Direct Pay - Frontend

Revision History

Version	Date	Content
1.0.0	2023.05.15	First Draft

Table of Contents

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

1. Setup three tappay field container.
2. Use TPDirect.setupSDK set up your enviroment.
3. Use TPDirect.card.setup set style.
4. TPDirect.card.onUpdated get TapPay Fields status.
5. Use TPDirect.card.getPrime to get prime.
6. Get SDK compatibility information, please refer to Web SDK compatibility

If you have any problem ,please refer to our Example Code

TapPay Fields

Add 3 div for TapPay fields container in your HTML:

```
<style>
    .tpfield {
        height: 40px;
        width: 300px;
        border: 1px solid gray;
        margin: 5px 0;
        padding: 5px;
    }
</style>
<div class="tpfield" id="card-number"></div>
<div class="tpfield" id="card-expiration-date"></div>
<div class="tpfield" id="card-ccv"></div>
```

SetupSDK

```
TPDirect.setupSDK(appID, appKey, serverType)
```

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

```
<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>
```

TPDirect.card.setup

```
TPDirect.card.setup(config)
```

Name	Type	Usage
fields	Object	For the format, please refer to the table shown below
styles	Object	Supported CSS style , please refer to TapPay Fields Styles
isMaskCreditCardNumber	Boolean	Set whether to enable the masking credit card number.
maskCreditCardNumberRange	Object	For the format, please refer to the table shown below

fields format

Name	Type	Usage
number	Object	element: CSS Selector or DOM element placeholder: String
expirationDate	Object	element: CSS Selector or DOM element placeholder: String
ccv	Object	element: CSS Selector or DOM element placeholder: String CCV is optional. If there is no input, it will not be verified. In contrast, it will be verified. If you don’t want to display the ccv field, you don’t have to set up the ccv object.

maskCreditCardNumberRange format

Name	Type	Usage
beginIndex	Int	The start index of masking card number
endIndex	Int	The end index of masking card number

```
TPDirect.card.setup({
    // Display ccv field
    let fields = {
        number: {
            // css selector
            element: '#card-number',
            placeholder: '**** **** **** ****'
        },
        expirationDate: {
            // DOM object
            element: document.getElementById('card-expiration-date'),
            placeholder: 'MM / YY'
        },
        ccv: {
            element: '#card-ccv',
            placeholder: 'ccv'
        }
    }
    // Not display ccv field
    let fields = {
        number: {
            // css selector
            element: '#card-number',
            placeholder: '**** **** **** ****'
        },
        expirationDate: {
            // DOM object
            element: document.getElementById('card-expiration-date'),
            placeholder: 'MM / YY'
        }
    }

    fields: fields,
    styles: {
```

```
            // Style all elements
            'input': {
                'color': 'gray'
            },
            // Styling ccv field
            'input.ccv': {
                // 'font-size': '16px'
            },
            // Styling expiration-date field
            'input.expiration-date': {
                // 'font-size': '16px'
            },
            // Styling card-number field
            'input.card-number': {
                // 'font-size': '16px'
            },
            // style focus state
            ':focus': {
                // 'color': 'black'
            },
            // style valid state
            '.valid': {
                'color': 'green'
            },
            // style invalid state
            '.invalid': {
                'color': 'red'
            },
            // Media queries
            // Note that these apply to the iframe, not the root window.
            '@media screen and (max-width: 400px)': {
                'input': {
                    'color': 'orange'
                }
            }
        },
        /*
         This setting will show that after the card number is entered correctly,
         the first six and last four digits of the credit card number will be *.
        */
        isMaskCreditCardNumber: true,
        maskCreditCardNumberRange: {
            beginIndex: 6,
            endIndex: 11
        }
    })
```

onUpdate

```
TPDirect.card.onUpdate(callback)
```

Implement TPDirect.card.onUpdate , get the current status of the TapPay Fields. The object returned by the callback function will contain the following properties：

Name	Usage
cardType (String)	mastercard, visa, jcb, amex, unknown
canGetPrime (boolean)	true = all fields are correct. You may now call getPrime.
hasError (boolean)	true = error in one of the field.
status.number (int)	For the returned status code, please refer to our reference.
status.expiry (int)	For the returned status code, please refer to our reference.
status.ccv (int)	For the returned status code, please refer to our reference

```
TPDirect.card.onUpdate(function (update) {
    // update.canGetPrime === true
    // --> you can call TPDirect.card.getPrime()
    if (update.canGetPrime) {
        // Enable submit Button to get prime.
        // submitButton.removeAttribute('disabled')
    } else {
        // Disable submit Button to get prime.
        // submitButton.setAttribute('disabled', true)
    }

    // cardTypes = ['mastercard', 'visa', 'jcb', 'amex','unknown']
    if (update.cardType === 'visa') {
        // Handle card type visa.
    }

    // number fields is error
    if (update.status.number === 2) {
        // setNumberFormGroupToError()
    } else if (update.status.number === 0) {
        // setNumberFormGroupToSuccess()
    } else {
        // setNumberFormGroupToNormal()
    }

    if (update.status.expiry === 2) {
        // setNumberFormGroupToError()
    } else if (update.status.expiry === 0) {
        // setNumberFormGroupToSuccess()
    } else {
        // setNumberFormGroupToNormal()
    }

    if (update.status.ccv === 2) {
        // setNumberFormGroupToError()
    } else if (update.status.ccv === 0) {
        // setNumberFormGroupToSuccess()
    } else {
        // setNumberFormGroupToNormal()
    }
})
```

getTappayFieldsStatus

```
TPDirect.card.getTappayFieldsStatus()
```

This method can get TapPay Fields status.
The same to TPDirect.card.onUpdate Callback.

Get Prime

```
TPDirect.card.getPrime(callback)
```

Use TPDirect.card.getPrime to get Prime.

