SDP Solution API Reference (AM, ParlayX 3.0)

Change History

Changes between document issues are cumulative. The latest document issue contains all the changes made in earlier issues.

Issue 13 (2016-04-28)

This issue is the thirteenth official release, and includes the following changes:

Update the namedParameters field in the request Header for balanceUpdate in 5.1 balanceUpdate.

Issue 12 (2016-01-28)

This issue is the twelfth official release, and includes the following changes:

Add the namedParameters field in the request Header for balanceUpdate in 5.1 balanceUpdate.

Issue 11 (2015-11-20)

This issue is the eleventh official release, and includes the following changes:
Password encryption algorithm supports SHA256.

Issue 10 (2015-06-18)

This issue is the tenth official release, and includes the following changes:
Update the description of **enduserDAAccountId** in getBalance message header.

Issue 09 (2014-12-05)

This issue is the ninth official release, and includes the following changes:
Add the fields **faultcode** and **faultstring** for Service Error and Policy Error.

Issue 08 (2014-05-19)

This issue is the eighth official release, and includes the following changes:
Update Chapter 3.1 for changing fields in getBalance message header parameters and getBalanceResponse message.

Issue 07 (2014-05-04)

This issue is the seventh official release, and includes the following changes:
Change the description of **serviceId** and **bundleID**.

Issue 06 (2014-02-15)

This issue is the sixth official release, and includes the following changes:

Add the fields **endUserDAAccountId**, **extensionInfo**, **expiryDate**, and **description** for getBalance.

Issue 05 (2013-12-23)

This issue is the fifth official release, and includes the following changes:

Add the fake ID function in Phase2.3 version.

To use the fake ID function, SPs must modify their systems to change all mobile numbers involved in old and new services to fake IDs so that numbers sent by the SP systems to the SDP are all fake IDs. The SDP converts the received fake IDs to mobile numbers for service processing.
SPs must obtain the mapping between mobile numbers involved in old services and fake IDs from the MTN carrier.

If SPs still use mobile numbers when the fake ID function is enabled, service processing will fail.

Issue 04 (2013-11-19)

This issue is the fourth official release, and includes the following changes:
Add the Appendix including details about currency, atomic charging unit, ISO 4217 codes and OpCoID list.

Issue 03 (2013-09-22)

This issue is the third official release, and includes the following changes:
Change the format from the Huawei style to MTN style.

Issue 02 (2013-08-05)

This issue is the second official release, and includes the following changes:
Update Chapter 3 ,4 ,5 for adding **bundleID** field.

Issue 01 (2013-06-22)

This issue is the first official release.

Contents

01

Overview

1.1 Scenario

Pre-paid subscribers, whether they have subscribed to pre-paid telephony, SMS, or data service, have credits with their service providers; the consumption of services will lead to reduction of their credit, or the credit may expire. Therefore, from time to time, subscribers may have to recharge their accounts. This occurs through an application that interfaces with the subscriber either directly or indirectly. Examples of direct interaction are voice prompts and WAP/web pages, or even SMS. Typically, such multi-modal applications either request a currency amount and, e.g. credit card information, or a voucher number plus credentials. The voucher number and credentials are then validated and causes a pre-determined currency amount to be transferred.
The Parlay X Account Management API described in the present document supports account querying, direct recharging and recharging through vouchers. As a side effect, it may prevent subscribers from having their account balance credits expire.

1.1.1 Obtaining Account Balance Information

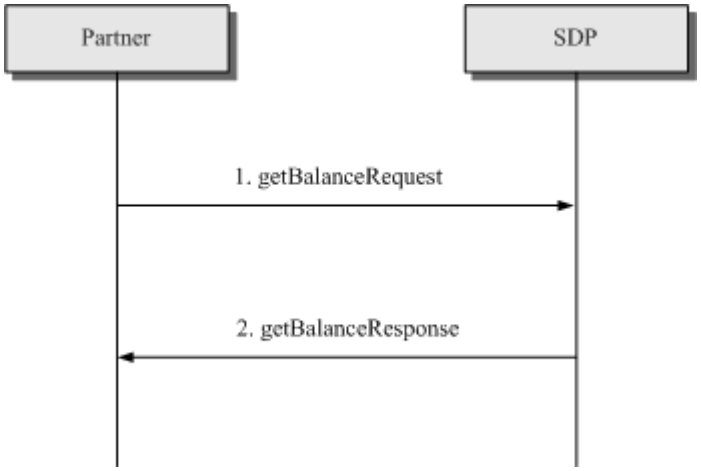
Introduction

Partner invokes SDP's API getBalance to send the request for obtaining the subscriber's account balance.

Obtaining Account Balance Information

Figure 1-1 shows the process for a Partner to get the account balance.

Figure 1-1 Obtaining Account Balance Information



The process is as follows:

1. Partner invokes SDP's API getBalance to send the request for obtaining the subscriber's account balance.
2. SDP returns the result to the Partner.

1.1.2 Querying the Expiration Date for the Balance

Introduction

Partner invokes the SDP's getCreditExpiryDate API to obtain the account balance expiration date.

Querying the Expiration Date for the Balance

Figure 1-2 shows the process for a Partner to obtain the account balance expiration date.

Figure 1-2 Querying the Expiration Date for the Balance

The process is as follows:

1. Partner invokes the getCreditExpiryDate API to send a request to SDP querying the account balance expiration date.
2. SDP returns the result to Partner.

1.1.3 Recharging using direct payment

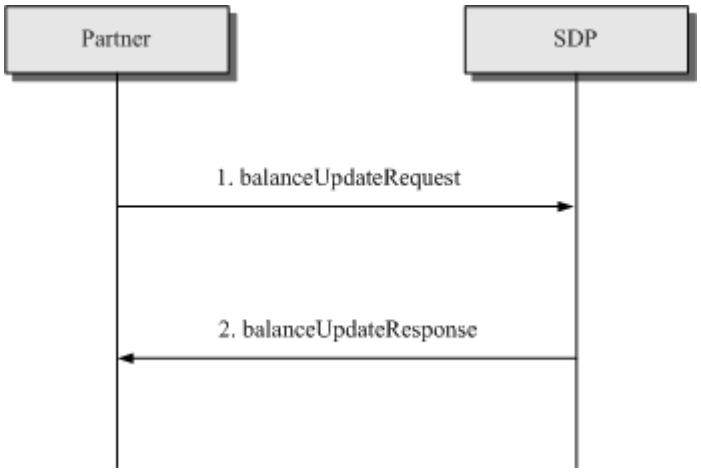
Introduction

Partner invokes the SDP's balanceUpdate API to recharge using direct payment.

Recharging Using Direct Payment

Figure 1-3 shows the process of recharging using direct payment.

Figure 1-3 Recharging Using Direct Payment



The process is as follows:

1. Partner invokes the balanceUpdate API to send a request of recharging using direct payment for a subscriber.
2. SDP returns the result to Partner.

1.1.4 Recharging using a voucher

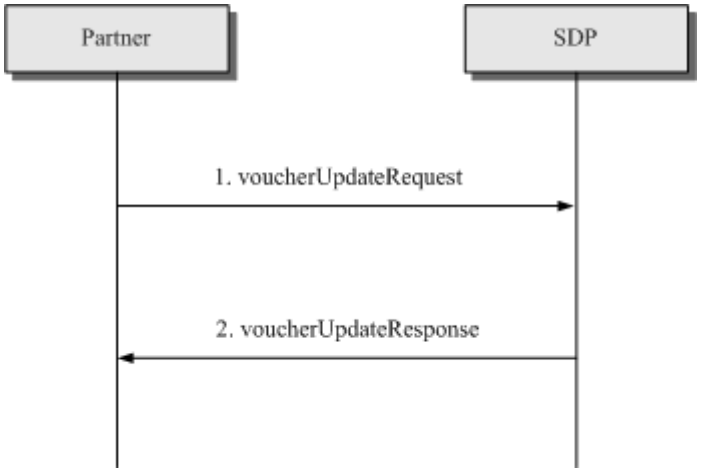
Introduction

Partners invoke the voucherUpdate interface provided by the SDP to recharge users' accounts using a voucher.

Recharging Using a Voucher

Figure 1-4 shows the process of recharging using a voucher.

Figure 1-4 Recharging Using a Voucher



The process is as follows:

1. Partner invokes the voucherUpdate API to send a request of recharging using a voucher for a subscriber
2. SDP returns the result to Partner.

1.1.5 Querying Users' Operation Records

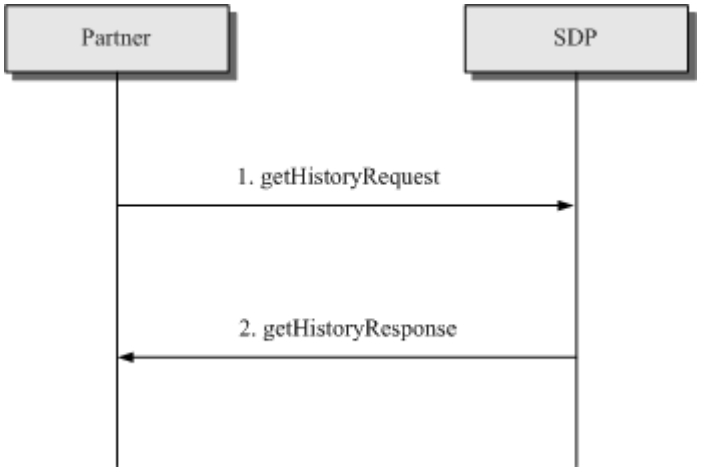
Introduction

Partner invokes the SDP's getHistory API to query the history records of a subscriber's account.

