



ITEXMO SMS API DOCUMENTATION v.2

TABLE OF CONTENTS

Overview	1
Prerequisites	1
Request Headers	2
• Authorization Header (PHP cURL)	3
• Authorization Header (Laravel-Guzzle)	4
• Authorization Header (Postman)	5
SENDING A MESSAGE API	6-8
SENDING MULTIPLE MESSAGES API	9-10
SENDING OTP MESSAGE API	11-14
RECEIVE DELIVERY REPORTS	15-18
QUERY API	19-20
Sample Response	21-33
SMS Length Calculation	34

OVERVIEW

ITEXMO SMS SERVICES INC. is made to help its clients integrate messaging (SMS) capabilities to their system. All connections in our API require authentication and are secured by TLS/SSL.

This paper provides the complete list of the APIs where clients can send, manage and/or view their messages.

The latest ITEXMO SMS API (with endpoint ***api.itexmo.com***) features a RESTful architecture and supports JSON media type.

PREREQUISITES

Following are the prerequisites for ITEXMO SMS API integration

- ***itexmo.com***
 - All request should contain ITEXMO credentials (ApiCode and Password) for authentication;
 - The requesting system should have an active ITEXMO credentials in our system;
 - Requests must contain all the valid information to perform operation on ITEXMO.
- ***api.itexmo.com***
 - All the requests should contain ITEXMO credentials (Email and Password) for authentication;
 - The requesting system should have an active APICODE in our system;
 - Requests must contain all the valid information to perform operation on ITEXMO.

REQUEST HEADERS

Requests to our API must contain ITEXMO credentials in request params. API requests are authenticated by the following HTTP params.

itexmo.com

Parameter	Expected Value	Description	Example
ApiCode	Required, String	ApiCode of active ITEXMO account	PR-SAMPL123456_ABCDE
passwd	Required, String	Password of active ITEXMO account	123456789ABCD

api.itexmo.com

Parameter	Expected Value	Description	Example
Email	Required, String, Email format	Email of active ITEXMO account	itexmoclient@gmail.com
Password	Required, String	Password of active ITEXMO account	123456789ABCD

Those are two parameters needed to access the ITEXMO API. All other parameters are properly validated in the request. If client use a wrong value or missing a required parameter, then API will provide the appropriate error messages.

Note: If the client doesn't want to include the following ITEXMO credentials in the requested params, the client can use an Authorization Header- Basic Auth which the latest API (*api.itexmo.com*) supports.

Adding Basic Auth using PHP cURL

(api.itexmo.com)

```
$endpoint = 'https://api.itexmo.com/api/broadcast';
$email = 'itexmoclient@gmail.com'; // Client email
$password = '123456789ABCD'; // Client password
$ch = curl_init($endpoint); // Initiate cURL
$headers = array(
    'Content-Type: application/json',
    'Authorization: Basic '. base64_encode("$email:$password")
); // Create the headers array containing authorization
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers); // Set the headers
// Set other cURL options here such as http method, data, etc.
$response = curl_exec($ch); // Execute the cURL request
if (curl_errno($ch)) { // Check for errors.
    echo curl_error($ch); // If an error occurred
}
echo $response; // Print out the response
curl_close ($ch); // Close cURL
```

Adding Basic Auth using Laravel Guzzle

(api.itexmo.com)

```
...  
$content = [  
    "ApiCode" => "APICODEABCD1234",  
    // Declare other data for contents  
]; // Must contain the data for the request  
$endpoint = 'https://api.itexmo.com/api/broadcast';  
$email = 'itexmoclient@gmail.com'; // Client email  
$password = '123456789ABCD'; // Client password  
$client = new \GuzzleHttp\Client();  
$response = $client->post($endpoint, [  
    'headers' => [  
        'Content-Type' => 'application/json',  
        'Accept' => 'application/json',  
        'Authorization' => 'Basic ' . base64_encode("$email:$password")  
    ],  
    'body' => json_encode($content)  
]);  
...
```

Adding Basic Auth using Postman

(api.itexmo.com)

Select POST method. Set the request url. In the Authorization Menu, select Basic Auth type. Type the email in username field and password in password field. Assuming the Body Menu has necessary and valid data, clicking Send will return the expected response.