Name	Usage
status	Error code. 0 = Success.
card	Card information For the format, please refer to the table shown below
clientip	IP address of the customer
card_identifier	Card identifier. Each credit card will only have one card identifier.
merchant_reference_info	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

card format

Name	Type (Max)	Usage
prime	String(67)	prime token used in Pay by Prime API
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

Name	Type (Max)	Usage
type	int	Card type -1 = Unknown 1 = VISA 2 = MasterCard 3 = JCB 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.

```
// call TPDirect.card.getPrime when user submit form to get tappay prime
// $('form').on('submit', onSubmit)

function onSubmit(event) {
    event.preventDefault()

    // Get TapPay Fields  status
    const tappayStatus = TPDirect.card.getTappayFieldsStatus()

    // Check can getPrime
    if (tappayStatus.canGetPrime === false) {
        alert('can not get prime')
        return
    }

    // Get prime
    TPDirect.card.getPrime((result) => {
        if (result.status !== 0) {
            alert('get prime error ' + result.msg)
            return
        }
        alert('get prime success, prime: ' + result.card.prime)

        // send prime to your server, to pay with Pay by Prime API .
        // Pay By Prime Docs: https://docs.tappaysdk.com/tutorial/zh/back.html#pay-by-
prime-api
    })
}
```

TapPay Fields focus style

Customize TapPay Fields focus state style, you can implement tappay-field-focus in tappay field container.

```
/* Bootstrap focus style */
.tappay-field-focus {
    border-color: #66afe9;
    outline: 0;
    -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075), 0 0 8px rgba(102, 175, 233, .6);
    box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075), 0 0 8px rgba(102, 175, 233, .6);
}
```

Get CCV Prime

When using Pay by Card Token API, if you want to bring the ccv data which has been hashed to do transaction , you can use TPDirect.ccv.getPrime() to get the ccv_prime.
ps. ccv_prime is a random string. It can help you not to handling sensitive ccv data.

Suggested Workflow

1. Setup CCV Field
2. Setup Card Type
3. onUpdate
4. Get Prime

Setup CCV field

Setup the ccv input attributes with TPDirect.ccv.setup

Name	Type	Usage
elds	JSONObject	Please refer to the fields table
styles	JSONObject	Supported CSS style, please refer to TapPay Fields Styles

fields structure follow bellow table

Name	Type	Usage
ccv	JSONObject	Including the following keys 1. element: CSS Selector or DOMelement 2. placeholder: String

```
TPDirect.ccv.setup({
    fields: {
        ccv: {
            element: '#card-ccv',
            placeholder: 'ccv'
        }
    },
    styles: {
        // Style all elements
        'input': {
            'color': 'gray'
        },
        // Styling ccv field
        'input.ccv': {
            // 'font-size': '16px'
        },
        // style focus state
        ':focus': {
            // 'color': 'black'
        },
        // style valid state
        '.valid': {
            'color': 'green'
        },
        // style invalid state
        '.invalid': {
            'color': 'red'
        },
        // Media queries
        // Note that these apply to the iframe, not the root window.
        '@media screen and (max-width: 400px)': {
            'input': {
                'color': 'orange'
            }
        }
    },
    // 此設定會顯示卡號輸入正確後，會顯示前六後四碼信用卡卡號
    isMaskCreditCardNumber: true,
    maskCreditCardNumberRange: {
        beginIndex: 6,
        endIndex: 11
    }
})
```

Setup Card Type

After set up the ccv input field, you could set up the ccv length by TPDirect.ccv.setupCardType. You could use following variables as parameter. If you don't set up any card type, the ccv input length will be valid from 3 to 4.

```
TPDirect.CardType.VISA
TPDirect.CardType.JCB
TPDirect.CardType.AMEX
TPDirect.CardType.MASTERCARD
TPDirect.CardType.UNKNOWN
```

Setup the ccv input attributes with TPDirect.ccv.setup

onUpdate

You can use TPDirect.ccv.onUpdate() to monitor the current status of ccv input field.

```
TPDirect.ccv.onUpdate((update) => {
    console.log(update)
})

// Example Data
{
  canGetPrime: true
  hasError: false
  status: {
    ccv: 0
  }
}
```

update structure follow below

Name	Type	Usage
canGetPrime	Boolean	Is can Get CCV Prime
hasError	Boolean	Is any error
status.ccv	Number	For the returned status code, please refer to our reference

Get Prime

Finally, use TPDirect.ccv.getPrime to get the CCV prime. And, use Pay by Card Token API to complete transaction with the CCV prime as request parameters. TPDirect.ccv.getPrime is supported by callback or promise.

Via callback to get prime

```
TPDirect.ccv.getPrime((error, response) => {
    if (error) {
        console.log(error)
        return
    }
    console.log(response)
})
```

Via promise to get prime

```
TPDirect.ccv.getPrime().then((response) => {
    console.log(response)
}).catch((error) => {
    console.log(error)
})
```

Response structure follow below table

Name	Type	Usage
status	Number	Error code. 0 = Success.
msg	String	Error message.
ccv_prime	String	CCV Prime token, used in Pay by Card Token API
client_ip	String	IP address of the customer

Get Device Id

If you use RBA service.
Call getDeviceId() get Id of each device.

```
TPDirect.getDeviceId()
```

