API Technical Guide: Alert Groups

Cheetah Messaging

Table of Contents

1 Introduction

Purpose

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

Overview

The ALERT GROUPS endpoint allows you to manage your Alert Groups. The primary purpose of an Alert Group is to notify a select group of individuals when some triggering event has occurred. That event could be any activity recorded by the platform, such as an email click or a Web Form submission. When the event takes place, the system deploys a message via Email, SMS Text, or Push Notification to the recipients included in the Alert Group.

When you assign an Alert Group to a Campaign, that Alert Group serves as the only Audience for that Campaign, overwriting any other Audience Filter selections you may have made. For this reason, Alert Groups should be assigned to a separate, dedicated Event-triggered Campaign whose only purpose is to send out alert messages when the desired triggering event takes place. The content of this Alert Group message can be anything you need.

As an example, let's say you want to send a promotional email Campaign to your consumers, and you want to notify your internal salespeople when a consumer clicks on that promotional message. In this scenario, you need to set up two different Campaigns within the platform. The primary Campaign is a typical email marketing Campaign, with your desired promotional content, and an Audience Filter of your intended recipients.

Next, you need a secondary Event-triggered Campaign, with the desired triggering event ("Click" activity in this example). You could define the content of this secondary Campaign to include the contact information for the consumer who clicked on the promotional message. That way, your salespeople could follow up with the consumer. You create an Alert Group consisting of your salespeople, then assign this Alert Group to this secondary Campaign. This Alert Group becomes the Audience for this secondary Campaign. You would then launch both Campaigns.

Let's say a consumer receives the promotional email message, and clicks on it. The platform tracks that activity, which in turn triggers the deployment of the secondary alert Campaign. The platform deploys this Campaign containing the consumer contact information to ONLY the Alert Group.

With this endpoint, you can define a new Alert Group, and view and edit existing Alert Groups. You can also manage the recipients within the Alert Group. Please note that you can't delete an Alert Group using the ALERT GROUPS endpoint; Alert Groups can be deleted only through the application user interface.

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/AlertGroups
- https://api.eccmp.com/services2/api/AlertGroups/{id}
- https://api.eccmp.com/services2/api/AlertGroups/{id}/Recipients
- **Europe**:
- https://api.ccmp.eu/services2/api/AlertGroups
- https://api.ccmp.eu/services2/api/AlertGroups/{id}
- https://api.ccmp.eu/services2/api/AlertGroups/{id}/Recipients
- **Japan**:
- https://api.marketingsuite.jp/services2/api/AlertGroups

- https://api.marketingsuite.jp/services2/api/AlertGroups/{id}
- https://api.marketingsuite.jp/services2/api/AlertGroups/{id}/Recipients

Methods

The ALERT GROUPS endpoint supports the following HTTP methods:

- **POST**: Create a new Alert Group.
- **GET**: Retrieve information about all Alert Groups.
- **GET**: Retrieve information about a specific Alert Group.
- **GET**: Retrieve a list of recipients in a specific Alert Group.
- **PUT**: Edit the details of a specific Alert Group.
- **PUT**: Replace all recipients in a specific Alert Group.
- **PATCH**: Add or remove individual recipients in a specific Alert Group.

Authentication

Access to the ALERT GROUPS 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 an Alert Group

Overview

This section describes how to create a new Alert Group using a POST request to the ALERT GROUPS endpoint.

Creating and populating an Alert Group with recipients is typically a two-step process. When you create a new Alert Group via a POST request, the Alert Group will contain only yourself as a recipient. As a second step, you'll need to send a PUT or a PATCH request to populate the new Alert Group with the desired recipients (see **Manage Alert Group Recipients** for more information).

Parameters

Parameters are the required or optional information contained within the API request message, and tell the system the details of the new Alert Group.

The parameters when sending a POST request to the ALERT GROUPS endpoint are described below in more detail.

name

This string parameter is required.

The **name** parameter contains the display name of the new Alert Group. The name must be unique within your client account.

Example:

```
"name": "API Alert Group"
```

table

This string parameter is required.

The **table** parameter represents the name of the source table for the Alert Group.

