

API Documentation

Version 1.0

messito.com



Contents

- Overview.....	4
- Input/Output Specification & Formats.....	4
- Error Codes.....	4
- Definitions.....	5
- API Key Parameter.....	5
- Function/Method Reference.....	6
1 - /Job.....	6
- Description.....	6
- URL.....	6
- HTTP Verb: PUT.....	6
- Input Parameters.....	6
- Output JSON Object Schema.....	8
- Output JSON Object Schema for CallbackURL Delivery Reports.....	9
2 - /Messages.....	11
- Description.....	11
- URL.....	11
- HTTP Verb: GET.....	11
- Input Parameters.....	11
- Output JSON Object Schema.....	13
3 - /Jobs.....	16
- Description.....	16
- URL.....	16
- HTTP Verb: GET.....	16
- Input Parameters.....	16
- Output JSON Object Schema.....	18
4 - /Users.....	21
- Description.....	21
- URL.....	21
- HTTP Verb: GET.....	21
- Input Parameters.....	21
- Output JSON Object Schema.....	21
5 - /Credits.....	23
- Description.....	23
- URL.....	23
- HTTP Verb: GET.....	23
- Input Parameters.....	23
- Output JSON Object Schema.....	23
6 - Appendix – Code Snippets.....	25
- PHP.....	25
- Java.....	30
- C#.....	36

Overview

This document provides a detailed explanation of the functionality of the Messito API v1.0. In this first version, the API provides a subset of the functionality offered by the full Messito product, allowing for programmatic access to some product features. The API can thus be used for integration with existing IT eco-systems supporting the relevant business processes.

Input/Output Specification & Formats

This API adheres to the REST paradigm. As such, it will demand conformance to this protocol, using the relevant HTTP verbs (PUT – for create; GET – for read; POST – for update and DELETE – for delete) and present a Resource-Oriented paradigm to provide the functionality required by its users.

Despite REST specifically allowing for multiple data representations, this API will constrain all output representations to JSON, where it will only provide its output in this format as specified in the schemas below.

For security purposes, this API will employ the HTTPS protocol for functionality/data interchange.

Error Codes

The API will – in all cases – return one of the following error-codes (as the *resCode* field of a *result* JSON object – as will be shown below):

- 0: SUCCESS

This code (0) denotes that the operation invoked was completed successfully.

- -3: AUTHENTICATION-FAILURE

This code (-3) identifies a failed authentication attempt due to incorrect credentials (invalid username and/or invalid password and/or missing/invalid apikey).

- -5: ACCESS-DENIED

This code (-5) is returned in cases where the specific user accessing API methods does not possess the required access permissions to the corresponding method/functionality being invoked.

- -7: INVALID-PARAMETERS

This code (-7) denotes cases where the required input parameters to a method are incorrect or incomplete.

- -9: EMPTY-MESSAGE-TEXT

This code (-9) denotes cases where input message text/body submitted is blank/null/empty or when as a result of an enforced GSM7 message encoding, the message text/body submitted is blank as a result of dropped unsupported characters within the GSM7 encoding. (see [/Job](#)).

- -11: GENERIC-ERROR

This code (-11) denotes that the operation encountered some generic/unspecified errors whilst processing the request.

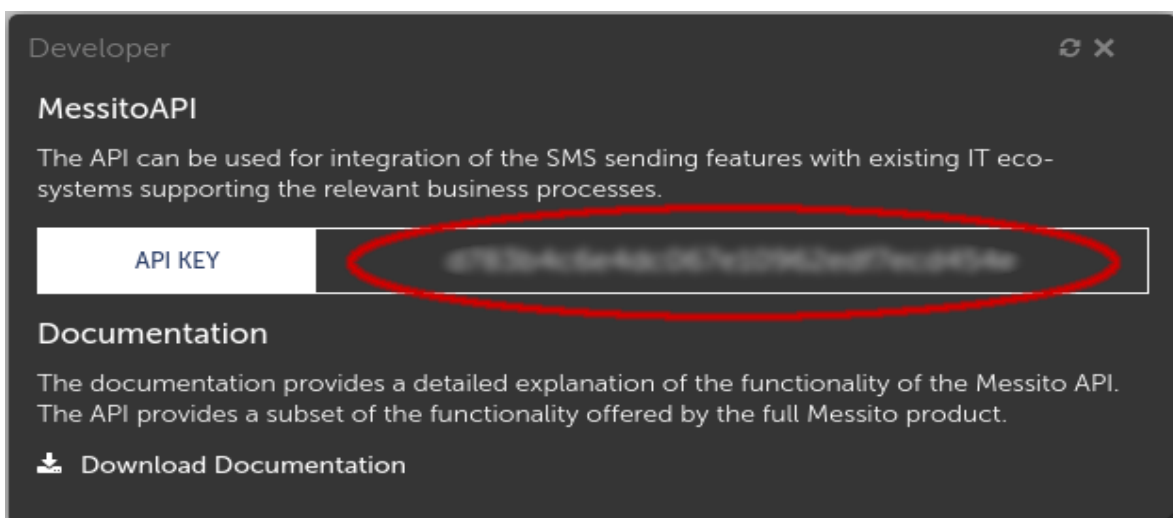
The Messito API will consistently return one of the above error codes for all method invocations. This code – along with more error specific detail where/when available – will be returned within a *result* JSON object as specified further below in each method specification detail.

Definitions

- **REST:** Representational state transfer (<http://www.ics.uci.edu/~taylor/documents/2002-REST-TOIT.pdf>).
- **JSON:** JavaScript Object Notation
- **Unix Time:** The number of seconds elapsed since 00:00:00 Coordinated Universal Time (UTC) on Thursday, 1 January 1970.
- **MCC:** Mobile Country Code.

