

# Cellocator+ API Guide



Cellocator Division  
Pointer Telocation Ltd.

Proprietary and Confidential

Version 2.1d

Revised and Updated: February 27, 2020



**POINTER**



# Cellocator+ v2.1 API Guide



## Legal Notices

### IMPORTANT

1. All legal terms and safety and operating instructions should be read thoroughly before the product accompanying this document is installed and operated.
2. This document should be retained for future reference.
3. Attachments, accessories or peripheral devices not supplied or recommended in writing by Pointer Telocation Ltd. may be hazardous and/or may cause damage to the product and should not, in any circumstances, be used or combined with the product.

### General

The product accompanying this document is not designated for and should not be used in life support appliances, devices, machines or other systems of any sort where any malfunction of the product can reasonably be expected to result in injury or death. Customers of Pointer Telocation Ltd. using, integrating, and/or selling the product for use in such applications do so at their own risk and agree to fully indemnify Pointer Telocation Ltd. for any resulting loss or damages.

### Warranty Exceptions and Disclaimers

Pointer Telocation Ltd. shall bear no responsibility and shall have no obligation under the foregoing limited warranty for any damages resulting from normal wear and tear, the cost of obtaining substitute products, or any defect that is (i) discovered by purchaser during the warranty period but purchaser does not notify Pointer Telocation Ltd. until after the end of the warranty period, (ii) caused by any accident, force majeure, misuse, abuse, handling or testing, improper installation or unauthorized repair or modification of the product, (iii) caused by use of any software not supplied by Pointer Telocation Ltd., or by use of the product other than in accordance with its documentation, or (iv) the result of electrostatic discharge, electrical surge, fire, flood or similar causes. Unless otherwise provided in a written agreement between the purchaser and Pointer Telocation Ltd., the purchaser shall be solely responsible for the proper configuration, testing and verification of the product prior to deployment in the field.

POINTER TELOCATION LTD.'S SOLE RESPONSIBILITY AND PURCHASER'S SOLE REMEDY UNDER THIS LIMITED WARRANTY SHALL BE TO REPAIR OR REPLACE THE PRODUCT HARDWARE, SOFTWARE OR SOFTWARE MEDIA (OR IF REPAIR OR REPLACEMENT IS NOT POSSIBLE, OBTAIN A REFUND OF THE PURCHASE PRICE) AS PROVIDED ABOVE. POINTER TELOCATION LTD. EXPRESSLY DISCLAIMS ALL OTHER WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, SATISFACTORY PERFORMANCE AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL POINTER TELOCATION LTD. BE LIABLE FOR ANY INDIRECT, SPECIAL, EXEMPLARY, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOSS OR INTERRUPTION OF USE, DATA, REVENUES OR PROFITS) RESULTING FROM A BREACH OF THIS WARRANTY OR BASED ON ANY OTHER LEGAL THEORY, EVEN IF POINTER TELOCATION LTD. HAS BEEN ADVISED OF THE POSSIBILITY OR LIKELIHOOD OF SUCH DAMAGES.

### Intellectual Property

Copyright in and to this document is owned solely by Pointer Telocation Ltd. Nothing in this document shall be construed as granting you any license to any intellectual property rights subsisting in or related to the subject matter of this document including, without limitation, patents, patent applications, trademarks, copyrights or other intellectual property rights, all of which remain the sole property of Pointer Telocation Ltd. Subject to applicable copyright law, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording or otherwise), or for any purpose, without the express written permission of Pointer Telocation Ltd.

© Copyright 2018. All rights reserved.



## Table of Contents

<b>1</b>	<b>Introduction</b> .....	<b>4</b>
1.1	Revisions .....	4
1.2	Glossary.....	4
<b>2</b>	<b>Getting Started</b> .....	<b>5</b>
2.1	Obtaining an Account ID .....	5
2.2	Authentication Method .....	5
2.3	Post Login .....	5
<b>3</b>	<b>API Actions</b> .....	<b>6</b>
3.1	Login/Logout .....	7
3.1.1	<i>Login</i> .....	7
3.1.2	<i>Logout</i> .....	7
3.2	Inventory Management.....	8
3.2.1	<i>GetGroupList</i> .....	8
3.2.2	<i>GetDeviceList</i> .....	10
3.2.3	<i>GetDeviceInfo</i> .....	11
3.3	PL Updates .....	12
3.3.1	<i>GetPLList</i> .....	12
3.3.2	<i>GetPLInfo</i> .....	13
3.3.3	<i>SetPLUpdate</i> .....	14
3.3.4	<i>CancelPLUpdate</i> .....	15
3.3.5	<i>GetPLUpdateStatus</i> .....	16
3.3.6	<i>GetDevicesForPLUpdate</i> .....	17
3.3.7	<i>GetPLCanLibStatus</i> .....	18
3.3.8	<i>SetPLCanLib</i> .....	19
3.4	Firmware Upgrades .....	20
3.4.1	<i>GetFWList</i> .....	20
3.4.2	<i>GetFWInfo</i> .....	21
3.4.3	<i>SetFWUpgrade</i> .....	22
3.4.4	<i>CancelFWUpgrade</i> .....	22
3.4.5	<i>GetDevicesForFWUpgrade</i> .....	23
3.4.6	<i>GetFWUpgradeStatus</i> .....	25
3.5	Groups Management.....	26
3.5.1	<i>Create Group</i> .....	26
3.5.2	<i>Delete Group</i> .....	28
3.5.3	<i>Assign Units to Group</i> .....	30
3.5.4	<i>Unassign Units from Account</i> .....	32
<b>4</b>	<b>Modem Types</b> .....	<b>34</b>
<b>5</b>	<b>Hardware Types</b> .....	<b>36</b>



## 1 Introduction

This guide is intended to provide customers using the Pointer Cellocator+ with complete instructions for creating, managing and monitoring activities related to PL updates, firmware upgrades, and end units & groups management.

### 1.1 Revisions

Author	Date	Version	Remarks
Alex Marais	July 26, 2019	2.1d	AutoCAN APIs added: <ol style="list-style-type: none"> <li>1. GetPLCanLibStatus</li> <li>2. SetPLCanLib</li> </ol>
Ziv Rotner	February 5, 2019	2.1c	Modified: <ol style="list-style-type: none"> <li>1. Create Group</li> <li>2. Delete Group</li> <li>3. Assign Units to Account &amp; Group</li> <li>4. Unassign Units from Account &amp; Group</li> </ol>
Ziv Rotner	December 15, 2018		APIs added: <ol style="list-style-type: none"> <li>1. Create Group</li> <li>2. Delete Group</li> <li>3. Associate Units</li> </ol>
Ziv Rotner	December 10, 2018		Document restructure
Ziv Rotner	October 2, 2018		New API URL: <a href="https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc">https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc</a>
Technical Writer	November 2013		Initial version

### 1.2 Glossary

Name	Definition
<b>XML</b>	Extensible Markup Language.
<b>External ID</b>	Entity identifier used solely by the client; for example, Task ID, Route ID and so on.
<b>POI</b>	Point of Interest.



