Open Platform Application Docking Guide

v 0.4

Revision History

Date	Version	Modified by	Description
2019-06-09	0.1	Marcus LeBlanc	Initial draft of English version
2019-06-10	0.2	Marcus LeBlanc	Added translated diagrams
2019-07-01	0.3	Marcus LeBlanc	Modified parameter order Added new UAT url Modified some wording
2019-07-31	0.4	Marcus LeBlanc	Fixed formatting Added additional samples Added more clarification

Table of Contents

Access preparation

Third party data list:

- App Name: PundixH5Demo
- Application (H5) address: https://open-auth-uat-2.pundix.com/platform/dist/index.html
- Payment callback: https://open-auth-uat-2.pundix.com/platform/pay
- Trading Currency (test platform): NPXS
- Application icon: 480 * 480, png
- Menu icon: 240 * 240, png

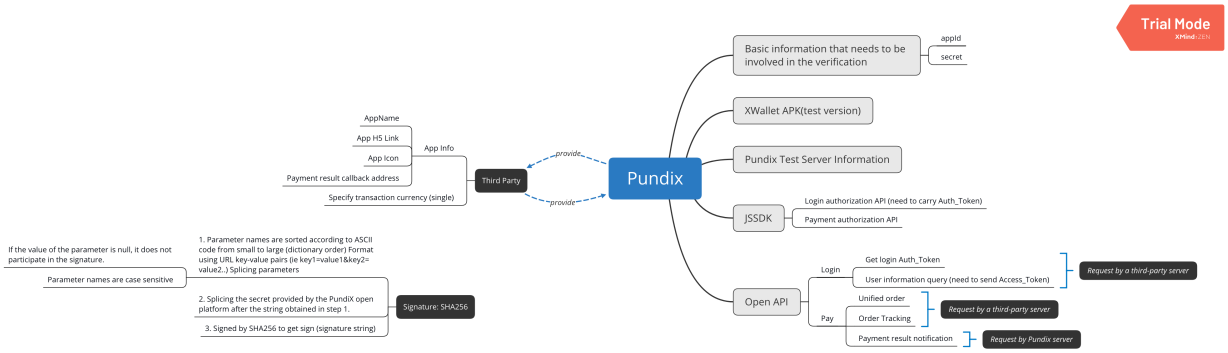
> Take the access application of the Pundi X application as an example. Please specify NPXS as the test environment transaction currency. In the production environment, you need to apply for your Token on the platform before you can specify the custom currency.

Pundix data sheet:

- Basic information (appId, secret): courtesy of Pundi X
- XWallet installation package (beta): courtesy of Pundi X
- Pundi X server information: https://open-auth-uat-2.pundix.com
- JSSDK link: https://open-auth-uat-2.pundix.com/platform/dist/pundix.js H5
- application demo link: https://open-auth-uat-2.pundix.com/platform/dist/index.html
- XWallet test Account: Provided by Pundi X

> After the commissioning is completed, the above information needs to be replaced with the corresponding production environment information. The production environment information needs to be completed on the Pundi X Open Platform. Obtained after the application is applied for

as the picture shows:


Trial Mode
XMind:ZEN
Pundix
Basic information that needs to be involved in the verification
appId
secret
XWallet APK(test version)
Pundix Test Server Information
JSSDK
Login authorization API (need to carry Auth_Token)
Payment authorization API
Open API
Login
Get login Auth_Token
User information query (need to send Access_Token)
Request by a third-party server
Pay
Unified order
Order Tracking
Request by a third-party server
Payment result notification
Request by Pundix server
provide
provide
Third Party
App Info
AppName
App H5 Link
App Icon
Payment result callback address
Specify transaction currency (single)
Signature: SHA256
1. Parameter names are sorted according to ASCII code from small to large (dictionary order) Format using URL key-value pairs (ie key1=value1&key2=value2..) Splicing parameters
If the value of the parameter is null, it does not participate in the signature.
Parameter names are case sensitive
2. Splicing the secret provided by the PundiX open platform after the string obtained in step 1.
3. Signed by SHA256 to get sign (signature string)

Signature algorithm

In the docking process, the system uses SHA256 to perform the line-by-line signing of the transmission parameters to ensure security during the transmission. The transmission parameters except the sign need to participate in the signature.

```
{
        appId: "20190513190506358790529",
        nonceStr: "2e92ab9c25fdf3f035233f7a5fe3e3f6",
        timestamp: 1557803736,
        sign: ""
}
```

JSON request example

Proceed as follows:

1. The parameters to be transmitted are **sorted according to alphabetical order from small to large (dictionary order)**, using the URL key-value pair format splicing parameters (i.e.: key1=value1&key2=value...); Lowercase.

