

# 『絶対リーチ！<sup>®</sup> SMS communicationPlatform』

## API 仕様書

*Version 1.47*

本ドキュメントは 2024 年 3 月末をもって更新を終了しております。

最新の API 仕様につきましては下記をご参照ください。

<https://doc.aossm.com/zettai-reach/openapi.html>

2023 年 12 月 27 日  
AI CROSS 株式会社

## 改版履歴

版数	日付	内容
1.40	2022/08/18	最新の API (P5) に統一 ・全キャリア長文対応に合わせ、最新の API のみを記載しております。 ・わかりづらい箇所等一部記述を見直しました。 ・一部構成を変更しました。
1.41	2022/08/25	2.4. ステータス取得(Common) で一部パラメータ名の表記を修正しました（先頭大文字→小文字）
1.42	2022/10/28	パラメータ名の誤記を修正しました（先頭大文字→小文字）
1.43	2022/12/6	1.3. TPS の表現を修正しました。 2.6.1 domain パラメータを追記しました。
1.44	2023/06/26	2.9. 通数取得 API を新規追加しました。
1.45	2023/11/21	2.4.2 ステータス取得(Common) のレスポンスに「separatedSuccessCount」項目を追加しました。
1.46	2023/12/7	smsCode の文字数を 5 桁または 6 桁に修正しました。
1.47	2023/12/27	CommonMT 送信、予約送信確認、予約送信一括キャンセルのパラメータに「groupTag」項目を追加しました。

## 目次

1. はじめに .....	5
1.1. 接続方式 .....	5
1.2. 接続元 IP アドレスの制限 .....	5
1.3. TPS(Transactions Per Second) .....	5
1.4. 推奨 API .....	5
1.5. 配信プラットフォーム刷新にともなう API URL の変更 .....	5
2. API 仕様 .....	6
2.1. CommonMT 送信 .....	6
2.1.1. リクエスト .....	6
2.1.2. レスポンス .....	14
2.1.3. 結果コード .....	15
2.2. CommonMT 予約送信確認 .....	16
2.2.1. リクエスト .....	16
2.2.2. レスポンス .....	17
2.2.3. 結果コード .....	19
2.3. CommonMT 予約送信キャンセル .....	20
2.3.1. リクエスト .....	20
2.3.2. レスポンス .....	21
2.3.3. 結果コード .....	22
2.3.4. リクエスト(一括キャンセル) .....	23
2.3.5. レスポンス(一括キャンセル) .....	24
2.3.6. 結果コード(一括キャンセル) .....	25
2.4. ステータス取得(Common) .....	26
2.4.1. リクエスト .....	26
2.4.2. レスポンス .....	27
2.4.3. 結果コード .....	31
2.5. DLR 通知(Common) .....	32
2.5.1. 通知パラメータ .....	32
2.5.2. レスポンス .....	42
2.6. ショート URL 登録 .....	43
2.6.1. リクエスト .....	43
2.6.2. レスポンス .....	44
2.6.3. 結果コード .....	45
2.7. 登録済み定型文取得 .....	46
2.7.1. リクエスト .....	46
2.7.2. レスポンス .....	47
2.7.3. 結果コード .....	49
2.8. 電話番号クリーニング .....	50
2.8.1. リクエスト .....	50
2.8.2. レスポンス .....	51
2.8.3. 結果コード .....	52
2.9. 通数集計 .....	53
2.9.1. リクエスト .....	53

---

2.9.2. レスポンス .....	54
2.9.3. 結果コード .....	55
3. 別記事項 .....	56
3.1. 国際電話番号形式について .....	56
3.2. ステータス ID について .....	56
3.3. レスポンスの rawdata について .....	56
3.4. 詳細内部コード(detailedStatusId)について .....	57
3.5. Y!Mobile (旧 Willcom、PHS 回線)について .....	57

## 1. はじめに

本書では「絶対リーチ!。SMS communicationPlatform」(以下、「本サービス」といいます)の外部 API 仕様を定義します。

### 1.1. 接続方式

API 接続は、HTTPS の GET/POST メソッドを用いて接続します。

※SSL/TLS の方式は TLS1.2 に限定しております。TLS1.0、1.1 では接続できません。

### 1.2. 接続元 IP アドレスの制限

セキュリティを強化するため API を使用するアクセス元 IP アドレスを制限することが可能です。管理画面の API 限定 IP 設定により設定できます。

### 1.3. TPS (Transactions Per Second)

CommonMT 送信 API リクエストは全体で 600TPS (1 秒間あたりのリクエスト件数)での受け付けが可能です。

※上限を超過した場合は一時的に流量制限を実施する場合があります。

### 1.4. 推奨 API

最新の「CommonMT 送信」をご利用下さい。

既に旧送信 API をご利用のお客様も「CommonMT 送信」に変更をお勧めいたします。

### 1.5. 配信プラットフォーム刷新にともなう API URL の変更

2022 年 3 月に配信プラットフォームを刷新し、API URL も変更となりました。

この仕様書では、各 API の新 URL を記載しております。

## 2. API 仕様

本サービスが提供する外部向け API 仕様を記載します。

### 2.1. CommonMT 送信

※ **KDDIは送信失敗となった場合に KDDI からの失敗ステータスの通知に 20 分程度時間を要する場合があります。**

#### 2.1.1. リクエスト

接続先 URL

https://sms-api.aossmms.com/p5/api/mt.json

※ 以下のパラメータを URL エンコードにて追加し、POST メソッドにて呼び出します。

No	項目名	データ型	必須	概要
1	token	String	○	アカウント登録時に発行されるアクセスキーを指定します。トークンの値につきましては、管理画面のアカウント情報をご確認ください。 ・半角英数字
2	clientId	String	○	契約クライアント ID を指定します。 割り当てられている契約クライアント ID につきましては、管理画面のクライアント情報をご確認ください。 ・半角数字
3	smsCode	String	○	送信元 SMS コードを指定します。 割り当てられている SMS コードにつきましては、管理画面のクライアント情報をご確認ください。 ・5桁または6桁の半角数字
4	message	String	○	送信メッセージを指定します。 ・最大 660 文字 (全角/半角問わず文字数でカウントされます)
5	phoneNumber	String	○	送信先電話番号を指定します。 日本国内形式 (070xxxxxxx/080xxxxxxx/090xxxxxxx)、国際電話番号形式のどちらも指定することが可能です。 ※国際電話番号形式については『別記事項・国際電話番号形式について』をご参照ください。
6	carrierId	String		キャリア ID を指定します。指定しない場合は本サービスにて電話番号より送信先キャリアを自動判定します。 101:AU 103:Docomo 105:Softbank 106:楽天モバイル

7	clientTag	String	送信ステータス確認用の任意の識別文字列を指定します。必ずユニークな値を指定してください。値の重複があるとエラーになります。 ・お客様側で設定可能な文字数は最大 200 文字になります。
8	scheduleTime	String	予約送信時間を指定します。 フォーマットは「YYYY-mm-dd HH:MM」となります
9	groupTag	String	予約確認・一括キャンセル用の任意識別文字列を指定します。 ・最大 200 文字

## リクエストサンプル

## ➤ CommonMt\_sample.ps1

```
[System.String] $logDir = Split-Path -Parent $MyInvocation.MyCommand.Path
[System.String] $url = "https://sms-api.aossms.com/p5/api/mt.json" #接続先URL
[System.String] $token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxx" #トークンを設定
[System.String] $clientId = "xxxx" #クライアントIDを設定
[System.String] $smsCode = "xxxx" #SMSコードを設定
[System.String] $message = "sample message (powershell)" #送信したいメッセージを設定
[System.String] $phoneNumber = "09001234567" #送信したい宛先電話番号を設定
[System.String] $clientTag = "uniqueTag" #一意となるタグを設定

#SMS送信をリクエスト
$response = Invoke-WebRequest -Method Post -Body @{token="{$token}"; clientId="{$clientId}"; smsCode="{$smsCode}";
message="{$message}"; phoneNumber="{$phoneNumber}"; clientTag="{$clientTag}"} $url}
#レスポンス結果を出力
$response.content | Out-File $logDir\{$clientTag}.txt
```

