

---

# Mobile Messaging Exchange API

---

## API Design Brief

Bhakti Mehta, Sr. Product Management Specialist

June 2020 | *Version 2.0*

## Contents

1. Introduction .....	3
1.1 Program Summary .....	3
1.2 Getting Started .....	3
1.1 Sandbox Implementation .....	4
1.2 Production Implementation.....	4
1.3 Swagger Documentation.....	4
1.4 Accessing APIs on the Developer Portal .....	4
1.5 Base URL .....	5
2. Specifications .....	5
2.1 RESTful .....	5
2.2 Authentication .....	6
2.1.1 Basic Authentication Key - Sandbox and Production.....	6
3. Mobile Messaging API Features .....	7
3.1 Messaging API .....	7
3.2 Account Administration .....	9
3.3 CDR Report .....	10
3.4 Usage Report .....	10
4. API Route Index .....	11
5. Other APIs.....	11

# 1. Introduction

This document is a high-level design brief for the Tata Communications Mobile Messaging Exchange API Program. The intent of this document is to provide explanatory information along with technical API specification files; and in no way replaces any publicly displayed information listed on <https://developer.tatacommunications.com> or other official TCL documentation.

## 1.1 Program Summary

The Mobile Messaging API Program provides an easy-to-integrate & scalable interface with REST for A2P SMS Messaging and it shortens the time required for ordering and reporting of Tata Communications' messaging services by enabling self-management capabilities for those services.

This initial release provides REST/HTTP API for mobile terminated messaging and administrative features such as sender whitelisting, real-time reporting & pricing.

Release	Features	Description
1.0	Messaging	Mobile Terminated SMS
2.0	Account Administration	Destination Pricing, SenderID Whitelisting, Usage Reporting
	CDR Report	Reporting & Analytics APIs for tracking message delivery records
	Usage Report	Reporting & Analytics APIs for tracking customer account usage records

## 1.2 Getting Started

Each Tata Communications API include a Sandbox and Live implementations. The Sandbox is a trial instance of our live backend with limited credit available for Messaging or provides a virtual backend that imitates the Live API features. It allows developers to learn the API calls structure and responses. It does not use nor return actual user data nor live network status. Both implementations use HTTPS connections and requires an authorization key that may be generated on the Developer Portal (see instructions below) or provided by your account representative.

Developers can start in the sandbox implementation before accessing Live API. The account representative will provide a set of POSTMAN collections and associated credentials so that you can start using the APIs immediately. In case you would like to access or change your existing secret key, you can visit our developer portal to view or change your secret key for added security.

## 1.1 Sandbox Implementation

The Sandbox Implementation connects with Tata Communications' trial account which provides a limited credit of 650 SMSes to the customers to try the features and get familiarized with the Messaging API. The Account Administration APIs have a virtual backend connected, that provides a Live-like experience for the APIs and exposes the key functions of the APIs for testing before integration with Live APIs.

## 1.2 Production Implementation

The production implementation connects with Tata Communications' Live network and operational systems. A user or developer must be a *valid & verified* TCL customer qualify for using the Production APIs. This validation is performed through your account representative and API product management before access is granted to the production API implementation.

## 1.3 Swagger Documentation

Swagger documentation for Mobile Messaging is available at <https://docs.tclaws.co/sb/sms/v1>. Developers can refer to the webpage to get information on the model of request and response parameters. The webpage is an interactive swagger UI for developers to quickly navigate between respective APIs.

## 1.4 Accessing APIs on the Developer Portal

Developers can register with the Developer Portal in cases where the Sandbox or Production access is not arranged through the account representative or API product management. In such cases, perform the following steps to gain access to Mobile Messaging APIs.

1. Go to our developer portal (<https://developer.tatacommunications.com/>) and register for an account
2. Share the email used to register with your account representative or API Product Manager.
3. Use the portal functions to create an "APP". An APP is an identifier assigned to your account that will be used for secure access to the APIs.
4. Using the search function, search for "Mobile Messaging" to find the APIs by name.
5. On the left-hand menu, click "Access" to request access to the sandbox implementation.
6. A notification will be sent to the API Product Manager and/or account representative for approval. You will be notified via email upon approval.
7. Access to the Live implementation follows the same steps using the same APP or a different APP if desired.
8. Tata Communications requires all developers to start in the sandbox, test and learn the API formatting and responses before moving to Production/Live.