API Key Parameter

In order to make use of this API, users will need to pass in an *apikey* parameter for each method call as listed in the method specifications below. In order to retrieve this value, log on to the Messito GUI (<http://messito.com/>) and navigate to *Settings > Developer* which displays this *apikey* value as shown below:



Function/Method Reference

1 /Job

Description

This method is used to submit an SMS transmission job via the API.

URL

```
https://api.messito.com/rest/Job
```

HTTP Verb: PUT

Input Parameters

These parameters are to be passed in within the body of the PUT request as HTTP standard *key=value* parameters in *application/x-www-form-urlencoded* mime-type encoding. Example:

```
username=username&password=password&apikey=apikey&name=name&messageText=The Message
Text&destinationNumbers="123456789,987654321,..."&source=Job Source&forceGSM7Text=false&callbackURL=
http://yourcallbackurl&token="User-defined token"
```

Name	Type	Req	Description/Notes
username	String	✓	The user-name of the specific user invoking the method.
password	String	✓	The authentication password of the specified user. <i>Note:</i> This parameter needs to be passed-in as an md5sum hash value.
apikey	String	✓	The application key issued to a Messito user upon registration to the system. <i>Note:</i> Refer to the API Key Parameter section above.
name	String	✓	A user-defined string value used by the API user to label the job.
messageText	String	✓	The actual SMS message content. <i>Note:</i> If the contents of this parameter are received as blank/null/empty, or – as a result

		<p>of the processing of the <i>forceGSM7Text</i> parameter (below) the resultant message text ends up as blank – then the API will return a -9 (<i>EMPTY-MESSAGE-TEXT</i>) error-code to denote this specific error and the Job is not processed.</p>								
destinationNumbers	List of Strings (comma separated)	<p>A list of destination mobile numbers (composed of digits [0-9]) to which the message is to be delivered.</p> <p><i>Note:</i> The numbers to be submitted need to include the country prefix without any leading zeros (e.g. '00') or plus (+) characters.</p> <p><i>Note:</i> This parameter can be used in conjunction with the contactListId parameter. Refer to the note below for further details.</p>								
contactListId	Integer	<p>An integer indicating the Contact List ID containing the list of destination numbers to which the message is to be sent. Such a list is created/maintained via the Messito GUI.</p> <p><i>Note:</i> Whilst this parameter is optional, a value has to be specified for one of these two parameters; furthermore, both parameters can be specified in conjunction in which case, the API will dispatch the message to each of the destination numbers specified within destinationNumbers and also for each destination number contained within the specified contactListID.</p> <p>As such, a call to this method can have the following combinations of these two parameters:</p> <table border="1" data-bbox="884 1675 1399 1895"> <thead> <tr> <th><i>destinationNumbers</i></th> <th><i>contactListId</i></th> </tr> </thead> <tbody> <tr> <td>✓</td> <td></td> </tr> <tr> <td></td> <td>✓</td> </tr> <tr> <td>✓</td> <td>✓</td> </tr> </tbody> </table> <p>Specifying neither of these parameters is invalid and the method call will be rejected.</p>	<i>destinationNumbers</i>	<i>contactListId</i>	✓			✓	✓	✓
<i>destinationNumbers</i>	<i>contactListId</i>									
✓										
	✓									
✓	✓									

source	String	✓	The sender source label which will be used when the SMS messages are delivered.
forceGSM7Text	Boolean (String – 'true' / 'false')		Flag to indicate if the API is to enforce GSM7 encoding when sending the messages. If this flag is set to on (true), the API will drop any characters specified within the message text (if any) in order to dispatch the message in GSM7 encoding.
callbackURL	String (URL)		An HTTP URL denoting the callback URL to which the API will subsequently send a delivery report (using HTTP POST) for each destination number specified. The API user is to ensure that the corresponding web server returns a 200 OK response when this URL is invoked. The schema/format of the data to be sent in this report is described in the Output JSON Object Schema for CallbackURL Delivery Reports section below. If this parameter is not specified, the API will not send delivery reports for messages.
token	String		A user-defined value specified by the method caller that is echoed back without any modification by the API.

Output JSON Object Schema

```
{
  "result": {
    "resCode": "0",
    "message": "",
    "exInfo": "Method Specific Extra Information (default none)"
  },
  "jobID": "999",
  "token": "passed-in user-defined token",
  "invalidNumbers": "987654321,321654987,..."
}
```

Name	Type	Description/Notes
result	Object	Standard composite result object containing three sub-fields providing method call response status/information.
- resCode	Integer	A numeric value denoting the overall result status of the method call; 0 (zero) denotes SUCCESS, negative values denote an error condition as specified/listed in the Error Codes section above.
- message	String	A description string (null/blank in case of SUCCESS) providing information vis-a-vis the actual error condition encountered.
- exInfo	String	A more verbose informative string providing detailed information (if applicable) about the error condition encountered. The API will only provide such extra information where it deems that such information is necessary/useful.
jobID	Integer	An integer value containing the jobID of the submitted Job.
token	String	The user-defined value submitted as the <i>token</i> parameter on method invocation, echoed back to the method caller.
invalidNumbers	List of Strings (comma separated)	A list of destination numbers which the API has identified as incorrect/invalid numbers and hence cannot dispatch the SMS message to them.

Output JSON Object Schema for CallbackURL Delivery Reports

For each destination number, the API will invoke (using the HTTP POST method) the input *callbackURL* specified, passing in a URL encoded JSON object as specified below:

```
{
  "messageID": "12345678901234560202",
  "jobID": 1,
  "destinationNumber": "35678941574",
  "deliveryStatus": delivery-status
}
```


The aforementioned JSON object will be sent to the callback URL as the body element of the POST HTTP request.

