

“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: HTTP API

HTTP API Reference Manual

Outline of directSMS’ Business Grade HTTP API.

August 2014



Table of Contents

Introduction	4
Features and functionality	4
Authenticating	5
Using Connection ID vs. Username/Password	5
Success Output	5
Error Output	5
Sample Request	6
Sample Response	6
Sending an SMS	7
Long SMS Support	7
Unicode SMS Support	8
Success Output	10
Error Output	10
Sample Request	11
Sample Response	11
Scheduling an SMS	12
Success Output	13
Error Output	14
Sample Request	14
Sample Response	14
Cancelling a Scheduled SMS	15
Success Output	16
Error Output	16
Sample Request	16
Sample Response	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
Success Output	21
Error Output	22
Sample Request	22
Sample Response	22
Getting Inbound SMS	23
1. Email Push	23
2. HTTP Push	23
Example	23
Error Handling	24
3. Polling API Server	24
Success Output	25
Error Output	26
Sample Request	26
Sample Response	26
Getting your balance	27
Success Output	27
Error Output	27
Sample Request	27
Sample Response	28
Disconnecting	29
Success Output	29

Error Output.....	29
Sample Request.....	29
Sample Response	29
How to get started.....	30
Partner Program	30
White-label Reseller.....	31

Introduction

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

The HTTP 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 HTTP API will allow you to send and receive SMS messages by issuing different HTTP POST or GET commands to our API server.

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.

Authenticating

Before you can execute any commands on the API server, you will need to authenticate using your username and password.

The **connect** command will return a Connection ID to be used with all subsequent calls in place of your username and password. This operation is typically called at the start of every client session.

The Connection ID returned will only be active for 4-8 hours. Using an inactive or stale connection will result in the server returning error messages.

Using Connection ID vs. Username/Password

If you are going to be sending multiple messages or executing multiple commands in succession, it's more efficient to call **connect** first to obtain a Connection ID followed by the other commands. It saves the servers from having to validate the username/password passed in of your calls.

If you are going to send messages sporadically, it's easier to pass your username/password instead of the extra steps required to obtain a Connection ID first.

URL: <http://api.directsms.com.au/s3/http/connect>

Parameter	Description	Required
username	Username you use to log onto the directSMS customer portal	Y
password	Password for the given user account	Y
lic_key	Enterprise license key if you were issued one	N

Y – Mandatory, N – Optional, C - Conditional

Success Output

Upon authenticating your security credentials, a connection identifier is returned which is to be used for subsequent calls. The ID returned is 32 characters in length.

```
id: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Error Output

If your credentials are invalid or your account does not have API access enabled. An error message will be returned instead.

