Pulse Secure®

Pulse One API Client Specification

Reference Guide

Release	1.0
Document Revision	1.0
Published Date	20 October 2020

Pulse Secure, LLC
2700 Zanker Road,
Suite 200 San Jose
CA 95134

https://www.pulsesecure.net

Pulse One API Client Specification - Reference Guide

The information in this document is current as of the date on the title page.

END USER LICENSE AGREEMENT

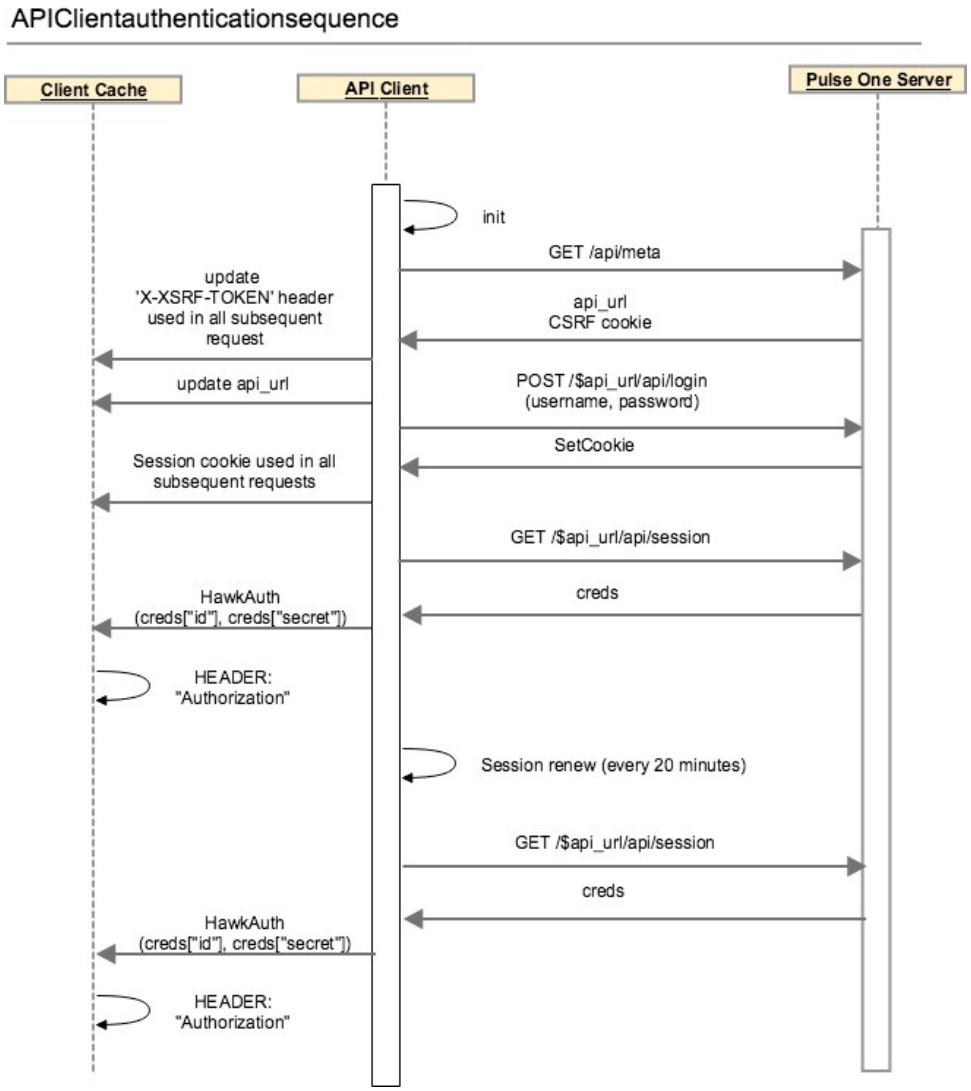
The Pulse Secure product that is the subject of this technical documentation consists of (or is intended for use with) Pulse Secure software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at https://www.pulsesecure.net. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

Authentication APIs

API endpoints related to authentication of a client that wants to use Pulse One REST API interface.
This document explains Pulse One API methods, data structures, and ways of using them together to achieve tasks like distributing configuration changes among managed appliances that belong to the same configuration group.

Authentication Flow



Note: In the rest of this document, all URLs are assumed to be prefixed with api_url value and not explicitly provided like in this diagram.

Authentication

Authentication with the v0 API is managed by sessions and cookies. When users log in, they are assigned a session and a session cookie is set with a token to allow more requests to be made with the session. See /api/login.
To use the v1 APIs, users are given Hawk authentication credentials. See */api/session*. All requests made to v1 APIs must be signed with Hawk authentication (https://github.com/hueniverse/hawk) -- a browser library (https://github.com/hueniverse/hawk/blob/master/lib/browser.js) is available.

Sessions and Session Cookies

Session cookies are used for the v0 API calls.
Session cookies expire after 20 minutes. If a cookie expires, then the user will be considered un-authenticated and will be prompted to re-log in when making a new request. However, every 2 minutes cookies are reissued for new requests. So, as long as a user is making requests within the 20-minute expiration window, the session cookie will be reissued every 2 minutes so it will stay fresh and therefore the session can stay valid forever.

Hawk Credentials

Hawk credentials are used for the v1 API calls.
Hawk credentials, like session cookies, expire after 20 minutes. To renew Hawk credentials, a request must be made to */api/session* within the expiration window to get new hawk credentials. Just like session cookies, new credentials will be made available every 2 minutes and they will be valid for up to 20 minutes. So, you can wait up-to 20 minutes before calling */api/session*, but if you call it sooner, you may get new credentials. See */api/session* for more details.

Hawk Authentication Header

API calls are authenticated by adding Authorization Hawk header to HTTP request. Hawk HTTP MAC code is calculated using following URI resource values as explained in Hawk authentication (https://github.com/hueniverse/hawk):

```
hawk.1.header\n
resource.timestamp\n
resource.nonce\n
resource.method\n
resource.name\n
resource.host\n
resource.port\n
\n
some-app-ext-data\n
\n
```

New lines are significant part of the content. Resource values are used before any HTTP related encoding. The calculated code is then plugged into Authorization header of the HTTP Request

```
Authorization: Hawk
id="id_from_cred",
ts="timestamp_value",
nonce="nonce_value", ext="some-
app-ext-data",
mac="calculated_mac_code"
```

For example, the following is the Hawk resource content of GET http://pulseone.com:8000/api/v1/groups

```
hawk.1.header 1353832234
j4h3g2 GET
/api/v1/groups pulseone.com
8000
```

Note: The nonce must be a unique and random value for every request.
The resulting header will look like (id and calculated MAC code will differ with cred value):

```
Authorization: Hawk
id="dh37fgj492je",
ts="1353832234",
nonce="j4h3g2", ext="",
mac="6R4rV5iE+NPoym+Ww
jeHzjAGXUtLNIxmo1vpMofpL
AE="
```

Authorization

Roles management will be explained here in future.

Requests

POST /api/login

Log in as a user.

JSON Body:

- username: (str) Username of the user to log in as.
- password: (str) Password for the user.

```
POST /api/login HTTP/1.1 Content-
Type: application/json Accept:
application/json
Host: customer.pulseone.net
{
   "username": "mreynolds",
   "password": "serenity"
}
```

If authenticated correctly, a JSON body containing a User Entity will be returned and a session cookie will be set. Invalid requests can expect the following errors:

- 401: Json Unauthorized - Login failed (user is locked, username not found, etc.)

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: length
Set-Cookie:cs_console_tkt="7e24...ZjU5!userid_type:b64str"; Path=/; Domain=.customer.pulseone.net

{
"display_name": "customer.pulseone.net", "domain_id": 1,
"email": "mreynolds@example.com",
"id": "88d40618-6039-42cf-bf87-243b0b2dceae", "username": "mreynolds",
...
}
```

GET /api/session

Get information about the current session.

```
GET /api/session HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseone.net
```

Response data will be in the form of a JSON dictionary with the following keys:

- domain: A dictionary containing data about the current customer domain.
- allow_security_appliances: (bool) Whether or not this domain is allowed to use security appliances or the related features. If this is disabled, all appliance features are not available to the user.
- entitlements: (list) A list of feature entitlements for this domain.
- user: A dictionary containing data about the currently authenticated user.
- display_name: (str) The display name of the current user.
- full_name: (str) The full name of the current user.
- group_name: (str) Deprecated.
- id: (str) A unique id for the current user.
- roles: (list of RoleEntity.List) The roles this user is associated with. Currently only one role is supported at a time so this list will have exactly one role.
- session_timeout: (int) The number of seconds of inactivity until this session will be timed out.
- username: (str) The username of the current user.
- permissions: (**RolePermissionsEntity**) Permissions this user has been granted. The complete set of permissions granted by all roles this user is associated with.
- api_url: A URL to use for v1 API requests. This URL must be used for all v1 requests and the value shouldn't be hard coded as it can change.
- credentials: A dictionary containing v1 API Hawk Credentials
  Note: Credentials will only be in the response if they have changed.
  - algorithm: (str) Hashing algorithm to use for Hawk -- always HS256
  - id: (str) Alphanumeric Hawk ID. This can be 100 characters or so
  - secret: (str) Alphanumeric secret to sign Hawk requests with
  - type: (str) Credential authentication type -- always "hawk"
  - expires: (int) UTC Unix timestamp in seconds when the credentials will expire. This value can be used to determine when it is time to renew the session.

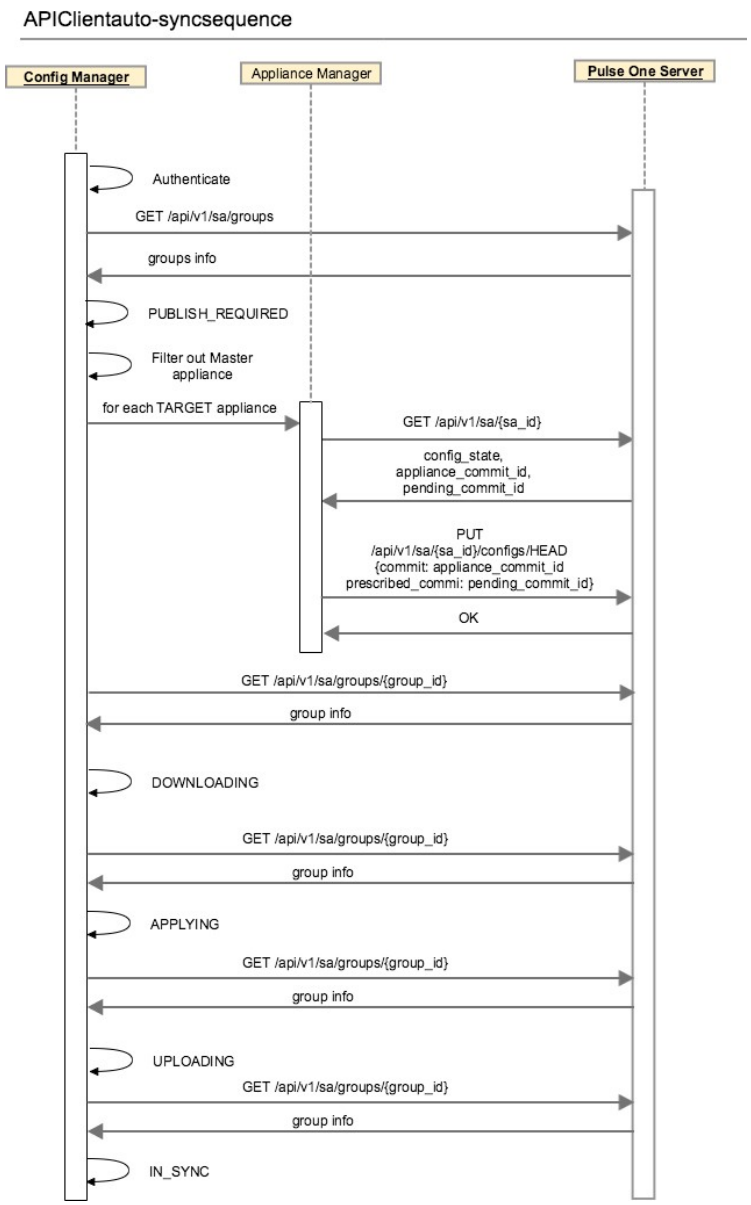
```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: length

{
"api_url": "https://api.pulseone.net",
"credentials": {
"type": "hawk", "algorithm":
"HS256",
"id": "{alphanumeric string}", "secret":
"{alphanumeric string}", "expires":
1428073884
},
"domain": { "allow_security_appliances":
true, "entitlements": [
"workspaces",
"appliances"
],
},
```

```
  "user": {
    "display_name": "admin",
    "full_name": null,
    "group_name": null,
    "id": "9822b71a-27cf-11e4-9440-1a1e7d989db4",
    "roles": [
      {
        "id": "8936a7de-b49e-47ee-acd5-9434da5508f8", "name":
        "Support Admin"
      }
    ],
    "session_timeout": 1200,
    "username": "admin",
    "permissions": {
    "action_map": {

          "WRITE_ACTIONS": ["CREATE", "READ", "UPDATE"],
          "DELETE_ACTIONS": ["CREATE", "READ", "UPDATE", "DELETE"],
        },
          "namespaces": {
"admin": ["READ"],
          "admin.settings": ["WRITE_ACTIONS"],
          "admin.appliances.appliance": ["DELETE_ACTIONS"],
          "admin.appliances.configdist": ["WRITE_ACTIONS"],
          "admin.appliances.appliance.operations": ["WRITE_ACTIONS"],
          "admin.appliances.log_aggregator": ["READ"], "admin.users.user":
          ["READ"]
        }
      }
    }
  }
```

Managed Appliance Configuration Management APIs

Configuration Distribution Flow

In the following diagram, Config Manager and Appliance Manager are assumed modules of the client using Pulse One API. Config Manager deals with Configuration Groups while Appliance Manager controls aspects of individual managed appliances. The diagram explains how client, as an example, can achieve automatic synchronization of Group targets with changes effected to Group's master. The method of configuration changes to Group's master is out of scope of this specification. One possible way is explained in PCS/PPS REST API Solutions Guide.



Note: PUT /api/v1/sa/{sa_id}/configs/HEAD call is using "Administrative Promote Pending Commit PUT" form explained in the corresponding section for this endpoint.

Appliance API

This is documentation for the API endpoints related to managed appliances.

Get By Id

Get managed appliance by ID.

Request

- Method: GET
- Resource: /api/v1/sa/{appliance-id}

Response

- Status: 200
- JSON Data: Response data in the form of a JSON dictionary with the following structure:
  - (SecurityAppliance) A **SecurityAppliance** entity

Example

Request

```
GET /api/v1/sa/1d89e147-a845-4825-93bf-154592454c25 HTTP/1.1
Accept: application/json Host: customer.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type:
application/json Content-
Length: 128
{
"created": "2015-02-23T23:53:09Z",
"group_id": "9ed4cea1-2f7b-4832-b5c1-82505548807e",
"id": "afbf5c72a95a482eae6697b823318e1e",
"name": "Excellent Appliance",
"state": "registered",
"notification_channel_status":
"online", "updated": "2015-02-
23T23:53:09Z"
"appliance_commit_id": "7dae60fc51088a99a72bad5c1434d805fee469f0",
"pending_commit_id":   "761ef2209fb2e7c06d870a672057cd5be002e324",
"config_size": "1389"
"config_created": "2015-02-23T23:53:09Z",
"type": "VPN",
"model": "MAG4610",
"serial_number": "0153M0TS00BII04U",
"cluster": {
"id": "bec7ed0a-7dcd-49d6-aa6c-ae5eaf47167f"
}
}
```

7List

Returns a list of managed appliances.

Request

- Method: GET
- Resource: /api/v1/sa

Response

- Status: 200
- JSON Data: Response data in the form of a JSON dictionary with the following structure:
  - items: (list of SecurityAppliance.GetAll) A list of SecurityAppliance.GetAll entities

Example

Request

```
GET /api/v1/sa
HTTP/1.1 Accept:
application/json
Host: customer.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 235
{
 "items": [
  {
   "created": "2015-02-23T23:53:09Z",
   "group_id": "9ed4cea1-2f7b-4832-b5c1-82505548807e",
   "id": "afbf5c72a95a482eae6697b823318e1e",
   "name": "Excellent Appliance",
   "state": "unregistered",
   "notification_channel_status": "offline", "updated":
   "2015-02-23T23:53:09Z"
   "appliance_commit_id":       "7dae60fc51088a99a72bad5c1434d805fee469f0",
   "pending_commit_id":         "761ef2209fb2e7c06d870a672057cd5be002e324",
   "config_size": "1389",
   "config_created": "2015-02-23T23:53:09Z",
   "type": "VPN",
   "model": "MAG4610",
   "serial_number": "0153M0TS00BII04U",
   "appliance_version": "8.1R4.1-32789",
   "cluster": {
       "id": "bec7ed0a-7dcd-49d6-aa6c-ae5eaf47167f",
        "node_name": "MyNodeNewName1", "leader_node": true,
   "active_node": false
   }
   }
   ]
   }
```

Reboot Appliance

Initiate a reboot of an appliance.

Request

- Method: POST
- Resource: /api/v1/sa/{appliance-id}/commands/reboot
- A body is not required for rebooting an appliance.

Response

- Status: 204

Example

Request

```
POST /api/v1/sa/f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b/commands/reboot HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseone.net
```

Response

```
HTTP/1.1 204 No Content
```

Appliance Groups

Master Appliance of the Group

An appliance whose configuration is considered as representative of the Group's configuration. All the members of the Group are expected to be compliant with this configuration unless explicitly requested otherwise. A compliant configuration is the one that has content of all Group's config block types replicated from the Master Appliance. The semantics of replication of blocks' content is out of the scope of this document but it does not always mean literal copy of the source content.

Target Appliance of the Group

An appliance that is a member of the Group and is a target of configuration replication process for the Group.

Creating a Group

Request

- Method: POST
- Resource: /api/v1/sa/groups
- JSON Data: JSON dictionary representing an ApplianceGroupEntity.Create entity.

Example

```
POST /api/v1/sa/groups HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseone.net
{
  "description": "Awesome group for appliances.",
  "name": "Foo Bar"
}
```

Response

- Status: 200
- JSON Data: Response data will be a ApplianceGroupEntity entity.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 3456
{
  "description": "Awesome group for appliances.", "id":
  "9ed4cea1-2f7b-4832-b5c1-82505548807e",
  "name": "Foo Bar"
}
```

Getting a Group

Request

- Method: GET
- Resource: /api/v1/sa/groups/{group-id}

Example

```
GET /api/v1/sa/groups/5a7b5f83-9bd2-462c-af8d-21aa31a76de2 HTTP/1.1
Accept: application/json
Host: customer.pulseone.net
```

Response

- Status: 200
- JSON Data: Response data will be a ApplianceGroupEntity entity.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 3456
{
   "description": "Shiney Group",
   "id": "5a7b5f83-9bd2-462c-af8d-21aa31a76de2",
   "name": "Shiney"
   "master_appliance_id": "d6c4e6d8-10a0-4570-ae01-ebed519439b9" "config_state":
   "in_sync",
   "members": [
      {
         "config_state": "in_sync",
         "appliance_id": "87f7c8a4-61cc-4f5c-85cb-e9338fc6b136"
      },
      {
         "config_state": "unknown",
         "appliance_id": "bc5e5a0d-4368-4338-a1d4-a3d6f0c71ee2"
      }
   ]
}
```

Getting A List Of Groups

Request

- Method: GET
- Resource: /api/v1/sa/groups

Example

```
GET /api/v1/sa/groups HTTP/1.1
Accept: application/json
Host: customer.pulseone.net
```

Response

- Status: 200
- JSON Data: Response data will be in the form of a JSON dictionary with the following keys:
  - items: (list) A list of ApplianceGroupEntitys.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 9456
{
 "items": [
  {
   "description": "Group A",
   "id": "7141c7a1-4e13-46b1-b8dc-1743003f341c",
   "name": "Group A",
   "master_appliance_id": "d6c4e6d8-10a0-4570-ae01-ebed519439b9"
   "config_state": "in_sync",
   "members": []
  },
  {
     "description": "Group B",
     "id": "50711ab8-f9f9-4405-87a7-35bb9c8dfcb9", "name":
     "Group B",
     "master_appliance_id": "ea26f353-e321-424c-8245-2d46c17d78e1"
     "config_state": "publishing",
     "members": [
        {
           "config_state": "publishing",
           "appliance_id": "87f7c8a4-61cc-4f5c-85cb-e9338fc6b136"
        },
        {
           "config_state": "downloading",
           "appliance_id": "bc5e5a0d-4368-4338-a1d4-a3d6f0c71ee2"
        }
     ]
  },
  {
     "description": "Group C",
     "id": "7143534f-6ad6-479b-8d1c-c9160ab13758",
     "name": "Group C",
     "master_appliance_id": "8c27d075-e24e-45a4-802f-f8a02e80f0bd"
     "config_state": "in_sync",
     "members": [
    {"config_state": "unknown",
    "appliance_id": "1ff7c8a4-61cc-5d5b-85cb-ea338fd6b136"
          },
       ]
    },
  ]
 }
```

Updating A Group

When updating a group, only registered appliances can be set as the master appliance.

Request

- Method: PUT
- Resource: /api/v1/sa/groups/{group-id}
- JSON Data: JSON dictionary representing an ApplianceGroupEntity.Update entity.

Example

```
PUT /api/v1/sa/groups/9ed4cea1-2f7b-4832-b5c1-82505548807e HTTP/1.1
Content-Type: application/json
Accept: application/json Host:
customer.pulseone.net
{
  "description": "Amazing group for appliances.",
  "name": "Foo Bar Baz"
}
```

Response

- Status: 200
- JSON Data: Response data will be a ApplianceGroupEntity entity.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 3456
{
  "description": "Amazing group for appliances.", "id":
  "9ed4cea1-2f7b-4832-b5c1-82505548807e",
  "name": "Foo Bar Baz"
}
```

Deleting A Group

Deleting a group will disassociate all appliances from this group.

Request

- Method: DELETE
- Resource: /api/v1/sa/groups/{group-id}

Example

```
DELETE /api/v1/sa/groups/9ed4cea1-2f7b-4832-b5c1-82505548807e HTTP/1.1
Host: customer.pulseone.net
```

Response

- Status: 204

Example

```
HTTP/1.1 204 No Content
```

Getting A List Of Members Of A Group

Request

- Method: GET
- Resource: /api/v1/sa/groups/{group-id}/members

Example

```
GET /api/v1/sa/groups/5a7b5f83-9bd2-462c-af8d-21aa31a76de2/members HTTP/1.1 Accept:
application/json
Host: customer.pulseone.net
```

Response

- Status: 200
- JSON Data: Response data will be an ApplianceGroupMemberListEntity entity.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 217
{
   {
      "id": "7e202f73-4713-43a9-94b2-a3f38db365e6",
      "members": [
         {
            "appliance_id": "9165d9ad-6eee-42fd-b308-3c2728989988"
         },
         {
            "appliance_id": "dc42aa5d-2bc4-4c4b-bbac-10a1cfdb9c1a"
         }
      ]
   }
}
```

Adding A Member to A Group

When adding to a group, member ID has to be one of a valid SecurityAppliance Entity IDs.

Request

- Method: POST
- Resource: /api/v1/sa/groups/{group-id}/members/{appliance-id}

Example

```
PUT /api/v1/sa/groups/9ed4cea1-2f7b-4832-b5c1-82505548807e/members/87f7c8a4-61cc-4f5c-85cb-e9338fc6b136 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseone.net
```

Response

- Status: 204

Example

```
HTTP/1.1 204 No Content
```

Removing A Member From A Group

Removing member appliance from a group.

Request

- Method: DELETE
- Resource: /api/v1/sa/groups/{group-id}/members/{appliance-id}

Example

```
DELETE /api/v1/sa/groups/9ed4cea1-2f7b-4832-b5c1-82505548807e/members/87f7c8a4-61cc-4f5c-85cb-e9338fc6b136 HTTP/1.1
Host: customer.pulseone.net
```

Response

- Status: 204

Example

```
HTTP/1.1 204 No Content
```

Adding Many Members To A Group

When adding many members to a group, each member ID has to be one of a valid SecurityAppliance Entity IDs.

Request

- Method: POST
- Resource: /api/v1/sa/groups/{group-id}
- JSON Data: JSON dictionary representing an ApplianceGroupMemberListEntity entity.

Example

```
PUT /api/v1/sa/groups/9ed4cea1-2f7b-4832-b5c1-82505548807e/members HTTP/1.1
Content-Type: application/json
Accept: application/json
```

```
Host: apicustomer.pulseone.net

{
   "id": "9ed4cea1-2f7b-4832-b5c1-82505548807e"
   "members": [
      {
         "appliance_id": "dc42aa5d-2bc4-4c4b-bbac-10a1cfdb9c1a"
      },
      {
         "appliance_id": "9165d9ad-6eee-42fd-b308-3c2728989988"
      }
   ]
}
```

Response

- Status: 204

Example

```
HTTP/1.1 204 No Content
```

Updating A Group's Config

Immediately after being created, group doesn't have Master Appliance set nor it has any config block type marked for replication. These values need to be set before group can publish prescribed configuration to Target Appliances.

Request

- Method: PUT
- Resource: /api/v1/sa/groups/{group-id}/config
- JSON Data: JSON dictionary representing an ApplianceGroupConfigEntity.Update entity.
- Errors:
  - 40003: Master Appliance conflict, see Errors section for details.
  - 404: Not Found

Example

```
PUT /api/v1/sa/groups/9ed4cea1-2f7b-4832-b5c1-82505548807e/config HTTP/1.1
Content-Type: application/json
Accept: application/json Host:
customer.pulseone.net
{
  "master_appliance_id": "53202f78-16b3-4171-9ecc-99b8890e3af9",
  "block_types": [
      "system.configuration.security",
      "user-roles.user-role"
  ]
}
```

Response

- Status: 204

Example

```
HTTP/1.1 204 No Content
```

Getting A Group's Config

Request

- Method: GET
- Resource: /api/v1/sa/groups/{group-id}/config
  Response - **Status:**200- **JSON Data:**
  Response data will be an ApplianceGroupConfigSettingsEntity entity. -
  Errors: -404`: Not Found

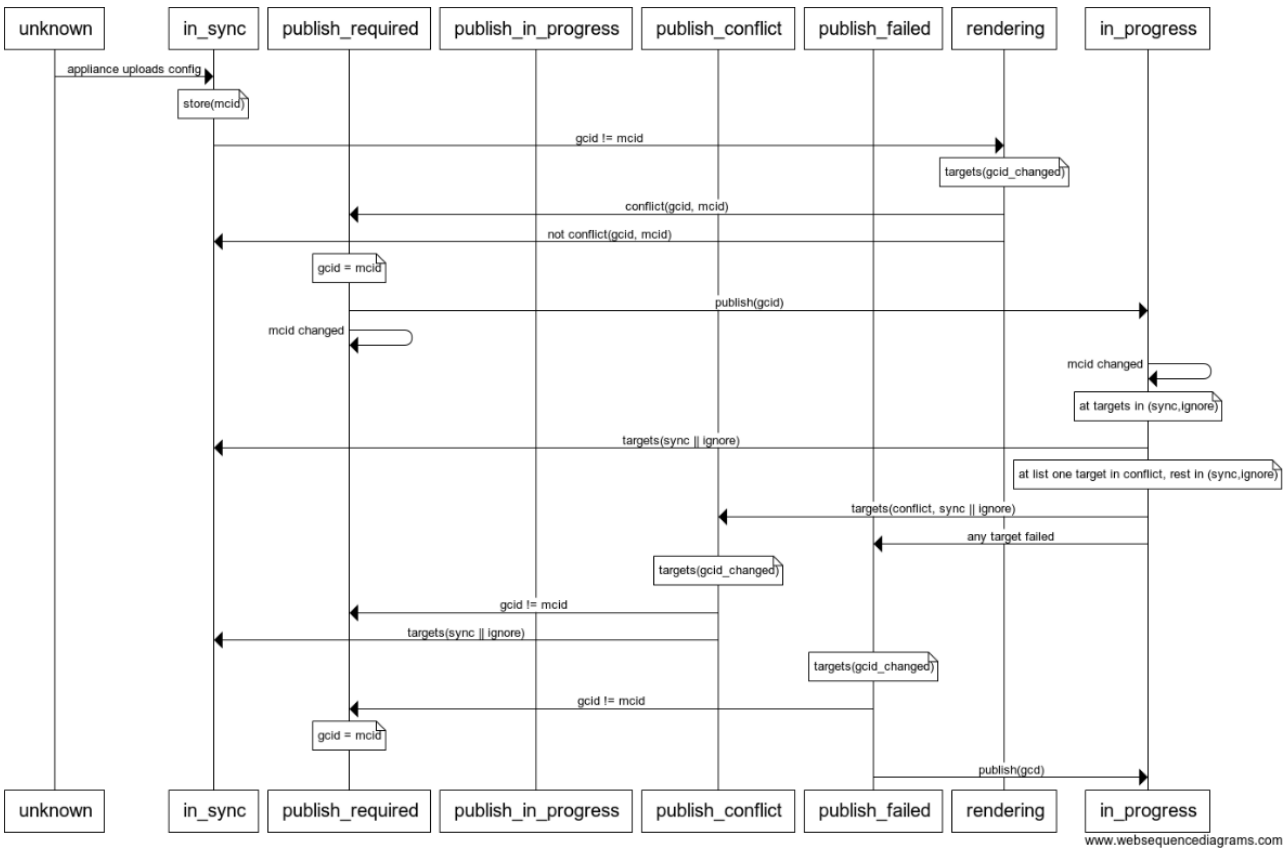
Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 3456
{
  "master_appliance_id": "53202f78-16b3-4171-9ecc-99b8890e3af9",
  "block_types": [
      "system.configuration.security",
      "user-roles.user-role"
  ],
  "config_state": "publish_required"
}
```

State Transition Diagram for the Group

Legend:

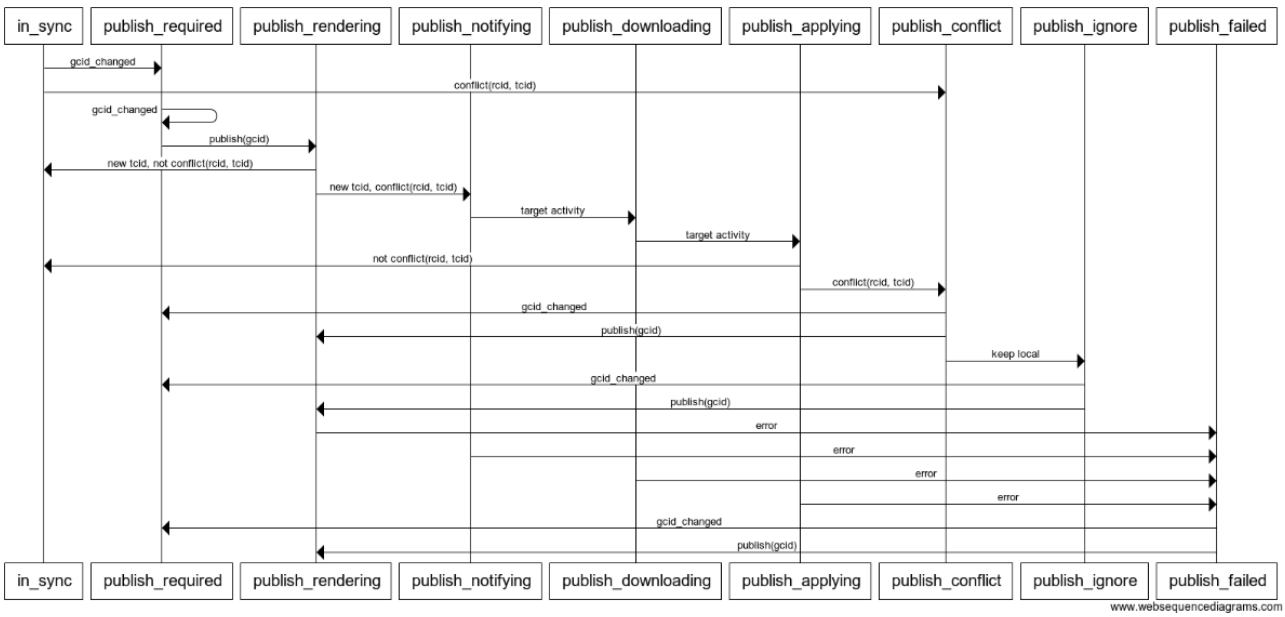
- mcid: Master Appliance Commit ID
- gcid: Group Commit ID - one of the recent Master Appliance's commit IDs considered to be a Group's current configuration. mcid will sometimes be ahead of gcid due to lag introduced by Group's publishing process
- publish: Action performed by the IT Admin in UI
- targets: send gcid_change event to all Target Appliances of the Group
- gcid != mcid: Group's configuration is different from Master Appliance configuration in parts significant to this Group (e.g., local changes on the Master Appliance are ignored). For illustration purpose, a state transition to group-rendering state is shown for in-sync state
- group-rendering: a state in which Pulse One is still calculating new configuration for at least one of the Target Appliances in the group



State Transition Diagram for Target Appliances in the Group

Legend:

- mcid: Master Appliance Commit ID
- gcid: Group Commit ID - one of the recent Master Appliance's commit IDs considered to be a Group's current configuration. mcid will sometimes be ahead of gcid due to lag introduced by Group's publishing process
- rcid: Target Appliance Commit ID of the configuration running on the system
- tcid: Target Commit ID that Pulse One wants to be downloaded and applied by the Target Appliance. Target Commit ID is a reference to a configuration created by the Renderer module for a given appliance. It is a combination of Group's and Target Appliance's specific configuration
- gcidchanged: Signal that Group's commit ID has been changed. Only significant changes are reported as gcidchanged (i.e., changes of local significance to Master Appliance will not trigger this signal)
- conflict(rcid, tcid): there is a change of significance between configuration running on the Target Appliance and target configuration for this appliance



Publish Config

Initiate a publish of group's configuration to its Target Appliances. For details on functionality, see Entities section explaining config_state attribute of ApplianceGroupConfigSettingsEntity.

Request

- **Method:** POST
- **Resource:** /api/v1/sa/groups/{group-id}/commands/publish
- **JSON Data:** Request data should be in the form of a JSON body representing ApplianceGroupMembersEntity. This is the optional field. If it is empty
  or not existent, all members subscribed to the group will be considered as
  Target Appliances.

Response

- **Status:** 204

Example

Request

