SDP Solution API Reference (Payment, ParlayX 3.0)

Change History

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

Issue 15 (2016-06-01)

This issue is the fifteenth official release, and includes the following changes:
Add the missing **<** character in the request example in chapter 3.1 and 4.1 for <currency>USD</currency>.

Issue 14 (2016-03-03)

This issue is the fourteenth official release, and includes the following changes:
Add new error code(SVC0272) for chargeAmount and refundAmount as per MTNSDP-DB-0262.
Add new error codes(SVC0273,SVC0274,SVC0275,SVC0276) for chargeAmount and refundAmount as per MTNSDP-DB-0262A.

Issue 13 (2015-11-20)

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

Issue 12 (2015-06-18)

This issue is the twelfth official release, and includes the following changes:
Add error code **SVC4001** for chargeAmount and refundAmount.
Add error code **SVC0906** for chargeAmount.
Update the description of **enduserDAAccountId** in chargeAmount/refundAmount message header.
Update the format description of **endUserIdentifier** in Table 3-2 and Table 4-2.
Update the description of **endUserDAAccountId** in Table 4-1.

Issue 11 (2015-01-12)

This issue is the eleventh official release, and includes the following changes:
Add the error code **SVC0271** for chargeAmount and refundAmount.
Optimize the description of **oauth_token** for chargeAmount.

Issue 10 (2014-12-05)

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

Issue 09 (2014-11-20)

This issue is the ninth official release, and includes the following changes:
Add the error code **SVC0007** for chargeAmount and refundAmount interfaces.

Issue 08 (2014-10-31)

This issue is the eighth official release, and includes the following changes:
Add the parameter **endUserDAAccountId** in Table 3-1 and Table 4-1.
Update the description about **bundleID**.

Issue 07 (2014-09-30)

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

Add the fake ID function in Phase2.3 version.

Change History

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 06 (2014-05-04)

This issue is the sixth official release, and includes the following changes:
Add the error codes for chargeAmount and refundAmount
Change the description of **serviceId** and **bundleID**.

Issue 05 (2014-01-24)

This issue is the fifth official release, and includes the following changes:
Add the oauth_token field in request message for chargeAmount.

Issue 04 (2013-11-19)

This issue is the fourth official release, and includes the following changes:
Add **OA** and **FA** fields in the request header of the interfaces.
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:
Updated chargeAmount and refundAmount for adding **bundleID** field.

Issue 01 (2013-06-22)

This issue is the first official release.

Contents

01

Overviews

1.1 Scenario

Payment APIs provided by the SDP are mainly used by the SP during Payment service development to implement charging and refund.

1.1.1 Deduction

The charging API provided by the SDP is used by the partner only for service development.

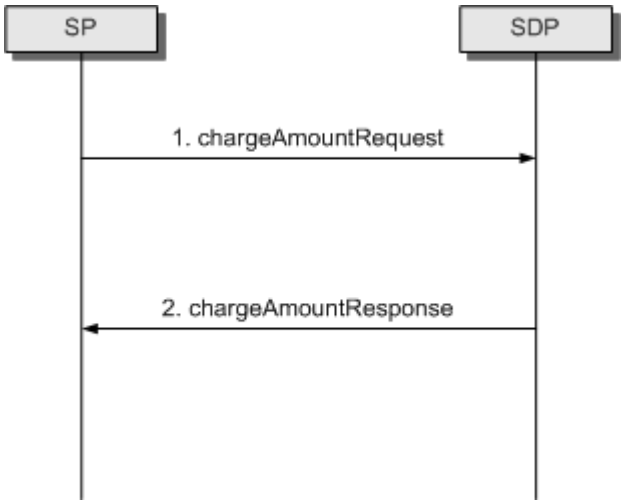
Introduction

When a user subscribes to a product provided by the partner on the partner portal, the partner can invoke the chargeAmount interface to deduct fees from the user's account.

