Service Exposure OneAPI Payment Development

Ericsson Composition Engine

USER GUIDE

Copyright

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.

Contents

1 About this Document 1

1.1 Revision Information 1

1.2 Purpose 1

1.3 Target Group 1

1.4 Prerequisites 1

2 OneAPI Payment Overview 3

3 Security and Authentication 5

4 Resources and URIs 7

5 Integration with ECE Authorization Server 9

5.1 Scope 9

5.2 Integration Point 10

6 Interface: Charging or Refunding 11

6.1 Operation: Charge User 11

6.2 Operation: Refund User 12

7 Interface: Reserving Amount Charging 15

7.1 Operation: Reserve Amount 15

7.2 Operation: Reserve Additional Amount 17

7.3 Operation: Charge Reservation 18

7.4 Operation: Release Reservation 20

8 Resource States 23

9 Response Codes and Exceptions 25

9.1 Response Codes 25

9.2 Exceptions 25

Glossary 31

Reference List 33

1 About this Document

This chapter contains the following parts:

- Revision information
- Purpose
- Target audience
- Prerequisites
- Typographic conventions

1.1 Revision Information

The following table shows the changes in recent revisions. Other than editorial changes, this document has been revised as follows:

Table 1 Revision Information

Revision	Section	Change
J	Section 5 on page 9	A new chapter.

1.2 Purpose

This document describes how to develop an application which is compatible with Service Exposure OneAPI Payment interface. The full specification of OneAPI Payment interface and examples with detailed explanation are provided in this document.

1.3 Target Group

This document is intended for application developers.

1.4 Prerequisites

It is assumed that the reader of this document is familiar with the following knowledge:

- Parlay REST API
- HTTP protocol

- RESTful HTTP
- ECE Service Exposure structure

2 OneAPI Payment Overview

The OneAPI Payment interface is used to charge mobile subscribers for use of Web application or content. This API is used to directly charge a user based on its consent. It also supports reserving funds from a user account for subsequent charging. This can offer additional content within the same session to the user and then apply one charge for the total, while ensuring they have sufficient funds to pay. This can be especially important for pre-pay users with limited credits.

OneAPI is a profile (subset) of **Parlay REST v1.0**. The full specifications and guidelines may be found at: ttp://www.openmobilealliance.org/Technical/current_releases.aspx.

For information about **Payment RESTful NetAPI**, you can visit: http://oneapi.gsma.com/Payment-restful-netapi/.

The supportability of the Payment RESTful NetAPI operations in OneAPI Payment service are listed in the following table:

Table 2 Supported Operations in OneAPI Payment

Resource URI	HTTP Method	Operation	Supportability in OneAPI
http://example.com/oneapi/payment/1/{endUserId}/transactions/amount	POST	Charge a user	Yes
		Refund a user	Yes
http://example.com/oneapi/payment/1/{endUserId}/transactions/amountReservation	POST	Reserve an amount to charge	Yes
http://example.com/oneapi/payment/1/{endUserId}/transactions/amountReservation/{transactionId}	POST	Reserve an additional amount (such as extending an existing reservation)	Yes
		Charge against the reservation	Yes
		Release the reservation	Yes

Note: It might be different on the supported operations between OneAPI and the RESTful NetAPI specification. For detailed operation supportability in OneAPI Payment service, see the corresponding section in the following chapters.

3 Security and Authentication

A server-side certificate is used to secure the HTTPS connection and to confirm the server identity. The client is authenticated with HTTP basic authentication or with OAuth authentication.

HTTP Basic Authentication

When the client uses HTTP basic authentication, add an Authorization header in the request as follows:

```
POST /oneapi/payment/1/tel:+8613580551660/transactions/amount HTTP/1.0
Authorization: Basic YXBwMUBwYXJ0bmVyMjphdXRob2s=
Content-Type: application/json
Accept: application/json
Content-Length: 173
Host: <DP-Exposure-Traffic>:27010
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.2 (java 1.5)
{"endUserId":"tel:+8613580551660",
 "transactionOperationStatus":"Charged",
 "description":"ECE SIG oneapi payment",
 "currency":"SEK",
 "amount":"10.0",
 "referenceCode":"REF-12345"
}
```

The format of the Authorization header is as follows:

`Authorization: Basic` *`<base64 encoded (user-pass)>`*

The following parameters are used in the authentication.