```
POST /api/v1/sa/config-groups/9ed4cea1-2f7b-4832-b5c1-82505548807e/commands/publish HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulsesecure.net

{
  "items": [ "c1e6e13b-0c76-49ec-a08a-5bcdfe66268b" ]
}
```

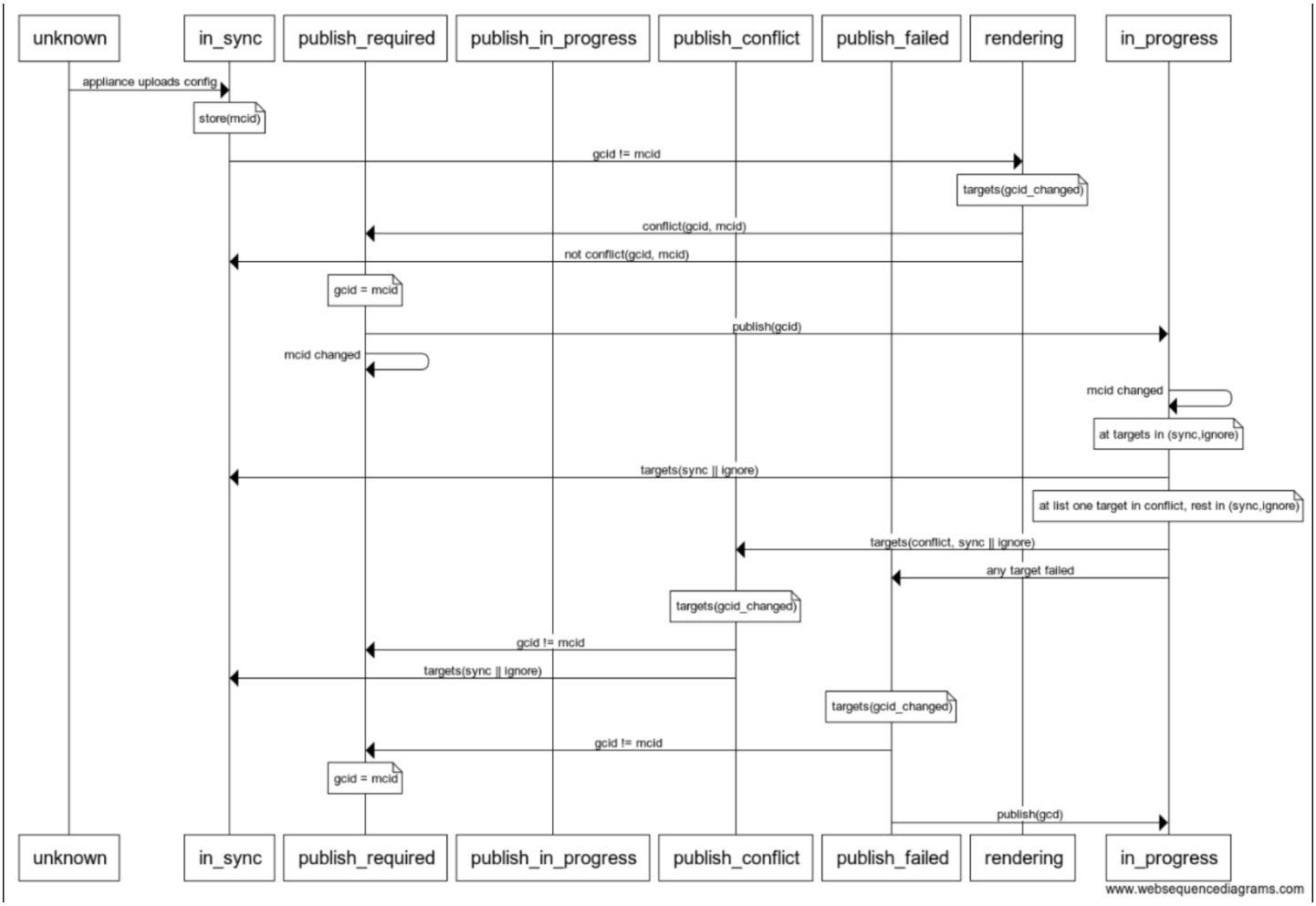
Response

```
HTTP/1.1 204 No Content
```

State Transition Diagram For The Group

Legend:

- mcid: Master Appliance Commit ID
- gcid: Group Commit ID - one of the recent Master Appliance's commit IDs considered to be a Group's current configuration. mcid will sometimes be ahead of gcid due to lag introduced by Group's publishing process
- publish: Action performed by the IT Admin in UI
- targets: send gcid_change event to all Target Appliances of the Group
- gcid != mcid: Group's configuration is different from Master Appliance configuration in parts significant to this Group (e.g., local changes on the Master Appliance are ignored). For illustration purpose, a state transition to group-rendering state is shown for in-sync state



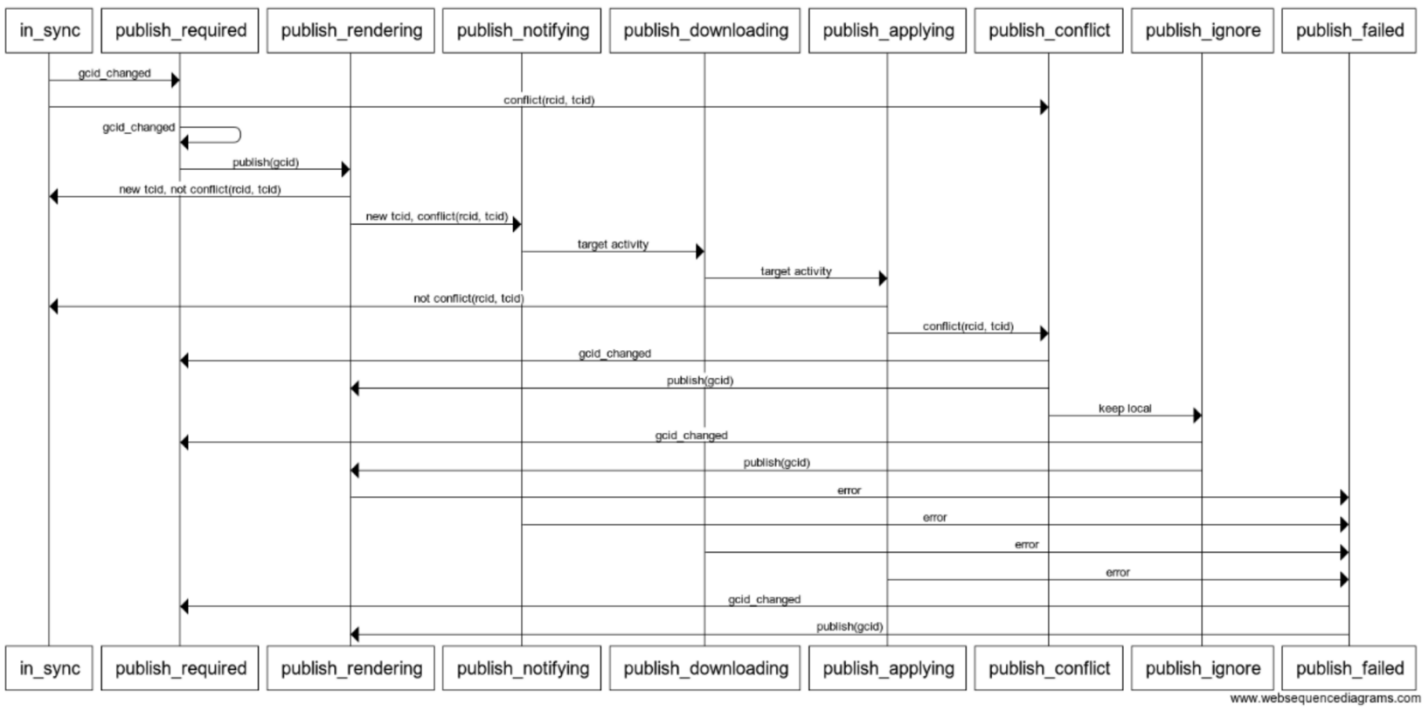

```
participant unknown
participant in_sync
participant publish_required
participant publish_in_progress
participant publish_conflict
participant publish_failed
participant rendering
unknown->in_sync: appliance uploads config
```

note over in_sync: store(mcid)
in_sync->rendering: gcid != mcid
note over rendering: targets(gcid_changed)
rendering->publish_required: conflict(gcid, mcid)
rendering->in_sync: not conflict(gcid, mcid)
note over publish_required: gcid = mcid
publish_required->in_progress: publish(gcid)
publish_required->publish_required: mcid changed
in_progress->in_progress: mcid changed
note over in_progress: at targets in (sync,ignore)
in_progress->in_sync: targets(sync || ignore)
note over in_progress: at list one target in conflict, rest in (sync,ignore)
in_progress->publish_conflict: targets(conflict, sync || ignore)
in_progress->publish_failed: any target failed
note over publish_conflict: targets(gcid_changed)
publish_conflict->publish_required: gcid != mcid
publish_conflict->in_sync: targets(sync || ignore)
note over publish_failed: targets(gcid_changed)
publish_failed->publish_required: gcid != mcid
note over publish_required: gcid = mcid
publish_failed->in_progress: publish(gcd)

State Transition Diagram For Target Appliances In The Group

Legend:

- mcid: Master Appliance Commit ID
- gcid: Group Commit ID - one of the recent Master Appliance's commit IDs considered to be a Group's current configuration. mcid will sometimes be ahead of gcid due to lag introduced by Group's publishing process
- rcid: Target Appliance Commit ID of the configuration running on the system
- tcid: Target Commit ID that Pulse One wants to be downloaded and applied by the Target Appliance. Target Commit ID is a reference to a configuration created by the Renderer module for a given appliance. It is a combination of Group's and Target Appliance's specific configuration
- gcid_changed: Signal that Group's commit ID has been changed. Only significant changes are reported as gcid_changed (i.e., changes of local significance to Master Appliance will not trigger this signal)
- conflict(rcid, tcid): there is a change of significance between configuration running on the Target Appliance and target configuration for this appliance



participant in_sync
participant publish_required
participant publish_rendering
participant publish_notifying
participant publish_downloading
participant publish_applying
participant publish_conflict
participant publish_ignore
participant publish_failed
in_sync->publish_required: gcid_changed
in_sync->publish_conflict: conflict(rcid, tcid)
publish_required->publish_required: gcid_changed
publish_required->publish_rendering: publish(gcid)
publish_rendering->in_sync: new tcid, not conflict(rcid, tcid)

publish_rendering->publish_notifying: new tcid, conflict(rcid, tcid)
publish_notifying->publish_downloading: target activity
publish_downloading->publish_applying: target activity
publish_applying->in_sync: not conflict(rcid, tcid)
publish_applying->publish_conflict: conflict(rcid, tcid)
publish_conflict->publish_required: gcid_changed
publish_conflict->publish_rendering: publish(gcid)
publish_conflict->publish_ignore: keep local
publish_ignore->publish_required: gcid_changed
publish_ignore->publish_rendering: publish(gcid)
publish_rendering->publish_failed: error
publish_notifying->publish_failed: error
publish_downloading->publish_failed: error
publish_applying->publish_failed: error
publish_failed->publish_required: gcid_changed
publish_failed->publish_rendering: publish(gcid)

Appliance Configuration

Hash IDs

This API uses sha1 hashes of contents. This has multiple advantages:

- A globally unique ID can be determined off of an object's contents
- Two objects with the same ID are statistically the same content
- Seeing if Pulse One is in sync with an appliance or vice-versa is just a matter of comparing IDs. All sha1 hashes are encoded as lowercase hex.

Commits

A commit object contains:

- A unique commit ID
- Commit meta information
- A reference to all config block IDs

A commit represents a specific set of configuration blocks. By simply comparing a commit ID it is possible to determine if an appliance's configuration is in sync with Pulse One and vice-versa.

- author: (utf8 str) Name and email of the last user to modify/create the commit. author is optional with caveats. See author for blocks above.
- date: (str) RFC-3339 formatted timestamp of the time of modification/creation. date is optional with caveats. See data for blocks above.
- blocks: Mapping of configuration block type to a list of block IDs. This allows blocks to repeat if the block type supports it. Order is preserved between blocks within a block type, However, order is not preserved between block types.

```
{
 "author": "Admin Name <admin@example.com>",
 "date": "2015-03-04T00:17:53Z",
 "blocks": {
  "{block_type}": ["{block_id}"],
  "system.configuration.security": [
     "0f63222d55444193e257eea1b3cf60102c07bbe7"
  ],
  "user-roles.user-role": [
     "c0d63554d6351cd0fc76e91c630b448808596ce0",
     "85bf0d4a0adb804506f0ee51eff1bd2fcf0a28d4"
  ]
 }
}
```

HEAD Commit

HEAD represents the current active commit for the appliance. It also MAY contain a prescribed commit information as an administrative request to change appliance's current active commit. The mechanism how this can be done is explained in head commit details.

Requests

Note: Most URLs do not include {device_id} because all config blocks and commits are shared for the domain.

PUT /api/v1/sa/{device_id}/configs/HEAD

Publish the commit ID for the configuration that the appliance is currently using. The content body of this operation is defined in Entities section by ApplianceConfigHeadCommitEntity and has commit value as mandatory. This value always represents the actual configuration of an appliance. In the case when commit value is not specified, 400 Bad Request is returned.
When used by an appliance, prescribed_commit is always empty.

- device_id: The appliance's Pulse One assigned ID
- Content-Type: application/json

Appliance PUT

This request is used by an appliance to inform Pulse One of appliance's actual configuration. The content body has commit set to the actual commit while prescribed_commit is empty. The commit must already be published. This will tell Pulse One what the complete configuration of the appliance is at the current time. Pulse One will track the commit history of the appliance. prescribed_commit is clear on successful PUT. Subsequent GET /api/v1/sa/{device_id}/configs/HEAD
calls will return empty prescribed_commit until a new prescribed commit gets created.

```
PUT /api/v1/sa/0f904595-7480-42f5-a07d-d2eda4cfa698/configs/HEAD
Content-Type: application/json
Content-Length: length
{
  "commit": "6962ca575e339134ea0ccf76720e13a7992ee0f8"
}
```

Administrative Rollback PUT

For rollback PUT, prescribed_commit has to be identical with the immediately preceding actual configuration of the appliance, otherwise request will fail with 409 Conflict error code. Once prescribed configuration is set to the rollback commit id, the appliance will have to rollback its configuration to get back into sync with Pulse One.
This request can only be invoked by Pulse One domain administrator.

```
PUT /api/v1/sa/0f904595-7480-42f5-a07d-d2eda4cfa698/configs/HEAD
Content-Type: application/json
Content-Length: length
{
  "commit": "6962ca575e339134ea0ccf76720e13a7992ee0f8",
  "prescribed_commit": "115ae467ed7f8b79a506ca9ce6642fa814237d46"
}
```

Administrative Roll-forward PUT

Roll-forward of prescribed_commit is an operation that puts Pulse One and an appliance into sync by accepting the actual commit to become the prescribed configuration for that appliance. For roll-forward, prescribed_commit has to be identical with the commit. Moreover, commit has to be identical with the actual commit of the appliance presently known to Pulse One. If any of these conditions are not satisfied, operation will fail with 409 Conflict error code.

This request can only be invoked by Pulse One domain administrator.

```
PUT /api/v1/sa/0f904595-7480-42f5-a07d-d2eda4cfa698/configs/HEAD
Content-Type: application/json
Content-Length: length
{
  "commit": "6962ca575e339134ea0ccf76720e13a7992ee0f8",
  "prescribed_commit": "6962ca575e339134ea0ccf76720e13a7992ee0f8"
}
```

Administrative Promote Pending Commit PUT

Promotion of pending_commit is an operation that prompts Pulse One to convert pending commit to the prescribed configuration. Pending commit is a commit generated by Pulse One and available through GET /api/security-appliances/{appliance-id} call defined in Appliances API section. For a successful promote operation, appliance's pending_commit in Pulse One has to be valid and non-empty value while prescribed_commit in the request has to be identical to it. Moreover, commit has to be identical with the actual commit of the appliance presently known to Pulse One. When prescribed_commit is different from pending_commit, 404 Not Found is returned. When pending_commit is not available, that means that Pulse One doesn't currently have any new configuration ready to be published to an appliance. In that case, promote operation fails for any value of prescribed_commit from the request, with 404 Not Found error code.
This request can only be invoked by Pulse One domain administrator.

```
PUT /api/v1/sa/0f904595-7480-42f5-a07d-d2eda4cfa698/configs/HEAD
Content-Type: application/json
Content-Length: length
{
  "commit": "6962ca575e339134ea0ccf76720e13a7992ee0f8",
  "prescribed_commit": "115ae467ed7f8b79a506ca9ce6642fa814237d46"
}
```

GET /api/v1/sa/{device_id}/configs/HEAD

Get the commit ID for the configuration that Pulse One prescribed for the appliance. Response body is defined by ApplianceConfigHeadCommitEntity in Entities section.

- device_id: The appliance's Pulse One assigned ID
- Content-Type: application/json

The prescribed commit ID is always returned along with the latest successfully PUT commit ID. If prescribed commit ID is not available with the latest commit ID, prescribed_commit value will not be present in the response body. If requested HEAD commit is not found (i.e. PUT never called), 204 No Content is returned.

```
Content-Type: application/json Content-Length: length
{
"commit": "6962ca575e339134ea0ccf76720e13a7992ee0f8"
"prescribed_commit": "115ae467ed7f8b79a506ca9ce6642fa814237d46"

}
```

PCS Session Management API

Fetch session information

This API will take a DSID as input and retrieve the username and the User's roles associated with that ID.
For a general guide and prerequisites on using PCS REST APIs please refer to the PCS/PPS : REST API Solutions Guide.

Request

- **Method**: GET
- **Resource**: /api/v1/sessions/info/<dsid>

Response

- Status:
  - 200 - When the request was authorized and valid session information were retrieved
  - 404 - Failed to export the session information from the database because the session is invalid
  - 500 - Failed to export the session information from the database because there was read error on one of the information
- JSON Data:
  Response data will be in the form of a JSON dictionary with the following keys:
  - roles: (list of *str*) List of all the roles that the user with this dsid is currently part of
  - username: (*str*) The username associated with the dsid

Example

Retrieving information with an invalid dsid
Request

```
GET   http://127.0.0.1:8090/api/v1/sessions/info/b4810731b0ff07f1d171cebce87c63f
Host: 127.0.0.1:8090
Accept: */*
```

Response

```
HTTP/1.1 404 NOT FOUND
Content-Type: application/json
Content-Length: 29

{"error": "Invalid session"}
```

Example - Retrieving information with a valid dsid

Request

```
GET   http://127.0.0.1:8090/api/v1/sessions/info/b4810731b0ff07f1d171cebce87c63fd
 Host: 127.0.0.1:8090
 Accept: */*
 HTTP/1.1 200 OK
 Content-Type: application/json
 Content-Length: 55

{
  "roles": [
    "Users",
    "SecondRole"
  ],
  "username": "user1"
}
```

Console API - Appliance Configuration

These APIs cover the configuration change indication which will allow the user to see that a new configuration has been uploaded and then drill down on the differences between the latest configuration and previous configurations.

Requests

GET /api/v1/sa/{device_id}/configs/commits

Return a list of config commits with created date and commit id for the last 30 days. The list will be ordered chronologically with the first record indicating the current config.

Response

- Status: 200
- JSON: Dictionary containing the following:
  - items: (list of ApplianceConfigCommitEntity.Commits)

Example

```
{
  "items": [
    {
      "id": "4e8b465aef1b3e24ef7b117ca98424b76ec7f735",
      "date": "2015-03-04T00:17:53Z"
    },
    {
      "id": "7b117ca98424b76ec7f7354e8b465aef1b3e24ef",
      "date": "2015-03-03T01:11:53Z"
    }
  ]
}
```

GET /api/v1/sa/configs/commits/diff?[from={commit_id1}]&to={commit_id2}

Return a list of blocks which are different between two commitids. To determine block differences, we will rely on the contentid since the block_id can change independently (based on metadata) of the actual block content.
If from parameter is not submitted, the diff will return all blocks of the to commit.

Response

- Status: 200
- JSON: Dictionary containing the following:
  - items: (list of ApplianceConfigBlockTypeDiffEntity)

Example


```
{
 "items": [
  {
  "type": "user-roles.user-role", "blocks": [

       {
         "change": "modified",
              "title": "User role
              data", "from": {
           "id": "00e9fcdcde2d912a72050637369d31c1e97a0115",
                "date": "2015-03-03T01:11:53Z",
           "content_id": "434fc1d1162f825ee92512ab1260893c1fc35254",
                "content_size": 1024
          },
          "to": {
           "id": "1e11cad47bc117c42f65be2f134ebe9ca892a957",
                "date": "2015-03-04T00:17:53Z",
           "content_id": "12ab1260893c1fc35254434fc1d1162f825ee925",
                "content_size": 2048
        }
       }
     ]
   },
   {

   "type": "system.configuration.security", "blocks":
          [
    {
      "change": "deleted",
              "title": "System
              Security", "from": {
        "id": "0637369d31c1e97a011500e9fcdcde2d912a7205",
                "date": "2015-03-04T00:17:53Z",
        "content_id": "fc35254434fc1d112ab125ee925260893c1162f8",
                "content_size": 6767
      },
      "to": null
    }
  ]
 },
 {
           "type": "system.resource.xyz",
           "blocks": [
             {
               "change": "added",
               "title": "System Resource",
               "from": null,
               "to": {
 "id": "e97a011500e9fcdcde2d912a72050637369d31c1", "date": "2015-03-04T00:17:53Z",
 "content_id": "3c1fc35254434fc1d1162f825ee92512ab126089", "content_size": 100000
             }
           }
         ]
       }
     ]
 }
```

Console API - EULA handling

This is documentation for the API endpoints related to management of EULA for users of Console UI.

See the *Errors* documentation for information about errors referenced in this document.

PUT (Signing of the current EULA)

Request

- **Method:** PUT
- **Resource:** /api/v1/users/{user_id}/eulas/{eula_uuid}
- **JSON Data:** Request data should be in the form of a JSON dictionary with the following structure:
  - agrees: (*bool*) explicit indication of the user's acceptance of this EULA

Response

- **Status:** 204

Example

Request

```
PUT /api/v1/users/1f7ac9e0-b3d1-11e4-99d6-0242ac11004b/eulas/22369d37-aab2-4b93-8644-0ddddc9f3df1
HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net

{
  "agrees": "true"
}
```

Response

```
HTTP/1.1 204 No Content
```

Get

Request

- **Method:** GET
- **Resource:** /api/v1/users/{use_id}/eulas/{eula_id}

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following structure:
  - (EulaEntity) A EULA's content

Example

Request

```
GET /api/v1/users/1f7ac9e0-b3d1-11e4-99d6-0242ac11004b/eulas/22369d37-aab2-4b93-8644-0ddddc9f3df1
HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 131

{
    "content": "EULA text here",
    "id": "22369d37-aab2-4b93-8644-0ddddc9f3df1",
    "created": "2015-01-23T23:53:09Z",
    "signed": "2015-03-23T23:53:09Z"
    "agrees": "True"
}
```

List

Request

- **Method:** GET
- **Resource:** /api/v1/users/{use_id}/eulas

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following structure:
  - items: (*list* of EulaEntity) A list of signed and unsigned EULA's for this user

Example

Request

```
GET /api/v1/users/1f7ac9e0-b3d1-11e4-99d6-0242ac11004b/eulas/22369d37-aab2-4b93-8644-0ddddc9f3df1
HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 231

{
   "items": [
      {
         "content": "EULA text here",
         "id": "22369d37-aab2-4b93-8644-0ddddc9f3df1",
         "created": "2015-01-23T23:53:09Z",
         "signed": "2015-03-23T23:53:09Z"
         "agrees": "True"
      },
      {
         "content": "EULA text here",
         "id": "22369d37-aab2-4b93-8644-0ddddc9f3df2",
         "created": "2015-03-23T23:53:09Z"
         "agrees": "False"
      },
   ]
}
```

Get Last Signed

Request

- **Method:** GET
- **Resource:** /api/v1/users/{user_id}/eulas/signed

Response

- Status: 200
  - JSON Data: Response data will be in the form of a JSON dictionary with the following structure:
    - (EulaEntity.Id) A EULA's Id
- **Status:** 404 Not found

Example

Request

```
GET /api/v1/users/1f7ac9e0-b3d1-11e4-99d6-0242ac11004b/eulas/signed HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 40

{
   "id": "22369d37-aab2-4b93-8644-0ddddc9f3df1"
}
```

Get Current Unsigned

Request

- **Method:** GET
- **Resource:** /api/v1/user/{user_id}/eulas/unsigned

Response

- **Status:** 204 Current EULA has already been signed, no need for further action for this user.
- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following structure:
  - (EulaEntity.Id) A EULA's Id

Example

Request

```
GET /api/v1/users/1f7ac9e0-b3d1-11e4-99d6-0242ac11004b/eulas/unsigned HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 40

{
   "id": "22369d37-aab2-4b93-8644-0ddddc9f3df1"
}
```

Console API - Policies

This is documentation for the API endpoints related to Policies within the Console UI.

Creation

Request

- **Method:** POST
- **Resource:** /api/policies
- **JSON Data:** A PolicyRequestEntity entity.

Response

- **Status:** 200
- **JSON Data:** Response data will be a Policy entity.

Example

Request

```
POST /api/policies HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net

{
    "ldap_groups": [
        {
            "id": "0a04ce58-73a9-4c83-b7a8-20d461bc5c93"
        }
    ],
    "name": "test-policy",
    "query": "tags: foobar",
    "type": "apps",
    "device_owner_mode": 0,
    "install_inside_geofencing_area": true,
    "geofencing_area": [
        {
            "id": "eb493232-be43-466b-bf14-af0f26c4e876",
            "name": "My Office",
            "address": "2700 Zanker Rd #200, San Jose, CA 95134",
            "radius": 1,
            "latitude": "+37.3908785",
            "longitude": "-121.9264253"
        }
    ]
}
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 235

{
    "created_on": "2015-02-17T21:57:43.642524",
    "edited": 1,
    "id": 13,
    "modified_on": "2015-02-17T21:57:43.642578",
    "ldap_groups": [
        {
            "id": "0a04ce58-73a9-4c83-b7a8-20d461bc5c93",
            "distinguished_name": "ou=people,dc=example,dc=com",
            "label": "People",
            "state": 1,
            "verified_time": "2015-02-13T22:39:08.575747"
        }
    ],
    "name": "test-policy",
    "query": "tags: foobar",
    "seq": 8,
    "state": "edited",
    "type": "apps",
    "device_owner_mode": 0,
    "install_inside_geofencing_area": true,
    "geofencing_area": [
        {
            "id": "eb493232-be43-466b-bf14-af0f26c4e876"
        }
    ]
}
```

Update

Request

- **Method:** PUT
- **Resource:** /api/policies/{policy-id}
- **JSON Data:** A PolicyRequestEntity entity.
  See the policy uniqueness rules above.

Response

- **Status:** 200
- **JSON Data:** Response data will be a Policy entity.

Example

Request

```
PUT /api/policies/13 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net

{
    "ldap_groups": [
        {
            "id": "0a04ce58-73a9-4c83-b7a8-20d461bc5c93"
        }
    ],
    "name": "test-policy-edited",
    "query": "tags: foobar",
    "device_owner_mode": 0,
    "install_inside_geofencing_area": true,
    "geofencing_area": [
        {
            "id": "eb493232-be43-466b-bf14-af0f26c4e876"
        }
    ]
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 235

{
    "created_on": "2015-02-17T21:57:43.642524",
    "edited": 1,
    "id": 13,
    "modified_on": "2015-02-17T21:59:43.642578",
    "ldap_groups": [
        {
            "id": "0a04ce58-73a9-4c83-b7a8-20d461bc5c93",
            "distinguished_name": "ou=people,dc=example,dc=com",
            "label": "People",
            "state": 1,
            "verified_time": "2015-02-13T22:39:08.575747"
        }
    ],
    "name": "test-policy-edited",
    "query": "tags: foobar",
    "seq": 8,
    "state": "edited",
    "type": "rules",
    "device_owner_mode": 0,
    "install_inside_geofencing_area": true,
    "geofencing_area": [
        {
            "id": "eb493232-be43-466b-bf14-af0f26c4e876",
            "name": "My Office",
            "address": "2700 Zanker Rd #200, San Jose, CA 95134",
            "radius": 1,
            "latitude": "+37.3908785",
            "longitude": "-121.9264253"
        }
    ]
}
```

Fetch

Request

- **Method:** GET
- **Resource:** /api/policies/{policy-id}

Response

- **Status:** 200
- **JSON Data:** Response data will be a Policy entity.

Example

Request

```
GET /api/policies/13 HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 235

{
    "created_on": "2015-02-17T21:57:43.642524",
    "edited": 1,
    "id": 13,
    "modified_on": "2015-02-17T21:59:43.642578",
    "ldap_groups": [
        {
            "id": "0a04ce58-73a9-4c83-b7a8-20d461bc5c93",
            "distinguished_name": "ou=people,dc=example,dc=com",
            "label": "People",
            "state": 1,
            "verified_time": "2015-02-13T22:39:08.575747"
        }
    ],
    "name": "test-policy-edited",
    "query": "tags: foobar",
    "seq": 8,
    "state": "edited",
    "type": "rules",
    "device_owner_mode": 0,
    "install_inside_geofencing_area": true,
    "geofencing_area": [
        {
            "id": "eb493232-be43-466b-bf14-af0f26c4e876",
            "name": "My Office",
            "address": "2700 Zanker Rd #200, San Jose, CA 95134",
            "radius": 1,
            "latitude": "+37.3908785",
            "longitude": "-121.9264253"
        }
    ]
}
```

Delete

Request

- **Method:** DELETE
- **Resource:** /api/policies/{policy-id}

A body is not required for deleting a policy.

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following keys:
  - msg: (*str*) A message regarding this request.
  - success: (*str*) True if the request to delete the policy was successful.

Example

Request

```
DELETE /api/policies/13 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 235

{
    "msg": "deleted policy group 13",
    "success": true
}
```

Console API - Properties

This is documentation for the API endpoints related to global properties and policy properties of Console UI.
See the Properties Entities documentation for information about entities referred to in this document.

Get Global Properties

Return a list of global properties.

Request

- **Method:** GET
- **Resource:** /api/properties

Response

- **Status:** 200
- **JSON Data:** Response data will be a list of DomainProperty:

Example

Request

```
GET /api/properties
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 2256

[
   {
      "read_only": false,
      "display_name": "welcome_timeout_hours",
      "name": "welcome_timeout_hours",
      "max_value": 720,
      "min_value": 1,
      "value": 48,
      "label": "Welcome Timeout Hours",
      "created_on": "2013-05-30T18:29:22",
      "prop_type": "int",
      "modified_on": "2014-06-20T17:31:10",
      "group": "Password",
      "choices": null,
      "id": 20,
      "sensitive": false
   }
   ...
]
```

Get Policy Properties

Return a list of policy properties by policy group id.

- **Method:** GET
- **Resource:** /api/policies/{policy-group-id}/properties

Response

- **Status:** 200
- **JSON Data:** Response data will be a JSON dictionary with the following structure:
  - properties: (*list* of *PolicyProperty*) A list of PolicyProperty entities
  - policies: (*list* of *Policy*) A list of Policy entities

Example

Request

```
GET /api/policies/1/properties
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 2256

{
    "properties": [
        {
            "display_name": "activesync_accept_all_certs",
            "name": "activesync_accept_all_certs",
            "max_value": null,
            "min_value": null,
            "policy_group_id": 1,
            "value": false,
            "policy_name": "Global",
            "platform": "all",
            "created_on": "2014-03-27T20:37:26",
            "choices": null,
            "prop_type": "bool",
            "modified_on": "2014-03-27T20:37:26",
            "group": "ActiveSync",
            "hidden": null,
            "label": "Activesync Accept All Certs",
            "id": 514
        }
        ...
    ],
    "policies": [
        {
            "policy_group_id": 1,
            "name": "Global"
        }
    ]
}
```

Console API - Registering

This is documentation for the API endpoint related to registering a workspace in the Console UI.

Workspace Registration

This endpoint is the entry-point for the workspace registration process. From here the user will be directed on how to complete registration such as downloading an Android app for their Android device, authenticating with their PIN and accepting a EULA.

Request

- **Method:** GET
- **Resource:** /register
- **Headers:**
  - User-Agent: This value will be used in determining the next step in the registration process. How this workflow is directed depends on the following values being found in this header value:
  - Android: The user is directed to download and install the Android app.
  - iPad or iPhone: The user is directed to the iOS registration workflow.

  If neither value is found in the User-Agent header value, the user is informed that their device is not supported for workspace registration.

Example

```
GET /register HTTP/1.1
Host: customer.pulseworkspace.net
User-Agent: Mozilla/5.0 (Linux; U; Android 4.0.3; ko-kr; LG-L160L Build/IML74K) AppleWebkit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30
```

Response

- **Status:** 200
- **Headers:**
  - Location: The location of the next step in the registration process. iOS users will be redirected to the iOS MDM daemon and Android users will be redirected to the Google Play market for downloading the Android app.

Example

```
HTTP/1.1 302 Found
Location: market://details?id=net.pulsesecure.workspace
```

Workspace Provisioning

Currently the registration route is not prefixed with /api/ even though this is a v0-style API. This pattern is preserved here in order to keep the existing route the same.

Request

- **Method:** POST
- **Resource:** /register/workspaces
- **Form Data:**
  - provision_email: (*str*) The email of the user registering, to be used for provisioning the workspace and associated with the workspace upon completion of registration process.

Example

```
POST /register/workspaces HTTP/1.1
Host: customer.pulseworkspace.net
Accept: text/html,application/xhtml+xml
Content-Type: application/x-www-form-urlencoded
Content-Length: 34

provision_email=test%40example.com
```

Response

The provisioning process will happen asynchronously and cannot be guaranteed at the time this request is received. The API can guarantee that it will attempt to provision a workspace for the provided email if the email is a valid customer email-domain.

- **Status:**
  - 200: The server will render a template instructing the user to check the email provided for provision_email. The email will contain a PIN and a link to follow to complete the registration process.
  - 404: The server will render a template providing a marketing message that entices them to sign up.

Example

```
HTTP/1.1 200 OK
Content-Length: 3425

<html>...</html>
```

Console API - Appliance Clusters

Read the API documentation for information about connecting and authenticating.
See the Entities documentation for information about entities referred to in this document.

Getting A List Of Clusters

Request

- **Method:** GET
- **Resource:** /api/v1/sa/clusters

Example

```
GET /api/v1/sa/clusters HTTP/1.1
Accept: application/json
Host: api.pulseworkspace.net
```

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following keys:
  - items: (*list*) A list of ApplianceClusterEntitys.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1024

{
  "items": [
    {
      "id": "bef5ad-ee31-4bd7-b2da-4d0b029f8945",
      "intrinsic_id": "2348-323-453",
      "name": "MyCluster1.hr.myorg.com",
      "type": "active-passive"
    },
    {
      "id": "5adbef-31ee-d74b-db2a-b029894d0f45",
      "intrinsic_id": "3284-232-434",
      "name": "MyCluster2.hr.myorg.com",
      "type": "active-active"
    },
  ]
}
```

Appliance Health Stats

This API documents the way in which an appliance can provide health statistics to Unity.
Read the API documentation for information about connecting and authenticating.
See the Entities documentation for information about entities referred to in this document.