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

Web SDK Error Code

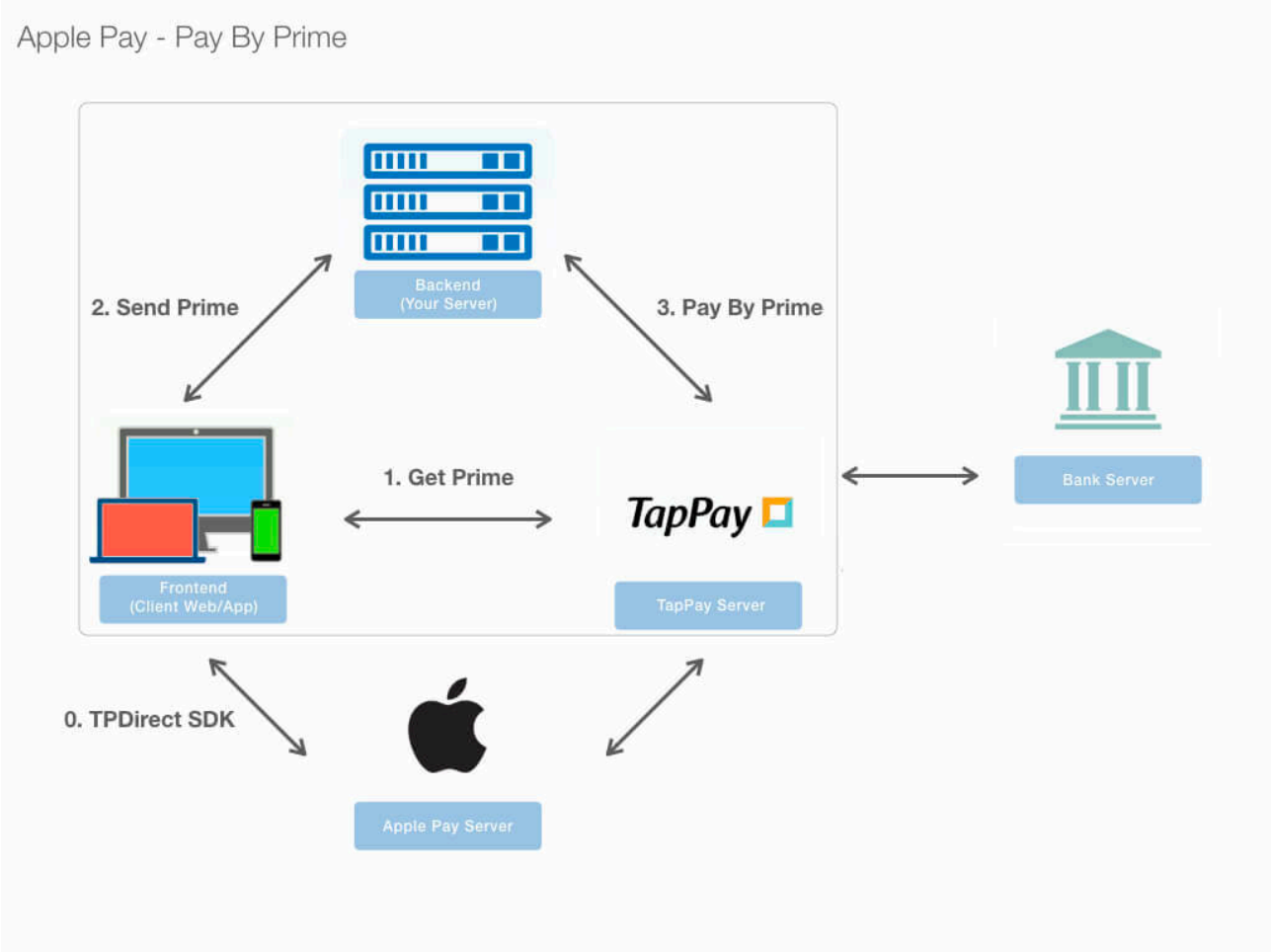
status	msg
-7	[Google Pay] failed to get prime
-6	[Apple Pay] failed to get prime
-5	Unknown Error
-3	failed to get prime
-1	SDK Loading Error
-	-
0	Success
5	Wrong JSON format
-	-
11	App ID not found
12	App name mismatch
13	Unknown app error
16	App key mismatch
-	-
41	Wrong Card Format
-	-
80	Invaild x-api-key or app key
81	Partner not found
84	Partner unauthorized
-	-
122	Card encrypt error
-	-

status	msg
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
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

status	msg
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
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

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



During this step, we will exchange customer's card information with a token called prime that does not contain any sensitive information.
In order to obtain this token in iOS, following the 5 steps below:

1. Download our SDK and integrate TPDirect.framework and TPDirectResource into your project: TPDirect.framework
2. Use TPDSetup to set up your environment.
3. Add UIView in your Main.storyboard
4. Use TPDForm to initialize TPDCard
5. Use the getPrime() function in TPDCard to obtain the prime token.

TPDSetup

iOS-objC

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

iOS-Swift

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

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.

Import TPDirect in your Appdelegate, and use TPDSetup to set up your environment in didFinishLaunchingWithOptions.

iOS-objC

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

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions {
  // Override point for customization after application launch.

  [TPDSetup setWithAppId:AppId withAppKey:@"AppKey"
withServerType:TPDServer_SandBox];

  return YES;
}
```

iOS-Swift

```
import TPDirect
...

func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions:
  [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
  // Override point for customization after application launch.

  TPDSetup.setWithAppId(AppId, withAppKey: AppKey, with: TPDServerType.sandBox)

  return true
}
```

UIView

Add UIView in your Main.storyboard, and and set UIButton to AutoLayout (The suggested dimension for UIView is (270,70))
In your ViewController, set up the @IBOutlet in both UIView and UIButton, and also the @IBAction in UIButton as well.

TPDForm

iOS-objC

```
+ (instancetype)setupWithContainer:(UIView *)view;
+ (instancetype _Nonnull)setup:(TPDForm * _Nonnull)tpdForm;
- (instancetype _Nonnull)onFormUpdated:(void(^_Nonnull)(TPDStatus *_Nonnull
status))callback;
```

iOS-Swift

```
class func setup(withContainer view: UIView) -> Self
class func setup(_ tpdForm: TPDForm) -> Self
func onFormUpdated(_ callback: @escaping (_ status: TPDStatus) -> Void) -> Self
```

Use TPDForm to obtain the customer's card information.
You need to pass your customized view to TPDForm, and use it to initialize TPDCard.
You can use onFormUpdated() to get the current status of TPDForm.
Use the returned TPDStatus to handle different cases, and use isCanGetPrime() to determine whether the entered information is correct or not.

* You must use our test card(4242424242424242, 01/23, 123) in the sandbox environment.

iOS-objC

```
#import "TapPayViewController.h"
#import <TPDirect/TPDirect.h>

@interface TapPayViewController ()

@property (weak, nonatomic) IBOutlet UIView *cardView;
@property (weak, nonatomic) IBOutlet UIButton *payButton;
@property (strong, nonatomic) TPDForm *tpdForm;
@property (strong, nonatomic) TPDCard *tpdCard;
@end

@implementation TapPayViewController

- (void)viewDidLoad {
  [super viewDidLoad];
```

```
    // 1. Setup TPDForm With Your Customized CardView(260, 70)
    self.tpdForm = [TPDForm setupWithContainer:self.cardView];
    self.tpdCard = [TPDCard setup:(self.tpdForm)];

