MARIGOLD™

API Technical Guide: SMS Campaign

Cheetah Messaging

Table of Contents

© 2025 Marigold. All rights reserved.

© 2025 Marigold. All rights reserved.

© 2025 Marigold. All rights reserved.

© 2025 Marigold. All rights reserved.

© 2025 Marigold. All rights reserved.

1 Introduction

Purpose

The purpose of this document is to provide an overview of the SMS CAMPAIGN API endpoint within the Cheetah Messaging platform. This document discusses the intended use of the SMS CAMPAIGN endpoint, and provides technical details for how to implement the endpoint.

Overview

The SMS CAMPAIGN endpoint is used to manage your SMS Text Campaigns in Messaging, by allowing you to create new Campaigns, and to edit, view, and delete your existing Campaigns.

This endpoint requires authentication using OAuth 2.0, and supports JSON and XML messages.

The URLs for this endpoint are:

- **North America:** https://api.eccmp.com/services2/api/SmsCampaign
- **Europe:** https://api.ccmp.eu/services2/api/SmsCampaign
- **Japan**: https://api.marketingsuite.jp/services2/api/SmsCampaign

Prerequisites

When creating a new SMS Text Campaign, the SMS CAMPAIGN endpoint requires that all the supporting assets needed for the Campaign must already be created and saved within the platform. These assets can include, for example, a Filter, a Seed List, a Content Block,

© 2025 Marigold. All rights reserved.

a Keyword Group, and an Exclusion List. In most cases, you must reference the existing asset by means of an identifier, in order to use that asset in the Campaign.

Unsupported Features

The SMS CAMPAIGN endpoint doesn't support all of the Campaign features that are available from within the Messaging application user interface. The following SMS Text Campaign features aren't supported by the SMS CAMPAIGN endpoint:

- Split Cells
- Custom Responses
- Personalized URLs
- Deriving the Message Creation Start date / time by calculating backward from the Send Start date / time
- Creating or editing an Event-triggered Campaign using any trigger type other than "Advanced Event Trigger" or "File Import"
- Multimedia Messaging Service (MMS) content in the message

Methods

The SMS CAMPAIGN endpoint supports the following HTTP methods:

- **POST**: Create a new SMS Text Campaign.
- **GET**: Retrieve information about a specified SMS Text Campaign.
- **PUT**: Submit modifications to an existing SMS Text Campaign.
- **DELETE**: Delete a specified SMS Text Campaign.

© 2025 Marigold. All rights reserved.

Authentication

Access to the SMS CAMPAIGN endpoint requires that you first be authenticated within the platform. Within Messaging, authentication is handled by OAuth 2.0. To authenticate with OAuth 2.0, you must first obtain a "Consumer Key" and a "Consumer Secret." Both of these values are managed at the user level, and can be obtained from within the Messaging application.

Next, you'll use your Consumer Key and Consumer Secret to request a "token." A token is a text string that, when provided in a request message, will allow the user access to the requested service. Tokens are valid only for a certain period of time.

For more details on how to authenticate your API request, please see the *Messaging: API How-to Guide*.

© 2025 Marigold. All rights reserved.

2 Create an SMS Text Campaign

Overview

This section describes how to create a new SMS Text Campaign via a POST request to the SMS CAMPAIGN endpoint.

The SMS CAMPAIGN POST request message contains a large number of different parameters, options, and identifiers. This document presents all these parameters in a functional manner, that mimics the way the options are presented to a user in the Messaging front-end interface:

- **Campaign Setup** -- Define the Campaign type, name, location, Metadata, etc.
- **Audience** -- Define / restrict the audience of intended recipients.
- **Message** -- Define the Campaign message content.
- **Sending** -- Define the Campaign schedule.
- **Responses** -- Configure expected responses to the Campaign, such as keyword responses.
- **Proofing** -- Configure proofing options.
- **Auditing** -- Configure pre-launch auditing options.

Parameters -- Campaign Setup

The parameters in this section explain the high-level options for creating and setting up an SMS Text Campaign, such as the Campaign type, name, location, and Metadata values.

© 2025 Marigold. All rights reserved.

custId

This integer parameter is required.

The **custId** parameter represents the Customer ID of your Messaging account. The Customer ID is a unique, system-generated identifier for every Messaging client account. This value isn't displayed anywhere within the Messaging application, so you must retrieve it by means of an API request (several different endpoints will return the Customer ID as part of the response message), or speak to your Client Services Representative, who can provide you with this value.

Example:

```
"custId": 446
```

entityId

This integer parameter is required.

The **entityId** parameter represents the **Entity ID** of the Campaign's source table.

Example:

```
"entityId": 100
```

typeId

This string parameter is required.

The **typeId** parameter indicates the type of Campaign. The valid values for this parameter are:

- "REGULAR" -- Regular One-Off Campaign
- "CALCULATED" -- Date-triggered Campaign
- "TRIGGERED" -- Event-triggered Campaign

Example:

```
"typeId": "REGULAR"
```


© 2025 Marigold. All rights reserved.

campTriggers

A trigger is a specific event or mechanism that causes Messaging to create and deploy a message within an Event-triggered Campaign. The **campTriggers** object is used to define the specific trigger mechanism for this Campaign.

Example:

```
"campTriggers": [
                                                                {
"typeId": "ADVANCED_EVENT_TRIGGER",
"triggerParams": []
                                                                }
]
```

The parameters in this object are described below in more detail.

typeId

This string parameter is optional.

The **typeId** parameter specifies the type of trigger mechanism. The only valid, supported values for this parameter when sending a POST message to the SMS CAMPAIGN endpoint are:

- "ADVANCED_EVENT_TRIGGER"
- "FILE_IMPORT"

If you need to create an Event-triggered SMS Campaign using some other trigger mechanism, you'll need to create the Campaign using the Messaging application.

triggerParams

This object is used to specify additional configuration options about the trigger event.

An "ADVANCED_EVENT_TRIGGER" trigger type doesn't have any additional configuration options, but the **triggerParams** object must still be included in the message. In this case, simply provide a blank **triggerParams** object.

Example:

```
"campTriggers": [
                                                                {
```


© 2025 Marigold. All rights reserved.

```
"typeId": "FILE_IMPORT",
"triggerParams": [
                                                                    {
"paramId": 1,
"paramName": "folder_id",
"integerVal": 11485
                                                                    }
                                                                     ]
                                                                    }
]
```

The parameters in this object are described below in more detail.

paramId

This integer parameter is required.

The **paramId** parameter is a sequential counter used when you're defining multiple trigger types.

paramName

This string parameter is required.

The **paramName** parameter is used to specify additional configuration details about the trigger. This parameter instructs the platform what information to expect in the **integerVal** parameter. When the trigger type is "FILE_IMPORT," the valid values for **paramName** are:

- "import_id" -- Used to specify an import file
- "folder_id" -- Used to specify a folder

integerVal

This integer parameter is required.

© 2025 Marigold. All rights reserved.

The **integerVal** is used to specify additional details about the trigger, based on the **paramName** value provided above.

- If **paramName** is "import_id," then you must provide the **Object Reference ID** for a specific Import file.
- If **paramName** is "folder_id," then you must provide the **Folder ID** for a folder where you will upload the Import file.

senderProfileId

This integer parameter is required.

The **senderProfileId** parameter represents the **Object Reference ID** of your SMS Text Sender Profile. A Sender Profile controls the Short Code from which your message is sent, as well as the recipient's eligibility to be contacted.

Example:

```
"senderProfileId": 7
```

obj

This object contains the name and location of the new Campaign.

Example:

```
"obj":
  {
    "display_name": "Test SMS Campaign API",
    "parent_obj_id": 37249
  }
```

The parameters in this object are described below in more detail.

display_name

This string parameter is required.

The **display_name** parameter contains the name of the Campaign. This name must be unique within the selected folder location.

parent_obj_id

This integer parameter is optional.

© 2025 Marigold. All rights reserved.

The **parent_obj_id** parameter represents the **Folder ID** of the folder where you want to save the new Campaign. If you don't provide this parameter, the system will save the Campaign in the default folder location for your account, which is typically the top-most Folder in your folder structure.

campMetaParams

This object is used to assign Metadata values to the Campaign. These Metadata fields can be used to group Campaigns together, for use in Filters and reports.

Metadata fields can either be a free-form text entry field, or they can optionally contain a set of valid values from which to pick.

