In develop enviroment you can use ngrok to set up an SSH tunnel , you can refer to our Ngrok tutorial.

2. Your domain must have a valid SSL certificate.

3. Your server must support the Transport Layer Security (TLS) 1.2 protocol and one of the cipher suites listed below:

Cipher Suite Value	Description
0xC02F	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
0xC027	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
0xC013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
0x009E	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
0x0067	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
0x009C	TLS_RSA_WITH_AES_128_GCM_SHA256
0x003C	TLS_RSA_WITH_AES_128_CBC_SHA256

4. Your server must allow access over HTTPS (TCP over port 443) to the Apple Pay IP Addresses provided below:

IP	Name	Type
17.171.85.7	apple-pay-gateway-cert.apple.com	Sandbox
17.171.78.7	apple-pay-gateway-nc-pod1.apple.com	Production
17.171.78.71	apple-pay-gateway-nc-pod2.apple.com	Production

IP	Name	Type
17.171.78.135	apple-pay-gateway-nc-pod3.apple.com	Production
17.171.78.199	apple-pay-gateway-nc-pod4.apple.com	Production
17.171.79.12	apple-pay-gateway-nc-pod5.apple.com	Production
17.141.128.7	apple-pay-gateway-pr-pod1.apple.com	Production
17.141.128.71	apple-pay-gateway-pr-pod2.apple.com	Production
17.141.128.135	apple-pay-gateway-pr-pod3.apple.com	Production
17.141.128.199	apple-pay-gateway-pr-pod4.apple.com	Production
17.141.129.12	apple-pay-gateway-pr-pod5.apple.com	Production
17.171.78.9	apple-pay-gateway-nc-pod1-dr.apple.com	Production
17.171.78.73	apple-pay-gateway-nc-pod2-dr.apple.com	Production
17.171.78.137	apple-pay-gateway-nc-pod3-dr.apple.com	Production
17.171.78.201	apple-pay-gateway-nc-pod4-dr.apple.com	Production
17.171.79.13	apple-pay-gateway-nc-pod5-dr.apple.com	Production
17.141.128.9	apple-pay-gateway-pr-pod1-dr.apple.com	Production
17.141.128.73	apple-pay-gateway-pr-pod2-dr.apple.com	Production
17.141.128.137	apple-pay-gateway-pr-pod3-dr.apple.com	Production
17.141.128.201	apple-pay-gateway-pr-pod4-dr.apple.com	Production
17.141.129.13	apple-pay-gateway-pr-pod5-dr.apple.com	Production

SetupSDK

Please use following html code to setup payment environment.

Please be aware of the differences of each Web SDK version to avoid loading errors.
https://js.tappaysdk.com/sdk/tpdirect/v5.14.0
https://js.tappaysdk.com/tpdirect/v5.13.1
Please refer to the following usage examples.

```
<!--
    Use TPDirect.setupSDK(appID, appKey, serverType)
    to setup APP ID, App Key, Server Type parameters
-->

<script src="https://js.tappaysdk.com/sdk/tpdirect/v5.14.0"></script>
<script>
    TPDirect.setupSDK(APP_ID, 'APP_KEY', 'sandbox')
</script>
```

When use web SDK version less than v5.14.0, please use the following path to include the web SDK

```
<script src="https://js.tappaysdk.com/tpdirect/v5.13.1"></script>
<script>
    TPDirect.setupSDK(APP_ID, 'APP_KEY', 'sandbox')
</script>
```

Name	Usage
appID	Please refer to appid
appKey	Please refer to appkey
serverType	Use ‘sandbox’ for sandbox environment.

checkAvailability()

Check if PaymentRequestAPI is available.

```
/**
    Use TPDirect.paymentRequestApi.checkAvailability() to detect user's browser
    whether is available to use Payment Request API or not.
*/
TPDirect.paymentRequestApi.checkAvailability()
```

Setup Apple Pay

```
/**
    Use TPDirect.paymentRequestApi.setupApplePay to setup parameters about Apple Pay.
*/
TPDirect.paymentRequestApi.setupApplePay({
    merchantIdentifier: 'merchant.tech.cherri',
    countryCode: 'TW'
})
```

Name (*=required)	Type	Content
merchantIdentifier*	String	You registed in Apple Developer Merchant Id If you enable Apple Pay on the Web Merchant Application Service, please fill in merchantIdentifier which generated by TapPay (You can find it from Portal > Payment Methods > Apple Pay > Apple Pay on the Web)
countryCode	String	countryCode , default is TW

PaymentRequest

Create a paymentRequest object setup transaction info to retrieve the Prime.

```
var paymentRequest = {
    supportedNetworks: ['AMEX', 'JCB', 'MASTERCARD', 'VISA'],
    supportedMethods: ['apple_pay'],
    displayItems: [{
        label: 'TapPay - iPhone8',
        amount: {
            currency: 'TWD',
            value: '1.00'
        }
    }],
    total: {
        label: '總金額',
        amount: {
            currency: 'TWD',
            value : '1.00'
        }
    },
    shippingOptions: [{
            id: "standard",
            label: "🚚 Ground Shipping (2 days)",
            detail: 'Estimated delivery time: 2 days',
            amount: {
                currency: "TWD",
                value: "5.00"
            }
        },
        {
            id: "drone",
            label: "🚀 Drone Express (2 hours)",
            detail: 'Estimated delivery time: 2 hours',
            amount: {
                currency: "TWD",
                value: "25.00"
            }
        }
    ],
    options: {
      requestPayerEmail: true,
      requestPayerName: true,
      requestPayerPhone: true,
      requestShipping: true,
      shippingType: 'shipping'
    }
}
```