Notifications

Unity can instruct an appliance to send current health statistics immediately using the following notification:

```
id: Unique notification ID
type: "system.health.stats" to indicate that the appliance should send all current health stats data.
{
  "id": "{alphanumeric string}",
  "type": "stats.health.pull"
}
```

Scheduling

For this version of the API it is expected that an appliance will send health statistics at a preconfigured interval of every 30 minutes.

Adding Health Statistics

Request

- **Method:** POST
- **Resource:** /api/v1/sa/{device-id}/health-stats
- device-id: *Required.* This is the value provided to the SA during registration
- **JSON Data:** Request data should be in the form of a JSON dictionary with the following keys:
  - stats: *Required.* A list of up to 100 ApplianceHealthStatsEntitys.
    All keys for ApplianceHealthStatsEntity are optional except the timestamp key.
    If timestamp is provided without any other keys, that entity is ignored.
    All missing keys are considered 'unavailable' at the time provided by timestamp.

Response

- **Status:** 204
- **Errors:**
  - 40001: Too Many Elements, see Errors for details.

Example

Request

```
POST /api/v1/sa/771b4124-f1b6-46ca-8035-172355924e02/health-stats HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: foobar.api.pulseworkspace.net

{
  "stats": [
    {
      "concurrent_users": 15038,
      "cpu_utilization": 46.2,
      "disk_utilization": 13.6,
      "memory_utilization": 68.02,
      "network_throughput": 76533456345,
      "timestamp": "2015-03-12T20:36:43.06Z"
    },
    {
      "concurrent_users": 15039,
      "timestamp": "2015-03-12T20:37:43.06Z"
    },
    {
      "concurrent_users": 15039,
      "activesync_users": 53,
      "cpu_utilization": 57.1,
      "disk_utilization": 13.8,
      "memory_utilization": 64.87,
      "network_throughput": 3257958496,
      "timestamp": "2015-03-12T20:38:43.06Z"
    },
    {
      "timestamp": "2015-03-12T20:39:43.06Z"
    }
  ]
}
```

Response

```
HTTP/1.1 204 No Content
```

Appliance Information

This document describes the API endpoint for retrieving appliance information.
Read the API documentation for information about connecting and authenticating.
See the Entities documentation for information about entities referred to in this document.

Retrieving appliance facts

Request

- **Method:** GET
- **Resource:** /api/v1/sa/{device-id}/info
- device-id: *Required* The appliance's Unity assigned ID.

Response

- **Status:** 200
- **JSON Data:** An ApplianceInfoEntity

Example

Request

```
GET /api/v1/sa/771b4124-f1b6-46ca-8035-172355924e02/info HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "appliance_version": "8.1R3.0-19864",
  "boot_timestamp": "2015-03-12T20:36:43.06Z",
  "last_config_timestamp": "2015-06-21T13:45:56.15Z",
  "if_map": "client",
  "concurrent_user_licenses": 5000,
  "concurrent_user_license_breakdown": {
    "access_licenses": 2000,
    "consec_licenses": 2000,
    "polsec_licenses": 1000
  }
  "license_role": null,
  "license_server": false,
```

```
  "cluster": {
    "id": "2348-323-453",
    "name": "MyCluster1.hr.myorg.com",
    "type": "active-passive",
    "node_name": "MyNodeNewName1",
    "leader_node": true,
    "active_node": false
  },
  "admin_url": "http://127.0.0.1/admin/config"
}
```

Cluster

Cluster Info
This endpoint provides a way to retrieve server cluster status and state.
cluster.info capability must be enabled to use this API.

Request

- **Method:** GET
- **Resource:** /api/v1/cluster/info

Example

```
GET /api/v1/cluster/info HTTP/1.1
Host: api.pulseworkspace.net
```

Response

- **Status:** 200
  - **JSON Data:**
    - A JSON dictionary representing ClusterInfoEntity entity
- **Status:** 204 - If the cluster info is not available.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: length

{
   "node": {
      "ip": "192.168.33.15",
      "id": "d6d2aac956c5482eb7612918fae385dc",
      "mode": "active",
      "hostname": "Zebra Node",
      "history": [
         {
            "mode": "standalone",
            "trigger": null,
            "time": "2019-07-15 03:00:00"
         },
         {
            "mode": "active",
            "trigger": "manual",
            "time": "2019-07-15 06:00:00"
         },
         {
            "mode": "standalone",
            "trigger": "auto_failover",
            "time": "2019-07-15 08:00:00"
         },
         # The last item in the list is the current state
```

```
            {
                "mode": "active",
                "trigger": "manual",
                "time": "2019-07-15 09:00:00"
            },
        ]
    },
    "cluster": {
        "nodes": ["192.168.33.15", "192.168.33.13"],
        "health": "green",
        "auto_failover": 10
    },
}
```

Appliance API

This is documentation for the API endpoints related to Appliances within the Console UI.
See the Errors documentation for information about errors referenced in this document.
Read the entities documentation for information about entities related to security appliances.

Creation

Request

- **Method:** POST
- **Resource:** /api/v1/sa
- **JSON Data:** Request data should be in the form of a JSON dictionary with supported fields defined in SecurityAppliance.Add. If orchestration field is presented in the request data, server will create an orchestrated appliance.

Response

- **Status:** 200
- **JSON Data:**
  - If the request is used for regular appliance creation, the response data will be in the form of a JSON dictionary with the following keys:
    - registration_code: A new code that can be used for appliance registration.
    - api_url: The URL to use for API requests. This will be entered in the appliance's UI by the admin.

If the request is used for orchestrated appliance creation:

- Response data will be a SecurityAppliance.Get entity.
- **Errors:**
  - 40906: orchestrated appliance creation is invalid. Could be one of the following reasons:
  - Create a SDP controller when there is already an existing controller.
  - Create a SDP controller in AWS.
  - Create a SDP gateway when there is not a SDP controller yet.

Regular Appliance Creation Example

Request

```
POST /api/v1/sa HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net

{
  "name": "Example Appliance"
  "dmi": {
    "username": "admindb",
    "password": "testpassword",
    "ip_address": "10.204.54.6,
    "port": "830"
  }
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 189

{
  "api_url": "example-api.pulseworkspace.net",
  "registration_code": "JZTAet96L"
}
```

Orchestration Appliance Creation Example

Request

```
POST /api/v1/sa HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net

{
  "name": "Example Appliance",
  "sdp_mode": "controller",
  "orchestration": {
    "type": "aws"
  }
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 189

{
  "id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
  "created": "2015-02-23T23:53:09Z",
  "name": "Excellent Appliance",
  "state": "unregistered",
  "notification_channel_status": "offline",
  "updated": "2015-02-23T23:53:09Z",
  "sdp_mode": "controller",
  "orchestration": {
    "type": "aws",
  }
  ...
}
```

Update

Request

- **Method:** PUT
- **Resource:** /api/v1/sa/{appliance-id}
- **JSON Data:** Request data should be in the form of a JSON dictionary with supported fields defined in SecurityAppliance.Update.
- name: (*str*) The updated name of the appliance.
- group_id: (*UUID*) The ApplianceGroup that this appliance belongs to, if any.
- sdp_mode: (*str*) (optional) Set the value to gateway or controller, to convert an already registered appliance to SDP gateway or SDP controller respectively. Possible errors:
    - 40906:
        - Selected appliance is already an SDP controller.
        - Selected appliance is already an SDP gateway.
        - There is not a SDP controller yet.
        - There is already an SDP controller.
        - Appliance version does not support SDP

Response

- **Status:** 200
- **JSON Data:** Response data will be a SecurityAppliance.Get entity.

Example

Request

```
PUT /api/v1/sa/f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net

{
  "group_id": "9ed4cea1-2f7b-4832-b5c1-82505548807e",
  "name": "Excellent Appliance",
  "sdp_mode": "gateway"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 189

{
  "created": "2015-02-23T23:53:09Z",
  "group_id": "9ed4cea1-2f7b-4832-b5c1-82505548807e",
  "id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
  "name": "Excellent Appliance",
  "state": "unregistered",
  "notification_channel_status": "offline",
  "updated": "2015-02-23T23:53:09Z",
  "sdp_mode": "controller",
  "orchestration": {
    "type": "aws",
    "state": "created"
  }
  ...
}
```

Delete

Request

- **Method:** DELETE
- **Resource:** /api/v1/sa/{appliance-id}

A body is not required for deleting an appliance.

Response

- **Status:** 204

Example

Request

```
DELETE /api/v1/sa/f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 204 No Content
```

Get By Id

Request

- **Method:** GET
- **Resource:** /api/v1/sa/{appliance-id}

Response

- **Status:** 200
- **JSON Data:** Request data should be in the form of a JSON dictionary with the following structure:
  - (*SecurityAppliance*) A SecurityAppliance entity

Example

Request

```
GET /api/v1/sa/1d89e147-a845-4825-93bf-154592454c25 HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 128

{
  "created": "2015-02-23T23:53:09Z",
  "group_id": "9ed4cea1-2f7b-4832-b5c1-82505548807e",
  "id": "afbf5c72a95a482eae6697b823318e1e",
  "name": "Excellent Appliance",
  "state": "registered",
  "notification_channel_status": "online",
  "updated": "2015-02-23T23:53:09Z"
  "appliance_commit_id": "7dae60fc51088a99a72bad5c1434d805fee469f0",
  "pending_commit_id": "761ef2209fb2e7c06d870a672057cd5be002e324",
  "config_size": "1389"
  "config_created": "2015-02-23T23:53:09Z",
```

```
  "type": "VPN",
  "model": "MAG4610",
  "serial_number": "0153M0TS00BII04U",
  "cluster": {
    "id": "bec7ed0a-7dcd-49d6-aa6c-ae5eaf47167f"
  }
  "dmi": {
    "username": "admindb",
    "ip_address": "10.204.54.6,
    "port": "830"
  }
}
```


List

Request

- **Method:** GET
- **Resource:** /api/v1/sa
- **Optional Parameters:**
  - search: (*str*) Limit results to appliances with a name, model, or appliance_version matching this string.
  - limit: (*int*) The maximum number of results to return. If not specified, the default limit is 10. The maximum is 100.
  - start: (*int*) The offset of the first result
  - sdp_mode: (*str*) SDP mode of the appliance. One of:
    - controller
    - gateway
    - all - Include all the SDP appliances
    - none - Include all the non-SDP appliances
  - orchestration_type: (*str*) Type of the orchestrated appliance. One of:
    - aws
    - vsphere
    - all - Include all the orchestrated appliances
    - none - Include all the non-orchestrated appliances

Response

- **Status:** 200
- **JSON Data:**
  - total: (*int*) The total number of appliances matching the search criteria.
  - items: (*list* of *SecurityAppliance.GetAll*) A list of SecurityAppliance.GetAll entities

Example

Request

```
GET /api/v1/sa?search=Excellent&orchestration_type=aws&sdp_mode=gateway HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 249

{
  "items": [
    {
      "created": "2015-02-23T23:53:09Z",
      "group_id": "9ed4cea1-2f7b-4832-b5c1-82505548807e",
      "id": "afbf5c72a95a482eae6697b823318e1e",
      "name": "Excellent Appliance",
      "state": "unregistered",
      "notification_channel_status": "offline",
      "updated": "2015-02-23T23:53:09Z"
      "appliance_commit_id": "7dae60fc51088a99a72bad5c1434d805fee469f0",
      "pending_commit_id": "761ef2209fb2e7c06d870a672057cd5be002e324",
      "config_size": "1389",
      "config_created": "2015-02-23T23:53:09Z",
      "type": "VPN",
      "model": "MAG4610",
      "serial_number": "0153M0TS00BII04U",
      "appliance_version": "8.1R4.1-32789",
      "sdp_mode": "gateway",
      "cluster": {
        "id": "bec7ed0a-7dcd-49d6-aa6c-ae5eaf47167f",
        "node_name": "MyNodeNewName1",
        "leader_node": true,
        "active_node": false
      }
      "dmi": {
        "username": "admindb",
        "ip_address": "10.204.54.6,
        "port": "830"
      },
      "orchestration": {
        "type": "aws",
        "state": "started"
      }
    }
  ],
  "total": 1
}
```

Regenerating Registration Codes

Administrators may need to regenerate a registration code if the code generated during creation of the appliance expires before the appliance can be registered with Unity.

Request

- Method: POST
- Resource: /api/v1/sa/{appliance-id}/commands/renew-registration-code

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following keys:
  - registration_code: A new code that can be used for appliance registration.
  - api_url: The URL to use for API requests. This will be entered in the appliance's UI by the admin.

Example

Request

```
POST /api/v1/sa/f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b/commands/renew-registration-code HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 189

{
  "api_url": "example-api.pulseworkspace.net",
  "registration_code": "JZTAet96L"
}
```

Reboot Appliance

Initiate a reboot of an appliance.

Request

- **Method:** POST
- **Resource:** /api/v1/sa/{appliance-id}/commands/reboot
- A body is not required for rebooting an appliance.

Response

- **Status:** 204

Example

Request

```
POST /api/v1/sa/f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b/commands/reboot HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 204 No Content
```

Perform actions on the Orchestrated Appliance

Start/stop/destroy/update an orchestrated appliance.

Request

- **Method:** POST
- **Resource:** /api/v1/sa/{appliance-id}/commands/{actions}
- action can be one of the following:
  - start
  - stop
  - update
  - destroy

Response

- **Status:** 204
- **Errors:**
  - 40905 If the orchestrated appliance state transition is invalid.
  - 50031 If the appliance is not orchestrated.
  - 50032 If the orchestrated appliance does not have config yet.
  - 40907: If the admin action on orchestrated appliance is invalid. Could be one of the following reasons:
    - Destroy a SDP controller when there is still at least one active SDP gateway.

Example

Request

```
POST /api/v1/sa/f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b/commands/start HTTP/1.1
Content-Length: 0
Host: api.pulseone.net
POST /api/v1/sa/f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b/commands/stop HTTP/1.1
Content-Length: 0
Host: api.pulseone.net
POST /api/v1/sa/f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b/commands/update HTTP/1.1
Content-Length: 0
Host: api.pulseone.net
POST /api/v1/sa/f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b/commands/destroy HTTP/1.1
Content-Length: 0
Host: api.pulseone.net
```

Response

```
HTTP/1.1 204 No Content
```

Cloud API - Uploading Activity Records

This is documentation for how a PPS or PCS appliance uploads activity records to the Cloud. Activity records are inspired by RFC 5424, The Syslog Protocol.
JSON bodies are expanded, for readability.
Read the API documentation for information about connecting and authenticating.

PUT Request

- **Method:** PUT
- **Resource:** /api/v1/sa/{device_id}/activities/{activity_id}
- *activity_id* is a permanently unique id. It is a 40-character string that uniquely identifies the activity object. It is simply a sha1 hash of the body of the activity in Canonical JSON form. The hash is encoded as lowercase hex.
- **JSON Data:** Request data should be a JSON body representing an ActivityEntity.Update.
  If the activity references a notification, the JSON body should represent an ActivityEntity.Reference entity. In this case, the reference values are as follows:
  - `reference.id`: This value is the `id` value from the notification.
  - `reference.type`: This value has the form `notification:{notification_type}` where `{notification_type}` is replaced with the `type` value from the notification.

  For both entities, the optional `params` value may contain the following keys:
  - `conflicting_operation`: value can be `rsa_key` or `upgrade`

Response

- **Status:** 204

Example Request

```
PUT /api/v1/sa/{device_id}/activities/da39a3ee5e6b4b0d3255bfef95601890afd80709 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseworkspace.net

{
  "activity_type": "requested_operation.not_performed",
  "message": "Could not reboot because an RSA operation was in progress",
  "params": {
    "conflicting_operation": "rsa_key"
  },
  "reference": {
    "id": "abka7890v",
    "type": "notification:system.operations.appliance.reboot"
  },
  "severity": "error",
  "time": "2015-06-23T05:59:03Z"
}
```

Expressing Appliance Log Messages as Activity Records

Today, the appliance can send its System, Admin Access, and User Access log messages to syslog sinks. It would make some sense for the appliance to be able to send equivalent messages to Unity.
Critical event log messages are especially relevant. They indicate that the appliance is not well. Hence, Appliance Critical events are sent as activities to Pulse One, so that Pulse One can alert administrators of important events.
Rather than creating a separate API for the purpose, we use this activity uploading API.
Read the Entities documentation for information about activity-log entities.
An appliance's Critical event logs are mapped to ActivityEntity.Update as follows:

- message: **Required.** Human readable event log message (which includes dynamic arguments)
- time: **Required.** Timestamp of when this event log occurred.
- severity: **Required.** Value will be critical for Critical Events
- activity_type: **Required.** system.event.logged
- params: message_id: Message ID of the event log.

Example Appliance Critical Log Message as Activity Record

```
PUT /api/v1/sa/{device_id}/activities/da39a3ee5e6b4b0d3255bfef95601890afd80709 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseworkspace.net

{
  "activity_type": "system.event.logged",
  "message": "[127.0.0.1] - System()[] - Program dshealthstatsunity recently failed.",
  "params": {
    "message_id": "ERR31093"
  },
  "severity": "critical",
  "time": "2015-06-23T05:59:03Z"
}
```

Getting An Activity

Request

- **Method:** GET
- **Resource:** /api/v1/sa/{device-id}/activities/{activity-id}
- device-id: *Required.* This is the value provided to the SA during registration
- activity-id: *Required.* This is the id of the activity.

Example

```
GET /api/v1/sa/771b4124-f1b6-46ca-8035-
172355924e02/activities/da39a3ee5e6b4b0d3255bfef95601890afd80709 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseworkspace.net
```

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON body representing an ActivityEntity.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 3567

{
  "activity_type": "requested_operation.not_performed",
  "actor": "notification-abka7890v",
  "message": "Could not reboot because an RSA operation was in progress",
  "params": {
    "conflicting_operation": "rsa_key"
  },
  "reference": {
    "id": "abka7890v",
    "type": "notification:system.operations.appliance.reboot"
  },
  "severity": "error",
  "target": "security_appliance-771b4124-f1b6-46ca-8035-172355924e02",
  "timestamp": "2015-06-23T05:59:03Z"
}
```

Getting All Activities For An Appliance

This request covers getting all activities for a specific appliance.

Request

- **Method:** GET
- **Resource:** /api/v1/sa/{device-id}/activities
- device-id: *Required.* This is the value provided to the SA during registration
- **Parameters:** The following request parameters are optional:
    - start: (*int*) The starting position for paging through results.
    - limit: (*int*) The limit of the number of documents to fetch.
    - sort: (*str*) The document field by which to sort.
    - direction: (*str*) The direction of sorting. Supported values are asc and desc.
    - fields: (*str*) A comma-delimited list of the fields of the documents to return.

Example

```
GET /api/v1/sa/771b4124-f1b6-46ca-8035-172355924e02/activities?limit=2&start=0&sort=time&direction=desc
HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseworkspace.net
```

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following keys:
  - items: (*list*) A list of ActivityEntitys.
  - total: (*int*) The total number of activities for this request.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 35671

{
  "items": [
    {
      "activity_type": "requested_operation.succeeded",
      "actor": "notification-abka7890v",
      "message": "Reboot successful.",
      "reference": {
        "id": "abka7890v",
        "type": "notification:system.operations.appliance.reboot"
      },
      "severity": "notice",
      "target": "security_appliance-771b4124-f1b6-46ca-8035-172355924e02",
      "time": "2015-06-23T06:05:03Z"
    },
    {
      "activity_type": "requested_operation.not_performed",
      "actor": "notification-abka7890v",
      "message": "Could not reboot because an RSA operation was in progress",
      "params": {
        "conflicting_operation": "rsa_key"
      },
      "reference": {
        "id": "abka7890v",
        "type": "notification:system.operations.appliance.reboot"
      },
      "severity": "error",
      "target": "security_appliance-771b4124-f1b6-46ca-8035-172355924e02",
      "time": "2015-06-23T05:59:03Z"
    }
  ],
  "total": 2
}
```

Getting Activities for All Appliances by Severity

This request covers getting all activities, filterable by severity, across all appliances sorted by most recent activity.

Request

- **Method:** GET
- **Resource:** /api/v1/sa/activities
- **Parameters:** The following request parameters are optional:
  - severity: (*str*) Activity severity. Must be one of: emergency, alert, critical, error, warning, notice, informational or debug per ActivityEntity.
  - start: (*int*) The starting position for paging through results.
  - limit: (*int*) The limit of the number of activities to fetch. The default is 20 and the max is 100.

Example

```
GET /api/v1/sa/activities?severity=critical&limit=2&start=0 HTTP/1.1
Accept: application/json
Host: api.pulseworkspace.net
```

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following keys:
  - items: (*list*) A list of ApplianceActivityEntitys.
  - total: (*int*) The total number of activities for this request.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1002

{
  "items": [
    {
      "activity_type": "system.event.logged",
      "actor": null,
      "message": "[127.0.0.1] - System()[] - Program dshealthstatsunity recently failed.",
      "params": {
        "message_id": "ERR31093"
      },
      "severity": "critical",
      "target": "security_appliance-771b4124-f1b6-46ca-8035-172355924e02",
      "time": "2015-06-23T06:05:03Z",
      "appliance": {
        "id": "771b4124-f1b6-46ca-8035-172355924e02",
        "name": "appliance1"
      }
    },
```

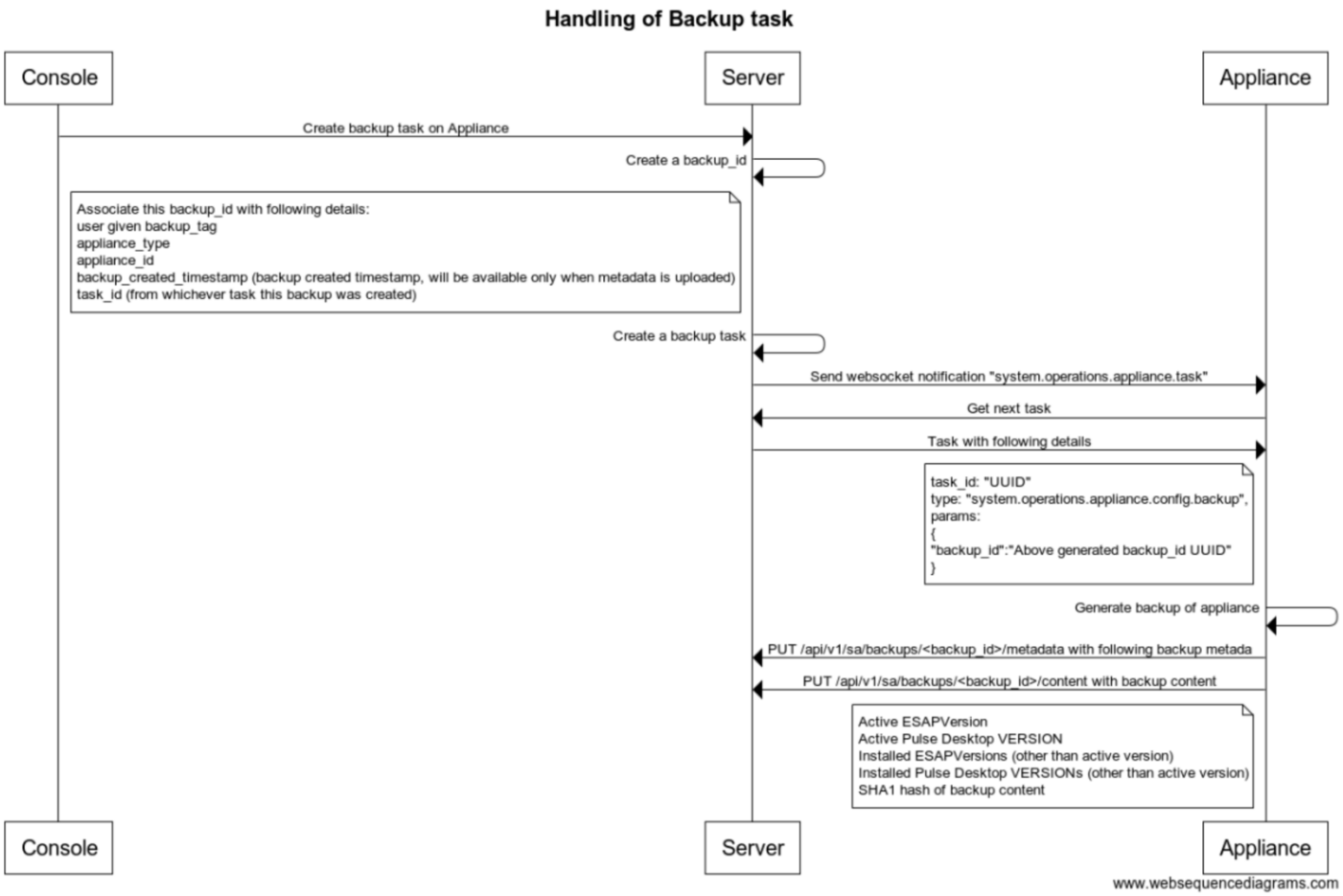
```
    {
      "activity_type": "system.event.logged",
      "actor": null,
      "message": "[127.0.0.1] - System()[] - Program dshealthstatsunity recently failed.",
      "params": {
        "message_id": "ERR31093"
      },
      "severity": "critical",
      "target": "security_appliance-d6972cad-efcb-48ef-bd07-e8ea266a946e",
      "time": "2015-06-23T05:59:03Z",
      "appliance" : {
        "id": "d6972cad-efcb-48ef-bd07-e8ea266a946e",
        "name": "appliance2"
      }
    }
  ],
  "total": 2
}
```

Managed Appliance backups management API

This API allows Pulse One to manage backups of managed appliances. These API would be used in following scenarios:
An appliance is tasked to create a backup of its configuration in Pulse One
An appliance is tasked to restore its configuration from a backup stored in Pulse One
Console wants to list and manage all managed appliance's backups stored in Pulse One

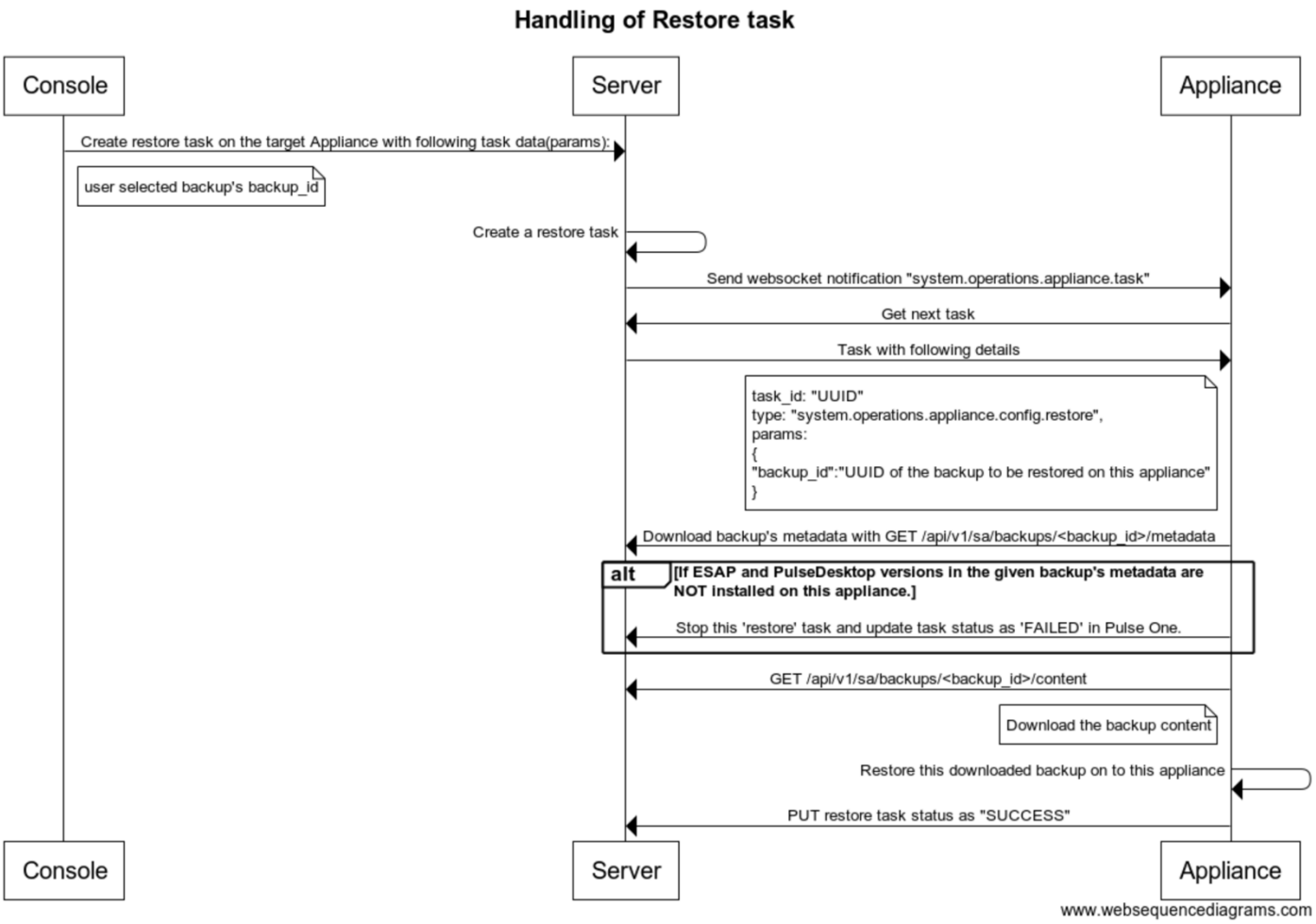
Pulse One server represents any backup of a managed appliance in two parts:

- Metadata that describes the backup
- Content which is actual binary content of the backup



Handling of Backup task

Console->Server: Create backup task on Appliance
Server->Server: Create a backup_id
note left of Server
Associate this backup_id with following details:
user given backup_tag
appliance_type
appliance_id
backup_created_timestamp (backup created timestamp, will be available only when metadata is uploaded)
task_id (from whichever task this backup was created)
end note
Server->Server: Create a backup task
Server->Appliance: Send websocket notification "system.operations.appliance.task"
Appliance->Server: Get next task
Server->Appliance: Task with following details
note left of Appliance
task_id: "UUID"
type: "system.operations.appliance.config.backup",
params:
  {
  "backup_id":"Above generated backup_id UUID"
  }
end note
Appliance->Appliance: Generate backup of appliance
Appliance->Server: PUT /api/v1/sa/backups/<backup_id>/metadata with following backup metada
Appliance->Server: PUT /api/v1/sa/backups/<backup_id>/content with backup content
note left of Appliance
Active ESAPVersion
Active Pulse Desktop VERSION
Installed ESAPVersions (other than active version)
Installed Pulse Desktop VERSIONs (other than active version)
SHA1 hash of backup content
end note



Handling of Restore task

Console->Server: Create restore task on the target Appliance with following task data(params):
note right of Console
user selected backup's backup_id
end note
Server->Server: Create a restore task
Server->Appliance: Send websocket notification "system.operations.appliance.task"
Appliance->Server: Get next task
Server->Appliance: Task with following details
note left of Appliance
task_id: "UUID"
type: "system.operations.appliance.config.restore",
params:
  {
  "backup_id":"UUID of the backup to be restored on this appliance"
  }
end note

Appliance->Server: Download backup's metadata with GET /api/v1/sa/backups/<backup_id>/metadata
alt If ESAP and PulseDesktop versions in the given backup's metadata are NOT installed on this appliance.
Appliance->Server: Stop this 'restore' task and update task status as 'FAILED' in Pulse One.
end
Appliance->Server: GET /api/v1/sa/backups/<backup_id>/content
note left of Appliance
Download the backup content
end note
Appliance->Appliance: Restore this downloaded backup on to this appliance
Appliance->Server: PUT restore task status as "SUCCESS"

PUT /api/v1/sa/backups/{backup_id}/metadata

This API can be used by appliance to set the backup's metadata, after backup content has been uploaded to Pulse One successfully.
If the metadata hasn't been submitted within 90 minutes of backup task creation, the task will be moved to failed state.

