Telefónica Tech

IoT

KITE PLATFORM

# API for customers

Version 11.1.0

Kite Platform team

15/07/2025

Public use

**CONFIDENTIAL**

# 1 Table of Contents

# 2 Document history

| Version | State | Date | Changes |
|---|---|---|---|
| 3.0.0 | Internal review | 06/09/2016 | The Glossary, which was placed in the Introduction, becomes an independent section. Style and format corrections. Pasting, from the Spanish version of this document, the link about how to access to the prepaid vouchers management, and the section about the API to manage prepaid vouchers. Eliminating of the Reports API synchronous interface. |
| 3.0.0 | Reviewed | 06/09/2016 | No changes |
| 3.0.0 | Internal review | 21/10/2016 | FAQs section is added. Changes in the content in How to communicate with Kite Platform API? Changes in the description of API to manage customers' sheet and API to manage billing accounts |
| 3.0.1 | Internal review | 14/11/2016 | No changes. |
| 3.1.0 | New | 12/01/2017 | Adding sections about SIM swap and Echo. Format changes. |
| 3.1.0 | New | 31/01/2017 | Adding new sections in FAQs about consumption, reports, versioning, trusted CAs, flow to sending SMS.<br>Format and style changes. |
| 3.1.0 | Reviewed | 31/01/2017 | No changes |
| 3.1.0 | External review | 01/03/2017 | Adding TPS configuration and reviewing of the Reports Availability section. |
| 3.2.0 | New | 04/05/2017 | Updated the information about Pagination mistakes in FAQs. |
| 3.3.0 | New | 30/10/2017 | Adding an example about the access to the API so the connection can be checked quickly.<br>Example with the command CURL<br>Example through soapUI<br><br>Updating the FAQs with clarifications about the permissions associated with the Customer's certificate, and new examples of requests:<br>distribution of the commercial group<br>adding and deleting of APNs |
| 4.0.0 | Internal review | 15/03/2018 | No changes |
| 4.0.0 | New | 16/04/2018 | Added clarifications about root and intermediate CA certificates.<br>Added maximum TPS value for organization. |
| 4.0.1 | New | 31/05/2018 | Only TLSv 1.2 allowed.<br>New report with the subscription detail billing. |
| 4.0.1 | New | 13/05/2018 | API Transaction report can be requested for the current day. |
| 4.1.0 | New | 03/07/2018 | Added new diagnostic synchronous operation for GPRS and GSM diagnostics.<br>Deleted getSubscriptionDetail from Inventory. You can use the getSubscriptions operation by filtering by subscription.<br>Added new reports:<br>changed imei report<br>manufacturer model |
| 4.2.0 | Internal review | 27/09/2018 | Removed asynchronous GPRS and GSM diagnostic. You can use the synchronous diagnostic. |
| 4.3.0 | Internal review | 27/11/2018 | Added new FAQ for changing the commercial group of the subscription. |
| 4.3.1 | Internal review | 11/01/2019 | No changes |

| | | | |
|---|---|---|---|
| 5.0.0 | New | 01/03/2019 | New user list report |
| 5.1.0 | New | 17/04/2019 | Added support for multiple static apns in Inventory API.<br>You can configure several static apns for the subscription by means of new parameters: staticIpAddress0, staticIpAddress1, … staticIpAddress9<br>StaticIp and staticApnIndex are deprecated.<br>Updated Static APN FAQ |
| 5.2.0 | Internal review | 13/12/2019 | No changes |
| 6.0.0 | Internal review | 20/01/2019 | TPS fixed for diagnostic operations |
| 6.0.0 | Internal review | 16/04/2020 | New alarms monthly report |
| 6.1.0 | Internal review | 25/06/2020 | New direct SIM manufacturer order report. Only for Customers belonging to VIVO Provider. |
| 6.2.0 | Internal review | 14/09/2020 | No changes |
| 6.3.0 | Internal review | 26/10/2020 | No changes |
| 6.4.0 | Internal review | 04/02/2021 | No changes |
| 7.0.0 | Internal review | 20/04/2021 | Preproduction certificate server is signed by DigiCert<br>Updated Kite platform server: kiteplatform-api.telefonica.com |
| 7.1.0 | New | 08/07/2021 | The maximum maxBatchSize allowed for the Inventory API increased to 1000.<br>Added TPS configuration for SIM blocking operation. Only available for Customers belonging to VIVO Provider.<br><br>Added clarifications about certificate generation and WSDL downloading. |
| 7.2.0 | New | 21/09/2021 | Added clarification about unavailability periods due to HLR backup |
| 7.3.0 | Internal review | 03/12/2021 | No changes |
| 7.4.0 | New | 14/02/2022 | Clarification is added about IPs associated with the URL of each environment. |
| 7.5.0 | Internal review | 09/03/2022 | No changes |
| 8.0.0 | Internal review | 17/05/2022 | No changes |
| 8.1.0 | Internal review | 11/07/2022 | No changes |
| 8.2.0 | New | 14/10/2022 | The report of retired daily subscriptions can be requested with start date from 2011-01-01.<br>The creation of the report of monthly expense detail for non monetary users may involve two downloadReportIds if there is a minicycle. |
| 9.0.0 | Internal review | 23/02/2023 | No changes |
| 9.1.0 | Internal review | 07/07/2023 | Added TPS config for the new operations of the Inventory API:<br>• consumptionHistory<br>• gprsHistory |
| 9.2.0 | Internal review | 27/09/2023 | No changes |
| 9.3.0 | Internal review | 16/11/2023 | No changes |
| 10.0.0 | Internal review | 04/03/2024 | No changes |
| 10.1.0 | Internal review | 04/03/2024 | No changes |
| 10.2.0 | Internal review | 26/11/2024 | No changes |
| 11.0.0 | Internal review | 27/03/2025 | Section 4.3.1 is modified to include the extraction of the private key from the certificate using OpenSSL version 3.0.<br><br>Updated availability periods for location reports based on the Customer's location supplementary service level |
| 11.1.0 | Internal review | 08/07/2025 | The report of daily presence can be requested with start date from current day. |

# 3 Glossary

- **Application**: software component developed in order to help in the business of an organization.
- **Billing Cycle**: day of the month in which the billing period of a customer begins. It is fixed for each customer and when the last day of the period ends, the Kite Platform generates a prebill for each billing account of the customer.
- **Life Cycle**: a group of states and transitions that an M2M subscription can have. Up to four types of life cycles are allowed: automatic activation with/without test coupon and manual activation with/without test coupon.
- **Billing account**: all billable events in the Kite Platform are associated with a billing account. These are monthly payments, one-time charges or voice, information and SMS consumption. By default, a platform client always has a billing account and the billable events to the client level (e.g. VPN monthly fee) are always associated with that billing account.
- **Customer**: an organization, which uses connectivity services, organized from the Kite Platform and which a Service Provider provides.
- **Commercial group**: name given in the API (Application Programming Interface) to the subscription's group.
- **Subscriptions Group**: it uses a business plan of basic services, a billing account, defines an expenditure control and, if the business plan allows it, a personalization of white and black lists. All subscriptions, at the moment of belonging to a subscription group, will use their business plan of basic services and will use their billing account for consumption made in pool and will have white and black lists defined in that subscription group. The Customer creates the Subscriptions Group using business plans and billing accounts provided by the Service provider.
- **Business Plan of Basic Services**: it defines a group of available services (voice, data, and SMS) and the way they are charged. Business plan and M2M subscriptions establish an association allowing that these last ones can be used. It is made up of services, operators' restrictions, lifecycle configuration, basic services and life cycle fees, and black and white lists of telephone numbers. A customer can have one or various business plans. The Service Provider creates and distributes them to the Customer.
- **Prebill**: for each expired billing cycle and billing account, it collects an added charge for monthly fees, charges associated with exact events such as service or subscription activation and subscriptions' voice, information and SMS consumption.
- **Service Pack**: given name in the API to refer to a Business Plan of Basic Services.
- **Service Provider**: an organization, which provides the M2M service to Customers.
- **Kite Platform**: organized connectivity product commercial name of Telefónica to the market.
- **Subscription**: a SIM card with Kite Platform service and assigned to a customer.

# 4 Introduction

## 4.1 What is Kite Platform API?

The Kite Platform offers an API that allows customers and suppliers of the service to integrate their systems with the Kite Platform to access its data and capabilities.

By means of this API, the applications using it can perform some of the available tasks via web. The API offers the following services:

- Customer management.
- Inventory management.
- Preinventory management.
- Billing accounts management.
- Commercial groups management.
- Prebill management.
- Basic services commercial plan listing.
- Diagnostics.
- Reports.
- EndCustomer management.
- User management.
- Prepaid vouchers management.
- Sim swap.
- Echo operation for checking platform status.

The operations of each service will be explained below.

## 4.2 How to access Kite Platform API?

The API is available on the Internet, in the address **kiteplatform-api.telefonica.com**, through **port 8010**. Therefore, the services this API exhibits can be used from any system having access to the Internet.