You must use the **Table System Name**, and not the table's user-friendly display name.

Example:

```
"table": "recipient"
```

status

This string parameter is optional.

The **status** parameter lets you set the status of this Alert Group -- either "DRAFT" or "READY." Setting an Alert Group to "Draft" status means it can't be selected for use in Campaigns.

If you don't provide this parameter, the system defaults the value to "READY."

Example:

```
"status": "READY"
```

type

This string parameter is required.

The **type** parameter is used to indicate the channel for this Alert Group -- either ""EMAIL," "SMS," or "PUSH."

Example:

```
"type": "EMAIL"
```

tags

This array is optional.

The **tags** array is used to assign one or more Tags to this Alert Group. Tags are an organizational tool, used to group together assets for reporting or Filters.

Example:

```
"tags":
  [
```

```
  "testing",
  "alerts"
]
```

3 View or Edit an Alert Group

Overview

This section describes how to work with existing Alert Groups via a GET or PUT request to the ALERT GROUPS endpoint.

Please note that you can't delete an Alert Group using the ALERT GROUPS endpoint; Alert Groups can be deleted only through the application user interface.

GET Parameters

The GET method is used to retrieve information about all of your Alert Groups, or about a single, specified Alert Group. Both of these options are described below in more detail.

Retrieve All Alert Groups

This endpoint allows you to retrieve a list of all available Alert Groups in your account; you can optionally filter and sort this list to narrow down the results.

The parameters for this endpoint are described below. All of these parameters are query type parameters that should be sent as part of the URL.

status

This string parameter is optional.

The **status** parameter lets you search for Alert Groups based on their status -- either "DRAFT" or "READY.

For example:

```
https://api.eccmp.com/services2/api/AlertGroups?status=READY
```

tags

This string parameter is optional.

The **tags** parameter lets you search for Alert Groups that have been assigned a particular Tag. Optionally, you can provide multiple Tag values, separated by commas. The system will return all the Alert Groups that contain any of the provided Tags.

For example:

```
https://api.eccmp.com/services2/api/AlertGroups?status=READY&tags=Testing
```

archived

This Boolean parameter is optional.

The **archived** parameter lets you search for Alert Groups based on whether the Alert Group has been archived (i.e., deleted) or not. To search only for archived Alert Groups, send a value of "true" in this parameter. If you don't provide this parameter, the system defaults to a value of "false."

For example:

```
https://api.eccmp.com/services2/api/AlertGroups?status=READY&tags=Testing&archived=true
```

count / index

These integer parameters are optional.

These two parameters are used to control how many Alert Groups, and which Alert Groups, should be included within the response message. By default, the system will return the first twenty Alert Groups that match the search criteria, sorted by **name** in ascending order.

However, you can use the **count** and **index** parameters to alter the response message to include a different quantity and selection of Alert Groups.

The **count** parameter will split the response message up into "pages" of the indicated size. The **index** parameter then tells the system which page you want to see in the response message.

Please note that the **index** parameter start with "0" as the first page, "1," for the second page, and so forth.

For example, let's say your account has thirty Alert Groups, and you want to see only the last fifteen Alert Groups in the response message. You could set **count** to "15" so that the system splits the response into two pages (page 0 with Alert Groups 1 through 15, and page 1 with Alert Groups 16 through 30). You would also set **index** to "1," so that you receive only the second page of Alert Groups in the response.

If you don't provide a value for these parameters, the system will default **index** to "0" and **count** to "20." The parameter must be sent as a query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/AlertGroups?status=READY&tags=Test
ing&archived=true&index=1&count=15
```

sort

This string parameter is optional.

The Alert Groups in the response message are always sorted by **name**. The **sort** parameter controls the sort sequence. The possible values are:

- "ASC" -- Ascending order
- "DESC" -- Descending order

If you don't provide a value for this parameter, the system will default to "ASC."

For example:

```
https://api.eccmp.com/services2/api/AlertGroups?status=READY&tags=Test
ing&archived=true&index=1&count=15&sort=DESC
```

Retrieve a Single Alert Group

This endpoint allows you to retrieve information about a single Alert Group in your account.

The request message must include the Alert Group's **Object Reference ID** as a path type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/AlertGroups/3025
```

PUT Parameters

The PUT method allows you to submit modifications to an existing Alert Group. Using this method, you can change most of the information about an Alert Group, including its name, status, table, and Tags; you can't change the Alert Group's channel.

