Netsurion®
Powering Secure and Agile Networks

API Reference Guide

EventTracker API Guide

Publication Date:

August 29, 2022

Abstract

The guide demonstrates how to successfully invoke the EventTracker APIs to acquire the EventTracker data on your app or website.

Scope

The configuration details in this guide are consistent with the API add-on and the EventTracker version 9.3 or later.

Audience

This guide is for the users to work with the EventTracker API endpoints.

Table of Contents

1 Overview

This section provides fundamental information about the EventTracker APIs and contains the following sections:

- About EventTracker API – This section includes the information about the API design and the EventTracker API features.
- Working with the API – This section provides information on how to use the API.

1.1 About EventTracker API

EventTracker framework is a security LLC flagship event log monitoring and management product. The EventTracker solution is a scalable, enterprise-class Security Information and Event Management (SIEM) solution for Windows systems, Syslog/Syslog NG (UNIX and many networking devices), SNMP V1/V2, legacy systems, applications, and databases.

The EventTracker API allows you to consume EventTracker data from outside EventTracker UI. The EventTracker APIs have the following features:

- Incident API
- Site API
- System API

Architectural design of EventTracker API

The EventTracker APIs are formulated in the REST architectural style, operating on standard HTTP methods and status codes.

The EventTracker APIs are resource-based. Each resource is linked with a URL that identifies a set of objects. An API endpoint is composed of the HTTP method and the URL associated with the resource. The API endpoints produce the JSON schema formatted responses.

Note

To view the open API specification, refer to the EventTracker-API-Addon file.

Paging with Hypermedia as the Engine of Applications State (HATEOAS)

HATEOAS returns not just the data, but also the related actions that can be performed around it.

With HATEOAS, a client interacts with a network application whose application servers provide information dynamically through hypermedia.

1.2 Working with the API

This section includes the following:

- Versioning
- Requests
- Responses
- TimeZone

1.2.1 Versioning

Versioning in API helps to iterate faster when the required changes are identified in the APIs. It facilitates you to demarcate between the older and the new APIs as and when they are released. The version of an API is indicated in the URL of the request.

For example, in the following request, the version of the Incident API is v1.0.

```
EventTrackerAPI/v1.0/incident
```

Note

When a newer version of an API is available, we recommend that you upgrade to the latest version as soon as possible.

1.2.2 Requests

An API request is made whenever you require to communicate with a server to access the EventTracker data. The Requests contain the following components:

- HTTP methods
- Authentication
- Request header
- Request parameters

HTTP methods

The HTTP methods determine the action to take on the resource such as GET, POST. EventTracker APIs support the standard GET method.

Authentication

The process of determining whether a client is qualified to access a resource is known as authentication.

- Pass the API key for User authentication.

- Use the JWT token for each request to access the EventTracker APIs.
- POST is the supported HTTP method used for authentication.

Note

Refer to the JWT Token Generation section to generate the JWT token.

Each EventTracker API request must contain information that identifies the requester as a trusted user. The APIs support the JWT authentication that has two ways, **http header** and **Cookie**.

Note

Passing the token from the cookie is only accepted on http connections. Ensure the cookie name is "BearerToken".

Note

Clients must pass the token in the secure connections by using the Authorization header (HTTPS).

Request header

The EventTracker APIs use the following request header fields:

- **Accept**

  Most APIs produce JSON-formatted responses. For each request, specify the MIME type **application/JSON** using the Accept header field. Refer to the supported formats documented in the API-specific documentation if an API response is in formats other than JSON.

- **Content-Type**

  POST requests use JSON-formatted input. For each POST request, specify the MIME type **application/JSON** using the Content-Type header field.

Request parameters

The input parameters vary by API. Depending on the API, the supported request parameters include URL parameters, POST parameters, and QUERY parameters.

1.2.3 Responses

When an API returns a response, the following elements are included:

- HTTP status codes
- Response parameters

HTTP status codes

The following table includes the most common HTTP status codes returned by the EventTracker APIs:

Status code	Response	Description
200	OK	The request was successful.
400	Bad Request	The request could not be understood by the server due to a syntax error. You must modify the request before attempting to submit it again. Possible errors include an invalid URL or missing parameters in the call.
401	Unauthorized	The request requires user authentication.
403	Forbidden	The server understood the request but refuses to fulfill it. Retry the call with the correct authentication.
404	Not Found	The request did not process successfully due to a permanent error in the URL. This code often generates when the server cannot find the identifier. To attempt the request again, fix the indicated error.
405	Method Not Allowed	The specified method is not allowed for the resource in the request.
406	Not Acceptable	The resource identified in the request can only generate response data with content characteristics that are unacceptable according to the Accept headers in the request.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource.
422	Un-processable Entity	The server understands the content type of the request entity, and the syntax of the request entity is correct, but it was unable to process the contained instructions.
429	Too Many Requests	The user has sent too many requests in a given amount of time ("rate limiting"). A Retry-After header might be included to this response indicating how long to wait before making a new request.
500	Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request.
501	Not Implemented	The server does not support the functionality required to fulfill the request.

Response parameters

The response parameters vary for each API.

1.2.4 Time Zone

All the DateTime-related values should be Universal Time Coordinated (UTC) in API requests and responses. All the datetime parameters will accept the YYYY-MM-DD HH:MM:SS format.

2 Authentication

The EventTracker APIs use the **API key** for authentication. The API key validates the identity when attempting to gain access to the EventTracker API endpoints. The EventTracker application generates a secret key, which is a long, difficult-to-guess string of numbers and letters—at least 30 characters long, though there is no set length. It is usually sent along with the API authorization header.

Note

Contact the EventTracker Support team for the **API key** details.

To generate the JWT token, include the API key in the request body for user authentication. The API verifies the specified key and accepts or rejects the request accordingly.

Example code snippet to generate a JWT token using an API key

Getting an access token using ASP.NET C#.

Function

```
RestClient client = new RestClient(endpointURL + "/v1.0/user/authenticateKey");
RestRequest request = new RestRequest();
request.Method = Method.Post;
request.AddJsonBody ("{"apiKey": " 00112233-4455-6677-8899-aabbccddeeff"}");
      // pass the API Key in the Request Body
IRestResponse restResponse = await client.ExecuteAsync(request);
if (restResponse.IsSuccessful)
{
      // parse the response JSON object to get the access token
}
else
{
      // Look out for appropriate error messages in the response JSON object
}
```

3 JWT Token Generation

This section describes how to obtain the JWT token by authenticating the User with the API key and includes the following information:

- About the User API
- Post user authenticatekey

3.1 About the User API

The User API lets you get the JWT token and using this token you can access the other EventTracker API endpoints.