## ➤ CommonMt\_sample.py

```
import requests

url = "https://sms-api.aossms.com/p5/api/mt.json" #接続先URL

data = {
    "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx" #トークンを設定
    , "clientId": "xxxx" #クライアントIDを設定
    , "smsCode": "xxxx" #SMSコードを設定
    , "message": "sample message (python)" #送信したいメッセージを設定
    , "phoneNumber": "09001234567" #送信したい宛先電話番号を設定
    , "clientTag": "uniqueTag" #一意となるタグを設定
}

req = requests.post(url, data=data) #SMS送信をリクエスト
print(req.text) #レスポンス結果を出力
```



➤ CommonMt\_sample.bat

```
@echo off
rem 接続先URL
set url=https://sms-api.aossm.com/p5/api/mt.json

rem トークンを設定
set token=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

rem クライアントIDを設定
set clientId=xxxx

rem SMSコードを設定
set smsCode=xxxxx

rem 送信したいメッセージを設定
set message=sample+message(curl)

rem 送信したい宛先電話番号を設定
set phoneNumber=09001234567

rem 一意となるタグを設定
set clientTag=uniqueTag

rem SMS送信をリクエスト、レスポンス結果を出力
curl -s -X POST -d
"token=%token%&clientId=%clientId%&smsCode=%smsCode%&message=%message%&phoneNumber=%phoneNumber%&clientTag=%clientTag%" "%url%" > result.txt
```

## ➤ CommonMt\_sample.php

```
<?php
$url = 'https://sms-api.aossms.com/p5/api/mt.json';
$data = array(
    'token' => 'xxxxxxxxxxxxxxxxxxxxxxxxxxxx' //トークンを設定
    , 'clientId' => 'xxxx' //クライアントIDを設定
    , 'smsCode' => 'xxxx' //SMSコードを設定
    , 'message' => 'sample message (php)' //送信したいメッセージを設定
    , 'phoneNumber' => '09001234567' //送信したい宛先電話番号を設定
    , 'clientTag' => 'uniqueTag' //一意となるタグを設定
);
$content = http_build_query($data); //URLエンコードされたクエリ文字列を生成
$header = array( //リクエストヘッダを設定
    "Content-Type: application/x-www-form-urlencoded"
    , "Content-Length: ".strlen($content)
);
$options = array( //HTTPコンテキストオプションを設定
    "http" => array(
        'method' => 'POST'
        , 'header' => implode("\r\n", $header)
        , 'content' => $content
    )
);
$content = file_get_contents($url, false, stream_context_create($options)); //SMS送信をリクエスト
file_put_contents('result.txt', $content); //レスポンス結果を出力
```

## ➤ CommonMt\_sample.java

```
package com.aicross.sms;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import javax.net.ssl.HttpURLConnection;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class SmsService {
    private static final Logger logger = LoggerFactory.getLogger(SmsService.class);
    public static void main(String[] args) throws Exception {
        HttpURLConnection conn = null;
        StringBuffer result = new StringBuffer();
        try {
            URL url = new URL("https://sms-api.aossm.com/p5/api/mt.json"); //接続先URL
            String token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"; //トークンを設定
            String clientId = "xxxx"; //クライアントIDを設定
            String smsCode = "xxxxx"; //SMSコードを設定
            String message = "sample message(java)"; //送信したいメッセージを設定
            String phoneNumber = "09001234567"; //送信したい宛先電話番号を設定
            String clientTag = "uniqueTag"; //一意となるタグを設定

            //リクエストパラメータを生成
            StringBuilder param = new StringBuilder();
            param.append("token="+token);
            param.append("&clientId="+clientId);
            param.append("&smsCode="+smsCode);
            param.append("&message="+message);
            param.append("&phoneNumber="+phoneNumber);
            param.append("&clientTag="+clientTag);

            //SMS送信をリクエスト
            conn = (HttpURLConnection)url.openConnection();
            conn.setDoOutput(true);
            conn.setRequestMethod("POST");
            conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded;");
            OutputStreamWriter out = new OutputStreamWriter(conn.getOutputStream());
            out.write(param.toString());
            out.close();
            conn.connect();

            =====次ページ=====
        }
    }
}
```

```
=====前ページ=====
```

```
//レスポンスコードを取得
int status = conn.getResponseCode();
if (status == HttpURLConnection.HTTP_OK) {
    //レスポンスコードOKの場合、レスポンス結果を取得
    final InputStream input = conn.getInputStream();
    String encoding = conn.getContentEncoding();
    if (null == encoding) {
        encoding = "UTF-8";
    }
    final InputStreamReader inReader = new InputStreamReader(input, encoding);
    final BufferedReader bufReader = new BufferedReader(inReader);
    String line = null;
    while ((line = bufReader.readLine()) != null) {
        result.append(line);
    }
    bufReader.close();
    inReader.close();
    input.close();
} else {
    //レスポンスコードOKでない場合、レスポンスコードを出力
    logger.error(String.valueOf(status));
}
} catch (Exception ex) {
    logger.error(ex.getLocalizedMessage());
} finally {
    if (conn != null) {
        conn.disconnect();
    }
}
//レスポンス結果を出力
logger.info(result.toString());
}
}
```

## ➤ CommonMt\_sample.js

```
const axios = require('axios')
const qs = require('qs')

const instance = axios.create({
  baseURL: 'https://sms-api.aossmms.com', // 接続先URL
  headers: {'Content-Type': 'application/x-www-form-urlencoded'}, // リクエストヘッダを設定
})
const data = {
  token: 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx', // トークンを設定
  clientId: 'xxxx', // クライアントIDを設定
  smsCode: 'xxxx', // SMSコードを設定
  phoneNumber: '09001234567', // 送信したい宛先電話番号を設定
  message: 'sample message (nodejs)', // 送信したいメッセージを設定
  clientTag: 'uniqueTag' // 一意となるタグを設定
}
const content = qs.stringify(data) // URLエンコードされたクエリ文字列を生成

instance.post('/p5/api/mt.json', content) // SMS送信をリクエスト
.then(function (response) {
  console.log(response.data) // レスポンス結果を出力
})
```

## 2.1.2. レスポンス

※ API 呼び出し結果として応答データ (JSON 形式) を返却します。

### レスポンスパラメータ

No	項目名	データ型	概要
1	responseCode	Integer	応答コードです。 正常時は0を返します。1以上の場合はエラーになります。 ※結果コードの項を参照してください。
2	responseMessage	String	応答結果を示すメッセージです。 ※結果コードの項を参照してください。
3	phoneNumber	String	リクエスト時に指定した送信先電話番号です。 国際電話番号形式で返却されます。 ※国際電話番号形式については『別記事項・国際電話番号形式について』をご参照ください。
4	smsMessage	String	送信時に指定した message 項目の内容です。

### レスポンスサンプル

➤ 成功時 (0) :

```
{
  "responseCode": 0,
  "responseMessage": "Success.",
  "phoneNumber": "+819012345678",
  "smsMessage": "test"
}
```

➤ 失敗時 (1~) :

```
{
  "responseCode": 300,
  "responseMessage": "Exceed sending count.",
  "phoneNumber": "+819012345678",
  "smsMessage": "test"
}
```

### 2.1.3. 結果コード

※ API で返却される responseCode と responseMessage を下記に定義します。

responseCode	responseMessage	説明
0	Success	リクエストの受付が成功しました。
100	Required {項目名}.	{項目名}の項目が不足しています。
110	Invalid {項目名}.	{項目名}の項目の内容が不正です。
120	{項目名} too long.	{項目名}の長さが上限を超過しています。
300	Exceed sending count.	契約プランで定められた送信可能な件数の上限を超過しています。
400	Duplicate client tag.	クライアントタグが以前にリクエストした際のクライアントタグと重複しています。
500	System error.	上記で定義外のエラーが発生しています。

