

“The Smart SMS Platform For Growing Businesses. Reliability Guaranteed”

Here’s why you should be using directSMS for all your messaging needs

You will get the help you need from an expert outfit with thousands of satisfied customers across Australia

You will be up and running in no time

“directSMS’ customer service team have provided a high level of service and have been very responsive with all our support queries.”

*Eliot Harper
Production Workflow
Marketing Manager
Fuji Xerox Australia*

“directSMS customer service is very efficient. Whenever I have had to ask questions, I always got quick replies and they were very helpful”

*Greg Gubiani
Service Express Manager
The Westin Melbourne*

directSMS: REST API

REST API Reference Manual

Outline of directSMS’ Business Grade REST API.

March 2015



Table of Contents

Introduction	3
Features and functionality.....	4
Authenticating.....	5
Sending an SMS.....	5
1-way Messages	5
2-way Messages	5
Long SMS Support	6
Unicode SMS Support	6
Sample Requests - JSON.....	9
Success Output - JSON	10
Error Output - JSON.....	10
Scheduling an SMS	11
Sample Requests - JSON.....	12
Success Output - JSON	14
Error Output - JSON.....	14
Cancelling a Scheduled SMS	15
Sample Requests - JSON.....	15
Success Output - JSON	15
Error Output - JSON.....	16
Receiving Delivery Receipts	17
Example.....	18
Error Handling	18
Getting SMS replies	19
1. Email Push.....	19
2. HTTP Push	19
Example.....	20
Error Handling	20

3. Polling API Server	20
Sample Requests - JSON	21
Success Output - JSON	21
Error Output - JSON.....	22
Getting Inbound SMS	23
1. Email Push.....	23
2. HTTP Push	23
Example.....	23
Error Handling	24
3. Polling API Server	24
Sample Requests - JSON	24
Success Output - JSON	25
Error Output - JSON.....	26
Getting your balance	27
Sample Requests - JSON	27
Success Output - JSON	27
Error Output - JSON.....	27
How to get started.....	28
Partner Program	28
White-label Reseller.....	29

Introduction

Thank you for considering directSMS for your messaging needs. This document provides a reference manual for all features available via the REST interface on our API servers to send and receive SMS messages.

The REST API allows you to SMS enable your applications and systems by integrating them with directSMS API servers over the HTTP protocol. HTTPS is also supported for secure transactions using SSL encryption over the HTTP protocol.

The REST API will allow you to send and receive SMS messages by issuing different HTTP GET, PUT, POST and DELETE commands to our API servers.

Features and functionality

The directSMS API servers will allow you to perform the following:

1. Send 1-way and 2-way SMS messages.
2. Schedule the sending of SMS messages for future delivery.
3. Retrieve 2-way SMS replies and correlate each reply with its original message.
4. Retrieve Inbound SMS messages if you have your own dedicated inbound number.
5. Receive real time notifications when SMS messages are delivered to their destinations.
6. Receive notifications of reply or inbound SMS messages received in real time.
7. Send Unicode messages (Chinese, Korean, Russian, Hebrew, Arabic and so on).

Access to the API servers comes with Australia based phone and email support.

Please note, all parameter names and values mentioned in this documentation are **Case Sensitive**.

Authenticating

In order to use any API function, you will need to pass your security credentials with every request you make to the directSMS servers.

These are the same username/password credentials you use to log onto the directSMS customer portal.

Parameter	Description	Required
username	Username you use to log onto the directSMS customer portal	Y
password	Password for the given user account	Y
lic	Enterprise license key if you were issued one. <u>This is non mandatory</u>	N

Y – Mandatory, N – Optional, C - Conditional

Sending an SMS

The directSMS gateway can send either 1-way SMS messages or 2-way messages.

1-way Messages

1-way messages allow you to control the “sending number” or the “source address” of the SMS. You can set the source address to be:

1. Any Australian mobile phone number. If recipients reply, their replies are routed directly to this number.
2. Any virtual or inbound SMS number. If recipients reply, their replies are routed back to our system and are logged online and forwarded back to your systems via email or HTTP push if you wish.
3. Your business name. E.g. “XYZ”. In effect, these messages are 1-way notifications only and recipients are blocked from replying to them.

2-way Messages

2-way messages allow you to carry out conversational transactions with your message recipients. In this case, the directSMS gateway is in control of the “source address” or sending number on these SMS.