Name	Type	Usage
supportedNetworks	Array	Setup supported networks. Support parameters “VISA"、"MASTERCARD"、"JCB” and “AMEX”.
supportedMethods	Array	Setup supported methods
displayItems	Array	For the format, please refer to the table shown below
total	JSONObject	For the format, please refer to the table shown below
shippingOptions	JSONObject	For the format, please refer to the table shown below
options	JSONObject	For the format, please refer to the table shown below

displayItems format

Name	Type	Usage
label	String	Item Name
amount	JSONObject	value: item amount currency: item currency
isAmountPending	Boolean	Whether to use apple pay deferred payment, default is false. If isAmountPending is true,transaction amount will show “Amount Pending”. You can only choose isAmountPending or amount.value for the same item More detail please refer to Deferred Payment

total format

Name	Type	Usage
label	String	Transaction Summary
amount	JSONObject	value: total amount currency: currency
isAmountPending	Boolean	Whether to use apple pay deferred payment, and default is false. More detail please refer to
isShowTotalAmount	Boolean	Apple Pay total field display amount or AMOUNT PENDING, default is true You can only choose isAmountPending or amount.value for the same item More detail please refer to Deferred Payment

shippingOptions format

Name	Type	Usage
id	String	Shipping Name
label	String	Shipping type description
detail	String	Shipping detail description (Apple Pay Only)
amount	JSONObject	value: Shipping amount currency: Shipping currency

options format

Name	Type	Usage
requestPayerEmail	Bool	Request Payer Email
requestPayerName	Bool	Request Payer Name
requestPayerPhone	Bool	Request Payer Phone
requestShipping	Bool	Request Payer Shipping address
shippingType	String	Shipping type , If you want to setup please refer to shippingType shipping , delivery , pickup

After you setup payment request object, please use following method to setup Payment Request API

```
/**
    Use TPDirect.paymentRequestApi.setupPaymentRequest to setup payment requset object
    and return object which decide the user's browser support Payment Request API
    and the user have the card to make payments or not in callback function
*/

TPDirect.paymentRequestApi.setupPaymentRequest(paymentRequest, function (result) {
    if (!result.browserSupportPaymentRequest) {
        console.log('Browser not support PaymentRequest')
        return
    }
    if (result.canMakePaymentWithActiveCard === true) {
        console.log('Current device have supported card')
    } else {
        console.log('Current device without supported card to make payment')
    }
})
```

setupPaymentRequest have a callback function, callback function have following object properties.

Name	Type	Usage
result	JSONObject	For the format, please refer to the table shown below

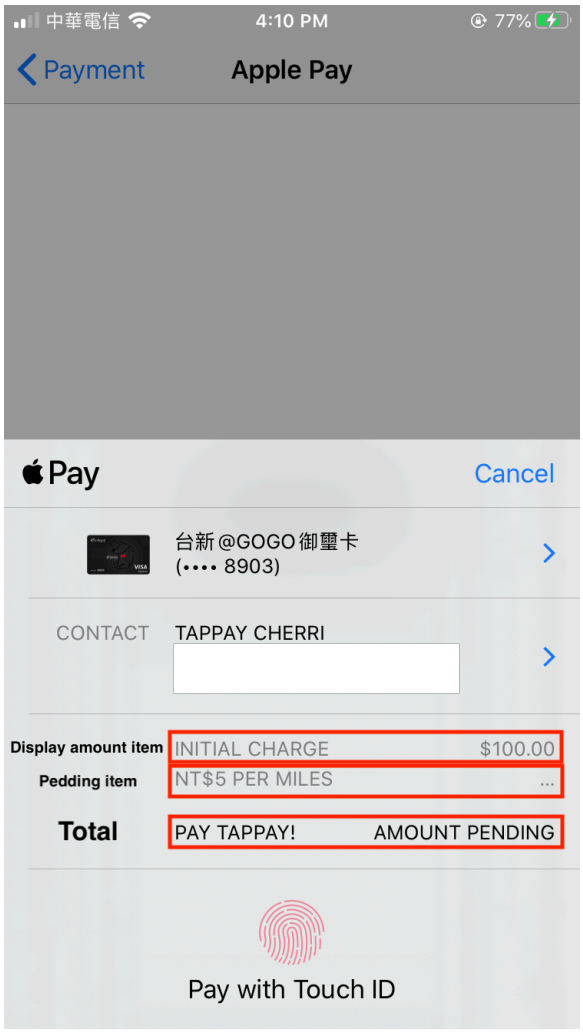
result format

Name	Type	Usage
browserSupportPaymentRequest	Boolean	Does Browser support Payment Request API true: Support false: Not Support
canMakePaymentWithActiveCard	Boolean	Does this device have card to make payment Apple pay only check user have setup card in Apple Pay. true: The device have supported card false: The device don’t have supported card

Deferred Payment

Please contact TapPay Customer Service (support@cherri.tech) to activate Apple Pay deferred payment.
Once the function has been activated, the prime could live for 30 days and payment amount is set by Pay by Prime, not by SDK.
The property total.isAmountPending of paymentRequest could determine whether this transaction enables deferred payment or not. Default is false
The property total.isShowTotalAmount of paymentRequest could determine that total amount of the payment sheet shows the sum of each payment item’s amount or AMOUNT PENDING. Default is false
If isAmountPending:true has been set in displayItems of paymentRequest, the payment item’s amount of the payment sheet will show 『・・・』, which means actual payment amount is unknown.