The screenshot shows the Postman interface for a POST request. The method is set to POST, and the URL is https://api.itexmo.com/api/broadcast. The Authorization tab is selected, and the type is set to Basic Auth. The username is itexmoclient@gmail.com and the password is 123456789ABCD. The Show Password checkbox is checked.

POST	https://api.itexmo.com/api/broadcast	Send
Params	Authorization	Headers (9)
Type	Basic Auth	Username
The authorization header will be automatically generated when you send the request. Learn more about authorization		Password
		itexmoclient@gmail.com
		123456789ABCD
		<input checked="" type="checkbox"/> Show Password

Note: For reference, ApiCodes with server type of Direct are called Sender ID ApiCodes, while the rest of ApiCodes with different server type (Shared or Dedicated) is called Non-Sender ID ApiCodes.

SENDING A MESSAGE API

This can be used by an authorized ITEXMO account to send a message directly to a recipient/s either with senderid or without.

Sending A Message API requests are protected by TLS and uses the following base URL:

Old Production: <https://itexmo.com/api/broadcast/>
<https://itexmo.com/api/broadcast/v2/>

New Production: <https://api.itexmo.com/api/broadcast>

Following are the requests to call the Sending A Message API:

itexmo.com

Parameter	Expected Value	Description	Example
ApiCode	Required, String	ApiCode of active ITEXMO account	PR-SAMPL123456_ABCDE
passwd	Required, String	Password of active ITEXMO account	123456789ABCD
SenderID	Optional, String	Only for senderid package type of ApiCode, the registered brand in Telco. If left blank, system will use the first found registered Sender ID under client's ApiCode. If none registered, will use ITEXMO SMS.	ITEXMO SMS
Numbers	Required, Array, Unique	Recipient/s number (maximum: 250)	["09123456789", "09123456788"]
Message	Required, String	Message content	Test message.

api.itexmo.com

Parameter	Expected Value	Description	Example
Email	Required, String, Email format	Email of active ITEXMO account	itexmoclient@gmail.com
Password	Required, String	Password of active ITEXMO account	123456789ABCD
ApiCode	Required, String	Active ApiCode purchased under client's ITEXMO account	PR-SAMPL123456_ABCDE
SenderId	Optional, String	Only for senderid package type of ApiCode, the registered brand in Telco. If left blank, system will use the first found registered Sender ID under client's ApiCode. If none registered, will use ITEXMO SMS.	ITEXMO SMS
ClientId	Optional, String (Alpha-Dash)	Client's transaction id, maximum of 50 characters	123ABC
Recipients	Required, Array, Unique	Recipient/s number (maximum: 250)	["09123456789", "09123456788"]
Message	Required, String	Message content	Test message.

Sample JSON Content (RAW)

itexmo.com

```
{
  "ApiCode": "PR-SAMPL123456_ABCDE ",
  "passwd": "123456789ABCD",
  "Numbers": [
    "09123456789",
    "09123456788"
  ],
  "Message": "Test message.",
  "SenderId": "ITEXMO SMS"
}
```

api.itexmo.com

```
{
  "Email": " itexmoclient@gmail.com",
  "Password": "123456789ABCD",
  "Recipients": [
    "09123456789",
    "09123456788"
  ],
  "Message": "Test message.",
  "ApiCode": "PR-SAMPL123456_ABCDE",
  "SenderId": "ITEXMO SMS"
}
```

Note: In *api.itexmo.com*, if client provides both Email and Password in the request parameters and in authorization header, the request parameters will be ignored and the basic authentication values will be used.

SENDING MULTIPLE MESSAGES API

This can be used by an authorized ITEXMO account with active APICODE to send different messages directly to various recipients either with senderid or without. This is only available on **api.itexmo.com**.