Request

- **Method:** PUT
- **Authorization:** Requires appliance
- **Resource:** /api/v1/sa/backups/{backup_id}/metadata
- **JSON Data:** A JSON dictionary representing an ApplianceBackupMetadataUpdateEntity entity.

Response Success

- **Status:** 200

Response Error

- **Status:** 404 (Not Found, if incorrect backup_id)
- **Status:** 409 (Conflict, if content is already available for this backup_id)

Example

Request

```
PUT /api/v1/sa/backups/28374bca-efef-11f5-8e5b-0242ac13000d/metadata HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
{
  "active_esap_version": "5635",
  "active_pulse_desktop_version": "3458",
  "installed_esap_versions": ["5636", "5637", "5638"],
  "installed_pulse_desktop_versions": ["3458", "3459", "3460"],
  "content_hash": "088623438e5e50d53b7cfe8d50943b90ea1989a9",
  "content_size": 368228
}
```

Response

```
HTTP/1.1 200 OK
```

HEAD /api/v1/sa/backups/{backup_id}/metadata

Check to see if a backup with the given backup_id exists in Pulse One.
If the backup exists, expect a 200 response with at least Content-Length of the backup content (in bytes).
If the backup does not exist, expect a 404.

Request

- **Method:** HEAD
- **Authorization:** Requires appliance (or) admin.appliances.backup|READ
- **Resource:** /api/v1/sa/backups/{backup_id}/metadata

Response

- **Status:** 200
- **Status:** 404 (If backup with given backup_id does not exist)

Example

```
HTTP/1.1 200 OK
Content-Length: 1230384
```

GET /api/v1/sa/backups/{backup_id}/metadata

API to get metadata of a backup with the given backup_id
If the backup_id does not exist, expect a 404.

Request

- **Method:** GET
- **Authorization:** Requires appliance (or) admin.appliances.backup|READ
- **Resource:** /api/v1/sa/backups/{backup_id}/metadata

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing an ApplianceBackupMetadataGetEntity entity.

Example

Request

```
GET /api/v1/sa/backups/28374bca-efef-11f5-8e5b-0242ac13000d/metadata HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 80

{
   "backup_id": "28374bca-efef-11f5-8e5b-0242ac13000d",
   "name": "My Backup-1",
   "description": "After fixing Users role and realm",
   "appliance_id": "28374bca-efef-11f5-8e5b-0242ac13000d",
   "type": "VPN",
   "created": "2017-09-04T18:55:38",
   "active_esap_version": "5634",
   "active_pulse_desktop_version": "3456",
   "installed_esap_versions": ["5636", "5637", "5638"],
   "installed_pulse_desktop_versions": ["3458", "3459", "3460"],
   "content_hash": "088623438e5e50d53b7cfe8d50943b90ea1989a8",
   "content_size": 368263
}
```

DELETE /api/v1/sa/backups/{backup_id}

Pulse One Console can use this API to delete a backup's metadata and content.

Request

- **Method:** DELETE
- **Authorization:** admin.appliances.backup|DELETE
- **Resource:** /api/v1/sa/backups/{backup_id}

Response

- **Status:** 204 (Successful)
- **Status:** 404 (Not Found)

GET /api/v1/sa/backups

Pulse One Console can use this API to get details of all managed appliance backups.

Request

- **Method:** GET
- **Authorization:** Requires appliance (or) admin.appliances.backup|READ
- **Resource:** /api/v1/sa/backups

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a list of ApplianceBackupMetadataGetEntity entities.

Example

Request

```
GET /api/v1/sa/backups HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 80

{
  "backups": [
    {
      "backup_id": "28374bca-efef-11f5-8e5b-0242ac13000d",
      "name": "My Backup-1",
      "description": "After fixing Users role and realm",
      "appliance_id": "28374bca-efef-11f5-8e5b-0242ac13000d",
      "type": "NAC",
      "created": "2017-09-04T18:55:38",
      "active_esap_version": "5634",
      "active_pulse_desktop_version": "3456",
      "installed_esap_versions": ["5636", "5637", "5638"],
      "installed_pulse_desktop_versions": ["3458", "3459", "3460"],
      "content_hash": "088623438e5e50d53b7cfe8d50943b90ea1989a8",
      "content_size": 368263
    },
    {
      "backup_id": "28374bca-efef-11f5-8e5b-0242ac13000e",
      "name": "My Backup-2",
      "description": "After fixing VPN connection IPv4 pools",
      "appliance_id": "28374bca-efef-11f5-8e5b-0242ac13000e",
      "type": "VPN",
```

```
    "created": "2017-09-05T18:55:38",
    "active_esap_version": "5635",
    "active_pulse_desktop_version": "3458",
    "installed_esap_versions": ["5636", "5637", "5638"],
    "installed_pulse_desktop_versions": ["3458", "3459", "3460"],
    "content_hash": "088623438e5e50d53b7cfe8d50943b90ea1989a9",
    "content_size": 368228
   }
  ]
}
```


PUT /api/v1/sa/backups/{backup_id}/content

Using this API Appliance can upload content of a backup.
If the content hasn't been submitted within 90 minutes of backup task creation, the task will be moved to failed state.

Request

- **Method:** PUT
- **Resource:** /api/v1/sa/backups/{backup_id}/content
- **Authorization:** Requires appliance
- **Content-Type:** application/octet-stream
- **Body:** The binary contents of the backup

Response

- **Status:** 204 (Successful)
- **Status:** 404 (Not Found, if incorrect backup_id)
- **Status:** 409 (Conflict, if metadata with size/hash was not already uploaded to server (or) uploaded content's hash/size does not match to this backup's metadata)

GET /api/v1/sa/backups/{backup_id}/content

Using this API Appliance can get content of a backup

Request

- **Method:** GET
- **Resource:** /api/v1/sa/backups/{backup_id}/content
- **Authorization:** Requires appliance

Response

- **Content-Type:** application/octet-stream
- **Body:** The binary contents of the backup
- **Status:** 204 (Successful) or 404 (Not Found, if incorrect backup_id)

Console API - CloudSecure Data

This is documentation for the API endpoint for retrieving CloudSecure stats.
Read the API documentation for information about connecting and authenticating.
See the Entities documentation for information about entities referred to in this document.

Aggregated CloudSecure Statistics

Request

- **Method:** GET
- Authorization**:** admin.appliances|READ
- **Resource:** /api/v1/sa/stats/cloudsecure? [appliance_ids=<appliance_ids>]&[limit=<max_entries>]& [duration=<duration>]&[resolution=<sample_resolution>]& [timestamp_start=<epoch_time>]&[timestamp_end=<epoch_time>]

Returns aggregated count of the cloud secure applications, device types and role details including the trend for the applications for the given duration with provided resolution.
Either duration or timestamp_start, timestamp_end parameters need to be passed in the API. If duration parameter is passed, data will be returned considering current time as the end time. If none of these duration parameters are provided, data will be returned for last 24 hours with hourly resolution.

Request Parameters

- appliance_ids: (*str*) Comma separated list of appliance ids. If appliance IDs are not provided, then data will be returned for the
  entire domain. (Default: None)
- limit: (*int*) Limit the number of entries for which data needs to be returned. This parameter must be greater than 0. If this parameter is passed as 0 or less than 0, returns invalid request error. If this parameter is not passed, entire data is returned.
  (Default: None)
- duration: (*int*) Desired duration for which data needs to be returned. This parameter must be greater than 0. If this parameter is passed as 0 or less than 0, returns invalid request error. If duration is specified along with timestamp_start and timestamp_end, returns invalid request error. Duration is specified in days. (Default: None)
- resolution: (*str*) Desired resolution of the values. Values:
  - minute: Minute resolution.
  - hour: Hour resolution.
  - day: Daily resolution.
  - week: Weekly resolution.
  - month: Monthly resolution. (Default: hour)
- timestamp_start: (int) Start time of the duration. (epoch timestamp). If this parameter is not provided, then data will be returned without enforcing the start time.
- timestamp_end: (int) End time of the duration. (epoch timestamp).
  If this parameter is not provided, then data will be returned without enforcing the end time.
  If this parameter value is smaller than timestamp_start parameter, invalid request error is returned.

Response

- **Status:** 200
- **JSON Data:** An CloudSecureStatsAggregationEntity
  - JSON Object with list of aggregated cloud secure applications, device types and role details. It also provides the application trend details for the given duration.
  - Returns empty JSON object if application result is not available

Example

Request

```
GET /api/v1/sa/stats/cloudsecure?duration=1&resolution=hour HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length:

{
   "timestamp_start": "2017-01-06T00:02:18Z",
   "timestamp_end": "2017-01-07T00:02:18Z",
   "resolution": "hour",
   "duration": 1,
   "applications": {
      "successful": {
         "Dropbox": 7,
         "OneDrive": 4,
         "Outlook": 10,
         "Salesforce": 9,
         "Zendesk": 20
      },
      "failed": {
         "Dropbox": 6,
         "OneDrive": 8,
         "Outlook": 4,
         "Salesforce": 12,
         "Zendesk": 10
      },
      "total_successful": 50,
      "total_failed": 40,
      "stats": [
         {
            "apps_count": {
               "Dropbox": 5,
               "OneDrive": 4,
               "Outlook": 10,
               "Salesforce": 9,
               "Zendesk": 20
            },
            "timestamp": "2017-01-06T00:02:18Z"
         },
```

```
        {
            "apps_count": {
                "Dropbox": 5,
                "OneDrive": 4,
                "Outlook": 10,
                "Salesforce": 9,
                "Zendesk": 20
            },
            "timestamp": "2017-01-06T01:02:18Z"
        },
        {
            ...
        }
    ]
  }
  "devices": {
     "iOS": 26,
     "Android": 24,
     "Windows": 19,
     "Maintosh": 15,
     "Windows Mobile": 12
  },
  "compliance": {
     "compliant": 24
     "not_compliant: 12,
     "not_assessed": 15,
     "not_applicable": 19,
  },
  "roles": {
     "Employees": 35,
     "Contractors": 21,
     "Guest": 15,
     "Full Access": 10,
     "Limited Connectivity": 15
  },
}
```

API for Configuration Upload and Download Status

This documents how a PCS or PPS informs Pulse One about configuration uploads and downloads that are pending or in progress.
JSON bodies are expanded, for readability.
Read the API documentation for information about connecting and authenticating.
The [Entities][entities.md] document defines the ConfigStatusEntity.

PUT Request

- **Method:** PUT
- **Resource:** /api/v1/sa/{device_id}/configs/status
- **JSON Data:** Request data should be a JSON body representing a ConfigStatusEntity.
  Sparse updates are permitted: if the caller omits fields, the values are unchanged.

Response

- **Status:** 204

Example

```
PUT /api/v1/sa/{device_id}/configs/status
Content-Type: application/json
Accept: application/json
Host: api.pulseworkspace.net

{
  "upload": {
    "status": "in_progress"
  }
}
```

GET Request

- **Method:** GET
- **Resource:** /api/v1/sa/{device_id}/configs/status

Response

- **Status:** 200
- **JSON Data:** A JSON body representing a ConfigStatusEntity

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: {nnn}

{
  "upload": {
    "status": "in_progress"
  }
  "download": {
    "status": "idle"
  }
}
```

Orchestration Config API

Get Orchestration Config

This API allows the IT admin to get the orchestration config by appliance ID.

Request

- **Method:** GET
- **Resource:** /api/v1/sa/{appliance_id}/orchestration

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing OrchestrationConfigEntity entity

Example

Request

```
GET /api/v1/sa/1d89e147-a845-4825-93bf-154592454c25/orchestration HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 128

{
  "appliance_id": "1d89e147-a845-4825-93bf-154592454c25",
  "id": "bec7ed0a-7dcd-49d6-aa6c-ae5eaf47167f",
  "appliance_config": {
    "admin_username": "admin",
    "internal_fqdn": "internal.pulseworkspace.net",
    "external_fqdn": "external.pulseworkspace.net",
    ...
    "management_ip_address": "10.204.54.6"
  },
  "deployment_config": {
    "id": "28374bca-efef-11f5-8e5b-0242ac13000d"
    "vsphere_config": {
      "datacenter_name": "datacenter-1",
      "datastore_name": "datastore-1",
      ...
      "appliance_master_template_name": "test-master-template"
    }
  }
}
```

Create Orchestration Config

This API allows the IT Admin to create an orchestration config.

Request

- **Method:** POST
- **Resource:** /api/v1/sa/{appliance_id}/orchestration
- **JSON Data:** A JSON dictionary representing OrchestrationConfigEntity entity

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing OrchestrationConfigEntity entity
- **Errors:** 404 If the appliance DI is invalid.

Example

Request

```
POST /api/v1/sa/1d89e147-a845-4825-93bf-154592454c25/orchestration HTTP/1.1
Accept: application/json
Host: api.pulseone.net

{
  "service_account_id": "bec7ed0a-7dcd-49d6-aa6c-ae5eaf47167f",
  "appliance_config": {
    "admin_username": "admin",
    "admin_password": "password",
    "internal_fqdn": "internal.pulseworkspace.net",
    "external_fqdn": "external.pulseworkspace.net",
    ...
    "management_ip_address": "10.204.54.6"
  },
  "deployment_config": {
    "vsphere_config": {
      "datacenter_name": "datacenter-1",
      "datastore_name": "datastore-1",
      ...
      "appliance_master_template_name": "test-master-template"
    }
  }
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: <length>

{
  "appliance_id": "1d89e147-a845-4825-93bf-154592454c25",
  "service_account_id": "bec7ed0a-7dcd-49d6-aa6c-ae5eaf47167f",
  "appliance_config": {
    "admin_username": "admin",
    "internal_fqdn": "internal.pulseworkspace.net",
    "external_fqdn": "external.pulseworkspace.net",
    ...
    "management_ip_address": "10.204.54.6"
  },
  "deployment_config": {
    "id": "28374bca-efef-11f5-8e5b-0242ac13000d",
    "vsphere_config": {
      "datacenter_name": "datacenter-1",
      "datastore_name": "datastore-1",
      ...
      "appliance_master_template_name": "test-master-template"
    }
  }
}
```

Update Orchestration Config

This API allows the IT Admin to update an orchestration config by appliance ID.

Request

- **Method:** PUT
- **Resource:** /api/v1/sa/{appliance_id}/orchestration
- **JSON Data:** A JSON dictionary representing OrchestrationConfigEntity entity

Response

- Status: 204
- Errors:
  - 404 If the appliance is not found.

Example

Request

```
PUT /api/v1/sa/1d89e147-a845-4825-93bf-154592454c25/orchestration HTTP/1.1
Accept: application/json
Host: api.pulseone.net

{
  "appliance_config": {
    "admin_username": "admin_new",
    "internal_fqdn": "internal-new.pulseworkspace.net"
  },
  "deployment_config": {
    "id": "28374bca-efef-11f5-8e5b-0242ac13000d",
    "vsphere_config": {
      "datacenter_name": "datacenter-2"
    }
  }
}
```

Response

```
HTTP/1.1 204 OK
```

Download Appliance Configuration

This API allows the IT admin to download the initial appliance configuration.

Request

- **Method:** GET
- **Resource:** /api/v1/sa/<appliance_id>/orchestration/init-config

Response

- Status: 200
- Text file: A text file which contains the data to be used during the appliance deployment procedure
- Errors:
  - 404 If the appliance is not found or if orchestration type is not found.

Example

Request

```
GET /api/v1/sa/1d89e147-a845-4825-93bf-154592454c25/orchestration/init-config HTTP/1.1
Host: customer.pulseworkspace.net
```

Response

```
http
HTTP/1.1 200 OK
Content-Type: text/plain;charset=UTF-8
Content-Disposition: attachment; filename=vsphere_user_data.txt
Content-Length: <length>

vaPrimaryDNS=None;vaExternalDefaultVlan=-1;vaExternalPortReconfigWithValueInVAppProperties=0;
vaExternalNetmask=None;vaWINSServer=localhost;vaAdminUsername=test_admin;
vaAdminPassword=test_password;vaManagementIPAddress=None;vaDNSDomain=None;
vaGateway=None;vaNetmask=None;vaAcceptLicenseAgreement=y;vaAdminEnableREST=y;
vaOrganization=None;vaExternalGateway=None;vaManagementDefaultVlan=-1;
vaConfigServerCACertPEM=
-----BEGIN CERTIFICATE-----
MIIDgjCCAuugAwIBAgIQdbhb/0UnTXGcjNvsRBoqBjANBgkqhkiG9w0BAQsFADCB
5zELMAkGA1UEBhMCV0MxCzAJBgNVBAgMAk9BMTIwMAYDVQQHDCnhiILli4zqtYvq
tLlOzrrqsZjqtJBmxYblibjjgZrlianqt4rRnuq1lTFbMFkGA1UECgxS6rOExI7E
K7roTs2U8brpksSU0a0pvgbcDpWAi0bamrYrgF2ckIx8osi4T1dC/7O1s1yAXztK
x1GuzKx3TOENyzleEeW+DRSWGJx4/fNuXrX1gV0vTQIDAQABoy0wKzAbBgNVHREE
FDASghBjcHFvbngucXJncC50ZXN0MAwGA1UdEwEB/wQCMAAwDQYJKoZIhvcNAQEL
BQADgYEAojDkMzL/VIawQS2nz7QYhLHNbq0PhcOeglJ+rOx0ac7UvkVZX/YTJ2qq
X6/L45pj0xkOzhF+S2AdAafL/6LZ0blnF3dc8cx9V9/vctXK9xNNS0JztqfoDXF6
tCDTh1YwfkSU/jEiXzxb/iac+iziC1PTwUYRhwHpH9FST9OY3TM=
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDhDCCAu2gAwIBAgIRALfENb+qVEd/uPNeJ3u/YVgwDQYJKoZIhvcNAQELBQAw
gecxCzAJBgNVBAYTAldDMQswCQYDVQQIDAJPQTEyMDAGA1UEBwwp4YiC5YuM6rWL
6rS5Ts666rGY6rSQZsWG5Ym444Ga5Ymp6reK0Z7qtZUxWzBZBgNVBAoMUuqzhMSO
xIjqtb/EqMW56rKF4YqH6rGyceqwluOCheOCk+q2l+OBmmvqs7Bz0ZDhiJbqtobj
jzMmJMz0O1pAqEMd4+2wNLAqevDlVSpOmM6pYF/GRFMBIJkxDKHgjrJueDiFv3ak
OBJnCQq1VykYZSTVT7p0DcTCLctfABcTNutxfnv+kDtkYmH9AgMBAAGjKjAoMBgG
A1UdEQQRMA+CDWdidS51bGRwLnRlc3QwDAYDVR0TAQH/BAIwADANBgkqhkiG9w0B
AQsFAAOBgQBYNfIkuyXEh5MRvHWW85XgtSlkxUsGZUev2DX8yLpnzRHG8oUGJ0Up
3lkyzUPhqmJfYUy8P5dRvNX1nPQxTLOJEILQmp3wOHjd8zu/MF7CyZfjvlsT3jdc
P8ufyMmSIyMbVURTyrUE/obw/pqX1HRD2FP9K5v6x6mQz5xtY/CUAw==
-----END CERTIFICATE-----;
vaRandomText=575fce705d9e6c92237438c7e3612c7d;vaDefaultVlan=-1;vaIPAddress=None;
vaInternalPortReconfigWithValueInVAppProperties=0;vaConfigURL=
'http://domain-1e1f5edf.unity.test/api/v1/sa/57540bd4-6f19-4f10-83af-
fc8017dc2c16/orchestration/initial-
config?t=159add10099134a46477eaebcf537af077de5089189fbe802bf4748398841520';
vaExternalIPAddress=None;vaManagementNetmask=None;vaManagementPortReconfigWithValueInVAppProperti
es=0;
vaCommonName=;vaSecondaryDNS=None;vaEnableLicenseServer=n;vaManagementGateway=None;
```

Unity API - Appliance Configuration

This is documentation for interactions between an appliance and Unity when exchanging configuration. JSON bodies expanded to be human-readable for convenience.
Read the API documentation for information about connecting and authenticating.

Configuration

After registration, an appliance will need to make this request in order to fetch its initial configuration.
Types
The configuration data structure is a list of dictionaries that indicate separate configuration payloads.
saml: A configuration payload of this type has no other properties and its presence indicates to the appliance that SAML should be configured, if the appliance supports SAML. For SAML configuration see SAML Configuration.

Request

- **Method:** GET
- **Resource:** /api/v1/sa/{device_id}/config

Example

Request

```
GET /api/v1/sa/b6484b93-edc1-4208-ab27-03d68f12534a/config HTTP/1.1
Host: foobar.api.pulseworkspace.net
Authorization: Hawk id="dh37fgj492je", ts="1353832234", nonce="j4h3g2",
hash="Yi9LfIIFRtBEPt74PVmbTF/xVAwPn7ub15ePICfgnuY=",mac="aSe1DERmZuRl3pI36/9BdZmnErTw3sNzOO
AUlfeKjVw="
```

Response

```
HTTP/1.1 200 OK
Server-Authorization: Hawk mac="Cs35A35Ik0mQZBO+Ux/Nh6h1BaKhp78pn99x1Cs3BYw=", [snip]...
Content-Type: application/json
Content-Length: 8934

{
    "id": "223178a2-0d13-4c4a-9860-6ceb0b031043",
    "result": {
        "config": [
            {
                "type": "saml"
            }
        ]
    }
}
```

Appliance CloudSecure Stats

This API documents the way in which appliance can provide CloudSecure stats to PulseOne.
Read the API documentation for information about connecting and authenticating.
See the Entities documentation for information about entities referred to in this document.
See the User Access History documentation for information about the user access history records. Records in CloudSecure stats will have reference to the corresponding user access history record.

Scheduling

For this version of the API, it is expected that an appliance will send CloudSecure stats at a pre-configured interval of every 2 minutes.

Adding CloudSecure Statistics

API to publish the CloudSecure endpoint stat details from appliance to Pulse One. To avoid appliance sending same endpoint stats data in every update, appliance needs to filter the endpoint stats based on session_update_time and send only the stats for endpoints that are updated from previous API call to current time.

Request

- Method: POST
- Resource: /api/v1/sa/{device-id}/stats/cloudsecure
- device-id: *Required.* This is the value provided to the appliance during registration
- Authorization: Appliance with ID {device-id}
- JSON Data:
- Request data should be in the form of a JSON dictionary with the following keys:
  - total: *Required.* Indicates number of entries available under 'items' Each entry in 'items' corresponds to details of one endpoint.
  - items: *Required.* (*List of CloudSecureEndpointStatsEntity*) If there is no data for endpoints for the given interval, None can be passed (empty JSON object for this key) A list of up to 100 CloudSecureEndpointStatsEntitys can be sent. In case appliance sends more than 100 entries, data will be rejected. If appliance has more than 100 endpoint details to be sent, it is expected that appliance will send the details in batches of 100. If any CloudSecureEndpointStatsEntity details are received without id and session_start_time keys, they will be ignored and logged.

Response

- Status: 204
- Errors:
  - 40001: Too Many Elements, see Errors for details.

Example

Request

```
POST /api/v1/sa/771b4124-f1b6-46ca-8035-172355924e02/stats/cloudsecure HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net

{
  "total":3,
  "items": [
    {
      "cloud_secure_record_id": "10153M4YK30P7Z0ID",
      "session_start_time": "2017-02-02T00:02:18Z",
      "session_update_time": "2017-02-02T00:02:48Z",
      "session_end_time": "2017-02-02T00:02:48Z",
      "auth_result": true,
      "os_type": 2,
      "os_version": 4,
      "compliance": "Compliant",
      "endpoint_model": 2,
      "endpoint_platform": 3,
      "user_access_record_id":"20153M4YK30P7Z0ID"
      "failed_applications": [1,2],
      "success_applications": [3,5,6],
      "roles": ["Engineering","Finance"],
    },
    {
      ...
    }
  ]
}
```

Response

```
HTTP/1.1 204 No Content
```

Updating Cloud Secure identifiers to display name mapping

Appliance stores identifiers for various attributes of endpoint details and keeps mappings for identifiers to display names. Appliance also provides option to administrators to change display names for any of these identifiers. In case administrator modifies display name for any of the identifier, it automatically gets updated for existing endpoint stats as well as it stores only the identifier and mapping from identifier to display name is updated.

Hence to support above flow in Pulse One console as well, appliance sends mapping between identifiers and display names.

Appliance will send the updated mappings of identifier to display names only for the updated/modified ones. For Ex, if there is a change in mapping for endpoint os versions, appliance will send only these mapping details.

While sending data to the console, conversion from identifiers to display names need to be done by the corresponding API and display strings need to be sent to console.

Request

- **Method:** POST
- **Resource:** /api/v1/sa/{device-id}/stats/cloudsecure/mappings
- device-id: *Required.* This is the value provided to the appliance during registration
- **JSON Data:** CloudSecureMappingsEntity

Response

- **Status:** 204 - Successful

Example

Request

```
POST /api/v1/sa/771b4124-f1b6-46ca-8035-172355924e02/stats/cloudsecure/mappings
HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net

{
  "os_types": [
    {"id" : 0, "value": "Unknown"},
    {"id" : 1, "value": "iOS"},
    {"id" : 2, "value": "Android"},
    {"id" : 3, "value": "Windows"},
    {"id" : 4, "value": "Mac"},
    {"id" : 5, "value": "Windows Phone"},
    {"id" : 6, "value": "Chrome"},
    {"id" : 7, "value": "Linux"}
  ],
```

```
  "os_versions": [
    {"id" : 1, "value": "iOS 8.1"},
    {"id" : 2, "value": "iOS 9.1"},
    {"id" : 3, "value": "Mac 10.11"},
    {"id" : 4, "value": "Mac 10.12"},
    {"id" : 5, "value": "Windows 8"},
    {"id" : 6, "value": "Windows 8.1"},
    {"id" : 7, "value": "Windows 10"},
  ],
  "endpoint_models": [
    {"id" : 1, "value": "iPhone 5S"},
    {"id" : 2, "value": "iPhone 6S"},
    {"id" : 3, "value": "iPhone 6S Plus"}
  ],
  "endpoint_platforms": [
    {"id" : 1, "value": "iPhone"},
    {"id" : 2, "value": "iPad"},
    {"id" : 3, "value": "iPad Mini"}
  ],
  "applications": [
    {"id" : 1, "value": "Aha"},
    {"id" : 2, "value": "Dropbox"},
    {"id" : 3, "value": "JIRA"},
    {"id" : 4, "value": "Google"},
    {"id" : 5, "value": "Microsoft"},
    {"id" : 6, "value": "Salesforce"},
    {"id" : 7, "value": "Box"},
    {"id" : 8, "value": "WebEx"},
    {"id" : 9, "value": "Zendesk"},
    {"id" : 10, "value": "Netsuite"}
  ]
}
```


Response

```
HTTP/1.1 204 No Content
```

Domains Info

Customer Domain Lookup
This endpoint provides a way for determining the correct customer URL that belongs to an email domain. An "email domain" is a way to associate a customer's domain with the domain portion of an email address. This allows a user to look up the customer's domain if all they have is an email address. This is particularly useful for figuring out where to send users to register their device for workspace auto-provisioning without having to know anything but their email address.

Request

- **Method:** GET
- **Resource:** /api/v1/info/domain/lookup?email_domain={email_domain}
- email_domain: The email domain to use for determining the customer domain. For example, if you have the email address "user@example.com" the email_domain is "example.com".

Example

```
GET /api/v1/info/domain/lookup?email_domain=example.com HTTP/1.1
Host: api.pulseworkspace.net
```

Response

If the email_domain matches a customer email domain then the domain's info is returned.
If the email_domain in the request does not match any customer email domains, an HTTP status of 404 is returned.

- **Status:** 200
- **JSON Data:**
  - name: The domain name for the customer
  - url: A URL with schema for the domain name

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: length

{
  "name": "example.pulseworkspace.net",
  "url": "https://example.pulseworkspace.net"
}
```

Appliance Endpoint Stats

This is documentation for the API endpoint for retrieving endpoint stats.
Read the API documentation for information about connecting and authenticating.
See the Entities documentation for information about entities referred to in this document.

Aggregated Endpoint Statistics

Request

- **Method:** GET
- **Resource:** /api/v1/sa/stats/endpoint?limit=<duration>&resolution=<sample_resolution>&[appliance_ids=<appliance_id(s)>]&[timestamp_end=<datetime>]

Returns a chronological list of endpoint stats based on the aggregation details.

Request Parameters

- timestamp_end: (*str*) End time of duration window. (Default is NOW)
- limit: (*int*) Limit the range in days for the requested resolution . If an appliance is new or because of data retention policies there may not be at least limit a day's worth of data. In this case, as many records as available will be returned. Example: A request for hourly resolution with a limit of 10 days when only 7 days of data is available will result in only 7 days of records.
- resolution: (*str*) Desired resolution of the results. Values:
  - minute: Minute resolution for 1 day.
  - hour: Hour resolution for 7 days.
  - day: Daily resolution for 30 days.
  - week: Weekly resolution for 90 days.
  - month: Monthly resolution for 365 days.
- appliance_ids: (*str*) Comma separated list of appliance ids. If appliance IDs are not provided, then data will be returned for the entire domain.

Response

- **Status:** 200
- **JSON:** An ApplianceStatsAggregationEntity
  Note: user_roles will not be returned. We do not currently support an aggregation of this data.
  - timestamp_start: (*str*) Start time of duration window. All returned samples set will be more recent than this time.
  - timestamp_end: (*str*) End time of duration window. No sample will be more recent than this time.
  - limit: Value returned from the request.
  - resolution: Value returned from the request.
  - latest_stats: (EndpointStatsAggregationEntity) Date histogram aggregation over all endpoint stats in the duration window.
  - stats: (list of EndpointStatsAggregationEntity) A list of all EndpointStatsAggregationEntity values in the duration window.

Example

Request

```
GET /api/v1/sa/stats/endpoint?limit=7&resolution=hour&appliance_ids=bef5ad-ee31-4bd7-b2da-4d0b029f8945
```

Response


```
{
  "timestamp_start": "2015-03-06T13:34:52Z",
  "timestamp_end": "2015-03-13T13:34:52Z",
  "resolution": "hour",
  "limit": 7,
  "latest_stats": {
    "hc_failure_reason": {
      "hc_failure_39": 0,
      "hc_failure_38": 0,
      "hc_failure_37": 0,
.
.
      "hc_failure_27": 0,
      "hc_failure_40": 0,
      "hc_failure_41": 0
    },
    "auth_result": {
      "failed": 2,
      "success": 4
    },
    "compliance": {
      "failed": 0,
      "compliant": 0,
      "not_assessed": 6,
      "remediated": 0
    },
    "auth_mechanism": {
      "eap": 0,
      "l3": 4,
      "other": 0,
      "mac": 0
    },
    "device_os": {
      "windows_8": 0,
      "windows_7": 0,
      "unknown": 0,
      "linux": 0,
      "ios": 0,
      "windows_xp": 0,
      "windows_vista": 0,
      "mac_os": 4,
      "blackberry": 0,
      "others": 0,
      "android": 0
    }
    "compliance": {
      "failed": 0,
      "compliant": 0,
      "not_assessed": 6,
      "remediated": 0
    },
    "timestamp_start": "2015-03-13T12:34:52Z",
    "timestamp_end": "2015-03-13T13:34:52Z"
  }
  "stats": [
```

```
  {
    "hc_failure_reason": {
      "hc_failure_39": 0,
      "hc_failure_38": 0,
      "hc_failure_37": 0,
.
.
      "hc_failure_27": 0,
      "hc_failure_40": 0,
      "hc_failure_41": 0
    },
    "auth_result": {
      "failed": 2,
      "success": 4
    },
    "compliance": {
      "failed": 0,
      "compliant": 0,
      "not_assessed": 6,
      "remediated": 0
    },
    "timestamp_end": "2015-03-13T13:00:00Z",
    "auth_mechanism": {
      "eap": 0,
      "l3": 4,
      "other": 0,
      "mac": 0
    },
    "device_os": {
      "windows_8": 0,
      "windows_7": 0,
      "unknown": 0,
      "linux": 0,
      "ios": 0,
      "windows_xp": 0,
      "windows_vista": 0,
      "mac_os": 4,
      "blackberry": 0,
      "others": 0,
      "android": 0
    }
  },
  {
    ...
  },
 ]
}
```

Appliance Endpoint Stats

This API documents the way in which an appliance can provide endpoint statistics to Unity.
Read the API documentation for information about connecting and authenticating.
See the Entities documentation for information about entities referred to in this document.
Scheduling
For this version of the API it is expected that an appliance will send endpoint statistics at a preconfigured interval of every 2 minutes.