The gateway can then use this to correlate any replies received from the message recipients back to the original SMS you sent out.

This is very useful for transactional use cases such as the confirmation of appointments, surveys and similar.

Long SMS Support

A single SMS can only carry 160 characters. By default, if you attempt to send a message which is more than 160 characters, the server will return an error.

You can explicitly indicate that you would like a long message to be split into multiple SMS segments if it exceeds this 160 character limit by passing the optional parameter **max_segments**.

This parameter indicates the maximum number of SMS messages the server should use while sending out messages longer than 160 characters.

Each message segment is restricted to 153 characters in length, with 7 bytes used by the gateway for headers so the destination handset can piece the message segments back together again once they have been received.

Each message segment is equal to sending a single SMS.

Please note; a message will only be split into multiple segments if it is longer than 160 characters. If your message is under this limit and you specify a **max_segments** value, the message will not be split and you will only be charged for a single SMS.

max_segments	Max. SMS Messages Sent	Characters Available
1	1	160
2	2	305 (153 x 2)
3	3	459 (153 x 3)
4	4	612 (153 x 4)
.	.	.
N	N	153 x N

Unicode SMS Support

Unicode messages are fully supported. This means you can send messages containing any 16-bit characters like Chinese, Korean, Hindi, Hebrew, Arabic, Russian and so on.

For Unicode messages, your **UCS-2** encoded message text must be converted into Hexadecimal digits. Each UCS-2 character is made up of 2-bytes (16-bits). Each byte is converted into 2 hexadecimal digits.

For example, a message like “**Hello World - 你好世界**” must be converted into the following Hex digits: **00480065006C006C006F00200057006F0072006C00640020002D00204F60597D4E16754C**

Due to the increased character size, Unicode messages can only have a maximum of **70 Unicode characters**. If you need to send longer messages, you can make use of the **max_segments** parameter to have our systems break up the long message into multiple segments.

Please note that Unicode message segments can only hold **67 Unicode characters**. The rest of the message payload contains a User Data Header (UDH) containing message segmentation information. This header is used by the destination handset to put the original message together based on all the received segments.

Another note; your Unicode message will only be split into multiple segments if it is longer than 70 characters. If your message is under this limit and you specify a **max_segments** value, the message will not be split and you will only be charged for a single SMS.

max_segments	Max. SMS Messages Sent	Unicode Characters Available
1	1	70
2	2	134 (67 x 2)
3	3	201 (67 x 3)
4	4	268 (67 x 4)
.	.	.
N	N	67 x N

Please see <http://www.Unicode.org/> for more information.

Method: POST
 URL: <https://api.directsms.com.au/s3/rest/sms/send>

The basic parameters to send a message are outlined below.

Parameter	Description	Required
type	The gateway can send two types of messages: <ol style="list-style-type: none"> 1. 1-way SMS where you get to set the Sender ID on each message to value of your choosing. For example your own mobile number or your company name. 2. 2-way SMS where our system will set the Sender ID and allow people receiving your message to reply back. Our system will track all 2-way replies and correlate them back to 	Y

your original SMS		
<u>Valid values are "1-way" or "2-way" only</u>		
to	<p>The list of mobile numbers you want to send this message to separated by “,”.</p> <p>Please keep in mind that you are sending a HTTP POST so best to keep the number of mobile numbers down to about 300 or so per request.</p>	Y
message	The message contents you want to deliver	Y
senderId	<p>This is the Sender ID on the message. This can be set to your own mobile number, your inbound number or company name.</p> <p>This is required on 1-way SMS only</p>	C
messageId	<p>This is a message identifier for a 2-way SMS that identifies this message in your system. When any replies are received for this message, this message key will be returned back to you in order for you to correlate the replies back to the original message.</p> <p>The messageid is limited to 12 characters in length. This is very useful for applications like order or appointment confirmations.</p> <p>This is used when sending 2-way SMS only</p>	N
maxSegments	<p>This option enables the splitting of a message if the number of characters exceeds 160. It specifies the maximum number of SMS segments to split the message into.</p> <p>Each segment carries a 153 character payload. The remaining 7 characters contains a User Data Header (UDH) which contains the concatenation information to be used by the destination handset to pull all message segments back together.</p>	N
unicode	This option tells the API servers that your message text is made up of 16-bit Unicode	N

characters.