Sending Multiple Message API requests are protected by TLS and uses the following base URL:

Production: <https://api.itexmo.com/api/broadcast-2d>

Following are the requests to call the Sending Multiple Message API:

Parameter	Expected Value	Description	Example
Email	Required, String, Email format	Email of active ITEXMO account	itexmoclient@gmail.com
Password	Required, String	Password of active ITEXMO account	123456789ABCD
ApiCode	Required, String	Active ApiCode purchased under client's ITEXMO account	PR-SAMPL123456_ABCDE
Contents	Required, Array	Contains the recipient and corresponding message (maximum: 250) Must have unique recipient in every request.	<pre>[{ "Message": "Test message 1.", "Recipient": "09123456789" }, { "Message": "Test message 2.", "Recipient": "09123456788" }]</pre>
SenderId	Optional, String	Only for senderid package type of ApiCode, the	ITEXMO SMS

		registered brand in Telco. If left blank, system will use the first found registered Sender ID under client's ApiCode. If none registered, will use ITEXMO SMS.	
ClientId	Optional, String (Alpha-Dash)	Client's transaction id, maximum of 50 characters	123ABC
PriorityLevel	Optional, String	The priority level of message. The default is NORMAL. HIGH means the request will be prioritized among client's other request.	HIGH * Only for non-senderid package with dedicated type of server.

Sample JSON Content (RAW)

```
{
  "Email": "itexmoclient@gmail.com",
  "Password": "123456789ABCD",
  "Contents": [
    {
      "Message": "Test message 1.",
      "Recipient": "09123456789"
    },
    {
      "Message": "Test message 2.",
      "Recipient": "09123456788"
    }
  ],
  "ApiCode": "PR-SAMPL123456_ABCDE",
  "SenderId": "ITEXMO SMS"
}
```

SENDING OTP MESSAGE API

This can be used by an authorized ITEXMO account with DIRECT SERVER TYPE APICODE (with Sender ID) to send OTP message directly to a recipient.

Sending OTP Message API requests are protected by TLS and uses the following base URL:

Old Production: <https://itexmo.com/api/otp/>

New Production: <https://api.itexmo.com/api/broadcast-otp>

Following are the requests to call the Sending Multiple Message API:

itexmo.com

Parameter	Expected Value	Description	Example
ApiCode	Required, String	ApiCode of active ITEXMO account	PR-SAMPL123456_ABCDE
passwd	Required, String	Password of active ITEXMO account	123456789ABCD
SenderID	Optional, String	The registered brand in Telco. If left blank, system will use the first found registered Sender ID under client's ApiCode. If none registered, will use ITEXMO SMS.	ITEXMO SMS
Numbers	Required, Array	Recipient/s number (maximum: 1)	["09123456789"]
Message	Required, String	Message content	OTP 12345.

api.itexmo.com

Parameter	Expected Value	Description	Example
Email	Required, String, Email format	Email of active ITEXMO account	itexmoclient@gmail.com
Password	Required, String	Password of active ITEXMO account	123456789ABCD
ApiCode	Required, String	Active ApiCode purchased under client's ITEXMO account	PR-SAMPL123456_ABCDE
SenderId	Optional, String	The registered brand in Telco. If left blank, system will use the first found registered Sender ID under client's ApiCode. If none registered, will use ITEXMO SMS.	ITEXMO SMS
ClientId	Optional, String (Alpha-Dash)	Client's transaction id, maximum of 50 characters	123ABC
Recipients	Required, Array	Recipient/s number (maximum: 1)	["09123456789"]
Message	Required, String	Message content	OTP 12345.
OtpResendCount	Optional, Integer	You may pass counter value for OTP Message. Example, in your platform the user clicked resend, so pass the value 1, if again clicked, increment.	1 * This data will be helpful to trace the counts of resent OTP Messages.

