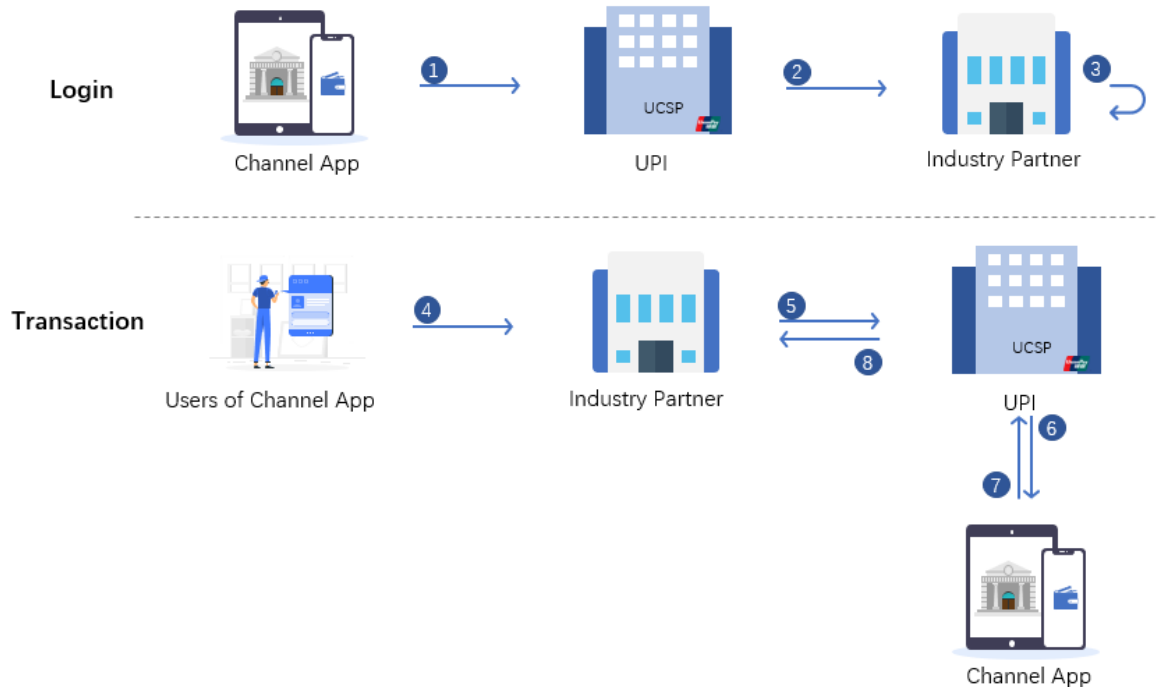


UnionPay International Developer

 developer.unionpayintl.com/cjweb/api/detail

Flow Chart



Login Process:

1. Users log in channel App
2. USCP transfer login information to Industry Partner
3. Industry Partner realize joint login;

Transaction Process:

4. Users place an order in channel App
5. Industry partner call USCP interface
6. Users pay the order and initiate transaction
7. Transaction result is returned to UCSP and Industry Partner

API Reference

Interface description

Industry Service Login Interface: After the UCSP receives user data from the Channel APP, it forwards user data to the Industry Service in Industry Service login interface INDUSTRY_LOGIN_INFO. The Industry Service receives the user data and completes user logs in after successful signature verification. ▪ Interface provider: Industry Service
▪ URL format: https://{INDUSTRY_INDEX_URL}?request_data={INDUSTRY_LOGIN_INFO}