Table 3 Authentication Parameters

Parameter	Value
`user-pass`	`userid:password`
`userid`	*`<application-id>@<partner-id>`*
`password`	`TEXT`

For example, if the `userid` is `app1@partner1` and the `password` is `authok`, then the header including the base64 encoded `user-pass` is the following: `Authorization: Basic cGFydG5lcjFAYXBwMTphdXRob2s=`

For more information, see Ericsson Composition Engine, System Administration Guide.

OAuth Authentication

For OAuth authentication, only the client credential flow is supported for the client to get the access token. The OAuth authorization flow which uses authorization code to get access token is not supported.

For more information about client integration, see Authorization Integration Guide.

When the client uses OAuth authentication, add an Authorization header in the request as follows:

```
POST /oneapi/payment/20/tel%3A%2B8613500000000/transactions/amount
Authorization: Bearer MkdPhwf2FCVm5rijcaCi
Host: <DP-Exposure-Traffic>:27010
Accept: application/json
Content-Type: application/x-www-form-urlencoded
endUserId=tel%3A%2B8613500000000&transactionOperationStatus=
Charged&referenceCode=REF-12345&description=
ECE_SIG_oneapi_payment&code=CODE-1
```

The Authorization header contains `Bearer`, followed by an identifier from the OAuth server.

If the application identifier is a globally unique access key in a Business to Business (B2B) scenario, add an Access Key header after the Authorization header in the request. The Access Key header contains the application access key as follows:

```
AccessKey: <access key of application>
```

Following is an example of the request containing the Access Key header:

```
POST /oneapi/payment/20/tel%3A%2B8613500000000/transactions/amount
Authorization: Bearer MkdPhwf2FCVm5rijcaCi
AccessKey: 1316kj
Host: <DP-Exposure-Traffic>:27010
Accept: application/json
Content-Type: application/x-www-form-urlencoded
endUserId=tel%3A%2B8613500000000&transactionOperationStatus=
Charged&referenceCode=REF-12345&description=
ECE_SIG_oneapi_payment&code=CODE-1
```

4 Resources and URIs

OneAPI Payment is a RESTful API. A RESTful API use HTTP commands `POST`, `GET`, `PUT`, and `DELETE` to operate on a resource at the server. The resource is addressed by a URI. What is returned by the server is a representation of that resource depending on its current state.

HTTP `POST`, `GET` and `DELETE` commands are used in OneAPI Payment.

Representation Formats

The Payment API supports `application/x-www-form-urlencoded` and `application/json` content type for `POST` operations. The response content type is normally `application/json`.

Note: The `POST` operations is shown as `application/x-www-form-urlencoded` in this document. However `JSON` can also be used in the request body.

5 Integration with ECE Authorization Server

This section introduces how to integrate OneAPI Payment service with ECE Authorization Server.

5.1 Scope

To integrate OneAPI Payment service with ECE authorization server, configure the related scopes.For detailed information of Scope Configuration and Scope Configuration for OneAPI. Refer to Authorization Server Configuration.

When ECE authorization server grants the scope to payment service, the scope in the authorization request is simple scope or parameterized scope. For detailed information of the Authorization, refer to Authorization Integration User Guide.

Only `amount`, `currency`, and `code` are currently supported in the parameterized scope. When a OneAPI Payment request is received, if the parameters exist, validate them with the granted value in parameterized scope.

The pattern of parameterized scope is the scope key followed by one or more parameterized key-value pairs. Each parameterized key-value pair is contained within parentheses, and all the pairs are contained within square brackets.

The pattern of parameterized scope is:

```
scopeKey[(paramaterName1:parameterValue1)(parameterName2:parameterValue2)]
```

The following table shows the scopes supported by OneAPI Payment interfaces.

Table 4 Scopes Supported by OneAPI Payment Interface

Applicable interface	Scope Key	Parameterized Scope Example
Charging interface with amount and currency in request.	`payment.chargerefund`	`payment.chargerefund[(amount:12.5)(currency:USD)]`
Charging interface with code in request	`payment.debitcode`	`payment.debitcode[(code:CODE-1)]`
Reserving amount charging interface.	`payment.reservation`	`payment.reservation[(amount:12.5)(currency:USD)]`

Note: Refunding interfaces are not applicable to be used with OAuth authorization.

5.2 Integration Point

If OneAPI Payment service integrates with ECE authorization server, an `ownerId` is included within the normal token given by authorization server to grant the access to resource when using authorization code grant flow. The `ownerId` is a unique ID of the end user, used as the resource owner identity in the interaction with the authorization server.

