VividCortex

API Documentation

Contents

Interacting with the VividCortex API

Our application is driven entirely via public calls to the VividCortex API. If you are curious as to how something works that is not documented below, you can find out by using the Network Inspector in your browser and view the API calls that are made while interacting with the app.

API Tokens

All requests against the VividCortex API require authentication in the form of an API token. Each environment in an organization has an API token assigned to it for agents to use; they can also be used for making custom API requests. The API token for a given environment can be found in the Add Host Wizard when the environment is selected.

Users

Get the list of Users in an organization

Getting the list of users in your organization can be done with the following request:

```
curl -X GET 'https://app.vividcortex.com/api/v2/users' \
  -H 'Authorization: Bearer <API Token>'
```

Which returns a response in the following format:

```
[
  {
    "id":1,
    "email":"brainiac@vividcortex.com",
    "name":"Test User",
    "status":"0",
    "created":0,
    "updated":1469465681,
    "deleted":0,
    "orgStatus":"0"
  },
  {
    "id":2,
    "email":"robot@vividcortex.com",
    "name":"Robot",
    "status":"1",
    "created":0,
    "updated":1405433595,
    "deleted":0,
    "orgStatus":"0"
  }
]
```

Note: You can determine a user's current state via the `status` field. The `status` field can have a few different values: - 1: The user has been invited but has not logged in yet. - 0: The user is an active member of the organization. - other: The user has been blocked from the organization. This is effectively the same as deleting a user.

Adding a User to an organization

You can add users with the following request:

```
curl -X POST 'https://app.vividcortex.com/api/v2/users' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer <API Token>' \
  --data-binary '{"name":"Test User","email":"brainiac@vividcortex.com"}'
```

Which returns the following response:

```
{
  "id":2274,
  "email":"brainiac@vividcortex.com",
  "name":"Test User",
  "status":"1",
  "created":1441245811,
  "updated":1441245811,
  "deleted":0
}
```

The above request takes a JSON payload with the full name and email address of the person you wish to add to the organization. The user in question will receive an email shortly after inviting them to log in for the first time.

Note: By default, new users are added to the `Everyone` team for the organization, which in turn is not affiliated with any environment by default. In order for a new user to have access to environments in the organization, it will either need to be added to a team that is affiliated with an environment, or the `Everyone` team must already be affiliated with an environment.

Removing a User from an organization

Removing a user is done by sending a request with the ID of the user in question:

```
curl -X DELETE 'https://app.vividcortex.com/api/v2/users/<userID>' \
  -H 'Authorization: Bearer <API Token>'
```

The request does not return a payload response, only a status code indicating whether the removal was successful.

Hosts

Get the list of Hosts in an environment

You can fetch the list of hosts in an environment with the following request:

```
curl 'https://app.vividcortex.com/api/v2/hosts?from=-3600&nest=tags' \
  -H 'Authorization: Bearer <API Token>' \
  -H 'X-Indent: true'
```

Which returns a response in the following format:

```
{
  "data": [
    {
      "id":132,
      "uuid":"3156bbb284211b36cc5bcbb1920b7a01",
      "created":1454340944,
      "updated":1460575464,
      "deleted":0,
      "lastSeen":1474990286,
      "parent":0,
      "name":"test-db",
      "bindings":[],
      "type":"os",
      "description":"Server test-db version 4.2.5-201.fc22.x86_64",
      "status":0,
      "tags":["db-test"],
      "os":"linux",
      "arch":"amd64",
      "discovered":0
    },
    {
      "id":133,
      "uuid":"60321d6f8e9e930a9039e29dbe8adcf6",
      "created":1454781822,
      "updated":1474990181,
      "deleted":0,
      "lastSeen":1474990280,
      "parent":132,
      "name":"test-db",
      "bindings":["[::]:3306"],
      "type":"mysql",
      "description":"MySQL Community Server (GPL) version 5.6.28-log on test-db
listening on [::]:3306",
      "status":0,
      "tags":["db-test"],
      "os":"linux",
      "arch":"amd64",
      "discovered":0
    }
  ]
}
```

The above response lists each active host within the last hour (as specified by the `from` parameter in the request). Note that hosts have different types; the example response is showing a MySQL instance and the linux host that it is running on. You can tell the two hosts apart via the `type` field in the response for each host. Additionally, a database host has a reference to the ID of the OS host that it is running on, via the `parent` field.