Endpoint:

- Production: kiteplatform-api.telefonica.com

The kiteplatform-api.telefonica.com URL can be resolved with two different IPs: 81.45.14.153 or 81.45.16.202.

The FW of the client's local network must not block these IPs, either by openinng access by IP or URL.

All the requests to the API must use protocol **HTTPS** to guarantee the confidentiality of the exchanged data.

To authenticate and authorize the requests, the consumer of the API has to use a certificate and a private key allowing him/her to carry out the **two-way SSL** negotiation that is required each time the connection to the API is established. Such certificate is clearly related to an organization registered in Kite Platform.

Each time a request is received in the API, the consumer's credentials are checked to verify the licenses to access to the required resources, otherwise returning the resources he/she is licensed to (for example, with the list of subscriptions).

This certificate and the private key are issued on demand of the customer who wants to access the API, and it is important to keep them in a safe place since they represent an access gateway to his/her data in the service. You can limit the API access by setting subnet whitelist.

You can manage your own certificates from the section related to your organization: MY ORGANIZATION -> API E2E applications -> Certificate management for API Kite.

For more information about to generate your own certificates and API access control by subnet whitelist, you can consult the Kite Platform User manual for customers.

All the certificates issued have the same permissions as an ADMIN user in the Portal, so they can execute CRUD operations (create, update and delete).

DigiCert signs the server certificates of Kite Platform in Preproduction and Production. Consequently, in order to be able to use the API Service of Kite Platform, the customer must aggregate the Intermediate CAs certificates to the System Certificates Store.

It is not recommended to directly validate the API Service Certificate of Kite Platform. This certificate is renewed each new year, thus, the consumer of the API could have security problems when changing the server certificate.

In the case of the certificate issued by Kite Platform in Preproduction or Production, the CAs of the route certificate (that should be added to the Certificates Store) are:

- **Root**: DigiCert Global Root CA.
- **Intermediate**: DigiCert SHA2 Secure Server CA

The most appropriate option is to validate only the CAs signers of the certificate and not the server certificate.

Each service in the API has its own URL access:

- Customer management:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Customers/v2/r12
```

- Inventory management:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Inventory/v5/r12
```

- Billing accounts management:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/BillingAccount/v
1/r12
```

- Commercial groups management:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/CommercialGroup/
v2/r12
```

- Prebill management:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Prebill/v1/r12
```

- Basic Services Commercial Plans listing:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/ServicePacks/v2/
r12
```

- Diagnostics:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/DiagnosticInvent
ory/v1/r12
```

- Reports:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Reports/v1/r12
```

- End Customer management:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/EndCustomers/v1/
r12
```

- User management:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Users/v1/r12
```

- Supplementary services management:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/SupplementarySer
vices/v2/r12
```

- Prepaid voucher management:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Vouchers/v1/r12
```

- SIM swap:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Simswap/v1/r12
```

- Platform status checking:

```
https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Echo/v1/r12
```

### 4.3 How to communicate with Kite Platform API?

The API uses the protocol SOAP; a standard that allows the definition (using an XML interface) of a group of available operations, the input and output interfaces and the description of the types of the exchanged data.

Each SOAP request is embedded within an HTTP message and as the version of SOAP is 1.1:

- All the HTTP messages use the POST method.
- The MIME type used is text/XML.

The API returns a 200 code in case of success and a 500 code if the request generates an exception (either because of a server failure or because of a violation of the business logic of the Kite Platform).

Kite Platform API is an **API UNICA,** this means that it follows a common and homogeneous style and design guideline for the rest of the APIs UNICA, defined by Telefónica facilitate access to its services layer.

The purpose of this document is not to cover the level of implementation detail, which is provided in the following documents that are also available for reference:

- **XSD/WSDL files** which define the SOAP interface, input/output parameters, and types.

These documents can be downloaded directly from the Kite Platform through the link: Help -> Manual download section

- Document describing the **Binding SOAP of Kite Platform APIs** with an explanation of the types and operations gathered in the XSD/WSDL, examples of requests, answers, and exceptions returned by the API.
- **Common UNICA document** defining common details (both at a functional level – agnostic in relation to the implementation and at an implementation level –used technology: e.g. SOAP) to all APIs UNICA.

The operations exposed in the API have a maximum runtime that the customer must respect, to avoid considering as failed operations the ones finished successfully in the platform.

In general, any operation has to end in 70 seconds, as a maximum time. So, it is recommended wait this amount of time, before reattempting a request.

> Some operations in the Inventory and Prebill API that, for processing requirements, have a maximum runtime of 3 minutes.
>
> The execution time of the lifecycle change operation from Inactive New to Trial or Activation Pending, can be different from 70 seconds. Please, you can check with your service provider the time to complete this operation.

### 4.3.1 CURL call

Below, is an example to call the API using the CURL command:

> From 4.0.1 version, only TLSV 1.2 is allowed, and you shall use CURL from 7.34.0

1. Install **openssl**, if you do not have it installed:

```
sudo apt-get install openssl
```

2. Extract the public key from Customer certificate (its nomenclature is similar to *Customer_certificate.pfx*, just change the first part with your username). You will need the password provided by the Kite Platform Support Team:

```
openssl pkcs12 -in Customer_certificate.pfx -clcerts -nokeys -
out Customer_certificate.cer
```

3. Extract the private key from the Customer certificate. You will need the password provided by the Kite Platform Support Team:

```
openssl pkcs12 -in Customer_certificate.pfx -nocerts -nodes  -
out Customer_certificate.key
```

Starting with openssl version 3.0, you must use the legacy option:

```
openssl pkcs12 -in Customer_certificate.pfx -nocerts -nodes
-legacy -out Customer_certificate.key
```

4. Following, there is an example about the access to the Commercial Plans API REST using the CURL command:

```
sudo curl --cert ./Customer_certificate.cer --key
./Customer_certificate.key https://kiteplatform-
api.telefonica.com:8010/services/REST/GlobalM2M/ServicePacks/v2/
r12/servicePack
```

5. The next example shows how to invoke the Echo API REST to check the connection:

```
 sudo curl -X POST -H "Content-Type: application/json" -H
"Accept-Encoding: gzip,deflate" -H "Cache-Control: no-cache" --
cert ./Customer_certificate.cer --key
```