The integration point `ipReceivedPaymentRequestPreSubscription`, which can be invoked by each payment operation, needs to convert the `endUserId` from OneAPI Payment request to an `ownerId`. This `ownerId` must be validated with the one in the normal token. Only if they are matched, the operation can be allowed to continue with the business. If OneAPI Payment service sends request with a trusted client token which does not contain an `ownerId`, the validation of `ownerId` is bypassed.

For detailed information on integration point, refer to Payment Service Integration in Integration Point Development.

6 Interface: Charging or Refunding

This interface defines following operations:

- Charge user.
- Refund user.

6.1 Operation: Charge User

The Charging or Refunding interface can be used to charge an end user.

6.1.1 Resource and URI

Table 5 Resource and URI

Resource URI	HTTP Method	Operation
`https://<serverRoot>/oneapi/payment/1/<endUserId>/transactions/amount`[1]	`POST`	Charge or refund a user

(1) <serverRoot> is replaced by the hostname of the OneAPI server that is being accessed.

6.1.2 Request

The following is an example of the request to charge an amount to the user bill.

```
POST /oneapi/payment/1/tel%3A%2B8613500000000/transactions/amount
Authorization: Bearer MkdPhwf2FCVm5rijcaCi
Host: ece.example.com:38080
Accept: application/json
Content-Type: application/x-www-form-urlencoded

endUserId=tel%3A%2B8613500000000&transactionOperationStatus=
Charged&referenceCode=REF-12345&description=
ECE_SIG_oneapi_payment&code=CODE-1
```

The following table shows the parameters for charging user request.

Table 6 Request Parameters for Charging User

Parameter	Location	Type	Description	Mandatory
`endUserId`	URL and payload	URL-escaped	It is the identity of the end user to be charged. `tel URI` is used as the user identity. Only global number is supported. “tel:” scheme must be given. For example, `tel%3A%2B8613500000000` The `endUserId` value in URL and payload must be same.	Yes

Parameter	Location	Type	Description	Mandatory
`referenceCode`	Payload	String	It is unique for each charge event. It is the reference for reconciliation purposes. The operator should include it in reports so that you can match their view of what has been sold with yours by matching the `referenceCode`.	Yes
`transactionOperationStatus`	Payload	Enumeration	This indicates the desired resource state. It is “Charged” in this case. See Resource States for further explanation.	Yes
`description`	Payload	String	It is the human-readable text to appear on the bill. Then user can easily see what they bought.	Yes
`currency`	Payload	String	It is the 3-figure code as per ISO 4217.	Either `amount` AND `currency`OR `code` must be provided.
`amount`	Payload	Decimal	It can be a whole number or decimal.	
`code`	Payload	String	It is a code provided by the OneAPI implementation. It is used to reference an operator price point.	

6.1.3 Response

The following is an example of the response for charging user.

```
201 Created
Content-Type: application/json
Date: Wed, 08 Aug 2012 02:26:50 GMT

{"amountTransaction":{
       "endUserId":"tel:+8613500000000",
       "paymentAmount":{
               "chargingInformation":{
                        "description":"ECE_SIG_oneapi_payment",
                        "code":"CODE-1"}
                        },
               "referenceCode":"REF_12345",
               "transactionOperationStatus":"Charged"
}}
```

The following table shows the response parameters.

Table 7 Response Parameters for Charging User

Parameter	Description
`transactionOperationStatus`	“Charged” confirms that the resource state is **charged** . See Table 18.
`totalAmountCharged`	It confirms how much was charged to the user. It does not exist if `amount` is not provided in request.
`<other information>`	The same information in request, for confirmation.

6.2 Operation: Refund User

The Charging or Refunding interface can be invoked to refund user.

6.2.1 Resource and URI

The resource and URI is same as Table 5.

6.2.2 Request

The following is an example of the request to refund user.

```
POST /oneapi/payment/1/tel%3A%2B8613500000000/transactions/amount
Authorization: Bearer MkdPhwf2FCVm5rijcaCi
Host: ece.example.com:38080
Accept: application/json
Content-Type: application/x-www-form-urlencoded

endUserId=tel%3A%2B8613500000000&transactionOperationStatus=
Refunded&referenceCode=REF-12345&description=
ECE_SIG_oneapi_payment&amount=10.0&currency=SEK
```

The request parameters for refunding user are similar to charging user, as shown in Request Parameters for Charging User.

