

Subscription Management API

Ericsson Device Connection Platform

INTERWORK DESCRIPTION

Copyright

© Ericsson AB 2018, 2019. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

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	Introduction	1
1.1	Prerequisites	1
1.2	Related Information	1
2	Revision Information	3
3	Interface Overview	4
3.1	Using the API	4
3.2	Versioning	4
3.3	Encapsulation	4
3.4	Security	4
3.4.1	Authentication	5
3.4.2	Authorization	5
3.4.3	Encryption	5
4	Endpoints and Operations	6
4.1	Login	7
4.1.1	POST /login	7
4.1.2	POST /login-form	8
4.1.3	POST /token-reissue	8
4.2	SubscriptionOrder	9
4.2.1	GET /subscriptions/order/metadata	9
4.2.2	POST /subscriptions/order	10
4.2.3	GET /subscriptions/stockOrders/metadata	12
4.2.4	POST /subscriptions/stockOrders	12
4.2.5	GET /subscriptions/pendingOrders	13
4.2.6	POST /subscriptions/orders/{orderId}:approve	14
4.2.7	DELETE /subscriptions/orders/{orderId}	14
4.3	SubscriptionInventory	15
4.3.1	POST /subscriptions/details	15
4.3.2	GET /subscriptions/details/{key}/{value}	16
4.3.3	GET /subscriptions/{id}/history	19
4.3.4	GET /subscriptions/requests/{requestId}	20
4.3.5	POST /subscriptions/requests/find	22
4.3.6	GET /subscriptions/returnedMsisdns	22
4.3.7	POST /subscriptions/{imsi}:reassignMsisdn	23
4.3.8	POST /subscriptions/{id}:assignImei	24
4.3.9	POST /subscriptions/{subscription_id}:suspendSubscription	25
4.3.10	POST /subscriptions/{subscription_id}:changeMsisdn	25
5	HTTP Response Codes	27



Reference List

28



1 Introduction

This document describes the Device Connection Platform (DCP) Subscription Management API. This API offers a web application interface to perform operations related to the inventory and management of subscriptions.

The DCP Subscription Management API is based on the Representational State Transfer (REST) interface.

From this point on, the DCP Subscription Management API is referred to as **API** in this document.

Scope

This document covers the following topics:

- An overview of the API
- A detailed description of all operations offered by the API
- A description of the possible HTTP response codes in the API

Target Groups

This document is intended for users of the DCP subscription management services who want to know how to use the API.

1.1 Prerequisites

In this document, it is assumed that the user is familiar with the following:

- Java Script Object Notation (JSON)
- Basic understanding of REST web services

1.2 Related Information

For the full list of permissions impacting the Subscription Management API, see the following document:

- DCP Service Portal Operator Guide, Reference [3]

The definition and explanation of acronyms and terminology and typographic conventions can be found in the following documents:

- DCP Glossary of Terms and Acronyms, Reference [1]
- Typographic Conventions, Reference [2]



Note: There is no one-to-one correspondence between the features available through the Service Portal and the Subscription Management API.



2 Revision Information

Table 1 shows the changes in the recent revisions of the Subscription Management API. Other than editorial changes, this document has been revised as follows:

Table 1 Revision Information

Section	Title	Change
DCP release 19B3, document revision U		
Section 4.3.9 on page 24	POST /subscriptions/{subscription_id}:suspendSubscription	The following endpoint is introduced: <ul style="list-style-type: none"> POST /subscriptions/{subscription_id}:suspendSubscription
Section 4.3.10 on page 25	POST /subscriptions/{subscription_id}:changeMsisdn	The following endpoint is introduced: <ul style="list-style-type: none"> POST /subscriptions/{subscription_id}:changeMsisdn
DCP release 19B2, document revision T		
Section 4.3.7 on page 23	POST /subscriptions/{imsi}:reassignMsisdn	Information added about reassigning MSISDN if a subscription has consumer connectivity V2 enabled.
DCP release 19B1, document revision N		
Section 4.3.1 on page 15	POST /subscriptions/details	The individualApn parameter was added and the response example was updated.
DCP release 19A3, document revision M		
Section 3.2 on page 4	Versioning	An example for V2 header was added. It enables the new version of secondary key query operation.
Section 3.3 on page 4	Encapsulation	Information was added about response and request content types for separate versions.
Section 4 on page 6	Endpoints and Operations	Information was added that a new version is available for the following endpoint: <ul style="list-style-type: none"> GET /subscriptions/details/{key}/{value}
Section 4.3.2 on page 16	GET /subscriptions/details/{key}/{value}	Information was added that a new version is available for this endpoint and a new response example was added for V2.
DCP release 19A2, document revision L		
The whole document	Subscription Management API	The POST /actions/assignImei endpoint was removed from the API. Use the POST /subscriptions/{id}:assignImei endpoint instead, see Section 4.3.8 on page 24.



3 Interface Overview

This section provides a brief overview of the Subscription Management API.

3.1 Using the API

Each API operation is available at its own endpoint that must be inserted to the end of the following URL, where the `<baseURI>` parameter is provided by the operator:

```
<baseURI>/dcpapi/rest/subscriptionManagement/v1
```

Operations are started by submitting a request on the related endpoint. See Section 4 on page 6 for a detailed description of operations, or refer to the following URL for the complete online documentation:

```
<baseURI>/dcpapi/rest/subscriptionManagement/v1/ =>  
docs/index.html
```

3.2 Versioning

The versioning of the API is based on media type, to ensure backwards compatibility during API changes. The version must be specified in the **Accept** header in each request. Example headers for API versions are the following:

— Accept: application/vnd.dcp-v1+json

— Accept: application/vnd.dcp-v2+json

3.3 Encapsulation

The API accepts content types `application/json` and `application/vnd.dcp-v1+json` for body parameters in V1 operations. (No current V2 operation includes a body parameter in the API.)