    [self.tpdForm setErrorColor:[self colorWithRGB:@"FF5858" withAlpha:1]];

    // 2. Use callback Get Status
    [self.tpdForm onFormUpdated:^(TPDStatus * _Nonnull status) {

      [self.payButton setEnabled:[status isCanGetPrime]];
      self.payButton.alpha = [status isCanGetPrime] ? 1.0 : 0.25;

    }];

    [self.payButton setEnabled:NO];
    self.payButton.alpha = 0.25;

  }
```

iOS-Swift

```
import UIKit
import TPDirect

class TapPayViewController: UIViewController {
  //MARK: - @IBOutlet

  @IBOutlet weak var cardView: UIView!
  @IBOutlet weak var payButton: UIButton!
  var tpdForm : TPDForm!
  var tpdCard : TPDCard!

  override func viewDidLoad() {
      super.viewDidLoad()

      // 1. Setup TPDForm With Your Customized CardView(260, 70)
      self.tpdForm = TPDForm.setup(withContainer: cardView)
      self.tpdCard = TPDCard.setup(self.tpdForm)

      self.tpdForm.setErrorColor(UIColor.red)

      // 2. Use callback Get Status
      self.tpdForm.onFormUpdated { (status) in
        self.payButton.isEnabled = status.isCanGetPrime()
        self.payButton.alpha = (status.isCanGetPrime()) ? 1.0 : 0.25
      }
      self.payButton.isEnabled = false
      self.payButton.alpha = 0.25
    }
}
```

setIsUsedCcv

iOS-objC

```
- (void)setIsUsedCcv:(BOOL)isUsedCcv;
```

iOS-Swift

```
func setIsUsedCcv(_ isUsedCcv: Bool)
```

Use this method set display ccv field.

Get Prime

If you pay with AE card and use 3DS2.0 verification ,the ccv codes must be brought in, otherwise the transaction will fail

iOS-objC

```
- (instancetype _Nonnull)onSuccessCallback:(void(^ _Nonnull)(NSString *_Nullable prime, TPDCardInfo *_Nullable cardInfo, NSString * cardIdentifier, NSDictionary *_Nonnull merchantReferenceInfo))callback;
- (instancetype _Nonnull)onFailureCallback:(void(^ _Nonnull)(NSInteger status, NSString *_Nonnull message))callback;
- (void)getPrime;
```

iOS-Swift

```
func onSuccessCallback(_ callback: @escaping (_ prime: String?, _ cardInfo: TPDCardInfo?, _ cardIdentifier: String?, _ merchantReferenceInfo: [AnyHashable:Any]) -> Void) -> Self
func onFailureCallback(_ callback: @escaping (_ status: Int, _ message: String) -> Void) -> Self
func getPrime()
```

After you have finished setting the success and failure function for TPDCard, you may call getPrime to obtain the prime token.
Pass the token to your server, and use Pay by Prime API to complete your transaction.

iOS-objC

```
- (IBAction)doneAction:(id)sender {
  // Example Card
  // Number : 4242 4242 4242 4242
  // DueMonth : 01
  // DueYear : 23
  // CCV : 123
```

```
  [[[tpdCard onSuccessCallback:^(NSString * _Nullable prime, TPDCardInfo * _Nullable
cardInfo, NSString * cardIdentifier, NSDictionary * merchantReferenceInfo) {

    NSLog(@"Prime : %@, LastFour : %@", prime, cardInfo.lastFour);
    NSLog(@"Bincode : %@, Issuer : %@, cardType : %lu, funding : %lu ,country : %@ ,
countryCode : %@ , level : %@", cardInfo.bincode, cardInfo.issuer, cardInfo.cardType,
cardInfo.funding , cardInfo.country,cardInfo.countryCode,cardInfo.level);
    [self showResult:[NSString stringWithFormat:@"Prime : %@, LastFour : %@, Bincode :
%@, Issuer : %@, cardType : %lu, funding : %lu ,country : %@ , countryCode : %@ ,
level : %@" , prime, cardInfo.lastFour, cardInfo.bincode, cardInfo.issuer, cardInfo.cardType,
cardInfo.funding, cardInfo.country,cardInfo.countryCode,cardInfo.level]];
  }] onFailureCallback:^(NSInteger status, NSString * _Nonnull message) {
    //
    NSLog(@"Status : %ld, Message : %@", status, message);
    [self showResult:[NSString stringWithFormat:@"Status : %ld, Message : %@", status,
message]];
  }] getPrime];

}
```

iOS-Swift

```
@IBAction func doneAction(_ sender: Any) {
  // Example Card
  // Number : 4242 4242 4242 4242
  // DueMonth : 01
  // DueYear : 23
  // CCV : 123
  tpdCard.onSuccessCallback { (prime, cardInfo, cardIdentifier, merchantReferenceInfo) in

    print("Prime : \(prime!), LastFour : \(cardInfo!.lastFour!)")
    print("Bincode : \(cardInfo!.bincode!), Issuer : \(cardInfo!.issuer!), cardType : \
(cardInfo!.cardType), funding : \(cardInfo!.funding) ,country : \(cardInfo!.country!) ,
countryCode : \(cardInfo!.countryCode!) , level : \(cardInfo!.level!)")

