SDP Solution API Reference (Payment, ParlayREST)

Change History

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

Issue 13 (2016-08-11)

This issue is the thirteenth official release, and includes the following changes:
Modify the description of the **amount** field for 2.2.5 Response and 3.2.5 Response.

Issue 12 (2016-08-10)

This issue is the twelfth official release, and includes the following changes:
Modify the description of the **amount** field.

Issue 11 (2016-06-15)

This issue is the eleventh official release, and includes the following changes:
Change the **description** field type to string array.

Issue 10 (2016-03-03)

This issue is the tenth 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 09 (2015-11-20)

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

Issue 08 (2015-06-18)

This issue is the eighth official release, and includes the following changes:
Add error code **SVC4001** for chargeAmount and refundAmount.
Add error code **SVC0906** for chargeAmount.
Update the format description for 2.2.3 Request URI and 3.2.3 Request URI.
Update the format description of **endUserId** in Table 2-2, Table 2-5, Table 3-2 and Table 3-5.

Issue 07 (2015-01-12)

This issue is the seventh official release, and includes the following changes:
Add the error code **SVC0271** for chargeAmount and refundAmount.
Change the **amount** example in JSON encoding format for chargeAmount and refundAmount.
Optimize the description of **oauth_token** for chargeAmount.

Issue 06 (2014-10-31)

This issue is the sixth official release, and includes the following changes:
Add the parameter **endUserDAAccountId** in Table 2-1 and Table 3-1.
Update the description about **bundleID**.

Issue 05 (2014-05-04)

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

Change History

Issue 04 (2014-01-24)

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

Issue 03 (2013-12-23)

This issue is the third 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 02 (2013-11-19)

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

Issue 01 (2013-10-30)

This issue is the first official release for phase2.3.

Contents

01 Overview

1.1 API Functions

Payment APIs provided by the SDP are mainly used by the SP during Payment service development to deduct a fee from or refund a fee to a user's account.

Table 1-1 describes functions of Payment capability APIs provided by the SDP.

Table 1-1 Functions of Payment capability APIs

Function	Subfunction	Description	API
Deduction	Deduct a fee from a user's account	The SP (functioning as the client) invokes the chargeAmount API provided by the SDP (functioning as the server) to deduct a fee from a user's account. SPs must develop the App code based on the API field requirements so that the App can send correct requests to the SDP. The SDP sends a response within 60 seconds by default.	chargeAmount

Overview

Function	Subfunction	Description	API
Refund	Refund a fee to a user's account	The SP (functioning as the client) invokes the refundAmount API provided by the SDP (functioning as the server) to refund a fee to a user's account. SPs must develop the App code based on the API field requirements so that the App can send correct requests to the SDP. The SDP sends a response within 60 seconds by default.	refundAmount

1.2 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.

Type	Description
Optional	A parameter is always optional. Parameters with the **Optional** requirement are not used for service processing.

1.3 Message Format

The SDP provides standard HTTP messages. A request consists of the request line, request header, and request body. A response consists of the status line, response header, and response body.
The request line consists of the following:

- Request method: POST, GET, PUT, and DELETE.
- Request URI: address used by the server to provide services.
- HTTP protocol version: HTTP/1.1.

The status line consists of the following:

- HTTP protocol version: HTTP/1.1.
- Response status code: complies with the HTTP protocol.
- Response status description: description of a status code, which complies with the HTTP protocol.

1.4 Status Codes

Status codes sent by the SDP comply with the HTTP protocol. Table 1-3 lists the status codes.

Table 1-3 HTTP status codes

Status Code	Description
200	Success.
201	Created.
204	No Content.
304	ConditionNotMet - Not Modified: The condition specified in the conditional header(s) was not met for a read operation.
400	Invalid parameters in the request
401	Authentication failure.
403	Application don't have permissions to access resource due to the policy constraints (request rate limit, etc).

Overview

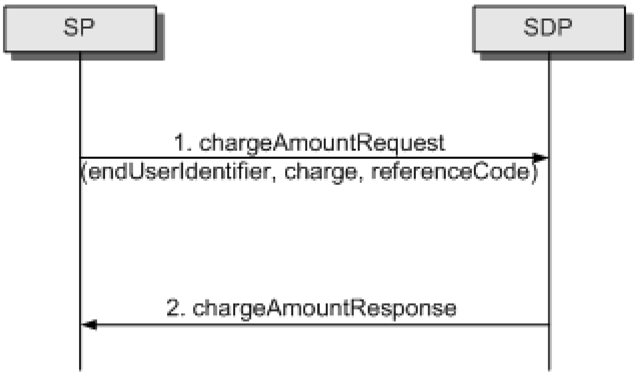
Status Code	Description
404	Not Found - The specified resource does not exist.
405	Method not allowed by the resource.
409	Conflict.
411	Length Required: The Content-Length header was not specified.
412	Precondition Failed: The condition specified in the conditional header(s) was not met for a write operation.
413	RequestBodyTooLarge - Request Entity Too Large: The size of the request body exceeds the maximum size permitted.
416	InvalidRange - Requested Range Not Satisfiable: The range specified is invalid for the current size of the resource.
500	Internal server error.
503	ServerBusy - Service Unavailable: The server is currently unable to receive requests. Please retry your request.