```
err: Error message
```

Sample Request

http://api.directsms.com.au/s3/http/connect?username=my_user&password=secret

Sample Response

id: 4c2c7d229b4d56bf249e9f6192cb7c11

Sending an SMS

Once you have called the connect operation and obtained a Connection ID, you can send SMS messages. The gateway can send either 1-way SMS messages or 2-way messages.

Alternatively, if you are sending messages sporadically, you can provide your username and password (and license key) in place of the Connection ID to identify your account when you come to send an SMS.

The second method is slightly slower because the server has to do two operations. It firstly authenticates the security credentials passed in against your account and then sends the message.

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 characters 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: **00480065006C006F00200057006F0072006C00640020002D0020204F60597D4E16754C**

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.

URL: http://api.directsms.com.au/s3/http/send_message

Parameter	Description	Required
connectionid	The Connection ID returned by the call to the connect operation at the beginning of your session. If you cannot call the connect operation first, you	Y

can pass the authentication parameters in place of the **connectionid**.

The authentication parameters are:

username – The username used to log onto directSMS' customer portal (Mandatory)

password – The password for the given user account (Mandatory)

lic_key – Enterprise license key if one was issued to you (Optional)

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	<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	This is a message identifier for a 2-way SMS that	N
------------------	---	---

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.

The messageid is limited to 12 characters in length. This is very useful for applications like order or appointment confirmations.

This is used when sending 2-way SMS only

max_segments	<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 153 characters and concatenation header (UDH) used by the destination handset to pull all message segments back together.</p>	N
unicode	<p>This option tells the API servers that your message text is made up of 16-bit Unicode characters.</p> <p>Please note, in order to submit the message successfully, your <u>UCS-2</u> encoded message text must be converted into <u>Hexadecimal digits</u>.</p> <p>Any value set against this parameter will be treated as true</p>	N

Y – Mandatory, N – Optional, C - Conditional

Success Output

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

id: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Error Output

If the parameters submitted are invalid for any reason, an error message will be returned instead.

err: Error message

Sample Request

The following will send a 1-way SMS with a Sender ID of “directSMS” using the Connection ID returned by the server when the connect operation was called.

http://api.directsms.com.au/s3/http/send_message?connectionid=4c2c7d229b4d56bf249e9f6192cb7c11&message=This+is+a+test&senderid=directSMS&to=61412345678&type=1-way

The following will authenticate and send a 1-way SMS with a Sender ID of “directSMS” in one request.

http://api.directsms.com.au/s3/http/send_message?username=my_user&password=secret&message=This+is+a+test&senderid=directSMS&to=61412345678&type=1-way

The following will send a 2-way SMS with the Message Key of “ID112”.

http://api.directsms.com.au/s3/http/send_message?connectionid=4c2c7d229b4d56bf249e9f6192cb7c11&message=This+is+a+test&messageid=ID122&to=61412345678&type=2-way

The following will send a 1-way Unicode SMS with the message text = “Hello World - 你好世界” and Sender ID = “Unicode”.

http://api.directsms.com.au/s3/http/send_message?connectionid=4c2c7d229b4d56bf249e9f6192cb7c11&message=00480065006C006C006F00200057006F0072006C00640020002D00204F60597D4E16754C&senderid=Unicode&to=61412345678&type=1-way&unicode=true

Sample Response

The following shows the server’s response to a successful message submission.

id: 5c2c7d229b4d56bf249e9f6192cb7c11

If you send long SMS messages (max_segments > 1), the server will respond with the ID of each SMS segment making up your long message on a new line. For example:

id: 5c2c7d229b4d56bf249e9f6192cb7c11
id: 439fce419a0f813fc8ea3f9abac3b71e
id: 211cf776a4eb9f0fc29229024ab1a5ff

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.

URL: http://api.directsms.com.au/s3/http/schedule_message

Parameter	Description	Required
connectionid	<p>The Connection ID returned by the call to the connect operation at the beginning of your session.</p> <p>If you cannot call the connect operation first, you can pass the authentication parameters in place of the connectionid.</p> <p>The authentication parameters are:</p> <p>username – The username used to log onto directSMS' customer portal (Mandatory)</p> <p>password – The password for the given user account (Mandatory)</p> <p>lic_key – Enterprise license key if one was issued to you (Optional)</p>	Y
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	Y

your original SMS

Valid values are “1-way” or “2-way” only

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	<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
timestamp	<p>The date and time to send this SMS, represented as the number of seconds after the Unix Epoch (January 1 1970 00:00:00 GMT).</p> <p>Please ensure you adjust this to your appropriate time zone and daylight savings setting.</p> <p>This approach ensures time zone independence</p>	Y

Y – Mandatory, N – Optional, C - Conditional

Success Output

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

id: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Error Output

If the parameters submitted are invalid for any reason, an error message will be returned instead.

err: Error message

Sample Request

The following will schedule a 1-way SMS with a Sender ID of “directSMS” for delivery at 12AM on January 1, 2012 (GMT).

http://api.directsms.com.au/s3/http/schedule_message?connectionid=4c2c7d229b4d56bf249e9f6192cb7c11&message=This+is+a+test&senderid=directSMS&to=61412345678&type=1-way×tamp=1325376000

The following will schedule a 2-way SMS with the Message Key of “ID112” for delivery at 12AM on January 1, 2012 (GMT).

http://api.directsms.com.au/s3/http/schedule_message?connectionid=4c2c7d229b4d56bf249e9f6192cb7c11&message=This+is+a+test&messageid=ID122&to=61412345678&type=2-way×tamp=1325376000

Please see <http://www.epochconverter.com/> for more information about using Unix Epoch timestamps, as well as an easy to use calculator when testing your application.

Sample Response

id: 5c2c7d229b4d56bf249e9f6192cb7c11

Cancelling 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.

Please note, only the user account used to create the original scheduled message is allowed to cancel it.

URL: http://api.directsms.com.au/s3/http/cancel_scheduled_message

Parameter	Description	Required
connectionid	<p>The Connection ID returned by the call to the connect operation at the beginning of your session.</p> <p>If you cannot call the connect operation first, you can pass the authentication parameters in place of the connectionid.</p> <p>The authentication parameters are:</p> <p>username – The username used to log onto directSMS' customer portal (Mandatory)</p> <p>password – The password for the given user account (Mandatory)</p> <p>lic_key – Enterprise license key if one was issued to you (Optional)</p>	Y
id	<p>This is the 32 character identifier returned by the gateway in response to your original call to the "schedule message" operation</p>	Y

Y – Mandatory, N – Optional, C - Conditional

Success Output

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

This is the same identifier that you would have sent in your request to cancel the message.

