iTextos Reference Document Manthan API

Table of Contents

Overview

This guide provides specifications of the QueryString based API connectivity provided by iTextos to its customers to subscribe to iTextos SMS Service. This guide is intended for developers planning on integrating their systems with the iTextos SMS service.

This guide assumes that you are familiar with and have experience in applying the following technologies and concepts:
- Invoking the HTTP API request using Query String payload.

Feature Supported by this API

The features supported by this Query String API are as follows.

- Simple Basic authentication protocol in the header
- One 2 One (Single message to a Single Recipient)
- One 2 Many (Single message to Multiple Recipients)

Parameters Specification

This API can include the following parameters as per the requirement. Any other parameters will be considered invalid and ignored.

Parameters Supported by the API

Parameter Type	Description	Comments	Parameter Name	Is Mandatory	Possible values	Default value
Account Id	A unique key to identify the customer.	Key will be provided by iTextos.	aid	Yes	XXX…X	Not applicable
PIN	Password for the user		pin	Yes	XXX…X	Not applicable
MSISDN	To specify the mobile number	The mobile numbers should be prepended with the proper country code.	mnumber	Yes	XXX	Not applicable
MESSAGE TEXT	Message to be sent out	No limitation for the number of characters in the message	message	Yes	XXX	Not applicable
From Address	To specify the Header	Max Length = 6	signature	Yes	XXX	Not applicable
Content-Type	Content-Type of the message		msgType	No	FLASH — FL PLAIN — PM UNICODE — UC BINARY — BM FLASH UNICODE — FU SPECIFIC PORT — SP SPECIFIC PORT UNICODE — SPU	PLAIN
Port	To specify the specific Ports to deliver the messages on those ports	Max Length = 5 Min Value = 0 Max Value = 65535	port	No	99999	Not applicable

Parameter Type	Description	Comments	Parameter Name	Is Mandatory	Possible values	Default value
Send delivery notification	To specify about the DLR is to be sent back or not. As per the account level configuration, the DLR will be sent back in XML API.		dlrtype	No	1 / 0	0
Validity Period	To specify the validity period for this message request.	The value will be treated in minutes. **Minimum = 1** **Maximum = 1440 (one day)**	expiry	No	9999	Not applicable
Template Id	To use a specific template for this customer for this message.		template_id	No	XXXXX…X	Not applicable
Template Values	The template values (dynamic values) to be used for the given template Id.	Supports a maximum of 10 dynamic parameters. The values are separated by tilde character.	template_values	No	template_values=XXXX~YYYY	Not Applicable
Customer Reference ID	A unique id sent by the customer for every message	Max Length = 50	custref	No	XXX	Not applicable
DLT Entity Id	To specify the DLT Entity Id		dlt_entity_id	No	XXX	Not applicable
DLT Template Id	To specify the DLT Template Id		dlt_template_id	No	XXX	Not applicable

Parameter Type	Description	Comments	Parameter Name	Is Mandatory	Possible values	Default value
Message Msg Tag	To specify the Message tag	Max Length = 50	mtag	No	XXX	Not applicable

Note:

i.   For the Message Text parameter, there is no limitation for the number of characters in the message.

ii.   After the submission of the message to the operator, the operator is the sole responsible for the validity period of the message.

iii. It is mandatory to provide the values to the Template value parameter whenever the template id parameter is used.

Note on Content Types

1.   By default, the API assumes that the incoming request is on the UTF-8 charset. If the customer wants to send in a different charset, then they have to specify the charset in the HTTP request's content type.

2.   The content type (type parameter) can be any one of the following.

	Content-Type	Value
a	FLASH	FL
b	PLAIN	PM
c	UNICODE	UC
d	BINARY	BM
e	FLASH UNICODE	FU
f	SPECIFIC PORT	SP
g	SPECIFIC PORT UNICODE	SPU
h	ADVANCE	AD

3.   In the case of the type 'a', 'b', 'c',' d' there is no need to set the DCS value.

4.   When the type is 'e', then the customers have to specify the DCS value based on the industry standard. The same will be sent to the SMSCs.