Request Method

HTTP GET

Request Parameter

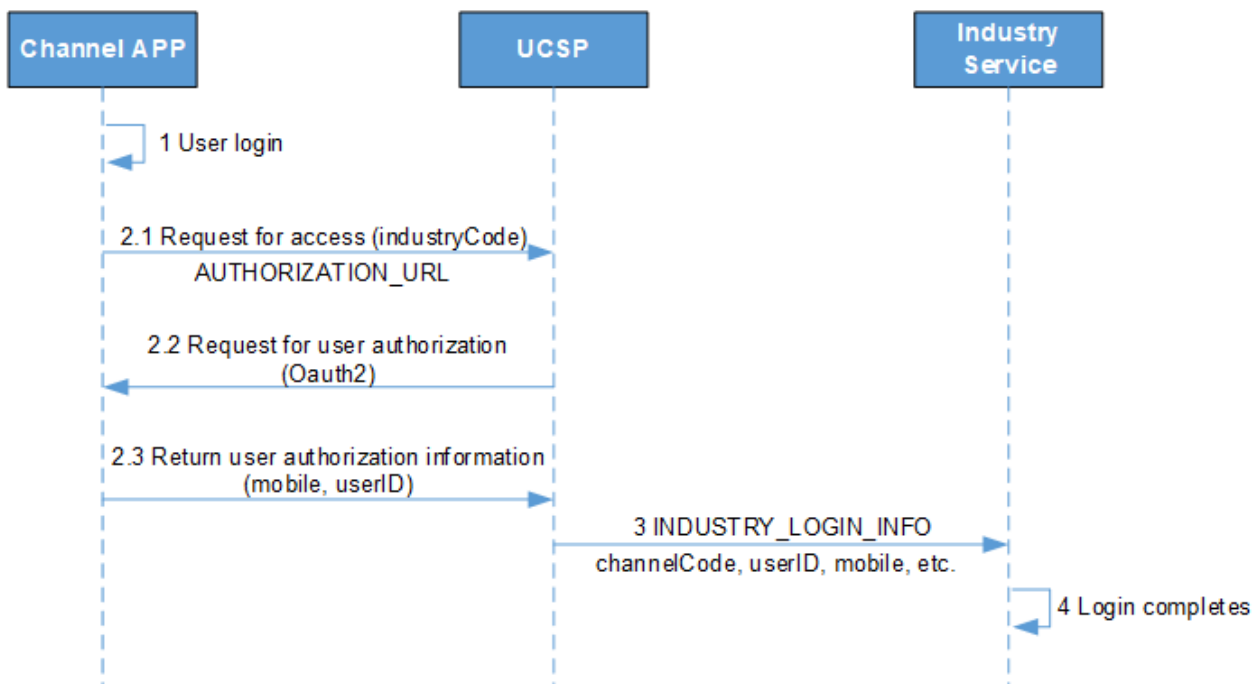
Field name	Identifier	Type	Length	Request	Default value	Note
Message Information	msgInfo	Object		M:Mandatory		
Version Number	versionNo	ANS	5	M:Mandatory		
Message ID	msgID	AN	29	M:Mandatory	Format: A + 8-digit industry code + YYYYMMDDhhmmss + 6-digit random number	
Time Stamp	timeStamp	N	14	M:Mandatory	YYYYMMDDhhmmss	
Message Type	msgType	ANS	1-50	M:Mandatory		
Industry Code	industryCode	AN	8	M:Mandatory	Industry code	
Transaction Information	trxInfo	Object		M:Mandatory		

Channel Code	channelCode	AN	8	M:Mandatory	Channel code
Mobile Number	mobile	ANS	1-64	M:Mandatory	Encrypted transmission
User Identification	userId	ANS	1-64	M:Mandatory	Fill in openID of UnionPay App when UnionPay APP is the Channel provider
Certificate and Signature	certificateSignature	Object		M:Mandatory	
UCSP Signature Certificate ID	ucspSignCertID	AN	1-128	M:Mandatory	Certificate serial ID, the UCSP generates signature using private key of the certificate.
App Signature Certificate ID	appSignCertID	AN	1-128	O:Optional	Certificate serial ID, the Industry Service backend generates signature using private key of the certificate.
App Encryption Certificate ID	appEncCertID	AN	1-128	C:Conditional	Present if encrypted information exists in request message. Certificate serial ID, the UCSP encrypts data using public key of the certificate.
Signature	signature	ANS	1-2048	M:Mandatory	

Sequence Chart

1. Authorized Login Flow

- Applies to the Channel APPs that provides OAUTH authorization.
- Message flow:



1.The user logs in on the Channel APP client side and clicks on the Industry Services icon.