Querying Users' Operation Records

Figure 1-5 shows the process of querying the history records of a subscriber's account.

Figure 1-5 Querying Users' Operation Records



The process is as follows:

1. Partner invokes the getHistory API to send a request of querying the history records of a subscriber's account.
2. SDP returns the result to Partner.

1.1.6 Querying Users' Balance Types

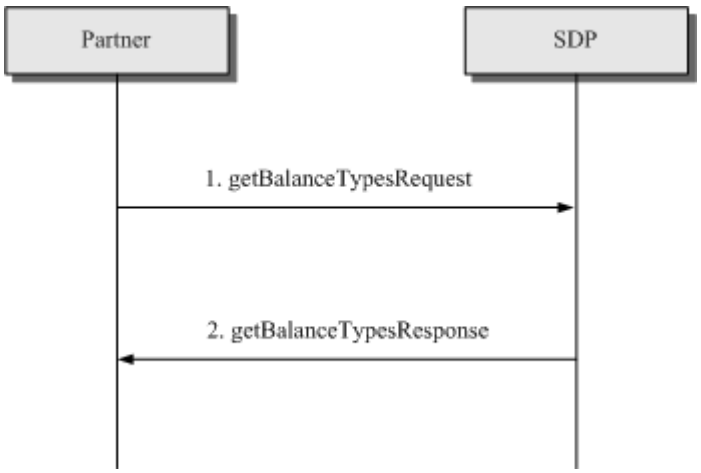
Introduction

The partner invokes the getBalanceTypes interface provided by the SDP to query users' balance types.

Querying Users' Balance Types

Figure 1-6 shows the process of querying users' balance types.

Figure 1-6 Querying Users' Balance Types



The process is as follows:

1. The partner uses the getBalanceTypes interface to send a request to the SDP to query a user's balance type.
2. The SDP sends the result to the partner.

1.2 Interface List

This topic describes Account Management capability interfaces provided by the SDP.
Table 1-1 lists the Account Management capability interfaces provided by the SDP.

Table 1-1 Account Management Parlay X 3.0 Interface List

Function	Interface	Description
Obtaining account balance information	getBalance	The SDP functions as the server and provides the interface for partners. The interface is used to get the balance of a subscriber's account.
Querying the expiration date for the balance	getCreditExpiryDate	The SDP functions as the server and provides the interface for partners. The interface is used to query the expiration date for the balance.
Recharging using direct payment	balanceUpdate	The SDP functions as the server and provides the interface for partners. The interface is used to recharge for a subscriber using direct payment.
Recharging using a voucher	voucherUpdate	The SDP functions as the server and provides the interface for partners. The interface is used to recharge for a subscriber using a voucher.

1 Overview

Function	Interface	Description
Querying users' operation records	getHistory	The SDP functions as the server and provides the interface for partners. The interface is used to query the operation records of a subscriber's account.
Querying users' balance types	getBalanceTypes	The SDP functions as the server and provides the interface for partners. The interface is used to query users' balance types.

02 Error Definition

2.1 Service Error

A service error is an error caused by service interfaces.
Table 2-1 defines the fields in a service error response sent by the server to the client.

Table 2-1 Service error fields

Parameter	Type	Mandatory or Optional	Description
faultcode	xsd:string	Mandatory	Result code. [Format] SVC*ABCD* In the format, **SVC** identifies a service error response, and *ABCD* is a number ranging from 0001 to 9999. [Example] SVC0001
faultstring	xsd:string	Mandatory	Error description. The value can contain the variable %# in definition. When sending a response, the server replaces the variable %# with the value of **variables**. [Example] Waiting for response timed out, message type is OutwardGetLocReq.
messageId	xsd:string	M	ID of an error message. The value is a character string starting with **SVC**. [Format] SVCABCD • **SVC** indicates that the message is an error message. • **ABCD** is a number ranging from 0001 to 9999.

text	xsd:string	M	Error message description, which can contain the variables % and #.
variables	xsd:string [0..unbounded]	O	Values of the variables used in the **text** value.

2.2 Policy Error

A policy error indicates an error reported by the server because of SLA violation.
Table 2-2 defines the fields in a policy error response sent by the server to the client.

Table 2-2 Policy error fields

Parameter	Type	Mandatory or Optional	Description
faultcode	xsd:string	Mandatory	Result code. [Format] POL*ABCD* In the format, **POL** identifies a policy error response, and *ABCD* is a number ranging from 0001 to 9999. [Example] POL0006
faultstring	xsd:string	Mandatory	Error description. The value can contain the variable %# in definition. When sending a response, the server replaces the variable %# with the value of **variables**. [Example] GroupAddr is not supported
messageId	xsd:string	M	ID of an error message. The value is a character string starting with **POL**. [Format] POL[ABCD] • **POL** indicates that the message is an error message. • **ABCD** is a number ranging from 0001 to 9999.
text	xsd:string	M	Error message description, which can contain the variables % and #.

variables	xsd:string [0..unbounded]	O	Values of the variables used in the **text** value.

03 Obtaining Account Balance Information

3.1 getBalance

The SDP functions as the server and provides the interface for partners. Partners use it to get the balance of a subscriber's account.

Request URI

http://*IP*:*Port*/AccountManagementService/services/AccountManagement/v3

In the preceding information, *IP* and *Port* indicate the IP address and port number for the SDP to provide Account Management services. Contact the carrier to obtain the IP address and port number.

Message Header

Table 3-1 describes the parameters in the message header of the getBalance interface.

Table 3-1 getBalance message header parameters

Parameter	Type	Mandatory or Optional	Description
spId	xsd:string	M	Partner ID. Partner ID automatically allocated by the SDP to the Partner after successful registration. To obtain the Partner ID: • For a partner, use the registered account to log in to the SDP Management Portal, choose **Home** > **Partner** > **Access Code**, and query the Partner ID. • For a developer, use the registered account to log in to the Developer Portal, choose **Home** > **My Account**, and query the Developer ID. [Example] 2500110000021

Parameter	Type	Mandatory or Optional	Description
spPassword	xsd:string	O	Authentication key for the SDP to authenticate partners. The SDP supports authentication by **SP ID + Password**, **SP ID + IP address + Password**, or **SP ID + IP address**. Partners select an authentication mode during registration. If a partner selects authentication by **SP ID + Password** or **SP ID + IP address + Password**, this parameter is mandatory in requests sent by this partner. The value is a character string encrypted. The encryption formula is as follows: • SHA-256: spPassword = Base64(SHA-256(spId + Password + timeStamp)) • MD5: spPassword = MD5(spId + Password + timeStamp) In the formula: • **spId** and **timeStamp**: authentication ID and timestamp. • **Password**: password for partners to access the SDP. – For a partner, set the authentication mode and password during registration. If the password is forgotten, use the registered account to log in to the SDP Management Portal, choose **Home** > **Partner** > **Partner Information**, and set a new one. – For a developer, set the authentication mode and password during registration. If the password is forgotten, use the registered account to log in to the Developer Portal, choose **Member Center** > **Account** > **Registration Information** > **Invoke Password**, and set a new one. **NOTE** To retain features of earlier versions, the SP uses the MD5 algorithm in the connection to the SDP, which might cause security risks. [Example] e6434ef249df55c7a21a0b45758a39bb

Parameter	Type	Mandatory or Optional	Description
timeStamp	xsd:string	O	Time stamp (UTC time). The value is required during MD5 encryption for **spPassword**. **NOTE** If the **spPassword** parameter must be set, this parameter is mandatory. [Format] yyyyMMddHHmmss [Example] 20100731064245
serviceId	xsd:string	O	Service ID. To obtain the service ID: • For a service partner, use the registered account to log in to the SDP Management Portal to get the ID. • For an API partner, set it blank. • For a developer: – If the developer has bought the capability, set the value to none. – If the developer has released the capability service, contact the carrier to obtain the service ID. The **serviceId** must be contained during invocation of a service interface developed by service partners and other partners, and must not be contained during invocation of a capability interface developed by API partners, other partners, and developers. [Example] 35000001000001

Parameter	Type	Mandatory or Optional	Description
bundleID	xsd:string	O	Bundle ID. When SDP creates a capability bundle, SDP allocates a bundleID to capability bundle. The **bundleID** must not be contained during invocation of a service interface developed by service partners and other partners, and must be contained during invocation of a capability interface developed by API partners, other partners, and developers. **NOTE** • When an API partner invokes the **getbalabce** interface, this parameter is mandatory. • The **bundleID** is allocated only after an API product bundle is released. Therefore, ensure that the corresponding product bundle has been released on the SDP management portal before using the interface. [Example] 256000039
OA	xsd:string	M	Mobile number or the fake ID of the service originator. [Format] • Mobile number: prefix+country code+mobile number **NOTE** • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639

Parameter	Type	Mandatory or Optional	Description
FA	xsd:string	O	Mobile number or the fake ID of the charged party. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639
endUserDAAccountId	xsd:int (1 to 2147483647)	O	Contain the Specific DA Account identifier. If endUserDAAccountId is NOT provided, SDP will return main balance ONLY. If endUserDAAccountId is provided and NOT 0, SDP will return main balance and ONE dedicatedAccount Information specified in the request. If endUserDAAccountId provided is 0, SDP will return main balance and ALL dedicatedAccounts belonging to the subscriber.

getBalanceRequest

Table 3-2 describes the input parameters in the **getBalanceRequest** message.

