**Colt SMS REST APIs Guide**

|  |
| --- |
| Copyright © |
| No part of this document may be reproduced or adapted in any form, including photocopying or storing it by electronic means, except as is necessary for the purpose of the recipient's dealings with Colt, without the written permission of Colt. Any copies made of any part of this document shall include the notice:  © 2023 Colt Technology Services. The Colt name and logos are trademarks. All rights reserved. |
| No information contained in this document shall be disclosed to any third party without the written permission of Colt. |

SMS REST APIs Implementation Guide

**Document Version** 1.0

**Release Date** 25/11/2022

Contents

[**1.** **Document History** 3](#_Toc126160991)

[**2.** **Introduction** 4](#_Toc126160992)

[2.1. REST API releases 4](#_Toc126160993)

[2.2. Access & End-Point URLs 4](#_Toc126160994)

[2.2.1. Error handling 4](#_Toc126160995)

[2.3. How to get access to new REST APIs, Colt API portal & support 5](#_Toc126160996)

[**3.** **Authorization** 7](#_Toc126160997)

[3.1. Oauth 2.0 Authentication 7](#_Toc126160998)

[3.1.1. API Details 7](#_Toc126160999)

[3.1.2. Input parameter(s) 7](#_Toc126161000)

[3.1.3. Error handling 7](#_Toc126161001)

[3.1.4. Response samples 7](#_Toc126161002)

[**4.** **Number Hosting** 9](#_Toc126161003)

[4.1. Guide Objective & Structure 9](#_Toc126161004)

[4.2. Product Documentation 9](#_Toc126161005)

[4.2.1. Service Matrix 9](#_Toc126161006)

[4.2.2. Fair use policy & Security 9](#_Toc126161007)

[4.2.3. Resource names for “SMS API” operations 9](#_Toc126161008)

[4.3. List of APIs for Colt SMS 9](#_Toc126161009)

[4.4. Message ID 10](#_Toc126161010)

[4.5. Status 10](#_Toc126161011)

[4.5.1. Current list of Colt SMS Countries 10](#_Toc126161012)

[4.5.2. Number Format 11](#_Toc126161013)

[4.6. Schemas 11](#_Toc126161014)

[**APPENDIX A: API Specifications** 16](#_Toc126161015)

[OAS Specification for SMS API 16](#_Toc126161016)

# **Document History**

Soft copies of this document are maintained under change control, but hard copies are not. This may not be a printout of the latest version.

**Revision history**

|  |  |  |
| --- | --- | --- |
| Revision | Date | Description |
| **1.0** | November 25 2022 | Document Launch |

# **Introduction**

We support Next Gen REST APIs for our SMS solution.

Our Next Gen APIs:

* Conform to the Industry standard REST-JSON APIs
* Aligns with TMF conventions
* Support OAuth2 and JWT based authentication and authorization
* Offers a standard format for numbers, date-time, etc.
* Optimizes fields
* No change with respect to synchronous/asynchronous API behavior

## REST API releases

Colt will update you with plans for future REST API release, and you can find the release notes for this at colt.net/cocom, alongside the plans for our other API releases per Colt Number Hosting.

## Access & End-Point URLs

Colt SMS REST APIs can be enabled in these environments below:

* Production: live environment.
* Sandbox: non-production environment for testing the production version of the APIs.

|  |  |
| --- | --- |
| Interface | Production |
| **REST** | **SMS API**  end point: <https://apimgd01.colt.net/message/v1>  Resource: resource name |

### Error handling

#### Error Schema

|  |  |
| --- | --- |
| description: | Used when an API throws an Error, typically with a HTTP error response-code (3xx, 4xx, 5xx) |
| code\* | string *example: ERR01*  Application relevant detail, defined in the API or a common list. |
| reason\* | string  Explanation of the reason for the error which can be shown to a client user. |
| message | string  More details and corrective actions related to the error which can be shown to a client user. |
| status | string *example: 401-02*  HTTP Error code extension |