The range of values (and explanatory code) for *deliveryStatus* are as below:

- 0 – DLRNOTRECEIVED
- 1 – ACCEPTD
- 2 – DELIVRD
- 3 – EXPIRED
- 4 – DELETED
- 5 – UNDELIV
- 7 – UNKNOWN
- 8 – REJECTED
- 9 – BADDLR

Note: The API will send the Integer code value for the above statuses and not the corresponding text for each code as listed above.

2 /Messages

Description

This method is used to retrieve Message information for a specific Job.

URL

```
https://api.messito.com/rest/Messages
```

HTTP Verb: GET

Input Parameters

These parameters – which are to be URL encoded prior to submission – are to be passed in standard HTTP request fashion, i.e. via URL parameters. Example:

```
https://api.messito.com/rest/Messages?
username=username&password=password&apikey=64865c96438f47fab8bcf92013fee8a2&jobID=18656&paginationOf
fset=-
1&paginationLimit=6000&attributeMaskFilter=jobID,submissionTimeStamp,messageID,source,destination
```

Name	Type	Req	Description/Notes
username	String	✓	The user-name of the specific user invoking the method.
password	String	✓	The authentication password of the specified user. <i>Note:</i> This parameter needs to be passed-in as an md5sum hash value.
apikey	String	✓	The application key issued to a Messito user upon registration to the system. <i>Note:</i> Refer to the API Key Parameter section above.
jobID	Integer	✓	The jobID whose messages – including their attributes – are to be retrieved. <i>Note:</i> In cases where the input <i>jobID</i> specified does not exist, the API will still return a successful return code (i.e. <i>result.resCode == 0</i>) as the operation is still deemed as successful, but the result <i>messageData</i> array will contain

		zero (0) entries.
paginationOffset	Integer	In context of this method, the API offers pagination functionality. This is done as the number of messages for a single job can be substantial and as such, this facility allows for the retrieval of messages in more manageable blocks. This parameter specifies the starting offset (within a job's overall list of messages) from which the returned set of messages is to start. If no value or a negative one is specified here, the API defaults the starting offset to zero (0).
paginationLimit	Integer	<p>The number of messages to retrieve in a single block of messages (i.e. page size), starting from the offset specified by the <i>paginationOffset</i> parameter.</p> <p>If no value or a negative one is specified here, the API will default this value to be <i>paginationOffset</i> plus 1000, effectively rendering the resultant page size to 1000.</p> <p><i>Note:</i> If the number of the Job's messages is less than the page-size specified or the calculated page-size is greater than the number of remaining messages for a job (which can be the case in cases when the <i>paginationOffset</i> + <i>paginationLimit</i> value exceeds the number of remaining messages), the API will return the number of messages that is less than the page size specified.</p>
attributeMaskFilter	List of Strings (comma separated)	A list of comma-separated Message attributes denoting which attributes are to be returned for each Message. The valid values for these names are the attribute names of a <i>messageData</i> JSON object as specified in the Output JSON Object Schema below. If one – or more – of these are specified, the <i>messageData</i> JSON object returned for each message will only contain these named/specified attributes; if no value is specified for this input parameter, then all attributes are returned for each message.

Output JSON Object Schema

```

{
  "result": {
    "resCode": "0",
    "message": "",
    "exInfo": "Method Specific Extra Information (default none)"
  },
  "messageData": [
    {
      "jobID": "job-id",
      "submissionTimeStamp": submission-timestamp,
      "sentTimeStamp": sent-timestamp,
      "deliveryTimeStamp": delivery-timestamp,
      "messageID": message-id,
      "source": "message-source",
      "destination": "destination-number",
      "messageBody": "message-body",
      "mcc": "mobile-country-code",
      "dataCoding": "data-coding",
      "creditsCost": "credits-cost",
      "messagePartCount": "message-part-count",
      "status": "status",
      "deliveryStatus": delivery-status
    }
  ]
}

```

Name	Type	Description/Notes
result	Object	Standard composite result object containing three sub-fields providing method call response status/information.
- resCode	Integer	A numeric value denoting the overall result status of the method call; 0 (zero) denotes SUCCESS, negative values denote an error condition as specified/listed in the Error Codes section above.
- message	String	A description string (null/blank in case of SUCCESS) providing information vis-

		a-vis the actual error condition encountered.
- exInfo	String	A more verbose informative string providing detailed information (if applicable) about the error condition encountered. The API will only provide such extra information where it deems that such information is necessary/useful.
messageData	Array of MessageData objects	An array of MessageData objects each providing the attributes listed below.
- jobId	Integer	The JobID identifying the job for which the message information is being retrieved.
- submissionTimeStamp	TimeStamp	The date/time (Unix Time) of when the Message was received for delivery.
- sentTimeStamp	TimeStamp	The date/time (Unix Time) of when the message was actually submitted for delivery.
- deliveryTimeStamp	TimeStamp	The date/time (Unix Time) of when the message was actually delivered to its destination.
- messageID	Integer	The unique message identifier of the Message.
- source	String	The sender ID (sender source label) of the message.
- destination	String	A string of digits containing the phone number to which the message was delivered.
- messageBody	String	The message body/text of the message.
- mcc	String	The Mobile Country Code (MCC) for this message.
- dataCoding	String	The encoding scheme (e.g. GMS7, UCS2, etc.) used to send the message.
- creditsCost	Float	The cost in credits charged for the delivery of the Message.
- messagePartCount	Integer	The number of actual SMS messages that were required to deliver the full message body/text of the Message.

