API Technical Guide: Schedule

Cheetah Messaging

Table of Contents

1 Introduction 5

Purpose 5

Overview 5

Methods 5

Authentication 6

2 Create a New Schedule 7

Overview 7

Parameters 7

startTime 7

endTime 7

timeZone 8

dayFrequency 8

frequencyType 8

daysInterval 8

weeklyInterval 9

monthlyInterval 9

intervalType 10

dayOfMonth 10

dayTypeInterval / dayType 10

yearlyInterval 12

intervalType 12

monthOfYear / dayOfMonth 12

dayTypeInterval / dayType / monthOfYear 13

timeFrequency 15

timeIntervalType 15

runAtTime 16
multipleTimesInterval 16
runIntervalUnit / runInterval 16
excludeTimeBefore / excludeTimeAfter 17
3 Working with an Existing Schedule 18
Overview 18
Retrieve a Schedule 18
id 18
Start / Stop a Schedule 18
id 18
4 Response 20
Success 20
Errors 20
5 Sample Messages 22
Sample Request 22
Sample Response 22
6 Appendix A -- Identifiers 24
Schedule ID 24
7 Appendix B -- Parameter Values 25
Time Zones 25

1 Introduction

Purpose

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

Overview

The SCHEDULE endpoint is used to create and manage repeatable schedule "objects." Using this endpoint, you can manage the schedule details (start and end time, execution frequency, etc.), and start and stop the schedule.

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/Schedule
- **Europe**: https://api.ccmp.eu/services2/api/Schedule
- **Japan**: https://api.marketingsuite.jp/services2/api/Schedule

Methods

The SCHEDULE endpoint supports the following HTTP methods:

- **POST**: Create a new Schedule

- **POST**: Start or stop an existing Schedule
- **GET**: Retrieve the details of an existing Schedule

Authentication

Access to the SCHEDULE 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*.

2 Create a New Schedule

Overview

This section describes how to create a new schedule.

Parameters

The parameters when creating a new schedule are as follows.

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."

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

Example:

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

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.

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. 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.
- "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 are used to 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 "

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:

```
"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"
- "Last"

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"
- "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).

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":
     {
       "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"
- "Last"

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 "
- "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"
      }
  }
```

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"
  }
```

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.

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"
       }
  }
```

3 Working with an Existing Schedule

Overview

This section describes how to work with existing schedules via a GET or POST request to the SCHEDULE endpoint.

Retrieve a Schedule

The GET method is used to retrieve information about a specified schedule. The parameters for this endpoint are described below.

id

When submitting a GET request to the SCHEDULE endpoint, the request message must include the **Schedule ID** as query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/Schedule?id=90255
```

Start / Stop a Schedule

The POST method is used to either start or stop an existing schedule. If the schedule is currently running, the request message will stop the schedule. Conversely, if the schedule is currently paused, the reqeust message will start the schedule.

The parameters for this endpoint are described below.

id

When submitting a POST request to the SCHEDULE endpoint, the request message must include the **Schedule ID** as query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/Schedule?id=90255
```

4 Response

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

Success

A successful response to a POST method to create a new schedule will generate a response code of "200," followed by the details of the new schedule, including the system-generated Schedule ID.

A successful response to a GET method to the schedule endpoint will generate a response code of "200," followed by the details of the specified schedule.

A successful response to a POST method to start / stop a schedule will generate a response code of "200." If you *started* the schedule, the response message references the current date / time. For example:

```
"2018-10-31T05:00:01Z"
```

If you *stopped* the schedule, the response message references the following "blank" date / time:

```
"0001-01-01T00:00:00"
```

Errors

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

Response Code	Error message	Description
500	A Schedule with this ID does not exist.	**id** parameter is missing or invalid.

5 Sample Messages

This section provides sample request and response messages for the SCHEDULE endpoint.

Sample Request

In this sample request message, the user is creating a new schedule.

```
{
     "startTime":
"2018-11-30T10:00:00",
     "timeZone": "Central_Standard_Time",
                                                    "dayFrequency": {

"frequencyType": "Weekly",

"weeklyInterval": "Monday, Wednesday, Friday"
                                                                    },
        "timeFrequency": {
            "timeIntervalType": "MutipleTimesADay",
            "multipleTimesInterval": {
                "runInterval": 15,
                "excludeTimeBefore": "2000-01-01T00:00:01",
                "excludeTimeAfter": "2000-01-01T23:59:59"
            }
      }
}
```

Sample Response

In the response to the above request message, the platform returns the details of the new schedule, including the system-generated Schedule ID.

```
{
    "scheduleId": 90266,
    "startTime": "2018-11-30T10:00:00",
    "timeZone": "Central_Standard_Time",
    "dayFrequency": {
        "frequencyType": "Weekly",
        "weeklyInterval": "Monday, Wednesday, Friday"
```

```
    },
    "timeFrequency": {
        "timeIntervalType": "MutipleTimesADay",
        "multipleTimesInterval": {
            "runInterval": 15,
            "excludeTimeBefore": "2000-01-01T00:00:01",
            "excludeTimeAfter": "2000-01-01T23:59:59"
        }
    }
}
```

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.

Schedule ID

The Schedule ID is a system-generated identifier for a schedule.

The Schedule ID is provided in the response to a POST message to create a new schedule. This response message is the only way to see the Schedule ID for a schedule, so you should make a note of the value when you create a new schedule. The platform doesn't provide any other mechanism -- either through the user interface or through an API -- to look up the Schedule ID.

If you need help finding a Schedule ID, please contact your Client Services Representative, who can look up the value in your database.

7 Appendix B -- Parameter Values

This Appendix lists all of the valid values for the Time Zone parameter.

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

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

timeZone
South_Africa_Standard_Time

timeZone
Tonga_Standard_Time