Response content type is `application/vnd.dcp-v1+json` for V1 operations and `application/vnd.dcp-v2+json` for V2 operations.

See Table 2 for the endpoints available in the API.

3.4 Security

This section describes the security measures in the API. Security measures include authentication, authorization and encryption.



3.4.1 Authentication

All API endpoints require authentication, which is performed by passing a token in the **X-Access-Token** header in each request.

The token can be acquired using the `/login` or `/login-form` endpoints. The login response contains the token expiration time in milliseconds. For prolonged interaction with the API, a new token can be reissued before the expiration of the current one by using the `/token-reissue` endpoint.

HTTP status 401 is returned in case of a missing, expired or invalid token.

See Table 2 for the endpoints available in the API and Table 20 for the list of HTTP status codes.

3.4.2 Authorization

The API uses the tokens described in Section 3.4.1 on page 4 to decide whether the user is authorized to use a particular endpoint.

3.4.3 Encryption

All API requests are encrypted using the HTTPS protocol to grant secure data transfer.



4

Endpoints and Operations

This section provides details on the available operations of the API.

Each of these operations use one of the following HTTP request methods combined with an appropriate endpoint:

- **POST:** These requests are used to submit data through the API.
- **GET:** These requests are used to query data through the API.
- **DELETE:** These requests are used to delete data through the API.

Table 2 shows the API endpoints together with the description of the operation they provide and the request type belonging to each endpoint. Click an endpoint to jump to the related section.

Table 2 API Endpoints

Endpoint	Provided Operation
Login	
POST /login	Signs in and authenticates the user to the API.
POST /login-form	Signs in and authenticates the user to the API.
POST /token-reissue	Requests the reissue of the access token.
SubscriptionOrder	
GET /subscriptions/order/metadata	Returns the selected values required for SIM Order parameters.
POST /subscriptions/order	Creates a SIM Order.
GET /subscriptions/stockOrders/metadata	Returns the selected values required for SIM Stock Order parameters.
POST /subscriptions/stockOrders	Creates a SIM Stock Order.
GET /subscriptions/pendingOrders	Finds pending orders.
POST /subscriptions/orders/{orderId}:approve	Approves a pending order.
DELETE /subscriptions/orders/{orderId}	Rejects a pending order.
SubscriptionInventory	
POST /subscriptions/details	Returns the details of subscriptions based on a primary key, which can be IMSI, ICC, or MSISDN.
GET /subscriptions/details/{key}/{value}	Returns the details of subscriptions based on a secondary key, which can be the company ID or the customer label. Two different versions are available for this endpoint.
GET /subscriptions/{id}/history	Queries the history of a subscription.
GET /subscriptions/requests/{requestId}	Returns detailed information on a single subscription request. The string {requestId} must be replaced with an actual subscription request ID in the endpoint URL.
GET /subscriptions/requests/find	Returns subscription request IDs based on search criteria.



Endpoint	Provided Operation
GET /subscriptions/returnedMsisdns	Queries MSISDNs which are returned from the quarantine period.
POST /subscriptions/{imsi}:reassignMsisdn	Reassigns the MSISDN to a new subscription which is identified by IMSI.
POST /subscriptions/{id}:assignImei	Updates the assigned IMEI of a subscription.
POST /subscriptions/{subscription_id}:suspendSubscription	Suspends a subscription and removes its MSISDN.
POST /subscriptions/{subscription_id}:changeMsisdn	Changes the MSISDN of a terminated subscription.

The sections below give comprehensive information on the operations provided by each API endpoint. For each operation, all input parameters are listed and described in table format if applicable. A generic response model is also provided.

4.1 Login

4.1.1 POST /login

The operation provided by the POST /login endpoint signs in and authenticates the user to the API. See Table 3 for the required parameters.

The POST /login endpoint provides the same functionality as the POST /login-form endpoint, but with a different parameter. See Section 4.1.2 on page 7 for details on the POST /login-form endpoint.

Table 3 POST /login Request Parameters

Parameter Name	Description	Parameter Type	Data Type
body	<p>The user credentials, that is, the ID and the password belonging to the user.</p> <p>The first parameter is called email, but it means user ID. Therefore, you must enter your user ID, together with the associated password to log in to the API. It is possible to have an email address as a user ID.</p>	body	Credentials { email (string), password (string) }

The following response model presents the variables in the response:

```
{
  message: (string),
  expirationTime (integer),
  userId: (string),
  token: (string)
}
```

Example 1 POST /login Response Model



4.1.2 POST /login-form

The operation provided by the POST /login-form endpoint signs in and authenticates the user to the API. See Table 4 for the required parameters.

The POST /login-form endpoint provides the same functionality as the POST /login endpoint, but with different, easier-to-use parameters. See Section 4.1.1 on page 7 for details on the POST /login endpoint.

Table 4 POST /login-form Request Parameters

Parameter Name	Description	Parameter Type	Data Type
username	The ID belonging to the user. The parameter is called username, but it means user ID. It is possible to have an email address as a user ID.	formData	string
password	The password belonging to the user.	formData	string

The following response model presents the variables in the response:

```
{
  message: (string),
  expirationTime (integer),
  userId: (string),
  token: (string)
}
```

Example 2 POST /login-form Response Model

4.1.3 POST /token-reissue

The operation provided by the POST /token-reissue endpoint requests the reissue of the access token. No parameters are needed.