6.2.3 Response

The following is an example of the response to refunding the user.

```
201 Created
Content-Type: application/json
Date: Wed, 08 Aug 2012 02:29:23 GMT

{"amountTransaction":{
        "endUserId":"tel:+8613500000000",
        "paymentAmount":{
              "chargingInformation":{
                     "amount":"10.0",
                     "currency":"SEK",
                     "description":"ECE_SIG_oneapi_payment"
                     },
              "totalAmountRefunded":"10.0"
              },
              "referenceCode":"REF_12345",
              "transactionOperationStatus":"Refunded "
}}
```

The following table shows the response parameters.

Table 8 Response Parameters for Refunding User

Parameter	Description
`transactionOperationStatus`	“Refunded” confirms that the resource state is **Refunded** . See Table 18.
`totalAmountRefunded`	It confirms how much was refunded to the user. It does not exist if `amount` is not provided in request.
`<other information>`	The same information in request, for confirmation.

7 Interface: Reserving Amount Charging

This interface defines following operations:

- Reserve amount.
- Reserve additional amount.
- Charge reservation.
- Release reservation.

7.1 Operation: Reserve Amount

The Amount Reservation Charging interface can be used to reserve an amount for subsequent charging to a user.

7.1.1 Resource and URI

Table 9 Resource and URI

Resource URI	HTTP Method	Operation
`https://<serverRoot>/oneapi/payment/1/<endUserId>/transactions/amountReservation`	POST	Reserve an amount for subsequent charging to a user.

7.1.2 Request

The following is an example of the request to reserve an amount for subsequent charging to a user.

```
POST /oneapi/payment/1/tel%3A%2B8613500000000/transactions/amountReservation
Authorization: Bearer MkdPhwf2FCVm5rijcaCi
Host: ece.example.com:38080
Accept: application/json
Content-Type: application/x-www-form-urlencoded

endUserId=tel%3A%2B8613500000000&transactionOperationStatus=
Reserved&referenceCode=REF-12345&description=
ECE_SIG_oneapi_payment&amount=10.0&currency=SEK
```

The following table shows the parameters for reserving amount request.

Table 10 Request Parameters for Reserving Amount

Parameter	Location	Type	Description	Mandatory
`endUserId`	URL and payload	URL-escaped	It is the identity of the end user to be charged. `tel URI` is used as the user identity. Only global number is supported. “tel:” scheme must be given. For example, `tel%3A%2B8613500000000` The `endUserId` value in URL and payload must be same.	Yes
`referenceCode`	Payload	String	It is unique for each charge event. It is the reference for reconciliation purposes. The operator should include it in reports so that you can match their view of what has been sold with yours by matching the `referenceCodes`.	Yes
`transactionOperationStatus`	Payload	Enumeration	This indicates the desired resource state. It is “Reserved” in this case. See Resource States.	Yes
`description`	Payload	String	It is the human-readable text to appear on the bill. Then user can easily see what they bought.	Yes
`currency`	Payload	String	It is the 3-figure code as per ISO 4217. All the subsequent operations use the currency specified in this Reserve Amount operation.	Yes
`amount`	Payload	Decimal	It can be a whole number or decimal.	Yes

7.1.3 Response

The following is an example of the response to reserving amount.

```
201 Created
Content-Type: application/json
Date: Wed, 08 Aug 2012 02:29:23 GMT
Location:https://ece.example.com:38080/oneapi/payment/1/tel:+8613500000000/
transactions/amountReservation/1208081000000001005

{"amountReservationTransaction":{
       "endUserId":"tel:+8613500000000",
       "paymentAmount":{
            "chargingInformation":{
                     "amount":"10.0",
                     "currency":"SEK",
                     "description":"ECE_SIG_oneapi_payment"
                     }
            },
            "referenceCode":"REF_12345",
            "resourceURL":"https://ece.example.com:38080/oneapi/payment/1/
            tel:+8613500000000/transactions/amountReservation/
            1208081000000001005”,
            "transactionOperationStatus":"Reserved"
}}
```

The following table shows the response parameters.

Table 11 Response Parameters for Reserving Amount

Parameter	Description
`Location`	The Location HTTP header provides the URI of the created resource. The final part of this URI ("1208081000000001005") is the `transactionId` which uniquely identifies this charge transaction.
`transactionOperationStatus`	"Reserved" confirms that the resource state is **Reserved** . See Table 18.
<other information>	The same information in request, for confirmation.

