
Nexmo Documentation

Release 1.4.0

Tim Craft

Sep 18, 2016

1	Nexmo Client Library for Python	1
1.1	Installation	1
1.2	Usage	1
1.3	SMS API	2
1.4	Voice API	2
1.5	Verify API	3
1.6	Application API	4
1.7	Validate webhook signatures	4
1.8	JWT parameters	5
1.9	API Coverage	5
1.10	License	6
2	API Reference	7
3	Indices and tables	9
	Python Module Index	11

Nexmo Client Library for Python

This is the Python client library for Nexmo's API. To use it you'll need a Nexmo account. Sign up [for free](https://nexmo.com) at nexmo.com.

- *Installation*
- *Usage*
- *SMS API*
- *Voice API*
- *Verify API*
- *Application API*
- *Coverage*
- *License*

1.1 Installation

To install the Python client library using pip:

```
pip install nexmo
```

Alternatively you can clone the repository:

```
git clone git@github.com:Nexmo/nexmo-python.git
```

1.2 Usage

Begin by importing the nexmo module:

```
import nexmo
```

Then construct a client object with your key and secret:

```
client = nexmo.Client(key=api_key, secret=api_secret)
```

For production you can specify the NEXMO_API_KEY and NEXMO_API_SECRET environment variables instead of specifying the key and secret explicitly.

For newer endpoints that support JWT authentication such as the Voice API, you can also specify the application_id and private_key arguments:

```
client = nexmo.Client(application_id=application_id, private_key=private_key)
```

In order to check signatures for incoming webhook requests, you'll also need to specify the signature_secret argument (or the NEXMO_SIGNATURE_SECRET environment variable).

1.3 SMS API

1.3.1 Send a text message

```
response = client.send_message({'from': 'Python', 'to': 'YOUR-NUMBER', 'text': 'Hello ↵
↵world'})

response = response['messages'][0]

if response['status'] == '0':
    print 'Sent message', response['message-id']

    print 'Remaining balance is', response['remaining-balance']
else:
    print 'Error:', response['error-text']
```

Docs: <https://docs.nexmo.com/messaging/sms-api/api-reference#request>

1.4 Voice API

1.4.1 Make a call

```
response = client.create_call({
    'to': [{'type': 'phone', 'number': '14843331234'}],
    'from': {'type': 'phone', 'number': '14843335555'},
    'answer_url': ['https://example.com/answer']
})
```

Docs: https://docs.nexmo.com/voice/voice-api/api-reference#call_create

1.4.2 Retrieve a list of calls

```
response = client.get_calls()
```

Docs: https://docs.nexmo.com/voice/voice-api/api-reference#call_retrieve

1.4.3 Retrieve a single call

```
response = client.get_call(uuid)
```

Docs: https://docs.nexmo.com/voice/voice-api/api-reference#call_retrieve_single

1.4.4 Update a call

```
response = client.update_call(uuid, action='hangup')
```

Docs: https://docs.nexmo.com/voice/voice-api/api-reference#call_modify_single

1.5 Verify API

1.5.1 Start a verification

```
response = client.start_verification(number='441632960960', brand='MyApp')

if response['status'] == '0':
    print 'Started verification request_id=' + response['request_id']
else:
    print 'Error:', response['error_text']
```

Docs: <https://docs.nexmo.com/verify/api-reference/api-reference#vrequest>

The response contains a verification request id which you will need to store temporarily (in the session, database, url etc).

1.5.2 Check a verification

```
response = client.check_verification('00e6c3377e5348cdaf567e1417c707a5', code='1234')

if response['status'] == '0':
    print 'Verification complete, event_id=' + response['event_id']
else:
    print 'Error:', response['error_text']
```

Docs: <https://docs.nexmo.com/verify/api-reference/api-reference#check>

The verification request id comes from the call to the start_verification method. The PIN code is entered into your application by the user.

