

TapPay 

Google Pay - Frontend

Table of Contents

Google Pay	4
Overview	4
Environment	4
TPDSetup	5
Prepare Payment Data	5
Check device can use Google Pay	6
TransactionInfo	7
onActivityResult	7
Get Prime	8
Example	9
Google Pay on the Web	10
Overview	10
Prerequisite	10
SetupSDK	10
setupGooglePay	11
setupPaymentRequest	12
setupTransactionPrice	12
setupGooglePayButton	12
Get Prime	13
Example Code	15
Q&A	17
Error	18
Web SDK Error Code	18
iOS SDK Error Code	21
Android SDK Error Code	24
Reference	27
1. appld	27
2. appKey	27
3. Production	27
4. Sandbox	28

Google Pay

If you want to know the latest SDK version and the difference between each version, please refer to Github Release Page: [TapPay Android Github](#).

Overview

Google Pay App are divided into 3 steps :

1. Setup project , use TPDSSetup setup enviroment
2. Prepare payment information , use TPDGooglePay to get paymentData
3. Use paymentData request to TapPay to get prime , pass to your server and complete the transaction using Pay by Prime API (please refer to Backend developer document of Tapay.)

Google Pay Button Design : [Google Pay Brand Guidelines](#)

Environment

Add following line to the build.gradle repositories.

```
allprojects {
    repositories {
        jcenter()
        maven {
            url "https://maven.google.com"
        }
    }
}
```

Add the following lines to the build.gradle in your app module:

```
compile 'com.google.android.gms:play-services-wallet:16.0.0'
compile 'com.android.support:appcompat-v7:24.2.1'
```

Add the following meta-data to your Manifest application:

```
<meta-data android:name="com.google.android.gms.version"android:value="@integer/
google_play_services_version" /> <meta-data
android:name="com.google.android.gms.wallet.api.enabled" android:value="true" />
```

Set up your environment with the following steps:

1. TPDSSetup
2. Prepare Payment Data
3. Check device can use Google Pay
4. TransactionInfo
5. onActivityResult
6. Get Prime

TPDSSetup

Use TPDSSetup to setup enviroment:

```
TPDSSetup.initInstance(getApplicationContext(), APP_ID, "APP_KEY",  
TPDServerType.SANDBOX);
```

Name	Type	Usage
context	Context	Android Context
applID	int	Application identifier
appKey	String	Application authorization key
serverType	TPDServerType	Types of server. Use Sandbox(TPDServerType.Sandbox) for testing, and switch to Production(TPDServerType.Production) after your product launches.

Prepare Payment Data

1. Create TPDMerchant instance and setup your merchant information

```
//Setup allowed card  
TPDCard.CardType[] allowedNetworks = new TPDCard.CardType[]  
{ TPDCard.CardType.Visa, TPDCard.CardType.MasterCard };  
TPDCard.AuthMethod[] allowedAuthMethods = new TPDCard.AuthMethod[]  
{ TPDCard.AuthMethod.PanOnly, TPDCard.AuthMethod.Cryptogram3DS };  
  
TPDMerchant tpdMerchant = new TPDMerchant();  
tpdMerchant.setSupportedNetworks(allowedNetworks);  
tpdMerchant.setSupportedAuthMethods(allowedAuthMethods);  
tpdMerchant.setMerchantName("Your Store Name")
```

Name	Type
setSupportedNetworks	Set supported payment networks.
setSupportedAuthMethods	Set allowed credit card authorization method. "PanOnly": Basic card transaction. Accept credit cards on Google Play stored in the cloud. "Cryptogram3DS": Token Card transaction, only accept token cards on mobile phones (e.g : Google Pay)
setMerchantName	Set up merchant name

2. Create TPDCConsumer instance and setup require information

```
TPDCConsumer tpdConsumer = new TPDCConsumer();
tpdConsumer.setPhoneNumberRequired(true);
tpdConsumer.setShippingAddressRequired(true);
tpdConsumer.setEmailRequired(true);
```

Name	Type
setPhoneNumberRequired	Phone number required or not.
setShippingAddressRequired	Address required or not.
setEmailRequired	Email required or not.

Check device can use Google Pay

Check user device can use Google Pay

```
/**
 * Create TPDGooglePay Instance and pass previous step created TPDMerchant and
 * TPDCConsumer.
 * Check device can use Google Pay
 * Get result from TPDGooglePayListener.onReadyToPayChecked
 */
tpdGooglePay = new TPDGooglePay(this, tpdMerchant , tpdConsumer);
tpdGooglePay.isGooglePayAvailable(this);
```

TransactionInfo

Create TransactionInfo instance and use tpdGooglePay.requestPayment method to get PaymentData.

```
tpdGooglePay.requestPayment(TransactionInfo.newBuilder()
    .setTotalPriceStatus(WalletConstants.TOTAL_PRICE_STATUS_FINAL)
    .setTotalPrice("3")
    .setCurrencyCode("TWD")
    .build(), LOAD_PAYMENT_DATA_REQUEST_CODE);
```