Use deferred payment, please note the following rules



All of the displayItems isAmountPending is false.

total.isAmountPending	total.isShowTotalAmount	Total field	Prime expire time
TRUE	TRUE	Display Total Amount	30days
TRUE	FALSE	Display Amount Pending	30days
FALSE	TRUE	Display Total Amount	90sec
FALSE	FALSE	Not Support	X

All of the displayItems isAmountPending is true.

total.isAmountPending	total.isShowTotalAmount	Total field	Prime expire time
TRUE	TRUE	Not Support	X
TRUE	FALSE	Display Amount Pending	30days
FALSE	TRUE	Not Support	X
FALSE	FALSE	Not Support	X

All of the displayItems isAmountPending have both.

total.isAmountPending	total.isShowTotalAmount	Total field	Prime expire time
TRUE	TRUE	Display Total Amount	30days
TRUE	FALSE	Display Amount Pending	30days
FALSE	TRUE	Not Support	X
FALSE	FALSE	Not Support	X

Notice

Due to total amount should be set above 0 in MacOS 10.14.1, iOS 12.1 and version lower.
For compatibility, Web SDK v5.4.0 and version higher will set total.amount.value:1 to Apple Pay in following situation.

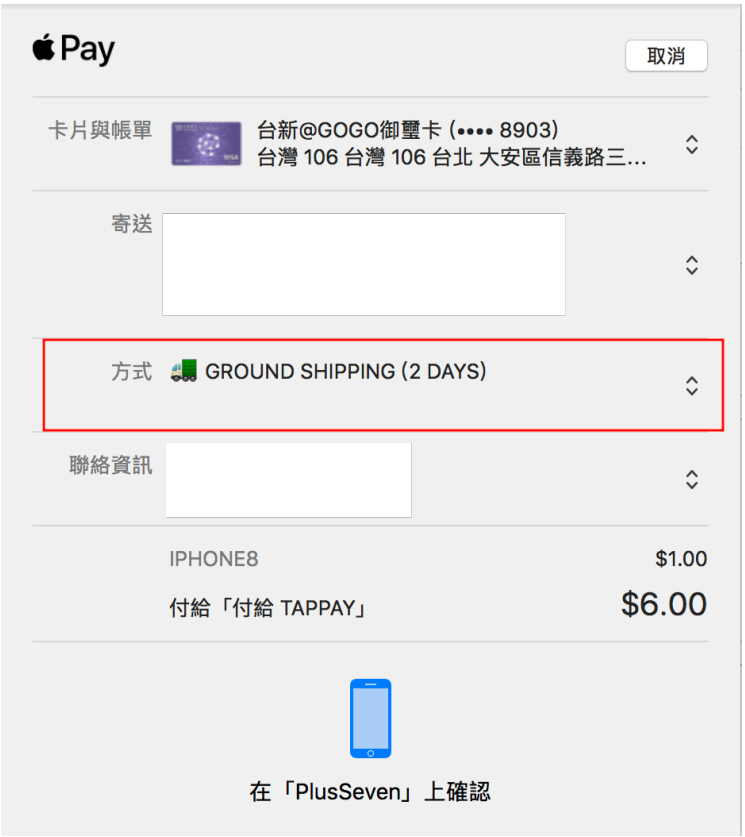
1. isAmountPending:true has been set in all displayItems of paymentRequest and total amount is set to show 『AMOUNT PENDING』 (total.isShowTotalAmount:false)
2. The total of displayItems' amount is 0 and total amount is set to show 『AMOUNT PENDING』(total.isShowTotalAmount:false)
3. There is no items in displayItems and total amount is set to show『AMOUNT PENDING』(total.isShowTotalAmount:false)

Besides, if you pay on Mac, but verify by fingerprint or face ID through iPhone, total amount of the payment sheet will show 1.

shippingOptions

In paymentRequest ShippingOption without detail.

```
shippingOptions: [{
    id: "standard",
    label: "🚚 Ground Shipping (2 days)",
    amount: {
        currency: "TWD",
        value: "5.00"
    }
}]
```



In paymentRequest ShippingOption add detail.

```
shippingOptions: [{
    id: "standard",
    label: "🚚 Ground Shipping (2 days)",
    // apple pay only
    detail: "Estimated delivery time: 2 days",
    amount: {
        currency: "TWD",
        value: "5.00"
    }
}]
```

Get Prime

```
/**
    Use TPDirect.paymentRequestApi.getPrime to get prime from TapPay.
*/

TPDirect.paymentRequestApi.getPrime(function(result) {
    console.log('paymentRequestApi.getPrime result', result)
    if (result.status !== 0) {
        console.error('getPrime failed: ' + result.msg)
        return
    }
    var prime = result.prime
})
```

Get Prime Result

Get the result and send the Prime back to your server to finish the transaction by calling Pay by Prime API.

```
{
    "prime": String,
    "total_amount": String,
    "status": Int,
    "msg": String,
    "client_ip": String,
    "payer": {
        "name": String,
        "phone": String,
        "email": String
    },
    "shippingAddress": {
        "country": String,
        "addressLine": ArrayString,
        "region": String,
        "city": String,
        "dependentLocality": String,
        "postalCode": String,
        "sortingCode": String,
        "languageCode": String,
        "organization": String,
        "recipient": String,
        "phone": String
    },
    "shippingOption": String,
    "billingAddress": {
        "country": String,
        "addressLine": ArrayString,
        "region": String,
        "city": String,
        "dependentLocality": String,
        "postalCode": String,
        "sortingCode": String,
        "languageCode": String,
        "organization": String,
        "recipient": String,
        "phone": String
    },
    "methodName": String ("apple_pay")
    "requestId": String,
    "card": {
        "lastfour": String,
        "funding": Int,
        "type": Int,
    },
    "card_info": {
        "bincode": String,
        "lastfour": String,
        "issuer": String,
        "issuer_zh_tw": String,
        "bank_id": String,
        "funding": Int,
        "type": Int,
```

```
            "level": String,
            "country": String,
            "countrycode", String,
        },
        "merchant_reference_info": {
            "affiliate_codes": ArrayString,
        },
        "prime_expiry_millis": Number
    }
```

