SMSWithoutBorders-OpenAPI

Afkanerd Technologies

Jun 28, 2022

CONTENTS

CHAPTER

ONE

OVERVIEW

Send bulk and single sms messages at scale

CHAPTER

TWO

REFERENCE DOCUMENTATION

2.1 Service endpoint

A service endpoint is a base URL that specifies the network address of an API service. One service might have multiple service endpoints. This service has the following service endpoint and all URIs below are relative to this service endpoint:

- `https://developers.smswithoutborders.com:14000`

2.2 API V1 Endpoints

Action	Endpoint	Parameters	Request body
Send SMS	POST /v1/sms	None	• auth_id = STRING • data = [{text = STRING, number = STRING, operator_name = STRING}] • callback_url = STRING • uuid = STRING
Get Phone number operator name	POST /v1/sms/operators	None	• [{text = STRING, number = STRING, operator_name = STRING}]

Warning: We advise you to use this endpoint safely in the back-end to avoid exposing your developer's token to unauthorized persons.

Examples using curl

2.2.1 Send single SMS message

Note: The `uuid` key is any random string provided by the user used to identify the request. If left empty, OpenAPI will populate the `uuid` key with a randomly generated uuid.

The `callback_url` will be invoked after the request is complete with a `POST` in the form:

```
{
    "errors": [{
        "operator_name":"",
        "number":"",
        "error_message": "",
        "timestamp": ""
        }],
    "uuid": ""
}
```

- If the `errors` array is empty all messages were sent out successfully.

```
curl --location --request POST 'https://developers.smswithoutborders.com:14000/v1/sms' \
--header 'Content-Type: application/json' \
--data-raw '{
"auth_id":"",
"data": [{
    "operator_name":"",
    "text":"",
    "number":""
    }],
"callback_url": "",
"uuid": ""
}'
```

2.2.2 Send bulk SMS messages

```
curl --location --request POST 'https://developers.smswithoutborders.com:14000/v1/sms' \
--header 'Content-Type: application/json' \
--data-raw '{
"auth_id":"",
"data": [{
    "operator_name":"",
    "text":"",
    "number":""
    },
    {
    "operator_name":"",
    "text":"",
    "number":""
    },
    {
    "operator_name":"",
```

(continues on next page)

(continued from previous page)

```
    "text":"",
    "number":""
    }],
"callback_url": "",
"uuid": ""
}'
```

2.2.3 Get Phone Number operator name

If the `operator_name` key is an empty string or not present in the request, It will be generated and populated in the response. But if the `operator_name` key is present it won't be modified in the response.

```
curl --location --request POST 'https://developers.smswithoutborders.com:14000/v1/sms/
↪operators' \
--header 'Content-Type: application/json' \
--data-raw '[
    {
    "operator_name":"",
    "text":"",
    "number":""
    },
    {
    "operator_name":"",
    "text":"",
    "number":""
    },
    {
    "operator_name":"",
    "text":"",
    "number":""
    }
]'
```

Note: The phone number format to be used in the request bodies of the API calls should be E.164.

CHAPTER

THREE

HOW TO SETUP OPENAPI

3.1 Setup configuration file

The configuration file is located in the `config` directory.

```
cp config/example.default.ini default.ini
```

3.2 Link OpenAPI to SMSWithoutBorders developer's console

In the configuration file `default.ini`, the section `DEVELOPER` is to setup SMSWithoutBorders developer's console.

1. HOST: The url pointing to SMSWithoutBorders developer's console (without port number)
2. PORT: The port number SMSWithoutBorders developer's console connects to.
3. VERSION: The version number of SMSWithoutBorders developer's console you're trying to connect to. Prefix the version number with a "v". Example v1, v2, e.t.c

3.3 Setup access credentials

Access credentials are found in the root directory of the repository named `setup.ini`.

```
cp example.setup.ini setup.ini
```

Note: The values of your access credentials most match those of the SMSWithoutBorders developer's console you're connecting to else connection will be denied.

CHAPTER

FOUR

USAGE

4.1 Connect Gateway Client

Note: This tutorial requires you to have the developer's token (Auth_key and Auth_id) from the SMS Without Borders developer console. If you don't have the developer's token, head over to SMS Without Borders developer console and create one.

Note: This tutorial also requires you to have a copy of the SMS Without Borders Gateway client set up on your device. If you do not have a copy of the SMS Without Borders Gateway client set up on your device, head over to SMS Without Borders Gateway Client and set it up

Having acquired your SMS Without Borders developer's token (Auth_key and Auth_id) and a copy of the SMS Without Borders Gateway client setup on your device, you can now connect OpenAPI to your gateway client's instance in a few steps.