Name	Type
setTotalPriceStatus	Set transaction status TOTAL_PRICE_STATUS_FINAL : The total price is the final total price of the transaction.
setTotalPrice	Set transaction price
setCurrencyCode	Set transaction currency

onActivityResult

Obtain paymentData from onActivityResult, this object contain user chose card and shipping information

```
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    switch (requestCode) {
        case LOAD_PAYMENT_DATA_REQUEST_CODE:
            switch (resultCode) {
                case Activity.RESULT_OK:
                    paymentData = PaymentData.getFromIntent(data);
                    break;
                case Activity.RESULT_CANCELED:
                    showMessage("Canceled by User");
                    break;
            }
    }
}
```

Get Prime

Request to TapPay to get Prime:

```
tpdGooglePay.getPrime(paymentData, TPDGooglePayGetPrimeSuccessCallback,  
TPDGetPrimeFailureCallback);  
TPDGooglePayGetPrimeSuccessCallback parameters
```

TPDGooglePayGetPrimeSuccessCallback parameters:

Name	Usage
prime	prime token used in Pay by Prime API
cardInfo	Card information For the format, please refer to the table shown below
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.

TPDGetPrimeFailureCallback parameters format

Name	Type	Usage
status	Int	Error code
reportMsg	String	Error Message

Example

Please refer to our Google Pay Example , and change the variables in MainActivity.java to your own value:

Name	Usage
EXAMPLE_APP_ID	Your app ID.
EXAMPLE_APP_KEY	Your app key in the sandbox environment.

Google 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

Overview

Google Pay Mobile Web has 2 steps:

1. Use SDK getPrime API to retrieve the Prime.
2. Pass the Prime back to your server, and call Pay by Prime API to complete the transaction.

Google Pay Button Design : Google Pay Brand Guidelines

Prerequisite

To use Google Pay on the Web, check the following information: <https://developers.google.com/payments/mobile-web-setup>

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/sdk/tpdirect/v5.13.1>

Please refer to the following usage examples.

```
// include google pay.js
<script src="https://pay.google.com/gp/p/js/pay.js"></script>
<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

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

setupGooglePay

```
TPDirect.googlePay.setupGooglePay
```

Setup Google Pay Merchant

Fill in you request from Google's Google Merchant ID

```
var googlePaySetting = {
  googleMerchantId: "Come from google portal", // Optional in sandbox, Required in
  production
  allowedCardAuthMethods: ["PAN_ONLY", "CRYPTOGRAM_3DS"],
  merchantName: "TapPay Test!",
  emailRequired: true, // optional
  shippingAddressRequired: true, // optional
  billingAddressRequired: true, // optional
  billingAddressFormat: "MIN", // FULL, MIN

  allowPrepaidCards: true,
  allowedCountryCodes: ['TW'],

  phoneNumberRequired: true // optional
}

TPDirect.googlePay.setupGooglePay(googlePaySetting)
```

Name	Type	Usage
googleMerchantId	String	Setup Google Merchant Id
allowedCardAuthMethods	Array	Setup card authentication methods "PAN_ONLY": Basic card transaction "CRYPTOGRAM_3DS": Token card transaction (e.g : Google Pay)
merchantName	String	Setup Merchant Name
emailRequired	Boolean	Request payer email

Name	Type	Usage
shippingAddressRequired	Boolean	Request shipping address
billingAddressRequired	Boolean	Request billing address
billingAddressFormat	String	Setup Address format, Please refer to Google Pay Address format
allowPrepaidCards	Boolean	Allow Prepaid card
allowedCountryCodes	Array	Allowed country, follow ISO 3166-1 alpha-2 format
phoneNumberRequired	Boolean	Request payer number

setupPaymentRequest

```
var paymentRequest = {
  allowedNetworks: ["AMEX", "JCB", "MASTERCARD", "VISA"],
  price: "123", // optional
  currency: "TWD", // optional
}

TPDirect.googlePay.setupPaymentRequest(paymentRequest, function(err, result){
  if (result.canUseGooglePay) {
    canUseGooglePay = true
  }
})
```

Name	Type	Usage
allowedNetworks	Array	Setup allow card type
price	String	Transaction amount
currency	String	Currency

setupTransactionPrice

This function is optional, if you setup amount in setupPaymentRequest, you don't need to implement this function, if not, please setup amount before you get prime.

```
TPDirect.googlePay.setupTransactionPrice({
  price : "123",
  currency: "TWD"
})
```

setupGooglePayButton

Setup Google Pay Button

This is optional setting, if you want to setup Google Pay Button by yourself, you don't need to implement this function.