※responseCode:0 が返却されていない場合、リクエストの受け付けには失敗しています。

## 2.2. CommonMT 予約送信確認

### 2.2.1. リクエスト

#### 接続先 URL

<https://sms-api.aossms.com/p5/api/checkreservation.json>

※ 以下のパラメータを URL エンコードにて追加し、GET メソッドにて呼び出します。

No	項目名	データ型	必須	概要
1	token	String	○	アカウント登録時に発行されるアクセスキーを指定します。トークンの値につきましては、管理画面のアカウント情報をご確認ください。 ・半角英数字
2	clientId	String	○	契約クライアント ID を指定します。割り当てられている契約クライアント ID につきましては、管理画面のクライアント情報をご確認ください。 ・半角数字
3	smsCode	String	○	送信元 SMS コードを指定します。 割り当てられている SMS コードにつきましては、管理画面のクライアント情報をご確認ください。 ・5桁または6桁の半角数字
4	clientTag	String	△	送信ステータス確認用の任意の識別文字列を指定します。 ・最大 200 文字 ※本項目または scheduleTime, scheduleDate のいずれかが必須。 ※優先度は clientTag>scheduleTime>scheduleDate になります。 ※本項目を指定した場合、指定文字列を保持する SMS のステータスを返却します。本項目を指定しない場合、SMS 送信日のデータを検索して返却します。
5	scheduleTime	String	△	予約送信時間を文字列で指定します。 フォーマットは「YYYY-mm-dd HH:MM」となります。 ※本項目または clientTag, scheduleDate のいずれかが必須。 ※優先度は clientTag>scheduleTime>scheduleDate になります。 groupTag も指定されている場合は組み合わせ条件になります。
6	scheduleDate	String	△	予約送信日を半角数字で指定します。 フォーマットは「YYYYMMDD」となります。 ※本項目または clientTag, scheduleTime のいずれかが必須。 ※優先度は clientTag>scheduleTime>scheduleDate になります。 groupTag も指定されている場合は組み合わせ条件になります。
7	groupTag	String	△	予約リクエスト時に指定した任意識別文字列を指定します。



## 2.2.2 レスポンス

※ API 呼び出し結果として応答データ (JSON 形式) を返却します。

### レスポンスパラメータ

No	階層	項目名	データ型	概要
1	1	responseCode	Integer	応答コードです。 正常時は0を返します。1以上の場合はエラーになります。 ※結果コードの項を参照してください。
2	1	responseMessage	String	送信処理結果を示すメッセージです。 ※結果コードの項を参照してください。
3	1	count	Integer	予約中のジョブの件数です。
4	1	status	List	階層2の項目を配列で保持します。
5	2	scheduleTime	String	予約送信時間です。 フォーマットは「YYYY-mm-dd HH:MM」となります。
6	2	phoneNumber	String	送信先の電話番号です。 国際電話番号形式で返却されます。
7	2	message	String	送信時に指定した message 項目の内容です。 (que_schedule_job.contents)
8	2	smsCode	Integer	送信元 SMS コードです。5桁または6桁の数字となります。
9	2	clientTag	String	SMS 送信のリクエスト時にパラメータとして指定した clientTag です。

## レスポンスサンプル

### ➤ 成功時 (0) :

```
{
  "responseCode": 0,
  "responseMessage": "Success.",
  "count": 1,
  "status": [
    {
      "scheduleTime": "2019-01-31 08:00",
      "phoneNumber": "+818012345678",
      "message": "yoyaku",
      "smsCode": 80013,
      "clientTag": "abcedfg"
    }
  ]
}
```

### ➤ 失敗時 (1~) :

```
{
  "responseCode":110,
  "responseMessage":"Invalid token.",
  "count":"0",
  "status":[]
}
```

### 2.2.3. 結果コード

※ API で返却される responseCode と responseMessage を下記に定義します。

responseCode	responseMessage	説明
0	Success.	リクエストが成功しました。
100	Required {項目名}.	{項目名}の項目が不足しています。
110	Invalid {項目名}.	{項目名}の項目が不正です。
120	{項目名} too long.	{項目名}の長さが上限を超過しています。
130	Already send.	確認対象のクライアントタグは既に送信されています。
140	Past {項目名}.	{項目名}は過去日時です。
500	System error.	上記で定義外のエラーが発生しました。

※responseCode:0 が返却されていない場合、リクエストの受け付けには失敗しています。

## 2.3. CommonMT 予約送信キャンセル

### 2.3.1. リクエスト

#### 接続先 URL

<https://sms-api.aossmms.com/p5/api/cancelreservation.json>

※ 以下のパラメータを URL エンコードにて追加し、GET メソッドにて呼び出します。

No	項目名	データ型	必須	概要
1	token	String	○	アカウント登録時に発行されるアクセスキーを指定します。トークンの値につきましては、管理画面のアカウント情報をご確認ください。 ・半角英数字
2	clientId	Integer	○	契約クライアント ID を指定します。割り当てられている契約クライアント ID につきましては、管理画面のクライアント情報をご確認ください。 ・半角数字
3	clientTag	String	○	送信ステータス確認用の任意の識別文字列を指定します。 ・最大 200 文字

### 2.3.2. レスポンス

※ API 呼び出し結果として応答データ (JSON 形式) を返却します。

#### レスポンスパラメータ

No	階層	項目名	データ型	概要
1	1	responseCode	Integer	応答コードです。 正常時は0を返します。1以上の場合はエラーになります。 ※結果コードの項を参照してください。
2	1	responseMessage	String	送信処理結果を示すメッセージです。 ※結果コードの項を参照してください。
3	1	clientTag	List	SMS 送信のリクエスト時にパラメータとして指定した clientTag です。

#### レスポンスサンプル

➤ 成功時 (0) :

```
{
  "responseCode": 0,
  "responseMessage": "Success.",
  "clientTag": "abcdefg"
}
```

➤ 失敗時 (1~) :

```
{
  "responseCode": 110,
  "responseMessage": "Invalid clientTag.",
  "clientTag": "abcdefg"
}
```

### 2.3.3. 結果コード

※ API で返却される responseCode と responseMessage を下記に定義します。

responseCode	responseMessage	説明
0	Success.	リクエストが成功しました。
100	Required {項目名}.	{項目名}の項目が不足しています。
110	Invalid {項目名}.	{項目名}の項目が不正です。
120	{項目名} too long.	{項目名}の長さが上限を超過しています。
130	Already send.	キャンセル対象は既送信されています。
500	System error.	上記で定義外のエラーが発生しました。

※responseCode:0 が返却されていない場合、リクエストの受け付けには失敗しています。

### 2.3.4. リクエスト（一括キャンセル）

#### 接続先 URL

<https://sms-api.aossm.com/p5/api/cancelreservationall.json>

※ 以下のパラメータを URL エンコードにて追加し、GET メソッドにて呼び出します。

No	項目名	データ型	必須	概要
1	token	String	○	アカウント登録時に発行されるアクセスキーを指定します。トークンの値につきましては、管理画面のアカウント情報をご確認ください。 ・半角英数字
2	clientId	String	○	契約クライアント ID を指定します。割り当てられている契約クライアント ID につきましては、管理画面のクライアント情報をご確認ください。 ・半角英数字
3	startDate	String	△	キャンセルする日付範囲の開始日付を指定します。（※1） ・YYYYMMDD
4	endDate	String	△	キャンセルする日付範囲の終了日付を指定します。（※1） ・YYYYMMDD
5	groupTag	String	△	キャンセルするグループタグを指定します。（※1）

（※1） startDate を指定し endDate を指定しない場合、startDate 以降の予約がキャンセルされます。

endDate を指定し startDate を指定しない場合、endDate 以前の予約がキャンセルされます。

startDate、endDate を共に指定しない場合はすべての予約がキャンセルされます。

groupTag を指定し、startDate、endDate を指定しない場合、groupTag が一致する予約がキャンセルされます。

groupTag を指定し、かつ startDate、endDate のいずれかまたは両方が指定されている場合、groupTag と日付範囲が一致する予約がキャンセルされます。

startDate、endDate、groupTag をすべて指定しない場合はすべての予約がキャンセルされます。

### 2.3.5. レスポンス (一括キャンセル)

※ API 呼び出し結果として応答データ (JSON 形式) を返却します。

#### レスポンスパラメータ

No	階層	項目名	データ型	概要
1	1	responseCode	Integer	応答コードです。 正常時は0を返します。1以上の場合はエラーになります。 ※結果コードの項を参照してください。
2	1	responseMessage	String	送信処理結果を示すメッセージです。 ※結果コードの項を参照してください。

#### レスポンスサンプル

➤ 成功時 (0) :

```
{
  "responseCode": 0,
  "responseMessage": "Success."
}
```

➤ 失敗時 (1~) :

```
{
  "responseCode": 110,
  "responseMessage": "Invalid clientTag."
}
```



### 2.3.6. 結果コード (一括キャンセル)

※ API で返却される responseCode と responseMessage を下記に定義します。

responseCode	responseMessage	説明
0	Success.	リクエストが成功しました。
100	Required {項目名}.	{項目名}の項目が不足しています。
110	Invalid {項目名}.	{項目名}の項目が不正です。
500	System error.	上記で定義外のエラーが発生しました。