Sample JSON Content (RAW)

itexmo.com

```
{
  "ApiCode": "PR-SAMPL123456_ABCDE ",
  "passwd": "123456789ABCD",
  "Numbers": ["09123456789"],
  "Message": "Test message.",
  "SenderId": "ITEXMO SMS"
}
```

api.itexmo.com

```
{
  "Email": " itexmoclient@gmail.com",
  "Password": "123456789ABCD",
  "Recipients": [
    "09123456789"
  ],
  "Message": "OTP 12345.",
  "ApiCode": "PR-SAMPL123456_ABCDE",
  "SenderId": "ITEMXO SMS"
}
```

Note: OTP messages are expected to be short in character length. With this, our system allows maximum of 160 characters for OTP regardless of the maximum characters client's apicode is allowed to.

In addition, to Send OTP Message API, we also have an endpoint that will enable us to calculate the **Conversion Rate of OTP Messages**.

The Conversion Rate is calculated by dividing the number of converted OTP messages by the number of sent OTP messages.

OTP Message can be tagged as Converted OTP Message only if received by the corresponding recipient and was able to use the OTP to wherever platform it is intended.

Our system saves logs and generates reports of all our client OTP message requests, hence the number of sent OTP messages. The data of converted OTP messages will be and should be passed by the client. We can count the numbers of converted OTP messages from that data.

Reminder, this feature is only for clients that has an active apicode in our new dashboard (<https://dashboard.itexmo.com>).

To provide the data of the converted OTP message use the endpoint below and follow the parameters provided in the table after that.

New Production: <https://api.itexmo.com/api/converted-report>

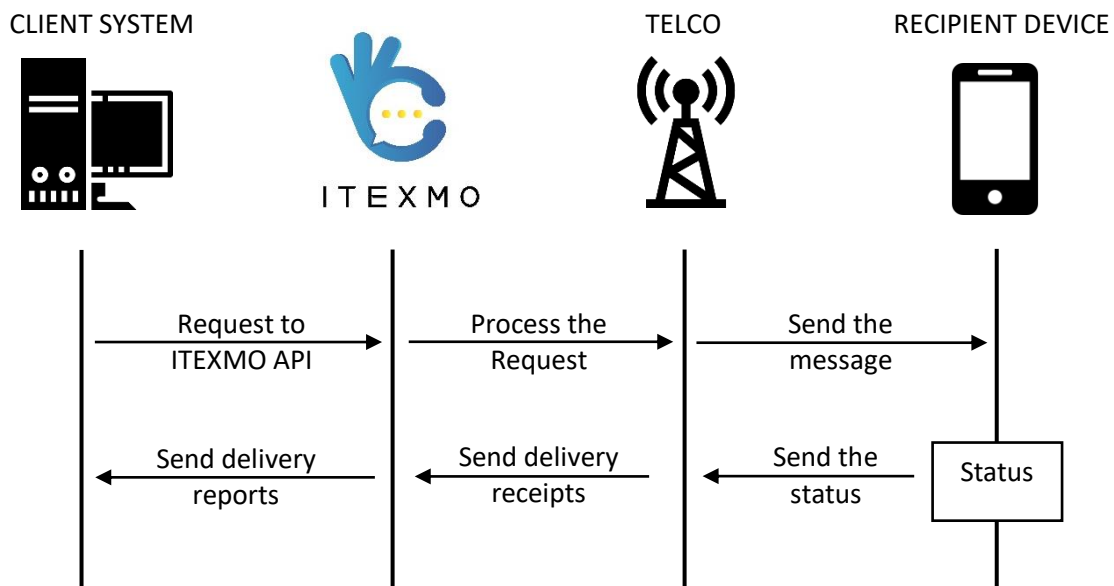
Parameter	Expected Value	Description	Example
Email	Required, String, Email format	Email of active ITEXMO account	itexmoclient@gmail.com
Password	Required, String	Password of active ITEXMO account	123456789ABCD
ApiCode	Required, String	Active ApiCode purchased under client's ITEXMO account	PR-SAMPL123456_ABCDE
Recipient	Required, String	Recipient number	09123456789
Content	Required, String	OTP message content	Your OTP is 12345.

Message key with value SUCCESS will be the response of the above endpoint if all data passed are valid.

RECEIVE DELIVERY REPORTS

This can be used by an authorized ITEXMO account to send message/s directly to recipient/s with senderid and received a delivery report.