```
./Customer_certificate.key https://kiteplatform-
api.telefonica.com:8010/services/REST/GlobalM2M/Echo/v1/r12/echo
 -d '{"data":"test"}'
```

### 4.3.2 SoapUI call

SoapUI is a very versatile application that allows us to call the web services in an agile way, from the basis of their contract in WSDL format.

From 4.0.1 version, only TLSV 1.2 is allowed, SoapUI clients use TLSv1.0 by default and you shall enable TLSv1.2.

1) If you run the SoapUI from desktop icon, you shall add TLSv1.2 support into the file SoapUI-5.0.0.vmoptions:

-Dsoapui.sslcontext.algorithm=TLSv1.2

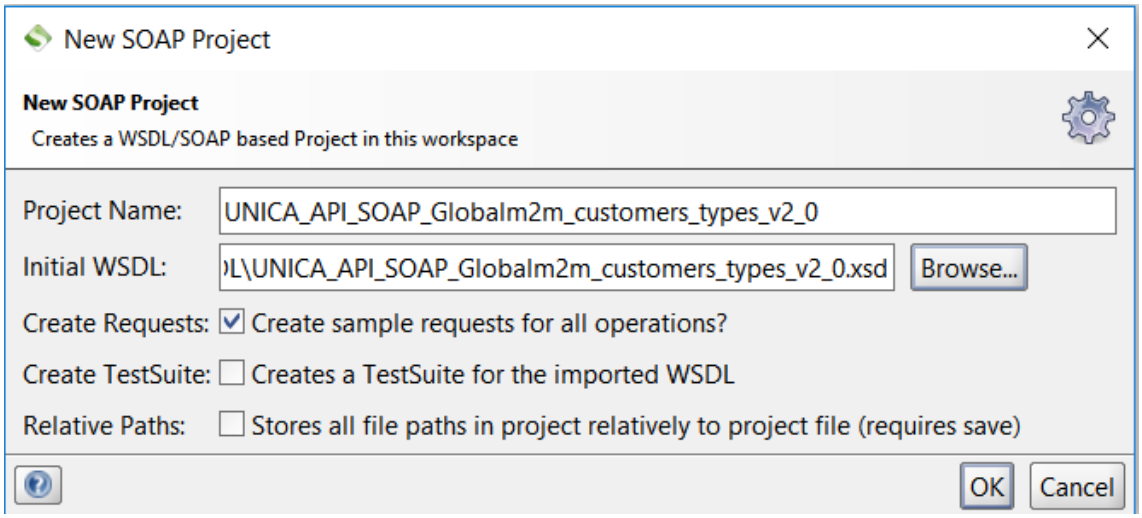
2) In you run the SoapUI from the start.bat, you shall add TLSv1.2 support into JAVA_OPTS:

-Dsoapui.https.protocols=TLSv1.2

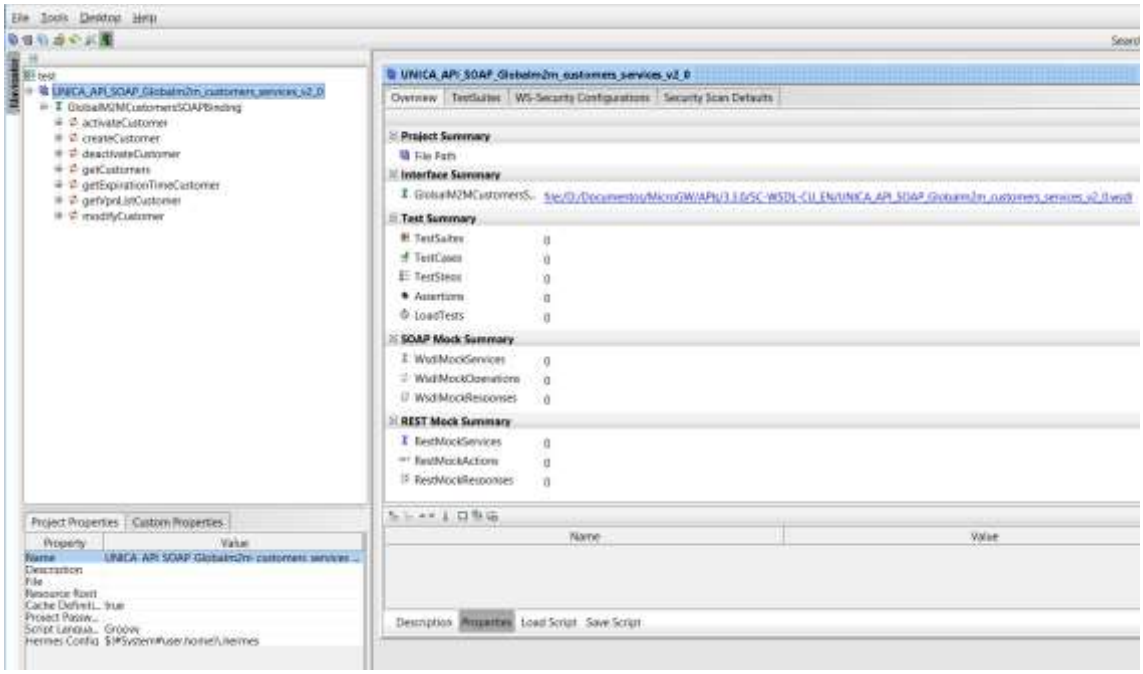
Pressing “Help” in the Kite Platform Portal you can download the WSDLs of the API.

Follow the next steps:

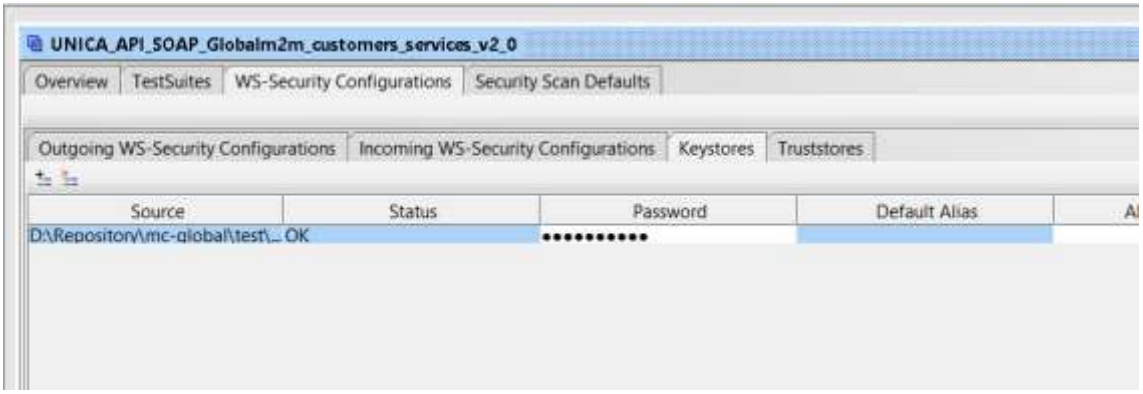
1. Select the option “Kite Platform APIs – WSDLs for Customers” to download the file *SC-WSDL-CU_EN.zip* that contains the WSDLs for the Customers API.
2. Unzip the file.
3. Open the soapUI and create a new project from the menu FILE > “New SOAP Project”.
4. Select “Browse” and import the WSDL file associated to the API we want to call.
5. Mark the checkbox in “Create sample requests for all the operations?” The soapUI will create the frame of the SOAP messages to call the API methods.



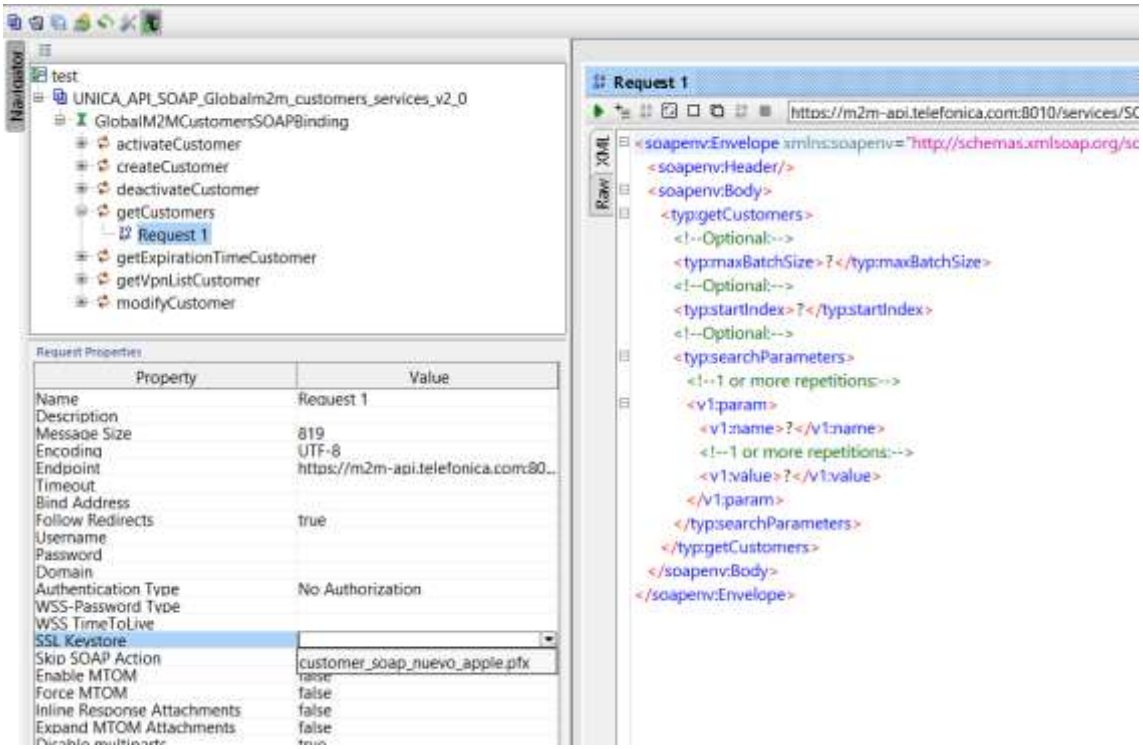

6. Following, import the Customer certificate given by the Kite Platform Support Team. To do so, double click over the new Project; a window in the right panel will be open.



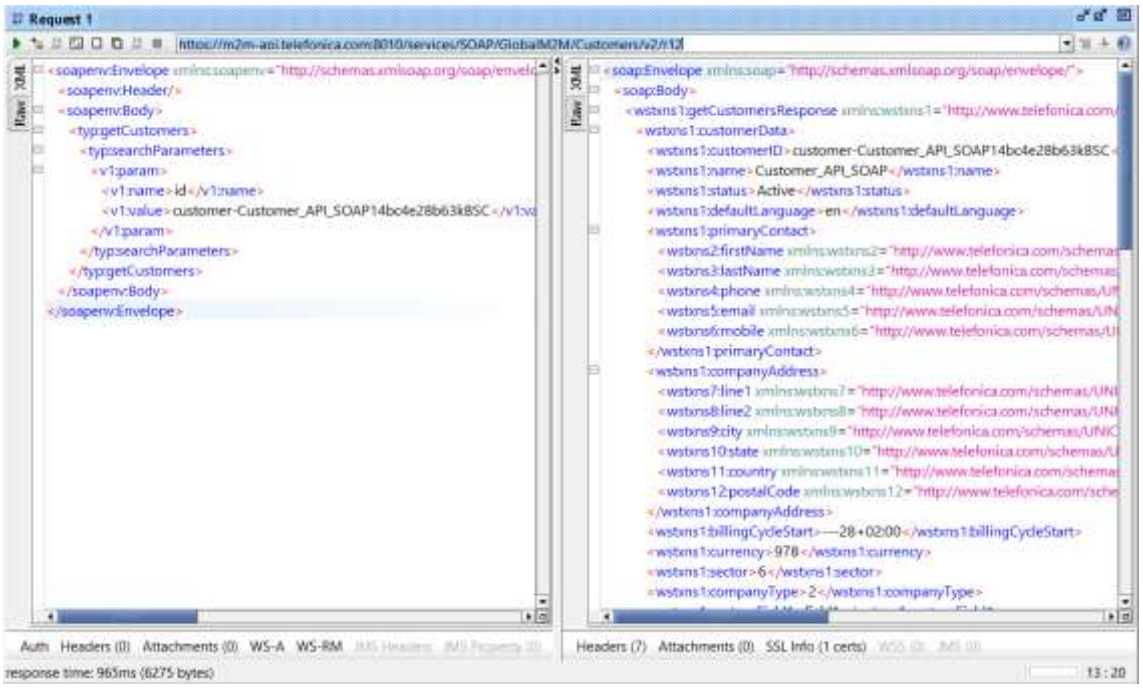
7. Select the tabs “WS – Security Configurations” > “Keystores” and import the certificate. You will need the password provided by the Kite Platform Support Team.



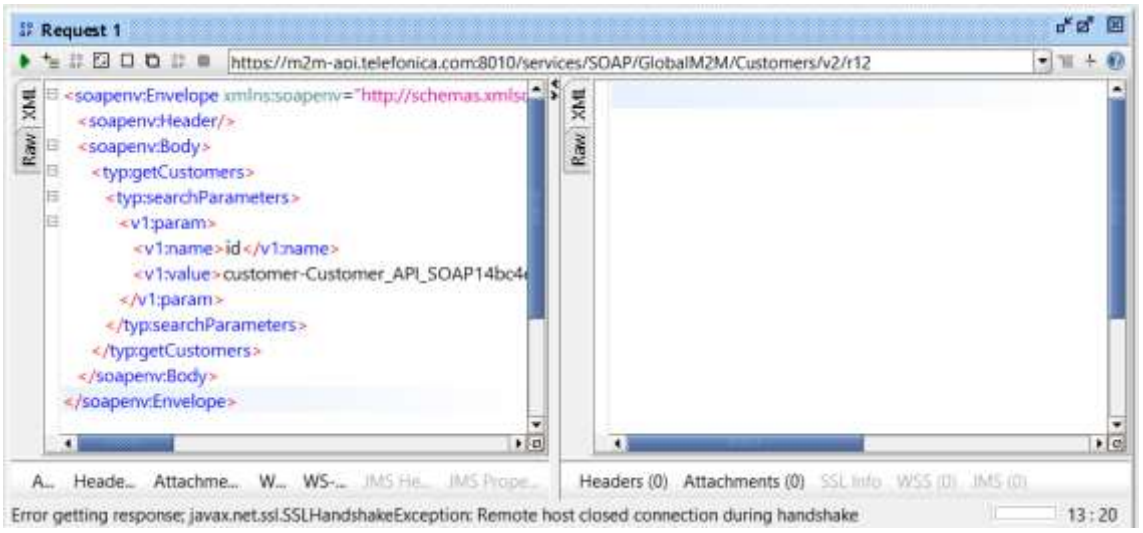

8. Once the keystore is imported, it is necessary configure the property “SSL Keystore” from the panel “Request properties”.
9. Pressing on the property, a drop-down menu will include the imported Customer certificate, so you can select it. The request will be send encrypted thanks to that certificate, so the server can recognize us.




10. If everything is configured correctly, you will receive a response in the right panel.




11. Otherwise, if you receive a "Remote Host closed connection during handshake", it is necessary check again that the certificate had been imported correctly and it is selected in the panel "Request Properties".



# 5 Available services

## 5.1 API to manage customers' sheet

This service allows modifying, listing, filtering and getting the VPN list and the expiration time about organizations of type *Customer*. The information of the customer account, which can be managed, includes the billing account by default.

**Operation modifyCustomer**: it modifies some information of the Customer's account in the Kite Platform service.

**Operation getCustomers**: it returns the own organization's information.

**Operation getVpnListCustomer**: it returns the own VPN list associated.

**Operation getExpirationTime**: returns the timeout, in seconds, that the customer needs to end the testing period.

## 5.2 API to manage the inventory

This service enables access to the subscriptions inventory, enumerating, details and modifying certain parameters of these subscriptions.

**Operation getSubscriptions**: it enumerates the subscriptions allowing to make a paginating and a filtering.

**Operation getPresenceDetail**: it obtains information of an existing subscription.

**Operation getLocationDetail**: it obtains the location of a subscription, not only the one collected automatically by a network but also the manually introduced one through the Kite Platform.

**Operation getStatusDetail**: it obtains expanded information of the actual life cycle status of a subscription.

**Operation getStatusHistory**: it obtains the life cycle status history through which a subscription has passed.

**Operation modifySubscription**: it allows modifying the life cycle status, customizable fields, billing account, APNs, static IP, expense limits, consumption thresholds, LTE and/or subscriptions group of a subscription.

**Operation getTimeAndConsumption**: it obtains the time and information combined ticket status of a subscription.

**Operation modifyTimeAndConsumption**: it modifies the time and information combined ticket status of a subscription.

**Operation sendSMS:** it allows sending an SMS to several subscriptions.

**Operation getSendSMSResult:** it obtains the SMS delivery status.

## 5.3 API to manage prebills

This service allows accessing to the pre-invoices created by the Kite Platform service.

**Operation getPrebills**: prebills list allowing making a filtering and pagination.

**Operation exportPrebill**: exports the prebill into XML format and returns the URI where the customer can retrieve it.

## 5.4 API to manage billing accounts

This service allows managing the billing accounts of a Customer, allowing listing and obtaining information from them.

**Operation getBillingAccounts**: it enumerates the billing accounts of a Customer.

**Operation getBillingAccount**: it consults the details of a billing account.

## 5.5 API to manage subscriptions groups

This service allows creating, modifying, enumerating and erasing subscriptions' groups.

**Operation createCommercialGroup**: it creates a new subscription group to the Customer who makes the request.

**Operation deleteCommercialGroup**: it erases a subscription group belonging to the Customer who makes the request.

**Operation getCommercialGroups**: it enumerates the subscriptions' groups of the Customer who makes the request.

**Operation getCommercialGroup**: it retrieves the subscription group information belonging to the Customer who makes the request.

**Operation updateCommercialGroup**: it modifies some of the subscription group information which belongs to the Customer who makes the request.

**Operation updateScreeningLists**: it updates the telephone numbers of the subscriptions' groups black and white lists belonging to the Customer who makes the request.

## 5.6 API for basic services commercial plan listing

This service allows listing Commercial plans of Basic Services and is intended to create subscription groups, as subscriptions groups have as a parameter a commercial plan of basic services.