Process

Figure 1-1 shows the process in which an partner uses the Payment capability provided by the SDP to deduct fees from users' accounts.

Figure 1-1 chargeAmount service process



The process is as follows:

- A partner uses the chargeAmount interface to send a request to the SDP.
- The SDP performs charging and sends a response to the partner.

1.1.2 Refund

The refund API provided by the SDP is used by the partner only for service development.

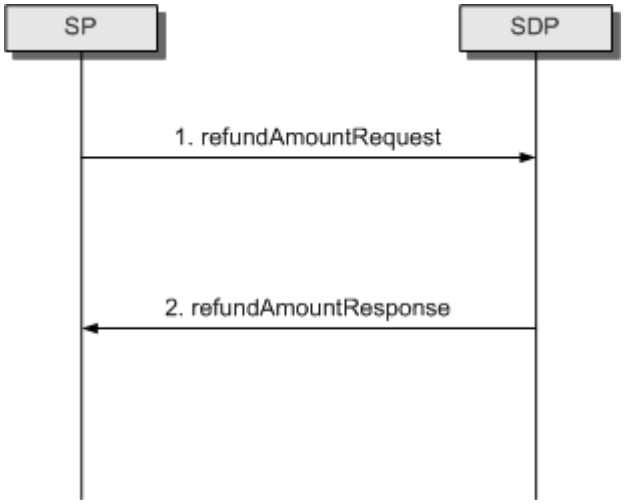
Introduction

The partner uses the refundAmount interface to refund a fee to the user.

Process

Figure 1-2 shows the process in which an partner uses the Payment capability provided by the SDP to refund fees to users.

Figure 1-2 refundAmount service process



The process is as follows:

- An partner uses the refundAmount interface to send a request to the SDP.
- The SDP performs refund and sends a response to the partner.

1.2 Interface List

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

Table 1-1 Payment Parlay X 3.0 interface list

Function	Interface	Description
Deduction	chargeAmount	This interface is provided by the SDP (when functioning as the server) for a partner to deduct fees from users' accounts.
Refund	refundAmount	This interface is provided by the SDP (when functioning as the server) for a partner to directly apply a refund to the account indicated by the end user identifier.

1.3 Level of Requirement for Parameters

The App must develop APIs based on the level of requirement for each parameter.

Table 1-2 Level of requirement for parameters

Type	Description
Mandatory	A parameter is always mandatory in a request. Parameters with the **Mandatory** requirement are used for access authentication or service processing. If a parameter with the **Mandatory** requirement is left empty in a request, access authentication or service processing fails and the request fails.
Conditional	A parameter is mandatory or optional in specified conditions. Parameters with the **Conditional** requirement are used for access authentication or service processing in specified conditions. If the specified conditions are met but a parameter with the **Conditional** requirement is left empty in a request, access authentication or service processing fails and the request fails.
Optional	A parameter is always optional. Parameters with the **Optional** requirement are not used for service processing.

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	Level of Requirement	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	Mandatory	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.

Parameter	Type	Level of Requirement	Description
			• **ABCD** is a number ranging from 0001 to 9999.
text	xsd:string	Mandatory	Error message description, which can contain the variables % and #.
variables	xsd:string [0..unbounded]	Optional	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	Level of Requirement	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	Mandatory	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	Mandatory	Error message description, which can contain the

Parameter	Type	Level of Requirement	Description
			variables % and #.
variables	xsd:string [0..unbounded]	Optional	Values of the variables used in the **text** value.

03

Deduction

3.1 chargeAmount

This interface is provided by the SDP (when functioning as the server) for a partner to deduct fees from users'accounts.

Request URI

http://*IP*:*Port*/AmountChargingService/services/AmountCharging/v3

In the preceding information, *IP* and *Port* indicate the IP address and port number for the SDP to provide 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 chargeAmount interface.

