VOSS Insights
Arbitrator API Guide

Release 23.1

Mar 15, 2023

Legal Information

DOCUMENT ID: 20230315101338

Contents

1. Introduction

1.1. References and Conventions

Interactive documentation may also be found directly on our Arbitrator product at the following url:

`https://<Arbitrator Ip Address>/api/`

The base URL for all routes is the following:

`https://<Arbitrator Ip Address>/api/`

All responses from the API will be in JSON format.

2. Resources

2.1. Resources Overview

The Arbitrator API is broken up into the resources below. Each resource represents an object in the Arbitrator system. A resource will have associated data and a set of methods in which the user may operate on it.

Resource	Description
`/alerts`	This resource will return data associated with alerts generated by the system.
`/assets`	This resource will return data associated with all the assets configured and discovered by the system. An asset can be added using the Arbitrator configuration screen. The Arbitrator will also automatically discover assets if the asset is configured to send logs to the Arbitrator.
`/ciscocdr`	This resource will return data associated with Cisco CDR and CMR files. The API will return a running total of various statistics associated with Cisco's call records. **Note:** As of release 22.2, only one optional parameter: `cm_ip` (Call Manager IP) is supported.
`/system`	This resource will return data about the Arbitrator system in general.

2.2. alerts

The alerts resource supports the following operations.

Method	URL	Description
GET	`alerts`	Get a list of all alerts.
GET	`alerts/{alert_id}`	Get a single alert by alert id.
POST	`alerts/disposition`	Disposition an Alert through API.

2.2.1. GET

`/alerts`

`/alerts/{alert_id}`

- GET Parameters

  `alert_id` - Optional input parameter specifying alert by `alert_id`.

  Example

  `/alerts/104`

- GET Query Parameters

  Query parameters are only used with GET requests and can be appended to the URL with a `?` sign:

  `?reference_id` - Optional query parameter specifying the alert by `reference_id`.

  Example

  `/alerts?reference_id=20000-55000002-00-01-2784-2`

- Response Codes

HTTP Status Code	Reason
200	Success

- Response Body

  AlertResult

- Example Curl Request

  Command with `alert_id`:

```
curl -k -w '\nRESP_CODE: %{response_code}\n'
     -X GET https://10.13.37.12/api/alerts/807
```

  Output:


```
{"alerts":[{
     "ALERTLOG_ID":"807",
     "ALERT_MESSAGE":"Node: tarb(127.0.0.1) - Alert 1 : User (admin) : Device (tarb)
↪",
     "STATMON_ID":"1",
     "STATMON_LIFE_ID":"1",
     "CYCLE_NUM":"0",
     "LOG_DATE":"1485531000",
     "LAST_ESC_DATE":"1485534603",
     "ACK_DATE":"1486064927",
     "LAST_UPDATED":"0",
     "ACK_LEVEL":-1,
     "PCOUNTER":1,
     "ASC_ID":11,
     "IRP_ID":"1",
     "IRP.IRP_NAME":"Default IRP (1626791390)",
     "IRS_ID":"107",
```


(continues on next page)

(continued from previous page)


```
     "AD_ID":2,
     "AD.AD_DESCRIPTION":null,
     "REFERENCE_ID":"20000- 55000002-00-01-2785-4",
     "USER_NAME":"admin",
     "NODE":"tarb",
     "SHORT_MESSAGE":"Alert 1 : User (admin) : Device (tarb)",
     "RULE_NAME":"Alert  1",
     "POLICY_NAME":"Kenny",
     "DISPOSITION_SCRIPT":".\/scripts\/disptest.php",
     "DISPOSITION_CONFIG":""}]
 }
```


RESP_CODE: 200

Command with reference_id:

```
curl -k -w '\nRESP_CODE: %{response_code}\n'
     -X GET https://10.13.37.12/api/alerts?
              reference_id=20000-55000002-00-01-2785-4
```

Output:


```
{"alerts":[{
     "ALERTLOG_ID":"807",
     "ALERT_MESSAGE":"Node: tarb(127.0.0.1) - Alert 1 : User (admin) : Device (tarb)
↪",
     "STATMON_ID":"1",
     "STATMON_LIFE_ID":"1",
     "CYCLE_NUM":"0",
     "LOG_DATE":"1485531000",
     "LAST_ESC_DATE":"1485534603",
     "ACK_DATE":"1486064927",
     "LAST_UPDATED":"0",
     "ACK_LEVEL":-1,
     "PCOUNTER":1,
     "ASC_ID":11,
     "IRP_ID":"1",
     "IRP.IRP_NAME":"Default IRP (1626791390)",
     "IRS_ID":"107",
     "AD_ID":2,
     "AD.AD_DESCRIPTION":null,
     "REFERENCE_ID":"20000- 55000002-00-01-2785-4",
     "USER_NAME":"admin",
     "NODE":"tarb",
     "SHORT_MESSAGE":"Alert 1 : User (admin) : Device (tarb)",
     "RULE_NAME":"Alert  1",
     "POLICY_NAME":"Kenny",
     "DISPOSITION_SCRIPT":".\/scripts\/disptest.php",
     "DISPOSITION_C  ONFIG":""}]
}
```


RESP_CODE: 200

2.2.2. POST

- POST Parameters

  The alerts POST API only accepts parameters form. The request `application/x-www-form-urlencode` body should be in the following format:

  `reference_id=value&disposition=14&username=value`

  The required fields are the following:

Field name	Type	Description
`reference_id`	Text	The reference id of the alert that needs to be dispositioned.
`disposition`	Number	The valid values are 0, 1, 2, 3, 4, 14, 15. Please see disposition table for description.
`username`	Text	The username making the request. Used only for tracking and logging.

- Disposition Description

Value	Description
0	Open.
1	Under Review.
2	Acknowledged.
3	Release.
4	Disregarded. This will delete Alert from system.
14	Closed.
15	Closed and Locked. This will delete Alert from system.

- Example Curl Request

```
curl -k -w '\nRESP_CODE: %{response_code}\n'
     -X POST https://10.13.37.14/api/alerts/disposition
     -d 'reference_id=10000-01000009-00-01-4607- 4&disposition=2&username=testuser'
```

Output:

```
{"alerts":{
    "ok":true,
    "data":{
       "ALERTLOG_ID":117760,
       "ALERT_MESSAGE":"Node: tarb50.14(10.13.37.14) - Tcritical : Severity
↪(critical)",
       "STATMON_ID":1,
       "STATMON_LIFE_ID":1,
       "CYCLE_NUM":0,
       "LOG_DATE":1571839153,
       "LAST_ESC_DATE":1571842756,
       "ACK_DATE":1571856666,
```

(continues on next page)

(continued from previous page)


```
        "LAST_UPDATED":0,
        "ACK_LEVEL":-1,
        "PCOUNTER" :1,
        "ASC_ID":11,
        "IRP_ID":2,
        "IRP.IRP_NAME":"2 IRP (1626791391)",
        "IRS_ID":5,
        "AD_ID":"2",
        "AD.AD_DESCRIPTION":null,
        "REFERENCE_ID":"10000-01000009-00-01- 4607-4",
        "USER_NAME":null,
        "NODE":"tarb50.14",
        "SHORT_MESSAGE":"Tcritical : Severity  (critical)",
        "RULE_NAME":"Tcritical",
        "POLICY_NAME":"Touy",
        "DISPOSITION_SCRIPT":"",
        "DISPOSITION_CONFIG":""},
    "query":""}}
```


RESP_CODE: `200`

2.3. assets

The assets resource supports the following operations.

Method	URL	Description
GET	`/assets`	Get a list of all assets.
GET	`/assets/{asset_id}`	Get a single asset by asset id.
POST	`/assets`	Add a new asset.
POST	`/assets`	Add multiple new assets.
PUT	`/assets`	Updates a single asset.
PUT	`/assets`	Updates multiple assets.
DELETE	`/assets`	Deletes a single asset.
DELETE	`/assets`	Deletes multiple assets.

2.3.1. GET

`/assets`

`/assets/{asset_id}`

- GET Parameters

  `asset_id` - Optional input parameter specifying the specific desired asset.

  `/assets/656`

- Response Codes

HTTP Status Code	Reason
200	Success

- Example Response

```
curl -kX GET https://10.13.37.12/api/assets/169


{
  "assets": [
   {
      "asset_id": "169",
      "asset_name": "10.13.37.1",
      "ipaddress": "10.13.37.1",
      "hostname": "10.13.37.1",
      "customer": "",
      "last_method": "snmp_trap",
      "last_byte_time": "1510782169"
   }
  ]
}
```

2.3.2. POST, PUT, DELETE Requests and Responses

Assets can be created, updated, and deleted using POST, PUT, and DELETE respectively. The same request parameter is required for all three requests.

This section will describe the desired request parameter and resulting response output.

- Request Parameters

  The `AssetInput` object describes the JSON object that can be used to modify a single Asset. The `AssetInputArray` object describes the JSON object that can be used to modify multiple Assets.

  Note: The names `AssetInput` and `AssetInputArray` will be referenced both in this documentation as well as the online interactive documentation that ships with the product.

- `AssetInput`

  The following is an example of the required parameter for creating or modifying single Asset.

  Note: The property `asset_id` is a required parameter for PUT (updates) and DELETE. It is optional and ignored for POST requests. The `asset_id` will be given as part of the response for a successful POST (insert) of an asset.

  The API does not support the use of `ipaddress` as the property for updates and deletes. The property is ambiguous because our system allows duplicate `ipaddress` to be inserted into the system. This gets even more complicated when we include the `customer` property in the discussion. In order to prevent updates and deletes to assets with duplicate IP addresses, this is a strict requirement for our API.

```
{
  "ipaddress": "(string) Ip address of asset.",
  "asset_id": "(long): Optional parameter on POST. Required parameter for PUT and␣
↪DELETE.",
  "hostname": "(string, optional): Hostname of the asset.",
  "did": "(string, optional): EM7 specific. Represents EM7 device id.",
  "mac_address": "(string, optional): Mac address of the asset.",
  "address": "(string, optional): Physical address of the asset.",
  "model": "(string, optional): Model of device.",
  "version": "(string, optional): Version of device.",
  "manufacturer": "(string, optional): Manufacturer of device.",
  "timezone": "(string, optional): Timezone of device location.",
  "description": "(string, optional): Description of device".,
  "asset_name": "(string, optional): Name or alias of device.",
  "customer": "(string, optional): Customer name of device.",
  "site": "(string, optional): Site location name where device resides."
}
```

- `AssetInputArray`

  The following is an example of the required POST parameter for adding multiple Assets.