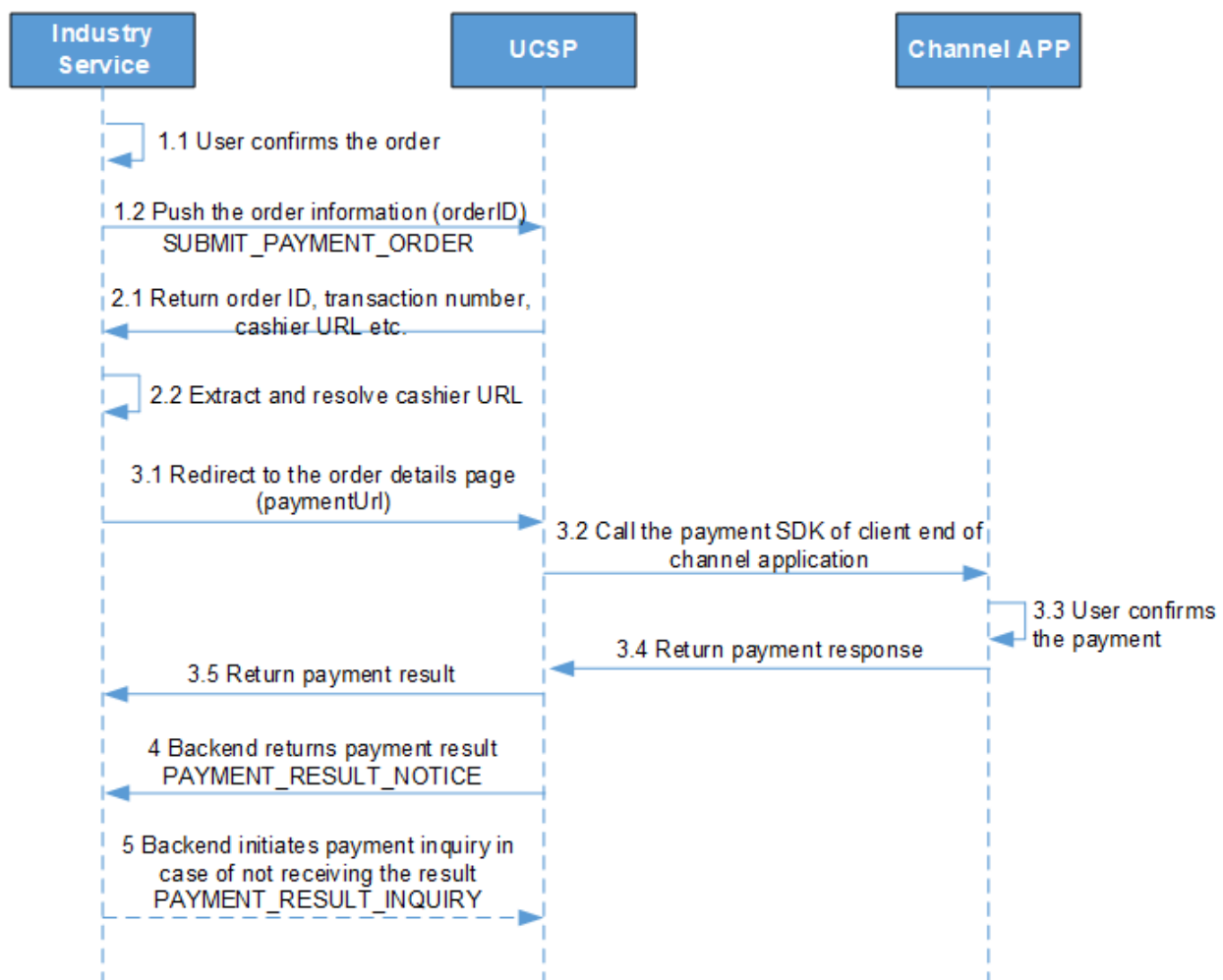
2.The Channel APP redirects to the UCSP authorized login interface AUTHORIZATION_URL with parameters including “industryCode” assigned to the Industry Service when the Industry Service first connects to UCSP and “channelCode”. The UCSP receives the access request, and then requests OAUTH authorization from the Channel APP to obtain user login credentials (“userID”, “mobile”). This step describes interactions between Channel APP and the UCSP, and technical details are out of the scope of the UCSP Specs.