Method	Resource	Description
POST	/v1.0/user/authenticateKey	This endpoint generates the JWT token by providing the API key.

3.2 POST User authenticateKey

URL	v1.0/user/authenticateKey
HTTP Method	POST

URL Parameter

Parameter	Required	Type	Description
API Key	True	JSON	In the request body, pass the JSON object as **{"apiKey":"string"}**. **Note:** **"string"** to be replaced with the actual **API key**. **Example** {"apiKey":" 00112233-4455-6677-8899-aabbccddeeff"}

Request

```
POST v1.0/user/authenticateKey
```

Response

Code	200
Description	Success


```
{
	"isSuccess": true,
	"status": 0,
	"message": "string",
	"pageDetails": {
		"totalRecords": 0,
		"totalPages": 0
	},
	"links": [
		{
			"href": "string",
			"rel": "string",
			"method": "string"
		}
	],
	"result": [
		{
			"token": "string"
		}
	]
}
```


Response headers


```
cache-control: public,max-age=10
content-type: application/json; charset=utf-8
date: Wed,06 Jul 2022 03:58:14 GMT
server:
transfer-encoding: chunked
vary: Accept-Encoding
x-powered-by:
x-rate-limit: 15
x-rate-limit-remaining: 14
```

For Example:

Request URL

```
http://localhost:8080/EventTrackerAPI/v1.0/user/authenticateKey
```

Response

```
{
      "isSuccess": true,
      "status": 200,
      "message": "Success",
      "pageDetails": null,
      "links": [],
      "result": [
            {
            "token": "eyJhbGcaaaaaiOiJIUzUxMiIsInR5cCI6IkpXVCJ9……"
            }
  ]
}
```

4 Authorization

Authorization determines the data and functions to which an API key has access.

- Copy the JWT token to access the Endpoints.
- After you receive the JWT token, you can access the endpoints.

Example for accessing an API endpoint by providing the JWT token

Sample code for accessing an API endpoint by providing the JWT token in the Authorization header using ASP.NET C#.

Function

```
RestClient client = new RestClient(endpointURL + "/v1.0/incident/details");
RestRequest request = new RestRequest();
request.Method = Method.Get;
request.AddHeader("Authorization", "Bearer
eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiQVBJVXNlciIsIlVzZXJOYW1l
```

```
IjoiQVBJVXNlciIsInVzZXJJZCI6IjIwMDQiLCJuYmYiOjE2NDkzMTUxNDAsImV4cCI6MTY0O
TMxNjA0MCwiaWF0IjoxNjQ5MzE1MTQwLCJpc3MiOiJodHRwczovL3d3dy5ldmVudHRyYWNrZX
IuY29tLyIsImF1ZCI6Imh0dHBzOi8vd3d3LmV2ZW50dHJhY2tlci5jb20vIn0.JGTDy86zPHk
sfM1ULOSIZIFkFRocarjulHI9ULgaAPVudFgdFl2V_D0rtT7FNlj09b6OwcwrKECdc3CEUiru
Ew");
// pass the Bearer Token details here.

request.AddHeader("Content-Type", "application/json");
request.AddParameter("FromDate", "2022-03-23");
request.AddParameter("ToDate", "2022-03-30");
IRestResponse restResponse = await client.ExecuteAsync(request);

if (restResponse.IsSuccessful)
{
// parse the response JSON object to get results
}
else
{
// Look out for appropriate error messages in the response JSON object
}
```

Sample code for accessing an API endpoint by using the JWT token stored in a cookie using ReactJS.

Function

```
let cookieToken = "BearerToken=
eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiQVBJVXNlciIsIlVzZXJ
OYW1lIjoiQVBJVXNlciIsInVzZXJJZCI6IjIwMDQiLCJuYmYiOjE2NDkzMTUxNDAsImV4cCI6MTY
0OTMxNjA0MCwiaWF0IjoxNjQ5MzE1MTQwLCJpc3MiOiJodHRwczovL3d3dy5ldmVudHRyYWNrZXI
uY29tLyIsImF1ZCI6Imh0dHBzOi8vd3d3LmV2ZW50dHJhY2tlci5jb20vIn0.JGTDy86zPHksfM1
ULOSIZIFkFRocarjulHI9ULgaAPVudFgdFl2V_D0rtT7FNlj09b6OwcwrKECdc3CEUiruEw";
    // pass the Bearer token details here.
let urlParam = endpointURL + "/v1.0/incident/details?fromDate=2022-03-
23&toDate=2022-03-30";
document.cookie = cookieToken;
let response = await fetch(urlParam, {
method: "GET",
    headers: {
          "Content-Type": "application/json;charset=utf-8",
                },
    credentials: "include"
```

```
        });
        let resp = await response.json();
```

Note

All the EventTracker API endpoint requests are async in nature.

The endpointURL in the above code references will be the actual URL (can be in the form of http:// or https://) where the API is hosted.

Example: http://localhost:8080/EventTrackerAPI or https://domain.com/EventTrackerAPI

The code snippet illustration above uses Request and Response class objects from the RestSharp library. Refer to the RestSharp documentation to know how to make use of the RestSharp library for accessing the endpoints. Clients can use any other method (HttpClient) to make requests against the endpoints.

Note

Contact the EventTracker support team if you require to re-generate the **API key**.

5 Accessing the EventTracker API Endpoints

Access the EventTracker API endpoints by using the JWT token.

Note

The JWT token generated will be valid only for 15 minutes.

Regenerating the Token to keep the Session Active

All EventTracker API endpoints have an expiry timestamp associated with it. Clients should take appropriate measures to make sure a valid access token is obtained before accessing API endpoints upon expiration of the earlier provided token.

The below section shows the headers that are available in the response header of an API request.

- **`x-expiry-timestamp`**
- **`serverdate`**

Any client wishing to keep the session alive can make use of the above response headers to automate the token regeneration process, rather than regenerating the token upon expiry.

Example: If you wish to regenerate the token automatically, a timer can be implemented by calculating the time difference between “x-expiry-timestamp” and “serverdate”. Embed the code logic to regenerate the access token when the timer is nearing the expiry timestamp of the endpoint.

Note

In accordance with rate limit (x-rate-limit), you can send up to 15 API requests per minute per endpoint.

5.1 Incident API

The Incident API retrieves incident information. An incident comprises correlated suspicious events that require action to maintain your security posture, achieve regulatory compliance, or both. EventTracker API generates incidents based on various predefined scenarios.