The following response model presents the variables in the response:

```
{
  message: (string),
  expirationTime (integer),
  userId: (string),
  token: (string)
}
```

Example 3 POST /token-reissue Response Model



4.2 SubscriptionOrder

4.2.1 GET /subscriptions/order/metadata

The operation provided by the GET /subscriptions/order/metadata endpoint requests the information required to populate the POST /subscriptions/order endpoint request parameters. See Table 5 for the request parameters.

Note: If requesting all data at once returns an error, try using separate requests by ignoring specific sets of data.

Table 5 GET /subscriptions/order/metadata Request Parameters

Parameter Name	Description	Parameter Type	Data Type
companyId	Mandatory. The ID of the company.	query	string
ignoreServiceContracts	Optional. If set to true, service contracts will not be returned.	query	string
ignoreDeliveryAddresses	Optional. If set to true, delivery addresses will not be returned.	query	string
ignoreSimSpecifications	Optional. If set to true, SIM specifications will not be returned.	query	string
ignoreMsisdnPools	Optional. If set to true, MSISDN pools for SIM specifications will not be returned.	query	string

The following response model presents the variables in the response:



```
{
  simSpecifications: [
    {
      * simSpecificationId: (string),
      minimumOrderVolume: (integer),
      orderIncrement: (integer),
      lowOrderThreshold: (integer),
      earliestDeliveryDate: (string),
      description: (string),
      operatorId: (string),
      numberSchemeId: (string),
      numberSchemeName: (string),
      networkResourceGroupId (integer),
      networkResourceGroupName (string),
      msisdnsPools: [
        {
          msisdnsPoolId: (string),
          * shortName: (string),
          description: (string)
        }
      ]
    }
  ],
  serviceContracts: [
    {
      serviceContractId: (string),
      * name: (string),
      description: (string),
      subscriptionPackages: [
        {
          * name: (string),
          description: (string),
          status: (string),
          networkResourceGroupId (integer)
        }
      ]
    }
  ],
  deliveryAddresses: [
    {
      * deliveryAddress: (string),
      phone: (string)
    }
  ]
}
```

Example 4 GET /subscriptions/order/metadata Response Model

Note: The following elements are marked by an asterisk in the example to indicate that their values must be used to populate the order data in the POST /subscriptions/order endpoint, see Section 4.2.2 on page 10:

- simSpecifications / simSpecificationId
- simSpecifications / msisdnsPools / shortName
- serviceContracts / name
- serviceContracts / subscriptionPackages / name
- deliveryAddresses / deliveryAddress

4.2.2 POST /subscriptions/order

The operation provided by the POST /subscriptions/order endpoint creates an order and forwards the generated order file through FTP. Order data fields have to be populated with correct values from the GET



/subscriptions/order/metadata response for the same enterprise. See Table 6 for the request parameters.

Note: Only standard SIM orders are supported and the subscription initial state must be active. The networkResourceGroupId of the SIM specification and the networkResourceGroupId of the subscription package must match.

Table 6 POST /subscriptions/order Request Parameters

Parameter Name	Description	Parameter Type	Data Type
body	The parameters needed for the SIM order.	body	See Example 5 for the request model.

```
{
  quantity: (integer),
  preferredDeliveryDate: (string),
  companyId: (string),
  deliveryAddress: (string),
  packageMarkings: (string),
  simSpecification: (string),
  contactName: (string),
  contactEmail: (string),
  contactPhone: (string),
  initialState: (string) = [Active, Deactivated],
  subscriptionPackage: (string),
  serviceContract: (string),
  poolShortName: (string)
}
```

Example 5 POST /subscriptions/order Request Model

The following elements must fulfill certain criteria:

- **preferredDeliveryDate:** Date must be formatted according to ISO8601, yyyy-MM-dd.
- **simSpecification:** Must be equal to one of the simSpecificationId from the GET /subscriptions/stockOrders/metadata response.
- **initialState:** Must be set to active.
- **subscriptionPackage:** Must be equal to one of the name of subscription packages listed for that service contract and the subscription package must not be a draft.
- **serviceContract:** Must be equal to one of the name of service contracts listed in the GET /subscriptions/order/metadata response.
- **poolShortName:** Must be equal to one of shortName for that SIM specification if msisdnsPools are listed for that SIM specification in the GET /subscriptions/order/metadata response.

The following response model presents the variables in the response:



```
{
  jobId: (string),
  resultCode: (string),
  errorCode: (string)
}
```

Example 6 POST /subscriptions/order Response Model

4.2.3 GET /subscriptions/stockOrders/metadata

The operation provided by the GET /subscriptions/stockOrders/metadata endpoint requests the information required to populate the POST /subscriptions/stockOrders endpoint request parameters.

No request parameters are needed.

The following response model presents the variables in the response:

```
{
  simSpecifications: [
    {
      simSpecificationId: (string),
      minimumOrderVolume: (integer),
      orderIncrement: (integer),
      lowOrderThreshold: (integer),
      earliestDeliveryDate: (string),
      description: (string),
      operatorId: (string),
      numberSchemeId: (string),
      numberSchemeName: (string),
      networkResourceGroupId: (integer),
      networkResourceGroupName: (string),
      msisdnsPools: [
        {
          msisdnsPoolId: (string),
          shortName: (string),
          description: (string)
        }
      ]
    }
  ]
}
```

Example 7 GET /subscriptions/stockOrders/metadata Response Model

4.2.4 POST /subscriptions/stockOrders

The operation provided by the POST /subscriptions/stockOrders endpoint creates a stock order and forwards the generated stock order file through FTP. Stock order data fields have to be populated with correct values from the GET /subscriptions/stockOrders/metadata response for the same enterprise. See Table 7 for the request parameters.