Table 3-1 chargeAmount message header parameters

Parameter	Type	Level of Requirement	Description
spId	xsd: string	Mandatory	Partner ID. Partner ID automatically allocated by the SDP to the partner after successful registration. For a partner, use the registered account to log in to the SDP management portal and query the account information. [Example] 000201

Parameter	Type	Level of Requirement	Description
spPassword	xsd: string	Conditional	Encrypted authentication password for partners to access the SDP. 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**: authentication password for partners to access the SDP. The value is set during registration with the SDP by a partner. **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
serviceId	xsd: string	Conditional	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] 3500001000012

Parameter	Type	Level of Requirement	Description
bundleID	xsd: string	Conditional	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** • In the capability sales scenario, if the API partner uses the Transaction Cost charging mode, the **bundleID** field does not need to be contained and the SDP sets **bundleID** to **–1**. If the API partner is a prepaid or postpaid user, the **bundleID** field must be contained. If the API partner does not send the **bundleID** field, the SDP uses the Transaction Cost charging mode by default and sets **bundleID** to **–1**. • 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
timeStamp	xsd: string	Conditional	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
OA	xsd: string	Conditional	Mobile number or the fake ID of the service originator. • When the Partner charges a user that subscribes to or orders a service by fee, this parameter value is the mobile number or the fake ID of the charged user. • When the Partner charges a user that receives a gift from another user by fee, this parameter value is the mobile number or the fake ID of the gift sender.

Parameter	Type	Level of Requirement	Description
			[Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639 **NOTE** When an API Partner sends the request, **OA** parameter is optional.
FA	xsd: string	Conditional	Mobile number or the fake ID of the charged party. The value must be the same as the OA value. **NOTE** When an API Partner sends the request, **FA** parameter is optional.
oauth_token	xsd: string	Conditional	Token authentication code synchronized by the MDSP to the SP if the SP has obtained the user consent. It must present in case of OTP or Flexible authorization scenario.
endUserDAAccountId	xsd: string	Optional	The specific DA account identifier. It indicates the subscriber DA Account. If it presents SDP will try to charge using DA Account if DA Account in mentioned. Otherwise SDP will charge from main account. There is no fallback to main balance if DA account charging is failed. Partner need to invoke Charge amount request again if necessary

chargeAmountRequest

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

Table 3-2 chargeAmount input parameters

Parameter	Type	Level of Requirement	Description
endUserIdentifier	xsd:anyURI	Mandatory	Account of the user to be charged. [Format] • Mobile number: tel:[Prefix][Country code][Mobile number] In the format, *[Prefix]* is optional. The value of *[Prefix]*, if contained, can be **+**, **+0**, **+00**, **0**, or **00**. • Fake ID: tel: [Prefix]- [opcoid]-[sequence].

Parameter	Type	Level of Requirement	Description
			[Example] • Mobile number: tel:8612312345678 • Fake ID: tel:f-245-11900000007639
charge	common:ChargingI informatio	Mandatory	Charging information. For the detailed parameters, see Table 3-3.
referenceCode	xsd:string	Mandatory	Unique ID of the request. [Example] 225

Table 3-3 ChargingInformatin parameters

Parameter	Type	Level of Requirement	Description
description	xsd:string	Mandatory	Charging description. [Example] charge
currency	xsd:string	Mandatory	Currency used for the charging. [Example] USD
amount	xsd:decimal	Conditional	Amount of the charged fee. In case of amount based charging, this parameter is mandatory. [Example] 100.00
code	xsd:string	Conditional	Charging code, which is relevant to the contract of the charged party. In case of code based charging, this parameter is mandatory. [Example] 4523

chargeAmountResponse

None.

Error Code

Table 3-4 describes the error codes that the SDP may return.