**Header:**

|  |  |  |
| --- | --- | --- |
| Name | Description | Type |
| Content-Type |  | string *Example: multipart/form-data; boundary=645501572420247968528808* |
| APIGWExceptionSystem |  | string |
| APIGWExceptionCode |  | string |
| APIGWTrackingId | Id to track the request end to end | string *Example: abcd456e-d45645-dfaafda-1232345667dd pattern: ^[a-zA-Z0-9\_.~:@-]{1,255}$* |

#### Example

Below is the TMF compliant error payload:

{

"code": "ERR01",

"reason": "string",

"message": "string",

"status": "401-02"

}

## How to get access to new REST APIs, Colt API portal & support

Please inform your account manager to request access to the sandbox and/or production environments for the REST APIs and access to Colt API portal.

**Sandbox support**

Support for sandbox testing will be provided by the Voice API Support team and is a chargeable service. Please contact your account manager for further information.

**Beta support**

Support in the beta environment will be provided by our testing team who are directly involved and responsible for the upcoming release. Contact details are published in the Number Hosting release note.

**Production support**

Please inform your account manager to request access to the sandbox and/or production environments for the REST APIs. You can raise any questions or issues to them directly and they will work to resolve these with our back office IT support.

* ***Oauth 2.0 authentication:***

Your Client ID and Client Secret will be shared with you by Colt’s Reseller Support Voice team

* ***JWT Authentication:***

If you wish to use JWT authentication, then you will need to share the JWT public certificate (zipped) and name of the token issuer. Your Application ID (s) to add into the JWT key as the ‘Audience’ value will be shared with you by Colt’s Reseller Support Voice team.

# **Authorization**

Industry standard authentication and authorization mechanism is now enabled

* **OAuth 2.0** access token (applicable for 30 mins only) with client\_credential grant type used to authenticate clients You can request up to 5 CLIENT\_ID + CLIENT\_Secret per company account.
* **JWT authentication type** is also supported for these APIs now.
* This will giveoptions to users to opt for any one of the authentication types

## Oauth 2.0 Authentication

### API Details

**Resource name for Authorisation API**

/token

### Input parameter(s)

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| **Header** | | |
| APIGWTrackingId | Mandatory | description: Unique identifier for the order  *example: abcd456e-d45645-dfaafda-1232345667dd*  pattern: ^[\w.~:@-]{1,255}$ |
| **Request Body** | | |
| grant\_type | Mandatory | Value= client\_credentials |

### Error handling

Below would be TMF complaint error payload:

{

"code": "ERR01",

"reason": "string",

"message": "string",

"status": "401-02"

}

### Response samples

|  |
| --- |
| Success response |
| {      "access\_token": "d3a1a83223e84bbe9d90173d784aaddf",      "token\_type": "Bearer",      "expires\_in": 1800,      "scope": "OauthScope NhScope"  } |

|  |
| --- |
| Expiry response |
| {      "code": "APIGW-400-00",      "reason": "Unauthorized",      "message": "Token specified is invalid or has expired..",      "status": "400-00"  } |

# **Number Hosting**

## Guide Objective & Structure

This guide aims to support you to set up and use SMS enabled mobile numbers via Colt Voice APIs, as well providing a short introduction and an explanation of the general principles.

## Product Documentation

This guide should be read together with the following documentation:

|  |  |
| --- | --- |
| Document | Description |
| Service Matrix | Business rules & features availability  error codes (system, porting, address validation, …)  Fields validation with applicable API & countries |

### Service Matrix

Please check Service Matrix for details of the number types offered by Colt and the business rules.

### Fair use policy & Security

Any misuse of number, whatever nature, might lead to service suspension or number claimed back.

### Resource names for “SMS API” operations

|  |  |
| --- | --- |
| Operation | Resource Name |
| Outbound SMS | /outbound-sms |

Please refer to *OAS Specification for functional Number Hosting APIs* for the OAS Specification for above API.