## 1.5 Base URL

Environment	URL
Account Administration API	
Production	<a href="https://api.tatacommunications.com/sms/v1">https://api.tatacommunications.com/sms/v1</a>
Sandbox	<a href="https://api.tatacommunications.com/sb/sms/v1">https://api.tatacommunications.com/sb/sms/v1</a>
Messaging API	
Production	<a href="https://api.tatacommunications.com/mmx/v1/messaging">https://api.tatacommunications.com/mmx/v1/messaging</a>
Sandbox	<a href="https://api.tatacommunications.com/sb/mmx/v1/messaging">https://api.tatacommunications.com/sb/mmx/v1/messaging</a>
CDR Report API	
Production	<a href="https://api.tatacommunications.com/mmx/v1/messaging">https://api.tatacommunications.com/mmx/v1/messaging</a>
Sandbox	<a href="https://api.tatacommunications.com/sb/mmx/v1/messaging">https://api.tatacommunications.com/sb/mmx/v1/messaging</a>
Usage Report API	
Production	<a href="https://api.tatacommunications.com/mmx/v1/messaging">https://api.tatacommunications.com/mmx/v1/messaging</a>
Sandbox	<a href="https://api.tatacommunications.com/sb/mmx/v1/messaging">https://api.tatacommunications.com/sb/mmx/v1/messaging</a>

# 2. Specifications

## 2.1 RESTful

Tata Communications' APIs are RESTful. REST API principles use distinct URLs for a resource (e.g. a contact is /contacts, an order is /orders, sms is /sms) and employs HTTP methods such as GET, POST, PUT and DELETE to

query and manage resource data. The Mobile Messaging APIs only accept application/json as input payloads and always return data in an application/json format.

## 2.2 Authentication

Production and Sandbox implementations use the Basic HTTP authentication scheme in which the user (e.g. your application) transmits credentials in the header field of each HTTP request. The form is Authorization: Basic <credentials>, where credentials is the base64 encoding of an id and password joined by a colon. This mechanism is specified in [RFC 7617](#).

### 2.1.1 Basic Authentication Key - Sandbox and Production

Mobile Messaging APIs operate only through the Tata Communications API gateway. The gateway checks each incoming HTTP request for a recognized Authentication key.

Most sandbox users will be provided pre-registered Authentication Key for immediate access into the Sandbox. Additional keys can be provided by Tata upon request or generated by the user itself.

In order to create more keys for accessing the APIs the user can register with the API gateway to create an ‘app’ on the platform. The application is given an ID (APPID) and secret. Developers can register at <https://developer.tatacommunications.com>.

## 3. Mobile Messaging API Features

### 3.1 Messaging API

The Mobile Messaging APIs enable end-customers and partners to be able to send Mobile Terminated SMS & on providing administrative functionalities to manage their accounts. A customer can send SMS from Application to Person use cases & use the API for self-service functions & real-time reporting functions to manage their accounts.

#### 3.1.1 Mobile Terminated SMS

This is a REST based API for Application to Person SMS use cases. The API enables the end user applications to connect to Mobile Messaging Exchange on REST/HTTP interface for building SMS communication channels over a global network for trusted routes and terminations.

The following key parameters and resources are used in the Messaging APIs.

Parameter	Description
from ( <i>string</i> )	Sender Id
to ( <i>string</i> )	Mobile Number/MSISDN for termination
msg ( <i>string</i> )	Content of the message
pid ( <i>integer, optional</i> ):	Set the PID (Protocol ID) value. For example, a SIM Toolkit message would use something like pid=127, coding=2, alt-dcs=1, mclass=3
dcs ( <i>Inline Model 1, optional</i> )	
mclass ( <i>integer, optional</i> ):	Sets the Message Class in DCS Field. Accepts value between 0 and 3. if we use 0=sends the message directly to display, 1=sends to mobile, 2=sends to SIM, 3=sends to SIM Toolkit = ['0', '1', '2', '3'],
mwi ( <i>integer, optional</i> ):	Set Message Waiting Indicator bits in the DCS field. If given, the message will be encoded as Message Waiting Indicator. if we use 0,1,2,3 , its for activating the voice, fax, e-mail and other indicator. Or 4,5,6,7 for deactivating respectively = ['0', '1', '2', '3', '4', '5', '6', '7']
altdcs ( <i>integer, optional</i> ):	If not specified then, by default=0 else 1=FX. = ['0', '1']
coding ( <i>integer, optional</i> ):	Sets the coding scheme bits in DCS field. Accepts 0,1,2. 0=7-bit, 1=8-bit 2=UCS2, If unset, defaults to 7-bit.

	= ['0', '1', '2']
dlr (Inline Model 2, optional)	
url (string, optional):	url callback to which the DLR will be delivered
mask (integer, optional):	0="No dlr is requested, same as no dlr block or no url specified" 1="Success and Failures", 2="Failures only". The default applies only when the url is specified = ['0', '1', '2']
segmentCntAck (Inline Model 3, optional):	Request message segment information in response to the API by selecting the option
responseAckType (integer, optional)	['1', '2', '3'] responseAckType = 3 : <b>length, bytes, segment count</b> will be sent to customer along with message id responseAckType = 2 : <b>bytes, segment count</b> will be sent to customer along with message id responseAckType = 1 : <b>segment count</b> will be sent to customer along with message id
responseByDcsCoding (boolean, optional)	[true,false] responseByDcsCoding = true: Segment count should be sent as per data coding standard (DCS) responseByDcsCoding = false: Segment count should be sent as per internal data coding logic.
enhldlr (Inline Model 4, optional)	Enhanced Delivery report
initialMCCMNC (integer, optional)	= ['0', '1'], Original MCCMNC of the "to" Number's Operator .
finalMCCMNC (integer, optional)	= ['0', '1'], Final MCCMNC of the "to" Number's Operator (If the number is ported, this will be different than the "initialMCCMNC". Else it will be same as "initialMCCMNC").
portability (integer, optional)	= ['0', '1'], To get the portability status.
priceCurrency (integer, optional)	= ['0', '1'],