Please note, in order to submit the message successfully, your UCS-2 encoded message text must be converted into Hexadecimal digits first.

By default, this parameter is set to “false”.

Y – Mandatory, N – Optional, C – Conditional

Sample Requests - JSON

The following will send a 1-way SMS with a Sender ID of “directSMS” to 2 recipients 61411000111 and 61411000222.

```
curl -i \  
-X POST \  
-H "Content-Type: application/json" \  
-H "Accept: application/json" \  
-H "Username: Your directSMS username" \  
-H "Password: Your directSMS password" \  
-d '{  
    "messageType": "1-way",  
    "senderId": "directSMS",  
    "messageText": "Test Message",  
    "to": ["61411000111", "61411000222"]  
}' \  
-s https://api.directsms.com.au/s3/rest/sms/send
```

The following will send a 2-way SMS with the Message Reference of “ID112”. ID112 might be the transaction or reference number of an appointment or similar entity in your system. This value will be returned when a reply to this message is received to this message.

```
curl -i \  
-X POST \  
-H "Content-Type: application/json" \  
-H "Accept: application/json" \  
-H "Username: Your directSMS username" \  
-H "Password: Your directSMS password" \  
-d '{  
    "messageType": "2-way",  
    "messageId": "ID112",  
    "messageText": "Test Message",  
    "to": ["61411000111", "61411000222"]  
}' \  
-s https://api.directsms.com.au/s3/rest/sms/send
```

Success Output - JSON

Upon successful submission of a message to the server, an identifier that uniquely identifies the message will be returned along with the message structure you submitted in a JSON array. The ID returned is 32 characters in length.

If the message submitted was converted into multiple concatenated SMS, the array returned will contain all SMS segments and their IDs.

The example below shows the response payload returned including the “id” parameter which identifies the message batch in directSMS’ systems.

If the message submitted is converted into multiple SMS segments, multiple message objects with different “id” property values will be returned.

```
[
  {
    "id": "1b71f477ef4bc6c50213642109fc537a",
    "messageType": "1-way",
    "senderId": "directSMS",
    "messageText": "Test Message",
    "to": ["61411000111", "61411000222"]
  }
]
```

Error Output - JSON

If the parameters submitted are invalid for any reason, an error message will be returned instead. Some errors will be conveyed through the HTTP Status and return code.

As with all REST APIs, any time a response containing anything other than a HTTP 400 – OK is an error.

Below is what the payload may look like when an error occurs. In this case, the username and/or password submitted to the server were incorrect.

```
{
  "httpStatus": "500",
  "errorCode": "300",
  "errorMessage": "Invalid login credentials"
}
```

Scheduling an SMS

The gateway can schedule SMS messages for delivery in the future. This can be very helpful for set and forget type messaging for client applications without the ability to schedule the sending of messages themselves.

This might be because the application is running on a device that might be off at the time you wish to send the message or because it has no scheduling capability of its own.

Method: POST
URL: <https://api.directsms.com.au/s3/rest/sms/schedule>

The basic parameters required to schedule a message for future delivery are listed below.

Parameter	Description	Required
type	<p>The gateway can send two types of messages:</p> <ol style="list-style-type: none">1. 1-way SMS where you get to set the Sender ID on each message to value of your choosing. For example your own mobile number or your company name.2. 2-way SMS where our system will set the Sender ID and allow people receiving your message to reply back. Our system will track all 2-way replies and correlate them back to your original SMS <p>Valid values are "1-way" or "2-way" only</p>	Y
to	<p>The list of mobile numbers you want to send this message to separated by ",".</p> <p>A maximum of 100 mobile numbers are possible, if you are sending the request via a HTTP GET. Alternatively, you can set a maximum of 300 mobile numbers if you are sending the request via HTTP POST</p>	Y
message	The message contents you want to deliver	Y
senderId	This is the Sender ID on the message. This can be	C

set to your own mobile number, your inbound number or company name.