```
TPDirect.googlePay.setupGooglePayButton({
  el: "#container",
  color: "black",
  type: "long",
  getPrimeCallback: function(err, prime){}
})
```

Name	Type	Usage
el	CSS Selector or DOM element	Setup Google Pay Button on element.
color	String	Setup Google Pay button color, black , white
type	String	Setup Google Pay type, long : "Buy with Google Pay", short : "Buy with text"
getPrimeCallback	function	Get Prime via Callback.

More information about Google Pay Button setting, please refer to [Google Button Options](#).

Get Prime

If you don't implement setupGooglePayButton, Please use getPrime to trigger Google Pay.

If use setupGooglePayButton, you don't have to implement this function.

```
TPDirect.googlePay.getPrime(function(err, prime, result) {
  // get the prime, and send it to Server
})
```

err object format

Name	Type
status	number
msg	String
originalError	string or object

result object format

Name	Type
client_ip	String
card_info	Card information For the format, please refer to the table shown below
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_info 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

merchant_reference_info format

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

Example Code

Example1

Custom Google Pay Button

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <script src="https://pay.google.com/gp/p/js/pay.js"></script>
  <script src="https://js.tappaysdk.com/tpdirect/v4"></script>
  <title>Document</title>
</head>

<body>
  <button id="test" onClick="getPrime()">Google Pay</button>
  <script>
    var canUseGooglePay = false
    var googlePaySetting = {
      googleMerchantId: "Come from google portal",
      allowedCardAuthMethods: ["PAN_ONLY", "CRYPTOGRAM_3DS"],
      merchantName: "TapPay Test!",
      emailRequired: true, // optional
      shippingAddressRequired: true, // optional,
      billingAddressRequired: true, // optional
      billingAddressFormat: "MIN", // FULL, MIN

      allowPrepaidCards: true,
      allowedCountryCodes: ['TW'],

      phoneNumberRequired: true // optional
    }
    TPDirect.googlePay.setupGooglePay(googlePaySetting)

    var paymentRequest = {
      allowedNetworks: ["AMEX", "JCB", "MASTERCARD", "VISA"],
      price: "123", // optional
      currency: "TWD", // optional
    }
    TPDirect.googlePay.setupPaymentRequest(paymentRequest, function (err, result) {
      if (result.canUseGooglePay) {
        canUseGooglePay = true
      }
    })
  </script>
</body>
</html>
```

```

function getPrime() {
  if (!canUseGooglePay) {
    return console.error("You can't use Google Pay.")
  }
  TPDirect.googlePay.getPrime(function (err, prime) {
    // get the prime, and send it to Server
  })
}
</script>
</body>

</html>

```

Example2

Use setupGooglePayButton to setup Google Pay Button

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <script src="https://pay.google.com/gp/p/js/pay.js"></script>
  <script src="https://js.tappaysdk.com/tpdirect/v4"></script>
  <title>Document</title>
</head>

<body>
  <div id="container"></div>
  <script>
    var googlePaySetting = {
      googleMerchantId: "Come from google portal",
      allowedCardAuthMethods: ["PAN_ONLY", "CRYPTOGRAM_3DS"],
      merchantName: "TapPay Test!",
      emailRequired: true, // optional
      shippingAddressRequired: true, // optional,
      billingAddressRequired: true, // optional
      billingAddressFormat: "MIN", // FULL, MIN

      allowPrepaidCards: true,
      allowedCountryCodes: ['TW'],

      phoneNumberRequired: true // optional
    }
    TPDirect.googlePay.setupGooglePay(googlePaySetting)

    var paymentRequest = {
      allowedNetworks: ["AMEX", "JCB", "MASTERCARD", "VISA"],
      price: "123", // optional
      currency: "TWD", // optional
    }
    TPDirect.googlePay.setupPaymentRequest(paymentRequest, function (err, result) {
      if (result.canUseGooglePay) {
        TPDirect.googlePay.setupGooglePayButton({

```



```
el: "#container",
color: "black",
type: "long",
getPrimeCallback: function(err, prime) {
    // get the prime, send it to server
}
})
}
})
</script>
</body>

</html>
```

Q&A

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

1. js.tappaysdk.com
2. fraud.tappaysdk.com
3. pay.google.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	Invalid x-api-key or app key
81	Partner not found
84	Partner unauthorized
-	-
122	Card encrypt error
-	-
401	Request Cancel, please see the detail of msg
402	Can not obtain payment data, please see the detail 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

status	msg
-	-
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
612	Out of range : pay_token_data > version

status	msg
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	Invalid 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	Invalid 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 > deviceId
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

App ID	1	
App Key	(show key) *****	
Platform	App Name	
Android	Package Name: *	
iOS	Bundle Name: *	
Web	Domain Name: *	

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

App ID	1	
App Key	(show key) *****	
Platform	App Name	
Android	Package Name: *	
iOS	Bundle Name: *	
Web	Domain Name: *	

3. Production

The official environment that charges customer's cards for transactions. Your 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/>)

4. 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()`.

5. TPDServerType

An enum used to differentiate sandbox and production environment.

It should be one of the following values:

Value	Usage
<code>TPDServerType.Sandbox</code>	Sandbox environment
<code>TPDServerType.Production</code>	Production environment