1.5.3 Cancel a verification

```
client.cancel_verification('00e6c3377e5348cdaf567e1417c707a5')
```

Docs: <https://docs.nexmo.com/verify/api-reference/api-reference#control>

1.5.4 Trigger next verification step

```
client.trigger_next_verification_event('00e6c3377e5348cdaf567e1417c707a5')
```

Docs: <https://docs.nexmo.com/verify/api-reference/api-reference#control>

1.6 Application API

1.6.1 Create an application

```
response = client.create_application(name='Example App', type='voice', answer_  
↪url=answer_url)
```

Docs: <https://docs.nexmo.com/tools/application-api/api-reference#create>

1.6.2 Retrieve a list of applications

```
response = client.get_applications()
```

Docs: <https://docs.nexmo.com/tools/application-api/api-reference#list>

1.6.3 Retrieve a single application

```
response = client.get_application(uuid)
```

Docs: <https://docs.nexmo.com/tools/application-api/api-reference#retrieve>

1.6.4 Update an application

```
response = client.update_application(uuid, answer_method='POST')
```

Docs: <https://docs.nexmo.com/tools/application-api/api-reference#update>

1.6.5 Delete an application

```
response = client.delete_application(uuid)
```

Docs: <https://docs.nexmo.com/tools/application-api/api-reference#delete>

1.7 Validate webhook signatures

```
client = nexmo.Client(signature_secret='secret')  
  
if client.check_signature(request.query):  
    # valid signature
```



```
else:
    # invalid signature
```

Docs: <https://docs.nexmo.com/messaging/signing-messages>

Note: you'll need to contact support@nexmo.com to enable message signing on your account before you can validate webhook signatures.

1.8 JWT parameters

By default the library generates short lived tokens for JWT authentication.

Use the auth method to specify parameters for a longer life token or to specify a different token identifier:

```
client.auth(nbf=nbf, exp=exp, jti=jti)
```

1.9 API Coverage

- Account
 - [X] Balance
 - [X] Pricing
 - [X] Settings
 - [X] Top Up
 - [X] Numbers
 - * [X] Search
 - * [X] Buy
 - * [X] Cancel
 - * [X] Update
- Number Insight
 - [X] Basic
 - [X] Standard
 - [X] Advanced
 - [] Webhook Notification
- Verify
 - [X] Verify
 - [X] Check
 - [X] Search
 - [X] Control
- Messaging
 - [X] Send

- [] Delivery Receipt
- [] Inbound Messages
- [X] Search
 - * [X] Message
 - * [X] Messages
 - * [X] Rejections
- [X] US Short Codes
 - * [X] Two-Factor Authentication
 - * [X] Event Based Alerts
 - [X] Sending Alerts
 - [X] Campaign Subscription Management
- Voice
 - [X] Outbound Calls
 - [] Inbound Call
 - [X] Text-To-Speech Call
 - [X] Text-To-Speech Prompt

1.10 License

This library is released under the MIT License

API Reference

exception nexmo.**AuthenticationError**

class nexmo.**Client** (***kwargs*)

auth (*params=None, **kwargs*)

buy_number (*params=None, **kwargs*)

cancel_number (*params=None, **kwargs*)

cancel_verification (*request_id*)

check_signature (*params*)

check_verification (*request_id, params=None, **kwargs*)

check_verification_request (*params=None, **kwargs*)

control_verification_request (*params=None, **kwargs*)

create_application (*params=None, **kwargs*)

create_call (*params=None, **kwargs*)

delete (*host, request_uri*)

delete_application (*application_id*)

get (*host, request_uri, params={}*)

get_account_numbers (*params=None, **kwargs*)

get_application (*application_id*)

get_applications (*params=None, **kwargs*)

get_available_numbers (*country_code, params=None, **kwargs*)

get_balance ()

get_basic_number_insight (*params=None, **kwargs*)

get_call (*uuid*)

get_calls (*params=None, **kwargs*)

get_country_pricing (*country_code*)

get_event_alert_numbers ()