	To get the price and currency for the sent message.
dlrText ( <i>integer</i> , optional)	= ['0', '1'] Not applicable currently. This field is reserved for future use.
udh ( <i>string</i> , optional)	User Data header for the sent message

## 3.2 Account Administration

Account administration APIs enable self-service & real-time reporting capabilities for MMX customers. Following are the important features released as part for the first release.

### 3.2.1 Destination Pricing

This API provides an Offer Price Card for a given customer account, based on the destination mobile country code and/or mobile network code.

### 3.2.2 Sender ID Whitelisting

This API is used to raise request for creating a new Sender ID to be whitelisted with MNOs, for a given pre-registered customer account.

### 3.2.3 Month-to-date usage lookup

This API provides the current month's usage summary for a given customer account ID in terms of total volume utilized till date and corresponding total amount in terms of the money value

Some of the important parameters used in this API are as follows,

Parameter	Description
mcc ( <i>string</i> , <i>required</i> ):	Mobile Country code from customer Price Card
mnc ( <i>string</i> , <i>optional</i> ):	Mobile Network code from customer price card
accountId ( <i>string</i> , <i>required</i> ):	Mobile messaging exchange Customer account ID
SenderId ( <i>string</i> , <i>required</i> ):	SenderId required to be whitelisted on registered account

### 3.3 CDR Report

CDR Report API provides SMS delivery statuses, errors, delivery time and destination records for tracking, analytics & troubleshooting purposes. These records are available after 72 hours of the actual message delivery. All the CDR records for a given customer account ID can be fetched for a specific time period using the CDR/account/{accountId} resource.

A specific Message CDR record can be fetched for a given message Unique ID obtained at the time of message delivery, using the CDR/message/{messageId}

Some of the important parameters in this API are described below:

Parameter	Description
messageId ( <i>string, required</i> ):	Message unique identifier received in response when sms is submitted
accountId ( <i>string, required</i> ):	Unique Customer Account ID provided at onboarding
fromDate ( <i>string, required</i> ):	This is the from date in format of dd-MM-yyyy HH:mm:ss (Ex. 03-03-2020 07:09:00) which customer account usage is available for usage report.
toDate ( <i>string, required</i> ):	This is the to date in format of dd-MM-yyyy HH:mm:ss (Ex. 03-03-2020 07:09:00) which customer account usage is available for usage report.
pageNo ( <i>double, optional</i> ):	Paginate the records fetched and navigate using the pageNo (default is 1)
pageSize ( <i>double, optional</i> ):	Page size specifying the number of records required in a single page (default is 25)

### 3.4 Usage Report

Usage Report API provides destination wise price, delivery timestamp, aggregate volumes in the given time range & the total usage for all or specific customer account IDs. This API provides near immediate & flexible access to obtain & track your message transaction information

Some of the important parameters in this API are described below:

Parameter	Description
service ( <i>string, required</i> ):	This is the service type used [MT, MO, MLNS]
customerId ( <i>string, required</i> ):	This is the unique customer ID provided at the time of onboarding
accountId ( <i>string, optional</i> ):	Mobile messaging exchange Customer account ID
mcc ( <i>string, required</i> ):	Mobile Country code from customer Price Card
mnc ( <i>string, optional</i> ):	Mobile Network code from customer price card

fromDate ( <i>string, required</i> ):	This is the from date in format of dd-mm-yyyy (Ex. 01-01-2020) for which customer account usage is available for usage report.
toDate ( <i>string, required</i> ):	This is the from date in format of dd-mm-yyyy (Ex. 01-01-2020) which customer account usage is available for usage report.

## 4. API Route Index

Method	Route
Messaging	
POST	/sms
CDR Report	
GET	/CDR/message/{messageId}
GET	/CDR/account/{accountId}
Usage Report	
GET	/report/usage
Account Administration	
GET	/destination
GET	/report
POST	/senderId

## 5. Other APIs

Horizontal APIs are those that are not specific to any one Tata Communications product or service. Examples include Standardized Address API (performs physical address validation) and Trouble Ticket API (allows user to raise tickets and incident management).

Additional product-specific APIs are available for Tata Communications' Voice, Networking and Cloud portfolio. Please refer to the document API Catalog Brief for a summary of horizontal and product-specific APIs and API programs.