3.Having obtained user credentials, the UCSP uses “industryCode” in the AUTHORIZATION_URL to obtain the login page URL of the Industry Service (INDUSTRY_INDEX_URL). Meanwhile, it generates signature based on data such as user credentials (“userID”, “mobile No.”) and forms the INDUSTRY_LOGIN_INFO message as the input parameter “request_data” of login page URL. The UCSP then redirects to the login page of the Industry Service.

4.The Industry Service receives the authorized login requests, and verifies the signature of the parameter data “request_data”. If the verification succeeds, the Industry Service obtains user login credential (“userID” and “mobile”), completes authorized login, and enters the service page.

2. Frontend Payment Flow

- Applies to all payment scenarios where the frontend calls the payment SDK.
- Message flow:



1 . The user clicks on the Submit order button on the Industry Service page displayed in the Channel APP. The Industry Service pushes the order information (“orderId”, “merId”, “acqIIN”, “txnAmt”, “currencyCode”) to the backend of the UCSP in the SUBMIT_PAYMENT_ORDER request.

2.The UCSP backend makes a request to the downstream system to place the order. When the order is placed, it returns the transaction number “tn” and the UCSP cashier page URL “paymentUrl” to the Industry Service backend in the SUBMIT_PAYMENT_ORDER response.

3.The Industry Service backend resolves the cashier page (“paymentUrl”) and returns to the industry page which then redirects to the cashier page. Users can check the order details and click to pay upon confirmation. The page calls the payment SDK of the Channel APP and invokes the cashier module of the client. Users can select cards and confirm payment in the cashier module. Upon acceptance, the payment SDK returns the transaction completion result to the cashier page which redirects to the payment success page.

4. After the UCSP receives the final payment result from downstream system, the UCSP forwards the result to the Industry Service in the PAYMENT_RESULT_NOTICE.

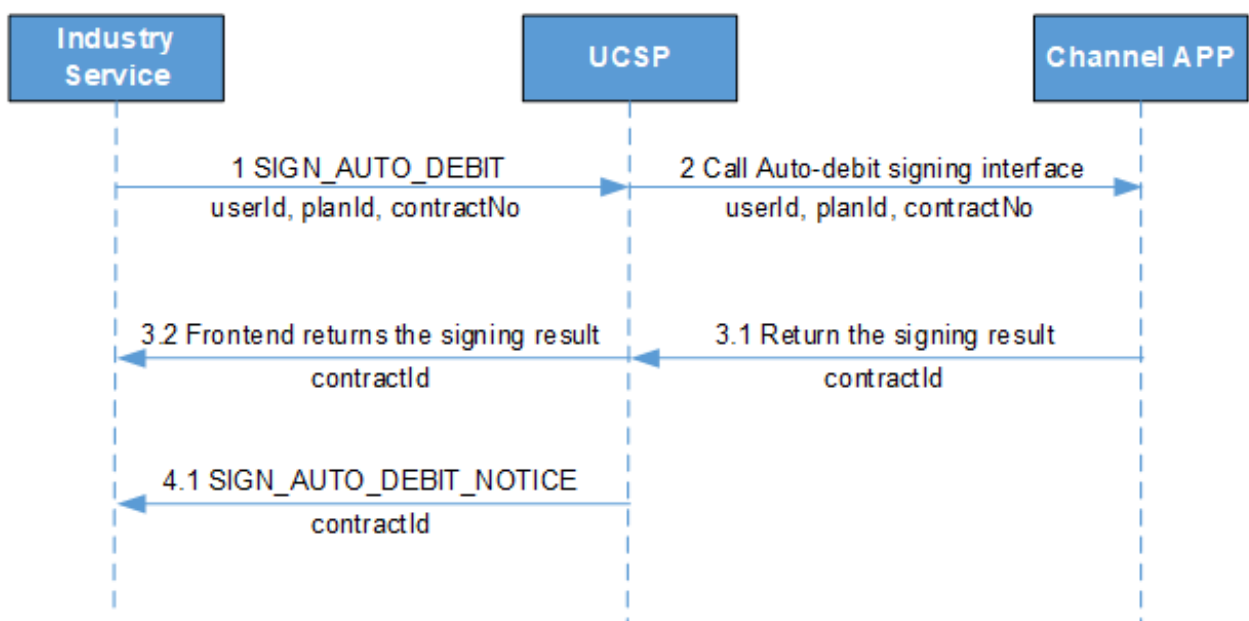
5. In case of network timeout where the backend of the Industry Service does not receive the payment result, the Industry Service can initiate PAYMENT_RESULT_INQUIRY to the UCSP to inquire the result.