Name	Type	Usage
prime	String	A one-time token that represents a customer's card. You would need this token to pay in the Pay by Prime API.
total_amount	String	This Transaction total amount.
status	Int	Response code. 0 indicates success.
msg	String	Error message
client_ip	String	Customer IP Address
payer	JSONObject	Customer Information For the format, please refer to the table shown below
shippingAddress	JSONObject	Delivery address information For the format, please refer to the table shown below
shippingOption	String	Delivery method
billingAddress	JSONObject	For the format, please refer to the table shown below
methodName	String	("card" , "android_pay" , "google_pay" or "apple_pay")
requestId	String	Unique payment request identifier.
card	JSONObject	For the format, please refer to the table shown below
card_info	JSONObject	Card information For the format, please refer to the table shown below
merchant_reference_info	JSONObject	If the merchant uses the co-branded card management in the TapPay portal, and the transaction card number meets the setting, this parameter will be returned. Set in TapPay Portal, must limit 20 character and half alphanumeric. For the format, please refer to the table shown below
prime_expiry_millis	Number	Prime Expiry Date Fiexd return parameter on latest Web SDK version (From v5.3)

payer format

Name	Type	Usage
name	String	Customer’s name
phone	String	Customer’s phone number
email	String	Customer’s email

shippingAddress / billingAddress format

Name	Type
country	String
addressLine	ArrayString
region	String
city	String
dependentLocality	String
postalCode	String
sortingCode	String
languageCode	String
organization	String
recipient	String
phone	String

card format

Name	Type	Usage
lastfour	String	Last four digits of the card
funding	Int	Card usage 0 = Credit Card 1 = Debit Card 2 = Prepaid Card
type	Int	Card type 1 = VISA 2 = MasterCard 3 = JCB 4 = Union Pay 5 = AMEX

card_info format

Name	Type	Usage
bincode	String(6)	First six digits of the card
lastfour	String(4)	Last four digits of the card
issuer	String	Card issuer
issuer_zh_tw	String	Issuer chinese name
bank_id	String	Bank identifier
funding	int	Card usage -1 = Unknown 0 = credit card 1 = debit card 2 = prepaid card
type	int	Card type -1 = Unknown 1 = VISA 2 = MasterCard 3 = JCB 4 = Union Pay 5 = AMEX
level	String	Card level
country	String	Country of card issuer
countrycode	String	Country code of card issuer

merchant_reference_info format

Name	Type	Usage
affiliate_codes	Array	Affiliated codes set by the merchant in the Co-brand card management area of the TapPay portal.

Example

If you have any questions, please refer our Example or Apple's Apple Pay Example. Keep in mind that Apple Pay JavaScript API is only available on iOS device running iOS 10 or later or Mac running macOS Sierra (10.12) or later.

Reference

If you want to integrate more payment methods, please refer to Payment Request API

Q&A

If you have to set frame-src of Content Security Policy, please set following two domains.

1. js.tappaysdk.com
2. fraud.tappaysdk.com

Error

Web SDK Error Code

status	msg
-7	[Google Pay] failed to get prime
-6	[Apple Pay] failed to get prime
-5	Unknown Error
-3	failed to get prime
-1	SDK Loading Error
-	-
0	Success
5	Wrong JSON format
-	-
11	App ID not found
12	App name mismatch
13	Unknown app error
16	App key mismatch
-	-
41	Wrong Card Format
-	-
80	Invaild x-api-key or app key
81	Partner not found
84	Partner unauthorized
-	-
122	Card encrypt error
-	-
401	Request Cancel, please see the defail of msg
402	Can not obtain payment data, please see the defail of msg
403	[Apple Pay] Get Session error: {Message from apple}
404	[Payment Request API] Unexcept Error, for detail please see originalError
421	Gateway Timeout

status	msg
422	Authorization Timeout
-	-
530	Invalid arguments : app_id
531	Missing arguments : app_key
532	Missing arguments : app_name
533	Missing arguments : card_number
534	Missing arguments : card_expiry_date
570	Out of range : app_key
571	Out of range : app_name
572	Out of range : card_number
573	Out of range : card_expiry_date
574	Out of range : card_ccv
589	Invalid arguments : platform_type
590	Missing arguments : android_merchant_id
591	Missing arguments : pay_token_data
592	Missing arguments : apple_merchant_id
594	Out of range : apple_merchant_id
598	Out of range : android_merchant_id
599	Missing arguments : pay_token_data > ephemeralPublicKey
600	Missing arguments : pay_token_data > encryptedMessage
601	Missing arguments : pay_token_data > tag
602	Out of range : pay_token_data > ephemeralPublicKey
603	Out of range : pay_token_data > encryptedMessage
604	Out of range : pay_token_data > tag
605	Missing arguments : pay_token_data > data
606	Missing arguments : pay_token_data > version
607	Missing arguments : pay_token_data > signature
608	Missing arguments : pay_token_data > header > ephemeralPublicKey
609	Missing arguments : pay_token_data > header > publicKeyHash
610	Missing arguments : pay_token_data > header > transactionId
611	Out of range : pay_token_data > data