```
id: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Error Output

If the parameters submitted are invalid for any reason, an error message will be returned instead.

```
err: Error message
```

Sample Request

The following will cancel a schedule 1-way SMS that was created earlier and has not yet been sent.

http://api.directsms.com.au/s3/http/cancel_scheduled_message?connectionid=4c2c7d229b4d56bf249e9f6192cb7c11&id=4c2c7d229b4d56bf249e9f6192cb7c32

Sample Response

```
id: 5c2c7d229b4d56bf249e9f6192cb7c11
```

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.

URL: http://api.directsms.com.au/s3/http/get_reply_messages

Parameter	Description	Required
connectionid	<p>The Connection ID returned by the call to the connect operation at the beginning of your session.</p> <p>If you cannot call the connect operation first, you can pass the authentication parameters in place of the connectionid.</p> <p>The authentication parameters are:</p>	Y

username – The username used to log onto directSMS’ customer portal (Mandatory)

password – The password for the given user account (Mandatory)

lic_key – Enterprise license key if one was issued to you (Optional)

messageid	This is the message identifier you passed when you sent the original 2-way SMS message. If this is set, only replies for the 2-way message(s) matching this identifier are retrieved. If this parameter is not set, all unread replies will be returned	N
------------------	---	---

mark_as_read	Mark the returned reply messages as read, to avoid returning them again in future. Please note; if you do not wish to mark the returned reply messages as read, do not include this parameter in your request. Any value set against this parameter will be treated as true	N
---------------------	---	---

Y – Mandatory, N – Optional, C - Conditional

Success Output

If there are no replies you will see the following

```
replies: 0
```

If replies are found for the parameters submitted, you will see the following

```
replies: xx
messageid: xxxx mobile: xxxx message: xxxxxxxxxxxxxxxx when: xxx
messageid: xxxx mobile: xxxx message: xxxxxxxxxxxxxxxx when: xxx
messageid: xxxx mobile: xxxx message: xxxxxxxxxxxxxxxx when: xxx
```

The “replies” field on the first line will tell you how many reply messages were found that matched your query. Each reply will occupy a new line in the output subsequent.

The output on each line will have the following fixed width fields:

Field	Description	Length
messageid	This is the message identifier you passed when you sent the original 2-way message Please note the messageid will be padded with spaces to achieve the length of 12 characters	12
mobile	The mobile number that sent this reply message in the format +61412345678 Please note the number will be padded with spaces to achieve the length of 20 characters	20
message	The message content Please note the message will be padded with spaces to achieve the length of 160 characters	160
when	This is the number of seconds since the reply was received	1+

Each field is separated from its header and the next field's header by 1 space as per the sample output above.

Error Output

If the parameters submitted are invalid for any reason, an error message will be returned instead.

```
err: Error message
```

Sample Request

The following will fetch any replies to 2-way messages submitted with the message identifier "ID122".

http://api.directsms.com.au/s3/http/get_reply_messages?connectionid=4c2c7d229b4d56bf249e9f6192cb7c11&messageid=ID122&mark_as_read=true

Sample Response

```
replies: 0
```

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

4. 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.