Adding Endpoint Statistics

Request

- **Method:** POST
- **Resource:** /api/v1/sa/{device-id}/stats/endpoint
- device-id: *Required.* This is the value provided to the SA during registration
- **JSON Data**: Request data should be in the form of a JSON dictionary with the following keys:
- stats: *Required.* A list of up to 100 ApplianceEndpointStatsEntity

All top level keys for ApplianceEndpointStatsEntity are optional except the timestamp_end and timestamp_start key. If timestamp_start and timestamp_end is provided without any other keys, that entity is ignored. All missing keys are considered 'unavailable' for the time period provided by timestamp_end and timestamp_start.

Response

- **Status:** 204
- **Errors:**
- 40001: Too Many Elements, see Errors for details.

Example

Request

```
POST /api/v1/sa/771b4124-f1b6-46ca-8035-172355924e02/stats/endpoint HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: foobar.api.pulseworkspace.net

{
  "stats": [
    {
      "hc_failure_reason": {
        "hc_failure_39": 0,
        "hc_failure_38": 0,
        "hc_failure_37": 0,
.
.
.
        "hc_failure_27": 0,
        "hc_failure_40": 0,
        "hc_failure_41": 0
```

```
    },
    "auth_result": {
      "failed": 2,
      "success": 4
    },
    "compliance": {
      "failed": 0,
      "compliant": 0,
      "not_assessed": 6,
      "remediated": 0
    },
    "timestamp_start": "2015-05-31T21:23:34Z",
    "auth_mechanism": {
      "eap": 0,
      "l3": 4,
      "other": 0,
      "mac": 0
    },
    "user_roles": {
      "Users": 4
    },
    "device_os": {
      "windows_8": 0,
      "windows_7": 0,
      "unknown": 0,
      "linux": 0,
      "ios": 0,
      "windows_xp": 0,
      "windows_vista": 0,
      "mac_os": 4,
      "blackberry": 0,
      "others": 0,
      "android": 0
    },
    "timestamp_end": "2015-05-31T21:25:34Z"
  }
 ]
}
```


Response

```
HTTP/1.1 204 No Content
```

Managed Appliance Firmware Management API

This API allows Pulse One to manage firmwares of managed appliances.

Note: *Managing ESAP packages on Pulse One also uses below API (similar to managing PCS and PPS firmwares on Pulse One). In all the below REST APIs 'firmware' can also mean 'ESAP' package.*

These API would be used in the following scenarios:

- Admin uploading a managed appliance firmware via Pulse One console
- A registered appliance downloading a firmware when Pulse One instructs it to upgrade to a specific version
- Console wants to list and manage all the managed appliance firmwares stored in Pulse One.

Pulse One server represents any firmware of a managed appliance in two parts:

- Metadata that describes the firmware
- Content which is actual binary content of the firmware

POST /api/v1/sa/firmwares

This API can be used by console to upload a managed appliance's firmware metadata and content, by making a multipart/form-data request.
MD5 hash (content_hash) field of the content would be entered by admin while uploading firmware package via console UI (i.e., it is not calculated by browser).

Note: content_hash needs to be calculated only for content being uploaded, which should not include metadata of the firmware package.

Request

- **Method:** POST
- **Authorization:** Requires admin.appliances.firmwares|WRITE
- **Resource:** /api/v1/sa/firmwares
- **Body:** Two parts (multipart), metadata and content as descibed below:

metadata: A JSON dictionary representing a ApplianceFirmwareMetadataUpdateEntity entity.
content: Binary contents of appliance firmware package.

Response
Success

- **Status:** -- 200
- **Body:** -- JSON dictionary containing ID of the newly created firmware (firmware_id), which is same as content_hash field of input ApplianceFirmwareMetadataUpdateEntity entity.

Error

- **Status:** -- 403 (Access forbidden)
- **Status:** -- 409 (Conflict, if content is already available for the given version and type in metadata (or) with same firmware_id/MD5 hash)

Example

Request

```
POST /api/v1/sa/firmwares HTTP/1.1
Content-Length: 56802
Content-Type: multipart/form-data; boundary=723e82b4e41f4d0d9c66db24f035a326
Host: api.pulseone.net

--723e82b4e41f4d0d9c66db24f035a326
Content-Disposition: multipart/form-data; name="metadata"

{
  "description": "Downloaded from support portal site",
  "version": "9.0R1-HF2",
  "type": "VPN",
  "content_hash": "088623438e5e50d53b7cfe8d50943b90ea1989a8"
}
--723e82b4e41f4d0d9c66db24f035a326
Content-Disposition: multipart/form-data; name="content"; filename="90R1-PCS.pkg"

<<binary-content of file 90R1-PCS.pkg>>
```

Response

```
HTTP/1.1 200 OK
{
  "firmware_id": "088623438e5e50d53b7cfe8d50943b90ea1989a8"
}
```

PUT /api/v1/sa/firmwares/{firmware_id}

This API can be used by console to modify a firmware's metadata whose firmware content has already been uploaded to Pulse One.

Request

- **Method:** PUT
- **Authorization:** Requires admin.appliances.firmwares|WRITE
- **Resource:** /api/v1/sa/firmwares/{firmware_id}
- **JSON Data:** A JSON dictionary representing an ApplianceFirmwareMetadataUpdateEntity entity.

Response Success

- **Status:** -- 204

Response Error

- **Status:** -- 409 (Conflict, if there is another firmware with new version and type combination already exists (other than this firmware_id)
- **Status:** -- 404 (Not Found, if incorrect firmware_id)

Example

Request

```
PUT /api/v1/sa/firmwares/28374bca-efef-11f5-8e5b-0242ac13000d HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net

{
  "description": "Downloaded from support portal site",
  "version": "9.0R1-HF2",
  "type": "VPN"
}
```

Response

```
HTTP/1.1 204 OK
```

HEAD /api/v1/sa/firmwares/{firmware_id}

Check to see if a firmware with the given firmware_id exists in Pulse One.
If the firmware exists, expect a 200 response, otherwise expect a 404.

Request

- **Method:** HEAD
- **Authorization:** Requires admin.appliances.firmware|READ
- **Resource:** /api/v1/sa/firmwares/{firmware_id}

Response

- **Status:** -- 200
- **Status:** -- 404 (If firmware with given firmware_id does not exist)

Example

```
HTTP/1.1 200 OK
```

GET /api/v1/sa/firmwares/{firmware_id}

API to get metadata of a firmware with the given firmware_id
If the firmware_id does not exist, expect a 404.

Request

- **Method:** GET
- **Authorization:** Requires admin.appliances.firmware|READ
- **Resource:** /api/v1/sa/firmwares/{firmware_id}

Response

- **Status:** -- 200
- **JSON Data:** A JSON dictionary representing a ApplianceFirmwareMetadataEntity entity.

Example

Request

```
GET /api/v1/sa/firmwares/28374bca-efef-11f5-8e5b-0242ac13000d HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 80

{
   "firmware_id": "088623438e5e50d53b7cfe8d50943b90ea1989a8",
   "description": "Downloaded from support portal site",
   "version": "9.0R1-HF2",
   "type": "VPN"
   "created": "2017-09-04T18:55:38",
   "content_size": 368263
}
```

DELETE /api/v1/sa/firmwares/{firmware_id}

Pulse One Console can use this API to delete a firmware (both metadata and content).

Request

- **Method:** DELETE
- **Authorization:** admin.appliances.firmware|DELETE
- **Resource:** /api/v1/sa/firmwares/{firmware_id}

Response

- **Status:** -- 204 (Successful)
- **Status:** -- 404 (Not Found)

GET /api/v1/sa/firmwares

Pulse One Console can use this API to get details of all managed appliance firmwares.

Request

- **Method:** GET
- **Authorization:** Requires admin.appliances.firmware|READ
- **Resource:** /api/v1/sa/firmwares

Response

- **Status:** -- 200
- **JSON Data:** A JSON dictionary representing a list of ApplianceFirmwareMetadataCollectionEntity entities.

Example

Request

```
GET /api/v1/sa/firmwares HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 80

{
  "items": [
    {
      "firmware_id": "088623438e5e50d53b7cfe8d50943b90ea1989a8",
      "description": "Downloaded from support portal site",
      "version": "9.0R1-HF2",
      "type": "VPN"
      "created": "2017-09-04T18:55:38",
      "content_size": 368263
    },
    {
      "firmware_id": "188623438e5e50d53b7cfe8d50943b90ea1989a8",
      "description": "Downloaded from support portal site",
      "version": "9.0R1-HF2",
      "type": "VPN"
      "created": "2017-09-04T18:55:38",
      "content_size": 368263
    }
  ],
  "total": 2
}
```

GET /api/v1/sa/firmwares/{firmware_id}/content

Using this API, one can get content of firmware package

Request

- **Method:** GET
- **Resource:** /api/v1/sa/firmwares/{firmware_id}/content
- **Authorization:** Requires admin.appliances.firmware|READ

Response

- **Content-Type:** application/octet-stream
- **Body:** The binary contents of the firmware
- **Status:** -- 200 (Successful)
- **Status:** -- 404 (Not Found, if incorrect firmware_id)

HEAD /api/v1/sa/firmwares/{firmware_id}/content

If the firmware exists, expect a 200 response with at least Content-Length of the firmware metadata (in bytes).
If the firmware does not exist, expect a 404.

Request

- **Method:** HEAD
- **Authorization:** Requires admin.appliances.firmware|READ
- **Resource:** /api/v1/sa/firmwares/{firmware_id}/content

Response

- **Status:** -- 200
- **Status:** -- 404 (If firmware with given firmware_id does not exist)

Example

```
HTTP/1.1 200 OK
Content-Length: 1230384
```

Console API - Appliance Health Stats

This is documentation for the API endpoint for retrieving health stats for an appliance.
See the Entities documentation for information about entities referred to in this document.

Request

- **Method:** GET
- **Resource:** /api/v1/sa/{sa_device_key}/health-stats

Returns the most recent health stats for a given appliance.

Response

- **Status:** 200
- **JSON Data:**
  - id: A UUID unique to the response.
  - result: A SaHealthStatsEntity entity.

There will be no content if the appliance does not have health data.

Example

Request

```
GET /api/v1/sa/f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b/health-stats HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Returns the latest health stats for a given appliance.

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 149

{
   "id": "223178a2-0d13-4c4a-9860-6ceb0b031043",
   "result": {
      "cpu_utilization": 13.45,
      "memory_utilization": 34.23,
      "disk_utilization": 78.56,
      "concurrent_users": 429,
      "network_throughput": 3902,
      "authentication_failures": 90,
      "timestamp": "2015-03-13T13:34:52Z"
   }
}
```

Request

- **Method:** GET
- **Resource:** /api/v1/sa/stats/health?limit=<duration>&resolution=<sample_resolution>&[appliance_ids=<appliance_id(s)>]&[timestamp_end=<datetime>]

Returns a chronological list of health stats based on the aggregation details.

Request Parameters

- timestamp_end: (*str*) Endtime of duration window. (Default is NOW)
- limit - (*int*) Limit the number of records for the requested resolution. If an appliance is new or because of data retention policies there may not be at least limit records. In this case, as many records as available will be returned. Example: A request for hourly resolution with a duration of 10 days when only 7 days of data is available will result in only 7 days of records.
- resolution - (*str*) Desired resolution of the results. Values:
  - minute: Minute resolution for 1 day.
  - hour: Hour resolution for 7 days.
  - day: Daily resolution for 30 days.
  - week: Weekly resolution for 90 days.
  - month: Monthly resolution for 365 days.
- appliance_ids: (*str*) Comma separated list of appliance ids.

Response

- **Status:** 200
- **JSON:** An ApplianceStatsAggregationEntity

Example


```
{
  "timestamp_start": "2015-03-06T13:34:52Z",
  "timestamp_end": "2015-03-13T13:34:52Z",
  "resolution": "hour",
  "limit": 7,
  "latest_stats": {
    "cpu_utilization": 15.00,
    "memory_utilization": 30.00,
    "disk_utilization": 78.01,
    "concurrent_users": 539,
    "network_throughput": 9898,
    "timestamp": "2015-03-13T13:00:00Z"
  },
  "stats": [
    {
      "cpu_utilization": 13.45,
      "memory_utilization": 34.23,
      "disk_utilization": 78.56,
      "concurrent_users": 429,
      "network_throughput": 3902,
      "timestamp": "2015-03-13T13:00:00Z"
    },
    {
```

```
    "cpu_utilization": 46.2,
    "memory_utilization": 68.02,
    "disk_utilization": 13.6,
    "concurrent_users": 15038,
    "network_throughput": 76533456345,
    "timestamp": "2015-03-13T12:00:00Z"
   },
   {
    ...
   }
  ]
}
```

Appliance Information

Every appliance has several facts (information) that change occasionally, but not often enough that they're useful for time-series.
This API documents the way in which an appliance can provide such facts (information) to Pulse One.
Read the API documentation for information about connecting and authenticating.
See the Entities documentation for information about entities referred to in this document.

Scheduling

Once in every two minutes appliance will periodically send facts to Pulse One.
Example appliance facts:

- Concurrent User licenses
- Named User licenses
- System Version
- Uptime (last reboot time)
- IF-MAP Federation
- License Member
- Build
- IP
- Cluster Info

See the Entities documentation for information about the JSON keys of the above facts.

Sending facts

Request

- **Method:** PUT
- **Resource:** /api/v1/sa/{device-id}/info
- device-id: *Required.* This is the value provided to the SA during registration
- **JSON Data:** Request data should be in the form of a JSON dictionary with the following keys:
  - All keys for ApplianceInfoEntity are optional.
  - Any missing key is considered 'not changed' from the previously sent value.

Response

- **Status:** 204
- **Errors:** See Errors for details.

Example

Request

```
PUT /api/v1/sa/771b4124-f1b6-46ca-8035-172355924e02/info HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseworkspace.net

{
  "appliance_version": "8.1R3.0-19864",
  "boot_timestamp": "2015-03-12T20:36:43.06Z",
  "concurrent_user_licenses": 5000,
  "concurrent_user_license_breakdown": {
    "access_licenses": 2000,
    "consec_licenses": 2000,
    "polsec_licenses": 1000
  }
  "if_map": "client",
  "license_role": null,
  "license_server": false,
  "cluster": {
    "id": "2348-323-453",
    "name": "MyCluster1.hr.myorg.com",
    "type": "active-passive",
    "node_name": "MyNodeNewName1",
    "leader_node": true,
    "active_node": false
    }
}
```

Response

```
HTTP/1.1 204 No Content
```

Orchestration API

Fetch Initial XML Appliance Config API

This API allows the PCS to fetch the XML appliance config.
This API uses temporary token to do the authorization, because the appliance doesn't have HAWK info when it tries to fetch the initial XML configurations.

Request

- **Method:** GET
- **Resource:** /api/v1/sa/<appliance_id>/orchestration/initial-config
- **Parameters:**
  - t: (*str*) The id of the temporary token.

Response

- **Status:** 200
  - **Content-Type:** application/xml
- **Errors:**
  - 403: If the temporary token is invalid.
  - 404: If the appliance is not found.

Example

Request

```
GET /api/v1/sa/31374bca-efef-11f5-8esd-0242as7h60d/orchestration/initial-
config?t=735o488ebcdd8fb0c4623a6awqsce39adkei9 HTTP/1.1
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: <length>

<xml>
...
</xml>
```

Console API - Profiler Data

This is documentation for the API endpoint for retrieving profiled endpoints of an appliance.
See the Entities documentation for information about entities referred to in this document.

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/profiler/endpoints?[appliance_ids=<appliance_ids>]&[search=<str>]&[collector_type=<collector_type>]&[start=<int>]&[limit=<int>]

Returns brief details of all the endpoints that are profiled by all registered appliances which has pushed device data to Pulse One. If the appliance ids are provided, then it returns only endpoints that are profiled by given appliances.

Request Parameters

- appliance_ids: (*str*) Comma separated list of appliance ids. (Default: None)
- search: (*str*) Free text search to filter the results. (Default: None)
- collector_type: (*str*) Filters the result based on collector type. Possible values:
  - dhcp
  - snmp
  - nmap
  - ssh
  - wmi
- start: (*int*) Indicates the number of initial results that should be skipped. (Default: 0)
- limit: (*int*) Indicates the number of results that should be returned. (Default: 10)

Response

- **Status:** 200
- **JSON Data:**
  - Returns ProfilerEndpointsEntity
  - 'total' field indicates number of available endpoints and can be used for pagination
  - Data is sorted based on last_seen field (most recently profiled ones come first)
  - Empty JSON array, if there are no profiled endpoints

Example

Request

```
GET /api/v1/sa/profiler/endpoints HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Returns all profiled endpoints from registered appliances.

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 18337

{
   "total": 321,
   "data": [{
      "appliance_id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
      "macaddr": "00:50:56:bf:6d:30",
      "first_seen": "2016-11-30T09:40:39Z",
      "last_seen": "2016-11-29T07:06:35Z",
      "os": "Microsoft Windows Kernel 6.x",
      "category": "Windows",
      "previous_category": "",
      "previous_os": "",
      "manufacturer": "VMware, Inc.",
      "ip": "10.204.90.74",
      "hostname": "admin2-PC",
      "notes": "RED: Test note.",
      "sid": "sidd797b442d0a83eb580a782f62a77e962ef403a4e7b407752"
   }, {
      "appliance_id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
      "macaddr": "00:50:56:86:4b:e6",
      "first_seen": "2016-12-08T03:47:23Z",
      "last_seen": "2016-12-15T06:41:50Z",
      "os": "Microsoft Windows Kernel 6.x",
      "category": "Windows",
      "previous_category": "",
      "previous_os": ""
      "manufacturer": "VMware, Inc.",
      "ip": "10.204.90.42",
      "hostname": "admin2-PC"
   }, {
      ...
   }]
}
```

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/profiler/endpoints/{mac_address}

Returns the given endpoint profile data.

Response

- **Status:** 200
- **JSON Data:**
  - Returns ProfilerEndpointEntity.
- **Errors:**
  - 404: If the given device not found.

Example

Request

```
GET /api/v1/sa/profiler/endpoints/00:50:56:bf:5c:98 HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Returns the given endpoint profile data.

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 12969

{
   "appliance_id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
   "category": "Windows",
   "first_seen": "2016-11-30T09:40:39Z",
   "history": [
      {
         "appliance_id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
         "ip_address": [
            {
               "collector_type": "session",
               "detected_time": "2016-11-30T09:40:42Z",
               "hostname": "",
               "ip": ""
            },
            {
               "collector_type": "session",
               "detected_time": "2016-12-02T09:18:05Z",
               "hostname": "",
               "ip": "172.21.17.51"
            }
         ],
         "profile": [
            {
               "category": "",
               "collector_type": "session",
               "detected_time": "2016-11-30T09:40:42Z",
               "os": ""
            }
         ],
         "session_details": [
```

```
            {
                "collector_type": "session",
                "detected_time": "2016-11-30T09:40:42Z",
                "session.login_host": "",
                "session.session_type": "",
                "session.state": ""
            },
            {
                "collector_type": "session",
                "detected_time": "2016-11-30T09:41:52Z",
                "session.login_host": "10.204.58.49",
                "session.session_type": "pps",
                "session.state": "active"
            },
            {
                "collector_type": "session",
                "detected_time": "2016-11-30T09:46:23Z",
                "session.login_host": "10.204.58.49",
                "session.session_type": "pps",
                "session.state": "inactive"
            },
            {
                "collector_type": "session",
                "detected_time": "2016-11-30T09:53:38Z",
                "session.login_host": "10.204.58.49",
                "session.session_type": "pps",
                "session.state": "active"
            },
            {
                "collector_type": "session",
                "detected_time": "2016-12-02T09:18:05Z",
                "session.login_host": "10.204.58.49",
                "session.session_type": "pps",
                "session.state": "inactive"
            }
        ]
    },
    {
        "appliance_id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
        "profile": [
            {
                "category": "Windows",
                "time_stamp": 1496108933.592008,
                "detected_time": "2017-05-30T01:48:53Z",
                "os": "Windows",
                "collector_type": "dhcp"
            }
        ],
        "ip_address": [
            {
                "ip": "10.204.49.174",
                "hostname": "PAVAN-VM-PC",
                "detected_time": "2017-05-30T01:48:53Z",
                "collector_type": "dhcp",
                "time_stamp": 1496108933.591265
            }
```

```
        ]
    },
    {
        "appliance_id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
        "profile": [
            {
                "category": "Windows",
                "time_stamp": 1496108933.592008,
                "detected_time": "2017-05-30T01:48:53Z",
                "os": "Windows",
                "collector_type": "dhcp"
            }
        ],
        "ip_address": [
            {
                "ip": "10.204.49.174",
                "hostname": "PAVAN-VM-PC",
                "detected_time": "2017-05-30T01:48:53Z",
                "collector_type": "dhcp",
                "time_stamp": 1496108933.591265
            }
        ]
    }
],
"hostname": "test-pc.psecure.net",
"ip": "172.21.17.39",
"last_seen": "2016-12-02T09:18:48Z",
"macaddr": "6c:88:14:e2:3b:3c",
"manufacturer": "Intel Corporate",
"manufacturer_id": 14843,
"nmap": {
    "classified_category": "Windows",
    "classified_os": "Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7",
    "distance": 4,
    "hostname": "test-pc.psecure.net",
    "ip": "172.21.17.39",
    "mac_addr": "6c:88:14:e2:3b:3c",
    "os_fingerprinted": true,
    "os_matches": [
        {
            "classes": [
                {
                    "accuracy": 95,
                    "cpe_list": [
                        "cpe:/o:microsoft:windows_vista::-",
                        "cpe:/o:microsoft:windows_vista::sp1"
                    ],
                    "osfamily": "Windows",
                    "osgen": "Vista",
                    "type": "general purpose",
                    "vendor": "Microsoft"
                },
                {
                    "accuracy": 95,
                    "cpe_list": [
                        "cpe:/o:microsoft:windows_server_2008::sp1"
```

```
                ],
                "osfamily": "Windows",
                "osgen": "2008",
                "type": "general purpose",
                "vendor": "Microsoft"
            },
            {
                "accuracy": 95,
                "cpe_list": [
                    "cpe:/o:microsoft:windows_7"
                ],
                "osfamily": "Windows",
                "osgen": "7",
                "type": "general purpose",
                "vendor": "Microsoft"
            }
        ],
        "line": 71668,
        "name": "Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7"
    },
    {
        "classes": [
            {
                "accuracy": 95,
                "cpe_list": [
                    "cpe:/o:microsoft:windows_7"
                ],
                "osfamily": "Windows",
                "osgen": "7",
                "type": "general purpose",
                "vendor": "Microsoft"
            },
            {
                "accuracy": 95,
                "cpe_list": [
                    "cpe:/o:microsoft:windows_server_2008:r2"
                ],
                "osfamily": "Windows",
                "osgen": "2008",
                "type": "general purpose",
                "vendor": "Microsoft"
            }
        ],
        "line": 69441,
        "name": "Microsoft Windows 7 SP1 or Windows Server 2008 R2"
    },
    {
        "classes": [
            {
                "accuracy": 94,
                "cpe_list": [
                    "cpe:/o:microsoft:windows_server_2008:r2:sp1"
                ],
                "osfamily": "Windows",
                "osgen": "2008",
                "type": "general purpose",
```

```
                "vendor": "Microsoft"
            },
            {
                "accuracy": 94,
                "cpe_list": [
                    "cpe:/o:microsoft:windows_8"
                ],
                "osfamily": "Windows",
                "osgen": "8",
                "type": "general purpose",
                "vendor": "Microsoft"
            }
        ],
        "line": 67469,
        "name": "Microsoft Windows Server 2008 R2 SP1 or Windows 8"
    },
    {
        "classes": [
            {
                "accuracy": 94,
                "cpe_list": [
                    "cpe:/o:microsoft:windows_server_2008:r2"
                ],
                "osfamily": "Windows",
                "osgen": "2008",
                "type": "general purpose",
                "vendor": "Microsoft"
            }
        ],
        "line": 67031,
        "name": "Microsoft Windows Server 2008 R2"
    },
    {
        "classes": [
            {
                "accuracy": 94,
                "cpe_list": [
                    "cpe:/o:microsoft:windows_7::sp1"
                ],
                "osfamily": "Windows",
                "osgen": "7",
                "type": "general purpose",
                "vendor": "Microsoft"
            },
            {
                "accuracy": 94,
                "cpe_list": [
                    "cpe:/o:microsoft:windows_server_2008::sp2",
                    "cpe:/o:microsoft:windows_server_2008:r2:sp1"
                ],
                "osfamily": "Windows",
                "osgen": "2008",
                "type": "general purpose",
                "vendor": "Microsoft"
            }
        ],
```

```
            "line": 69508,
            "name": "Microsoft Windows 7 SP1 or Windows Server 2008 SP2 or 2008 R2 SP1"
        },
        {
            "classes": [
                {
                    "accuracy": 94,
                    "cpe_list": [
                        "cpe:/o:microsoft:windows_vista::sp2"
                    ],
                    "osfamily": "Windows",
                    "osgen": "Vista",
                    "type": "general purpose",
                    "vendor": "Microsoft"
                }
            ],
            "line": 71897,
            "name": "Microsoft Windows Vista SP2"
        },
        {
            "classes": [
                {
                    "accuracy": 93,
                    "cpe_list": [
                        "cpe:/o:microsoft:windows_vista::sp2"
                    ],
                    "osfamily": "Windows",
                    "osgen": "Vista",
                    "type": "general purpose",
                    "vendor": "Microsoft"
                },
                {
                    "accuracy": 93,
                    "cpe_list": [
                        "cpe:/o:microsoft:windows_7::sp1"
                    ],
                    "osfamily": "Windows",
                    "osgen": "7",
                    "type": "general purpose",
                    "vendor": "Microsoft"
                },
                {
                    "accuracy": 93,
                    "cpe_list": [
                        "cpe:/o:microsoft:windows_server_2008"
                    ],
                    "osfamily": "Windows",
                    "osgen": "2008",
                    "type": "general purpose",
                    "vendor": "Microsoft"
                }
            ],
            "line": 71940,
            "name": "Microsoft Windows Vista SP2, Windows 7 SP1, or Windows Server 2008"
        },
        {
```

```
            "classes": [
                {
                    "accuracy": 93,
                    "cpe_list": [
                        "cpe:/o:microsoft:windows_7"
                    ],
                    "osfamily": "Windows",
                    "osgen": "7",
                    "type": "general purpose",
                    "vendor": "Microsoft"
                }
            ],
            "line": 68358,
            "name": "Microsoft Windows 7"
        },
        {
            "classes": [
                {
                    "accuracy": 93,
                    "cpe_list": [
                        "cpe:/o:microsoft:windows_7::-:professional"
                    ],
                    "osfamily": "Windows",
                    "osgen": "7",
                    "type": "general purpose",
                    "vendor": "Microsoft"
                },
                {
                    "accuracy": 93,
                    "cpe_list": [
                        "cpe:/o:microsoft:windows_8"
                    ],
                    "osfamily": "Windows",
                    "osgen": "8",
                    "type": "general purpose",
                    "vendor": "Microsoft"
                }
            ],
            "line": 68837,
            "name": "Microsoft Windows 7 Professional or Windows 8"
        },
        {
            "classes": [
                {
                    "accuracy": 92,
                    "cpe_list": [
                        "cpe:/o:microsoft:windows_vista::-",
                        "cpe:/o:microsoft:windows_vista::sp1"
                    ],
                    "osfamily": "Windows",
                    "osgen": "Vista",
                    "type": "general purpose",
                    "vendor": "Microsoft"
                }
            ],
            "line": 71588,
```

```
            "name": "Microsoft Windows Vista SP0 - SP1"
        }
    ],
    "ports": {
        "tcp": {
            "closed": [
                21,
                22,
                23,
                25,
                53,
                80,
                110,
                143,
                443,
                3306,
                3389,
                8080
            ],
            "filtered": [
                88,
                135,
                8085,
                8086
            ],
            "open": [
                139,
                445
            ]
        },
        "udp": {
            "closed": [
                53,
                123,
                161
            ],
            "filtered": [
                67,
                68,
                135,
                138,
                139,
                445,
                500,
                520,
                631,
                1434,
                1900
            ],
            "open": [
                137
            ]
        }
    },
    "snmp_sysdescr": "",
    "status": "up",
```

```
    "timestamp": 1480670326.5092349,
    "type": "nmap"
},
"os": "Windows 7",
"ports": {
    "tcp": {
        "closed": [
            21,
            22,
            23,
            25,
            53,
            80,
            110,
            143,
            443,
            3306,
            3389,
            8080
        ],
        "filtered": [
            88,
            135,
            8085,
            8086
        ],
        "open": [
            139,
            445
        ]
    },
    "udp": {
        "closed": [
            53,
            123,
            161
        ],
        "filtered": [
            67,
            68,
            135,
            138,
            139,
            445,
            500,
            520,
            631,
            1434,
            1900
        ],
        "open": [
            137
        ]
    }
},
"previous_category": "",
```

```
"previous_os": "",
"session": {
    "MDM": {
        "Manufacturer": "Intel Corporate",
        "manufacturer": "Intel Corporate"
    },
    "login_host": "10.204.58.49",
    "login_host_addr": "10.204.58.49",
    "mac_addr": "6c:88:14:e2:3b:3c",
    "session_type": "pps",
    "sid": "sidd797b442d0a83eb580a782f62a77e962ef403a4e7b407752",
    "source_ip": "172.21.17.39",
    "state": "active",
    "switch_ip": "",
    "type": "session",
    "user_agent": "Pulse-Secure/8.2.5.689 (Windows 7) Pulse/5.2.5.689"
},
"sid": "sidd797b442d0a83eb580a782f62a77e962ef403a4e7b407752",
"user_agent": {
    "classified_category": "Windows",
    "classified_os": "Windows 7",
    "type": "user-agent",
    "user_agent": "Pulse-Secure/8.2.5.689 (Windows 7) Pulse/5.2.5.689"
},
"wmi": {
    "category": "Windows",
    "mac_addr": "6c:88:14:e2:3b:3c",
    "status": "down",
    "timestamp": 1480499213.8365581,
    "type": "wmi"
},
"snmp": {
    "con_time": 1478618593.161855,
    "discon_time": 1478618934.161752,
    "ifindex": "516",
    "mac_addr": "005056bf5c98",
    "port": "ge-0/0/15.0",
    "snmp_version": 2,
    "switch_ip": "10.204.89.197",
    "switch_name": "JuniperSwitch",
    "switch_support": "6",
    "switch_vendor": "JUNIPER",
    "timestamp": 1478618934.161757,
    "trap": {
        "discon_time": 1478618934.161752,
        "ifindex": "516",
        "mac_addr": "005056bf5c98",
        "port": "ge-0/0/15.0",
        "snmp_version": 2,
        "switch_ip": "10.204.89.197",
        "switch_name": "JuniperSwitch",
        "switch_support": "6",
        "switch_vendor": "JUNIPER",
        "timestamp": 1478618934.161757,
        "trap_type": "mac_removed",
        "type": "trap",
```

```
            "valid_port": true,
            "vendor": "JUNIPER",
            "vlan": "5"
        },
        "poll": {
            "ifindex": "516",
            "mac_addr": "005056bf5c98",
            "port": "ge-0/0/15.0",
            "snmp_version": 2,
            "switch_ip": "10.204.89.197",
            "switch_name": "JuniperSwitch",
            "switch_support": "6",
            "switch_vendor": "JUNIPER",
            "timestamp": 1478618934.161757,
            "type": "snmp",
            "valid_port": true,
            "vendor": "JUNIPER",
            "vlan": "5"
        },
        "trap_type": "mac_removed",
        "type": "trap",
        "valid_port": true,
        "vendor": "JUNIPER",
        "vlan": "5"
    }
}
```


Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/profiler/stats/devices?[appliance_ids=<appliance_ids>]&[timestamp_start=<datetime>]&[timestamp_end=<datetime>]

Returns aggregated endpoints data based on Operating System of the profiled device.

Request Parameters

- appliance_ids: (*str*) Comma separated list of appliance ids. (Default: None)
- timestamp_start: (*str*) Start time of the duration.
- timestamp_end: (*str*) End time of the duration.