status	msg
612	Out of range : pay_token_data > version
613	Out of range : pay_token_data > header > ephemeralPublicKey
614	Out of range : pay_token_data > header > publicKeyHash
615	Out of range : pay_token_data > header > transactionId
616	Missing arguments : pay_token_data > header
617	Out of range : fraud_id
618	Out of range : pay_token_data > protocolVersion
619	Out of range : pay_token_data > signature
626	Invalid arguments : platform_type
634	Missing arguments : pay_token_data > protocolVersion
638	Missing arguments : merchant_app_launch_uri
639	Out of range : merchant_app_launch_uri
655	Missing arguments : reference_id
660	Out of range : reference_id
-	-
915	System error, please contact TapPay customer service
916	Signature verification is not proceed
917	Signature verification error
-	-
2000	[Direct Pay] Card number is empty
2001	[Direct Pay] Card month/year is empty
2012	[Direct Pay] CCV is wrong format
2013	Expired Card
2200	[Google Pay] Unexcept Error, for detail please see originalError
-	-
13002	This apple pay merchant identifier is disabled

iOS SDK Error Code

status	msg
0	Success
5	Wrong JSON format
-	-
11	App ID not found
12	App name mismatch
13	Unknown app error
16	App key mismatch
-	-
30	Device not support
31	iOS SDK version is too old
32	Android SDK version is too old
33	SDK version is not sent
-	-
41	Wrong card format
-	-
80	Invaild x-api-key or app key
81	Partner not found
84	Partner unauthorized
-	-
122	Card encrypt error
-	-
421	Gateway Timeout
422	Authorization Timeout
-	-
530	Invalid arguments : app_id
531	Missing arguments : app_key
532	Missing arguments : app_name
533	Missing arguments : card_number
534	Missing arguments : card_expiry_date

status	msg
570	Out of range : app_key
571	Out of range : app_name
572	Out of range : card_number
573	Out of range : card_expiry_date
574	Out of range : card_ccv
581	Invalid arguments : devicetype
582	Invalid arguments : sdkversion
583	Out of range : deviceInfo > deviceIdentifier
584	Out of range : deviceInfo > deviceModel
585	Out of range : deviceInfo > deviceOsVersion
586	Out of range : geoloc
589	Invalid arguments : platform_type
591	Missing arguments : pay_token_data
592	Missing arguments : apple_merchant_id
594	Out of range : apple_merchant_id
-	-
600	Missing arguments : pay_token_data > encryptedMessage
605	Missing arguments : pay_token_data > data
606	Missing arguments : pay_token_data > version
607	Missing arguments : pay_token_data > signature
608	Missing arguments : pay_token_data > header > ephemeralPublicKey
609	Missing arguments : pay_token_data > header > publicKeyHash
610	Missing arguments : pay_token_data > header > transactionId
611	Out of range : pay_token_data > data
612	Out of range : pay_token_data > version
613	Out of range : pay_token_data > header > ephemeralPublicKey
614	Out of range : pay_token_data > header > publicKeyHash
615	Out of range : pay_token_data > header > transactionId
616	Missing arguments : pay_token_data > header
617	Out of range : fraud_id
626	Invalid arguments : platform_type

status	msg
638	Missing arguments : merchant_app_launch_uri
639	Out of range : merchant_app_launch_uri
801	Missing arguments : merchant_app_launch_uri or merchant_universal_link
802	Doesn’t support merchant_app_launch_uri or merchant_universal_links concurrently
803	Out of range : merchant_universal_links
-	-
915	System error, please contact TapPay customer service
-	-
1001	Certificate server connection exception (http status code: {http status code})
1002	Certificate local connection exception (exception message: {exception message})
1003	Certificate data error
-	-
88001	SDK Is Not Activate
88003	Lost Parameter
88009	Can not obtain Apple Pay payment data
88010	Please Running on the Device
88012	Invalid cart setting . While isAmountPending is false , it should not have pending item.
88013	isAmountPending and isShowTotalAmount couldn’t be false at the same time.
88014	Total amount couldn’t be displayed if there is only pending items.
88015	isAmountPending must be true if the amount of payment item is pending.
88016	paymentItem amount could’t not be 0.00

Android SDK Error Code

status	msg
-4	Unknown Error
-3	Internet Unavailable
-	-
0	Success
5	Wrong JSON format
-	-
11	App ID not found
12	App name mismatch
13	Unknown app error
16	App key mismatch
-	-
30	Device not support
31	iOS SDK version is too old
32	Android SDK version is too old
33	SDK version is not sent
41	Wrong card format
80	Invaild x-api-key or app key
81	Partner not found
84	Partner unauthorized
-	-
122	Card encrypt error
-	-
421	Gateway Timeout
422	Authorization Timeout
-	-
530	Invalid arguments : app_id
531	Missing arguments : app_key
532	Missing arguments : app_name
533	Missing arguments : card_number