7.2 Operation: Reserve Additional Amount

The Reserving Amount Charging interface can be invoked to reserve an additional amount.

7.2.1 Resource and URI

Table 12 Resource and URI

Resource URI	HTTP Method	Operation
`https://`*`<serverRoot>`*`/oneapi/payment/1/`*`<endUserId>`*`/transactions/amountReservation/`*`<transactionId>`*	`POST`	Reserve an additional amount for subsequent charging to a user.

7.2.2 Request

The following is an example of the request to reserve an additional amount.

```
POST /oneapi/payment/1/tel%3A%2B8613500000000/transactions/amountReservation/
1208081000000001005
Authorization: Bearer MkdPhwf2FCVm5rijcaCi
Host: ece.example.com:38080
Accept: application/json
Content-Type: application/x-www-form-urlencoded

endUserId=tel%3A%2B8613500000000&transactionOperationStatus=Reserved&
referenceCode=REF-12345&description=ECE_SIG_oneapi_payment&amount=5.0
```

The following table shows the parameters for reserving additional amount request.

Table 13 Request Parameters for Reserving Additional Amount

Parameter	Location	Type	Description	Mandatory
`endUserId`	URL and payload	URL-escaped	It is the identity of the end user to be charged. `tel URI` is used as the user identity. Only global number is supported. "tel:" scheme must be given. For example, `tel%3A%2B8613500000000` The `endUserId` value in URL and payload must be same.	Yes

Parameter	Location	Type	Description	Mandatory
`referenceCode`	Payload	String	It is unique for each charge event. It is the reference for reconciliation purposes. The operator should include it in reports so that you can match their view of what has been sold with yours by matching the `referenceCodes`.	Yes
`transactionOperationStatus`	Payload	Enumeration	This indicates the desired resource state. It is “Reserved” in this case. See Resource States.	Yes
`description`	Payload	String	It is the human-readable text to appear on the bill. Then user can easily see what they bought.	Yes
`amount`	Payload	Decimal	It can be a whole number or decimal.	Yes

7.2.3 Response

The following is an example of response to reserving additional amount.

```
200 OK
Content-Type: application/json
Date: Wed, 08 Aug 2012 02:27:33 GMT

{"amountReservationTransaction":{
      "endUserId":"tel:+8613500000000",
      "paymentAmount":{
            "chargingInformation":{
                   "amount":"10.0",
                   "description":"ECE_SIG_oneapi_payment"
                   }
            },
            "referenceCode":"REF_12345",
            "resourceURL":"https://ece.example.com:38080/oneapi/payment/1/
            tel:+8613500000000/transactions/amountReservation/
            1208081000000001005”,
            "transactionOperationStatus":"Reserved"
}}
```

The following table shows the response parameters.

Table 14 Response Parameters for Reserving Additional Amount

Parameter	Description
`transactionOperationStatus`	“Reserved” confirms that the resource state is **Reserved** . See Table 18.
`<other information>`	The same information in request, for confirmation.

7.3 Operation: Charge Reservation

The Reserving Amount Charging interface can be invoked to charging against the reservation.

7.3.1 Resource and URI

Table 15 Resource and URI

Resource URI	HTTP Method	Operation
`https://<serverRoot>/oneapi/payment/1/<endUserId>/transactions/amountReservation/<transactionId>`	POST	Charge against the reservation.

7.3.2 Request

The following is an example of request to charge against the reservation.

```
POST /oneapi/payment/1/tel%3A%2B8613500000000/transactions/amountReservation/
1208081000000001005
Authorization: Bearer MkdPhwf2FCVm5rijcaCi
Host: ece.example.com:38080
Accept: application/json
Content-Type: application/x-www-form-urlencoded

endUserId=tel%3A%2B8613500000000&transactionOperationStatus=
Charged&referenceCode=REF-12345&description=
ECE_SIG_oneapi_payment&amount=15.0
```

The request parameters for charging reservation are similar to reserving additional amount, as shown in Request Parameters for Reserving Additional Amount.

7.3.3 Response

Following is an example of the response to charging against the reservation.

```
200 OK
Content-Type: application/json
Date: Wed, 08 Aug 2012 02:31:17 GMT

{"amountReservationTransaction":{
      "endUserId":"tel:+8613500000000",
      "paymentAmount":{
            "chargingInformation":{
                   "amount":"15.0",
                   "description":"ECE_SIG_oneapi_payment"
                   }
            },
            "referenceCode":"REF_12345",
            "resourceURL":"https://ece.example.com:38080/oneapi/payment/1/
            tel:+8613500000000/transactions/amountReservation/
            1208081000000001005”,
            "transactionOperationStatus":"Charged"
}}
```

The response parameters for charging reservation are similar to reserving additional amount, as shown in Table 14.