5.   **Note:** The UDH and DCS values are applicable only for the 'ADVANCE' Content-type.

Authorization

- The authorization of user will be done through the **accesskey** parameter or using the **Basic Authentication** through the HTTP headers.

- The preferred way is to use the **accesskey** parameter. *As all the requests are received through the HTTPS protocol no need worry about the security constraints*.

- In case authorization is using the accesskey parameter, refer to the Request Format for the sample given in the next section.

- In the case of using the Basic Authentication, we need to pass the header as below. The Key is the Base64 Encoded value of user and accesskey combination as `<user>:<accesskey>`

```
'Authorization': 'Basic <KEY>'
```

Sample Query String API

Request Format

The sample requests of this QUERY STRING API will be as follows.

One 2 One (single message to a single recipient)

It refers specifically to the sending of a single message to a single recipient.

```
http://<HOST:PORT>/genericapi/QSCustomReceiver?aid=XXXXX&pin=YYYYY&
mnumber=911234567890&signature=HEADER&message=Dear%20manthan1%20cus
tomer%20this%20is%20sample%20test%20message.&msgType=PM
```

One 2 Many (single message to multiple recipients)

- One 2 Many SMS is a type of group SMS in which the same message is broadcast to a large number of recipients.

- This feature enables the user to send a single message content to multiple mobile numbers. In this scenario, all the predetermined mobile numbers will receive the same format of message content.

- The mobile numbers will be given in the dest parameter and each mobile number will be separated by a comma (,) special character. For example, dest=9876543210,9876543210.

```
http://<HOST:PORT>/genericapi/QSCustomReceiver?aid=XXXXX&pin=YYYYY
&mnumber=911234567890,911234567899&signature=HEADER&message=Dear%2
0manthan1%20customer%20this%20is%20sample%20test%20message.&msgTyp
e=PM&expiry=10&custref=CustomerRef&dlrtype=0
```

Response Format

The below section outline the response format for every request that was successfully received by the iTextos messaging platform.

Below contents are formatted responses for reference and understanding. In real-time, the responses will not be as formatted as shown.

Note:

1. All the error responses will be moreover same. But the status code and reason will differ for every rejection.
2. Some of the failure / rejection will have the req_id in the responses. For others it will be Not applicable (N / A).

Success Response

The following sample response format would be received when the request was successfully accepted by the platform.

```
{
    "req_id":"12345678901234567890123",
    "req_time":"yyyy-MM-dd HH:mm:ss.SSS",
    "status":{
        "reason":"Application Accepted",
        "code":"200",
        "info":"ACCEPTED"
    }
}
```

Failure Response

The following sample response format would be received when the request was not accepted by the platform.

```
{
    "req_id":"N/A",
    "req_time":"yyyy-MM-dd HH:mm:ss.SSS",
    "status":{
        "reason":"<Error Description>",
        "code":"<Error Code>",
        "info":"REJECTED"
    }
}
```

Status Code Reference

The following API status and errors codes that you may get in case there is a wrong parameter entered or issue with the account:

Status Code	Status Description	Comment
200	Success	
305	Account Deactivated	

306	Invalid Credentials	
307	Account Expired	
308	Account Suspended	
309	Account Inactivated	
311	Empty Mobile Number	
312	Mobile Number is not valid	
313	Empty Message	
317	Message Type is not valid	
318	Destination Port is not valid	
319	Expiry Value is not valid	
320	Invalid Expiry Time	
324	Customer Reference Id exceeds length	
325	Message Send failed - TRA Blockout Time	
328	Invalid Data Coding Value	
329	Invalid Header - Empty Header	
330	Invalid Header	
331	Unauthorized Request	
336	DLT Entity Id is not valid	
337	DLT Template Id is not valid	
338	Invalid Source IP	
339	API Service Disabled	
340	International Service Disabled	
398	Internal Server Error	
399	Unknown Error Code	