This is required on 1-way SMS only

messageId	<p>This is a message identifier for a 2-way SMS that identifies this message in your system. When any replies are received for this message, this message key will be returned back to you in order for you to correlate the replies back to the original message</p> <p>The messageId is limited to 12 characters in length. This is very useful for applications like order or appointment confirmations</p> <p>This is used when sending 2-way SMS only</p>	N
scheduledDate	<p>The date and time to send this SMS. This value must be in the future.</p> <p>You can use the number of milliseconds since the Unix Epoch (1 January, 1970, 00:00:00 in GMT). For example 1427886000 will schedule the message on April 1, 2015 at 11AM in Greenwich Mean Time (GMT).</p> <p>Alternatively, you can use JavaScript's Date notation which uses Coordinated Universal Time (UTC) e.g. "2015-04-01T11:00:00.000Z" will schedule the message on April 1, 2015 at 11AM in Greenwich Mean Time (GMT).</p>	Y

Y – Mandatory, N – Optional, C - Conditional

Sample Requests - JSON

The following will schedule a 1-way SMS with a Sender ID of "directSMS" to 2 recipients 61411000111 and 61411000222 to go out on April 1, 2015 at 11AM GMT.

```
curl -i \  
-X POST \  
-H "Content-Type: application/json" \  
-H "Accept: application/json" \  
-H "Username: Your directSMS username" \  
-H "Password: Your directSMS password" \  
-d '{  
    "messageType": "1-way",
```

```
"senderId": "directSMS",
"messageText": "Happy April Fools Day!",
"to": ["61411000111", "61411000222"],
"scheduledDate": "2015-04-01T11:00:00.000Z"
}'\
-s https://api.directsms.com.au/s3/rest/sms/schedule
```

Here is the same exact request scheduled by passing the number of milliseconds since the Unix Epoch instead of formatting the scheduled date in JavaScript (UTC) notation.

Note that in the following example, the `scheduledDate` parameter is passed as an integer (no quote characters).

```
curl -i \
-X POST \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-H "Username: Your directSMS username" \
-H "Password: Your directSMS password" \
-d '{
  "messageType": "1-way",
  "senderId": "directSMS",
  "messageText": "Happy April Fools Day!",
  "to": ["61411000111", "61411000222"],
  "scheduledDate": 1427886000
}'\
-s https://api.directsms.com.au/s3/rest/sms/schedule
```

The following will schedule a 2-way SMS with the Message Reference of "ID112". ID112 might be the transaction or reference number of an appointment or similar entity in your system. This value will be returned when a reply to this message is received to this message.

```
curl -i \
-X POST \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-H "Username: Your directSMS username" \
-H "Password: Your directSMS password" \
-d '{
  "messageType": "2-way",
  "messageId": "ID112",
  "messageText": "Happy April Fools Day!",
  "to": ["61411000111", "61411000222"],
  "scheduledDate": "2015-04-01T11:00:00.000Z"
}'\
-s https://api.directsms.com.au/s3/rest/sms/schedule
```

Success Output - JSON

Upon successful submission of a message to the server, an identifier that uniquely identifies the message will be returned along with the message structure you submitted. The ID returned is 32 characters in length.

The example below shows the response payload returned including the “id” parameter, which identifies the newly scheduled message in directSMS’ systems.

Additionally, the `scheduledDate` parameter returned contains the timestamp that the message WILL actually be sent at. Our servers will approximate the timestamp value submitted in your request to the nearest 10 minutes.

This value can be returned as the number of milliseconds since the Unix Epoch or in JSON (UTC) notation complete with timezone.

```
{
  "id": "1b71f477ef4bc6c50213642109fc537a",
  "messageType": "1-way",
  "senderId": "directSMS",
  "messageText": "Happy April Fools Day!",\
  "to": ["61411000111", "61411000222"],\
  "scheduledDate": 1427886000000\
}
```

Error Output - JSON

If the parameters submitted are invalid for any reason, an error message will be returned instead. Some errors will be conveyed through the HTTP Status and return code. Generally anything other than a HTTP 400 – OK is an error.

Below is what the payload may look like when an error occurs. In this case, the username and/or password submitted to the server were incorrect.

```
{
  "httpStatus": "500",
  "errorCode": "300",
  "errorMessage": "Invalid login credentials"
}
```

Canceling a Scheduled SMS

The gateway allows you to cancel any SMS messages that are scheduled for future delivery.

This can be very helpful for appointment confirmation type and similar applications where you schedule a reminder message but then need to cancel it once the appointment is rescheduled.

In order to cancel a scheduled message you need to specify the ID returned by the gateway when the message was first created and scheduled.

Please note, only the user account used to create the original scheduled message is allowed to cancel it. You can NOT schedule a message as user1 and cancel it as user2.