Table 3-4 Error codes of the chargeAmount interface

Error Code	Description	Cause
SVC0001	Get addr route info failed!the baseScfType = %1,the sourceAddr = %2,the destAddr = %3.	An internal SDP service is abnormal.
	Waiting for response timed out, message type is %1.	An internal SDP service is abnormal.
	The next NE is out of service, message type is %1.	An internal SDP service is abnormal.
SVC0002	ChargingInformation is null.	The **ChargingInformation** value in the request body is blank.
	Description is null in ChargingInformation.	The **Description** value in the request body is blank.
	Amount is null.	The **Amount** value in the request body is blank.
	Invalid input value.	A parameter value in the request is invalid.
SVC0007	Amount field should not be NULL.	The **Amount** value in the request body is blank.
	Code field should not be NULL.	The **Code** value in the request body is blank.
	Amount field should be NULL.	The **Amount** value in the request body is not blank.
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.
	SPID %1 is not exist!	The partner specified by **SPID** in the request header does not exist in the SDP.

Error Code	Description	Cause
	SP ip is null!	The IP address in the request header is blank.
	Sp ip %1 is not accepted!	The IP address in the request header is incorrect.
	Sp password is null!	The **password** value in the request header is blank.
	Timestamp is empty in soapheader.	The **timeStamp** value in the request header is blank.
	local SP password is null!	An internal error occurs in the SDP.
	Sp password is not accepted!	The **password** value in the request header is incorrect.
	The authentication type is unknown!	An internal SDP service is abnormal.
	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.
	SP %1 is in blacklist!	The partner account is in the blacklist.
	The sp's Status is pre-deregistered.	The partner is in the pre-deregistered state.
	The sp's Status is deregistered.	The partner is in the deregistered state.
	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.
	The sp's Status is unknown.	An internal SDP service is abnormal.
	The ScfType %1 is inactive!	An internal SDP service is abnormal.
	The ScfType %1 is uninstalled!	An internal SDP service is abnormal.
	Service ID %1 is invalid!	The **serviceID** value in the request header is incorrect.

Error Code	Description	Cause
	Service ID %1 is not existed!	The **serviceID** value in the request header does not exist in the SDP.
	Service %1 is in blacklist!	The **serviceId** value in the request header in the blacklist.
	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.
	The service status is unknown.	An internal SDP service is abnormal.
	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.
	The sp %1 has not ordered the service %2!	The partner has not subscribed to the Payment service.
	The service %1 has not orderd the api %2.	An internal SDP service is abnormal.
	The service %1 has not orderd the SCF %2.	An internal SDP service is abnormal.
	The sp %1 has not orderd the api %2 in current date.	This partner does not have the permission for using the API.
	The sp %1 has not orderd the SCF %2.	The partner has not subscribed to the Payment service.
SVC3101	Insufficient Balance	The user's account balance is insufficient.
POL002	Privacy verification failed	The API invoked by the SP requires user authorization. However, the

Error Code	Description	Cause
		current SP is not authorized by the user.
POL0904	SP level gross control not pass.	The number of requests delivered by the partner exceeds the signed TPS.
POL0908	User Consent Required	The API invoked by the SP requires user authorization. However, the authorization code contained in the SP request fails to be verified.
POL0910	Minimum Amount per Transaction.	The **Amount** value in the request is out of the transaction amount range.
SVC0270	User charging is blocked.	User is blocked.
SVC0271	User not exist.	User does not exist.
SVC4001	The subscriber is in blacklist	The subscriber is in blacklist.
SVC0906	Token does not exist	oauth_token in the SOAP Header does not exist.
SVC0272	Invalid User	IN return error: user is invalid
SVC0273	A temporary lack of space	IN return error:A temporary lack of space
SVC0274	Disconnected the transport connection	IN return error:Disconnected the transport connection
SVC0275	A request is rejected for unspecified reasons	IN return error:A request is rejected for unspecified reasons
SVC0276	Timeout while waiting for the response	Timeout while waiting for the response

Example

The following are examples of the chargeAmount request:

- 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/payment/amount_charging/v3_1/local">
  <soapenv:Header>
    <ns1:RequestSOAPHeader xmlns:ns1="http://www.huawei.com.cn/schema/common/v2_1">
      <spId>000201</spId>
      <spPassword>e6434ef249df55c7a21a0b45758a39bb</spPassword>
      <serviceId>35000001000012</serviceId>
```