URL: http://api.directsms.com.au/s3/http/get_inbound_messages

Parameter	Description	Required
connectionid	<p>The Connection ID returned by the call to the connect operation at the beginning of your session.</p> <p>If you cannot call the connect operation first, you can pass the authentication parameters in place of the connectionid.</p> <p>The authentication parameters are:</p> <p>username – The username used to log onto directSMS' customer portal (Mandatory)</p> <p>password – The password for the given user</p>	Y

	account (Mandatory)	
	lic_key – Enterprise license key if one was issued to you (Optional)	
inbound_number	You can restrict the search to get only the messages received on a given inbound number if you have multiple numbers. If nothing is specified, all unread messages received will be retrieved	N
mark_as_read	Mark the returned inbound messages as read, to avoid fetching them again in future Please note; if you do not wish to mark the returned messages as read, do not include this parameter in your request. Any value set against this parameter will be treated as true	N

Y – Mandatory, N – Optional, C - Conditional

Success Output

If there are no inbound messages you will see the following

```
messages: 0
```

If inbound messages are found for the search parameters submitted, you will see the following

```
messages: xx
inbound: xxxx mobile: xxxx message: xxxxxxxxxxxxxxxx when: xxx
inbound: xxxx mobile: xxxx message: xxxxxxxxxxxxxxxx when: xxx
inbound: xxxx mobile: xxxx message: xxxxxxxxxxxxxxxx when: xxx
```

The “messages” field on the first line will tell you how many inbound messages were found that matched your query. Each message will occupy a new line in the output subsequent.

The output on each line will have the following fixed width fields:

Field	Description	Length
inbound	This is the inbound number the message was received on in the format +61412345678.	20

	Please note the number will be padded with spaces to achieve the length of 20 characters	
mobile	The mobile number sending this reply message in the format +61412345678.	20
	Please note the number will be padded with spaces to achieve the length of 20 characters	
message	The message content	160
	Please note the message will be padded with spaces to achieve a length of 160 characters	
when	This is the number of seconds since the reply was received	1+

Each field is separated from its header and the next field's header by 1 space.

Error Output

If the parameters submitted are invalid for any reason, an error message will be returned instead.

```
err: Error message
```

Sample Request

The following will fetch any inbound message received on inbound number "0429 007 007".

http://api.directsms.com.au/s3/http/get_inbound_messages?connectionid=4c2c7d229b4d56bf249e9f6192cb7c11&inbound_number=61429007007&mark_as_read=true

Sample Response

```
messages: 0
```

Getting your balance

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

URL: http://api.directsms.com.au/s3/http/get_balance

Parameter	Description	Required
connectionid	<p>The Connection ID returned by the call to the connect operation at the beginning of your session.</p> <p>If you cannot call the connect operation first, you can pass the authentication parameters in place of the connectionid.</p> <p>The authentication parameters are:</p> <ul style="list-style-type: none">username – The username used to log onto directSMS' customer portal (Mandatory)password – The password for the given user account (Mandatory)lic_key – Enterprise license key if one was issued to you (Optional)	Y

Y – Mandatory, N – Optional, C - Conditional

Success Output

If you have pre-paid account, the number of pre-paid SMS credits left on your account is returned. If you have a monthly invoiced account, this operation will return -1

```
credits: xxxx
```

Error Output

If the parameters submitted are invalid for any reason, an error message will be returned instead.

```
err: Error message
```

Sample Request

The following will retrieve the pre-paid credit balance using the Connection ID returned by the server after the call to the connect operation at the beginning of the user's session.

http://api.directsms.com.au/s3/http/get_balance?connectionid=4c2c7d229b4d56bf249e9f6192cb7c11

The following will authenticate and retrieve the pre-paid credit balance in one call.

http://api.directsms.com.au/s3/http/get_balance?username=my_user&password=secret

Sample Response

credits: 3331.5

Disconnecting

To end your session and terminate your connection to the API server, you have to call the disconnect operation. This will signal to the server to clean up any resources allocated to your connection during your session.

URL: <http://api.directsms.com.au/s3/http/disconnect>

Parameter	Description	Required
Connectionid	Connection ID that was returned when you called the connect operation at the beginning of your session	Y

Y – Mandatory, N – Optional, C - Conditional

Success Output

If the call succeeds the connection identifier for the closed connection is echoed back.

id: XX

Error Output

If the parameters submitted are invalid in any way, an error message will be returned.

err: Error message

Sample Request

<http://api.directsms.com.au/s3/http/disconnect?connectionid=4c2c7d229b4d56bf249e9f6192cb7c11>

Sample Response

id: 4c2c7d229b4d56bf249e9f6192cb7c11

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.