3. Backend Payment Flow

- Applies to the scenario where the user has signed auto-debit payment in the Channel APP in advance and the backend performs payment automatically without further cardholder verification.

- Message flow:

3.1 Auto-debit Payment Signing



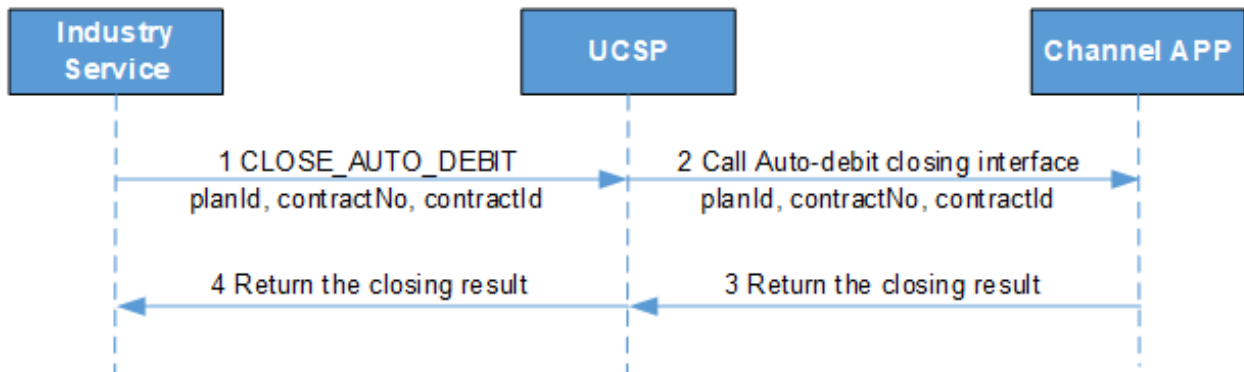
1-2. On the Industry Service page, when the user clicks the Sign Auto-debit Payment button, the Industry Service sends SIGN_SUTO_DEBIT to the UCSP, and the Service page will be redirected to the signing page of the Channel APP through UCSP.

3. The user reads the service content and the contract, and clicks to switch on the auto-debit payment service. The Channel APP returns the “contractId” and the contract result to UCSP. The UCSP obtains the “frontUrl” submitted in the SIGN_AUTO_DEBIT interface and redirects to the Industry Service frontend. The Industry Service refreshes its page to display the result.

4. The UCSP obtains the “backUrl” submitted in the SIGN_AUTO_DEBIT interface and pushes the signing result (“contractId”) to the Industry Service backend in

SIGN_AUTO_DEBIT_NOTICE. The “contractId” is saved by the Industry Service as a key element of the auto-debit payment.