Note: The property `asset_id` is a required parameter for PUT (updates) and DELETE. It is optional and ignored for POST requests. The `asset_id` will be given as part of the response for a successful POST(insert) of an asset.

```
[
 {
   "ipaddress": "(string) Ip address of asset.",
   "asset_id": "(long): Optional parameter on POST. Required parameter for PUT and␣
↪DELETE.",
   "hostname": "(string, optional): Hostname of the asset.",
   "did": "(string, optional): EM7 specific. Represents EM7 device id.",
   "mac_address": "(string, optional): Mac address of the asset.",
   "address": "(string, optional): Physical address of the asset.",
   "model": "(string, optional): Model of device.",
   "version": "(string, optional): Version of device.",
   "manufacturer": "(string, optional): Manufacturer of device.",
   "timezone": "(string, optional): Timezone of device location.",
   "description": "(string, optional): Description of device".,
   "asset_name": "(string, optional): Name or alias of device.",
   "customer": "(string, optional): Customer name of device.",
   "site": "(string, optional): Site location name where device resides."
 },
 {
   "ipaddress": "(string) Ip address of asset.",
   "asset_id": "(long): Optional parameter on POST. Required parameter for PUT and␣
↪DELETE.",
   "hostname": "(string, optional): Hostname of the asset.",
   "did": "(string, optional): EM7 specific. Represents EM7 device id.",
   "mac_address": "(string, optional): Mac address of the asset.",
   "address": "(string, optional): Physical address of the asset.",
```

(continues on next page)

(continued from previous page)

```
   "model": "(string, optional): Model of device.",
   "version": "(string, optional): Version of device.",
   "manufacturer": "(string, optional): Manufacturer of device.",
   "timezone": "(string, optional): Timezone of device location.",
   "description": "(string, optional): Description of device".,
   "asset_name": "(string, optional): Name or alias of device.",
   "customer": "(string, optional): Customer name of device.",
   "site": "(string, optional): Site location name where device resides." }
]
```

- Response Code

Note: Delete will only return a response code.

`200`

- Response Body

Below is an example of successful Insert or Update. Result may contain one or many assets. The `asset_id` must be saved by the application in order do make any subsequent Updates or Delete requests.

`AssetResult`

```
{
  "assets":
  [
   {
     "asset_id":"656",
     "asset_name":"",
     "ipaddress":"192.168.1.1",
     "hostname":"test hostname",
     "customer":"Customer1",
     "last_method":"",
     "last_byte_time":"0"
   }
  ]
}
```

- Other Possible Responses

Response Code	Reason	Remedy
200	Success	
400	Exception: UPDATE error: `asset_id` property required.	Include `asset_id` in request body
400	Exception: UPDATE error: Received null asset input.	See `AssetInput`
400	Exception: DELETE: error: `asset_id` property required.	Include `asset_id` in request body
400	Exception: DELETE error: Received null asset input..	See `AssetInput`

2.3.3. POST

`/assets`

- Example Curl Request

  Command:

```
curl -k -w '\nRESP_CODE: %{response_code}\n'
     -X POST https://10.13.37.12/api/assets
     - d'{"did": "1234",
          "hostname":"test hostname",
          "ipaddress":"192.168.100.1",
          "alias":"alias  natted ip address",
          "model":"test model",
          "version":"test version",
          "manufacturer":"test  vendor",
          "timezone":"test timezone",
          "description":"test description",
          "address":"test  physical address",
          "customer":"Test Customer1",
          "site":"Test Site1"}'
```

  Output:

```
{"assets":  [
  {"asset_id":"657",
   "asset_name":"",
   "ipaddress":"192.168.100.1",
   "hostname":"test  hostname",
   "customer":"Test Customer1",
   "last_method":"","last_byte_time":"0"}
  ]}
```

  RESP_CODE: `200`

2.3.4. PUT

`/assets`

- Example Curl Request

Note: `asset_id` was included.

Command:

```
curl -k -w '\nRESP_CODE: %{response_code}\n'
     -X PUT https://10.13.37.12/api/assets
     - d'{"asset_id":"657",
          "did": "1234",
          "hostname":"test hostname2",
          "ipaddress":"192.168.100.1",
          "alias":"alias natted ip address",
          "model":"test  model",
          "version":"test version",
          "manufacturer":"test vendor",
          "timezone":"test  timezone",
          "description":"test description",
          "address":"test physical address",
          "customer":"Test Customer1",
          "site":"Test Site1"}'
```

Output:

```
{"assets":
  [{"asset_id":"657",
    "asset_name":"",
    "ipaddress":"192.168.100.1",
    "hostname":"test  hostname2",
    "customer":"Test Customer1",
    "last_method":"",
    "last_byte_time":"0"}]
}
```

RESP_CODE: `200`

2.3.5. DELETE

`/assets`

Command:

```
curl -k -w '\nRESP_CODE: %{response_code}\n'
     -X DELETE https://10.13.37.12/api/assets
     - d'{"asset_id":"657",
          "did": "1234",
          "hostname":"test hostname2",
          "ipaddress":"192.168.100.1",
```

(continues on next page)

(continued from previous page)

```
        "alias":"alias natted ip address",
        "model":"test  model",
        "version":"test version",
        "manufacturer":"test vendor",
        "timezone":"test  timezone",
        "description":"test description",
        "address":"test physical address",
        "customer":"Test Customer1",
        "site":"Test Site1"}'
```

Output:

RESP_CODE: 200

2.4. ciscocdr

The ciscocdr resource supports the following operations.

Method	URL	Description
GET	/ciscocdr	Get running totals for ciscocdr stats.

2.4.1. GET

- Query Parameters

  Query parameters are only used with GET requests and can be appended to the URL with a ? sign.

  cm_ip - Optional query parameter. If provided, the total stats returned will only be for the specific Call Manager.

  Note: cm_ip can be an actual IP address or a free form string.

  Example:

  /ciscocdr?cm_ip=10.13.37.11

  /ciscocdr?cm_ip=customer_a

- Response Codes

HTTP Status Code	Reason
200	Success

- Example Output

  Command:

```
curl -k -w '\nRESP_CODE: %{response_code}\n'
     -X GET https://10.13.37.12/api/ciscocdr
```

Output (abbreviated):


```
{"ciscocdrs":
  {"call_stats":{"total_call_attempts":9003512,
      "total_audio_calls":9002735,
      "total_video_calls":777,
      "total_conferences":15783,
      "total_audio_conferences":15783,
      "total_video_conferences":0,
      "tota  l_abandoned_calls":877955,
      "total_completed_calls":14103104,
      "total_connected_calls":576162,
      "total_failed_calls":1066745,
      "total_processed_calls":8938909,
      "total_rejected_calls":  64603,
      "total_short_calls":19181,
      "total_minutes":34970017.2,
      "total_video_minutes":9132.3833333333,
      "total_audio_minutes":32397357.666667,
      "lower_timestamp":0,
      "upper_timestamp":0,
      "average_hold_time":718,
      "erlangs":42679.134722222,
      "grade_of_service":0,
      "call_failure_ratio":0,
      "total_mobile_calls":7,
      "total_mobile_orig_calls":0,
      "total_mobile_dest_calls":7,
      "total_mobile_minutes":0.45,
      "total_mobile_orig_minutes":266.91666666667,
      "total_mobile_dest_minutes":266.91666666667,
      "total_split_calls":103485,
      "total_split_minutes":214988.7} ,
```

```
"mos_stats":{
     "excellent":{"count":0, "duration":0},
     "good":{"count":2601813, "duration":82398665},
     "fair":{"count":1873602, "duration":74861679},
     "poor":{"count":14932, "duration":2273635},
     "bad":{"count":0, "duration":0},
     "unknown":{"count":2474100, "duration":19864626}
    },
```

```
"metric_stats":{"numberPacketsSent":{"avg":12087.783361046, "count":6920916, "max
↪":47033496, "min":2, "sum":83658533268},
    "numberOctetsSent":{"avg":1217508.3804676, "count":6920916, "max":740301392,
↪"min":-667814800, "sum":842627323 0512},
    "numberPacketsReceived":{"avg":11824.050579865, "count":6920916, "max":2375697,
↪"min":1, "sum":81833260843},
    "numberOctetsReceived":{"avg":1416922.0078419, "count":6920916, "max":2147483647,
↪ "min":32, "sum":9806398194825},
    "numberPacketsLost":{"avg":48407.375317516, "count":6920916, "max":2147483647,
↪"min":-65536, "sum":335023378353 },
```


(continues on next page)

(continued from previous page)


```
    "jitter":{"avg":318.44610438849, "count":6920916, "max":2147483647, "min":0,
↪"sum":2203938739},
    "latency":{"avg":60.928584742251, "count":6920916, "max":16383, "min":0, "sum
↪":421681617},
    "mos":{"avg":2.686694992065, "count":6920916, "max":4.5, "min":0, "sum":18594390.
↪357703},
    "cumalitveConcealRatio":{"avg":0.00081466591994468, "count":6920916, "max":0.
↪9995, "min":0, "sum":5638.2343999998},
    "intervalConcealRatio":{"avg":0.0015669875201491, "count":6920916, "max":1, "min
↪":0, "sum":10844.989},
    "intervalConcealRatioMax":{"avg":0.011221271880773, "count":6920916, "max":1,
↪"min":0, "sum":77661.480099994},
    "concealSeconds":{"avg":3.616535874731, "count":6920916, "max":37719, "min":0,
↪"sum":25029741},
    "severelyConcealSeconds":{"avg":1.594539219953, "count":6920916, "max":37094,
↪"min":0, "sum":11035672},
    "numberVideoPacketsSent":{"avg":0,"count":6920916,"max":17324840,"min":0,"sum":0}
↪,
    "numberVideoOctetsSent":{"avg":0,"count":6920916,"max":1970765368,"min":0,"sum
↪":0},
    "numberVideoPacketsReceived" :{"avg":0,"count":6920916,"max":7696079,"min":0,
↪"sum":0},
    "numberVideoOctetsReceived":{"avg":0,"count":6920916,"max":3822641129,"min":0,
↪"sum":0},
    "numberVideoPacketsLost":{"avg":0,"count":6920916,"max":4294967292,"min":0,"sum
↪":0},
    "videoAverageJitter":{"avg":0,"count":6920916,"max":139,"min":0,"sum":0},
    "videoRoundTripTime":{"avg":0,"count":6920916,"max":1608,"min":0,"sum":0},
    "videoOneWayDelay":{"avg":0,"count":6920916,"max":0,"min":0,"sum":0},
    "cmr_duration":{"avg":0.042237617101551,"count":6920916,"max":43199,"min":0,"sum
↪":292323},
    "videoDuration_channel2":{"avg":0,"count":6920916,"max":0,"min":0,"sum":0},
    "numberVideoPacketsSent_channel2":{"avg":0,"count":6920916,"max":0,"min":0,"sum
↪":0},
    "numberVideoOctetsSent_channel2":{"avg":0,"count":6920916,"max":0,"min":0,"sum
↪":0},
    "numberVideoPacketsReceived_channel2" :{"avg":0,"count":6920916,"max":0,"min":0,
↪"sum":0},
    "numberVideoOctetsReceived_channel2":{"avg":0,"count":6920916,"max":0,"min":0,
↪"sum":0},
    "numberVideoPacketsLost_channel2":{"avg":0,"count":6920916,"max":0,"min":0,"sum
↪":0},
    "videoAverageJitter_channel2":{"avg":0,"count":6920916,"max":0,"min":0,"sum":0},
    "videoRoundTripTime_channel2":{"avg":0,"count":6920916,"max":0,"min":0,"sum":0},
    "videoOneWayDelay_channel2":{"avg":0,"count":6920916,"max":0,"min":0,"sum":0}},
```

```
    "termination_stats":{"17":{"count":190713,"description":"User busy","duration
↪":2581},
        "-1593835503":{"count":9,"description":"Invalid termination code(-
↪1593835503)","duration":0},
        "1174405137":{"count":3889,"description":"CCM_SIP_486_BUSY_HERE (Cisco ␣
↪specific)","duration":0},
```