  }.onFailureCallback { (status, message) in

    print("status : \(status) , Message : \(message)")

  }.getPrime()

}
```

Response

Name	Usage
status	Error code. 0 = Success.
message	Error message
prime	prime token used in Pay by Prime API

Name	Usage
cardInfo	Card information For the format, please refer to the table shown below
cardIdentifier	Card identifier. Each credit card will only have one card identifier.
merchantReferenceInfo	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

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 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.

Example

If you have any questions, please refer to our sample project.

Get CCV Prime

When using Pay by Card Token API, if you want to bring the ccv data which has been hashed to do transaction , you can use TPDCcv class getPrime() method to get the ccv_prime.

ps. ccv_prime is a random string. It can help you not to handling sensitive ccv data.

Setup CCV Form

Use this method to set display ccv field.

iOS-objC

```
+ (instancetype)setupCCVWithContainer:(UIView *)view;
```

iOS-Swift

```
TPDCcvForm.setupCCV(withContainer: UIView)
```

Example:

iOS-objC

```
self.tpdCcvForm = [TPDCcvForm setupCCVWithContainer:self.ccvFormView];
```

iOS-Swift

```
tpdCcvForm = TPDCcvForm.setupCCV(withContainer: ccvFormView)
```

onCCVFormUpdated

Use this method to get the current status of ccv.

iOS-objC

```
- (instancetype _Nonnull)onCCVFormUpdated:(void (^)(TPDStatus * _Nullable))callback;
```

iOS-Swift

```
tpdCcvForm.onCCVFormUpdated(callback: (TPDStatus?) -> Void)
```

Example:

iOS-objC

```
[self.tpdCcvForm onCCVFormUpdated:^(TPDStatus * _Nullable status) {
    NSLog(@"isCanGetCCVPrime : %d", [status isCanGetCCVPrime]);
}];
```

iOS-Swift

```
tpdCcvForm.onCCVFormUpdated { (status) in
  print("isCanGetCCVPrime : \(status!.isCanGetCCVPrime())")
}
```

setup TPDCcv

Use this method to initial TPDCcv object.

iOS-objC

```
+ (instancetype _Nonnull)setup:(TPDCcvForm * _Nonnull)ccvForm;
```

iOS-Swift

```
tpdCcv = TPDCcv.setup(ccvForm: TPDCcvForm)
```

Example:

```
TPDCcv.setup(self.tpdCcvForm)
```

Get Prime

If calling getPrime() successfully, you will get prime from onSuccessCallback.
If failed, you will get error code and error message from onFailureCallback.

iOS-objC

```
[[[tpdCcv onSuccessCallback:^(NSString * _Nonnull prime) {
    // Send ccv_prime to pay by card token API
}]onFailureCallback:^(NSInteger status, NSString * _Nonnull message) {
    NSLog(@"status : %d, message : %@", status, message);
}] getPrime];
```

iOS-Swift

```
tpdCcv.onSuccessCallback { (prime) in
    // Send ccv_prime to Pay by card token API
}.onFailureCallback { (status, message) in
    print("status : \(status), message : \(message)")
}.getPrime()
```

Get Device ID

If you use RBA service.
Call getDeviceId() method return ID of each device.

iOS-objC

```
[[TPDSetup shareInstance] getDeviceId];
```

iOS-Swift

```
TPDSetup.shareInstance().getDeviceId()
```

iOS SDK Error Code

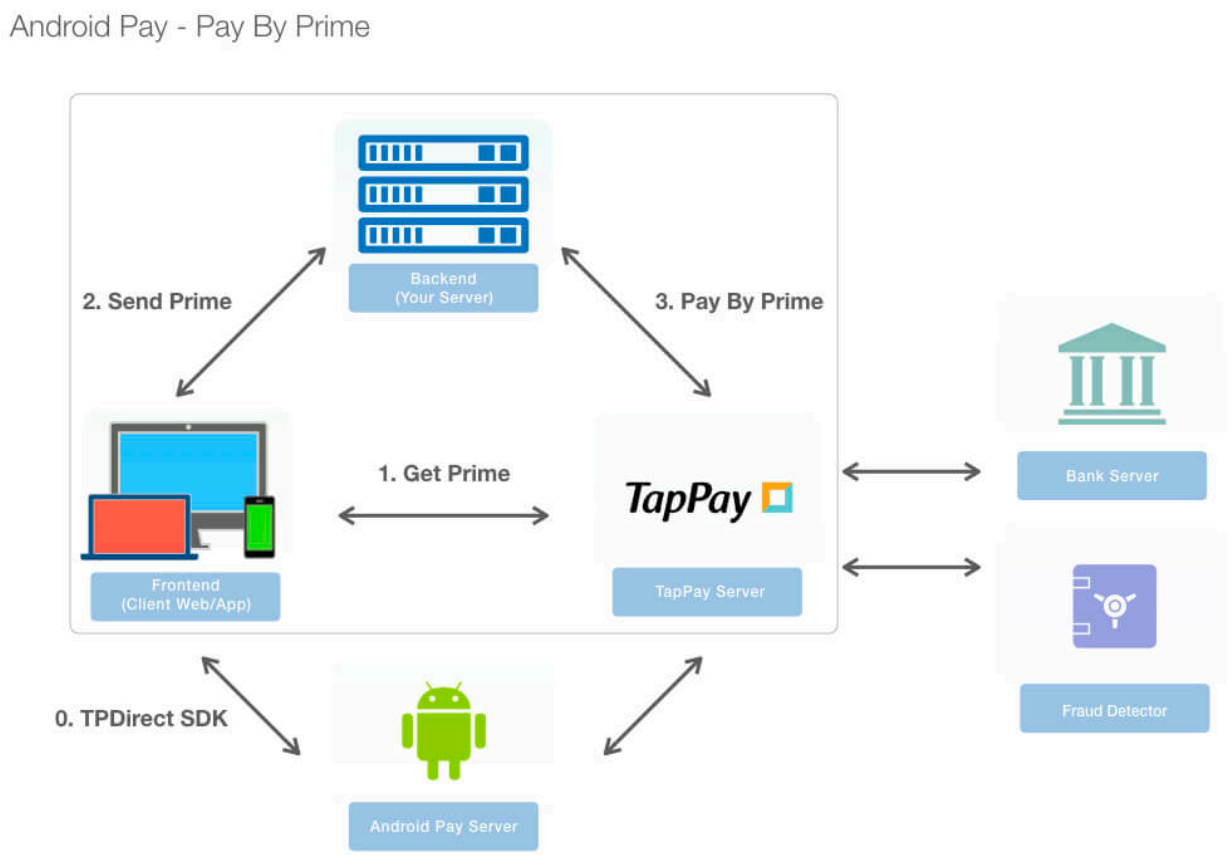
status	msg
0	Success
5	Wrong JSON format
-	-
11	App ID not found
12	App name mismatch
13	Unknown app error
16	App key mismatch
-	-
30	Device not support
31	iOS SDK version is too old
32	Android SDK version is too old
33	SDK version is not sent
-	-
41	Wrong card format
-	-
80	Invaild x-api-key or app 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

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



During this step, we will exchange customer's card information with a token called prime that does not contain any sensitive information.
In order to obtain this token in Android, following the 5 steps below:

1. Download our SDK and integrate it into your project: tpdirect.aar.
2. Use TPDSetup to set up your environment.
3. Add TPDForm to your layout file.
4. Use TPDForm to initialize TPDCard, and set the corresponding callback functions.
5. Use the getPrime() function in TPDCard to obtain the prime token.

TPDSetup

```
static void initInstance(Context context, int appID, String appKey, TPDServerType serverType);
```

Name	Type	Usage
context	Context	Android Context
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.