**Operation getServicePacks:** it lists the available basic services commercial plans for the Customer.

## 5.7 API for diagnostics

This service allows Diagnostics over a SIM.

**Operation administrativeDiagnostic:** it performs an administrative diagnostic of the given subscription.

**Operation ipDiagnostic:** it performs an IP diagnostic of the given subscription. This operation is asynchronous

**Operation getDiagnosticResult:** it gets the result of IP diagnostic, as those are asynchronous operations.

**Operation gsmSyncDiagnostic:** it performs a GSM diagnostic of the given subscription. This operation is synchronous.

**Operation gprsSyncDiagnostic:** it performs a GPRS diagnostic of the given subscription. This operation is synchronous.

## 5.8 API to manage asynchronous reports

This service allows creating and downloading reports about consumption, presence, network errors...

**Operation createDownloadReport:** it requests the creation of a report in the background to download it later.

**Operation getDownloadReportList:** it lists the reports that an API consumer has requested to be generated in the background.

**Operation getDownloadReportLink:** it obtains the download link of a report that previously has been requested to be generated in the background.

**Operation deleteDownloadReport:** it deleted a report that has been generated using the asynchronous report interface, allowing the platform to deallocate resources.

## 5.9 API to manage end customers

This service allows the management of End Customers.

**Operation createEndCustomer:** it creates a new End Customer.

**Operation getEndCustomers:** it lists the End Customers.

**Operation getEndCustomer:** it retrieves information of an End Customer.

**Operation modifyEndCustomer:** it modifies an End Customer.

**Operation deleteEndCustomer:** it deletes an End Customer.

**Operation deactivateEndCustomer:** it deactivates an End Customer.

**Operation activateEndCustomer:** it activates an End Customer.

## 5.10 API to manage users

This service allows the creation of user accounts to access the Kite Platform (self-management web portal).

**Operation createUser:** it creates a new user.

**Operation deleteUser**: it deletes a user.

**Operation modifyUser**: it modifies a user.

**Operation getUsers:** it lists the users.

**Operation blockUser:** it blocks a user, so he cannot access to the Kite Platform.

**Operation unblockUser:** it unblocks a user, so he has rights again to login in Kite Platform.

**Operation getRoles:** it lists available roles; the role is needed as a parameter for user creation.

**Operation resetPassword:** it resets the password of a user, receiving an email with a link to activate the account configuring a new password.

## 5.11 API for supplementary service listing

This service allows retrieving the supplementary service associated with the Customer.

**Operation getSuppplementaryServices:** it retrieves the supplementary service information belonging to the Customer who makes the request.

**Operation getSuppplementaryServiceDetail:** it retrieves the supplementary service detail information belonging to the Customer who makes the request.

## 5.12 API to manage prepaid vouchers

This service allows the access to manage the prepaid vouchers associated to the Customer's subscription.

**Operation getAvailableVoucher**: it obtains a list of the available prepaid vouchers that can be bought for a subscription.

**Operation getActiveVouchers**: it obtains a list of the active prepaid vouchers not expired.

**Operation purchaseRecurrentVoucher:** it buys a prepaid voucher with an automatic renewal for the specified subscription.

**Operation purchaseOneTimeVoucher:** it buys a prepaid voucher without automatic renewal for the specified subscription.

**Operation cancelVoucher:** it cancels the non-recurring prepaid voucher specified.

**Operation cancelRecurrentVoucherRenewal:** it cancels the recurring prepaid voucher specified.

## 5.13 API for SIM swap

This service allows the exchange of the internal data of a chip (ICC-IMSI) between two subscriptions of the Inventory.

| Only available for Mexico OB. |
|---|

**Operation swapSubscriptions**: allows the exchange of the internal data of a chip (ICC-IMSI) between two subscriptions of the Inventory. That means, there is a transfer of the MSISDN from a SIM to another, making possible that this last inherits the history, the consume expenses and the billing state of the original SIM. Usually, this exchange is due to a broken device or because of the acquisition of a new one.

## 5.14 API for checking platform status

**Operation echo**: this operation allows checking if the Platform is up by sending a text which shall be returned in the response, otherwise, the Platform is down.

# 6 FAQs

## 6.1 Validation of the intermediate CAs certificates

DigiCert signs the server certificates of Kite Platform, both in Preproduction and Production. Consequently, in order to be able to use the API Service of Kite Platform, the customer must aggregate the Intermediate CAs certificates to the System Certificates Store.

> It is not recommended to directly validate the API Service Certificate of Kite Platform. This certificate is renewed each new year, thus, the consumer of the API could have security problems when changing the server certificate.

For the certificate submitted for SM2M, the CAs of the route certificate that should be added to the certificates store are:

- Root: DigiCert SHA2 Secure Server CA.
- Intermediate: DigiCert SHA2 Secure Server CA

The most convenient option is to validate only the CAs signers of the certificate and not the server certificate.

## 6.2 Pagination mistakes

The queries in the subscriptions inventory include pagination parameters to request a certain page with a particular size. For that, it is necessary an initial element (*startIndex*) and a number of required elements (*maxBatchSize*). Therefore, the solicited page will be the elements between *startIndex* and *startIndex + maxBatchSize*.

If these elements are not included, the default paging will be applied and the first ten subscriptions will be shown.

> Started with the version 3.2, it is not obligatory that the *startIndex* is a multiple of *maxBatchSize*, thanks to that, the paging can start at any moment.
>
> The default value of *maxBatchSize* is 10 and the maximum value 1000.
>
> The element *startIndex* can take any value equal or bigger than zero.

The next snippet code is an example of a request with paging:

```
POST https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Inventory/v4/r12

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2
m/inventory/v4/types"
xmlns:v1="http://www.telefonica.com/schemas/UNICA/SOAP/common/v1
">
```

```
   <soapenv:Header/>
   <soapenv:Body>
      <typ:getSubscriptions>
         <typ:maxBatchSize>100</typ:maxBatchSize>
         <typ:startIndex>200</typ:startIndex>
      </typ:getSubscriptions>
   </soapenv:Body>
</soapenv:Envelope>
```

If the requested page does not contain any element, the response is empty:

```
HTTP/1.1 200 OK

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <wstxns1:getSubscriptionsResponse
xmlns:wstxns1="http://www.telefonica.com/schemas/UNICA/SOAP/Glob
alm2m/inventory/v9/types"/>
   </soap:Body>
</soap:Envelope>
```

The common procedure is the following:

12. If we are iterating over the pages consecutively, each request receives a number of subscriptions equal to the value of the parameter *maxBatchSize*. When the *maxBatchSize* is not indicated, 10 subscriptions are displayed from the *startIndex*, otherwise, from the beginning of the inventory.
13. If the returned number of subscriptions is less than the requested number of subscriptions that means that the required page is the last one available. There is an exception to this case, when the last page has, exactly, the same number of elements than the requested ones; the only way to know if it is the last page will be asking for the next one.
14. If one more page is submitted, the result will be an empty page.

## 6.3 Availability periods

Throughout the day, some provision operations cannot be done because of the unavailable periods of the system, used to make maintenance tasks. This unavailability can be identified when, as a response to an operation, the next error is received: *Service temporarily unavailable: <detail of the service>.*

When this error appears, it is preferable to avoid retry immediately, as the unavailability time will be higher than the timeout by default. It is better to wait until the unavailability period finishes because during this time the response will be wrong.

The unavailability periods are:

- First Sunday of each month, since 00:59 in Europe/Madrid time zone for Spain HLR and 00:59 US/Eastern (06:59 Madrid time) for Latam HLR, for 61 min
- Every day, since 01:59 in Europe/Madrid for Spain HLR and US/Eastern (07:59 Madrid time) for Latam HLR, for 17 minutes

These tasks will affect the following operations in API:

- modifySubscription in the API to manage the inventory.
- modifyCustomer in the API to modify the customer's sheet.

From time to time, it is possible to find other no-planned unavailability periods that can affect to a higher number of operations. In these cases, it is urgent to apply the same politic of delayed reattempts.

## 6.4 Reach the limit of requests

The system is designed to share resources among users, so it has been established a maximum number of operations that a certain customer can do in a particular period. When this maximum is reached, the next error is returned:

```
{
  "ClientException": {
    "exceptionId": 1005,
    "text": "Request Rate Per Period Control Not Pass",
    "exceptionCategory": "POL"
  }
}
```

It is recommendable to configure the schema of reattempts to wait, at least, one second. If after this second, the error persists, the waiting time must be increased.

Telefónica can decide to vary the maximum of requests, temporally or permanently, to assure the best quality of services for their customers.

## 6.5 Asynchronous reports

### 6.5.1 Creation and download

All the asynchronous reports follow the same method for its creation. Here, we are going to see how to create a Presence Report through the API REST.

#### 6.5.1.1 Creation of the report

The following snippet of code is an example of the request to create the Presence Report:

```
POST https://kiteplatform-
api.telefonica.com:8010/services/REST/GlobalM2M/Reports/v1/r12/r
eport/downloadReport

{
```

```
"presence_monthly": {
     "supervisionGroup": "default_group",
          "date": "2016-10"
     }
}
```

#### 6.5.1.2 Visualization of the report

To obtain the report, it is necessary to know its identifier. This data is obtained after making the next query and get a list of the created reports. The identifier will be in the field d*ownloadReportId.*

```
GET https://kiteplatform-
api.telefonica.com:8010/services/REST/GlobalM2M/Reports/v1/r12/
report/downloadReport
```

The following snippet shows an example where it is possible to visualize the field *downloadReportId.*

```
HTTP/1.1 200 OK
{
    "downloadReports": [
        {
            "downloadReportId": "588f2386e907ac26ed820f80",
            "created": "2017-01-30T11:29:10Z",
            "expire": "2017-01-31T11:29:10Z",
            "modified": "2017-01-30T11:29:10Z",
            "status": "finished",
            "params": {
                "presence_monthly": {
                    "date": "2016-10",
                    "supervisionGroup": "default_group",
                    "orgId": "customer-
Customer_API_SOAP14bc4e28b63kBSC"
                }
            },
            "remove": "2018-01-30T11:29:10Z"
        }
    ]
}
```

#### 6.5.1.3 Download link

The following snippet shows the method to obtain the download link:

```
GET https://kiteplatform-
api.telefonica.com:8010/REST/GlobalM2M/Reports/v1/r12/report/dow
nloadReport/588f2386e907ac26ed820f80/link
```

The response will show the next information:

```
HTTP/1.1 200 OK
{
    "resultURL": [
```

```
        "http://m2m-movistar-es.cloncloud-
m2mglobserv02.hi.inet/download/3pKv0q63n04mEdCn9GBvXE94MGkr2zXeK
P494cS78O9A1cvDk3D21pvYZiCy"
    ]
}
```

#### 6.5.1.4 Download the report

The method GET applied to the link allows the download of the report without authentication. The response is a CSV file associated with the requested report.