Table 7 POST /subscriptions/stockOrders Request Parameters

Parameter Name	Description	Parameter Type	Data type
body	The parameters needed for the SIM stock order.	body	See Example 8 for the request model.



```
{
  quantity: (integer),
  preferredDeliveryDate: (string),
  companyId: (string),
  deliveryAddress: (string),
  packageMarkings: (string),
  simSpecification: (string),
  contactName: (string),
  contactEmail: (string),
  contactPhone: (string)
}
```

Example 8 POST /subscriptions/stockOrders Request Model

The following elements must fulfill certain criteria:

- **preferredDeliveryDate:** Date must be formatted according to ISO8601, yyyy-MM-dd.
- **simSpecification:** Must be equal to one of the simSpecificationId from the GET /subscriptions/stockOrders/metadata response.

The following response model presents the variables in the response:

```
{
  jobId: (string),
  resultCode: (string),
  errorCode: (string)
}
```

Example 9 POST /subscriptions/stockOrders Response Model

4.2.5 GET /subscriptions/pendingOrders

The operation provided by the GET /subscriptions/pendingOrders endpoint finds orders pending for approval.

No request parameters are needed.

The following response model presents the variables in the response:

```
{
  jobsPendingApproval: [
    {
      jobId: (string),
      orderedBy: (string),
      companyId: (string),
      createdAt: (integer),
      updatedAt: (integer),
      orders: [
        {
          quantity: (integer),
          serviceContractId: (string),
          subscriptionPackageId: (string),
          simSpecificationId: (string),
          orderType: (string),
          companyId: (string)
        }
      ]
    }
  ]
}
```

Example 10 GET /subscriptions/pendingOrders Response Model



4.2.6 POST /subscriptions/orders/{orderId}:approve

The operation provided by the POST /subscriptions/orders/{orderId}:approve endpoint approves a pending order. The string {orderId} must be replaced with an actual order ID in the endpoint URL. See Table 8 for the request parameters.

Table 8 POST /subscriptions/orders/{orderId}:approve Request Parameters

Parameter Name	Description	Parameter Type	Data type
orderId	Mandatory. The ID of the order that is waiting for approval or rejection.	path	string
additionalInfo	Optional. Additional information for approval.	query	string

The following response model presents the variables in the response:

```
{
  request: {
    userToken: (string),
    uuid: (string)
  },
  serviceRequestId: (string),
  success: (boolean),
  errorCode: (integer)
}
```

Example 11 POST /subscriptions/orders/{orderId}:approve Response Model

4.2.7 DELETE /subscriptions/orders/{orderId}

The operation provided by the DELETE /subscriptions/orders/{orderId} endpoint rejects a pending order. The string {orderId} must be replaced with an actual order ID in the endpoint URL. See Table 9 for the request parameters.

Table 9 DELETE /subscriptions/orders/{orderId} Request Parameters

Parameter Name	Description	Parameter Type	Data type
orderId	Mandatory. The ID of the order that is waiting for approval or rejection.	path	string
additionalInfo	Optional. Additional information for rejection.	query	string

The following response model presents the variables in the response:

```
{
  request: {
    userToken: (string),
    uuid: (string)
  },
  serviceRequestId: (string),
  success: (boolean),
  errorCode: (integer)
}
```

Example 12 DELETE /subscriptions/orders/{orderId} Response Model



4.3 SubscriptionInventory

4.3.1 POST /subscriptions/details

The operation provided by the POST /subscriptions/details endpoint returns the details of subscriptions based on a primary key, which can be IMSI, ICC, or MSISDN. The amount of primary keys must be between 1 and 10000. See Table 10 for the parameters.

Table 10 POST /subscriptions/details Request Parameters

Parameter Name	Description	Parameter Type	Data Type
body	<p>The following parameters are included:</p> <ul style="list-style-type: none"> • type • values • groups 	body	<pre>GetSubscriptionDetailsByPrimarykey { type (string) = [imsi, icc, msisdn] values groups (string) = [enterpriseInfo, packageInfo, dates, label, simSpecification, lockStateInfo, individualApn] }</pre>

Groups in request body return the following parameters in the response:

- **enterpriseInfo:** Returns the companyName in the response.
- **packageInfo:** Returns the subscriptionPackageName and subscriptionPackageDescription in the response.
- **dates:** Returns the firstActivationDate and pbrExitDate in the response.
- **label:** Returns the customerLabel in the response.
- **simSpecification:** Returns the simSpecificationId in the response.
- **lockStateInfo:** Returns the lockState and lockingReason in the response.
- **individualApn:** Returns the individualApns in the response. If the list is empty, APNs are used from package.