Example:

```
"campMetaParams": [
  {
    "optionId": 4,
    "stringVal": "Test Metadata Value"
  },
  {
    "optionId": 10,
    "selectionId": 3
  },
  {
    "optionId": 50,
    "integerVal": 25
  },
  {
    "optionId": 54,
    "datetimeVal": "2018-03-15T00:00:00"
  }
]
```

The parameters in this object are described below in more detail.

optionId

This integer parameter is optional.

The **optionId** references the ID of the desired Metadata field. This ID is not displayed within the user interface, and you can't search for it via an API endpoint. Please speak to your Client Services Representative who can provide this value for you.

© 2025 Marigold. All rights reserved.

selectionId

This integer parameter is optional.

The **selectionId** parameter is used for Metadata fields with pre-defined values; this ID references a specific pre-defined value. This ID is not displayed within the user interface, and you can't search for it via an API endpoint. Please speak to your Client Services Representative who can provide this value for you.

integerVal

This integer parameter is optional.

The **integerVal** parameter is used to provide an integer value for Metadata fields that accept Integer type data.

stringVal

This string parameter is optional.

The **stringVal** parameter is used to provide a string value for Metadata fields that accept String type data.

datetimeVal

This string parameter is optional.

The **datetimeVal** parameter is used to provide a date value for Metadata fields that accept Date type data.

Date values should be provided in the format: "YYYY-MM-YYThh:mm:ss."

Parameters -- Audience

The term "Audience" refers to the universe of recipients that you're targeting with your Campaign.

For Regular One-off Campaigns and Date-triggered Campaigns, the Audience is defined using a Filter.

© 2025 Marigold. All rights reserved.

For an Event-triggered Campaign, you're not required to select an Audience Filter, as the event itself defines who the recipients are; the default Audience for an Event-triggered Campaign is "all triggered records." Optionally, however, you can select a Filter if you need to apply additional restrictions to identify and select only a sub-set of the triggered records.

For all Campaign types, the Audience can optionally be restricted through the use of other assets, such as Exclusions Lists, De-duping Logic, etc.

The options and parameters described in this section explain how to specify a Filter, and apply other optional Audience-related assets and restrictions.

toFilterId

This integer parameter is optional.

The **toFilterId** parameter represents the **Object Reference ID** of the Campaign's main audience Filter. This Filter is used to identify and select the intended recipients of the SMS Text Campaign.

Please note that you can't change the Filter in a Regular One-off Campaign, once the Campaign is launched.

Example:

```
"toFilterId": 29094
```

ccFilterId

This integer parameter is optional.

The **ccFilterId** parameter is used to specify a Seed List. A Seed List is a group of one or more individuals who are designated to receive a copy of a Campaign message.

Seed Lists can be built in one of two ways: either by manually entering one or more individuals into the Seed List, or by using a Filter to define the logic of who should be included in the Seed List.

If using a Filter to define the Seed List, you must provide that Filter's **Object Reference ID** in the **ccFilterId** parameter.

© 2025 Marigold. All rights reserved.

If you're using a manually-defined Seed List, you must provide the **Object Reference ID** for that Seed List in the **ccFilterId** parameter

Example:

```
"ccFilterId": 40966
```

campToList

This object is used to add additional assets to your Campaign, such as an Alert Group.

> **Note** **campToList** object is also used to add a Proofing Group to your Campaign. The parameters related to Proofing are described later in this document in the **Parameters -- Proofing** section.

Example:

```
"CampToList":
  {
    "alertListId": 2803
  }
```

The parameters in this object are described below in more detail.

alertListId

This integer parameter is optional.

The **alertListId** parameter is used to specify an Alert Group. The primary purpose of an Alert Group is to notify a select group of individuals when some triggering event has occurred.

To assign an Alert Group, you must provide the **Object Reference ID** for the desired Alert Group.

campToPropLists

This object is used to add Exclusion Lists to the Campaign. An Exclusion List consists of individuals who should not be targeted in your Campaign, such as individuals who work for competitors, for example. The Exclusion List automatically overrides all other criteria,

© 2025 Marigold. All rights reserved.

including Filters, Seed Lists, and Proofing Groups. A Campaign can optionally include multiple Exclusion Lists.

Example:

```
"campToPropLists": [
    {
      "listId": 1050,
      "excludeFlag": 1
    },
    {
      "listId": 1078,
      "excludeFlag": 1
    }
],
```

The parameters in this object are described below.

listId

This integer parameter is optional.

The **listId** parameter contains the **Object Reference ID** for the desired Exclusion List.

excludeFlag

This integer parameter is optional.

The value in the **excludeFlag** parameter should be "1."

campParam

The **campParam** object contains a wide variety of different parameters. This section describes the parameters in this object that are related to Audience selection.

Example:

```
"campParam":
  {
    "ignoreConfirmationFlag": 1
  }
```

These parameters are described below in more detail.

ignoreConfirmationFlag

This integer parameter is optional.

© 2025 Marigold. All rights reserved.

In order to comply with certain regional marketing regulations, a Sender Profile can optionally be configured to enforce a "double opt-in" method for validating consumer eligibility to be contacted. If using this method, the consumer must register to receive your messages, AND confirm the registration. Only consumers who have completed both steps in this process (i.e., the "double opt-in") will be considered eligible to receive messages.

> **Note**
>
> Enforcement of the double opt-in process must be enabled for a Sender Profile. For more information on enabling this feature, please speak to your Client Services Representative.

If using the double opt-in process, you typically need to send a "Confirmation request" text message to a consumer who has submitted his or her initial registration. However, the consumer's status at this point in the process is still "Not confirmed" because he or she has not yet confirmed the registration. In order to deploy text messages to an "unconfirmed" consumer, the platform offers an override feature.

To enable this override for a Campaign, provide a value of "1" in the **ignoreConfirmationFlag** parameter.

trackingOffFlag

This integer parameter is optional.

When the trackingOffFlag is enabled, the campaign will be sent without any tracking process, such as no link tracking, spy pixel, etc. For cells and splits or AB Tests, when the tracking is switched off at the parent campaign, it is cascaded to all child cells.

campLimit

The parameters in this object are used to apply optional restrictions to the Campaign Audience.

Example:

```
"campLimit":
  {
    "msgLimit": 5000,
```


© 2025 Marigold. All rights reserved.

```
    "msgPerPkLimit": 1,
    "dedupePropId": 1150,
    "dedupeFilterId": 29094,
    "dedupeSortPropId": 15972,
    "dedupeSortOrder": "ASC",
    "dedupeScopeId": 200,
}
```

These parameters are described below in more detail.

msgLimit

This integer parameter is optional.

The **msgLimit** parameter is used to set a hard limit on the total number of messages sent out in this Campaign.

msgPerPkLimit

This integer parameter is optional.

Messaging allows you to limit the Campaign to send only one message to each unique recipient over the entire lifetime of the Campaign. The platform identifies and removes duplicates using the Unique Identifier (also referred to as the "Alternate Key"). The One Message Per ID feature is typically used for triggered Campaigns, where the same individual could theoretically qualify to receive a message more than once. Using this feature, the recipient will receive only one message from the Campaign.

The One Message Per ID feature is conceptually similar to the De-Duplication Logic feature (described below), but that feature allows you to dedupe on any single field, not just on the Unique Identifier, and to define the rules for how to pick the "winner" from among a set of duplicate records.

To enable the One Message Per ID feature, provide a value of "1" in the **msgPerPkLimit** parameter.

dedupePropId

This integer parameter is optional.

De-duplication (or "dedupe") refers to the process of identifying and removing duplicate records from your Campaign Audience, in order to ensure that recipients don't receive unwanted multiple copies of your message. The De-Duplication Logic feature allows you

© 2025 Marigold. All rights reserved.

to select what field you want to use for identifying duplicate records, as well as the rules for picking the "winner" from among a set of duplicate records.

The De-duplication Logic feature is conceptually similar to the "One Message Per ID" feature (described above), but the De-duplication Logic feature allows you to define more complex logic, and you can dedupe on a field other than the Unique Identifier.

The **dedupePropId** references the **Field ID** for the field that you want to use to identify duplicate records. The system will perform a byte-for-byte match on the values in this field to attempt to find duplicates.

dedupeFilterId

This integer parameter is optional.

The **dedupeFilterId** parameter is part of the De-Duplication Logic feature (see **dedupePropId** above for more details on this feature).

Optionally, you can use a Filter to define the "winner" from among a set of duplicate records. For example, you could define a Filter that selects records that have click activity, or that made a purchase. The **dedupeFilterId** references the **Object Reference ID** of the desired De-Duplication Logic Filter.