- status	<p>Integer (Enum):</p> <ul style="list-style-type: none"> 1 – NOTFETCHED 2 – SENT 3 – NOCREDIT 4 – DESTINATIONINVALID 5 – DELETED 6 – UPDATING 7 – NOPRICE 8 – SENTACK 9 – OVERRIDDEN 	<p>The status identifying the current state of message dispatch.</p> <p><i>Note:</i> The API will return the integer code (and not the corresponding code text) within this <i>status</i> attribute.</p>
- deliveryStatus	<p>Integer (Enum):</p> <ul style="list-style-type: none"> 0 – DLRRNOTRECEIVED 1 – ACCEPTD 2 – DELIVRD 3 – EXPIRED 4 – DELETED 5 – UNDELIV 7 – UNKNOWN 8 – REJECTED 9 – BADDLR 	<p>The delivery status of the message (e.g. delivered, rejected, expired, etc.).</p> <p><i>Note:</i> The API will return the integer code (and not the corresponding code text) within this <i>deliveryStatus</i> attribute.</p>

3 /Jobs

Description

This method is used to retrieve a list of Jobs (and their corresponding information) which had been previously submitted.

URL

```
https://api.messito.com/rest/Jobs
```

HTTP Verb: GET

Input Parameters

These parameters – which are to be URL encoded prior to submission – are to be passed in standard HTTP request fashion, i.e. via URL parameters. Example:

```
https://api.messito.com/rest/Jobs?
username=username&password=password&apikey=64865c96438f47fab8bcf92013fee8a2&jobIDList=18656,18638,18637&startTimeStamp=1403614904&endTimeStamp=1403614904&paginationOffset=0&paginationLimit=10&userIDList=1,2,3
```

Name	Type	Req	Description/Notes
username	String	✓	The user-name of the specific user invoking the method.
password	String	✓	The authentication password of the specified user. <i>Note:</i> This parameter needs to be passed-in as an md5sum hash value.
apikey	String	✓	The application key issued to a Messito user upon registration to the system. <i>Note:</i> Refer to the API Key Parameter section above.
jobIDList	List of Strings (comma-separated)	✓	A list of jobID's identifying the jobs for which message information is to be returned. <i>Note:</i> This can be used in conjunction with the date-range filter specified via <i>StartTimeStamp-EndTimeStamp</i> . The API will guarantee that if a specific job satisfies both filters, only one

		jobID is returned in the result set.
startTimeStamp	Timestamp	<p>The start date/time (Unix Time) value of the period during which jobs were submitted. If not specified, then no filtering will be applied vis-a-vis job submission timestamps; in such cases, if the subsequent <i>endTimeStamp</i> is still specified, that parameter will also be ignored and no corresponding filtering is applied. In other words, the API will only apply filtering when <i>both</i> these parameters are specified.</p> <p><i>Note:</i> This can be used in conjunction with the date-range filter specified via <i>StartTimeStamp-EndTimeStamp</i>. The API will guarantee that if a specific job satisfies both filters, only one JobID is returned in the result set.</p>
endTimeStamp	Timestamp	<p>The end date/time (Unix Time) value of the period during which Jobs were submitted. If not specified, then no filtering will be applied vis-a-vis job submission timestamps; in such cases, if the previous <i>startTimeStamp</i> is still specified, that parameter will also be ignored and no corresponding filtering is applied. In other words, the API will only apply filtering when <i>both</i> these parameters are specified.</p> <p><i>Note:</i> This can be used in conjunction with the date-range filter specified via <i>StartTimeStamp-EndTimeStamp</i>. The API will guarantee that if a specific job satisfies both filters, only one jobID is returned in the result set.</p>
paginationOffset	Integer	<p>In context of this method, the API offers pagination functionality. This is done as the number of message for a single job can be substantial and as such, this facility allows for the retrieval of messages in more manageable lots. This parameter specifies the starting offset (within a Job's overall list of messages) from which the returned set of messages is to start. If no value or a negative one is specified here, the API defaults to the starting offset to zero (0).</p>
paginationLimit	Integer	<p>The number of messages to retrieve in a single lot of messages (i.e. page size).</p>

		<p>If no value or a negative one is specified here, the API will default this value to be <i>paginationOffset</i> plus 1000, effectively rendering the resultant page size to 1000.</p> <p><i>Note:</i> If the number of the Job's messages is less than the page-size specified or the calculated page-size is greater than the number of remaining messages for a job (which can be the case in cases when the <i>paginationOffset</i> + <i>paginationLimit</i> value exceeds the number of remaining messages), the API will return the number of messages that is less than the page size specified.</p>
userIDList	List of Integers (comma-separated)	The list of userID's to filter on for the jobs to be returned; i.e. if specified, only jobs submitted by the specified users will be returned. If this parameter is not specified, then no such filtering is effected and all qualifying jobs – irrespective of the user who submitted the job – will be returned.

Output JSON Object Schema

```
{
  "result": {
    "resCode": "0",
    "message": "",
    "exInfo": "Method Specific Extra Information (default none)"
  },
  "jobData": [
    {
      "jobID": "job-id",
      "name": "job-name",
      "source": "source",
      "messageBody": "message-body",
      "submissionTimeStamp": "submission-time-stamp",
      "scheduledTimeStamp": "scheduled-time-stamp",
      "processedTimeStamp": "processed-time-stamp",
      "token": "user-defined job token value",
      "userID": "user-id",
      "jobStatus": "job-status"
    }
  ]
}
```

```

    }
  ]
}

```

Name	Type	Description/Notes
result	Object	Standard composite result object containing three sub-fields providing method call response status/information.
- resCode	Integer	A numeric value denoting the overall result status of the method call; 0 (zero) denotes SUCCESS, negative values denote an error condition as specified/listed in the Error Codes section above.
- message	String	A description string (null/blank in case of SUCCESS) providing information vis-a-vis the actual error condition encountered.
- exInfo	String	A more verbose informative string providing detailed information (if applicable) about the error condition encountered. The API will only provide such extra information where it deems that such information is necessary/useful.
jobData	Array of JobData objects	An array of JobData objects each providing the attributes listed below.
- jobID	Integer	The jobID identifying the corresponding Job.
- name	String	The job name.
- source	String	The sender ID (sender source label) of each message within the job.
- messageBody	String	The message body/text of each Message within the job.
- submissionTimeStamp	TimeStamp	The date/time (Unix Time) of when the job was submitted for processing.
- scheduledTimeStamp	TimeStamp	The date/time (Unix Time) of when the job was scheduled for processing.