The Incident API also provides the ability to retrieve incident notes, which provide details about the incident and recommended remediation actions.

The Incident API, includes three endpoints:

- GET Incident Details
- GET Incident Count
- GET Incident Summary

Method	Resource	Description
GET	/v1.0/incident/details	This endpoint gets the details about an incident.
GET	/v1.0/incident/count	This endpoint gets the total number of incidents.
GET	/v1.0/incident/summary	This endpoint gets the incident summary details.

5.1.1 GET Incident Details

This endpoint is used to retrieve the incident details.

URL	`/v1.0/incident/details`
HTTP Method	`GET`

Query Parameters

Parameter	Required	Type	Description
`fromDate`	true*	string	Pass the **from date** in UTC time zone. Format should be "YYYY-MM-DD HH:MM:SS" where time is optional.
`toDate`	true*	string	Pass the **to date** in UTC time zone. Format should be "YYYY-MM-DD HH:MM:SS" where time is optional.
`groupName`	false	string	This parameter obtains the incident details belonging to a group(s) . This parameter accepts comma(,) separated values.
`computer`	false	string	This parameter obtains the incident details belonging to a computer(s) or system(s). This parameter accepts comma(,) separated values.
`riskValue`	false	string	This parameter obtains the incident details based on the risk value(s) (Critical, Serious, High, Medium, Low, Undefined). This parameter accepts comma(,) separated values.
`incidentName`	false	string	This parameter obtains the incident details based on the incident name(s). This parameter accepts comma(,) separated values.
`incidentNumber`	false	integer	This parameter obtains the incident details based on the incident number. Maximum value - 64 bit integer.
`isAcknowledged`	false	boolean	This parameter obtains the incident details based on the Acknowledgement/Unacknowledgement. Pass true or false. By default, the acknowledged incidents will not be displayed in the result.

Parameter	Required	Type	Description
sortBy	false	string	This parameter sorts the results based on a specific column. Sort order is ascending (ASC) by default except for logtime which is sorted by descending (DESC) order by default. **Example**: incidentName asc, logTime desc. Refer to sortBy Results for more details.
siteId	false	integer	This parameter obtains the incident details belonging to a site. By default, it will consider **siteId** as 0 (Local Site\console). Get the **siteId** using the site details endpoint. Maximum length - 32 bit integer.
pageNum	false	integer	Pass the page number to retrieve the incident details belonging to a page. By default, the value for **pageNum** is 1, and the maximum value depends upon the **totalPages** in response body Maximum length - 32 bit integer.
pageSize	false	integer	Pass the record count to be viewed on a page. By default, the value for **pageSize** is 25, and you can increase the value to a maximum of 100 bit integer.

Note

* You must specify the value for the following mandatory parameters **fromDate, toDate.**

Note

If a specific detail is not provided for any of the optional parameters (that is the Parameters - **false**), all the details that are contained in those optional parameters will be included.

For example, if the **groupName** parameter is left blank then all the groups of that server will get displayed.

sortBy Results

Parameters	**sortBy** Results (by passing only the parameter)
computer	Ascending by default
riskValue	Ascending by default
logTime	Descending by default
incidentName	Ascending by default
eventLogType	Ascending by default

eventSource	Ascending by default
eventType	Ascending by default
eventCategory	Ascending by default
eventId	Ascending by default
eventUser	Ascending by default

Note

The **sortBy** sorts the values in the alphanumeric order according to the specified sorting format.

Request

Get	http://***.**.*.***:8080/EventTrackerAPI/v1.0/incident/details?fromDate=1%2F1%2F2022&toDate=1%2F1%2F2022&siteId=0&pageNum=1&pageSize=25
Curl	curl -X 'GET' \ 'http://***.**.*.***:8080/EventTrackerAPI/v1.0/incident/details?fromDate=1%2F1%2F2022&toDate=1%2F1%2F2022&siteId=0&pageNum=1&pageSize=25' \ -H 'accept: application/json' \ -H 'Authorization: Token'

Note

Curl request has been formatted to improve readability.

Response

Code	200
Description	Success


```
{
	"isSuccess": true,
	"status": 0,
	"message": "string",
	"pageDetails": {
		"totalRecords": 0,
		"totalPages": 0
	},
	"links": [
		{
			"href": "string",
			"rel": "string",
```

```
                    "method": "string"
              }
        ],
        "result": [
              {
                    "id": "string",
                    "incidentNo": "string",
                    "computer": "string",
                    "assetValue": "string",
                    "riskValue": "string",
                    "riskDescription": "string",
                    "logTime": "2022-03-08T00:10:28.554Z",
                    "incidentName": "string",
                    "eventLogType": "string",
                    "eventType": "string",
                    "eventSource": "string",
                    "eventCategory": "string",
                    "eventId": "string",
                    "eventUser": "string",
                    "eventDescription": "string",
                    "alertshortdescription": "string",
                    "isAcknowledged": true,
                    "annotationStatus": true
              }
        ]
}
```


Response headers


```
cache-control: public,max-age=10
content-security-policy: default-src 'self';
content-type: application/json; charset=utf-8
date: Wed,06 Jul 2022 04:04:50 GMT
server:
serverdate: 2022-07-06 04:04:50
transfer-encoding: chunked
vary: Accept-Encoding
x-content-type-options: nosniff
x-expiry-timestamp: 2022-07-06 04:13:14
```

```
x-frame-options: DENY
x-powered-by:
x-rate-limit: 15
x-rate-limit-remaining: 14
x-xss-protection: 1; mode=block
```

For Example:

Request URL

```
http://localhost:8080/EventTrackerAPI/v1.0/incident/details?fromDate=2022-08-10&toDate=2022-08-25&siteId=0&pageNum=1&pageSize=25
```