#### Error handling

Below would be TMF complaint error payload:

{

"code": "ERR01",

"reason": "string",

"message": "string",

"status": "401-02"

}

## List of APIs for Colt SMS

Colt APIs are listed and described in the below table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| API | API Type: SOAP or REST | Order Type | Description | MessageID created  (Yes/No) |
| SMS API | REST | N/A | Enables you to send SMS from your account and receive messages to your number | Yes |

## Message ID

After an SMS operation, a unique ID [**messageId**] will be returned to you.

messageId follows the below pattern and format:

|  |  |
| --- | --- |
|  | Description |
| Pattern | 32 character long (128 bit) |
| Format | “[A-Za-z0-9\-]{1,36}“ |
| Maximum length | 36, including ‘-’ characters |
| Example | 8700206f-c3b3-4c10-8cc2-2490f41eedc0 |

## Status

Your message will have a status [**status**], that will vary over the time and the scenarios.

Please find below *Status* and description:

|  |  |  |
| --- | --- | --- |
| orderStatus | Description | Next action |
| Success | The message was successfully accepted for delivery. | No action required. |
| Throttled | You are sending SMS faster than the account limit | Contact support to check your throughput limit and can request to increase this |
| Missing Parameters | Your request is missing one of the required parameters | Review the parameters to ensure all fields are entered |
| Invalid Parameters | The value of one or more parameters is invalid. | Check your entered parameters to ensure the correct values are input |

### Current list of Colt SMS Countries

Please find below country codes supported in Colt APIs:

|  |  |
| --- | --- |
| Country | Country Code |
| Netherlands | NL |
| Sweden | SE |
| United Kingdom | GB |

### Number Format

Colt REST APIs support these number formats:

* **E164 international format**

|  |  |
| --- | --- |
| Résultat de recherche d'images pour "important message icon" | * Only e164 format is supported in REST APIs request/response. * Order Detail response will have the same format as the request. |