## 2 Getting Started

This section describes how to get started with the Pointer Cellocator+ API, including the initial steps you need to take when trying to integrate your system with the Pointer Cellocator+ using the API.

**NOTE:** For this release, the new API uses methods and calls created through this link: <https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc>

### 2.1 Obtaining an Account ID

The account ID value is required to work with Cellocator+. If it was not provided, contact Pointer Customer Support.

### 2.2 Authentication Method

The Pointer Cellocator+ API uses a token-based mechanism to authenticate users and manage their permissions and sessions.

Upon a successful execution of the login function with the relevant account username and password, the function returns a token key.

During its lifetime (1 hour by default), the token key should be used to execute other functions. When there is no activity for a period of 1 hour, or your application has executed the logoff function, the token will expire, and a new token will be generated again when logging in once again.

Note that without the return token key, you will not be able to work with the Pointer Cellocator+ API.

### 2.3 Post Login

Following a successful login and receiving the token, the entire function list is available for you to use, per the permissions granted to your account.



## 3 API Actions

The following API actions can be called using the Cellocator+ API; each of the actions are described in detail in the following sections, including the relevant available parameters, and example calls and their responses.

- ◆ **Login/Logout**
  - Login, page 7
  - Logout, page 7
- ◆ **Inventory Management**
  - Get Group List, page 8
  - Get Device List, page 10
  - Get Device Info, page 11
- ◆ **PL Updates**
  - Get PL List, page 12
  - Get PL Info, page 13
  - Set PL Update, page 14
  - Cancel PL Update, page 15
  - Get PL Update Status, page 16
  - Get Devices for PL Update, page 17
  - Get PL CAN Library Status, page 18
  - Set PL CAN Library, page 19
- ◆ **Firmware Upgrades**
  - Get FW List, page 20
  - Get FW Info, page 21
  - Set FW Upgrade, page 22
  - Cancel FW Upgrade, page 22
  - Get Devices for FW Upgrade, page 23
  - Get FW Upgrade Status, page 25
- ◆ **Group Management**
  - Create Group, page 26
  - Delete Group, page 28
  - Assign Units to Group, page 30



## 3.1 Login/Logout

### 3.1.1 Login

The following table describes how to work with the *Login* function.

<b>Description</b>	The <i>Login</i> function authenticates your username and password and provides you with an authentication token that is used in further API calls. The token is valid for a specific period, as mentioned in section 2.2.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	loginName	String	Mandatory
	Password	String	Mandatory
<b>URL</b>	Server IP/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/Login		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/Login?loginName=MyLogin&password=MyPassword		
<b>Response</b>	User certificate (string).  <pre>&lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="a:string" xmlns:a="http://www.w3.org/2001/XMLSchema"&gt;06E0D5C2-9088-4642-BC39-4A0E5A278B20&lt;/message&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.3679509785053146E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt;</pre>		

### 3.1.2 Logout

The following table describes how to work with the *Logout* function.

<b>Description</b>	The <i>Logout</i> function ends the active session initiated with the <i>Login</i> function and switches the token ID to an expired state. From the moment the <i>Logout</i> function is used, no further API calls will be possible with the token ID.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	loginName	String	Mandatory
	password	String	Mandatory



# Cellocator+ v2.1 API Guide



<b>URL</b>	Server IP/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/Logout
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/Logout?loginName=MyLogin&password=MyPassword
<b>Response</b>	No data.  <pre>&lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.3679509870322893E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt;</pre>

## 3.2 Inventory Management

### 3.2.1 GetGroupList

The following table describes how to work with the *GetGroupList* function.

<b>Description</b>	The <i>GetGroupList</i> function lists all the account groups. Every account has by default a set of groups, where each group is dedicated to a specific hardware type. For example, group name <i>Brom - 12</i> is dedicated to CelloTrack unit types, and <i>Brom - 31</i> is dedicated to Cello-iQ.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	GroupName	String	Optional; single value
<b>URL</b>	Server IP/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetGroupList		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetGroupList?certificate=ZRA59F5D-1EAA-4685-BC50-9102B459017B&AccountId=34		





# Cellocator+ v2.1 API Guide



<b>Response</b>	<p>Group Data list.</p> <pre>&lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPl usModels" xmlns:i="http://www.w3.org/2001/XMLSchema- instance"&gt;   &lt;message i:type="ArrayOfGroupData"&gt;     &lt;GroupData&gt;       &lt;AccountName&gt;PIL&lt;/AccountName&gt;       &lt;CreateDate i:nil="true"/&gt;       &lt;Description i:nil="true"/&gt;       &lt;Id&gt;555&lt;/Id&gt;       &lt;Name&gt;Shagrir - 5&lt;/Name&gt;       &lt;ProviderName i:nil="true"/&gt;       &lt;Type&gt;Group&lt;/Type&gt;       &lt;Author i:nil="true"/&gt;       &lt;DeviceType&gt;Compact Fleet / Security&lt;/DeviceType&gt;       &lt;DevicesCount&gt;0&lt;/DevicesCount&gt;     &lt;/GroupData&gt;     &lt;GroupData&gt;       &lt;AccountName&gt;PIL&lt;/AccountName&gt;       &lt;CreateDate i:nil="true"/&gt;       &lt;Description i:nil="true"/&gt;       &lt;Id&gt;557&lt;/Id&gt;       &lt;Name&gt;Shagrir - 7&lt;/Name&gt;       &lt;ProviderName i:nil="true"/&gt;       &lt;Type&gt;Group&lt;/Type&gt;       &lt;Author i:nil="true"/&gt;       &lt;DeviceType&gt;Compact CAN 8 Sensors&lt;/DeviceType&gt;       &lt;DevicesCount&gt;4&lt;/DevicesCount&gt;     &lt;/GroupData&gt;     ...     ...   &lt;/message&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.368002352128384E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt;</pre>
-----------------	--



# Cellocator+ v2.1 API Guide



## 3.2.2 *GetDeviceList*

The following table describes how to work with the *GetDeviceList* function.

<b>Description</b>	The <i>GetDeviceList</i> function details the entire devices associated with a specific group.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	groupId	Integer	Mandatory
<b>URL</b>	Server IP/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetDeviceList		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetDeviceList?certificate=ZRA59F5D-1EAA-4685-BC50-9102B459017B&AccountId=34&groupId=37995		
<b>Response</b>	<p>Unit Model list.</p> <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="ArrayOfUnitModel"&gt;     &lt;UnitModel&gt;       &lt;AccountName/&gt;       &lt;CMUID&gt;371105&lt;/CMUID&gt;       &lt;FWName&gt;F0440_PCelloHW24_Telit3-Automotive_STM32RCT6_S512_31p_Fleet.csf&lt;/FWName&gt;       &lt;FWStatus&gt;1&lt;/FWStatus&gt;       &lt;GroupInAccount/&gt;       &lt;GroupName&gt;Caterpillar Generators&lt;/GroupName&gt;       &lt;HWDescr&gt;Cello&lt;/HWDescr&gt;       &lt;LastStatusDate&gt;2014-11-20T22:21:05.187&lt;/LastStatusDate&gt;       &lt;ProgStatus&gt;5&lt;/ProgStatus&gt;       &lt;Remark/&gt;     &lt;/UnitModel&gt;     &lt;UnitModel&gt;       &lt;AccountName/&gt;       &lt;CMUID&gt;371139&lt;/CMUID&gt;       &lt;FWName&gt;F0469_PCelloHW24_Telit3-Automotive_STM32RCT6_S512_31x_Fleet.csf&lt;/FWName&gt;       &lt;FWStatus&gt;5&lt;/FWStatus&gt;       &lt;GroupInAccount/&gt;       &lt;GroupName&gt;Caterpillar Generators&lt;/GroupName&gt;       &lt;HWDescr&gt;Cello&lt;/HWDescr&gt;       &lt;LastStatusDate&gt;2014-11-20T21:31:50.877&lt;/LastStatusDate&gt; </pre>		