```
GET https://kiteplatform-
api.telefonica.com:8010/api/download/x02NJJx0wkf06YEvE4kv4cdsVa6
0IA25F73U6G4Iusz5MF3K35yr0YbvDYQb
```

Before the creation and download of the report, it is recommended calculate a delay of 10 seconds to allow the end of the asynchronous tasks.

The download only can occur one time, since the associated token expires after the use of the link. In any case, if the customer tries to use the download link two times or more will see the next error:

```
HTTP/1.1 400 OK
{
   "message": "Token does not exist or has expired",
   "error": 400
}
```

### 6.5.2 Format

The report is downloaded in a ZIP file and each one can contain one or more CSV. In turn, the CSV can contain, as maximum, one million of lines, which is the maximum number of lines supported by Excel 2007.

The format for the name of the CSV file is:

[tipo]_[organizationId]_[transactionId]_[periodo].csv

### 6.5.3 Availability

- **Daily reports about presence:** it is only possible to create and/or visualize the ones between the current day and the previous N days. The number of days is not fixed, as it depends on the difference in days between the first day of the month two months ago. If today is May 29, 2025, you can only order from March 1st onwards. This kind of report is created using the particular day.
- **Daily reports about alarms**: it is only possible to create and/or visualize the ones between the day before the petition and the previous 100 days. It is not available for the present day. This kind of report is created using the particular day.
- **Daily location**: this kind of report is created using the particular day. It is only possible to create and/or visualize the ones between:

  - **Advanced location**: current and previous day
  - **Basic location**: current and the previous 100 days

- **Monthly reports about presence, location, and alarms**: it is only possible to create and/or show the ones between the month before the petition and the previous 3 months. It is not available for the present month. This kind of report is created using the particular month.
- **Accumulated daily summary about presence**: it is only possible to create and/or visualize the ones between the previous month to the petition and the previous 3 months. It is not available for the present month. This type of report is created using the particular month.
- **Daily consumption**: in a range of two dates, as long as it is below the 31 days. This type of report is created indicating the start and finish date.
- **Monthly expense detail**: it is only possible to create and/or visualize the ones between the previous month to the petition and the previous 24 months. It is not available for the present month. This type of report is created indicating the particular month.
- **Daily consumption detail**: it is only possible to create and/or visualize the ones between the petition day and the previous 124 days. It is available for the present day. This type of report is created indicating the start and finish day.
- **Daily expense detail**: it is only possible to create and/or visualize the ones between the petition day and the previous 124 days. It is available for the present day. This type of report is created indicating the start and finish day.
- **Daily subscription snapshot**: it is only possible to create and/or show the ones between the previous two days to the petition and the previous 30 days. It is not available neither the previous day nor the present day. This type of report is created indicating the particular day.
- **Monthly state subscriptions:** it is only possible to create and/or visualize the ones between the present day and the last 12 months. This type of report is created indicating the particular month.
- **Charges aggregated**: it is only possible to create and/or visualize the ones between the previous month to the petition and the previous 11 months. It is not available for the present month. This type of report is created indicating the particular month.
- **Network errors**: it is only possible to create and/or visualize the ones between the prior day to the petition and the previous 31 days. It is not available for the present day. This type of report is created indicating the particular day. It will be available for downloading during three months.
- **API transactions**: it is only possible to create and/or visualize the ones between the current day to the petition and the previous 31 days. It is available for the present day. This kind of report is created indicating the particular day. It will be available for downloading during three months.

- **Retired daily subscriptions:** it is possible create and/or visualize the ones between the present day and the previous 732 days. Two-year periods with start date from 2011-01-01 and end date up to the present day are supported. This type of report is created indicating the start and finish day. It will be available for downloading during one year.
- **Monthly activity SWAP:** it is possible to create and/or visualize all the ones between the present month and the last 12 months. This type of report is created indicating the particular month. It will be available for downloading during three months.

| Only available for the customers with SWAP funcionality. |
|---|

- **Detail subscription billing:** it is possible to create and/or visualize all the ones between the present month and the last 24 months. This type of report is created indicating the billing cycle closing date and the related billing account. It is not available for the day when the cycle is closed. You must wait one day after. It will be available for downloading during one year.
- **Changed IMEI**: it is only possible to create and/or visualize the ones between the current day to the petition and the previous 30 days. It is available for the present day. This kind of report is created indicating the particular day. It will be available for downloading during three months.
- **Manufacturer model summary**: no date period is required. It is created on demand. It will be available for downloading during three months. It will be available for downloading during three months.
- **User list**: no date period is required. It is created on demand. It will be available for downloading during three months. It will be available for downloading during three months.
- **SMS delivery**: operation identifier of the sms sending through Inventory API. It will be available for downloading during one week.
- **Monthly activity alarms:** it is possible to create and/or visualize all the ones between the present month and the last 12 months. This type of report is created indicating the particular month. It will be available for downloading during three months.
- **Direct SIM manufacturer order:** this type of report is created indicating the particular order identifier. It will be available for downloading during five years.
- **Legacy monthly expense detail**: it is only possible to create and/or visualize the ones between the previous day to the petition and the previous 22 months. This type of report is created indicating the invoice date.

## 6.6 Flow of SMS with delivery report

The Inventory API allows the sending of SMS to an activated SIM. This action also allows you to generate and view a report about the state of the delivery, in a similar way to an asynchronous report.

There is a limit of 1000 SIMs per invocation.

The next example shows how to generate the report through an API SOAP.

### 6.6.1 Delivery

The next snippet shows the request for the delivery of an SMS and activate the option of delivery report

```
POST https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Inventory/v9/r12

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2
m/inventory/v9/types">

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2
m/inventory/v9/types">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:sendSMS>
         <!--1 or more repetitions:-->
         <typ:destSubscriptions>
            <!--You have a CHOICE of the next 4 items at this
level-->
            <typ:icc>8934072100251270859</typ:icc>
         </typ:destSubscriptions>
         <typ:text>hello world</typ:text>
         <typ:hasDeliveryReport>true</typ:hasDeliveryReport>
      </typ:sendSMS>
   </soapenv:Body>
</soapenv:Envelope>
```

The response returns an identifier of the asynchronous task, used to create the delivery report.

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <wstxns1:sendSMSResponse
xmlns:wstxns1="http://www.telefonica.com/schemas/UNICA/SOAP/Glob
alm2m/inventory/v9/types">
         <wstxns1:watcherId>56c32ac8e907ac270793e2bb</wstxns1:wa
tcherId>
      </wstxns1:sendSMSResponse>
   </soap:Body>
</soap:Envelope>
```

### 6.6.2 Creation

The next snippet shows an example about a request to create the delivery report associated to the SMS sending identifier.

```
POST https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Reports/v1/r12

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2
m/reports/v1/types">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:createDownloadReport>
         <typ:sms_ao>
            <typ:watcherId>56c32ac8e907ac270793e2bb
</typ:watcherId>
         </typ:sms_ao>
      </typ:createDownloadReport>
   </soapenv:Body>
</soapenv:Envelope>
```

### 6.6.3 Visualization

To obtain a report about the SMS delivery is necessary to know the identifier. The next snippet shows the query to check the created reports:

```
POST https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Reports/v1/r12

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2
m/reports/v1/types">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:getDownloadReportList/>
   </soapenv:Body>
</soapenv:Envelope>
```

The response is a list of the created reports. Just look in the field *downloadReportId* to know the identifier and find the correct report.

```
HTTP 1.1 200 OK

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <wstxns1:getDownloadReportListResponse
xmlns:wstxns1="http://www.telefonica.com/schemas/UNICA/SOAP/Glob
alm2m/reports/v1/types">
         <wstxns1:downloadReports>
```

```
            <wstxns1:downloadReportId>56c32ad3e907ac270793e2bd</
wstxns1:downloadReportId>
            <wstxns1:created>2016-02-
16T13:57:39Z</wstxns1:created>
            <wstxns1:expire>2016-02-
17T14:57:39Z</wstxns1:expire>
            <wstxns1:modified>2016-02-
16T13:57:39Z</wstxns1:modified>
            <wstxns1:status>active</wstxns1:status>
            <wstxns1:params>
               <wstxns1:sms_ao>
                  <wstxns1:watcherId>56c32ac8e907ac270793e2bb</w
stxns1:watcherId>
               </wstxns1:sms_ao>
            </wstxns1:params>
            <wstxns1:remove>2016-02-
23T14:57:39Z</wstxns1:remove>
         </wstxns1:downloadReports>
      </wstxns1:getDownloadReportListResponse>
   </soap:Body>
</soap:Envelope>
```

### 6.6.4 Download link

The request to obtain the download link is as follows:

```
POST https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Reports/v1/r12

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2
m/reports/v1/types">\r
   <soapenv:Header/>
   <soapenv:Body>
      <typ:getDownloadReportLink>
         <typ:downloadReportId>56c32ad3e907ac270793e2bd
</typ:downloadReportId>
      </typ:getDownloadReportLink>
   </soapenv:Body>
</soapenv:Envelope>
```

The response returns the link:

```
HTTP 1.1 200 OK

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <wstxns1:getDownloadReportLinkResponse
xmlns:wstxns1="http://www.telefonica.com/schemas/UNICA/SOAP/Glob
alm2m/reports/v1/types">
```

```
         <wstxns1:resultURL>https://kiteplatform-
api.telefonica.com:8010/api/download/x02NJJx0wkf06YEvE4kv4cdsVa6
0IA25F73U6G4Iusz5MF3K35yr0YbvDYQb</wstxns1:resultURL>
      </wstxns1:getDownloadReportLinkResponse>
   </soap:Body>
</soap:Envelope>
```

### 6.6.5 Download the report

Use a GET method to get the report without authentication. The petition returns a CSV file with all the details about the delivery of SMS.

```
GET https://kiteplatform-
api.telefonica.com:8010/api/download/x02NJJx0wkf06YEvE4kv4cdsVa6
0IA25F73U6G4Iusz5MF3K35yr0YbvDYQb
```

Before the creation of the report, it is recommended to calculate a delay of 10 seconds to allow the end of the asynchronous tasks

The download only can occur one time, since the associated token expires after the use of the link. In any case, if the customer tries to use the download link two times or more will see the next error:

```
{
   "message": "Token does not exist or has expired",
   "error": 400
}
```

## 6.7 Obtain the current consumption

The inventory API allows knowing the consumption of a subscription during the current billing cycle. Also, this data can be obtained from the generation of a daily consumption report.