7.4 Operation: Release Reservation

The Reserving Amount Charging interface can be used to release the reservation.

7.4.1 Resource and URI

Table 16 Resource and URI

Resource URI	HTTP Method	Operation
`https://<serverRoot>/oneapi/payment/1/<endUserId>/transactions/amountReservation/<transactionId>`	`POST`	Release the leftover reservation to user.

7.4.2 Request

The following is an example of the request to release the reservation.

```
POST /oneapi/payment/1/tel%3A%2B8613500000000/transactions/amountReservation/
1208081000000001005
Authorization: Bearer MkdPhwf2FCVm5rijcaCi
Host: ece.example.com:38080
Accept: application/json
Content-Type: application/x-www-form-urlencoded

endUserId=tel%3A%2B8613500000000&transactionOperationStatus=
Released&referenceCode=REF-12345
```

The following table shows the parameters for releasing reservation request.

Table 17 Request Parameters for Releasing Reservation

Parameter	Location	Type	Description	Mandatory
`endUserId`	URL and payload	URL-escaped	It is the identity of the end user to be charged. `tel URI` is used as the user identity. Only global number is supported. “tel:” scheme must be given. For example, `tel%3A%2B8613500000000` The `endUserId` value in URL and payload must be same.	Yes
`referenceCode`	Payload	String	It is unique for each charge event. It is the reference for reconciliation purposes. The operator need to include it in reports so that you can match their view of what has been sold with yours by matching the `referenceCodes`.	Yes
`transactionOperationStatus`	Payload	Enumeration	This indicates the desired resource state. It is “Released” in this case. See Resource States.	Yes

7.4.3 Response

The following is an example of the response to releasing the reservation.

```
200 OK
Content-Type: application/json
Date: Wed, 08 Aug 2012 02:31:17 GMT

{"amountReservationTransaction":{
      "endUserId":"tel:+8613500000000",
      "paymentAmount":{
            "chargingInformation":{}
            },
            "referenceCode":"REF_12345",
            "resourceURL":"https://ece.example.com:38080/oneapi/payment/1/
            tel:+8613500000000/transactions/amountReservation/
            1208081000000001005”,
            "transactionOperationStatus":"Released"
}}
```

The response parameters for releasing reservation are similar to reserving additional amount, as shown in Table 14.

8 Resource States

The client passes the `transactionOperationStatus` in the request body, so that the resource can be placed into a desired state. The server either confirms this desired state in the `transactionStatus` response field, or shows a failure state as listed below.

The following table lists the available status.

Table 18 Resource States

Status	Description
`Charged`	Charge successfully.
`Refunded`	Refund successfully.
`Reserved`	The reservation is created successfully.
`Charged`	The reservation has been charged successfully
`Released`	The reservation is ended

9 Response Codes and Exceptions

This chapter describes some response codes and exceptions.

9.1 Response Codes

The following table shows some HTTP response codes and their indications.

Table 19 Response Codes

Response Codes	Indication
201	Created. The charge or refund is successful.
400	Bad request. Check the error message and correct the request syntax.
401	Authentication failure. Check the authentication requirements from your OneAPI provider.
403	Forbidden. Please provide authentication credentials.
404	Not found: mistake in the host or path of the service URI.
405	Method not supported. For example, only `GET` is supported in OneAPI Location v1.
503	Server busy and service unavailable. Please retry the request.

More details please visit: http://www.ietf.org/rfc/rfc2616.txt

9.2 Exceptions

Following is an example of exception.

```
400 Bad Request
Date:  Tue, 17 Jul 2012 09:33:49 GMT
Content-Type:  application/json

{"requestError": {
    "serviceException": {
        "messageId": "SVC0002",
        "text": " Invalid input value for message part %1",
        "variables": " tel:+016309700000"
    }
}}
```

Following table shows the meaning of the parameters in the exception example.

Table 20 Parameters Description

Parameter	Description
`400`	Error code. It indicates either a service exception or a policy exception.
`requestError`	Exception reason. It contains `serviceException` and `policyException`. They share the same body `messageId`, `text`, and `variables`.