Method: DELETE
URL: <https://api.directsms.com.au/s3/rest/sms/deschedule/{id}>

This operation accepts one parameter only. This parameter is passed as part of the URL called.

Parameter	Description	Required
id	This is the 32 character identifier returned by the gateway in response to your original call to the "schedule message" operation	Y

Y – Mandatory, N – Optional, C – Conditional

Sample Requests - JSON

The following will cancel a scheduled message with ID of `1b71f477ef4bc6c50213642109fc537a`.

```
curl -i \  
-X DELETE \  
-H "Content-Type: application/json" \  
-H "Accept: application/json" \  
-H "Username: Your directSMS username" \  
-H "Password: Your directSMS password" \  
-s  
https://api.directsms.com.au/s3/rest/sms/deschedule/1b71f477ef4bc6c50213642109fc537a
```

Success Output - JSON

Upon successful submission, the message identified should be removed from the system and the ID will be echoed back.

```
1b71f477ef4bc6c50213642109fc537a
```

Error Output - JSON

If the parameters submitted are invalid for any reason, an error message will be returned instead. Some errors will be conveyed through the HTTP Status and return code. Generally anything other than a HTTP 400 – OK is an error.

Below is what the payload may look like when an error occurs. In this case, the username and/or password submitted to the server were incorrect.

```
{
  "httpStatus": "500",
  "errorCode": "300",
  "errorMessage": "Invalid login credentials"
}
```


Receiving Delivery Receipts

The gateway can send a HTTP GET/POST in real time to your server whenever a Delivery Receipt (DR) is processed for one of the messages you sent earlier. Delivery receipts are short messages sent from the networks denoting the successful or otherwise delivery of an SMS message.

You can turn this notification mechanism on and control the URL the gateway will push the DR notification messages to by updating the User Profile page on the directSMS Customer Portal. You will receive each DR notification in real time in a separate HTTP request.

The parameters in the HTTP GET/POST that are sent to your server are as follows:

Parameter	Description
id	This is the identifier of the original message that this DR is correlated to within directSMS' systems e.g. 6df259577165fd4c7fa6ba0cc8fa41d5
type	The type of the original sent message e.g. 1-way or 2-way
mobile	The mobile number message was delivered to. This will be presented in International format e.g. 61412345678 instead of 0412 345 678
status	<p>The status of the message in question. The different values are as follows:</p> <p>DELIVRD – Message was delivered successfully</p> <p>UNDELIV – Message was not delivered</p> <p>EXPIRED – The gateway attempted to send the message for 48 hours but was not able to deliver it because the handset was turned off or not in coverage</p> <p>REJECTD - The message was rejected by the network and was not delivered. This may happen if your messages are deemed as SPAM or if the subscriber does not exist. This depends on the destination network and whether they run a overt vs. covert filtering scheme</p> <p>DELETED - The message was aborted by the network because the it is a duplicate or the subscriber is invalid</p>
when	The number of seconds since the message identified in the DR was received. For example, if the message was received by the handset in question 10 seconds ago, this value will be 10

Example

1. On the User Profile page, you have selected to send a HTTP push to the URL http://www.mywebsite.com/process_delivery_receipt?security_option=xyz when a new delivery receipt is received.
2. One of your 1-way messages has been delivered to its destination 10 seconds earlier.
3. The gateway will make the following HTTP GET call to your server:
http://www.mywebsite.com/process_delivery_receipt?security_option=xyz&id=6df259577165fd4c7fa6ba0cc8fa41d5&type=1-way&mobile=61444123123&status=DELIVRD&when=10

Error Handling

If for any reason your server does not respond with a **200 OK** HTTP response code, the HTTP Push will be deemed a failure and the gateway will retry to push this DR notification again at a later time.

The gateway retries failed notifications every 20 minutes for between 1 and 2 hours. After this, the HTTP push is aborted for the given DR.

Getting SMS replies

You can receive reply messages to 2-way SMS messages sent out earlier in 3 ways:

1. *Email Push*

The gateway can send you an email in real time when a new reply message is received. The email will contain the original 2-way message, as well as the reply that was just received. To turn this notification mechanism on, you need to update your User Profile by logging onto directSMS' Customer Portal using your username and password.

Please note, the email will be sent to the email address you have nominated under your user profile.