```
TPDSetup.initInstance(getApplicationContext(), APP_ID, "APP_KEY", TPDServerType.Sandbox);
```

TPDForm

Use TPDForm to obtain the customer’s card information.
The EditText will use the style you set for your theme.
Please add the following lines in your layout.xml:

```
<tech.cherri.tpdirect.API.TPDForm
  android:id=“@+id/tpdForm"
  android:layout_width=“wrap_content"
  android_layout_height=“wrap_content">
</tech.cherri.tpdirect.API.TPDForm>
```

After you use findViewById() to obtain your TPDForm object, you can use setOnFormUpdateListener() to get the current status of TPDForm.
Use the returned TPDStatus to handle different cases, and use isCanGetPrime() to determine whether the entered information is correct or not.

* You must use our test card(4242424242424242, 01/23, 123) in the sandbox environment.

```
tpdForm.setTextErrorColor(Color.RED); // Set color of error messages.
tpdForm.setOnFormUpdateListener(new TPDFormUpdateListener() {
    @Override
    public void onFormUpdated(TPDStatus tpdStatus) {
        errorText.setText("");
        if(tpdStatus.getCardNumberStatus() == TPDStatus.STATUS_ERROR){
            errorText.setText("Invalid Card Number");
        }else if(tpdStatus.getExpirationDateStatus() == TPDStatus.STATUS_ERROR){
            errorText.setText("Invalid Expiration Date");
        }else if(tpdStatus.getCcvStatus() == TPDStatus.STATUS_ERROR){
            errorText.setText("Invalid CCV");
        }
        payButton.setEnabled(tpdStatus.isCanGetPrime());
    }
});
```

setIsUsedCcv

```
tpdForm.setIsUsedCcv(Bool)
```

Use this method set display ccv field.

TPDCard

Initialize a TPDCard object using the correct TPDForm, and set the success and failure callback functions.

```
TPDCard card = TPDCard.setup(TPDForm tpdForm);
  .onSuccessCallback(new TPDCardGetPrimeSuccessCallback() {
    @Override
    public void onSuccess(String prime, TPDCardInfo cardInfo, String cardIdentifier,
TPDMerchantReferenceInfoDto merchantReferenceInfo) {
      Log.i("TPDCard", "Success");
    }
  })
  .onFailureCallback(new TPDGetPrimeFailureCallback()) {
    @Override
    public void onFailure(int status, String reportMsg) {
      Log.i("TPDCard", "Failed");
    }
  });
```

Get Prime

After you have completed the above steps, you may call getPrime() to obtain the prime token.
Pass the token to your server, and use Pay by Prime API to complete your transaction.
If you pay with AE card and use 3DS2.0 verification ,the ccv codes must be brought in, otherwise the transaction will fail

```
card.getPrime();
```

Response

TPDCardGetPrimeSuccessCallback format:

Name	Usage
prime	prime token used in Pay by Prime API
cardInfo	Card information For the format, please refer to the table shown below
cardIdentifier	Card identifier. Each credit card will only have one card identifier.
merchantReferenceInfo	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

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

Name	Type (Max)	Usage
cardType	int	Card type -1 = Unknown 1 = VISA 2 = MasterCard 3 = JCB 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.

TPDGetPrimeFailureCallback format

Name	Usage
status	Error code. 0 = Success.
message	Error message

Example

Please refer to our sample projects, and change the variables in MainActivity.java to your own value:

Name	Usage
EXAMPLE_APP_ID Your	Your app ID.
EXAMPLE_APP_KEY Your	Your app key in the sandbox environment.
TEST_PARTNER_SERVER	The path to your server responsible for handling payments. We will POST the prime token to this path.

Get CCV Prime

When using Pay by Card Token API, if you want to bring the ccv data which has been hashed to do transaction , you can use TPDCcv Class getPrime method to get the ccv_prime.
ps. ccv_prime is a random string. It can help you not to handling sensitive ccv data.

1. Import tpdirect.aar into your project.

2. Use TPDSetup to initialize the SDK and setup environment.

```
TPDSetup.initInstance(getApplicationContext(), "APP_ID", "APP_KEY",
TPDServerType.Sandbox);
```

3. Add TPDCcvForm in your layout.

```
<tech.cherri.tpdirect.api.TPDCcvForm
        android:id="@+id/tpdCcvForm"
        android:layout_width="270dp"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true">
</tech.cherri.tpdirect.api.TPDCcvForm>
```

4. Setup TPDCcv with TPDCcvForm.

```
TPDCcv tpdCcv = TPDCcv.setup(tpdCcvForm)
    .onSuccessCallback(new TPDCcvGetPrimeSuccessCallback() {
        @Override
        public void onSuccess(String ccvPrime) {
            // do something
        }
    }).onFailureCallback(new TPDGetPrimeFailureCallback() {
        @Override
        public void onFailure(int status, String msg) {
            // do something
        }
    });
```

5. Fill ccv in TPDCcvForm and get Prime from TapPay.

```
tpdCcv.getPrime();
```

Get Device Id

If you use RBA service.
Call gerRbaDeviceId() method return ID of each device.