The required parameters are described below. All of the other Alert Group parameters are optional, and should be used to change the desired information. The parameters for the PUT method are the same as described in the **Create an Alert Group** section.

Please note that this method and endpoint is used to edit the high-level Alert Group information, but not the recipients within the Alert Group. See **Manage Alert Group Recipients** for more information on how to modify the recipients within the Alert Group.

id

This integer parameter is required.

The request message must include the Alert Group's **Object Reference ID** as a path type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/AlertGroups/3025
```

type

The request message must include the current **type** value for the specified Alert Group. You can't change this value after the Alert Group has been created.

For example:

```
"type": "EMAIL"
```

4 Manage Alert Group Recipients

Overview

This section describes how to work with the recipients in an Alert Group via a GET, PUT, or PATCH request to the ALERT GROUPS endpoint. Using these methods, you can retrieve a list of the recipients in an Alert Group, replace all of the existing recipients, or add or remove individual recipients.

GET Parameters

The GET method is used to retrieve all of the recipients within a single, specified Alert Group; you can optionally filter and sort this list to narrow down the results.

The request message must include the Alert Group's **Object Reference ID** as a path type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/AlertGroups/3025/Recipients
```

The additional, optional parameters for this endpoint are described below. All of these parameters should be sent as query type parameters within the URL.

count / index

These integer parameters are optional.

These two parameters are used to control how many recipients, and which recipients, should be included within the response message. By default, the system will return the first twenty recipients that match the search criteria, sorted in ascending order.

However, you can use the **count** and **index** parameters to alter the response message to include a different quantity and selection of recipients.

The **count** parameter will split the response message up into "pages" of the indicated size. The **index** parameter then tells the system which page you want to see in the response message.

Please note that he **index** parameter start with "0" as the first page, "1," for the second page, and so forth.

For example, let's say your account has thirty recipients, and you want to see only the last fifteen recipients in the response message. You could set **count** to "15" so that the system splits the response into two pages (page 0 with recipients 1 through 15, and page 1 with recipients 16 through 30). You would also set **index** to "1," so that you receive only the second page of recipients in the response.

If you don't provide a value for these parameters, the system will default **index** to "0" and **count** to "20."

For example:

```
https://api.eccmp.com/services2/api/AlertGroups/3025/Recipients?index=
1&count=15
```

sort

This string parameter is optional.

The **sort** parameter controls the sort sequence of the recipients in the response message. The possible values are:

- "ASC" -- Ascending order
- "DESC" -- Descending order

If you don't provide a value for this parameter, the system will default to "ASC."

For example:

```
https://api.eccmp.com/services2/api/AlertGroups/3025/Recipients?index=1&cou
nt=15?sort=DESC
```

PUT Parameters

The PUT method is used to replace all of the existing recipients in the specified Alert Group with the recipients in the request message.

The request message must include the Alert Group's **Object Reference ID** as a path type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/AlertGroups/3025/Recipients
```

The additional parameters are described below.

alertGroupRecipients

This array is required.

The **alertGroupRecipients** array is used to provide the full set of replacement recipients for this Alert Group.

For each recipient, you must include the contact information in the **value** parameter, and the content type in the **type** parameter. The valid values for **type** are:

- "MULTIPART" -- Recipient will receive a multi-part message (i.e., HTML and plain text).
- "TEXT" -- Recipient will receive only the plain text message.

For example:

```
[
    {
        "value": "mary.smith@cheetahdigital.com",
        "type": "MULTIPART"
    },
    {
        "value": "john.doe@cheetahdigital.com",
        "type": "TEXT"
    }
]
```

PATCH Parameters

The PATCH method is used to either add or remove individual recipients in an existing Alert Group.

If you want to replace all of the recipients in an Alert Group, you can use the PUT method, described above.

The request message must include the Alert Group's **Object Reference ID** as a path type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/AlertGroups/3025/Recipients
```

The additional parameters are described below.

remove

This Boolean parameter is optional.