- processedTimeStamp	TimeStamp	The date/time (Unix Time) of when the job was actually processed.
- token	String	The user-defined token that was specified when the corresponding job was submitted.
- userID	Integer	The ID of the user that submitted this job.
- jobStatus	Integer (enum): 0 – PREFETCH 1 – NOTFETCHED 2 – PROCESSING 3 – SUCCESS 4 – MESSAGESNOTSENT 5 – DELETED 6 – UPDATING 7 – RESEND	The status denoting the current stage of Job processing. <i>Note:</i> The API will return the integer code (and not the corresponding code text) within this <i>jobStatus</i> attribute.

4 /Users

Description

This method is used to retrieve information relating to the users within the API user's Messito account, where the username and userID of each user within the Account is returned.

URL

```
https://api.messito.com/rest/Users
```

HTTP Verb: GET

Input Parameters

These parameters – which are to be URL encoded prior to submission – are to be passed in standard HTTP request fashion, i.e. via URL parameters. Example:

```
https://api.messito.com/rest/Users?
username=username&password=password&apikey=64865c96438f47fab8bcf92013fee8a2
```

Name	Type	Req	Description/Notes
username	String	✓	The user-name of the specific user invoking the method.
password	String	✓	The authentication password of the specified user. <i>Note:</i> This parameter needs to be passed-in as an md5sum hash value.
apikey	String	✓	The application key issued to a Messito API user upon registration to the system. <i>Note:</i> Refer to the API Key Parameter section above.

Output JSON Object Schema

```
{
  "result": {
    "resCode": "0",
    "message": ""
  }
}
```

```

    "exInfo": "Method Specific Extra Information (default none)"
  },
  "userData": [
    {
      "userID": "user-id",
      "username": "user-name"
    }
  ]
}

```

Name	Type	Description/Notes
result	Object	Standard composite result object containing three sub-fields providing method call response status/information.
- resCode	Integer	A numeric value denoting the overall result status of the method call; 0 (zero) denotes SUCCESS, negative values denote an error condition as specified/listed in the Error Codes section above.
- message	String	A description string (null/blank in case of SUCCESS) providing information vis-a-vis the actual error condition encountered.
- exInfo	String	A more verbose informative string providing detailed information (if applicable) about the error condition encountered. The API will only provide such extra information where it deems that such information is necessary/useful.
userData	Array of userData objects	An array of userData objects each providing the attributes listed below.
- userID	Integer	The user's userID.
- username	String	The user's username.

5 /Credits

Description

This method is used to retrieve the current Credit Balance for the corresponding Account as identified by the *username* parameter.

URL

```
https://api.messito.com/rest/Credits
```

HTTP Verb: GET

Input Parameters

These parameters – which are to be URL encoded prior to submission – are to be passed in standard HTTP request fashion, i.e. via URL parameters. Example:

```
https://api.messito.com/rest/Credits?
username=username&password=password&apikey=64865c96438f47fab8bcf92013fee8a2
```

Name	Type	Req	Description/Notes
username	String	✓	The user-name of the specific user invoking the method.
password	String	✓	The authentication password of the specified user. <i>Note:</i> This parameter needs to be passed-in as an md5sum hash value.
apikey	String	✓	The application key issued to a Messito API user upon registration to the system. <i>Note:</i> Refer to the API Key Parameter section above.

Output JSON Object Schema

```
{
  "result": {
    "exInfo": "",
    "message": ""
  }
}
```

```

    "resCode": "0"
  },
  "credits": "16095.2"
}

```

Name	Type	Description/Notes
result	Object	Standard composite result object containing three sub-fields providing method call response status/information.
- resCode	Integer	A numeric value denoting the overall result status of the method call; 0 (zero) denotes SUCCESS, negative values denote an error condition as specified/listed in the Error Codes section above.
- message	String	A description string (null/blank in case of SUCCESS) providing information vis-a-vis the actual error condition encountered.
- exInfo	String	A more verbose informative string providing detailed information (if applicable) about the error condition encountered. The API will only provide such extra information where it deems that such information is necessary/useful.
credits	Number	A decimal number denoting the current Credit Balance for the corresponding Account.

