



**Business
Services**

M2M API for PHP

SDK Manual

M2M API for PHP: SDK Manual

Copyright © 2010 Orange

Revision History

Revision 1.0.6-000	2010/03/19
• First release.	

Table of Contents

Preface	iv
1. Overview	1
1. Features	1
2. Concepts	1
2. Prerequisites	3
1. What you need to consume M2M API web service	3
2. PHP requirements	3
3. PHP environment	4
4. Proxy and firewall settings	5
3. Install and Set Up	6
1. Install SDK	6
2. Configure SDK	6
3. One shot installation	7
4. First API call in PHP	11
4. The M2M SDK	12
1. Common Objects	12
2. M2M	14
3. SDK configuration	15
4. Logging	16
5. Methods	20
1. Basic scenarios	20
2. GetSubscriptionStatus method	21
3. GetConnectivityDirectory method	23
4. SearchInConnectivityDirectory method	26
5. UpdateConnectivityDirectory method	31
6. SubmitUpdateSimStatus and GetUpdateSimStatus methods	33
7. SubmitConsumptionTracking and GetConsumptionTracking methods	37
8. GetNetworkStatus method	41
9. GetIncidentDiagnostics method	43
10. SubmitSessionHistory and GetSessionHistory methods	45
A. Error codes	50
1. Connectivity	56

Preface

Welcome to the M2M API SDK manual!

This manual is a collection of topics related to develop code using all the advanced features of Orange™ M2M API with our SDK.

Note : M2M API is also known as Malima API.

By the end of this manual you will be able to make advanced applications using our API. You will discover the implementation details (architecture and libraries used) of SDK we made for you. Our SDK has been designed to hide all the complexity of the underlying Webservice communication and to help leverage the power of Object Oriented Programming.

You will first learn how to setup your machine, then how to configure the SDK and finally how to use every single methods provided by the API.

You will find some code snippets that are ready to go, just copy and paste the code into your favorite IDE!

Chapter 1. Overview

1. Features

The M2M webservice offers a way of managing SIM cards set, during the construction phase, the setting phase, or the operating phase.

M2M API offers the following features:

- Get the data set of a SIM card and refresh it.
- Get SIM information according to a collection of up to 100 line identifiers.
- Search SIM information.
- Update a set of SIM referentials, including devices and machines.
- Update a set of SIM status.
- Request for a set of SIM traffic in a period of time.
- Get the network covering status into an area defined by its latitude and longitude.
- Get the diagnostic analysis of a SIM card, including the network area covering the SIM card.
- Request for a SIM card set of statistics by session in a period of time.

2. Concepts

The provided services are web services included in a Software Development Toolkit, that can be inserted in software applications. They allow automating the deployment and the management of a set of machines, embedding SIM cards, and exchanging data with a central server thanks to a mobile connection (GSM / GPRS).

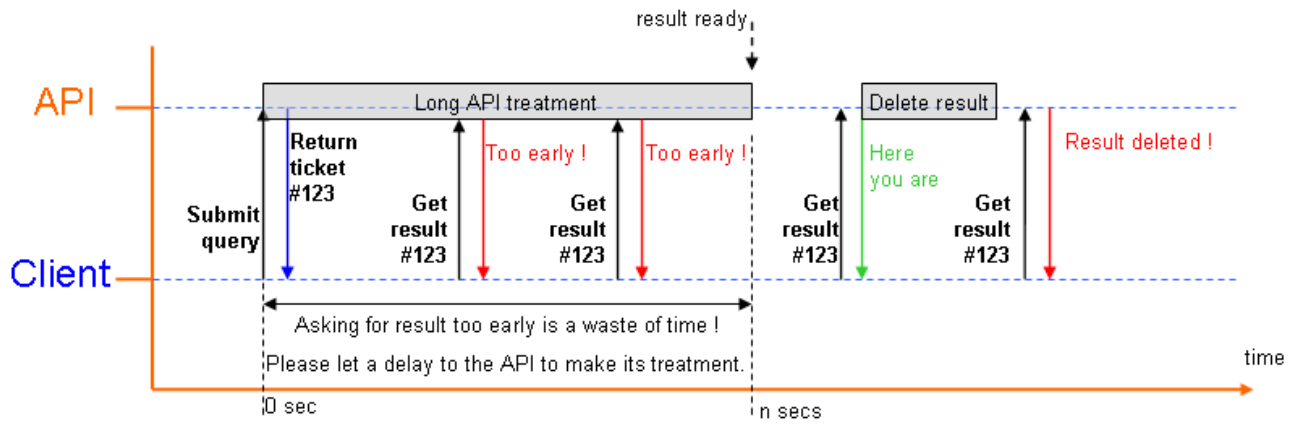
Two kind of actors are concerned by the Orange M2M services :

- The operator of the machines set, who expresses functional demands in relation to his management applications.
- The developers who insert these services in the given application.

Tip

Please note that some operations are done in asynchronous mode. They need a long time treatment (in case of M2M updates for example). A ticket number is returned when the method is called. Later, this ticket number can be used to get final result. Please use these operations very carefully. As the methods can be long lasting, it is unnecessary to require the operation result using the ticket number too frequently and too early after getting ticket number. A small figure below explain how to query your result.

Figure 1.1. Example of asynchronous sequence call with result get later



Chapter 2. Prerequisites

Before getting started, please read the information of this chapter, and follow any installation / download instructions. You'll be then ready to move onto the other chapters.

In this chapter you will learn how to get your credentials. Then, we will explain how to have a well configured PHP environment.

1. What you need to consume M2M API web service

To access the M2M API, you first have to get from Orange some credentials. Once you have these items, please keep them, they will be necessary to access the M2M API.

1.1. API access credentials needed

This section focuses on how to access the API and consume the webservice.

Developing with the API, you'll need:

- *Service URLs*: the URLs to consume the webservices
- *Access Key* and *Access Key Password* for basic authentication.

1.2. M2M API requirements

You will need the relevant end point URL when using the API. You need to use different URLs for different functionalities of your development.

The following table recaps the WSDL endpoint locations depending on the M2M functionality you are working with.

Table 2.1. Which endpoint should I use?

Functionality	URL
Subscription Status	https://iosw-ba.francetelecom.com/MLM/SubscriptionStatus-1
Connectivity Directory	https://iosw-ba.francetelecom.com/MLM/ConnectivityDirectory-1
Sim Lifecycle Management	https://iosw-ba.francetelecom.com/MLM/SimLifecycleManagement-1
Consumption Tracking	https://iosw-ba.francetelecom.com/MLM/ConsumptionTracking-1
Network Status	https://iosw-ba.francetelecom.com/MLM/NetworkStatus-1
Incident Diagnostics	https://iosw-ba.francetelecom.com/MLM/IncidentDiagnostics-1
Session History	https://iosw-ba.francetelecom.com/MLM/SessionHistory-1

Download and install the server certificate from M2M API on your machine in order to authenticate your application for the Secure Socket Layer (SSL).

2. PHP requirements

If you are going to use the SDK you can skip this section.

On the contrary, if you want to develop from scratch, you have to read this section.

You will need to download and to activate through `php.ini` the Soap, Curl and OpenSSL extension. The Zend Framework manages Soap objects and the logs.

Table 2.2. Required libraries

Library type	Library examples and comments
ZendFramework	The Zend Framework is required to use this sdk. It is provided with the sdk in the folder "COMMON\External". The version used is ZendFramework-1.9.3PL1.
SOAP Library	We recommend to use Wamp server 2.0 which provides the Soap extension. You'll have to activate the extension through <code>php.ini</code>).
Curl Library	The recommended version is "libcurl/7.19.6 OpenSSL/0.9.8k zlib/1.2.3" which is included in Wamp server 2.0. You'll have to activate the extension through <code>php.ini</code>).
OpenSsl Library	The recommended version is OpenSSL 0.9.8k which is included in Wamp server 2.0. You'll have to activate the extension through <code>php.ini</code>).

Table 2.3. Optional libraries

Library type	Library examples and comments
Unit Testing	We recommend the use of PHP unit or SimpleTest .
Logging	We recommend the use of Zend Log .

For the matter of simplicity and when the library license allows, we have included in our SDKs the above mentioned libraries.

3. PHP environment

We recommend the use of Eclipse PDT or Netbeans in association with xDebug as IDE to develop, debug and run your application. We encourage you as well to use the WAMP framework to test your application. The code has been tested and developed with the PHP5 version. You'll find the exact version number in the changelog of the SDK. So make sure to use the right PHP5 version to ensure compatibility.

To recap, here is what we recommend you to use:

Table 2.4. PHP environment recommended tools

Tool	Comment
Eclipse PDT IDE	PDT Eclipse IDE is freeware and provides developer with intellisense-like functionality. You may also use Netbeans with xDebug.
Wamp (WampServer Version 2.0)	Wamp (Windows Apache Mysql Php) allows you to switch easily between PHP and Apache versions.

Tip

Apache localhost instruction for Windows users: to modify the localhost directory (by default under Windows `c : /wamp/www` if you use Wampserver), find the `httpd.conf` file in the directory `wamp/`

bin/apache/apache.<version>/conf/ (where <version> is the version of Apache you are working with). Edit the following lines in the file:

```
DocumentRoot "<path to your directory>" and Directory "<path to your directory>".
```

Save the file and restart Wamp.

Example:

```
DocumentRoot "c:/devarea"  
<Directory "c:/devarea">
```

4. Proxy and firewall settings

In order to use our APIs, you have to check your proxy and firewall settings.

The firewall basically inspects network traffic passing through it. It may deny passage based on how rules have been set by your system administrator. We recommend to first check with the administrator if the port is open and if the URL request won't be blocked. Once the proxy and firewall are well set, we encourage you to test the network with a simple function that takes a few easy-to-build parameters and just returns a status. This will allow you to verify that everything is working fine.

Chapter 3. Install and Set Up

To kick-start your development download the PHP SDK. The reference manual (generated by PHPdocumentor) is also available in the ZIP of the SDK.

1. Install SDK

The SDK is composed of two libraries:

- The Common library and M2M (the core libraries),

These libraries are all mandatory.

```
<project>
- M2M
- COMMON
```

What you have to do now is rather simple if you followed previous document sections. Just copy the core libraries folders to your workspace and you are ready to go!

Tip

We encourage you to use Integrated Development Environment (IDE) which will ease your everyday work with programming. You can use Eclipse and Netbeans for free.

2. Configure SDK

The SDK is shipped with a configuration file that stores required settings: web proxy parameters, service URL and M2M credentials. Alternatively you can hardcode these settings.

The default configuration file is named `m2m.ini` and is located in M2M folder. An exception will be thrown if the configuration file is not found or not valid.

Tip

If your login or password contains a backslash, don't forget to double it.

Example 3.1. Example of m2m.ini file

```

;-----
; (1) Connectivity settings
;-----
; http proxy (optional)
; Don't forget to surround the values with double-quotes.
proxyHost      = "your_proxy_host"
proxyPort      = "your_proxy_port"
proxyUsername  =
proxyPassword  =

;-----
;      (2) API Settings
;-----
; Mandatory
; Don't forget to surround the values with double-quotes.
; Credentials
;AccessKey
login          = "your_login"

;AccessKey Password
password       = "your_password"

; End Points
connectivityDirectoryUrl = "your_ConnectivityDirectory_url"
consumptionTrackingUrl   = "your_ConsumptionTracking_url"
incidentDiagnosticsUrl   = "your_incidentDiagnostics_url"
networkStatusUrl        = "your_networkStatus_url"
sessionHistoryUrl        = "your_sessionHistory_url"
simLifecycleManagementUrl = "your_SimLifecycleManagement_url"
subscriptionStatusUrl    = "your_SubscriptionStatus_url"

;-----
; (3) Misc
;-----
; set it to "no" to log.
DisableLogger = yes

```

To apply these settings, you have to use the IniParser and the ServiceConfigurator classes:

```

$IniFile = new M2M_IniParser(<PATH_TO>."m2m.ini"); // Path to ini file
$Settings = new M2M_ServiceConfigurator($IniFile);

```

3. One shot installation

You will learn here how to run the code snippets depending on your operating system.

3.1. Windows

The following steps will allow you to run the code snippets within a few minutes under Windows with WAMP server.

The 2 first steps describe how to install a PHP server (namely Wampserver). Skip them if you already have your own PHP server running.

- Download and install WampServer at <http://www.wampserver.com/en/index.php>. Just leave the default parameters when installing.
- Check that PHP server runs well by starting Wamp: Start -> All Programs -> WampServer -> Start WampServer, wait a few seconds et type in your favorite internet browser the URL "<http://localhost>". You should get a page that displays information about PHP. From now, your PHP server is properly installed!

Once Wamp is installed, you can see its icon in the tray. When clicking on the icon, you'll get the menu.

Figure 3.1. WampServer 2.0 Menu



The following steps explain how to run your first code snippets.

- Activate first curl extension through your `php.ini` file (the one of the PHP folder if running CLI or the one of the Apache folder if running the server).
- Unzip the archive you've downloaded in the folder `C:\wamp\www\mytest`. The folder `C:\Wamp\www\mytest` should contain 2 folders, namely `COMMON` and `<Api long Name>` (where `<Api long Name>` is `ContactEveryone`, `deviceCapabilities`, `MultimediaConference` or `Malima` depending on the API you've downloaded).
- Edit the properties file `<API short name>.ini` (where `<API short name>` is `CEO`, `DCE`, `MMC` or `M2M`) stored in the `<API long name>` folder to input your own connection settings (credentials, URL, proxy, Misc ...).
- Create a new PHP file, say `myfirsttest.php`. The content of this file is stored in the "first API call" section that you can find under the "getting started" tab, or take whatever code sample you find in the SDK manual PDF located in "advanced development" tab. Store this file in `C:\wamp\www\mytest\`.
- Edit `myfirsttest.php` to set the following initialization files :

```
InitializationPath="PATH_TO/<API short name>/Initialization.php"
```

```
IniFile="PATH_TO/<API short name>/m2m.ini"
```

- You are ready to run `myfirsttest.php`! Check that WampServer does run (check step 2).
- On the <http://localhost> webpage, you can see a section "Vos Projets" or "Your Projects". Below, you'll find the project called "mytest". Click on that link.
- Click now on `myfirsttest.php`.