※responseCode:0 が返却されていない場合、リクエストの受け付けには失敗しています。

## 2.4. ステータス取得 (Common)

### 2.4.1. リクエスト

#### 接続先 URL

<https://sms-api.aossmms.com/p5/api/status.json>

※ 以下のパラメータを URL エンコードにて追加し、GET メソッドにて呼び出します。

No	項目名	データ型	必須	概要
1	token	String	○	アカウント登録時に発行されるアクセスキーを指定します。トークンの値につきましては、管理画面のアカウント情報をご確認ください。 ・半角英数字
2	clientId	String	○	契約クライアント ID を指定します。割り当てられている契約クライアント ID につきましては、管理画面のクライアント情報をご確認ください。 ・半角数字
3	date	String	△	SMS 送信日を指定します。 ・半角数字 ・YYYYMMDD 形式 ※本項目または clientTag のいずれかが必須。 ※優先度は clientTag>date になります。
4	clientTag	String	△	送信ステータス確認用の任意の識別文字列を指定します。 ・最大 200 文字 ※本項目または date のいずれかが必須。 ※優先度は clientTag>date になります。 ※本項目を指定した場合、指定文字列を保持する SMS のステータスを返却します。本項目を指定しない場合、SMS 送信日のデータを検索して返却します。

## 2.4.2. レスポンス

※ API 呼び出し結果として応答データ (JSON 形式) を返却します。

### レスポンスパラメータ

No	階層	項目名	データ型	概要
1	1	responseCode	Integer	応答コードです。 正常時は0を返します。1以上の場合エラーになります。 ※結果コードの項を参照してください。
2	1	responseMessage	String	送信処理結果を示すメッセージです。 ※結果コードの項を参照してください。
3	1	status	List	階層2の項目を配列で保持します。
4	2	dateTime	String	本サービスがキャリアよりステータスを取得した時間です。 フォーマットは「YYYYMMDD HHMMSSsss」(sssはミリ秒)となります。
5	2	phoneNumber	String	送信先の電話番号です。 国際電話番号形式で返却されます。 ※国際電話番号形式については『別記事項・国際電話番号形式について』をご参照ください。
6	2	message	String	送信先に送信されたSMS本文です。
7	2	smsCode	Integer	送信元SMSコードです。 5桁または6桁の数字となります。
8	2	carrierId	String	キャリアIDです。以下のいずれかのIDになります。 101:AU 103:NTT Docomo 105:Softbank 106:楽天モバイル ※statusIdが0,1の場合本項目の値は不定です ※該当キャリアが判別できない場合、nullまたは999となります。
9	2	carrierName	String	送信先のキャリア名です。 上記のキャリアIDに応じたキャリア名が出力されます。 ※statusIdが0,1の場合本項目の値は不定です
10	2	statusId	Integer	ステータスIDです。下記いずれかの値が出力されます。 0: 状態不明 1: 配信中 2: 送達 3: 配信失敗
11	2	clientTag	String	SMS送信のリクエスト時にパラメータとして指定したclientTagです。SMS送信のリクエスト時に指定しなかった場合は空となります。
12	2	detailedStatusId	String	ステータスに関する詳細内部コードです。 ※statusIdが0,1の場合本項目の値は不定です
13	2	separatedSuccessCount	Integer	課金通数です。 ※statusIdが2,3の場合のみ有効値が出力されます ※statusIdが0,1の場合本項目の値は不定です

14	2	rawdata	List	<p>キャリアより得られた詳細エラーデータです。階層 3 の項目を配列で保持します。</p> <p>※statusId が 0, 1 の場合本項目の値は不定です</p> <p>※本項目は将来廃止予定です</p>
15	3	carrier	String	<p>キャリア ID です。以下のいずれかの ID になります。</p> <p>101:AU 102:Y!mobile 103:NTT Docomo 105:Softbank 106:楽天モバイル</p> <p>※statusId が 0, 1 の場合本項目の値は不定です</p> <p>※該当キャリアが判別できない場合、null または 999 となります。</p>
16	3	type	String	<p>詳細エラーデータのタイプを示します。</p> <p>※statusId が 3 の場合に値が出力されます。出力される値は下記のいずれかになります。その他の場合は null となります。</p> <p>1:本サービスからキャリアへの SMS 送信リクエストが失敗した際のレスポンス 2: キャリアから端末への SMS 送信が失敗した際にキャリアから受信した DLR</p> <p>※statusId が 3 以外の場合は null となります。</p>
17	3	data	Object	<p>階層 4 の項目をオブジェクトで保持します。</p> <p>※statusId が 3 の場合に出力されます。</p> <p>※statusId が 3 以外の場合は null となります。</p>
18	4	sepmaxnum	String	<p>Docomo に SMS 送信した際の 70 文字毎の分割数を示します。</p> <p>※statusId が 3 かつ carrier が 103 の場合に出力されます。</p> <p>※statusId が 3 であるが carrier が 103 でない場合は null となります。</p> <p>※statusId が 3 以外の場合は出力されません。</p>
19	4	sepseqnum	String	<p>Docomo への SMS 送信が失敗した際に、送信が失敗した SMS が 70 文字毎に分割された SMS の中で何番目に送信されたかの順番を示します。</p> <p>※statusId が 3 かつ carrier が 103 の場合に出力されます。</p> <p>※statusId が 3 であるが carrier が 103 でない場合は null となります。</p> <p>※statusId が 3 以外の場合は出力されません。</p>
20	4	code	String	<p>Docomo から端末への SMS 送信が失敗した際に Docomo から返されるエラーコードです。</p> <p>2 桁の数字が出力されます。</p> <p>※statusId が 3 かつ carrier が 103 の場合に出力されます。</p> <p>※submission_status が出力されている場合は null となります。</p> <p>※statusId が 3 であるが carrier が 103 でない場合は null となります。</p> <p>※statusId が 3 以外の場合は出力されません。</p>
21	4	submission_status	Integer	<p>本サービスから Docomo への SMS 送信リクエストが失敗した際に Docomo から返されるエラーコードです。</p> <p>5 桁の数字が出力されます。</p> <p>※statusId が 3 かつ carrier が 103 の場合に出力されます。</p> <p>※code が出力されている場合は null となります。</p> <p>※statusId が 3 であるが carrier が 103 でない場合は null となります。</p> <p>※statusId が 3 以外の場合は出力されません。</p>

22	4	msgState	String	Softbank から端末への SMS 送信が失敗した際に Softbank から返される配信結果です。 8 桁までの英数字が出力されます。 ※statusId が 3 かつ carrier が 105 の場合に出力されます。 ※submission_err_code が出力されている場合は null となります。 ※statusId が 3 であるが carrier が 105 でない場合は null となります。 ※statusId が 3 以外の場合は出力されません。
23	4	submission_err_code	Integer	本サービスから Softbank への SMS 送信リクエストが失敗した際に Softbank から返されるエラーコードです。 5 桁の数字が出力されます。 ※statusId が 3 かつ carrier が 105 の場合に出力されます。 ※msgState と ntwkErr が出力されている場合は null となります。 ※statusId が 3 であるが carrier が 105 でない場合は null となります。 ※statusId が 3 以外の場合は出力されません。
24	4	ntwkErr	String	Softbank から端末への SMS 送信が失敗した際に Softbank から返されるエラーコードです。 8 桁までの英数字が出力されます。 ※statusId が 3 かつ carrier が 105 の場合に出力されます。 ※submission_err_code が出力されている場合は null となります。 ※statusId が 3 であるが carrier が 105 でない場合は null となります。 ※statusId が 3 以外の場合は出力されません。
25	4	status	String	常に null となります。
26	4	commandStatus	String	常に null となります。
27	4	networkErrorCode	String	常に null となります。

## レスポンスサンプル

## ➤ 成功時 (0) :

```
{
  "responseCode": 0,
  "responseMessage": "Success.",
  "status": [
    {
      "dateTime": "20170208 175417155",
      "phoneNumber": "+818012345678",
      "message": "test",
      "smsCode": 80079,
      "carrierId": "101",
      "carrierName": "AU",
      "statusId": 2,
      "clientTag": "001180079_20170208175402",
      "detailedStatusId": "2",
      "separatedSuccessCount": 3,
      "rawdata": [
        {
          "carrier": "101",
          "type": null,
          "data": null,
        }
      ]
    }
  ]
}
```

