



API Technical Guide: Activity

Cheetah Messaging

Note

This endpoint has been temporarily deactivated. Please speak to your Client Services Representative if you have any questions

Table of Contents

1	Introduction	4
	Purpose	4
	Overview	4
	Methods	5
	Authentication	5
2	Retrieve Recipient Activity	6
	Overview	6
	Retrieve via Entity ID / Primary Key ID	6
	entityId	6
	pkId	6
	activityName	7
	count / page	7
	Retrieve via Table Name / Primary Key ID	8
	tableName	8
	pkId	8
	activityName	9
	count / page	9
	Retrieve via Entity ID / Alternate Key ID	10
	entityId	10
	akValues	10
	count / page	11
	Retrieve via Table Name / Alternate Key ID	12
	tableName	12
	akValues	12
	count / page	13



3	Response	15
	Success	15
	Errors	16
4	Sample Messages	17
	Sample Response	17
5	Appendix A -- Identifiers	18
	Entity ID	18
	Table Name	19
	Primary Key ID	20
	Alternate Key ID	22



1 Introduction

Purpose

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



Note ~~Endpoint has been temporarily deactivated. Please speak to your Client Services Representative if you have any questions.~~

Overview

The **ACTIVITY** endpoint allows you to retrieve all of the activity data associated with a single recipient. This activity data includes information such as Campaigns sent, link clicks, Web Form submissions, and opt-outs. The endpoint supports a variety of different methods of identifying the recipient whose activity you want to retrieve.

The **ACTIVITY** endpoint requires authentication using OAuth 2.0, and support JSON and XML messages.

The URLs for these endpoints are:

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



Methods

The **ACTIVITY** endpoint supports the following HTTP methods:

- **GET:** Retrieve activity information (optionally filtered by activity type) for a recipient based on the Table's Entity ID and the recipient's Primary Key ID.
- **GET:** Retrieve activity information (optionally filtered by activity type) for a recipient based on the Table's name and the recipient's Primary Key ID.
- **GET:** Retrieve activity information for a recipient based on the Table's Entity ID and the recipient's Alternate Key ID.
- **GET:** Retrieve activity information for a recipient based on the Table's name and the recipient's Alternate Key ID.

Authentication

Access to the **ACTIVITY** 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 *Cheetah Messaging: API How-to Guide*.



2 Retrieve Recipient Activity



Overview

This section describes how to retrieve recipient activity using a GET request to the **ACTIVITY** endpoint. The endpoint provides several different methods of identifying the desired table and recipient.

Retrieve via Entity ID / Primary Key ID

Using a GET method, you can retrieve activity information about a single recipient by specifying the table's Entity ID, and also the recipient's Primary Key ID. You can also optionally filter the response to a specific activity type.

entityId

This integer parameter is required.

When submitting a GET request to the **ACTIVITY** endpoint, the request message must include the table's **Entity ID** as a query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/Activity?entityId=100&pkId=2311113
```

pkId

This integer parameter is required.

When submitting a GET request to the **ACTIVITY** endpoint, the request message must include the recipient's **Primary Key ID** as a query type parameter within the URL.

For example:



`https://api.eccmp.com/services2/api/Activity?entityId=100&pkId=2311113`

activityName

This string parameter is optional.

When submitting a GET request to the **ACTIVITY** endpoint, you may optionally include a specific activity type as a query type parameter within the URL. The response message will contain only the activity records of the specified type.

The valid values for **activityName** are:

- "BOUNCE"
- "CLICK"
- "FORMSUBMISSION"
- "OPTOUT"
- "SENT"
- "UPDATE"

For example:

`https://api.eccmp.com/services2/api/Activity?entityId=100&pkId=2311113
&activityName=CLICK`

count / page

These two integer parameters are optional, and should be provided as query type parameters in the URL.

If you don't provide a value for these parameters, the system will default them to blank.

These parameters are used to control how many activity records, and which records, should be included within the response message. By default, the system will return all activity records for the specified recipient, sorted by **time** in descending order.

However, you can use the **count** and **page** parameters to limit the response message to only a certain quantity and selection of activity records.

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



For example, let's say a recipient has twenty associated activity records, and you want to see only the last ten records in the response message. You could set **count** to "10" so that the system splits the response into two pages (page 1 with records 1 through 10, and page 2 with records 11 through 20). You would also set **page** to "2," so that you receive only the second page of records in the response.

For example:

```
https://api.eccmp.com/services2/api/Activity?entityId=100&pkId=2311113  
&page=2&count=10
```

Retrieve via Table Name / Primary Key ID

Using a GET method, you can retrieve activity information about a single recipient by specifying the table's name, and also the recipient's Primary Key ID. You can also optionally filter the response to a specific activity type.

tableName

This string parameter is required.

When submitting a GET request to the **ACTIVITY** endpoint, the request message must include the **Table Name** as a query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/Activity?tableName=recipient&pkId=  
2311113
```

pkId

This integer parameter is required.

When submitting a GET request to the **ACTIVITY** endpoint, the request message must include the recipient's Primary Key ID as a query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/Activity?tableName=recipient&pkId=  
2311113
```



activityName

This string parameter is optional.

When submitting a GET request to the **ACTIVITY** endpoint, you may optionally include a specific activity type as a query type parameter within the URL. The response message will contain only the activity records of the specified type.

The valid values for **activityName** are:

- "BOUNCE"
- "CLICK"
- "FORMSUBMISSION"
- "OPTOUT"
- "SENT"
- "UPDATE"

For example:

```
https://api.eccmp.com/services2/api/Activity?tableName=recipient&pkId=2311113&activityName=CLICK
```

count / page

These two integer parameters are optional, and should be provided as query type parameters in the URL.

If you don't provide a value for these parameters, the system will default them to blank.

These parameters are used to control how many activity records, and which records, should be included within the response message. By default, the system will return all activity records for the specified recipient, sorted by **time** in descending order.

However, you can use the **count** and **page** parameters to limit the response message to only a certain quantity and selection of activity records.

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



For example, let's say a recipient has twenty associated activity records, and you want to see only the last ten records in the response message. You could set **count** to "10" so that the system splits the response into two pages (page 1 with records 1 through 10, and page 2 with records 11 through 20). You would also set **page** to "2," so that you receive only the second page of records in the response.

For example:

```
https://api.eccmp.com/services2/api/Activity?tableName=recipient&pkId=2311113&page=2&count=10
```

Retrieve via Entity ID / Alternate Key ID

Using a GET method, you can retrieve all of the activity information about a single recipient by specifying the table's Entity ID, and also the recipient's Alternate Key ID.

entityId

This integer parameter is required.

When submitting a GET request to the **ACTIVITY** endpoint, the request message must include the table's **Entity ID** as a query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/Activity?entityId=100&akValues=john.smith@cheetahdigital.com
```

akValues

This integer parameter is required.

When submitting a GET request to the **ACTIVITY** endpoint, the request message must include the recipient's Alternate Key ID (also known as the Unique Identifier) as a query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/Activity?entityId=100&akValues=john.smith@cheetahdigital.com
```

If the Alternate Key for this table comprises multiple fields, the **akValues** parameter must include all of the values, in the correct sequence, separated by commas.



For example, let's say the Alternate Key for a table is made up of two fields -- Email Address and Environment ID.

^ Unique Identifier(s)	
1	Email Address [email_address]
2	Environment_ID [environment_id]
^ Active Fields	
1	First_Name [first_name] X
2	Last_Name [last_name] X

When looking up activity via the Alternate Key, you would need to provide the values for both the "Email Address" field and the "Environment ID" field, in the proper sequence, separated by commas.

For example:

```
https://api.eccmp.com/services2/api/Activity?entityId=100&akValues=john.smith@cheetahdigital.com,qa
```

count / page

These two integer parameters are optional, and should be provided as query type parameters in the URL.

If you don't provide a value for these parameters, the system will default them to blank.

These parameters are used to control how many activity records, and which records, should be included within the response message. By default, the system will return all activity records for the specified recipient, sorted by **time** in descending order.

However, you can use the **count** and **page** parameters to limit the response message to only a certain quantity and selection of activity records.

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

For example, let's say a recipient has twenty associated activity records, and you want to see only the last ten records in the response message. You could set **count** to "10" so that the system splits the response into two pages (page 1 with records 1 through 10, and page



2 with records 11 through 20). You would also set **page** to "2," so that you receive only the second page of records in the response.

For example:

```
https://api.eccmp.com/services2/api/Activity?entityId=100&akValues=john.smith@cheetahdigital.com&page=2&count=10
```

Retrieve via Table Name / Alternate Key ID

Using a GET method, you can retrieve all of the activity information about a single recipient by specifying the table's name, and also the recipient's Alternate Key ID.

tableName

This integer parameter is required.

When submitting a GET request to the **ACTIVITY** endpoint, the request message must include the **Table Name** as a query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/Activity?tableName=recipient&akValues=john.smith@cheetahdigital.com
```

akValues

This integer parameter is required.

When submitting a GET request to the **ACTIVITY** endpoint, the request message must include the recipient's Alternate Key ID (also known as the Unique Identifier) as a query type parameter within the URL.

For example:

```
https://api.eccmp.com/services2/api/Activity?tableName=recipient&akValues=john.smith@cheetahdigital.com
```

If the Alternate Key for this table comprises multiple fields, the **akValues** parameter must include all of the values, in the correct sequence, separated by commas.

For example, let's say the Alternate Key for a table is made up of two fields -- Email Address and Environment ID.



^ Unique Identifier(s)	
1	Email Address [email_address]
2	Environment_ID [environment_id]
^ Active Fields	
1	First_Name [first_name] ✕
2	Last_Name [last_name] ✕

When looking up activity via the Alternate Key, you would need to provide the values for both the "Email Address" field and the "Environment ID" field, in the proper sequence, separated by commas.

For example:

```
https://api.eccmp.com/services2/api/Activity?tableName=recipient&akValues=john.smith@cheetahdigital.com,qa
```

count / page

These two integer parameters are optional, and should be provided as query type parameters in the URL.

If you don't provide a value for these parameters, the system will default them to blank.

These parameters are used to control how many activity records, and which records, should be included within the response message. By default, the system will return all activity records for the specified recipient, sorted by **time** in descending order.

However, you can use the **count** and **page** parameters to limit the response message to only a certain quantity and selection of activity records.

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

For example, let's say a recipient has twenty associated activity records, and you want to see only the last ten records in the response message. You could set **count** to "10" so that the system splits the response into two pages (page 1 with records 1 through 10, and page 2 with records 11 through 20). You would also set **page** to "2," so that you receive only the second page of records in the response.

For example:



<https://api.eccmp.com/services2/api/Activity?tableName=recipient&akValues=john.smith@cheetahdigital.com&page=2&count=10>



3 Response

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



Success

A successful response to a GET method will generate a response code of "200," followed by the activity records associated with the specified recipient. The parameters in the response message are as follows:

- **campaignName:** For activities that can be attributed to a Campaign (such as "CLICK" and "SENT" for example), this parameter displays the Campaign name.
- **time:** The date / time that the activity occurred.
- **linkName:** For "CLICK" activities, this parameter shows the Link name.
- **formName:** For "FORMSUBMISSION" activities, this parameter shows the Web Form name.
- **activityTypeName:** The type of activity; the possible values in this parameter are:
 - "BOUNCE"
 - "CLICK"
 - "FORMSUBMISSION"
 - "OPTOUT"
 - "SENT"
 - "UPDATE"



- **updateTypeName:** For "UPDATE" activities, this parameter indicates the update method, such as "Form," "Import," "Manual Import," or "Relational_Insert" for example.