Response

- **Status:** 200
- **JSON Data:**
  - JSON Object with OS as key and count as value.
  - If no profiled endpoints are available, then return empty JSON Object

Example

Request

```
GET /api/v1/sa/profiler/stats/devices HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Returns aggregated endpoints data based on Operating System of the profiled device.

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length:

{
   "Others": 1,
   "Ruckus Wireless AP": 1,
   "Microsoft Windows Kernel 6.x": 39,
   "Windows": 6,
   "Mac OS X": 2,
   "Apple iPod, iPhone or iPad": 1,
   "Microsoft Windows Kernel 6.0": 4,
   "Generic Linux": 2,
   "Ubuntu": 1,
   "Generic Android": 1,
   "PXE": 9
}
```

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/profiler/stats/manufacturers?[appliance_ids=<appliance_ids>]&[timestamp_start=<datetime>]&[timestamp_end=<datetime>]

Returns aggregated count of endpoints based on the manufacturer of the profiled endpoints.

Request Parameters

- appliance_ids: (*str*) Comma separated list of appliance ids. (Default: None)
- timestamp_start: (*str*) Start time of the duration.
- timestamp_end: (*str*) End time of the duration.

Response

- **Status:** 200
- **JSON Data:**
  - JSON Object with Manufacturer as key and corresponding device count as value.
  - If no profiled endpoints are available, then returns empty JSON object.

Example

Request

```
GET /api/v1/sa/profiler/starts/manufacturers HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Returns aggregated endpoints count based on the manufacturer of the profiled endpoints.

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 417

{
   "Ruckus Wireless": 1,
   "Apple, Inc.": 3,
   "VMware, Inc.": 42,
   "Qumranet Inc.": 4,
   "Xiaomi Communications Co Ltd": 1,
   "Hon Hai Precision Ind. Co.,Ltd.": 1,
   "Armorlink shanghai Co. Ltd": 2,
   "Universal Global Scientific Industrial Co., Ltd": 1,
   "Super Micro Computer, Inc.": 5,
   "IBM Corp": 2,
   "LG Electronics": 1,
   "Intel Corporate": 3,
   "Flextronics International": 1
}
```

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/profiler/stats/categories?[appliance_ids=<appliance_ids>]&[timestamp_start=<datetime>]&[timestamp_end=<datetime>]

Returns aggregated endpoints count based on the category of the profiled endpoints.

Request Parameters

- appliance_ids: (*str*) Comma separated list of appliance ids. (Default: None)
- timestamp_start: (*str*) Start timestamp of the duration.
- timestamp_end: (*str*) End timestamp of the duration.

Response

- **Status:** 200
- **JSON Data:**
  - JSON Object with Category as key and corresponding endpoints count as value.
  - If no profiled endpoints are availableavailable, then returns empty JSON Object.

Example

Request

```
GET /api/v1/sa/profiler/stats/categories HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Returns aggregated endpoints count based on the category of the profiled endpoints.

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 164

{
   "Other": 1,
   "Network Boot Agents": 9,
   "Windows": 49,
   "Smartphones/PDAs/Tablets": 2,
   "Routers and APs": 1,
   "Linux": 3,
   "Macintosh": 2
}
```

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/profiler/stats/states?[appliance_ids=<appliance_ids>]&[timestamp_start=<datetime>]&[timestamp_end=<datetime>]

Returns aggregated endpoints count based on the state of the discovered endpoints.

Request Parameters

- appliance_ids: (*str*) Comma separated list of appliance ids. (Default: None)
- timestamp_start: (*str*) Start time of the duration.
- timestamp_end: (*str*) End time of the duration.

Response

- **Status:** 200
- **JSON Data:**
  - Returns ProfilerStateStatsEntity

Example

Request

```
GET /api/v1/sa/profiler/stats/states HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Returns aggregated endpoints count based on the state of the discovered endpoints.

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 45

{
   "profiled": 66,
   "not_profiled": 1
}
```

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/profiler/stats/managed-states?[appliance_ids=<appliance_ids>]&[timestamp_start=<datetime>]&[timestamp_end=<datetime>]

Returns aggregated endpoint count based on the managed state of the profiled endpoints.

Request Parameters

- appliance_ids: (*str*) Comma separated list of appliance ids. (Default: None)
- timestamp_start: (*str*) Start time of the duration.
- timestamp_end: (*str*) End time of the duration.

Response

- **Status:** 200
- **JSON Data:**
  - JSON Object with Managed State (Managed/ Unmanaged) as key and corresponding devices count as value.
  - If no discovered devices are available, then returns empty JSON Object.

Example

Request

```
GET /api/v1/sa/profiler/stats/managed-states
Accept: application/json
Host: customer.pulseworkspace.net
```

Returns aggregated endpoint count based on the managed state of the profiled endpoint.

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 41

{
   "managed": 66,
   "unmanaged": 1
}
```

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/profiler/stats/profile-changes?[appliance_ids=<appliance_ids>]&[timestamp_start=<datetime]&[timestamp_end=<datetime>]

Returns number of endpoints that are changed its profile in given time period.

Request Parameters

- appliance_ids: (*str*) Comma separated list of appliance ids. (Default: None)
- timestamp_start: (*str*) Start time of the duration.
- timestamp_end: (*str*) End time of the duration.

Response

- **Status:** 200
- **JSON Data:**
- Returns ProfilerProfileChangeStatsEntity

Example

Request

```
GET /api/v1/sa/profiler/stats/profile-changes HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Returns number of endpoints that are changed its endpoint profile.

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 41

{
   "others": 139,
   "profile_changed": 2
}
```

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/profiler/stats/sessions?[appliance_ids=<appliance_ids>]

Returns number of endpoints have active session with all appliances or given appliances.

Request Parameters

- appliance_ids: (*str*) Comma separated list of appliance ids. (Default: None)

Response

- **Status:** 200
- **JSON Data:**
  - Returns ProfilerSessionStatsEntity

Example

Request

```
GET /api/v1/sa/profiler/stats/sessions HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Returns number of profiled endpoints that are having active session with profiler.

Request

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 41

{
   "sessions": 31,
   "no_session": 40
}
```

Profiler Data

This API documents the way in which an appliance can provide profiled endpoints data to Pulse One.
See the Entities documentation for information about entities referred to in this document.

Adding Profiled Endpoints

The REST API endpoint to add or update the profiled endpoints of an appliance. It accepts gzip compressed JSON array with multiple profiled endpoints. Each endpoint is identified using its MAC address. If the same endpoint comes from two appliances, then later one replaces exiting one, but with few exceptions stated below:

1. Last seen is updated from whichever is seen most recently.
2. Fist seen is update from whichever is seen first.
3. History is merged and maintained per appliance under history attribute.

Request

- **Method:** POST
- **Authorization:** Appliance with ID {appliance_id}
- **Resource:** /api/v1/sa/{appliance_id}/profiler/endpoints
- appliance_id: *Required.* This is the value provided to the SA during registration
- **JSON Data:** Request data should be in form of a JSON array with 2

Response

- **Status:** 204

Example

Request

In below example, JSON array represented, but in actual request this needs to be compressed with gzip.

```
POST /api/v1/sa/771b4124-f1b6-46ca-8035-172355924e02/profiler/endpoints HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: foobar.api.pulseworkspace.net

[{
   "category": "Smartphones/PDAs/Tablets",
   "dhcp": {
      "classified_category": "Smartphones/PDAs/Tablets",
      "classified_os": "Apple iPod, iPhone or iPad",
      "combination_id": 615084,
      "hostname": "darumugiPhone6s",
      "mac_addr": "44:00:10:26:ec:f2",
      "message_type": 3,
      "parameter_request_list": "1,121,3,6,15,119,252",
      "requested_ip": "172.21.16.86",
      "src_mac": "0026886ece01",
      "timestamp": 1480479200.8340991,
```

```
    "type": "dhcp"
},
"first_seen": "2016-11-24T07:37:41Z",
"history": {
    "ip_address": [{
        "collector_type": "dhcp",
        "detected_time": "2016-11-24T07:37:41Z",
        "hostname": "",
        "ip": ""
    }, {
        "collector_type": "dhcp",
        "detected_time": "2016-11-24T07:37:41Z",
        "hostname": "",
        "ip": ""
    }, {
        "collector_type": "dhcp",
        "detected_time": "2016-11-25T03:35:46Z",
        "hostname": "darumugiPhone6s",
        "ip": "172.21.16.89"
    }, {
        "collector_type": "dhcp",
        "detected_time": "2016-11-28T03:11:21Z",
        "hostname": "darumugiPhone6s",
        "ip": "172.21.16.124"
    }, {
        "collector_type": "dhcp",
        "detected_time": "2016-11-29T02:46:32Z",
        "hostname": "darumugiPhone6s",
        "ip": "172.21.16.93"
    }, {
        "collector_type": "dhcp",
        "detected_time": "2016-11-30T03:01:51Z",
        "hostname": "darumugiPhone6s",
        "ip": "172.21.16.75"
    }],
    "profile": [{
        "category": "",
        "collector_type": "dhcp",
        "detected_time": "2016-11-24T07:37:41Z",
        "os": ""
    }, {
        "category": "",
        "collector_type": "dhcp",
        "detected_time": "2016-11-24T07:37:41Z",
        "os": ""
    }]
},
"hostname": "darumugiPhone6s",
"ip": "172.21.16.86",
"last_seen": "2016-11-30T04:13:20Z",
"macaddr": "44:00:10:26:ec:f2",
"manufacturer": "Apple, Inc.",
"manufacturer_id": 18453,
"os": "Apple iPod, iPhone or iPad",
"previous_category": "",
"previous_os": ""
```

```
}, {
    "category": "Windows",
    "dhcp": {
        "classified_category": "Windows",
        "classified_os": "Microsoft Windows Kernel 6.x",
        "combination_id": 5557,
        "hostname": "ananthm-PC",
        "mac_addr": "10:0b:a9:b7:cc:d4",
        "message_type": 8,
        "parameter_request_list": "1,15,3,6,44,46,47,31,33,121,249,43,252",
        "requested_ip": "172.21.16.149",
        "src_mac": "0026886ece01",
        "timestamp": 1480479200.6655741,
        "type": "dhcp",
        "vendor_class": "MSFT 5.0"
    },
    "first_seen": "2016-11-24T07:30:24Z",
    "history": {
        "ip_address": [{
            "collector_type": "dhcp",
            "detected_time": "2016-11-24T07:30:27Z",
            "hostname": "",
            "ip": ""
        }, {
            "collector_type": "dhcp",
            "detected_time": "2016-11-24T07:30:27Z",
            "hostname": "",
            "ip": ""
        }, {
            "collector_type": "dhcp",
            "detected_time": "2016-11-28T04:25:53Z",
            "hostname": "ananthm-PC",
            "ip": "172.21.16.100"
        }, {
            "collector_type": "dhcp",
            "detected_time": "2016-11-28T04:25:54Z",
            "hostname": "ananthm-PC",
            "ip": "192.168.1.7"
        }, {
            "collector_type": "dhcp",
            "detected_time": "2016-11-29T04:28:51Z",
            "hostname": "ananthm-PC",
            "ip": "172.21.16.165"
        }, {
            "collector_type": "dhcp",
            "detected_time": "2016-11-29T04:28:52Z",
            "hostname": "ananthm-PC",
            "ip": "192.168.1.7"
        }, {
            "collector_type": "dhcp",
            "detected_time": "2016-11-30T03:58:22Z",
            "hostname": "ananthm-PC",
            "ip": "172.21.16.185"
        }, {
            "collector_type": "dhcp",
            "detected_time": "2016-11-30T03:58:23Z",
```

```
            "hostname": "ananthm-PC",
            "ip": "192.168.1.7"
        }],
        "profile": [{
            "category": "",
            "collector_type": "dhcp",
            "detected_time": "2016-11-24T07:30:27Z",
            "os": ""
        }, {
            "category": "",
            "collector_type": "dhcp",
            "detected_time": "2016-11-24T07:30:27Z",
            "os": ""
        }]
    },
    "hostname": "ananthm-PC",
    "ip": "172.21.16.149",
    "last_seen": "2016-11-30T04:13:17Z",
    "macaddr": "10:0b:a9:b7:cc:d4",
    "manufacturer": "Intel Corporate",
    "manufacturer_id": 13088,
    "os": "Microsoft Windows Kernel 6.x",
    "previous_category": "",
    "previous_os": ""
}]
```


Response

```
HTTP/1.1 204 No Content
```

Cloud API - Appliance Registration

This is documentation for interactions between a VPN or NAC appliance and Cloud during the appliance registration workflow. JSON bodies expanded to be human-readable for convenience.
Read the API documentation for information about connecting and authenticating.
See the Entities documentation for information about entities referred to in this document.

A Note On Request Hosts

Registration requests must be made to a specific domain. This domain value is configured in the appliance by the appliance admin during initial setup. The domain to use is provided to the appliance admin in the PWS Cloud console during setup of the appliance in the PWS Cloud console.

Example

reg.pulseworkspace.net

Registration

After initial admin configuration, the appliance will make this request to Cloud which will facilitate the registration process. PWS will provide the api URL that future requests should be made to by the appliance.
Note: The initial registration request is unique in that it does not require HAWK authentication.

Request

- **Method:** POST
- **Resource:** /api/v1/sa/register
- **JSON Data:** Request data should be in the form of a JSON body representing ApplianceRegistrationInfoEntity. All the fields in the entity are required to be filled in.

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following keys:
  - credentials: The credentials to use for HAWK authentication for future requests. This data structure is described in more detail in the Auth documentation.
  - device_id: A unique identifier for the appliance making the request. The appliance should remember this value and include it in future requests.
  - notifications_url: The websocket URL that the appliance should subscribe to in order to receive notifications.
  - api_url: The URL to use for API requests for this device.
  - client_certificate: Client certificate of the PZT gateway.
  - server_certificate: Server certificate of the PZT gateway.
- **Errors:**
  - 40304: Registration Code Invalid, see Errors for details.

Example

Request

```
POST /api/v1/sa/register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: foobar.api.pulseworkspace.net

{
   "registration_code": "JZTAet96L"
   "type": "VPN"
   "model": "MAG4610"
   "serial_number": "0153M0TS00BII04U"
   "appliance_version": "8.1R4.1-32789"
   "client_certificate_csr":
      "-----BEGIN CERTIFICATE REQUEST-----\n
      ...
      oNML1gqUppaiYZSQMMDvVSAmeyIPdYo9TDcJbgkfBnwlqGgAg3MtJ8tJ39Fx9BDY\n
      nOncm6gQaNJ+Hw==\n
      -----END CERTIFICATE REQUEST-----\n",
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 189

{
   "id": "4a26a3b1-fe7d-438f-b328-d6d37dbd04b2",
   "result": {
      "credentials": {
        "type": "hawk",
        "algorithm": "HS256",
        "id": "{alphanumeric string}",
        "secret": "{alphanumeric string}"
      },
      "device_id": "b6484b93-edc1-4208-ab27-03d68f12534a",
      "notifications_url": "wss://foobar-api.pulseworkspace.net/api/v1/notifications",
      "api_url": "https://foobar-api.pulseworkspace.net",
      "client_certificate":
         "-----BEGIN CERTIFICATE-----\n
         ...
         oNML1gqUppaiYZSQMMDvVSAmeyIPdYo9TDcJbgkfBnwlqGgAg3MtJ8tJ39Fx9BDY\n
         nOncm6gQaNJ+Hw==\n
         -----END CERTIFICATE-----\n",
   }
}
```

Scheduled Tasks API

This API allows Pulse One console to manage scheduled tasks. Currently only the following type of tasks can be *scheduled* by Pulse One console:

- system.operations.appliance.firmware.stage: Stage a firmware package
- system.operations.appliance.firmware.install: Install a staged firmware package

Creation of new schedule task
POST /api/v1/sa/tasks/schedules

This API can be used by console to create a new scheduled task.

Request

- **Method:** POST
- **Authorization:** admin.appliances.tasks|CREATE
- **Resource:** /api/v1/sa/tasks/schedules
- **JSON Data:** A JSON dictionary representing a ScheduledTaskUpdateEntity entity.

Response
Success

- **Status:** -- 200
- **Body:** -- A JSON dictionary representing a ScheduledTaskEntity entity.

Example

Request

```
POST /api/v1/sa/tasks/schedules HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
{
   "appliance_id": "28374bca-efef-11f5-8e5b-0242ac13000d",
   "comments": "Scheduled upgrade during september holidays",
   "once": "2018-09-04T18:55:38",
   "task": {
      "type": "system.operations.appliance.firmware.stage",
      "params": {
         "firmware_id": "088623438e5e50d53b7cfe8d50943b90ea1989a8"
      }
   }
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 80

{
   "id": "38374bca-efef-11f5-8e5b-0242ac13000a",
   "enabled": True,
   "appliance_id": "28374bca-efef-11f5-8e5b-0242ac13000d",
   "comments": "Scheduled upgrade during september holidays",
   "once": "2018-09-04T18:55:38",
   "task": {
      "type": "system.operations.appliance.firmware.stage",
      "params": {
         "firmware_id": "088623438e5e50d53b7cfe8d50943b90ea1989a8"
      }
   }
}
```

PUT /api/v1/sa/tasks/schedules/{scheduled_task_id}

This API can be used by console to modify an existing schedule task identified by id.

Request

- **Method:** PUT
- **Authorization:** Requires admin.appliances.tasks|WRITE
- **Resource:** /api/v1/sa/tasks/schedules/{scheduled_task_id}
- **JSON Data:** A JSON dictionary representing a ScheduledTaskUpdateEntity entity.

Response Success

- **Status:** -- 204

Response Error

- **Status:** -- 409 (Conflict, if this scheduled task is already in-progress or started)
- **Status:** -- 404 (Not Found, if incorrect id)

Example

Request

```
PUT /api/v1/sa/tasks/schedules/28374bca-efef-11f5-8e5b-0242ac13000d HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net

{
   "group_id": "28374bca-efef-11f5-8e5b-0242ac13000d",
   "comments": "Scheduled upgrade during september holidays",
   "once": "2018-09-04T18:55:38",
   "task": {
      "type": "system.operations.appliance.firmware.stage",
      "params": {
         "firmware_id": "088623438e5e50d53b7cfe8d50943b90ea1989a8"
      }
   }
}
```

Response

```
HTTP/1.1 204 OK
```

GET /api/v1/sa/tasks/schedules/{scheduled_task_id}

API to get an existing scheduled task with the given id

Request

- **Method:** GET
- **Authorization:** Requires admin.appliances.tasks|READ
- **Resource:** /api/v1/sa/tasks/schedules/{scheduled_task_id}

Response Success

- **Status:** -- 200
- **JSON Data:** A JSON dictionary representing a ScheduledTaskEntity entity.

Response Error

- **Status:** -- 404 (Not Found, if incorrect id)

Example

Request

```
GET /api/v1/sa/tasks/schedules/28374bca-efef-11f5-8e5b-0242ac13000d HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 80

{
   "id": "28374bca-efef-11f5-8e5b-0242ac13000d",
   "enabled": True,
   "cluster_id": "48374bca-efef-11f5-8e5b-0242ac13000f",
   "comments": "Scheduled upgrade during september holidays",
   "once": "2018-09-04T18:55:38",
   "task": {
      "type": "system.operations.appliance.firmware.stage",
      "params": {
         "firmware_id": "088623438e5e50d53b7cfe8d50943b90ea1989a8"
      }
   }
}
```

DELETE /api/v1/sa/tasks/schedules/{scheduled_task_id}

Pulse One Console can use this API to delete an existing scheduled task.

Request

- **Method:** DELETE
- **Authorization:** admin.appliances.tasks|DELETE
- **Resource:** /api/v1/sa/tasks/schedules/{scheduled_task_id}

Response

- **Status:** -- 204 (Successful)

GET /api/v1/sa/tasks/schedules

Pulse One Console can use this API to get details of all scheduled tasks.

Request

- **Method:** GET
- **Authorization:** Requires admin.appliances.tasks|READ
- **Resource:** /api/v1/sa/tasks/schedules

Response

- **Status:** -- 200
- **JSON Data:** A JSON dictionary representing a list of ApplianceFirmwareMetadataCollectionEntity entities.

Example

Request

```
GET /api/v1/sa/tasks/schedules HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 80

{
  "items": [
    {
        "id": "28374bca-efef-11f5-8e5b-0242ac13000d",
        "enabled": True,
        "appliance_id": "48374bca-efef-11f5-8e5b-0242ac13000f",
        "comments": "Scheduled upgrade during september holidays",
        "once": "2018-09-04T18:55:38",
        "task": {
            "type": "system.operations.appliance.firmware.stage",
            "params": {
                "firmware_id": "088623438e5e50d53b7cfe8d50943b90ea1989a8"
            }
        }
    },
```

```
  {
    "id": "38374bca-efef-11f5-8e5b-0242ac13000f",
    "enabled": "False",
    "group_id": "48374bca-efef-11f5-8e5b-0242ac13000f",
    "comments": "Scheduled upgrade during december holidays",
    "once": "2018-10-04T18:55:38",
    "task": {
      "type": "system.operations.appliance.firmware.stage",
      "params": {
        "firmware_id": "088623438e5e50d53b7cfe8d50943b90ea1989a8"
      }
    }
    "last_task": {
      "status": "partial_success",
      "completed": "2018-12-04T18:55:38"
    }
  }
 ],
 "total": 2
}
```

SDP API

Get the URL of the SDP controller login page

This API will return the URL of the SDP controller login page.

Request

- **Method:** GET
- **Resource:** /api/v1/sa/sdp/controller/login

Response

- **Status:** 200
- **Errors:**
  - 404 If there isn't a SDP controller.

Example

Request

```
GET /api/v1/sa/sdp/controller/login HTTP/1.1
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: length

{
  "controller_login_url": "https://controller.example.com"
}
```

Console API - Appliance Stat Thresholds

This is documentation for the API endpoint for retrieving a list of appliances that match the specified criteria for comparison.
Read the API documentation for information about connecting and authenticating.
See the Entities documentation for information about entities referred to in this document.

Request

- **Method:** GET
- **Resource:** /api/v1/sa/stats/threshold?type=<type>&metric=<metric>&value=<value>&operation=<operation>

Request Parameters

- type: (*str*) The type of stat to test.
- health: Health related stats, see entity for metrics.
- endpoint: Endpoint related stats, see entity for metrics.
- metric: (*str*) The metric which we want to test. For example:
  - cpu_utilization
  - compliance.failed
- value: (*int*) Value to compare against.
- operation: (*str*) Desired operation for metric comparison. Values:
  - lt: Metric value is less than the value specified.
  - lte: Metric value is less than or equal to the value specified.
  - gt: Metric value is greater than the value specified.
  - gte: Metric value is greater than or equal to the value specified.
  - eq: Metric value is equal to the value specified.

Response

- **Status:** 200
- **JSON:** dictionary containing the following:
  - items: (list of _ApplianceStatsThresholdEntity_s)

Example

Request

```
GET /api/v1/sa/stats/threshold?type=health&metric=cpu_utilization&value=30&operation=gt
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "items": [
    {
      "appliance": {
        "id": "771b4124-f1b6-46ca-8035-172355924e02",
        "name": "appliance1"
      },
      "value": 60
    },
    {
      "appliance": {
        "id": "41771b24-bf16-382d-0385-72355192402e",
        "name": "appliance5"
      },
      "value": 50
    },
    {
      "appliance": {
        "id": "b4127714-b6f1-82da-0358-1235597024e2",
        "name": "appliance10"
      },
      "value": 99
      "timestamp": "2015-03-12T20:36:43.06Z"
    }
  ]
}
```

Console API - Domain Appliance Info

This is documentation for the API endpoint for retrieving appliance information for an entire domain.
See the Entities documentation for information about entities referred to in this document.

Request

- **Method:** GET
- **Resource:** /api/v1/sa/stats/info

Returns appliance information for an entire domain.

Response

- **Status:** 200
- **JSON Data:** An ApplianceStatsInfoEntity

Example

Request

```
GET /api/v1/sa/stats/info HTTP/1.1
Accept: application/json
Host: api.pulseworkspace.net
```

Returns the overall appliance information for a customer's domain.

Response

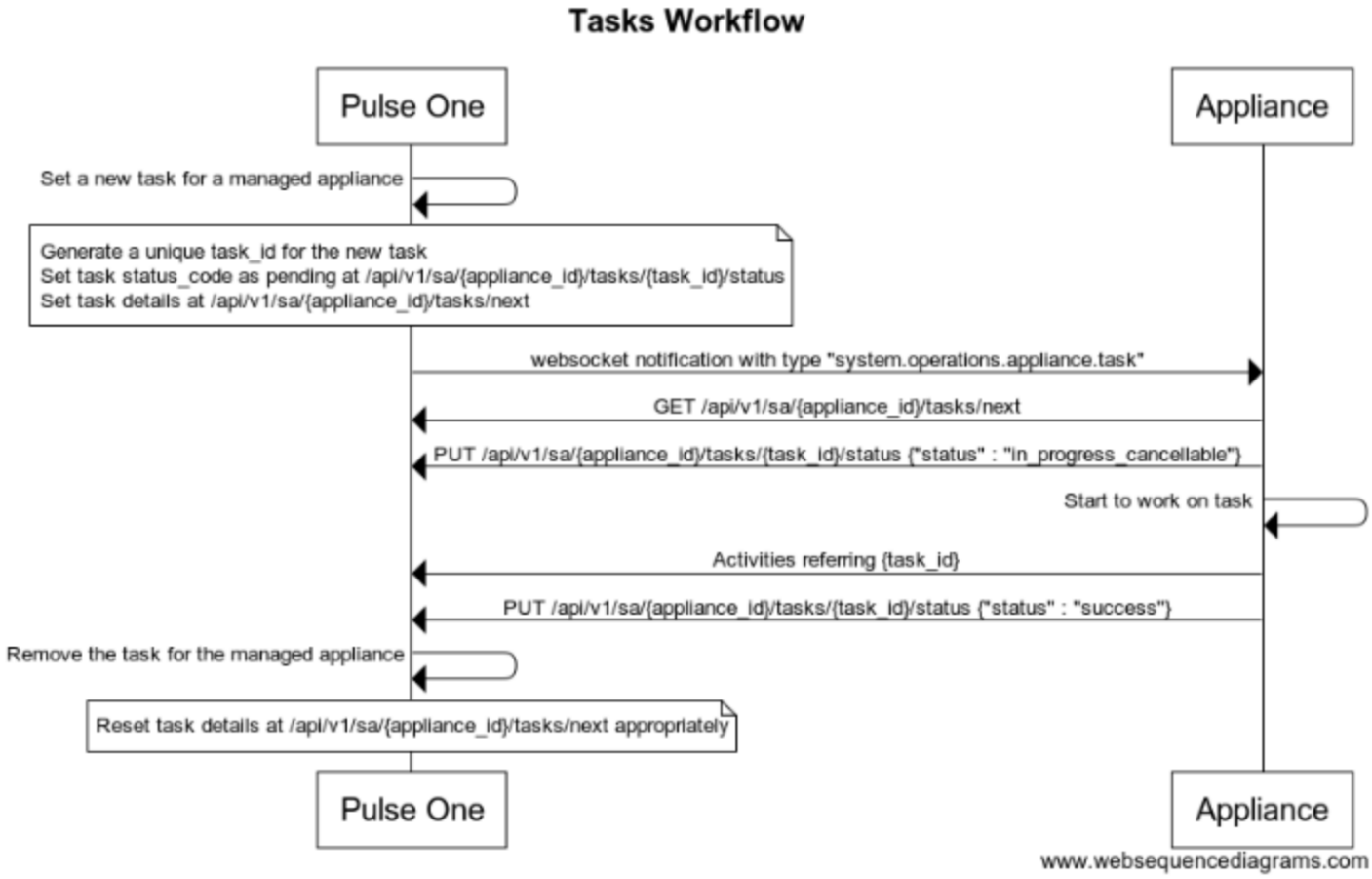
```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "clusters": 2,
  "appliances": [
    {
      "type": "pcs",
      "count": 100,
      "concurrent_users": 3,
      "concurrent_user_licenses": 100,
      "concurrent_user_license_breakdown": {
        "access_licenses": 30,
        "consec_licenses": 70
      },
      "named_user_licenses": {
        "installed": 3000,
        "consumed": 200,
        "leased_in": 400,
        "leased_out": 400,
      },
      "license_mode": {
```

```
      "concurrent_users": 30,
      "named_users": 70,
    },
    "license_role": {
      "client": 40,
      "server": 20,
      "standalone": 40,
    },
    "appliance_versions": {
      "6.1R1-28393": 70,
      "8.3R3-39382": 30,
    },
  },
  {
    "type": "pps",
    "count": 25,
    "concurrent_users": 5,
    "concurrent_user_licenses": 150,
    "concurrent_user_license_breakdown": {
       "access_licenses": 30,
       "polsec_licenses": 120
    },
    "appliance_versions": {
      "5.0R2.3-15112": 5,
      "5.4R3-23223": 20,
    }
  }
 ]
}
```

Appliance Tasks API

This API allows Pulse One console to command a task on a managed appliance. Some examples of the tasks performed by a managed appliance could be:

- Take a configuration backup and upload to Pulse One. Task type: system.operations.appliance.config.backup
- Restore configuration from a specific configuration backup stored in Pulse One. Task type: system.operations.appliance.config.restore
- Upgrade firmware of a managed appliance Task type: system.operations.appliance.firmware.upgrade
- Stage a firmware package Task type: system.operations.appliance.firmware.stage
- Install a staged firmware package Task type: system.operations.appliance.firmware.install
- Import XML Task type: system.operations.appliance.config.xml_import



Tasks Workflow

```
Pulse One->Pulse One: Set a new task for a managed appliance
   note over Pulse One
      Generate a unique task_id for the new task
      Set task status_code as pending at /api/v1/sa/tasks/{task_id}/status
      Set task details at /api/v1/sa/{appliance_id}/tasks/next
   end note
Pulse One->Appliance: websocket notification with type "system.operations.appliance.task"
Appliance->Pulse One: GET /api/v1/sa/{appliance_id}/tasks/next
Appliance->Pulse One: PUT /api/v1/sa/tasks/{task_id}/status {"status" : "in_progress_cancellable"}

Appliance->Appliance: Start to work on task
Appliance->Pulse One: Activities referring {task_id}
Appliance->Pulse One: PUT /api/v1/sa/tasks/{task_id}/status {"status" : "success"}
Pulse One->Pulse One: Remove the task for the managed appliance
   note over Pulse One
      Reset task details at /api/v1/sa/{appliance_id}/tasks/next appropriately
   end note
```

Notifications

Pulse One can instruct an appliance to work on a task using the following notification:

```
{
  "id": "{alphanumeric string}",
  "type": "system.operations.appliance.task",
}
```

- id: Unique notification ID
- type: "system.operations.appliance.task" to indicate that the appliance should pick up the pending task

The unique notification ID is different from the unique task ID described in the section below.

Tasks

A task describes the details of what Pulse One instructs the appliance to do.

```
{
  "id": "{alphanumeric string}",
  "type": "{task type}",
  "params": {
    {Optional key/value pairs of further details on the task.
     Each task type may define relevant key/value pairs}
  }
}
```

Current appliance task

The appliance can get the details on its current task specified by Pulse One

Request

- **Method:** GET
- **Authorization:** Requires the appliance that is assigned to the task.
- **Resource:** /api/v1/sa/{appliance_id}/tasks/next

Response

- **Status:** 200
- **JSON Data:** A task as described in the above section
- **No pending tasks for the appliance:** 204

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 345

{
  "id": "fbb2a630-42fb-48c8-ab8f-0f72c9746e08",
  "type": "system.operations.appliance.config.backup",
  "params": {
    "upload_url": "https://foobar-api.pulseworkspace.net/config/fbb2a630-42fb-48c8-ab8f-0f72c9746e08"
  }
}
```

Task status update

The appliance can update the status of the task using the following API.

Request

- **Method:** PUT
- **Authorization:** Requires the appliance that is assigned to the task.
- **Resource:** /api/v1/sa/tasks/{task_id}/status
- **JSON Data:** {"status" : {status_code}} where status_code can have one of the following values:
  - pending The appliance has not started working on the task yet.
  - in_progress_cancellable The appliance is working on the task. The appliance can cancel the task at this point in time.
  - in_progress_not_cancellable The appliance is working on the task. The appliance cannot cancel the task at this point in time.
  - failed The appliance finished working on the task. The task execution resulted in one or more errors.
  - success The appliance finished working on the task successfully.
  - cancelling The appliance has started working on cancelling the task.
  - cancelled The appliance has cancelled the task successfully.