Table 3-2 getBalanceRequest input parameters

Parameter	Type	Mandatory or Optional	Description
endUserIdentifier	xsd:anyURI	M	This parameter identifies the end user's account. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639
endUserPin	xsd:string	O	Contains the end user's credentials for authorizing access to the account. [Example] 1212

getBalanceResponse

Table 3-3 describes the parameters in the **getBalanceResponse** message.

Table 3-3 getBalanceResponse parameters

Parameter	Type	Description
result	Balance [1...unbounded]	It is a set of Balance records, where each record specifies a balance type and the associated amount. For the error codes, see Table 3-4.

Table 3-4 Balance parameters

Parameter	Type	Description
accountID	xsd:string	ID of the returned balance. 0 for Main Account. Dedicated AccountID for DA Account.
balanceType	xsd:string	Identifies the type of balance. A user account may have multiple types of balances for different usage, for example, voice, SMS, and game

Parameter	Type	Description
		services.
amount	xsd:decimal	Amount of balance.
expiryDate	xsd:dateTime	Expiry Date of the account.
description	xsd:string	Description of the account.
extensionInfo	NamedParameterList	Extended parameter.

Table 3-5 NamedParameter

Parameter	Mandatory/ Optional	Type	Description
key	M	String	Name of the extended parameter.
value	M	String	Value of the extended parameter.

Table 3-6 NamedParameterList

Parameter	Mandatory/ Optional	Type	Description
NamedParameters	Optional	NamedParameter[ ]	List of extended parameters.

Error Code

Table 3-7 describes the error codes that the SDP may return.

Table 3-7 getBalance error codes

Error Code	Description	Cause
SVC0001	Service error.	A service exception occurred in the internal SDP.
	A service error occurred.	An internal SDP service is abnormal.
SVC0901	SPID is null!	The **spId** value in the request header is blank or does not exist in the SDP.
	SPID %1 is invalid!	The **spId** value in the request header is in an incorrect format.
	Sp password is null!	The **password** value in the request header is blank.

Error Code	Description	Cause
	Service ID is null!	The **serviceId** value in the request header is blank or does not exist in the SDP.
	Service ID %1 is invalid!	The **serviceId** value in the request header is in an incorrect format.
	Timestamp is empty in soapheader.	The **timeStamp** value in the request header is blank.
	SP ip is null!	The IP address in the request header is blank.
	The Created is null.	The **timeStamp** value in the request header is blank.
	The Nonce is null.	The **Nonce** value in the request header is blank.
	Sp password is null!	The **password** value in the request header is blank.
	The PasswordDigest is not accepted.	The password in the cipher text is incorrect.
	Sp ip %1 is not accepted!	The IP address in the request header is incorrect.
	Sp password is not accepted!	The **password** value in the request header is incorrect.
	The Sp has not logged in.	The **Token** value in the request header does not exist in the SDP and the Partner needs to log in first.
	The token is not accepted!	The **Token** format in the request header is incorrect.
	The token is expired!	The **Token** value in the request header is expired.
	local SP password is null!	An exception occurred in the internal SDP.
	The sp's Status is deregistered.	The Partner is in the deregistered state.
	The sp's Status is pre-deregistered.	The Partner is in the pre-deregistered state.
	SP %1 is in blacklist!	The Partner account is in the blacklist.
	The sp's Status is unknown.	An internal SDP service is abnormal.
	SPID %1 is not exist!	The Partner specified by **spId** in the request header does not exist in the SDP.

Error Code	Description	Cause
	The service status is configuring.	The service specified by **serviceId** in the request header is in the configuring state.
	The service status is suspended.	The service specified by **serviceId** in the request header is in the paused state.
	The service status is pre-deregistered.	The service specified by **serviceId** in the request header is in the pre-deregistered state.
	The service status is deregistered.	The service specified by **serviceId** in the request header is in the deregistered state.
	Service ID %1 is not exist!	The **serviceID** value in the request header does not exist in the SDP.
	The service status is unknown.	An internal SDP service is abnormal.
	Service %1 is in blacklist!	The **serviceId** value in the request header in the blacklist.
	The API %1 is not existed.	This Partner does not have the permission for using the API.
	The API status is disabled.	This Partner does not have the permission for using the API.
	SP status is locked.	The Partner is in the locked state.
	The sp's Status is forbidden.	The Partner is in the forbidden state.
	The sp's status is pause.	The Partner is in the paused state.
SVC0905	AuthToken %1 is invalid	The **AuthToken** value in the request header sent by the partner is invalid.

Example

Examples of getBalance messages in typical scenarios are as follows:

- When a service partner sends the request message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc="http://www.csapi.org/schema/parlayx/account_management/v3_1/local">
   <soapenv:Header>
      <tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com.cn/schema/common/v2_1">
         <tns:spId>011104</tns:spId>
         <tns:spPassword>557d5fd8fb01089acd744245fc011036</tns:spPassword>
         <tns:timeStamp>20120809114701</tns:timeStamp>
         <tns:serviceId>35000001000119</tns:serviceId>
         <tns:OA>8613900000001</tns:OA>
```

```
        <tns:FA>8613900000001</tns:FA>
      </tns:RequestSOAPHeader>
    </soapenv:Header>
    <soapenv:Body>
      <loc:getBalance>
        <loc:endUserIdentifier>8613812345678</loc:endUserIdentifier>
        <loc:endUserPin>1212</loc:endUserPin>
      </loc:getBalance>
    </soapenv:Body>
</soapenv:Envelope>
```

- When an API partner sends the request message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc="http://www.csapi.org/schema/parlayx/account_management/v3_1/local">
   <soapenv:Header>
      <tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com.cn/schema/common/v2_1">
         <tns:spId>011104</tns:spId>
         <tns:spPassword>557d5fd8fb01089acd744245fc011036</tns:spPassword>
         <tns:timeStamp>20120809114701</tns:timeStamp>
         <tns:bundleID>256000039</tns:bundleID>
         <tns:OA>8613900000001</tns:OA>
         <tns:FA>8613900000001</tns:FA>
      </tns:RequestSOAPHeader>
   </soapenv:Header>
   <soapenv:Body>
      <loc:getBalance>
         <loc:endUserIdentifier>8613812345678</loc:endUserIdentifier>
         <loc:endUserPin>1212</loc:endUserPin>
      </loc:getBalance>
   </soapenv:Body>
</soapenv:Envelope>
```

- The following is an example of the response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <ns1:getBalanceResponse
xmlns:ns1="http://www.csapi.org/schema/parlayx/account_management/v3_1/local">
         <ns1:result>
            <accountID>0</accountID>
            <balanceType>SMS</balanceType>
            <amount>600</amount>
           <expiryDate>2004-02-15T02:44:14</expiryDate>
            <description>temperat iras</description>
            <extensionInfo>
               <item>
                  <key>turbine corripuit</key>
                  <value>regemque dedit</value>
```

