ThreatConnect™

# Management API

## User Guide

**Document Version 1.0**

**March 31, 2021**

10020-01 EN Rev. B

**ThreatConnect, Inc.**
3865 Wilson Blvd., Suite 550, Arlington, VA 22203

P: 1.800.965.2708 | F: 1.703.229.4489
www.ThreatConnect.com

# Table of Contents

# Overview

The Management API allows administrators to query their ThreatConnect® instance to check health, status, and activity. Using REST API commands, administrators can monitor their instance to ensure it is in a healthy state. Additional endpoints allow in-depth checks, like thread dumps and Java® virtual machine (JVM) statistics. The API also supports a few administrative tasks, like clearing the entity cache and resetting locked user accounts. The API is also extensible using custom SQL queries that can be called directly using REST endpoints.

***NOTE: Extending the API with custom SQL is considered an advanced technique, and care must be taken in a production environment to not tax the database with expensive queries. This feature is not supported in Dedicated Cloud instances.***

# Configuration

To use the API, the system configuration 'statuskey' property must be set with a user-generated key. This key will be used in all API request headers to authenticate to the instance. Because this is an administrative feature, standard ThreatConnect API keys will not work.

***NOTE: These instructions apply only to On Premise instances of ThreatConnect. Users running Dedicated Cloud instances should contact their Customer Success Engineer to generate and provide a statuskey.***

To apply this key, users must run the following SQL on their instance with their custom value (limit is 255 characters). This property is hidden in the System Settings UI screen.

```
update systemconfig set value = 'abcd' where name = 'statusKey';
```

All references in this document will use the variables in Table 1 to make requests. Users must use their instance details in place of these variables.

**Table 1**

| Variable | Description |
|---|---|
| {{statuskey}} | The user's status key defined in this section |
| {{baseUrl}} | The user's instance fully qualified domain name (example: https://app.threatconnect.com) |

Use the following header to authenticate to the Management API:

```
Header: statuskey: {{statuskey}}
```

# API Detail

License Service

## Commands

| | |
|---|---|
| **Curl** | curl -X GET -H "statuskey: {{statuskey}}" "{{baseUrl}}/api/v1/management/license" |
| **HTTP** | GET {{baseUrl}}/api/v1/management/license HTTP/1.1 |

## Sample Response

**Status 200 OK**


```
{
  "status": "Success",
  "data": {
    "indicatorLimit": 2147483647,
    "userLimit": 999,
    "organizationLimit": 2147483647,
    "docStorageLimit": 9223372036854775807,
    "apiLimit": 999,
    "taxiiLimit": 999
  }
}
```

## JVM Stats Service

### Commands

| | |
|---|---|
| **Curl** | curl -X GET -H "statuskey: {{statuskey}}" "{{baseUrl}}/api/v1/management/jvm/stats" |
| **HTTP** | GET {{baseUrl}}/api/v1/management/jvm/stats |

### Sample Response

**Status 200 OK**


```
{
  "status": "Success",
  "data": {
    "maxMemoryMb": 629,
    "openFileHandles": 2632
  }
}
```


## JVM Thread Dump Service

### Commands

| | |
|---|---|
| **Curl** | curl -X GET -H "statuskey: {{statuskey}}" "{{baseUrl}}/api/v1/management/jvm/threadDump" |
| **HTTP** | GET {{baseUrl}}/api/v1/management/jvm/threadDump |

### Sample Response

| Status 200 OK |
| --- |
| "Reference Handler"<br>java.lang.Thread.State: RUNNABLE<br>at java.base@11.0.6/java.lang.ref.Reference.waitForReferencePendingList(Native Method)<br>at java.base@11.0.6/java.lang.ref.Reference.processPendingReferences(Reference.java:241)<br>at java.base@11.0.6/java.lang.ref.Reference$ReferenceHandler.run(Reference.java:213)<br>"Finalizer"<br>java.lang.Thread.State: WAITING<br>at java.base@11.0.6/java.lang.Object.wait(Native Method)<br>at java.base@11.0.6/java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:155)<br>at java.base@11.0.6/java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:176)<br>at java.base@11.0.6/java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:170)<br>... |

## Feeds Status Service

### JOB Status Commands

| | |
| --- | --- |
| **Curl** | curl -X GET -H "statuskey: {{statuskey}}" "{{baseUrl}}/api/v1/management/feeds?type=JOB" |
| **HTTP** | GET {{baseUrl}}/api/v1/management/feeds?type=JOB |