```
        <timeStamp>20100731064245</timeStamp>
        <OA>989377123456</OA>
        <FA>989377123456</FA>
        <oauth_token>1215616AA</oauth_token>
      </RequestSOAPHeader>
    </soapenv:Header>
    <soapenv:Body>
      <loc:chargeAmount>
        <loc:endUserIdentifier>86123456</loc:endUserIdentifier>
        <loc:charge>
          <description>charge</description>
          <currency>USD</currency>
          <amount>100.00</amount>
          <code>4523</code>
        </loc:charge>
        <loc:referenceCode>225</loc:referenceCode>
      </loc:chargeAmount>
    </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/payment/amount_charging/v3_1/local">
  <soapenv:Header>
    <ns1:RequestSOAPHeader xmlns:ns1="http://www.huawei.com.cn/schema/common/v2_1">
      <spId>000201</spId>
      <spPassword>e6434ef249df55c7a21a0b45758a39bb</spPassword>
      <timeStamp>20100731064245</timeStamp>
      <bundleID>256000039</bundleID>
      <oauth_token>1215616AA</oauth_token>
    </RequestSOAPHeader>
  </soapenv:Header>
  <soapenv:Body>
    <loc:chargeAmount>
      <loc:endUserIdentifier>86123456</loc:endUserIdentifier>
      <loc:charge>
        <description>charge</description>
        <currency>USD/currency>
        <amount>100.00</amount>
        <code>4523</code>
      </loc:charge>
      <loc:referenceCode>225</loc:referenceCode>
    </loc:chargeAmount>
  </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:chargeAmountResponse
xmlns:ns1="http://www.csapi.org/schema/parlayx/payment/amount_charging/v3_1/local"/>
   </soapenv:Body>
</soapenv:Envelope>
```

09

Appendix

4.1 refundAmount

This interface is provided by the SDP (when functioning as the server) for a partner to directly apply a refund to the account indicated by the end user identifier.

Request URI

http://*IP*:*Port*/AmountChargingService/services/AmountCharging/v3

In the preceding information, *IP* and *Port* indicate the IP address and port number for the SDP to provide 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 refundAmount interface.

Table 4-1 refundAmount message header parameters

Parameter	Type	Level of Requirement	Description
spId	xsd: string	Mandatory	Partner ID. Partner ID automatically allocated by the SDP to the partner after successful registration. For a partner, use the registered account to log in to the SDP management portal and query the account information. [Example] 000201

Parameter	Type	Level of Requirement	Description
spPassword	xsd: string	Conditional	Encrypted authentication password for partners to access the SDP. 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**: authentication password for partners to access the SDP. The value is set during registration with the SDP by a partner. **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
serviceId	xsd: string	Conditional	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] 3500001000012

Parameter	Type	Level of Requirement	Description
bundleID	xsd: string	Conditional	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** • In the capability sales scenario, if the API partner uses the Transaction Cost charging mode, the **bundleID** field does not need to be contained and the SDP sets **bundleID** to **–1**. If the API partner is a prepaid or postpaid user, the **bundleID** field must be contained. If the API partner does not send the **bundleID** field, the SDP uses the Transaction Cost charging mode by default and sets **bundleID** to **–1**. • 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
timeStamp	xsd: string	Conditional	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
OA	xsd: string	Conditional	Mobile number or the fake ID of the service originator. • When the Partner charges a user that subscribes to or orders a service by fee, this parameter value is the mobile number or the fake ID of the charged user. • When the Partner charges a user that receives a gift from another user by fee, this parameter value is the mobile number or the fake ID of the gift sender.

Parameter	Type	Level of Requirement	Description
			[Example] • Mobile number: 8612312345678 • Fake ID: f-245-11900000007639 **NOTE** When an API Partner sends the request, **OA** parameter is optional.