4.1.1 1. Configure your gateway client

The configuration file (config.ini) is located in the .configs directory of your gateway client's directory. Place your developer's token (Auth_key and Auth_id) under the `[NODE]` section of your configuration file.

```
[NODE]
api_id=
api_key=
```

also configure the `connection_url` to point the server you're trying to conect to. To connect to the SMSWithoutBorders server use `developers.smswithoutborders.com`

```
connection_url=developers.smswithoutborders.com
```

Note: The `connection_url` can be your custom server visit RabbitMQ to set-up your instance.

4.1.2 2. Restart your gateway client

You will need to restart your gateway and cluster for changes to take effect. In the root of the repo use the command:

```
make restart
```

4.1.3 All done!

You are now ready to send out bulk SMS messages with OpenAPI. Head over to your API agent and send out bulk SMS messages with your developer's token.

Example using curl

```
curl --location --request POST 'https://developers.smswithoutborders.com:14000/v1/sms' \
--header 'Content-Type: application/json' \
--data-raw '{
"auth_id":"",
"data": [{
    "operator_name":"",
    "text":"",
    "number":""
    }]
}'
```

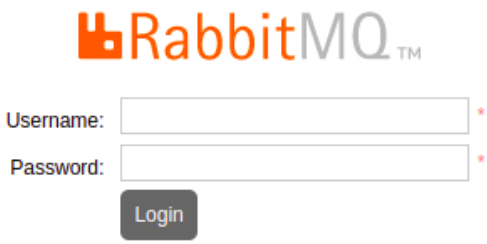
See *Reference Documentation* for more API references

4.2 Manage OpenAPI messages

Manage your OpenApi messages through the SMSWithoutBorders RabbitMQ dashboard.

4.2.1 1. Login

Visit SMSWithoutBorders RabbitMQ dashboard.



- Username = Your developer's auth_id
- Password = Your developer's auth_key

Note: If you do not have the developer's token (Auth_key and Auth_id), head over to SMS Without Borders developer console and create one.

CHAPTER

FIVE

EXAMPLES

5.1 View errors (if any) from Open API's callback URL

Note: This tutorial requires you to have the developer's token (Auth_key and Auth_id) from the SMS Without Borders developer console. If you don't have the developer's token, head over to SMS Without Borders developer console and create one. You will also have to set up Open API and have it running.

Once you have Open API all set up, you can start sending out single or bulk SMS messages.

In order to actually see the output of the `callback_url` in your post request which should look somewhat like this:

```
curl --location --request POST 'https://developers.smswithoutborders.com:14000/v1/sms' \
--header 'Content-Type: application/json' \
--data-raw '{
"auth_id":"",
"data": [{
    "operator_name":"",
    "text":"",
    "number":""
    }],
"callback_url": "",
"uuid": ""
}'
```

You can locally set up a tiny Flask app like this and run it

```
from flask import Flask, request
import logging

app = Flask(__name__)

logging.basicConfig(level='INFO', format='%(asctime)s-%(levelname)s-%(message)s')

@app.route("/", methods=['GET', 'POST'])
def log_openapi_errors():
    if request.method == 'POST':
        error_data = request.get_json()
        logging.error("\033[31m%s\033[00m", error_data)
        return error_data
    return "<h1>Check your logs to see if you got errors</h1>"
```

(continues on next page)

(continued from previous page)

```
if __name__=='__main__':
    app.run(debug=True)
```

Once your tiny Flask app is running, add it's localhost URL as the value of the `callback_url` of your Open API post request.

```
curl --location --request POST 'https://developers.smswithoutborders.com:14000/v1/sms' \
--header 'Content-Type: application/json' \
--data-raw '{
"auth_id":"",
"data": [{
    "operator_name":"",
    "text":"",
    "number":""
    }],
"callback_url": "http://127.0.0.1:5000",
"uuid": ""
}'
```

From now when you send a post request to Open API, you can now check the logs of your tiny Flask app to see the the result of the callback URL. It should return an array of json objects if there happen to be any errors in your post request to Open API. Your error log should look like this:

```
2022-05-11 11:16:31,934-ERROR-{'errors': [{'operator_name': 'MTN Cameroon', 'number':
↪'+2376728-+72885', 'error_message': '(1) The string supplied did not seem to be a␣
↪phone number.', 'timestamp': '2022-05-11 11:16:31.931214'}], 'uuid': '6d6b83e2-d113-
↪13ec-ae9a-cba900762ab3'}
```

CHAPTER

SIX

HOST OPENAPI

read more ...

CHAPTER

SEVEN

HOST GATEWAY CLIENT

Deku is a linux SMS management Gateway. It can both receive and send out SMS messages using the Linux ModemManager utilities. It is aimed at being a complete toolset of everything SMS linux. It functions best with USB 2G/3G Modems

read more ..

CHAPTER

EIGHT

LICENSING

This project is licensed under the GNU General Public License v3.0.