### Sample JOB Response

| Status 200 OK |
| --- |
| {<br>"status": "Success",<br>"data": [<br>{ |

```
    "name": "SNP TI",
    "nextRunTime": "2021-03-01T16:00:00.000Z",
    "cronSchedule": "0 0 19-23/2,0-18/2 ? * *",
    "type": "JOB",
    "lastExecutionResponse": {
      "sessionId": "c3c50137-a65b-4fee-9ebe-27f494895998",
      "status": "Detached"
    }
  },
  {
    "name": "Group IB Threat Intelligence v1",
    "nextRunTime": "2021-03-01T16:00:00.000Z",
    "cronSchedule": "0 0 1-23/1 ? * *",
    "type": "JOB",
    "lastExecutionResponse": {
      "sessionId": "4aa91f75-4930-4bfb-8592-5de02194ffcf",
      "status": "Completed",
      "startTime": "2021-02-09T22:43:37.518Z",
      "exitTime": "2021-02-10T04:56:24.096Z"
    }
  },
...
```


## TAXII Status Commands

| | |
|---|---|
| **Curl** | curl -X GET -H "statuskey: {{statuskey}}" "{{baseUrl}}/api/v1/management/feeds?type=TAXII" |
| **HTTP** | GET {{baseUrl}}/api/v1/management/feeds?type=TAXII |

## Response

Status 200 OK


```
{
  "status": "Success",
  "data": [
    {
      "name": "guest.Abuse_ch",
      "nextRunTime": "2016-04-04T20:24:00.000Z",
      "type": "TAXII",
      "lastExecutionResponse": {
        "startTime": "2016-04-03T20:24:00.000Z"
      }
    },
    {
      "name": "Threatconnect",
      "nextRunTime": "2016-04-05T19:59:00.000Z",
      "type": "TAXII",
      "lastExecutionResponse": {
        "startTime": "2016-04-04T19:59:00.000Z"
      }
    },
...
```


# Playbook Activity Stats Service

## Commands

| | |
|---|---|
| **Curl** | curl -X GET -H "statuskey: {{statuskey}}" "{{baseUrl}}/api/v1/management/playbooks/activity/stats" |
| **HTTP** | GET {{baseUrl}}/api/v1/management/playbooks/activity/stats |

## Response

Status 200 OK


```
{
  "status": "Success",
  "data": {
    "queueSize": 0,
    "workerCount": 6,
    "publicServers": [
      {
        "name": "tc-job",
        "specs": {
          "os": "CentOS Linux | GNU/Linux  7 (Core) build 4.14.219-119.340.amzn1.x86_64 ",
          "diskGb": 196,
          "cpuCount": 16,
          "memGb": 61
        },
        "consumerCount": 6
      }
    ],
    "privateServers": [],
    "executionsTotal": 31614,
    "executionsToday": 248,
    "maxExecutionLimit": 10000,
    "brokerConnectionCount": 27
  }
}
```

## Playbook Activity Running Service

### Commands

| | |
|---|---|
| **Curl** | curl -X GET -H "statuskey: {{statuskey}}" "{{baseUrl}}/api/v1/management/playbooks/activity/running" |
| **HTTP** | GET {{baseUrl}}/api/v1/management/playbooks/activity/running |

### Response

**Status 200 OK**


```
{
  "status": "Success",
  "data": [
    {
      "sessionId": "d071342b-5496-4f65-b536-4c4b6afb5958",
      "playbookId": 8876,
      "playbookName": "PB Execution Monitor - DO NOT TOUCH",
      "startTime": "2021-02-28T23:15:00.270Z"
    }
  ]
}
```


## Playbook Activity Workers Service

### Commands

| | |
|---|---|
| **Curl** | curl -X GET -H "statuskey: {{statuskey}}" "{{baseUrl}}/api/v1/management/playbooks/activity/workers" |
| **HTTP** | GET {{baseUrl}}/api/v1/management/playbooks/activity/running |