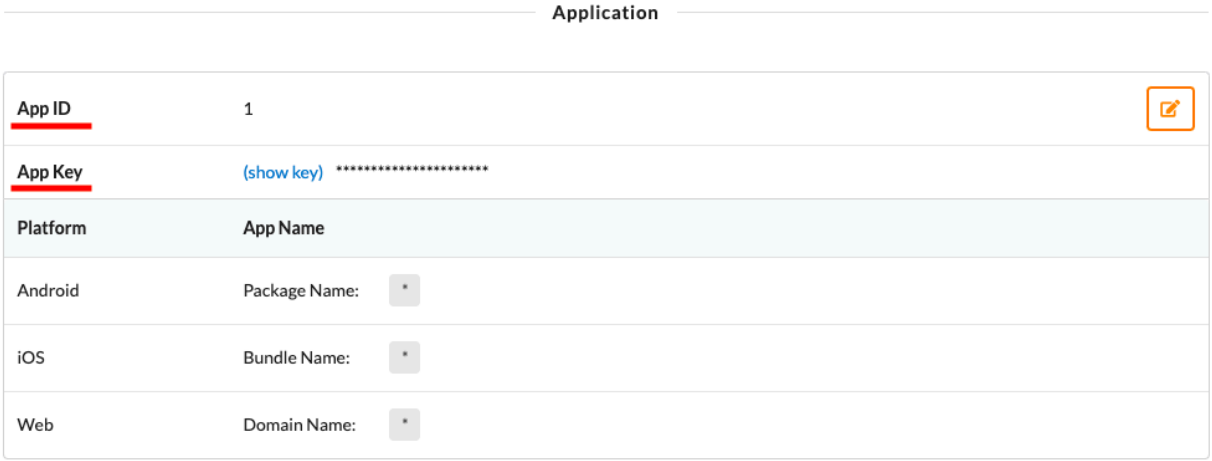
status	msg
534	Missing arguments : card_expiry_date
570	Out of range : app_key
571	Out of range : app_name
572	Out of range : card_number
573	Out of range : card_expiry_date
574	Out of range : card_ccv
581	Invalid arguments : devicetype
582	Invalid arguments : sdkversion
583	Out of range : deviceInfo > deviceIdentifier
584	Out of range : deviceInfo > deviceModel
585	Out of range : deviceInfo > deviceOsVersion
586	Out of range : geoloc
589	Invalid arguments : platform_type
590	Missing arguments : android_merchant_id
591	Missing arguments : pay_token_data
598	Out of range : android_merchant_id
599	Missing arguments : pay_token_data > ephemeralPublicKey
-	-
600	Missing arguments : pay_token_data > encryptedMessage
601	Missing arguments : pay_token_data > tag
602	Out of range : pay_token_data > ephemeralPublicKey
603	Out of range : pay_token_data > encryptedMessage
604	Out of range : pay_token_data > tag
605	Missing arguments : pay_token_data > data
606	Missing arguments : pay_token_data > version
607	Missing arguments : pay_token_data > signature
611	Out of range : pay_token_data > data
612	Out of range : pay_token_data > version
617	Out of range : fraud_id
618	Out of range : pay_token_data > protocolVersion
619	Out of range : pay_token_data > signature

status	msg
626	Invalid arguments : platform_type
634	Missing arguments : pay_token_data > protocolVersion
638	Missing arguments : merchant_app_launch_uri
639	Out of range : merchant_app_launch_uri
658	Missing arguments : pay_token_data > type
659	Out of range : pay_token_data > type
801	Missing arguments : merchant_app_launch_uri or merchant_universal_link
802	Doesn’t support merchant_app_launch_uri or merchant_universal_links concurrently
803	Out of range : merchant_universal_links
-	-
915	System error, please contact TapPay customer service
916	Signature verification is not proceed
917	Signature verification error
-	-
1001	Certificate server connection exception (http status code: {http status code})
1002	Certificate local connection exception (exception message: {exception message})
1003	Certificate data error
-	-
88004	Parameter Format Error
88007	Input Form Not Set
88009	Can not obtain payment data

Reference

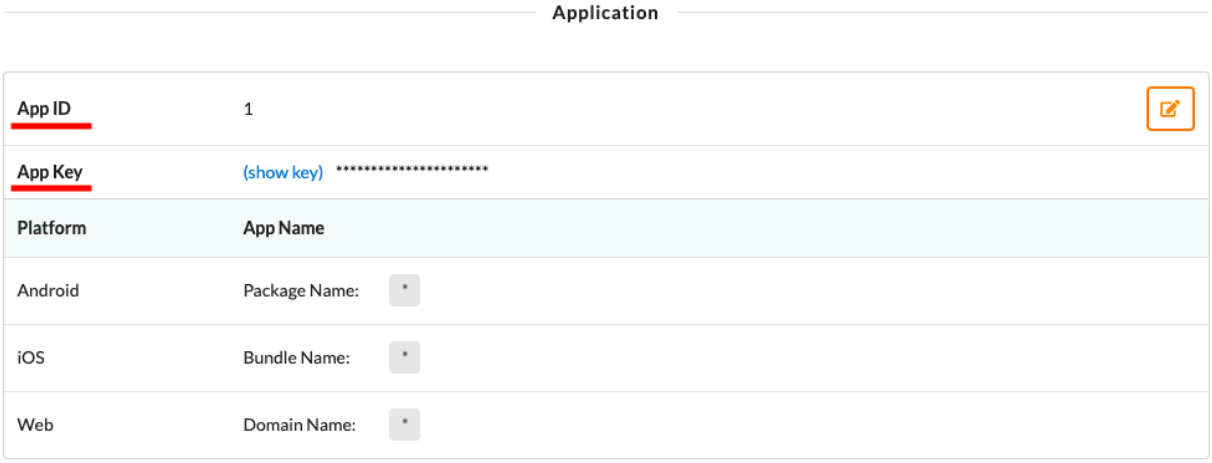
1. appId

An integer identifier for your application or website.
This is used to activate your application or website in order to start using our services.
You can find it via Portal > Developer > Application.