## ➤ 失敗時 (1~) :

```
{"responseCode":110,"responseMessage":"Invalid date.,"status":[]}
```

### 2.4.3. 結果コード

※ API で返却される responseCode と responseMessage を下記に定義します。

responseCode	responseMessage	説明
0	Success.	リクエストが成功しました。
100	Required {項目名}.	{項目名}の項目が不足しています。
110	Invalid {項目名}.	{項目名}の項目が不正です。
120	{項目名} too long.	{項目名}の項目の文字長が上限を超えています。
500	System error.	上記で定義外のエラーが発生しました。

※responseCode:0 が返却されていない場合、リクエストの受け付けには失敗しています。

## 2.5. DLR 通知 (Common)

DLR のプッシュ型通知をお申込みいただいた場合、XML もしくは JSON 形式にて、弊社サーバより SMS 送信ステータスをお客様サーバに対する HTTPS コールを行って通知します (XML/JSON 形式のどちらで DLR 通知を行うかはお申し込み時にお選びいただく形となります)

DLR のプッシュ型通知は、MT 送信のリクエスト時に clientTag を指定することにより DLR 通知を受信できます。clientTag を指定しなかった場合は DLR 通知を受信できません。

### 2.5.1. 通知パラメータ

※ XML 形式の場合、お客様よりご指定いただいた URL に対し、下記の XML を POST メソッドにて通知します。

No	階層	項目名	データ型	概要
1	1	xml	---	xml を記載するパラメータです。
2	1	eventNotifications	---	ルート要素です。
3	2	eventNotification	---	DLR 毎に提供される要素です。 1 通の DLR 通知で 1 つの<eventNotification/>が通知されます。 階層 3 の項目をタグで保持します。
4	3	dateTime	String	本サービスがキャリアより DLR を取得した時間です。 フォーマットは「YYYY/mm/dd HH:MM:SS」となります。
5	3	messageId	String	本サービス内部で生成/使用されるメッセージ ID です。 32 桁の文字列となります。
6	3	phoneNumber	String	送信先電話番号です。国際電話番号形式で通知されます。 ※詳細は『別記事項・国際電話番号形式について』をご参照ください。
7	3	smsCode	Integer	送信元 SMS 用コードです。 5 桁または 6 桁の数字となります。
8	3	carrierId	Integer	キャリア ID です。 101: AU 103: NTT Docomo 105: Softbank 106: 楽天モバイル ※statusId が 0, 1 の場合本項目の値は不定です ※該当キャリアが判別できない場合、null または 999 となります。
9	3	carrierName	String	キャリア名です。 上記のキャリア ID に応じたキャリア名が出力されます。 ※statusId が 0, 1 の場合本項目の値は不定です
10	3	messageText	String	送信先に送信された SMS 本文です。
11	3	clientId	Integer	お客様毎に割り当てられた本サービスのクライアント ID です。
12	3	clientTag	String	SMS 送信時に指定したクライアントタグです。



13	3	statusId	Integer	<p>ステータス ID です。下記いずれかの値が出力されます。</p> <p>0: 状態不明 1: 配信中 2: 送達 3: 配信失敗</p>
14	3	statusDescription	String	<p>ステータス概要です。ステータス ID に応じて下記のステータス概要が出力されます。</p> <p>0: Unknown 1: Message is Sending 2: Message delivered 3: Message delivery failed</p>
15	3	detailedStatusId	String	<p>ステータスに関する詳細内部コードです。</p> <p>※statusId が 0, 1 の場合本項目の値は不定です</p>
16	3	separatedSuccessCount	Integer	<p>課金通数です。</p> <p>※statusId が 2 の場合のみ有効値が出力されます。</p>
17	3	rawdata	---	<p>詳細エラーデータの要素です。</p> <p>階層 4 の要素をタグで保持します。</p> <p>※statusId が 2, 3 の場合に出力されます。</p> <p>※statusId が 0, 1 の場合は出力されません。</p> <p>※本項目は将来廃止予定です</p>
18	4	carrier	Integer	<p>SMS 送信先のキャリア ID です。下記のいずれかになります。</p> <p>101: AU 103: NTT Docomo 105: Softbank 106: 楽天モバイル</p> <p>※該当キャリアが判別できない場合、null または 999 となります。</p>
19	4	type	Integer	<p>詳細エラーデータのタイプを示します。</p> <p>※statusId が 3 の場合に値が出力されます。出力される値は下記のいずれかになります。</p> <p>1: 本サービスからキャリアへの SMS 送信リクエストが失敗した際のレスポンス 2: キャリアから端末への SMS 送信が失敗した際にキャリアから受信した DLR</p> <p>※statusId が 2 の場合は空となります。</p>
20	4	data	---	<p>キャリアから取得したエラーデータの要素です。</p> <p>階層 5 の要素をタグで保持します。</p> <p>※statusId が 2, 3 の場合に出力されます。</p>
21	5	sepmaxnum	Integer	<p>Docomo に SMS 送信した際の 70 文字毎の分割数を示します。</p> <p>※statusId が 3 かつ carrier が 103 の場合に出力されます。</p> <p>※statusId が 3 であるが carrier が 103 でない場合は null となります。</p> <p>※statusId が 3 以外の場合は出力されません。</p>
22	5	sepseqnum	Integer	<p>Docomo への SMS 送信が失敗した際に、送信が失敗した SMS が 70 文字毎に分割された SMS の中で何番目に送信されたかの順番を示します。</p> <p>※statusId が 3 かつ carrier が 103 の場合に出力されます。</p> <p>※statusId が 3 であるが carrier が 103 でない場合は null となります。</p> <p>※statusId が 3 以外の場合は出力されません。</p>

23	5	code	Integer	<p>Docomo から端末への SMS 送信が失敗した際に Docomo から返されるエラーコードです。</p> <p>2 桁の数字が出力されます。</p> <p>※statusId が 3 かつ carrier が 103 の場合に出力されます。</p> <p>※submission_status が出力されている場合は空となります。</p> <p>※statusId が 3 であるが carrier が 103 でない場合は null となります。</p> <p>※statusId が 2 の場合は空となります。</p>
24	5	submission_status	Integer	<p>本サービスから Docomo への SMS 送信リクエストが失敗した際に Docomo から返されるエラーコードです。</p> <p>5 桁の数字が出力されます。</p> <p>※statusId が 3 かつ carrier が 103 の場合に出力されます。</p> <p>※code が出力されている場合は空となります。</p> <p>※statusId が 3 であるが carrier が 103 でない場合は null となります。</p> <p>※statusId が 2 の場合は空となります。</p>
25	5	msgState	String	<p>Softbank から端末への SMS 送信が失敗した際に Softbank から返される配信結果です。</p> <p>8 桁までの英数字が出力されます。</p> <p>※statusId が 3 かつ carrier が 105 の場合に出力されます。</p> <p>※submission_err_code が出力されている場合は null となります。</p> <p>※statusId が 3 であるが carrier が 105 でない場合は null となります。</p> <p>※statusId が 2 の場合は空となります。</p>
26	5	submission_err_code	Integer	<p>本サービスから Softbank への SMS 送信リクエストが失敗した際に Softbank から返されるエラーコードです。</p> <p>5 桁の数字が出力されます。</p> <p>※statusId が 3 かつ carrier が 105 の場合に出力されます。</p> <p>※msgState と ntwkErr が出力されている場合は null となります。</p> <p>※statusId が 3 であるが carrier が 105 でない場合は null となります。</p> <p>※statusId が 2 の場合は空となります。</p>
27	5	ntwkErr	String	<p>Softbank から端末への SMS 送信が失敗した際に Softbank から返されるエラーコードです。</p> <p>8 桁までの英数字が出力されます。</p> <p>※statusId が 3 かつ carrier が 105 の場合に出力されます。</p> <p>※submission_err_code が出力されている場合は null となります。</p> <p>※statusId が 3 であるが carrier が 105 でない場合は null となります。</p> <p>※statusId が 2 の場合は空となります。</p>
28	5	status	String	常に null となります。
29	5	commandStatus	String	常に null となります。
30	5	networkErrorCode	String	常に null となります。