02 APIs for Deduction

2.1 Process

Figure 2-1 shows the process in which an SP uses the Payment capability provided by the SDP to deduct a fee from a user's account.

Figure 2-1 chargeAmount service process



The process is as follows:

- An SP uses the chargeAmount interface to send a request to the SDP.

  The request body must contain the endUserIdentifier, charge, and referenceCode parameters.
- The SDP performs charging and sends a response to the SP.

2.2 chargeAmount

2.2.1 Function

The SP (functioning as the client) invokes the chargeAmount API provided by the SDP (functioning as the server) to deduct a fee from a user's account.

SPs must develop the App code based on the API field requirements so that the App can send correct requests to the SDP. The SDP sends a response within 60 seconds by default.

2.2.2 Request Method

Set it to **POST**.

2.2.3 Request URI

The request URI is the destination URI of chargeAmount request sent by the SP to the SDP. The URI is provided by the SDP in the following format:

http://*IP*:*Port*/*apiVersion*/payment/*endUserId*/transactions/amount

In the format,

- **IP** indicates the service IP address provided by the SDP. Contact carriers to obtain the IP address.
- **Port** indicates the ParlayREST port number of the API provided by the SDP. Contact carriers to obtain the port number.
- **apiVersion** has a fixed value of **1**.
- **endUserId** indicates the 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].

2.2.4 Request

The SP functions as the client and sends a **chargeAmountRequest** message to the SDP to deduct a fee from a user's account.

Example

Request examples in different request body encoding formats are as follows:

- In XML encoding format:

```
POST ../1/payment/861234567/transactions/amount HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="D6ri9fAmdFhGijaU6c+9myLB/9s=",Nonce="2013042718472200001",
Created="2013-04-27T18:47:22Z"
X-RequestHeader: request ServiceId="35000001000012"
Accept-Encoding: gzip,deflate
Accept: application/xml
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/xml; charset=UTF-8
Content-Length: 12345

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:861234567</endUserId>
 <paymentAmount>
   <chargingInformation>
     <description>Test amount transaction in "Charged" state</description>
     <currency>USD</currency>
     <amount>10</amount>
     <code>TEST-012345</code>
   </chargingInformation>
 </paymentAmount>
 <transactionStatus>Charged</transactionStatus>
 <referenceCode>REF-12345</referenceCode>
 </payment:amountTransaction>
```

- In JSON encoding format:

```
POST ../1/payment/861234567/transactions/amount HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="D6ri9fAmdFhGijaU6c+9myLB/9s=",Nonce="2013042718472200001",
Created="2013-04-27T18:47:22Z"
X-RequestHeader: request ServiceId="35000001000012"
Accept-Encoding: gzip,deflate
Accept: application/json
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/json
Content-Length: 12345

{"amountTransaction": {
    "endUserId": "tel:861234567",
    "paymentAmount": {"chargingInformation": {
        "amount": 10,
        "code": "TEST-012345",
        "currency": "USD",
        "description": ["Test amount transaction in \"Charged\" state"]
    }},
    "referenceCode": "REF-12345",
    "transactionStatus": "Charged"
}}
```

- In URL encoding format:

```
POST ../1/payment/861234567/transactions/amount HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="D6ri9fAmdFhGijaU6c+9myLB/9s=",Nonce="2013042718472200001",
Created="2013-04-27T18:47:22Z"
X-RequestHeader: request ServiceId="35000001000012"
Accept-Encoding: gzip,deflate
```

```
Accept: application/xml
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 10.137.213.125:14312
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345

endUserId=tel:861234567&transactionStatus=Charged&description=Test amount transaction in "Charged"
state&currency=USD&amount=10&code=TEST-012345&referenceCode=REF-12345
```