- If this worked, you should have the same result as this is described below the code snippets you've copied pasted (you did that in step 5). You're now ready to test more functions by copying pasting code snippets from the documentation!

3.2. Unix

You'll find hereafter some commands to install PHP under Unix. These commands were run under Ubuntu and Debian. You may need to modify them for your environment.

- `sudo apt-get install apache2 apache2-doc mysql-server php5 libapache2-mod-php5 php5-mysql phpmyadmin`
- `sudo apt-get install php5-cli` (if you wish / need to use PHP through commandline)
- `sudo apt-get install php5-curl` (you will need this option to call the APIs)

Figure 3.2. Example of output snippets when installing PHP

```
Lecture des listes de paquets... fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
 apache2-mpm-prefork apache2-utils apache2.2-common libapr1 libaprutil1
 libdbd-mysql-perl libdbi-perl libgd2-xpm libhtml-template-perl libltdl3
 libmcrypt4 libmysqlclient15off libnet-daemon-perl libplrpc-perl libt1-5
 libterm-readkey-perl mysql-client-5.0 mysql-common mysql-server-5.0 openssl
 openssl-blacklist php5-common php5-gd php5-mcrypt psmisc ssl-cert
Paquets suggérés :
 apache2-suexec apache2-suexec-custom php-pear dbshello libgd-tools
 libipc-sharedcache-perl libmcrypt-dev mcrypt tinyca ca-certificates
Les NOUVEAUX paquets suivants seront installés :
 apache2 apache2-doc apache2-mpm-prefork apache2-utils apache2.2-common
 libapache2-mod-php5 libapr1 libaprutil1 libdbd-mysql-perl libdbi-perl
 libgd2-xpm libhtml-template-perl libltdl3 libmcrypt4 libnet-daemon-perl
 libplrpc-perl libt1-5 libterm-readkey-perl mysql-client-5.0 mysql-server
 mysql-server-5.0 openssl openssl-blacklist php5 php5-common php5-gd
 php5-mcrypt php5-mysql phpmyadmin psmisc ssl-cert
Les paquets suivants seront mis à jour :
 libmysqlclient15off mysql-common
2 mis à jour, 31 nouvellement installés, 0 à enlever et 45 non mis à jour.
Il est nécessaire de prendre 54,9Mo dans les archives.
Après cette opération, 156Mo d'espace disque supplémentaires seront utilisés.
```

```
Creating config file /etc/php5/apache2/php.ini with new version
Reloading web server config: apache2.
Paramétrage de libgd2-xpm (2.0.36~rc1~dfsg-3) ...
Paramétrage de libhtml-template-perl (2.9-1) ...
Paramétrage de libltdl3 (1.5.26-4) ...
Paramétrage de libmcrypt4 (2.5.7-5) ...
Paramétrage de libt1-5 (5.1.2-3) ...
Paramétrage de libterm-readkey-perl (2.30-4) ...
Paramétrage de mysql-server (5.0.51a-24+lenny1) ...
Paramétrage de php5 (5.2.6.dfsg.1-1+lenny3) ...
Paramétrage de php5-gd (5.2.6.dfsg.1-1+lenny3) ...
Paramétrage de php5-mcrypt (5.2.6.dfsg.1-1+lenny3) ...
Paramétrage de php5-mysql (5.2.6.dfsg.1-1+lenny3) ...
Paramétrage de ssl-cert (1.0.23) ...
Paramétrage de phpmyadmin (4:2.11.8.1-5) ...
```

CLI install output snippet:

```

2.6.dfsg.1-1+lenny3 12475kB]
2475ko réceptionnés en 0s (4467ko/s)
Sélection du paquet php5-cli précédemment désélectionné.
(Lecture de la base de données... 45681 fichiers et répertoires déjà installés.)
Dépaquetage de php5-cli (à partir de .../php5-cli_5.2.6.dfsg.1-1+lenny3_i386.de
) ...
Traitement des actions différées (« triggers ») pour « man-db »...
Paramétrage de php5-cli (5.2.6.dfsg.1-1+lenny3) ...

Creating config file /etc/php5/cli/php.ini with new version
uami8516@a-linuxasc:~$

```

Curl install snippet:

```

Préconfiguration des paquets...
Sélection du paquet ca-certificates précédemment désélectionné.
(Lecture de la base de données... 45493 fichiers et répertoires déjà installés.)
Dépaquetage de ca-certificates (à partir de .../ca-certificates_20080809_all.de
) ...
Sélection du paquet libssh2-1 précédemment désélectionné.
Dépaquetage de libssh2-1 (à partir de .../libssh2-1_0.18-1_i386.deb) ...
Sélection du paquet libcurl3 précédemment désélectionné.
Dépaquetage de libcurl3 (à partir de .../libcurl3_7.18.2-8lenny2_i386.deb) ...
Sélection du paquet php5-curl précédemment désélectionné.
Dépaquetage de php5-curl (à partir de .../php5-curl_5.2.6.dfsg.1-1+lenny3_i386.
eb) ...
Traitement des actions différées (« triggers ») pour « man-db »...
Paramétrage de ca-certificates (20080809) ...
Updating certificates in /etc/ssl/certs...done.
Running hooks in /etc/ca-certificates/update.d...done.
Paramétrage de libssh2-1 (0.18-1) ...
Paramétrage de libcurl3 (7.18.2-8lenny2) ...
Paramétrage de php5-curl (5.2.6.dfsg.1-1+lenny3) ...

```

Edit apache2.conf file: Alias /sdk/ "/project/php/" <Directory "/project/php/">Order allow,deny Allow from all </Directory>.

Then, you are ready to run the SDK. Since you're under Unix, mind the upper- and lowercase!

- Unzip the archive you've downloaded in the folder mytest. The folder mytest should contain 2 folders, namely COMMON and <API long name> (where <API long name> is ContactEveryone, deviceCapabilities, MultimediaConference or Malima depending on the API you've downloaded).
- Edit the properties file <API short name>.ini (where <API short name> is CEO, DCE, MMC or M2M) stored in the <API long name> folder to input your own connection settings (credentials, URL, proxy, Misc ...).
- Create a new PHP file, say myfirsttest.php. The content of this file is stored in the "first API call" section that you can find under the "getting started" tab, or take whatever code sample you find in the SDK manual PDF located in "advanced development" tab. Store this file in C:\wamp\www\mytest\.
- Edit myfirsttest.php to set the following initialization files :

```
InitializationPath="PATH_TO/<API short name>/Initialization.php"
```

```
IniFile="PATH_TO/<API short name>/m2m.ini"
```

- You are ready to run myfirsttest.php! Check that WampServer does run (check step 2).
- Go to the `http://localhost:port/webpage` to run myfirsttest.php.
- If this worked, you should have the same result as this is described below the code snippets you've copied pasted. You're now ready to test more functions by copying pasting code snippets from the documentation!

4. First API call in PHP

If you reach this step it means that you are ready to deploy our SDK libraries and the SDK configuration file to get informations on a Sim card in PHP. This is what we will achieve here with this small sample demonstrating that you can successfully communicate with our API by calling the `getConnectivityDirectory` method.

This `getConnectivityDirectory` method returns a datas list of the Sim card selected.

The first code listing below uses the configuration file `m2m.ini`. Do not forget to edit this file to type your own credentials, proxy and end-point data before running this example.

Here is an example of a sample code using the configuration file `m2m.ini`:

```
<?php
// Prior run this snippet, don't forget to edit the file "m2m.ini"
// TODO: Edit m2m.ini file in "M2M" folder and copy it to "sample" folder
$output = "";
$initializationPath = dirname ( __FILE__ ) . "../M2M/Initialization.php";
try {
    if ( ! require_once ($initializationPath) ) {
        throw new Exception ( "Cannot find Initialization file at $initializationPath" );
    }
    $iniFile = new M2M_IniParser(dirname(__FILE__).DIRECTORY_SEPARATOR."m2m.ini"); // Path to ini
    file
    $settings = new M2M_ServiceConfigurator($iniFile);
    $output .= "<h5>Using URL: " . $settings->get_sConnectivityDirectoryUrl() . "</h5>";
    $connectivityDirectoryClient = new M2M_ConnectivityDirectoryClient($settings);

    //*****//
    //*** GETCONNECTIVITYDIRECTORY ***//
    //*****//

    $getConnectivityDirectory= new getConnectivityDirectory();
    $getConnectivityDirectory->lineIdentifiers= new LineIdentifierCollection();
    $getConnectivityDirectory->lineIdentifiers->subscriptionNumber = "your_subscription_number";
    $output .= M2M_Utility::getBeautifiedOutput($connectivityDirectoryClient-
>call_getConnectivityDirectory($getConnectivityDirectory), "getConnectivityDirectory" );
    echo $output;
?>
```

Chapter 4. The M2M SDK

Hereafter is described a global view of the M2M SDK architecture. For a more functional view, please refer to the functional User document.

Basically, the M2M SDK is based on PHP 5 and is composed of:

- Common: this library contains reusable classes in the Open Developer Network context.
- M2M: this library contains
 - the `M2mClient` service with access to the M2M API webmethods.
 - the `Proxy` and `Credentials` class, encapsulating authentication data.
 - the Web services, that lets the developer get quickly involved in the M2M API
 - A set of php tools to initialize, to log, to generate outputs of the answers sent by M2M

Figure 4.1. M2M Sdk global architecture diagram

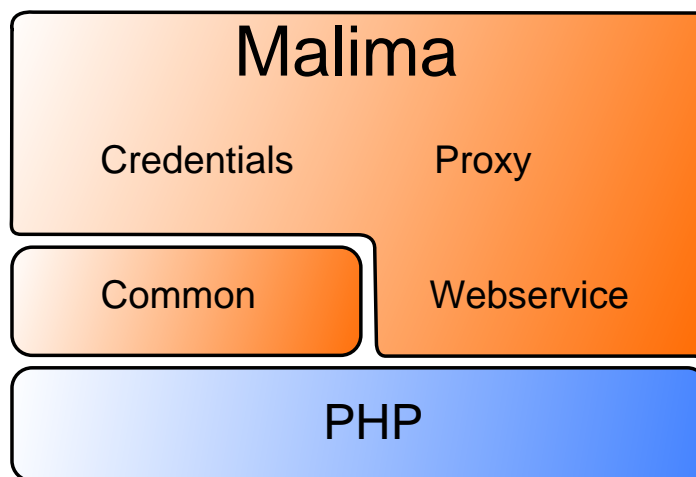
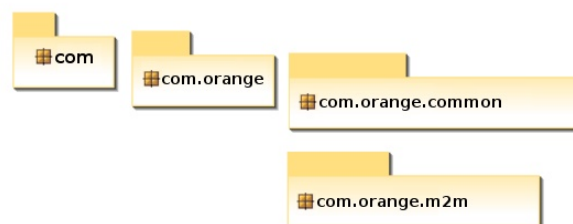


Figure 4.2. M2M Sdk UML model diagram



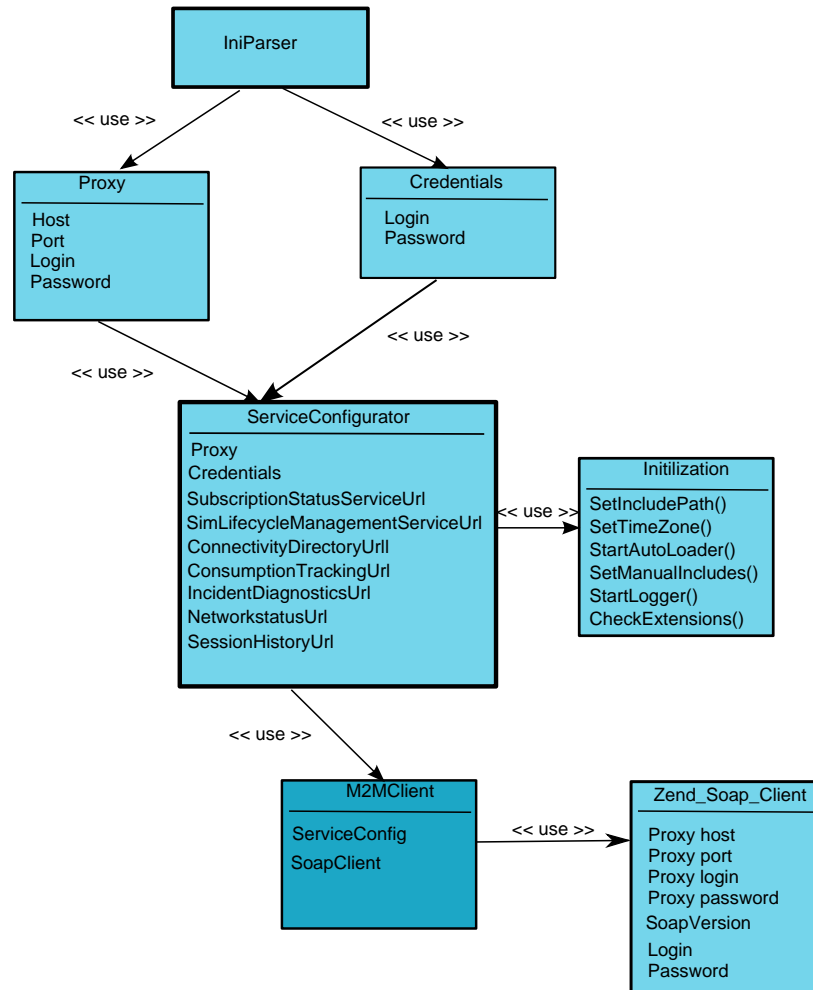
1. Common Objects

The M2M folder contains all common required classes. The common classes are `IniParser` (`IniParserException.php`), `Proxy`, `Credentials`, `ServiceConfigurator`, `Initialization` (`InitializationException.php`), `M2mClient` (`M2mClientException`), `Zend_Soap_Client` (`Zend Framework`), `FileStream`, `Utility`.