2. appKey

An authenticator key for your application or website.
This is used to activate your application or website in order to start using our services.
You can find it via Portal > Developer > Application.



3. Amount

Total price for a particular transaction.

For refunds, a smaller than original amount is acceptable if partial refund is requested.

TWD：Lower limit of each transaction is NTD 1, upper limit of each transaction is NTD 20,000,000. Please fill in 1 when the transaction is TWD 1.

HKD：Lower limit of each transaction is HKD 0.01, upper limit of each transaction is HKD 252,500. Please fill in 101 when the transaction is HKD 1.01

MYR：Lower limit of each transaction is MYR 0.01, upper limit of each transaction is MYR 135,000. Please fill in 101 when the transaction is MYR 1.01

USD limit of each transaction is USD 0.01, upper limit of each transaction is USD 33,169. Please fill in 101 when the transaction is USD 1.01

4. Currency

The letter abbreviation for currency, following ISO 4217.
For example,

Currency	Code
Taiwan New Dollar	TWD
Hong Kong Dollar	HKD
Malaysia Dollar	MYR
US Dollar	USD
Japanese Yen	JPY

5. Partial Refund

Partial refund could only be executed after captured successfully.
Payment process is as follows:

Transaction happens > Payment is captured based on delay_capture_in_days > Partial refund is available only after captured successfully

e.g. A transaction happened on October 9th.
If the delay_capture_in_days parameter is 0, then the transaction will be captured on the same day. However, Partial refund only could be executed after captured successfully.

e.g. A transaction happened on October 9th.
If the delay_capture_in_days parameter is 3, then the transaction will be captured on October 12th. Therefore, partial refund is available starting from October 13th if the transaction has been successfully captured.

＊If it shows “capture successfully” from TapPay, but couldn’t do partial refund. Please try it again one day after due to the bank issue.

6. Production

The official environment that charges customer's cards for transactions.
You product should be in this environment when it officially launches and goes live.
You can specify the environment for your application during TPDSetup.initInstance().
You can specify the environment for your website during TPDirect.setupSDK().

Things to remember：

1. Register your merchants in the production environment on Portal and use the official bank's acquirer account.
2. Register your IP in the production environment on Portal.
3. Use the production app key instead of the sandbox one.
4. When setting the Frontend environment, change the server type to production.
5. The Backend APIs should use the production domain instead of the sandbox one.(https://sandbox.tappaysdk.com/tpc/ -> https://prod.tappaysdk.com/tpc/)

7. Refund

An API that allows you to refund transaction, fully or partially. Please refer to the backend documentation: Refund API.