   As shown below:

```
appId=20190513190506358790529&nonceStr=2e92ab9c25fdf3f035233f7a5fe3e3f6&timestamp
=155780373&yourField1=yourValue1&yourField2=yourField2Value2
```

2. After the string obtained in step 1. **the secret key issued by the platform is added**. For this example, the key is assumed to be “**IPq0Cx4sJXqUwdrst1VVYg==“**.

As follows:

```
appId=20190513190506358790529&nonceStr=2e92ab9c25fdf3f035233f7a5fe3e3f6&timestamp=15
5780373&yourField1=yourValue1&yourField2=yourField2Value2IPq0Cx4sJXqUwdrst1VVYg==
```

3. From step 2. splicing to get the tempSign string, the string is by encrypted using SHA256. signature on tempSign, as shown below:

```
sign = SHA256Enc.encrypt(tempSign)
// sign字符串
6d606bc7082dd9319908b7e798d1a7a7580075bf2023bd9987ee2109f05a335c
```

> After encoding by SHA256, it needs to be converted to **Hex**. If the parameter is **null**, it will not participate in the signature.

Open Platform API

The open API root path is: /apiPlatformAuth/

Take the auth token for the interface port as an example: http://test-0514api.pundix.com/apiPlatformAuth/api/v1/auth/token

The API adopts the RESTful format, **the Open Platform API is unified, and the JSON format is used for interaction.**

Universal response format

To receive the response, format your JSON request as follows:

```
{
  "code": 0,
  "data": {
    "appId": "string",
    "authToken": "string"
  },
  "msg": "string"
}
```

Parameter column list

Field name	Variable Name	Type	Sample Value	Description
Response Code	code	Number	200	The normal response is 200; when an exception occurs, the code is an exception code, and detailed exception information can be viewed in the exception code column list.
Data Pack	data	JSON	{"appId":"","authToken":""}	The normal response is a packet; when an exception occurs, it is null
Message	msg	String	success	The normal response is "success"; when an exception occurs, this is the error message.

Get Auth Token

Application scenario

The auth token obtains the interface port, and uses the auth token required to obtain the login authorization to exchange the access token, to gain access to user data.

Interface Address

/api/v1/auth/token

Requester method

GET

Parameter list

Field name	Variable Name	Req'd	Type	Sample Value	Description
Application Unique Identifier	appId	Yes	String	20190513100506846790529	Application id assigned by the platform
Timestamp	timestamp	Yes	Number	1557812279	Timestamp, in seconds
Random String	nonceStr	Yes	String	3d31858ed6dbd7ee6df1db11036af6d7	The normal response is "success"; when an exception occurs, this is the error message.
Sign	Sign	Yes	String	52b5de0a2eb3899683f20401b0a83b0dd918...	SHA256 signature string

Response

The response is in JSON format

```
{
  "code": 0,
  "data": {
    "appId": "string",
    "authToken": "string"
  },
  "msg": "string"
}
```

Field Name	Variable Name	Type	Sample Value	Description
Application Unique Identifier	appId	String	20190513100506846790529	Application id assigned by the platform
Authorization Token	authToken	String	i/W0SddsNEONsSy+fd29+w==	Used to exchange accessToken, authToken can only be used once, and it can be used immediately after expiration. The expiration time is 10 minutes.

Unify the single API

Application Scenario

Used to make prepaid orders on the open platform to complete the payment operation.