```
            </item>
          </extensionInfo>
        </ns1:result>
      </ns1:getBalanceResponse>
    </soapenv:Body>
  </soapenv:Envelope>
```

04

Querying the Expiration Date for the Balance

4.1 getCreditExpiryDate

The SDP functions as the server and provides the interface for partners. Partners use it to get the account balance expiration date.

Request URI

http://*IP*:*Port*/AccountManagementService/services/AccountManagement/v3

In the preceding information, *IP* and *Port* indicate the IP address and port number for the SDP to provide Account Management services. Contact the carrier to obtain the IP address and port number.

Message Header

Table 4-1 describes the parameters in the message header of the getCreditExpiryDate interface.

Table 4-1 getCreditExpiryDate message header parameters

Parameter	Type	Mandatory or Optional	Description
spId	xsd:string	M	Partner ID. Partner ID automatically allocated by the SDP to the Partner after successful registration. To obtain the Partner ID: • For a partner, set the authentication mode and password during registration. If the password is forgotten, contact the carrier to retrieve the password. • For a partner, use the registered account to log in to the SDP Management Portal, choose **Home** > **Partner** > **Access Code**, and query the Partner ID. • For a developer, use the registered account to log in to the Developer Portal, choose **Home** > **My Account**, and query the Developer ID. [Example] 2500110000021

Parameter	Type	Mandatory or Optional	Description
spPassword	xsd:string	O	Authentication key for the SDP to authenticate partners. The SDP supports authentication by **SP ID + Password**, **SP ID + IP address + Password**, or **SP ID + IP address**. Partners select an authentication mode during registration. If a partner selects authentication by **SP ID + Password** or **SP ID + IP address + Password**, this parameter is mandatory in requests sent by this partner. The value is a character string encrypted. The encryption formula is as follows: • SHA-256: spPassword = Base64(SHA-256(spId + Password + timeStamp)) • MD5: spPassword = MD5(spId + Password + timeStamp) In the formula: • **spId** and **timeStamp**: authentication ID and timestamp. • **Password**: password for partners to access the SDP. – For a partner, set the authentication mode and password during registration. If the password is forgotten, use the registered account to log in to the SDP Management Portal, choose **Home** > **Partner** > **Partner Information**, and set a new one. – For a developer, set the authentication mode and password during registration. If the password is forgotten, use the registered account to log in to the Developer Portal, choose **Member Center** > **Account** > **Registration Information** > **Invoke Password**, and set a new one. **NOTE** To retain features of earlier versions, the SP uses the MD5 algorithm in the connection to the SDP, which might cause security risks. [Example] e6434ef249df55c7a21a0b45758a39bb

Parameter	Type	Mandatory or Optional	Description
timeStamp	xsd:string	O	Time stamp (UTC time). The value is required during MD5 encryption for **spPassword**. **NOTE** If the **spPassword** parameter must be set, this parameter is mandatory. [Format] yyyyMMddHHmmss [Example] 20100731064245
serviceId	xsd:string	O	Service ID. To obtain the service ID: • For a service partner, use the registered account to log in to the SDP Management Portal to get the ID. • For an API partner, set it blank. • For a developer: – If the developer has bought the capability, set the value to none. – If the developer has released the capability service, contact the carrier to obtain the service ID. The **serviceId** must be contained during invocation of a service interface developed by service partners and other partners, and must not be contained during invocation of a capability interface developed by API partners, other partners, and developers. [Example] 35000001000001

Parameter	Type	Mandatory or Optional	Description
bundleID	xsd:string	O	Bundle ID. When SDP creates a capability bundle, SDP allocates a bundleID to capability bundle. The **bundleID** must not be contained during invocation of a service interface developed by service partners and other partners, and must be contained during invocation of a capability interface developed by API partners, other partners, and developers. **NOTE** • When an API partner invokes the **getCreditExpiryDate** interface, this parameter is mandatory. • The **bundleID** is allocated only after an API product bundle is released. Therefore, ensure that the corresponding product bundle has been released on the SDP management portal before using the interface. [Example] 256000039
OA	xsd:string	M	Mobile number or the fake ID of the service originator. [Format] • Mobile number: prefix+country code+mobile number **NOTE** • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639

Parameter	Type	Mandatory or Optional	Description
FA	xsd:string	O	Mobile number or the fake ID of the charged party. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639

getCreditExpiryDateRequest

Table 4-2 describes the input parameters in the **getCreditExpiryDateRequest** message.

Table 4-2 getCreditExpiryDateRequest input parameters

Parameter	Type	Mandatory or Optional	Description
endUserIdentifier	xsd:anyURI	M	This parameter identifies the end user's account. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639
endUserPin	xsd:string	O	Contains the end user's credentials for authorizing access to the account. [Example] 1212

getCreditExpiryDateResponse

Table 4-3 describes the parameters in the **getBalanceResponse** message.

Table 4-3 getCreditExpiryDateResponse parameters

Parameter	Type	Description
result	BalanceExpireDetails [1…unbounded]	It is a BalanceExpireDetails array that consists of types with a string and a date field. i.e. Balance type and date balance will expire respectively. For the error codes, see Table 4-4.

Table 4-4 BalanceExpireDetails parameters

Parameter	Type	Description
balanceType	xsd:string	Identifies the type of balance. End user accounts may have one or more balances for different types of usage (e.g. Voice, SMS, gaming etc)
date	xsd:dateTime	It is the date the identified balance will expire. Do not specify if the balance does not expire

Error Code

Table 4-5 describes the error codes that the SDP may return.

Table 4-5 getCreditExpiryDate error codes

Error Code	Description	Cause
SVC0001	Service error.	A service exception occurred in the internal SDP.
	A service error occurred.	An internal SDP service is abnormal.
SVC0901	SPID is null!	The **spId** value in the request header is blank or does not exist in the SDP.
	SPID %1 is invalid!	The **spId** value in the request header is in an incorrect format.
	Sp password is null!	The **password** value in the request header is blank.
	Service ID is null!	The **serviceId** value in the request header is blank or does not exist in the SDP.
	Service ID %1 is invalid!	The **serviceId** value in the request header is in an incorrect format.

Error Code	Description	Cause
	Timestamp is empty in soapheader.	The **timeStamp** value in the request header is blank.
	SP ip is null!	The IP address in the request header is blank.
	The Created is null.	The **timeStamp** value in the request header is blank.
	The Nonce is null.	The **Nonce** value in the request header is blank.
	Sp password is null!	The **password** value in the request header is blank.
	The PasswordDigest is not accepted.	The password in the cipher text is incorrect.
	Sp ip %1 is not accepted!	The IP address in the request header is incorrect.
	Sp password is not accepted!	The **password** value in the request header is incorrect.
	The Sp has not logged in.	The **Token** value in the request header does not exist in the SDP and the Partner needs to log in first.
	The token is not accepted!	The **Token** format in the request header is incorrect.
	The token is expired!	The **Token** value in the request header is expired.
	local SP password is null!	An exception occurred in the internal SDP.
	The sp's Status is deregistered.	The Partner is in the deregistered state.
	The sp's Status is pre-deregistered.	The Partner is in the pre-deregistered state.
	SP %1 is in blacklist!	The Partner account is in the blacklist.
	The sp's Status is unknown.	An internal SDP service is abnormal.
	SPID %1 is not exist!	The Partner specified by **spId** in the request header does not exist in the SDP.
	The service status is configuring.	The service specified by **serviceId** in the request header is in the configuring state.
	The service status is suspended.	The service specified by **serviceId** in the request header is in the paused state.

Error Code	Description	Cause
	The service status is pre-deregistered.	The service specified by **serviceId** in the request header is in the pre-deregistered state.
	The service status is deregistered.	The service specified by **serviceId** in the request header is in the deregistered state.
	Service ID %1 is not exist!	The **serviceID** value in the request header does not exist in the SDP.
	The service status is unknown.	An internal SDP service is abnormal.
	Service %1 is in blacklist!	The **serviceId** value in the request header in the blacklist.
	The API %1 is not existed.	This Partner does not have the permission for using the API.
	The API status is disabled.	This Partner does not have the permission for using the API.
	SP status is locked.	The Partner is in the locked state.
	The sp's Status is forbidden.	The Partner is in the forbidden state.
	The sp 's status is pause.	The Partner is in the paused state.
SVC0905	AuthToken %1 is invalid	The **AuthToken** value in the request header sent by the partner is invalid.

Example

Examples of getCreditExpiryDate messages in typical scenarios are as follows:

- When a service partner sends the request message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc="http://www.csapi.org/schema/parlayx/account_management/v3_1/local">
   <soapenv:Header>
      <tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com.cn/schema/common/v2_1">
         <tns:spId>011104</tns:spId>
         <tns:spPassword>557d5fd8fb01089acd744245fc011036</tns:spPassword>
         <tns:timeStamp>20120809114701</tns:timeStamp>
         <tns:serviceId>35000001000119</tns:serviceId>
         <tns:OA>8613900000001</tns:OA>
         <tns:FA>8613900000001</tns:FA>
      </tns:RequestSOAPHeader>
   </soapenv:Header>
   <soapenv:Body>
      <loc:getCreditExpiryDate>
         <loc:endUserIdentifier>8613812345678</loc:endUserIdentifier>
```

```
        <loc:endUserPin>1212</loc:endUserPin>
      </loc:getCreditExpiryDate>
    </soapenv:Body>
</soapenv:Envelope>
```