> **Note** If you don't define a Filter and / or Sort option to select the "winner" from among a set of duplicate records, the system will sort the set by the Primary Key ID ("pk_id") field in descending order, then pick the top-most record. This default option roughly approximates picking the "most recently added" record.

dedupeSortPropId

This integer parameter is optional.

The **dedupeSortPropId** parameter is part of the De-Duplication Logic feature (see **dedupePropId** above for more details on this feature).

Optionally, you can sort the set of duplicate records on a specified field. This option can be used with or without the Filter (see **dedupeFilterId** above for details). The system will sort the records in the duplicate set by the field you select, in the sequence you select

© 2025 Marigold. All rights reserved.

(see **dedupeSortOrder** below), then pick the "topmost" record. For example, you could decide to pick the record with the most recent click activity, or the biggest purchase.

The **dedupeSortPropId** references the **Field ID** for the field that you want to use to sort the set of duplicate records.

dedupeSortOrder

This string parameter is optional.

The **dedupeSortOrder** parameter is part of the De-Duplication Logic feature (see **dedupePropId** above for more details on this feature).

The **dedupeSortOrder** parameter is used in conjunction with the **dedupeSortPropId** parameter described above to define the sort sequence for the specified sort field.

The valid values for **dedupeSortOrder** are:

- "ASC" -- ascending order
- "DESC" -- descending order

dedupeScopeId

This integer parameter is optional.

The **dedupeScopeId** parameter is part of the De-Duplication Logic feature (see **dedupePropId** above for more details on this feature).

For Regular One-off Campaigns, the De-Duplication Logic is always applied to the entire Campaign Audience. For Event-triggered and Date-triggered Campaigns, the **dedupeScopeId** parameter allows you to dedupe on just the records for the current batch of recipients in the message queue, and not the entire Campaign history. The valid values for this parameter are:

- "100" -- Dedupe on just the current batch of recipients in the message queue.
- "200" -- Dedupe on the entire Campaign history.

© 2025 Marigold. All rights reserved.

Parameters -- Message Content

The options and parameters in this section describe how to define the content of your SMS Text message.

contId

This integer parameter is optional.

The **contId** parameter allows you to specify an asset that you want to use as the content source for this Campaign. This parameter contains the **Object Reference ID** of the desired asset. This asset must be a Content Block or a Dynamic Block, and it must have the same source table as the Campaign.

Example:

```
"contId": 52686
```

contBodies

Unlike other channels, such as email, the SMS Text channel doesn't support the use of multiple "format versions" of the message content. An SMS Text Campaign will only have one format version.

The **contBodies** object specifies the format version and the message content.

Example:

```
"contBodies": [
  {
    "type": "TEXT",
    "usageMask": "SMS",
    "body": "Hello, {(name_first)}. This is my SMS Text message
content."
  }
]
```

The parameters in this object are described below in more detail.

type

This string parameter is optional.

© 2025 Marigold. All rights reserved.

The **type** parameter is used in combination with the **usageMask** parameter (described below) to define the format version of the content. For an SMS Text Campaign, the value of this parameter must be "TEXT."

usageMask

This string parameter is optional.

The **usageMask** parameter is used in combination with the **type** parameter described above to define the format version of the content. For an SMS Text Campaign, the value of this parameter must be "SMS."

body

This string parameter is optional.

The **body** parameter is used to provide the actual text message content.

Please note that certain characters and formatting need to be escaped, or the JSON will be considered invalid. The below characters within the HTML or Plain Text message content need to be escaped:

- **Backspace** is replaced with **\b**
- **Form feed** is replaced with **\f**
- **Newline** is replaced with **\n**
- **Carriage return** is replaced with **\r**
- **Tab** is replaced with **\t**
- **Double quote** is replaced with **\"**
- **Backslash** is replaced with ****

For assistance with escaping JSON code, the following tool will parse your JSON code and escape any problematic characters: https://www.freeformatter.com/json-escape.html.

smsMsgTemplate

This object contains a variety of parameters related to the text message.

Example:

© 2025 Marigold. All rights reserved.

```
"SmsMsgTemplate": {
    "toAddressPropId": 10043,
    "expireTime": "2018-07-19T03:00:00"
    }
```

These parameters are described below in more details.

toAddressPropId

This integer parameter is optional.

The **toAddressPropId** is used to specify which field in the Campaign's source table contains the mobile phone number that should be used to contact recipients. You can specify any field in the source table that has a Data Type of "Phone." In this parameter, provide the **Field ID** of the desired mobile phone field.

expireTime

This string parameter is optional.

The **expireTime** parameter is used to set an end-date for any Keyword-driven automated responses sent out by this Campaign. If you don't provide a value for this parameter, the Keyword-driven automated responses will run indefinitely.

Date values should be provided in the format: "YYYY-MM-DDThh:mm:ss."

Example:

```
"expireTime": "2018-03-06T06:30:00"
```

Parameters -- Schedule

The parameters and options described in this section are mostly related to the Campaign's schedule.

When a Campaign is launched, it goes through two separate and distinct phases: building messages and sending messages.

In the "building messages" phase, the platform identifies the intended recipients of the Campaign, identifies all the possible content variations based on the Dynamic Content options used in the Campaign, and determines the Personalization values based on any

© 2025 Marigold. All rights reserved.

Personalization fields used in the content. The steps in this first phase are often collectively referred to as the "queue" process.

In the "sending messages" phase, the system merges together the data and the content in order to assemble the final message. This message is then transmitted to the recipients.

Each of these phases is controlled by its own dedicated schedule that defines when to start the schedule for that phase, and (for triggered Campaigns) how often to execute the process.

> **Note**
>
> For more details on the scheduling options available within the platform, please see the Online Help system, or the *Messaging -- Campaign Scheduling* document.

The SMS CAMPAIGN endpoint uses two separate objects to define the build schedule and the send schedule. The scheduling parameters in the POST message are largely the same within these two objects. The two objects are:

- **queueSchedule**: This object defines the start / end of the build schedule, and the build frequency. In the context of a Date-triggered Campaign, this object controls the "Recurrence Schedule."
- **sendSchedule**: This object defines the start / end of the send schedule, and the send frequency.

The above two objects should be submitted as nested objects beneath the **campParam** object.

The basic structure for defining the schedule in a Campaign is as follows:

```
"campParam":
  {
    "sendSchedule":   <define the details of the send process>
       {
         "startTime":
         "endTime":
         "timeZone":
         "dayFrequency":
            {
```


© 2025 Marigold. All rights reserved.

```
                <frequency details>
            }
        "timeFrequency":
            {
                <frequency details>
            }
    }
"queueSchedule":    <define the details of the build process>
    {
      "startTime":
      "endTime":
      "timeZone":
      "dayFrequency":
          {
              <frequency details>
          }
      "timeFrequency":
          {
              <frequency details>
          }
    }
}
```

Depending on the type of Campaign, not all of the scheduling options and parameters are relevant. As an example, a Regular One-Off Campaign builds and sends messages only once, so there's no need to define a build frequency, or a send frequency; the frequency options are intended for triggered Campaigns which typically build and send messages multiple times.

The scheduling parameters are described below in more detail.

startTime

This string parameter is optional.

Optionally, you can use the **startTime** parameter to specify the date / time when you want the schedule to go "live." If you don't provide this parameter, the system defaults to "immediately," meaning the schedule will go live when the Campaign is launched.

Date values should be provided in the format: "YYYY-MM-DDThh:mm:ss."

Example:

```
"startTime": "2018-03-06T00:00:00"
```


© 2025 Marigold. All rights reserved.

endTime

This string parameter is optional.

Optionally, you can use the **endTime** parameter to define an end date / time for the schedule. If you don't provide this parameter, the system will default to running the schedule indefinitely, until the Campaign finishes, or is stopped or cancelled.

Date values should be provided in the format: "YYYY-MM-DDThh:mm:ss."

Example:

```
endTime": "2018-03-16T00:00:00"
```

timeZone

This string parameter is optional.

The **timeZone** parameter specifies the Time Zone to use for all date-related features in this Campaign. If you don't provide this parameter, the system will default to using the time zone selected in your User Profile. The valid values for this parameter are specified in Appendix B.

Example:

```
"timeZone": "Central_Standard_Time"
```

dayFrequency

The **dayFrequency** object is used to define when, and how often, the process should execute. You can define the frequency based on a daily, weekly, monthly, or yearly occurrence.

The parameters in this object are described below in more detail.

frequencyType

This string parameter is optional.