Interface Address

/api/v1/order

Request Method

POST

Parameter List

Field name	Variable Name	Req’d	Type	Sample Value	Description
Currency Price	amount	Yes	Number	0.05	Commodity price, subject to the user-specified currency unit
Application Unique Identifier	appId	Yes	String	20190513100506846790529	Application id assigned by the platform
Application Order Number	appOrderNo	Yes	String	20320513100506846790529	The payment order saved by the third party application service number
Packet	attach	Yes	String	{tag,'myTag'}	The packet information will be in the payment notice Return
Product Name	body	Yes	String	Gifts (roses, fireworks, etc.)	Product name from 3rd party
Currency Unit	currencyUnit	Yes	String	VOF	Use the payment currency unit specified when the user places an order
Product Detail	detail	Yes	String	Virtual gift, rose	Product Detail information
Random String	nonceStr	Yes	String	3d31858ed6dbd7ee6df1db11036af6d7	The normal response is "success"; when an exception occurs, this is the error message.
Quantity	num	Yes	Number	1	Quantity of goods purchased
Order Expiration Time	orderExpireTime	Yes	String	1557812279000	How long before the order expires, timestamp, in milliseconds
Timestamp	timestamp	Yes	Number	1557812279	Timestamp, in seconds
Sign	sign	Yes	String	52b5de0a2eb3899683f20401b0a83b0dd918...	SHA256 signature string

Response

The response is in JSON format.

```
{
  "code": 0,
  "data": {
    "prepayOrderNo": "string"
  },
  "msg": "string"
}
```

Field Name	Variable	Type	Sample Value	Description
Prepay Order Number	prepayOrderNo	String	20320513100506846790529	The prepay order number needs to be stored in the and used for payment authorization request and future order inquiry.

> When the third party places an order, the user needs to specify the payment currency. When the user pays in XWallet, the payment can be made only by the currency specified when the user places the order.

Order Query API

Application Scenario

Used to query the booking order details on the platform.

Interface Address

/api/v1/order

Request Method

GET

Parameter List

Field name	Variable Name	Req'd	Type	Sample Value	Description
Application Unique Identifier	appId	Yes	String	20190513100506846790529	Application id assigned by the platform

```
{
  "code": 0,
  "data": {
    "amount": 0,
    "appId": "string",
    "appOrderNo": "string",
    "currencyUnit": "string",
    "orderStatus": 0,
    "orderTime": 0,
    "prepayOrderNo": "string",
    "productDetail": "string",
    "productName": "string"
  },
  "msg": "string"
}
```

Field name	Variable Name	Req'd	Type	Sample Value	Description
Random String	nonceStr	Yes	String	3d31858ed6dbd7ee6df1db11036af6d7	The normal response is "success"; when an exception occurs, this is the error message.
Prepay Order Number	prepayOrderNo	Yes	String	20320513100506846790529	How long before the order expires, timestamp, in milliseconds
Timestamp	timestamp	Yes	Number	1557812279	Timestamp, in seconds
Sign	sign	Yes	String	52b5de0a2eb3899683f20401b0a83b0dd918...	SHA256 signature string

Response

The response is in JSON format.

Parameter List

Field name	Variable Name	Req'd	Type	Sample Value	Description
Currency Price	amount	Yes	Number	0.05	Commodity price, subject to the user-specified currency unit
Application Unique Identifier	appId	Yes	String	20190513100506846790529	Application id assigned by the platform
Application Order Number	appOrderNo	Yes	String	20320513100506846790529	The payment order saved by the third party application service number
Currency Unit	currencyUnit	Yes	String	VOF	Use the payment currency unit specified when the user places an order
Quantity	num	Yes	Number	1	Quantity of goods purchased
Order Status	orderStatus	Yes	Number	0	Platform order status, as shown in the order status
Order Time	orderTime	Yes	Number	1557812279000	Platform reservation order time
Product Detail	productDetail	Yes	String	Virtual gift, rose	Product Detail information
Product Name	productName	Yes	String	Gifts (roses, fireworks, etc.)	Product name from 3rd party
Sign	sign	Yes	String	52b5de0a2eb3899683f20401b0a83b0dd918...	SHA256 signature string

Order Status (orderStatus)

Field Name	Variable Name	Value	Description
Order Status	orderStatus	0	Waiting for payment, waiting for payment by user
Order Status	orderStatus	1	Completed, completed with user payment
Order Status	orderStatus	2	Cancel, cancel the order with the user or the timeout of the order has not been paid

Query user information

Application Scenario

After the user login authorization is completed, the third-party application obtains the accessToken and accesses the user information by utilizing the accessToken.

