Zoom Connect REST API

Authentication

Requests to Zoom Connect are authenticated by providing your account email address and API key as request header attributes named *email* and *token* respectively. You may also provide the email and token as URL parameters, although this is not advised and should be used for testing purposes only.

Note, your API key is securely encrypted by the SSL channel. If you have not been provided an API key, please contact Zoom Connect support.

Retrieving your account balance

URL: https://zoomconnect.com/zoom/api/rest/v1/account/balance
HTTP METHOD: GET

Example XML response

URL: https://zoomconnect.com/zoom/api/rest/v1/account/balance.xml

```
<account>
    <creditBalance>33</creditBalance>
</account>
```

Example JSON response

URL: https://zoomconnect.com/zoom/api/rest/v1/account/balance.json

```
{ "creditBalance":33 }
```

Sending an SMS

We provide three ways of connecting to our API to send a single SMS, namely: by means of URL Parameters, JSON objects, and XML data.

URL PARAMETERS

URL: https://zoomconnect.com/zoom/api/rest/v1/sms/send-url-parameters
HTTP METHOD: GET or POST

URL Parameter Example

https://zoomconnect.com/zoom/api/rest/v1/sms/send-url-parameters?token=yourPassword&email=you@example.com&recipientNumber=27725551234&message=testing&dataField=Campaign1

XML

URL: https://zoomconnect.com/zoom/api/rest/v1/sms/send.xml
HTTP METHOD: POST

XML Example

```
<sendSmsRequest>
    <message>This is an example of the data to post. The dateToSend parameter is optional.</message>
    <recipientNumber>27725551234</recipientNumber>
    <campaign>e.g Optional campaign name</campaign>
    <dataField>e.g Optional extra field</dataField>
</sendSmsRequest>
```

Example XML response

messageId e.g 77654

JSON

URL: https://zoomconnect.com/zoom/api/rest/v1/sms/send.json
HTTP METHOD: POST

JSON Example

```
{
        "recipientNumber":"27725551234",
        "message":"This is an example of the data to post. The dateToSend parameter is optional.",
        "campaign":"e.g Optional campaign name",
        "dataField":"e.g Optional extra field"
}
```

Example JSON response

```
{
        "messageId":"4635029",
        "error":null
}
```

Sending an SMS batch

We provide you the ability to send a batch of messages in one go. You are able to do this by means of JSON objects and XML data.

XML

URL: https://zoomconnect.com/zoom/api/rest/v1/sms/send-bulk.xml
HTTP METHOD: POST

XML Example

```xml
<sendSmsRequests>
    <sendSmsRequest>
        <message>Message 1.</message>
        <recipientNumber>27725551234</recipientNumber>
        <campaign>e.g Optional campaign name</campaign>
        <dataField>e.g Optional extra field</dataField>
    </sendSmsRequest>
    <sendSmsRequest>
        <message>Message 2.</message>
        <recipientNumber>27725553456</recipientNumber>
        <campaign>e.g Optional campaign name</campaign>
        <dataField>e.g Optional extra field</dataField>
    </sendSmsRequest>
</sendSmsRequests>
```

Example XML response

messageId e.g 77654
messageId e.g 77655

JSON

URL: https://zoomconnect.com/zoom/api/rest/v1/sms/send-bulk.json
HTTP METHOD: POST

JSON Example

```
{
        "sendSmsRequests":
        [
                {
                        "recipientNumber":"27725551234",
                        "message":"Message 1.",
                        "campaign":"e.g Optional campaign name",
                        "dataField":"e.g Optional extra field"
                },
                {
                        "recipientNumber":"27725553456",
                        "message":"Message 2.",
                        "campaign":"e.g Optional campaign name",
                        "dataField":"e.g Optional extra field"
                }
        ]
}
```

Example JSON response

```
{
        "sendSmsResponses":
        [
                {
                        "messageId":"4635873",
                        "error":null
                },
                {
                        "messageId":"4635874",
                        "error":null
                }
        ]
}
```

Scheduling an SMS

Scheduling an SMS through our API follows the same pattern as sending an SMS with the addition of adding the date and time of the message to be scheduled. The format of the date is yyyyMMddHHmm.

JSON Example

```
{
        "recipientNumber":"27725551234",
        "message":"This is an example of the data to post. The dateToSend parameter is optional.",
        "campaign":"e.g Optional campaign name",
        "dataField":"e.g Optional extra field",
        "dateToSend":"201404282149"
}
```

XML Example

```
<sendSmsRequest>
   <message>This is an example of the data to post. The dateToSend parameter is optional.</message>
   <recipientNumber>27725551234</recipientNumber>
   <campaign>e.g Optional campaign name</campaign>
   <dataField>e.g Optional extra field</dataField>
   <dateToSend>201404282149</dateToSend>
</sendSmsRequest>
```

Scheduling an SMS batch

We provide you the ability to schedule a batch of messages in one go. Scheduling an SMS through our API follows the same pattern as sending a batch SMS with the addition of adding the date and time of the message to be scheduled. The format of the date is yyyyMMddHHmm. You are able to do this by means of JSON objects, and XML data.