It is recommended to generate the Daily consumption report. It is an asynchronous operation and has a lower impact.

### 6.7.1 Inventory

The next snippet code is an example of the request in the inventory:

```
POST https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Inventory/v9/r12

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2
m/inventory/v9/types"
xmlns:v1="http://www.telefonica.com/schemas/UNICA/SOAP/common/v1
">
   <soapenv:Header/>
```

```
   <soapenv:Body>
      <typ:getSubscriptions>

         <typ:searchParameters>
            <v1:param>
               <v1:name>imsi</v1:name>
               <v1:value>214071418484295</v1:value>
            </v1:param>
         </typ:searchParameters>

      </typ:getSubscriptions>
   </soapenv:Body>
     </soapenv:Envelope>
```

In the response, the field *value* is the datum about the daily or monthly consumption, taking into account the time zone and the billing cycle of the customer.

The voice consumption is in seconds and the data consumption in bytes.

```
….
            <consumptionDaily>
               <voice>
                  <limit>0</limit>
                  <value>30</value>
                  <thrReached>0</thrReached>
               </voice>
               <sms>
                  <limit>0</limit>
                  <value>21</value>
                  <thrReached>0</thrReached>
               </sms>
               <data>
                  <limit>0</limit>
                  <value>0</value>
                  <thrReached>0</thrReached>
               </data>
            </consumptionDaily>
            <consumptionMonthly>
               <voice>
                  <limit>0</limit>
                  <value>4031</value>
                  <thrReached>0</thrReached>
               </voice>
               <sms>
                  <limit>0</limit>
                  <value>172</value>
                  <thrReached>0</thrReached>
               </sms>
               <data>
                  <limit>0</limit>
                  <value>200001</value>
                  <thrReached>0</thrReached>
               </data>
```

```
            </consumptionMonthly>
……
```

### 6.7.2 Reports

The next example shows the request to know the consumption of a subscription using a daily consumption report:

```
POST https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Reports/v1/r12

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2
m/reports/v1/types">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:createDownloadReport>
         <typ:consumption_daily>
         <!--Optional:-->
         <typ:showCommercialGroup>true</typ:showCommercialGroup>
         <!--Optional:-->
         <typ:showBillingAccount>true</typ:showBillingAccount>
         <!--Optional:-->
         <typ:showSubscriptions>true</typ:showSubscriptions>
         <typ:start>2017-01-01</typ:start>
         <typ:end>2017-01-30</typ:end>
         </typ:consumption_daily>
      </typ:createDownloadReport>
   </soapenv:Body>
</soapenv:Envelope>
```

See the 6.5 Asynchronous reports chapter to download the created report .

The next example shows the report as CSV file:

```
Date;Service Provider;Customer;Fiscal Number;Commercial Group
ID;Commercial Group Name;Billing Account ID;Billing Account
Name;Subscription ID;Voice;SMS;Data
01/01/2017;;Customer_API_SOAP;12345678L;;;0;default;55970765060;
0;0;0
```

## 6.8 WSDL versioning

The releases of Kite Platform, without changes in a certain API, don't change the versioning of its WSDL. However, every time there is a change, the WSDL versioning increases. The numbering for the WSDL releases not necessary has to match with the one in Kite Platform.

## 6.9 TPSs values for each operation

Each API operation has an individual configuration for a limit of requests performed by the users of an organization during a period of time. This avoids the overloading in the heaviest operations.

> In addition to the limit of requests for each operation, there is also a maximum TPS value for each organization set to 14.

The next snippet code is an example of the error received when the TPS value is reached:

```
HTTP/1.1 500 Internal Server Error

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <soap:Fault
xmlns:fault="http://www.telefonica.com/wsdl/UNICA/SOAP/common/v1
/faults">
         <faultcode>soap:Client</faultcode>
         <faultstring>Client exception</faultstring>
         <detail>
            <fault:ClientException
xmlns:wstxns1="http://www.telefonica.com/schemas/UNICA/SOAP/comm
on/v1">

<wstxns1:exceptionCategory>POL</wstxns1:exceptionCategory>
               <wstxns1:exceptionId>1005</wstxns1:exceptionId>
               <wstxns1:text>Request Rate Per Period Control Not
Pass </wstxns1:text>
            </fault:ClientException>
         </detail>
      </soap:Fault>
   </soap:Body>
</soap:Envelope>
```

The following tables show the TPS value (transactions per second) for each operation, according to the API:

| API | Operation | TPS |
|---|---|---|
| Clients | getCustomers | 4 |
| | modifyCustomer | 4 |
| | getVpnListCustomer | 4 |
| | getExpirationTimeCustomer | 4 |

| API | Operation | TPS |
|---|---|---|
| Inventory | getSubscriptions | 4 |

| | | |
|---|---|---|
| | getSubscriptionDetail | 2 |
| | getPresenceDetail | 4 |
| | getLocationDetail | 4 |
| | getStatusDetail | 4 |
| | getStatusHistory | 4 |
| | modifySubscription | 2 |
| | getTimeAndConsumption | 2 |
| | modifyTimeAndConsumption | 2 |
| | sendSMS | 2 |
| | getSendSMSResult | 4 |
| | block | 4 |
| | consumptionHistory | 1 |
| | gprsHistory | 1 |

| API | Operation | TPS |
|---|---|---|
| Prebills | getPrebills | 4 |
| | exportPrebill | 4 |

| API | Operation | TPS |
|---|---|---|
| Billing account | getBillingAccount | 4 |
| | getBillingAccounts | 4 |

| API | Operation | TPS |
|---|---|---|
| Subscriptions groups | createCommercialGroup | 4 |
| | getCommercialGroup | 4 |
| | getCommercialGroups | 4 |
| | deleteCommercialGroup | 4 |
| | updateCommercialGroup | 4 |
| | updateScreeningLists | 2 |
| | updateLpwaPowerSaving | 4 |

| API | Operation | TPS |
|---|---|---|
| Commercial plans | getServicePacks | 4 |

| API | Operation | TPS |
|---|---|---|
| Diagnosis | administrativeDiagnostic | 2 |
| | gsmSyncDiagnostic, gprsSyncDiagnostic, ipDiagnostic | 2 |
| | getDiagnosticResult | 4 |

| API | Operation | TPS |
|---|---|---|
| Reports | createDownloadReport | 2 |
| | getDownloadReportList | 4 |

| | | |
|---|---|---|
| | getDownloadReportLink | 4 |
| | deleteDownloadReport | 4 |
| | createDownloadReport | 4 |

| API | Operation | TPS |
|---|---|---|
| End customers | createEndCustomer | 4 |
| | getEndCustomers | 4 |
| | getEndCustomer | 4 |
| | modifyEndCustomer | 4 |
| | deleteEndCustomer | 4 |
| | activateEndCustomer | 4 |
| | deactivateEndCustomer | 4 |

| API | Operation | TPS |
|---|---|---|
| Users | createUser | 4 |
| | deleteUser | 4 |
| | modifyUser | 4 |
| | getUsers | 4 |
| | blockUser | 4 |
| | unblockUser | 4 |
| | getUserRoles | 4 |
| | resetPassword | 4 |

| API | Operation | TPS |
|---|---|---|
| Supplementary Services | getSupplementaryServices | 2 |
| | getSupplementaryServiceDetail | 2 |

| API | Operation | TPS |
|---|---|---|
| Prepaid vouchers | getAvailableVouchers | 4 |
| | purchaseOneTimeVoucher | 4 |
| | getActiveVouchers | 4 |
| | cancelRecurrentVoucherRenewal | 2 |
| | cancelOneTimeVoucher | 2 |

| API | Operation | TPS |
|---|---|---|
| SIMs swap | swapSubscriptions | 4 |

| API | Operation | TPS |
|---|---|---|
| State | echo | 4 |

## 6.10 Permissions of the Customer certificate

The permissions of the certificate are the same that an ADMIN user in the Portal may have. However, the allowed operations change depending on the organization (Customer or Service), even though the role is always ADMIN.

| The ADMIN role performs CRUD operations (Creation, Updating and Deleting). |
|---|

### 6.10.1 Error in unauthorised operation

In the case of an unauthorised operation, the platform returns an error SVC 1013.