3.2 Auto-debit Payment Closing



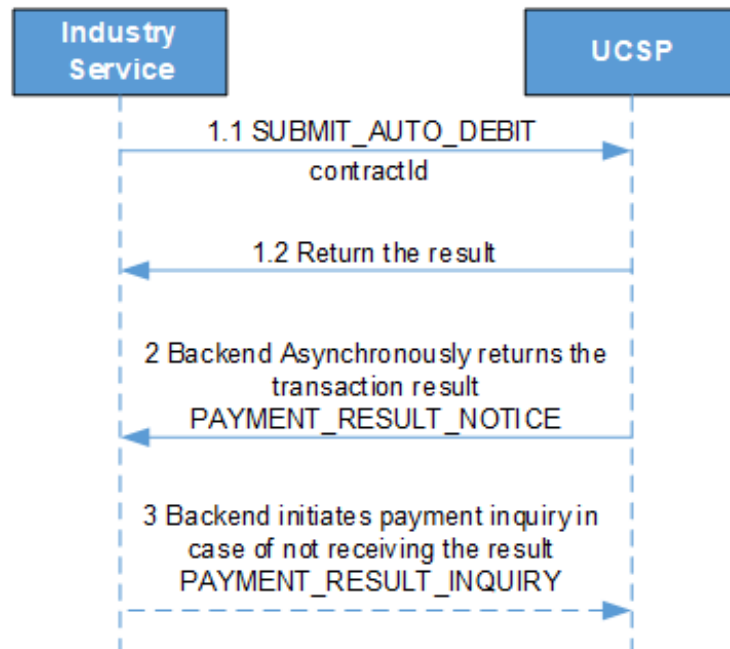
1. On the Industry Service page, when the user clicks the Close Auto-debit Payment button, the Industry Service backend will send the CLOSE_AUTO_DEBIT request to UCSP.

2. Based on the “contractID”, the UCSP calls the auto-debit closing interface to the corresponding contracted Channel APP.

3. When the contract is closed, the Channel APP returns the result to the UCSP backend.

4. The UCSP returns the result to the Industry Service backend in the CLOSE_AUTO_DEBIT response.

3.3 Backend Payment Flow

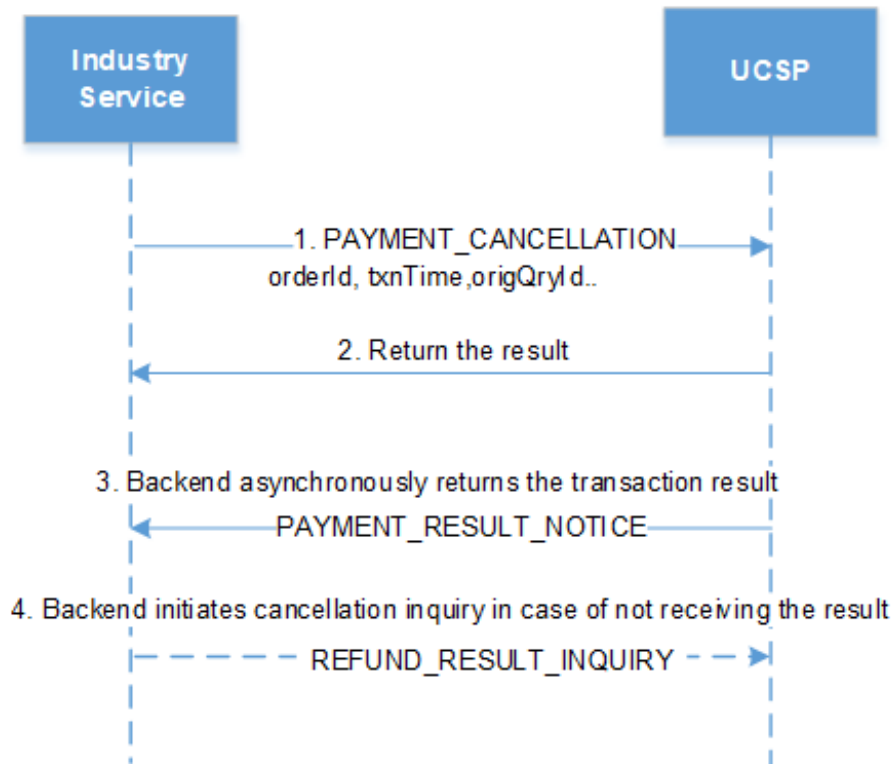


1. In the payment scenario, the user triggers fee deduction of the Industry Service. The backend submits the automatic debit request SUBMIT_AUTO_DEBIT (“contractId”) to the UCSP. The UCSP forwards the payment request to the downstream auto-debit payment system signed by the user.

2. When the payment result from the Issuer reaches the UCSP, the UCSP returns the result to the Industry Service in PAYMENT_RESULT_NOTICE.