## Response

Status 200 OK


```
{
  "status": "Success",
  "data": [
    {
      "workerName": "Takodana",
      "serverName": "tc-job",
      "status": "IDLE",
      "lastSessionId": "bb1b3f83-dfe9-411d-a518-f964d97cc3a2",
      "lastActivityTime": "2021-03-01T15:15:00.503Z"
    },
    {
      "workerName": "Kashyyyk",
      "serverName": "tc-job",
      "status": "IDLE",
      "lastSessionId": "16e1ac4e-da0b-42bd-83cf-21b8d57c7cbc",
      "lastActivityTime": "2021-03-01T15:10:00.441Z"
    },
    {
      "workerName": "Polis Massa",
      "serverName": "tc-job",
      "status": "IDLE",
      "lastSessionId": "1dc4daf9-0a10-4f3d-9ab6-d464633ddeb5",
      "lastActivityTime": "2021-03-01T15:00:00.515Z"
    },
    {
      "workerName": "Naboo",
      "serverName": "tc-job",
```

```
      "status": "IDLE",
      "lastSessionId": "55260fb7-b08b-423f-9fca-183d0c2448ad",
      "lastActivityTime": "2021-03-01T15:20:00.426Z"
    },
    {
      "workerName": "Shili",
      "serverName": "tc-job",
      "status": "IDLE",
      "lastSessionId": "4141a4c5-364e-4bf5-8d08-e02c80a5f265",
      "lastActivityTime": "2021-03-01T15:05:00.399Z"
    },
    {
      "workerName": "Jedha",
      "serverName": "tc-job",
      "status": "IDLE",
      "lastSessionId": "14617e4e-ff0b-48a7-8b70-a872cad406b9",
      "lastActivityTime": "2021-03-01T15:17:16.231Z"
    }
  ]
}
```


## Playbook Download Service

### Commands

| | |
|---|---|
| **Curl** | curl -X GET -H "statuskey: {{statuskey}}" "{{baseUrl}}/api/v1/management/playbooks/:groupXid/download"" |
| **HTTP** | GET {{baseUrl}}/api/v1/management/playbooks/:groupXid/download |

### Response

| Status 200 OK |
| --- |
| {<br>"name": "AWS SNS Trigger",<br>"type": "Standard",<br>"panX": 20,<br>"panY": 20,<br>"logLevel": "WARN",<br>"description": "",<br>"version": "1.0",<br>"jobList": [<br>... |

## Playbook Queue Stats Service

### Commands

| | |
| --- | --- |
| **Curl** | curl -X GET -H "statuskey: {{statuskey}}" "{{baseUrl}}/api/v1/management/queue/stats" |
| **HTTP** | GET {{baseUrl}}/api/v1/management/queue/stats |

### Response

| Status 200 OK |
| --- |
| {<br>"status": "Success",<br>"data": [<br>{<br>"queueName": "BatchJobQueue",<br>"queueSize": 0, |

```
    "messagesAdded": 0,
    "messagesProcessed": 0
  },
  {
    "queueName": "DNSMonitorQueue",
    "queueSize": 0,
    "messagesAdded": 0,
    "messagesProcessed": 0
  },
  {
    "queueName": "ElasticSearchQueue",
    "queueSize": 0,
    "messagesAdded": 0,
    "messagesProcessed": 0
  },
  {
    "queueName": "InboundMailQueue",
    "queueSize": 0,
    "messagesAdded": 0,
    "messagesProcessed": 0
  },
  {
    "queueName": "JobExecutionQueue",
    "queueSize": 0,
    "messagesAdded": 0,
    "messagesProcessed": 0
  },
  {
    "queueName": "JobLogRequestQueue",
    "queueSize": 0,
```

```
      "messagesAdded": 0,
      "messagesProcessed": 0
    },
    {
      "queueName": "PlaybookExecutionQueue",
      "queueSize": 0,
      "messagesAdded": 265,
      "messagesProcessed": 265
    }
  ]
}
```


## Unlock User Service

### Commands

| | |
|---|---|
| **Curl** | curl -X POST -H "statuskey: {{statuskey}}" "{{baseUrl}}/api/v1/management/user/:username/unlock" |
| **HTTP** | POST {{baseUrl}}/api/v1/management/user/:username/unlock |

### Response

**Status 200 OK**


```
{
  "status": "Success"
}
```

## Health Check Args Service

### Commands

| | |
|---|---|
| **Curl** | curl -X GET -H "statuskey: {{statuskey}}" "{{baseUrl}}/api/v1/management/health/args" |
| **HTTP** | GET {{baseUrl}}/api/v1/management/health/args |

### Response

**Status 200 OK**


```
{
  "status": "Success",
  "data": [
    {
      "name": "Monthly Playbook Query",
      "args": [
        "enddate",
        "startdate"
      ]
    }
  ]
}
```


## Custom Health Check (Count Playbooks in TRACE)

With the custom health checks, administrators can build queries and pull results via API calls. For example: showing the SQL process list, Playbooks in TRACE mode, or Playbook execution activity. Any SQL script that returns a dataset can be exposed as an API using the custom health-check feature.

