

Large Account

Web Service Interface Description

v 3.6

Doc Revision 4 | 9 November 2026

This document is delivered subject to the following conditions and restrictions:

- This document contains proprietary information belonging to Cellact Ltd. Such information is supplied solely for the purpose of assisting explicitly and appropriately authorized users of Cellact products.
- No part of this document may be used for any other purpose, disclosed to any person or firm or reproduced by any means, electronic or mechanical, without the express prior written permission of Cellact Ltd.
- The software described in this document is furnished under a license. The Cellact software may be used or copied only in accordance with the terms of that agreement.
- The information contained in this document, specifications on which the document is based and the examples contained in it are all subject to change without notice.
- Corporate and individual names and data used in examples herein are fictitious unless otherwise noted. The text and the graphics included in this document are for the purposes of instruction, illustration and reference only.
- The names of other companies, product brands, and services mentioned in this document are trademarks or registered trademarks of their respective holders.

Copyright ©2016, Cellact Ltd. All rights reserved.

Purpose

This document specifies web services that allow integrating Large-Account Service in client's applications

Audience

This guide is intended for developers of Applications that facilitate Web Services provided by the Large-Account

Contents

1 Introduction 1

2 Before you start 1

2.1 Preconditions 1

2.2 The Method and End-Points on LA 2

2.3 End-Points on CP for Asynchronous Messages 2

2.4 Quick Glance at the Service 2

2.5 Authorization to use the Service 3

2.6 Description of attributes of Service 4

3 Common Principles 5

3.1 Structure of Request 5

3.2 Response 6

3.2.1 The Structure of Response 6

3.2.2 Acceptance of the request 8

3.3 Handling Asynchronous Messages 8

3.3.1 Async-Anchor provided by CP Client 9

3.3.2 Async-Anchor provided by LA/WS 10

3.3.3 Summary: Mapping of Async-Anchor Names 10

3.4 Reports - Tracking the Progress of Delivery 11

3.4.1 Confirmation of Delivery 11

3.4.2 Reporting by means of HTTP 13

4 "Send" group of Web Services 15

4.1 Send:sendtextmt 16

4.1.1 Request 16

4.1.2 Response 21

4.1.3 Confirmation of Delivery 21

4.1.4 Errors 22

4.2 Send:sendivr 23

4.2.1 Request 23

4.2.2 Response 25

4.2.3 Confirmation of Delivery 25

4.2.4 Rejection of the request 25

4.2.5 Errors 26

4.3 Send:sendivrmt 26

4.3.1 Request 27

4.3.2 Response 27

4.3.3 Confirmation of Delivery 27

4.3.4 Errors 27

5 Additional Web Services 28

5.1 SendText 28

5.1.1 Request 28

5.1.2 Response 29

5.1.3 Errors 30

5.2 SendMail 30

5.2.1 Request 30

5.2.2 Response 34

5.2.3 Errors 34

6 Appendix 35

6.1 Format of the Destination Number 35

6.2 Message Delivery Status (REASON) Codes 35

6.3 Types and XML Structures 38

6.3.1 Enums 38

6.3.2 Credentials 38

6.3.3 Result 38

6.3.4 SendRequest 39

6.3.5 SendRequestMail 40

List of Tables

Table 1 Endpoints of LA 2

Table 2 Credentials 6

Table 3 Structure of the Result 7

Table 4 Attributes of Result 8

Table 5 SMS Delivery Status 12

Table 6 SMS Delivery Timeline and the corresponding reports 12

Table 7 Attribute of Confirmation Report (HTTP) 13

Table 8 Parameters of Send 17

Table 9 Parameters of SendText 28

Table 10 Parameters of SendMail 31

Table 11 Interpretation of the “Reason” Code 36

List of Figures

Figure 1 LA Server in Network 1

Figure 2 Top-Level Structure of LA/WS Request and Reply 5

Figure 3 Delivery Reports are Async messages 9

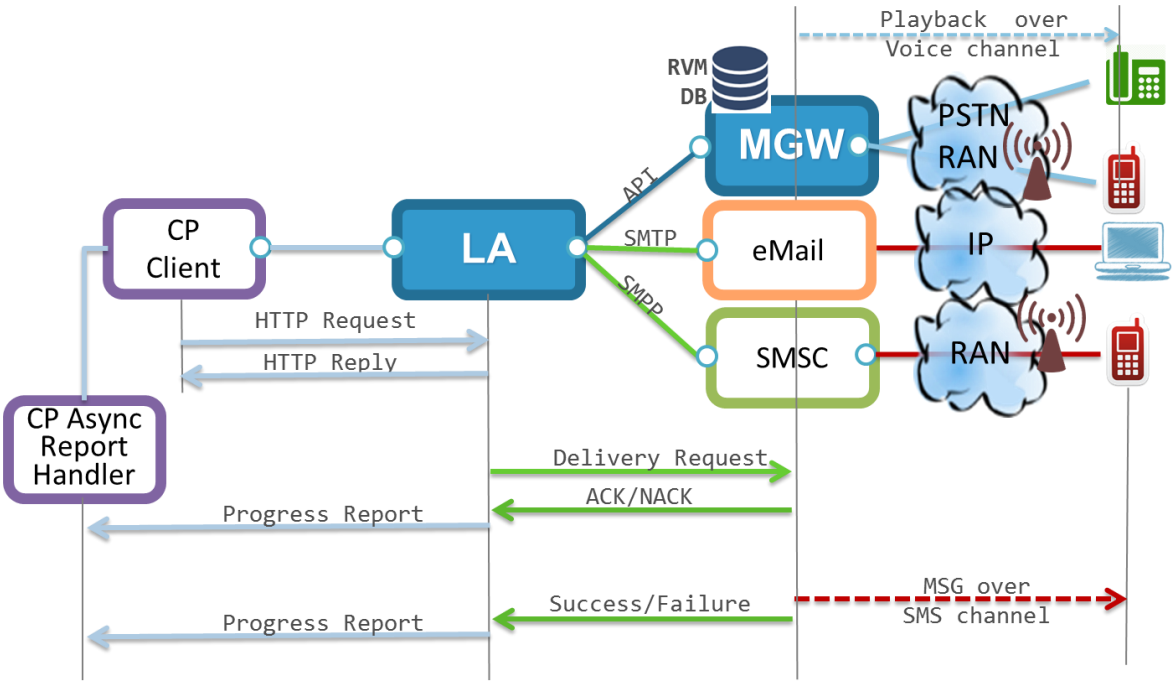
Figure 4 Mapping of Anchor Name 11

Figure 5 Tracking the progress of message delivery 12

1 Introduction

Large-Account Web Service (LA/WS) offers to its clients personalized delivery of content that targets groups of recipients using SMS, Email or recorded announcement (IVR). The raw content delivery service is enhanced by optional features that allow to use ready-to-use content from the libraries of templates, schedule the delivery time and a number of retransmits in case the recipient is not available, receive confirmation reports and so forth.

Figure 1 LA Server in Network



This User Guide specifies APIs provided by **LA/WS**.

> Note: registration with LA/WS that permits its use is described in LA Admin Guide

Structure of the Document

The structure of the document is as follows:

1. Introduction and Common Principles shared by all **LA/WS** services
2. Reference Guide to the Services
3. Appendix

2 Before you start

2.1 Preconditions

The following requirements must be met before using LA/WS (see Section 2.5 for detailed description)

To use SMS or Email Delivery Service

- **Credentials of the User and Account**: The name of your Account, Name of User and its Password must be registered with LA. User must be associated with Account
- **Source IP address** – all LA/WS requests from a client must originate from IP associated with the Account of the client.

To use IVR Delivery Service

- As above
- **VN of the Account**: Virtual Number used by the client of the service must be associated with its Account

2.2 The Method and End-Points on LA

LA/WS is provided over **HTTP** and **HTTPS**. All service requests are **POST** requests.

The end-points of the service are defined in the following table:

Table 1 Endpoints of LA

Type of Access	The Endpoint
HTTPS	**https://la.cellactpro.com/lawebservice/SendSms.asmx**
HTTP	**http://la1.cellactpro.com/SendSms.asmx**

Note: In this document we use acronym HTTP to represent both HTTP and HTTPS options

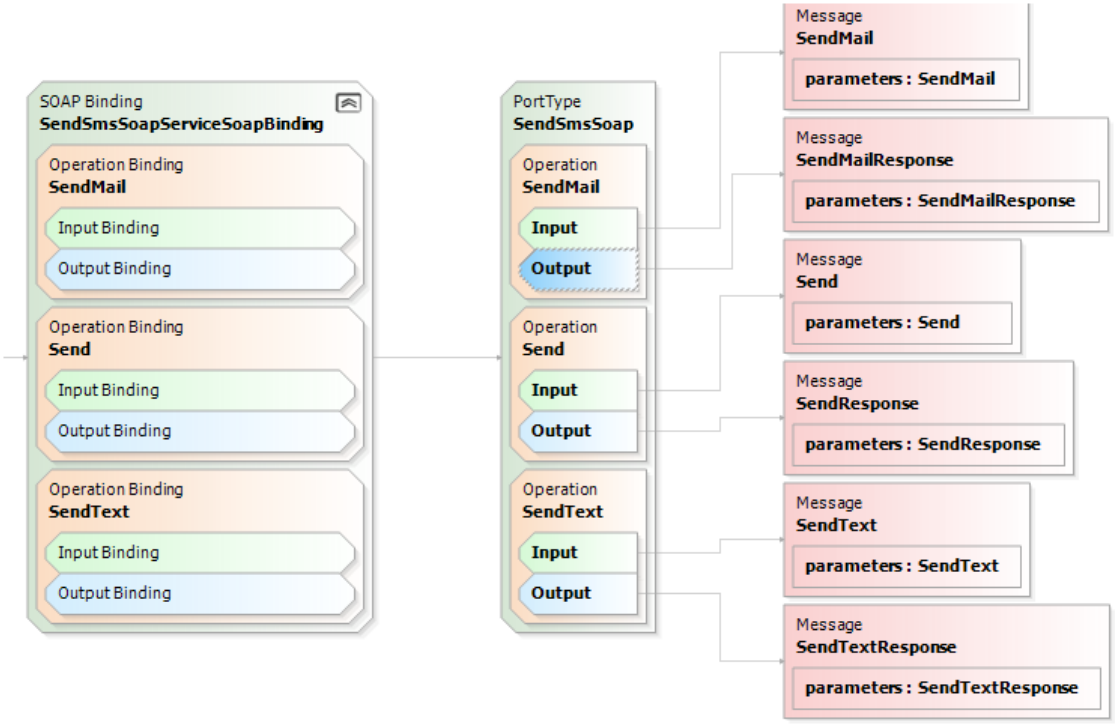
2.3 End-Points on CP for Asynchronous Messages

CP may provide endpoints to receive asynchronous messages, such as “delivery progress reports”. LA supports “delivery progress” reporting using HTTP. The endpoints specified by CP must allow access from LA.