The following response model presents the variables in the response:



```

{
  subscriptionDetails: [
    {
      imsi: (string),
      msisdn: (string),
      subscriptionState: (string) = [ACTIVE, DEACTIVATED, PAUSE, ⇒
        TERMINATED, TERMINATED_PENDING, OPERATOR_BLOCKED, ⇒
        DEACTIVATED_NO_BILLING, ACTIVE_NO_BILLING],
      icc: (string),
      imei: (string),
      companyId: (string),
      companyName: (string),
      subscriptionPackageName: (string),
      subscriptionPackageDescription: (string),
      firstActivationDate: (string),
      lockState: (string) = [UNLOCKED, LOCKED, UNKNOWN],
      lockingReason: (string),
      customerLabel: (string),
      simSpecificationId: (string),
      pbrExitDate: (string),
      individualApns: [
        {
          ordinal: (integer),
          apn: (string),
          type: (string)
        }
      ]
    }
  ],
  size: (integer)
}

```

Example 13 POST /subscriptions/details Response Model

4.3.2 GET /subscriptions/details/{key}/{value}

The operation provided by the GET /subscriptions/details/{key}/{value} endpoint returns the details of subscriptions based on a secondary key, which can be the company ID or the customer label. Two different versions are available for this endpoint, as follows:

- V1, visible on the Swagger page or triggered by sending the request header Accept: application/vnd.dcp-v1+json.
- V2, not visible on the Swagger page. It must be triggered by sending the request header Accept: application/vnd.dcp-v2+json.

The limit for retrieving subscriptions for V1 (using the start parameter) is 200000. The start parameter is not available for V2, because it does not have a limit. Instead the response contains a link to the next page of the result set. To iterate through the pages of the result set, follow the link to the next page at JSON path `_links.next.href` of a response. For example using the jq `'._links.next.href'` command. See Table 11 for the parameters.

Note: V1 is going to be removed six months after 19A3.



Table 11 GET /subscriptions/details/{key}/{value} Request Parameters

Parameter Name	Description	Parameter Type	Data Type
key	Mandatory. The secondary key used for the query. It can be company for the company ID or label for the customer label. Use only one secondary key per query.	path	string
value	Mandatory. The value of the secondary key used for the query.	path	string
order	Optional. The results are ordered based on the subscription field specified in this element (default IMSI).	query	string
group	Optional. Additional subscription field groups to be included in the response.	query	array [string]
start ⁽¹⁾	Optional. Return requests starting from index, the default value is 0 and the maximum allowed value is 200000.	query	integer

(1) This parameter is only available for V1.

Groups in the request return the following parameters in the response:

- **enterpriseInfo:** Returns the companyName in the response.
- **packageInfo:** Returns the subscriptionPackageName and subscriptionPackageDescription in the response.
- **dates:** Returns the firstActivationDate and pbrExitDate in the response.
- **label:** Returns the customerLabel in the response.
- **simSpecification:** Returns the simSpecificationId in the response.
- **lockStateInfo:** Returns the lockState and lockingReason in the response.

The following response model presents the variables in the V1 response:



```

{
  subscriptionDetails: [
    {
      imsi: (string),
      msisdn: (string),
      subscriptionState: (string) = [ACTIVE, DEACTIVATED, PAUSE, ⇒
        TERMINATED, OPERATOR_BLOCKED, ⇒
        DEACTIVATED_NO_BILLING, ACTIVE_NO_BILLING],
      icc: (string),
      imei: (string),
      companyId: (string),
      companyName: (string),
      subscriptionPackageName: (string),
      subscriptionPackageDescription: (string),
      firstActivationDate: (string),
      lockState: (string) = [UNLOCKED, LOCKED, UNKNOWN],
      lockingReason: (string),
      customerLabel: (string),
      simSpecificationId: (string),
      pbrExitDate: (string)
    }
  ],
  moreResultsAvailable: (boolean),
  start: (integer),
  size: (integer)
}

```

Example 14 GET /subscriptions/details/{key}/{value}V1 Response Model

The following response model presents the variables in the V2 response:

```

{
  subscriptionDetails: [
    {
      imsi: (string),
      msisdn: (string),
      subscriptionState: (string) = [ACTIVE, DEACTIVATED, PAUSE, ⇒
        TERMINATED, TERMINATED_PENDING, OPERATOR_BLOCKED, ⇒
        DEACTIVATED_NO_BILLING, ACTIVE_NO_BILLING],
      icc: (string),
      imei: (string),
      companyId: (string),
      companyName: (string),
      subscriptionPackageName: (string),
      subscriptionPackageDescription: (string),
      firstActivationDate: (string),
      lockState: (string) = [UNLOCKED, LOCKED, UNKNOWN],
      lockingReason: (string),
      customerLabel: (string),
      simSpecificationId: (string),
      pbrExitDate: (string)
    }
  ],
  _links: {
    self: {
      href: (string) = [<baseURI>/dcpapi/rest/subscriptionManagement/⇒
        <version>/subscriptions/details/<companyName>/<companyId>?group=label&group=dates]
    }
    next: {
      href: (string) = [<baseURI>/dcpapi/rest/subscriptionManagement/⇒
        <version>/subscriptions/details/<companyName>/<companyId>?⇒
        group=label&group=dates&begin=imsi]
    }
  }
}

```

Example 15 GET /subscriptions/details/{key}/{value} V2 Response Model



4.3.3 GET /subscriptions/{id}/history

The operation provided by the GET /subscriptions/{id}/history endpoint queries the history of a subscription. The string {id} must be replaced with an actual subscription ID in the endpoint URL. See Table 12 for the parameters.