***NOTE: Queries can break after an upgrade to the platform, based on database changes. There is no guarantee that ThreatConnect database tables will stay the same between upgrades, which may directly impact health-check queries. This feature is not supported in Dedicated Cloud instances.***

Below is a sample SQL query to build a custom health-check API that returns all Playbooks in TRACE mode.

```
--- Playbooks in TRACE
INSERT INTO systemhealthcheck (name, type, category, description, command,
validation, filterDb, filterScript, hidden)
VALUES ('Playbooks in TRACE', 'DATABASE', 'Database', 'Playbooks in TRACE',
'select count(*) as Playbook_Count from blueprint where status = ''Active''
and loglevel = ''TRACE''', null, null, null, 1);
```

Once the record is inserted into the database, the following API call will retrieve the results directly from the database.

## Commands

| | |
|---|---|
| **Curl** | curl -X POST -H "statuskey: {{statuskey}}" -d '[<br>{<br>"name": "Playbooks in TRACE"<br>}<br>]' "{{baseUrl}}/api/v1/management/health?category=Database&name=Playbooks in TRACE" |
| **HTTP** | POST {{baseUrl}}/api/v1/management/health?category=Database&name=Playbooks in TRACE |

## Response

**Status 200 OK**


```
{
 "status": "Success",
 "data": [
  {
   "category": "Database",
   "name": "Playbooks in TRACE",
   "result": "184",
   "passed": true
```

```
  }
 ]
}
```

## Custom Health Check (Top Table Counts)

This custom health check (sample only) allows administrators to check the largest tables by row count. Follow the steps below to create this health-check API.

***NOTE: Queries can break after an upgrade to the platform, based on database changes. There is no guarantee that ThreatConnect database tables will stay the same between upgrades, which may directly impact health-check queries. This feature is not supported in Dedicated Cloud instances.***

```
-- Top Table Counts
INSERT INTO systemhealthcheck (name, type, category, description, command,
validation, filterDb, filterScript, hidden)
VALUES ('Statement For Counts', 'DATABASE', 'Database', 'Statement For
Counts', 'select (select count(*) from indicator) indicator_count, (select
count(*) from indicatorattribute) indicatorattribute_count, (select count(*)
from indicator_tag) indicator_tag_count, (select count(*) from alert)
alert_count, (select count(*) from jobexecution) jobexecution_count, (select
count(*) from blueprintexecution) blueprintexecution_count from license;',
null, null, null, 1);
```

Once the record is inserted into the database, the following API call will retrieve the results directly from the database.

### Commands

| | |
|---|---|
| **Curl** | curl -X POST -H "statuskey: {{statuskey}}" -d '[<br>{<br>"name": "Statement For Counts"<br>}<br>]' "{{baseUrl}}/api/v1/management/health?category=Database&name=Statement For Counts" |
| **HTTP** | POST {{baseUrl}}/api/v1/management/health?category=Database&name=Statement For Counts |

### Response

Status 200 OK


```
{
  "status": "Success",
  "data": [
    {
      "category": "Database",
      "name": "Top table counts",
      "result": [
        {
          "indicator_count": 16627832,
          "indicatorattribute_count": 38487028,
          "indicator_tag_count": 20003691,
          "alert_count": 1200154,
          "jobexecution_count": 238904,
          "blueprintexecution_count": 31620
        }
      ],
      "passed": true
    }
  ]
}
```


## Status Service

### Commands

| | |
|---|---|
| **Curl** | curl -X GET -H "statuskey: {{statuskey}}" "{{baseUrl}}/status" |
| **HTTP** | GET {{baseUrl}}/status |

## Response

Status 200 OK


```
{
  "Product Version": "6.1.0",
  "DB Status": "OK",
  "HTTP Status": "OK",
  "Filesystem status (JBoss Server Log)": "OK (122861MB remaining)",
  "Filesystem status (Bulk Reports)": "OK (122861MB remaining)",
  "Filesystem status (Local Storage)": "N/A",
  "Filesystem status (TC Server Log)": "OK (122861MB remaining)",
  "Current Time": "2021-03-01T15:44:36.279+0000",
  "Message": "System OK."
}
```