Errors

If Messaging encounters a problem with an **ACTIVITY** 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	No results for input	No activity records were found for the criteria provided in the request message.



4 Sample Messages

This section contains a sample response message for the **ACTIVITY** endpoint.

Sample Response

Below is a sample response message from the **ACTIVITY** endpoint.

JSON Payload

```
{
  "campaignName": "",
  "time": "2018-09-13T16:08:40.13",
  "linkName": "",
  "formName": "",
  "activityTypeName": "UPDATE",
  "updateTypeName": "Form"
},
{
  "campaignName": "Welcome Email Campaign",
  "time": "2018-09-13T16:08:16.59",
  "linkName": "",
  "formName": "Subscription Confirm Form",
  "activityTypeName": "FORMSUBMISSION",
  "updateTypeName": ""
},
{
  "campaignName": "Welcome Email Campaign",
  "time": "2018-09-13T16:06:39.903",
  "linkName": "Link 01 - /ats/show.aspx?cr=394&fm=2981",
  "formName": "",
  "activityTypeName": "CLICK",
  "updateTypeName": ""
},
{
  "campaignName": "Welcome Email Campaign",
  "time": "2018-09-13T16:05:36.623",
  "linkName": "",
  "formName": "",
  "activityTypeName": "SENT",
  "updateTypeName": ""
}
}
```



5 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 Cheetah Messaging Online Help system or the *Cheetah Messaging -- Table API Technical Guide*.



Table 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 **ACTIVITY** message, you must use the system name.

You can look up table names within the application:

1. From the System Tray, select *Data Management > Structures > Tables*. The system provides a list of all the Tables in your account, using the tables' display names.
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.

TABLE	EDIT
Item Details	
Item Details & Revision History <i>which users created/modified this item and its system ids</i>	
Modified	6/8/2018 5:18 PM [Api User]
Created	3/12/2018 1:42 PM [Api User]
Owner	Api User [change]
Obj Id	53041
Obj Ref Id	2714
Table Name	[aet_table_test]

You can also retrieve table names using the **TABLE** API endpoint.



To retrieve table names:

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"
}
```

Primary Key ID

The Primary Key ID (or "PKID") is a system-generated identifier assigned to every record, in every table, in your Messaging database.

You can look up the PKID for a recipient using the Record Lookup screen in the Messaging application. However, before you can look up this value, you need to make sure that the PKID field is listed as an Active Field on the table.

1. From the System Tray, select *Data Management > Structures > Tables*. The system provides a list of all the Tables in your account.
2. Select the desired Table. The Table Details screen is displayed.
3. Click the "Inactive Fields" section to expand it, and see if the PKID field for this table is currently marked as inactive. The default name of the PKID field is:
"pk_<tablename>_id."



Note

ctive field still exists on the table, but it's removed from all "views" of this table (including the Record Lookup search results).

4. If the PKID field is listed within "Inactive Fields," click the change button ("X" icon) to the right of the field name. The system moves this field from the "Inactive Fields" section up to the "Active Fields" section.

The screenshot displays a table management interface. At the top, there are two active fields: '12 first_name' and '13 long string'. Below them is a section titled 'Inactive Fields' which contains three fields: '1 ak_aet_table_test', '2 domain_prefix', and '# 3 pk_aet_table_test_id'. Each field has a small 'X' icon to its right. The 'X' icon for the 'pk_aet_table_test_id' field is circled in red, indicating it is the field to be activated.

5. In the Tool Ribbon, click *Edit* > *Actions* > *Save*.

Now that the PKID is an "Active Field," you'll be able to see it on the Record Lookup screen.

1. From the System Tray, select *Data Management* > *Management* > *Record Lookup*.
2. Select the desired table, and enter your search criteria. Click **Search**. The search results are displayed.
3. Within the search results, click "Edit" next to the desired record. The Record Details screen is displayed. The Primary Key ID value is listed on this screen.

The screenshot shows a record lookup screen with a label 'pk_recipient_id' and a text input field containing the value '2311113'.

You can also look up the PKID for a recipient using the **SEARCH RECORDS** API. This endpoint allows you to search a table for a value in a specific field; the response message will include the Primary Key ID.

To use the **SEARCH RECORDS** endpoint to look up a PKID:

1. Submit a request to the **SEARCH RECORDS** endpoint, using the desired search criteria.
For example:



```
https://api.eccmp.com/services2/api/SearchRecords?viewName=recipient&prop=email&columnName=email&operation==&param=john.smith@cheetahdigital.com
```

2. In the search results, the Primary Key ID is displayed within the **id** parameter.

```
{
  "id": 2311113,
  "properties": [
    {
      "propName": "email",
      "value": "john.smith@cheetahdigital.com"
    }
  ]
}
```

For more details on the **SEARCH RECORDS** endpoint, please see the Cheetah Messaging Online Help system or the *Cheetah Messaging -- Search Records API Technical Guide*.

Alternate Key ID

In order to ensure that records in your table are created and updated properly, each table can optionally be configured with an Alternate Key ID (also referred to as the "Unique Identifier" or the AKID). The AKID can be based on a single field, or on a combination of multiple fields.

You can look up the AKID for a recipient using the Record Lookup screen in the Messaging application. However, before you can look up this value, you need to make sure that the AKID field is listed as an Active Field on the table.

1. From the System Tray, select *Data Management > Structures > Tables*. The system provides a list of all the Tables in your account.
2. Select the desired Table. The Table Details screen is displayed.
3. Click the "Inactive Fields" section to expand it, and see if the AKID field for this table is currently marked as inactive. The default name of the AKID field is: "ak_<tablename>."

Note

Active field still exists on the table, but it's removed from all "views" of this table (including the Record Lookup search results).



4. If the AKID is listed within "Inactive Fields," click the change button ("X" icon) to the right of the field name. The system moves this field from the "Inactive Fields" section up to the "Active Fields" section.

The screenshot shows a list of fields. At the top are two active fields: 'first_name' (ID 12) and 'long string' (ID 13). Below them is a section titled 'Inactive Fields' with an expand/collapse arrow. Inside this section are three fields: 'ak_aet_table_test' (ID 1), 'domain_prefix' (ID 2), and 'pk_aet_table_test_id' (ID 3). The 'ak_aet_table_test' field has a red circle around its 'X' icon, indicating it is the target for activation.

5. In the Tool Ribbon, click *Edit* > *Actions* > *Save*.

Now that the Alternate Key ID is an "Active Field," you'll be able to see it on the Record Lookup screen.

1. From the System Tray, select *Data Management* > *Management* > *Record Lookup*.
2. Select the desired table, and enter your search criteria. Click **Search**. The search results are displayed.
3. Within the search results, click "Edit" next to the desired record. The Record Details screen is displayed. The Alternate Key ID value is listed on this screen.

The screenshot shows a search result for the 'ak_recipient' field. The value entered is 'john.smith@cheetahdigital.com'.

If the Alternate Key comprises more than one field, the Record Details screen displays all values, separated by a plus sign ("+").

For example, let's say you have a table that uses "Email Address and "Environment ID" fields to make up the Alternate Key ID. On the Record Lookup screen, this field would be displayed with the two values separated by a plus sign.

The screenshot shows a search result for the 'ak_recipient' field. The value entered is 'john.smith@cheetahdigital.com+qa', demonstrating how multiple fields are concatenated with a plus sign.



If you want to look up this recipient's activity using the **ACTIVITY** endpoint, you need to populate the **akValues** parameter with all of the values that make up the AKID, in the proper sequence, and separated by commas. The field sequence can be found by looking at the "Unique Identifier" section of the Tables screen, which lists all of the fields that make up the AKID, and the sequence.

For example:

```
https://api.eccmp.com/services2/api/Activity?tableName=recipient&akValues=john.smith@cheetahdigital.com,qa
```