Table 12 GET /subscriptions/{id}/history Request Parameters

Parameter Name	Description	Parameter Type	Data Type
idType	Optional. The type of the subscription identifier can be the following: <ul style="list-style-type: none"> • IMSI • ICCID • MSISDN The default identifier is IMSI.	query	string
id	The subscription identifier. IMSI if the idType is not specified. If the idType is specified the corresponding subscription identifier type must be used.	path	string

The following response model presents the variables in the response:

```
[
  {
    timestamp: (integer),
    subscriptionHistoryType: (string) = [CHANGE_EXISTING_SUBSCRIPTIONSTATE, ⇒
    CHANGE_SUBSCRIPTIONOWNER, CHANGE_SUBSCRIPTIONOWNER_EUICC, ⇒
    CHANGE_SUBSCRIPTIONREGION, CHANGE_SUBSCRIPTIONSTATE, ⇒
    CHANGE_SUBSCRIPTIONSTATE_EUICC, CHANGE_SUBSCRIPTIONPACKAGE, ⇒
    CHANGE_SUBSCRIPTIONPACKAGE_EUICC, CHANGE_SUBSCRIPTIONLABEL, ⇒
    CHANGE_SUBSCRIPTIONNOTE, CHANGE_ALTROAMINGPROVIDER, CREATE_SUBSCRIPTION, ⇒
    CREATE_SUBSCRIPTION_EUICC, THROTTLE_IMSI, POLICYBASEDROUTING_EXIT, ⇒
    TRIGGER_ACTION, TIME_BASED_CHANGE_SUBSCRIPTIONSTATE, ⇒
    TIME_BASED_CHANGE_SUBSCRIPTIONPACKAGE, UPDATE_IMEI, REASSIGN_MSISDN, ⇒
    UPDATE_PROVISIONED_IMEI, PCL_LIMIT_EXCEEDED, ⇒
    CHANGE_SUBSCRIPTIONSTATE_AND_UNLINK_SUBSCRIPTION, ⇒
    LOCK_SUBSCRIPTIONSTATE, UNLOCK_SUBSCRIPTIONSTATE, UNKNOWN],
    serviceRequestId: (string),
    initiatorName: (string),
    description: (string),
    subscriptionState: (string) = [NEW, ACTIVE, DEACTIVATED, PAUSE, ⇒
    TERMINATED, PENDING_TERMINATION, REMOVED, OPERATOR_BLOCKED, ⇒
    ACTIVE_NO_BILLING, DEACTIVATED_NO_BILLING, UNKNOWN],
    subscriptionPackage: (string),
    serviceConfiguration: (string),
    msisdn: (string),
    imei: (string),
    provisionedImei: (string)
  }
]
```

Example 16 GET /subscriptions/{id}/history Response Model



4.3.4 GET /subscriptions/requests/{requestId}

The operation provided by the GET /subscriptions/requests/{requestId} endpoint returns the details of a single subscription request. The string {requestId} must be replaced with an actual subscription request ID in the endpoint URL. See Table 13 for the parameters.

Table 13 GET /subscriptions/requests/{requestId} Request Parameters

Parameter Name	Description	Parameter Type	Data Type
requestId	The ID of the subscription request.	path	string

The following response model presents the variables in the response:



```

{
  subscriptionRequestId: (string),
  subscriptionRequestType: (string) = [CESS, CHANGE_SUBSCRIPTIONOWNER, ⇒
    CHANGE_SUBSCRIPTIONREGION, CHANGE_SUBSCRIPTIONSTATE, ⇒
    CHANGE_SUBSCRIPTIONPACKAGE, CHANGE_SUBSCRIPTIONLABEL, ⇒
    CHANGE_SUBSCRIPTION_NOTE, CHANGE_ALTROAMINGPROVIDER, EUICC_ORDER, ⇒
    EUICC_PROFILE_ORDER, EUICC_STOCK_ORDER, EUICC_UPDATE_LABEL, ⇒
    EUICC_LOCALIZATION, EUICC_DELOCALIZATION, EUICC_LOCALIZATION_BATCH, ⇒
    EUICC_DELOCALIZATION_BATCH, EUICC_ADD_LOCALE, EUICC_ADD_LOCALE_BATCH, ⇒
    EUICC_REMOVE_LOCALE, EUICC_DOWNLOAD_PROFILE, EUICC_PREPARE_SMSR_CHANGE, ⇒
    EUICC_SMSR_CHANGE, EUICC_DELETE_PROFILE, EUICC_ALLOCATE_PROFILE, ⇒
    EUICC_ENABLE, EUICC_RESET_COUNTERS, SIM_ORDER, STOCK_ORDER, ⇒
    TRIGGERPACKAGE_ASSIGN, TRIGGERPACKAGE_REMOVE, LOGIN, LOGOUT, ⇒
    REGISTER_CALLBACK, EXECUTE_USE_CASE, UNKNOWN],
  subscriptionRequestState: (string) = [IN_PROGRESS, PENDING, CANCELED, ⇒
    REJECTED, COMPLETED, COMPLETED_WITH_FAILURES, UNKNOWN],
  createdBy: (string),
  companyId: (string),
  companyName: (string),
  timeCreated: {
    epochSecond: (integer),
    nano: (integer)
  },
  lastUpdated: {
    epochSecond: (integer),
    nano: (integer)
  },
  size: (integer),
  subscriptionRequestResourceCounters: {
    inProgress: (integer),
    pending: (integer),
    canceled: (integer),
    rejected: (integer),
    completed: (integer),
    failed: (integer)
  },
  subscriptionRequestFailureInfos: [
    {
      errorObject: {},
      getErrorType: (string) = [ES_ERROR, DCP_ERROR]
    }
  ],
  subscriptionRequestResources: [
    {
      resourceProperties: [
        {
          type: (string) = [ICCID, IMSI, MSISDN, EUICCID, INTERNAL_ID, ⇒
            TARGET_SUBSCRIPTION_STATE, TARGET_SUBSCRIPTION_PACKAGE],
          value: (string),
          operation: (string) = [SUBSCRIPTION_STATE_CHANGE, ⇒
            SUBSCRIPTION_PACKAGE_CHANGE, DOWNLOAD, DOWNLOAD_AND_ENABLE, ⇒
            ENABLE, DELETE, LOCALIZATION, ADD_LOCALE, ⇒
            DELOCALIZATION, REMOVE_LOCALE, UNKNOWN]
        }
      ],
      resourceState: (string) = [IN_PROGRESS, PENDING, CANCELED, REJECTED, ⇒
        COMPLETED, FAILED, UNKNOWN],
      resourceFailureInfos: [
        {
          errorObject: {},
          getErrorType: (string) = [ES_ERROR, DCP_ERROR]
        }
      ],
      resourceId: (string)
    }
  ]
}