Response

```
{
  "isSuccess": true,
  "status": 200,
  "message": "Success",
  "pageDetails": {
    "totalRecords": 142,
    "totalPages": 6
  },
  "links": [
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/incident/details?PageNum=1&PageSize=25",
      "rel": "First",
      "method": "GET"
    },
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/incident/details?PageNum=1&PageSize=25",
      "rel": "Previous",
      "method": "GET"
    },
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/incident/details?PageNum=2&PageSize=25",
```

```
      "rel": "Next",
      "method": "GET"
    },
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/incident/details?PageNum=6&PageSi
ze=25",
      "rel": "Last",
      "method": "GET"
    }
  ],
  "result": [
    {
      "id": "595",
      "incidentNo": "202208000010594",
      "computer": "THRESHOLD~ALERT",
      "assetValue": "Undefined",
      "riskValue": "2",
      "riskDescription": "Threat level = Low\n Asset Value = Undefined\n
Vulnerability = Undefined\n",
      "logTime": "2022-08-24T08:26:30",
      "incidentName": "Alert Threshold",
      "eventLogType": "Security",
      "eventType": "Error",
      "eventSource": "EventTracker",
      "eventCategory": "0",
      "eventId": "9858",
      "eventUser": "",
      "eventDescription": "Generated through EvtGen",
      "alertshortdescription": null,
      "isAcknowledged": false,
      "annotationStatus": false
    },
    {
      "id": "585",
      "incidentNo": "202208000010584",
      "computer": "THRESHOLD~ALERT",
      "assetValue": "Undefined",
      "riskValue": "5",
```

```
      "riskDescription": "Threat level = Serious\n Asset Value = Undefined\n
Vulnerability = Undefined\n",
      "logTime": "2022-08-24T08:25:34",
      "incidentName": "Alert Threshold",
      "eventLogType": "Security",
      "eventType": "Error",
      "eventSource": "EventTracker",
      "eventCategory": "0",
      "eventId": "9858",
      "eventUser": "",
      "eventDescription": "Generated through EvtGen",
      "alertshortdescription": null,
      "isAcknowledged": false,
      "annotationStatus": false
    }
  ]
}
```


5.1.2 GET Incident Count

This endpoint is used to retrieve the total number of incidents.

URL	`/v1.0/incident/count`
HTTP Method	`GET`

Query Parameters

Parameter	Required	Type	Description
`fromDate`	true*	string	Pass the **from date** in UTC time zone. Format should be “YYYY-MM-DD HH:MM:SS” where time is optional.
`toDate`	true*	string	Pass the **to date** in UTC time zone. Format should be “YYYY-MM-DD HH:MM:SS” where time is optional.
`groupName`	false	string	This parameter obtains the incident count belonging to a group(s). This parameter accepts comma(,) separated values.

Parameter	Required	Type	Description
computer	false	string	This parameter obtains the incident count belonging to a computer or system(s). This parameter accepts comma(,) separated values.
riskValue	false	string	This parameter obtains the incident count based on the risk value(s) (Critical, Serious, High, Medium, Low, Undefined). This parameter accepts comma(,) separated values.
incidentName	false	string	This parameter obtains the incident count based on the incident name(s). This parameter accepts comma(,) separated values.
isAcknowledged	false	boolean	This parameter obtains the incident count based on the Acknowledgement/Unacknowledgement. Pass true or false. By default, the acknowledged incidents will not be displayed in the result.
siteId	false	integer	This parameter obtains the incident count belonging to a site. By default, it will consider **siteId** as 0 (Local Site\console). Get the **siteId** using the site details endpoint. Maximum length - 32 bit integer.

Note

* You must specify the value for the following mandatory parameters **fromDate, toDate**.

Note

If a specific detail is not provided for any of the optional parameters, all the details that are contained in those optional parameters will be included.

For example, if the **groupName** parameter is left blank then all the groups of that server will get displayed.

Request

Get	`http://***.**.*.**:8080/EventTrackerAPI/v1.0/incident/count?fromDate=1%2F1%2F2022&toDate=1%2F2%2F2022&siteId=0`

Curl	`curl -X 'GET' \`
	
	`'http://***.**.*.***:8080/EventTrackerAPI/v1.0/incident/count?fromDate=1%2F1%2F2022&toDate=1%2F1%2F2022&siteId=0' \`
	`  -H 'accept: application/json' \`
	`  -H 'Authorization: Token'`

Note

Curl request has been formatted to improve readability.

Response

Code	200
Description	Success


```
{
	"isSuccess": true,
	"status": 0,
	"message": "string",
	"pageDetails": {
		"totalRecords": 0,
		"totalPages": 0
	},
	"links": [
		{
			"href": "string",
			"rel": "string",
			"method": "string"
		}
	],
	"result": {
		"totalCount": 0
		}
}
```


Response headers


```
cache-control: public,max-age=10
content-security-policy: default-src 'self';
content-type: application/json; charset=utf-8
date: Wed,06 Jul 2022 04:07:32 GMT
server:
serverdate: 2022-07-06 04:07:32
transfer-encoding: chunked
vary: Accept-Encoding
x-content-type-options: nosniff
```

```
x-expiry-timestamp: 2022-07-06 04:13:14
x-frame-options: DENY
x-powered-by:
x-rate-limit: 15
x-rate-limit-remaining: 14
x-xss-protection: 1; mode=block
```

For Example:

Request URL

```
http://localhost:8080/EventTrackerAPI/v1.0/incident/count?fromDate=2022-08-10&toDate=2022-08-25&siteId=0
```

Response

```
{
  "isSuccess": true,
  "status": 200,
  "message": "Success",
  "pageDetails": null,
  "links": [],
  "result": {
    "totalCount": 142
  }
}
```

5.1.3 GET Incident Summary

This endpoint is used to retrieve the incident summary details.

URL	`/v1.0/incident/summary`
HTTP Method	`GET`

Query Parameters

Parameter	Required	Type	Description
`fromDate`	true*	string	Pass the **from date** in UTC time zone. Format should be "YYYY-MM-DD HH:MM:SS" where time is optional.
`toDate`	true*	string	Pass the **to date** in UTC time zone. Format should be "YYYY-MM-DD HH:MM:SS" where time is optional.
`groupName`	false	string	This parameter obtains the incident summary belonging to a group(s). This parameter accepts comma(,) separated values.
`riskValue`	false	string	This parameter obtains the incident details based on the risk value(s) (Critical, Serious, High, Medium ,Low, Undefined). This parameter accepts comma(,) separated values.
`incidentName`	false	string	This parameter filters the incident summary based on the incident name.
`siteId`	false	`integer`	This parameter obtains the incident summary belonging to a site. By default, it will consider `siteId` as 0(Local Site\console). Get the `siteId` using the site details endpoint. Maximum length - 32 bit integer

Note

* You must specify the value for the following mandatory parameters `fromDate, toDate`.

Note

If a specific detail is not provided for any of the optional parameters, all the details that are contained in those optional parameters will be included.

For example, if the **groupName** parameter is left blank then all the groups of that server will get displayed.