Note: If you want to get the details of a specific host, you can do so by specifying `?id=<hostID>` or `?uuid=<hostUUID>` in the request.

Set tags for a given Host

You can change the tags associated with a host with a request like the one below:

```
curl -X PUT 'https://app.vividcortex.com/api/v2/hosts/<hostID>' \
  -H 'Authorization: Bearer <API Token>' \
  -H 'Content-Type: application/json' \
  -d '{"tags":["tag1","tag2"]}'
```

The above request sets the tags for a given host to `["tag1", "tag2"]`, but the `name` and `description` fields can also be set.

Note: If you wish to remove all tags for a given host, you can do so by passing `[]` for the tags field.

Deleting a Host

You can delete a given host with a request like the one below:

```
curl -X DELETE 'https://app.vividcortex.com/api/v2/hosts/<hostID>' \
  -H 'Authorization: Bearer <API Token>'
```

The above request sets the deleted field for a given host, which will shutdown the running agents and free the license it is consuming immediately. If the specified hostID has one or more child hosts, they will also be deleted.

Events

Creating an Event

Creating custom events can be done with a request like the one below:

```
curl -x POST https://app.vividcortex.com/api/v2/events \
  -H 'Authorization: Bearer <API Token>' \
  -H 'Content-Type: application/json' \
  --data-binary \
  '{
      "start": 1341446430,
      "host": 132,
      "duration": 60,
      "type":  "Custom Event",
      "level": "info",
      "message": "Here is a custom event I created!"
  }'
```

The above request will create a custom event with the given parameters: - `start` : The unix timestamp to create the event at. - `host` : The ID of the host to associate the event with. If unspecified, a global event is created. - `duration` : The duration of the event in seconds. Defaults to 0. - `type` : The category to create the event under. - `level` : The severity of the event. This can be `info` , `warn` , or `crit` . Defaults to `info` . - `message` : A description explaining what the event is.

Note: `host` , `duration` , and `level` are all optional, so an event can be created as long as it has a start time, a type and a message.

Get the list of recent Events

Getting a list of recent events can be done with a request like the one below:

```
curl -x GET https://app.vividcortex.com/api/v2/events?
from=-3600&until=0&host=100,101 \
  -H 'Authorization: Bearer <API Token>' \
  -H 'X-Indent: true'
```

The above request will return a response containing the list of events found within the last hour for the host IDs listed in the request, formatted like this:

```
{
  "data": [
    {
      "start": 1474999720,
      "uuid": "c3275df8f07d223fc260451b6b453365",
      "host": 100,
      "type": "Agent Startup",
      "duration": 0,
      "level": "info",
      "digest": "12999091233602395932",
      "message": "vc-os-metrics starting version 1.7.337",
    },
    {
      "start": 1475000395,
      "uuid": "4f03b88bd29188301e5ddaedf26b2428",
      "host": 101,
      "type": "Agent Startup",
      "duration": 0,
      "level": "info",
      "digest": "12999091233602395932",
      "message": "vc-mysql-query starting version 1.7.337",
    }
  ]
}
```

Note: Some events are not associated with any hostID. In order to return them as well, specify `?addGlobal=1` in the request address to indicate that global events should be included.

Get a list of Event Categories

Getting a list of the event categories that have been seen can be done with a request like the one below:

```
curl -x GET https://app.vividcortex.com/api/v2/events/types \
  -H 'Authorization: Bearer <API Token>' \
  -H 'X-Indent: true'
```

The above request will return a response containing the list of all event types that have been created in the given environment:

```
{
  "data": [
    "Agent Startup",
    "Agent Shutdown",
    "New Important Query"
  ]
}
```

Charts

- Charts are defined in VividCortex as a set of collections, with each collection containing one or more chart templates. By default there is a set of pre-defined chart collections in the app for you to use, but you can also define your own collections from the pool of available templates.