- When an API partner sends the request message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc="http://www.csapi.org/schema/parlayx/account_management/v3_1/local">
   <soapenv:Header>
      <tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com.cn/schema/common/v2_1">
         <tns:spId>011104</tns:spId>
         <tns:spPassword>557d5fd8fb01089acd744245fc011036</tns:spPassword>
         <tns:timeStamp>20120809114701</tns:timeStamp>
         <tns:OA>8613900000001</tns:OA>
         <tns:FA>8613900000001</tns:FA>
      </tns:RequestSOAPHeader>
   </soapenv:Header>
   <soapenv:Body>
      <loc:getCreditExpiryDate>
         <loc:endUserIdentifier>8613812345678</loc:endUserIdentifier>
         <loc:endUserPin>1212</loc:endUserPin>
      </loc:getCreditExpiryDate>
   </soapenv:Body>
</soapenv:Envelope>
```

- The following is an example of the response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <ns1:getCreditExpiryDateResponse
xmlns:ns1="http://www.csapi.org/schema/parlayx/account_management/v2_2/local">
         <ns1:result>
            <balanceType>SMS</balanceType>
            <date>20130101T02:29:03+0000</date>
         </ns1:result>
      </ns1:getCreditExpiryDateResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

05 Recharging using direct payment

5.1 balanceUpdate

The SDP functions as the server and provides the interface for partners. Partners use it to recharge using direct payment.

Request URI

http://*IP*:*Port*/AccountManagementService/services/AccountManagement/v3

In the preceding information, *IP* and *Port* indicate the IP address and port number for the SDP to provide Account Management services. Contact the carrier to obtain the IP address and port number.

Message Header

Table 5-1 describes the parameters in the message header of the balanceUpdate interface.

Table 5-1 balanceUpdate message header parameters

Parameter	Type	Mandatory or Optional	Description
spId	xsd:string	M	Partner ID. Partner ID automatically allocated by the SDP to the Partner after successful registration. To obtain the Partner ID: • For a partner, set the authentication mode and password during registration. If the password is forgotten, contact the carrier to retrieve the password. • For a partner, use the registered account to log in to the SDP Management Portal, choose **Home** > **Partner** > **Access Code**, and query the Partner ID. • For a developer, use the registered account to log in to the Developer Portal, choose **Home** > **My Account**, and query the Developer ID. [Example] 2500110000021

Parameter	Type	Mandatory or Optional	Description
spPassword	xsd:string	O	Authentication key for the SDP to authenticate partners. The SDP supports authentication by **SP ID + Password**, **SP ID + IP address + Password**, or **SP ID + IP address**. Partners select an authentication mode during registration. If a partner selects authentication by **SP ID + Password** or **SP ID + IP address + Password**, this parameter is mandatory in requests sent by this partner. The value is a character string encrypted. The encryption formula is as follows: • SHA-256: spPassword = Base64(SHA-256(spId + Password + timeStamp)) • MD5: spPassword = MD5(spId + Password + timeStamp) In the formula: • **spId** and **timeStamp**: authentication ID and timestamp. • **Password**: password for partners to access the SDP. – For a partner, set the authentication mode and password during registration. If the password is forgotten, use the registered account to log in to the SDP Management Portal, choose **Home** > **Partner** > **Partner Information**, and set a new one. – For a developer, set the authentication mode and password during registration. If the password is forgotten, use the registered account to log in to the Developer Portal, choose **Member Center** > **Account** > **Registration Information** > **Invoke Password**, and set a new one. **NOTE** To retain features of earlier versions, the SP uses the MD5 algorithm in the connection to the SDP, which might cause security risks. [Example] e6434ef249df55c7a21a0b45758a39bb

Parameter	Type	Mandatory or Optional	Description
timeStamp	xsd:string	O	Time stamp (UTC time). The value is required during MD5 encryption for **spPassword**. **NOTE** If the **spPassword** parameter must be set, this parameter is mandatory. [Format] yyyyMMddHHmmss [Example] 20100731064245
serviceId	xsd:string	O	Service ID. To obtain the service ID: • For a service partner, use the registered account to log in to the SDP Management Portal to get the ID. • For an API partner, set it blank. • For a developer: – If the developer has bought the capability, set the value to none. – If the developer has released the capability service, contact the carrier to obtain the service ID. The **bundleID** must not be contained during invocation of a service interface developed by service partners and other partners, and must be contained during invocation of a capability interface developed by API partners, other partners, and developers. [Example] 35000001000001

Parameter	Type	Mandatory or Optional	Description
OA	xsd:string	M	Mobile number or the fake ID of the service originator. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 Fake ID: f-245-11900000007639
FA	xsd:string	O	Mobile number or the fake ID of the charged party. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639
namedParameters	NamedParameterList	O	Extended information. For the **NamedParameterList** information ,see Table 5-2.

Table 5-2 NamedParameterList parameters

Parameter	Type	Length	Level of Requirement	Description
item	NamedParameter[0...unbounded]	-	Optional	Data type corresponding to the extended field, which consists of **key** and **value**. For the **NamedParameters** information, see Table 5-3.

Table 5-3 NamedParameter parameters

Parameter	Type	Length	Level of Requirement	Description
key	String	-	Mandatory	ID of the extended field. The value is specified by the SDP. [Example] try
value	String	-	Mandatory	Value of the key. [Example] true

Table 5-4 Extended parameters transmitted in extensionInfo

Key			Value	
Parameter	**Type**	**Length**	**Level of Requirement**	**Description**
validateSubscriberLocation	int	4	Conditional	The parameter is used in a refill request to indicate if a validation of the subscriber location should be performed or not. ● 0 (false) (default value) Do not perform validation ● 1 (true) Perform validation [Example] 0

balanceUpdateRequest

Table 5-5 describes the input parameters in the **balanceUpdateRequest** message.

Table 5-5 balanceUpdateRequest input parameters

Parameter	Type	Mandatory or Optional	Description
endUserIdentifier	xsd:anyURI	M	This parameter identifies the end user's account. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639
endUserPin	xsd:string	O	Contains the end user's credentials for authorizing access to the account. [Example] 1212
referenceCode	xsd:string	M	Textual information to uniquely identify the request, e.g. in case of disputes.
balanceType	xsd:string	M	Identifies the type of balance. End user accounts may have one or more balances for different types of usage (e.g. Voice, SMS, gaming etc).
amount	xsd:decimal	M	Currency amount that should be added to the end user's account.
period	xsd:int	O	The balance is requested to expire in the number of days indicated by this parameter. The operator's policies may overrule this parameter. NOTE If this optional parameter is not present, the operator's policy on balance expiration is always in effect.

balanceUpdateResponse

None.

Error Code

Table 5-6 describes the error codes that the SDP may return.

Table 5-6 balanceUpdate error codes

Error Code	Description	Cause
SVC0001	Service error.	A service exception occurred in the internal SDP.
	A service error occurred.	An internal SDP service is abnormal.
SVC0901	SPID is null!	The **spId** value in the request header is blank or does not exist in the SDP.
	SPID %1 is invalid!	The **spId** value in the request header is in an incorrect format.
	Sp password is null!	The **password** value in the request header is blank.
	Service ID is null!	The **serviceId** value in the request header is blank or does not exist in the SDP.
	Service ID %1 is invalid!	The **serviceId** value in the request header is in an incorrect format.
	Timestamp is empty in soapheader.	The **timeStamp** value in the request header is blank.
	SP ip is null!	The IP address in the request header is blank.
	The Created is null.	The **timeStamp** value in the request header is blank.
	The Nonce is null.	The **Nonce** value in the request header is blank.
	Sp password is null!	The **password** value in the request header is blank.
	The PasswordDigest is not accepted.	The password in the cipher text is incorrect.
	Sp ip %1 is not accepted!	The IP address in the request header is incorrect.
	Sp password is not accepted!	The **password** value in the request header is incorrect.
	The Sp has not logged in.	The **Token** value in the request header does not exist in the SDP and the Partner needs to log in first.

Error Code	Description	Cause
	The token is not accepted!	The **Token** format in the request header is incorrect.
	The token is expired!	The **Token** value in the request header is expired.
	local SP password is null!	An exception occurred in the internal SDP.
	The sp's Status is deregistered.	The Partner is in the deregistered state.
	The sp's Status is pre-deregistered.	The Partner is in the pre-deregistered state.
	SP %1 is in blacklist!	The Partner account is in the blacklist.
	The sp's Status is unknown.	An internal SDP service is abnormal.
	SPID %1 is not exist!	The Partner specified by **spId** in the request header does not exist in the SDP.
	The service status is configuring.	The service specified by **serviceId** in the request header is in the configuring state.
	The service status is suspended.	The service specified by **serviceId** in the request header is in the paused state.
	The service status is pre-deregistered.	The service specified by **serviceId** in the request header is in the pre-deregistered state.
	The service status is deregistered.	The service specified by **serviceId** in the request header is in the deregistered state.
	Service ID %1 is not exist!	The **serviceID** value in the request header does not exist in the SDP.
	The service status is unknown.	An internal SDP service is abnormal.
	Service %1 is in blacklist!	The **serviceId** value in the request header in the blacklist.
	The API %1 is not existed.	This Partner does not have the permission for using the API.
	The API status is disabled.	This Partner does not have the permission for using the API.
	SP status is locked.	The Partner is in the locked state.
	The sp's Status is forbidden.	The Partner is in the forbidden state.
	The sp 's status is pause.	The Partner is in the paused state.

Error Code	Description	Cause
SVC0905	AuthToken %1 is invalid	The **AuthToken** value in the request header sent by the partner is invalid.

Example

Examples of balanceUpdate messages in typical scenarios are as follows:

- The following is an example of request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc="http://www.csapi.org/schema/parlayx/account_management/v3_1/local">
   <soapenv:Header>
      <tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com.cn/schema/common/v2_1">
         <tns:spId>011104</tns:spId>
         <tns:spPassword>557d5fd8fb01089acd744245fc011036</tns:spPassword>
         <tns:timeStamp>20120809114701</tns:timeStamp>
         <tns:serviceId>35000001000119</tns:serviceId>
         <tns:OA>8613900000001</tns:OA>
         <tns:FA>8613900000001</tns:FA>
         <tns:namedParameters>
            <tns:item>
               <tns:key>validateSubscriberLocation</v2:key>
               <tns:value>1</v2:value>
            </tns:item>
         </tns:namedParameters>
      </tns:RequestSOAPHeader>
   </soapenv:Header>
   <soapenv:Body>
      <loc:balanceUpdate>
        <loc:endUserIdentifier>8613812345678</loc:endUserIdentifier>
         <loc:endUserPin>1212</loc:endUserPin>
         <loc:referenceCode>121</loc:referenceCode>
         <loc:balanceType>SMS</loc:balanceType>
         <loc:amount>60</loc:amount>
         <loc:period>10</loc:period>
      </loc:balanceUpdate>
   </soapenv:Body>
</soapenv:Envelope>
```