```
String deviceId = TPDSetup.gerInstance(getApplicationContext()).getRbaDeviceId()
```

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

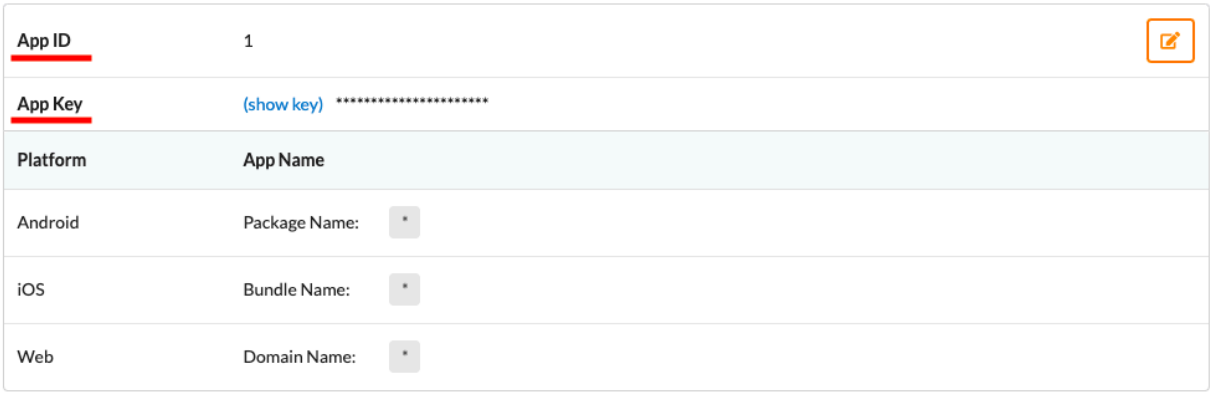
status	msg
626	Invalid arguments : platform_type
634	Missing arguments : pay_token_data > protocolVersion
638	Missing arguments : merchant_app_launch_uri
639	Out of range : merchant_app_launch_uri
658	Missing arguments : pay_token_data > type
659	Out of range : pay_token_data > type
801	Missing arguments : 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.

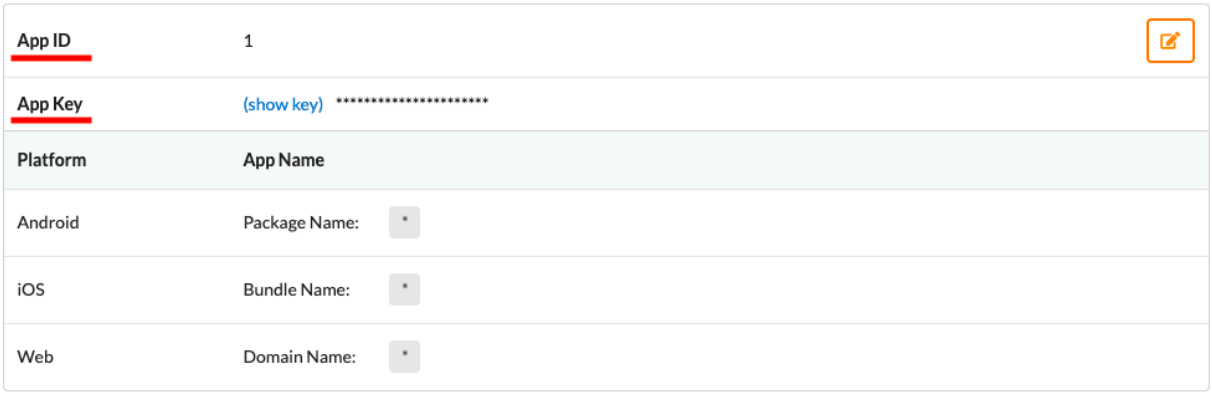
Application



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.

Application



3. onSuccessCallBack()

A callback method for the createToken() method in TPDCard.
It is invoked when the method is successful and returns the prime token and the last four digits of the customer's card number.

4. payByPrime

An API that allows you to pay using the prime token.
You would need to use this API to pay if the customer hasn't registered his or her card before or does not wish to save his or her card.

5. Prime

A one-time token that represents a customer's card.
It is a token returned by TPDirect.card.createToken() or
the onSuccessCallBack() of TPDCard.createToken().
You would need this token to pay in the payByPrime API.
This token will expire after 90 seconds upon creation.
If using Apple Pay Deferred Payments, please keep prime by yourself.
The duration of each prime is set up for 30 days as default. It will be used by calling Pay By Prime API.
You can use this prime for test in sandbox environment.

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. 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. Status Code

Status code returned by our JavaScript SDK that depicts the current situation of the card form the customer is filling out:

code	Usage
0	OK
1	EMPTY
2	ERROR
3	TYPING

9. TapPay Fields Styles

TapPay Fields Supported CSS Style

color , font , font-family , font-size , font-size-adjust , font-stretch , font-style , font-variant , font-variant-alternates , font-variant-caps , font-variant-east-asian , font-variant-ligatures , font-variant-numeric , font-weight , line-height , outline , opacity , text-shadow , transition , -moz-osx-font-smoothing , -moz-transition , -webkit-font-smoothing , -webkit-transition

10. Test Card

All the test cards below only support the test transaction in the test environment. To test Token Pay, please use real card in the test environment. You will not be charged if you use real cards to test in the test environment of Token Pay.
Please contact TapPay Support to get the test information of all the wallet pay.

PS: Use a valid future date for card expired year and expired month

Card Number	CCV	Result
4242 4242 4242 4242	123	0 - Success (type : Visa)
3543 9234 8838 2426	123	0 - Success (type : JCB)
3454 5465 4604 563	1234	0 - Success (type : AMEX)
5451 4178 2523 0575	123	0 - Success (type : MASTERCARD) Complete three-domain-secure transaction without OTP validation
4716 3139 6829 4359	123	0 - Success (type : Visa) The bank_id and issuer_zh_tw will be blank, if you test with this test card
4242 4202 3507 4242	123	915 - Unknown Error, please contact TapPay customer service
4242 4216 0218 4242	123	10003 - Card Error

Card Number	CCV	Result
4242 4222 0418 4242	123	10005 - Bank System Error
4242 4240 1026 4242	123	10006 - Duplicate Transaction
4242 4246 1228 4242	123	10008 - Bank Merchant Account Data Error
4242 4264 1829 4242	123	10009 - Amount Error
4242 4276 2229 4242	123	10013 - Order number duplicate
4242 4288 2639 4242	123	10023 - Bank Error
4242 4210 0008 4242	123	10015 = Redeem Failed

If you would like to test Card Metadata API in the sandbox environment, you have to use following test cards, which will produce the corresponding result.

Card Number	CCV	Result
5480 3543 1215 3777	123	It will return real card face.(Mastercard) token_status="ACTIVE", card art status="SUPPORT", is_real_card_face="TRUE"
4622 9431 2741 6387	123	It will return real card face.(Visa) token_status="ACTIVE", card art status="SUPPORT", is_real_card_face="TRUE"
4242 4217 4599 4242	123	It will return fake card face. (Merchant call card metadata API by an unsupported credit card) token_status="NOT_SUPPORT", card art status="NOT SUPPORT", is_real_card_face="FALSE"