2.4 Quick Glance at the Service

LA/WS provides the following services using format specified by SOAP:

- **SendMail**
- **Send (**SMS or Recorded Announcement)
- **SendText (** stripped-down method to deliver SMS using default attributes of **Send)**



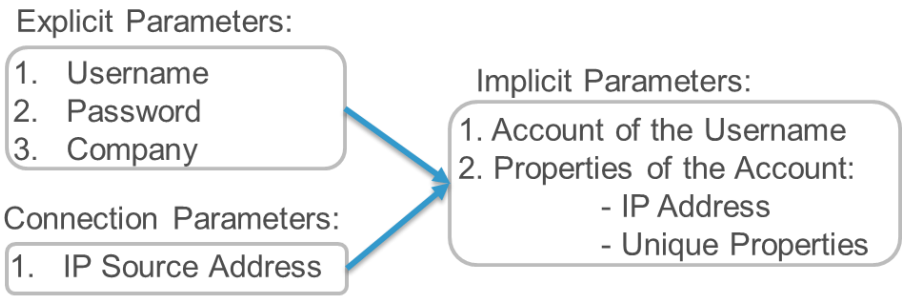
2.5 Authorization to use the Service

Validation of Account's properties

LA/WS is provided to clients that operate on behalf of their accounts. The following properties of an account are essential to use LA/WS services:

- **IP addresses** – any request LA/WS service must originate from IP Address that is associated with the account.
- The caller (username, password) is listed as the member of the **account** ("company") that is registered to use LA

The following chart shows how parameters of the caller (Credentials and IP) are used to authorize client's access to the service



The value specified by the **Company** tag is used to validate that:

- The **IP Source Address** of the caller is registered ("white-listed") on **`Large-Account`** Server and is associated with the account
- **Username** is associated with the account

Validation of User's properties

Once properties of the caller's Account have been validated (known "Company" name and registered source IP address), LA/WS asserts properties of the caller:

- The **Username** is registered on LA Server as the member of the account (**Company)**
- The **Password** is registered on LA Server and is associated with the **Username**
- The status of the **Username** allows using LA/WS services (the user is not disabled)

2.6 Description of attributes of Service

Attributes of WS Service are specified in this document using the following table template:

Name	M	Description	Type	Range
Name of the Attribute	**Yes** - Mandatory **No** - Optional	Description of the attribute	Type of the attribute	Allowed range

3 Common Principles

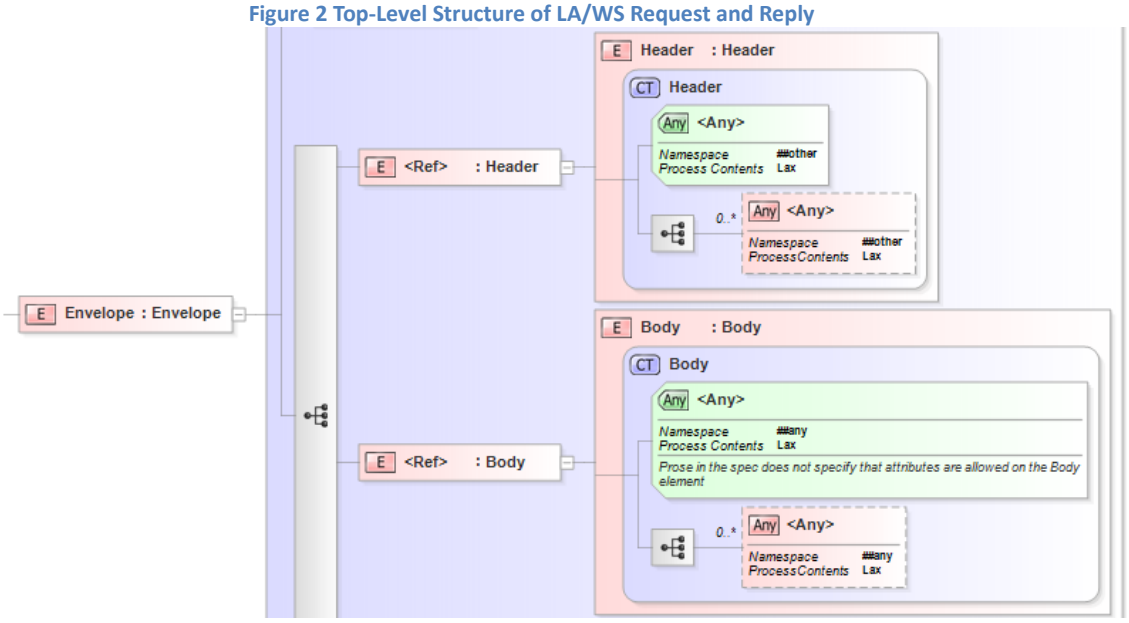
This chapter specifies common characteristics shared by all **LA/WS** Requests and Responses

3.1 Structure of Request

All **LA/WS** Requests are HTTP **POST**s. The HTTP Body of any request is structured according to the SOAP <Envelope> structure as defined in **http://schemas.xmlsoap.org/soap/envelope/**.

In particular, the SOAP <Envelope> carries <**soapenv:Header**> and <**soapenv:Body**> sections

Figure 2 Top-Level Structure of LA/WS Request and Reply



In the case of **LA/WS:**

- the **Header** section is empty: <soapenv:Header/>
- the **Body**(<soapenv:Body>) is structured as follows:
  - The "root" attribute of the **Body** specifies the intended service:

    Send, SendText, SendMail
  - Parameters subtended to the "root" are specific to the specified service. Nevertheless, each service request carries in its body the **Credentials** of the caller.

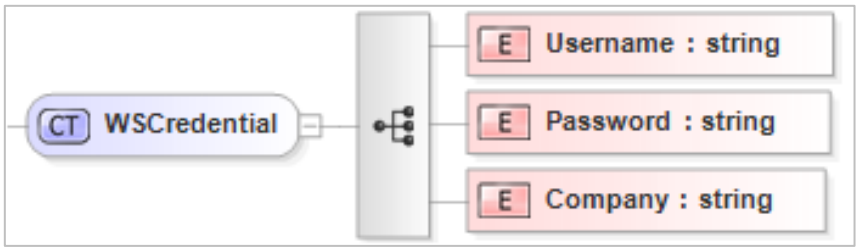


Table 2 Credentials

Name	M	Description	Type	Range
Username	yes	Name of a **representative of account** as it is registered in **LA**	string	**Up to 30 chars**
Password	yes	Password of the **UserName** as it is registered in **LA**	string	
Company	yes	Name of the **account** as it is registered in **LA**	string	

Example

All uppercase sample values are placeholders. Replace them with your own account details, authorized sender and recipient numbers, and callback URL before use. X characters in number-format examples represent digits. Content-Length depends on the exact UTF-8 request body and must be calculated by your HTTP client.
The following capture shows the content of the HTTP Body of the **SendText** request that intends to deliver SMS to the list of recipients. The content of the soapenv:Body carries the command (**SendText**) and three sections associated with this particular command:

- Credentials of the requestor
- Content to be delivered: Hello from LA/WS SendText!
- List of targets: RECIPIENT_PHONE_1, RECIPIENT_PHONE_2, RECIPIENT_PHONE_3

```
POST https://la.cellactpro.com/lawebservice/SendSms.asmx HTTP/1.1
Content-Length: 744
Host: la.cellactpro.com
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://www.cellact.com/webservices/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:SendText>
      <web:credentials>
        <web:username>YOUR USERNAME</web:username>
        <web:password>YOUR PASSWORD</web:password>
        <web:company>YOUR COMPANY</web:company>
      </web:credentials>
      <web:content>Hello from LA/WS SendText!</web:content>
      <web:destinationAddressSet>
        <web:string>RECIPIENT PHONE 1</web:string>
        <web:string>RECIPIENT PHONE 2</web:string>
        <web:string>RECIPIENT PHONE 3</web:string>
      </web:destinationAddressSet>
    </web:SendText>
  </soapenv:Body>
</soapenv:Envelope>
```

3.2 Response

3.2.1 The Structure of Response

Requests that are **not accepted** by **LA/WS** are responded using standard HTTP **4xx** and **5xx** codes.

Requests that are **accepted** by **LA/WS** are responded with the **HTTP 200 (OK)** message while the content of **HTTP body** indicates the outcome of the initial processing of the request.

The body of the HTTP 200 response carries standard SOAP <Envelope> structure (see Figure 2):

- the **Header** section is empty: <soapenv:Header/>
- The **Body** (<soapenv:Body>) carries XML structure that is labeled using the following naming convention:

  DoSomething LA/WS request is responded using XML label
  "**DoSomethingResponse** that contains section labeled
  DoSomethingResult that contains generic structure
  Result that contains
  the **result** and parameters associated with it

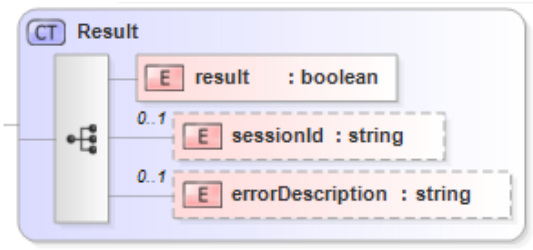
Example: *Acceptance of the SendText* ***request***

The following is an example of the response to the **SendText** API:

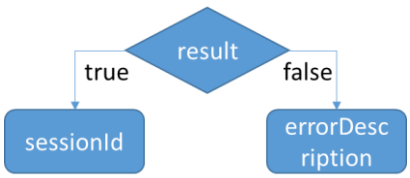
```
HTTP/1.1 200 OK
Content-Type: text/xml;charset=UTF-8
Content-Length: 434
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <SendTextResponse xmlns="http://www.cellact.com/webservices/">
      <SendTextResult>
        <result>true</result>
        <sessionId>00000000-0000-4000-8000-000000000001</sessionId>
        <errorDescription/>
      </SendTextResult>
    </SendTextResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The following table presents the structure of the **Result.** The <result> indicates whether LA/WS accepted the request or declined to process it. The other fields provide further details:

Table 3 Structure of the Result



The relevance of the **SessionId** and **errorDescription** depends on the value of the <result>



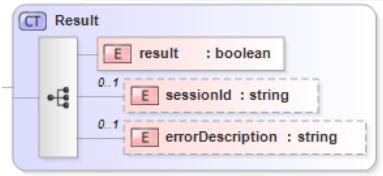
Cellact Proprietary

Table 4 Attributes of Result

Name	M	Description	Type	Note
result	yes	**true** – request has been accepted by LA/WS **false** – request has been declined	bool	
sessionId	y/n	Applicable when the **result==true**. Specifies the reference number assigned to this request. Allows to coordinate further messages associated with this request	string	See Section 3.3.3
errorDescription	y/n	Applicable when the **result==false**. Provides verbal description of the root cause of the rejection	string	See Sections 4.1.4, 5.2.3, 5.1.3

3.2.2 Acceptance of the request

Acceptance of the request by **LA/WS** is indicated by <result>**true**</result> in the **Result**.



It is accompanied by <sessionId> that is provided to facilitate processing of the follow-up messages and reports associated with the request (See **Confirmation of SMS Delivery** in Section 4.1.3).

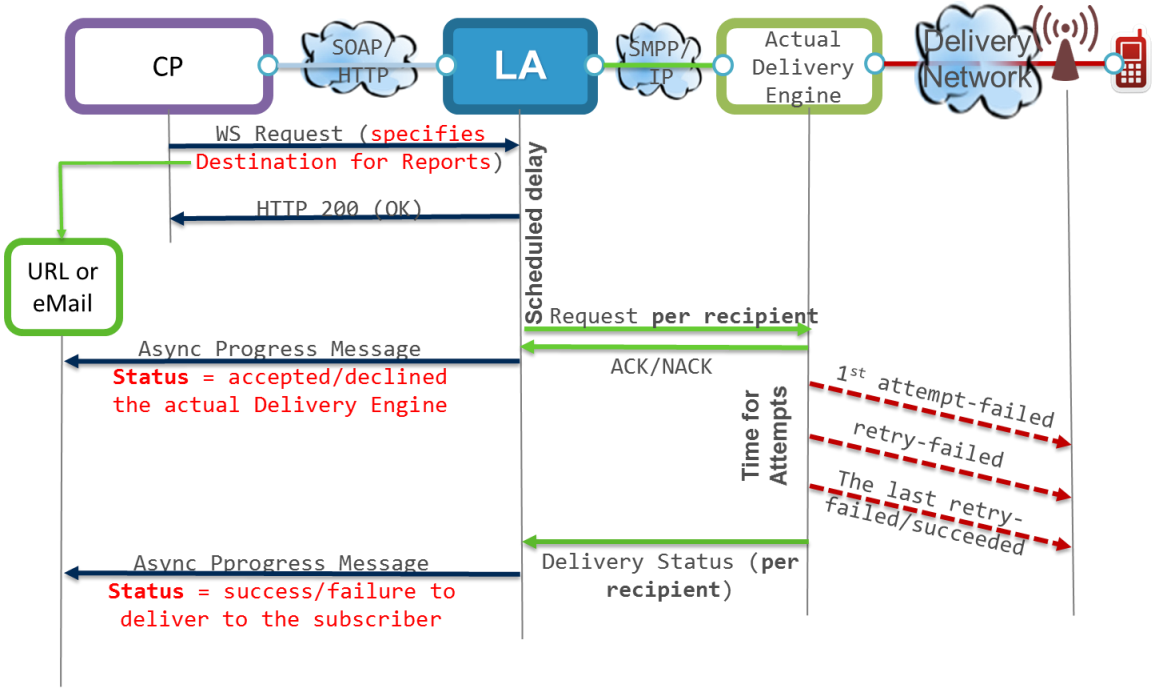
> Note: <result>**true**</result> does not indicate that the target recipient(s) received the message. It just indicates that:
>
> 1. LA accepted the request, and
> 2. LA will forward the request (or necessary requests) to the corresponding service realization engine, such as SMSC, IVR or Email Server (see Figure 1 LA Server in Network).

3.3 Handling Asynchronous Messages

The HTTP 200 (OK) is a **synchronous** response. It indicates that the request has been accepted by LA/WS. However, the client of the LA/WS service may request to receive the outcome of the request (by means of HTTP) when the actual server attempts to fulfill it.

> Note: Execution of the request may be explicitly delayed by the caller.In addition, the "Actual Delivery Engine" may use multiple attempts to fulfill it. Therefore, the actual result and the progress message are delivered asynchronously to the request. They are delivered to the specified destination while using the specified delivery method(HTTP)

Figure 3 Delivery Reports are Async messages



LA/WS supports 2 methods to receive Status-of-Delivery reports as HTTP GET to the specified URL.

> Note: Please see section 3.4.2 for the internal structure of the reports

In addition, LA/WS supports 2 methods that allow the caller to associate async messages (such as progress reports) with the original request:

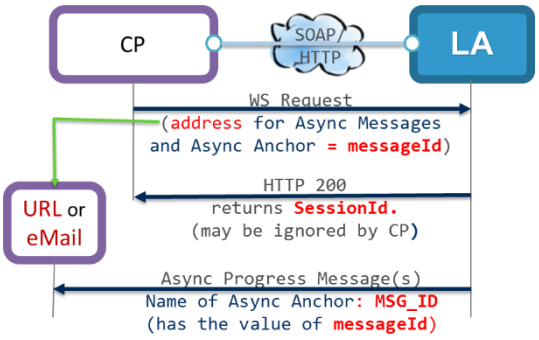
- Using association "anchor" provided by CP Client in the request
- Using association "anchor" provided by LA/WS in HTTP 200 response to the request

3.3.1 Async-Anchor provided by CP Client

Using this method, CP Client provides the "anchor" (**messageId**) in the body of the request. LA/WS applies the provided "anchor" in all follow-up messages associated with the original request.

Cellact Proprietary

Please note that the name of this Anchor in the Async Progress Messages is **MSG_ID**, yet it carries the value tagged by the **messageId** in the original request.



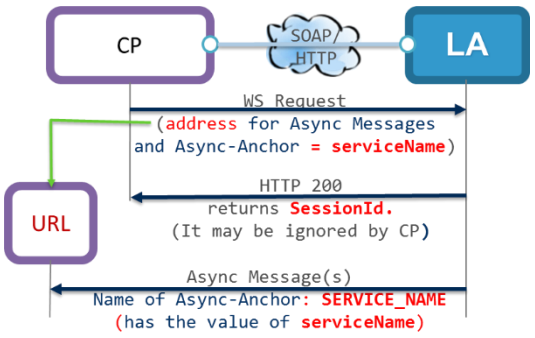
Type of Async	Name of Async Anchor in the Async Message
URL	MSG_ID

Alternative method that applies to URL only

Using this method, CP Client provides the "anchor" (**serviceName**) in the body of the request. LA/WS applies the provided "anchor" in all follow-up messages associated with the original request.

Please note that the name of this Anchor in the Async Progress Messages is **SERVICE_NAME**, yet it carries the value tagged by the **serviceName** in the original request.

Please note that this method applies to URL only

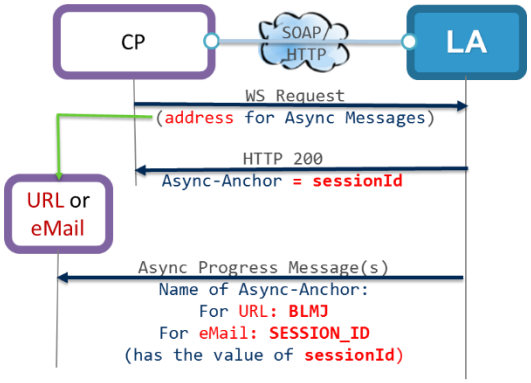


Type of Async	Name of Async Anchor in the Async Message
URL	SERVICE_NAME

3.3.2 Async-Anchor provided by LA/WS

In this method, LA/WS returns an "anchor" (**sessionId**) in the HTTP 200 (OK) response to the request. LA/WS applies it in all follow-up messages associated with the original request.

Please note that the name of this Anchor in the Async Progress Messages depends on the type of Async Message Handler as specified below. Nevertheless, the value carried in the "anchor" field is equal to the value tagged by the **sessionId** in the response to the original request.



Type of Async	Name of Async Anchor in the Async Message
URL	BLMJ

3.3.3 Summary: Mapping of Async-Anchor Names

The following table summarizes the mapping of Anchor Names

Cellact Proprietary

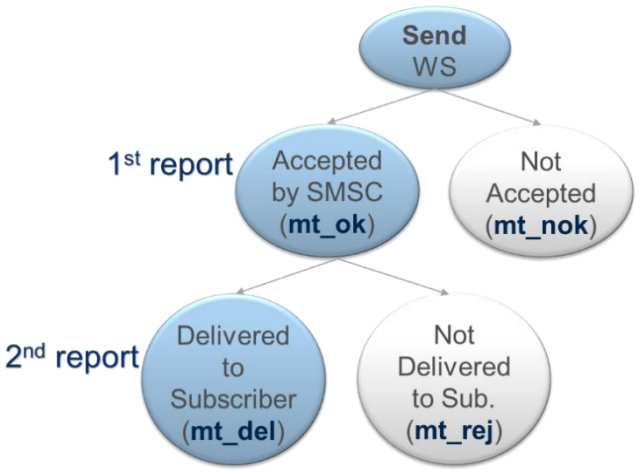
Figure 4 Mapping of Anchor Name

Async-Anchor provided by	How and when it is provided	Name of the corresponding Anchor in the Async Message to: Name of Anchor	Name of the corresponding Anchor in the Async Message to: URL (HTTP)
CP Client	In HTTP POST Request to LA/WS	messageId	MSG_ID
	In HTTP POST Request to LA/WS	serviceName	SERVICE_NAME
LA/WS	In HTTP 200 Response to CP (result=true)	sessionId	BLMJ

3.4 Reports - Tracking the Progress of Delivery

3.4.1 Confirmation of Delivery

The success of SMS delivery to a subscriber depends on the status of network elements in the Mobile Operator's Network and the status of each designated recipient of SMS.



This feature provides progress status as reported by SMSC. It is composed of up to 2 reports per subscriber. The 1st report indicates whether the request has reached the designated SMSC and has been accepted by it. The 2nd report indicates whether the SMS has been delivered to the subscriber.

Cellact Proprietary

Figure 5 Tracking the progress of message delivery

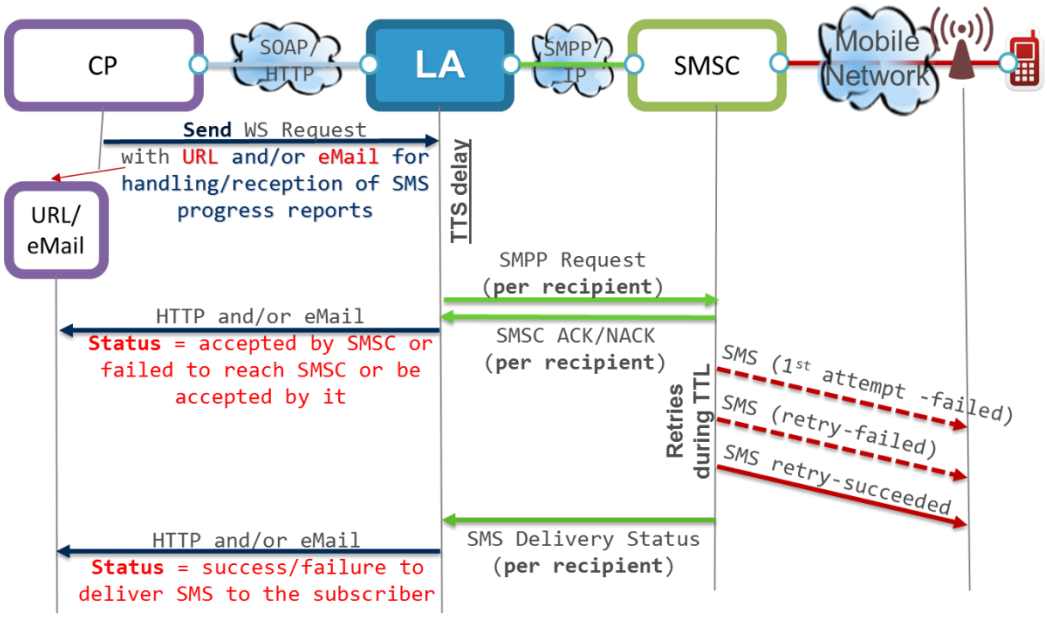


Table 5 SMS Delivery Status

	Description	"REASON"
mt_nok	The request has been declined by SMSC or SMPP-aware elements on the route to SMSC	See Appendix
mt_ok	The message was accepted by SMSC	By default - 5000
mt_rej	All attempts (by SMSC) to deliver the message to the destination have failed.	See Appendix
mt_del	The message was delivered to the destination	By default - 1000

Example

The following timeline shows reports generated when processing WS **Send** Request with the following delivery attributes:

- TimeToSend - in 45 min,
- 5 Recipients: A, B, C, X and Y
- Status of each recipient during an attempt to deliver SMS – see the table
- TimeToLive – 90 min (pause between retries – some default value (i.e. 30 min))

Table 6 SMS Delivery Timeline and the corresponding reports

Time relative to WS Send reception (by LA)		45+0	45+30	45+60	45+90	
Recipient	**Status of the recipient during an attempt to deliver SMS**		**1st retry**	**2nd retry**	**3rd retry**	**Description**
A	Subscriber is available	**mt_ok, mt_del**				**Delivery instantly confirmed**
B	Subscriber becomes available during the 2nd retry	**mt_ok**		**mt-del**		**Delivery confirmed after the 2nd retry**
C	Not available during all attempts	**mt_ok**			**mt_rej**	**All attempts failed**
X	Subscriber has unknown MSISDN	**mt_ok, mt_rej**				**Delivery instantly rejected**

Cellact Proprietary

Y	Not Applicable	**mt_nok**		**Request failed to reach SMSC**

> **Note**: For a list of N recipients, LA generates N requests.
>
> Each request results in 1 or 2 reports.
>
> Therefore, the total number of reports regarding the delivery shall be in the range between N and 2N

3.4.2 Reporting by means of HTTP

Delivery-Progress Report is delivered to the provided URL using HTTP/GET. The CONFIRMATION query parameter contains the URL-encoded XML report:

```
https://example.com/la/status?CONFIRMATION=%3CPALO%3E%3CBLMJ%3E00000000-
0000-4000-8000-
000000000001%3C%2FBLMJ%3E%3CSENDER%3ESENDER_NUMBER%3C%2FSENDER%3E%3CRECIP
IENT%20MNP=%2297253%22%3ERECIPIENT_PHONE_1%3C%2FRECIPIENT%3E%3CFINAL_DATE
%3E20150420184355%3C%2FFINAL_DATE%3E%3CEVT%3Emt_ok%3C%2FEVT%3E%3CREASON%3
E5000%3C%2FREASON%3E%3CSERVICE_NAME%3Eaaa$bbb%3C%2FSERVICE_NAME%3E%3CMESS
AGE_COUNT%3E1%3C%2FMESSAGE_COUNT%3E%3CMSG_ID%3E00000000-0000-4000-8000-
000000000002%3C%2FMSG_ID%3E%3C%2FPALO%3E
```

The same data using pretty-print:

```
CONFIRMATION=
<PALO>
    <BLMJ>00000000-0000-4000-8000-000000000001</BLMJ>
    <SENDER>SENDER_NUMBER</SENDER>
    <RECIPIENT MNP="97253">RECIPIENT_PHONE_1</RECIPIENT>
    <FINAL_DATE>20150420184355</FINAL_DATE>
    <EVT>mt_ok</EVT>
    <REASON>5000</REASON>
    <SERVICE_NAME>aaa$bbb</SERVICE_NAME>
    <MESSAGE_COUNT>1</MESSAGE_COUNT>
    <MSG_ID>00000000-0000-4000-8000-000000000002</MSG_ID>
</PALO>
```

Table 7 Attribute of Confirmation Report (HTTP)

Name	Description	Type
BLMJ	Async-Anchor that carries the value of **`SessionId`** (see Section 3.3.3)	**string**
SENDER	Sender of the SMS	**string**
RECIPIENT.MNP	ID of Mobile Operator that delivered the message	**string**
RECIPIENT	MSISDN of the recipient	**string**
FINAL_DATE	The timestamp of the report Format: yyyymmddhhmmss	**string**

Cellact Proprietary

Name	Description	Type
EVT	One of **mt_ok, mt_nok, mt_del, mt_rej**	**enum**
REASON	The code that further identifies the cause of failure (See Table 11)	**int**
SERVICE_NAME	Async-Anchor that carries the value of **`serviceName`** (see Section 3.3.3) Void if **`serviceName`** was not provided by the original **`Send`** request.	**string**
MESSAGE_COUNT	Number of SMSs used to deliver the message	**int**
MSG_ID	Async-Anchor that carries the value of **`messageId`** (see Section 3.3.3) Void if **`messageId`** was not provided in the original **`Send`** request.	**string**

The following are examples of the content of HTTP-based status reports

MT_OK

See the example above

MT_NOK

Indicates failure to deliver the SMS to the MNO

```
CONFIRMATION=
<PALO>
    <BLMJ>00000000-0000-4000-8000-000000000001</BLMJ>
    <SENDER>SENDER_NUMBER</SENDER>
    <RECIPIENT MNP="97250">RECIPIENT_PHONE_1</RECIPIENT>
    <FINAL_DATE>20150420184355</FINAL_DATE>
    <EVT>mt_nok</EVT>
    <REASON>2010</REASON>
    <SERVICE_NAME>aaa$bbb</SERVICE_NAME>
    <MESSAGE_COUNT>1</MESSAGE_COUNT>
    <MSG_ID>00000000-0000-4000-8000-000000000002</MSG_ID>
</PALO>
```

Note: REASON=2010 indicates that the “Subscriber is blocked”

MT_DEL

Indicates success to deliver SMS to the recipient.

```
CONFIRMATION=
<PALO>
    <BLMJ>00000000-0000-4000-8000-000000000001</BLMJ>
    <SENDER>SENDER_NUMBER</SENDER>
    <RECIPIENT MNP="97253">RECIPIENT_PHONE_1</RECIPIENT>
    <FINAL_DATE>20150420184355</FINAL_DATE>
    <EVT>mt_del</EVT>
    <REASON>1000</REASON>
    <SERVICE_NAME>aaa$bbb</SERVICE_NAME>
    <MESSAGE_COUNT>1</MESSAGE_COUNT>
    <MSG_ID>00000000-0000-4000-8000-000000000002</MSG_ID>
</PALO>
```

MT_REJ

Indicates failure to deliver SMS to the recipient.

```
CONFIRMATION=
<PALO>
    <BLMJ>00000000-0000-4000-8000-000000000001</BLMJ>
    <SENDER>SENDER_NUMBER</SENDER>
    <RECIPIENT MNP="97253">RECIPIENT_PHONE_1</RECIPIENT>
    <FINAL_DATE>20150420184355</FINAL_DATE>
    <EVT>mt_rej</EVT>
    <REASON>1002</REASON>
    <SERVICE_NAME>aaa$bbb</SERVICE_NAME>
    <MESSAGE_COUNT>1</MESSAGE_COUNT>
    <MSG_ID>00000000-0000-4000-8000-000000000002</MSG_ID>
</PALO>
```

Note: REASON=1002 indicates that the "Subscriber is not available."

4 "Send" group of Web Services

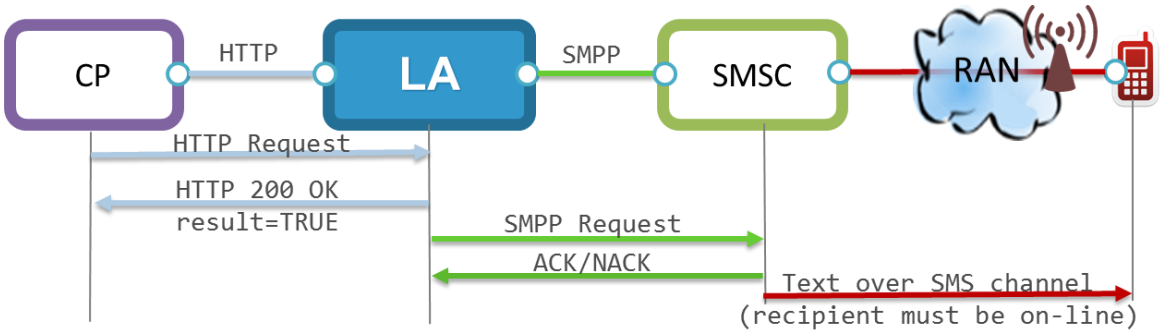
This chapter is the reference specification for variants of the **Send** Service. The actual service is specified by the **command** attribute:

- sendtextmt - deliver SMS with various selection and control options
- sendivr – deliver IVR with various selection and control options
- sendivrmt – deliver the default Recorded Announcement

The distinctive feature of this group of services is an option to track the progress of the delivery by means of asynchronous status reports

Cellact Proprietary

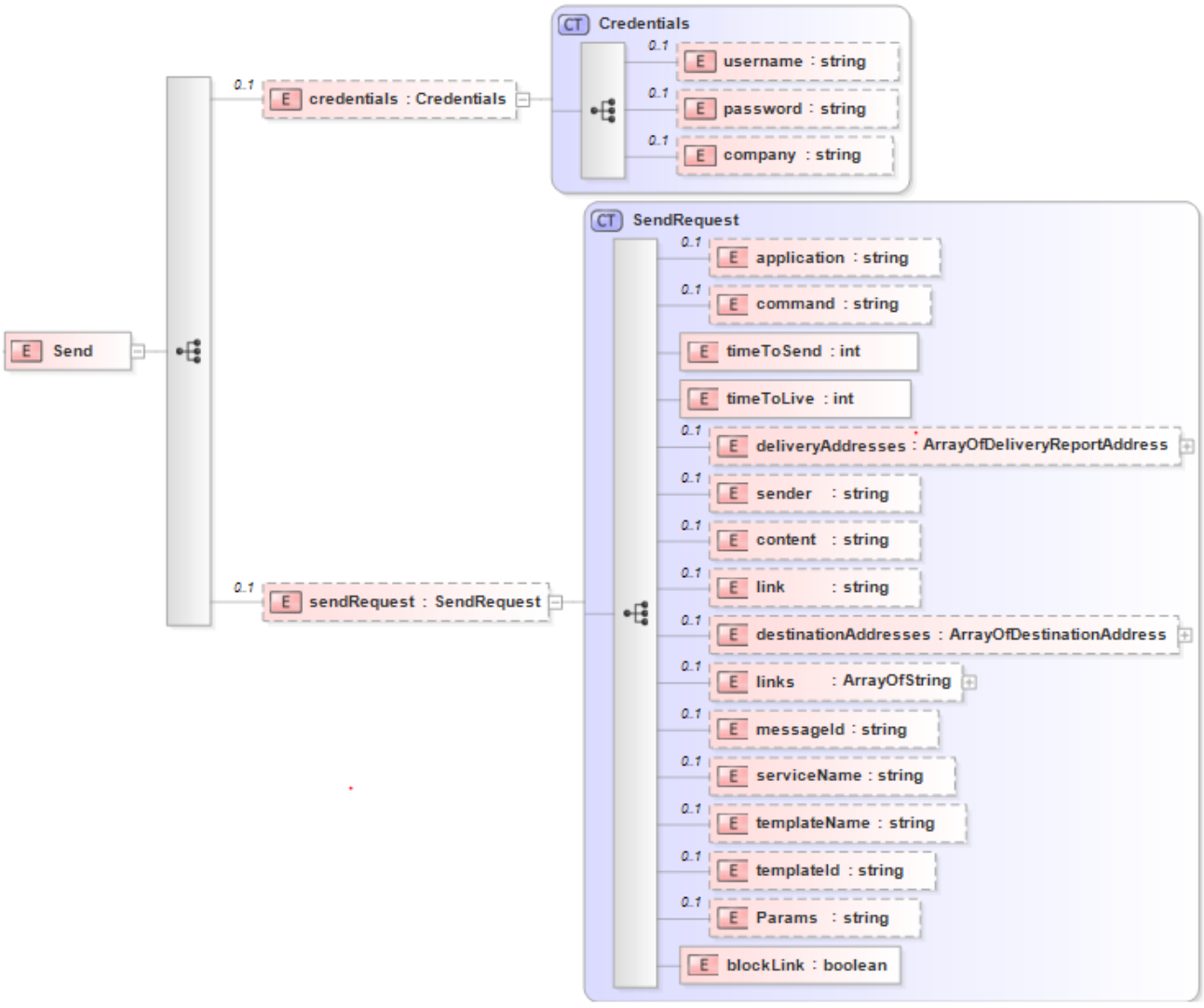
4.1 Send:sendtextmt



This variant of the **Send** WS (**command=sendtextmt**) attempts to trigger delivery of (personalized) text to the specified list of recipients using SMS as the delivery media. Optional parameters of the WS are used to specify features that control the delivery of SMS and reporting attributes.

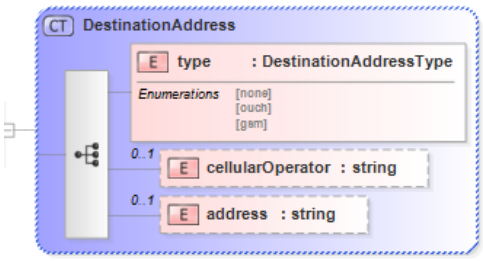
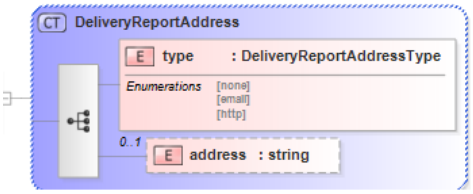
4.1.1 Request

Top-Level Diagram



Cellact Proprietary

Subtypes

destinationAddresses – array of	deliveryReportAddresses – array of
	

Parameters

Parameters of the <**soapenv**:**Body**> of the Send WS are:

Table 8 Parameters of Send

Name	M	Description	Type	Range/Note
credentials				
Username	Yes	**See Table 2**		
Password	Yes			
Company	Yes			
End of credentials				
sendRequest – attributes of the delivery				
application	No	Legacy attribute If used, must be set to "**LA**" Default: **LA**	string	**LA**
command	No	Allows to specify variants of Send Service (such as Send Text, Send IVR) Default: "**sendtextmt.**"	enum	**sendtextmt, sendivrmt, sendivr**
timeToSend	No	Indicates to **LA/WS** when to deliver the message relative to the time of the request. Units – minutes. Default: 0 (Immediately)	int	**0..144000 (minutes)**
timeToLive	No	The period of time (in minutes) that the SMSC (element of Mobile Network) attempts to deliver SMS to the subscriber. Default: 1440	int	**15..1440 (minutes)**
deliveryAddresses (array of DeliveryReportAddress) – specifies addresses for the Delivery Progress Reports (see Section 3.4)				

Name	M	Description	Type	Range/Note
type	No	Type of address – for receiving confirmation to URL set "http"; Default – no confirmation	enum	**http**
address	No	Mandatory if "**type**" is specified Example of URL: 'https://example.com/la/status'		
End of deliveryAddresses				
links	No	Reserved for future use		
sender	No	String that is presented to the recipient as identification of the author of the content	string	**Up to 50 char**
content	Yes	The content of the message.	string	**Up to 800 chars**
link	No	Applicable to command **sendivr**. Indicates the ID of the Recorded Announcements (of the Account that is associated with the **UserName**)	string	**Up to 512 chars**
destinationAddresses (array of DestinationAddress) – specifies the designated recipients of the message				
type	No	Type of Radio Network. Legacy attribute – not applicable	string	
cellularOperator	No	Name of the Cellular operator Legacy attribute – not applicable	string	
address	Yes	MSISDN of the designated recipient. Possible formats are set in LA	string	**Up to 16 chars**
End of destinationAddresses				
links	No	Reserved for Future Use		
messageId	No	The "Async Anchor" provided by the caller. See Section 3.3.1	string	**Up to 40 char**
serviceName	No	The "Async Anchor" provided by the caller. See Section 3.3.1	string	**Up to 50 char**
templateName	No	Name of Template ("Prerecorded Content") associated with the Account of the caller	string	**Up to 50 char**
templateId	No	Reserved for the future use		

Cellact Proprietary

Name	M	Description	Type	Range/Note
Params	No	semicolon-separated list. Each entry substitutes the corresponding parameter in the message. See Specification of Params Order below	string	**Up to 1000 chars**
blockLink	No	Reserved for the future use		
End of sendRequest				

Note: if both "content" and "templateName" are specified - the content is ignored

Specification of Params order

#	English	Hebrew
1.	{1:Lastname}	{1:שמשפחה}
2.	{2:Firstname}	{2:שפרטי}
3.	{3:Role}	{3:תפקיד}
4.	{4:Phone}	{4:טלפון}
5.	{5:Email}	{5:דואל}
6.	{6:Birthdate}	{6:לידה}
7.	{7:Reserved1}	{7:שמור-א}
8.	{8:Reserved2}	{8:שמור-ב}
9.	{9:Reserved3}	{9:שמור-ג}
10.	{10:Reserved4}	{10:שמור-ד}
11.	{11:Reserved5}	{11:שמור-ה}
12.	{12:Reserved6}	{12:שמור-ו}
13.	{13:Reserved7}	{13:שמור-ז}
14.	{14:Reserved8}	{14:שמור-ח}
15.	{15:Reserved9}	{15:שמור-ט}
16.	{16:Reserved10}	{16:שמור-י}

XML Spec

HTTP Request has the following body

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://www.cellact.com/webservices/">
  <soapenv:Body>
    <web:Send>
      <web:credentials>
        <web:username>string</web:username>
        <web:password>string</web:password>
        <web:company>string</web:company>
      </web:credentials>
      <web:sendRequest>
        <web:application>string</web:application>
```

```
        <web:command>string</web:command>
        <web:timeToSend>int</web:timeToSend>
        <web:timeToLive>int</web:timeToLive>
        <web:deliveryAddresses>
          <web:DeliveryReportAddress>
            <web:type>string</web:type>
            <web:address>string</web:address>
          </web:DeliveryReportAddress>
        </web:deliveryAddresses>
        <web:sender>string</web:sender>
        <web:content>string</web:content>
        <web:link>string</web:link>
        <web:destinationAddresses>
          <web:DestinationAddress>
            <web:type>string</web:type>
            <web:cellularOperator>string</web:cellularOperator>
            <web:address>string</web:address>
          </web:DestinationAddress>
        </web:destinationAddresses>
        <web:messageId>string</web:messageId>
        <web:serviceName>string</web:serviceName>
        <web:templateName>string</web:templateName>
        <web:templateId>string</web:templateId>
        <web:Params>string</web:Params>
        <web:blockLink>boolean</web:blockLink>
      </web:sendRequest>
    </web:Send>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

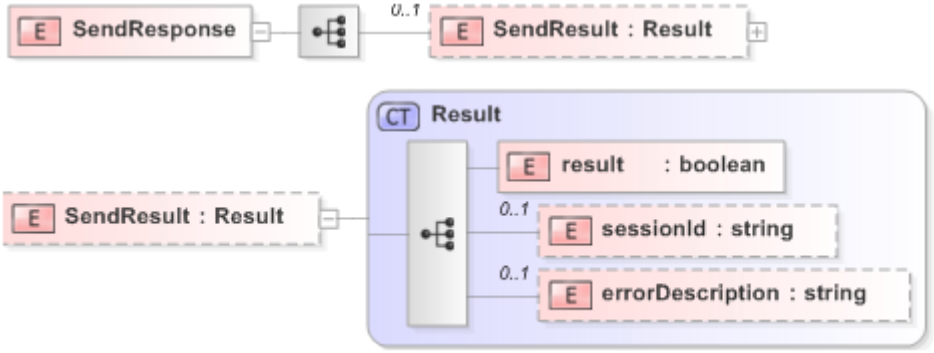
```
POST https://la.cellactpro.com/lawebservice/SendSms.asmx HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: "http://www.cellact.com/webservices/Send"
Content-Length: 1453
Host: la.cellactpro.com
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://www.cellact.com/webservices/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:Send>
      <web:credentials>
        <web:username>YOUR USERNAME</web:username>
        <web:password>YOUR PASSWORD</web:password>
        <web:company>YOUR COMPANY</web:company>
      </web:credentials>
      <web:sendRequest>
        <web:application>LA</web:application>
        <web:command>sendtextmt</web:command>
        <web:deliveryAddresses>
          <web:DeliveryReportAddress>
            <web:type>http</web:type>
            <web:address>https://example.com/la/status</web:address>
          </web:DeliveryReportAddress>
          <web:DeliveryReportAddress>
            <web:type>http</web:type>
            <web:address> https://example.org/la/status</web:address>
          </web:DeliveryReportAddress>
        </web:deliveryAddresses>
        <web:sender>Delivery</web:sender>
        <web:content>Testing LA WS delivery POST</web:content>
        <web:destinationAddresses>
          <web:DestinationAddress>
```