Interface Address

/api/v1/user

Request Method

GET

Parameter List

Field Name	Variable Name	Req'd	Type	Value	Description
Access Token	accessToken	Yes	String	i/W0SddsNEONsSy+fd29+w==	accessToken expires for 30 days
Application Unique Identifier	appId	Yes	String	20190513100506846790529	Application id assigned by the platform
Random String	nonceStr	Yes	String	3d31858ed6dbd7ee6df1db11036af6d7	Random string, not much larger than 32 bits
Timestamp	timestamp	Yes	Number	1557812279	Timestamp, in seconds
Signature	sign	Yes	String	52b5de0a2eb3899683f20401b0a83b0dd918...	SHA256 signature string

Response

The response is in JSON format

Field Name	Variable Name	Req'd	Type	Value	Description
Application Unique Identifier	appId	Yes	String	20190513100506846790529	Application id assigned by the platform

Field Name	Variable Name	Req’d	Type	Value	Description
User Unique identifier	openId	Yes	String	20190513100506846790529	XWallet uses the user's unique identifier, and the role domain can take effect in the current app.
User Unique Identifier	unionId	Yes	String	20190513100506846790529	XWallet uses the user's unique identifier, and the role domain can take effect under the open platform developer account.
Nickname	nickName	Yes	String	Jack	XWallet user nickname
Avatar	profilePicture	Yes	String	Image link	XWallet link for user avatar
Gender	gender	Yes	Number	1	0 = Confidential 1 = Male 2 = Female

```
{
  "code": 0,
  "data": {
    "appId": "string",
    "openId": "string",
    "unionId": "string",
    "nickname": "string",
    "profilePicture": "string",
    "gender": "string"
  },
  "msg": "string"
}
```

Refresh AccessToken - New Interface

Application Scenario

Update the AccessToken through the interface address before the AccessToken expires.

Interface Address

/api/v1/auth/token

Request Method

PUT

Parameter List

Field Name	Variable Name	Req’d	Type	Value	Description
Application Unique Identifier	appId	Yes	String	20190513100506846790529	Application id assigned by the platform
Random String	nonceStr	Yes	String	3d31858ed6dbd7ee6df1db11036af6d7	Random string, not much larger than 32 bits
Refresh Token	refreshToken	Yes	String	41de9123c544e02a99a8bb58616e04	The refreshToken issued after login with the user authorization is used to update the accessToken, and the expiration time is 35 days.
Timestamp	timestamp	Yes	Number	1557812279	Timestamp, in seconds
Signature	sign	Yes	String	52b5de0a2eb3899683f20401b0a83b0dd918...	SHA256 signature string

Response

The response is in JSON format.

```
{
  "code": 0,
  "data": {
    "accessToken": "string",
    "refreshToken": "string"
  },
  "msg": "string"
}
```

Field Name	Variable Name	Req'd	Type	Value	Description
Access Token	accessToken	Yes	String	i/W0SddsNEONsSy+fd29+w==	accessToken expires in 30 days
Refresh Token	refreshToken	Yes	String	i/W0SddsNEONsSy+fd29+w==	The refreshToken issued after login with the user authorization is used to update the accessToken, and the expiration time is 35 days.

Payment result notification

When the status of the platform reservation order changes (complete, cancel), the platform initiates payment notice.

Application scenario

When the user completes the order or cancels the order, the platform actively informs the third party server. The third party application needs to provide the payment notification callback address in advance;

Interface Address

Provided by the third party and will only support absolute path (with https)

Request Method

POST

Parameter List (in JSON format)

Field name	Variable Name	Req'd	Type	Sample Value	Description
Currency Price	amount	Yes	Number	0.05	Commodity price, subject to the user-specified currency unit
Application Unique Identifier	appId	Yes	String	2019051310050684679O529	Application id assigned by the platform

Field name	Variable Name	Req’d	Type	Sample Value	Description
Application Order Number	appOrderNo	Yes	String	20320513100506846790529	The payment order saved by the third party application service number
Packet	attach	Yes	String	{tag,'myTag'}	The packet information will be in the payment notice Return
Product Name	body	Yes	String	Gifts (roses, fireworks, etc.)	Product name from 3rd party
Currency Unit	currencyUnit	Yes	String	VOF	Use the payment currency unit specified when the user places an order
Product Detail	detail	Yes	String	Virtual gift, rose	Product Detail information
Random String	nonceStr	Yes	String	3d31858ed6dbd7ee6df1db11036af6d7	The normal response is "success"; when an exception occurs, this is the error message.
Notify Number	notifyNum			20190513100506846790529	Platform payment result notification record number
Notify Type	notifyType	Yes	Number	1	Notification type, as shown below Notification type 1 = Completed, completed with user payment 2 = Cancel, cancel the order with the user or the order has not been paid
Quantity	num	Yes	Number	1	Quantity of goods purchased
Order Number	orderNo			20320513100506846790529	The payment order saved by the third party application service number
Timestamp	timestamp	Yes	Number	1557812279	Timestamp, in seconds
Sign	sign	Yes	String	52b5de0a2eb3899683f20401b0a83b0dd918...	SHA256 signature string