The **frequencyType** is used to define the unit of time for setting up the frequency. The valid values for this parameter are:

- "Daily" -- Define a daily interval at which to execute the process.

© 2025 Marigold. All rights reserved.

- "Weekly" -- Specify the day (or days) of the week on which to execute the process.
- "Monthly" -- Specify the day of the month on which to execute the process.
- "Yearly" -- Specify the date on which to execute the process.

Depending on which frequency type you select, the other parameters in the **dayFrequency** object will then define the details.

daysInterval

This integer parameter is optional.

If you're executing the process at a daily frequency, the **daysInterval** parameter allows you to define how many days between intervals. For example, if you want to execute the process every three days, you would provide a value of "3" in this parameter.

Example of a daily frequency:

```
"dayFrequency":
  {
    "frequencyType": "Daily",
    "daysInterval": 3
  }
```

weeklyInterval

This string parameter is optional.

If you're executing the process at a weekly frequency, the **weeklyInterval** parameter allows you to define on which days of the week you want to run the process. The valid values for this parameter are:

- "Sunday"
- "Monday"
- "Tuesday"
- "Wednesday"
- "Thursday"
- "Friday"
- "Saturday "

© 2025 Marigold. All rights reserved.

You can provide one or more values in this parameter; multiple values should be separated by commas.

Example of a weekly frequency:

```
"dayFrequency":
  {
    "frequencyType": "Weekly",
    "weeklyInterval": "Monday, Wednesday, Friday"
  }
```

monthlyInterval

The **monthlyInterval** object is used to define a monthly frequency.

The **monthlyInterval** object should be submitted as a nested object beneath the **dayFrequency** object.

The parameters in this object are described below in more detail.

intervalType

This string parameter is optional.

When setting up a monthly frequency, the system provides two different options for specifying which day in a month to run the process. The **intervalType** parameter is used to select which method to use. The valid values are:

- "DayofMonth" -- Execute process every month on a specific number ("15th of the month" for example).
- "DayType" -- Use a business rule to calculate a day on which to execute the process ("second Tuesday of every month" for example).

dayOfMonth

This integer parameter is optional.

The **dayOfMonth** parameter is used when defining a monthly frequency based on a specific day of the month. In this parameter, provide the day of the month when you want the process to execute. For example, if you want the process to run on the 15th of every month, you would provide a value of "15" in this parameter.

Example of a monthly frequency that runs on the same day every month:

© 2025 Marigold. All rights reserved.

```
"dayFrequency":
  {
    "frequencyType": "Monthly",
    "monthlyInterval":
      {
        "intervalType": "DayOfMonth",
        "dayOfMonth": 15
      }
  }
```

dayTypeInterval / dayType

These string parameters are optional.

The **dayTypeInterval** and **dayType** parameters are used together when defining a monthly frequency based on a business rule to calculate a date.

In the **dayTypeInterval** parameter, provide the interval for the business rule. For example, if you want the process to run on the "second Tuesday" of the month, you would indicate "second" in this parameter. The valid values for this parameter are:

- "First"
- "Second"
- "Third"
- "Fourth"
- "Fifth"

In the **dayType** parameter, indicate which day of the week, or which type of day, is needed for the business rule. For example, if you want the process to run on the "second Tuesday" of the month, you would indicate "Tuesday" in this parameter. The valid values for this parameter are:

- "Sunday"
- "Monday"
- "Tuesday"
- "Wednesday"
- "Thursday"

© 2025 Marigold. All rights reserved.

- "Friday"
- "Saturday "
- "WeekDay"
- "WeekendDay"
- "Day"

Example of a monthly frequency controlled by a business rule:

```
"dayFrequency":
  {
    "frequencyType": "Monthly",
    "monthlyInterval":
      {
        "intervalType": "DayType",
        "dayTypeInterval": "Second",
        "dayType": "Tuesday"
      }
  }
```

yearlyInterval

The **yearlyInterval** object is used to define a yearly frequency.

The **yearlyInterval** object should be submitted as a nested object beneath the **dayFrequency** object.

The parameters in this object are described below in more detail.

intervalType

This string parameter is optional.

When setting up a yearly frequency, the system provides two different options of specifying which day of the year to run the process. The **intervalType** parameter is used to select which method to use. The valid values are:

- "DayofYear" -- Execute process every year on a specific date ("January 15" for example).
- "DayType" -- Use a business rule to calculate a date on which to execute the process ("first day of July" for example).

© 2025 Marigold. All rights reserved.

monthOfYear / dayOfMonth

The **monthOfYear** string parameter is optional; the **dayOfMonth** integer parameter is optional.

The **monthOfYear** and **dayOfMonth** parameters are used together when defining a yearly frequency based on a specific date.

In the **monthOfYear** parameter, provide the name of the month when you want the process to execute. For example, if you want the process to run on January 15, you would provide a value of "January" in this parameter. The valid values for this parameter are:

- "January"
- "February"
- "March"
- "April"
- "May"
- "June"
- "July"
- "August"
- "September"
- "October"
- "November"
- "December"

In the **dayOfMonth** parameter, provide the day of the month when you want the process to execute. For example, if you want the process to run on January 15, you would provide a value of "15" in this parameter.

Example of a yearly frequency that runs on the same date every year:

```
"dayFrequency":
  {
   "frequencyType": "Yearly",
   "yearlyInterval":
```


© 2025 Marigold. All rights reserved.

```
        {
          "intervalType": "DayOfYear",
          "monthOfYear": "January",
          "dayOfMonth": 15
        }
    }
```

dayTypeInterval / dayType / monthOfYear

These three string parameters are all optional.

These parameters are used together when defining a yearly frequency based on a business rule.

In the **dayTypeInterval** parameter, provide the interval for the business rule. For example, if you want the process to run on the "first day of July," you would indicate "first" in this parameter. The valid values for this parameter are:

- "First"
- "Second"
- "Third"
- "Fourth"
- "Fifth"

In the **dayType** parameter, indicate which day of the week, or which type of day, needed for the business rule. For example, if you want the process to run on the "first day of July," you would indicate "day" in this parameter. The valid values for this parameter are:

- "Sunday"
- "Monday"
- "Tuesday"
- "Wednesday"
- "Thursday"
- "Friday"
- "Saturday "

© 2025 Marigold. All rights reserved.

- "WeekDay"
- "WeekendDay"
- "Day"

In the **monthOfYear** parameter, provide the name of the month when you want the process to execute. For example, if you want the process to run on the "first day of July," you would provide a value of "July" in this parameter. The valid values for this parameter are:

- "January"
- "February"
- "March"
- "April"
- "May"
- "June"
- "July"
- "August"
- "September"
- "October"
- "November"
- "December"

Example of a yearly frequency controlled by a business rule:

```
"dayFrequency":
  {
    "frequencyType": "Yearly",
    "yearlyInterval":
      {
        "intervalType": "DayType",
        "monthOfYear": "July",
        "dayType": "Day",
        "dayTypeInterval": "First"
      }
  }
```


© 2025 Marigold. All rights reserved.

timeFrequency

This **timeFrequency** object is used to control at what time of day, or how many times a day, the process should execute.

The parameter in this object are described below in more detail.

timeIntervalType

This string parameter is optional.

When setting up the "time of day" frequency, the system provides two different options for specifying at what time, or how often, to run the process. The **timeIntervalType** parameter is used to select which method to use. The valid values are:

- "OnceADay" -- Execute the process at a specified time of day.
- "MultipleTimesADay" -- Execute the process periodically throughout the day, optionally with the use of a "window" during which time the system will run the process.

runAtTime

This integer parameter is optional.

The **runAtTime** parameter is used to specify a time of day at which to execute the process. The specified time should either be on the hour, or on the half-hour. For example, 1:00, 1:30, 2:00, 2:30, and so forth.

Date values should be provided in the format: "YYYY-MM-DDThh:mm:ss." For the date component of this value, simply use "2000-01-01."

Example of a daily frequency that runs at a specified time of day:

```
"timeFrequency":
  {
    "timeIntervalType": "OnceADay",
    "runAtTime": "2000-01-01T09:00:00"
  }
```


© 2025 Marigold. All rights reserved.

multipleTimesInterval

This **multipleTimesInterval** object is used when you want to execute the process multiple times throughout a day, optionally with the use of a "window" during which time the system will run the process.

The **multipleTimesInterval** object should be submitted as a nested object beneath the **timeFrequency** object.