- An example of the **chargeAmountRequest** message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc="http://www.csapi.org/schema/parlayx/terminal_location/v2_2/local">
  <soapenv:Header>
    <tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com.cn/schema/common/v2_1">
      <tns:spId>270110000420</tns:spId>
      <tns:spPassword>268f43ce43e4af823c3c79dba88fd116</tns:spPassword>
      <tns:serviceId>27012000003285</tns:serviceId>
      <tns:timeStamp>20121010134322</tns:timeStamp>
      <tns:OA>27136600114</tns:OA>
      <tns:FA>27136600114</tns:FA>
      <tns:oauth_token> 1215616AA </tns:oauth_token>
    </tns:RequestSOAPHeader>
  </soapenv:Header>
```

Message Header Parameters

Table 2-1 describes parameters in a **chargeAmountRequest** message header.

Table 2-1 Parameters in the chargeAmountRequest message header

Header Field	Parameter	Type	Length	Level of Requirement	Description
Accept	-	String	16	Optional	Format in which responses are encoded. The options are as follows: • application/xml • application/json When this parameter is not transmitted, the SDP uses the **application/xml** mode to encode responses by default. [Example] application/xml

Header Field	Parameter	Type	Length	Level of Requirement	Description
Content-Type	-	String	33	Mandatory	Format in which requests are encoded. • application/xml • application/json • application/x-www-form-urlencoded [Example] application/xml
Content-length	-	int	2	Mandatory	Number of bytes in the request body. [Example] 1024
Authorization	/	String	10	Mandatory	Authentication mode. The SDP uses the WSSE authentication mode, in which the **realm** and **profile** parameters are involved. **NOTE** The **Authorization** value indicates the policy used by the SDP to perform authentication at the application layer. The SDP authenticates a third-party in WSSE UsernameToken mode. [Example] WSSE
	realm	String	20	Mandatory	Party that performs authentication. Set it to **SDP**. [Example] SDP
	profile	String	20	Mandatory	Third-party access mechanism. Set it to **UsernameToken**. [Example] UsernameToken
X-WSSE	/	String	30	Mandatory	WSSE authentication flag. To enable the SDP to perform authentication at the application layer in the WSSE UsernameToken mode, set this parameter to **UsernameToken**. [Example] UsernameToken

Header Field	Parameter	Type	Length	Level of Requirement	Description
	Username	String	21	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • A service Partner and an API Partner can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. [Example] 000201

Header Field	Parameter	Type	Length	Level of Requirement	Description
	PasswordDigest	String	100	Mandatory	Authentication key for the SDP to authenticate partners. The value is a character string encrypted. The encryption formula is as follows: • SHA-256: PasswordDigest = Base64(SHA-256(Nonce + Created + Password)) • SHA-1: PasswordDigest = Base64(SHA-1(Nonce + Created + Password)) In the formula: • **Nonce** and **Created**: random number and its generation time. • **Password**: password for partners to access the SDP. An SP can obtain the password from the email notification received after successful registration. **NOTE** To retain features of earlier versions, the SP uses the SHA-1 algorithm in the connection to the SDP, which might cause security risks. [Example] 7647734A6C73437A426D337153682F4D67627A6C53734666706B733D
	Nonce	String	30	Mandatory	Random number generated by an SP when sending a message. [Example] 66C92B11FF8A425FB8D4CCFE0ED9ED1F
	Created	String	20	Mandatory	UTC time when the **Nonce** value is generated. [Format] yyyy-MM-dd'T'HH:mm:ss'Z' [Example] 2013-04-27T18:47:22Z
X-RequestHeader	/	String	10	Mandatory	Type of the message sent by a third party. Set it to **request**. [Example] request

Header Field	Parameter	Type	Length	Level of Requirement	Description
	ServiceId	String	21	Mandatory	Service ID. The ID is automatically allocated by the SDP to services after successful release. Partner can log in to the SDP Management Portal and query service information for the 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

APIs for Deduction

Header Field	Parameter	Type	Length	Level of Requirement	Description
	bundleID	String	21	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
	oauth_token	String	20	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.

Header Field	Parameter	Type	Length	Level of Requirement	Description
	endUserDAAccountId	xsd: string	10	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.

Message Body Parameters

Table 2-2 describes parameters in a **chargeAmountRequest** message body.

Table 2-2 Parameters in a chargeAmountRequest message body

Parameter	type	Length	Level of Requirement	Description
endUserId	xsd:anyURI	30	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]. [Example] • Mobile number: tel:8612312345678 • Fake ID: tel: f-245-11900000007639
paymentAmount	paymentAmount	-	Mandatory	Charging information. For the detailed parameters, see Table 2-3.

Parameter	type	Length	Level of Requirement	Description
transactionStatus	TransactionStatus	-	Mandatory	Status of the transaction. The value is a numeration value and has the following options: • Charged • Refunded
referenceCode	xsd:string	30	Mandatory	Unique ID of the charge request. [Example] 225

Table 2-3 ChargingInformation parameters

Parameter	type	Length	Level of Requirement	Description
description	xsd:string[1…unbounded]	255	Mandatory	Charging description. [Example] charge
currency	xsd:string	3	Mandatory	Currency used for the charging. See *ISO 4217, Codes for the representation of currencies and funds*. [Example] USD
amount	xsd:decimal	4	Conditional	Amount of the charged fee. In case of amount based charging, this parameter is mandatory. This parameter can be decimal in case of the "major currency support" configured for this partner, otherwise, it should be an integer value for the minimum currency unit. [Example] 100

Parameter	type	Length	Level of Requirement	Description
code	xsd:string	30	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

2.2.5 Response

The SDP functions as the server, processes **chargeAmountRequest** messages received from the SP, and sends **chargeAmountResponse** messages to the SP.

This topic provides a success response example. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see 4 API Error Responses.

Example

Response examples in different response body encoding formats are as follows:

- In XML encoding format:

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: 12345
Location: http://10.135.101.26:8312/1/payment/861234567/transactions/amount/123456789

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:861234567</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Charged" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountCharged>10</totalAmountCharged>
  </paymentAmount>
  <transactionStatus>Charged</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
<resourceURL>http://10.135.101.26:8312/1/payment/861234567/transactions/amount/123456789</resourc
eURL>
</payment:amountTransaction>
```

- In JSON encoding format:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Location: http://10.135.101.26:8312/1/payment/861234567/transactions/amount/123456789