(continues on next page)

(continued from previous page)


```
        "111":{"count":2109,"description":"Protocol  error","duration":271454},
        "47":{"count":55696,"description":"Resource  unavailable","duration
↪":2424315},
        "458752":{"count":167,"description":"Drop any  party\/drop last party␣
↪(Cisco specific)","duration":35490},
        "42":{"count":931,"description":"Switching equipment congestion","duration
↪":1998},
        "127":{"count":828,"description":"Interworking","duration":2},
        "-1543503841":{"count":9,"description":"Invalid termination code(-
↪1543503841)","duration":0},
        "82":{"count":135,"description":"Identified channel does not exist",
↪"duration":0},
        "393216":{"count":325690,"description":"Call split (Cisco specific)",
↪"duration":16263385},
        "63":{"count":14,"description":"Service or option not available","duration
↪":150},
        "29":{"count":2,"description":"Facility rejected","duration":268},
        "27":{"count":18772,"description":"Destination out of order","duration
↪":4401},
        "41":{"count":63192,"description":"Temporary failure","duration":4201087},
        "28":{"count":139,"description":"Invalid number format (address incomplete)
↪","duration":378},
        "57":{"count":2,"description":"Bearer capability not  authorized","duration
↪":282},
        "1":{"count":401406,"description":"Unallocated (unassigned)  number",
↪"duration":3855},
        "31":{"count":3652,"description":"Normal","duration":6},
        "50":{"count":83,"description":"Requested facility not subscribed",
↪"duration":0},
        "16":{"count":7746066,"description":"Normal call clearing","duration
↪":2081913388},
        [...]
        },
    "elapsed_time":0.048560857772827
    }


RESP_CODE: ``200``
```


2.5. system

The system resource supports the following operations.

Method	URL	Description
GET	/system/stats	Get system stats.

2.5.1. GET

- Example Output

  Command:

```
curl -k -w '\nRESP_CODE: %{response_code}\n'
     -X GET  https://10.13.37.12/api/system/stats
```

  Output:


```
{"data":
  {"cpu_idle":"14",
   "cpu_cores":"8",
   "mem_total":"12338068",
   "mem_used":"702032",
   "mem_free":  "11636036",
   "mem_used_percent":"6.000",
   "diskspace_total":"481419520",
   "diskspace_used":"3  55646212",
   "diskspace_used_percent":"74.000",
   "diskspace_free":"125773308",
   "diskspace_free_percent":"26.000",
   "system_load":"27.250",
   "total_process_threads":"0",
   "sum_cdrs_unprocessed":"0",
   "sum_cmrs_unprocessed":"0",
   "sum_total_unprocessed":"0",
   "customer":"TARB",
   "hostname":"tarb",
   "version":"4.0001-14m",
   "services": {"postgres":"running",
      "ndxserver":"running",
      "scdtsd":"running",
      "apache":"running",
      "sshd  ":"running",
      "slapd":"running"}}
}
```


  RESP_CODE: 200

3. Resources v2

3.1. Resources v2 Overview

The following URIs will require authentication. LayerX currently implements a token based authentication system. Every resource under the v2 route requires a token property to be set in the HTTP header. A token can be requested from the `/v2/login` URI.

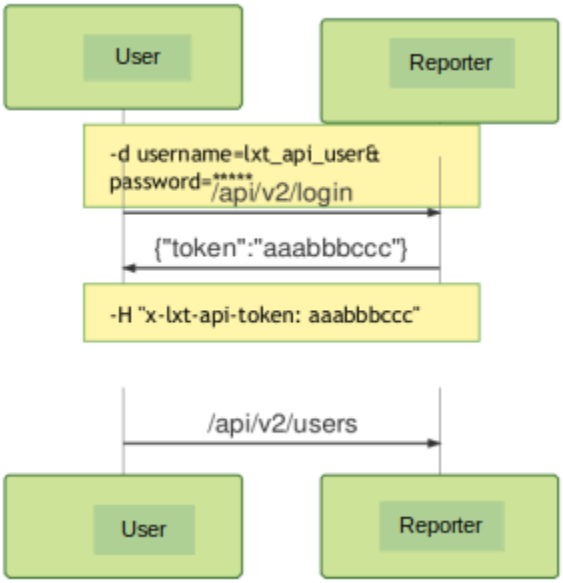
Resource	Description
`/v2/alerts`	This resource will return data associated with alerts generated by the system.
`/v2/login`	Use this resource to request a token.
`/v2/lxt_updates`	Use this resources to manage software updates for the product.
`/v2/users`	Use this resource to manage users.
`/v2/configs`	Use this resource to manage Arbitrator configuration items.

3.2. `/v2/login`

The `login` route is required before access to any route under `v2` is requested. The system will respond with a `token` that needs to be included in the header of all subsequent API requests. The following methods are supported for the `login` route.

Method	URI	Description
POST	`/v2/login`	Retrieves dashboard data for a specific user.

3.2.1. High Level login API Flow



3.2.2. POST

`/v2/login`

- Required Parameters

  The login request requires a `username` and `password` parameter to be sent as part of the POST request.

  Note: the `username` and `password` should be sent as a multipart form parameter. The username should be a userid that already exists in the system. A user can be added through our User Interface or via the API.

  By default, the system contains a user named `lxt_api_user`. This userid can be used for first time API users. The `lxt_api_user` password is set at install time by your system administrator.

- Example Curl Request

  Command:

```
curl -k -w '\nRESP_CODE: %{response_code}\n'
     -X POST https://<IP or FQDN>/api/v2/login
     -d"username=lxt_api_user&password=password1"
```

  Output (truncated):

```
{"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9[...]"}
```

  RESP_CODE: `200`

3.3. /v2/alerts

The alerts resource supports the following operations.

Method	URL	Description
GET	/v2/alerts	Get a list of all alerts.
GET	/v2/alerts/{alert_id}	Get a single alert by alert id.
POST	/v2/alerts/disposition	Disposition an Alert through API.

3.3.1. GET

/v2/alerts

/v2/alerts/{alert_id}

- Example

```
https://<host>/api/v2/alerts?limit=3

{
  "status": 200,
  "message": "Success",
  "data": [
    {
      "alert_id": "<alert_ID>",
      "message": "Node: () - RayRule : Jabber (JabberNew)",
      "log_date": 1665689762,
      "last_escalated_date": 1665689763,
      "acknowledged_date": 0,
      "last_updated": 0,
      "acknowledged_level": 1,
      "status_condition_id": 12,
      "status_condition": "Critical",
      "irp_id": 2,
      "irp_name": "Ray IRP",
      "ad_id": 0,
      "disposition": "The alert is open.",
      "reference_id": "119012-01000011-00-01-101-1",
      "user_name": "",
      "node": "",
      "short_message": "RayRule : Jabber (JabberNew)",
      "rule_name": "RayRule",
      "policy_name": "RayPolicy",
      "response_procedures": [
        {
          "response_procedure_id": "<resp_ID>",
          "name": "Ray IRP"
        }
      ]
    },
```

(continues on next page)

(continued from previous page)


```
    {
      "alert_id":  "<alert_ID>",
      "message": "Node: () - RayRule : Jabber (JabberNew)",
      "log_date": 1665689760,
      "last_escalated_date": 1665689761,
      "acknowledged_date": 0,
      "last_updated": 0,
      "acknowledged_level": 1,
      "status_condition_id": 12,
      "status_condition": "Critical",
      "irp_id": 2,
      "irp_name": "RayIRP",
      "ad_id": 0,
      "disposition": "The alert is open.",
      "reference_id": "119012-01000011-00-01-101-1",
      "user_name": "",
      "node": "",
      "short_message": "RayRule : Jabber (JabberNew)",
      "rule_name": "RayRule",
      "policy_name": "RayPolicy",
      "response_procedures": [
        {
          "response_procedure_id": "<resp_ID>",
          "name": "RayIRP"
        }
      ]
    }
  ]
}
```


RESP_CODE: 200

3.4. /v2/lxt_updates

Method	URL	Description
GET	/v2/lxt_updates	Retrieves current list of all update requests.
GET	/v2/lxt_updates/{id}	Retrieves information about a specific update request.
POST	/v2/lxt_updates	Adds a new update request.
PUT	/v2/lxt_updates	Modifies an existing update request.
PUT	/v2/lxt_updates/{id}	Modifies an existing update request.
DELETE	/v2/lxt_updates	Deletes an existing update request.
DELETE	/v2/lxt_updates/{id}	Deletes an existing update request.

3.4.1. Header (required)

x-lxt-api-token: "token from login"

3.4.2. GET

/v2/lxt_updates

- Example 1: Get All Updates

  Command:

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -X GET https://<IP or FQDN>/api/v2/lxt_updates
```

  Output (formatted):

```
{
 "status":200,
 "message":"Success",
 "data":[
  {"id":"12"},
  {"id":"13"}
 ]
}
```

  RESP_CODE: 200

- Example 2: Get Updates with specified id

  Command:

```
curl -s
-H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9[...]
--insecure
-w "RESP_CODE: %{response_code}"
-X GET https://<IP or FQDN>/api/v2/lxt_updates/12
```

  Output (formatted):

```
  {
    "status":200,
    "message":"Success",
    "data":[
      {
      "id":"12",
      "status":{"comment":"Error encountered. Please reference the install log."},
      "log":"This file does not look like a service pack.\n"
      }
    ]
```

(continues on next page)

(continued from previous page)

```
  }


RESP_CODE: ``200``
```

3.4.3. POST

Use POST to add a new update request.

/v2/lxt_updates

- Input defines

  The following definitions may be used when creating a new request. The new software should be copied into the drop account, or else provide a URL for fetching.