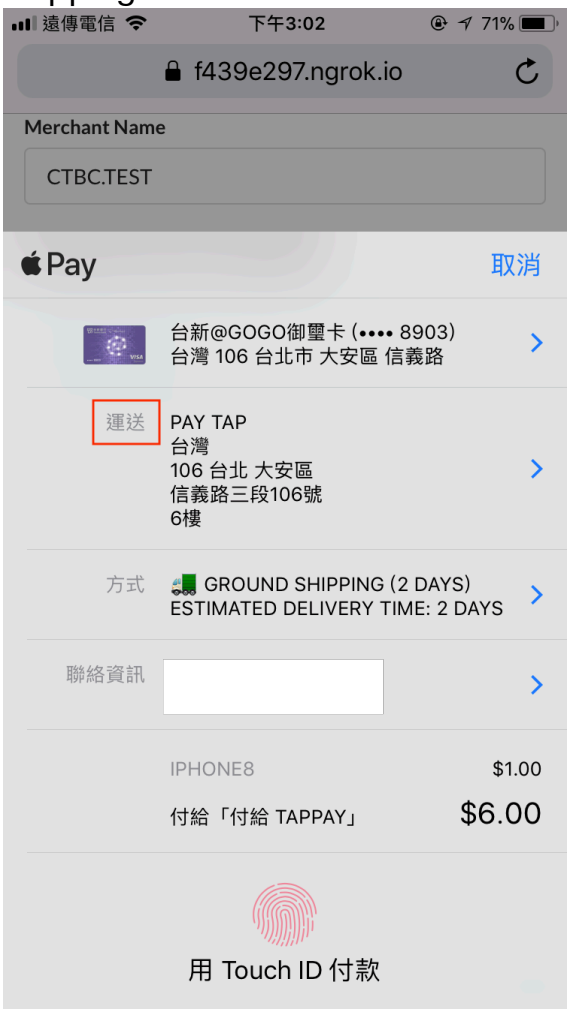
8. Sandbox

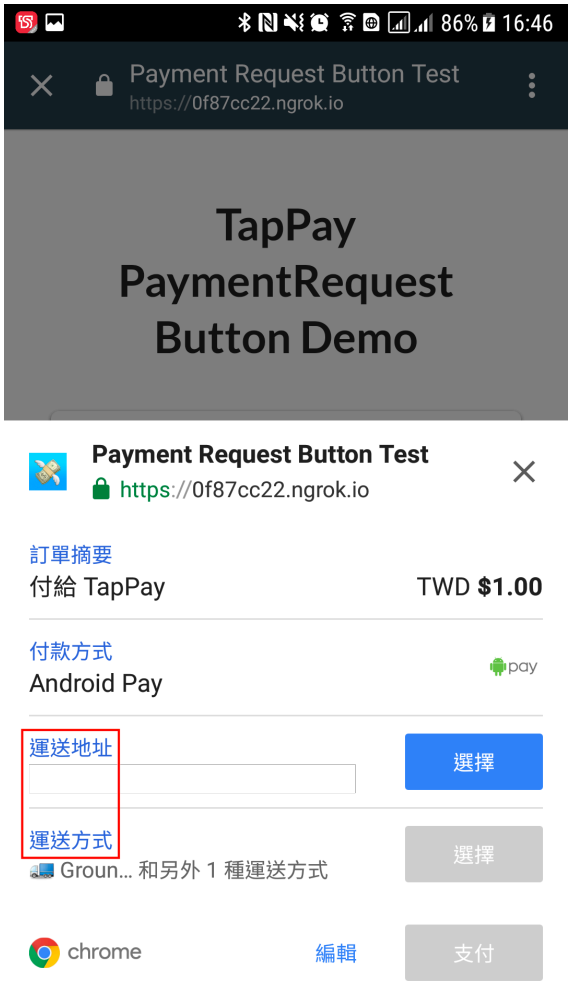
A type of environment that has all the functions of a production environment except no money will be charged for the transactions.
You should first test your application or website in this environment before you move on to the production environment.
You can specify the environment for your application during TPDSetup.initInstance().
You can specify the environment for your website during TPDirect.setupSDK().

8. shippingType

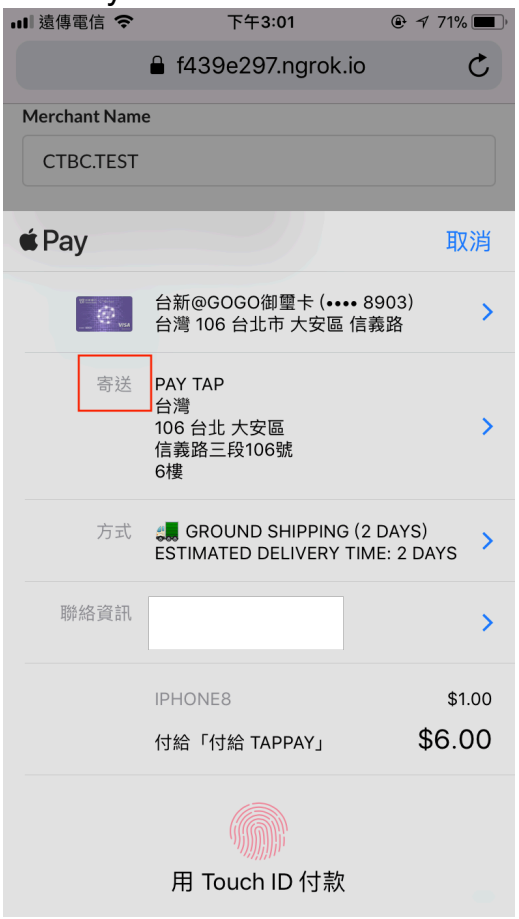
Shipping type , show on Payment Request API

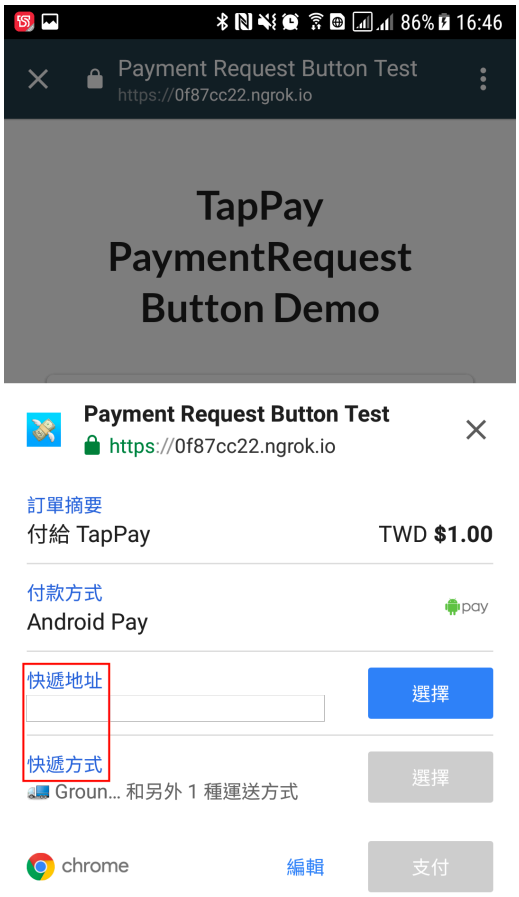
1. shipping


遠傳電信
下午3:02
71%
f439e297.ngrok.io
Merchant Name
CTBC.TEST
Pay
取消
台新@GOGO御璽卡 (•••• 8903)
台灣 106 台北市 大安區 信義路
運送
PAY TAP
台灣
106 台北 大安區
信義路三段106號
6樓
方式
GROUND SHIPPING (2 DAYS)
ESTIMATED DELIVERY TIME: 2 DAYS
聯絡資訊
IPHONE8
$1.00
付給「付給 TAPPAY」
$6.00
用 Touch ID 付款


86% 16:46
Payment Request Button Test
https://0f87cc22.ngrok.io
TapPay
PaymentRequest
Button Demo
Payment Request Button Test
https://0f87cc22.ngrok.io
訂單摘要
付給 TapPay
TWD $1.00
付款方式
Android Pay
pay
運送地址
選擇
運送方式
Groun... 和另外 1 種運送方式
選擇
chrome
編輯
支付

2. delivery





3. pickup