6 Appendix – Code Snippets

PHP

Note: This code uses the PHP::cURL library (<http://php.net/manual/en/book.curl.php>); other methods and modules can be used to achieve the same functionality.

```
<?php
//global variables
$baseURL = "https://api.messito.com/rest";
$username = "test-user";
$password = "a60e197eb5024188a2572ccdfa250dd8";
$apikey = "ae99544138c84e35ad0c0b67d96548e7";

function submitJob($baseURL, $username, $password, $apikey)
{
    $data = array(
        "username" => $username,
        "password" => $password,
        "apikey" => $apikey,
        "name" => "Test Job",
        "messageText" => "Test Message Text",
        "destinationNumbers" => "35612345678,35698765432",
        "source" => "MessitoAPISource",
        "forceGSM7Text" => "false",
        "callbackURL" => "http://echo.jsontest.com/",
        "token" => uniqid()
    );
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($curl, CURLOPT_URL, $baseURL . "/Job");
    curl_setopt($curl, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/x-www-form-urlencoded'
    ));
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "PUT");
    curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($data));
    $f = curl_exec($curl);
    $res = json_decode($f, true);
    curl_close($curl);
    return $res;
}
```



```
function getMessages($baseUrl, $username, $password, $apikey, $jobID)
{
    $targetURL = sprintf("%s/Messages?username=%s" . "&password=%s" . "&apikey=%s" . "&jobID=%d" .
"&paginationOffset=%d" . "&paginationLimit=%d", $baseUrl, urlencode($username),
urlencode($password), urlencode($apikey), urlencode($jobID), urlencode(0), urlencode(500));
    $curl      = curl_init();
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($curl, CURLOPT_URL, $targetURL);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    $f         = curl_exec($curl);
    $res = json_decode($f, true);
    curl_close($curl);
    return $res;
}

function getUsers($baseUrl, $username, $password, $apikey)
{
    $targetURL = sprintf("%s/Users?username=%s" . "&password=%s" . "&apikey=%s", $baseUrl,
urlencode($username), urlencode($password), urlencode($apikey));
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($curl, CURLOPT_URL, $targetURL);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    $f = curl_exec($curl);
    $res = json_decode($f, true);
    curl_close($curl);
    return $res;
}

function getJobs($baseUrl, $username, $password, $apikey, $jobIDs)
{
    $targetURL = sprintf("%s/Jobs?username=%s" . "&password=%s" . "&apikey=%s" . "&jobIDList=%s" .
"&startTimeStamp=%d" . "&endTimeStamp=%d", $baseUrl, urlencode($username), urlencode($password),
urlencode($apikey), urlencode($jobIDs), urlencode(1401580800), //1st June 2014 00:00:00
urlencode(1406764800) //31st July 2014 00:00:00
    );
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($curl, CURLOPT_URL, $targetURL);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    $f = curl_exec($curl);
```

```

$res = json_decode($f, true);
curl_close($curl);
return $res;
}

////////////////////////////////////

/* /Job */
$jobRes = submitJob($baseUrl, $username, $password, $apikey);
if ($jobRes == null) {
    //handle null response here
} else {
    $result = $jobRes["result"];
    switch ($result["resCode"]) {
        case 0:
            //handle success
            printf("jobID: %s; token: %s\r\n", $jobRes["jobID"], $jobRes["token"]);
            break;
        default:
            //handle specific errors as per API document
            break;
    }
}

/* /Messages */
$messagesRes = getMessages($baseUrl, $username, $password, $apikey, 20166);
if ($messagesRes == null) {
    //handle null response here
} else {
    $result = $messagesRes["result"];
    $messageData = $messagesRes["messageData"];
    switch ($result["resCode"]) {
        case 0:
            //handle success
            foreach ($messageData as $message) {
                //loop through messageData
                printf("creditsCost: %d; dataCoding: %s; deliveryStatus: %d; " . "deliveryTimeStamp: %d; destination: %s; jobID: %d; " . "mcc: %s; messageBody: %s; messageID: %d; messagePartCount: %d; " . "sentTimeStamp: %d; source: %s; status: %d; submissionTimeStamp: %d", $message["creditsCost"], $message["dataCoding"], $message["deliveryStatus"], $message["deliveryTimeStamp"], $message["destination"], $message["jobID"], $message["mcc"], $message["messageBody"], $message["messageID"], $message["messagePartCount"], $message["sentTimeStamp"], $message["source"], $message["status"], $message["submissionTimeStamp"]);
            }
        }
    }
}

```

```

        break;
    default:
        //handle specific errors as per API document
        break;
    }
}

/* /Jobs */
$jobsRes = getJobs($baseUrl, $username, $password, $apikey, "20164,20165,20181,20180");
if ($jobsRes == null) {
    //handle null response here
} else {
    $result = $jobsRes["result"];
    switch ($result["resCode"]) {
        case 0:
            //handle success
            $jobData = $jobsRes["jobData"];
            foreach ($jobData as $job) {
                printf("jobID: %d; jobStatus: %d; messageBody: %s; name: %s; " .
                    "processedTimeStamp: %d; scheduledTimeStamp: %d; " . "source: %d; submissionTimeStamp: %d; token:
                    %s; userID: %d\r\n", $job["jobID"], $job["jobStatus"], $job["messageBody"], $job["name"],
                    $job["processedTimeStamp"], $job["scheduledTimeStamp"], $job["source"], $job["submissionTimeStamp"],
                    $job["token"], $job["userID"]);
            }
            break;
        default:
            //handle specific errors as per API document
            break;
    }
}

/* /Users */
$usersRes = getUsers($baseUrl, $username, $password, $apikey);
if ($usersRes == null) {
    //handle null response here
} else {
    $result = $usersRes["result"];
    $userData = $usersRes["userData"];
    switch ($result["resCode"]) {
        case 0:
            //handle success
            foreach ($userData as $user) {

```

```
        //loop through userData
        printf("userID: %d; username: %s\r\n", $user["userID"], $user["username"]);
    }
    break;
default:
    //handle specific errors as per API document
    break;
}
}
?>
```