The parameters in this object are described below in more detail.

runIntervalUnit / runInterval

The **runIntervalUnit** string parameter is optional; the **runInterval** integer parameter is optional.

These parameters are used together when defining a frequency to execute the process multiple times a day.

The **runIntervalUnit** parameter controls the unit of measure for the frequency interval. For example, if you want the process to run every 3 hours, you would indicate "hour" as the desired unit. The valid values for this parameter are:

- "Minute"
- "Hour"

The **runInterval** parameter is used to define the frequency interval, for how often you want the process to run. For example, if you want the process to run every 3 hours, you would provide a value of "3" in this parameter.

If the **runIntervalUnit** value is "Minute," then the valid values for **runInterval** are: "10," "20," "30," "40," and "50."

If the **runIntervalUnit** value is "Hour," then the valid values for **runInterval** are the integers "1" through "11."

excludeTimeBefore / excludeTimeAfter

These string parameters are optional.

© 2025 Marigold. All rights reserved.

These two parameters are used to define a window during which time the process will execute. For example, let's say you want the process to run only during the normal business hours of 8:00 AM to 5:00 PM. You would use these parameters to instruct the system to exclude all frequency intervals before 8:00 AM, and after 5:00 PM.

Date values should be provided in the format: "YYYY-MM-DDThh:mm:ss." For the date component of this value, simply use "2000-01-01."

Example of a daily frequency that runs every three hours, but only during a specified processing window:

```
"timeFrequency":
  {
    "timeIntervalType": "MultipleTimesADay",
    "multipleTimesInterval":
       {
         "runIntervalUnit": "Hour",
         "runInterval": 3,
         "excludeTimeBefore": "2000-01-00T08:00:00",
         "excludeTimeAfter": "2000-01-00T17:00:00"
       }
  }
```

campLimit

Most of the parameters in the **campLimit** object are used to restrict the Campaign Audience. The following parameter relates to sending, and is described below in more detail.

Example:

```
"campLimit":
  {
    "msgPerHourLimit": 3000
  }
```

msgPerHourLimit

This integer parameter is optional.

By default, the platform will send messages as quickly as it can after they've been created. The **msgPerHourLimit** parameter is used to set a limit on how many messages are deployed per hour. The system will round up the value you provide to the next highest increment of 500.

© 2025 Marigold. All rights reserved.

Parameters -- Responses

The parameters described in this section are used to configure what responses you expect to receive from your Campaign message, such as a keyword response, for example.

linkTrackingUsageMask

This string parameter is optional.

The **linkTrackingUsageMask** parameter is used to control the specific link tracking details, such as which links to track. The platform will parse your Campaign message content to attempt to identify links within the specified format version. The valid values for this parameter are:

- "NONE" -- Do not track links.
- "TEXT" -- Track all links.

If you don't provide the **linkTrackingUsageMask** parameter, the system defaults to using a value of "TEXT."

Example:

```
"linkTrackingUsageMask": "TEXT"
```

linkTrackingDomainId

This string parameter is optional.

The **linkTrackingDomainId** parameter represents the ID of a link tracking domain. If you have multiple domains set up for your account, the **linkTrackingDomainId** is used to specify the desired domain.

The value for this identifier isn't displayed within the user interface, and you can't look it up via an API endpoint. If you need the ID for a link tracking domain, please speak with your Client Services Representative, who can provide this value for you.

If you don't provide the **linkTrackingDomainId** parameter, the system defaults to using the default Link Tracking Domain set in your account.

Example:

© 2025 Marigold. All rights reserved.

```
"linkTrackingDomainId": "1506"
```

campParam

This object contains a variety of different parameters. The parameters related to responses are described below in more detail.

shortenLinksFlag

The **shortenLinksFlag** parameter is used to enable the platform's URL link shortener. Provide a value of "1" in this parameter to enable this feature. If you don't include this parameter, the system defaults to a value of "0" (disabled).

Example:

```
"campParam":
  {
                                                "shortenLinksFlag": 1
  }
```

linkRedirectUrlSuffix

This string parameter is optional.

The **linkRedirectUrlSuffix** parameter is used to provide append codes that you want appended to the tracked links in your message content. Append Codes are an optional feature used for web tracking purposes through your own tracking system, or through third-party vendors like Omniture and Google Analytics. The code (or codes) provided in this parameter will be appended to each tracked link in the campaign.

Example:

```
"campParam":
  {
    "linkRedirectUrlSuffix": "trackingcode"
  }
```

smsResponseTemplates

This object is used to define an automated SMS Keyword response action. Keywords are special words or phrases that, when detected within a recipient's SMS text message, can be used to either:

- Trigger an automated text message response back to the recipient.

© 2025 Marigold. All rights reserved.

- Capture the data in the recipient's text message, and store it in your database.

The **smsResponseTemplates** object is used for both of the above actions, although the relevant parameters vary slightly.

Example of an automated text message action:

```
"smsResponseTemplates":
  [
    {
      "senderId": 1,
      "groupId": 701,
      "matchPhonePropId": 11139,
      "confirmationMessage": "Thanks for your message!"
    }
  ]
```

Example of a data capture action:

```
"smsResponseTemplates":
  [
    {
      "senderId": 1,
      "groupId": 646,
      "matchPhonePropId": 11139,
      "responsePropId": 10994
    }
```

The parameters in this object are described below in more detail.

senderId

This integer parameter is optional.

The **senderId** parameter references the ID of the desired Short Code (Short Codes are associated with Sender Profiles). This Short Code is the number to which recipients will text their responses.

The value for this identifier isn't displayed within the user interface, and you can't look it up via an API endpoint. If you need the ID for a Short Code, please speak with your Client Services Representative, who can provide this value for you.

groupId

This integer parameter is optional.

© 2025 Marigold. All rights reserved.

The **groupId** refers to the **Object Reference ID** of the desired SMS Keyword Group. Keywords can be organized into groups, which allows you to link the same triggered activity to all of the words or phrases contained within that group. By using Keyword Groups, you can create a list of similar words or spelling variations.

matchPhonePropId

This integer parameter is optional.

The **matchPhonePropId** parameter references the **Field ID** of a "Phone" field on the Campaign's source table. The system will look for a match between the mobile phone number on the recipient's response and the field you specify in this parameter, in order to identify the recipient.

confirmationMessage

This string parameter is optional, and is used if using the recipient's SMS text message to trigger an automated response back to the recipient.

The **confirmationMessage** parameter contains the content of the automated text message response. This message will be sent from the platform back to the recipient, when the recipient texts one of the keywords in the specified Keyword Group.

responsePropId

This string parameter is optional, and is used if you're capturing data from the recipient's SMS text messages, and storing it in your database.

The **responsePropId** parameter references the **Field ID** on the Campaign's source table where you want to store the data you're capturing in the text message.

Parameters -- Proofing

The parameters in this section are used to configure the Campaign's proofing options. A proof is a sample message that gets sent to a select individual, or group of individuals, in order to verify that the content, format, and appearance of the message is accurate.

© 2025 Marigold. All rights reserved.

proofFilterId

This integer parameter is optional.

By default, the platform will use the main Campaign Audience to select and populate proofing records. However, you can optionally designate an alternate Audience from which to select the records that should be used for generating proofs. This alternate Audience is defined through the use of a Filter.

The **proofFilterId** parameter represents the **Object Reference ID** of the Proofing Filter.

Example:

```
"proofFilterId": 29094
```

campToList

The **campToList** object is used to add additional assets to your Campaign.

The parameters in this object related to Proofing are described below in more detail.

testListId

This integer parameter is optional.

The **testListId** is used to specify a Proofing Group. A Proofing Group consists of one or more individuals who will receive a test message, or "proof" prior to the Campaign being launched.

To assign a Proofing Group, you must provide the **Object Reference ID** for the desired Proofing Group.

Example:

```
"campToList":
  {
    "testListId": 2442
  )
```

campLimit

Most of the parameters in this object are used to apply optional restrictions to the Campaign Audience. The parameters related to Proofing are described below in more detail.

© 2025 Marigold. All rights reserved.

msgLimitTest

This integer parameter is optional.

The **msgLimitTest** parameter is used to specify the maximum number of proofing messages that you want to generate.

Example:

```
"campLimit":
  {
    "msgLimitTest": 25
  }
```

Parameters -- Auditing

The options and parameters in this section control the pre-launch audit options. The pre-launch audit step is intended as a final check, prior to launching the Campaign.

campStatProps

This object is used to define optional Cross Tab Reports that you want to generate as part of the pre-launch audit step. To create a Cross Tab, you must specify the desired field (or fields) on the Campaign source table. The system will then generate a Cross Tab Report on that field. The Cross Tab Report displays all the unique values stored in this field, along with the number of records that contain each unique value.