- The following is an example of the response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <ns1:balanceUpdateResponse
xmlns:ns1="http://www.csapi.org/schema/parlayx/account_management/v3_1/local"/>
   </soapenv:Body>
</soapenv:Envelope>
```

5 Recharging using direct payment

06

Recharging using a voucher

6.1 voucherUpdate

The SDP functions as the server and provides the interface for partners. Partners use it to recharge using a voucher.

Request URI

http://*IP*:*Port*/AccountManagementService/services/AccountManagement/v3

In the preceding information, *IP* and *Port* indicate the IP address and port number for the SDP to provide Account Management services. Contact the carrier to obtain the IP address and port number.

Message Header

Table 6-1 describes the parameters in the message header of the voucherUpdate interface.

Table 6-1 voucherUpdate message header parameters

Parameter	Type	Mandatory or Optional	Description
spId	xsd:string	M	Partner ID. Partner ID automatically allocated by the SDP to the Partner after successful registration. To obtain the Partner ID: • For a partner, set the authentication mode and password during registration. If the password is forgotten, contact the carrier to retrieve the password. • For a partner, use the registered account to log in to the SDP Management Portal, choose **Home** > **Partner** > **Access Code**, and query the Partner ID. • For a developer, use the registered account to log in to the Developer Portal, choose **Home** > **My Account**, and query the Developer ID. [Example] 2500110000021

Parameter	Type	Mandatory or Optional	Description
spPassword	xsd:string	O	Authentication key for the SDP to authenticate partners. The SDP supports authentication by **SP ID + Password**, **SP ID + IP address + Password**, or **SP ID + IP address**. Partners select an authentication mode during registration. If a partner selects authentication by **SP ID + Password** or **SP ID + IP address + Password**, this parameter is mandatory in requests sent by this partner. The value is a character string encrypted. The encryption formula is as follows: • SHA-256: spPassword = Base64(SHA-256(spId + Password + timeStamp)) • MD5: spPassword = MD5(spId + Password + timeStamp) In the formula: • **spId** and **timeStamp**: authentication ID and timestamp. • **Password**: password for partners to access the SDP. – For a partner, set the authentication mode and password during registration. If the password is forgotten, use the registered account to log in to the SDP Management Portal, choose **Home** > **Partner** > **Partner Information**, and set a new one. – For a developer, set the authentication mode and password during registration. If the password is forgotten, use the registered account to log in to the Developer Portal, choose **Member Center** > **Account** > **Registration Information** > **Invoke Password**, and set a new one. **NOTE** To retain features of earlier versions, the SP uses the MD5 algorithm in the connection to the SDP, which might cause security risks. [Example] e6434ef249df55c7a21a0b45758a39bb

Parameter	Type	Mandatory or Optional	Description
timeStamp	xsd:string	O	Time stamp (UTC time). The value is required during MD5 encryption for **spPassword**. **NOTE** If the **spPassword** parameter must be set, this parameter is mandatory. [Format] yyyyMMddHHmmss [Example] 20100731064245
serviceId	xsd:string	O	Service ID. To obtain the service ID: • For a service partner, use the registered account to log in to the SDP Management Portal to get the ID. • For an API partner, set it blank. • For a developer: – If the developer has bought the capability, set the value to none. – If the developer has released the capability service, contact the carrier to obtain the service ID. The **bundleID** must not be contained during invocation of a service interface developed by service partners and other partners, and must be contained during invocation of a capability interface developed by API partners, other partners, and developers. [Example] 35000001000001

Parameter	Type	Mandatory or Optional	Description
OA	xsd:string	M	Mobile number or the fake ID of the service originator. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639
FA	xsd:string	O	Mobile number or the fake ID of the charged party. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639

voucherUpdateRequest

Table 6-2 describes the input parameters in the **voucherUpdateRequest** message.

Table 6-2 voucherUpdateRequest input parameters

Parameter	Type	Mandatory or Optional	Description
endUserIdentifier	xsd:anyURI	M	This parameter identifies the end user's account. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639
endUserPin	xsd:string	O	Contains the end user's credentials for authorizing access to the account. [Example] 1212
referenceCode	xsd:string	M	Textual information to uniquely identify the request, e.g. in case of disputes.
voucherIdentifier	xsd:string	M	This parameter identifies the voucher.
voucherPin	xsd:decimal	O	Contains the voucher's credentials for authentication.

voucherUpdateResponse

None.

Error Code

Table 6-3 describes the error codes that the SDP may return.

Table 6-3 getBalance error codes

Error Code	Description	Cause
SVC0001	Service error.	A service exception occurred in the internal SDP.
	A service error occurred.	An internal SDP service is abnormal.

Error Code	Description	Cause
SVC0901	SPID is null!	The **spId** value in the request header is blank or does not exist in the SDP.
	SPID %1 is invalid!	The **spId** value in the request header is in an incorrect format.
	Sp password is null!	The **password** value in the request header is blank.
	Service ID is null!	The **serviceId** value in the request header is blank or does not exist in the SDP.
	Service ID %1 is invalid!	The **serviceId** value in the request header is in an incorrect format.
	Timestamp is empty in soapheader.	The **timeStamp** value in the request header is blank.
	SP ip is null!	The IP address in the request header is blank.
	The Created is null.	The **timeStamp** value in the request header is blank.
	The Nonce is null.	The **Nonce** value in the request header is blank.
	Sp password is null!	The **password** value in the request header is blank.
	The PasswordDigest is not accepted.	The password in the cipher text is incorrect.
	Sp ip %1 is not accepted!	The IP address in the request header is incorrect.
	Sp password is not accepted!	The **password** value in the request header is incorrect.
	The Sp has not logged in.	The **Token** value in the request header does not exist in the SDP and the Partner needs to log in first.
	The token is not accepted!	The **Token** format in the request header is incorrect.
	The token is expired!	The **Token** value in the request header is expired.
	local SP password is null!	An exception occurred in the internal SDP.
	The sp's Status is deregistered.	The Partner is in the deregistered state.
	The sp's Status is pre-deregistered.	The Partner is in the pre-deregistered state.

Error Code	Description	Cause
	SP %1 is in blacklist!	The Partner account is in the blacklist.
	The sp's Status is unknown.	An internal SDP service is abnormal.
	SPID %1 is not exist!	The Partner specified by **spId** in the request header does not exist in the SDP.
	The service status is configuring.	The service specified by **serviceId** in the request header is in the configuring state.
	The service status is suspended.	The service specified by **serviceId** in the request header is in the paused state.
	The service status is pre-deregistered.	The service specified by **serviceId** in the request header is in the pre-deregistered state.
	The service status is deregistered.	The service specified by **serviceId** in the request header is in the deregistered state.
	Service ID %1 is not exist!	The **serviceID** value in the request header does not exist in the SDP.
	The service status is unknown.	An internal SDP service is abnormal.
	Service %1 is in blacklist!	The **serviceId** value in the request header in the blacklist.
	The API %1 is not existed.	This Partner does not have the permission for using the API.
	The API status is disabled.	This Partner does not have the permission for using the API.
	SP status is locked.	The Partner is in the locked state.
	The sp's Status is forbidden.	The Partner is in the forbidden state.
	The sp 's status is pause.	The Partner is in the paused state.
SVC0905	AuthToken %1 is invalid	The **AuthToken** value in the request header sent by the partner is invalid.

Example

Examples of voucherUpdate messages in typical scenarios are as follows:

- The following is an example of request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc="http://www.csapi.org/schema/parlayx/account_management/v3_1/local">
  <soapenv:Header>
```

```
      <tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com.cn/schema/common/v2_1">
        <tns:spId>011104</tns:spId>
        <tns:spPassword>557d5fd8fb01089acd744245fc011036</tns:spPassword>
        <tns:timeStamp>20120809114701</tns:timeStamp>
        <tns:serviceId>35000001000119</tns:serviceId>
        <tns:OA>35713111113</tns:OA>
        <tns:FA>35713111113</tns:FA>
      </tns:RequestSOAPHeader>
    </soapenv:Header>
    <soapenv:Body>
      <loc:voucherUpdate>
         <loc:endUserIdentifier>35713111113</loc:endUserIdentifier>
        <loc:endUserPin>1212</loc:endUserPin>
        <loc:referenceCode>131</loc:referenceCode>
        <loc:voucherIdentifier>141</loc:voucherIdentifier>
        <loc:voucherPin>11</loc:voucherPin>
      </loc:voucherUpdate>
    </soapenv:Body>
</soapenv:Envelope>
```