Whenever client sends a request through our APIs, our system provide a Long ID which serves as the reference number for every request our system processed. That only gives information from our side. If the client wanted additional information from the side of recipient/s of their request, our system can provide an acknowledgement or delivery reports. For the client to understand more how this works, check the diagram below.



Sending Message With Delivery Reports are protected by TLS and uses the following base URL:

Old Production: <https://itexmo.com/api/broadcast/v2/>

<https://itexmo.com/api/otp/>

New Production: <https://api.itexmo.com/api/broadcast>

<https://api.itexmo.com/api/broadcast-2d>

<https://api.itexmo.com/api/broadcast-otp>

Noticeable above that the base URL is the same with the API we provided from previous pages. This means that the request parameters are the same but with additional.

Following are the additional request parameters when client want to receive a delivery report:

itexmo.com

Parameter	Expected Value	Description	Example
DeliveryReports	1	This will serve as trigger that Delivery Reports is necessary. The value must always be 1. Not declaring this will not make Delivery Reporting successful.	1
DeliveryURL	URL Format	The URL in which our system will push data for the delivery report.	https://client.com/ /delivery-reports

api.itexmo.com

Parameter	Expected Value	Description	Example
DeliveryReportsURL	URL Format	The URL in which our system will push data for the delivery report.	https://client.com/ /delivery-reports

Possible values of Delivery Status- **DELIVERED** (delivered to destination), **EXPIRED** (validity period- 8 hrs. has expired), **DELETED** (message has been deleted), **UNDELIVERABLE** (undelivered), **ACCEPTED** (accepted by Telco), **UNKNOWN** (invalid), and **REJECTED** (not accepted).

The URL provided in the DeliveryReportsURL/ DeliveryURL params should handle the data our system will push. Check the details below:

METHOD:	POST		
CONTENT TYPE:	FORM URL ENCODED (application/x-www-form-urlencoded)		
PARAMETERS:	'LongID'	=>	ReferencelD of the request
	'Recipient'	=>	Recipient of the message
	'Status'	=>	Status of the message
	'NetworkSubmitTime'	=>	Time when platform submits the request to the Telco.
	'ClientSubmitTime'	=>	Time the platform receives the request
	'DoneTime'	=>	Expiration time for the request (8 hours)

Note: For multiple recipients in a request, our system will send individual delivery reports for each recipient in the request.

Sample JSON Content (RAW)

itexmo.com

```
{
  "ApiCode": "PR-SAMPL123456_ABCDE ",
  "passwd": "123456789ABCD",
  "Numbers": ["09123456789",],
  "Message": "Test message 12345.",
  "SenderID": "ITEXMO SMS",
  "DeliveryReports": "1"
  "DeliveryURL": "https://client.com/delivery-reports"
}
```

api.itexmo.com

```
{
  "Email": "itexmoclient@gmail.com",
  "Password": "123456789ABCD",
  "Contents": [
    {
      "Message": "Test message 1.",
      "Recipient": "09123456789"
    },
    {
      "Message": "Test message 2.",
      "Recipient": "09123456788"
    }
  ],
  "ApiCode": "PR-SAMPL123456_ABCDE",
  "SenderId": "ITEXMO SMS",
  "DeliveryReportsURL": "https://client.com/delivery-reports"
}
```

QUERY API

This can be used by an authorized ITEXMO account with active APICODE to query various data. This is only available on ***api.itexmo.com***.

Query API requests are protected by TLS and uses the following base URL:

Production: <https://api.itexmo.com/api/query>

Following are the requests to call the Sending Multiple Message API:

Parameter	Expected Value	Description	Example
Email	Required, String, Email format	Email of active ITEXMO account	itexmoclient@gmail.com
Password	Required, String	Password of active ITEXMO account	123456789ABCD
ApiCode	Required, String	Active ApiCode purchased under client's ITEXMO account	PR-SAMPL123456_ABCDE
Action	Required, Any from the following: 1. ServerStatus 2. ApiCodeInfo 3. PendingSMS 4. SMSReport For clients with ApiCode under Postpaid Plan, use the following: 1. PostpaidServerStatus 2. PostpaidApiCodeInfo 3. PostpaidPendingSMS 4. PostpaidSMSReport	ServerStatus- return information about the server of client's ApiCode such as type, status, etc. ApiCodeInfo- return information about the ApiCode client availed such as plan, character limit, messages left, total credit usage, filtering status, and expiration date. PendingSMS- return the count of pending messages and its corresponding	ServerStatus

		<p>information such as recipient, message content, sender id, delivery report url, and long id.</p> <p>SMSReport- return the count of SMS sent belonging to the Client or Reference ID and its corresponding information such as recipient, message, credit used, reference id, time requested, time sent, delay, etc.</p>	
--	--	--	--

Sample JSON Content (RAW)

```
{
  "Email": " itexmoclient@gmail.com",
  "Password": "123456789ABCD",
  "Action ": "ServerStatus",
  "ApiCode": "PR-SAMPL123456_ABCDE"
}

{
  "Email": " itexmoclient@gmail.com",
  "Password": "123456789ABCD",
  "Action ": "SMSReport",
  "ApiCode": "PR-SAMPL123456_ABCDE",
  "ReferenceId": "PRSAM19394137945432625844"
}

{
  "Email": " itexmoclient@gmail.com",
  "Password": "123456789ABCD",
  "Action ": "SMSReport",
  "ApiCode": "PR-SAMPL123456_ABCDE",
  "ClientId": "123ABC"
}
```

SAMPLE RESPONSE

- **Successful request** (*api.itexmo.com*)- 200 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": false,
  "TotalSMS": 2,
  "Accepted": 2,
  "TotalCreditUsed": 2,
  "Failed": 0,
  "DeliveryReportStatus": "Pushing",
  "ReferenceId": "PRSAM19394137945432625844"
}
```

DateTime	return the date and time of the response.
Error	true if request failed, false if successful.
TotalSMS	returns the count of recipients in the request.
Accepted	returns the count of valid recipients accepted.
TotalCreditUsed	returns the credit used (1 credit per 160 character) multiplied to the value of Accepted.
Failed	return the count of failed SMS.
DeliveryReportStatus	return " Pushing " if delivery reports were possible for client's apicode, return " Rejected " otherwise.
ReferenceId	returns the long ID value.

Note: Error may return **false** but still have failed SMS. Example: among 10 recipients, 9 are valid the other 1 is not. The response will be:

```
"Error": false,
...
"Failed": 1,
"FailedList": [
  {
    ...
  }
]
...
```

See **Other Responses** for more information.

- **Successful request** (*itexmo.com*)- 200 Status Code

```
{
  "Result": "Success",
  "Accepted": 1,
  "Failed": 0,
  "TotalSMS": 1,
  "TotalCreditUsed": 1,
  "Reports": "Pushing",
  "RefID": "PRSAM19394137945432625844"
}
```

Result	returns the result of request.
Accepted	returns the count of valid recipients accepted.
Failed	returns the count of failed SMS.
TotalSMS	returns the count of recipients in the request.
TotalCreditUsed	returns the credit used (1 credit per 160 characters for GSM-7BIT) multiplied to the value of Accepted.
Reports	return "Pushing" if delivery reports were possible for client's apicode, return "Rejected" otherwise.
Referenceld	returns the long ID value.

Note: The old production (*itexmo.com*) returns Invalid Response for invalid request.
Example:

```
{
  "Result": 401,
  "Error": 3
}
```

To know more about the description of Error Code
please check <https://itexmo.com/Developers/apidocs.php>.

- **Successful request (Query- ServerStatus) - 200 Status Code**

```
{
  "DateTime": "2022-01-06 14:34:16",
  "ApiCode": "PR-SAMPL123456_ABCDE",
  "ServerType": "DIRECT",
  "ServerStatus": "ONLINE"
}
```

- **Successful request (Query- ApiCodeInfo) - 200 Status Code**