Response

The response is in JSON format.

```
{
  "returnCode": "Success",
  "returnMsg": "Ok"
}
```

Field Name	Variable Name	Req'd	Type	Value	Description
Return Code	returnCode	Yes	String	Success	Please fill in the value of the example
Return Message	returnMsg	Yes	String	Ok	Please fill in the value of the example

> Be sure to respond according to the given response format. In order to increase security, it is recommended that the third party check the notification information of the platform through the private key.

Platform notification failure compensation strategy

If the third-party server does not respond correctly to the notification in the given format, the platform will adopt the following strategy:

1. The platform will send 8 notifications within 25 hours if the notification is not properly responded to.
2. 8 notifications are notified by asynchronous method

Open API error codes

Error Code	Description
2001	Invalid appId, or the application has expired, please contact the platform for confirmation.
2002	Invalid signature, please confirm the correctness of the signature
2003	Invalid order number, please confirm the correctness of the platform reservation number
2005	Invalid authToken, authToken has expired or has been used, please re-acquire
2007	Invalid receipt currency, not within the specified receipt currency
3001	Unconfigured payment result notification callback address
3002	User information for XWallet not found
3004	No valid developer account, platform developer account has expired, please contact the platform for confirmation
3005	This feature has been closed or expired. Please contact the platform for confirmation.

JSSDK

1. In the current debugging environment, the authorization login and payment authorization functions (auth, pay) are provided by default.
2. Introduce js file into html header

```
<script src="https://open-auth-uat-2.pundix.com/platform/dist/pundix.js"></script>
```