- The following is an example of the response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <ns1:voucherUpdateResponse
xmlns:ns1="http://www.csapi.org/schema/parlayx/account_management/v3_1/local"/>
   </soapenv:Body>
</soapenv:Envelope>
```

07

Querying Users' Operation Records

7.1 getHistory

The SDP functions as the server and provides the interface for partners. Partners use it to query the history records of a subscriber's account.

Request URI

http://*IP*:*Port*/AccountManagementService/services/AccountManagement/v3

In the preceding information, *IP* and *Port* indicate the IP address and port number for the SDP to provide Account Management services. Contact the carrier to obtain the IP address and port number.

Message Header

Table 7-1 describes the parameters in the message header of the getHistory interface.

Table 7-1 getHistory message header parameters

Parameter	Type	Mandatory or Optional	Description
spId	xsd:string	M	Partner ID. Partner ID automatically allocated by the SDP to the Partner after successful registration. To obtain the Partner ID: • For a partner, set the authentication mode and password during registration. If the password is forgotten, contact the carrier to retrieve the password. • For a partner, use the registered account to log in to the SDP Management Portal, choose **Home** > **Partner** > **Access Code**, and query the Partner ID. • For a developer, use the registered account to log in to the Developer Portal, choose **Home** > **My Account**, and query the Developer ID. [Example] 2500110000021

Parameter	Type	Mandatory or Optional	Description
spPassword	xsd:string	O	Authentication key for the SDP to authenticate partners. The SDP supports authentication by **SP ID + Password**, **SP ID + IP address + Password**, or **SP ID + IP address**. Partners select an authentication mode during registration. If a partner selects authentication by **SP ID + Password** or **SP ID + IP address + Password**, this parameter is mandatory in requests sent by this partner. The value is a character string encrypted. The encryption formula is as follows: • SHA-256: spPassword = Base64(SHA-256(spId + Password + timeStamp)) • MD5: spPassword = MD5(spId + Password + timeStamp) In the formula: • **spId** and **timeStamp**: authentication ID and timestamp. • **Password**: password for partners to access the SDP. – For a partner, set the authentication mode and password during registration. If the password is forgotten, use the registered account to log in to the SDP Management Portal, choose **Home** > **Partner** > **Partner Information**, and set a new one. – For a developer, set the authentication mode and password during registration. If the password is forgotten, use the registered account to log in to the Developer Portal, choose **Member Center** > **Account** > **Registration Information** > **Invoke Password**, and set a new one. **NOTE** To retain features of earlier versions, the SP uses the MD5 algorithm in the connection to the SDP, which might cause security risks. [Example] e6434ef249df55c7a21a0b45758a39bb

Parameter	Type	Mandatory or Optional	Description
timeStamp	xsd:string	O	Time stamp (UTC time). The value is required during MD5 encryption for **spPassword**. **NOTE** If the **spPassword** parameter must be set, this parameter is mandatory. [Format] yyyyMMddHHmmss [Example] 20100731064245
serviceId	xsd:string	O	Service ID. To obtain the service ID: • For a service partner, use the registered account to log in to the SDP Management Portal to get the ID. • For an API partner, set it blank. • For a developer: – If the developer has bought the capability, set the value to none. – If the developer has released the capability service, contact the carrier to obtain the service ID. The **bundleID** must not be contained during invocation of a service interface developed by service partners and other partners, and must be contained during invocation of a capability interface developed by API partners, other partners, and developers. [Example] 35000001000001

Parameter	Type	Mandatory or Optional	Description
OA	xsd:string	M	Mobile number or the fake ID of the service originator. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639
FA	xsd:string	O	Mobile number or the fake ID of the charged party. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639

getHistoryRequest

Table 7-2 describes the input parameters in the **getHistoryRequest** message.

Table 7-2 getHistoryRequest input parameters

Parameter	Type	Mandatory or Optional	Description
endUserIdentifier	xsd:anyURI	M	This parameter identifies the end user's account. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence

Parameter	Type	Mandatory or Optional	Description
			[Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639
endUserPin	xsd:string	O	Contains the end user's credentials for authorizing access to the account. [Example] 1212
date	xsd:dateTime	O	This parameter indicates the desired starting date for the entries to be returned. [Format] [-]*YYYY-MM-DDThh*:*mm*:*ss*.*ms*[Z|(+|-)*hh*:*mm*], in which, **-** is optional, *YYYY* indicates the year, *MM* indicates the month, *DD* indicates the day, *hh* indicates the hour, *mm* indicates the minute, *ss* indicates the second, and *ms* indicates millisecond (1 ms = 0.001 s). The following time formats are supported: • Local time The value **2012-01-01T12:12:12.001** indicates that the local time is 12 minutes, 12 seconds, and 1 millisecond past twelve on Jan. 1, 2012. • Adjusted time The value **2012-01-01T12:12:12.001+02:00** indicates that the UTC time is the local time minus 2. That is, the UTC time is **2012-01-01T10:12:12.001Z**. • UTC time The value **2012-01-01T10:12:12.001Z** indicates that the UTC time is 12 minutes, 12 seconds, and 1 millisecond past ten on Jan. 1, 2012. **NOTE** If this parameter is not present, it is up to the discretion of the service to decide this date.
maxEntries	xsd:int	O	This parameter indicates the maximum number of entries that shall be returned. **NOTE** If this parameter is not present, it is up to the discretion of the service to decide how many entries to return.

getHistoryResponse

Table 7-3 describes the parameters in the **getHistoryResponse** message.

Table 7-3 getHistoryResponse parameters

Parameter	Type	Description
result	DatedTransaction [0 .. unbounded]	It is a DatedTransaction array that consists of types with a date field and a string field, which are the date of the occurrence and the transaction details respectively. For the error codes, see Table 7-4.

Table 7-4 DatedTransaction parameters

Parameter	Type	Description
transactionDate	xsd:dateTime	The date the transaction occurred.
transactionDetails	xsd:string	The transaction details.

Error Code

Table 7-5 describes the error codes that the SDP may return.

Table 7-5 getHistory error codes

Error Code	Description	Cause
SVC0001	Service error.	A service exception occurred in the internal SDP.
	A service error occurred.	An internal SDP service is abnormal.
SVC0901	SPID is null!	The **spId** value in the request header is blank or does not exist in the SDP.
	SPID %1 is invalid!	The **spId** value in the request header is in an incorrect format.
	Sp password is null!	The **password** value in the request header is blank.
	Service ID is null!	The **serviceId** value in the request header is blank or does not exist in the SDP.
	Service ID %1 is invalid!	The **serviceId** value in the request header is in an incorrect format.
	Timestamp is empty in soapheader.	The **timeStamp** value in the request header is blank.

Error Code	Description	Cause
	SP ip is null!	The IP address in the request header is blank.
	The Created is null.	The **timeStamp** value in the request header is blank.
	The Nonce is null.	The **Nonce** value in the request header is blank.
	Sp password is null!	The **password** value in the request header is blank.
	The PasswordDigest is not accepted.	The password in the cipher text is incorrect.
	Sp ip %1 is not accepted!	The IP address in the request header is incorrect.
	Sp password is not accepted!	The **password** value in the request header is incorrect.
	The Sp has not logged in.	The **Token** value in the request header does not exist in the SDP and the Partner needs to log in first.
	The token is not accepted!	The **Token** format in the request header is incorrect.
	The token is expired!	The **Token** value in the request header is expired.
	local SP password is null!	An exception occurred in the internal SDP.
	The sp's Status is deregistered.	The Partner is in the deregistered state.
	The sp's Status is pre-deregistered.	The Partner is in the pre-deregistered state.
	SP %1 is in blacklist!	The Partner account is in the blacklist.
	The sp's Status is unknown.	An internal SDP service is abnormal.
	SPID %1 is not exist!	The Partner specified by **spId** in the request header does not exist in the SDP.
	The service status is configuring.	The service specified by **serviceId** in the request header is in the configuring state.
	The service status is suspended.	The service specified by **serviceId** in the request header is in the paused state.
	The service status is pre-deregistered.	The service specified by **serviceId** in the request header is in the pre-deregistered state.

Error Code	Description	Cause
	The service status is deregistered.	The service specified by **serviceId** in the request header is in the deregistered state.
	Service ID %1 is not exist!	The **serviceID** value in the request header does not exist in the SDP.
	The service status is unknown.	An internal SDP service is abnormal.
	Service %1 is in blacklist!	The **serviceId** value in the request header in the blacklist.
	The API %1 is not existed.	This Partner does not have the permission for using the API.
	The API status is disabled.	This Partner does not have the permission for using the API.
	SP status is locked.	The Partner is in the locked state.
	The sp's Status is forbidden.	The Partner is in the forbidden state.
	The sp 's status is pause.	The Partner is in the paused state.
SVC0905	AuthToken %1 is invalid	The **AuthToken** value in the request header sent by the partner is invalid.

Example

Examples of getHistory messages in typical scenarios are as follows:

- The following is an example of request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc="http://www.csapi.org/schema/parlayx/account_management/v3_1/local">
   <soapenv:Header>
     <tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com.cn/schema/common/v2_1">
        <tns:spId>011104</tns:spId>
        <tns:spPassword>557d5fd8fb01089acd744245fc011036</tns:spPassword>
        <tns:timeStamp>20120809114701</tns:timeStamp>
        <tns:serviceId>35000001000119</tns:serviceId>
        <tns:OA>35713111113</tns:OA>
        <tns:FA>35713111113</tns:FA>
     </tns:RequestSOAPHeader>
  </soapenv:Header>
  <soapenv:Body>
     <loc:getHistory>
        <loc:endUserIdentifier>35713111113</loc:endUserIdentifier>
        <loc:endUserPin>1212</loc:endUserPin>
        <loc:date>2012-01-01T12:12:12.001Z</loc:date>
```

```
        <loc:maxEntries>2</loc:maxEntries>
      </loc:getHistory>
    </soapenv:Body>
</soapenv:Envelope>
```

- The following is an example of the response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <ns1:getHistoryResponse
xmlns:ns1="http://www.csapi.org/schema/parlayx/account_management/v3_1/local">
         <ns1:result>
            <transactionDate>2012-06-06T12:12:12.001Z</transactionDate>
            <transactionDetails>ok</transactionDetails>
         </ns1:result>
         <ns1:result>
            <transactionDate>2012-06-08T12:12:12.001Z</transactionDate>
            <transactionDetails>ok</transactionDetails>
         </ns1:result>
      </ns1:getHistoryResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

09

Appendix

8.1 getBalanceTypes

The SDP functions as the server and provides the interface for partners. Partners use it to obtain users' balance types.

Request URI

http://*IP*:*Port*/AccountManagementService/services/AccountManagement/v3

In the preceding information, *IP* and *Port* indicate the IP address and port number for the SDP to provide Account Management services. Contact the carrier to obtain the IP address and port number.

Message Header

Table 8-1 describes the parameters in the message header of the getBalanceTypes interface.