Example:

```
"campStatProps": [
    {
        "propId": 15078
    },
    {
        "propId": 15972
    }
]
```

The parameters in this object are described below in more detail.

propId

This integer parameter is optional.

© 2025 Marigold. All rights reserved.

The propId parameter references the Field ID of the desired field on which you want to generate the Cross Tab Report.

campReviewFlags

This object is used to set optional audit "check points" during the Campaign's launch. At each indicated step, the platform will stop and wait for approval before continuing on to the next step. The possible audit check points are:

- **Queuing Statistics**: Confirm the audience counts before letting content calculation begin.
- **Content Permutations**: Review content permutations before letting personalization begin.
- **Sending**: Require approval of the overall Campaign before sending messages.

Example:

```
"campReviewFlags":
   {
     "contCalculationFlag": 0,
     "personalizationFlag": 0,
     "sendingFlag": 1
   }
```

The parameters in this object are described below in more detail.

contCalculationFlag

This integer parameter is optional.

The contCalculationFlag parameter controls the "Queuing Statistics" check point. To enable this audit check point, provide a value of "1" in this parameter. If you don't provide this parameter, the system defaults it to "0."

personalizationFlag

This integer parameter is optional.

The personalizationFlag parameter controls the "Content Permutations" check point. To enable this audit check point, provide a value of "1" in this parameter. If you don't provide this parameter, the system defaults it to "0."

© 2025 Marigold. All rights reserved.

sendingFlag

This integer parameter is optional.

The **sendingFlag** parameter controls the "Sending" check point. To enable this audit check point, provide a value of "1" in this parameter. If you don't provide this parameter, the system defaults it to "1."

© 2025 Marigold. All rights reserved.

3 Edit an SMS Text Campaign

Overview

This section describes how to work with existing SMS Text Campaigns via a GET, PUT, or DELETE request to the SMS CAMPAIGN endpoint.

Retrieve an SMS Campaign

The GET method is used to retrieve all of the information about a specified SMS Text Campaign.

If the Campaign is an Event-triggered Campaign, the response message will not include any details of the trigger type or the trigger configuration (except for the "File Import" trigger type).

When submitting a GET request to the SMS CAMPAIGN endpoint, the request message must include the Campaign's **Object Reference ID** as a query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/SmsCampaign?id=22492
```

Delete an SMS Campaign

The DELETE method is used to delete a specified SMS Campaign.

When submitting a DELETE request to the SMS CAMPAIGN endpoint, the request message must include the Campaign's **Object Reference ID** as a query type parameter within the URL.

For example:

© 2025 Marigold. All rights reserved.

```
https://api.eccmp.com/services2/api/SmsCampaign?id=22492
```

Edit an SMS Campaign

The PUT method allows you to submit modifications to an existing SMS Text Campaign. Using this method, you can change the Campaign name or other attributes, add or remove assets, change the message content, or execute various actions, such as send proofs, run pre-launch audits, or launch the Campaign.

When submitting a PUT request to the SMS CAMPAIGN endpoint, the request message must include the Campaign's **Object Reference ID** as a query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/SmsCampaign?id=22492
```

The parameters for the PUT method are the same as described in the **Create an SMS Text Campaign** section, with the following additions, described below.

campId

This integer parameter is required.

The **campId** represents the Campaign's **Object Reference ID**. The value you provide in this parameter must match the **id** value sent within the URL.

Example:

```
"campId": 22492
```

campAction

This string parameter is optional.

The **campAction** parameter allows you to execute an action, or process, within the specified Campaign. The valid values for this parameter are:

- "SAVE" -- Save the Campaign.
- "PROOF" -- Send proofs.
- "AUDIT" -- Run Pre-Launch Audits.

© 2025 Marigold. All rights reserved.

- "LAUNCH" -- Launch the Campaign.
- "APPROVE" -- Approve the current launch step that's waiting for approval.
- "PAUSE" -- Un-approve the "Sending" step. The system immediately stops sending messages, but continues to queue any new triggered messages, and continues to collect activity data (opens, clicks, etc.) on any messages previously deployed. The status of the Campaign is changed to “Pending Approval.” The Sending step can later be re-approved in order to resume sending
- "SUSPEND" -- Suspend a launched Campaign; this action will cause the system to temporarily stop the Campaign from sending any messages until you resume it again. If you suspend a Date-triggered or an Event-triggered Campaign using this action, the platform will NOT queue any new records while the Campaign is suspended, even if the triggering event, or the Recurrence Frequency, occurs. New triggered records are essentially ignored until you later resume the Campaign.
- "SUSPEND_WITH_QUEUE" (Event-triggered Campaigns only) -- Suspend a launched Event-triggered Campaign; this action will cause the system to temporarily stop the Campaign from sending any messages until you resume it again. Unlike the "SUSPEND" action described above, the platform will continue to queue new triggered records while the Campaign is suspended. If you then later resume the Campaign, the platform will deploy those queued records. This action is not available for Date-triggered and Regular One-off Campaigns.
- "RESUME" -- Resume a suspended Campaign; this action will cause the system to resume sending messages in a suspended Campaign. Any messages that were in the queue at the moment you suspended the Campaign will be deployed.
- "CANCEL" -- Cancel a launched Campaign; this action will cause the system to stop the Campaign from queueing or sending any more messages. Any messages currently in the queue at the moment you cancel the Campaign are removed, and the platform doesn’t retain any history of the fact that these messages were “queued but not sent.” The platform will continue to collect activity data (opens, clicks, etc.) on any records deployed prior to you canceling it. This action can’t be undone, and the Campaign can’t later be resumed.

© 2025 Marigold. All rights reserved.

- "CHANGE" -- Execute the Pick Up Changes process. When executing Pick Up Changes, please note the following conditions:

- You must first suspend the Campaign.
- You can't submit Campaign revisions AND execute Pick Up Changes within the same PUT message. If you need to modify your Campaign, send the desired changes in a PUT message (without the **campAction** parameter). Then, send a second PUT message with campAction equal to "SUSPEND" to suspend the Campaign (as noted above, you must suspend the Campaign before running Pick Up Changes). Then finally, send a third PUT message with **campAction** equal to "CHANGE" to execute Pick Up Changes.

© 2025 Marigold. All rights reserved.

4 Response

This section describes the possible response messages sent back from the SMS CAMPAIGN endpoint.

Success

A successful response to a POST message will generate a response code of "200," followed by the details of the new SMS Text Campaign contained within the body of the response message.

A successful response to a GET message will generate a response code of "200," followed by the details of the specified SMS Text Campaign contained within the body of the response message.

A successful response to a PUT message will generate a response code of "200," followed by the details of the modified SMS Text Campaign contained within the body of the response message.

A successful response to a DELETE message will generate a response code of "204;" the body of the response message will be empty.

Errors

If Messaging encounters a problem with an SMS CAMPAIGN request message, the platform will send an "error" message with details of the problem. Below is a list of error codes and their descriptions.

© 2025 Marigold. All rights reserved.

General Errors

Response Code	Error message	Description
500	Campaign and audience entities do not match	Source table for an asset (such as a Filter) does not match the source table specified for the Campaign. All assets used in a Campaign must be built off the same source table as the Campaign itself.
500	Processing of the HTTP request resulted in an exception.	For a PUT or GET request, mismatch in the Campaign's Object Reference ID. The ID used in the URL doesn't match the ID in the **campId** parameter within the body of the message.
500	Execution Timeout Expired. The timeout period elapsed prior to completion of the operation or the server is not responding. System.ComponentModel.Win32Exception (0x80004005): The wait operation timed out	Database server operation timeout.
500	A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was not found or was not accessible. Verify that the instance name is correct and that SQL Server is configured to allow remote connections.	Database server can not be reached.
400	Invalid EntityId	**EntityId** parameter is missing or invalid.
400	CampAction SUSPEND_WITH_QUEUE is only available for Event Triggered campaigns	The "SUSPEND_WITH_QUEUE" value for **campAction** is valid only for an Event-triggered Campaign.
401	Authorization has been denied for this request	Unauthorized request.
400	This SMS Campaign does not exist	In a GET request, the Campaign's ID does not exist.

© 2025 Marigold. All rights reserved.