- `IniParser`: this class parses the `m2m.ini` file to extract the proxy settings, the credentials, the end points and the log settings.
- `Proxy`: this class encapsulates proxy data extracted by the `IniParser`.

- **Credentials**: this class encapsulates credentials data extracted by the IniParser.
- **ServiceConfigurator**: this class encapsulates all data contained in Proxy, Credentials objects and the end points.
- **Initialization**: this class initializes the following methods : SetIncludePath SetTimeZone StartAutoLoader SetManualIncludes StartLogger CheckExtensions
- **M2mClient**: this abstract class is the parent class of each class of service and links ServiceConfigurator, EndPointUrl, WsdPath and ClassMap. It also manages logs.
- **Zend_Soap_Client**: this Zend Framework class provides us the soap object used.
- **FileStream**: this class extends the Zend Framework class Zend_Log_Writer_Stream which allow us to set our writer.
- **Utility**: this class provides output functionalities.

Figure 4.3. Common classes



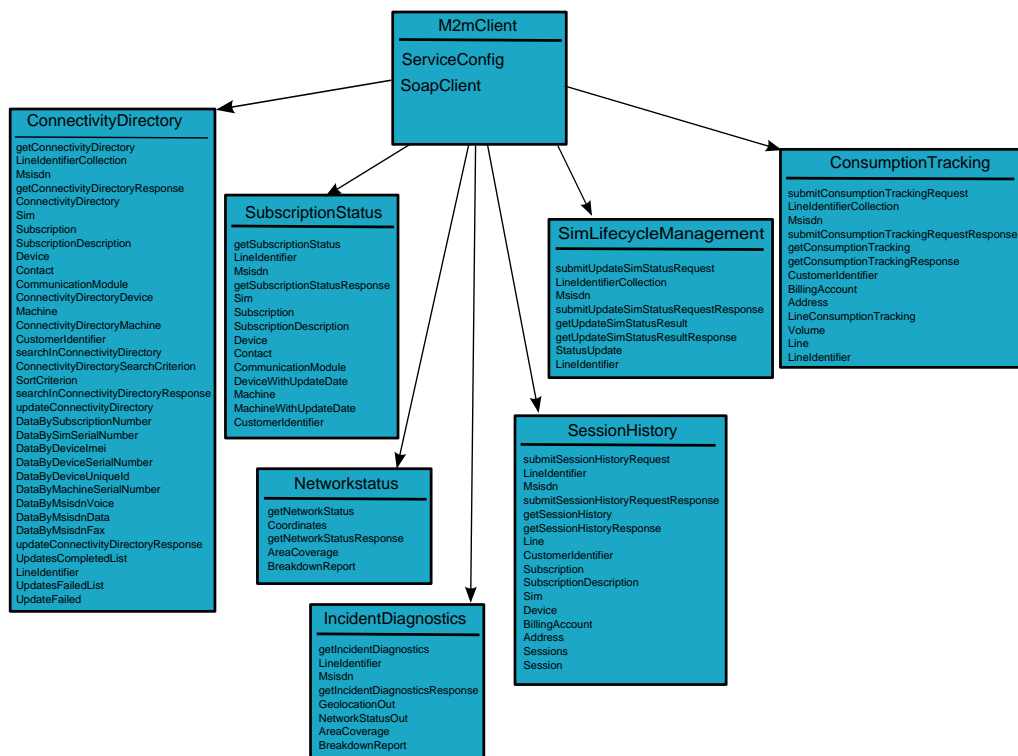
2. M2M

The `com.orange.M2M` package contains several classes.

- **M2M**: this class represents a simple entry point to call webmethods of the M2M API. It inherits from the `com.orange.api.common.OrangeApiService`, thus encapsulates static initialization methods. Moreover, it provides business and configuration methods :
 - `getSubscriptionStatus`: Get the data set of a SIM card and refresh it, according to one of its identifiers.

- `getConnectivityDirectory`: Get SIM card information according to a collection of up to 100 line identifiers.
- `searchInConnectivityDirectory`: Search SIM card information according to a set of sorted parameters, returned by range.
- `updateConnectivityDirectory`: Update a set of SIM card referentials, including devices and machines, according to SIM cards's identifiers.
- `submitUpdateSimStatus`: Update a set of SIM cards's statuses, according to their identifiers. This method is asynchrone and needs a call to `getUpdateSimStatus` method in order to get result.
- `getUpdateSimStatus`: Retrieve result of a `submitUpdateSimStatus`, according to a ticket number.
- `submitConsumptionTrackingRequest`: Request for a set of SIM cards's traffic in a date range. This method is asynchrone and needs a call to `getConsumptionTracking` method in order to get result.
- `getConsumptionTracking`: Retrieve result of a `submitConsumptionTrackingRequest` according to a ticket number.
- `getNetworkStatus`: Get the network covering status into an area defined by its latitude and longitude.
- `getIncidentDiagnostics`: Get the diagnostic analysis of a SIM card, including the network covering status of the SIM card.
- `submitSessionHistoryRequest`: Request for a SIM card's set of statistics by sessions in a date range. This method is asynchrone and needs a call to `getSessionHistory` method in order to get result.
- `getSessionHistory`: Retrieve result of a `submitSessionHistoryRequest` according to a ticket number.

Figure 4.4. M2M class diagram.



3. SDK configuration

The `M2mClient` and her subclasses encapsulates initialization objects named `Credentials`, `Proxy` and `ServiceConfigurator`.

This M2mClient class use the ServiceConfigurator class which is based on two other class :

- Credentials
- Proxy

The ServiceConfigurator class read a configuration file (m2m.ini by default) and set the following parameters:

- the web proxy: you can disable the web proxy or specify your own proxy.
- the credentials: you'll have to set your personal credentials.
- the service URL.

ServiceConfigurator loads settings from the properties file named m2m.ini and is located in the M2M folder.

An exception is thrown if the initialization file is not found or contains invalid data.

Here is an sample of how to get the configuration datas:

```

/* Inialization.php manages includes and starts some required functionalities of the Zend
Framework
 * SetIncludePath (Zend)
 * SetTimeZone
 * StartAutoLoader (Zend)
 * SetManualIncludes
 * StartLogger (Zend)
 * CheckExtensions
 */
$sInitializationPath = dirname ( __FILE__ ) . "../M2M/Initialization.php";
if ( ! require_once ($sInitializationPath) ) {
throw new Exception ( "Cannot find Initialization file at $sInitializationPath" );
}

/* M2M_IniParser read the m2m.ini to extract required datas */
$oIniFile = new M2M_IniParser(dirname(__FILE__).DIRECTORY_SEPARATOR."m2m.ini"); // Path to
ini file

/*M2M_ServiceConfigurator contains all configurations datas*/
$oSettings = new M2M_ServiceConfigurator($oIniFile);

```

Here's a example on how to get the Connectivity Directory Url from the m2m.ini file. You can acces to a dedicated data this way :

```
$oSettings->get_sConnectivityDirectoryUrl();
```

You can find this example in the ConnectivityDirectoryExample.php code snippet provided with the sdk located in the sample folder.

4. Logging

The SDK uses the Zend Framework for purposes of application debugging and auditing. If you don't want to use this feature, just skip this section: it won't impact the execution of your program (by default, logging is disabled).

Zend_Log is a tool to help the programmer output log statements to a variety of output targets. .

Typically the Zend_Log activation is specified using a properties file.

The following content shows you the properties file `m2m.ini` that is shipped with the SDK where you can enable the Zend logger.

```

;-----
; (3) Misc
;-----
; set it to "no" to log.
DisableLogger = yes

```

Here's a sample of `Initialization.php` where the zend logger is loaded :

```

$log = new Zend_Log ( );

// Formatting
$defaultFormat = Zend_Log_Formatter_Simple::DEFAULT_FORMAT;
$format = $defaultFormat . '%client_ip% %user_agent%' . PHP_EOL;

$path = dirname ( __FILE__ ) . '/../logs/events.log';
if ( file_exists ( dirname ( $path ) ) ) {
    $writer = new M2M_FileStream ( $path );
    $writer->setFormatter ( new Zend_Log_Formatter_Simple ( $format ) );
    $log->addWriter ( $writer );
}

// Adding parameters to log: IP and browser
if ( array_key_exists ( 'HTTP_USER_AGENT', $_SERVER ) ) {
    $log->setEventitem ( 'user_agent', $_SERVER [ 'HTTP_USER_AGENT' ] );
}
if ( array_key_exists ( 'REMOTE_ADDR', $_SERVER ) ) {
    $log->setEventitem ( 'client_ip', $_SERVER [ 'REMOTE_ADDR' ] );
}

Zend_Registry::set ( 'log', $log );

```

You can create your personal log file, and set path to it as shown above. By default, you can find the log file `event.log` in the "logs" folder of the sdk.

This log configuration file defines a writer which write into the file `event.log` but Zend_log provides other way to log :

- directly in the output of your browser
- in a database
- [firebug](#) console

You can find more information about Zend_log [here](#)

You'll find hereafter an example of a Zend_log output:

```

*****SOAP request HTTP header: POST /MLM/SessionHistory-1 HTTP/1.1
Host: 8X.1X.21X.20X
Connection: Keep-Alive

```

```

User-Agent: PHP-SOAP/5.2.11
Content-Type: text/xml; charset=utf-8
SOAPAction: ""
Content-Length: 662
Authorization: Basic NjZNM1YzZmQ4N1ZVOVBLOVXXXXXXXXXXY
127.0.0.1 Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.2) Gecko/20100115 Firefox/3.6
2010-03-22T10:31:58+01:00 DEBUG (7):

*****SOAP request HTTP body:
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns1="http://common.types.malima.francetelecom.com"
  xmlns:ns2="http://sessionHistory.types.malima.francetelecom.com"
  xmlns:ns3="http://webservice.malima.francetelecom.com/v1">
  <SOAP-ENV:Body>
    <ns3:submitSessionHistoryRequest>
      <ns2:lineIdentifier>
        <ns1:subscriptionNumber>23697253</ns1:subscriptionNumber>
      </ns2:lineIdentifier>
      <ns2:periodStartDateTime>2010-01-11T08:02:09+01:00</ns2:periodStartDateTime>
      <ns2:periodEndDateTime>2010-02-11T08:02:09+01:00</ns2:periodEndDateTime>
    </ns3:submitSessionHistoryRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

--- Using style and using use
127.0.0.1 Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.2) Gecko/20100115 Firefox/3.6
2010-03-22T10:31:58+01:00 DEBUG (7): Success
127.0.0.1 Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.2) Gecko/20100115 Firefox/3.6
2010-03-22T10:31:58+01:00 DEBUG (7):

*****SOAP response HTTP header: HTTP/1.1 200 OK

X-Backside-Transport: OK OK
Connection: Keep-Alive
Transfer-Encoding: chunked
Server: Apache-Coyote/1.1
Content-Type: text/xml; charset=UTF-8
Date: Mon, 22 Mar 2010 09:29:55 GMT
X-Client-IP: 1X.6X.7X.230

127.0.0.1 Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.2) Gecko/20100115 Firefox/3.6
2010-03-22T10:31:58+01:00 DEBUG (7):

*****SOAP response HTTP body:
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:trackingHeader
      xmlns:t="http://www.francetelecom.com/iosw/v1"
      xmlns:date="http://exslt.org/dates-and-times">
      <t:requestId>c6e4e2c6-a470-4b64-b4db-6d238408f7a8</t:requestId>
      <t:timestamp>2010-03-22T10:29:55+01:00</t:timestamp>
    </t:trackingHeader>
  </soap:Header>
  <soap:Body>
    <ns2:submitSessionHistoryRequestResponse
      xmlns:ns2="http://webservice.malima.francetelecom.com/v1"
      xmlns:ns3="http://common.types.malima.francetelecom.com"

```

```
xmlns:ns4="http://sessionHistory.types.malima.francetelecom.com"  
xmlns:ns5="http://exception.malima.francetelecom.com"  
xmlns:ns6="http://networkstatus.types.malima.francetelecom.com">  
<ns4:ticketNumber>400</ns4:ticketNumber>  
</ns2:submitSessionHistoryRequestResponse>  
</soap:Body></soap:Envelope>  
127.0.0.1 Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.2) Gecko/20100115 Firefox/3.6  
2010-03-22T10:31:58+01:00 DEBUG (7): Success  
127.0.0.1 Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.2) Gecko/20100115 Firefox/3.6  
2010-03-22T10:31:58+01:00 DEBUG (7):
```

(...)

Chapter 5. Methods

You should first make sure you've read the prerequisites before coding and running the following program listing. The prerequisites explain, among others, how to set up your proxy information and your account information through a configuration file.

1. Basic scenarios

In this chapter you will learn how to use the base objects we defined for you inside our SDK through some ready-to-use samples.

What are the basic steps to get your subscription status for a set of SIM cards ?

1. Create a `M2MUser` and initialize its parameters.
2. Create a `LineIdentifierCollection` for a set of SIM cards, based on their `SubscriptionIdentifier` for instance.
3. In the end, call the `GetSubscriptionStatus` API method. You will know everything about a set of SIM cards and their connectivity directory : their status, in which devices the SIM cards are set, in which machines the devices are set.

What are the basic steps to search in your connectivity directory ?

1. Create a `M2MUser` and initialize its parameters.
2. If you don't have information about your SIM cards or serial numbers, you could Search in your Connectivity Directory according to a big range of searching criterions such as the state of searched SIM cards, the properties of a device of machine, dates of SIM cards changing states, phone numbers,
3. In the end, call the `SearchInConnectivityDirectory` API method. You will get the set of SIM cards corresponding to your searchin criterions.