## DLR サンプル

## ➤ 配信成功時 (0) :

```
<?xml version="1.0" encoding="UTF-8"?>
<eventNotifications>
  <eventNotification type="DLR">
    <dateTime><![CDATA[2017/02/03 13:00:45]]></dateTime>
    <messageId><![CDATA[000002517020320030241455L3V3V5d]]></messageId>
    <phoneNumber><![CDATA[+818012345678]]></phoneNumber>
    <smsCode><![CDATA[80051]]></smsCode>
    <carrierId><![CDATA[105]]></carrierId>
    <carrierName><![CDATA[Softbank]]></carrierName>
    <messageText><![CDATA[test 20170203 13:00:34]]></messageText>
    <clientId><![CDATA[25]]></clientId>
    <clientTag><![CDATA[002580051_20170203130034]]></clientTag>
    <statusId><![CDATA[2]]></statusId>
    <statusDescription><![CDATA[Message delivered]]></statusDescription>
    <rawdata>
      <carrier><![CDATA[105]]></carrier>
      <type><![CDATA[]]></type>
      <data>
        <code><![CDATA[]]></code>
        <submission_status><![CDATA[]]></submission_status>
        <msgState><![CDATA[]]></msgState>
        <submission_err_code><![CDATA[]]></submission_err_code>
        <ntwkErr><![CDATA[]]></ntwkErr>
        <commandStatus><![CDATA[]]></commandStatus>
        <networkErrorCode><![CDATA[]]></networkErrorCode>
      </data>
    </rawdata>
    <detailedStatusId><![CDATA[2]]></detailedStatusId>
    <separatedSuccessCount><![CDATA[1]]></separatedSuccessCount>
  </eventNotification>
</eventNotifications>
```

## ➤ 配信失敗時(0) :

```
<?xml version="1.0" encoding="UTF-8"?>
<eventNotifications>
  <eventNotification type="DLR">
    <dateTime><![CDATA[2017/02/03 09:50:29]]></dateTime>
    <messageId><![CDATA[000000317020320018132331wkyYHLX]]></messageId>
    <phoneNumber><![CDATA[+818012345678]]></phoneNumber>
    <smsCode><![CDATA[80022]]></smsCode>
    <carrierId><![CDATA[105]]></carrierId>
    <carrierName><![CDATA[Softbank]]></carrierName>
    <messageText><![CDATA[test message]]></messageText>
    <clientId><![CDATA[3]]></clientId>
    <clientTag><![CDATA[000380022_20170203_0938]]></clientTag>
    <statusId><![CDATA[3]]></statusId>
    <statusDescription><![CDATA[Message delivery failed]]></statusDescription>
    <rawdata>
      <carrier><![CDATA[105]]></carrier>
      <type><![CDATA[2]]></type>
      <data>
        <code><![CDATA[null]]></code>
        <submission_status><![CDATA[null]]></submission_status>
        <msgState><![CDATA[EXPIRED]]></msgState>
        <submission_err_code><![CDATA[null]]></submission_err_code>
        <ntwkErr><![CDATA[G020600]]></ntwkErr>
        <commandStatus><![CDATA[null]]></commandStatus>
        <networkErrorCode><![CDATA[null]]></networkErrorCode>
      </data>
    </rawdata>
    <detailedStatusId><![CDATA[12832]]></detailedStatusId>
    <separatedSuccessCount><![CDATA[0]]></ separatedSuccessCount>
  </eventNotification>
</eventNotifications>
```

※ JSON 形式の場合、お客様よりご指定いただいた URL に対し、下記の JSON を POST メソッドにて通知します。

No	階層	項目名	データ型	概要
1	1	json	---	json を記載するパラメータです。
2	1	eventNotifications	List	階層 2 の項目を配列で保持します。
3	2	type	string	固定値「DLR」となります。
4	2	dateTime	String	本サービスがキャリアより DLR を取得した時間です。 フォーマットは「YYYY/mm/dd HH:MM:SS」となります。
5	2	messageId	String	本サービス内部で生成/使用されるメッセージ ID です。 32 桁の文字列となります。
6	2	phoneNumber	String	送信先電話番号です。国際電話番号形式で通知されます。 ※詳細は『別記事項・国際電話番号形式について』をご参照ください。
7	2	smsCode	String	送信元 SMS 用コードです。 5 桁または 6 桁の数字となります。
8	2	carrierId	String	キャリア ID です。 101: AU 103: NTT Docomo 105: Softbank 106: 楽天モバイル ※statusId が 0, 1 の場合本項目の値は不定です ※該当キャリアが判別できない場合、null または 999 となります。
9	2	carrierName	String	キャリア名です。 上記のキャリア ID に応じたキャリア名が出力されます。 ※statusId が 0, 1 の場合本項目の値は不定です
10	2	messageText	String	送信先に送信された SMS 本文です。
11	2	clientId	String	お客様毎に割り当てられた本サービスのクライアント ID です。
12	2	clientTag	String	SMS 送信時に指定したクライアントタグです。
13	2	statusId	String	ステータス ID です。下記いずれかの値が出力されます。 0: 状態不明 1: 配信中 2: 送達 3: 配信失敗
14	2	statusDescription	String	ステータス概要です。ステータス ID に応じて下記のステータス概要が出力されます。 0: Unknown 1: Message is Sending 2: Message delivered 3: Message delivery failed
15	2	detailedStatusId	Integer	ステータスに関する詳細内部コードです。 ※statusId が 0, 1 の場合本項目の値は不定です

16	2	separatedSuccessCount	Integer	課金通数です。 ※statusId が 2 の場合のみ有効値が出力されます。
17	2	rawdata	List	詳細エラーデータです。 階層 3 の項目を配列で保持します。 ※statusId が 2, 3 の場合に出力されます。 ※statusId が 0, 1 の場合は null となります。 ※本項目は将来廃止予定です
18	3	carrier	String	SMS 送信先のキャリア ID です。下記のいずれかになります。 101: AU 103: NTT Docomo 105: Softbank 106: 楽天モバイル ※該当キャリアが判別できない場合、null または 999 となります。
19	3	type	String	詳細エラーデータのタイプを示します。 ※statusId が 3 の場合に値が出力されます。出力される値は下記のいずれかになります。 1: 本サービスからキャリアへの SMS 送信リクエストが失敗した際のレスポンス 2: キャリアから端末への SMS 送信が失敗した際にキャリアから受信した DLR ※statusId が 2 の場合は空となります。
20	3	data	List	キャリアから取得したエラーデータです。 階層 4 の項目を配列で保持します。 ※statusId が 2, 3 の場合に出力されます。
21	4	sepmaxnum	String	Docomo に SMS 送信した際の 70 文字毎の分割数を示します。 ※statusId が 3 かつ carrier が 103 の場合に出力されます。 ※statusId が 3 であるが carrier が 103 でない場合は null となります。 ※statusId が 3 以外の場合は出力されません。
22	4	sepseqnum	String	Docomo への SMS 送信が失敗した際に、送信が失敗した SMS が 70 文字毎に分割された SMS の中で何番目に送信されたかの順番を示します。 ※statusId が 3 かつ carrier が 103 の場合に出力されます。 ※statusId が 3 であるが carrier が 103 でない場合は null となります。 ※statusId が 3 以外の場合は出力されません。
23	4	code	String	Docomo から端末への SMS 送信が失敗した際に Docomo から返されるエラーコードです。 2 桁の数字が出力されます。 ※statusId が 3 かつ carrier が 103 の場合に出力されます。 ※submission_status が出力されている場合は空となります。 ※statusId が 3 であるが carrier が 103 でない場合は null となります。 ※statusId が 2 の場合は空となります。
24	4	submission_status	String	本サービスから Docomo への SMS 送信リクエストが失敗した際に Docomo から返されるエラーコードです。 5 桁の数字が出力されます。 ※statusId が 3 かつ carrier が 103 の場合に出力されます。 ※code が出力されている場合は空となります。 ※statusId が 3 であるが carrier が 103 でない場合は null となります。 ※statusId が 2 の場合は空となります。

25	4	msgState	String	Softbank から端末への SMS 送信が失敗した際に Softbank から返される配信結果です。 8 桁までの英数字が出力されます。 ※statusId が 3 かつ carrier が 105 の場合に出力されます。 ※submission_err_code が出力されている場合は null となります。 ※statusId が 3 であるが carrier が 105 でない場合は null となります。 ※statusId が 2 の場合は空となります。
26	4	submission_err_code	String	本サービスから Softbank への SMS 送信リクエストが失敗した際に Softbank から返されるエラーコードです。 5 桁の数字が出力されます。 ※statusId が 3 かつ carrier が 105 の場合に出力されます。 ※msgState と ntwkErr が出力されている場合は null となります。 ※statusId が 3 であるが carrier が 105 でない場合は null となります。 ※statusId が 2 の場合は空となります。
27	4	ntwkErr	String	Softbank から端末への SMS 送信が失敗した際に Softbank から返されるエラーコードです。 8 桁までの英数字が出力されます。 ※statusId が 3 かつ carrier が 105 の場合に出力されます。 ※submission_err_code が出力されている場合は null となります。 ※statusId が 3 であるが carrier が 105 でない場合は null となります。 ※statusId が 2 の場合は空となります。
28	4	status	String	常に null となります。
29	4	commandStatus	String	常に null となります。
30	4	networkErrorCode	String	常に null となります。