# Cellocator+ v2.1 API Guide



	<pre>         &lt;ProgStatus&gt;5&lt;/ProgStatus&gt;         &lt;Remark/&gt;       &lt;/UnitModel&gt;       ...       ...     &lt;/message&gt;     &lt;state&gt;success&lt;/state&gt;     &lt;timestamp&gt;6.3680024461299891E+17&lt;/timestamp&gt;   &lt;/ResponseModel&gt; </pre>
--	---

### 3.2.3 GetDeviceInfo

The following table describes how to work with the *GetDeviceInfo* function.

<b>Description</b>	The <i>GetDeviceInfo</i> function provides the entire information available in Cellocator+ regarding a specific unit, starting from its manufacturing date, warranty information, modem, SIM card number, and more.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	cmuId	Integer	Mandatory
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetDeviceInfo		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetDeviceInfo?certificate=ZRA59F5D-1EAA-4685-BC50-9102B459017B&AccountId=34&cmuid=591795		
<b>Response</b>	ExtendedUnitInfoModel data.  <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="ExtendedUnitInfoModel"&gt;     &lt;Account&gt;&amp;lt;..&amp;gt;&lt;/Account&gt;     &lt;AccountId i:nil="true"/&gt;     &lt;AssemblyDate&gt;12/29/2011&lt;/AssemblyDate&gt;     &lt;CMUID&gt;591795&lt;/CMUID&gt;     &lt;CurrentFWVersion&gt;279&lt;/CurrentFWVersion&gt;     &lt;FWStatus&gt;Normal Operation&lt;/FWStatus&gt;     &lt;FirstFWVersion i:nil="true"/&gt;     &lt;FirstPLVersion i:nil="true"/&gt;     &lt;FirstStatusDate&gt;09/07/2012&lt;/FirstStatusDate&gt;     &lt;GroupID&gt;37995&lt;/GroupID&gt;     &lt;GroupName&gt;Caterpillar Generators&lt;/GroupName&gt; </pre>		



# Cellocator+ v2.1 API Guide



	<pre> &lt;Modem&gt;Telit GE864 Automotive&lt;/Modem&gt; &lt;PLName i:nil="true"/&gt; &lt;ProgStatus&gt;Normal Operation&lt;/ProgStatus&gt; &lt;Prov&gt;PIL&lt;/Prov&gt; &lt;ProviderId&gt;34&lt;/ProviderId&gt; &lt;PurchaseOrder&gt;1&lt;/PurchaseOrder&gt; &lt;SIMNumber&gt;425010450957900&lt;/SIMNumber&gt; &lt;SerialId/&gt; &lt;ShipmentDate/&gt; &lt;ShipmentTrackingNumber&gt;0&lt;/ShipmentTrackingNumber&gt; &lt;TesterName&gt;Lab L&lt;/TesterName&gt; &lt;TestingDate&gt;05/01/2013&lt;/TestingDate&gt; &lt;WarrantyExpDate&gt;04/09/2014&lt;/WarrantyExpDate&gt; &lt;/message&gt; &lt;state&gt;success&lt;/state&gt; &lt;timestamp&gt;6.3680024902467469E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt; </pre>
--	---

## 3.3 PL Updates

### 3.3.1 GetPLList

The following table describes how to work with the *GetPLList* function.

<b>Description</b>	The <i>GetPLList</i> function lists all the PL files uploaded or assigned to your account.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	fwVersion (fw id)	Integer	Mandatory
	groupId	Integer	Mandatory
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetPLList		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetPLList?certificate=ZRA59F5D-1EAA-4685-BC50-9102B459017B&AccountId=007&groupId=2190&fwVersion=217		
<b>Response</b>	IDNameModel list (Id: PL ID, Desc: PL Name).  <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt; </pre>		



# Cellocator+ v2.1 API Guide



```

<message i:type="ArrayOfIDNameModel">
  <IDNameModel>
    <Desc>30_31_hw_24_fw_31e_Cello-
AR_V232_05042012_ODED.PL</Desc>
    <id>1493</id>
  </IDNameModel>
  <IDNameModel>
    <Desc>AR_UPDATE_hw_24_fw_30g_30_31_hw_24_fw_31e_Cello-
AR_V232_05042012_ODED.PL</Desc>
    <id>1557</id>
  </IDNameModel>
  <IDNameModel>
    <Desc>CT7700270-
000_30g_hw_24_fw_30g_CelloEdge_Fleet_CelloAR_SPC_Shagrir_Ve
r1.PL</Desc>
    <id>724</id>
  </IDNameModel>
  <IDNameModel>
    <Desc>CT7700271-
000_30g_CELLOAR_30G_16102011_Shagrir_Ver_13112011.PL</Desc>
    <id>1039</id>
  </IDNameModel>
</message>
<state>success</state>
<timestamp>6.3680026205430362E+17</timestamp>
</ResponseModel>

```

### 3.3.2 GetPLInfo

The following table describes how to work with the *GetPLInfo* function.

<b>Description</b>	The <i>GetPLInfo</i> function lists the HW type, PL author, creation date, relevant FW versions and the name of each PL file uploaded or assigned to your account.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	pIVersion	Integer	Mandatory
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetPLInfo		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetPLInfo?certificate=ZRA59F5D-1EAA-4685-BC50-9102B459017B&AccountId=007&pIVersion=2190		
<b>Response</b>	PLInfoModel.		



# Cellocator+ v2.1 API Guide



```

<ResponseModel
xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <message i:type="PLInfoModel">
    <HWDescr>Cello</HWDescr>
    <PLAuthor>Oded</PLAuthor>
    <PLCreateDate>Jun 27 2012 2:48PM
  </PLCreateDate>
    <PLDescr/>
    <PLFWVersions>Ver. 30a to 31e</PLFWVersions>
    <PLName>30_31_hw_24_fw_31e_Cello-AR_V232_05042012_ODED.PL</PLName>
  </message>
  <state>success</state>
  <timestamp>6.3680026543134592E+17</timestamp>
</ResponseModel>

```

### 3.3.3 SetPLUpdate

The following table describes how to work with the *SetPLUpdate* function.