What are the basic steps to get then update your connectivity directory ?

1. Create a `M2MUser` and initialize its parameters.
2. Create a `LineIdentifierCollection` for a set of SIM cards.
3. Call the `GetConnectivityDirectory` method.
4. Modify the `ConnectivityDirectory` array.
5. In the end, call the `UpdateConnectivityDirectory` API method using the modified `ConnectivityDirectory`.

What are the basic steps to update a SIM card status ?

1. Create an `M2MUser` and initialize its parameters.
2. Create a `LineIdentifierCollection` for a set of SIM cards.
3. Create a (`SimLifecycleManagement`) `SimStatus` corresponding to the requested updated status.
4. Create a `TestMode` for the area covered (`INTERNATIONAL`, `EUROPE`, `METROPOLE`, `AU_COMPTEUR`).
5. Call the asynchronous `SubmitUpdateSimStatus` API method.
6. Get the `TicketNumber` returned.
7. Call the `GetUpdateSimStatus` API method until the global response status of the method is either `TERMINATED` or `ERROR`, meaning the asynchronous modification has ended.

What are the basic steps to get consumption tracking of a set of SIM cards ?

1. Create an `M2MUser` and initialize its parameters.
2. Create a `LineIdentifierCollection` for a set of SIM cards.
3. Create a `start date` and an `end date` for a given period.
4. Call the asynchronous `SubmitConsumptionTracking` API method.
5. Get the `TicketNumber` returned.
6. Call the `GetConsumptionTracking` API method until the global response status of the method is either `TERMINATED` or `ERROR`, meaning the asynchronous generation has ended.

What are the basic steps to get your network status ?

1. Create a `M2MUser` and initialize its parameters.
2. Create a `Coordinates` object corresponding to GPS coordinates of the area you where want the network status.
3. In the end, call the `GetNetworkStatus` API method. You will know everything about the network in the area, for each kind of network technology.

What are the basic steps to get incident disgnostics of a SIM card ?

1. Create a `M2MUser` and initialize its parameters.
2. Create a `LineIdentifier` for the SIM card.
3. In the end, call the `GetIncidentDiagnostics` API method. You will know everything about the state of the SIM card, the network status in the area, troubles occuring in the area.

What are the basic steps to get session history of a given SIM cards ?

1. Create an `M2MUser` and initialize its parameters.
2. Create a `LineIdentifier` for the SIM card.
3. Create a `start date` and an `end date` for a given period.
4. Call the asynchronous `SubmitSessionHistory` API method.
5. Get the `TicketNumber` returned.
6. Call the `GetSessionHistory` API method until the global response status of the method is either `TERMINATED` or `ERROR`, meaning the asynchronous generation has ended.

2. GetSubscriptionStatus method

2.1. Description

Refresh SIM card's data and return the full data set information concerning one SIM card

2.2. Input parameter

Table 5.1. getSubscriptionStatus: Input parameters.

Name	Type	Description	Cardinality
Service Configurator	Service Configurator	A Configurator for the Malima service	1..1
LineIdentifier	LineIdentifier	The SIM card identifier referenced by a subscriptionNumber, simSerialNumber, deviceImei, deviceSerialNumber, deviceUniqueId, machineSerialNumber, or a phone number associated to voice, data or fax.	1..1

2.3. Output parameter

Table 5.2. getSubscriptionStatus: output parameters.

Name	Type	Description	Cardinality
Response	GetSubscriptionStatusResponse	A description of the SIM card data, made of a customer environment Id, a SIM card description, subscription state, device handling the SIM card description, machine handling the device description, an array of customer identifiers	1..1

2.4. Exceptions

This web method may throw `TechnicalException` or `FunctionalException`.

2.5. Sample code

This sample shows how to Get Subscription Status of a SIM card.

```
<?
// Prior run this snippet, don't forget to edit the file "m2m.ini"
// TODO: Edit m2m.ini file in "M2M" folder and copy it to "sample" folder
$output = "";
$initializationPath = dirname ( __FILE__ ) . "../M2M/Initialization.php";
try {
    if ( ! require_once ($initializationPath) ) {
        throw new Exception ( "Cannot find Initialization file at $initializationPath" );
    }
    $iniFile = new M2M_IniParser(dirname(__FILE__).DIRECTORY_SEPARATOR."m2m.ini"); // Path to ini
    file
    $settings = new M2M_ServiceConfigurator($iniFile);
    $subscriptionStatusClient = new M2M_SubscriptionStatusClient($settings);
    $output .= "<h5>Using URL: " . $settings->get_sSubscriptionStatusUrl() . "</h5>";

    /*******//
    /*** GETSUBSCRIPTIONSTATUS ***/
    /*******//

    $getSubscriptionStatus = new getSubscriptionStatus();
    $lineIdentifier= new LineIdentifierCollection();
    $lineIdentifier->subscriptionNumber = "your_subscription_number";
    $getSubscriptionStatus->lineIdentifier = $lineIdentifier;
    $getSubscriptionStatusRequest = $subscriptionStatusClient-
>call_getSubscriptionStatus($getSubscriptionStatus);
    $output .=
    M2M_Utility::getBeautifiedOutput($getSubscriptionStatusRequest, "getSubscriptionStatus");
}
catch ( Exception $e ) {
    $output .= M2M_Utility::getBeautifiedException($e);
}

echo $output;
?>
```

The output looks like:

```
<pre>getSubscriptionStatusResponse Object
(
  [customerEnvironmentIdentifier] => 64095147
  [sim] => Sim Object
    (
      [serialNumber] => 1956473384156
      [puk1] =>
      [puk2] =>
      [imei] =>
      [status] => PRE_ACTIVATED
      [suspensionReason] =>
      [lastStatusRefreshDate] => 2010-03-19T12:59:03.599+01:00
      [lastStatusChangeDate] => 2010-03-17T16:06:25+01:00
      [requestedStatus] =>
      [statusChangeRequestDate] =>
    )
  [subscription] => Subscription Object
    (
      [identifier] => 23697255
      [description] => SubscriptionDescription Object
        (
          [value] => Business M To M Pack 2 Mo
          [user] => Mr FOOBAR
          [userRef] =>
          [service] =>
        )
      [creationDate] => 2010-03-17T00:00:00+01:00
      [connectionDate] =>
      [msisdnData] =>
      [msisdnVoice] =>
      [msisdnFax] =>
    )
  [device] =>
  [machine] =>
  [customerIdentifier] => CustomerIdentifier Object
    (
      [type] => SIREN
      [identifier] => 304827900
    )
)
</pre>
```

3. GetConnectivityDirectory method

3.1. Description

Get your Connectivity Directory information concerning a set of SIM cards.

3.2. Input parameter

Table 5.3. getConnectivityDirectory: Input parameters.

Name	Type	Description	Cardinality
Service Configurator	Service Configurator	A Configurator for the Malima service	1..1
LineIdentifier Collection	LineIdentifier Collection	A collection of Line Identifiers	1..1

3.3. Output parameter

Table 5.4. getConnectivityDirectory: output parameters.

Name	Type	Description	Cardinality
Response	Connectivity DirectoryResponse	An array of ConnectivityDirectory and a Collection of unknown LineIdentifiers	1..1

3.4. Exceptions

This web method may throw `TechnicalException` or `FunctionalException`.

3.5. Sample code

This sample shows you how get a connectivity directory referential from a collection of line identifiers

```
<?php
// Prior run this snippet, don't forget to edit the file "m2m.ini"
// TODO: Edit m2m.ini file in "M2M" folder and copy it to "sample" folder
$output = "";
$initializationPath = dirname ( __FILE__ ) . "../M2M/Initialization.php";
try {
    if ( ! require_once ($initializationPath) ) {
        throw new Exception ( "Cannot find Initialization file at $initializationPath" );
    }
    $iniFile = new M2M_IniParser(dirname(__FILE__).DIRECTORY_SEPARATOR."m2m.ini"); // Path to ini file
    $settings = new M2M_ServiceConfigurator($iniFile);
    $output .= "<h5>Using URL: " . $settings->get_sConnectivityDirectoryUrl() . "</h5>";
    $connectivityDirectoryClient = new M2M_ConnectivityDirectoryClient($settings);

    //*****//
    //*** GETCONNECTIVITYDIRECTORY ***//
    //*****//

    $getConnectivityDirectory= new getConnectivityDirectory();
    $getConnectivityDirectory->lineIdentifiers= new LineIdentifierCollection();
    $getConnectivityDirectory->lineIdentifiers->subscriptionNumber = "your_subscription_number";
    $output .= M2M_Utility::getBeautifiedOutput($connectivityDirectoryClient->call_getConnectivityDirectory($getConnectivityDirectory), "getConnectivityDirectory" );
```

```
echo $sOutput;
?>
```

The output looks like:

```
<pre>getConnectivityDirectoryResponse Object
(
    [connectivityDirectory] => ConnectivityDirectory Object
        (
            [customerEnvironmentIdentifier] => 64095147
            [sim] => Sim Object
                (
                    [serialNumber] => 1956473384107
                    [puk1] =>
                    [puk2] =>
                    [imei] =>
                    [status] => PRE_ACTIVATED
                    [suspensionReason] =>
                    [lastStatusRefreshDate] => 2010-03-17T16:06:25+01:00
                    [lastStatusChangeDate] => 2010-03-17T16:06:25+01:00
                    [requestedStatus] =>
                    [statusChangeRequestDate] =>
                )
            [subscription] => Subscription Object
                (
                    [identifier] => 23697248
                    [description] => SubscriptionDescription Object
                        (
                            [value] =>
                            [user] =>
                            [userRef] =>
                            [service] =>
                        )
                    [creationDate] => 2010-03-17T00:00:00+01:00
                    [connectionDate] =>
                    [msisdnData] =>
                    [msisdnVoice] =>
                    [msisdnFax] =>
                )
            [device] => ConnectivityDirectoryDevice Object
                (
                    [updateDate] => 2010-03-18T17:13:27+01:00
                    [uniqueIdentifier] => MyPHPdeviceIdentifier
                    [serialNumber] => MyPHPdeviceSerialNumber
                    [contact] => Contact Object
                        (
                            [name] => M2M
                            [email] => m2m@orange.com
                            [phone] => 0612345678
                        )
                    [communicationModule] => CommunicationModule Object
                        (
```

```

        [brand] => PHP TM
        [model] => PHP Model
    )

    [deviceLocation] => MyPHPdeviceLocation
    [category] => MyPHPdeviceCategory
    [description] => MyPHPdeviceDescription
    [address] => MyPHPdeviceAddress
    [deviceHolder] =>
)

[machine] => ConnectivityDirectoryMachine Object
(
    [updateDate] => 2010-03-18T17:13:27+01:00
    [serialNumber] => MyPHPmachineSerialNumber
    [name] => MyPHPmachineName
    [description] => MyPHPmachineDescription
)

[customerIdentifier] => CustomerIdentifier Object
(
    [type] => SIREN
    [identifier] => 304827900
)

)

[unknownLinesIdentifiers] =>
)
</pre></h4>

```

4. SearchInConnectivityDirectory method

4.1. Description

The SearchInConnectivityDirectory operation enable to search SIM cards according to search criterions.

4.2. Input parameter

Table 5.5. searchInConnectivityDirectory: Input parameters.

Name	Type	Description	Cardinality
Service Configurator	Service Configurator	A Configurator for the Malima service	1..1
Connectivity Directory SearchCriterion	Connectivity Directory SearchCriterion	The object containing the criterions.	1..1
RangeSize	Integer	The max size returned by the method.	1..1
RangeStart	Integer	The index where the method must start from.	1..1

4.3. Output parameter

Table 5.6. searchInConnectivityDirectory: output parameters.

Name	Type	Description	Cardinality
Response	SearchInConnectivityDirectoryResponse	The response made of an array of ConnectivityDirectory objects, the global number of results, and a boolean telling whether the number of results has exceeded or not	1..1

4.4. Exceptions

This web method may throw `TechnicalException` or `FunctionalException`.