Get a List of Chart Collections

Getting a list of the available chart collections for an environment can be done with a request like the one below:

```
curl 'https://app.vividcortex.com/api/v2/charts/collections?ids=cpu,disk' \
  -H 'Authorization: Bearer <API Token>' \
  -H 'Accept: application/json' -H 'X-Indent: true'
```

The above request will return a response containing the list of the chart collections in the environment that match one of the listed ids. If no ids are specified, all chart collections are returned:

```
[
  {
    "id": "cpu",
    "title": "CPU",
    "attributes": {},
    "templateIds": [
      "os-context-switches",
      "os-cpu-ec2-balance",
      "os-cpu-ec2-usage",
      "os-cpu-utilization",
      "os-processes"
    ]
  },
  {
    "id": "disk",
    "title": "Disk",
    "attributes": {},
    "templateIds": [
      "os-disk-io-latency",
      "os-disk-io-ops",
      "os-disk-io-sectors",
      "os-disk-space",
      "os-per-disk-space"
    ]
  }
]
```

Each collection object is defined as having a `title` that is used to identify it in the app, and an `id` for interacting with it in the API. The `templateIds` field lists the ids of each chart template that belongs to the collection.

Get a List of Chart Templates

Getting a list of the available chart templates can be done with a request like the one below:

```
curl 'https://app.vividcortex.com/api/v2/charts/templates?ids=os-context-
switches,os-processes' \
  -H 'Authorization: Bearer <API Token>' \
  -H 'Accept: application/json' -H 'X-Indent: true'
```

The above request will return a response containing the list of chart templates that match one of the listed ids. If no ids are specified, all chart templates are returned:

```
[
  {
    "id": "os-context-switches",
    "title": "Context Switches",
    "layout": "lines",
    "hostType": "os",
    "attributes": {},
    "metrics": [
      {
        "name": "os.cpu.ctxt",
        "attributes": {}
      }
    ]
  },
  {
    "id": "os-processes",
    "title": "Processes",
    "layout": "lines",
    "hostType": "os",
    "attributes": {},
    "metrics": [
      {
        "name": "os.cpu.processes",
        "attributes": {}
      },
      {
        "name": "os.cpu.procs_blocked",
        "attributes": {}
      },
      {
        "name": "os.cpu.procs_running",
        "attributes": {}
      }
    ]
  }
]
```

Similarly to chart collections, each template object is defined as having a `title` that is used to identify it in the app, and an `id` for interacting with it in the API. The `metrics` field lists the individual metrics in VivdCortex that are displayed on the chart represented by the template.

Note: While you can create custom collections of charts to use as dashboards in the app, templates are pre-defined and cannot be modified. If you wish to have a custom chart template created, you will need to submit a feature request by sending a message to support@vividcortex.com.

Creating a custom Chart Collection

You can create your own chart collection with a request similar to the one below:

```
curl -X POST 'https://app.vividcortex.com/api/v2/charts/collections' \
  -H 'Authorization: Bearer <API Token>' -H 'Content-Type: application/json' \
  -data-binary '{
        "title": "My Dashboard",
        "attributes": {},
        "templateIds": [
            "mongo-op-counters",
            "os-cpu-utilization",
            "os-disk-io-ops",
            "os-memory-usage"
        ]
    }'
```

Which gives the following response:

```
{ "id": "my-custom-dashboard" }
```

The above request creates a collection titled “My Dashboard”, with the charts defined by the listed template IDs. The response contains the ID of the created collection, which can be used to retrieve it, or delete it if desired.

Deleting a custom Chart Collection

If you wish to remove a custom collection that you have created, you can do so with a request like the one below:

```
curl -X DELETE 'https://app.vividcortex.com/api/v2/charts/collections/<id>' \
  -H 'Authorization: Bearer <API Token>'
```

The above request will delete a collection with the specified ID. For instance, specifying `my-custom-dashboard` would delete the collection created by the example request for creating a custom collection shown previously.