Parameter	Description
`serviceException`	The reason why the service can not accept the request. In this exception example, the phone number is too long.
`policyException`	It shows that the request syntax is valid, however an operator policy is broken. For example, the amount to be charged in the request exceeds a limit that is set.
`messageId`	The identifier of the exception.
`text`	The description for the exception.
`variables`	It indicates any specific cause of the error.

9.2.1 Service Exceptions

This section lists the available service exceptions. The following table shows error codes, the possible reasons why the exception occurred, and the possible solutions.

Table 21 Service Exceptions

ID	Exception Text	Variables	HTTP Code
`SVC0001`	A service error occurred. Error code is %1	**%1** – explanation of the error[1]	400 Bad Request
`SVC0002`	Invalid input value for message part %1	**%1** – the part of the request that is invalid	400 Bad Request
`SVC0003`	Invalid input value for message part %1, valid values are %2.	**%1** – message part **%2** – list of valid values	400 Bad Request
`SVC0004`	No valid addresses provided in message part %1. Addresses means phone numbers and some like that. So this exception is thrown if the MSISDN does not follow the correct format (For example,+441234567890). The MSISDN includes URL encoding where necessary. If the address is part of the resource URL, the status code 404 is used. Otherwise the status code 400 is used.	**%1** – message part	404 Not Found 400 Bad Request
`SVC0005`	Correlator %1 specified in message part %2 is a duplicate. This exception is thrown if the `clientCorrelator` has already been used, for example, when creating a previous resource.	**%1** – correlator **%2** – message part	409 Conflict
`SVC0006`	Group %1 in message part %2 is not a valid group.	**%1** – identifier for the invalid group **%2** – message part	400 Bad Request
`SVC0007`	Invalid charging information	None	400 Bad Request
`SVC0008`	Overlapped criteria %1	%1 – Message Part with the overlapped criteria	400 Bad Request
`SVC1000`	No resources. This exception is thrown if there are no server resources available to process the request.	None	503 Service unavailable
`SVC0270`	Charging operation failed. The charge was not applied.	None	400 Bad Request

ID	Exception Text	Variables	HTTP Code
`SVC0271`	Refunds not supported – %1	%1 – guidance from the implementor on what to do instead	400 Bad Request
`SVC0273`	Refund failed – %1	%1 – the reason why the refund failed. The possible reasons include following: • User do not accept the refund • The `originalServerReference` code is invalid • The refund request is for an amount greater than the original charge	400 Bad Request

[1] For a reference to the error codes, see Section 9.2.3 on page 28.

9.2.2 Policy Exceptions

This section lists the available policy exceptions. The following table shows error codes, the possible reasons why the exception occurred, and the possible solutions.

Table 22 Policy Exceptions

ID	Exception Text	Variables	HTTP Code
`POL0001`	A policy error occurred. Error code is %1	**%1** – explanation of the error[1]	403 Forbidden
`POL0002`	Privacy verification fails for address %1, request is refused	**%1** – address privacy verification fails for	403 Forbidden
`POL0003`	Too many addresses specified in message part %1	**%1** – message part	403 Forbidden
`POL0004`	Unlimited notification request not supported	None	403 Forbidden
`POL0005`	Too many notifications requested	None	403 Forbidden
`POL0006`	Group specified in message part %1 not allowed[2]	**%1** – message part	403 Forbidden
`POL0007`	Nested groups specified in message part %1 not allowed[3]	**%1** – message part	403 Forbidden
`POL0008`	Charging is not supported	None	403 Forbidden
`POL0009`	Invalid frequency requested	None	403 Forbidden

ID	Exception Text	Variables	HTTP Code
`POL0010`	Requested information unavailable as the retention time interval is expired.	None This exception is thrown if, for example, the delivery status of an old SMS is requested, which means that the server no longer maintains the resource. In case the information that has become unavailable is addressed by a resource URL, the following applies: • If the resource URL refers to a resource that has existed in the past and the server is aware of that fact, the status code 410 is used. • If the server is not aware, the status code 404 is used. • In all other cases, the status code 403 is used.	403 Forbidden 404 Not Found 410 Gone
`POL0011`	Media type not supported	None	403 Forbidden 406 Not Acceptable
`POL0012`	Too many description entries specified in message part %1	**%1** – message part	403 Forbidden
`POL0013`	Duplicated addresses %1	%1 – duplicated addresses	400 Bad Request
`POL1009`	User has not been provisioned for %1.	**%1** – name of the service	403 Forbidden
`POL1010`	User has been suspended from %1.	**%1** – name of the service	403 Forbidden
`POL0251`	Chargeable amount exceeded – %1	**%1** – indicate whether the one-off charge limit, or the cumulative charge for a given time period, is reached.	403 Forbidden

(1) For a reference to the error codes, see Section 9.2.3 on page 28.