4.5. Sample code

This sample shows you how to Search in your Connectivity Directory.

```
<?php
// Prior run this snippet, don't forget to edit the file "m2m.ini"
// TODO: Edit m2m.ini file in "M2M" folder and copy it to "sample" folder
$output = "";
$initializationPath = dirname ( __FILE__ ) . "../M2M/Initialization.php";
try {
    if ( ! require_once ($initializationPath) ) {
        throw new Exception ( "Cannot find Initialization file at $initializationPath" );
    }
    $iniFile = new M2M_IniParser(dirname(__FILE__).DIRECTORY_SEPARATOR."m2m.ini"); // Path to ini file
    $settings = new M2M_ServiceConfigurator($iniFile);
    $output .= "<h5>Using URL: " . $settings->get_sConnectivityDirectoryUrl() . "</h5>";
    $connectivityDirectoryClient = new M2M_ConnectivityDirectoryClient($settings);

    //*****
    //*** SEARCHINCONNECTIVITYDIRECTORY ***
    //*****
    $searchCriteria = new ConnectivityDirectorySearchCriterion ( );
    $searchCriteria->requiredStatus = "your_required_status"; // Optional
    // If you want to filter on currentStatus, you can pass an array to "currentStatus" parameter:
    // $searchCriteria->currentStatus = array("ACTIVATED"); // Optional

    // To know how many results can be found, you can set $iRangeSize to 0 and $iRangeStart to 0
    $iRangeSize = "your_iRangeSize"; // The number of items to be retrieved during this search
    $iRangeStart = "your_iRangeStart"; // The index from which search will begin

    $sortCriterion1 = new sortCriterion ( );
    $sortCriterion1->order = "ASC";
    $sortCriterion1->attribute = "SIM_STATUS";

    $sortCriterion2 = new sortCriterion ( );
    $sortCriterion2->order = "DESC";
    $sortCriterion2->attribute = "SIM_SSN";

    $searchInConnectivityDirectory = new searchInConnectivityDirectory();
```

M2M API for PHP - Methods

```
$oSearchInConnectivityDirectory->searchCriteria = $oSearchCriteria;
$oSearchInConnectivityDirectory->rangeSize      = $iRangeSize;
$oSearchInConnectivityDirectory->rangeStart    = $iRangeStart;
$oSearchInConnectivityDirectory->sortCriterion1 = $oSortCriterion1;
$oSearchInConnectivityDirectory->sortCriterion2 = $oSortCriterion2;
$oOutput .= M2M_Utility::getBeautifiedOutput($oConnectivityDirectoryClient-
>call_searchInConnectivityDirectory($oSearchInConnectivityDirectory),"searchInConnectivityDirectory");

/**
 * Result
 *
 * isNumberExceeded: If there are more results found than range size, isNumberExceeded is true.
 * totalNumber: the total number of elements in the database (independently of the search request made)
 * If search matches no result, no other parameter is return
 * Else, an array of results is sent back OR an object if there is a single result
 */
?>
```

The output looks like:

```
<pre>searchInConnectivityDirectoryResponse Object
(
    [connectivityDirectory] => Array
        (
            [0] => ConnectivityDirectory Object
                (
                    [customerEnvironmentIdentifier] => 64095147
                    [sim] => Sim Object
                        (
                            [serialNumber] => 1956473384107
                            [puk1] =>
                            [puk2] =>
                            [imei] =>
                            [status] => PRE_ACTIVATED
                            [suspensionReason] =>
                            [lastStatusRefreshDate] => 2010-03-17T16:06:25+01:00
                            [lastStatusChangeDate] => 2010-03-17T16:06:25+01:00
                            [requestedStatus] =>
                            [statusChangeRequestDate] =>
                        )
                    [subscription] => Subscription Object
                        (
                            [identifier] => 23697248
                            [description] => SubscriptionDescription Object
                                (
                                    [value] =>
                                    [user] =>
                                    [userRef] =>
                                    [service] =>
                                )
                            [creationDate] => 2010-03-17T00:00:00+01:00
                            [connectionDate] =>
```

```

        [msisdnData] =>
        [msisdnVoice] =>
        [msisdnFax] =>
    )
[device] => ConnectivityDirectoryDevice Object
(
    [updateDate] => 2010-03-18T17:13:27+01:00
    [uniqueIdentifier] => MyPHPdeviceIdentifier
    [serialNumber] => MyPHPdeviceSerialNumber
    [contact] => Contact Object
    (
        [name] => M2M
        [email] => m2m@orange.com
        [phone] => 0612345678
    )

    [communicationModule] => CommunicationModule Object
    (
        [brand] => PHP TM
        [model] => PHP Model
    )

    [deviceLocation] => MyPHPdeviceLocation
    [category] => MyPHPdeviceCategory
    [description] => MyPHPdeviceDescription
    [address] => MyPHPdeviceAddress
    [deviceHolder] =>
)

[machine] => ConnectivityDirectoryMachine Object
(
    [updateDate] => 2010-03-18T17:13:27+01:00
    [serialNumber] => MyPHPmachineSerialNumber
    [name] => MyPHPmachineName
    [description] => MyPHPmachineDescription
)

[customerIdentifier] => CustomerIdentifier Object
(
    [type] => SIREN
    [identifier] => 304827900
)
)

[1] => ConnectivityDirectory Object
(
    [customerEnvironmentIdentifier] => 64095147
    [sim] => Sim Object
    (
        [serialNumber] => 1956473384115
        [puk1] =>
        [puk2] =>
        [imei] =>
        [status] => PRE_ACTIVATED
        [suspensionReason] =>
        [lastStatusRefreshDate] => 2010-03-17T16:06:25+01:00
        [lastStatusChangeDate] => 2010-03-17T16:06:25+01:00
        [requestedStatus] =>
    )
)

```

```

        [statusChangeRequestDate] =>
    )

    [subscription] => Subscription Object
    (
        [identifier] => 23697251
        [description] => SubscriptionDescription Object
        (
            [value] =>
            [user] =>
            [userRef] =>
            [service] =>
        )

        [creationDate] => 2010-03-17T00:00:00+01:00
        [connectionDate] =>
        [msisdnData] =>
        [msisdnVoice] =>
        [msisdnFax] =>
    )

    [device] =>
    [machine] =>
    [customerIdentifier] => CustomerIdentifier Object
    (
        [type] => SIREN
        [identifier] => 304827900
    )
)

[2] => ConnectivityDirectory Object
(
    [customerEnvironmentIdentifier] => 64095147
    [sim] => Sim Object
    (
        [serialNumber] => 1956473384123
        [puk1] =>
        [puk2] =>
        [imei] =>
        [status] => PRE_ACTIVATED
        [suspensionReason] =>
        [lastStatusRefreshDate] => 2010-03-17T16:06:25+01:00
        [lastStatusChangeDate] => 2010-03-17T16:06:25+01:00
        [requestedStatus] =>
        [statusChangeRequestDate] =>
    )

    [subscription] => Subscription Object
    (
        [identifier] => 23697252
        [description] => SubscriptionDescription Object
        (
            [value] =>
            [user] =>
            [userRef] =>
            [service] =>
        )

        [creationDate] => 2010-03-17T00:00:00+01:00
    )
)

```

```

        [connectionDate] =>
        [msisdnData] =>
        [msisdnVoice] =>
        [msisdnFax] =>
    )

    [device] =>
    [machine] =>
    [customerIdentifier] => CustomerIdentifier Object
    (
        [type] => SIREN
        [identifier] => 304827900
    )

)

)

[resultsNumberExceeded] => 1
[totalResultsNumber] => 20
)
</pre>

```

5. UpdateConnectivityDirectory method

5.1. Description

The UpdateConnectivityDirectory operation updates a set of connectivity directory data, managed by different kinds of identifiers.

5.2. Input parameter

Table 5.7. updateConnectivityDirectory: Input parameters.

Name	Type	Description	Cardinality
Service Configurator	Service Configurator	A Configurator for the M2M service	1..1
Update Connectivity Directory	Update Connectivity Directory	The set of connectivity directory data, managed by different kinds of identifiers (SimSerialNumber, DeviceImei, DeviceSerialNumber, DeviceUniqueId, MachineSerialNumber, MsisdnVoice, MsisdnData and/or MsisdnFax.	1..1

5.3. Output parameter

Table 5.8. updateConnectivityDirectory: output parameters.

Name	Type	Description	Cardinality
Response	UpdateConnectivityDirectoryResponse	An array of connectivity directories updated successfully, and an array of connectivity directories update failed	1..1

5.4. Exceptions

This web method may throw `TechnicalException` or `FunctionalException`.

5.5. Sample code

This sample shows you how to Update a set of Connectivity Directory..

```
<?php
// Prior run this snippet, don't forget to edit the file "m2m.ini"
// TODO: Edit m2m.ini file in "M2M" folder and copy it to "sample" folder
$output = "";
$initializationPath = dirname ( __FILE__ ) . "../M2M/Initialization.php";
try {
    if ( ! require_once ($initializationPath) ) {
        throw new Exception ( "Cannot find Initialization file at $initializationPath" );
    }
    $iniFile = new M2M_IniParser(dirname(__FILE__).DIRECTORY_SEPARATOR."m2m.ini"); // Path to ini
    file
    $settings = new M2M_ServiceConfigurator($iniFile);
    $output .= "<h5>Using URL: " . $settings->get_sConnectivityDirectoryUrl() . "</h5>";
    $connectivityDirectoryClient = new M2M_ConnectivityDirectoryClient($settings);

    //*****//
    //*** UPDATECUSTOMERDEVICEDATA ***//
    //*****//
    $updateConnectivityDirectory = new updateConnectivityDirectory();
    $updateConnectivityDirectory->dataBySubscriptionNumber= new dataBySubscriptionNumber();
    $updateConnectivityDirectory->dataBySubscriptionNumber->subscriptionNumber
= "your_subscription_Number";
    /* Device */
    $updateConnectivityDirectory->dataBySubscriptionNumber->deviceData=new Device();
    $updateConnectivityDirectory->dataBySubscriptionNumber->deviceData->uniqueIdentifier
= "device_unique_identifier";
    $updateConnectivityDirectory->dataBySubscriptionNumber->deviceData->serialNumber
= "device_serial_Number";
    $updateConnectivityDirectory->dataBySubscriptionNumber->deviceData->description
= "device_description";
    $updateConnectivityDirectory->dataBySubscriptionNumber->deviceData->address
= "device_address";
    $updateConnectivityDirectory->dataBySubscriptionNumber->deviceData->category
= "device_category";
    $updateConnectivityDirectory->dataBySubscriptionNumber->deviceData->deviceLocation
= "device_location";
    $updateConnectivityDirectory->dataBySubscriptionNumber->deviceData->contact
= "device_contact";
    /*Machine*/
    $machineData = new Machine();
    $machineData->serialNumber = "123456";
    $machineData->name = "machine_name";
    $machineData->description = "machine_description";
    $updateConnectivityDirectory->dataBySubscriptionNumber->machineData = $machineData;

    $output .= M2M_Utility::getBeautifiedOutput($connectivityDirectoryClient-
>call_updateConnectivityDirectory($updateConnectivityDirectory), "updateConnectivityDirectory");
    echo $output;
}
```

```
?>
```

The output looks like:

```
<pre>updateConnectivityDirectoryResponse Object
(
  [updatesCompleted] => UpdatesCompletedList Object
    (
      [lineIdentifier] => LineIdentifier Object
        (
          [subscriptionNumber] => 23697248
          [simSerialNumber] =>
          [deviceImei] =>
          [deviceSerialNumber] =>
          [deviceUniqueId] =>
          [machineSerialNumber] =>
          [msisdnVoice] =>
          [msisdnData] =>
          [msisdnFax] =>
        )
      )
    )
  [updatesFailed] =>
)
</pre>
```

6. SubmitUpdateSimStatus and GetUpdateSimStatus methods

6.1. SubmitUpdateSimStatus method

6.1.1. Description

Submit an update of SIM cards's status for a set of SIM cards.

Tip

The method `SubmitUpdateSimStatus` is asynchronous and has to be used prior the `GetUpdateSimStatus` method call which gets back the submit result.

6.1.2. Input parameter

Table 5.9. submitUpdateSimStatus: Input parameters.

Name	Type	Description	Cardinality
Service Configurator	Service Configurator	A Configurator for the M2M service	1..1
LineIdentifier Collection	LineIdentifier Collection	A collection of Line Identifiers	1..1
RequestedStatus	String	Can be ACTIVATED, ACTIVATED_FOR_TEST or SLEEPING	1..1
TestMode	String	Can be INTERNATIONAL, EUROPE, METROPOLE or AU_COMPTEUR	1..1

6.1.3. Output parameter

Table 5.10. submitUpdateSimStatus: output parameters.

Name	Type	Description	Cardinality
Response	SubmitUpdateSim StatusRequestResponse	A ticket number useable in the getUpdateSimStatus request, and a collection of unknown requested LineIdentifiers if any.	1..1

6.1.4. Exceptions

This web method may throw `TechnicalException` or `FunctionalException`.

6.2. GetUpdateSimStatus method

6.2.1. Description

Get the result of a `SubmitUpdateSimStatus` method call.

Tip

The method `GetUpdateSimStatus` has to be used after a `SubmitUpdateSimStatus` method call, in order to get the asynchronous submit result.

As the method is the result of a preliminar asynchrone submit call, the result of the get method contains a Ticket Status telling wether the submit call work is finished, in progress or not started yet.

6.2.2. Input parameter

Table 5.11. getUpdateSimStatus: Input parameters.

Name	Type	Description	Cardinality
Service Configurator	Service Configurator	A Configurator for the M2M service	1..1
TicketNumber	Integer	A ticket number returned by the submitUpdateSimStatus method call.	1..1

6.2.3. Output parameter

Table 5.12. getUpdateSimStatus: output parameters.

Name	Type	Description	Cardinality
Response	GetUpdateSimStatus ResultResponse	A global ticket status telling whether the request is being processed, in progress or terminated ; an array of StatusUpdate (one for each requested LineIdentifier); a reminder of SImSimStatus and TestMode values ; an array of unknown line identifier if any.	1..1

6.2.4. Exceptions

This web method may throw `TechnicalException` or `FunctionalException`.