Sample Format of Possible Responses

The possible responses will be as follows

Success

Error Code: 200	Description: The message is successfully accepted by the interface and proceed for the further processing.

```
{
   "req_id":"12345678901234567890123",
   "req_time":"yyyy-MM-dd HH:mm:ss.SSS",
   "status":{
      "reason":"Application Accepted",
      "code":"200",
      "info":"ACCEPTED"
   }
}
```

Account Deactivated

Error Code: 305	Description: This error response is thrown when the Access key is valid but the account is deactivated.

```
{
   "req_id": "N/A",
   "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
```

```
    "status": {
        "reason": "Account Deactivated ( Account Deactivated )",
        "code": "305",
        "info": "REJECTED"
    }
}
```

Invalid Credential

Error Code: 306	Description: This error response is thrown when the user didn't provide the PIN parameter
| ```
{
    "req_id": "N/A",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Invalid Credentials",
        "code": "306",
        "info": "REJECTED"
    }
}
``` | |

Error Code: 306	Description: This error response is thrown when the user provided access key parameter is not valid
| ```
{
    "req_id": "N/A",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Invalid Credentials ( Account Invalid )",
        "code": "306",
        "info": "REJECTED"
    }
}
``` | |

Account Expired

Error Code: 307	Description: This error response is thrown when the Access key is valid but the account is expired.
| ```
{
    "req_id": "N/A",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Account expired ( Account Expired )",
        "code": "307",
        "info": "REJECTED"
    }
}
``` | |

Account Suspended

Error Code: 308	Description: This error response is thrown when the Access key is valid but the account is expired.
| ```
{
    "req_id": "N/A",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Account Suspended ( Account Suspended )",
        "code": "308",
        "info": "REJECTED"
    }
}
``` | |

Account Inactivated

Error Code: 309	Description: This error response is thrown when the Access key is valid but the account is in inactive mode.
| ```
{
    "req_id": "N/A",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Account Inactive ( Account Inactive )",
        "code": "308",
        "info": "REJECTED"
    }
}
``` | |

Empty Mobile Number

Error Code: 311	Description: This error response is thrown when the dest parameter is not passed
| ```
{
    "req_id": "123456789012345678902",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Empty Mobile Number",
        "code": "311",
        "info": "REJECTED"
    }
}
``` | |

Invalid Mobile Number

Error Code: 312	Description: This error response is thrown when the dest parameter is passed and the mobile not does meet the required lengths.
| ```
{
    "req_id": "123456789012345678902",
``` | |

```
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Mobile Number is not valid",
        "code": "312",
        "info": "REJECTED"
    }
}
```

Empty Message

Error Code: 313	Description: This error response is thrown when the dest parameter is passed and the mobile not does meet the required lengths.
| ```
{
    "req_id": "1234567890123456789012",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Empty Message",
        "code": "313",
        "info": "REJECTED"
    }
}
``` | |

Invalid Message Type

Error Code: 317	Description: This error response is thrown when the request is having invalid message type than the predefined one.
| ```
{
    "req_id": "12345678901234567890123",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Message Type is not valid",
        "code": "317",
        "info": "REJECTED"
    }
}
``` | |

Destination Port Invalid

Error Code: 318	Description: This error response is thrown when the request is having port specific message type and invalid port is specified.
| ```
{
    "req_id": "12345678901234567890123",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Destination Port is not valid",
        "code": "318",
        "info": "REJECTED"
    }
}
``` | |

Invalid Expire Value

Error Code: 319	Description: This error response is thrown when the request is having expiry parameter, but it is not a valid number.

```
{
    "req_id": "12345678901234567890123",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Expiry Value is not valid",
        "code": "319",
        "info": "REJECTED"
    }
}
```