2. *HTTP Push*

The gateway can send a HTTP GET/POST in real time to your server whenever a reply message is received. You can turn this notification mechanism on and control the URL the gateway will push the messages to by updating the User Profile page on the directSMS Customer Portal. This is the recommended way to receive reply messages. You will receive each reply in real time in a separate HTTP request.

The parameters in the HTTP GET/POST that are sent to your server are as follows:

Parameter	Description
id	This is the identifier of the original 2-way message that this reply is correlated to within directSMS' systems e.g. 6df259577165fd4c7fa6ba0cc8fa41d5
message_text	The content of the SMS message. This will be URL encoded
mobile	The mobile number of the person replying. This will be presented in International format e.g. 61412345678 instead of 0412 345 678
when	The number of seconds since this SMS was received. For example, if the message was received by directSMS' gateway 2 seconds ago, this value will be 2
messageid	This is the message identifier of the original 2-way SMS that identifies that message in your system, <u>if one was supplied when the original 2-way SMS was sent out</u>

Example

1. On the User Profile page, you have selected to send a HTTP push to the URL http://www.mywebsite.com/process_reply_sms?security_option=xyz when a new 2-way reply message is received.
2. One of your clients replies to your 2-way message from earlier today to confirm their appointment at 3PM with the message “Yes I will be there”. The original 2-way message sent to this client is identified as “APP1234” in your system.
3. The gateway will make the following HTTP GET call to your server:
http://www.mywebsite.com/process_sms?security_option=xyz&id=6df259577165fd4c7fa6ba0cc8fa41d5&message_text=Yes+I+will+be+there&mobile=61444123123&messageid=APP1234&when=0

Error Handling

If for any reason your server does not respond with a **200 OK** HTTP response code, the HTTP Push will be deemed a failure and the gateway will retry to push this message again at a later time.

The gateway retries failed messages at 20 minute intervals for between 4 and 8 hours. After this, the HTTP push is aborted for that message.

3. Polling API Server

The API server can fetch the replies for all or a given 2-way message depending on the parameters you pass during the call. This operation also takes an optional parameter to mark the replies as read in order to ensure you do not retrieve them again later.

This is the least favoured method of retrieving replies.

PLEASE NOTE: Polling for replies excessively will see your account suspended. The most efficient solution is to use the HTTP Push feature where reply messages are pushed to your server as they are received in real time.

Method: GET
URL: <https://api.directsms.com.au/s3/rest/sms/reply>

This operation accepts one parameter only. This parameter is passed as part of the URL called as a query string parameter. E.g. <https://api.directsms.com.au/s3/rest/sms/reply?markAsRead=true>

Parameter	Description	Required
markAsRead	Mark the returned reply messages as read, to avoid returning them again in future. By default, this value is set to true .	N

Y – Mandatory, N – Optional, C - Conditional

Sample Requests - JSON

The following request will return all reply messages received for the given user account, as well as mark those messages as read. This will stop the system from returning them in subsequent calls.

```
curl -i \  
-X GET \  
-H "Content-Type: application/json" \  
-H "Accept: application/json" \  
-H "Username: Your directSMS username" \  
-H "Password: Your directSMS password" \  
-s https://api.directsms.com.au/s3/rest/sms/reply
```

Success Output - JSON

Upon successful submission, all the unread reply messages are sent back in a JSON array. Please see the sample output below.

Please note the replyDate parameter is returned in JSON (UTC) date format complete with timezone offset.

```
[  
  {  
    "id": "93f7fc364b96481be07db82e3913ed76",  
    "mobile": "61407263977",  
    "messageId": "Appointment123",  
    "messageText": "Yes, I will be attending",  
    "replyDate": "2014-11-11T13:34:10.325+1100"  
  },  
  {  
    "id": "b0d9e8aaf21148ddf27127aae3df88e0",  
    "mobile": "61407263977",  
    "messageText": "This is a test reply",  
    "replyDate": "2014-11-09T17:50:44.325+1100"  
  },  
  {  
    "id": "b0d9e8aaf21148ddf27127aae3df88e0",  
    "mobile": "61407263977",  
    "messageText": "And another reply",  
    "replyDate": "2014-11-09T17:50:49.325+1100"  
  }  
]
```

Each reply message is represented as a JSON object with the following fields:

Field	Description
id	ID of the original 2-way message

messageId	This is the message identifier you passed when you sent the original 2-way message. This is only present if you submitted the original 2-way message with a messageId
mobile	The mobile number that sent this reply message in the format 61412345678
messageText	The message payload
replyDate	The date and time the reply message was received

Error Output - JSON

If the parameters submitted are invalid for any reason, an error message will be returned instead. Some errors will be conveyed through the HTTP Status and return code. Generally anything other than a HTTP 400 – OK is an error.