Java

```
package se.ft.MMMPublicAPI.SampleAPICalls;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
import javax.net.ssl.HttpURLConnection;

public class HTTPSClient {

    private static final String PUTJOB_COMMAND = "--putJob";
    private static final String GETJOBS_COMMAND = "--getJobs";
    private static final String GETMESSAGES_COMMAND = "--getMessages";
    private static final String GETUSERS_COMMAND = "--getUsers";
    private static final String ALL_COMMAND = "--all";

    private static final String USERNAME = "test-user";
    private static final String PASSWORD = "a60e197eb5024188a2572ccdfa250dd8";
    private static final String API_KEY = "ae99544138c84e35ad0c0b67d96548e7";

    private static final String UTF_8 = "UTF-8";

    private static final String PUT_JOB_ENDPOINT = "https://api.messito.com/rest/Job";
    private static final String GET_JOBS_ENDPOINT = "https://api.messito.com/rest/Jobs";
    private static final String GET_MESSAGES_ENDPOINT = "https://api.messito.com/rest/Messages";
    private static final String GET_USERS_ENDPOINT = "https://api.messito.com/rest/Users";

    private static HttpURLConnection createHttpsURLConnection(final URL url)
        throws IOException {
        return (HttpURLConnection) url.openConnection();
    }

    public static void main(String[] args) throws IOException {
```

```
if (args.length > 0) {
    final String command = args[0].trim();

    switch (command) {
        case PUTJOB_COMMAND:
            System.out.println("EXECUTING - putJob()");
            putJob();
            break;
        case GETJOBS_COMMAND:
            System.out.println("EXECUTING - getJobs()");
            getJobs();
            break;
        case GETMESSAGES_COMMAND:
            if (args.length > 1) {
                final String jobid = args[1];
                System.out.format("EXECUTING - getMessages(%s)\n", jobid);
                getMessages(jobid);
            } else {
                System.out.format("usage: - %s [jobid]\n", GETMESSAGES_COMMAND);
            }
            break;
        case GETUSERS_COMMAND:
            System.out.println("EXECUTING - getUsers()");
            getUsers();
            break;
        case ALL_COMMAND:
            System.out.println("EXECUTING - putJob()");
            putJob();
            System.out.println("EXECUTING - getJobs()");
            getJobs();
            if (args.length > 1) {
                final String jobid = args[1];
                System.out.format("EXECUTING - getMessages(%s)\n", jobid);
                getMessages(jobid);
            } else {
                System.out.format("skipping getMessages, usage: - %s [jobid]\n", ALL_COMMAND);
            }
            System.out.println("EXECUTING - getUsers()");
            getUsers();
            break;
        default:
```

```
        outputUsageCommand();
    }
} else {
    outputUsageCommand();
}
}

private static void outputUsageCommand() {
    System.out.format("usage: [%s, %s, %s, %s, %s]\n",
        PUTJOB_COMMAND,
        GETJOBS_COMMAND,
        GETMESSAGES_COMMAND,
        GETUSERS_COMMAND,
        ALL_COMMAND);
}

private static void putJob() throws MalformedURLException, IOException {

    HttpURLConnection httpsConn = createHttpsURLConnection(new URL(PUT_JOB_ENDPOINT));

    httpsConn.setRequestMethod("PUT");
    httpsConn.setDoInput(true);
    httpsConn.setDoOutput(true);

    OutputStream os = httpsConn.getOutputStream();
    BufferedWriter writer = new BufferedWriter(
        new OutputStreamWriter(os, UTF_8));

    String data = String.format(
        "username=%s&"
        + "password=%s&"
        + "apikey=%s&"
        + "name=%s&"
        + "messageText=%s&"
        + "destinationNumbers=%s&"
        + "source=%s&"
        + "callbackURL=%s&"
        + "token=%s",
        USERNAME,
        PASSWORD,
        API_KEY,
```

```
        "name",
        "message-text",
        "35612345678,35698765432",
        "source",
        "http://echo.jsontest.com/",
        "token");

writer.write(data);
writer.flush();
writer.close();
os.close();

BufferedReader br = new BufferedReader(
    new InputStreamReader(httpsConn.getInputStream()));

String response;
while ((response = br.readLine()) != null) {
    System.out.println(response);
}

}

private static void getJobs() throws MalformedURLException, IOException {

String url = String.format("%s?"
    + "username=%s&"
    + "password=%s&"
    + "apikey=%s&"
    + "jobIDList=%s&"
    + "startTimeStamp=%s&"
    + "endTimeStamp=%s&"
    + "paginationOffset=%s&"
    + "paginationLimit=%s&"
    + "userIDList=%s",
    GET_JOBS_ENDPOINT,
    URLEncoder.encode(USERNAME, UTF_8),
    URLEncoder.encode(PASSWORD, UTF_8),
    URLEncoder.encode(API_KEY, UTF_8),
    URLEncoder.encode("", UTF_8),
    URLEncoder.encode("", UTF_8),
    URLEncoder.encode("", UTF_8),
```



```
        URLEncoder.encode("0", UTF_8),
        URLEncoder.encode("-1", UTF_8),
        URLEncoder.encode("", UTF_8));

    HttpURLConnection httpsConn = createHttpsURLConnection(new URL(ur1));
    BufferedReader br = new BufferedReader(
        new InputStreamReader(httpsConn.getInputStream()));

    String response;
    while ((response = br.readLine()) != null) {
        System.out.println(response);
    }
}

private static void getMessages(final String jobID)
    throws MalformedURLException, IOException {

    String url = String.format("%s?"
        + "username=%s&"
        + "password=%s&"
        + "apikey=%s&"
        + "jobID=%s&"
        + "paginationOffset=%s&"
        + "paginationLimit=%s",
        GET_MESSAGES_ENDPOINT,
        URLEncoder.encode(USERNAME, UTF_8),
        URLEncoder.encode(PASSWORD, UTF_8),
        URLEncoder.encode(API_KEY, UTF_8),
        URLEncoder.encode(jobID, UTF_8),
        URLEncoder.encode("0", UTF_8),
        URLEncoder.encode("-1", UTF_8));

    HttpURLConnection httpsConn = createHttpsURLConnection(new URL(ur1));
    BufferedReader br = new BufferedReader(
        new InputStreamReader(httpsConn.getInputStream()));

    String response;
    while ((response = br.readLine()) != null) {
        System.out.println(response);
    }
}
```

```
}

private static void getUsers() throws MalformedURLException, IOException {

    String url = String.format("%s?"
        + "username=%s&"
        + "password=%s&"
        + "apikey=%s",
        GET_USERS_ENDPOINT,
        URLEncoder.encode(USERNAME, UTF_8),
        URLEncoder.encode(PASSWORD, UTF_8),
        URLEncoder.encode(API_KEY, UTF_8));

    HttpURLConnection httpsConn = createHttpsURLConnection(new URL(url));
    BufferedReader br = new BufferedReader(
        new InputStreamReader(httpsConn.getInputStream()));

    String response;
    while ((response = br.readLine()) != null) {
        System.out.println(response);
    }
}

}
```