Example

```
PUT /api/v1/sa/tasks/fbb2a630-42fb-48c8-ab8f-0f72c9746e08/status HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net

{
    "status": "in_progress_cancellable"
}
```

Task status activities

Detailed status logs related to a task apart from the status_code mentioned in the above section can be updated by the appliance using activities.
For example, the appliance can indicate its progress periodically with relevant activities, though the status_code remains as "in_progress_cancellable" or "in_progress_not_cancellable".
Activities related to a task execution in the appliance will include a reference to the notification which triggered the task:

- reference.type: "system.operations.appliance.task:{task type}"
- reference.id: ID of the task

Example

```
PUT /api/v1/sa/{device_id}/activities/da39a3ee5e6b4b0d3255bfef95601890afd80709 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net

{
  "activity_type": "appliance_task",
  "message": "Started to take the configuration backup",
  "reference": {
    "id": "fbb2a630-42fb-48c8-ab8f-0f72c9746e08",
    "type": "system.operations.appliance.task:system.operations.appliance.config.backup"
  },
  "severity": "informational",
  "time": "2017-03-23T15:30:00Z"
}
```

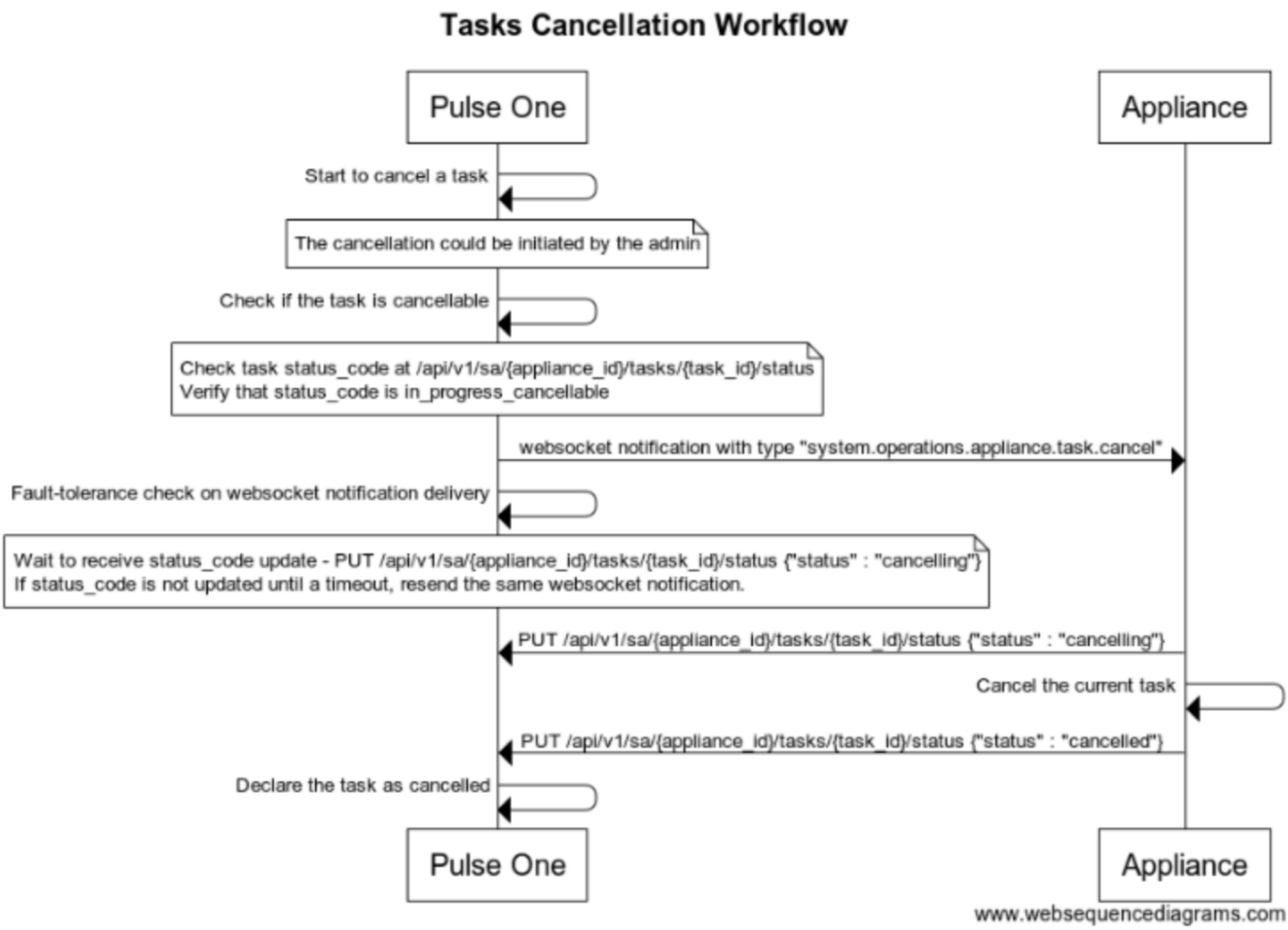
Cancel a task

Pulse One can instruct an appliance to cancel a task using the following notification:

```
{
  "id": "{alphanumeric string}",
  "type": "system.operations.appliance.task.cancel",
  "params": {
    "task_id": "{alphanumeric string}"
  }
}
```

- id: Unique notification ID
- type: "system.operations.appliance.task.cancel" to indicate that the appliance should cancel a task in progress
- params.task_id: Unique task ID

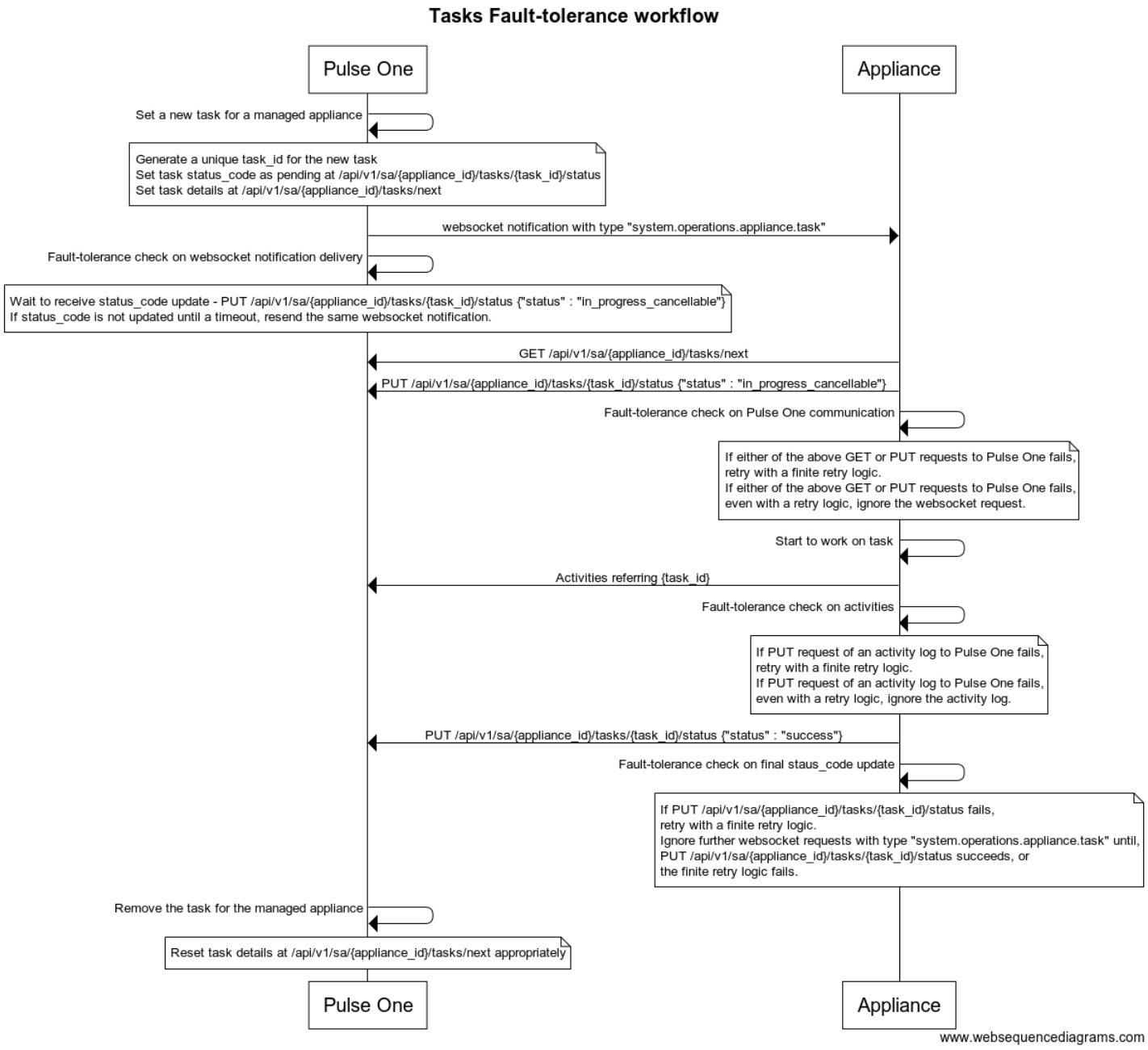
Pulse One can notify the appliance to cancel a task whose current status_code is in_progress_cancellable.
The appliance responds to a websocket notification of type "system.operations.appliance.task.cancel" by updating the status_code of the task as cancelling, and eventually cancelled. If the appliance is not able to cancel the task it can update the status_code of the task as in_progress_not_cancellable.



Tasks Cancellation Workflow

```
Pulse One->Pulse One: Start to cancel a task
    note over Pulse One
        The cancellation could be initiated by the admin
    end note
Pulse One->Pulse One: Check if the task is cancellable
    note over Pulse One
        Check task status_code at /api/v1/sa/tasks/{task_id}/status
        Verify that status_code is in_progress_cancellable
    end note
Pulse One->Appliance: websocket notification with type "system.operations.appliance.task.cancel"
Pulse One->Pulse One: Fault-tolerance check on websocket notification delivery
    note over Pulse One
        Wait to receive status_code update - PUT /api/v1/sa/tasks/{task_id}/status {"status" : "cancelling"}
        If status_code is not updated until a timeout, resend the same websocket notification.
    end note
Appliance->Pulse One: PUT /api/v1/sa/tasks/{task_id}/status {"status" : "cancelling"}
Appliance->Appliance: Cancel the current task
Appliance->Pulse One: PUT /api/v1/sa/tasks/{task_id}/status {"status" : "cancelled"}
Pulse One->Pulse One: Declare the task as cancelled
```

Fault tolerance

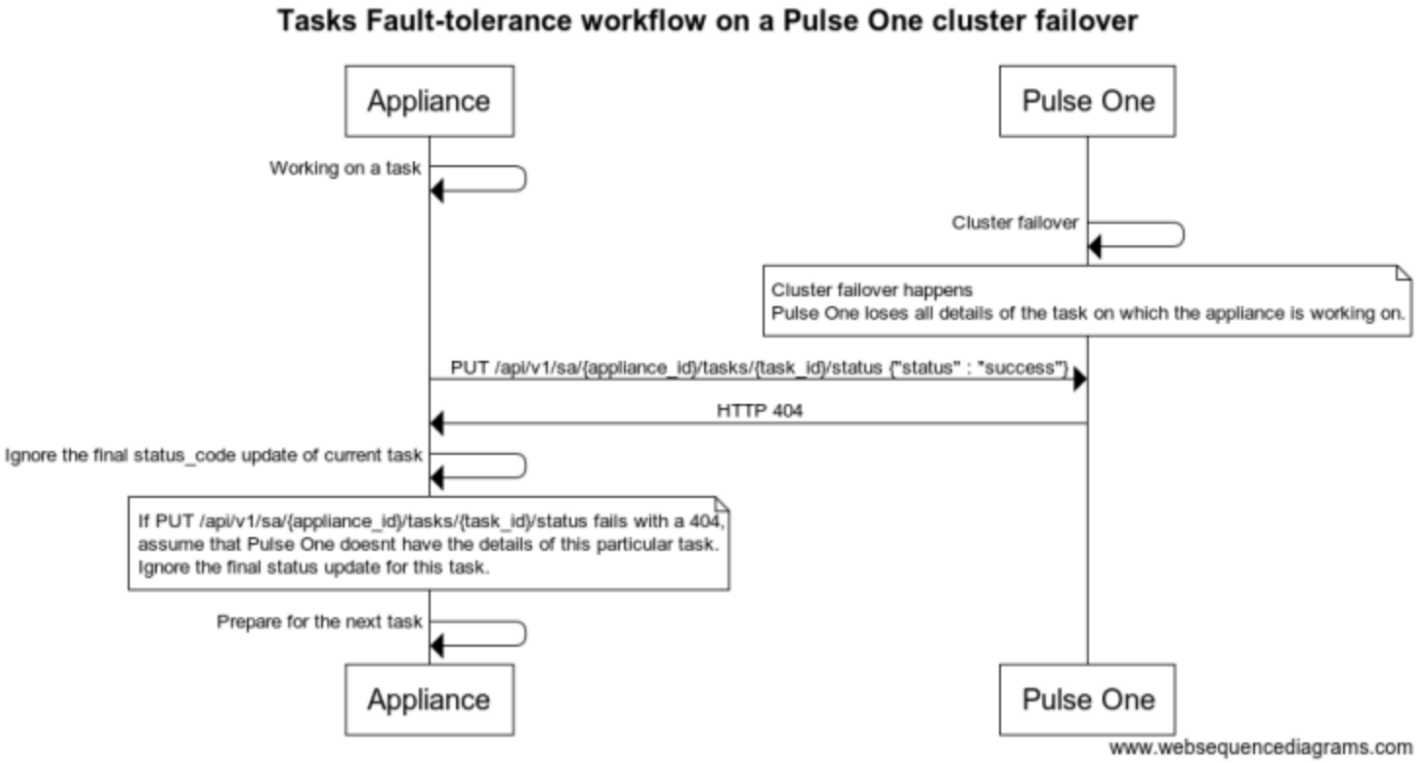


Tasks Fault-tolerance workflow

```
Pulse One->Pulse One: Set a new task for a managed appliance
   note over Pulse One
      Generate a unique task_id for the new task
      Set task status_code as pending at /api/v1/sa/tasks/{task_id}/status
      Set task details at /api/v1/sa/{appliance_id}/tasks/next
   end note
Pulse One->Appliance: websocket notification with type "system.operations.appliance.task"
Pulse One->Pulse One: Fault-tolerance check on websocket notification delivery
   note over Pulse One
      Wait to receive status_code update - PUT /api/v1/sa/tasks/{task_id}/status {"status" : "in_progress_cancellable"}
      If status_code is not updated until a timeout, resend the same websocket notification.
   end note

Appliance->Pulse One: GET /api/v1/sa/{appliance_id}/tasks/next
Appliance->Pulse One: PUT /api/v1/sa/tasks/{task_id}/status {"status" : "in_progress_cancellable"}

Appliance->Appliance: Fault-tolerance check on Pulse One communication
   note over Appliance
      If either of the above GET or PUT requests to Pulse One fails,
      retry with a finite retry logic.
      If either of the above GET or PUT requests to Pulse One fails,
      even with a retry logic, ignore the websocket request.
   end note

Appliance->Appliance: Start to work on task
Appliance->Pulse One: Activities referring {task_id}

Appliance->Appliance: Fault-tolerance check on activities
   note over Appliance
      If PUT request of an activity log to Pulse One fails,
      retry with a finite retry logic.
      If PUT request of an activity log to Pulse One fails,
      even with a retry logic, ignore the activity log.
   end note

Appliance->Pulse One: PUT /api/v1/sa/tasks/{task_id}/status {"status" : "success"}
Appliance->Appliance: Fault-tolerance check on final staus_code update
   note over Appliance
      If PUT /api/v1/sa/tasks/{task_id}/status fails,
      retry with a finite retry logic.
      Ignore further websocket requests with type "system.operations.appliance.task" until,
      PUT /api/v1/sa/tasks/{task_id}/status succeeds, or
      the finite retry logic fails.
   end note

Pulse One->Pulse One: Remove the task for the managed appliance
   note over Pulse One
      Reset task details at /api/v1/sa/{appliance_id}/tasks/next appropriately
   end note
```

Fault tolerance on a Pulse One cluster failover

In the remote case where a cluster failover happens on the Pulse One server, and the details of the current task which the appliance is working on is lost on Pulse One, the workflow on the appliance while updating the final status_code is described below.



Tasks Fault-tolerance workflow on a Pulse One cluster failover

```
Appliance->Appliance: Working on a task

Pulse One->Pulse One: Cluster failover
   note over Pulse One
      Cluster failover happens
      Pulse One loses all details of the task on which the appliance is working on.
   end note

Appliance->Pulse One: PUT /api/v1/sa/tasks/{task_id}/status {"status" : "success"}
Pulse One->Appliance: HTTP 404
Appliance->Appliance: Ignore the final status_code update of current task
   note over Appliance
      If PUT /api/v1/sa/tasks/{task_id}/status fails with a 404,
      assume that Pulse One does not have the details of this particular task.
      Ignore the final status update for this task.
   end note
Appliance->Appliance: Prepare for the next task
```

Task management APIs

Create Task for an appliance

Currently following types of appliance tasks can be created from Pulse One console.

- Backup
- Restore
- Firmware Upgrade
- ESAP Upload

Backup task creation
Request

- **Method:** POST
- **Authorization:** admin.appliances.backup|CREATE
- **Resource:** /api/v1/sa/{appliance_id}/tasks/backups
- **JSON Data:** A JSON dictionary representing an ApplianceBackupTaskCreationEntity entity.

Response Success

- **Status:** 200
- **JSON Data:** A JSON dictionary representing an ApplianceTaskEntity entity, where-in params would contain backup_id of the backup being created.

Response Error

HTTP status 409 will be returned if there is already a pending/active backup task on this appliance.

- **Status:** 409

Example

Request

```
POST /api/v1/sa/28374bca-efef-11f5-8e5b-0242ac13000d/tasks/backups HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
{
  "params": {
    "name": "My Backup-1",
    "description": "After fixing Users role and realm"
  }
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 80

{
   "id": "16417bca-efef-11e5-8e5a-0242ac13000c",
   "appliance_id": "28374bca-efef-11f5-8e5b-0242ac13000d",
   "type": "system.operations.appliance.config.backup",
   "status": "pending",
   "params": {
      "backup_id": "28374bca-ffff-11f5-8e5b-0242ac13000f"
   }
   "created": "2017-09-04T18:55:38",
   "completed": null
}
```

Restore task creation

Request

- **Method:** POST
- **Authorization:** admin.appliances.restore|CREATE
- **Resource:** /api/v1/sa/{appliance_id}/tasks/restores
- **JSON Data:** A JSON dictionary representing an ApplianceRestoreTaskCreationEntity entity.

Response Success

- **Status:** 200
- **JSON Data:** A JSON dictionary representing an ApplianceTaskEntity entity.

Response Error

HTTP status 409 will be returned if there is already a pending/active restore task on this appliance.

- **Status:** 409

Example

Request

```
POST /api/v1/sa/28374bca-efef-11f5-8e5b-0242ac13000d/tasks/restores HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
{
  "params": {
    "backup_id": "28374bca-efef-11f5-8e5b-0242ac13000e"
  }
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 80

{
    "id": "16417bca-efef-11e5-8e5a-0242ac13000c",
    "appliance_id": "28374bca-efef-11f5-8e5b-0242ac13000d",
    "type": "system.operations.appliance.config.restore",
    "status": "pending",
    "params": {
        "backup_id": "28374bca-efef-11f5-8e5b-0242ac13000e"
    },
    "created": "2017-09-04T18:55:38",
    "completed": null
}
```

Creation of Add ESAP package task

Request

- **Method:** POST
- **Authorization:** admin.appliances.upgrade|CREATE
- **Resource:** /api/v1/sa/{appliance_id}/tasks/add-esap
- **JSON Data:** A JSON dictionary representing a AddEsapTaskCreationEntity entity.

Response Success

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a ApplianceTaskEntity entity.

Response Error

HTTP status 409 will be returned if there is already a pending/active task on this appliance.

- **Status:** 409

Example

Request

```
POST /api/v1/sa/28374bca-efef-11f5-8e5b-0242ac13000d/tasks/add-esap HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
{
  "params": {
    "firmware_id": "28374bca-efef-11f5-8e5b-0242ac13000e",
    "activate": "True"
  }
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 725

{
    "appliance_id": "ec8d6737-4fcf-43b9-beff-86321de27388",
    "completed": null,
    "completion_dal_method": "sa.xml_imports.handle_task_completion",
    "content_dal_method": "sa.xml_imports.get_content_for_task",
    "created": "2019-09-18T07:21:38Z",
    "dal_method": null,
    "deadline": null,
    "expires": "2019-09-18T09:21:38Z",
    "group_id": null,
    "id": "f0dd4da7-df9e-4ad1-af60-7adde1a184fb",
    "params": {
      "activate": true,
      "config_type": "add_esap_package_to_managed_appliance",
      "firmware_id": "eec297901756b020953a3b41b1391c31"
    },
    "parent_task_id": null,
    "requeue": "2019-09-18T07:51:38Z",
    "scheduled_task_id": null,
    "status": "pending",
    "type": "system.operations.appliance.config.xml_import"
}
```

Creation of Add ESAP package for appliances group

Request

- **Method:** POST
- **Authorization:** admin.appliances.upgrade|CREATE
- **Resource:** /api/v1/sa/{group_id}/tasks/group-add-esap
- **JSON Data:** A JSON dictionary representing a AddEsapTaskCreationEntity entity.

Response

Success

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a ApplianceTaskCollectionEntity entity, where each item in the collection represents the newly created upgrade task for each appliance under the given group.

Example

Request

```
POST /api/v1/sa/28374bca-efef-11f5-8e5b-0242ac13000d/tasks/group-add-esap HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
{
  "params": {
    "firmware_id": "28374bca-efef-11f5-8e5b-0242ac13000e",
    "activate": "True"
  }
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 597

{
    "appliance_id": null,
    "completed": null,
    "completion_dal_method": null,
    "content_dal_method": null,
    "created": "2019-09-18T07:17:54Z",
    "dal_method": null,
    "deadline": null,
    "expires": "2019-09-18T09:17:54Z",
    "group_id": "507fae8f-eeb9-4863-8ac9-af6b752f9a09",
    "id": "afade59c-37ca-4e34-95f7-3e1c59e1d09c",
    "params": {
      "activate": true,
      "firmware_id": "eec297901756b020953a3b41b1391c31"
    },
    "parent_task_id": null,
    "requeue": "2019-09-18T07:47:54Z",
    "scheduled_task_id": null,
    "status": "pending",
    "type": "system.operations.appliance.config.xml_import"
}
```

Creation of appliance firmware upgrade task

Request

- **Method:** POST
- **Authorization:** admin.appliances.upgrade|CREATE
- **Resource:** /api/v1/sa/{appliance_id}/tasks/upgrades
- **JSON Data:** A JSON dictionary representing a FirmwareUpgradeTaskCreationEntity entity.

Response Success

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a ApplianceTaskEntity entity.

Response Error

HTTP status 409 will be returned if there is already a pending/active task on this appliance.

- **Status:** 409

Example

Request

```
POST /api/v1/sa/28374bca-efef-11f5-8e5b-0242ac13000d/tasks/upgrades HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
{
  "params": {
    "firmware_id": "28374bca-efef-11f5-8e5b-0242ac13000e"
  }
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 80

{
    "id": "16417bca-efef-11e5-8e5a-0242ac13000c",
    "appliance_id": "28374bca-efef-11f5-8e5b-0242ac13000d",
    "type": "system.operations.appliance.firmware.upgrade",
    "status": "pending",
    "params": {
        "firmware_id": "28374bca-efef-11f5-8e5b-0242ac13000e"
    },
    "created": "2017-09-04T18:55:38",
    "completed": null
}
```

Creation of firmware upgrade task for appliances group

Request

- **Method:** POST
- **Authorization:** admin.appliances.upgrade|CREATE
- **Resource:** /api/v1/sa/{group_id}/tasks/group-upgrades
- **JSON Data:** A JSON dictionary representing a FirmwareUpgradeTaskCreationEntity entity.

Response Success

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a ApplianceTaskCollectionEntity entity, where each item in the collection represents the newly created upgrade task for each appliance under the given group.

Example

Request

```
POST /api/v1/sa/28374bca-efef-11f5-8e5b-0242ac13000d/tasks/group-upgrades HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseone.net
{
  "params": {
    "firmware_id": "28374bca-efef-11f5-8e5b-0242ac13000e"
  }
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 80

{ [
   {
     "id": "16417bca-efef-11e5-8e5a-0242ac13000c",
     "appliance_id": "28374bca-efef-11f5-8e5b-0242ac13000d",
     "type": "system.operations.appliance.firmware.upgrade",
     "status": "pending",
     "params": {
        "firmware_id": "28374bca-efef-11f5-8e5b-0242ac13000e"
     },
     "created": "2017-09-04T18:55:38",
     "completed": null
   },
   {
     "id": "26417bca-efef-11e5-8e5a-0242ac13000d",
     "appliance_id": "38374bca-efef-11f5-8e5b-0242ac13000e",
     "type": "system.operations.appliance.firmware.upgrade",
     "status": "pending",
     "params": {
```

```
        "firmware_id": "28374bca-efef-11f5-8e5b-0242ac13000e"
      },
      "created": "2017-09-04T18:55:38",
      "completed": null
    }
  ]
}
```

Get Task details

Console can get the details of an appliance's task using the following API.

Request

- **Method:** GET
- **Authorization:** admin.appliances.tasks|READ
- **Resource:** /api/v1/sa/tasks/{task_id}
- **JSON Data:** A JSON dictionary representing an ApplianceTaskEntity entity.

Example

Request

```
GET /api/v1/sa/28374bca-efef-11f5-8e5b-0242ac13000d/tasks/16417bca-efef-11e5-8e5a-0242ac13000c
HTTP/1.1
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 75

{
    "id": "16417bca-efef-11e5-8e5a-0242ac13000c",
    "appliance_id": "28374bca-efef-11f5-8e5b-0242ac13000d",
    "type": "system.operations.appliance.config.restore",
    "params": {
        "backup_id": "38374bca-efef-11f5-8e5b-0242ac13000d"
    },
    "status": "success",
    "created": "2017-09-04T18:55:38",
    "completed": "2017-09-04T19:55:38"
}
```

Get all active/pending tasks of an appliance

Console can get all active/pending tasks of an appliance's task using the following API.

Request

- **Method:** GET
- **Authorization:** admin.appliances.tasks|READ
- **Resource:** /api/v1/sa/{appliance_id}/tasks
- **JSON Data:** A JSON dictionary representing a list of ApplianceTaskSummaryEntity entities.

Example

Request

```
GET /api/v1/sa/28374bca-efef-11f5-8e5b-0242ac13000d/tasks HTTP/1.1
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 75

[
   {
      "id": "16417bca-efef-11e5-8e5a-0242ac13000c",
      "type": "system.operations.appliance.config.restore",
      "status": "in_progress_cancellable",
   },
   {
      "id": "26417bca-efef-11e5-8e5a-0242ac13000c",
      "type": "system.operations.appliance.config.reboot",
      "status": "pending"
   }
]
```

Cancel a task from Console

Pulse One Console can use this API to cancel a task that is in in_progress_cancellable status:

Request

- **Method:** DELETE
- **Authorization:** admin.appliances.tasks|DELETE
- **Resource:** /api/v1/sa/tasks/{task_id}

Response

- **Status:** 204 (Successful)
- **Status:** 409 (task that's in progress and could not be canceled)

Get Task content

Some tasks can have content associated with them. This API can be used to fetch that content.
Content-Type and Body of the response will be specific to the task.
For Import XML task, the http response would have: - Content-Type as application/xml - Body as the XML being imported

Request

- **Method:** GET
- **Authorization:** admin.appliances.tasks|READ
- **Resource:** /api/v1/sa/tasks/{task_id}/content

Example

Request

```
GET /api/v1/sa/tasks/16417bca-efef-11e5-8e5a-0242ac13000c/content HTTP/1.1
Accept: application/xml
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 7500

<config>
...
</config>
```

Console API - User Access History

This is documentation for the API endpoints for Pulse One console to retrieve users sign-in activities of an appliance. See the Entities documentation for information about entities referred to in this document.

Get summary of users sign-in history

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/access/summary?[username=<username>][&start=<int>][&limit=<int>][&duration=<duration-type>][&realm=<realm>]

Returns summary of sign-in history (UserAccessSummariesEntity) of all users (or) specified user based on filters explained below.

Request Parameters

- start: (*int*) Indicates the number of initial results that should be skipped. (Default: 0)
- limit: (*int*) Indicates the number of results that should be returned. (Default: 10)
- username: (*str*) If specified a value, then results will include sign-in summary of users whose username match to given value. Default (without this filter) is to return sign-in summary for *all* users.
- duration: (*str*) If specified a value, then results will be filtered by the given duration. Possible values are:
  - 1d - last 24 hours (Default)
  - 7d - last 7 days
  - 30d - last 30 days
- realm: (*str*) If specified a value, then results will include sign-in summary of users whose *realm* match to given value. Default (without this filter) is to return sign-in summary from users of *all* realms.

Response

- **Status:** 200
- **JSON Data:**
  - Returns UserAccessSummariesEntity
  - items are sorted based on username field alphabetically in ascending order
  - Empty JSON array, if there are no matching user access summary entities

Example

Request

```
GET /api/v1/sa/access/summary HTTP/1.1
Accept: application/json
Host: customer.pulseone.net
```

Returns all users sign-in summary from all registered appliances.

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 18337

{
    "total": 10,
    "items": [{
        "username": "User1",
        "last_appliance_id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
        "last_login_roles": ["HR", "Engg"],
        "last_login_realm": "Users",
        "last_login_time": "2017-12-30T09:40:39Z",
        "last_login_ip": "10.204.54.20",
        "successful_logins": 21,
        "failed_logins": 2,
        "compliant_sessions": 10,
        "noncompliant_sessions": 11,
        "remediated_sessions": 4,
        "total_sessions_length": 5640,
        "average_sessions_length": 245
    }, {
        "username": "User2",
        "last_appliance_id": "d4287ae0-63b7-4c3c-b7e1-d3b3421ace0c",
        "last_login_roles": ["HR", "Admins"],
        "last_login_realm": "Admins",
        "last_login_time": "2017-12-30T09:40:39Z",
        "last_login_ip": "10.204.54.21",
        "successful_logins": 12,
        "failed_logins": 4,
        "compliant_sessions": 8,
        "noncompliant_sessions": 4,
        "remediated_sessions": 2,
        "total_sessions_length": 2620,
        "average_sessions_length": 135
    }, {
        ...
    }]
}
```

Get a specific user's sign-in history

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/access/history/<username>?[&start=<int>][&limit=<int>][&duration=<duration-type>][&realm=<realm>][&mac-address=<mac-address>][&auth-result=<auth-result>][&auth-mechanism=<auth-mechanism>][&compliance-result=<compliance-result>]

Returns sign-in history of given user () based on filters explained below.

Request Parameters

- start: (*int*) Indicates the number of initial results that should be skipped. (Default: 0)
- limit: (*int*) Indicates the number of results that should be returned. (Default: 10)
- duration: (*str*) If specified a value, then results will be filtered by the given duration. Possible values are:
  - 1d - last 24 hours (Default)
  - 7d - last 7 days
  - 30d - last 30 days
- realm: (*str*) If specified a value, then results will include history of user's sign-ins that were done using given *realm*.
- mac-address: (*str*) If specified a value, then results will include history of user's sign-ins that were done from a machine with given MAC address
- auth-result: (*int*) If specified a value, then results will be filtered by the given authentication result. Possible values are:
  - all - Include both authentication success and failures (Default behavior, when this filter was not specified)
  - success - Include only authentication success
  - fail - Include only authentication failures
  - auth-mechanism: (*str*) If specified a value, then results will be filtered by the given authentication mechanism. Possible values are:
  - all - Include user's sign-in history with all authentication mechanisms (Default, if this filter was not specified)
  - L3_Auth - Include sign-ins only with L3 auth
  - MAC_Auth - Include sign-ins only with MAC auth
  - Auth_802_1X - Include sign-ins only with 801.1x auth
  - Other_Method - Include sign-ins only other authentication mechanisms
- compliance-result: (*str*) If specified a value, then results will be filtered by the given compliance result. Possible values are:
  - all - Include user's sign-in history with any compliant result (Default behavior, if this filter was not specified)
  - COMPLIANT_YES - Include only compliant sign-ins
  - COMPLIANT_NO - Include only non-compliant sign-ins
  - COMPLIANT_REMEDIATED - Include only remediated sign-ins
  - COMPLIANT_NOT_ASSESSED - Include only compliance not assessed sign-ins

Response

- **Status:** 200
- **JSON Data:**
  - Returns UserSigninHistoryRecordsEntity
  - items are sorted based on username field alphabetically in ascending order
  - Empty JSON array, if there are no matching user access history record entities
  - If device_profiled_status is True for any sign-in history record, then the value of mac_address field can be used to make a REST call to /api/v1/sa/profiler/endpoints/{mac_address} REST endpoint to fetch additional information about the endpoint

Example

Request

```
GET /api/v1/sa/access/history/joeuser HTTP/1.1
Accept: application/json
Host: customer.pulseone.net
```

Returns joeuser's sign-in history across all registered appliances.

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 18337

{
   "total": 10,
   "items": [
   {
      'user_access_record_id': '871b4124-f1b6-46ca-8035-172355924e04'
      'login_time': '2016-11-24T07:37:41Z',
      'logout_time': '2016-11-24T07:39:41Z',
      'authentication_mechanism': 'L3_Auth',
      'authentication_succeeded': True,
      'authentication_failure_reason': null,
      'roles': ['Users', 'HR'],
      'username': 'user1',
      'realm': 'Users',
      'authentication_server_name': 'Local Auth-Server',
      'source_ip': '1.2.3.4',
      'mac_address': '12:23:34:12:23:32',
      'device_os': 'Windows_7',
      'mdm_info': null,
      'compliance': 'COMPLIANT_YES',
      'first_host_check_succeeded': True,
      'first_host_check_time': '2016-11-24T07:35:41Z',
      'first_host_check_failed_policies': null,
      'first_host_check_failure_reasons': null,
      'appliance_id': '101b4124-f1b6-46ca-8035-172355924e04',
      'device_profiled_status': True
   },
   {
      'user_access_record_id': '971b4124-f1b6-46ca-8035-172355924e05'
      'login_time': '2016-11-24T07:36:41Z',
      'logout_time': null,
      'authentication_mechanism': 'Auth_802_1X',
      'authentication_succeeded': False,
      'authentication_failure_reason': 'Max Sessions'
      'roles': ['Users', 'Engineering'],
      'username': 'user2',
      'realm': 'Users',
      'authentication_server_name': 'Corp Active Directory',
      'source_ip': '1.2.3.5',
```

```
      'mac_address': '14:23:34:12:23:32',
      'device_os': 'Mac_OS_10_13',
      'mdm_info': null,
      'compliance': 'COMPLIANT_NO',
      'first_host_check_succeeded': False,
      'first_host_check_time': '2016-11-24T07:30:41Z',
      'first_host_check_failed_policies': null,
      'first_host_check_failure_reasons': null,
      'appliance_id': '111b4124-f1b6-46ca-8035-172355924e04',
      'device_profiled_status': False
   },
   ...],
}
```

Get top users by number of login sessions

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/access/stats/login-sessions?[&duration=<duration-type>][&top-users-count=<top-users-count>]

Returns aggregated top users (usernames) by their login sessions count in the given duration.

