



 medianasms / **python-rest-sdk** Public

medianasms python SDK

 BSD-3-Clause license 0 stars  0 forks  Activity Star Watch[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) master ▾

...

**bulwarkzero** increase package version ...on Feb 12, 2020  13[View code](#)

# Mediana SMS api SDK

This repository contains open source Python client for `mediana_sms` api.  
Documentation can be found at: <http://docs.medianasms.com>.

**build** **passing**

## Installation

you can install this package with either pip:

```
pip install medianasms
```

or running bellow command after downloading source:

```
python setup.py install
```

## Examples

For using sdk, you have to create a client instance that gives you available methods on API

```
from medianasms import Client

# you api key that generated from panel
api_key = "api-key"

# create client instance
```

☰ README.md

## Credit check

```
# return float64 type credit amount
credit = sms.get_credit()
```

## Send one to many

For sending sms, obviously you need `originator` number, `recipients` and `message`.

```
bulk_id = sms.send(
    "+9810001",          # originator
    ["98912xxxxxxx"],    # recipients
    "mediana is awesome" # message
)
```

If send is successful, a unique tracking code returned and you can track your message status with that.

## Get message summery

```
bulk_id = "message-tracking-code"

message = sms.get_message(bulk_id)

print(message.status) # get message status
```

```
print(message.cost)      # get message cost
print(message.payback)   # get message payback
```

## Get message delivery statuses

```
bulk_id = "message-tracking-code"

statuses, pagination_info = sms.fetch_statuses(bulk_id, 0, 10)

# you can loop in messages statuses list
for status in statuses {
    print("Recipient: %s, Status: %s" % (status.recipient, status.status))
}

print("Total: ", pagination_info.total)
```

## Inbox fetch

fetch inbox messages

```
messages, pagination_info = sms.fetch_inbox(0, 10)

for message in messages {
    print("Received message %s from number %s in line %s" % (message.message,
    })
```

## Pattern create

For sending messages with predefined pattern(e.g. verification codes, ...), you have to create a pattern. a pattern at least have a parameter. parameters defined with `%param_name%`.

```
pattern = sms.create_pattern(r"%name% is awesome", False)

print(pattern.code)
```

## Send with pattern

```
pattern_values = {
    "name": "Mediana",
```

```
}

bulk_id = sms.send_pattern(
    "t2cfmnyo0c",    # pattern code
    "+9810001",      # originator
    "98912xxxxxxx",  # recipient
    pattern_values,  # pattern values
)
```

## Error checking

```
from medianasms import HTTPError, Error, ResponseCode

try:
    bulk_id = sms.send("9810001", ["98912xxxxx"], "mediana is awesome")
except Error as e: # mediana sms error
    print("Error handled => code: %s, message: %s" % (e.code, e.message))
    if e.code == ResponseCode.ErrUnprocessableEntity.value:
        for field in e.message:
            print("Field: %s , Errors: %s" % (field, e.message[field]))
except HTTPError as e: # http error like network error, not found, ...
    print("Error handled => code: %s" % (e))
```

## Releases

 3 tags

## Packages

No packages published

## Languages

● Python 100.0%