C#

```
public void submitJob()
{
    String ip = "api.messito.com";
    int port = 443;
    String username = "test-user";
    String passwordMD5 = "a60e197eb5024188a2572ccdfa250dd8";
    String apikey = "ae99544138c84e35ad0c0b67d96548e7";

    String name = "testjob";
    String messageText = "testmessage";
    String destinationNumbers = "35698765432,35612345678";
    String source = "testsource";
    String callbackURL = "http://echo.jsontest.com/";
    String token = "testToken";
    String forceGSM7Text = "false";

    HttpWebRequest request = (HttpWebRequest) HttpWebRequest.Create("https://" + ip + ":" +
port + "/rest/Job");
    string putData = "username=" + username +
        "&password=" + passwordMD5 +
        "&apikey=" + apikey +
        "&name=" + name +
        "&messageText=" + messageText +
        "&destinationNumbers=" + destinationNumbers +
        "&source=" + source +
        "&callbackURL=" + callbackURL +
        "&token=" + token +
        "&forceGSM7Text=" + forceGSM7Text;
    byte[] byteArray = Encoding.UTF8.GetBytes(putData);
    request.Method = "PUT";
    // Set the ContentType property of the WebRequest.
    request.ContentType = "application/x-www-form-urlencoded";
    // Set the ContentLength property of the WebRequest.
    request.ContentLength = byteArray.Length;
    // Get the request stream.
    Stream dataStream = request.GetRequestStream();
    // Write the data to the request stream.
    dataStream.Write(byteArray, 0, byteArray.Length);
    // Close the Stream object.
```

```
        dataStream.Close();

        HttpResponseMessage response = (HttpResponseMessage) request.GetResponse();
        // Get the stream associated with the response.
        Stream receiveStream = response.GetResponseStream();
        // Pipes the stream to a higher level stream reader with the required encoding format.
        StreamReader readStream = new StreamReader(receiveStream, Encoding.UTF8);
        string resp = readStream.ReadToEnd();
        response.Close();
        readStream.Close();

        MessageBox.Show(resp, "title");
    }

    public void getMessages()
    {
        String ip = "api.messito.com";
        int port = 443;
        String username = "test-user";
        String passwordMD5 = "a60e197eb5024188a2572ccdfa250dd8";
        String apikey = "ae99544138c84e35ad0c0b67d96548e7";

        int jobID = 18656;
        int paginationOffset = 0;
        int paginationLimit = 5000;

        HttpRequest request = (HttpRequest) HttpRequest.Create("https://" + ip + ":" +
port + "/rest/Messages?username=" + username + "&password=" + passwordMD5 + "&apikey=" + apikey +
"&jobID=" + jobID + "&paginationOffset=" + paginationOffset + "&paginationLimit=" +
paginationLimit);
        HttpResponseMessage response = (HttpResponseMessage) request.GetResponse();
        // Get the stream associated with the response.
        Stream receiveStream = response.GetResponseStream();
        // Pipes the stream to a higher level stream reader with the required encoding format.
        StreamReader readStream = new StreamReader(receiveStream, Encoding.UTF8);
        string resp = readStream.ReadToEnd();
        response.Close();
        readStream.Close();

        MessageBox.Show(resp, "title");
    }
}
```

```
public void getJobs()
{
    String ip = "api.messito.com";
    int port = 443;
    String username = "test-user";
    String passwordMD5 = "a60e197eb5024188a2572ccdfa250dd8";
    String apikey = "ae99544138c84e35ad0c0b67d96548e7";

    String jobIDList = "18656,18638,18637";
    String userIDList = "3,4,5";
    long startTimestampSeconds = 0;
    long endTimestampSeconds = 1403614904;
    int paginationOffset = 0;
    int paginationLimit = 5000;

    HttpRequest request = (HttpRequest) HttpRequest.Create("https://" + ip + ":" +
port + "/rest/Jobs?username=" + username +
"&password=" + passwordMD5 + "&apikey=" + apikey + "&jobIDList=" + jobIDList + "&userIDList=" +
userIDList + "&startTimestamp=" + startTimestampSeconds + "&endTimestamp=" + endTimestampSeconds +
"&paginationOffset=" + paginationOffset + "&paginationLimit=" + paginationLimit);

    HttpResponse response = (HttpResponse)request.GetResponse();
    // Get the stream associated with the response.
    Stream receiveStream = response.GetResponseStream();
    //Pipes the stream to a higher level stream reader with
// the required encoding format.
    StreamReader readStream = new StreamReader(receiveStream, Encoding.UTF8);
    string resp = readStream.ReadToEnd();
    response.Close();
    readStream.Close();

    MessageBox.Show(resp, "title");
}

public void getUsers()
{
    String ip = "api.messito.com";
    int port = 443;
    String username = "test-user";
    String passwordMD5 = "a60e197eb5024188a2572ccdfa250dd8";
    String apikey = "ae99544138c84e35ad0c0b67d96548e7";

    HttpRequest request = (HttpRequest) HttpRequest.Create("https://" + ip + ":" +
port + "/rest/Users?username=" + username + "&password=" + passwordMD5 + "&apikey=" + apikey);
```

```
HttpWebResponse response = (HttpWebResponse) request.GetResponse();
// Get the stream associated with the response.
Stream receiveStream = response.GetResponseStream ();
// Pipes the stream to a higher level stream reader
// with the required encoding format.
StreamReader readStream = new StreamReader (receiveStream, Encoding.UTF8);
string resp = readStream.ReadToEnd();
response.Close();
readStream.Close();

MessageBox.Show(resp, "title");
}
```