Request

Get	http://***.**.*.***:8080/EventTrackerAPI/v1.0/incident/summary?fromDate=1%2F1%2F2022&toDate=1%2F1%2F2022&siteId=0
Curl	curl -X 'GET' \ 'http://***.**.*.***:8080/EventTrackerAPI/v1.0/incident/summary?fromDate=1%2F1%2F2022&toDate=1%2F1%2F2022&siteId=0' \ -H 'accept: application/json' \ -H 'Authorization: Token'

Note

Curl request has been formatted to improve readability.

Response

Code	200
Description	Success


```
{
	"isSuccess": true,
	"status": 0,
	"message": "string",
	"pageDetails": {
		"totalRecords": 0,
		"totalPages": 0
	},
	"links": [
		{
			"href": "string",
			"rel": "string",
			"method": "string"
		}
	],
	"result": [
		{
			"alertId": 0,
			"incidentName": "string",
			"riskValue": 0,
			"risk": "string",
			"totalSystems": 0,
```

```
                    "totalCount": 0
                }
            ]
}
```

Response headers

```
cache-control: public,max-age=10
content-security-policy: default-src 'self';
content-type: application/json; charset=utf-8
date: Wed,06 Jul 2022 04:11:00 GMT
server:
serverdate: 2022-07-06 04:11:00
transfer-encoding: chunked
vary: Accept-Encoding
x-content-type-options: nosniff
x-expiry-timestamp: 2022-07-06 04:13:14
x-frame-options: DENY
x-powered-by:
x-rate-limit: 15
x-rate-limit-remaining: 14
x-xss-protection: 1; mode=block
```

For Example:

Request URL

```
http://localhost:8080/EventTrackerAPI/v1.0/incident/summary?fromDate=2022-08-10&toDate=2022-08-25&siteId=0
```

Response

```
{
  "isSuccess": true,
  "status": 200,
  "message": "Success",
  "pageDetails": null,
  "links": [],
  "result": [
    {
```

```
      "alertId": 4691,
      "incidentName": "Alert Threshold",
      "riskValue": 2,
      "risk": "Low",
      "totalSystems": 1,
      "totalCount": 17
    },
    {
      "alertId": 4690,
      "incidentName": "testalert1",
      "riskValue": 4,
      "risk": "High",
      "totalSystems": 2,
      "totalCount": 91
    }
  ]
}
```


5.2 Site API

The Site API gives you the details about the local site (Collection Master) and sites (Collection Points) which are reporting to the Collection Master console.

The Site API, includes the following endpoint:

- Get Site Details

Method	Resource	Description
GET	/v1.0/site/details	This endpoint gets the site details (Collection master/Collection point)

5.2.1 GET Site Details

This endpoint is used to retrieve the details of Collection Master and Collection Point.

URL	`/v1.0/site/details`
HTTP Method	`GET`

Request

GET	http://***.**.*.***:8080/EventTrackerAPI/v1.0/site/details
Curl	curl -X 'GET' \ 'http://***.**.*.***:8080/EventTrackerAPI/v1.0/site/details' \ -H 'accept: application/json' \ -H 'Authorization: Token'

Response

Code	200
Description	Success


```
{
	"isSuccess": true,
	"status": 0,
	"message": "string",
	"pageDetails": {
		"totalRecords": 0,
		"totalPages": 0
	},
	"links": [
		{
			"href": "string",
			"rel": "string",
			"method": "string"
		}
	],
	"result": [
		{
			"siteId": 0,
			"siteName": "string",
			"timeZone": "string"
		}
	]
}
```


Response headers


```
cache-control: public,max-age=10
content-security-policy: default-src 'self';
content-type: application/json; charset=utf-8
date: Wed,06 Jul 2022 04:11:00 GMT
server:
```

```
serverdate: 2022-07-06 04:11:00
transfer-encoding: chunked
vary: Accept-Encoding
x-content-type-options: nosniff
x-expiry-timestamp: 2022-07-06 04:13:14
x-frame-options: DENY
x-powered-by:
x-rate-limit: 15
x-rate-limit-remaining: 14
x-xss-protection: 1; mode=block
```

For Example:

Request URL

```
http://localhost:8080/EventTrackerAPI/v1.0/site/details
```

Response

```
{
  "isSuccess": true,
  "status": 200,
  "message": "Success",
  "pageDetails": null,
  "links": [],
  "result": [
    {
      "siteId": 0,
      "siteName": "ETTVMBLRENT03",
      "timeZone": "(UTC-06:00) Central Time (US & Canada)"
    }
  ]
}
```

5.3 System API

The System API endpoints are used to get the details about the Systems and Groups, and Non-Reporting systems (Offline systems).

The System API, includes three endpoints:

- Get System Details
- Get System Groups
- Get System Offline Systems

Method	Resource	Description
GET	/v1.0/system/details	This endpoint gets the details about a system.
GET	/v1.0/system/groups	This endpoint gets the details about a group.
GET	/v1.0/system/offlineSystems	This endpoint gets the details about the system which are offline (non-reporting)

5.3.1 GET System Details

This endpoint is used to get the details about a System.

URL	/v1.0/system/details
HTTP Method	GET

Query Parameters

Parameter	Required	Type	Description
groupName	false	string	This parameter obtains the system details belonging to a group(s). This parameter accepts comma(,) separated values.
computer	false	string	This parameter obtains the system details based on the computer(s) or system(s). This parameter accepts comma(,) separated values.
port	false	string	This parameter obtains the system details based on the port number(s). This parameter accepts comma(,) separated values.
sortBy	false	string	This parameter sorts the results based on a specific column.

Parameter	Required	Type	Description
			Sort order is ascending (ASC) by default. **Example**: computer asc, lastEventReceivedTime desc. Refer to sortBy Results for more details.
`siteId`	false	integer	This parameter obtains the system details belonging to a site. By default, it will consider **`siteId`** as 0(Local Site\console). Get the **`siteId`** using the site details endpoint. Maximum length - 32 bit integer.
`pageNum`	false	integer	Pass the page number to retrieve the system details belonging to a page. Maximum length - 32 bit integer. The default value is 1, and the maximum value depends upon the **`totalPages`** in the Response body.
`pageSize`	false	integer	Pass the record count to be veiwed on a page. Maximum length - 32 bit integer. The default value is 25, and the maximum is 5000.
`includeDLASystem`	false	boolean	This parameter obtains the system details based on the DLA systems. By default, DLA systems will NOT be displayed in the results.
`includeSyslogSystem`	false	boolean	This parameter obtains the system details based on the Syslog systems. By default, Syslog systems will NOT be displayed in the results.
`includeUnmanaged`	false	boolean	This parameter obtains the system details based on the unmanaged systems. By default, unmanaged systems will NOT be displayed in the results.