```
            <web:address>RECIPIENT PHONE 1</web:address>
          </web:DestinationAddress>
        </web:destinationAddresses>
        <web:messageId>deliver</web:messageId>
        <web:serviceName>Service Name</web:serviceName>
      </web:sendRequest>
    </web:Send>
  </soapenv:Body>
</soapenv:Envelope>
```

4.1.2 Response

HTTP 200 (OK) SendResponse is used as response to the request

Top-level Diagram



Parameters

See Section 3.2.1

XML Spec

See Section 3.2.1

Example

```
HTTP/1.1 200 OK
Content-Type: text/xml;charset=UTF-8
Content-Length: 418
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <SendResponse xmlns="http://www.cellact.com/webservices/">
      <SendResult>
        <result>true</result>
        <sessionId>00000000-0000-4000-8000-000000000001</sessionId>
        <errorDescription/>
      </SendResult>
    </SendResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.1.3 Confirmation of Delivery

See Section 3.4

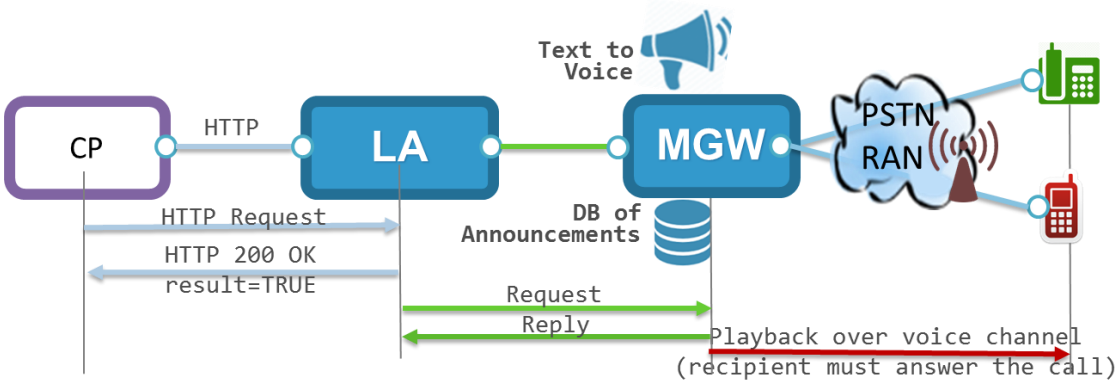
4.1.4 Errors

#	Message in the < errorDescription > field	Description
1.	Invalid account	Invalid credentials
2.	cvc-minLength-valid: Value '' with length = '0' is not facet-valid with respect to minLength '1' for type 'CELLACT-STRING-1-30'.	Credentials are empty
3.	cvc-pattern-valid: Value 'XXX' is not facet-valid with respect to pattern '[Ll][Aa]|[Mm][Aa][Rr][Kk][e][Tt][Ii][Nn][Gg]|[Gg][Ll][Oo][Bb][Aa][Ll][Ss][Mm][Ss]|[Mm][Aa][Ii][Ll]2[Ss][Mm][Ss]|[Mm][Aa][Ii][Ll][Tt][Ee][Xx][Tt]|[Rr][Ee][Dd][Aa][Ll][Ee][Rr][Tt]|[Mm][Tt][Rr][Aa][Dd][Ee]|[Vv][Oo][Tt][Ii][Nn][Gg]|[Mm][Ee][Nn][Tt][Ee][Rr][Pp][Rr][Ii][Ss][Ee]' for type 'CELLACT-APP'.	Invalid value in the application field
4.	cvc-enumeration-valid: Value XXX is not facet-valid with respect to enumeration '[sendtextmt, sendlinkmt, sendringtonemt, sendpicturemt, sendsimplevcardmt, sendoperatorlogomt, sendivrmt, sendivr, sendiconmt, sendwaptextmt, sendbookmarkmt, send-ems-melody, send-ems-melody-predefined, send-motorola-melody, send-ceo-melody, send-ceo-bitmap, send-ceo-vcard, sendmms, sendgamemt, sendkaraokemt, provision, sendmail]'. It must be a value from the enumeration.	Invalid value in the command field
5.	cvc-minInclusive-valid: Value 'XXX' is not facet-valid with respect to minInclusive '1' for type 'CELLACT-TTS'.	Invalid min value in the TTS field
6.	cvc-maxInclusive-valid: Value 'XXX' is not facet-valid with respect to maxInclusive '65535' for type 'CELLACT-TTS'.	Invalid max value in the TTS field
7.	cvc-minInclusive-valid: Value 'XXX' is not facet-valid with respect to minInclusive '15' for type 'CELLACT-TTL'.	Invalid min value in the TTL field
8.	cvc-maxInclusive-valid: Value 'XXX' is not facet-valid with respect to maxInclusive '65535' for type 'CELLACT-TTL'.	Invalid max value in the TTL field
9.	cvc-enumeration-valid: Value 'XXX' is not facet-valid with respect to enumeration '[post]'. It must be a value from the enumeration.	Invalid value in the COD types field
10.	cvc-minLength-valid: Value 'XXX' with length = '0' is not facet-valid with respect to minLength '1' for type 'CELLACT-STRING-1-150'.	Invalid value in the COD addresses field
11.	cvc-maxLength-valid: Value 'XXX' with length = 'XX' is not facet-valid with respect to maxLength '50' for type 'CELLACT-STRING-0-50'.	Invalid value in the sender field
12.	cvc-maxLength-valid: Value 'XXX' with length = 'XXX' is not facet-valid with respect to maxLength '800' for type 'CELLACT-STRING-0-800'.	Invalid value in the content field
13.	cvc-maxLength-valid: Value 'XXX' with length = 'XXX' is not facet-valid with respect to maxLength '512' for type 'CELLACT-STRING-0-512	Invalid value in the link field
14.	cvc-pattern-valid: Value 'XXX' is not facet-valid with respect to pattern '05[0,2,3,4,5,6,7,8,9]-?[0-9]{7}|\+[0-9]{7,16}|([\+\.a-zA-Z0-9_-])+@([a-zA-Z0-9_-])+(([a-zA-Z0-9_-])*\.([a-zA-Z0-9_-])+)+' for type 'CELLACT-MOBILE-INTERNATIONAL-NUMBER'.	Invalid value in the destination addresses field
15	cvc-maxLength-valid: Value 'XXX' with length = 'XX' is not facet-valid with respect to maxLength '40' for type 'CELLACT-STRING-1-40'.	Invalid value in the destination messageId field
16.	cvc-maxLength-valid: Value 'XXX' with length = 'XX' is not facet-valid with respect to maxLength '50' for type 'CELLACT-STRING-1-50'.	Invalid value in the destination ServiceName field
17.	cvc-maxLength-valid: Value 'XXX' with length = 'XX' is not facet-valid with respect to maxLength '50' for type 'CELLACT-STRING-1-50'.	Invalid value in the templateName field
18.	cvc-maxLength-valid: Value 'XXX' with length = 'XXX' is not facet-valid with respect to maxLength '512' for type 'CELLACT-STRING-0-512'.	Invalid value in the templateId field
19.	cvc-maxLength-valid: Value 'XXX' with length = 'XXX' is not facet-valid with respect to maxLength '1000' for type 'CELLACT-STRING-0-1000'.	Invalid value in the Params field

4.2 Send:sendivr

Delivers Voice Announcement to the list of recipients.