## DLR サンプル

## ➤ 配信成功時(0) :

```
{
  "eventNotifications": [
    {
      "type": "DLR",
      "dateTime": "2017/02/24 21:45:47",
      "messageId": "00002891702242187613918627udCsc",
      "phoneNumber": "+819012345678",
      "smsCode": "80366",
      "carrierId": "103",
      "carrierName": "NTT Docomo",
      "messageText": "test message",
      "clientId": "289",
      "clientTag": "CID0289_20170224_2145",
      "statusId": "2",
      "statusDescription": "Message delivered",
      "rawdata": {
        "carrier": "103",
        "type": "",
        "data": {
          "code": "",
          "submission_status": "",
          "msgState": "",
          "submission_err_code": "",
          "ntwkErr": "",
          "commandStatus": "",
          "networkErrorCode": ""
        }
      }
    },
    "detailedStatusId": 2,
    "separatedSuccessCount": 1
  ]
}
```



## ➤ 配信失敗時(0) :

```
{
  "eventNotifications": [
    {
      "type": "DLR",
      "dateTime": "2017/02/24 21:46:57",
      "messageId": "0000289170224218762151655Xf1Hve",
      "phoneNumber": "+819012345678",
      "smsCode": "80366",
      "carrierId": "103",
      "carrierName": "NTT Docomo",
      "messageText": "test message",
      "clientId": "289",
      "clientTag": "CID0289_20170224_2146",
      "statusId": "3",
      "statusDescription": "Message delivery failed",
      "rawdata": {
        "carrier": "103",
        "type": "2",
        "data": {
          "code": "21",
          "submission_status": null,
          "msgState": null,
          "submission_err_code": null,
          "ntwkErr": null,
          "commandStatus": null,
          "networkErrorCode": null
        }
      },
      "detailedStatusId": 3,
      "separatedSuccessCount": 0
    }
  ]
}
```

## 2.5.2. レスポンス

※ API 呼び出し結果としてお客様サーバより HTTP ステータスコードを返却してください。

HTTP ステータスコード	成功／失敗	備考
2xx	成功	
2xx 以外	失敗	

HTTP ステータスコード 2xx を 10 秒以内に返却した場合に成功とみなします。

200 以外のステータスコードを返却した場合、またはタイムアウト（10 秒）が発生した場合は受信に失敗したと判断され、3 回のリトライ処理が行われます。

## 2.6. ショート URL 登録

API によりショート URL を登録することが可能です。

### 2.6.1. リクエスト

#### 接続先 URL

`https://sms-api.aossm.com/p1/api/shortenurl.json`

※ 以下のパラメータを URL エンコードにて追加し、POST メソッドにて呼び出します。

No	項目名	データ型	必須	概要
1	token	String	○	アカウント登録時に発行されるアクセスキーを指定します。トークンの値につきましては、管理画面のアカウント情報をご確認ください。 ・半角英数字
2	clientId	String	○	契約クライアント ID を指定します。 割り当てられている契約クライアント ID につきましては、管理画面のクライアント情報をご確認ください。 ・半角数字
3	name	String		生成されるショート URL の識別名を指定します。 ・最大 256 文字 指定しない場合、自動的に下記が name として登録されます。 Shorturl-YYYYMMDDHHMSSsss (sss はミリ秒)
4	longUrl	String	○	ショート URL に変換するオリジナル URL を指定します。 ・最大 2048 文字
5	domain	String		ショート URL に変換する際のドメインを指定します。 ※独自ドメインオプションをお申し込みいただいているお客様のみご指定可能です

## 2.6.2. レスポンス

※ API 呼び出し結果として応答データ (JSON 形式) を返却します。

### レスポンスパラメータ

No	階層	項目名	データ型	概要
1	1	responseCode	Integer	応答コードです。 正常時は0を返します。1以上の場合はエラーになります。 ※結果コードの項を参照してください。
2	1	responseMessage	String	送信処理結果を示すメッセージです。 ※結果コードの項を参照してください。
3	1	shortUrl	String	生成されたショート URL です。
4	1	name	String	リクエスト時に指定した生成されたショート URL の識別名です。

### レスポンスサンプル

➤ 成功時 (0) :

```
{"responseCode":0,"responseMessage":"Success.,"shortUrl":"https://ai9.jp/iBJL10","name":"shortUrlSample"}
```

➤ 失敗時 (1~) :

```
{"responseCode":110,"responseMessage":"Invalid longUrl.,"shortUrl":null,"name":null}
```

### 2.6.3. 結果コード

※ API で返却される responseCode と responseMessage を下記に定義します。

responseCode	responseMessage	説明
0	Success.	リクエストが成功しました。
100	Required {項目名}.	{項目名}の項目が不足しています。
110	Invalid {項目名}.	{項目名}の項目が不正です。
120	{項目名} too long.	{項目名}の項目の文字長が上限を超えています。
500	System error.	上記で定義外のエラーが発生しました。

※responseCode:0 が返却されていない場合、リクエストの受け付けには失敗しています。

## 2.7. 登録済み定型文取得

APIにより、管理画面で登録した定型文の一覧を取得することが可能です。

### 2.7.1. リクエスト

接続先 URL

<https://sms-api.aossm.com/p5/api/template.json>

※ 以下のパラメータを URL エンコードにて追加し、GET メソッドにて呼び出します。

No	項目名	データ型	必須	概要
1	token	String	○	アカウント登録時に発行されるアクセスキーを指定します。トークンの値につきましては、管理画面のアカウント情報をご確認ください。 ・半角英数字
2	clientId	String	○	契約クライアント ID を指定します。 割り当てられている契約クライアント ID につきましては、管理画面のクライアント情報をご確認ください。 ・半角数字

## 2.7.2. レスポンス

※ API 呼び出し結果として応答データ (JSON 形式) を返却します。

### レスポンスパラメータ

No	階層	項目名	データ型	概要
1	1	responseCode	String	応答コードです。 正常時は0を返します。1以上の場合はエラーになります。 ※結果コードの項を参照してください。
2	1	responseMessage	String	送信処理結果を示すメッセージです。 ※結果コードの項を参照してください。
3	1	count	Integer	登録済の定型文の件数です。
4	1	templates	List	階層2の項目を配列にて保持します。
5	2	title	String	定型文のタイトルです。
6	2	default_content	String	全キャリア共通の定型文メッセージです。
7	2	parameters	List	階層3の項目を配列にて保持します。
8	3	parameter_name	String	定型文に埋め込まれているパラメータです。本項目は定型文に埋め込まれたパラメータの件数分だけ繰り返されます。

## レスポンスサンプル

## ➤ 成功時 (0) :

```
{
  "responseCode": 0,
  "responseMessage": "Success.",
  "count": 2,
  "templates": [
    {
      "title": "demo_sample_01",
      "default_content": "demo_sample_01",
      "parameters": []
    },
    {
      "title": "demo_sample_02",
      "default_content": "{{demo%}}_{{sample%}}_02",
      "parameters": [
        {
          "parameter_name": "demo"
        },
        {
          "parameter_name": "sample"
        }
      ]
    }
  ]
}
```

## ➤ 失敗時 (1~) :

```
{"responseCode":100,"responseMessage":"Required
clientId.","count":0,"templates":null}
```



### 2.7.3. 結果コード

※ API で返却される responseCode と responseMessage を下記に定義します。

responseCode	responseMessage	説明
0	Success.	リクエストが成功しました。
100	Required {項目名}.	{項目名}の項目が不足しています。
110	Invalid {項目名}.	{項目名}の項目が不正です。
500	System error.	上記で定義外のエラーが発生しました。