```
id=12 (optional)
delay=60 (optional)
url=http://www.layerxtech.com/downloads/arbitratorhawaii/updates (optional)
```

- Example 1: Add new update request

  Command:

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -X POST https://<IP or FQDN>/api/v2/lxt_updates
```

  or

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1Ni[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d 'id=12'
     -d 'delay=60'
     -d 'url=http://www.layerxtech.com/downloads/arbitratorhawaii/updates'
     -X POST https://<IP  or FQDN>/api/v2/lxt_updates
```

  Output (formatted):

```
{
 "status":200,
 "message":"Success",
 "data":[{"id":"12"}]
}
```

  RESP_CODE: 200

3.4.4. PUT

Use PUT to modify an existing update request. An error will be returned if the update request does not exist in the system.

/v2/lxt_updates

- Input defines

  The following definitions may be used when creating a new request.

```
id=12 (optional)
delay=60 (optional)
url=http://www.layerxtech.com/downloads/arbitratorhawaii/updates (optional)
```

- Example 1: Update existing update request

  Command:

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d'id=12&delay=0'
     -X PUT https://<IP or  FQDN>/api/v2/lxt_updates
```

  or

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d 'delay=0'
     -X PUT https://<IP or FQDN>/api/v2/users/12
```

  Output (formatted):

```
{
 "status":201,
 "message":"Success",
 "data":{ "id":"12" }
 }
```

  RESP_CODE: 200

3.4.5. DELETE

/v2/lxt_updates

/v2/lxt_updates/{id}

- Example 1: Delete existing update request

  Command:

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d"id=12"
     -X DELETE https://<IP or FQDN>/api/v2/lxt_udpates
```

or

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -X DELETE https://<IP or FQDN>/api/v2/lxt_updates/12
```

Output (formatted):

```
  {
  "status":201,
  "message":"Success",
  "data":[]
  }

RESP_CODE: ``200``
```

- Example 2: Delete existing update request again

  Command:

```
curl -s
     -H x-lxt-apitoken:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d"id=12"
     -X DELETE https://<IP or FQDN>/api/v2/lxt_updates
```

  or

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -X DELETE https://<IP or FQDN>/api/v2/lxt_updates/12
```

  Output (formatted):

```
{
  "status":404,
  "message":"Could not find existing entry for {id}",
  "data":[]
}
```

  RESP_CODE: 404

3.5. /v2/configs

The Configs API exists to replicate the browser's Lxtconfig portion of the User Interface (UI). A knowledge of the UI will greatly aid in the understanding of the Config API.

There are several objects that may be configured via the API. As a general rule, the full set of GET, POST, PUT, and DELETE calls are supported for all of the objects. JSON objects are used for the API requests and response.

Note: For security reasons, GET of the Credentials will not return the username/password associated with the credential.

3.5.1. Naming, `object_id`

While the objects contain their own attributes, they all share `object_id`, where the `object_id` matches the object type. The `object_id` is a Universally Unique IDentifier (UUID) that allows identification for API invocations. Because the identifier is universal, it may be reused on POST calls to allow coordination between different arbitrators. (This is useful for generalizing Probes, as well as the localhost Asset.)

The object_id may also be used to perform actions on a specific entry

`https:///api/v2/configs/asset_groups`

- Create asset_groups using a supplied array
- Retrieve all asset_groups
- Update asset_groups using a supplied array
- Delete asset_groups using a supplied array

  `https:///api/v2/configs/asset_groups/object_id`
- Create a single named asset_group
- Retrieve a specific asset_group
- Update the specified asset_group
- Delete the specified asset_group

3.5.2. Relationships between object types

In addition to the objects' attributes, the relationship between the objects can be seen via the presence of arrays entries for the other objects. For example, an Asset Group will contain an array named "assets" with 0..n entries for its children assets.

Generally speaking, the objects follow a parent-child relationship. The relationships are m:n, which means the same object can share a parent or child object with its peer objects. For instance, the same Probe Group can be used with multiple Assets.

```
Configs
Asset Group
+> [Asset Group]
-> Assets
---> Profiles
-----> Credentials
-----> Probe Groups
-------> Probes
Policy Modules
-> Policy Filters
-> Correlation Rules
---> Response Procedures
-----> Response Methods+Details
-------> Controls
```

When fetching data for a parent object, its children will be included. An entry for the parent object will also be present, but only the parent's UUID will be supplied.

Note: For large configurations, timeout restrictions may prevent fetching an entire object group.

Developer Hint: Performing a cut-paste-replace of an existing object is the fastest way to create a new entry.

3.6. /v2/configs/asset_groups

Method	URL	Description
GET	/v2/configs/asset_groups	Retrieves current list of all asset_groups and all children underneath.
POST	/v2/configs/asset_groups	Adds a new asset_groups and all children.
PUT	/v2/configs/asset_groups	Modifies an existing asset_groups and all children.
DELETE	/v2/configs/asset_groups	Deletes an existing asset_groups.

Note: asset_group_id of 0 is reserved for the "Ungrouped" Asset Group

3.6.1. Header (required)

x-lxt-api-token: "token from login"

3.6.2. GET

/v2/configs/asset_groups

- Example 1: Get All Asset Groups

  Command:

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -X GET https://<IP or FQDN>/api/v2/configs/asset_groups
```

  Output (formatted):

```
{
 "status":200,
 "message":"Success",
 "data":[
  {"asset_group_id": 0,
   "name": "Ungrouped",
   "assets": [{
      "asset_id":"07a3525a54cacf44d00b220a77a3d16046[...]",
      "name": "Local System",
      "ipaddress": "10.13.37.172",
      "hostname": "RayHawaiiArb2",
      "customer": "",
      "last_method": "",
      "last_byte_time": 0,
      "did": "Unknown",
      "mac_address": "Unknown",
      "address": "",
      "version": "Unknown",
      "manufacturer": "LayerX Technologies",
      "timezone": "UTC",
      "description": "Local Arbitrator Platform",
      "model": "Unknown",
      "asset_groups": [],
      "assets": [],
      "profiles": []
      },
      {
      "asset_id":"3c7bada1022172641f5e4e319[...]",
      "name": "127.0.0.1",
      "ipaddress": "127.0.0.1",
      "hostname": "local",
      "customer": "",
      "last_method": "raw_udp",
      "last_byte_time": 1582644506,
      "did": "Unknown",
      "mac_address": "Unknown",
      "address": "",
      "version": "Unknown",
```

(continues on next page)

(continued from previous page)

```
      "manufacturer": "Unknown",
      "timezone": "Unknown",
      "description": "",
      "model": "Unknown",
      "asset_groups": [],
      "assets": [],
      "profiles": []
      }]
  }
```

RESP_CODE: 200

3.6.3. POST

/v2/configs/asset_groups

Input Data

This is an example input object describing a single Asset Group, with no child asset groups and no assets. Note: Sub-object “asset_groups” and “assets” may be pre-populated, including their sub- objects.

```
{
  "name": "A Group",
  "description": "Anything",
  "enabled": 1,
  "locked": 0,
  "parent_id": "",
  "physical_address": "123 Main Street",
  "assets": [],
  "asset_groups": []
}
```

Filled entry example

```
{
  "name": "A Group",
  "description": "Anything",
  "enabled": 1,
  "locked": 0,
  "parent_id": "",
  "physical_address": "123 Main Street",
  "asset_groups": [
  {
    "asset_group_id": "GG56YBLHYBDMFVQH161[...]",
    "name": "testEmpty.sh8",
    "description": "Anything",
    "enabled": 1,
    "locked": 0,
    "parent_id": 10,
    "physical_address": "123 Main Street",
    "assets": [],
    "asset_groups": []
```

(continues on next page)

(continued from previous page)


```
 } ]
 "assets": [
 {
  "asset_id": "XQATXO524O3Q0HZQ16[...]",
  "name": "172",
  "ipaddress": "10.13.37.172",
  "hostname": "hawaiiIsoSp10",
  "customer": "",
  "last_method": "",
  "last_byte_time": 0,
  "did": "Unknown",
  "mac_address": "Unknown",
  "address": "",
  "version": "Unknown",
  "manufacturer": "LayerX Technologies",
  "timezone": "UTC",
  "description": "Local Arbitrator Platform",
  "model": "Unknown",
  "asset_groups": [],
  "assets": [],
  "profiles": [{
    "profile_id": "LXTAE3NIJH37W3C81560[...]",
    "asset_id": "XQATXO524O3Q0HZQ161099[...]",
    "probe_group_id": "LXTCSP5DDP62CC6C[...]",
    "enabled": 1,
    "interval": 30,
    "start_time": -1,
    "start_weekdays": 127,
    "end_window": 86400,
    "assets": [ { "asset_id": "XQATXO524O3Q0HZQ1610[...]" } ],
    "credentials": [],
    "probe_groups": [ {
      "probe_group_id": "LXTCSP5DDP62CC6C156[...]",
      "name": "Local System Stats",
      "description": "Probes the local Arbitrator platform for statistics.",
      "profiles": [{"profile_id": "LXTAE3NIJH37W3C8156O5[...]"}],
      "probes": [{ "probe_id": "LXTQITGJ6DFZ00TD15605399[...]",
        "name": "Disk Stats",
        "short_message": "DISK",
        "command": "get_arb_stat.sh -d",
        "description": "Local System Disk Usage",
        "locked": 0,
        "enabled": 1,
        "unit": "",
        "autoscale": 0,
        "probe_groups": [{"probe_group_id": "LXTCSP5DDP62CC6C156O54[...]"}]},
        {"probe_id": "LXTQITGJ6DFZ00TD15605399[...]",
         "name": "Memory Stats",
         "short_message": "MEM",
         "command": "get_arb_stat.sh -m",
         "description": "Watches Memory Consumption",
         "locked": 0,
```


(continues on next page)

(continued from previous page)