(2) Group means an address that refers to more than one end user.

(3) Group means an address which refers to more than one end user. Groups cannot contain addresses which are themselves groups.

9.2.3 Error Codes

This section gives a reference to the error codes in service exception `SVC0001` in Table 21 and in policy exception `POL0001` in Table 22.

Table 23 Error Code Reference

Error Code	Indication
`1`	Unexpected network or system related errors
`2`	Request time out
`3`	Service Capability is inactive.

Error Code	Indication
4	Could not contact Rule engine
5	Subscription limitations are violated
6	Throttling rejected on service capability level
7	Distribution list adapter is not deployed
8	There is no valid address in Distribution List.
20	SLA limitations violated
21	Application or Service Provider is inactive or does not exist.
22	Too many addresses are specified.
23	Service activation number does not exist in the allowed list.
24	Error during application of SLA enforcement rule
26	Throttling rejected on application or service provider level
27	All addresses were rejected by integration adapter.
28	If the SLA property `DelayToleranceNoDelayAction` violated and the violation action contains `reject`.
29	If the SLA property `DelayToleranceLowDelayAction` violated and the violation action contains `reject`.
30	If the SLA property `DelayToleranceDelayTolerantAction` violated and the violation action contains `reject`.
31	If the SLA property `MinGetLoactionRequestAccuracy` violated and the corresponding violation action contains `reject`.
40	Southbound Resource Specific Error
42	Request Specific Error
44	Network Traffic Error
46	Vendor Specific Error
47	GMLC Error
48	The value of response time required from the request is `NO_DELAY` but the system cache is disabled.
49	The value of response time required from the request is `NO_DELAY` but the value of type of location is `CURRENT` or `CURRENT_LAST`.
50	The value of the type of location from request is `LAST` but the system cache is disabled.
51	Congestion in GMLC
52	Not in cache
53	The value of the type of the session is not "DIAL".
54	The coordinate reference system value in request is not "WGS84".
55	Resource related service enabler don't have cache.
56	GMLC not reachable
57	ADC response error
58	Device capabilities not found session error
61	The user is not allowed to perform this operation.
70	Operation not allowed in Southbound Resource

Error Code	Indication
71	Requester or GroupOwner is belong to SP, for ParlayX ALM.
72	Requester is unauthorized in southbound resource.
80	Too many subscriptions for a watcher or a presentity (ParlayX Presence).
81	Subscription is expired or not existing.
60001	The reservation expires.
60003	The communication to the southbound enabler fails.
60005	The SLA property MaxAmountOfPaymentRequest is violated. For the detailed information, refer to the document Service Exposure Service Properties - OneAPI Payment Web Service, Reference [1].
60006	The SLA property MinAmountOfPaymentRequest is violated. For the detailed information, refer to the document Service Exposure Service Properties - OneAPI Payment Web Service, Reference [1].
60007	The application is not allowed to use this operation. This is enforced by the SLA property AllowChargeAmount, AllowRefundAmount or AllowReserveAmountCharging. For the detailed information, refer to the document Service Exposure Service Properties - OneAPI Payment Web Service, Reference [1].
60011	The southbound enabler reports the error.
60012	The balance is not enough.
60013	Error while creating or parsing data to/from enabler.
60014	OwnerId is violated.
9000 - 9499	Error occurs in the integration adapter, used for service exceptions.
9500 - 9999	The failure occurs in the integration adapter, used for policy exceptions.

Glossary

API
Application Programming Interface

B2B
Business to Business

ECE
Ericsson Composition Engine

HTTP
Hypertext Transfer Protocol

HTTPS
HTTP over SSL

ID
Identification

OAuth
Open standard for Authorization

OneAPI
Open Network Enablers API

Payment
Short Message Service

REST
Representational State Transfer

SLA
Service Level Agreement

URI
Uniform Resource Identifier

URL
Uniform Resource Locator

Reference List

Ericsson Documents

[1] *Service Exposure Service Properties - OneAPI Payment Web Service*, 40/1553-CXP 904 0189 Uen