Voice Announcement may be either fetched from DB of Announcements or vocalized from the text provided in the **content** field.



Note: This section contains preliminary info regarding flexible IVR delivery service. It supersedes the current capability to deliver Single Announcement (see: **sendivrmt**)

Realization of this service reuses the structure introduced by Send (SMS) service while specifying **command=sendivr**, and making mandatory some attributes as specified below.

4.2.1 Request

See Section4.1.1 for the description of the parameters and XML Format

The following attributes have special meaning in the case of IVR:

Name	M	Description	Type	Range/Note
command	Yes	must be **sendivr**	enum	
sender	Yes	Must be VN number associated with the account	string	**Up to 16 char**
content	No	If **content** is provided, it shall be conveyed as recorded announcement	string	**Up to 800 chars**
link	No	Serial Number of the recorded announcement owned by the account. Note: If both content and link fields are specified - only one source shall be broadcasted, according to the setting in the configuration of LA	string	**1..100**

Name	M	Description	Type	Range/Note
address	Yes	MSISDN of the designated recipient. May be either cellular or **stationary** phone.	string	**Possible formats are set in LA**
timeToSend	No	Not Relevant. Attempting to deliver Immediately		
timeToLive	No	Not Relevant. Default: Number of attempts and the total time designated for retries is controlled by Admin of MGW		

Example 1 – sendivr message with link to predefined recording

```
POST https://la.cellactpro.com/lawebservice/SendSms.asmx HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: "http://www.cellact.com/webservices/Send"
Content-Length: 1266
Host: la.cellactpro.com
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://www.cellact.com/webservices/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:Send>
      <web:credentials>
        <web:username>YOUR USERNAME</web:username>
        <web:password>YOUR PASSWORD</web:password>
        <web:company>YOUR COMPANY</web:company>
      </web:credentials>
      <web:sendRequest>
        <web:application>LA</web:application>
        <web:command>sendivr</web:command>
        <web:deliveryAddresses>
          <web:DeliveryReportAddress>
            <web:type>http</web:type>
            <web:address>https://example.com/la/status</web:address>
          </web:DeliveryReportAddress>
        </web:deliveryAddresses>
        <web:sender>ACCOUNT VN</web:sender>
        <web:link>00000000-0000-4000-8000-000000000001</web:link>
        <web:destinationAddresses>
          <web:DestinationAddress>
            <web:address>RECIPIENT PHONE 1</web:address>
          </web:DestinationAddress>
        </web:destinationAddresses>
        <web:messageId>deliver</web:messageId>
        <web:serviceName>Service Name</web:serviceName>
      </web:sendRequest>
    </web:Send>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 2 – TTS (Text2Speech) sendivr message

```
POST https://la.cellactpro.com/lawebservice/SendSms.asmx HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: "http://www.cellact.com/webservices/Send"
Content-Length: 1250
Host: la.cellactpro.com
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://www.cellact.com/webservices/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:Send>
      <web:credentials>
        <web:username>YOUR USERNAME</web:username>
        <web:password>YOUR PASSWORD</web:password>
        <web:company>YOUR COMPANY</web:company>
      </web:credentials>
      <web:sendRequest>
        <web:application>LA</web:application>
        <web:command>sendivr</web:command>
        <web:deliveryAddresses>
          <web:DeliveryReportAddress>
            <web:type>http</web:type>
            <web:address>https://example.com/la/status</web:address>
          </web:DeliveryReportAddress>
        </web:deliveryAddresses>
        <web:sender>ACCOUNT VN</web:sender>
        <web:content>test test test</web:content>
        <web:destinationAddresses>
          <web:DestinationAddress>
            <web:address>RECIPIENT PHONE 1</web:address>
          </web:DestinationAddress>
        </web:destinationAddresses>
        <web:messageId>deliver</web:messageId>
        <web:serviceName>Service Name</web:serviceName>
      </web:sendRequest>
    </web:Send>
  </soapenv:Body>
</soapenv:Envelope>
```

4.2.2 Response

See Section 4.1.2

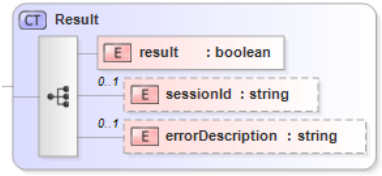
4.2.3 Confirmation of Delivery

See Section 3.4

4.2.4 Rejection of the request

Rejection of the request by **LA/WS** is indicated by <result>**false**</result> in the **Result**.

It is accompanied by < errorDescription > that specifies the 1st error that caused the decline of the request.



> Note: < errorDescription> presents only the 1st error detected by LA/WS software.
>
> That is, if , for example, the list of recipients has few ill-formatted numbers – only the 1st one shall be identified in the **errorDescription**

Example:** Rejection of the Send **request

The following example shows a rejected Send request with an invalid destination number ("911").

```
HTTP/1.1 200 OK
Content-Type: text/xml;charset=UTF-8
Content-Length: 534
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <SendResponse xmlns="http://www.cellact.com/webservices/">
      <SendResult>
        <result>false</result>
        <sessionId/>
        <errorDescription>
          cvc-pattern-valid: Value '911' is not facet-valid with respect
to pattern '05[0,2,3,4,5,6,7,8,9]-?[0-9]{7}|\+[0-9]{7,16}'.
        </errorDescription>
      </SendResult>
    </SendResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

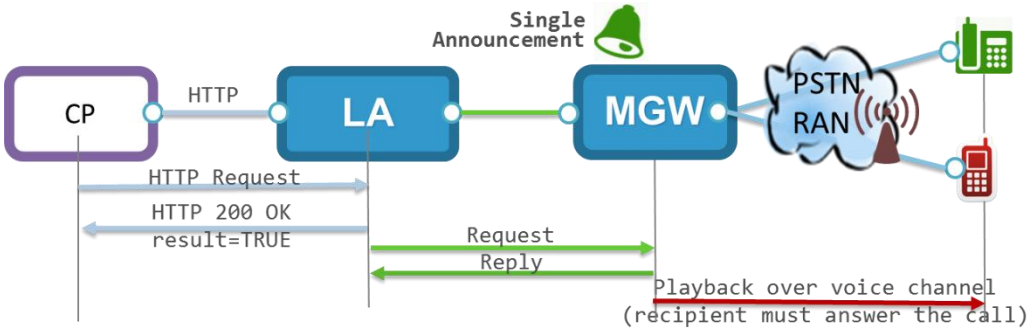
> Note: The content of the error message is generated by SW package that validates the range of the parameters.
>
> In particular, the term "not facet-valid" is the term used by the SW that validates values of parameters in terms of their range, pattern or enum values.

4.2.5 Errors

See Section 4.1.4

4.3 Send:sendivrmt

This is the legacy version of IVR service. It is used to broadcast **the One-and-only-One (default) Announcement**.



Realization of this service reuses the structure introduced by Send service while specifying **command=sendivrmt**, and changing mandatory/optional requirement of some attributes as specified below.

4.3.1 Request

See Section4.1.1 for the description of the parameters and XML Format

The following attributes have special meaning in the case of IVR:

Name	M	Description	Type	Range/Note
command	**Yes**	must be **sendivrmt**	enum	
content	**No**	Is ignored	string	
link	**No**		string	
address	Yes	MSISDN of the designated recipient. May be either cellular or **stationary** phone.	string	**Possible formats are set in LA**
timeToSend	No	Is ignored. Attempting to deliver Immediately		
timeToLive	No	Is ignored. Number of attempts is controlled by MGW Admin		

4.3.2 Response

See Section 4.1.2

4.3.3 Confirmation of Delivery

See Section 3.4

4.3.4 Errors

See Section 4.1.4

Cellact Proprietary

5 Additional Web Services

This chapter is the reference specification for

- SendText (straightforward version of Send SMS)
- SendMail

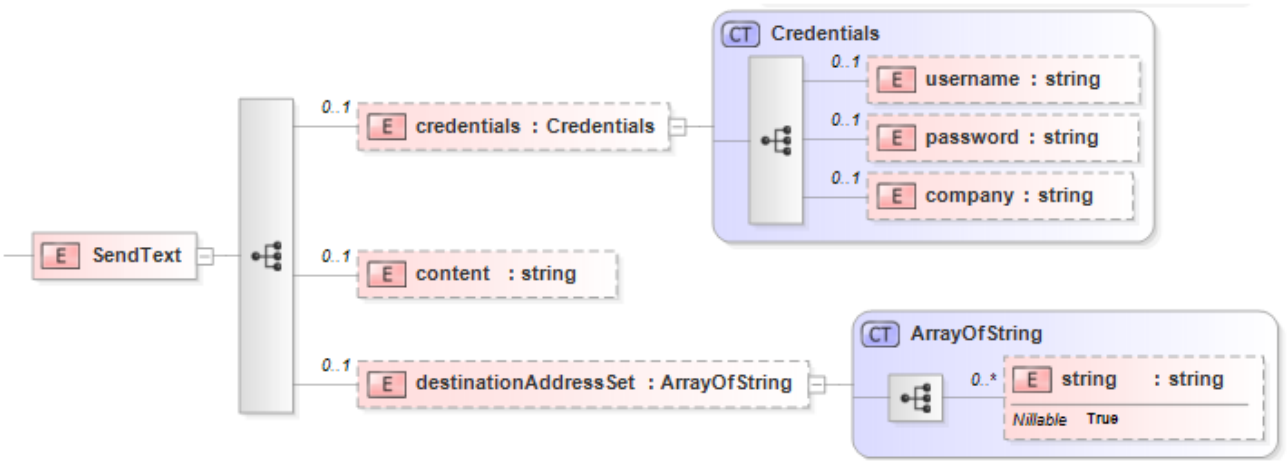
```
Unlike services in Chapter 4,
services in this Chapter do not
provide option to receive
status-of-delivery reports
```

5.1 SendText

SendText is the legacy version of the **Send**. It specifies straightforward method to deliver SMS to the specified recipient/list-of-recipients

5.1.1 Request

Top-Level Diagram



Parameters

Parameters of the <**soapenv**:**Body**> of the Send WS are:

Table 9 Parameters of SendText

Name	M	Description	Type	Range/Note
credentials				
Username	Yes	**See Table 2**		
Password	Yes			
Company	Yes			
End of credentials				
content	Yes	The content of the message.	string	**Up to 800 chars**
destinationAddressSet	yes	Phone Number(s) of the recipient(s).	Array of Strings	**See Patterns in Section 6.1**

Cellact Proprietary

> Note: If the same destination appears multiple times in the list of the recipients - the message is delivered the corresponding number of times

XML Spec

HTTP Request has the following body

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://www.cellact.com/webservices/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:SendText>
      <web:credentials>
        <web:username>string</web:username>
        <web:password>string</web:password>
        <web:company>string</web:company>
      </web:credentials>
      <web:content>string</web:content>
      <web:destinationAddressSet>
        <!--one or more repetitions:-->
        <web:string>PhoneNumber</web:string>
      </web:destinationAddressSet>
    </web:SendText>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

In the following example message Hello from **LA/WS** SendText! is delivered two times to RECIPIENT_PHONE_2 and one time to RECIPIENT_PHONE_3

