API Document

MISSALL OTP – ASYNC – API

Documentation

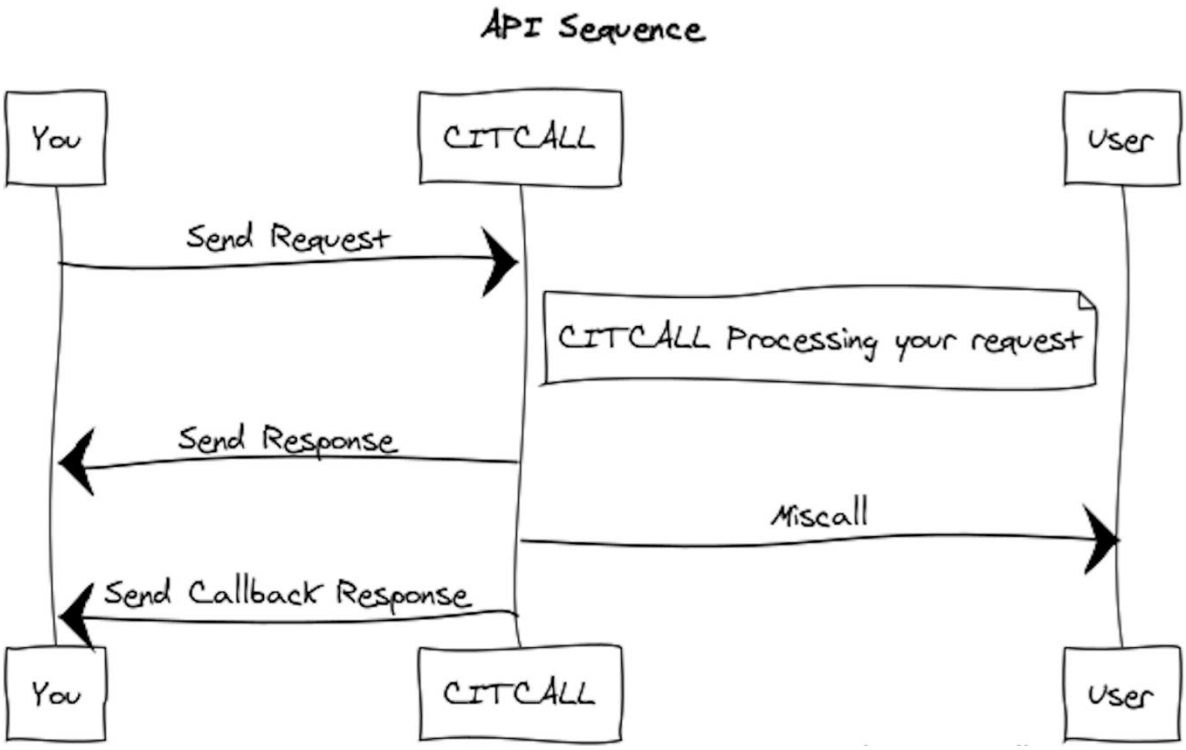
V 1.0

Date	:	5 Februari 2018
Version	:	1.0
Prepared by	:	Muhammad Chairul

Document History

Version	Author (s)	Date	Description
1.0	Muhammad Chairul	5 Feb 2018	

1. Introduction



This document will provide instructions on how to integrate CITCALL services by using CITCALL HTTP application programming interface (HTTP API). Use HTTP API for making miscall. CITCALL's API is based on REST standards, enabling you to use your browser for accessing URLs. Use any HTTP client in any programming language to interact with our API.

2. Integration

2.1. API key authorization

This is the most secure authorization type and the one with the most flexibility.

API keys can be generated by calling the dedicated API method. Furthermore, API keys can have a limited scope and cover only some API methods. Lastly, they can be revoked at any time. This range of possibilities makes API keys well suited for separating the API access rights across multiple applications or use cases. Finally, the loss of an API key is easily manageable.

in this case, the credentials included in the Authorization header should be a Base64 encoded username and apikey combination. More formally, basic authentication header can be constructed in two steps:

- Username and password are concatenated using the colon (:) as a separator username:apikey.
- The resulting string is encoded using the RFC2045-MIME variant of Base64.

in this case, the credentials included in the Authorization header should be a Base64 encoded username and apikey combination. More formally, basic authentication header can be constructed in two steps:

Example:

```
Userid: "Aladdin"
Apikey: "openSesame"
Concatenated string: "Aladdin:openSesame"
Base64 encoded string: "QWxhZGRpbjpvcGVuIHNlc2FtZQ=="
Authorization header: "QWxhZGRpbjpvcGVuIHNlc2FtZQ=="
```

2.2. Data Security System

For security purpose, all messages that sent from and to CITCALL will be encrypted with a library (provided, see Appendix A). The encryption algorithm needs a API Key which will be provided by CITCALL. The API Key is a regular 32 hexadecimal characters.