The **remove** parameter is used to tell the system whether you're adding recipients to the Alert Group, or removing recipients from the Alert Group. The possible values for this parameter are as follows:

- "True" -- Remove the recipients in the request payload from the Alert Group.
- "False" -- Add the recipients in the request payload to the Alert Group.

If you don't provide a value for this parameter, the system will default it to "false."

This parameter must be sent as a query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/AlertGroups/3025/Recipients?remove
=true
```

alertGroupRecipients

This array is required.

The **alertGroupRecipients** array is used to provide one or more recipients being added to, or removed from, the Alert Group. For each recipient, you must include the contact information in the **value** parameter, and (if adding a recipient) the content type in the

type parameter. If you're removing a recipient, you don't need to provide the **type** parameter.

The valid values for **type** are:

- "MULTIPART" -- Recipient will receive a multi-part message (i.e., HTML and plain text).
- "TEXT" -- Recipient will receive only the plain text message.

For example:

```
[
    {
        "value": "john.doe@cheetahdigital.com",
        "type": "TEXT"
    }
]
```

5 Response

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

Success

A successful response to a POST message will generate a response code of "200," followed by the details of the new Alert Group 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 requested information within the body of the response message -- either all Alert Groups, a specified Alert Group, or the recipients within an Alert Group.

A successful response to a PUT message will generate a response code of "200," followed by the details of the modified Alert Group, or the replaced list of recipients, contained within the body of the response message.

A successful response to a PATCH message will generate a response code of "200;" the body of the response message will contain the recipients that were added or removed.

Errors

If Messaging encounters a problem with an ALERT GROUPS 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
400	"Parameter 'table' is required for POST."	Parameter **table** is missing.
400	"A table with this name does not exist."	Parameter **table** is invalid.

Response Code	Error message	Description
400	"Parameter 'type' is required for POST."	Parameter **type** is missing or invalid.
400	"Parameter 'type' is required for PUT."	Parameter **type** is missing or invalid.
400	"An Obj with this name already exists"	Duplicate Alert Group name. The **name** value must be unique within your client account.
400	"An Alert Group with this type and ID does not exist."	In a PUT request, mismatch between the Object Reference ID and the current **type** parameter.
400	"An Alert Group with this ID does not exist."	Invalid Object Reference ID.
400	"Invalid email address detected: <x>"	Email address syntax is invalid.

6 Sample Messages

This section contains several sample request and response messages for the ALERT GROUPS endpoint.

Creating an Alert Group

This POST request creates a new Alert Group. Please note that this Alert Group will initially contain only you as a recipient.

POST Request Payload

```
{
  "name": "Alert Group API Test",
  "table": "recipient",
  "status": "READY",
  "type": "EMAIL",
  "tags":
  [
    "testing",
    "alert"
  ]
}
```

The response message contains the details of the Alert Group, along with the system-generated Object Reference ID. Make a note of this value ("3025"), as you'll need it in the next step.

Response Message

```
{
    "id": 3025,
    "name": "Alert Group API Test",
    "table": "recipient",
    "status": "READY",
    "type": "EMAIL",
    "tags": [
        "testing",
        "alert"
    ]
}
```

Next, you need to populate the new Alert Group with the desired recipients. Send a PUT request to the following URL, using the Object Reference ID ("3025") in the URL:

```
https://api.eccmp.com/services2/api/AlertGroups/3025/Recipients
```

In the payload of the PUT request, include an array containing all of the recipients and content types for the Alert Group.

PUT Request Payload

```
[
    {
        "value": "mary.smith@cheetahdigital.com",
        "type": "MULTIPART"
    },
    {
        "value": "john.doe@cheetahdigital.com",
        "type": "TEXT"
    }
]
```

Sample Response

This sample response shows the results of the **Retrieve a Single Alert Group** version of the ALERT GROUPS endpoint.

JSON Payload