{"amountTransaction": {
   "clientCorrelator": "54321",
   "endUserId": "tel:861234567",
   "paymentAmount": {
      "chargingInformation": {
         "amount": 10,
         "code": "TEST-012345",
         "currency": "USD",
         "description": ["Test amount transaction in \"Charged\" state"]
      },
      "totalAmountCharged": "10"
   },
   "referenceCode": "REF-12345",
   "resourceURL": "http://10.135.101.26:8312/1/payment/861234567/transactions/amount/123456789",
   "transactionStatus": "Charged"
}}
```

Message header Parameters

Table 2-4 describes parameters in the **chargeAmountResponse** message header.

Table 2-4 Parameters in the chargeAmountResponse message header

Parameter	type	Length	Level of Requirement	Description
Content-Type	xsd:string	16	Mandatory	Format in which the response is encoded. The options are as follows: • application/xml • application/json [Example] application/xml
Content-length	int	2	Mandatory	Number of bytes in the response body. [Example] 1024

Parameter	type	Length	Level of Requirement	Description
Location	xsd:string	512	Mandatory	Message redirected-to URL sent by the SDP. [Format] http://*serverRoot*/*apiVersion*/payment/*endUserId*/transactions/amount/*requestId* In the format: • *serverRoot*: service root address provided by the SDP. The format is *IP*:*Port*. • *apiVersion*: the value must be **1**. • *endUserId*: account of the user to be charged. • *requestId*: unique ID sent by the SDP during the message delivery. [Example] http://10.135.101.26:8312/1/payment/861234567/transactions/amount/123456789

Message Body Parameters

Table 2-5 describes parameters in a **chargeAmountResponse** message body.

Table 2-5 Parameters in a chargeAmountResponse message body

Parameter	type	Length	Level of Requirement	Description
endUserId	xsd:anyURI	30	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]. [Example] • Mobile number: tel:8612312345678 • Fake ID: tel: f-245-11900000007639
paymentAmount	paymentAmount	-	Mandatory	Charging information. For the detailed parameters, see Table 2-6.
transactionStatus	TransactionStatus	-	Mandatory	Status of the transaction. The value is a numeration value and has the following options: • Charged • Refunded
referenceCode	xsd:string	30	Mandatory	Unique ID of the charge request. [Example] 225

Parameter	type	Length	Level of Requirement	Description
resourceURL	xsd:anyURI	512	Optional	Message redirected-to URL sent by the SDP. [Format] http://*serverRoot*/*apiVersion*/payment/*endUserId*/transactions/amount/*requestId* In the format: • *serverRoot*: service root address provided by the SDP. The format is *IP*:*Port*. • *apiVersion*: the value must be **1**. • *endUserId*: account of the user to be charged. • *requestId*: unique ID sent by the SDP during the message delivery. [Example] http://10.135.101.26:8312/1/payment/861234567/transactions/amount/123456789

Table 2-6 ChargingInformation parameters

Parameter	type	Length	Level of Requirement	Description
description	xsd:string[1…unbounded]	255	Mandatory	Charging description. [Example] charge
currency	xsd:string	3	Mandatory	Currency used for the charging. See *ISO 4217, Codes for the representation of currencies and funds*. [Example] USD

Parameter	type	Length	Level of Requirement	Description
amount	xsd:decimal	4	Mandatory	Amount of the charged fee. In case of amount based charging, this parameter is mandatory. This parameter can be decimal in case of the "major currency support" configured for this partner, otherwise, it should be an integer value for the minimum currency unit. [Example] 100
code	xsd:string	30	Optional	Charging code, which is relevant to the contract of the charged party. [Example] 4523

2.2.6 Error Codes

Table 2-7 describes chargeAmount error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 2-7 Error codes of the chargeAmount API

Error Code	Description	Cause
SVC0001	invalid_client.	Client is invalid.
SVC0002	endUserId is null.	The **endUserId** value in the request body is blank.
	endUserId is invalid.	The **endUserId** value in the request is invalid.
	referenceCode is null.	The **referenceCode** value in the request body is blank.
	transactionStatus is null.	The **transactionStatus** value in the request body is blank.
	transactionStatus is not in enumeration.	The **transactionStatus** value in the request is invalid.

APIs for Deduction

Error Code	Description	Cause
	PaymentAmount/ChargingInformation is null.	The **ChargingInformation** value in the request body is blank.
	PaymentAmount/ChargingInformation/descriptions is null	The **Description** value in the request body is blank.
	%1 is null.	A parameter value in the request is blank.
	PaymentAmount is null.	The **PaymentAmount** value in the request body is blank.
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	UserName is null!	The **SPID** value in the request header is blank or does not exist in the SDP.
	UserName %1 is invalid!	The **SPID** value in the request header is in an incorrect format.
	Authentication Failed, cause by SP,because of timestamp expired.	The **timestamp** value in the request header is invalid.
	SP ip is null!	The IP address in the request header is blank.
	Created is empty in X-WSSE httpheader.	The **Created** value in the request header is blank.
	Nonce is empty in X-WSSE httpheader.	The **Nonce** value in the request header is blank.
	Sp password is null!	The **password** value in the request header is blank.
	Sp ip %1 is not accepted!	The IP address in the request header is incorrect.
	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.

APIs for Deduction

Error Code	Description	Cause
	The authentication type is unknown!	An internal SDP service is abnormal.
	UserName %1 is not exist!	The SP specified by **SPID** in the request header does not exist in the SDP.
	SP %1 is in blacklist!	The SP account is in the blacklist.
	The sp's Status is pre-deregistered.	The SP is in the pre-deregistered state.
	The sp's Status is deregistered.	The SP is in the deregistered state.
	SP status is locked.	The SP is in the locked state.
	The sp's Status is forbidden.	The SP is in the forbidden state.
	The sp 's status is pause.	The SP is in the paused state.
	The sp's Status is unknown.	An internal SDP service is abnormal.
	Service ID %1 is not existed!	The **serviceID** value in the request header does not exist in the SDP.
	The API %1 is not existed.	This SP does not have the permission for using the API.
	The API status is disabled.	This SP does not have the permission for using the API.
	The sp %1 has not ordered the service %2!	The SP 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 SP does not have the permission for using the API.
	The sp %1 has not orderd the SCF %2.	The SP has not subscribed to the Payment service.
SVC0905	Authentication Failed, cause by SP,because of timestamp is not valid.	The **timestamp** value in the request header is invalid.

APIs for Deduction

Error Code	Description	Cause
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 current SP is not authorized by the user.
POL0907	Real MSISDN not allowed.	An internal SDP service is abnormal.
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	The amount to be charged is Invalid Range.	The **Amount** value in the request is out of the transaction amount range.
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 HTTP 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

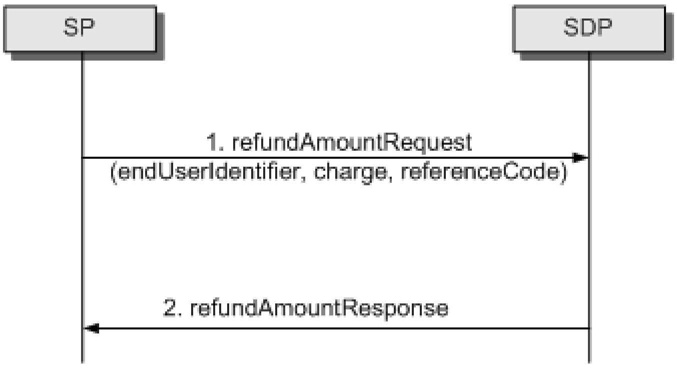
03

APIs for Refund

3.1 Process

Figure 3-1 shows the process in which an SP uses the Payment capability provided by the SDP to refund a fee to a user's account.

Figure 3-1 refundAmount service process



The process is as follows:

- An SP uses the refundAmount interface to send a request to the SDP.

  The request body must contain the **endUserIdentifier**, **charge**, and **referenceCode** parameters.
- The SDP performs refund and sends a response to the SP.

3.2 refundAmount

3.2.1 Function

The SP (functioning as the client) invokes the refundAmount API provided by the SDP (functioning as the server) to refund a fee to a user's account.

SPs must develop the App code based on the API field requirements so that the App can send correct requests to the SDP. The SDP sends a response within 60 seconds by default.

3.2.2 Request Method

Set it to **POST**.

3.2.3 Request URI

The request URI is the destination URI of refundAmount request sent by the SP to the SDP. The URI is provided by the SDP in the following format:

http://*IP*:*Port*/*apiVersion*/payment/*endUserId*/transactions/amount

In the format,

- **IP** indicates the service IP address provided by the SDP. Contact carriers to obtain the IP address.
- **Port** indicates the ParlayREST port number of the API provided by the SDP. Contact carriers to obtain the port number.
- **apiVersion** has a fixed value of **1**.
- **endUserId** indicates the 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].

3.2.4 Request

The SP functions as the client and sends a **refundAmountRequest** message to the SDP to refund a fee to a user's account.

Example

Request examples in different request body encoding formats are as follows:

- In XML encoding format:

```
POST ../1/payment/861234567/transactions/amount HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="D6ri9fAmdFhGijaU6c+9myLB/9s=",Nonce="2013042718472200001",
Created="2013-04-27T18:47:22Z"
X-RequestHeader: request ServiceId="35000001000012"
Accept: application/xml
Host: example.com:80
Content-Type: application/xml
Content-Length: 12345

<?xml version="1.0" encoding="UTF-8"?>
<payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:861234567</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Refunded" state</description>
```

```
     <currency>USD</currency>
     <amount>10</amount>
     <code>TEST-012345</code>
   </chargingInformation>
  </paymentAmount>
  <transactionStatus>Refunded</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
</payment:amountTransaction>
```

- In JSON encoding format:

```
POST ../1/payment/861234567/transactions/amount HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="D6ri9fAmdFhGijaU6c+9myLB/9s=",Nonce="2013042718472200001",
Created="2013-04-27T18:47:22Z"
X-RequestHeader: request ServiceId="35000001000012"
Accept: application/json
Host: example.com:80
Content-Type: application/json
Content-Length: 12345

{"amountTransaction": {
   "clientCorrelator": "54321",
   "endUserId": "tel:861234567",
   "paymentAmount": {"chargingInformation": {
      "amount": 10,
      "code": "TEST-012345",
      "currency": "USD",
      "description": ["Test amount transaction in \"Refunded\" state"]
   }},
   "referenceCode": "REF-12345",
   "transactionStatus": "Refunded"
}}
```

- In URL encoding format:

```
POST ../1/payment/861234567/transactions/amount HTTP/1.1
Authorization: WSSE realm="SDP",profile="UsernameToken"
X-WSSE: UsernameToken
Username="35000001",PasswordDigest="D6ri9fAmdFhGijaU6c+9myLB/9s=",Nonce="2013042718472200001",
Created="2013-04-27T18:47:22Z"
X-RequestHeader: request ServiceId="35000001000012"
Accept: application/xml
Host: example.com:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 12345
```

```
endUserId=tel:861234567&transactionStatus=refunded&description= Test amount transaction in
"Refunded"
state&currency=USD&amount=10&code=TEST-012345&referenceCode=REF-12345&clientCorrelator=54321
```

Message Header Parameters

Table 3-1 describes parameters in a **refundAmountRequest** message header.

Table 3-1 Parameters in the refundAmountRequest message header

Header Field	Parameter	Type	Length	Level of Requirement	Description
Accept	-	String	16	Optional	Format in which responses are encoded. The options are as follows: • application/xml • application/json When this parameter is not transmitted, the SDP uses the **application/xml** mode to encode responses by default. [Example] application/xml
Content-Type	-	String	33	Mandatory	Format in which requests are encoded. • application/xml • application/json • application/x-www-form-urlencoded [Example] application/xml
Content-length	-	int	2	Mandatory	Number of bytes in the request body. [Example] 1024
Authorization	/	String	10	Mandatory	Authentication mode. The SDP uses the WSSE authentication mode, in which the **realm** and **profile** parameters are involved. **NOTE** The **Authorization** value indicates the policy used by the SDP to perform authentication at the application layer. The SDP authenticates a third-party in WSSE UsernameToken mode. [Example] WSSE

Header Field	Parameter	Type	Length	Level of Requirement	Description
	realm	String	20	Mandatory	Party that performs authentication. Set it to **SDP**. [Example] SDP
	profile	String	20	Mandatory	Third-party access mechanism. Set it to **UsernameToken**. [Example] UsernameToken
X-WSSE	/	String	30	Mandatory	WSSE authentication flag. To enable the SDP to perform authentication at the application layer in the WSSE UsernameToken mode, set this parameter to **UsernameToken**. [Example] UsernameToken
	Username	String	21	Mandatory	Partner ID. The ID is automatically allocated by the SDP to partners after successful registration. To obtain the ID: • A service Partner and an API Partner can log in to the SDP management portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. • A Developer can log in to the Developer Portal and query account information, or log in to the mailbox used for registration and view the email notification received after successful registration. [Example] 000201

APIs for Refund

Header Field	Parameter	Type	Length	Level of Requirement	Description
	PasswordDigest	String	100	Mandatory	Authentication key for the SDP to authenticate partners. The value is a character string encrypted. The encryption formula is as follows: • SHA-256: PasswordDigest = Base64(SHA-256(Nonce + Created + Password)) • SHA-1: PasswordDigest = Base64(SHA-1(Nonce + Created + Password)) In the formula: • **Nonce** and **Created**: random number and its generation time. • **Password**: password for partners to access the SDP. An SP can obtain the password from the email notification received after successful registration. **NOTE** To retain features of earlier versions, the SP uses the SHA-1 algorithm in the connection to the SDP, which might cause security risks. [Example] 7647734A6C73437A426D337153682F4D67627A6C53734666706B733D
	Nonce	String	30	Mandatory	Random number generated by an SP when sending a message. [Example] 66C92B11FF8A425FB8D4CCFE0ED9ED1F
	Created	String	20	Mandatory	UTC time when the **Nonce** value is generated. [Format] yyyy-MM-dd'T'HH:mm:ss'Z' [Example] 2013-04-27T18:47:22Z

APIs for Refund

Header Field	Parameter	Type	Length	Level of Requirement	Description
X-RequestHeader	/	String	10	Mandatory	Type of the message sent by a third party. Set it to **request**. [Example] request
	ServiceId	String	21	Mandatory	Service ID. The ID is automatically allocated by the SDP to services after successful release. Partner can log in to the SDP Management Portal and query service information for the 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

Header Field	Parameter	Type	Length	Level of Requirement	Description
	bundleID	String	21	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
	endUserDAAccountId	xsd: string	10	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.

Message Body Parameters

Table 3-2 describes parameters in a **refundAmountRequest** message body.

Table 3-2 Parameters in a refundAmountRequest message body

Parameter	type	Length	Level of Requirement	Description
endUserId	xsd:anyURI	30	Mandatory	Account of the user to whom a third party 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
paymentAmount	paymentAmount	-	Mandatory	Charging information. For the detailed parameters, see Table 3-3.
transactionStatus	TransactionStatus	-	Mandatory	Status of the transaction. The value is a numeration value and has the following options: • Charged • Refunded
referenceCode	xsd:string	30	Mandatory	Unique ID of the refund request. [Example] 225

Table 3-3 ChargingInformation parameters

Parameter	type	Length	Level of Requirement	Description
description	xsd:string[1…unbounded]	255	Mandatory	Charging description. [Example] charge
currency	xsd:string	3	Mandatory	Currency used for the charging. See *ISO 4217, Codes for the representation of currencies and funds*. [Example] USD
amount	xsd:decimal	4	Conditional	Amount of the refund fee. In case of amount based charging, this parameter is mandatory. This parameter can be decimal in case of the "major currency support" configured for this partner, otherwise, it should be an integer value for the minimum currency unit. [Example] 100
code	xsd:string	30	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

3.2.5 Response

The SDP functions as the server, processes **refundAmountRequest** messages received from the SP, and sends **refundAmountResponse** messages to the SP.

This topic provides a success response example. If a request fails, the SDP sends an error response that contains an error code. For details about error responses, see 4 API Error Responses.

Example

Response examples in different response body encoding formats are as follows:

- In XML encoding format:

```
HTTP/1.1 201 Created
Content-Type: application/xml
Content-Length: 12345
Location: http://10.135.101.26:8312/1/payment/861234567/transactions/amount/987654321

<?xml version="1.0" encoding="UTF-8"?>
 <payment:amountTransaction xmlns:payment="urn:oma:xml:rest:payment:1">
  <endUserId>tel:861234567</endUserId>
  <paymentAmount>
    <chargingInformation>
      <description>Test amount transaction in "Refunded" state</description>
      <currency>USD</currency>
      <amount>10</amount>
      <code>TEST-012345</code>
    </chargingInformation>
    <totalAmountRefunded>10</totalAmountRefunded>
  </paymentAmount>
  <transactionStatus>Refunded</transactionStatus>
  <referenceCode>REF-12345</referenceCode>
  <clientCorrelator>54321</clientCorrelator>
<resourceURL>http://10.135.101.26:8312/1/payment/861234567/transactions/amount/987654321</resourc
eURL>
</payment:amountTransaction>
```

- In JSON encoding format:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 12345
Location: http://10.135.101.26:8312/1/payment/861234567/transactions/amount/987654321

{"amountTransaction": {
   "clientCorrelator": "54321",
   "endUserId": "tel:861234567",
   "paymentAmount": {
      "chargingInformation": {
         "amount": 10,
         "code": "TEST-012345",
         "currency": "USD",
         "description": ["Test amount transaction in \"Refunded\" state"]
      },
      "totalAmountRefunded": "10"
   },
   "referenceCode": "REF-12345",
   "resourceURL": "http://10.135.101.26:8312/1/payment/861234567/transactions/amount/987654321",
   "transactionStatus": "Refunded"
}}
```

Message header Parameters

Table 3-4 describes parameters in the **refundAmountResponse** message header.

Table 3-4 Parameters in the refundAmountResponse message header

Parameter	type	Length	Level of Requirement	Description
Content-Type	xsd:string	16	Mandatory	Format in which the response is encoded. The options are as follows: • application/xml • application/json [Example] application/xml
Content-length	int	2	Mandatory	Number of bytes in the response body. [Example] 1024

Parameter	type	Length	Level of Requirement	Description
Location	xsd:string	512	Mandatory	Message redirected-to URL sent by the SDP. [Format] http://*serverRoot*/*apiVersion*/payment/*endUserId*/transactions/amount/*requestId* In the format: • *serverRoot*: service root address provided by the SDP. The format is *IP*:*Port*. • *apiVersion*: the value must be **1**. • *endUserId*: account of the user to be charged. • *requestId*: unique ID sent by the SDP during the message delivery. [Example] http://10.135.101.26:8312/1/payment/861234567/transactions/amount/987654321

Message Body Parameters

Table 3-5 describes parameters in a **refundAmountResponse** message body.

Table 3-5 Parameters in a refundAmountResponse message body

Parameter	type	Length	Level of Requirement	Description
endUserId	xsd:anyURI	30	Mandatory	Account of the user to whom a third party 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
paymentAmount	paymentAmount	-	Mandatory	Charging information. For the detailed parameters, see Table 3-6.
transactionStatus	TransactionStatus	-	Mandatory	Status of the transaction. The value is a numeration value and has the following options: • Charged • Refunded
referenceCode	xsd:string	30	Mandatory	Unique ID of the refund request. [Example] 225

Parameter	type	Length	Level of Requirement	Description
resourceURL	xsd:anyURI	512	Optional	Message redirected-to URL sent by the SDP. [Format] http://*serverRoot*/*apiVersion*/payment/*endUserId*/transactions/amount/*requestId* In the format: • *serverRoot*: service root address provided by the SDP. The format is *IP*:*Port*. • *apiVersion*: the value must be **1**. • *endUserId*: account of the user to be charged. • *requestId*: unique ID sent by the SDP during the message delivery. [Example] http://10.135.101.26:8312/1/payment/861234567/transactions/amount/987654321

Table 3-6 ChargingInformation parameters

Parameter	type	Length	Level of Requirement	Description
description	xsd:string[1…unbounded]	255	Mandatory	Charging description. [Example] charge
currency	xsd:string	3	Mandatory	Currency used for the charging. See *ISO 4217, Codes for the representation of currencies and funds*. [Example] USD

Parameter	type	Length	Level of Requirement	Description
amount	xsd:decimal	4	Mandatory	Amount of the charged fee. In case of amount based charging, this parameter is mandatory. This parameter can be decimal in case of the "major currency support" configured for this partner, otherwise, it should be an integer value for the minimum currency unit. [Example] 100
code	xsd:string	30	Optional	Charging code, which is relevant to the contract of the charged party. [Example] 4523

3.2.6 Error Codes

Table 3-7 describes refundAmount error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

Table 3-7 Error codes of the refundAmount API

Error Code	Description	Cause
SVC0001	invalid_client.	Client is invalid.
SVC0002	endUserId is null.	The **endUserId** value in the request body is blank.
	endUserId is invalid.	The **endUserId** value in the request is invalid.
	referenceCode is null.	The **referenceCode** value in the request body is blank.
	transactionStatus is null.	The **transactionStatus** value in the request body is blank.
	transactionStatus is not in enumeration.	The **transactionStatus** value in the request is invalid.