Note

If a specific detail is not provided for any of the optional parameters, all the details that are contained in those optional parameters will be included.

For example, if the **groupName** parameter is left blank then all the groups of that server will get displayed.

`sortBy` Results

Parameters	**sortBy** Results (by passing only the parameter)
computer	Ascending by default
groupName	Ascending by default
assetValue	Descending by default
type	Ascending by default
System_group_Id	Ascending by default
port	Ascending by default
last_Event_Received_Time	Ascending by default
agent_Installation_Time	Ascending by default
eventTracker_Version	Ascending by default
changeAudit_Version	Ascending by default
SerialNumber	Ascending by default

Note

The **sortBy** sorts the values based on the alphanumeric order except for the **assetValue** parameter.

Request

GET	http://***.**.*.***:8080/EventTrackerAPI/v1.0/system/details?siteId=0&pageNum=1&pageSize=25&includeDLASystem=false&includeSyslogSystem=false&includeUnmanaged=false
Curl	curl -X 'GET' \ 'http://***.**.*.***:8080/EventTrackerAPI/v1.0/system/details?siteId=0&pageNum=1&pageSize=25&includeDLASystem=false&includeSyslogSystem=false&includeUnmanaged=false' \ -H 'accept: application/json' \ -H 'Authorization: Token'

Note

Curl request has been formatted to improve readability.

Response

Code	200
Description	Success


```
{
      "isSuccess": true,
```

```
"status": 0,
"message": "string",
"pageDetails": {
      "totalRecords": 0,
      "totalPages": 0
      },
"links": [
      {
            "href": "string",
            "rel": "string",
            "method": "string"
            }
],
"result": [
      {
            "systemGroupId": 0,
            "computer": "string",
            "description": "string",
            "groupName": [
            "string"
            ],
            "agentInstallationTime": "2022-04-06T12:29:50.584Z",
            "status": "string",
            "lastEventReceivedTime": "2022-04-06T12:29:50.584Z",
            "eventTrackerVersion": "string",
            "changeAuditVersion": "string",
            "type": "string",
            "port": "string",
            "ipAddress": "string",
            "assetValue": "string",
            "fqdn": "string",
            "serialNumber": "string",
            "publicIP": "string",
            "macAddress": "string",
            "timeZone": "string",
            "latitude": "string",
            "longitude": "string"
```

```
                    }
        ]
}
```

Response headers

```
cache-control: public,max-age=10
content-security-policy: default-src 'self';
content-type: application/json; charset=utf-8
date: Wed,06 Jul 2022 04:17:38 GMT
server:
serverdate: 2022-07-06 04:17:38
transfer-encoding: chunked
vary: Accept-Encoding
x-content-type-options: nosniff
x-expiry-timestamp: 2022-07-06 04:28:55
x-frame-options: DENY
x-powered-by:
x-rate-limit: 15
x-rate-limit-remaining: 14
x-xss-protection: 1; mode=block
```

For Example:

Request URL

```
http://localhost:8080/EventTrackerAPI/v1.0/system/details?sortBy=serialNumbe
r%20desc&siteId=0&pageNum=1&pageSize=25&includeDLASystem=false&includeSyslog
System=false&includeUnmanaged=false
```

Response


```
{
  "isSuccess": true,
  "status": 200,
  "message": "Success",
  "pageDetails": {
    "totalRecords": 95,
    "totalPages": 4
  },
  "links": [
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/system/details?pageNum=1&pageSize
=25",
      "rel": "First",
      "method": "GET"
    },
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/system/details?pageNum=1&pageSize
=25",
      "rel": "Previous",
      "method": "GET"
    },
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/system/details?pageNum=2&pageSize
=25",
      "rel": "Next",
      "method": "GET"
    },
    {
```

```
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/system/details?pageNum=4&pageSize
=25",
      "rel": "Last",
      "method": "GET"
    }
  ],
  "result": [
    {
      "systemGroupId": 45,
      "computer": "ETTVMBLRENT03",
      "description": "586, osver 10, ",
      "groupName": [
        "GROUP1"
      ],
      "agentInstallationTime": "2020-04-28T09:43:03",
      "status": "Managed",
      "lastEventReceivedTime": "2022-08-08T08:51:14",
      "eventTrackerVersion": "9.3 - Build 5",
      "changeAuditVersion": null,
      "type": "2019",
      "port": "14505",
      "ipAddress": "localhost",
      "assetValue": "Serious",
      "fqdn": "ETTVMBLRENT03.NTPL.LOCAL",
      "serialNumber": " VMware-42 30 3c 42 32 4b 05 94-e9 e2 69 20 9e 97 b5
db\n\t",
      "publicIP": "182.74.234.198",
      "macAddress": "00-50-56-B0-AB-7D",
      "timeZone": "Asia/Kolkata",
      "latitude": "19.0748",
      "longitude": "72.8856"
    },
    {
      "systemGroupId": 46,
      "computer": "TEST1~GROUP1",
      "description": "",
      "groupName": [
        "GROUP1"
      ],
```

```
    "agentInstallationTime": "0001-01-01T00:00:00",
    "status": "Managed",
    "lastEventReceivedTime": "2022-08-08T02:39:27",
    "eventTrackerVersion": "",
    "changeAuditVersion": null,
    "type": "Unknown",
    "port": "14505",
    "ipAddress": "localhost",
    "assetValue": "Undefined",
    "fqdn": null,
    "serialNumber": null,
    "publicIP": null,
    "macAddress": null,
    "timeZone": null,
    "latitude": null,
    "longitude": null
  },
  {
    "systemGroupId": 52,
    "computer": "DELL~LAPTOP",
    "description": "",
    "groupName": [
      "GROUP1",
      "LAPTOP"
    ],
    "agentInstallationTime": "0001-01-01T00:00:00",
    "status": "Managed",
    "lastEventReceivedTime": "2022-08-08T02:39:27",
    "eventTrackerVersion": "",
    "changeAuditVersion": null,
    "type": "Unknown",
    "port": "14505",
    "ipAddress": "localhost",
    "assetValue": "Undefined",
    "fqdn": null,
    "serialNumber": null,
    "publicIP": null,
    "macAddress": null,
    "timeZone": null,
```

```
      "latitude": null,
      "longitude": null
    },
    {
      "systemGroupId": 92,
      "computer": "ALERT5~CRITICAL",
      "description": "",
      "groupName": [
        "CRITICAL"
      ],
      "agentInstallationTime": "0001-01-01T00:00:00",
      "status": "Managed",
      "lastEventReceivedTime": "2022-08-11T06:52:01",
      "eventTrackerVersion": "",
      "changeAuditVersion": null,
      "type": "Unknown",
      "port": "14505",
      "ipAddress": "localhost",
      "assetValue": "Undefined",
      "fqdn": null,
      "serialNumber": null,
      "publicIP": null,
      "macAddress": null,
      "timeZone": null,
      "latitude": null,
      "longitude": null
    }
  ]
}
```


5.3.2 GET System Groups

This endpoint is used to retrieve the groups belonging to a particular customer (system).