4242 4278 8927 4242	123	It will return fake card face. token_status="REQUEST_PROCESSING", card art status="REQUEST_PROCESSING", is_real_card_face="FALSE"
4242 4230 1382 4242	123	It will return fake card face. token_status="ACTIVE", card art status="NOT_SUPPORT", is_real_card_face="FALSE"
4242 4235 5081 4242	123	It will return fake card face. token_status="ACTIVE", card art status="REQUEST_PROCESSING", is_real_card_face="FALSE"
4242 4246 9569 4242	123	It will return fake card face. token_status="SUSPENDED", card art status="NOT_SUPPORT", is_real_card_face="FALSE"
4242 4253 9494 4242	123	It will return fake card face. token_status="SUSPENDED", card art status="REQUEST_PROCESSING", is_real_card_face="FALSE"

Card Number	CCV	Result
4242 4266 2411 4242	123	It will return fake card face. token_status=“DELETED”, card art status=“NOT_SUPPORT”, is_real_card_face=“FALSE”
4242 4272 1860 4242	123	It will return fake card face. token_status=“DELETED”, card art status=“REQUEST_PROCESSING”, is_real_card_face=“FALSE”
4242 4259 9407 4242	123	It will return fake card face. token_status=“DELETED”, card art status=“SUPPORT”, is_real_card_face=“FALSE”
4242 4264 8626 4242	123	It will return fake card face. (When the transaction was done, but card holder called issuers to delete the card before merchant call Card Metadata API.) token_status=“NOT_SUPPORT”,card art status=“NOT_SUPPORT”, is_real_card_face=“FALSE”
5455 7339 9439 8399	123	It will return real card face. (This is an unsupported credit card in TSP system, but this card’s real card face is stored in TapPay server) token_status=“NOT_SUPPORT”, card art status=“SUPPORT”, is_real_card_face=“TRUE”
4242 4241 8178 4242	123	It will return real card face. token_status=“SUSPENDED”, card art status=“SUPPORT”, is_real_card_face=“TRUE”

If you want to test Card Notify API in sandbox, please use follow test card number：

Card Number	CCV	Result
4242 4241 8178 4242	123	token_status=“SUSPENDED”, card art status=“SUPPORT”, is_real_card_face=“TRUE”

If you want to test identity verification(KYC) in sandbox, please use testing national id and phone number as below.

National id	Phone number
A123456789	0912345678

11. TPDCard
An class for setting up the client's card.

12. TPDirect.card.createToken()
An method for exchanging the customer's card information with a prime token for transaction.
If successful, the prime token returned in the card array will allow you to pay using payByPrime API.

13. TPDSetup
An class for setting up the environment.

14. TPDServerType
An enum used to differentiate sandbox and production environment.
It should be one of the following values:

Value	Usage
TPDServerType.Sandbox	Sandbox environment
TPDServerType.Production	Production environment

15. Web SDK compatibility

Safari

Device	Mac	iOS
Direct Pay	O	O
Apple Pay	O	O
Google Pay (Real Card)	O	O
Google Pay (Token Card)	X	X
PaymentRequestAPI	X	X
LINE Pay	O	O
Samsung Pay	X	X
JKOPAY	O	O

Chrome

Device	Mac	Windows	iOS	Android
Direct Pay	O	O	O	O
Apple Pay	X	X	X	X
Google Pay (Real Card)	O	O	O	O
Google Pay (Token Card)	X	X	X	O
PaymentRequestAPI	O	O	X	O
LINE Pay	O	O	O	O
Samsung Pay	O	O	O	O
JKOPAY	O	O	O	O

Firefox

Device	Mac	Windows	iOS	Android
Direct Pay	O	O	O	O
Apple Pay	X	X	X	X
Google Pay (Real Card)	O	O	X	O
Google Pay (Token Card)	X	X	X	X
PaymentRequestAPI	X	X	X	X
LINE Pay	O	O	X	O
Samsung Pay	O	O	O	O
JKOPAY	O	O	O	O

Opera

Device	Mac	Windows	iOS	Android
Direct Pay	O	O	O	O
Apple Pay	X	X	X	X
Google Pay (Real Card)	O	O	X	O
Google Pay (Token Card)	X	X	X	X
PaymentRequestAPI	O	O	X	X
LINE Pay	O	O	O	O
Samsung Pay	O	O	X	X
JKOPAY	O	O	O	O

Edge

	Windows
Direct Pay	O
Apple Pay	X
Google Pay (Real Card)	O
Google Pay (Token Card)	X
PaymentRequestAPI	X
LINE Pay	O
Samsung Pay	O
JKOPAY	O

IE11

	Windows
Direct Pay	O
Apple Pay	X
Google Pay (Real Card)	X
Google Pay (Token Card)	X
PaymentRequestAPI	X
LINE Pay	O
Samsung Pay	X
JKOPAY	O

Facebook Browser

Device	iOS	Android
Direct Pay	O	O
Apple Pay	X	X
Google Pay (Real Card)	X	X
Google Pay (Token Card)	X	X
PaymentRequestAPI	X	X

LINE Browser

Device	iOS	Android
Direct Pay	O	O
Apple Pay	X	X
Google Pay (Real Card)	X	X
Google Pay (Token Card)	X	X
PaymentRequestAPI	X	X