APIs for Refund

Error Code	Description	Cause
	PaymentAmount/ChargingInformation is null.	The **ChargingInformation** value in the request body is blank.
	PaymentAmount/ChargingInformation/descriptions is null	The **Description** value in the request body is blank.
	%1 is null.	A parameter value in the request is blank.
	PaymentAmount is null.	The **PaymentAmount** value in the request body is blank.
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	UserName is null!	The **SPID** value in the request header is blank or does not exist in the SDP.
	UserName %1 is invalid!	The **SPID** value in the request header is in an incorrect format.
	Authentication Failed, cause by SP,because of timestamp expired.	The **timestamp** value in the request header is invalid.
	SP ip is null!	The IP address in the request header is blank.
	Created is empty in X-WSSE httpheader.	The **Created** value in the request header is blank.
	Nonce is empty in X-WSSE httpheader.	The **Nonce** value in the request header is blank.
	Sp password is null!	The **password** value in the request header is blank.
	Sp ip %1 is not accepted!	The IP address in the request header is incorrect.
	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.

APIs for Refund

Error Code	Description	Cause
	The authentication type is unknown!	An internal SDP service is abnormal.
	UserName %1 is not exist!	The SP specified by **SPID** in the request header does not exist in the SDP.
	SP %1 is in blacklist!	The SP account is in the blacklist.
	The sp's Status is pre-deregistered.	The SP is in the pre-deregistered state.
	The sp's Status is deregistered.	The SP is in the deregistered state.
	SP status is locked.	The SP is in the locked state.
	The sp's Status is forbidden.	The SP is in the forbidden state.
	The sp 's status is pause.	The SP is in the paused state.
	The sp's Status is unknown.	An internal SDP service is abnormal.
	Service ID %1 is not existed!	The **serviceID** value in the request header does not exist in the SDP.
	The API %1 is not existed.	This SP does not have the permission for using the API.
	The API status is disabled.	This SP does not have the permission for using the API.
	The sp %1 has not ordered the service %2!	The SP 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 SP does not have the permission for using the API.
	The sp %1 has not orderd the SCF %2.	The SP has not subscribed to the Payment service.
SVC0905	Authentication Failed, cause by SP,because of timestamp is not valid.	The **timestamp** value in the request header is invalid.

APIs for Refund

Error Code	Description	Cause
POL0907	Real MSISDN not allowed.	An internal SDP service is abnormal.
POL0910	The amount to be charged is Invalid Range.	The **Amount** value in the request is out of the transaction amount range.
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

04

API Error Responses

4.1 Service Error Response

A service error is caused by service operation exceptions irrelevant to policies. When a service error occurs, the server sends a service error response to the client. This topic provides a service error response example and describes parameters in the response.

Example

```
<requestError>
   <serviceException>
      <messageId>SVC0280</messageId>
      <text>Message too long.SLA level is %1, maximum length is %2 characters, actual length is %3
characters.</text>
      <variables>System self</variables>
      <variables>1200</variables>
      <variables>1796</variables>
   </serviceException>
</requestError>
```

Parameter Description

Table 4-1 describes parameters in a service error response.

Table 4-1 Parameters in a service error response

Parameter	Type	Level of Requirement	Description
messageId	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
text	xsd:string	Mandatory	Error description. The value can contain the variable *%#* in definition. [Example] Waiting for response timed out, message type is OutwardGetLocReq.
variables	xsd:string [0..unbounded]	Optional	Value of the variable defined in the value of **text**. [Example] OutwardGetLocReq

4.2 Policy Error Response

A policy error is caused by service level agreement (SLA) violation. When a policy error occurs, the server sends a policy error response to the client. This topic provides a policy error response example and describes parameters in the response.

Example

```
<requestError>
   <policyException>
      <messageId>POL0280</messageId>
      <text>Message too long.SLA level is %1, maximum length is %2 characters, actual length is %3 
characters.</text>
      <variables>System self</variables>
      <variables>1200</variables>
      <variables>1796</variables>
   </serviceException>
</requestError>
```

API Error Responses

Parameter Description

Table 4-2 describes parameters in a policy error response.

Table 4-2 Parameters in a policy error response

Parameter	Type	Level of Requirement	Description
messageId	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
text	xsd:string	Mandatory	Error description. The value can contain the variable *%#* in definition. [Example] GroupAddr is not supported
variables	xsd:string [0..unbounded]	Conditional	Value of the variable defined in the value of **text**. [Example] GroupAddr

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

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