3. In case of network timeout, the Industry Service can inquire the payment result of UCSP through PAYMENT_RESULT_INQUIRY. After the UCSP receives synchronous response, it can initiate payment inquiry at intervals of 0s, 1s, 2s, 4s, 8s, 16s, 32s. The UCSP should stop sending inquiries once it receives successful/failed transaction status or it receives backend notification.

4. Payment Cancellation Flow



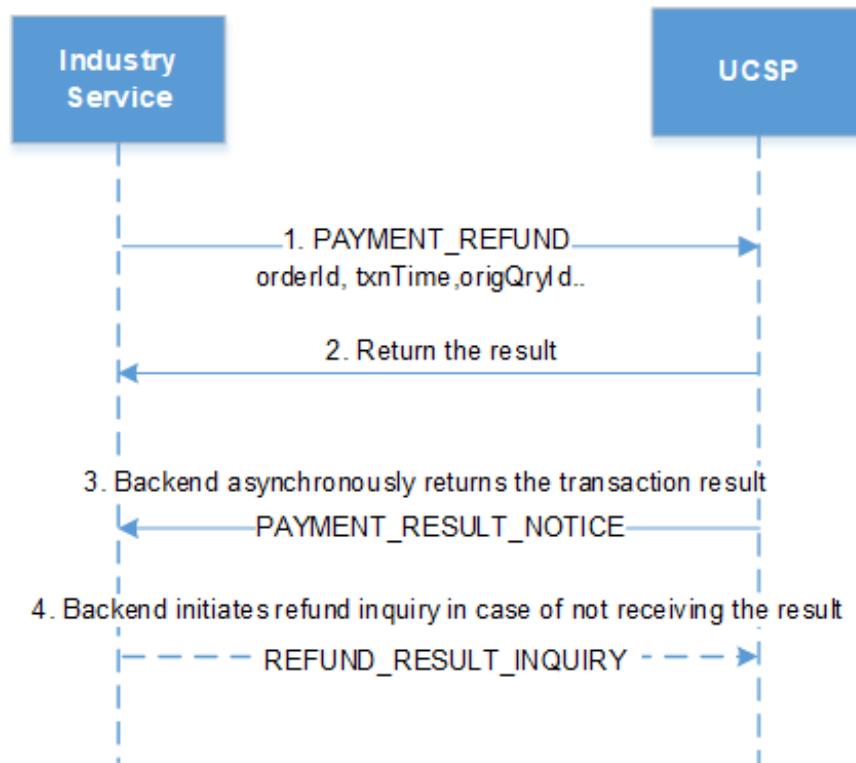
1. When the user requires payment cancellation, the Industry Service initiates payment cancellation request `PAYMENT_CANCELLATION` (“`orderId`”, “`txnTime`”, “`origQryId`”) to the UCSP.

2. The UCSP processes the request and returns the processing result.

3. When the payment result from the Issuer reaches the UCSP, the UCSP returns the result in `PAYMENT_RESULT_NOTICE` to the Industry Service.

4. In case of network timeout, the Industry Service can inquire the cancellation result by sending `REFUND_RESULT_INQUIRY`.

5. Refund Flow



1. When the user requires refund, Industry Service initiates refund request `PAYMENT_REFUND` (“orderId”, “txnTime”, “origQryId”) to the UCSP.

2. The UCSP processes the request and returns the processing result.

3. When the payment result from the Issuer reaches the UCSP, the UCSP returns the result in `PAYMENT_RESULT_NOTICE` to the Industry Service.

4. In case of network timeout, the Industry Service can inquire the refund result of UCSP by sending `REFUND_RESULT_INQUIRY`.

Steps to Launch

When merchants and UnionPay International carry out UCSP business, they will mainly go through steps as "agreement signing", "merchant enrollment", "system development & testing", "going online" and "production verification". Among these steps, “agreement signing”, “merchant enrollment” and “system development & testing” can be conducted in parallel, and other steps need to be carried out in order.

For merchant, please submit order including following information: merchant name, country/region, contact person and contact information

Response Code Reference

Response code	Description
00	Approved
01	Transaction Fail
05	Accepted, need to inquiry transaction result
10	Message Format Invalid
11	Signature Verification Fails
34	Transaction Order not Found
36	Order Info is Incorrect
99	Common Error