```
          "enabled": 1,
          "unit": "",
          "autoscale": 0,
          "probe_groups": [{"probe_group_id": "LXTCSP5DDP62CC6C15[...]"}]},
         {"probe_id": "LXTQITGJ6DFZ00TD15605[...]",
          "name": "CPU Stats",
          "short_message": "CPU",
          "command": "get_arb_stat.sh -c",
          "description": "Watches CPU Usage",
          "locked": 0,
          "enabled": 1,
          "unit": "",
          "autoscale": 0,
          "probe_groups": [{"probe_group_id": "LXTCSP5DDP62CC6C15605400[...]"}]},
         {"probe_id": "LXTQITGJ6DFZ00TD1560539[...]",
          "name": "Outbound Network Traffic (kBps)",
          "short_message": "OBNET",
          "command": "get_arb_stat.sh -o",
          "description": "Watches Outbound Network Statistics",
          "locked": 0,
          "enabled": 1,
          "unit": "",
          "autoscale": 0,
          "probe_groups": [{"probe_group_id": "LXTCSP5DDP62CC6C156054[...]"}]},
         {"probe_id": "LXTQITGJ6DFZ00TD156053[...]",
          "name": "Inbound Network Traffic (kBps)",
          "short_message": "IBNET",
          "command": "get_arb_stat.sh -i",
          "description": "Watches Inbound Network Statistics",
          "locked": 0,
          "enabled": 1,
          "unit": "",
          "autoscale": 0,
          "probe_groups": [{"probe_group_id": "LXTCSP5DDP62CC6C1560[...]"}]} ]
        } ] } ] } ]
}
```


- Example 1: POST asset group

  Command:

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/asset_group1
     -X POST https://<IP or FQDN>/v2/configs/asset_groups
```

  ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiO[...]
     --insecure
```

(continues on next page)

(continued from previous page)

```
    -w "RESP_CODE: %{response_code}"
    -d@test_data/asset_group1
    -X POST https://10.13.37.14/v2/configs/asset_groups
    -d '{"name":"ExampleName",
         "description":"Anything",
         "enabled":1,
         "locked":0,
         "parent_id":"",
         "physical_address":"123 Main Street"}'
```

Output (formatted):

```
{"status": 200,
 "message": "Success",
 "data": [{"asset_group_id": "S33HDW6Z5205ICNK1[...]",
           "name": "ExampleName",
           "description": "Anything",
           "enabled": 1,
           "locked": 0,
           "parent_id": "",
           "physical_address": "123 Main Street",
           "assets": [],
           "asset_groups": []
            }
    ]
 }
```

RESP_CODE: 200

- Example 2: POST asset group with asset

  Input Data

  This is an example input object describing a single Asset Group with a single Asset.

```
{"name": "ExampleName",
 "description": "ExampleDescription",
 "enabled": 1,
 "locked": 0,
 "parent_id": "",
 "physical_address": "1234 Main Street",
 "assets": [{"name": "TestAsset",
             "ipaddress": "10.13.37.55",
             "hostname": "",
             "customer": "",
             "last_method": "",
             "last_byte_time": 0,
             "did": "",
             "mac_address": "",
             "address": "",
             "version": "",
             "manufacturer": "",
             "timezone": "UTC",
             "description": "",
```

(continues on next page)

(continued from previous page)

```
            "model": "",
            "asset_groups": [],
            "assets": [],
            "profiles": []
           }
    ]
}
```

Command

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/asset_group1
     -X POST https://<IP or FQDN>/v2/configs/asset_groups
```

ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/asset_group1
     -X POST https://10.13.37.14/v2/configs/asset_groups
```

Output (formatted):

```
{"status": 200,
 "message": "Success",
 "data": [{"asset_group_id": "SZ3HWDCRGS69SLR0158[...]",
           "name": "ExampleName",
           "description": "ExampleDescription",
           "enabled": 1,
           "locked": 0,
           "parent_id": "",
           "physical_address": "1234 Main Street",
           "assets": [{"asset_id":"TMWPJN8312ZUVX9V1582[...]",
                       "name": "TestAsset",
                       "ipaddress": "10.13.37.55",
                       "hostname": "Unknown",
                       "customer": "",
                       "last_method": "",
                       "last_byte_time": 0,
                       "did": "Unknown",
                       "mac_address": "Unknown",
                       "address": "",
                       "version": "Unknown",
                       "manufacturer": "Unknown",
                       "timezone": "UTC",
                       "description": "",
                       "model": "Unknown",
```

(continues on next page)

(continued from previous page)

```
                        "asset_groups": [{"asset_group_id":"SZ3HWDCRGS69SLR0158[...]"}
↪],
                        "assets": [],
                        "profiles": []}],
            "asset_groups": []
            }]
}
```

RESP_CODE: 200

- Example 3: PUT asset_groups: Modify Asset Group

  Input data

```
{"asset_group_id": "R4ICXBVWMU5C21YL1582[...]",
 "name": "New Name",
 "description": "New Description",
 "enabled": 1,
 "locked": 0,
 "parent_id": "",
 "physical_address": "1234 New Main Street",
 "assets": [],
 "asset_groups": []
}
```

  Command:

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/asset_group1
     -X POST https://<IP or FQDN>/v2/configs/asset_groups
```

  ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1Q[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/asset_group2
     -X POST https://10.13.37.14/v2/configs/asset_groups
```

  Output (formatted):

```
{"status": 200,
 "message": "Success",
 "data": [{"asset_group_id": "R4ICXBVWMU5C21YL158266549[...]",
           "name": "New Name",
           "description": "New Description",
           "enabled": 1,
           "locked": 0,
           "parent_id": "",
           "physical_address": "1234 New Main Street",
```

(continues on next page)

(continued from previous page)

```
            "assets": [],
            "asset_groups": []
             }]
}
```

RESP_CODE: 200

3.6.4. PUT

/v2/configs/asset_groups

- Example 1: Put Asset Group

  Input Data

```
{"asset_group_id": "GG56YBLHYBDMFVQH16110060[...]",
 "name": "New Name 2"
}
```

  Command

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiL[...]
     --insecure
     -w RESP_CODE: %{response_code}
     -d@./test/input_data.json
     -X PUT https://10.13.37.14/api/v2/configs/asset_groups
```

  Output (formatted):

```
  {"status": 200,
   "message": "Success",
   "data": [{"asset_group_id": "GG56YBLHYBDMFVQ[...]",
             "name": "New Name 2"
             "description": "New Description",
             "enabled": 1,
             "locked": 0,
             "parent_id": "",
             "physical_address": "1234 New Main Street",
             "assets": [],
             "asset_groups": []
            }]
  }


RESP_CODE: ``200``
```

3.6.5. DELETE

/v2/configs/asset_groups

- Example 1: Delete Asset Group

  Input Data

```
{"asset_group_id": "GG56YBLHYBDMFVQH161[...]" }
```

  Command

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1Q[...]
     --insecure
     -w RESP_CODE: %{response_code}
     -X DELETE https://10.13.37.14/api/v2/configs/asset_groups/GG56[...]
```

  or

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJ[...]
     --insecure
     -w RESP_CODE: %{response_code}
     -d@./test/DELETE.7.input_data.json
     -X DELETE https://10.13.37.14/api/v2/configs/asset_groups
```

  Output (formatted):

```
{"status":null,"message":null,"data":[]}
```

  RESP_CODE: 200

3.7. /v2/configs/assets

Method	URL	Description
GET	/v2/configs/assets	Retrieves current list of all assets.
POST	/v2/configs/assets	Adds new assets.
PUT	/v2/configs/assets	Modifies an existing asset.
DELETE	/v2/configs/assets	Deletes an existing asset.

3.7.1. Header (required)

x-lxt-api-token: “token from login”

3.7.2. GET

/v2/configs/assets

- Example 1: Get All Assets

  Command:

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1Qi[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -X GET https://<IP or FQDN>/api/v2/configs/assets
```

  Output (formatted):


```
{"status":200,
 "message":"Success",
 "data":[{"asset_id":"a3b7d01cd3b7a3[...]",
          "name":"Local System",
          "ipaddress":"10.13.37.173",
          "hostname":"ATestArb",
          "customer":"",
          "last_method":"",
          "last_byte_time":0,
          "did":"Unknown",
          "mac_address":"Unknown",
          "address":"",
          "version":"Unknown",
          "manufacturer":"LayerX Technologies",
          "timezone":"UTC",
          "description":"Local Arbitrator Platform",
          "model":"Unknown",
          "asset_groups":[],
          "assets":[],
          "profiles":[{
            "profile_id":"LXTAE3NIJH37W3C8[...]",
            "asset_id":"a3b7d01cd3b7a3860b[...]",
            "probe_group_id":"LXTCSP5DDP62[...]",
            "enabled":1,
            "interval":30,
            "start_time":-1,
            "start_weekdays":127,
            "end_window":86400,
            "assets":[{"asset_id":"a3b7d01cd[...]"}],
            "credentials":[],
            "probe_groups":[{
              "probe_group_id":"LXTCSP5DDP62[...]",
```


(continues on next page)

(continued from previous page)


```
              "name":"Local System Stats",
              "description":"Probes the local Arbitrator platform for statistics.",
              "profiles":[{"profile_id":"LXTAE3NIJH[...]"}],
              "probes":[{"probe_id":"LXTQITGJ6DFZ00[...]",
                "name":"Disk Stats",
                "short_message":"DISK",
                "command":"get_arb_stat.sh -d",
                "description":"Local System Disk Usage",
                "locked":0,
                "enabled":1,
                "unit":"",
                "autoscale":0,
                "probe_groups":[{"probe_group_id":"LXTCSP5DDP62[...]"}]
                }]}
            ]
          }]
         },
         {"asset_id":"b9ae6a7a12cbb238[...]",
           "name":"CUCM_8",
           "ipaddress":"CUCM_8",
           "hostname":"CUCM_8",
           "customer":"",
           "last_method":"",
           "last_byte_time":0,
           "did":"Unknown",
           "mac_address":"Unknown",
           "address":"",
           "version":"Cisco Call Manager",
           "manufacturer":"Cisco",
           "timezone":"Unknown",
           "description":"",
           "model":"Unknown",
           "asset_groups":[],
           "assets":[],
           "profiles":[]
         }
        ]
}
```


RESP_CODE: 200

- Example 2: Get One Asset

  Command:

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1Qi[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -X GET https://<IP or FQDN>/api/v2/configs/assets/a3b7d01cd3[...]
```