Response Code	Error message	Description
400	Campaign has been launched already	When using a **campAction** value of "Launch," the specified Campaign has already been launched.
400	Invalid or missing folder selected for '{0}'	The **parent_obj_id** for the folder where you want to save the Campaign is invalid or does not exist.
400	Link Tracking Domain Id is wrong, it is no entry in the database	Client's link tracking domain has not been configured.
400	Link Tracking Domain is not set	Link tracking domain name is missing for the Campaign.
400	Using Campaign Content ID and Content Bodies in combination is forbidden	Request payload has both **contId** and **contBodies**.
400	Using Campaign Content ID of type DOCUMENT is forbidden	Request payload has **contId** but the content type is of type DOCUMENT.
400	Trigger Type is not supported	The only suported trigger types for this endpoint are ADVANCED_EVENT_TRIGGER and FILE_IMPORT
400	Triggers are only allowed for event triggered campaigns	Campaign is Event-triggered but **campTriggers** was not defined.
400	Load and Send is only supported for EMAIL channel	Selected **entityId** is a Load and Send table.
400	Pickup change is not available until the campaign is suspended.	You must suspend a Campaign before running Pick Up Changes.
400	Filter is NOT allowed to change after launch.	You can't change the Filter in a Regular One-off Campaign once the Campaign is launched.

© 2025 Marigold. All rights reserved.

Advanced Event Trigger Validations

Response Code	Error message	Description
400	Advance Event Trigger not enabled. Please contact administrator	When trying to use AET as the trigger, but AET has not been enabled for this account.
400	SMS Advanced Event Trigger not enabled. Please contact administrator.	AET for the SMS channel has not been enabled for this account.
400	Advanced Event Trigger does not support Cells and Split campaign	The platform does not support the use of AET in a Campaign with split cells.
400	Advanced Event Trigger does not support Share to Social Block, Facebook Like Button, Google+ Button at this time	In a Campaign using AET as the trigger, the platform doesn't support the use of these content objects.
400	Advanced Event Trigger does not support Looping Blocks that contain Filters.	In a Campaign using AET as the trigger, the platform doesn't support Looping Blocks which have filters via updaward join or download join.
400	Advanced Event Trigger support only Dynamic Blocks that contain Field Filters	In a Campaign using AET as the trigger, the platform supports only Dynamic Blocks with filters that use fields in the rules.

Content Validaton

Response Code	Error message	Description
400	Invalid content body type detected: {0}	Content body type is invalid.
400	Mismatch between channel type ({0}) and content body type ({1}) detected	The channel of the Campaign is different than the channel of the selected content source.
400	Invalid content body usage mask detected: {0}	Content body **usageMask** is invalid.
400	Invalid merge symbol detected in the content	Content is invalid.

© 2025 Marigold. All rights reserved.

Object Validaton

Respons e Code	Error message	Description
400	Invalid obj	When trying to update a Campaign, **obj_id** is invalid or does not exist.
400	Invalid Obj ref_id	When trying to update a Campaign, **ref_id** is invalid or does not exist
400	Invalid Obj display name	When trying to update a Campaign, **display_name** is invalid or missing.
400	An Obj with this name already exists	Duplicate Campaign name; **display_name** value must be unique within the specified folder.
400	Content is missing for sendout	**cont_id** is missing and **campAction** is either Launch, Proof or Audit.
400	CampAction SUSPEND_WITH_QUEUE is only available for Event Triggered campaigns	The "SUSPEND_WITH_QUEUE" value for **campAction** is valid only for an Event-triggered Campaign.
400	Default TimeZone is not set	API user does not have a time zone defined

© 2025 Marigold. All rights reserved.

5 Sample Code

This section contains sample messages to help illustrate the structure and functionality of the SMS CAMPAIGN message.

Request Message #1

The following sample POST message creates a simple SMS Text Campaign with most of the default options, and without any of the more complex, sophisticated assets. This "bare bones" message is essentially the minimum required information needed to successfully create a new SMS Text Campaign via the SMS CAMPAIGN endpoint.

This SMS Text Campaign will be a Regular One-off Campaign, with the default scheduling options, and an audience Filter.

JSON Payload

```
{
    "custId": 446,
    "entityId": 100,
    "typeId": "REGULAR",
    "toFilterId": 12902,
    "contBodies": [
        {
            "type": "TEXT",
            "usageMask": "SMS",
            "body": "SMS Text message content goes here."
        }
    ],
    "senderProfileId": 7,
    "campParam": {
        "sendSchedule": {
            "timeZone": "Central_Standard_Time"
                },
        "queueSchedule": {
            "timeZone": "Central_Standard_Time",
            "startTime": "2018-07-24T03:00:00"
```


© 2025 Marigold. All rights reserved.

```
                }
            },
    "smsMsgTemplate": {
        "toAddressPropId": 10043,
        "expireTime": "2018-07-19T03:00:00"
    },
    "obj": {
        "display_name": "Test SMS Campaign API",
        "parent_obj_id": 22803
    }
}
```

Request Message #2

This sample message builds on the previous example by adding a few new assets and options to the new Campaign. These additions are highlighted in red in the sample message below.

This Campaign is a Date-triggered Campaign, with a specified start date to the Recurring Schedule and the Send Schedule, and a Recurring Frequency of "daily, every day, at 9:00 AM." The Campaign also has a Keyword-driven response and a Seed List.

JSON Payload

```
{
    "custId": 446,
    "entityId": 100,
    "typeId": "CALCULATED",
    "toFilterId": 12902,
    "ccFilterId": 6902,
    "contBodies": [
        {
            "type": "TEXT",
            "usageMask": "SMS",
            "body": "SMS Text message content goes here."
        }
    ],
    "senderProfileId": 7,
    "campParam": {
        "sendSchedule": {
           "startTime": "2018-07-24T09:00:00"
       },
       "queueSchedule": {
           "startTime": "2018-07-24T09:00:00",
           "dayFrequency": {
               "frequencyType": "Daily",
               "daysInterval": 1
           },
```


© 2025 Marigold. All rights reserved.

```
            "timeFrequency": {
                "runAtTime": "2000-01-01T09:00:00"
            }
        }
    },
    "smsResponseTemplates": [
        {
            "senderId": 7,
            "groupId": 844,
            "matchPhonePropId": 10043,
            "confirmationMessage": "Confirmation Message goes here."
        }
    ],
    "smsMsgTemplate": {
        "toAddressPropId": 10043,
        "expireTime": "2018-07-19T03:00:00"
    },
    "obj": {
        "display_name": "Test SMS Campaign API",
        "parent_obj_id": 22803
    }
}
```


© 2025 Marigold. All rights reserved.

6 Appendix A -- Identifiers

Messaging uses several different types of IDs when referencing assets, such as tables, fields, folders, Filters, and so forth. This appendix describes these different types of IDs, and provides steps on how to look up the value of an ID.

Entity ID

The Entity ID is a unique, system-generated identifier for every table in your database. This value is not displayed within the application user interface anywhere, so to get the Entity ID for a table, you must retrieve it by means of the TABLE API endpoint.

To retrieve the Entity ID for a table:

1. Submit a request to the TABLE API endpoint. The simplest method is to use the version of the TABLE endpoint that allows you to retrieve information based on the table's name. For example:

   ```
   https://api.eccmp.com/services2/api/Table?tableName=recipient
   ```

2. Within the API response message, the system lists every field in this table. As part of the field details, the response message provides the Entity ID for this table.

Sample Response:

```
{
  "viewId": 1002,
  "entityId": 100,
  "displayName": "create_date",
  "propId": 1030,
  "columnName": "create_date"
}
```

For more details on the TABLE endpoint, please see the Messaging Online Help system or the *Messaging -- Table API Technical Guide*.