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <soap:Fault
xmlns:fault="http://www.telefonica.com/wsdl/UNICA/SOAP/common/v1
/faults">
         <faultcode>soap:Client</faultcode>
         <faultstring>Client exception</faultstring>
         <detail>
            <fault:ClientException
xmlns:wstxns1="http://www.telefonica.com/schemas/UNICA/SOAP/comm
on/v1">

<wstxns1:exceptionCategory>SVC</wstxns1:exceptionCategory>
               <wstxns1:exceptionId>1013</wstxns1:exceptionId>
               <wstxns1:text>Operation is not allowed. Access to
perform "list" on a resource of type "organization" has been
denied for user with id "54eee152e907ac3fe338b4f5" and role
"admin-customer"</wstxns1:text>
            </fault:ClientException>
         </detail>
      </soap:Fault>
   </soap:Body>
</soap:Envelope>
```

## 6.11 Delete the Commercial Group assignation to an End Customer

The request to delete the assignation of a commercial group would be the same as the updating, but the value in the field “endCustomer” will be empty:

```
POST https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/CommercialGroup/
v4/r12

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2
m/commercialgroup/v2/types">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:updateCommercialGroup>
         <typ:id>43349</typ:id>
```

```
         <typ:endCustomer></typ:endCustomer>
      </typ:updateCommercialGroup>
   </soapenv:Body>
</soapenv:Envelope
```

## 6.12 Static APN

### 6.12.1 Adding a static APN

The following snippet of code is an example of the request to add a static APN to the subscription:

```
POST https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Inventory/v9/r12

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2
m/inventory/v9/types"
xmlns:v1="http://www.telefonica.com/schemas/UNICA/SOAP/common/v1
">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:modifySubscription>
         <!--You have a CHOICE of the next 4 items at this
level-->
         <typ:icc>89536222421979385264</typ:icc>
          <typ:staticIpAddress0><v1:ipv4>33.13.14.90</v1:ipv4>
          </typ:staticIpAddress0>
          <typ:apn0>api.customer.vpn</typ:apn0>
      </typ:modifySubscription>
   </soapenv:Body>
</soapenv:Envelope>
```

When consulting the data subscription, the response will be as follows:

```
<soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <wstxns1:getSubscriptionsResponse
xmlns:wstxns1="http://www.telefonica.com/schemas/UNICA/SOAP/Glob
alm2m/inventory/v8/types">
         <wstxns1:subscriptionData>
            <wstxns1:icc>89536222421979385264</wstxns1:icc>

……

            <wstxns1:apn/>
            <wstxns1:apn0>api.customer.vpn</wstxns1:apn0>
            <wstxns1:apn1/>
            <wstxns1:apn2/>
            <wstxns1:apn3/>
            <wstxns1:apn4/>
```

```
            <wstxns1:apn5/>
            <wstxns1:apn6/>
            <wstxns1:apn7/>
            <wstxns1:apn8/>
            <wstxns1:apn9/>

     <wstxns1:staticIpAddress0>33.13.14.90</wstxns1:staticIpAddr
ess0>
     <wstxns1:staticIpAddress1></wstxns1:staticIpAddress1>
     <wstxns1:staticIpAddress2></wstxns1:staticIpAddress2>
     <wstxns1:staticIpAddress3></wstxns1:staticIpAddress3>
     <wstxns1:staticIpAddress4></wstxns1:staticIpAddress4>
     <wstxns1:staticIpAddress5></wstxns1:staticIpAddress5>
     <wstxns1:staticIpAddress6></wstxns1:staticIpAddress6>
     <wstxns1:staticIpAddress7></wstxns1:staticIpAddress7>
     <wstxns1:staticIpAddress8></wstxns1:staticIpAddress8>
     <wstxns1:staticIpAddress9></wstxns1:staticIpAddress9>


……
         </wstxns1:subscriptionData>
      </wstxns1:getSubscriptionsResponse>
   </soap:Body>
</soap:Envelope>
```

### 6.12.2 Remove an static APN

The following snippet of code is an example of the request to remove the static APN from the subscription. It removes its static IP address as well.

```
POST https://kiteplatform-
api.telefonica.com:8010/services/SOAP/GlobalM2M/Inventory/v9/r12

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2
m/inventory/v9/types"
xmlns:v1="http://www.telefonica.com/schemas/UNICA/SOAP/common/v1
">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:modifySubscription>
         <!--You have a CHOICE of the next 4 items at this
level-->
         <typ:icc>89536222421979385264</typ:icc>
         <typ:apn0>-</typ:apn0>
      </typ:modifySubscription>
   </soapenv:Body>
</soapenv:Envelope>
```

When consulting the data subscription, the response will be as follows:

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <wstxns1:getSubscriptionsResponse
xmlns:wstxns1="http://www.telefonica.com/schemas/UNICA/SOAP/Glob
alm2m/inventory/v9/types">
         <wstxns1:subscriptionData>
            <wstxns1:icc>895362224219793852 64</wstxns1:icc>

……..

            <wstxns1:staticIp/>
            <wstxns1:apn/>
            <wstxns1:apn0/>
            <wstxns1:apn1/>
            <wstxns1:apn2/>
            <wstxns1:apn3/>
            <wstxns1:apn4/>
            <wstxns1:apn5/>
            <wstxns1:apn6/>
            <wstxns1:apn7/>
            <wstxns1:apn8/>
            <wstxns1:apn9/>
      <wstxns1:staticIpAddress0></wstxns1:staticIpAddress0>
      <wstxns1:staticIpAddress1></wstxns1:staticIpAddress1>
      <wstxns1:staticIpAddress2></wstxns1:staticIpAddress2>
      <wstxns1:staticIpAddress3></wstxns1:staticIpAddress3>
      <wstxns1:staticIpAddress4></wstxns1:staticIpAddress4>
      <wstxns1:staticIpAddress5></wstxns1:staticIpAddress5>
      <wstxns1:staticIpAddress6></wstxns1:staticIpAddress6>
      <wstxns1:staticIpAddress7></wstxns1:staticIpAddress7>
      <wstxns1:staticIpAddress8></wstxns1:staticIpAddress8>
      <wstxns1:staticIpAddress9></wstxns1:staticIpAddress9>
……..
         </wstxns1:subscriptionData>
      </wstxns1:getSubscriptionsResponse>
   </soap:Body>
</soap:Envelope>
```

## 6.13 Change the Commercial Group of the subscription

### 6.13.1 Get the target Commercial Group identifier

The request to retrieve the list of your commercial groups is as follows:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2m/commercialgroup/
v4/types">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:getCommercialGroups>
    </typ:getCommercialGroups>
  </soapenv:Body>
</soapenv:Envelope>
```

You can get the identifier related to the target commercial group name by inspecting into the response:

```
<soap:Envelope
   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <wstxns1:getCommercialGroupsResponse
xmlns:wstxns1="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2m/commercialg
roup/v4/types">
         <wstxns1:commercialGroup>
            <wstxns1:id>26</wstxns1:id>
            <wstxns1:name>CG_Test1</wstxns1:name>
            <wstxns1:description>description1</wstxns1:description>
            <wstxns1:customerId>customer-
Customer_API_SOAP14bc4e28b63kBSC</wstxns1:customerId>
         </wstxns1:commercialGroup>
         <wstxns1:commercialGroup>
            <wstxns1:id>61</wstxns1:id>
            <wstxns1:name>CG_Test2</wstxns1:name>
            <wstxns1:description>description2</wstxns1:description>
            <wstxns1:customerId>customer-
Customer_API_SOAP14bc4e28b63kBSC</wstxns1:customerId>
         </wstxns1:commercialGroup>
      </wstxns1:getCommercialGroupsResponse>
   </soap:Body>
</soap:Envelope>
```

### 6.13.2 Change the Commercial Group

The request to modify the commercial group of the subscription is as follows:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://www.telefonica.com/schemas/UNICA/SOAP/Globalm2m/inventory/v9/type
s">
  <soapenv:Header/>
  <soapenv:Body>
    <typ:modifySubscription>
      <typ:icc>8934071279000009098</typ:icc>
      <typ:commercialGroup>26</typ: commercialGroup>
    </typ:modifySubscription>
  </soapenv:Body>
</soapenv:Envelope>
```

The parameter commercialGroup shall contain the target commercial group identifier that it is numeric, not the name.

## 6.14 Error of access control when origin IP is not authorized

If you try to invoke the API through a machine whose IP is not included in the list of authorized subnets, you will get an error.

Example:


```
{
   "ClientException": {
      "exceptionId": "1013",
      "exceptionCategory": "SVC",
      "text": "Operation is not allowed: Remote IP '32.10.11.1' is not authorized"
   }
}
```