```

Example 17 GET /subscriptions/requests/{requestId} Response Model



4.3.5 POST /subscriptions/requests/find

The operation provided by the POST /subscriptions/requests/find endpoint returns subscription request IDs based on search criteria. See Table 14 for the parameters.

Table 14 POST /subscriptions/requests/find Request Parameters

Parameter Name	Description	Parameter Type	Data Type
body	The search criteria for the subscription request.	body	See Example 18 for the model of the search criteria.

```
{
  subscriptionRequestTypes: [
    (string) = [CESS, CHANGE_SUBSCRIPTIONOWNER, ⇒
      CHANGE_SUBSCRIPTIONREGION, CHANGE_SUBSCRIPTIONSTATE, ⇒
      CHANGE_SUBSCRIPTIONPACKAGE, CHANGE_SUBSCRIPTIONLABEL, ⇒
      CHANGE_SUBSCRIPTION_NOTE, CHANGE_ALTROAMINGPROVIDER, ⇒
      EUICC_ORDER, EUICC_PROFILE_ORDER, EUICC_STOCK_ORDER, ⇒
      EUICC_UPDATE_LABEL, EUICC_LOCALIZATION, EUICC_DELOCALIZATION, ⇒
      EUICC_LOCALIZATION_BATCH, EUICC_DELOCALIZATION_BATCH, ⇒
      EUICC_ADD_LOCALE, EUICC_ADD_LOCALE_BATCH, EUICC_REMOVE_LOCALE, ⇒
      EUICC_DOWNLOAD_PROFILE, EUICC_DELETE_PROFILE, EUICC_RESET_COUNTERS, ⇒
      SIM_ORDER, STOCK_ORDER, TRIGGERPACKAGE_ASSIGN, ⇒
      TRIGGERPACKAGE_REMOVE, UNKNOWN]
  ],
  subscriptionRequestStates: [
    (string) = [IN_PROGRESS, PENDING, CANCELED, ⇒
      REJECTED, COMPLETED, COMPLETED_WITH_FAILURES, UNKNOWN]
  ],
  createdBy: (string),
  limit: (integer),
  companyIds: [
    (string)
  ],
  timeCreatedBefore: (string),
  timeCreatedAfter: (string),
  lastUpdatedBefore: (string),
  lastUpdatedAfter: (string),
  sortOrder: {
    sortField: (string) = [SUBSCRIPTION_REQUEST_TYPE, ⇒
      SUBSCRIPTION_REQUEST_STATE, TIME_CREATED_BEFORE, ⇒
      TIME_CREATED_AFTER, LAST_UPDATED_BEFORE, ⇒
      LAST_UPDATED_AFTER, CREATED_BY],
    descending: (boolean)
  }
}
```

Example 18 Search Criteria Model

The response returns the subscription request IDs based on the search criteria.

4.3.6 GET /subscriptions/returnedMsisdns

The operation provided by the GET /subscriptions/returnedMsisdns endpoint queries MSISDNs which are returned from the quarantine period. This is done based on the operator ID and the specified time period. See Table 15 for the request parameters.

Note: Internal number management (that is, number management by DCP) is the requirement for this operation to work. In case of external number management, the endpoint returns an empty list.



Table 15 GET /subscriptions/returnedMsisdns Request Parameters

Parameter Name	Description	Parameter Type	Data Type
operatorId	The ID of the operator to which the MSISDNs belong.	query	string
startDate	The starting date of the specified time period. The format is YYYY-MM-DD.	query	string
endDate	The ending date of the specified time period. The format is YYYY-MM-DD.	query	string
simSpecificationId	Optional. The SIM specification ID.	query	string

The following response model presents the variables in the response:

```
{
  items: [
    {
      startNumber: (string),
      endNumber: (string),
      dateTime: (string),
      poolId: (string),
      reservedFor: (string),
      available: (string),
      ordered: (string)
    }
  ]
}
```

Example 19 GET /subscriptions/returnedMsisdns Response Model

4.3.7

POST /subscriptions/{imsi}:reassignMsisdn

The operation provided by the POST /subscriptions/{imsi}:reassignMsisdn endpoint reassigns the MSISDN (of a subscription) to a new subscription which is identified by IMSI.

For the operation to work, the target subscription must be in paused state, and both of the original and target subscriptions must be under the same enterprise. After the operation, MSISDN (provided in the request body) of the original subscription and its label are transferred to the target subscription (identified by IMSI in the request).

If both the original and the target subscription have Consumer Connectivity V2 enabled in the corresponding subscription packages, then the data buckets, purchases and recurrences are transferred from the original subscription to the target subscription. Other properties of the new subscription come from the corresponding subscription package.