Sample of a secret key: **fe01ce2a7fbac8fafaed7c982a04e229**

***Note**: API Key will be generated manual, please contact Citcall's administrator to get your API key*

2.3. You will interact with CITCALL API using JSON POST on a single URL:

URL: **https://gateway.citcall.com/v2/asynccall**

These are the basic rules to integrate with CITCALL API system:

- Using HTTP POST
- Header Content-Type should be application/json
- Request data should be encrypted with CITCALL encryption method
- Response data should be decrypted with CITCALL decryption method

Example request

Raw data:

```
{
     "msisdn":"081234567890",
     "gateway":1,
     "clientUniqueId":"1234567890"
}
```

Encrypt the raw data before making a request, result of the encryption will look like this:

```
X3wBT1F3AlAFTHlKcjQQPEMuNHQNPXheWEh5UVh_Tldh
emxJQi4rEzUSLEhyAU5_ci0yOzgJPHsVfzcSCBJ7QWJs
e09Te39VB1MEf3JE
```

Note: this is not the real result from encrypting the above JSON, because the result will vary over time

Send the encrypted data along with the provided Client ID:

```
{
     "data":"X3wBT1F3AlAFTHlKcjQQPEMuNHQNPXhe
     WEh5UVh_TldhemxJQi4rEzUSLEhyAU5_ci0yOzgJ
     PHsVfzcSCBJ7QWJse09Te39VB1MEf3JE"
}
```

Example response of successful request:

```
{
     "data":"Xn8AUFF3AlAFTHlKcjkAdQpsdgJ1R2oP
     FXw1P11rChQLLS8QE2l2QT0ONEI0NT8NcnV8BatM
     ellbS3xUWwJHQVRsPgwLLDhBCj17AQF8TQsAAn8H
     CD10SA53OgIaKhBBYnt2PwMzMwQ-
     DyA9OTgSOBkuaAx1THpZW0t8VFsCR0El"
}
```

Decrypt the response data:

```
{
     "rc":"00",
     "info":"success",
     "msisdn":"+6281234567890",
     "token":"02150808955",
     "gateway":1,
     "clientUniqueId":"1234567890"
}
```

Example response of failed request (will be sent unencrypted):

```
{
     "rc":"98",
     "info":"Authorization failed"
}
```

2.4. Miscall

This section provide the information for miscall to the MSISDN.

Accepted parameters to create an miscall

Parameter	Mandatory	Notes
msisdn	Yes	End-User mobile number
gateway	Yes	Gateway number (1,2,3,4)
clientUniqueId	No	Unique id that you want to associate with the Request

The response returns the following:

- Rc: Respon Code.

Possible status values:

Code	Description
00	queued
99	Wrong Method
98	Authorization failed
96	apikey not found or non active
88	missing parameter
77	userid not found
06	unknown error / failed
34	Service temporary unavilable

66	Maintenance in progress
14	insuficient amount

- **trxid** : unique message ID automatically generated by Citcall.
- **msisdn** : Recipients information.
- **token :** Number received by the end user**.**
- **gateway** : number of gateway.
- **clientUniqueId** : Unique id that you want to associate with the Request.

2.5. Callback

Parameters of payment Callback.

Parameter	Notes
rc	Respon Code
trx_id	unique message ID automatically generated by Citcall
msisdn	End-User mobile number
via	Route used to end-user.
token	Number received by the end user.
dial_code	Dial Code
dial_status	Dial Status
call_status	Call Status
result	Result
clientUniqueId	Unique id that you want to associate with the Request

Notes:

- **Callback URL**: provided by client,
- Callback using JSON POST
- To add callback on dashboardIt is still manually doing by Citcall's administrator, please send the callback url

APPENDIX

A. Encryption library:

   a. PHP:

      https://1drv.ms/u/s!Asa8cPF5omkWgxeoG5yUcQ4eLpQG

   b. JAVA:

      https://1drv.ms/f/s!Asa8cPF5omkWgxh-OktFtQO0r5F2

   c. Python:

      https://1drv.ms/f/s!Asa8cPF5omkWgxtY34TFNqhI-jUN