Below is what the payload may look like when an error occurs. In this case, the username and/or password submitted to the server were incorrect.

```
{
  "httpStatus": "500",
  "errorCode": "300",
  "errorMessage": "Invalid login credentials"
}
```

Getting Inbound SMS

If you have a dedicated inbound SMS number, you can retrieve your inbound SMS messages in three ways.

1. Email Push

The gateway can send you an email in real time when the message is received. You can control the email address the gateway will forward messages to through the Inbound SMS settings page on the directSMS Customer Portal.

2. HTTP Push

The gateway can send a HTTP GET/POST in real time when the message is received to your server. You can control the URL the gateway will push the message to through the Inbound SMS settings page on the directSMS Customer Portal.

This is the recommended way to receive messages. You will receive them in real time.

The parameters in the HTTP GET/POST that are sent to your server are as follows:

Parameter	Description
inbound_number	This will be the inbound number you are leasing from directSMS, this will be represented in International format e.g. 61429007007
id	This is the identifier of the inbound message within directSMS' systems e.g. 6df259577165fd4c7fa6ba0cc8fa41d5
message_text	The content of the SMS message. This will be URL encoded
mobile	The mobile number of the SMS message sender. This will be presented in International format e.g. 61412345678 instead of 0412 345 678
when	The number of seconds since this SMS was received. For example, if the message was received by directSMS' gateway 2 seconds ago, this value will be 2

Example

- On the Inbound SMS page, you have selected to send a HTTP push to the URL http://www.mywebsite.com/process_sms?security_option=xyz when a new inbound message is received.

5. A customer has decided to contact you via SMS on your inbound number 0428 001 001. The customer sends the SMS "This is John Citizen, please provide more info".
6. The gateway will make the following HTTP GET call to your server:
http://www.mywebsite.com/process_sms?security_option=xyz&id=6df259577165fd4c7fa6ba0cc8fa41d5&inbound_number=61428001001&message_text=This+is+John+Citizen,+please+provide+more+info&mobile=61444123123&when=0

Error Handling

If for any reason your server does not respond with a **200 OK** HTTP response code, the HTTP Push will be deemed a failure and the gateway will retry to push this message again at a later time.

The gateway retries failed messages at 20 minute intervals for between 4 and 8 hours.

3. Polling API Server

Alternatively, you can poll the API server periodically looking for the messages that have been received on your inbound number of choice or all inbound numbers if you have more than one number.

Again, this is the least recommended of the methods available due to the load it places on both client and server.

PLEASE NOTE: Polling for received messages excessively will see your account suspended. The most efficient solution is to use the HTTP Push feature where messages are pushed to your server as they are received in real time.

Method: GET
URL: <https://api.directsms.com.au/s3/rest/sms/inbound>

This operation accepts one parameter only which is passed as part of the URL (a query string parameter). E.g. <https://api.directsms.com.au/s3/rest/sms/inbound?markAsRead=true>

Parameter	Description	Required
markAsRead	Mark the returned reply messages as read, to avoid returning them again in future. By default, this value is set to true .	N

Y – Mandatory, N – Optional, C – Conditional

Sample Requests - JSON

The following request will fetch all new messages received on 61429815038 as well as mark them as read.


```
curl -i \  
-X GET \  
-H "Content-Type: application/json" \  
-H "Accept: application/json" \  
-H "Username: Your directSMS username" \  
-H "Password: Your directSMS password" \  
-s https://api.directsms.com.au/s3/rest/sms/inbound/61429815038
```

Success Output - JSON

Upon successful submission, all the unread messages are sent back in a JSON array. Please see the sample output below.