URL	/v1.0/system/groups
HTTP Method	GET

Query Parameters

Parameter	Required	Type	Description
searchText	false	string	This parameter obtains the groups matching to a specified text.
siteId	false	integer	This parameter obtains the groups belonging to a site. By default, it will consider **siteId** as 0(Local Site\console). Get the **siteId** using the site details endpoint. Maximum length - 32 bit integer
pageNum	false	integer	Pass the page number to retrieve the groups belonging to a page. Maximum length - 32 bit integer. The default value is 1, and the maximum value depends upon the **totalPages** in the Response body.
pageSize	false	integer	Pass the record count to be veiwed on a page. Maximum length - 32 bit integer. The default value is 25, and the maximum is 5000.
includeGroupWithSystem	true	boolean	This parameter obtains the groups with systems. By default, a group without systems will not be displayed in the results.

Note

If a specific detail is not provided for any of the optional parameters, all the details that are contained in those optional parameters will be included.

For example, if the **searchText** parameter is left blank then all the group details of that server will get displayed.

Request

Get	/v1.0/system/groups
Curl	curl -X 'GET' \ 'http://***.**.*.***:8080/EventTrackerAPI/v1.0/system/groups?siteId=0&pageNum=1&pageSize=25&includeGroupWithSystem=true&includeVirtualGroup=false' \ -H 'accept: text/plain'

Note

Curl request has been formatted to improve readability.

Response

Code	200
Description	Success


```
{
	"isSuccess": true,
	"status": 0,
	"message": "string",
	"pageDetails": {
		"totalRecords": 0,
		"totalPages": 0
	},
	"links": [
		{
			"href": "string",
			"rel": "string",
			"method": "string"
		}
	],
	"result": [
		{
			"id": 0,
			"name": "string"
		}
	]
}
```


Response headers


```
cache-control: public,max-age=10
content-security-policy: default-src 'self';
content-type: application/json; charset=utf-8
date: Wed,06 Jul 2022 04:19:42 GMT
server:
serverdate: 2022-07-06 04:19:42
```

```
transfer-encoding: chunked
vary: Accept-Encoding
x-content-type-options: nosniff
x-expiry-timestamp: 2022-07-06 04:28:55
x-frame-options: DENY
x-powered-by:
x-rate-limit: 15
x-rate-limit-remaining: 14
x-xss-protection: 1; mode=block
```

For Example:

Request URL

```
http://localhost:8080/EventTrackerAPI/v1.0/system/groups?siteId=0&pageNum=1&
pageSize=05&includeGroupWithSystem=true
```

Response

```
{
  "isSuccess": true,
  "status": 200,
  "message": "Success",
  "pageDetails": {
    "totalRecords": 64,
    "totalPages": 13
  },
  "links": [
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/system/groups?PageNum=1&PageSize=
5",
      "rel": "First",
      "method": "GET"
    },
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/system/groups?PageNum=1&PageSize=
5",
      "rel": "Previous",
      "method": "GET"
```

```
    },
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/system/groups?PageNum=2&PageSize=
5",
      "rel": "Next",
      "method": "GET"
    },
    {
      "id": 147,
      "name": "CONTOSO"
    }
  ]
}
```

5.3.3 GET Offline Systems

This endpoint is used to retrieve the details of the system which are offline (non-reporting).

Note

This endpoint provides all the systems details (excluding the DLA systems) not reporting for the past 24 hours.

URL	`/v1.0/system/offlineSystems`
HTTP Method	`GET`

Query Parameters

Parameter	Required	Type	Description
`groupName`	false	string	This parameter obtains the system details belonging to a group(s). This parameter accepts comma(,) separated values.
`computer`	false	string	This parameter obtains the system details based on the computer or system(s). This parameter accepts comma(,) separated values.
`siteId`	false	integer	This parameter obtains the systems belonging to a site. By default, it will consider **`siteId`** as 0(Local Site\console). Get the **`siteId`** using the site details endpoint.

Parameter	Required	Type	Description
			Maximum length - 32 bit integer.
pageNum	false	integer	Pass the page number to retrieve the systems belonging to a page. Maximum length - 32 bit integer. The default value is 1, and the maximum value depends upon the **totalPages** in the Response body.
pageSize	false	integer	Pass the record count to be veiwed on a page. Maximum length - 32 bit integer. The default value is 25, and the maximum is 5000
days	false	integer	Pass the number of days the systems are offline. By default, it will consider one day. Maximum length - 32 integer.

Note

If a specific detail is not provided for any of the optional parameters, all the details that are contained in those optional parameters will be included.

For example, if the **groupName** parameter is left blank then all the groups of that server will get displayed.

Request

Get	`http://***.**.*.***:8080/EventTrackerAPI/v1.0/system/offlineSystems?siteId=0&pageNum=1&pageSize=25&days=1`

Curl	`curl -X 'GET' \`
	
	`'http://***.**.*.***:8080/EventTrackerAPI/v1.0/system/offlineSystems?siteId=0&pageNum=1&pageSize=25&days=1' \`
	`  -H 'accept: application/json' \`
	`  -H 'Authorization: Token'`

Note

Curl request has been formatted to improve readability.