## Schemas

|  |  |
| --- | --- |
| APIGWTrackingIdType | APIGWTrackingIdTypestring *example: abcd456e-d45645-dfaafda-1232345667dd pattern: ^[a-zA-Z0-9\_.~:@-]{1,255}$*  Unique identifier for the transaction |
| deliveryReceipt | {   |  |  | | --- | --- | | msisdn | string *example: +447700900000*  The number the message was sent to. Numbers are specified in  E.164 format. | | to | string *example: AcmeInc*  The SenderID you set in from in your request. | | network-code | string *example: 12345*  The Mobile Country Code Mobile Network Code (MCCMNC) of the  carrier this phone number is registered with. | | messageId | string *example: 0A0000001234567B*  The ID for this message. | | status | string *example: delivered x-possible-values: List [ "delivered", "expired", "failed", "rejected",*  *"accepted", "buffered", "unknown" ]*  A code that explains where the message is in the delivery process. | | scts | string *example: 2001011400*  When the DLR was received from the carrier in the following format  YYMMDDHHMM. For example, 2001011400 is at 2020-01-01 14:00 | | err-code | string *example: 0*  The status of the request. Will be a non 0 value if there has been  an error, or if the status is unknown. | | serviceProfile | serviceProfileTypestring  service profile associated with customer request | | client-ref | string *example: my-personal-reference*  If the client-ref is set when the SMS is sent, it will be  included in the delivery receipt | | message-timestamp | string *example: 2020-01-01 12:00:00*  The time when when this delivery receipt is pushed to your  webhook endpoint. | |  |  |   } |
| outboundMsgRespType | {   |  |  | | --- | --- | | message-count | string *example: 1*  The amount of messages in the request | | messages | [messageResponseType{   |  |  | | --- | --- | | to | string *example: +447700900000*  The number the message was sent to.  Numbers are specified in E.164 format. | | message-id | string *example: 0A0000000123ABCD1 xml: OrderedMap { "name": "messageId" }*  The ID of the message  *xml:    name: messageId* | | status | string *example: 0*  The status of the message. . | | network | string *example: 12345*  The estimated ID of the network of the  recipient | | client-ref | string *example: my-personal-reference*  If a client-ref was included when sending  the SMS, this field will be included and hold  the value that was sent. | |  |  |   }] | |  |  |   } |
| messageResponseType | {   |  |  | | --- | --- | | to | string *example: +447700900000*  The number the message was sent to. Numbers are  specified in E.164 format. | | message-id | string *example: 0A0000000123ABCD1 xml: OrderedMap { "name": "messageId" }*  The ID of the message  *xml:    name: messageId* | | status | string *example: 0*  The status of the message. . | | network | string *example: 12345*  The estimated ID of the network of the recipient | | client-ref | string *example: my-personal-reference*  If a client-ref was included when sending the SMS, this field will  be included and hold the value that was sent. | |  |  |   } |
| messagingError | {   |  |  | | --- | --- | | description: | Error | | message-count | string *example: 1*  The amount of messages in the request | | messages | [messagingErrorDetail{   |  |  | | --- | --- | | status | string *example: 2*  The error status of the message | | error-text | string *example: Missing to param*  The description of the error | |  |  |   }] | |  |  |   } |
| messagingErrorDetail | {   |  |  | | --- | --- | | status | string *example: 2*  The error status of the message | | error-text | string *example: Missing to param*  The description of the error | |  |  |   } |
| serviceProfileType | string  service profile associated with customer request |
| inboundMessageType | {   |  |  | | --- | --- | | serviceProfile\* | serviceProfileTypestring  service profile associated with customer request | | msisdn\* | string *example: +447700900001*  The phone number that this inbound message was sent from.  Numbers are specified in E.164 format. | | to\* | string *example: +447700900000*  The phone number the message was sent to. This is your  virtual number. Numbers are specified in E.164 format. | | messageId\* | string *example: 0A0000000123ABCD1*  The ID of the message | | text\* | string *example: Hello world*  The message body for this inbound message. | | type\* | string *example: text*  Possible values are:   * text - standard text. * unicode - URLencoded unicode . This is valid for   standard GSM, Arabic, Chinese, double-encoded  characters and so on.   * binary - a binary message. | | keyword\* | string *example: HELLO*  The first word in the message body. Converted to upper case. | | message-timestamp\* | string *example: 2020-01-01 12:00:00*  The time when the message is pushed to customer's end point | | timestamp | string *example: 1578787200*  A unix timestamp representation of message-timestamp. | | concat | string *example: true*  True - if this is a concatenated message. This field does  not exist if it is a single message | | concat-ref | string *example: 1*  The transaction reference. All parts of this message share  this value. | | concat-total | string *example: 3*  The number of parts in this concatenated message. | | concat-part | string *example: 2*  The number of this part in the message. Counting starts at 1. | | data | string($binary)  The content of this message, if type is binary. | | udh | string  The hex encoded User Data Header, if type is binary | |  |  |   } |