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://www.cellact.com/webservices/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:SendText>
      <web:credentials>
        <web:username>YOUR USERNAME</web:username>
        <web:password>YOUR PASSWORD</web:password>
        <web:company>YOUR COMPANY</web:company>
      </web:credentials>
      <web:content>Hello from LA/WS SendText!</web:content>
      <web:destinationAddressSet>
        <web:string>RECIPIENT PHONE 2</web:string>
        <web:string>RECIPIENT PHONE 2</web:string>
        <web:string>RECIPIENT PHONE 3</web:string>
      </web:destinationAddressSet>
    </web:SendText>
  </soapenv:Body>
</soapenv:Envelope>
```

5.1.2 Response

HTTP 200 (OK) SendTextResponse is used as response to the request

Top-Level Diagram



Cellact Proprietary

Parameters

See Section 3.2.1

XML Spec

See Section 3.2.1

Example

See Section 3.2.1

5.1.3 Errors

#	Message in the < errorDescription > field	Description
1.	Invalid account	Invalid credentials
2.	cvc-minLength-valid: Value '' with length = '0' is not facet-valid with respect to minLength '1' for type 'CELLACT-STRING-1-30'.	Empty credentials
3.	cvc-maxLength-valid: Value 'XXX' with length = 'XXX' is not facet-valid with respect to maxLength '800' for type 'CELLACT-STRING-0-800'.	Invalid value in the content field
4.	cvc-pattern-valid: Value 'XXX' is not facet-valid with respect to pattern '05[0,2,3,4,5,6,7,8,9]-?[0-9]{7}|\+[0-9]{7,16}|([\+\.a-zA-Z0-9_-])+@([a-zA-Z0-9_-])+(([a-zA-Z0-9_-])*\.([a-zA-Z0-9_-])+)+' for type 'CELLACT-MOBILE-INTERNATIONAL-NUMBER'.	Invalid value in the destination field
5.	cvc-complex-type.2.4.b: The content of element 'DEST_LIST' is not complete. One of '{TO}' is expected.	Empty value in the destination field

5.2 SendMail

Requests email delivery to the specified recipients using a template configured for the authenticated Large Account. Set templateName to a template belonging to that account and use Params for substitutions.

A successful SOAP response confirms request acceptance; it does not confirm email delivery.

5.2.1 Request

Top-level Diagram





Parameters

Table 10 Parameters of SendMail

Name	M	Description	Type	Range/Note
credentials				
Username	Yes	**See Table 2**		
Password	Yes			
Company	Yes			
End of credentials				
sendRequestMail – attributes of the delivery				
application	No	Legacy attribute If used, must be set to "**LA**" Default: **LA**	string	**LA**
command	No	Use sendmailmt for template-based email. The service also accepts sendmail.	string	**sendmailmt, sendmail**

Cellact Proprietary

Name	M	Description	Type	Range/Note
timeToSend	No	Delay before delivery, in minutes. Use 0 for immediate delivery.	int	**0..144000 (minutes)**
timeToLive	No	The period of time (in minutes) that the Email Server attempts to deliver an Email to the subscriber. Default: 1440	int	**15..1440 (minutes)**
sender	Yes	Email address of the sender. The sender Email address should be defined in Cellact systems and actively approved by the customer in a preliminary authorization process. The sender address is **case sensitive** and should be used in the same format as initially defined and approved.	string	**Email format** **Up to 50 char**
senderName	No	String that is presented to the recipient as identification of the author of the content	string	**Up to 50 char**
DestinationMailAddresses (array of DestinationMailAddress) – specifies Email addresses of the designated recipients				
messageId	No	The "Async Anchor" provided by the caller. See Section 3.3.1	string	**Up to 40 char**
serviceName	No		string	**Up to 50 char**
subject	Yes	The Subject field of the Email.	string	**Up to 100 chars**
templateName	Yes	Name of an email template configured for the authenticated account.	string	**Up to 50 char**
templateId	No	ID of the Email template **that is defined in LA**	string	**Up to 512 char**
Params	No	Semicolon-separated list. Each entry substitutes the corresponding parameter in the message. See "Params" in Section 4.1.1	string	**Up to 1000 chars**
campaignName	no	name of the campaign	string	**Up to 50 char**
replyTo	no	The Email address to be used for the reply	string	**Email format** **Up to 50 char**
End of sendRequestMail				

XML Spec

HTTP Request is comprised of the standard Header and the following Body

```
<soap:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendMail xmlns="http://www.cellact.com/webservices/">
      <credentials>
        <username>string</username>
        <password>string</password>
        <company>string</company>
      </credentials>
      <sendRequestMail>
        <application>string</application>
        <command>string</command>
        <timeToSend>int</timeToSend>
        <timeToLive>int</timeToLive>
        <sender>string</sender>
        <senderName>string</senderName>
        <destinationMailAddresses>
          <DestinationMailAddress>
            <address>string</address>
          </DestinationMailAddress>
        </destinationMailAddresses>
        <messageId>string</messageId>
        <serviceName>string</serviceName>
        <subject>string</subject>
        <templateName>string</templateName>
        <templateId>string</templateId>
        <Params>string</Params>
        <campaignName>string</campaignName>
        <replyTo>string</replyTo>
      </sendRequestMail>
    </SendMail>
  </soap:Body>
</soap:Envelope>
```

Example

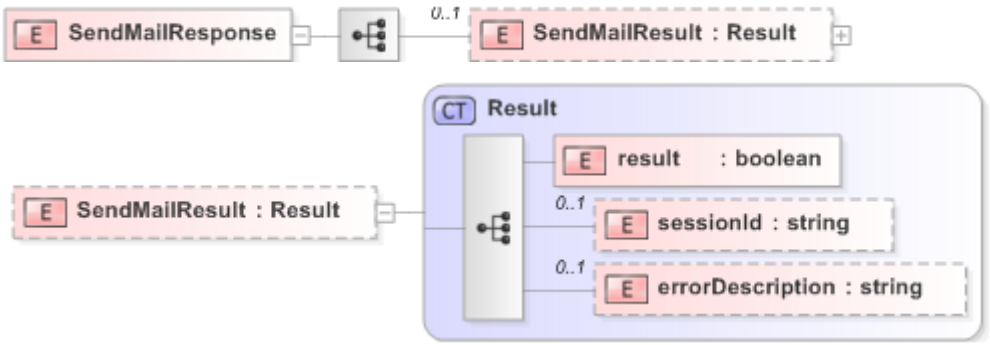
```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://www.cellact.com/webservices/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:SendMail>
      <web:credentials>
        <web:username>YOUR USERNAME</web:username>
        <web:password>YOUR PASSWORD</web:password>
        <web:company>YOUR COMPANY</web:company>
      </web:credentials>
      <web:sendRequestMail>
        <web:application>LA</web:application>
        <web:command>sendmailmt</web:command>
        <web:timeToSend>0</web:timeToSend>
        <web:timeToLive>15</web:timeToLive>
        <web:sender>sender@example.com</web:sender>
        <web:senderName>Example Sender</web:senderName>
        <web:destinationMailAddresses>
          <web:DestinationMailAddress>
            <web:address>recipient@example.com</web:address>
          </web:DestinationMailAddress>
        </web:destinationMailAddresses>
        <web:messageId>123456</web:messageId>
        <web:serviceName>Service Name</web:serviceName>
        <web:subject>Template email example</web:subject>
        <web:templateName>YOUR ACCOUNT TEMPLATE NAME</web:templateName>
        <web:templateId/>
```