  Output (formatted):

```
{"status":200,
 "message":"Success",
 "data":[{"asset_id":"a3b7d01cd3b7a3860b6a[...]",
          "name":"Local System",
          "ipaddress":"10.13.37.173",
          "hostname":"ATestArb",
          "customer":"",
          "last_method":"",
          "last_byte_time":0,
          "did":"Unknown",
          "mac_address":"Unknown",
          "address":"",
          "version":"Unknown",
          "manufacturer":"LayerX Technologies",
          "timezone":"UTC",
          "description":"Local Arbitrator Platform",
          "model":"Unknown",
          "asset_groups":[],
          "assets":[],
          "profiles":[{"profile_id":"LXTAE3NIJH37W3[...]",
             "asset_id":"a3b7d01cd3b7a386[...]",
             "probe_group_id":"LXTCSP5DDP62CC6C[...]",
             "enabled":1,
             "interval":30,
             "start_time":-1,
             "start_weekdays":127,
             "end_window":86400,
             "assets":[{"asset_id":"a3b7d01cd3b7a3860[...]"}],
             "credentials":[],
             "probe_groups":[{"probe_group_id":"LXTCSP5DDP62CC6C1560[...]",
               "name":"Local System Stats",
               "description":"Probes the local Arbitrator platform for statistics.",
               "profiles":[{"profile_id":"LXTAE3NIJH37W3C81[...]"}],
               "probes":[{"probe_id":"LXTQITGJ6DFZ00TD156053[...]",
                  "name":"Disk Stats",
                  "short_message":"DISK",
                  "command":"get_arb_stat.sh -d",
                  "description":"Local System Disk Usage",
                  "locked":0,
                  "enabled":1,
                  "unit":"",
                  "autoscale":0,
                  "probe_groups":[{"probe_group_id":"LXTCSP5DDP62CC6C1[...]"}]}]
             }]
                    }]
       }]
}
```


RESP_CODE: 200

3.7.3. POST

/v2/configs/assets

Input Data

This is an example input object describing a single Asset.

```
{"name": "run.sh",
 "ipaddress": "3.3.3.3",
 "hostname": "3.3.3.3",
 "customer": "",
 "last_method": "",
 "last_byte_time": 0,
 "did": "Unknown",
 "mac_address": "Unknown",
 "address": "",
 "version": "Cisco Call Manager",
 "manufacturer": "Cisco",
 "timezone": "Unknown",
 "description": "",
 "model": "Unknown",
 "asset_groups": [],
 "assets": [],
 "profiles": []
}
```

- Example 1: POST asset

  Command:

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/asset1
     -X POST https://<IP or FQDN>/v2/configs/assets
```

  ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLC[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/asset1
     -X POST https://10.13.37.14/v2/configs/assets
```

  Output (formatted):

```
{"status": 200,
 "message": "Success",
 "data": [{"asset_id": "S33HDW6Z5205ICNK1582657[...]",
           "name": "run.sh",
           "ipaddress": "3.3.3.3",
           "hostname": "3.3.3.3",
```

(continues on next page)

(continued from previous page)

```
            "customer": "",
            "last_method": "",
            "last_byte_time": 0,
            "did": "Unknown",
            "mac_address": "Unknown",
            "address": "",
            "version": "Cisco Call Manager",
            "manufacturer": "Cisco",
            "timezone": "Unknown",
            "description": "",
            "model": "Unknown",
            "asset_groups": [],
            "assets": [],
            "profiles": []
          }]
}
```

RESP_CODE: 200

- Example 2: POST asset with profile

  Input Data

  This is an example input object describing a single Asset Group with a single Asset.

```
{"asset_id": "d9506ab97e091007a74bf[...]",
 "name": "Voss-IMP",
 "ipaddress": "10.30.1.104",
 "hostname": "Voss-IMP",
 "customer": "",
 "last_method": "syslog",
 "last_byte_time": 1611069910,
 "did": "Unknown",
 "mac_address": "Unknown",
 "address": "Main Street",
 "version": "Unknown",
 "manufacturer": "Unknown",
 "timezone": "Unknown",
 "description": "VOSS-EMEA-IMP",
 "model": "Unknown",
 "asset_groups": [{"asset_group_id": "LQ1CUQCIUF985[...]"}],
 "assets": [],
 "profiles": [{"profile_id": "WL7B4CWE5Y6N414N1597174[...]",
 "asset_id": "d9506ab97e0910[...]",
 "probe_group_id": "CUEJIM0KAA28G[...]",
 "enabled": 1,
 "interval": 60,
 "start_time": -1,
 "start_weekdays": 127,
 "end_window": 86400,
 "assets": [{"asset_id": "d9506ab97[...]" } ],
 "credentials": [],
 "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GWS[...]",
```

(continues on next page)

(continued from previous page)


```
      "name": "1b-PING Monitor",
      "description": null,
      "profiles": [{"profile_id": "WL7B4CWE5Y6N41[...]"}],
      "probes": [{"probe_id": "LOQ465XQQXJHS0JW[...]",
                  "name": "PING Monitor",
                  "short_message": "",
                  "command": "icmp_echo.exp %s",
                  "description": "",
                  "locked": 0,
                  "enabled": 1,
                  "unit": "",
                  "autoscale": 0,
                  "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GWSB1[...]"}]
                }]
      }]
    }]
}
```


Command

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/asset1
     -X POST https://<IP or FQDN>/v2/configs/assets
```

ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiL[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/asset1
     -X POST https://10.13.37.14/v2/configs/assets
```

Output (formatted):


```
{"status": 200,
 "message": "Success",
 "data": [{"asset_id":"d9506ab97e091007a74bf[...]",
     "name": "Voss-IMP",
     "ipaddress": "10.30.1.104",
     "hostname": "Voss-IMP",
     "customer": "",
     "last_method": "syslog",
     "last_byte_time": 1611069910,
     "did": "Unknown",
     "mac_address": "Unknown",
     "address": "Main Street",
     "version": "Unknown",
     "manufacturer": "Unknown",
```


(continues on next page)

(continued from previous page)


```
    "timezone": "Unknown",
    "description": "VOSS-EMEA-IMP",
    "model": "Unknown",
    "asset_groups": [{"asset_group_id": "LQ1CUQCIUF985LHY[...]"}],
    "assets": [],
    "profiles": [{"profile_id": "WL7B4CWE5Y6N414N1[...]",
        "asset_id":"d9506ab97e091007a74[...]",
        "probe_group_id": "CUEJIM0KAA28GWSB156[...]",
        "enabled": 1,
        "interval": 60,
        "start_time": -1,
        "start_weekdays": 127,
        "end_window": 86400,
        "assets": [{"asset_id":"d9506ab97e091007a74bf16[...]"}],
        "credentials": [],
        "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GWSB1[...]",
           "name": "1b-PING Monitor",
           "description": null,
           "profiles": [{"profile_id": "WL7B4CWE5Y6N414[...]"}],
           "probes": [{"probe_id": "LOQ465XQQXJHS0JW1568[...]",
              "name": "PING Monitor",
              "short_message": "",
              "command": "icmp_echo.exp %s",
              "description": "",
              "locked": 0,
              "enabled": 1,
              "unit": "",
              "autoscale": 0,
              "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GWSB1[...]"}]
           }]
        }]
    }]
 }]
}
```


RESP_CODE: 200

- Example 3: PUT assets: Modify Asset

  Input data


```
{"asset_id": "d9506ab97e091007a74b[...]",
 "name": "New Name"
}
```


  Command:

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/asset1
     -X POST https://<IP or FQDN>/v2/configs/assets
```

ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJ[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/asset1
     -X POST https://10.13.37.14/v2/configs/assets
```

Output (formatted):


```
{"status": 200,
 "message": "Success",
 "data": [{"asset_id":"d9506ab97e091007a74bf169[...]",
     "name": "New Name",
     "ipaddress": "10.30.1.104",
     "hostname": "Voss-IMP",
     "customer": "",
     "last_method": "syslog",
     "last_byte_time": 1611069910,
     "did": "Unknown",
     "mac_address": "Unknown",
     "address": "Main Street",
     "version": "Unknown",
     "manufacturer": "Unknown",
     "timezone": "Unknown",
     "description": "VOSS-EMEA-IMP",
     "model": "Unknown",
     "asset_groups": [{"asset_group_id": "LQ1CUQCIUF985LHY1[...]"}],
     "assets": [],
     "profiles": [{"profile_id": "WL7B4CWE5Y6N414N1[...]",
         "asset_id":"d9506ab97e091007a74[...]",
         "probe_group_id": "CUEJIM0KAA28GWS[...]",
         "enabled": 1,
         "interval": 60,
         "start_time": -1,
         "start_weekdays": 127,
         "end_window": 86400,
         "assets": [{"asset_id":"d9506ab97e091007a74b[...]"}],
         "credentials": [],
         "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GW[...]",
             "name": "1b-PING Monitor",
             "description": null,
             "profiles": [{"profile_id": "WL7B4CWE5Y6N414N1[...]"}],
             "probes": [{"probe_id": "LOQ465XQQXJHS0JW15[...]",
                "name": "PING Monitor",
                "short_message": "",
                "command": "icmp_echo.exp %s",
                "description": "",
                "locked": 0,
                "enabled": 1,
                "unit": "",
                "autoscale": 0,
                "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GWSB1[...]"}]
```


(continues on next page)

(continued from previous page)

```
            }]
         }]
      }]
   }]
}
```

RESP_CODE: 200

3.7.4. DELETE

/v2/configs/assets

- Example 1: Delete Asset

  Input Data

```
{"asset_id": "d9506ab97e091[...]"}
```

  Command

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1[...]
     --insecure
     -w RESP_CODE: %{response_code}
     -d@./test/input_data.json
     -X DELETE https://10.13.37.14/api/v2/configs/assets
```

  or

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLC[...]
     --insecure
     -w RESP_CODE: %{response_code}
     -X DELETE https://10.13.37.14/api/v2/configs/assets/d9506ab97e09100[...]
```

  Output (formatted):

```
{"status":null,
 "message":null,
 "data":[]}
```

  RESP_CODE: 200

3.8. /v2/configs/profiles

Method	URL	Description
GET	/v2/configs/profiles	Retrieves current list of all profiles.
POST	/v2/configs/profiles	Adds a new profile.
PUT	/v2/configs/profiles	Modifies an existing profile.
DELETE	/v2/configs/profiles	Deletes an existing profile.