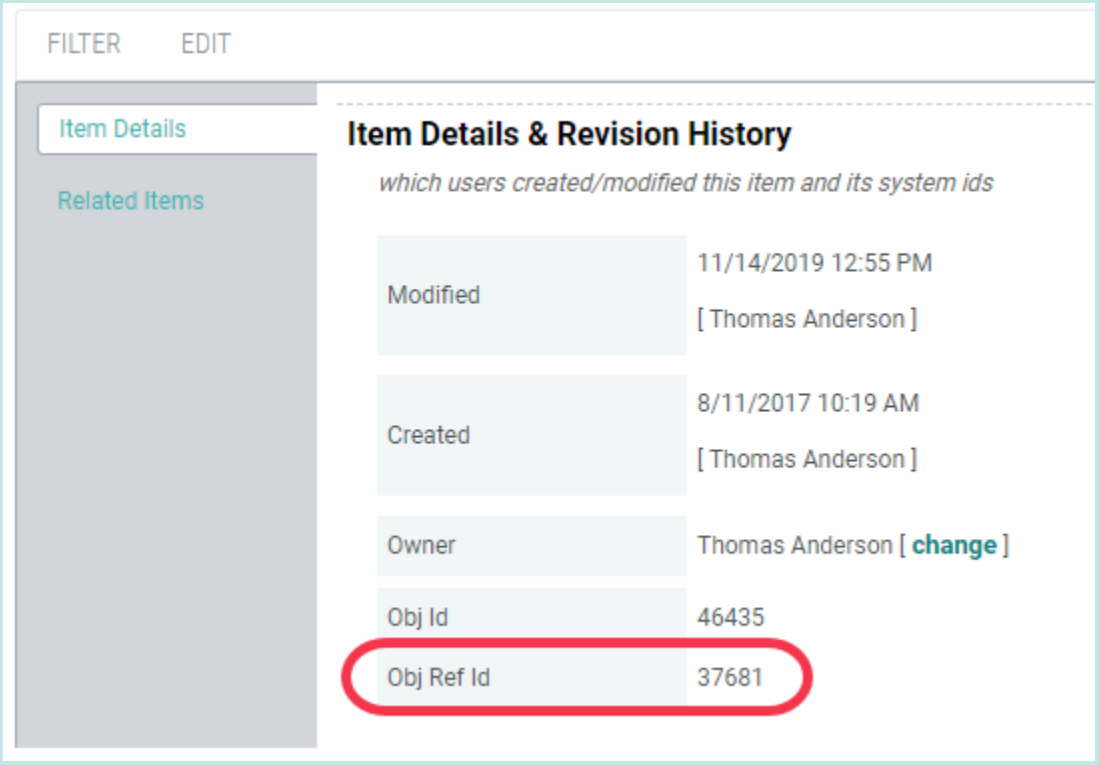
© 2025 Marigold. All rights reserved.

Object Reference ID

The Object Reference ID is a system-generated identifier for every item and asset in your account.

For some asset types, the value for this identifier can be found within the Messaging application:

1. From the System Tray, navigate to desired screen for this asset type.
2. In the Tool Ribbon, click the first tab; the name of this tab corresponds to the asset type, such as "Filter" if you're on the Filter screen, for example.
3. The "Item Details" screen is displayed. The Object Reference ID is listed on this screen.



Optionally, for many asset types, you can use the SEARCH endpoint, and search for the desired asset:

1. Submit a GET request to the SEARCH API endpoint. The simplest method is to use the versions of the SEARCH endpoint that allow you to retrieve information based on

© 2025 Marigold. All rights reserved.

either the asset's name or its type. For example, to retrieve information about all of your Filters:

```
https://api.eccmp.com/services2/api/Object?type=Filter
```

2. The response message provides a list of all the assets in your system that match the search criteria. Find the desired asset in the response message.
3. As part of the API response message, the system provides the Object Reference ID, which is referred to as the "**ref_id**." For example:

```
{
  "obj_id": 44737,
  "display_name": "Reward Members Filter",
  "type_id": "Filter",
  "ref_id": 40329,
  "parent_obj_id": 43269,
  "eligibility_status_id": "READY"
}
```

Field ID

The Field ID (or "Property ID") is a unique, system-generated identifier for every field in a table. This value is not displayed within the application user interface anywhere, so to get the Field ID for a field, you must retrieve it by means of the TABLE API endpoint.

To retrieve the Field ID for a field:

1. Submit a GET request to the TABLE API endpoint. The simplest method is to use the version of the TABLE endpoint that allows you to retrieve information based on the table's name. For example:

```
https://api.eccmp.com/services2/api/Table?tableName=recipient
```

2. Within the API response message, the system lists every field in this table. As part of that field definition, the response includes the Field ID (referred to as the **propId**).

Sample Response:

```
{
 "viewId": 1002,
 "entityId": 100,
 "displayName": "create_date",
 "propId": 1030,
```


© 2025 Marigold. All rights reserved.

```
  "columnName": "create_date"
}
```

Folder ID

The Folder ID is a unique, system-generated identifier for each folder and sub-folder in your system. This value is not displayed within the application user interface anywhere, so to get your Folder ID, you must retrieve it by means of the SEARCH API endpoint.

1. Submit a GET request to the SEARCH endpoint. The easiest method is to use the version that lets you search by object type -- use a type value of "Folder." For example:

```
https://api.eccmp.com/services2/api/Object?type=Folder
```

2. The response message provides a list of all the folders in your system. Find the desired folder in the response message.
3. As part of the API response message, the system provides the Folder ID, which is referred to as the "**obj_id**."

> **Note**
>
> If this Folder is a sub-folder, the "**parent_obj_id**" is the Folder ID of the parent folder.

Sample Response:

```
{
  "obj_id": 37465,
  "display_name": "Content Block Folder",
  "type_id": "Folder",
  "ref_id": 37465,
  "parent_obj_id": 22817,
  "eligibility_status_id": "READY"
}
```


© 2025 Marigold. All rights reserved.

7 Appendix B -- Parameter Values

This Appendix lists all of the valid values for different parameters.

Time Zones

The valid values for the **timeZone** parameter are as follows:

timeZone
UTC_11
Samoa_Standard_Time
Hawaiian_Standard_Time
Alaskan_Standard_Time
Pacific_Standard_Time_Mexico
Pacific_Standard_Time
US_Mountain_Standard_Time
Mountain_Standard_Time_Mexico
Mountain_Standard_Time
Central_America_Standard_Time
Central_Standard_Time
Central_Standard_Time_Mexico
Canada_Central_Standard_Time
SA_Pacific_Standard_Time
Eastern_Standard_Time
US_Eastern_Standard_Time
Venezuela_Standard_Time
Paraguay_Standard_Time

timeZone
FLE_Standard_Time
Israel_Standard_Time
E_Europe_Standard_Time
Arabic_Standard_Time
Arab_Standard_Time
Russian_Standard_Time
E_Africa_Standard_Time
Iran_Standard_Time
Arabian_Standard_Time
Azerbaijan_Standard_Time
Mauritius_Standard_Time
Gegian_Standard_Time
Caucasus_Standard_Time
Afghanistan_Standard_Time
Ekaterinburg_Standard_Time
Pakistan_Standard_Time
West_Asia_Standard_Time
India_Standard_Time

© 2025 Marigold. All rights reserved.

timeZone
Atlantic_Standard_Time
Central_Brazilian_Standard_Time
SA_Western_Standard_Time
Pacific_SA_Standard_Time
Newfoundland_Standard_Time
E_South_America_Standard_Time
Argentina_Standard_Time
SA_Eastern_Standard_Time
Greenland_Standard_Time
Montevideo_Standard_Time
UTC_02
Mid_Atlantic_Standard_Time
Azes_Standard_Time
Cape_Verde_Standard_Time
Mocco_Standard_Time
UTC
GMT_Standard_
Greenwich_Standard_Time
W_Europe_Standard_Time
Central_Europe_Standard_Time
Romance_Standard_Time
Central_European_Standard_Time
W_Central_Africa_Standard_Time
Namibia_Standard_Time
Jdan_Standard_Time
GTB_Standard_Time
Middle_East_Standard_Time
Egypt_Standard_Time
Syria_Standard_Time

timeZone
Sri_Lanka_Standard_Time
Nepal_Standard_Time
Central_Asia_Standard_Time
Bangladesh_Standard_Time
N_Central_Asia_Standard_Time
Myanmar_Standard_Time
SE_Asia_Standard_Time
Nth_Asia_Standard_Time
China_Standard_Time
Nth_Asia_East_Standard_Time
Singape_Standard_Time
W_Australia_Standard_Time
Taipei_Standard_Time
Ulaanbaatar_Standard_Time
Tokyo_Standard_Time
Kea_Standard_Time
Yakutsk_Standard_Time
Cen_Australia_Standard_Time
AUS_Central_Standard_Time
E_Australia_Standard_Time
AUS_Eastern_Standard_Time
West_Pacific_Standard_Time
Tasmania_Standard_Time
Vladivostok_Standard_Time
Central_Pacific_Standard_Time
New_Zealand_Standard_Time
UTC_12
Fiji_Standard_Time
Kamchatka_Standard_Time

© 2025 Marigold. All rights reserved.

timeZone
South_Africa_Standard_Time

timeZone
Tonga_Standard_Time

© 2025 Marigold. All rights reserved.