XML

URL: https://zoomconnect.com/zoom/api/rest/v1/sms/send-bulk.xml
HTTP METHOD: POST

XML Example

```
<sendSmsRequests>
    <sendSmsRequest>
        <dateToSend>201406060909</dateToSend>
        <message>Message 1.</message>
        <recipientNumber>27725551234</recipientNumber>
        <campaign>e.g Optional campaign name</campaign>
        <dataField>e.g Optional extra field</dataField>
    </sendSmsRequest>
    <sendSmsRequest>
        <dateToSend>201406060909</dateToSend>
        <message>Message 2.</message>
        <recipientNumber>27725553456</recipientNumber>
        <campaign>e.g Optional campaign name</campaign>
        <dataField>e.g Optional extra field</dataField>
    </sendSmsRequest>
</sendSmsRequests>
```

JSON

URL: https://zoomconnect.com/zoom/api/rest/v1/sms/send-bulk.json
HTTP METHOD: POST

JSON Example

```
{
        "sendSmsRequests":
        [
                {
                        "recipientNumber":"27725551234",
                        "message":"Message 1.",
                        "dateToSend":"201406060924",
                        "campaign":"e.g Optional campaign name",
                        "dataField":"e.g Optional extra field"
                },
                {
                        "recipientNumber":"27725553456",
                        "message":"Message 2.",
                        "dateToSend":"201406060924",
                        "campaign":"e.g Optional campaign name",
```

```
                    "dataField":"e.g Optional extra field"
                }
            ]
}
```

Receiving a reply

All replies received are forwarded to a specified URL. To set the message forward URL please login to Zoom Connect, click on your company name, click on "Developer settings" and paste the URL to which inbound replies are to be forwarded.

HTTP METHOD: POST

Example

```
http://yourserver.com/receive-sms.php?
        &from: 27725551234
        &messageId=124567
        &message: REPLY
        &date: 201404282055
        &charset: UTF-8
        &campaign: shoe campaign
        &dataField: custom
        &nonce: 89430dc5-cac1-4795-9dd0-46c9b8233e52
        &nonce-date: 20140428210027
        &checksum: 180f27d80dd05886e81da0c2ccebf830bc85c7ae
```

Please note that the "dataField" is only present when set

Receiving a delivery report

All delivery reports received are forwarded to a specified URL. To set the delivery report URL please login to Zoom Connect, click on your company name, click on "Developer settings" and paste the URL to which delivery reports are to be forwarded.

HTTP METHOD: POST

Example

```
http://yourserver.com/delivery-report.php?
        &from: 27725551234
        &messageId=124567
        &status: REPLY
        &campaign: optional campaign name
        &dataField: optional extra field
        &date: 201404282055
        &charset: UTF-8
        &nonce: 89430dc5-cac1-4795-9dd0-46c9b8233e52
        &nonce-date: 20140428210027
        &checksum: 180f27d80dd05886e81da0c2ccebf830bc85c7ae
```

Validating a received message

A shared secret key is used to generate an HMAC-SHA1 checksum based on the parameter data sent from Zoom Connect to your server. This checksum is sent along as a parameter with the other data. If a would be attacker alters any of the parameters, the checksum will no longer match, and you should reject the request.

An expiry date, also included in the checksum, is sent as a parameter with each request. This expiry date should be checked to ensure requests are not resent. The field name of the expiry date is nonce-date, and the date format of the is yyyyMMddHHmmss.

If you have not been provided a secret key, please login to Zoom Connect, click on your company name, click on "Developer settings" and click on the "Regenerate" button in order to generate your secret key.

Example

```
$privateKey = "secret";

$params['to'] = '27725551234';
$params['from'] = '27726661234';
$params['dataField'] = 'data';
$params['date'] = '201405161005';
$params['messageId']='124567'
$params['message'] = 'Test Message';
$params['charset'] = 'UTF-8';
$params['nonce'] = 'e273f1776275974f1a120d8b92c5b3cb';
$params['nonce-date'] = '2014051610050102';

ksort($params);
$message = http_build_query($params, null, '&');
$checksum = hash_hmac('sha1', $message, $privateKey);

charset=UTF-8&dataField=data&date=201405161005&from=27726661234&message=Test+Message&nonce=e273f17
76275974f1a120d8b92c5b3cb&nonce-date=2014051610050102&to=27725551234
e0f97132b829301592874356f76e47edc355688f
```