Note: The original subscription which is associated with the specified MSISDN is removed from the system permanently. An MSISDN which is not used by an existing subscription cannot be specified. The operation cannot be undone.



See Table 16 for the required parameters.

Table 16 POST /subscriptions/{imsi}:reassignMsisdn Request Parameters

Parameter Name	Description	Parameter Type	Data Type
imsi	Identifies the target subscription to which the original MSISDN is to be reassigned.	path	string
body	The organization ID and the original MSISDN. Optionally, additional information for the request.	body	Assign IMEI request { companyId (string) additionalInfo (string) msisdn (string) }

The response returns the service request ID and a message that the MSISDN is reassigned without errors.

4.3.8

POST /subscriptions/{id}:assignImei

The operation provided by the POST /subscriptions/{id}:assignImei endpoint updates the assigned IMEI of a subscription. See Table 17 for the parameters.

Table 17 POST /subscriptions/{id}:assignImei Request Parameters

Parameter Name	Description	Parameter Type	Data Type
idType	Optional. The type of the subscription identifier can be the following: <ul style="list-style-type: none">• IMSI• ICCID• MSISDN The default identifier is IMSI.	query	string
id	The subscription identifier. IMSI if the idType is not specified. If the idType is specified the corresponding subscription identifier type must be used.	path	string
body	The organization ID and the assigned IMEI. Optionally, additional information for the request. To remove the assigned IMEI of the subscription, set the assigned IMEI field value to an empty string as null value is not supported.	body	Assign IMEI request { companyId (string), additionalInfo (string), assignedImei (string) }

The response returns the service request ID and a message that the IMEI is assigned and updated successfully.



4.3.9

POST /subscriptions/{subscription_id}:suspendSubscription

The operation provided by the POST /subscriptions/{subscription_id}:suspendSubscription removes the MSISDN of a single subscription and suspends it. See Table 18 for the parameters.

Table 18 POST /subscriptions/{subscription_id}:suspendSubscription Request Parameters

Parameter Name	Description	Parameter Type	Data Type
subscription_id	Mandatory. The ID of the subscription to be suspended.	query	string
body	Mandatory. The type of the subscription identifier: <ul style="list-style-type: none"> IMSI ICC 	body	string

The following response model presents the variables in the response:

Request received successfully.

Response code 200:

```
{
  "serviceRequestId": "string"
}
```

Invalid arguments for the operation

Response code 400:

```
{
  "code": 2040,
  "httpStatus": 401,
  "message": "Request validation failed",
  "detailedMessage": "Could not parse JSON body"
}
```

Operation failed for internal reasons.

Response code 500:

```
{
  "code": 2040,
  "httpStatus": 401,
  "message": "Request validation failed",
  "detailedMessage": "Could not parse JSON body"
}
```

Example 20 POST /subscriptions/{subscription_id}:suspendSubscription Response Model

4.3.10

POST /subscriptions/{subscription_id}:changeMsisdn

The operation provided by the POST /subscriptions/{subscription_id}:changeMsisdn reactivates a subscription and assigns a new MSISDN. See Table 19 for the parameters.



Table 19 POST /subscriptions/{subscription_id}:changeMsisdn Request Parameters

Parameter Name	Description	Parameter Type	Data Type
subscription_id	Mandatory. The ID of the subscription whose MSISDN is to be changed.	query	string
body	<p>The change information for the subscription identified by IMSI, or ICC.</p> <ul style="list-style-type: none">• Mandatory. Subscription ID type• Optional. New MSISDN• Optional. Company ID• Optional. Additional information	body	<pre>{ "subscription_id_type": "string", "msisdn": "string", "companyId": "string", "additional_info": "string" }</pre>

The following response model presents the variables in the response:

Request received successfully.

Response code 200:

```
{
  "serviceRequestId": "string"
  "msisdn": "string"
}
```

Invalid arguments for the operation

Response code 400:

```
{
  "code": 2040,
  "httpStatus": 401,
  "message": "Request validation failed",
  "detailedMessage": "Could not parse JSON body"
}
```

Operation failed for internal reasons.

Response code 500:

```
{
  "code": 2040,
  "httpStatus": 401,
  "message": "Request validation failed",
  "detailedMessage": "Could not parse JSON body"
}
```

Example 21 POST /subscriptions/{subscription_id}:changeMsisdn Response Model



5 HTTP Response Codes

Table 20 contains the HTTP response codes and a description applicable to the API.

Table 20 HTTP Response Codes

Response Code	Response Text ⁽¹⁾	Description
200	OK	The request was processed successfully.
201	Created	The request has been fulfilled, resulting in the creation of a new resource.
400	Bad Request	Error due to incorrect syntax. Check the error message and correct the request format.
401	Unauthorized	The authentication has failed due to a missing, expired or invalid token. Check the authentication requirements of the API.
404	Not Found	The specified resource is not found. Check if you specified the resource correctly.
500	Internal Server Error	An unexpected error prevented the server from fulfilling the request. Try again later.
503	Service Unavailable	The server is currently unavailable because it is overloaded or down for maintenance. Try again later.

(1) This document lists the generic HTTP response texts. The actual response texts returned by the API can be different.



Reference List

Ericsson Documents

- [1] DCP Glossary of Terms and Acronyms
GLOSSARY, 1/0033-HRC 105 050/1
- [2] Typographic Conventions
DESCRIPTION, 1/1551-FCK 101 05
- [3] DCP Service Portal Operator Guide
USER GUIDE, 33/1553-HRC 105 050/1