6.2.5. Sample code

This sample shows how to Get SIM Status Update of a SIM card.

```
<?
// Prior run this snippet, don't forget to edit the file "m2m.ini"
// TODO: Edit m2m.ini file in "M2M" folder and copy it to "sample" folder
$output = "";
$InitializationPath = dirname ( __FILE__ ) . "../M2M/Initialization.php";
try {
    ini_set("max_execution_time", 50); // for demo purpose, as we will make php "sleep" for a while
    if ( ! require_once ($InitializationPath) )
        throw new Exception ( "Cannot find Initialization file at $InitializationPath" );
    $IniFile = new M2M_IniParser(dirname(__FILE__).DIRECTORY_SEPARATOR."m2m.ini"); // Path to ini file
    $Settings = new M2M_ServiceConfigurator($IniFile);
    $SimLifecycleManagementClient = new M2M_SimLifecycleManagementClient($Settings);
    $Output .= "<h5>Using URL: " . $Settings->get_sSimLifecycleManagementUrl() . "</h5>";

    //*****
    /*** SUBMITUPDATESIMSTATUS ***/
    //*****

    $Params = new submitUpdateSimStatusRequest();
    $Params->requestedStatus = "your_requested_Status";
    $Params->testMode = "your_testMode";
```

```

$oIdentifiers = new LineIdentifierCollection();
$oIdentifiers->subscriptionNumber = "your_subscription_number";
$oParams->lineIdentifiers=$oIdentifiers;
$oSubmitUpdateSimStatusRequestOutput = $oSimLifeCycleManagementClient-
>call_submitUpdateSimStatusRequest($oParams);

$sOutput .= M2M_Utility::getBeautifiedOutput($oSubmitUpdateSimStatusRequestOutput,
    "submitUpdateSimStatusRequest");

//*****//
//*** GETUPDATESIMSTATUS ***//
//*****//
// To test this webmethod, you may want to run call_submitUpdateSimStatusRequest so you can get
a ticket number
$iTicketNumber = $oSubmitUpdateSimStatusRequestOutput->simStatusUpdateOut->idTicket;
$oGetUpdateSimStatusResponse = $oSimLifeCycleManagementClient-
>call_getUpdateSimStatus($iTicketNumber);
$sIds = "your_Ids";
$sOutput .= M2M_Utility::getBeautifiedOutput($oGetUpdateSimStatusResponse, "getUpdateSimStatus
(ticket Number $iTicketNumber and for ids $sIds)");

sleep(20);

// Call the operation, later
$oGetUpdateSimStatusResponse = $oSimLifeCycleManagementClient-
>call_getUpdateSimStatus($iTicketNumber);
$sOutput .= M2M_Utility::getBeautifiedOutput($oGetUpdateSimStatusResponse, "getUpdateSimStatus
(a bit later, ticket Number $iTicketNumber and for ids $sIds)", "getUpdateSimStatusLater");
}
catch ( Exception $e ) {
    $sOutput .= M2M_Utility::getBeautifiedException($e);
}
echo $sOutput;
?>

```

The output looks like:

```

<pre>submitUpdateSimStatusRequestResponse Object
(
    [ticketNumber] => 322
    [unknownLineIdentifiers] =>
)
</pre></h4><hr><a name="getUpdateSimStatus (ticket Number 322 )" ></a><font
color="blue"><h4>Response of getUpdateSimStatus (ticket Number 322 ) method:</font>
<pre>getUpdateSimStatusResultResponse Object
(
    [globalProcessingStatus] => WAITING
    [errorCode] =>
    [errorMessage] =>
    [requestedStatus] => ACTIVATED_FOR_TEST
    [testMode] => INTERNATIONAL
    [statusUpdate] => StatusUpdate Object
        (
            [inputLineIdentifier] => LineIdentifier Object
                (

```

```

        [subscriptionNumber] => 23697253
        [simSerialNumber] =>
        [deviceImei] =>
        [deviceSerialNumber] =>
        [deviceUniqueId] =>
        [machineSerialNumber] =>
        [msisdnVoice] =>
        [msisdnData] =>
        [msisdnFax] =>
    )

    [processingStatus] => WAITING
    [progressMessage] =>
    [errorCode] =>
    [errorMessage] =>
    [errorDate] =>
    [updateDate] =>
    [plannedModeExitDate] =>
)

[unknownLineIdentifiers] =>
)
</pre>

```

7. SubmitConsumptionTracking and GetConsumptionTracking methods

7.1. SubmitConsumptionTracking method

7.1.1. Description

Submit an Consumption Tracking request for a set of SIM cards in a given period of time. This method enables to consult a full traffic exchanged between a set of SIM cards, devices or machines and the Orange network.

Tip

The method SubmitConsumptionTracking is asynchronous and has to be used prior the getConsumptionTracking method call which gets back the submit result.

7.1.2. Input parameter

Table 5.13. submitConsumptionTracking: Input parameters.

Name	Type	Description	Cardinality
Service Configurator	Service Configurator	A Configurator for the M2M service	1..1
LineIdentifier Collection	LineIdentifier Collection	A collection of Line Identifiers	1..1
StartDate	Date	The start date of the request	1..1
EndDate	Date	The end date of the request	1..1

7.1.3. Output parameter

Table 5.14. submitConsumptionTracking: output parameters.

Name	Type	Description	Cardinality
Response	SubmitConsumptionTrackingRequestResponse	A ticket number useable in the getConsumptionTracking request, and a collection of unknown requested LineIdentifiers if any.	1..1

7.1.4. Exceptions

This web method may throw `TechnicalException` or `FunctionalException`.

7.2. GetConsumptionTracking method

7.2.1. Description

Get the result of a `SubmitConsumptionTracking` method call.

Tip

The method `GetConsumptionTracking` has to be used after a `SubmitConsumptionTracking` method call, in order to get the asynchronous submit result.

As the method is the result of a preliminar asynchrone submit call, the result of the get method contains a `Ticket Status` telling wether the submit call work is finished, in progress or not started yet.

7.2.2. Input parameter

Table 5.15. getConsumptionTracking: Input parameters.

Name	Type	Description	Cardinality
ServiceConfigurator	ServiceConfigurator	A Configurator for the M2M service	1..1
TicketNumber	Integer	A ticket number returned by the <code>submitConsumptionTracking</code> method call.	1..1

7.2.3. Output parameter

Table 5.16. getConsumptionTracking: output parameters.

Name	Type	Description	Cardinality
Response	GetConsumptionTrackingResultResponse	A global ticket status telling wether the request is being processed, in progress or terminated ; an array of Customer Identifier ; an array of Billing Account ; an array of Line (i.e. volume of consumption, i.e. CDRs for each Line) ; the global number of CDRs returned, the first CDR date and last CDR date, and an array of unknown line identifier if any.	1..1

7.2.4. Exceptions

This web method may throw `TechnicalException` or `FunctionalException`.

7.2.5. Sample code

This sample shows how to Get Subscription Status of a SIM card

```
<?php
// Prior run this snippet, don't forget to edit the file "m2m.ini"
// TODO: Edit m2m.ini file in "M2M" folder and copy it to "sample" folder

$output = "";
$initializationPath = dirname ( __FILE__ ) . "../M2M/Initialization.php";
try {
    ini_set("max_execution_time", 50); // for demo purpose, as we will make php "sleep" for a while
    if ( ! require_once ($initializationPath) ) {
        throw new Exception ( "Cannot find Initialization file at $initializationPath" );
    }
    $iniFile = new M2M_IniParser(dirname(__FILE__).DIRECTORY_SEPARATOR."m2m.ini"); // Path to ini
    file
    $settings = new M2M_ServiceConfigurator($iniFile);
    $consumptionTrackingClient = new M2M_ConsumptionTrackingClient($settings);
    $output .= "<h5>Using URL: " . $settings->get_sConsumptionTrackingUrl() . "</h5>";

    /**
     * SUBMITCONSUMPTIONTRACKINGREQUEST
     */
    $submitConsumptionTrackingRequest = new submitConsumptionTrackingRequest();
    $lineIdentifierCollection=new LineIdentifierCollection();
    $lineIdentifierCollection->subscriptionNumber="your_subscription_number";
    $submitConsumptionTrackingRequest->lineIdentifiers=$lineIdentifierCollection;

    $startDate = new Zend_Date("2010-01-11T08:02:09+01:00", Zend_Date::ATOM);
    $submitConsumptionTrackingRequest->periodStartDateTime= $startDate-
toString(Zend_Date::ATOM);
    $endDate = new Zend_Date("2010-02-11T08:02:09+01:00", Zend_Date::ATOM); // Comparison greater
or equals.
    $submitConsumptionTrackingRequest->periodEndDateTime=$endDate->toString(Zend_Date::ATOM);

    // call the operation
    $submitConsumptionTrackingResponse = $consumptionTrackingClient-
>call_submitConsumptionTrackingRequest ($submitConsumptionTrackingRequest);

    //output
    $output =
M2M_UTILITY::getBeautifiedOutput($submitConsumptionTrackingResponse, "submitConsumptionTrackingRequest");
    $ticketNumber = $submitConsumptionTrackingResponse->ticketNumber;

    /**
     * GETCONSUMPTIONTRACKING
     */
    /* $ticketNumber:@var string */
    // create input object and set values
    $getConsumptionTracking = new getConsumptionTracking();
    $getConsumptionTracking->ticketNumber = $ticketNumber; // Retrieving ticket number we've just
created.
```

```

// call the operation
$rGetConsumptionTracking = $oConsumptionTrackingClient-
>call_getConsumptionTracking($oGetConsumptionTracking);
//output
$output .= M2M_Utility::getBeautifiedOutput($rGetConsumptionTracking, "GetConsumptionTracking
(for ticket number $sTicketNumber)");
sleep(20); // Call the operation 20s later
// call the operation, later
$output .= "A bit later, recall the operation... ";
$output .= M2M_Utility::getBeautifiedOutput($oConsumptionTrackingClient-
>call_getConsumptionTracking($oGetConsumptionTracking), "GetConsumptionTracking (for ticket
number $sTicketNumber)", "GetConsumptionTrackingLater");
}
catch ( Exception $e ) {
    $output .= M2M_Utility::getBeautifiedException($e);
}
echo $output; // Display HTML page
?>

```

The output looks like:

```

<pre>submitConsumptionTrackingRequestResponse Object
(
    [ticketNumber] => 295
    [unknownLineIdentifiers] =>
)
</pre></h4><hr><a name="GetConsumptionTracking (for ticket number 295)"></a><font
color="blue"><h4>Response of GetConsumptionTracking (for ticket number 295) method:</font>
<pre>getConsumptionTrackingResponse Object
(
    [globalProcessingStatus] => WAITING
    [errorCode] =>
    [errorMessage] =>
    [totalCdrNumber] =>
    [firstCdrDateTime] =>
    [lastCdrDateTime] =>
    [customerIdentifier] =>
    [billingAccount] =>
    [line] => Line Object
        (
            [inputLineIdentifier] => LineIdentifier Object
                (
                    [subscriptionNumber] => 23697249
                    [simSerialNumber] =>
                    [deviceImei] =>
                    [deviceSerialNumber] =>
                    [deviceUniqueId] =>
                    [machineSerialNumber] =>
                    [msisdnVoice] =>
                    [msisdnData] =>
                    [msisdnFax] =>
                )
            [processingStatus] => WAITING
            [errorCode] =>

```

```

        [errorMessage] =>
        [lineConsumptionTracking] =>
    )

    [unknownLineIdentifiers] =>
)
</pre>

```

8. GetNetworkStatus method

8.1. Description

Get Network Status information from given GPS coordinates, i.e. in a given geographic area.

8.2. Input parameter

Table 5.17. getNetworkStatus: Input parameters.

Name	Type	Description	Cardinality
Service Configurator	Service Configurator	A Configurator for the M2M service	1..1
coordinates	Coordinates	GPS coordinates in WGS84 / DMS0 format. Examples : N48d48m41s0.0 - E2d19m39s0.0 N45d0m14s0.0 - W1d11m49s0.0	1..1

8.3. Output parameter

Table 5.18. getNetworkStatus: output parameters.

Name	Type	Description	Cardinality
Response	GetNetwork StatusResponse	A Network Stat Status telling whether the response is OK or not ; an Area Coverage array describing for each technology (2G, 3G, ...) if the network is covered or not, with a brief description ; an Breakdown Report array telling for each kind of network (2G Voice, 2G Data, 3G voice, ...) the kind of incident occurring, with a brief description.	1..1

8.4. Exceptions

This web method may throw `TechnicalException` or `FunctionalException`.

8.5. Sample code

This sample shows you how Get a Network Status from a GPS coordinates data.

```

<?php
$sOutput = "";
$sInitializationPath = dirname ( __FILE__ ) . "../M2M/Initialization.php";
try {
    ini_set("max_execution_time", 50); // for demo purpose, as we will make php "sleep" for a while
    if ( ! require_once ($sInitializationPath) ) {
        throw new Exception ( "Cannot find Initialization file at $sInitializationPath" );
    }
    $oIniFile = new M2M_IniParser(dirname(__FILE__).DIRECTORY_SEPARATOR."m2m.ini"); // Path to ini
file
    $oSettings = new M2M_ServiceConfigurator($oIniFile);
    $oNetworkStatusClient = new M2M_NetworkStatusClient($oSettings);
    $sOutput .= "<h5>Using URL: " . $oSettings->get_sNetworkStatusUrl() . "</h5>";

    /*******//
    /*** GET NetworkStatus ***/
    /*******//
    $oNetworkStatus = new getNetworkStatusResponse();
    // coordinates: @var (object)Coordinates
    $oNetworkStatus->coordinates=new Coordinates();
    $oNetworkStatus->coordinates->latitude="N48d49m43s0.0";
    $oNetworkStatus->coordinates->longitude="E2d18m26s0.0";
    // call the operation
    $sOutput .= M2M_Utility::getBeautifiedOutput($oNetworkStatusClient-
>call_getNetworkStatus($oNetworkStatus), "GetNetworkStatus");
}
catch ( Exception $e ) {
    $sOutput .= M2M_Utility::getBeautifiedException($e);
}
echo $sOutput; // Display HTML page
?>

```

The output looks like:

```

<pre>getNetworkStatusResponse Object
(
    [status] => OK
    [errorMessage] =>
    [areaCoverage] => Array
        (
            [0] => AreaCoverage Object
                (
                    [technology] => TECH_2G
                    [coverage] => 1
                )
            [1] => AreaCoverage Object
                (
                    [technology] => TECH_3G_2100
                    [coverage] =>
                )
            [2] => AreaCoverage Object
                (

```

```

        [technology] => TECH_3G_900
        [coverage] =>
    )
)
[areaCoverageDescription] => 2G EDGE
[breakdownReport] => Array
(
    [0] => BreakdownReport Object
    (
        [communicationType] => DATA_3G
        [classification] => NO_INFORMATION_AVAILABLE
    )
    [1] => BreakdownReport Object
    (
        [communicationType] => VOICE_3G
        [classification] => MINOR_INCIDENT
    )
    [2] => BreakdownReport Object
    (
        [communicationType] => VOICE_2G
        [classification] => MAJOR_INCIDENT
    )
    [3] => BreakdownReport Object
    (
        [communicationType] => DATA_2G
        [classification] => NO_INCIDENT
    )
)
[breakdownReportDescription] =>
)
</pre>

```

9. GetIncidentDiagnostics method

9.1. Description

Get an Incident diagnostics information concerning a Line Identifier.