Table 8-1 getBalanceTypes message header parameters

Parameter	Type	Mandatory or Optional	Description
spId	xsd:string	M	Partner ID. Partner ID automatically allocated by the SDP to the Partner after successful registration. To obtain the Partner ID: • For a partner, set the authentication mode and password during registration. If the password is forgotten, contact the carrier to retrieve the password. • For a partner, use the registered account to log in to the SDP Management Portal, choose **Home** > **Partner** > **Access Code**, and query the Partner ID. • For a developer, use the registered account to log in to the Developer Portal, choose **Home** > **My Account**, and query the Developer ID. [Example] 2500110000021

Parameter	Type	Mandatory or Optional	Description
spPassword	xsd:string	O	Authentication key for the SDP to authenticate partners. The SDP supports authentication by **SP ID + Password**, **SP ID + IP address + Password**, or **SP ID + IP address**. Partners select an authentication mode during registration. If a partner selects authentication by **SP ID + Password** or **SP ID + IP address + Password**, this parameter is mandatory in requests sent by this partner. The value is a character string encrypted. The encryption formula is as follows: • SHA-256: spPassword = Base64(SHA-256(spId + Password + timeStamp)) • MD5: spPassword = MD5(spId + Password + timeStamp) In the formula: • **spId** and **timeStamp**: authentication ID and timestamp. • **Password**: password for partners to access the SDP. – For a partner, set the authentication mode and password during registration. If the password is forgotten, use the registered account to log in to the SDP Management Portal, choose **Home** > **Partner** > **Partner Information**, and set a new one. – For a developer, set the authentication mode and password during registration. If the password is forgotten, use the registered account to log in to the Developer Portal, choose **Member Center** > **Account** > **Registration Information** > **Invoke Password**, and set a new one. **NOTE** To retain features of earlier versions, the SP uses the MD5 algorithm in the connection to the SDP, which might cause security risks. [Example] e6434ef249df55c7a21a0b45758a39bb

Parameter	Type	Mandatory or Optional	Description
timeStamp	xsd:string	O	Time stamp (UTC time). The value is required during MD5 encryption for **spPassword**. **NOTE** If the **spPassword** parameter must be set, this parameter is mandatory. [Format] yyyyMMddHHmmss [Example] 20100731064245
serviceId	xsd:string	O	Service ID. To obtain the service ID: • For a service partner, use the registered account to log in to the SDP Management Portal to get the ID. • For an API partner, set it blank. • For a developer: – If the developer has bought the capability, set the value to none. – If the developer has released the capability service, contact the carrier to obtain the service ID. The **bundleID** must not be contained during invocation of a service interface developed by service partners and other partners, and must be contained during invocation of a capability interface developed by API partners, other partners, and developers. [Example] 35000001000001

Parameter	Type	Mandatory or Optional	Description
OA	xsd:string	M	Mobile number or the fake ID of the service originator. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639
FA	xsd:string	O	Mobile number or the fake ID of the charged party. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**. • Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639

getBalanceTypesRequest

Table 8-2 describes the input parameters in the **getBalanceTypesRequest** message.

Table 8-2 getBalanceTypesRequest input parameters

Parameter	Type	Mandatory or Optional	Description
endUserIdentifier	xsd:anyURI	M	This parameter identifies the end user's account. [Format] • Mobile number: prefix+country code+mobile number NOTE • The **prefix** is optional. • The **prefix** can be any one of **+**, **+0**, **+00**, **0**, and **00**.

Parameter	Type	Mandatory or Optional	Description
			• Fake ID: Prefix-opcoid-sequence [Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639
endUserPin	xsd:string	O	Contains the end user's credentials for authorizing access to the account. [Example] 1212

getBalanceTypesResponse

Table 8-3 describes the parameters in the **getBalanceTypesResponse** message.

Table 8-3 getBalanceTypesResponse parameters

Parameter	Type	Description
result	xsd:string [1unbounded]	Identifies the types of balance. End user accounts may have one or more balances for different types of usage (e.g. Voice, SMS, gaming etc).

Error Code

Table 8-4 describes the error codes that the SDP may return.

Table 8-4 getBalance error codes

Error Code	Description	Cause
SVC0001	Service error.	A service exception occurred in the internal SDP.
	A service error occurred.	An internal SDP service is abnormal.
SVC0901	SPID is null!	The **spId** value in the request header is blank or does not exist in the SDP.
	SPID %1 is invalid!	The **spId** value in the request header is in an incorrect format.
	Sp password is null!	The **password** value in the request header is blank.
	Service ID is null!	The **serviceId** value in the request header is blank or does not exist in the SDP.

Error Code	Description	Cause
	Service ID %1 is invalid!	The **serviceId** value in the request header is in an incorrect format.
	Timestamp is empty in soapheader.	The **timeStamp** value in the request header is blank.
	SP ip is null!	The IP address in the request header is blank.
	The Created is null.	The **timeStamp** value in the request header is blank.
	The Nonce is null.	The **Nonce** value in the request header is blank.
	Sp password is null!	The **password** value in the request header is blank.
	The PasswordDigest is not accepted.	The password in the cipher text is incorrect.
	Sp ip %1 is not accepted!	The IP address in the request header is incorrect.
	Sp password is not accepted!	The **password** value in the request header is incorrect.
	The Sp has not logged in.	The **Token** value in the request header does not exist in the SDP and the Partner needs to log in first.
	The token is not accepted!	The **Token** format in the request header is incorrect.
	The token is expired!	The **Token** value in the request header is expired.
	local SP password is null!	An exception occurred in the internal SDP.
	The sp's Status is deregistered.	The Partner is in the deregistered state.
	The sp's Status is pre-deregistered.	The Partner is in the pre-deregistered state.
	SP %1 is in blacklist!	The Partner account is in the blacklist.
	The sp's Status is unknown.	An internal SDP service is abnormal.
	SPID %1 is not exist!	The Partner specified by **spId** in the request header does not exist in the SDP.
	The service status is configuring.	The service specified by **serviceId** in the request header is in the configuring state.

Error Code	Description	Cause
	The service status is suspended.	The service specified by **serviceId** in the request header is in the paused state.
	The service status is pre-deregistered.	The service specified by **serviceId** in the request header is in the pre-deregistered state.
	The service status is deregistered.	The service specified by **serviceId** in the request header is in the deregistered state.
	Service ID %1 is not exist!	The **serviceID** value in the request header does not exist in the SDP.
	The service status is unknown.	An internal SDP service is abnormal.
	Service %1 is in blacklist!	The **serviceId** value in the request header in the blacklist.
	The API %1 is not existed.	This Partner does not have the permission for using the API.
	The API status is disabled.	This Partner does not have the permission for using the API.
	SP status is locked.	The Partner is in the locked state.
	The sp's Status is forbidden.	The Partner is in the forbidden state.
	The sp 's status is pause.	The Partner is in the paused state.
SVC0905	AuthToken %1 is invalid	The **AuthToken** value in the request header sent by the partner is invalid.

Example

Examples of getBalanceTypes messages in typical scenarios are as follows:

- The following is an example of request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc="http://www.csapi.org/schema/parlayx/account_management/v3_1/local">
   <soapenv:Header>
      <tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com.cn/schema/common/v2_1">
         <tns:spId>011104</tns:spId>
         <tns:spPassword>557d5fd8fb01089acd744245fc011036</tns:spPassword>
         <tns:timeStamp>20120809114701</tns:timeStamp>
         <tns:serviceId>35000001000119</tns:serviceId>
         <tns:OA>35713111113</tns:OA>
         <tns:FA>35713111113</tns:FA>
      </tns:RequestSOAPHeader>
   </soapenv:Header>
```

```
  <soapenv:Body>
    <loc:getBalanceTypes>
      <loc:endUserIdentifier>35713111113</loc:endUserIdentifier>
      <loc:endUserPin>1212</loc:endUserPin>
    </loc:getBalanceTypes>
  </soapenv:Body>
</soapenv:Envelope>
```

- The following is an example of the response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getBalanceTypesResponse
xmlns:ns1="http://www.csapi.org/schema/parlayx/account_management/v3_1/local">
      <ns1:result>SMS</ns1:result>
      <ns1:result>GPRS</ns1:result>
      <ns1:result>Voice</ns1:result>
    </ns1:getBalanceTypesResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

09

Appendix

9.1 The Currency and Atomic Charging Unit of each OpCo country

Country Name	Currency	Atomic Charging Unit
Afghanistan	AFA	pul
Benin	XOF	N/A
Bissau	CFA	N/A
Cameroon	XAF	NA
Cote D'Ivoire	XOF	NA
Cyprus	Euro	cent
Ghana	GHS	Pesewa(1GHS=100Pesewa)
Guinea	GNF(Guinea Franc)	NA
Liberia	LRD(Liberia Dollar)	NA
N Sudan	SDG	Piaster
Nigeria	NGN	Kobo
Rwanda	Rwandan Franc (RWF)	RWF
SA	ZAR	N/A
Swaziland	SZL	cent
Uganda	Ugandan shilling (Code UGX)	UGX
Yemen	Yemeni Riyal	Fils (1 Riyal = 100 Fils)
Zambia	ZMK	ngwee, 1ZMK = 100Ngwee

9.2 ISO 4217 Codes for the representation of currencies and funds

ENTITY	CURRENCY	ALPHABETI CCODE	NUMERI CCODE
Afghanistan	Afghani	AFA	4
Benin	CFA Franc BCEAO	XOF	952
Botswana	Pula	BWP	72
Cameroon	CFA Franc BEAC	XAF	950
Congo	CFA Franc BEAC	XAF	950
Cote D'Ivoire	CFA Franc BCEAO	XOF	952
Cyprus	Cyprus Pound	CYP	196
Ethiopia	Ethiopian Birr	ETB	230
Ghana	Cedi	GHC	288
Guinea	Guinea Franc	GNF	324
Guinea-Bissau	Guinea-Bissau Peso	GWP	624
	CFA Franc BCEAO	XOF	952
Liberia	Liberian Dollar	LRD	430
Nigeria	Naira	NGN	566
Rwanda	Rwanda Franc	RWF	646
South Africa	Rand	ZAR	710
Sudan	Sudanese Dinar	SDD	736
Swaziland	Lilangeni	SZL	748
Uganda	Uganda Shilling	UGX	800
Yemen	Yemeni Rial	YER	886
Zambia	Kwacha	ZMK	894

9.3 OpCoID List

MTN OpCo Name	OpCoID Value
Afghanistan	9301
Benin	22901

9 Appendix

MTN OpCo Name	OpCoID Value
Bissau	24501
Botswana	26701
Cameroon	23701
Congo	24201
Cote D'Ivoire	22501
Cyprus	35701
Ghana	23301
Guinea Republic	22401
Liberia	23101
Nigeria	23401
Rwanda	25001
South Africa	2701
Sudan North	21101
Sudan South	24901
Swaziland	26801
Syria	96301
Uganda	25601
Yemen	96701
Zambia	26001


MTN