FA	xsd: string	Conditional	Mobile number or the fake ID of the charged party. The value must be the same as the OA value. **NOTE** When an API Partner sends the request, **FA** parameter is optional.
endUserDAAccountId	xsd: string	Optional	The specific DA account identifier. It indicates the subscriber DA Account. If it presents SDP will try to refund using DA Account if DA Account in mentioned. Otherwise SDP will refund to main account. There is no fallback to main balance if DA account refunding is failed. Partner need to invoke refund amount request again if necessary.

refundAmountRequest

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

Table 4-2 refundAmount input parameters

Parameter	Type	Level of Requirement	Description
endUserIdentifier	xsd:anyURI	Mandatory	Account of the user to whom a partner needs to refund fees. [Format] • Mobile number: tel:[Prefix][Country code][Mobile number] In the format, *[Prefix]* is optional. The value of *[Prefix]*, if contained, can be **+**, **+0**, **+00**, **0**, or **00**. • Fake ID: tel: [Prefix]- [opcoid]-[sequence]. [Example] • Mobile number: tel:8612312345678 • Fake ID: tel:f-245-11900000007639

Parameter	Type	Level of Requirement	Description
charge	common:ChargingI informatio	Mandatory	Refunding information. For the detailed parameters, see Table 4-3.
referenceCode	xsd:string	Mandatory	Unique ID of the request. [Example] 225

Table 4-3 ChargingInformatin parameters

Parameter	Type	Level of Requirement	Description
description	xsd:string	Mandatory	Charging description. [Example] charge
currency	xsd:string	Mandatory	Currency used for the charging. [Example] USD
amount	xsd:decimal	Conditional	Amount of the refund fee. In case of amount based charging, this parameter is mandatory.[Example] 100.00
code	xsd:string	Conditional	Charging code, which is relevant to the contract of the charged party. In case of code based charging, this parameter is mandatory. [Example] 4523

refundAmountResponse

None.

Error Code

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

Table 4-4 Error codes of the refundAmount interface

Error Code	Description	Cause
SVC0001	Get addr route info failed!the baseScfType = %1,the sourceAddr = %2,the destAddr = %3.	An internal SDP service is abnormal.
	Waiting for response timed out, message type is %1.	An internal SDP service is abnormal.
	The next NE is out of service, message type is %1.	An internal SDP service is abnormal.
SVC0002	ChargingInformation is null.	The **ChargingInformation** value in the request body is blank.
	Description is null in ChargingInformation.	The **Description** value in the request body is blank.
	Amount is null.	The **Amount** value in the request body is blank.
	Invalid input value.	A parameter value in the request is invalid.
SVC0007	Amount field should not be NULL.	The **Amount** value in the request body is blank.
	Code field should not be NULL.	The **Code** value in the request body is blank.
	Amount field should be NULL.	The **Amount** value in the request body is not blank.
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.
	SPID %1 is not exist!	The partner specified by **SPID** in the request header does not exist in the SDP.

9 Appendix

Error Code	Description	Cause
	SP ip is null!	The IP address in the request header is blank.
	Sp ip %1 is not accepted!	The IP address in the request header is incorrect.
	Sp password is null!	The **password** value in the request header is blank.
	Timestamp is empty in soapheader.	The **timeStamp** value in the request header is blank.
	local SP password is null!	An internal error occurs in the SDP.
	Sp password is not accepted!	The **password** value in the request header is incorrect.
	The authentication type is unknown!	An internal SDP service is abnormal.
	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.
	SP %1 is in blacklist!	The partner account is in the blacklist.
	The sp's Status is pre-deregistered.	The partner is in the pre-deregistered state.
	The sp's Status is deregistered.	The partner is in the deregistered state.
	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.
	The sp's Status is unknown.	An internal SDP service is abnormal.