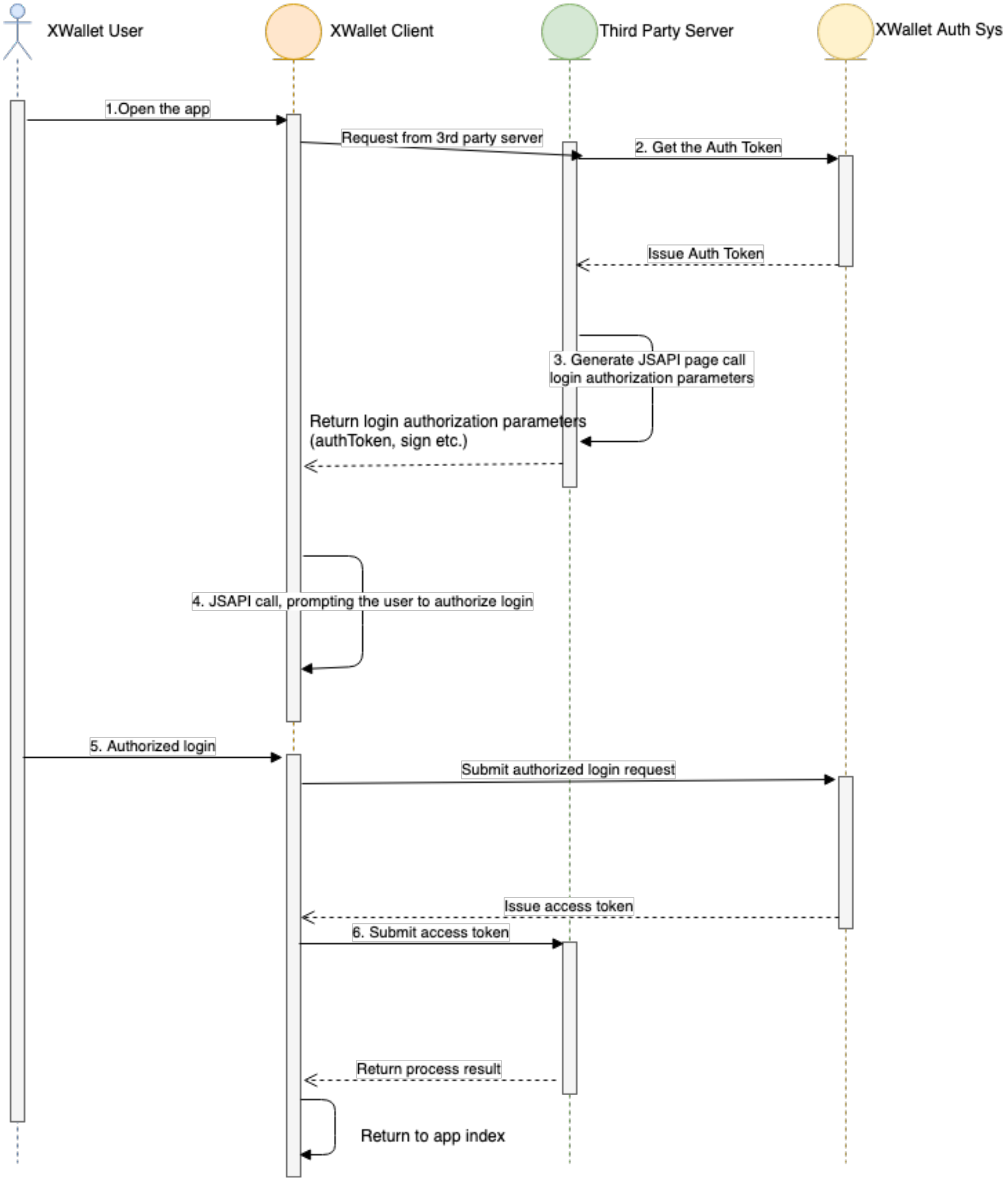
```
// Initialize
PX.init({
    appId: '', // required
    timestamp: '', // Required timestamp, seconds
    nonceStr: '', // Required random number, not greater than 32 bits
    sign: '', // required (signature method 见 above) sign by appId, nonceStr, timestamp + Secret signature
    debug: false,
    jsApiList: ['auth', 'pay'], // required,
    success: function(res) {
         /*
             // The available api value is true, not available as false
            { "auth": true, "pay": true }
        */
      // Return to success can call other methods
     // Login authorization PX.auth({
     authToken: '', // Required (obtained by auth token to get the interface)
     success: function(res) {
        /*
        // access user data
        { accessToken: 'accessToken', refreshToken: 'refreshToken'}
        */
          },
          fail: function(res) {
             { "code":"","msg":"" }
          }
});

// Payment Note: There is no need to pass the appId on this interface, but the signature still needs the appId to participate
 PX.pay({
      timestamp: 0, // timestamp
      nonceStr: '', // random string, no less than 32 bits
      data: '',
      // Unify the value of the prepayOrderNo parameter returned by the single interface port.
     //The submission format is as follows:
     prepayOrderNo=2019052014x
     sign: '', // Required (signature method see above)
          success: function(res) {
            /*
             // payment success information
            {"result":"pay Success"}
            */
          },
          fail: function(res){
            { "code":"","msg":"" }
          }
});
         // Customize shared content
         // Note: (If you don't call this method, the default page share will be shared by default)
         PX.sharedData({
             title: '', // share the title
             imgUrl: '', // image link
             description: '', // share description
             link: '', // Share the link, the link domain name must be the same as the domain name. filled in by
         the open platform.
             success: function(res) {
             // set successfully //{
                     // title='Pundi X',
                     // imgUrl='https://pundix.com/logo/logo.png',
                     // description='Making cryptocurrency accessible to everyone.',
                     // link='https://pundix.com',
                     // url='https%3A%2F%2Ftest-0514api.pundix.com%2Fplatform%2Fdist%2Findex.html'
                   //}
                 },
                 fail: function(res) {
         // Setup failed }
         });
              },
              fail: function(res) {
         //
         } });
```

3. Method Calls

> Note: The PX.pay interface does not need to pass the appId, but the sign at that place still needs the appId to participate in the signature.

Login authorization

Authorized login timing diagram:

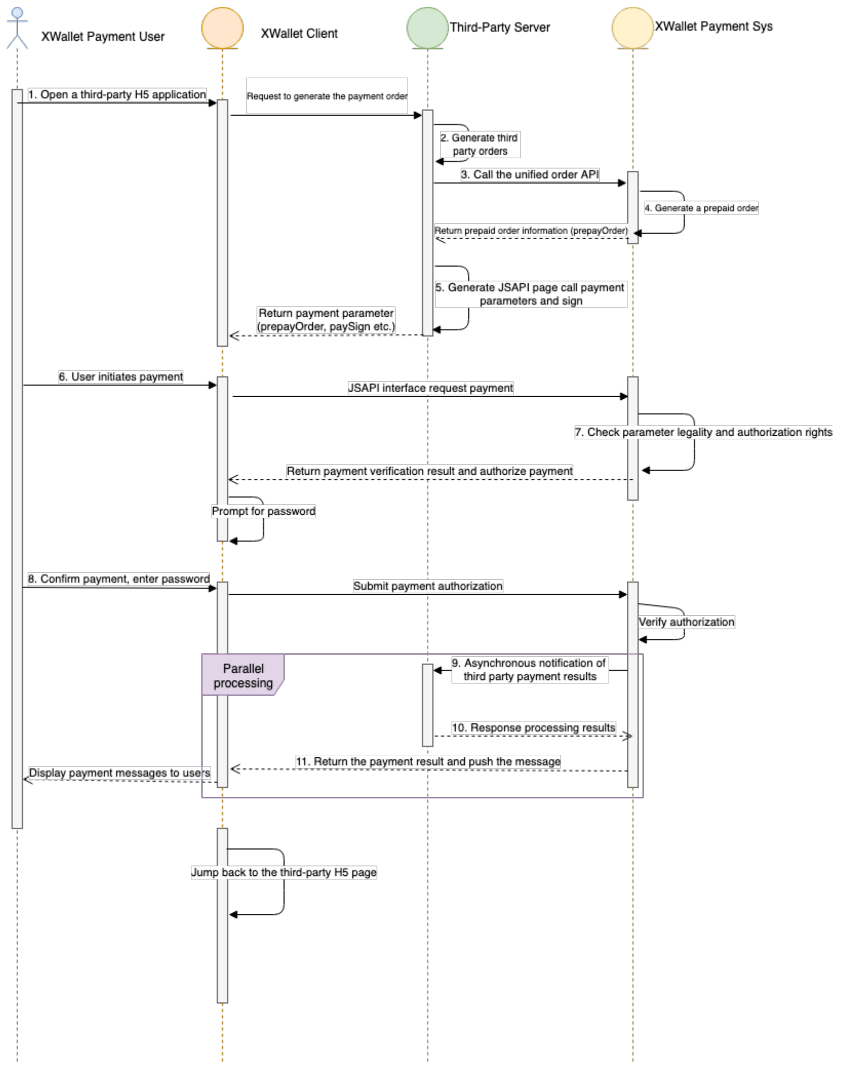


Access process

1. When the user opens the H5 application page, the third-party server is requested to obtain the authToken.
2. The third-party server obtains the authToken through the authToken API.
3. Generate the page surface JSAPI call with the required parameters and signature
4. Called with JSAPI, XWallet application uses the outgoing call authorization page to guide the user authorization.
5. Use the user to perform the login authorization, submit the authorization request to the platform verification server to complete the verification, and issue the accessToken after the verification is passed.
6. The third-party application needs to store the accessToken so that it can be used for accessing user data at a later time.

Payment authorization

Authorization payment timing diagram:


XWallet Payment User
XWallet Client
Third-Party Server
XWallet Payment Sys
1. Open a third-party H5 application
Request to generate the payment order
2. Generate third party orders
3. Call the unified order API
4. Generate a prepaid order
Return prepaid order information (prepayOrder)
5. Generate JSAPI page call payment parameters and sign
Return payment parameter (prepayOrder, paySign etc.)
6. User initiates payment
JSAPI interface request payment
7. Check parameter legality and authorization rights
Return payment verification result and authorize payment
Prompt for password
8. Confirm payment, enter password
Submit payment authorization
Verify authorization
Parallel processing
9. Asynchronous notification of third party payment results
10. Response processing results
11. Return the payment result and push the message
Display payment messages to users
Jump back to the third-party H5 page

Access process

1. Open the application with the user, and complete the order operation within the application.
2. Third-party server generates orders
3. Call the platform to unify the single API to complete the prepaid order
4. The platform generates a prepaid order and returns the prepaid order number.
5. The third-party server generates the JSAPI, the page surface payment authorization call parameters and the signature
6. The user initiates the payment, and the JSAPI calls the platform to perform the line parameter and authorization authority check.
7. After the platform payment authorization verification is completed, the payment payment panel is called out, and the user enters the password to complete the payment.
8. Confirm the payment with the user, enter the password, and submit the payment information.
9. The platform asynchronously notifies the third party server to pay the result
10. Third-party server response notification processing
11. Return the payment result to XWallet and inform the user to pay the information

Signature tool

Signature tool entry address