Invalid Expire Time

Error Code: 320	Description: This error response is thrown when the request is having expiry parameter, but it is not within the applicable time range.

```
{
    "req_id": "12345678901234567890123",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Invalid Expiry Time",
        "code": "320",
        "info": "REJECTED"
    }
}
```

URL tracking is not valid

Error Code: 322	Description: This error response is thrown when the request is not having a proper formatted value in the message.

```
{
    "req_id": "12345678901234567890123",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "URL Track is not valid",
        "code": "322",
        "info": "REJECTED"
    }
}
```

Customer Reference Id exceeds length

Error Code: 324	Description: This error response is thrown when the request is having customer reference id more length than allowed length.

```
{
    "req_id": "12345678901234567890123",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Customer Reference Id exceeds length",
        "code": "324",
        "info": "REJECTED"
    }
}
```

TRA Blockout

Error Code: 325	Description: This error response is thrown when the request within the TRA blockout time.

```
{
    "req_id": "12345678901234567890123",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Message Send failed - TRA Blockout Time",
        "code": "325",
        "info": "REJECTED"
    }
}
```

Empty Header

Error Code: 329	Description: This error response is thrown when the request has an Empty Header.

```
{
    "req_id": "12345678901234567890123",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Invalid Header - Empty Header",
        "code": "329",
        "info": "REJECTED"
    }
}
```

Invalid Header

Error Code: 330	Description: This error response is thrown when the request does not have the a proper header as per the specification.

```
{
    "req_id": "12345678901234567890123",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Invalid Header",
```

```
        "code": "330",
        "info": "REJECTED"
    }
}
```

Unauthorized Request

Error Code: 331	Description: This error response is thrown when the request has received in non specified segment type. For Example: Promo messages in OTP segments.
| ```
{
    "req_id": "12345678901234567890123",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Unauthorized Request",
        "code": "331",
        "info": "REJECTED"
    }
}
``` | |

Invalid DLT Entity Id

Error Code: 336	Description: This error response is thrown when the request has passed an DLT Entity ID and that is not in proper length.
| ```
{
    "req_id": "12345678901234567890123",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "DLT Entity Id is not valid",
        "code": "336",
        "info": "REJECTED"
    }
}
``` | |

Invalid DLT Template Id

Error Code: 337	Description: This error response is thrown when the request has passed an DLT Template ID and that is not in proper length.
| ```
{
    "req_id": "12345678901234567890123",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "DLT Template Id is not valid",
        "code": "337",
        "info": "REJECTED"
    }
}
``` | |

Invalid Source IP

Error Code: 338	Description: This error response is thrown when the request received from a non whiltlisted IP for that customer.

```json
{
    "req_id": "N/A",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "Invalid source IP",
        "code": "338",
        "info": "REJECTED"
    }
}
```

API Service Disabled

Error Code: 339	Description: This error response is thrown when the user does not have API Service enabled.

```json
{
    "req_id": "N/A",
    "req_time": "yyyy-MM-dd HH:mm:ss.SSS",
    "status": {
        "reason": "API Service Disabled",
        "code": "339",
        "info": "REJECTED"
    }
}
```

Internal Error

Error Code: 398	Description: This error response is thrown whenever an internal error occurs.

```json
{
   "req_id":"N/A",
   "req_time":"yyyy-MM-dd HH:mm:ss.SSS",
   "status":{
      "reason":"Internal Error",
      "code":"398",
      "info":"REJECTED"
   }
}
```

Unknown Error Code

Error Code: 399	Description: This error response is thrown when unable to process the request due to some reason.

```json
{
   "req_id":"N/A",
```

```
    "req_time":"yyyy-MM-dd HH:mm:ss.SSS",
    "status":{
       "reason":"Unknown Error Code",
       "code":"399",
       "info":"REJECTED"
    }
}
```