3.8.1. Header (required)

x-lxt-api-token: "token from login"

3.8.2. GET

/v2/configs/profiles

- Example 1: Get All Profiles

  Command:

```
curl -s
-H x-lxt-api-token:eyJ0eXAiOiJKV1Qi[...]
--insecure
-w "RESP_CODE: %{response_code}"
-X GET https://<IP or FQDN>/api/v2/configs/profiles
```

  Output (formatted):

```
{"status":200,
 "message":"Success",
 "data":[{"profile_id": "WL7B4CWE5Y6N414N159[...]",
       "asset_id":"d9506ab97e091007a74bf1[...]",
       "probe_group_id": "CUEJIM0KAA28GWSB1568[...]",
       "enabled": 1,
       "interval": 60,
       "start_time": -1,
       "start_weekdays": 127,
       "end_window": 86400,
       "assets": [{"asset_id":"d9506ab97e0910076[...]"}],
     "credentials": [],
     "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GW[...]",
        "name": "1b-PING Monitor",
        "description": null,
        "profiles": [{"profile_id": "WL7B4CWE5Y6N414N159[...]"}],
        "probes": [{"probe_id": "LOQ465XQQXJHS0JW15[...]",
           "name": "PING Monitor",
           "short_message": "",
           "command": "icmp_echo.exp %s",
```

(continues on next page)

(continued from previous page)

```
            "description": "",
            "locked": 0,
            "enabled": 1,
            "unit": "",
            "autoscale": 0,
            "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GWS[...]"}]
            }]
       }]
    }]
}
```

RESP_CODE: 200

- Example 2: Get One Profile

  Command:

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1Qi[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -X GET https://<IP or FQDN>/api/v2/configs/profiles/WL7B4CWE5Y6N[...]
```

  Output (formatted):

```
{"status":200,
 "message":"Success",
 "data":[{"profile_id": "WL7B4CWE5Y6N414N15[...]",
     "asset_id":"d9506ab97e091007a74bf16[...]",
     "probe_group_id": "CUEJIM0KAA28GWSB1[...]",
     "enabled": 1,
     "interval": 60,
     "start_time": -1,
     "start_weekdays": 127,
     "end_window": 86400,
     "assets": [{"asset_id":"d9506ab97e0910[...]"}],
     "credentials": [],
     "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GW[...]",
           "name": "1b-PING Monitor",
           "description": null,
           "profiles": [{"profile_id": "WL7B4CWE5Y6N414N[...]"}],
           "probes": [{"probe_id": "LOQ465XQQXJHS0JW[...]",
              "name": "PING Monitor",
              "short_message": "",
              "command": "icmp_echo.exp %s",
              "description": "",
              "locked": 0,
              "enabled": 1,
              "unit": "",
              "autoscale": 0,
              "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GWSB[...]"}]
            }]
     }]
```

(continues on next page)

(continued from previous page)

```
 }]
}
```

RESP_CODE: 200

3.8.3. POST

/v2/configs/profiles

Input Data

This is an example input object describing a single Profile, mapping an existing asset to an existing probe_group, with no credentials supplied.

```
{
 "asset_id": "d9506ab97e091007a74bf1[...]",
 "probe_group_id": "CUEJIM0KAA28GWS[...]",
 "enabled": 1,
 "interval": 60,
 "start_time": -1,
 "start_weekdays": 127,
 "end_window": 86400,
 "credentials" : []
}
```

- Example 1: POST profile

  Command:

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/profile1
     -X POST https://<IP or FQDN>/v2/configs/profiles
```

  ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1Qi[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data
     -X POST https://10.13.37.14/v2/configs/profiles
```

  Output (formatted):

```
{"profile_id": "WL7B4CWE5Y6N414N15[...]",
 "asset_id": "d9506ab97e0910[...]",
 "probe_group_id": "CUEJIM0KAA28GWSB1[...]",
 "enabled": 1,
 "interval": 60,
 "start_time": -1,
```

(continues on next page)

(continued from previous page)


```
 "start_weekdays": 127,
 "end_window": 86400,
 "assets": [{"asset_id": "d9506ab97e09100[...]"}],
 "credentials": [],
 "probe_groups": [{"probe_group_id": "CUEJIM0KAA28G[...]",
     "name": "1b-PING Monitor",
     "description": null,
     "profiles": [{"profile_id": "WL7B4CWE5Y6N414N[...]"}],
     "probes": [{"probe_id": "LOQ465XQQXJHS0JW156[...]",
        "name": "PING Monitor",
        "short_message": "",
        "command": "icmp_echo.exp %s",
        "description": "",
        "locked": 0,
        "enabled": 1,
        "unit": "",
        "autoscale": 0,
        "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GWS[...]"}]
     }]
  }]
}
```


RESP_CODE: 200

- Example 3: PUT profile: Modify Profile

  Input data


```
{"profile_id": "WL7B4CWE5Y6N414[...]",
 "interval": 30
}
```


  Command:

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data
     -X POST https://<IP or FQDN>/v2/configs/profiles
```

  ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1Qi[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/profile1
     -X POST https://10.13.37.14/v2/configs/profiles
```

  Output (formatted):


```
{"profile_id": "WL7B4CWE5Y6N414N15[...]",
 "asset_id": "d9506ab97e091007a74[...]",
```


(continues on next page)

(continued from previous page)


```
 "probe_group_id": "CUEJIM0KAA28GWSB15[...]",
 "enabled": 1,
 "interval": 30,
 "start_time": -1,
 "start_weekdays": 127,
 "end_window": 86400,
 "assets": [{"asset_id": "d9506ab97e09100[...]"}],
 "credentials": [],
 "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GWSB[...]",
    "name": "1b-PING Monitor",
    "description": null,
    "profiles": [{"profile_id": "WL7B4CWE5Y6N414[...]"}],
    "probes": [{"probe_id": "LOQ465XQQXJHS0JW15[...],
         "name": "PING Monitor",
         "short_message": "",
         "command": "icmp_echo.exp %s",
         "description": "",
         "locked": 0,
         "enabled": 1,
         "unit": "",
         "autoscale": 0,
         "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GW[...]"}]
       }]
   }]
}
```


RESP_CODE: 200

3.8.4. DELETE

/v2/configs/profiles

- Example 1: Delete Profile

  Input Data


```
{"profile_id": "WL7B4CWE5Y6N41[...]"}
```


  Command

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiL[...]
     --insecure
     -w RESP_CODE: %{response_code}
     -d@./test/input_data.json
     -X DELETE https://10.13.37.14/api/v2/configs/profiles
```

  or

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiL[...]
     --insecure
```

(continues on next page)

(continued from previous page)

```
    -w RESP_CODE: %{response_code}
    -X DELETE https://10.13.37.14/api/v2/configs/profiles/WL7B4CWE5Y6N41[...]
```

Output (formatted):

```
{"status":null,
 "message":null,
 "data":[]}
```

RESP_CODE: 200

3.9. /v2/configs/probe_groups

Method	URL	Description
GET	/v2/configs/probe_groups	Retrieves current list of all probe_groups and all children underneath.
POST	/v2/configs/probe_groups	Adds a new probe group and all children.
PUT	/v2/configs/probe_groups	Modifies an existing probe group and all children.
DELETE	/v2/configs/probe_groups	Deletes an existing probe group.

3.9.1. Header (required)

x-lxt-api-token: "token from login"

3.9.2. GET

/v2/configs/probe_groups

- Example 1: Get All Probe Groups

  Command:

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJh[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -X GET https://<IP or FQDN>/api/v2/configs/probe_groups
```

  Output (formatted):

```
{"status":200,
 "message":"Success",
 "data":[{"probe_group_id": "CUEJIM0KAA28[...]",
      "name": "1b-PING Monitor",
      "description": null,
      "profiles": [],
```

(continues on next page)

(continued from previous page)


```
      "probes": [{"probe_id": "LOQ465XQQXJ[...]",
         "name": "PING Monitor",
         "short_message": "",
         "command": "icmp_echo.exp %s",
         "description": "",
         "locked": 0,
         "enabled": 1,
         "unit": "",
         "autoscale": 0,
         "probe_groups": [{"probe_group_id": "CUEJIM0KAA28G[...]"}]
      }]
   }]
}
```


RESP_CODE: 200

3.9.3. POST

/v2/configs/probe_groups

Input Data


```
{
  "name": "Probe Group",
  "description": null,
  "profiles": [],
  "probes": [{"probe_id": "LOQ465XQQXJHS0[...]",
      "name": "PING Monitor",
      "short_message": "",
      "command": "icmp_echo.exp %s",
      "description": "",
      "locked": 0,
      "enabled": 1,
      "unit": "",
      "autoscale": 0,
    }]
}
```


- Example 1: POST probe group with probe

  Command:

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/probe_group1
     -X POST https://<IP or FQDN>/v2/configs/probe_groups
```

  ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiL[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/probe_group1
     -X POST https://10.13.37.14/v2/configs/probe_groups'
```

Output (formatted):


```
{"status":200,
 "message":"Success",
 "data":[{"probe_group_id": "CUEJIM0KAA28G[...]",
    "name": "1b-PING Monitor",
    "description": null,
    "profiles": [],
    "probes": [{"probe_id": "LOQ465XQQXJHS0J[...]",
      "name": "PING Monitor",
      "short_message": "",
      "command": "icmp_echo.exp %s",
      "description": "",
      "locked": 0,
      "enabled": 1,
      "unit": "",
      "autoscale": 0,
      "probe_groups": [{"probe_group_id": "CUEJIM0KAA28G[...]"}]
     }]
   }]
}
```


RESP_CODE: 200

3.9.4. PUT

/v2/configs/probe_groups

- Example 1: PUT probe_groups: Modify Probe Group

  Input data

```
{"probe_group_id": "R4ICXBVWMU5C21YL158[...]",
 "name": "New Name" }
```

  Command:

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/probe_group1
     -X POST https://<IP or FQDN>/v2/configs/probe_groups
```

  ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLC[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/probe_group2
     -X POST https://10.13.37.14/v2/configs/probe_groups
```

Output (formatted):

```
{"status":200,
 "message":"Success",
 "data":[{"probe_group_id": "CUEJIM0KAA28[...]",
    "name": "New Name"
    "description": null,
    "profiles": [],
    "probes": [{"probe_id": "LOQ465XQQXJHS[...]",
         "name": "PING Monitor",
         "short_message": "",
         "command": "icmp_echo.exp %s",
         "description": "",
         "locked": 0,
         "enabled": 1,
         "unit": "",
         "autoscale": 0,
         "probe_groups": [{"probe_group_id": "CUEJIM0K[...]"}]
       }]
   }]
}
```

RESP_CODE: 200

3.9.5. DELETE

/v2/configs/probe_groups

- Example 1: Delete Probe Group

  Input Data

```
{"probe_group_id": "GG56YBLHYBDMFVQH161[...]" }
```

  Command

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1Q[...]
     --insecure
     -w RESP_CODE: %{response_code}
     -X DELETE https://10.13.37.14/api/v2/configs/probe_groups/GG56YBLHYBDM[...]
```

  or

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1Qi[...]
```

(continues on next page)

(continued from previous page)

```
    --insecure
    -w RESP_CODE: %{response_code}
    -d@./test/DELETE.7.input_data.json
    -X DELETE https://10.13.37.14/api/v2/configs/probe_groups
```

Output (formatted):

```
{"status":null,
 "message":null,
 "data":[]}
```

RESP_CODE: 200

3.10. /v2/configs/probes

Method	URL	Description
GET	/v2/configs/probes	Retrieves current list of all update probes.
POST	/v2/configs/probes	Adds a new probe.
PUT	/v2/configs/probes	Modifies an existing probe.
DELETE	/v2/configs/probes	Deletes an existing probe.