get_message (*message_id*)

```
get_message_rejections (params=None, **kwargs)
get_number_insight (params=None, **kwargs)
get_prefix_pricing (prefix)
get_sms_pricing (number)
get_verification (request_id)
get_verification_request (request_id)
get_voice_pricing (number)
initiate_call (params=None, **kwargs)
initiate_tts_call (params=None, **kwargs)
initiate_tts_prompt_call (params=None, **kwargs)
parse (host, response)
post (host, request_uri, params)
put (host, request_uri, params)
request_number_insight (params=None, **kwargs)
resubscribe_event_alert_number (params=None, **kwargs)
search_messages (params=None, **kwargs)
send_2fa_message (params=None, **kwargs)
send_event_alert_message (params=None, **kwargs)
send_marketing_message (params=None, **kwargs)
send_message (params)
send_ussd_prompt_message (params=None, **kwargs)
send_ussd_push_message (params=None, **kwargs)
send_verification_request (params=None, **kwargs)
signature (params)
start_verification (params=None, **kwargs)
topup (params=None, **kwargs)
trigger_next_verification_event (request_id)
update_application (application_id, params=None, **kwargs)
update_call (uuid, params=None, **kwargs)
update_number (params=None, **kwargs)
update_settings (params=None, **kwargs)
exception nexmo.ClientError
exception nexmo.Error
exception nexmo.ServerError
```

Indices and tables

- `genindex`
- `modindex`
- `search`

n

nexmo, [7](#)

A

auth() (nexmo.Client method), 7
AuthenticationError, 7

B

buy_number() (nexmo.Client method), 7

C

cancel_number() (nexmo.Client method), 7
cancel_verification() (nexmo.Client method), 7
check_signature() (nexmo.Client method), 7
check_verification() (nexmo.Client method), 7
check_verification_request() (nexmo.Client method), 7
Client (class in nexmo), 7
ClientError, 8
control_verification_request() (nexmo.Client method), 7
create_application() (nexmo.Client method), 7
create_call() (nexmo.Client method), 7

D

delete() (nexmo.Client method), 7
delete_application() (nexmo.Client method), 7

E

Error, 8

G

get() (nexmo.Client method), 7
get_account_numbers() (nexmo.Client method), 7
get_application() (nexmo.Client method), 7
get_applications() (nexmo.Client method), 7
get_available_numbers() (nexmo.Client method), 7
get_balance() (nexmo.Client method), 7
get_basic_number_insight() (nexmo.Client method), 7
get_call() (nexmo.Client method), 7
get_calls() (nexmo.Client method), 7
get_country_pricing() (nexmo.Client method), 7
get_event_alert_numbers() (nexmo.Client method), 7
get_message() (nexmo.Client method), 7
get_message_rejections() (nexmo.Client method), 7

get_number_insight() (nexmo.Client method), 8
get_prefix_pricing() (nexmo.Client method), 8
get_sms_pricing() (nexmo.Client method), 8
get_verification() (nexmo.Client method), 8
get_verification_request() (nexmo.Client method), 8
get_voice_pricing() (nexmo.Client method), 8

I

initiate_call() (nexmo.Client method), 8
initiate_tts_call() (nexmo.Client method), 8
initiate_tts_prompt_call() (nexmo.Client method), 8

N

nexmo (module), 7

P

parse() (nexmo.Client method), 8
post() (nexmo.Client method), 8
put() (nexmo.Client method), 8

R

request_number_insight() (nexmo.Client method), 8
resubscribe_event_alert_number() (nexmo.Client method), 8

S

search_messages() (nexmo.Client method), 8
send_2fa_message() (nexmo.Client method), 8
send_event_alert_message() (nexmo.Client method), 8
send_marketing_message() (nexmo.Client method), 8
send_message() (nexmo.Client method), 8
send_ussd_prompt_message() (nexmo.Client method), 8
send_ussd_push_message() (nexmo.Client method), 8
send_verification_request() (nexmo.Client method), 8
ServerError, 8
signature() (nexmo.Client method), 8
start_verification() (nexmo.Client method), 8

T

topup() (nexmo.Client method), 8

`trigger_next_verification_event()` (nexmo.Client method),
8

U

`update_application()` (nexmo.Client method), 8

`update_call()` (nexmo.Client method), 8

`update_number()` (nexmo.Client method), 8

`update_settings()` (nexmo.Client method), 8