```
{
    "id": 3025,
    "name": "Alert Group API Test",
    "table": "recipient",
    "status": "READY",
    "type": "EMAIL",
    "tags": [
        "testing",
        "alert"
    ]
}
```

7 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.

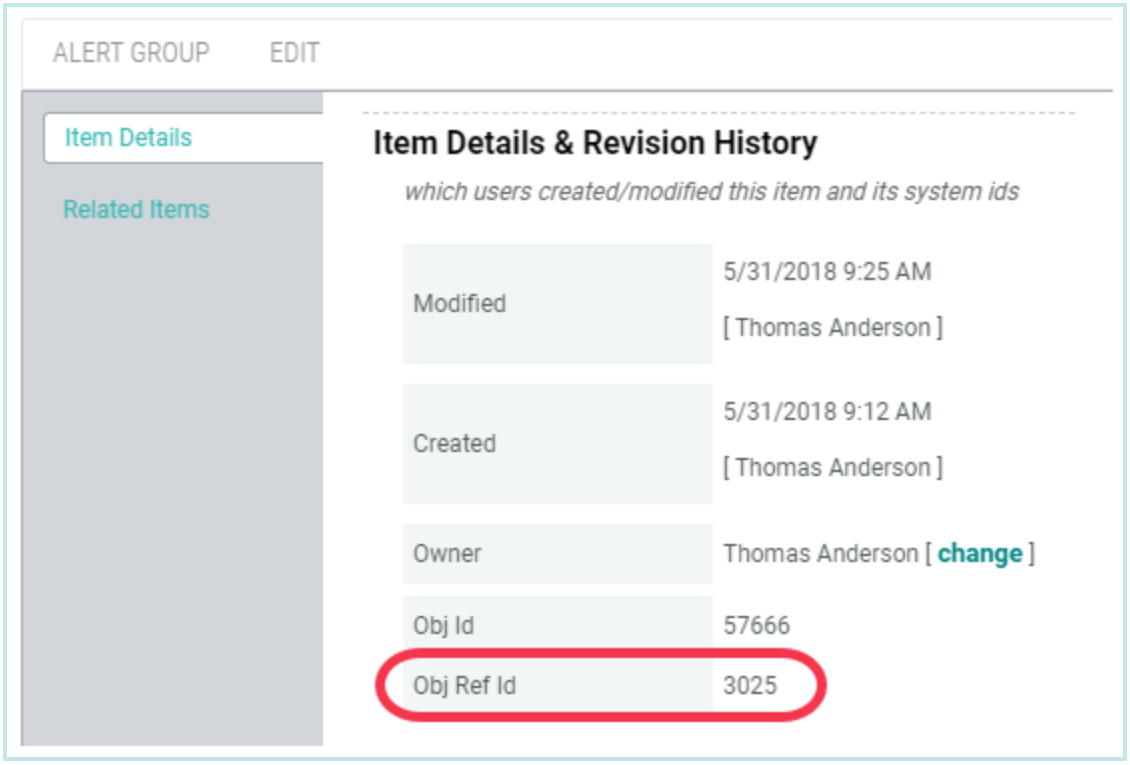
Object Reference ID

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

The value for this identifier for an Alert Group can be found within the Messaging application, or by using the ALERT GROUPS endpoint, which will return the Object Reference ID in the response message.

To find the Object Reference ID within the application:

1. From the System Tray, select *Campaigns* > *Execution* > *Alert Groups*. The system displays a list of all the Alert Groups in your account.
2. Select the desired Alert Group. The Alert Group Details screen is displayed.
3. In the Tool Ribbon, click the "Alert Group" tab. The "Item Details" screen is displayed. The Object Reference ID is listed on this screen.



You can also look up the Object Reference ID for an Alert Group using the **Retrieve All Alert Groups** version of the ALERT GROUPS endpoint. As part of the API response message, the system provides the Object Reference ID in the **id** parameter. For example:

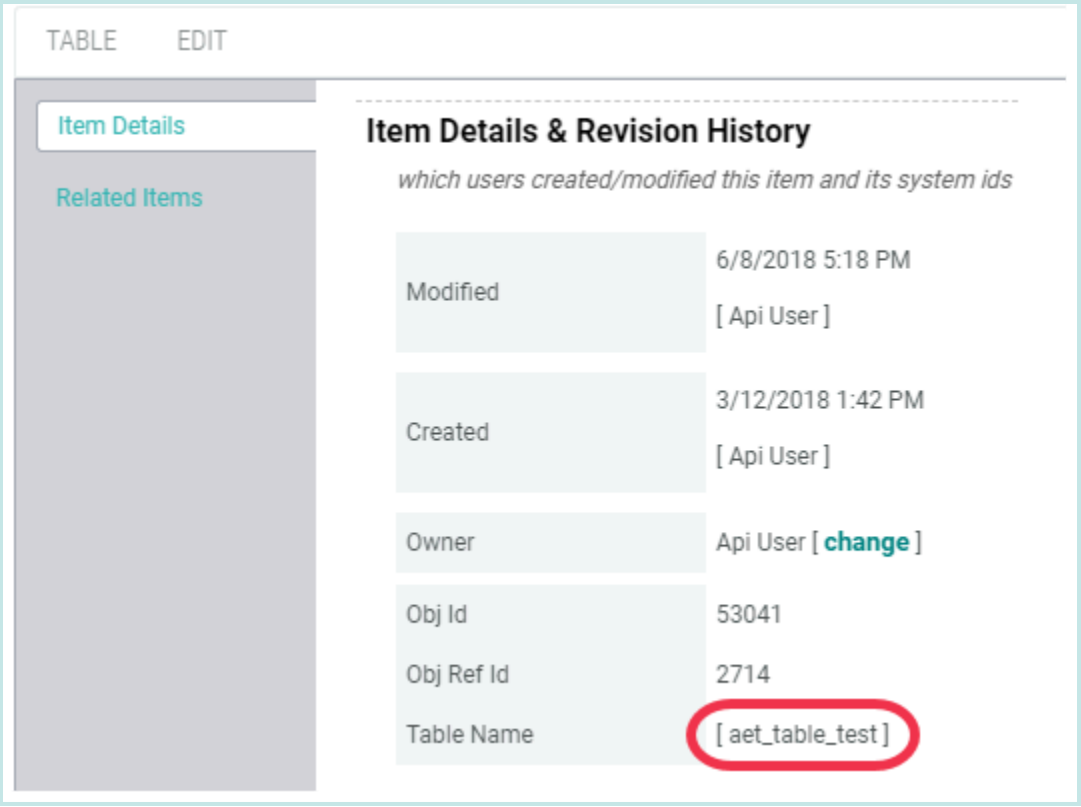
```
{
    "id": 3026,
    "name": "Sales Team Alert Group",
    "table": "recipient",
    "status": "READY",
    "type": "EMAIL",
    "tags": [
        "alert"
    ]
}
```

Table System Name

Tables in Messaging have a user-friendly display name, and a corresponding system-generated name. For example, let's say you have a table with a display name of "Order Item Table." By default, the platform will automatically generate the system name for this table as "order_item_table." When you're submitting the ALERT GROUPS POST or PUT message, you must use the table's system name.

You can look up a table's system name within the application:

1. From the System Tray, select *Data Management* > *Structures* > *Tables*. The system displays a list of all the Tables in your account.
2. Select the desired Table. The Table Details screen is displayed.
3. In the Tool Ribbon, click the "Table" tab. The "Item Details" screen is displayed. The system name for this table is displayed.



You can also retrieve the table's system name using the TABLE API endpoint.

To retrieve the table's system name:

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 for all tables. Please note that depending on the number of tables in your account, you may need to increase the response message size (by default, the system returns the first 20 tables). Within the URL, add the **count** query type parameter, and enter the number of tables you want to return. For example:

```
https://api.eccmp.com/services2/api/Table?count=50
```

2. Within the API response message, the system returns the table details. The table's system name is provided in the **tableName** parameter.

Sample Response:

```
{
  "viewId": 2714,
  "viewName": "AET table test",
  "entityId": 816,
  "tableName": "aet_table_test"
}
```