9.2. Input parameter

Table 5.19. getIncidentDiagnostics: Input parameters.

Name	Type	Description	Cardinality
Service Configurator	Service Configurator	A Configurator for the M2Mservice	1..1
LineIdentifier	LineIdentifier	A Line Identifier	1..1

9.3. Output parameter

Table 5.20. getIncidentDiagnostics: output parameters.

Name	Type	Description	Cardinality
Response	GetIncidentDiagnosticsResponse	An IncidentDiagnostics response for the given LineIdentifier concerning geolocation data of the SIM card, the last known geolocation area of the SIM card, and the network status concerning the SIM card area.	1..1

9.4. Exceptions

This web method may throw `TechnicalException` or `FunctionalException`.

9.5. Sample code

This sample shows you how get an incident diagnostics result from a collection of line identifiers

```
<?php
// Prior run this snippet, don't forget to edit the file "m2m.ini"
// TODO: Edit m2m.ini file in "M2M" folder and copy it to "sample" folder
$sOutput = "";
$sInitializationPath = dirname ( __FILE__ ) . "../M2M/Initialization.php";
try {
    ini_set("max_execution_time", 50); // for demo purpose, as we will make php "sleep" for a while

    if ( ! require_once ($sInitializationPath) ) {
        throw new Exception ( "Cannot find Initialization file at $sInitializationPath" );
    }
    $oIniFile = new M2M_IniParser(dirname(__FILE__).DIRECTORY_SEPARATOR."m2m.ini"); // Path to ini
file
    $oSettings = new M2M_ServiceConfigurator($oIniFile);
    $oIncidentDiagnosticsClient = new M2M_IncidentDiagnosticsClient($oSettings);
    $sOutput .= "<h5>Using URL: " . $oSettings->get_sIncidentDiagnosticsUrl() . "</h5>";

    /*******
    /**** GET INCIDENT DIAGNOSTICS      ***/
    /*******
    $oIncidentDiagnostics = new getIncidentDiagnostics();
    $oLineIdentifier= new LineIdentifierCollection();
    $oLineIdentifier->subscriptionNumber="your_subscription_number";
    $oIncidentDiagnostics->lineIdentifier=$oLineIdentifier;
    // call the operation
    $sOutput .= M2M_Utility::getBeautifiedOutput($oIncidentDiagnosticsClient-
>call_getIncidentDiagnostics($oIncidentDiagnostics), "GetIncidentDiagnostics");
}
catch ( Exception $e ) {
    $sOutput .= M2M_Utility::getBeautifiedException($e);
}
echo $sOutput; // Display HTML page
?>
```

The output looks like:

```

<pre>
<h2>Created on 13 avr. 2010 09:24:57</H2><h5>Using URL: https://80.12.215.206/
MLM/IncidentDiagnostics-1</h5><hr><a name="GetIncidentDiagnostics"></
a><font color="blue"><h4>Response of GetIncidentDiagnostics method:</font>
<pre>getIncidentDiagnosticsResponse Object
(
    [incidentDiagnosticsStatus] => NO_GEOLOCATION_AVAILABLE
    [geolocationInformation] => GeolocationOut Object
        (
            [geolocationErrorCode] => 12
            [geolocationErrorMessage] => Not a member
            [geolocationDate] =>
        )
    [networkStatusResponse] =>
)
</pre></h4>
</pre>

```

10. SubmitSessionHistory and GetSessionHistory methods

10.1. SubmitSessionHistory method

10.1.1. Description

Submit an update of SIM cards's status for a set of SIM cards.

Tip

The method `SubmitSessionHistory` is asynchronous and has to be used prior the `getSessionHistory` method call which gets back the submit result.

10.1.2. Input parameter

Table 5.21. submitSessionHistory: Input parameters.

Name	Type	Description	Cardinality
Service Configurator	Service Configurator	A Configurator for the M2M service	1..1
LineIdentifier	LineIdentifier	A unique Line Identifiers	1..1
StartDate	Date	The start date of the request	1..1
EndDate	Date	The end date of the request	1..1

10.1.3. Output parameter

Table 5.22. submitSessionHistory: output parameters.

Name	Type	Description	Cardinality
TicketNumber	Integer	A ticket number useable in the getSessionHistory request.	1..1

10.1.4. Exceptions

This web method may throw `TechnicalException` or `FunctionalException`.

10.2. GetSessionHistory method

10.2.1. Description

Get the result of a `SubmitSessionHistory` method call.

Tip

The method `GetSessionHistory` has to be used after a `SubmitSessionHistory` method call, in order to get the asynchronous submit result.

As the method is the result of a preliminar asynchrone submit call, the result of the get method contains a `Ticket Status` telling wether the submit call work is finished, in progress or not started yet.

10.2.2. Input parameter

Table 5.23. getSessionHistory: Input parameters.

Name	Type	Description	Cardinality
Service Configurator	Service Configurator	A Configurator for the M2M service	1..1
TicketNumber	Integer	A ticket number returned by the submitSessionHistory method call.	1..1

10.2.3. Output parameter

Table 5.24. getSessionHistory: output parameters.

Name	Type	Description	Cardinality
Response	<code>GetSessionHistoryResultResponse</code>	A global ticket status telling wether the request is being processed, in progress or terminated ; a Billing Account data ; a Line data (referring an array of <code>CustomerData</code> , a <code>Subscription</code> , a <code>SIM card</code> , a <code>Device</code> , a machine identifier) ; and an array of <code>Session</code> .	1..1

10.2.4. Exceptions

This web method may throw `TechnicalException` or `FunctionalException`.

10.2.5. Sample code

This sample shows how to Get Session History of a SIM card.

```
<?php
// Prior run this snippet, don't forget to edit the file "m2m.ini"
// TODO: Edit m2m.ini file in "M2M" folder and copy it to "sample" folder
$output = "";
$initializationPath = dirname ( __FILE__ ) . "../M2M/Initialization.php";
try {
    ini_set("max_execution_time", 50); // for demo purpose, as we will make php "sleep" for a while

    if ( ! require_once ($initializationPath) ) {
        throw new Exception ( "Cannot find Initialization file at $initializationPath" );
    }
    $iniFile = new M2M_IniParser(dirname(__FILE__).DIRECTORY_SEPARATOR."m2m.ini"); // Path to ini
    file
    $settings = new M2M_ServiceConfigurator($iniFile);
    $sessionHistoryClient = new M2M_SessionHistoryClient($settings);
    $output .= "<h5>Using URL: " . $settings->get_sSessionHistoryUrl() . "</h5>";
    //***** submitSessionHistory *****//
    //***** submitSessionHistoryRequest *****//
    $submitSessionHistoryRequest = new submitSessionHistoryRequest();
    $submitSessionHistoryRequest->lineIdentifier = new LineIdentifier();
    $submitSessionHistoryRequest->lineIdentifier->subscriptionNumber='23697253';
    $startDate = new Zend_Date("2010-01-11T08:02:09+01:00", Zend_Date::ATOM);
    $submitSessionHistoryRequest->periodStartDateTime= $startDate->toString(Zend_Date::ATOM);
    $endDate = new Zend_Date("2010-02-11T08:02:09+01:00", Zend_Date::ATOM); // Comparison greater
    or equals.
    $submitSessionHistoryRequest->periodEndDateTime=$endDate->toString(Zend_Date::ATOM);
    $submitSessionHistoryOutput = $sessionHistoryClient-
>call_submitSessionHistoryRequest($submitSessionHistoryRequest);
    $ticketNumber=$submitSessionHistoryOutput->ticketNumber;
    // call the operation
    $output .= M2M_Utility::getBeautifiedOutput($sessionHistoryClient-
>call_submitSessionHistoryRequest($submitSessionHistoryRequest), "SubmitSessionHistoryRequest");

    //***** getSessionHistory *****//
    $sessionHistory= new getSessionHistory();
    $sessionHistory->ticketNumber=$ticketNumber;
    // call the operation
    $output .= M2M_Utility::getBeautifiedOutput($sessionHistoryClient-
>call_getSessionHistory($sessionHistory), "GetSessionHistory");
    sleep(30); // Call the operation 30s later
    $output .= "Call the operation 30s later";
    $output .= M2M_Utility::getBeautifiedOutput($sessionHistoryClient-
>call_getSessionHistory($sessionHistory), "GetSessionHistory");
}
catch ( Exception $e ) {
```

```

    $sOutput .= M2M_Utility::getBeautifiedException($e);
}
echo $sOutput; // Display HTML page
?>

```

The output looks like:

```

<?xml version="1.0" encoding="UTF-8"?>
<![CDATA[
</pre></h4><hr><a name="GetSessionHistory"></a><font color="blue"><h4>Response of
GetSessionHistory method:</font> <pre>getSessionHistoryResponse Object
(
    [globalProcessingStatus] => WAITING
    [errorCode] =>
    [errorMessage] =>
    [line] =>
    [billingAccount] =>
    [sessions] =>
)
</pre></h4>Call the operation 30s later<hr><a name="GetSessionHistory"></a><font
color="blue"><h4>Response of GetSessionHistory method:</font> <pre>getSessionHistoryResponse
Object
(
    [globalProcessingStatus] => TERMINATED
    [errorCode] =>
    [errorMessage] =>
    [line] => Line Object
        (
            [inputLineIdentifier] =>
            [processingStatus] =>
            [errorCode] =>
            [errorMessage] =>
            [lineConsumptionTracking] =>
            [customerIdentifier] => CustomerIdentifier Object
                (
                    [type] => SIREN
                    [identifier] => 304827900
                )
            [subscription] => Subscription Object
                (
                    [identifier] => 23697253
                    [description] => SubscriptionDescription Object
                        (
                            [value] => Business M To M Pack 2 Mo
                            [user] => CHAB3
                            [userRef] =>
                            [service] =>
                        )
                    [creationDate] => 2010-03-17T00:00:00+01:00
                    [connectionDate] =>
                    [msisdnData] =>
                    [msisdnVoice] =>
                    [msisdnFax] =>
                )
            [sim] => Sim Object

```

```

        (
            [serialNumber] => 1956473384131
            [puk1] =>
            [puk2] =>
            [imei] =>
            [status] => PRE_ACTIVATED
            [suspensionReason] =>
            [lastStatusRefreshDate] => 2010-04-19T10:50:03.333+02:00
            [lastStatusChangeDate] =>
            [requestedStatus] =>
            [statusChangeRequestDate] =>
        )
    )

[billingAccount] => BillingAccount Object
(
    [billingAccountNumber] => 64095147
    [billingAddress] => Address Object
    (
        [streetNameAndNumber] => RUE VICTOR
        [complements] =>
        [localityOrPOBox] => RUE VICTOR
        [postCode] => 33600
        [town] => PESSAC
        [country] => FR
    )

    [linesConsumptionTrackingSynthesis] =>
    [lineNumber] =>
    [linesReferences] =>
)

[sessions] => Sessions Object
(
    [periodStartDateTime] => 2010-01-11T08:02:09+01:00
    [periodEndDateTime] => 2010-02-11T08:02:09+01:00
    [totalNumberOfResults] => 0
    [firstSessionDateTime] =>
    [lastSessionDateTime] =>
    [numberOfRemainingResults] => 0
    [session] =>
)
)
</pre></h4>

```

Appendix A. Error codes

Caution

The error message returned by the M2M API may contain a reference to <user name>. It refers to the couple Access Key - Access Key Password used for authentication.

Table A.1. Error codes

Code	Detail
malimaErrorFault	<p>Error message: Une erreur Malima est survenue..</p> <p>Can be of several kinds:</p> <p>MLM_UNKNOWN_USER : L'utilisateur <user name> est inconnu de Malima.</p> <p>Input parameter: <i>user name</i>.</p> <p>CANNOT_CONNECT_TO_DATABASE : Accès impossible à la base de donnée.</p> <p>MLM_TICKET_CREATION : La création du ticket a échoué.</p> <p>MLM_NETWORKSTATUS_SERVICE_CONNECTION_ERROR : Impossible de se connecter au service de météo du réseau.</p> <p>MLM_NETWORKSTATUS_SERVICE_ACCESS_DENIED : Accès au service de météo du réseau non autorisé.</p> <p>MLM_GEOLOC_SERVICE_CONNECTION_ERROR : Impossible de se connecter au service de géolocalisation.</p> <p>MLM_GEOLOC_SERVICE_ACCESS_DENIED : Accès au service de géolocalisation non autorisé.</p>
unknownLineIdentifier ErrorFault	<p>Error message: Unknwon line identifier / Identifiant de ligne inconnu..</p> <p>Description: choose only one of the following element:</p> <p>subscriptionNumber, simSerialNumber, deviceImei, deviceSerialNumber, deviceUniqueId, machineSerialNumber, msisdnVoice, msisdnData, msisdnFax.</p>
tooManyLine IdentifiersFault	<p>Error message: Too many lines identifiers.</p> <p>Description: Le nombre d'identifiants de ligne est trop élevé (<maximum lines number> au maximum).</p> <p>Input parameter: <i>maxAllowedLineIdentifiersNumber</i>.</p>

Code	Detail
invalidParameterFault	Error message: At least one parameter is not valid. Description: Au moins un paramètre n'est pas valide.
unknownTicketFault	Error message: Unknown ticket number. Description: Le ticket <ticket number> est inconnu. Input parameter: <i>unknownTicketNumber</i>
resultAlreadyRetrievedFault	Error message: If a user try to retrieve a result he has already retrieved. Description: Le résultat a déjà été récupéré le <first retrieval date>. Input parameter: <i>retrievalDate</i>
networkStatusServiceErrorFault	Error message: networkStatusServiceErrorFault : Network status IS error. Description: Une erreur est survenue dans le service de météo du réseau.
invalidPeriodErrorFault	Error message: Invalid period error. Description: Période de recherche non valide. Malima error messages listing : La date de fin est antérieure à la date de début. La date de début de recherche remonte à plus d'un an. La durée de la période de recherche excède 30 jours.