Response

Code	200
Description	Success

```
{
```

```
"isSuccess": true,
"status": 0,
"message": "string",
"pageDetails": {
      "totalRecords": 0,
      "totalPages": 0
      },
"links": [
      {
            "href": "string",
            "rel": "string",
            "method": "string"
      }
],
"result": [
      {
            "systemGroupId": 0,
            "computer": "string",
            "description": "string",
            "groupName": [
                  "string"
            ],
            "agentInstallationTime": "2022-04-06T12:29:50.584Z",
            "status": "string",
            "lastEventReceivedTime": "2022-04-06T12:29:50.584Z",
            "eventTrackerVersion": "string",
            "changeAuditVersion": "string",
            "type": "string",
            "port": "string",
            "ipAddress": "string",
            "assetValue": "string",
            "fqdn": "string",
            "serialNumber": "string",
            "publicIP": "string",
            "macAddress": "string",
            "timeZone": "string",
            "latitude": "string",
            "longitude": "string"
      }
```

```
      ]
}
```

Response headers

```
cache-control: public,max-age=10
content-security-policy: default-src 'self';
content-type: application/json; charset=utf-8
date: Wed,06 Jul 2022 04:21:42 GMT
server:
serverdate: 2022-07-06 04:21:42
transfer-encoding: chunked
vary: Accept-Encoding
x-content-type-options: nosniff
x-expiry-timestamp: 2022-07-06 04:28:55
x-frame-options: DENY
x-powered-by:
x-rate-limit: 15
x-rate-limit-remaining: 14
x-xss-protection: 1; mode=block
```

For Example:

Request URL

```
http://localhost:8080/EventTrackerAPI/v1.0/system/offlineSystems?siteId=0&pa
geNum=1&pageSize=05&days=1
```

Response

```
{
  "isSuccess": true,
  "status": 200,
  "message": "Success",
  "pageDetails": {
    "totalRecords": 88,
    "totalPages": 18
  },
  "links": [
    {
```

```
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/system/offlineSystems?siteId=0&pa
geNum=1&pageSize=5",
      "rel": "First",
      "method": "GET"
    },
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/system/offlineSystems?siteId=0&pa
geNum=1&pageSize=5",
      "rel": "Previous",
      "method": "GET"
    },
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/system/offlineSystems?siteId=0&pa
geNum=2&pageSize=5",
      "rel": "Next",
      "method": "GET"
    },
    {
      "href":
"http://localhost:8080/EventTrackerAPI/v1.0/system/offlineSystems?siteId=0&pa
geNum=18&pageSize=5",
      "rel": "Last",
      "method": "GET"
    }
  ],
  "result": [
    {
      "systemGroupId": 55,
      "computer": "ACER~LAPTOP",
      "description": "",
      "groupName": [
        "GROUP1",
        "LAPTOP"
      ],
      "agentInstallationTime": "0001-01-01T00:00:00",
      "status": "Managed",
      "lastEventReceivedTime": "2022-08-08T02:39:27",
      "eventTrackerVersion": "",
```

```
    "changeAuditVersion": null,
    "type": "Unknown",
    "port": "14505",
    "ipAddress": "localhost",
    "assetValue": "Undefined",
    "fqdn": null,
    "serialNumber": null,
    "publicIP": null,
    "macAddress": null,
    "timeZone": null,
    "latitude": null,
    "longitude": null
  },
  {
    "systemGroupId": 65,
    "computer": "ALERT~GENERATOR",
    "description": "",
    "groupName": [
      "GENERATOR"
    ],
    "agentInstallationTime": "0001-01-01T00:00:00",
    "status": "Managed",
    "lastEventReceivedTime": "2022-08-10T05:45:27",
    "eventTrackerVersion": "",
    "changeAuditVersion": null,
    "type": "Unknown",
    "port": "14505",
    "ipAddress": "localhost",
    "assetValue": "Undefined",
    "fqdn": null,
    "serialNumber": null,
    "publicIP": null,
    "macAddress": null,
    "timeZone": null,
    "latitude": null,
    "longitude": null
  },
  {
    "systemGroupId": 70,
```

```
    "computer": "ALERT~GROUP1",
    "description": "",
    "groupName": [
      "GROUP1"
    ],
    "agentInstallationTime": "0001-01-01T00:00:00",
    "status": "Managed",
    "lastEventReceivedTime": "2022-08-10T12:51:47",
    "eventTrackerVersion": "",
    "changeAuditVersion": null,
    "type": "Unknown",
    "port": "14505",
    "ipAddress": "localhost",
    "assetValue": "Undefined",
    "fqdn": null,
    "serialNumber": null,
    "publicIP": null,
    "macAddress": null,
    "timeZone": null,
    "latitude": null,
    "longitude": null
  },
  {
    "systemGroupId": 85,
    "computer": "ALERT~GROUP10",
    "description": "",
    "groupName": [
      "GROUP10"
    ],
    "agentInstallationTime": "0001-01-01T00:00:00",
    "status": "Managed",
    "lastEventReceivedTime": "2022-08-10T12:52:11",
    "eventTrackerVersion": "",
    "changeAuditVersion": null,
    "type": "Unknown",
    "port": "14505",
    "ipAddress": "localhost",
    "assetValue": "Undefined",
    "fqdn": null,
```

```
      "serialNumber": null,
      "publicIP": null,
      "macAddress": null,
      "timeZone": null,
      "latitude": null,
      "longitude": null
    },
    {
      "systemGroupId": 68,
      "computer": "ALERT~GROUP2",
      "description": "",
      "groupName": [
        "GROUP2"
      ],
      "agentInstallationTime": "0001-01-01T00:00:00",
      "status": "Managed",
      "lastEventReceivedTime": "2022-08-10T12:51:45",
      "eventTrackerVersion": "",
      "changeAuditVersion": null,
      "type": "Unknown",
      "port": "14505",
      "ipAddress": "localhost",
      "assetValue": "Undefined",
      "fqdn": null,
      "serialNumber": null,
      "publicIP": null,
      "macAddress": null,
      "timeZone": null,
      "latitude": null,
      "longitude": null
    }
  ]
}
```

About Netsurion

Netsurion® delivers an adaptive managed security solution that integrates our XDR platform with your existing security investments and technology stack, easily scaling to fit your business needs. Netsurion's managed offering includes our 24x7 SOC that operates as your trusted cybersecurity partner, working closely with your IT team to strengthen your cybersecurity posture. Our solution delivers Managed Threat Protection so you can confidently focus on your core business.

Headquartered in Ft. Lauderdale, FL with a global team of security analysts and engineers, Netsurion is a leader in Managed Detection and Response (MDR) and a Top 25 Global MSSP. Learn more at www.netsurion.com.

Contact Us

Corporate Headquarters
Netsurion
Trade Centre South
100 W. Cypress Creek Rd
Suite 530
Fort Lauderdale, FL 33309

Contact Numbers
EventTracker Enterprise SOC: 877-333-1433 (Option 2)
EventTracker Enterprise for MSP's SOC: 877-333-1433 (Option 3)
EventTracker Essentials SOC: 877-333-1433 (Option 4)
EventTracker Software Support: 877-333-1433 (Option 5)
https://www.netsurion.com/eventtracker-support