3.10.1. Header (required)

x-lxt-api-token: “token from login”

3.10.2. GET

/v2/configs/probes

- Example 1: Get All Probes

  Command:

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLC[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -X GET https://<IP or FQDN>/api/v2/configs/probes
```

  Output (formatted):

```
{"status":200,
 "message":"Success",
 "data":[{"probe_id": "LOQ465XQQXJHS0[...]",
     "name": "PING Monitor",
     "short_message": "",
```

(continues on next page)

(continued from previous page)

```
    "command": "icmp_echo.exp %s",
    "description": "",
    "locked": 0,
    "enabled": 1,
    "unit": "",
    "autoscale": 0,
    "probe_groups": [{"probe_group_id": "CUEJIM0KAA2[...]"}]
  }]
}
```

RESP_CODE: 200

- Example 2: Get One Probe

  Command:

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJhbG[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -X GET https://<IP or FQDN>/api/v2/configs/probes/LOQ465XQQXJHS[...]
```

  Output (formatted):

```
{"status":200,
 "message":"Success",
 "data":[{"probe_id": "LOQ465XQQXJHS0JW15[...]",
          "name": "PING Monitor",
          "short_message": "",
          "command": "icmp_echo.exp %s",
          "description": "",
          "locked": 0,
          "enabled": 1,
          "unit": "",
          "autoscale": 0,
          "probe_groups": [{"probe_group_id": "CUEJIM0KAA2[...]"}]
     }]
}
```

RESP_CODE: 200

3.10.3. POST

/v2/configs/probes

Input Data

This is an example input object describing a single Probe.

```
{"name": "PING Monitor",
 "short_message": "",
 "command": "icmp_echo.exp %s",
 "description": "",
```

(continues on next page)

(continued from previous page)

```
 "locked": 0,
 "enabled": 1,
 "unit": "",
 "autoscale": 0,
 "probe_groups": []
}
```

- Example 1: POST probe

  Command:

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/probe1
     -X POST https://<IP or FQDN>/v2/configs/probes
```

  ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJh[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/probe1
     -X POST https://10.13.37.14/v2/configs/probes
```

  Output (formatted):

```
{"status": 200,
 "message": "Success",
 "data": [{"probe_id": "LOQ465XQQXJHS0J[...]",
           "name": "PING Monitor",
           "short_message": "",
           "command": "icmp_echo.exp %s",
           "description": "",
           "locked": 0,
           "enabled": 1,
           "unit": "",
           "autoscale": 0,
           "probe_groups": []
          }]
}
```

  RESP_CODE: 200

3.10.4. PUT

/v2/configs/probes

- Example 3: PUT probes: Modify Probe

  Input data

```
{"probe_id": "d9506ab97e091007a74[...]",
"name": "New Name" }
```

  Command:

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/probe1
     -X POST https://<IP or FQDN>/v2/configs/probes
```

  ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLC[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/probe1
     -X POST https://10.13.37.14/v2/configs/probes
```

  Output (formatted):

```
{"status": 200,
 "message": "Success",
 "data": [{"probe_id": "d9506ab97e091007a7[...]",
      "name": "New Name",
      "short_message": "",
      "command": "icmp_echo.exp %s",
      "description": "",
      "locked": 0,
      "enabled": 1,
      "unit": "",
      "autoscale": 0,
      "probe_groups": [{"probe_group_id": "CUEJIM0KAA28GWS[...]"}]
   }]
}
```

  RESP_CODE: 200

3.10.5. DELETE

/v2/configs/probes

- Example 1: Delete Probe

  Input Data

```
{"probe_id": "d9506ab97e091007a7[...]"}
```

  Command

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJ[...]
     --insecure
     -w RESP_CODE: %{response_code}
     -d@./test/input_data.json
     -X DELETE https://10.13.37.14/api/v2/configs/probes
```

  or

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLC[...]
     --insecure
     -w RESP_CODE: %{response_code}
     -X DELETE https://10.13.37.14/api/v2/configs/probes/d9506ab97e[...]
```

  Output (formatted):

```
{"status":null,"message":null,"data":[]}
```

  RESP_CODE: 200

3.11. /v2/configs/credentials

Method	URL	Description
GET	/v2/configs/credentials	Retrieves current list of all credentials.
POST	/v2/configs/credentials	Adds a new credential.
PUT	/v2/configs/credentials	Modifies an existing credential.
DELETE	/v2/configs/credentials	Deletes an existing credential.

3.11.1. Header (required)

x-lxt-api-token: "token from login"

3.11.2. GET

/v2/configs/credentials

- Example 1: Get All Credentials

  Command:

```
curl -s
     -H "x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJh[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -X GET https://10.13.37.14/api/v2/configs/credentials
```

  Output (formatted):

```
{"status": 200,
 "message": "Success",
 "data": [{"credential_id": "E613X4TEDYG[...]",
           "name": "ray",
           "username": "LayerX does not return usernames.",
           "password": "LayerX does not return passwords.",
           "profiles": [],
           "response_methods": []},
          {"credential_id":"HWO7ZVMUDG7SGWY7[...]",
           "name": "run.sh",
           "username": "LayerX does not return usernames.",
           "password": "LayerX does not return passwords.",
           "profiles": [],
           "response_methods": [] }
        ]
}
```

  RESP_CODE: 200

3.11.3. POST

/v2/configs/credentials

Input Data

This is an example input object describing a single Credential.

```
{"name": "admin2",
 "username": "admin2User",
 "password": "admin2Pass" }
```

- Example 1: POST probe group and probes

  Command:

```
curl -s
     -H "x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJh[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/credential
     -X POST https://10.13.37.14/v2/configs/credentials
```

Output (formatted):

```
{"status": 200,
 "message": "Success",
 "data": [{"credential_id":"IZP3WG47RN66R0AR1[...]",
           "name":"admin2",
           "username":"LayerX does not return usernames.",
           "password":"LayerX does not return passwords.",
           "profiles":[],
           "response_methods":[] } ]
}
```

RESP_CODE: 200

3.11.4. PUT

/v2/configs/credentials

- Example 3: PUT credentials: Modify Credential

  Input data

```
{"credential_id":"IZP3WG47RN66R0AR[...]",
 "name":"New Name" }
```

  Command:

```
curl -s
     -H "x-lxt-api-token: xxx"
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/credential1
     -X POST https://<IP or FQDN>/v2/configs/credentials
```

  ie.

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1QiLCJhb[...]
     --insecure
     -w "RESP_CODE: %{response_code}"
     -d@test_data/credential1
     -X POST https://10.13.37.14/v2/configs/credentials
```

  Output (formatted):

```
{"status": 200,
 "message": "Success",
 "data": [{"credential_id":"IZP3WG47RN66R0AR16110[...]",
           "name":"New Name"
           "username":"LayerX does not return usernames.",
           "password":"LayerX does not return passwords.",
           "profiles":[],
           "response_methods":[] } ]
}
```


RESP_CODE: 200

3.11.5. DELETE

/v2/configs/credentials

- Example 1: Delete Credentials

  Input Data

```
{"credential_id": "d9506ab97e091007a74[...]"}
```

  Command

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1Q[...]
     --insecure
     -w RESP_CODE: %{response_code}
     -d@./test/input_data.json
     -X DELETE https://10.13.37.14/api/v2/configs/credentials
```

  or

```
curl -s
     -H x-lxt-api-token:eyJ0eXAiOiJKV1Qi[...]
     --insecure
     -w RESP_CODE: %{response_code}
     -X DELETE https://10.13.37.14/api/v2/configs/credentials/d9506ab97e09[...]
```

  Output (formatted):


```
{"status":null,
 "message":null,
 "data":[]}
```


  RESP_CODE: 200

3.12. /v2/system

Method	URL	Description
GET	/v2/system	Retrieves current list of all update requests.
GET	/v2/system/daysRemaining	Retrieves days remaining on license.
GET	/v2/system/license	Retrieves license key.
PUT	/v2/system/productkey/{key}	Updates license key with {key}.

3.12.1. GET

/v2/system

Output (formatted):


```
{"status": 200,
 "message": "Success",
 "data": {"Hostname": "arb",
          "Version": "sp23",
          "Theme": "themes/VOSS",
          "Flavor": "arbitrator",
          "License": "9H3EJ-aaaaa-7X79K-nnnnn-ccccc",
          "DaysLicensed": "365",
          "DaysRemaining": "171",
          "ProductKey": "7a426f46634971795a446e...",
          "Kernel": "Linux 4.14.17-lxt-3 x86_64 GNU/Linux"
  }
}
```


/v2/system/daysRemaining

Output (formatted):


```
{"status": 200,
 "message": "Success",
 "data": {"daysRemaining": "171"}
}
```


/v2/system/license

Output (formatted):


```
{"status": 200,
 "message": "Success",
 "data": {"license": "9H3EJ-aaaaa-7X79K-nnnnn-ccccc"}
}
```

4. Appendix

4.1. References

4.1.1. Complete API Reference

Method	URL	Description
GET	`alerts`	Get a list of all alerts.
GET	`alerts/{alert_id}`	Get a single alert by alert id.
POST	`alerts/disposition`	Disposition an Alert through API.
GET	`/assets`	Get a list of all assets.
GET	`/assets/{asset_id}`	Get a single asset by asset id.
POST	`/assets`	Add a new asset.
POST	`/assets`	Add multiple new assets.
PUT	`/assets`	Updates a single asset.
PUT	`/assets`	Updates multiple assets.
DELETE	`/assets`	Deletes a single asset.
DELETE	`/assets`	Deletes multiple assets.
GET	`/ciscocdr`	Get running totals for `ciscocdr` stats.
GET	`/system/stats`	Get system stats.

4.1.2. Status Codes

All status codes are standard HTTP status codes. The table below are status codes commonly used in LayerX's API.

Status Code	Description
200	OK
400	Bad Request by Client
403	Forbidden
404	Resource no found
500	Internal LayerX Error
501	Not Implemented
503	Service Unavailable

4.1.3. Disposition Description

Value	Description
0	Open.
1	Under Review.
2	Acknowledged.
3	Release.
4	Disregarded. This will delete Alert from system.
14	Closed.
15	Closed and Locked. This will delete Alert from system.