Please note that the receivedDate field is returned in JSON (UTC) date format.

```
[  
  {  
    "id": "f111deca8d78e1427e5641095be9eb8f",  
    "mobile": "61417583070",  
    "inboundNumber": "61429815038",  
    "messageText": "Hello... Is it me you're looking for?",  
    "receivedDate": "2014-06-04T08:10:24.584+1000"  
  },  
  {  
    "id": "6119cc2abefec26d668217b8ffa8df23",  
    "mobile": "61413495900",  
    "inboundNumber": "61429815038",  
    "messageText": "Enter me into your competition",  
    "receivedDate": "2014-06-04T11:01:21.584+1000"  
  },  
  {  
    "id": "3c4ad52478c0969f6a1d72f832575569",  
    "mobile": "61427798558",  
    "inboundNumber": "61429815038",  
    "messageText": "How do I enter this comp?",  
    "receivedDate": "2014-06-04T15:01:18.584+1000"  
  },  
]
```

Each reply message is represented as a JSON object with the following fields:

Field	Description
id	ID of this mobile originated message
inboundNumber	The virtual number this message was received on
mobile	The mobile number that sent this reply message in the format

61412345678

messageText The message payload

receivedDate The date and time the reply message was received

Error Output - JSON

If the parameters submitted are invalid for any reason, an error message will be returned instead. Some errors will be conveyed through the HTTP Status and return code. Generally anything other than a HTTP 400 – OK is an error.

Below is what the payload may look like when an error occurs. In this case, the username and/or password submitted to the server were incorrect.

```
{
  "httpStatus": "500",
  "errorCode": "300",
  "errorMessage": "Invalid login credentials"
}
```

Getting your balance

You can fetch your pre-paid SMS credit balance from the server at any time by calling the `get_balance` operation.

Method:	GET
URL:	https://api.directsms.com.au/s3/rest/sms/balance

This operation takes no parameters or JSON objects as input

Sample Requests - JSON

The following will enquire on your account's pre-paid credit balance.

```
curl -i \  
-X GET \  
-H "Content-Type: application/json" \  
-H "Accept: application/json" \  
-H "Username: Your directSMS username" \  
-H "Password: Your directSMS password" \  
-s https://api.directsms.com.au/s3/rest/sms/balance
```

Success Output - JSON

Upon successful submission, the message identified should be removed from the system and the ID will be echoed back.

```
3312.0
```

Error Output - JSON

If the parameters submitted are invalid for any reason, an error message will be returned instead. Some errors will be conveyed through the HTTP Status and return code. Generally anything other than a HTTP 400 – OK is an error.

Below is what the payload may look like when an error occurs. In this case, the username and/or password submitted to the server were incorrect.

```
{  
  "httpStatus": "500",  
  "errorCode": "300",  
  "errorMessage": "Invalid login credentials"  
}
```

How to get started

There are four simple steps to get started:

Step 1: Register online through our website. The system will give you some free credits to trial our services.

Step 2: Log in using the username/password sent to you.

Step 3: Email our support team to enable API access on your new account. Once activated, your account will have API access so you can test your solution.

Step 4: Start integrating your application, website or system.

If you require any more information or help, please call us on 1300 724 387 or email support@directsms.com.au and we'll be happy to assist.

If you are a software vendor, speak to us about our Partner Program or about becoming a Whitelabel Reseller. Please see details below.

Support Team

directSMS Pty Ltd

Email: support@directsms.com.au

Phone: 1300 724 387

Fax: 02 8569 0306

Web: www.directsms.com.au

Partner Program

The directSMS Partner Program is aimed at software vendors who wish to add SMS enabled features to their software offerings.

For each customer you introduce to directSMS through this Partner program, you will receive a generous trailing commission based on the revenue generated by that customer for the lifetime of their account.

As well as offering your customers the SMS features they need in your software, you can have the confidence that they will be backed by local-based technical and customer support. Not to mention a 100% money back reliability guarantee.

Best of all, directSMS will pay you a commission on each transaction as well as take care of all your clients' SMS related customer service and billing. What are partners for at the end of the day?

Contact us for more information.

White-label Reseller

The white-label solution is aimed at software vendors who wish to SMS enable their applications while generating revenue from their clients' SMS services. Add profits to your software as an SMS services reseller.

Brand your own reseller sub-domain with your own logo and corporate colours. Your clients will not know that directSMS is providing the underlying service. We will be your perfect silent partner.

As a white-label reseller, you can resell using the method that is most convenient for your organisation:

1. You can avoid all billing and paperwork headaches by having our systems bill your clients on your behalf. All you do is set your own prices and collect the margin you set above our low wholesale rates.
2. Alternatively, you can go "full service" where we bill you for all your clients' usage each month. You can then bill your clients any way you see fit.

Ultimately, the choice is yours.

Contact us for more information.