More specific error can be returned into M2M API responses, they are returned into M2M tickets responses when requesting asynchronous (i.e. submitXxx, then getXxx requests) queries:

Table A.2. Ticket Error codes

Code	Detail
MLM_TELCO_IS_ERROR	[0000000573] Un seul critère de positionnement doit être renseigné.
MLM_TELCO_IS_ERROR	[0000000574] Un critère de positionnement doit être renseigné.
MLM_TELCO_IS_ERROR	[0000000575] Le N° d'abonnement est inconnu.
MLM_TELCO_IS_ERROR	[0000000581] Non autorisé à modifier l'abonnement.
MLM_TELCO_IS_ERROR	[0000000584] La date d'effet doit être supérieure ou égale à la date du jour.
MLM_TELCO_IS_ERROR	[0000000585] Le Code service est inconnu.

M2M API for PHP - Error codes

Code	Detail
MLM_TELCO_IS_ERROR	[0000000586] Le code service est inconnu
MLM_TELCO_IS_ERROR	[0000000597] Le code service a une date d'effet inférieure à la date de connexion.
MLM_TELCO_IS_ERROR	[0000000598] Le code service est recurrent et existe déjà sur l'abonné.
MLM_TELCO_IS_ERROR	[0000000600] La date d effet est invalide elle doit être au premier du mois.
MLM_TELCO_IS_ERROR	[0000000655] La zone numérique passée en paramètre n'est pas valide.
MLM_TELCO_IS_ERROR	[0000000675] Aucun abonné n'est éligible.
MLM_TELCO_IS_ERROR	[0000000711] Positionnement impossible le service existe déjà sur l'abonné.
MLM_TELCO_IS_ERROR	[0000000794] La zone numerique passée en parametre n'est pas valide.
MLM_TELCO_IS_ERROR	[0000000795] La date passée en parametre n'est pas valide.
MLM_TELCO_IS_ERROR	[0000000796] Le N° d'abonnement est obligatoire.
MLM_TELCO_IS_ERROR	[0000000800] Le N° d'abonnement est inconnu.
MLM_TELCO_IS_ERROR	[0000000842] Le nombre d'abonnés dépasse le plafond.
MLM_TELCO_IS_ERROR	[0000000844] La zone numérique passée en paramètre n'est pas valide.
MLM_TELCO_IS_ERROR	[0000000848] Le nombre d'itérations demandées dépasse le seuil max. de restitution.
MLM_TELCO_IS_ERROR	[0000000987] Un seul critère de positionnement doit être renseigné.
MLM_TELCO_IS_ERROR	[0000000988] Un critère de positionnement doit être renseigné.
MLM_TELCO_IS_ERROR	[0000000989] Le N° d'abonnement est inconnu.
MLM_TELCO_IS_ERROR	[0000000995] Non autorisé à modifier l'abonnement.
MLM_TELCO_IS_ERROR	[0000000996] Renseigner code option ou code type de modification.
MLM_TELCO_IS_ERROR	[0000000997] Le code modification est obligatoire.
MLM_TELCO_IS_ERROR	[0000000998] Code option invalide.
MLM_TELCO_IS_ERROR	[0000000999] Code type de modification et/ou Code modification invalides.
MLM_TELCO_IS_ERROR	[0000001000] Type de suspension obligatoire.
MLM_TELCO_IS_ERROR	[0000001001] N° d'abonnement obligatoire.
MLM_TELCO_IS_ERROR	[0000001008] Attention, abonné déconnecté ou en attente de résiliation.

M2M API for PHP - Error codes

Code	Detail
MLM_TELCO_IS_ERROR	[0000001009] Attention, abonné non connecté ou en attente de connexion.
MLM_TELCO_IS_ERROR	[0000001010] Attention, abonné suspendu ou en attente de suspension.
MLM_TELCO_IS_ERROR	[0000001012] Demande incompatible avec le tarif de l'abonné.
MLM_TELCO_IS_ERROR	[0000001013] Service incompatible avec l'option demandée
MLM_TELCO_IS_ERROR	[0000001019] Migration non autorisée.
MLM_TELCO_IS_ERROR	[0000001023] Date d'effet incorrecte.
MLM_TELCO_IS_ERROR	[0000001024] Type de suspension incorrect.
MLM_TELCO_IS_ERROR	[0000001025] Demande non applicable.
MLM_TELCO_IS_ERROR	[0000001026] Contact inconnu.
MLM_TELCO_IS_ERROR	[0000001028] Demande non applicable.
MLM_TELCO_IS_ERROR	[0000001035] N° abonnement obligatoire.
MLM_TELCO_IS_ERROR	[0000001037] Date d'effet incorrecte.
MLM_TELCO_IS_ERROR	[0000001038] Demande non applicable.
MLM_TELCO_IS_ERROR	[0000001039] Valeur non autorisée.
MLM_TELCO_IS_ERROR	[0000001040] L'événement n'a pas été annulé.
MLM_TELCO_IS_ERROR	[0000001044] Demande non applicable.
MLM_TELCO_IS_ERROR	[0000001045] Aucune connexion en attente.
MLM_TELCO_IS_ERROR	[0000001046] Abonné sans suspension programmée.
MLM_TELCO_IS_ERROR	[0000001047] Abonné sans résiliation programmée.
MLM_TELCO_IS_ERROR	[0000001048] Abonné sans rétablissement programmé.
MLM_TELCO_IS_ERROR	[0000001049] Abonné sans activation programmée.
MLM_TELCO_IS_ERROR	[0000001095] L'abonné doit être actif pour ce service.
MLM_TELCO_IS_ERROR	[0000001097] Le Numéro de Carte SIM (NSCE) n'existe pas.
MLM_TELCO_IS_ERROR	[0000001098] Numéro de Carte SIM 2 (NSCE) non renseigné.
MLM_TELCO_IS_ERROR	[0000001101] Le Numéro de Carte SIM 2 (NSCE) n'est pas associé à l'abonnement.
MLM_TELCO_IS_ERROR	[0000001102] Le Service doit être positionné sur l'abonnement.
MLM_TELCO_IS_ERROR	[0000001103] Paramètre d'entrée 'Type de carte' incorrect.
MLM_TELCO_IS_ERROR	[0000001104] Le service n'a pas pu être ajouté sur certains(s) abonné(s).
MLM_TELCO_IS_ERROR	[0000001105] Aucun service n'a été positionné.
MLM_TELCO_IS_ERROR	[0000001107] L'abonné possède déjà une carte de renouvellement en cours.

M2M API for PHP - Error codes

Code	Detail
MLM_TELCO_IS_ERROR	[0000001108] L'abonné ne possède pas de carte de ce type active.
MLM_TELCO_IS_ERROR	[0000001122] La valeur du paramètre Type de suspension est incorrecte.
MLM_TELCO_IS_ERROR	[0000001123] L'une des dates passée en paramètre est invalide.
MLM_TELCO_IS_ERROR	[0000001126] Date d'effet incorrecte.
MLM_TELCO_IS_ERROR	[0000001129] Code modification inexistant ou inactif.
MLM_TELCO_IS_ERROR	[0000001132] Demande non applicable.
MLM_TELCO_IS_ERROR	[0000001269] Le N° d'abonnement doit être renseignés
MLM_TELCO_IS_ERROR	[0000001270] L'état de la ligne ne permet pas sa suspension
MLM_TELCO_IS_ERROR	[0000001275] Impossible de suspendre un abonné qui va être résilié
MLM_TELCO_IS_ERROR	[0000001277] Cette option n'est pas valable pour une ligne terminée
MLM_TELCO_IS_ERROR	[0000001279] On ne peut rétablir qu'un abonné déjà suspendu
MLM_TELCO_IS_ERROR	[0000001280] Cette option n'est pas valable pour une ligne terminée
MLM_TELCO_IS_ERROR	[0000001282] Cette option n'est pas disponible pour une ligne principale
MLM_TELCO_IS_ERROR	[0000001284] Cette option n'est pas valable pour une ligne terminée
MLM_TELCO_IS_ERROR	[0000001286] Cette option n'est pas disponible pour une ligne principale
MLM_TELCO_IS_ERROR	[0000001325] Demande non applicable
MLM_TELCO_IS_ERROR	[0000001326] Demande incompatible avec tarif de l'abonné.
MLM_TELCO_IS_ERROR	[0000001327] Service incompatible avec l'option demandée.
MLM_TELCO_IS_ERROR	[0000001328] Date effet obligatoire.
MLM_TELCO_IS_ERROR	[0000001334] Résiliation impossible.
MLM_TELCO_IS_ERROR	[0000001335] Résiliation impossible
MLM_TELCO_IS_ERROR	[0000001357] N° abonnement obligatoire
MLM_TELCO_IS_ERROR	[0000001360] Date d'effet incorrecte.
MLM_TELCO_IS_ERROR	[0000001364] Le N° d'abonnement est obligatoire.
MLM_TELCO_IS_ERROR	[0000001366] la date de début des appels à rechercher est obligatoire.
MLM_TELCO_IS_ERROR	[0000001367] Le N° d'abonnement est inconnu.

M2M API for PHP - Error codes

Code	Detail
MLM_TELCO_IS_ERROR	[0000001368] Le nombre d'itérations demandées dépasse le seuil.
MLM_TELCO_IS_ERROR	[0000001369] Le format de la date de début est incorrect.
MLM_TELCO_IS_ERROR	[0000001370] Le format de la date de fin est incorrect.
MLM_TELCO_IS_ERROR	[0000001371] Le format de l'heure de début est incorrect.
MLM_TELCO_IS_ERROR	[0000001372] Le format de l'heure de fin est incorrect.
MLM_TELCO_IS_ERROR	[0000001375] Date & heure de fin de recherche inférieure à date & heure de début.
MLM_TELCO_IS_ERROR	[0000002000] Action impossible : cet abonné est suspendu/déconnecté.
MLM_TELCO_IS_ERROR	[0000002001] Cet ajout de service porte sur un abonné suspendu/déconnecté.
MLM_TELCO_IS_ERROR	[0000002008] Ne pas renseigner Type numéro et Numéro si N° d'abonnement renseigné
MLM_TELCO_IS_ERROR	[0000002009] Ne pas renseigner N° d'abonnement si Type numéro et Numéro renseignés
MLM_TELCO_IS_ERROR	[0000002010] Numéro doit être renseigné si Type numéro est renseigné
MLM_TELCO_IS_ERROR	[0000002011] Type de numéro doit être renseigné si Numéro est renseigné.
MLM_TELCO_IS_ERROR	[0000002012] Le type de numéro n'est pas valide.
MLM_TELCO_IS_ERROR	[0000002013] Le couple Type et Numéro DISE n'est pas valide
MLM_TELCO_IS_ERROR	[0000002014] N° abonnement ou Type de numéro/Numéro ne sont pas renseignés
MLM_TELCO_IS_ERROR	Opération non réalisée. Veuillez vous rapprocher du service assistance client au {0} pour plus de précisions.
UPDATE_IN_PROGRESS	Une requête de mise à jour du statut de la SIM est déjà en cours pour cet abonnement.
STATUS_NOT_ALLOWED	Le nouveau statut demandé (<requested status>) n'est pas compatible avec le statut courant (<current status>).
UPDATE_SIM_SUSPENDED_NOT_BY_CUSTOMER	Opération interdite car la SIM n'a pas été suspendue par le client.
SIM_NOT_ENOUGH_IN_ACTIVATION	La SIM doit avoir été activée pendant <duration> mois avant de pouvoir passer dans l'état activée pour test.
UPDATE_SIM_STATUS_SLEEPING	La SIM sera activée à la première communication.
MLM_TELCO_IS_CONNECTION_ERROR	Impossible de se connecter à DISE (<error details>).
MLM_INTERNAL_ERROR	Une erreur interne Malima est survenue.

Code	Detail
MLM_NETWORK_STATUS_ERROR	Erreur Requête mal-formée
MLM_NETWORK_STATUS_ERROR	Format des coordonnées incorrect
MLM_NETWORK_STATUS_ERROR	Coordonnées XY ne correspondent pas à des coordonnées en France
MLM_NETWORK_STATUS_ERROR	Interface Network Status indisponible

1. Connectivity

Table A.3. Connectivity errors

Code	Detail
REQ.BACKSIDE_CONNECTION_FAILURE	Failed to establish a backside connection with the back-end service (500) Network connection error between technical platform and WSP server.
GENERAL	Internal Error (500) SOAP message specify an unknown operation name like "operation_C" or SOAP message has a missing '>' for the closing tag soapenv:Envelope.
REQ.SERVICE_NOT_FOUND	The back-end service could not handle the request because it has not found any service corresponding to the received request (500) Service is not available.
IOSW.UNKNOWN_ERROR	An unknown error has occurred during the processing of the request/response (500) « Internal error » of the technical platform.
REQ.AUTHENTICATION_FAILURE	The credentials provided are not valid (401) Unauthorized.
REQ.AUTHORISATION_FAILURE	The client is not authorized to consume the requested service (500).
REQ.SERVICE_AUTHENTICATION_FAILURE	The credentials provided by the service broker when connecting to the back-end service are not valid (500).

M2M API for PHP - Error codes

Code	Detail
REQ.SERVICE_AUTORISATION_FAILURE	The service broker is not authorized to consume the requested back-end service (500).
REQ.SCHEMA_VALIDATION_ERROR	The request sent by the client does not conform to the xml schema of the service (500) Schema Validation Error.
IOSW.XML_THREAT_ERROR	An xml threat was detected by the technical validation platform XML Threat protection, attacks, XDoS.
MESSAGE_TOO_LARGE	The size of the request or response message is too large XML Threat protection - Message too large.
REQ.SERVICE_INTERNAL_ERROR	The service failed to respond to the request because of an internal error Server Internal error.