※responseCode:0 が返却されていない場合、リクエストの受け付けには失敗しています。

## 2.8. 電話番号クリーニング

APIによって電話番号クリーニングを行うことができます。

※ この機能を利用するためにはオプション契約が必要となります。

### 2.8.1. リクエスト

#### 接続先 URL

<https://sms-api.aossm.com/p5/api/numbercleaning.json>

※ 以下のパラメータを URL エンコードにて追加し、GET メソッドにて呼び出します。

No	項目名	データ型	必須	概要
1	token	String	○	アカウント登録時に発行されるアクセスキーを指定します。 トークンの値につきましては、管理画面のアカウント情報をご確認ください。 ・半角英数字
2	clientId	String	○	契約クライアント ID を指定します。 割り当てられている契約クライアント ID につきましては、 管理画面のクライアント情報をご確認ください。 ・半角数字
3	phoneNumber	String	○	判定対象電話番号を指定します。 日本国内形式 (070xxxxxxx/080xxxxxxx/090xxxxxxx)、国際電話番号形式の どちらも指定することが可能です。
4	referenceDate	String	○	判定を行う基準となる日を指定します。 ・2年前～当日まで有効 ・半角数字 ・YYYYMMDD 形式

## 2.8.2. レスポンス

※ API 呼び出し結果として応答データ (JSON 形式) を返却します。

### レスポンスパラメータ

No	階層	項目名	データ型	概要
1	1	responseCode	Integer	応答コードです。 正常時は0を返します。1以上の場合はエラーになります。 ※結果コードの項を参照してください。
2	1	responseMessage	String	送信処理結果を示すメッセージです。 ※結果コードの項を参照してください。
3	1	phoneNumber	String	リクエスト時に指定した送信先電話番号です。 国際電話番号形式で返却されます。 ※国際電話番号形式については『別記事項・国際電話番号形式について』をご参照ください。
4	1	referenceDate	String	リクエスト時に指定した判定を行う基準日です。 ・半角数字 ・YYYYMMDD 形式
5	1	status	String	名義人変更判定結果の状態です。 T : 他人名義と判定された N : 本人 ERROR : 判定 DB になく判定不可

### レスポンスサンプル

➤ 成功時 (0) :

```
{
  "responseCode": 0,
  "responseMessage": "Success.",
  "phoneNumber": "+819012345678",
  "referenceDate": "20170531",
  "status": "N"
}
```

➤ 失敗時 (1~) :

```
{
  "responseCode": 1,
  "responseMessage": "Required clientId.",
  "phoneNumber": null,
  "referenceDate": null,
  "status": null
}
```

### 2.8.3. 結果コード

※ API で返却される responseCode と responseMessage を下記に定義します。

responseCode	responseMessage	説明
0	Success.	リクエストが成功しました。
100	Required {項目名}.	{項目名}の項目が不足しています。
110	Invalid {項目名}.	{項目名}の項目が不正です。
500	System error.	上記で定義外のエラーが発生しました。

※responseCode:0 が返却されていない場合、リクエストの受け付けには失敗しています。

## 2.9. 通数集計

APIによってキャリア別の通数を取得することができます。

### 2.9.1. リクエスト

#### 接続先 URL

<https://sms-api.aossm.com/p5/api/separatedsuccesscount.json>

※ 以下のパラメータを URL エンコードにて追加し、GET メソッドにて呼び出します。

No	項目名	データ型	必須	概要
1	token	String	○	アカウント登録時に発行されるアクセスキーを指定します。 トークンの値につきましては、管理画面のアカウント情報をご確認ください。 ・半角英数字
2	clientId	String	○	契約クライアント ID または all を指定します。 ※all は代理店参照者のみ使用可能です。 割り当てられている契約クライアント ID につきましては、 管理画面のクライアント情報をご確認ください。 ・半角数字 ・all (代理店参照者のみ)
3	month	String	○	取得する年月を指定します。取得できる期間はデータ保持期間の間のみです。 ・半角数字 ・YYYYMM 形式 ※当月を指定する場合、取得できる件数は前日までのものとなります。

## 2.9.2. レスポンス

※ API 呼び出し結果として応答データ (JSON 形式) を返却します。

### レスポンスパラメータ

No	階層	項目名	データ型	概要
1	1	responseCode	Integer	応答コードです。 正常時は 0 を返します。1 以上の場合はエラーになります。 ※結果コードの項を参照してください。
2	1	responseMessage	String	送信処理結果を示すメッセージです。 ※結果コードの項を参照してください。
3	1	separatedSuccessCount	List	階層 2 の項目を配列で保持します。
4	2	clientId	Integer	お客様毎に割り当てられた本サービスのクライアント ID です。 ※0 通の場合は表示されません。
5	2	101	Integer	指定年月の au 通数です。 ※0 通の場合は表示されません。
6	2	103	Integer	指定年月のドコモ通数です。 ※0 通の場合は表示されません。
7	2	105	Integer	指定年月のソフトバンク通数です。 ※0 通の場合は表示されません。
8	2	106	Integer	指定年月の楽天モバイル通数です。 ※0 通の場合は表示されません。

※取得した日時によっては請求書と差異が発生する可能性があります。

差異が発生した場合は請求書を正とさせていただきます。

## レスポンスサンプル

## ➤ 成功時 (0) :

```
{
  "responseCode": 0,
  "responseMessage": "Success.",
  "separatedSuccessCount": [
    {
      "clientId": 12345,
      "101": 1274094,
      "105": 1163321,
      "103": 1608969,
      "106": 258990
    }
  ]
}
```

## ➤ 失敗時 (1~) :

```
{
  "responseCode": 100,
  "responseMessage": "Required clientId.",
  "separatedSuccessCount": null
}
```

### 2.9.3. 結果コード

※ API で返却される responseCode と responseMessage を下記に定義します。

responseCode	responseMessage	説明
0	Success.	リクエストが成功しました。
100	Required {項目名}.	{項目名}の項目が不足しています。
110	Invalid {項目名}.	{項目名}の項目が不正です。
500	System error.	上記で定義外のエラーが発生しました。

※responseCode:0 が返却されていない場合、リクエストの受け付けには失敗しています。

### 3. 別記事項

#### 3.1. 国際電話番号形式について

本資料上のリクエストパラメータで『国際電話番号』形式としてある場合、以下に記載する国際電話番号形式に則った電話番号の指定が必要になります。

- ・ 先頭が半角記号+より始まる (注意: GET/POST 送信時は URL エンコードが必要)
  - ・ 国番号 81
  - ・ 国番号の次の 1 文字は 7, 8, 9 のいずれか
  - ・ 以降、任意の半角数字 8 桁
- xml タグ指定例)  
+818010007765
  - URL エンコード指定例)  
%2b818010007765 (%2b が半角プラス+を URL エンコードした文字列になります)

#### 3.2. ステータス ID について

本サービスのステータス ID 設定値とメッセージは以下の通りです。

設定値	メッセージ	備考
0	Unknown	状態不明
1	Message is Sending	配信中
2	Message delivered	送達
3	Message delivery failed	配信失敗

#### 3.3. レスポンスの rawdata について

各レスポンスの rawdata は廃止予定のため参照は推奨いたしません。



### 3.4. 詳細内部コード (detailedStatusId) について

本サービスの詳細内部コードは配信失敗の場合、「モジュールコード(先頭2桁) + 分類コード(末尾3桁)」の5桁で構成されます。詳細内部コード設定値の主要な値は以下の通りです。

- ※ 詳細内部コードは将来予告なく追加・変更される可能性があります。そのためプログラム実装では厳密に参照することは避け例外を許容することをお勧めいたします。

モジュールコード (先頭2桁)	該当キャリア
11~14	AU
31, 32	Docomo
51, 52	Softbank
61, 62	楽天モバイル

分類コード (末尾3桁)	内容	起因箇所
721	輻輳	キャリア、送信先端末
732	欠番、顧客都合	送信先端末
821	輻輳	送信先端末
823	ネットワーク起因 (圏外、電源 OFF 含む)	キャリア、送信先端末
824	圏外、電源 OFF 等	送信先端末
833	圏外、話中、電源 OFF 等	送信先端末
834	移動機側エラー (SMS 拒否設定等)	送信先端末

### 3.5. Y!Mobile (旧 Willcom、PHS 回線) について

Y!Mobile (旧 Willcom、PHS 回線) は 2021 年 1 月 31 日 (日) にサービス終了しました。

以上

CONFIDENTIAL

本文書は、AI CROSS 株式会社が著作権その他の権利を有する営業秘密です。当社の許可なく複製し利用すること、また漏洩することは「著作権法」「不正競争防止法」によって禁じられております。