| outboundMsgType | {   |  |  | | --- | --- | | serviceProfile\* | serviceProfileTypestring  service profile associated with customer request | | from\* | string  The name or number the message should be sent from.  Alphanumeric senderID's are not supported in all countries,  please contact support for more details. If alphanumeric,  spaces will be ignored. Numbers are specified in E.164 format. | | to\* | string *minLength: 7 maxLength: 16 pattern: \d{7,16} example: +447700900000*  The number that the message should be sent to.  Numbers are specified in E.164 format. | | text\* | string *example: Hello World!*  The body of the message being sent. If your message contains characters that can be encoded according to the GSM Standard and Extended tables then you can set the type to text. If your message contains characters outside this range, then you will need to set the type to unicode. | | ttl | {   |  |  | | --- | --- | | description: | Advanced: The duration in milliseconds  the delivery of an SMS will be attempted.  By default Colt attempts delivery for 72  hours, however the maximum effective  value depends on the operator and is  typically 24 - 48 hours. We recommend  this value should be kept at its default or  at least 30 minutes. | |  |  |   } | | status-report-req | boolean *example: false default: true*  Advanced: Boolean indicating if you like to receive  a [Delivery Receipt](https://editor.swagger.io/messaging/sms/building-blocks/receive-a-delivery-receipt). | | type | string *example: text default: text*  Advanced: The format of the message body  Enum: [ text, binary, wappush, unicode, vcal, vcard ] | | vcard | string($vcard)  Advanced: A business card in [vCard format](https://en.wikipedia.org/wiki/VCard). Depends on  type parameter having the value vcard. | | vcal | string($vcal)  Advanced: A calendar event in [vCal format](https://en.wikipedia.org/wiki/VCal" \t "_blank). Depends on type  parameter having the value vcal. | | body | string *example: 0011223344556677*  Advanced: Hex encoded binary data. Depends on type  parameter having the value binary. | | udh | string *example: 06050415811581*  Advanced: Your custom Hex encoded [User Data Header](https://en.wikipedia.org/wiki/User_Data_Header).  Depends on type parameter having the value binary. | | protocol-id | integer *example: 127*  Advanced: The value of the [protocol identifier](https://en.wikipedia.org/wiki/GSM_03.40#Protocol_Identifier) to use.  Ensure that the value is aligned with udh. | | title | string *example: Welcome*  Advanced: The title for a wappush SMS. Depends on  type parameter having the value wappush. | | url | string *example: https://example.com*  Advanced: The URL of your website. Depends on type  parameter having the value wappush. | | validity | string *example: 300000*  Advanced: The availability for an SMS in milliseconds.  Depends on type parameter having the value wappush. | | client-ref | string *example: my-personal-reference*  Advanced: You can optionally include your own reference  of up to 100 characters. | | entity-id | string *example: 1101456324675322134*  Advanced: A string parameter that satisfies regulatory  requirements when sending an SMS to specific countries.  For more information please contact support | | content-id | string *example: 1107457532145798767*  Advanced: A string parameter that satisfies regulatory  requirements when sending an SMS to specific countries.  For more information please contact support | |  |  |   } |
| Error | {   |  |  | | --- | --- | | description: | Used when an API throws an Error, typically with  a HTTP error response-code (3xx, 4xx, 5xx) | | code\* | string *example: ERR01*  Application relevant detail, defined in the API or a  common list. | | reason\* | string  Explanation of the reason for the error which can be  shown to a client user. | | message | string  More details and corrective actions related to the error  which can be shown to a client user. | | status | string *example: 401-02*  HTTP Error code extension | |  |  |   } |

## Auto-redact

Auto-redact is a service that automatically redacts Personal Identifiable Information (PII) from our systems without any actions from your side. You can define whether you want redaction to be done immediately (once the message or call has been processed) or with a delay of several days. Supported values are 15, 30, 60, and 90 days.

PII includes the message content and the receiver phone number for outbound messages and the sender phone number for inbound messages. The SMS API uses a data pipeline software to transport CDRs to various databases. The data pipeline keeps CDRs along with the receiver/sender phone number for 7 days. Thus, besides server logs and the long-term storage of CDRs, PII is also stored in the data pipeline logs.

Auto-redact for SMS redacts server logs, CDRs, and the data pipeline logs. The scope of auto-redaction is configurable and can include the following options:

1. Message content redaction only.  
2. Phone number redaction only.  
3. Phone number encryption only  
4. Message content redaction together with redaction or encryption of the phone number.

To enable this on your account, please contact the Reseller Support Voice team.

# **APPENDIX A: API Specifications**

## OAS Specification for SMS API

|  |  |
| --- | --- |
| Type | Attachment |
| YAML |  |