```
{
  "DateTime": "2022-01-06 14:34:16",
  "ApiCode": "PR-SAMPL123456_ABCDE",
  "CurrentPlan": "100K Initial",
  "MaxCharacters": 640,
  "MessagesLeft": 409995,
  "TotalCreditUsed": 5,
  "MessageFiltering": "YES",
  "ExpirationDate": "2023-01-05 12:53:00",
}
```

- **Successful request (Query- PendingSMS) - 200 Status Code**

```
{
  "Count": 2,
  "Contents": [
    {
      "Recipient": "09123456789",
      "Message": "Test message 1.",
      "SenderID": "ITEXMO SMS",
      "DeliveryReportURL": null,
      "ReferenceID": "PRSAM19394137945432625844"
    }, {
      "Recipient": "09123456788",
      "Message": "Test message 2.",
      "SenderID": "ITEXMO SMS",
      "DeliveryReportURL": null,
      "ReferenceID": "PRSAM19394137945432625845"
    }
  ]
}
```

- **Successful request (Query- SMSReport) - 200 Status Code**

```
{
  "Count": 1,
  "Contents": [
    {
      "Recipient": "09123456789",
      "Message": "Test message 1.",
      "CreditUsed": 1,
      "SenderID": "ITEXMO SMS",
      "PricePerCredit": "0.5",
      "TotalPrice": "0.5",
      "DeliveryStatus": "DELIVERED",
      "Type": "OTP",
      "ReferenceID": "PRSAM19394137945432625844",
      "TimeRequested": "2022-01-06 14:34:16",
      "TimeSent": "2022-01-06 14:34:16",
      "Delay": 0
    }
  ]
}
```

```
{
  "Count": 1,
  "Contents": [
    {
      "Recipient": "09123456789",
      "Message": "Test message 1.",
      "CreditUsed": 1,
      "ReferenceID": "DESAM19394137945953625882",
      "TimeRequested": "2022-01-06 14:34:16",
      "TimeSent": "2022-01-06 14:34:18",
      "Delay": 2
    }
  ]
}
```

- **Other Responses** (*api.itexmo.com*)

- Missing email - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "ITEXMO email is required."
}
```

- Missing password - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "ITEXMO password is required."
}
```

- Missing email and password - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "ITEXMO email and password is required."
}
```

- Invalid email or password - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "The credentials do not match our records. Kindly review your ITEXMO credentials."
}
```

- Email of account is not yet verified - *400 Status Code*

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "User email is not yet verified."
}
```

- Account information is not yet complete and phone is not verified - *400 Status Code*

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "Account is not yet fully verified."
}
```

- ITEXMO account is not active or suspended - *400 Status Code*

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "Account is not active."
}
```

- Delivery Reports URL is not in URL format - *400 Status Code*

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": {
    "DeliveryReportsURL": [
      "The delivery reports u r l must be a valid URL."
    ]
  }
}
```

- Missing required parameters - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": {
    "Recipients": [
      "The recipients field is required."
    ],
    "Message": [
      "The message field is required."
    ],
    "ApiCode": [
      "The api code field is required."
    ]
  }
}
```

- Recipient data format is invalid - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": {
    "Recipients": [
      "The recipients must be an array."
    ]
  }
}
```

- Invalid number for single recipient - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "Invalid number."
}
```

- Invalid number for multiple recipient (all are invalid) - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": [
    {
      "Recipient": "123",
      "Message": "Invalid number."
    },
    {
      "Recipient": "456",
      "Message": "Invalid number."
    }
  ]
}
```

- Invalid ApiCode - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "ApiCode not found."
}
```

- Expired ApiCode - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "ApiCode is expired."
}
```

- Inactive/suspended ApiCode - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "ApiCode is inactive/suspended."
}
```

- ITEXMO error - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "ITEXMO error. Contact support at support@itexmo.com."
}
```

- Message length exceed allowed length - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "Maximum allowed characters for message reached."
}
```

- For **Non Sender ID**, if daily limit reached - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "Maximum message per day reached. This will be reset by 12MN."
}
```

- For **Sender ID**, if SMS credits is zero - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "No more SMS credits left. Please top up to continue."
}
```