	The ScfType %1 is inactive!	An internal SDP service is abnormal.
	The ScfType %1 is uninstalled!	An internal SDP service is abnormal.
	Service ID %1 is invalid!	The **serviceID** value in the request header is incorrect.

9 Appendix

Error Code	Description	Cause
	Service ID %1 is not existed!	The **serviceID** value in the request header does not exist in the SDP.
	Service %1 is in blacklist!	The **serviceId** value in the request header in the blacklist.
	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.
	The service status is unknown.	An internal SDP service is abnormal.
	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.
	The sp %1 has not ordered the service %2!	The partner has not subscribed to the Payment service.
	The service %1 has not orderd the api %2.	An internal SDP service is abnormal.
	The service %1 has not orderd the SCF %2.	An internal SDP service is abnormal.
	The sp %1 has not orderd the api %2 in current date.	This partner does not have the permission for using the API.
	The sp %1 has not orderd the SCF %2.	The partner has not subscribed to the Payment service.
POL0904	SP level gross control not pass.	The number of requests delivered by the partner exceeds the signed TPS.

Error Code	Description	Cause
POL0910	Minimum Amount per Transaction.	The **Amount** value in the request is out of the transaction amount range.
SVC0270	User charging is blocked	User is blocked
SVC0271	User not exist.	User does not exist.
SVC4001	The subscriber is in blacklist	The subscriber is in blacklist
SVC0272	Invalid User	IN return error: user is invalid
SVC0273	A temporary lack of space	IN return error: A temporary lack of space
SVC0274	Disconnected the transport connection	IN return error: Disconnected the transport connection
SVC0275	A request is rejected for unspecified reasons	IN return error: A request is rejected for unspecified reasons
SVC0276	Timeout while waiting for the response	Timeout while waiting for the response

Example

The followings are examples of the refundAmount request:

- 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/payment/amount_charging/v3_1/local">
   <soapenv:Header>
      <ns1:RequestSOAPHeader xmlns:ns1="http://www.huawei.com.cn/schema/common/v2_1">
         <spId>000201</spId>
         <spPassword>e6434ef249df55c7a21a0b45758a39bb</spPassword>
         <serviceId>35000001000012</serviceId>
         <timeStamp>20100731064245</timeStamp>
         <OA>989377123456</OA>
         <FA>989377123456</FA>
       </RequestSOAPHeader>
    </soapenv:Header>
    <soapenv:Body>
      <loc:refundAmount>
        <loc:endUserIdentifier>86123456</loc:endUserIdentifier>
        <loc:charge>
          <description>charge</description>
          <currency>USD/currency>
          <amount>100.00</amount>
          <code>4523</code>
```

```
          </loc:charge>
          <loc:referenceCode>225</loc:referenceCode>
        </loc:refundAmount>
      </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/payment/amount_charging/v3_1/local">
   <soapenv:Header>
     <ns1:RequestSOAPHeader xmlns:ns1="http://www.huawei.com.cn/schema/common/v2_1">
       <spId>000201</spId>
       <spPassword>e6434ef249df55c7a21a0b45758a39bb</spPassword>
       <timeStamp>20100731064245</timeStamp>
       <bundleID>256000039</bundleID>
      </RequestSOAPHeader>
    </soapenv:Header>
    <soapenv:Body>
      <loc:refundAmount>
        <loc:endUserIdentifier>86123456</loc:endUserIdentifier>
        <loc:charge>
          <description>charge</description>
          <currency>USD</currency>
          <amount>100.00</amount>
          <code>4523</code>
         </loc:charge>
         <loc:referenceCode>225</loc:referenceCode>
        </loc:refundAmount>
      </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:refundAmountResponse
xmlns:ns1="http://www.csapi.org/schema/parlayx/payment/amount_charging/v3_1/local"/>
   </soapenv:Body>
</soapenv:Envelope>
```

05

Appendix

5.1 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

5.2 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

5.3 OpCoID List

MTN OpCo Name	OpCoID Value
Afghanistan	9301
Benin	22901

5 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