<b>Description</b>	The <i>SetPLUpdate</i> function associates one or more units for specific PL update.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	Certificate	String	Mandatory
	accountId	Integer	Mandatory
	unitIds	String	Mandatory
	plVersion	Integer	
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/SetPLUpdate?		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/SetPLUpdate?certificate=ZRVE2E234-D38D-4540-A7E6-CB676D85C145&accountId=007&plVersion=16181&unitIds=2190		
<b>Response</b>	State: success - Message: No Data State: partialSuccess - Message: string list (failed unitId, reason for fail)  <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="a:ArrayOfstring" xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"&gt;     &lt;a:string&gt;2191695, success&lt;/a:string&gt; </pre>		



# Cellocator+ v2.1 API Guide



	<pre> &lt;/message&gt; &lt;state&gt;success&lt;/state&gt; &lt;timestamp&gt;6.3680051334918976E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt; </pre>
--	---

### 3.3.4 CancelPLUpdate

The following table describes how to work with the *CancelPLUpdate* function.

<b>Description</b>	The <i>CancelPLUpdate</i> function cancel a PL update operation for one or several units.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	unitIds	String	Mandatory
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/CancelPLUpdate?		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/CancelPLUpdate?certificate=ZRVE2E234-D38D-4540-A7E6-CB676D85C145&accountId=007&unitIds=2190		
<b>Response</b>	State: success - Message: No Data State: partialSuccess - Message: string list (failed unitId, reason for fail)  <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="a:ArrayOfstring" xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"&gt;     &lt;a:string&gt;2191695, success&lt;/a:string&gt;   &lt;/message&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.3680051485350042E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt; </pre>		



## 3.3.5 *GetPLUpdateStatus*

The following table describes how to work with the *GetPLUpdateStatus* function.

<b>Description</b>	<p>The <i>GetPLUpdateStatus</i> function displays the latest PL update status for the selected units, from the following list:</p> <ul style="list-style-type: none"> <li>• On Wait</li> <li>• In Process</li> <li>• Successfully Completed</li> <li>• Failed</li> <li>• Normal Operation</li> <li>• Interrupt</li> <li>• Abort</li> </ul>		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	plVersion	Integer	Mandatory
	unitIds	String	Mandatory
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetPLUpdateStatus?		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetPLUpdateStatus?certificate=ZRC10149-B2FA-4212-8498-D3AF50323F00&accountId=13361&plVersion=17155&unitIds=2220565		
<b>Response</b>	<p>String list (unit ID / PL status).</p> <pre>&lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="ArrayOfIDNameModel"&gt;     &lt;IDNameModel&gt;       &lt;Desc&gt;Unit Upgraded&lt;/Desc&gt;       &lt;id&gt;2220565&lt;/id&gt;     &lt;/IDNameModel&gt;   &lt;/message&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.3680041798723136E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt;</pre>		





### 3.3.6 *GetDevicesForPLUpdate*

The following table describes how to work with the *GetDevicesForPLUpdate* function.

<b>Description</b>	The <i>GetDevicesForPLUpdate</i> function lists of all the units that are associated for PL upgrade.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	providerId	Integer	Mandatory
	modemType	Integer	Mandatory
	groupId	Integer	Mandatory List of group ids can be obtained using the GetGroupList function.
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetDevicesForPLUpdate		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetDevicesForPLUpdate?certificate=ZRA59F5D-1EAA-4685-BC50-9102B459017B&AccountId=34&providerId=1&modemType=1493&groupId=37995		
<b>Response</b>	UnitPLModel list.		



## 3.3.7 GetPLCanLibStatus

The following table describes how to work with the *GetPLCanLibStatus* function.

<b>Description</b>	<p>The GetPLCanLibStatus function retrieves the current PL CAN library upgrade status for a selected unit from the following list:</p> <ul style="list-style-type: none"> <li>• Done</li> <li>• In Process</li> <li>• Cancelled</li> <li>• Failed</li> <li>• Last try failed due to unit in PLUpgrade</li> <li>• Last try failed due to unit in FWUpgrade</li> <li>• Last try failed due to no match find in arbitration IDs (or no PLVersion found for CanLibId)</li> </ul>		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	unitId	Integer	Mandatory
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetPLCanLibStatus?		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetPLCanLibStatus?certificate=ZRC10149-B2FA-4212-8498-D3AF50323F00&accountId=13361&plVersion=17155&unitIds=2220565		
<b>Response</b>	<p>UnitCanLib (PL Can Library ID / PL Can Library Progress Sstatus).</p> <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="UnitCanLib"&gt;     &lt;LibId&gt;4738&lt;/LibId&gt;     &lt;ProgStatus&gt;Last try failed due to unit in PLUpgrade&lt;/ProgStatus&gt;   &lt;/message&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.3680041798723136E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt; </pre>		



# Cellocator+ v2.1 API Guide



## 3.3.8 SetPLCanLib

The following table describes how to work with the SetPLCanLib function.

<b>Description</b>	<p>The SetPLCanLib function sets a PL CAN Library to staging for a selected unit so that it can be ready to start an PL CAN upgrade. The latter request will return one of the following results:</p> <ul style="list-style-type: none"> <li>• CAN library PL attached</li> <li>• PL not attached</li> <li>• Failed, No such CAN library</li> <li>• Failed, No match find in arbitration IDs (or no PLVersion found for CanLibId)</li> </ul>			
<b>HTTP Method</b>	GET			
<b>Parameters</b>	Name	Type	Comments	
	certificate	String	Mandatory	
	accountId	Integer	Mandatory	
	unitId	Integer	Mandatory	
	ArbitrationID sCSV	string	List of commas separated arbitration id's	Only one of this 2 parameters should be defined!
	CanLibId	Integer	Libray Id for the unit	
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/SetP LCanLib?			
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/Cellocator PlusPublicAPI.svc/XMLService/SetPLCanLib?certificate=ZRC10149-B2FA- 4212-8498- D3AF50323F00&accountId=13361&plVersion=17155&unitIds=2220565			
<b>Response</b>	<p>UnitCanStartLibUpgrade (PL Can Library Start Upgrade Result).</p> <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPl usModels" xmlns:i="http://www.w3.org/2001/XMLSchema- instance"&gt;   &lt;message i:type="UnitCanStartLibUpgrade"&gt;     &lt;LibId&gt;4738&lt;/LibId&gt;     &lt;ProgStatus&gt;Last try failed due to unit in PLUgrade&lt;/ProgStatus&gt;   &lt;/message&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.3680041798723136E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt; </pre>			



## 3.4 Firmware Upgrades

### 3.4.1 GetFWList

The following table describes how to work with the *GetFWList* function.