```
        <web:Params/>
        <web:campaignName/>
        <web:replyTo/>
      </web:sendRequestMail>
    </web:SendMail>
  </soapenv:Body>
</soapenv:Envelope>
```

5.2.2 Response

HTTP 200 (OK) SendMailResponse is used as response to the request

Top-Level Diagram



Parameters

See Section 3.2.1

XML Spec

See Section 3.2.1

Example

See Section 3.2.1

5.2.3 Errors

#	Message in the < errorDescription > field	Description
1.	Invalid account	Invalid credentials
2.	cvc-minLength-valid: Value '' with length = '0' is not facet-valid with respect to minLength '1' for type 'CELLACT-STRING-1-30'.	Empty credentials field
3.	cvc-pattern-valid: Value 'XXX' is not facet-valid with respect to pattern '[Ll][Aa]|[Mm][Aa][Rr][Kk][e][Tt][Ii][Nn][Gg]|[Gg][Ll][Oo][Bb][Aa][Ll][Ss][Mm][Ss]|[Mm][Aa][Ii][Ll]2[Ss][Mm][Ss]|[Mm][Aa][Ii][Ll][Tt][Ee][Xx][Tt]|[Rr][Ee][Dd][Aa][Ll][Ee][Rr][Tt]|[Mm][Tt][Rr][Aa][Dd][Ee]|[Vv][Oo][Tt][Ii][Nn][Gg]|[Mm][Ee][Nn][Tt][Ee][Rr][Pp][Rr][Ii][Ss][Ee]' for type 'CELLACT-APP'.	Invalid value in the application field
4.	cvc-enumeration-valid: Value 'send' is not facet-valid with respect to enumeration '[sendtextmt, sendlinkmt, sendringtonemt, sendpicturemt, sendsimplevcardmt, sendoperatorlogomt, sendivrmt, sendivr, sendiconmt, sendwaptextmt, sendbookmarkmt, send-ems-melody, send-ems-melody-predefined, send-motorola-melody, send-ceo-melody, send-ceo-bitmap, send-ceo-vcard, sendmms, sendgamemt, sendkaraokemt, provision, sendmail]'. It must be a value from the enumeration.	Invalid value in the command field
5.	cvc-maxInclusive-valid: Value 'XXX' is not facet-valid with respect to maxInclusive '65535' for type 'CELLACT-TTS'.	Invalid value in the TTS field
6.	cvc-maxInclusive-valid: Value 'XXX' is not facet-valid with respect to maxInclusive '65535' for type 'CELLACT-TTL'.	Invalid value in the TTL field
7.	cvc-maxLength-valid: Value 'XXX' with length = 'XX' is not facet-valid with respect to maxLength '50' for type 'CELLACT-STRING-0-50'.	Invalid value in the Sender field
8.	cvc-maxLength-valid: Value 'XXX' with length = 'XX' is not facet-valid with respect to maxLength '50' for type 'CELLACT-STRING-0-50'.	Invalid value in the Sender Name

#	Message in the < errorDescription > field	Description
		field
9.	cvc-pattern-valid: Value '' is not facet-valid with respect to pattern '05[0,2,3,4,5,6,7,8,9]-?[0-9]{7}|\+[0-9]{7,16}|([\+\.a-zA-Z0-9_-])+@([a-zA-Z0-9_-])+(([a-zA-Z0-9_-])*\.([a-zA-Z0-9_-])+)+' for type 'CELLACT-MOBILE-INTERNATIONAL-NUMBER'.	Invalid value in the DestinationMail field
10.	cvc-maxLength-valid: Value 'XXX' with length = 'XX' is not facet-valid with respect to maxLength '40' for type 'CELLACT-STRING-1-40'.	Invalid value in the destination messageId field
11.	cvc-maxLength-valid: Value 'XXX' with length = 'XX' is not facet-valid with respect to maxLength '50' for type 'CELLACT-STRING-1-50'.	Invalid value in the destination ServiceName field
12.	cvc-maxLength-valid: Value 'XXX' with length = 'XX' is not facet-valid with respect to maxLength '100' for type 'CELLACT-STRING-0-100'.	Invalid value in the destination subject field
13.	cvc-maxLength-valid: Value 'XXX' with length = 'XX' is not facet-valid with respect to maxLength '50' for type 'CELLACT-STRING-1-50'.	Invalid value in the templateName field
14.	cvc-maxLength-valid: Value 'XXX' with length = 'XXX' is not facet-valid with respect to maxLength '512' for type 'CELLACT-STRING-0-512'.	Invalid value in the templateId field
15.	cvc-maxLength-valid: Value 'XXX' with length = 'XXX' is not facet-valid with respect to maxLength '1000' for type 'CELLACT-STRING-0-1000'.	Invalid value in the Params field
16.	cvc-maxLength-valid: Value 'XXX' with length = 'XX' is not facet-valid with respect to maxLength '50' for type 'CELLACT-STRING-1-50'.	Invalid value in the campaignName field
17.	cvc-maxLength-valid: Value 'XXX' with length = 'XX' is not facet-valid with respect to maxLength '50' for type 'CELLACT-STRING-1-50'.	Invalid value in the replyTo field

6 Appendix

6.1 Format of the Destination Number

The Destination Number for the purpose of SMS Delivery must match one of the following regular expressions:

Pattern	Description
05[0,2,3,4,5,6,7,8,9]-?[0-9]{7}	1. 05x where x is one of the [0,2,3,4,5,6,7,8,9] 2. Optional "-" 3. 7 digits Example: 054-XXXXXXX or 054XXXXXXX
+[0-9]{7,16}	1. "+" 2. 7 to 16 digits Example (10 digits): +XXXXXXXXXX

6.2 Message Delivery Status (REASON) Codes

The following Codes are used to indicate the status of message delivery in the Progress/ Confirmation messages:

Cellact Proprietary

Note: The numbering convention is as follows: value X000 indicates success; any other value indicates an issue.

Table 11 Interpretation of the "Reason" Code

REASON	Description
1000	Message delivered
1002	Subscriber is unavailable
1003	Message inbox full
1005	Incorrect subscriber number
1101	Registration request parameters are missing/incorrect
1103	Password not authenticated
1104	Subscriber is already registered with this service
1105	Subscriber is not registered to this service
1106	Pending request in process
1107	Internal connection error
1108	System SQL Connection Error
1140	Invalid SIGI / Subscriber is not registered to this service
1141	TRID or SIGI is missing
2006	Destination is blocked to RB messages
2010	Subscriber is blocked
2013	Sender is invalid
2014	Sender from a different operator cannot be charged
2016	Destination does not support the message's type
2018	Unknown link
2020	Spam filter
2050	Subscriber block in Guardian
2060	Number is blocked for SMS reception
3001	Account balance has exceeded
3002	Message contains invalid attributes
3004	MNP Validator Error
3007	"Night Block"– account can't send messages in the predefined hours
4002	Prepaid system did not recognize message's Tariff
5000	Message was delivered to the operator
5001	Operator error
5007	Recipient is temporarily unavailable / Device inbox is full
5014	SQL: System connection error
5016	Message size exceeded

Cellact Proprietary

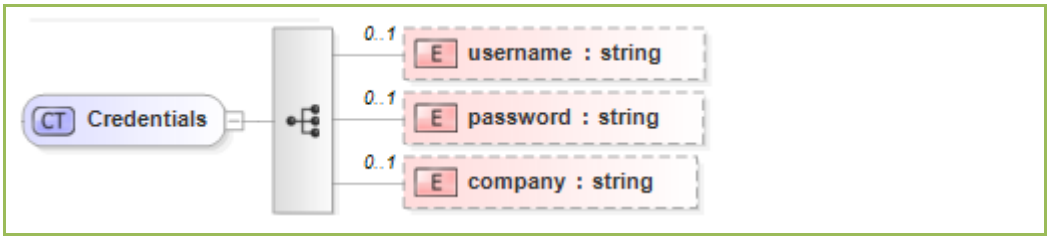
REASON	Description
5028	Error sending overseas message
6001	Communication error
7000	Subscriber replied to MT message
7001	Message not delivered
8000	WAP text message was opened
8018	MMS message rejected
9000	MO message delivered
9005	Subscriber has insufficient balance
9801	Email message was not delivered
9821	Recipient's Server is not available.
9831	Message declined by recipient
9840	Recipient is unavailable temporarily / Mail inbox is full
9845	Recipient is Not Found
9850	Invalid Email Address

6.3 Types and XML Structures

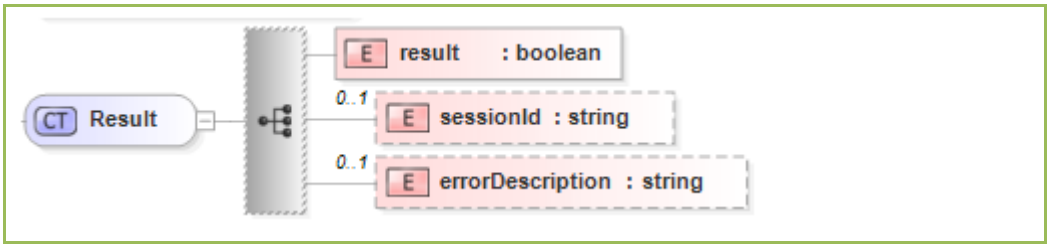
6.3.1 Enums



6.3.2 Credentials

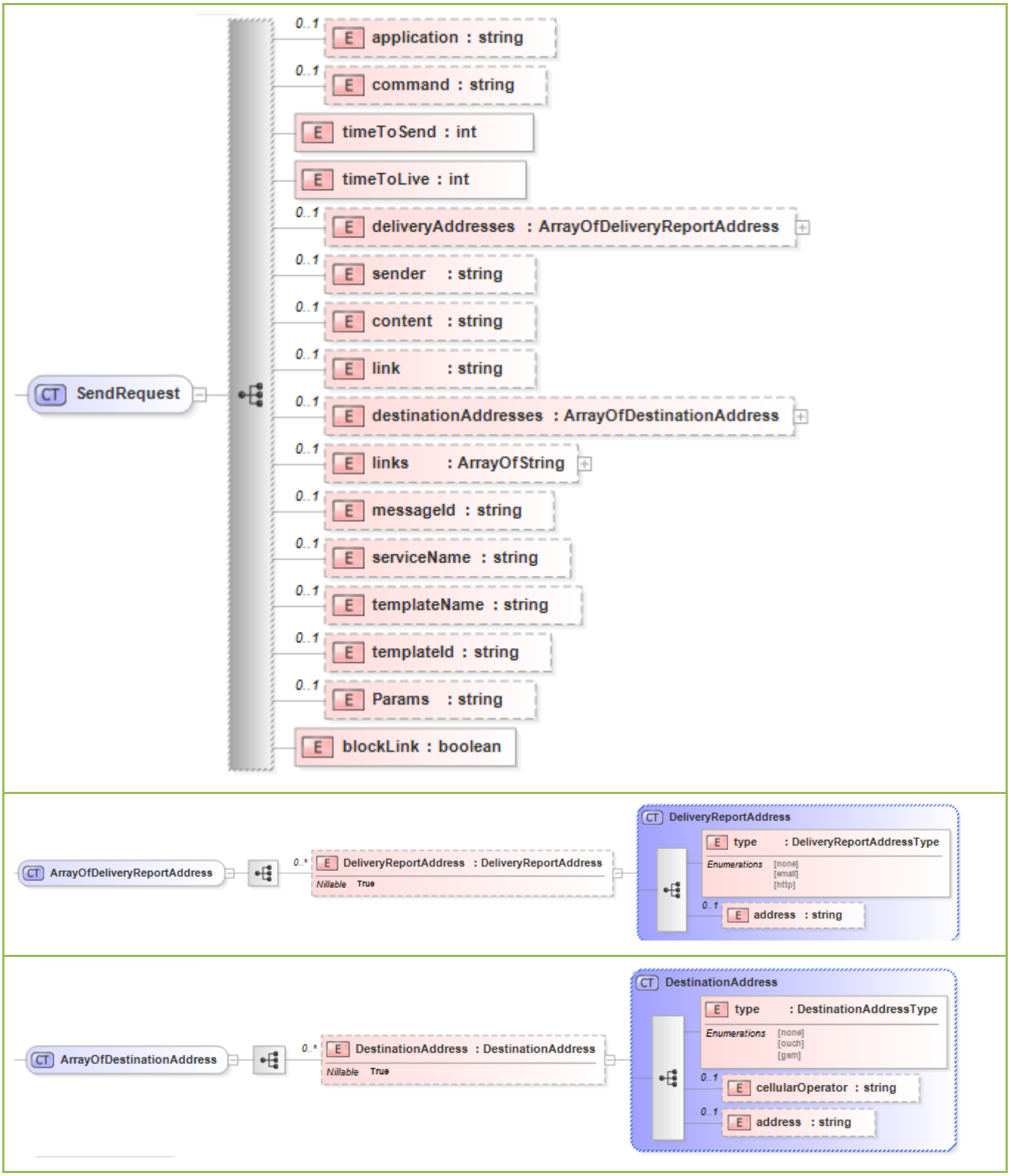


6.3.3 Result

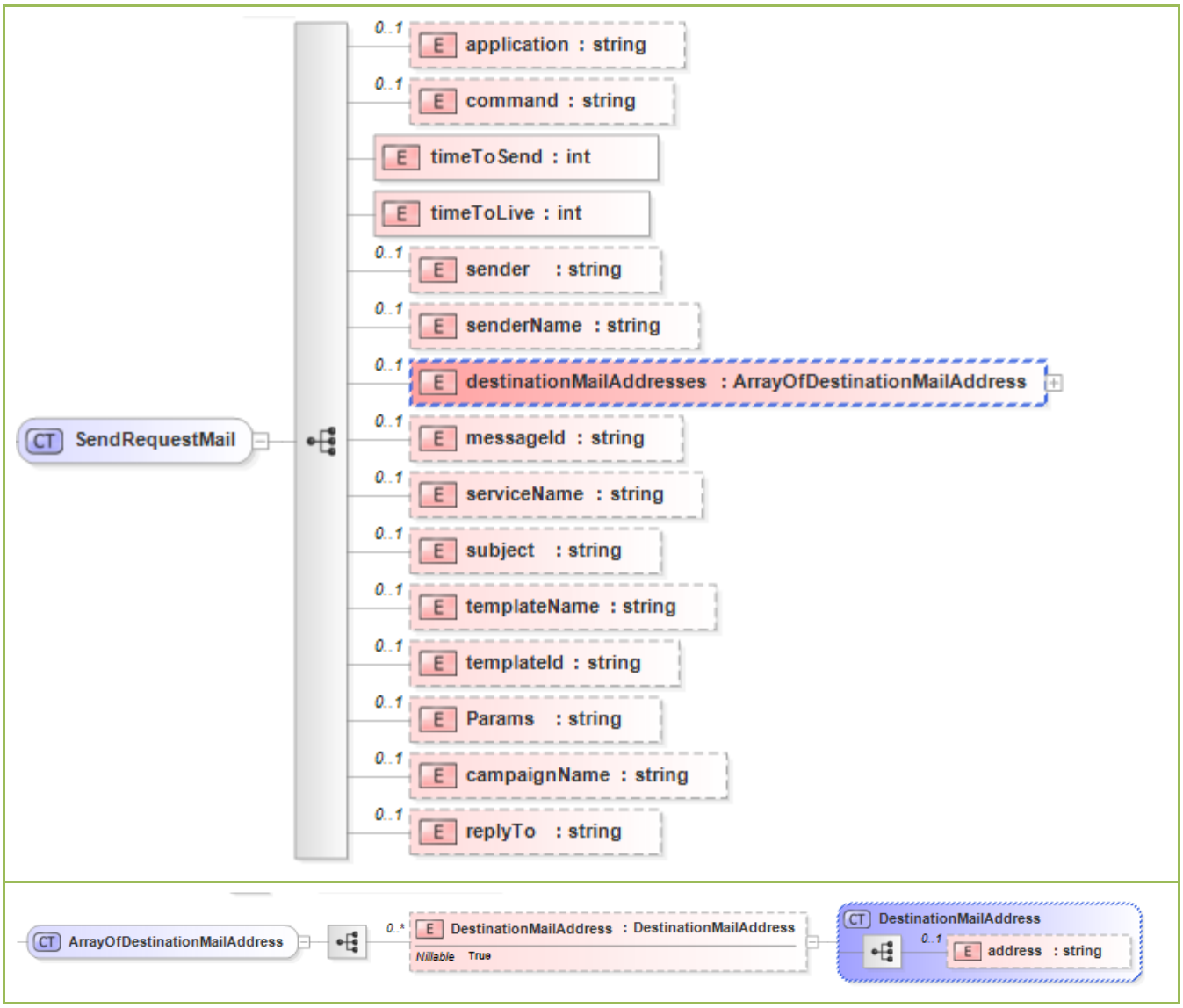


Cellact Proprietary

6.3.4 SendRequest



6.3.5 SendRequestMail



Cellact Proprietary

Contacting Cellact

Contact us at: mailto:support@cellact.com

Phone: +972-9-9704181

Cellact Ltd.

Shefayim Business Center

P.O. Box 286

Shefayim 60990 Israel

Tel: +972-9-970-4110

Fax: +972-9-970-4210

Visit our website:

https://www.cellact.com or https://www.cellact.co.il