- For **Sender ID**, uses an unregistered Sender ID - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "Invalid or not registered Sender ID."
}
```

- Exceeded the 250 limits for Recipients - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "Maximum recipient count exceeded. Found 360 recipients.
Allowed recipient count is 250."
}
```

- For **Sender ID**, sms credits not zero but not enough - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "Insufficient SMS credits. Credits left 10. You are trying to send
237."
}
```


- For **OTP**, exceeded maximum character allowed - *400 Status Code*

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "Allowed maximum length for OTP is 160. You are trying to send a message with 161 characters."
}
```

- For **OTP** and **Non-Sender ID** - *400 Status Code*

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "OTP API is not supported for Non-Sender ID ApiCodes."
}
```

- Invalid contents (Sending Multiple Message API) - *400 Status Code*

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": [
    {
      "Content": {
        "Recipient": "123",
        "Message": "Test message."
      },
      "Message": "Invalid number."
    },
    {
      "Content": {
        "Recipient": "09123456789",
        "Message": null
      },
      "Message": "Missing Message data."
    }
  ]
}
```

- Successful request but with failed SMS (Sending A Message API) - 200 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": false,
  "TotalSMS": 3,
  "Accepted": 2,
  "TotalCreditUsed": 2,
  "Failed": 1,
  "FailedList": [
    {
      "Recipient": "123",
      "Message": "Invalid number."
    }
  ],
  "DeliveryReportsStatus": "Pushing",
  "ReferenceId": "PRSAM19394137945432625844"
}
```

(Sending A Message API)

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": false,
  "TotalSMS": 3,
  "Accepted": 2,
  "TotalCreditUsed": 2,
  "Failed": 1,
  "FailedList": [
    {
      "Recipient": "09123456789",
      "Message": "Duplicate number."
    }
  ],
  "DeliveryReportsStatus": "Rejected",
  "ReferenceId": "PRSAM19394137945432625844"
}
```

(Sending Multiple Message API) - 200 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": false,
  "TotalSMS": 2,
  "Accepted": 1,
  "TotalCreditUsed": 1,
  "Failed": 1,
  "FailedList": [
    {
      "Content": {
        "Recipient": "09123456789",
        "Message": ""
      },
      "Message": "Message is empty."
    }
  ],
  "DeliveryReportsStatus": "Pushing",
  "ReferenceId": "PRSAM19394137945432625844"
}
```

- For **Query**, missing action - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "Action is required."
}
```

- For **Query**, invalid action - 400 Status Code

```
{
  "DateTime": "2022-01-06 14:34:16",
  "Error": true,
  "Message": "Action is invalid."
}
```

SMS LENGTH CALCULATION

This section discussed the way our API calculates the length of the message client want to send. This length will determine the credits used/deducted from client's ApiCode SMS credits.

First, we define the type of encoding of SMS our system supports.

- GSM-7BIT

This is the standard character encoding for usage on GSM networks. Most commonly letters and symbols in various language are accepted in this encoding. The maximum character count GSM-7BIT can carry is 160 characters if one part only and 153 characters per part if more than 1 part.

- UTF16

This is the standard method of encoding Unicode character data. Specials characters, emoticons and any characters not found in GSM-7BIT character set are considered Unicode character. Unicode SMS is limited to 70 characters if 1 part only and 67 characters per part if more than 1 part.

Our system calculates SMS length with the above encodings. If the message client is trying to send has special characters, it automatically converts the calculation with UTF16, if none, GSM-7BIT, never both.

To fully understand, let's check this example. Client sent a message with 90 characters without special characters, that is 1 SMS Credit. Client sent again with 160 characters without special characters again, still 1 SMS Credits. If client sent a message more than 160 characters, like 161, the SMS Credits equivalent is 2 (153/153 character for part 1, 8/153 characters for part 2). Just remember if the message without special character and is more than 160 characters, the rule will be 153 per part. If with special characters, 70 characters per part if only 1 part, but if more than 70, the rule will be 67 characters per part.