<b>Description</b>	The <i>GetFWList</i> function displays a list of firmware versions that can be used in conjunction with a specific group (unit hardware type), and specific modem.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	Certificate	String	Mandatory
	AccountId	Integer	Mandatory
	groupId	Integer	Mandatory Use the <i>GetGroupList</i> function to retrieve your account groups list.
	modemType	Integer	Mandatory A list of supported modem types is available in the Modem Types section.
<b>URL</b>	Server IP/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetFWList		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetFWList?certificate=ZRA59F5D-1EAA-4685-BC50-9102B459017B&AccountId=007&groupId=2190&modemType=6		
<b>Response</b>	<p>IDNameModel list (Id: FW Id, Desc: FW Name).</p> <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="ArrayOfIDNameModel"&gt;     &lt;IDNameModel&gt;       &lt;Desc&gt;F0001_PCelloHW24_Telit3-Automotive_STM32RCT6_S512_30g_Ar_RC3236.csf&lt;/Desc&gt;       &lt;id&gt;217&lt;/id&gt;     &lt;/IDNameModel&gt;     &lt;IDNameModel&gt;       &lt;Desc&gt;F0001_PCelloHW24_Telit3-Automotive_STM32RCT6_S512_30j_Fleet_RC3727.csf&lt;/Desc&gt;       &lt;id&gt;178&lt;/id&gt;     &lt;/IDNameModel&gt;     &lt;IDNameModel&gt;       &lt;Desc&gt;F0001_PCelloHW24_Telit3-Automotive_STM32RCT6_S512_31k_Ar_RC7660.csf&lt;/Desc&gt;       &lt;id&gt;232&lt;/id&gt;     &lt;/IDNameModel&gt;     &lt;IDNameModel&gt; </pre>		



# Cellocator+ v2.1 API Guide



	<pre>                 &lt;Desc&gt;F0002_PCelloHW24_Telit3- Automotive_STM32RCT6_S512_31k_Ar_RC7660.csf&lt;/Desc&gt;                 &lt;id&gt;249&lt;/id&gt;             &lt;/IDNameModel&gt;         &lt;/message&gt;         &lt;state&gt;success&lt;/state&gt;         &lt;timestamp&gt;6.3680025513395136E+17&lt;/timestamp&gt;     &lt;/ResponseModel&gt; </pre>
--	---

## 3.4.2 GetFWInfo

The following table describes how to work with the *GetFWInfo* function.

<b>Description</b>	The <i>GetFWInfo</i> function provides detailed information on a specific FW version, including the HW type it is designed for, modem type and other information.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	Certificate	String	Mandatory
	AccountId	Integer	Mandatory
	fwVersion	Integer	Mandatory List of relevant firmware versions can be obtained using the <i>GetFWList</i> function.
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetFWInfo		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetFWInfo?certificate=ZRA59F5D-1EAA-4685-BC50-9102B459017B&AccountId=007&fwVersion=217		
<b>Response</b>	<p>FWInfoModel.</p> <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;     &lt;message i:type="FWInfoModel"&gt;         &lt;CreateDate&gt;07/02/2012&lt;/CreateDate&gt;         &lt;Description/&gt;         &lt;HWType&gt;Cello&lt;/HWType&gt;         &lt;ModemType&gt;Telit GE864 Automotive&lt;/ModemType&gt;         &lt;Name&gt;F0001_PCelloHW24_Telit3- Automotive_STM32RCT6_S512_30g_Ar_RC3236.csf&lt;/Name&gt;         &lt;Version&gt;30g&lt;/Version&gt;     &lt;/message&gt;     &lt;state&gt;success&lt;/state&gt;     &lt;timestamp&gt;6.3680025880111987E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt; </pre>		



### 3.4.3 SetFWUpgrade

The following table describes how to work with the *SetFWUpgrade* function.

<b>Description</b>	The <i>SetFWUpgrade</i> function associate one or several units for firmware upgrade.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	unitIds	String	Mandatory
	fwVersion	Integer	Mandatory List of relevant firmware versions can be obtained using the <i>GetFWList</i> function.
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/SetFWUpgrade?		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/SetFWUpgrade?certificate=FDE2E234-D38D-4540-A7E6-CB676D85C145&accountId=007&fwVersion=709&unitIds=2190		
<b>Response</b>	State: success - message: No Data State: partialSuccess - message: string list (failed unitId, reason for fail)  <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="a:ArrayOfstring" xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"&gt;     &lt;a:string&gt;2191695, success&lt;/a:string&gt;   &lt;/message&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.3680051748857843E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt; </pre>		

### 3.4.4 CancelFWUpgrade

The following table describes how to work with the *CancelFWUpgrade* function.

<b>Description</b>	The <i>CancelFWUpgrade</i> function cancel the firmware upgrade process for one or several units.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>



## Cellocator+ v2.1 API Guide



	certificate	String	Mandatory
	accountId	Integer	Mandatory
	unitIds	String	Mandatory
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/CancelFWUpgrade?		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/CancelFWUpgrade?certificate=ZRVE234-D38D-4540-A7E6-CB676D85C145&accountId=007&unitIds=2190		
<b>Response</b>	<p>State: success - message: No Data            State: partialSuccess - message: string list (failed unitId, reason for fail)</p> <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="a:ArrayOfstring" xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"&gt;     &lt;a:string&gt;2191695, success&lt;/a:string&gt;   &lt;/message&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.3680051816359155E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt;           </pre>		

### 3.4.5 *GetDevicesForFWUpgrade*

The following table describes how to work with the *GetDevicesForFWUpgrade* function.

<b>Description</b>	The <i>GetDevicesForFWUpgrade</i> function generates a list of units that can be upgraded to firmware version.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	providerId	Integer	Optional
	fwVersion	Integer	Optional
	groupId	Integer	Mandatory



# Cellocator+ v2.1 API Guide



<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetDevicesForFWUpgrade?
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetDevicesForFWUpgrade?certificate=ZRV2E234-D38D-4540-A7E6-CB676D85C145&accountId=13350&providerId=5186&groupId=204374&fwVersion=1113
<b>Response</b>	<p>UnitFWModel list.</p> <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="ArrayOfUnitFWModel"&gt;     &lt;UnitFWModel&gt;       &lt;CMUID&gt;1463881&lt;/CMUID&gt;       &lt;CurrentFW&gt;F0830_PCelloHW107_TelitGE910V3_STM32RDT6_S512_44i_CR300B2G_Unified.csf&lt;/CurrentFW&gt;       &lt;DevType&gt;CR300B_2G&lt;/DevType&gt;       &lt;Duration&gt;375&lt;/Duration&gt;       &lt;FWStatus&gt;5&lt;/FWStatus&gt;       &lt;LastStatusDate&gt;2018-12-10T11:04:55.993&lt;/LastStatusDate&gt;       &lt;ModemDescr&gt;Default Modem&lt;/ModemDescr&gt;       &lt;NewFW/&gt;       &lt;reqtime i:nil="true"/&gt;       &lt;updatetime i:nil="true"/&gt;     &lt;/UnitFWModel&gt;     &lt;UnitFWModel&gt;       &lt;CMUID&gt;1475511&lt;/CMUID&gt;       &lt;CurrentFW&gt;F07EA_PCelloHW107_TelitGE910V3_STM32RDT6_S512_44d_CR300B2G_Unified.csf&lt;/CurrentFW&gt;       &lt;DevType&gt;CR300B_2G&lt;/DevType&gt;       &lt;Duration i:nil="true"/&gt;       &lt;FWStatus&gt;5&lt;/FWStatus&gt;       &lt;LastStatusDate&gt;2018-12-10T10:10:54.25&lt;/LastStatusDate&gt;       &lt;ModemDescr&gt;Default Modem&lt;/ModemDescr&gt;       &lt;NewFW/&gt;       &lt;reqtime i:nil="true"/&gt;       &lt;updatetime i:nil="true"/&gt;     &lt;/UnitFWModel&gt;   &lt;/message&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.3680041229150541E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt; </pre>





# Cellocator+ v2.1 API Guide



## 3.4.6 *GetFWUpgradeStatus*

The following table describes how to work with the *GetFWUpgradeStatus* function.

<b>Description</b>	The <i>GetFWUpgradeStatus</i> function lists of all of the units associated for firmware upgrade, and their on-going status.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	fwVersion	Integer	
	unitIds	Integer	Mandatory
<b>URL</b>	Server IP/ CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetFWUpgradeStatus?		
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/GetFWUpgradeStatus?certificate=ZRV1A4E3-01DC-4572-9908-AA9DD1F45326&accountId=67&fwVersion=1018&unitIds=3007296		
<b>Response</b>	String list (unit ID / FW status).  <pre> &lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="ArrayOfIDNameModel"&gt;     &lt;IDNameModel&gt;       &lt;Desc&gt;Unit On wait&lt;/Desc&gt;       &lt;id&gt;3007296&lt;/id&gt;     &lt;/IDNameModel&gt;   &lt;/message&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.3680042237458317E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt; </pre>		



## 3.5 Groups Management

### 3.5.1 Create Group

The following table describes how to work with the *CreateGroup* function.

<b>Description</b>	The <i>CreateGroup</i> function creates a new group (according to specific hardware type), where units from the same hardware type can be associated to.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	groupName	String	Mandatory, up to 50 characters.
	accountId	Integer	Mandatory
	deviceType	Integer	Mandatory
	Description	String	Optional, up to 100 characters.
	author	String	Optional, up to 50 characters.
<b>Usage</b>	<ol style="list-style-type: none"> <li>1. The accountId should match the accountId of the login certificate, and the input parameter (accountId) should be the same or the accountId should be child account of the certificate (one level down not including grandchild), which means that parent account can create groups to its account and to its children's accounts.</li> <li>2. In case of an attempt to create a group in different account than the certificate, the following error will be returned. <code>&lt;state&gt;insufficientPermissions&lt;/state&gt;</code></li> <li>3. In case the the certificate is expired, the following error will be displayed. <code>&lt;state&gt;certificateExpired&lt;/state&gt;</code></li> <li>4. The groupName has to be unique in an account-level. In case the groupName already exists, the following error will be returned. <code>&lt;message&gt;Group with name 'groupName' already exists&lt;/message&gt;</code> <code>&lt;state&gt;unsuccess&lt;/state&gt;</code></li> <li>5. The deviceType value must be from the list of existing device types (HW types), otherwise the following error will be returned: <code>&lt;message&gt;Non-existing HWType = {deviceType} is provided&lt;/message&gt;</code> <code>&lt;state&gt;unsuccess&lt;/state&gt;</code></li> <li>6. In case some of the mandatory parameters is missing, the following error will be returned. <code>&lt;message&gt;Mandatory parameter '{paramName}' is missing&lt;/message&gt;</code></li> </ol>		



## Cellocator+ v2.1 API Guide



	<pre>&lt;state&gt;invalidRequest&lt;/state&gt;</pre> <p>7. In case a parameter exceeds the maximum allowed string size, the following error will be returned.</p> <pre>&lt;message&gt;Parameter '{paramName}' must be between 0 and 100 characters &lt;/message&gt;</pre> <pre>&lt;state&gt;invalidRequest&lt;/state&gt;</pre>
<b>URL</b>	Server IP/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/ XMLService/AddGroup?
<b>URL Example</b>	<pre>https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/AddGroup?certificate={{Certificate}}&amp;groupName=Test1&amp;accountId=0&amp;deviceType=20&amp;description=Test Group Description&amp;author=Test</pre>
<b>Response</b>	Returns the groupId of the new group:  <pre>&lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="a:int" xmlns:a="http://www.w3.org/2001/XMLSchema"&gt;98085&lt;/message&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.3680979892223578E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt;</pre>



## 3.5.2 Delete Group

The following table describes how to work with the *DeleteGroup* function.

<b>Description</b>	The <i>DeleteGroup</i> function delete one or several groups. If there are units associated with these groups, they will be associated back to the relevant system groups.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountId	Integer	Mandatory
	groupIds	String	Mandatory
<b>Usage</b>	<ol style="list-style-type: none"> <li>The accountId should match the accountId of the login certificate, and the input parameter (accountId) should be the same or the accountId should be child account of the certificate (one level down not including grandchild), which means that parent account can create groups to its account and to its children's accounts.</li> <li>In case some user from one account try to delete group from another account in which does not have privileges (according to point 1), following error will be returned: &lt;state&gt;insufficientPermissions&lt;/state&gt;</li> <li>In case the logged in certificate is expired, below error will be returned and the user will need to get a new certificate: &lt;state&gt;certificateExpired&lt;/state&gt;</li> <li>All group Ids must be of an existing group, otherwise the following error will be returned for the specific groupId: &lt;message&gt;&lt;a:string&gt;{groupId} - Group does not exists&lt;/a:string&gt;&lt;/message&gt; &lt;state&gt;unsuccess&lt;/state&gt;</li> <li>All group Ids must belong to the provided accountId, otherwise the following error will be returned for the specific groupId: &lt;message&gt;&lt;a:string&gt;{groupId} - Group does not belong to provided AccountId&lt;/a:string&gt;&lt;/message&gt; &lt;state&gt;unsuccess&lt;/state&gt;</li> <li>Group can be deleted only if there are no assigned units in it. Otherwise, the following error message will be returned. &lt;message &gt;&lt;a:string&gt;{groupId} - Group is in use&lt;/a:string&gt;&lt;/message&gt; &lt;state&gt;unsuccess&lt;/state&gt;</li> <li>In case some of the mandatory parameters are missing the following error will be returned. &lt;message&gt;Mandatory parameter '{paramName}' is missing&lt;/message&gt; &lt;state&gt;invalidRequest&lt;/state&gt;</li> </ol>		



## Cellocator+ v2.1 API Guide



	<p>8. In case multiple group ids are provided for deletion, the 'state' field in the response can be</p> <ul style="list-style-type: none"><li>- Success – all groups from groupIds have been deleted successfully</li><li>- Unsuccess – no groups have been deleted</li><li>- Partial Success – some of the groups have been deleted and some of them haven't.</li></ul>
<b>URL</b>	Server IP/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/ XMLService/DeleteGroups?
<b>URL Example</b>	https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/ XMLService/DeleteGroups?certificate={{Certificate}}&accountId=0&groupIds=98088
<b>Response</b>	Returns a deleted successfully message:  <pre>&lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt; &lt;message i:type="a:ArrayOfstring" xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"&gt; &lt;a:string&gt;98098 - Successfully deleted&lt;/a:string&gt; &lt;/message&gt; &lt;state&gt;success&lt;/state&gt; &lt;timestamp&gt;6.3680979892223578E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt;</pre>



### 3.5.3 Assign Units to Group

The following table describes how to work with the *AssignUnits* function.

<b>Description</b>	<p>The Assign Units to Group function allows migration of units from one group to another, as well as from one account id to another (parent to sub-account). Units can be assigned from one of the account’s default groups to a custom group created (using the CreateGroup function), and also from a custom group, back to the account default group. The destination group should be with the same device type configuration.</p>		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	accountIdTo	Integer	Mandatory
	groupIdTo	Integer	Mandatory
	UnitIds	String	Mandatory
<b>Usage</b>	<ol style="list-style-type: none"> <li>Only units that belong to the current logged-in user account id can be assigned to other account or group, otherwise following error will be returned  <pre>&lt;message&gt;   &lt;a:string&gt;Unit {unitId} does not belong to current logged in user   accountId &lt;/a:string&gt; &lt;/message&gt; &lt;state&gt;unsuccess&lt;/state&gt;</pre> </li> <li>The account from the logged in certificate and the input parameter accountIdTo (where the unit should be assigned) should be the same or the accountIdTo should be child account of the certificate (one level down not including grandchild), which means that parent account can assign units to groups from its account or groups from its children’s accounts.</li> <li>In case some user from one account try to assign unit to group from other account in which does not have privileges (according to point 2), following error will be returned  <pre>&lt;state&gt;insufficientPermissions&lt;/state&gt;</pre> </li> <li>In case the logged in certificate is expired, below error will be returned and the user will need to get a new certificate:  <pre>&lt;state&gt;certificateExpired&lt;/state&gt;</pre> </li> <li>GroupId must be of an existing group, otherwise the following error will be returned:  <pre>&lt;message&gt;   &lt;a:string&gt;Group {groupId} does not exists&lt;/a:string&gt; &lt;/message&gt; &lt;state&gt;unsuccess&lt;/state&gt;</pre> </li> <li>GroupId must belong to the provided accountIdTo, otherwise the following error will be returned:  <pre>&lt;message&gt;   &lt;a:string&gt;Group {groupId} does not belong to AccountId {AccountIdTo} &lt;/a:string&gt; &lt;/message&gt; &lt;state&gt;unsuccess&lt;/state&gt;</pre> </li> </ol>		



## Cellocator+ v2.1 API Guide



	<p>7. All unit Ids must be of an existing unit, otherwise the following error will be returned for the specific unitId:  <code>&lt;message&gt;  &lt;a:string&gt;Unit {unitId} does not exists&lt;/a:string&gt;  &lt;/message&gt;  &lt;state&gt;unsuccess&lt;/state&gt;</code></p> <p>8. Only units that have the same HW type as the group HW type can be assigned to that group, otherwise the following error will be returned for the specific unit  <code>&lt;message&gt;  &lt;a:string&gt;Unit {unitId} with HWType = {unitHWType} cannot be assigned to group with HWType = {groupHWType}&lt;/a:string&gt;  &lt;/message&gt;  &lt;state&gt;unsuccess&lt;/state&gt;</code></p> <p>9. In case the user try to assign the unit to group that the unit already belong to, the following error will be returned  <code>&lt;message&gt;  &lt;a:string&gt;Unit {unitId} is already assigned to group {groupIdTo}&lt;/a:string&gt;  &lt;/message&gt;  &lt;state&gt;unsuccess&lt;/state&gt;</code></p> <p>10. In case some of the mandatory parameters is missing, following error will be returned  <code>&lt;message&gt;Mandatory parameter '{paramName}' is missing&lt;/message&gt;  &lt;state&gt;invalidRequest&lt;/state&gt;</code></p> <p>11. In case multiple unit ids are provided for assignment, the 'state' field in the response can be</p> <ul style="list-style-type: none"> <li>- Success – all units from unitIds have been assigned successfully</li> <li>- Unsuccess – no units have been assigned</li> <li>- PartialSuccess – some of the units have been assigned and some of them haven't</li> </ul>
<b>URL</b>	<p>Server  IP/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/  XMLService/AssignUnitsToAccountAndGroup?</p>
<b>URL Example</b>	<p><a href="https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/AssignUnitsToAccountAndGroup?certificate={{Certificate}}&amp;accountId=11 &amp;groupIdTo=98072&amp;unitIds=4545">https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/AssignUnitsToAccountAndGroup?certificate={{Certificate}}&amp;accountId=11 &amp;groupIdTo=98072&amp;unitIds=4545</a></p>
<b>Response</b>	<p>Returns the appropriate assignment status message for each unit</p> <pre>&lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusModels" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt; &lt;message i:type="a:ArrayOfstring" xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"&gt; &lt;a:string&gt;Unit 4545 successfully assigned&lt;/a:string&gt; &lt;/message&gt; &lt;state&gt;success&lt;/state&gt; &lt;timestamp&gt;6.3680989502145472E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt;</pre>



### 3.5.4 Unassign Units from Account

The following table describes how to work with the *UnassignUnits* function.

<b>Description</b>	The <i>UnassignUnits</i> function rollback unit(s) from the child to the parent account.		
<b>HTTP Method</b>	GET		
<b>Parameters</b>	<b>Name</b>	<b>Type</b>	<b>Comments</b>
	certificate	String	Mandatory
	UnitIds	String	Mandatory
<b>Usage</b>	<ol style="list-style-type: none"> <li>Parent account can unassign unit(s) from his child account, given the following conditions. <ol style="list-style-type: none"> <li>The units are associated to one of his sub-accounts (one level down), not including grandchild account.</li> <li>The units aren't already assigned to a 3<sup>rd</sup> level grandchild.</li> <li>The <i>Allow Parent Control</i> parameter of the child account (where the unit is assigned) is enabled.</li> </ol> </li> <li>In case another account tries to unassign a unit from an account in which does not have privileges (according to point 1), the following error will be returned.  <pre>&lt;message&gt; &lt;a:string&gt; Current logged-in user cannot unassign the Unit {unitId} &lt;/a:string&gt; &lt;/message&gt; &lt;state&gt;unsuccess&lt;/state&gt;</pre> </li> <li>In case the child account has already assigned the unit to 3<sup>rd</sup> level grandchild, the same error message from point 2 will be displayed.</li> <li>In case the <i>Allow Parent Control</i> of the child account is not enabled, the following error message will be displayed:  <pre>&lt;message&gt; &lt;a:string&gt;AllowParentControl is not enabled for the account of the Unit {unitId} &lt;/a:string&gt; &lt;/message&gt; &lt;state&gt;unsuccess&lt;/state&gt;</pre> </li> <li>UnitId must be of an existing unit, otherwise the following error will be returned:  <pre>&lt;message&gt; &lt;a:string&gt;Unit {groupId} does not exists&lt;/a:string&gt; &lt;/message&gt; &lt;state&gt;unsuccess&lt;/state&gt;</pre> </li> <li>In case the logged in certificate is expired, the error below will be returned, and the user will need to get a new certificate:  <pre>&lt;state&gt;certificateExpired&lt;/state&gt;</pre> </li> <li>In case some of the mandatory parameters is missing, following error will be returned</li> </ol>		





## Cellocator+ v2.1 API Guide



	<pre>&lt;message&gt;Mandatory parameter '{paramName}' is missing&lt;/message&gt; &lt;state&gt;invalidRequest&lt;/state&gt;</pre>
<b>URL</b>	Server IP/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/ XMLService/UnassignUnits?
<b>URL Example</b>	<a href="https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/XMLService/UnassignUnits?certificate={{Certificate}}&amp;unitIds=190004">https://cellocatorplus.pointer.com/CellocatorPlusPublicAPI/Services/CellocatorPlusPublicAPI.svc/ XMLService/UnassignUnits?certificate={{Certificate}}&amp;unitIds=190004</a>
<b>Response</b>	Returns the appropriate unassignment status message for each unit  <pre>&lt;ResponseModel xmlns="http://schemas.datacontract.org/2004/07/CellocatorPlusMode ls" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;message i:type="a:ArrayOfstring" xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays "&gt;     &lt;a:string&gt;Unit 190004 successfully unassigned from account 7040&lt;/a:string&gt;   &lt;/message&gt;   &lt;state&gt;success&lt;/state&gt;   &lt;timestamp&gt;6.3680989502145472E+17&lt;/timestamp&gt; &lt;/ResponseModel&gt;</pre>



## Cellocator+ v2.1 API Guide



### 4 Modem Types

Modem Type	Modem Description	Modem Version Text	Hardware Type	Unit Modem Type
0	Default Modem	NULL	0	0
1	Enfora Enabler II - G	0.7.8	4	1
2	Telit GE864 - No Mute Support	7.02.004	-1	2
3	Telit GE864 - Mute Support	7.02.004	4	3
5	Enfora III (23)	1.1.1	23	5
6	Telit GE864 Automotive	7.0.30.0	24	6
7	Cinterion BGS3	1.02	25	7
8	Simple Modem (ATD dial up)	Astro\Tetra	-1	8
9	CELLOTRACK T 2G GE910	40205_A00	18	0
11	CELLOTRACK T 3G HE910	40105_B00	18	2
12	CELLOTRACK T 3G UE910	40115_A00	18	3
13	CELLOTRACK T POWER 2G GE910	40200_A00	19	0
15	CELLOTRACK T POWER 3G HE910	40100_B00	19	2
16	CELLOTRACK T POWER 3G UE910	40110_A00	19	3
17	Cello CAN IQ GE910-V3	214TR_A03	20	2
18	Cello-CANiQ (NA) UE910 NAR	16.00.303	20	0
19	Cello-CANiQ (EU) UE910 EUR	12.00.404	20	1
20	GE910 QUAD V3	GE910 QUAD V3	14	2
21	Cello F/R GE910-V3	212TR_A03	24	2
22	COMPACT GE864	TOB	28	6
23	CR200_GE864 V2 ARCAM	180TA_A06	29	6
24	CR200B_GE864 V2 ARCAM	181TA_A06	30	6
25	Cello F/R UE910-NAR	215TR_A02	24	0
26	CELLO IQ GE864 V2	174TR_A02	31	6
27	CR200S4	200TR_B03	15	6
28	CR200B S4	201TR_B03	16	6
29	CR300 GE864	204TR_A00	1	7
30	CR300B GE864	203TR_A00	11	7
31	Nano Cinterion	Nano Cinterion modem	26	0
32	Reserved		1	0
33	No Modem		2	0
34	No Modem	Tetra/Astro external modem	3	1
35	Simple Modem- AT		3	2
36	Simple Modem- No AT		3	3
37	Sony/Erickson GR47		4	0



## Cellocator+ v2.1 API Guide



Modem Type	Modem Description	Modem Version Text	Hardware Type	Unit Modem Type
38	Telit GE864	Old retrofit board (obsolete)	4	2
39	Sony/Erickson GR47		5	0
40	Enfora Enabler II-G		5	1
41	Telit GE864	Old retrofit board (obsolete)	5	2
42	Telit GE864, mute support (HW 5)		5	3
43	Sony/Erickson GR47		7	0
44	Enfora Enabler II-G		7	1
45	Telit GE864	Old retrofit board (obsolete)	7	2
46	Telit GE864	Mute support	7	3
47	Telit GE864	Automotive	7	6
50	Reserved		10	0
51	Enfora 3		10	5
52	Reserved		11	0
53	Sony/Erickson GR47		12	0
54	Enfora III (12)		12	5
55	Reserved		13	0
56	Reserved		14	0
57	CT LL	1000_B03	47	0
58	Reserved		16	0
59	Sony/Erickson GR47		17	0
60	Telit GE864	Automotive	17	6
61	Reserved		21	0
62	Telit GE864	Non-Automotive	21	7
63	Sony/Erickson GR47		22	0
64	Enfora III (22)		22	5
65	Sony/Erickson GR47		23	0
66	Sony/Erickson GR47		25	0
75	Reserved		28	0
76	Telit GE864, mute support		28	3
77	Reserved		29	0
78	Reserved		30	0
79	Reserved		31	0
80	GE910 QUAD V3 (not in use)	CELLO IQ GE910-V3	14	0



## 5 Hardware Types

Hardware Type	Description
0	Default Hardwares
1	CR300
2	CFEdont use
3	Olympic-03GM
4	Compact Fleet
5	Compact Fleet / Security
6	Compact-01GR Low Cost
7	Compact CAN 8 Sensors
8	Compact-01GR Security
9	Cello-CANiQ (NA)
10	CELL POWER OUTPUT
11	CR300B
12	CelloTrack
13	Compact Basic
14	Cello IQ (ST)
15	Cellotrack LL
16	CR200_S4
17	Compact CAN 25 Sensors
18	Cellotrack3G
19	CelloTrackPower3g
20	CelloCan-IQ
21	CR300E
22	Compact EOB Fleet / Security
23	CelloTrack Power / Lighter
24	Cello
25	CelloCinterion
26	Nano
27	CELLO Big Flash Telit
28	Compact TOB Fleet / Security
29	CR200
30	CR200b
31	Cello-IQ
38	Cello-D
41	Cello-CANiQ (EU)
43	CR300 3G NAD
47	Cellotrack 10Y (LV100)
58	Nano (Modem EHS5-E)



## Cellocator+ v2.1 API Guide



Hardware Type	Description
73	Cello-CANiQ (2G)
75	CR300 3G EUD
77	Cello-IQ GNSS, Cello-Gen3
90	Nano (Modem EHS5-US)
100	Hardware100
105	CANiQ-Siren-NAR
107	CR300B 2G
116	CAN-IQ CarSharing
122	Nano20 EHS6-A
136	Nano10 2G
137	CANiQ-Siren-EUR
139	CR300B 2G GNSS
143	CelloTrack SOLAR
168	Nano10 EHS6-A
169	Cello-CANiQ V3
180	CELLO CAN IQ 3G NA TLV
201	Cello-CANiQ (NA) - Aux
212	CELLO CDMA Verizon
233	Cello-CANiQ (EU) - Aux
244	CELLO CDMA Sprint
514	CFE BT
515	CFE Basic
516	CFE I/O
517	CFE premium
65535	DFD