Request Parameters

- duration: (*str*) If specified a value, then results will be filtered by the given duration. Possible values are:
  - 1d - last 24 hours (Default)
  - 7d - last 7 days
  - 30d - last 30 days
- top-users-count: (*int*) Number of top users to be aggregated. Minimum value can be 4 and maximum value can be 15. When no filter specified (default), this API returns 5 users.

Response

- **Status:** 200
- **JSON Data:**
  - JSON Object with username as key and login-sessions count as value
  - If no user sessions are available, then returns empty JSON Object
  - Results will be sorted by login sessions count in descending order

Example

Request

```
GET /api/v1/sa/access/stats/login-sessions HTTP/1.1
Accept: application/json
Host: customer.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length:

{
   "user1": 12,
   "user2": 10,
   "user4": 9,
   "user3": 6,
   "user5": 5
}
```

Get top users by average login session time

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/access/stats/average-session-times?[&duration=<duration-type>][&top-users-count=<top-users-count>]

Returns aggregated top users (usernames) by their average login session time (in seconds) in the given duration.

Request Parameters

- duration: (*str*) If specified a value, then results will be filtered by the given duration. Possible values are:
  - 1d - last 24 hours (Default)
  - 7d - last 7 days
  - 30d - last 30 days
- top-users-count: (*int*) Number of top users to be aggregated. Minimum value can be 4 and maximum value can be 15. When no filter specified (default), this API returns 5 users.

Response

- **Status:** 200
- **JSON Data:**
  - JSON Object with username as key and average login session time as value
  - If no user sessions are available, then returns empty JSON Object
  - Results will be sorted by average session time in descending order

Example

Request

```
GET /api/v1/sa/access/stats/average-session-times HTTP/1.1
Accept: application/json
Host: customer.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length:

{
   "user1": 12000,
   "user2": 10000,
   "user4": 9000,
   "user3": 6000,
   "user5": 5000
}
```

Get top users by their compliance results

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/access/stats/compliance-results?[&duration=<duration-type>][&top-users-count=<top-users-count>][&compliance-result=<compliance-result>]

Returns aggregated top users (usernames) by the count of given type of compliance results in the given duration.

Request Parameters

- duration: (*str*) If specified a value, then results will be filtered by the given duration. Possible values are:
  - 1d - last 24 hours (Default)
  - 7d - last 7 days
  - 30d - last 30 days
- top-users-count: (*int*) Number of top users to be aggregated. Minimum value can be 4 and maximum value can be 15. When no filter specified (default), this API returns 5 users.
- compliance-result: (*str*) If specified a value, then results will contain counts of top users based on given compliance result type. Possible values are:
  - COMPLIANT_YES - Include results with count of only compliant sign-ins for each top user (Default)
  - COMPLIANT_NO - Include results with count of only non-compliant sign-ins for each top user
  - COMPLIANT_REMEDIATED - Include results with count of only remediated sign-ins for each top user
  - COMPLIANT_NOT_ASSESSED - Include results with count of only compliance not assessed sign-ins for each top user

Response

- **Status:** 200
- **JSON Data:**
  - JSON Object with username as key and count of given compliance result as value
  - If no user sessions are available, then returns empty JSON Object
  - Results will be sorted by average session time in descending order

Example

Request

```
GET /api/v1/sa/access/stats/compliance-results HTTP/1.1
Accept: application/json
Host: customer.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length:

{
   "user1": 12,
   "user2": 10,
   "user4": 9,
   "user3": 6,
   "user5": 5
}
```

Get top users by their authentication mechanism

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/access/stats/auth-mechanisms?[&duration=<duration-type>][&top-users-count=<top-users-count>][&authentication-mechanism=<authentication-mechanism>]

Returns aggregated top users (usernames) by the count of given type of authentication mechanism in the given duration.

Request Parameters

- duration: (*str*) If specified a value, then results will be filtered by the given duration. Possible values are:
  - 1d - last 24 hours (Default)
  - 7d - last 7 days
  - 30d - last 30 days
- top-users-count: (*int*) Number of top users to be aggregated. Minimum value can be 4 and maximum value can be 15. When no filter specified (default), this API returns 5 users.
- authentication-mechanism: (*str*) If specified a value, then results will contain counts of top users based on given authentication mechanism type. Possible values are:
  - L3_Auth - Include results with counts of only L3 auths (Default)
  - MAC_Auth - Include results with counts of only MAC auths
  - Auth_802_1X - Include results with counts of only 801.1x auths
  - Other_Method - Include results with counts of only other authentication mechanisms

Response

- **Status:** 200
- **JSON Data:**
  - JSON Object with username as key and count of given authentication mechanism as value
  - If no user sessions are available, then returns empty JSON Object
  - Results will be sorted by average session time in descending order

Example

Request

```
GET /api/v1/sa/access/stats/auth-mechanisms HTTP/1.1
Accept: application/json
Host: customer.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length:

{
   "user1": 12,
   "user2": 10,
   "user4": 9,
   "user3": 6,
   "user5": 5
}
```

Get top users by count of successful authentications

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/access/stats/auth-success?[&duration=<duration-type>][&top-users-count=<top-users-count>]

Returns aggregated top users (usernames) with count of successful authentications in the given duration.

Request Parameters

- duration: (*str*) If specified a value, then results will be filtered by the given duration. Possible values are:
  - 1d - last 24 hours (Default)
  - 7d - last 7 days
  - 30d - last 30 days
- top-users-count: (*int*) Number of top users to be aggregated. Minimum value can be 4 and maximum value can be 15. When no filter specified (default), this API returns 5 users.

Response

- **Status:** 200
- **JSON Data:**
  - JSON Object with username as key and count of successful authentications as value
  - If no user sessions are available, then returns empty JSON Object
  - Results will be sorted by count of successful authentications in descending order

Example

Request

```
GET /api/v1/sa/access/stats/auth-success HTTP/1.1
Accept: application/json
Host: customer.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length:

{
   "user1": 12,
   "user2": 10,
   "user4": 9,
   "user3": 6,
   "user5": 5
}
```

Get top users by count of failed authentications

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/access/stats/auth-failures?[&duration=<duration-type>][&top-users-count=<top-users-count>]

Returns aggregated top users (usernames) with count of failed authentications in the given duration.

Request Parameters

- duration: (*str*) If specified a value, then results will be filtered by the given duration. Possible values are:
  - 1d - last 24 hours (Default)
  - 7d - last 7 days
  - 30d - last 30 days
- top-users-count: (*int*) Number of top users to be aggregated. Minimum value can be 4 and maximum value can be 15. When no filter specified (default), this API returns 5 users.

Response

- **Status:** 200
- **JSON Data:**
  - JSON Object with username as key and count of failed authentications as value
  - If no user sessions are available, then returns empty JSON Object
  - Results will be sorted by count of failed authentications in descending order

Example

Request

```
GET /api/v1/sa/access/stats/auth-failures HTTP/1.1
Accept: application/json
Host: customer.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length:

{
   "user1": 12,
   "user2": 10,
   "user4": 9,
   "user3": 6,
   "user5": 5
}
```

Get top roles by count of login sessions

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/access/stats/roles?[&duration=<duration-type>][&top-roles-count=<top-roles-count>]

Returns aggregated top roles by their login sessions count in the given duration.

Request Parameters

- duration: (*str*) If specified a value, then results will be filtered by the given duration. Possible values are:
  - 1d - last 24 hours (Default)
  - 7d - last 7 days
  - 30d - last 30 days
- top-roles-count: (*int*) Number of top roles to be aggregated. Minimum value can be 4 and maximum value can be 15. When no filter specified (default), this API returns 5 roles.

Response

- **Status:** 200
- **JSON Data:**
  - JSON Object with role-name as key and login-sessions count as value
  - If no user sessions are available, then returns empty JSON Object
  - Results will be sorted by login sessions count in descending order

Example

Request

```
GET /api/v1/sa/access/stats/roles HTTP/1.1
Accept: application/json
Host: customer.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length:

{
   "role1": 12,
   "role2": 10,
   "role4": 9,
   "role3": 6,
   "role5": 5
}
```

Get top users by number of login sessions with given OS type

Request

- **Method:** GET
- **Authorization:** admin.appliances|READ
- **Resource:** /api/v1/sa/access/stats/os-type?[&duration=<duration-type>][&top-users-count=<top-users-count>][&os-type=<os-type>]

Returns aggregated top users (usernames) by their login sessions count in the given duration.

Request Parameters

- duration: (*str*) If specified a value, then results will be filtered by the given duration. Possible values are:
  - 1d - last 24 hours (Default)
  - 7d - last 7 days
  - 30d - last 30 days
- top-users-count: (*int*) Number of top users to be aggregated. Minimum value can be 4 and maximum value can be 15. When no filter specified (default), this API returns 5 users.
- os-type: (*str*) Type of operating system on which login session counts needs to be aggregated. Possible values:
  - Unknown
  - Android
  - iOS
  - Blackberry
  - Windows_XP
  - Windows_Vista
  - Windows_7
  - Windows_8
  - Mac_OS
  - Linux
  - Other_OS
  - Windows_8_1
  - Windows_10
  - Chrome_OS
  - Windows (Default, when os-type filter is not specified)
  - Mac_OS_10_8
  - Mac_OS_10_9
  - Mac_OS_10_10
  - Mac_OS_10_11
  - Mac_OS_10_12
  - Mac_OS_10_13`

Response

- **Status:** 200
- **JSON Data:**
  - JSON Object with username as key and login-sessions count on the given OS type as value
  - If no user sessions are available, then returns empty JSON Object
  - Results will be sorted by login sessions count in descending order

Example

Request

```
GET /api/v1/sa/access/stats/os-type HTTP/1.1
Accept: application/json
Host: customer.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length:

{
    "user1": 12,
    "user2": 10,
    "user4": 9,
    "user3": 6,
    "user5": 5
}
```

User Access History

This API documents the way in which an appliance can provide user sign-in history and statistics data to Pulse One. See the Entities documentation for information about entities referred to in this document.

Adding or updating User Access History

This REST API endpoint will be used by appliance (PCS/PPS) to *add* or *update* its User Access records in Pulse One. It accepts JSON array with multiple *User Access* records in one call. Each User Access record is identified using its unique identifier user_access_record_id, generated by appliance. When a user access record with already existing user_access_record_id is received from an appliance, then Pulse One replaces new user access record with the exiting one.

Request

- **Method:** POST
- **Authorization:** Requires appliance
- **Resource:** /api/v1/sa/{appliance_id}/access/history
- appliance_id: *Required.* This is the value provided to the managed appliance during registration
- **JSON Data:** Request data should be in form of UserAccessRecordsEntity entity.

Response

- **Status:** 204

Example

Request

```
POST /api/v1/sa/771b4124-f1b6-46ca-8035-172355924e02/access/history HTTP/1.1
Content-Type: application/json
Transfer-Encoding: gzip
Accept: application/json
Host: foobar.api.pulseworkspace.net

{
"total": 2,
"items": [
   {
      'user_access_record_id': '871b4124-f1b6-46ca-8035-172355924e04'
      'login_time': '2016-11-24T07:37:41Z',
      'logout_time': '2016-11-24T07:39:41Z',
      'authentication_mechanism': 'L3_Auth',
      'authentication_succeeded': True,
      'authentication_failure_reason': null,
      'roles': ['Users', 'HR'],
      'username': 'user1',
      'realm': 'Users',
      'authentication_server_name': 'Local Auth-Server',
      'source_ip': '1.2.3.4',
      'mac_address': '12:23:34:12:23:32',
```

```
        'device_os': 'Windows_7',
        'mdm_info': null,
        'compliance': 'COMPLIANT_YES',
        'first_host_check_succeeded': True,
        'first_host_check_time': '2016-11-24T07:35:41Z',
        'first_host_check_failed_policies': null,
        'first_host_check_failure_reasons': null
    },
    {
        'user_access_record_id': '971b4124-f1b6-46ca-8035-172355924e05'
        'login_time': '2016-11-24T07:36:41Z',
        'logout_time': null,
        'authentication_mechanism': 'Auth_802_1X',
        'authentication_succeeded': False,
        'authentication_failure_reason': 'Max Sessions'
        'roles': ['Users', 'Engineering'],
        'username': 'user2',
        'realm': 'Users',
        'authentication_server_name': 'Corp Active Directory',
        'source_ip': '1.2.3.5',
        'mac_address': '14:23:34:12:23:32',
        'device_os': 'Mac_OS_10_13',
        'mdm_info': null,
        'compliance': 'COMPLIANT_NO',
        'first_host_check_succeeded': False,
        'first_host_check_time': '2016-11-24T07:30:41Z',
        'first_host_check_failed_policies': null,
        'first_host_check_failure_reasons': null
    }],
}
```

Response

```
HTTP/1.1 204 No Content
```

Email Domains

See the Entities documentation for information about entities referred to in this document.

Getting An Email Domain

Request

- **Method:** GET
- **Resource:** /api/domains/{domain_id}/email-domains/{email_domain_id}

Example

```
GET /api/domains/1/email-domains/5a7b5f83-9bd2-462c-af8d-21aa31a76de2 HTTP/1.1
Accept: application/json
Host: api-env.pulseworkspace.net
```

Response

- **Status:** 200
- **JSON Data:** Response data will be a EmailDomainEntity entity.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 3456

{
  "id": "9ed4cea1-2f7b-4832-b5c1-82505548807e",
  "email_domain": "example.com",
  "domain_id": "1",
  "created": "2015-02-23T23:53:09Z"
}
```

Getting A List Of Email Domains

Request

- **Method:** GET
- **Resource:** /api/domains/{domain_id}/email-domains

Example

```
GET /api/domains/1/email-domains HTTP/1.1
Accept: application/json
Host: api-env.pulseworkspace.net
```

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following keys:
- items: (*list*) A list of EmailDomainEntitys.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 9456

{
  "items": [
    {
      "id": "9ed4cea1-2f7b-4832-b5c1-82505548807e",
      "email_domain": "example.com",
      "domain_id": "1",
      "created": "2015-03-23T23:53:09Z"
    },
    {
      "id": "9ed4cea1-2f7b-4832-b5c1-82505548807a",
      "email_domain": "example2.com",
      "domain_id": "1",
      "created": "2015-04-23T23:53:09Z"
    }
  ]
}
```

Deleting An Email Domain

Request

- **Method:** DELETE
- **Resource:** /api/domains/{domain_id}/email-domains/{email_domain_id}

Example

```
DELETE /api/domains/1/email-domains/9ed4cea1-2f7b-4832-b5c1-82505548807e HTTP/1.1
Host: api-env.pulseworkspace.net
```

Response

- **Status:** 204

Example

```
HTTP/1.1 204 No Content
```

Un-enroll Google AFW domain

Request

- **Method:** POST
- **Resource:** /api/domains/{domain_id}/afw/unenroll
- **Request Data:** No Request data

Response

- **Status:** 200
- **JSON Data:** msg (str): A message indicates that the domain has been un-enrolled.

Example

Request

```
POST /api/domains/1/afw/unenroll HTTP/1.1
Host: api-env.pulseworkspace.net
```

Response

HTTP/1.1 200 OK Content-Type: application/json
{ "msg": "Un-enrolled Google AFW domain." }

Recover a deleted domain

When an admin deletes a domain, PWS server will just mark the domain as deleted. The data which belongs to the domain will be deleted by a scheduled job later (default 1 month). Before the data is deleted, admin will be able to recover the domain.

Request

- **Method:** POST
- **Resource:** /domains/{domain_id}/recover
- **Request Data:** No Request data

Response

- **Status:** 204
- **Errors:**
  - 404 Not Found - The domain with that id could not be found.

Example

Request

```
POST /domains/1/recover HTTP/1.1
Host: mgmt.pulseone.net
```

Response

```
HTTP/1.1 204 No Content
```

Getting A List Of Domains

Request

- **Method:** GET
- **Resource:** /domains
- **Optional Parameters:**
  - limit: (*int*) The maximum number of results to return. If not specified, the default limit is 10. The maximum is 500.
  - start: (*int*) The offset of the first result
  - q: (*str*) Limit results to domain name matching this string.
  - deleted: (*bool*) If deleted is True (It has to be exactly True, not 'true' or 'yes', etc), the API returns domains which were marked to be deleted; otherwise, the API returns domains which are not to be deleted.

Example

```
GET /domains HTTP/1.1
Accept: application/json
Host: mgmt.pulseone.net
```

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following keys:
  - data: (list) A list of JSON dictionary with the following keys:
    - id: (*int*) ID of a domain
    - name: (str) Name of a domain
    - db_name: (*str*) Database name of a domain
    - db_username: (*str*) The username of a domain database
    - provider_name: (*str*) The hostname
  - total: (*int*) The count of the records.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 9456

{
  "data": [
    {
      "id": 1,
      "name": "apitests",
      "db_name": "apitests_f9b3d97e",
      "db_username": "apitests_12",
      "provider": "pulseone.net"
    },
    {
      "id": 2,
      "name": "test",
      "db_name": "tests_f9b3d97e",
      "db_username": "tests_12",
      "provider": "pulseone.net"
    },
  ],
  "total": 2
}
```

PCLS API - Pulse Cloud Licensing Service API

This API documentation defines the interaction between Virtual License Server (VLS),associated with a PCS appliance and Pulse Cloud Licensing Service(PCLS).
The Pulse Cloud Licensing Server (PCLS) is responsible for providing license keys to the VLS, given an *auth code* as well as ensuring that multiple VLS do not use the same license.
Read the Architecture of Pulse One Cloud Licensing Service documentation, for PCLS architecture.
There are two requests, the license provisioning request and a heartbeat request which is sent periodically by the VLS to PCLS.

PCLS Provisioning License

This request provisions the license and provides the license keys to the VLS, given a valid *auth code* and three *identifying attributes* that uniquely identifies a VLS.

Request

- **Method:** POST
- **Resource:** /api/v1/pcls/prov
  HTTP Custom Header: The following custom header is sent as part of the Request
  `X-PCLS-AUTH`: (*string*) The unique authorization code that identifies a customer product purchase.
- **Parameters:** The following request parameters are mandatory. (Atleast one of ipv4 or ipv6 is needed): ```
  - machineid: PCS derives the machineid from the vmid assigned by the hypervisor.
  - mac: The MAC address of the VM's internal interface.
  - ipv4 : The IPV4 address of the VM's internal interface.
  - ipv6 : The IPV6 address of the VM's internal interface. ```

Example

PCLS Provisioning License

```
POST /api/v1/pcls/prov HTTP/1.1
Content-Type: application/json
X-PCLS-AUTH: 6e29-deda-1987-8611
Accept: application/json
Host: pcls.pulseone.net

{
  "machineid": "VASPH0P5JUKBNLBHS",
  "mac": "5a:00:01:78:84:10",
  "ipv4": "172.20.61.67"
}
```

Response

- **Status:**
  - 200 - When the request was authorized, i.e., a valid authcode was supplied.
  - 403 - When the request was unauthorized, i.e., the authcode supplied was invalid.
  - 503 - Service Unavailable/PCLS out-of-service
- **JSON Data:**
  Response data will be in the form of a JSON dictionary with the following key:
  - license_keys: (*list*) A list of *license keys*

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 254

{
  "license_keys": [
    {
      "hydrogen freshman video wrench molecule harness
       shoulder stadium pride indigo hinge",
      "nitrogen sophomore audio plier atom prop knee field honor orange nail"
    }
  ]
}
```

PCLS Heartbeat

VLS sends a heartbeat periodically to the PCLS using this request.

Request

- **Method:** POST
- **Resource:** /api/v1/pcls/heartbeat
- HTTP Custom Header: The following custom header is sent as part of the Request:
  'X-PCLS-AUTH`: (*string*)
  A hashed license key of any **one** of the license keys,that were sent during the provisioning of the license.
  The license keys are hashed using SHA-1 cryptographic hash function.
- **Parameters:** The following request parameters are mandatory. Atleast one of ipv4 or ipv6 is needed:
  - machineid: PCS derives the machineid from the vmid assigned by the hypervisor.
  - mac: The MAC address of the VM's internal interface.
  - One of ipv4 or ipv6 is required:
  - ipv4 : The IPV4 address of the VM's internal interface.
  - ipv6 : The IPV6 address of the VM's internal interface.

Example - PCLS Heartbeat

```
POST /api/v1/pcls/heartbeat HTTP/1.1
Content-Type: application/json
X-PCLS-AUTH: e8dedad1j1e05a42decb58ee3e7fdae1f540
Accept: application/json
Host: pcls.pulseone.net

{
  "machineid": "VASPH0P5JUKBNLBHS",
  "mac": "5a:00:01:78:84:10",
  "ipv4": "172.20.61.67"
}
```

Response

- **Status:**
  - 200 - When the request was authorized, i.e., a valid hashed license key was supplied.
  - 401 - When the request was unauthorized, i.e., the hashed license key supplied was invalid.
  - 500 - Internal Server Error: When the request was authorized, but it gave a non-zero response code back
  - 503 - Service Unavailable/PCLS out-of-service
  - **JSON Data:**
    Response data will be in the form of a JSON dictionary with the following keys:
    - return_code: (_int_) A return code of 0 is success and would return one of the following error codes. The VLS must disable the license keys, if it receives an error return code along with an error message.
    - return_message: (*string*) The following would be the error code, along with their corresponding error message:
      10000 - "This hashed license key is unknown to PCLS"
      10001 - "This hashed license key is in use by another VLS"
    - grace_period: (*int*) *grace period* in hours.

Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 87

{
  "return_code": 0,
  "return_message" : "OK",
  "grace_period" : 24
}
```

Workspace Android Mobile Apps API

API endpoints to manage Android mobile apps in the app catalog.
This API mirrors the iOS API of the same name, except iOS app bundle upload.
Note: Android apps use the platform value of 'android', but their API endpoints are served under the afw route.

Create Mobile App

Request

- **Method:** POST
- **Resource:** /api/v1/afw/apps
- **JSON Data:** A MobileAppEntity.Update entity.

Response

- **Status:** 200

Example

Request

```
PUT /api/v1/afw/apps HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net

{
  "package_name": "com.cisco.webex",
  "title": "Cisco Webex",
  "tags": [
    "productivity"
  ],
  "app_config": {
    "schema": {
      "schema": ...,
      "version": ....
    },
    "values": ...
  },
  "app_permissions": {
    "schema": {
      "schema": ...,
      "version": ....
    },
    "values": ...
  },
```

```
  "category": "Internet",
  "creator": "Interwebs",
  "version": "1.0.1",
  "is_manual": false,
  "download_url": "",
  "is_required": true,
  "network_access": "direct"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 575

{
  "id": "c54677e8-b0f5-11e6-9ee1-0242c0aa0007",
  "platform": "afw",
  "package_name": "com.cisco.webex",
  "title": "Cisco Webex",
  "tags": [
    "productivity"
  ],
  "app_config": {
    "schema": {
      "schema": ...,
      "version": ....
    },
    "values": ...
  },
  "app_permissions": {
    "schema": {
      "schema": ...,
      "version": ....
    },
    "values": ...
  },
  "category": "Internet",
  "creator": "Interwebs",
  "version": "1.0.1",
  "is_manual": false,
  "download_url": "",
  "icon_url": "https://s3.amazonaws.com/unitydev-io-icons/com.cisco.webex.png",
  "licenses": "not licensed",
  "vpp_total": 0,
  "vpp_used": 0
  "source": "appstore",
  "is_required": true,
  "network_access": "direct",
  "modified_on": "2016-11-22T20:53:50",
  "created_on": "2016-11-22T20:53:50",
}
```

Get Mobile App

Retrieve a Mobile App by package name.

Request

- **Method:** GET
- **Resource:** /api/v1/afw/apps/{package-name}

Response

- **Status:** 200
- **JSON Data:** Response will be a MobileAppEntity.

Example

Request

```
GET /api/v1/afw/apps/com.cisco.webex /1.1
Accept: application/json
Host: customer.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 575

{
  "id": "c54677e8-b0f5-11e6-9ee1-0242c0aa0007",
  "platform": "android",
  "package_name": "com.cisco.webex",
  "title": "Cisco Webex",
  "tags": [
    "productivity"
  ],
  "app_config": {
    "schema": {
      "schema": ...,
      "version": ....
    },
    "values": ...
  },
  "app_permissions": {
    "schema": {
      "schema": ...,
      "version": ....
    },
    "values": ...
  },
```

```
  "category": "Internet",
  "creator": "Interwebs",
  "version": "1.0.1",
  "is_manual": false,
  "download_url": "",
  "icon_url": "https://s3.amazonaws.com/unitydev-io-icons/com.cisco.webex.png",
  "licenses": "not licensed",
  "vpp_total": 0,
  "vpp_used": 0
  "source": "appstore",
  "is_required": true,
  "network_access": "direct",
  "modified_on": "2016-11-22T20:53:50",
  "created_on": "2016-11-22T20:53:50",
}
```


Update Mobile App

Request

- **Method:** PUT
- **Resource:** /api/v1/afw/apps/{package-id}
- **JSON Data:** A MobileAppEntity.Update object.

Response

- **Status:** 204

Example

Request

```
PUT /api/v1/afw/apps/com.cisco.webex HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net

{
  "package_name": "com.cisco.webex",
  "title": "Super Awesome Cisco Webex",
  "tags": [
    "meetings"
  ],
  "app_config": {
    "schema": {
      "schema": ...,
      "version": ....
    },
    "values": ...
  },
```

```
  "app_permissions": {
    "schema": {
      "schema": ...,
      "version": ....
    },
    "values": ...
  },
  "category": "Intranet",
  "creator": "Interwebs",
  "version": "1.0.1",
  "is_manual": false,
  "download_url": "",
  "is_required": false,
  "network_access": "direct"
}
```

Response

```
HTTP/1.1 204 No Content
```

Delete Mobile App

An endpoint for deleting mobile apps from the app catalog.
The server will reject the request if the mobile app belongs to any app rules on any policies. The server will return a list of policy entities with id and name. The name is suitable for displaying a helpful list to the IT Admin so that they know what policies must be edited to remove the app rule for the mobile app being deleted. The id is useful for linking directly to the policy.

Request

- **Method:** delete
- **Resource:** /api/v1/afw/apps/{package-id}

Response

- **Status:** 204 - The mobile app has been deleted.
- **Status:** 400 - The mobile app belongs to policies and cannot be deleted.
- **JSON Data:** Response will be a dictionary with a key of items that will contain a list of PolicyEntity. The policy items will contain id and name.

Example

Request

```
DELETE /api/v1/afw/apps/com.cisco.webex HTTP/1.1
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 204 No Content
```

OR

HTTP/1.1 400 Bad Request

```
{
  "items": [
    {
      "id": 1,
      "name": "Example Policy"
    }
  ]
}
```

List Apps

In order to list all apps for this platform, please use the (Workspaces Mobile Apps Search API)[/api/unity/workspaces/mobile-apps.md] and specify the platform.

Searching For Mobile apps

Use Google's Product API to search for Apps.

Request

- **Method:** GET
- **Resource:** /api/v1/afw/apps/search
- **Parameters:**
  - q: *required* Search keyword.
  - country_code: *optional* Two-letter country code. Default is "US".
  - start: Number of results to skip (default 0)
  - limit: Maximum number of results to return (default 10, maximum 100)

Response

- **Status:** 200
- **JSON Data:**
  - o total (int): Total number of results available
  - o items : A list of MobileAppSearchResult.
- **Errors:**
  - o 50202: raised when the Google API is not returning expected results. This is expected to be temporary.

Example

```
GET /api/v1/afw/apps/search?q=WebEx&start=0&limit=1
Host: api.pulseworkspace.net
```

Request

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "total": 2910
  "items": [
    {
      "name": "Cisco WebEx Meetings",
      "package_name": "com.cisco.webex.meetings",
      "image": "https://lh3.ggpht.com/MmfNO-
OC2YuTrxTYhhPypVgxPIPLe1uDAW5HlcUWbvawzIAevWqqa2_dMOgjWAJTuy8=w300",
      "url": "https://itunes.apple.com/store/apps/details?id=com.cisco.webex.meetings",
      "description": "Take your Web meetings anywhere!  Join any web conference...",
      "creator": "Cisco",
      "category": "Productivity",
      "source": "googleplay"
    }
  ]
}
```

Upload App Bundle

Request

- **Method:** POST
- **Resource:** /api/v1/afw/apps/{app title}/bundle
- **multipart/form-data:** Form data field named version which contains package version, also file contains the ipa bundle file.

Response

- **Status:** 200
- **JSON Data:** Response will be a MobileAppEntity.UploadResult.

Example

Request

```
POST /api/v1/afw/apps/Webex/bundle HTTP/1.1
Content-Length: <length>
Content-Type: multipart/form-data; boundary=723e82b4e41f4d0d9c66db24f035a326
Host: api.pulseone.net

--723e82b4e41f4d0d9c66db24f035a326
Content-Disposition: form-data; name="version"

1.0.1
--723e82b4e41f4d0d9c66db24f035a326
Content-Disposition: form-data; name="file"; filename="com.cisco.webex.1.0.1.apk"
Content-Type: application/octet-stream

--723e82b4e41f4d0d9c66db24f035a326--
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: <length>

{
  "title": "Webex",
}
```

AFW - Products Approval / Search / Lookup

These endpoints provide a wrapper around the google APIs for product approval, search, and lookup by package name.

Product Search

Use Google Search to find Android products.

Request

- **Method:** GET
- **Resource:** /api/v1/afw/products?q=WebEx
- **Parameters:**
- q: Search keywords
- start: Number of results to skip (default 0)
- limit: Maximum number of results to return (default 10, maximum 100)

Response

- **Status:** 200
- **JSON Data:**
- total (int): Total number of results available
- items : A JSON array of objects, each with at least these fields:
  - name: App name
  - package_name: Package name
  - image: Image icon URL
  - url: Play store URL
  - description: App short description

Example

```
GET /api/v1/afw/products?q=WebEx
Host: customer.pulseworkspace.net
```

Request

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "total": 2910
  "items": [
    {
      "name": "Cisco WebEx Meetings",
      "package_name": "com.cisco.webex.meetings",
      "image": "https://lh3.ggpht.com/MmfNO-bvawzIAevWqqa2_dMOgjWAJTuy8=w300",
      "url": "https://play.google.com/store/apps/details?id=com.cisco.webex.meetings",
      "description": "Take your Web meetings anywhere!  Join any web conference..."
    }
  ]
}
```

Generate Approval URL

Generates a URL that can be rendered in an iframe to display the permissions (if any) of a product. An enterprise admin must view these permissions and accept them on behalf of their organization in order to approve that product.

Request

- **Method:** POST
- **Resource:** /api/v1/afw/products/<product-id>/generate-approval-url
- **Request Data:** No Request data

Response

- **Status:** 200
- **Errors:**
  - 404 Not Found - The product ID is invalid
- **JSON Data:** A JSON dictionary representing a GenerateProductApprovalUrlEntity entity.

Example

Request

```
POST /api/v1/afw/products/net.pulsesecure.pulsesecure/generate-approval-url
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK Content-Type: application/json
{ "url": "
https://play.google.com/work/permissionsiframe?token=Qa6GMM_CJC4 " }
```

Approve Product

Approves the specified product (and the relevant app permissions, if any).

Request

- **Method:** POST
- **Resource:** /api/v1/afw/products/<product-id>/approve
- **Request Parameters:**
  - approve_forever: True if the IT Admin wishes to approve all permissions for this application forever, False otherwise.
- **Request Data:** Request data should be in the form of a JSON dictionary representing a [ApproveProductEntity] entity.

Response

- **Status:** 204
- **Errors:**
- 404 Not Found - The product ID is invalid

Example

Response

```
POST /api/v1/afw/products/net.pulsesecure.pulsesecure/approve?approve_forever=true
Content-Type: application/json
Host: customer.pulseworkspace.net

{
  "url": "https://play.google.com/work/permissionsiframe?token=Qa6GMM_CJC4"
}
```

Response

```
HTTP/1.1 204 No Content
```

AFW - WIFI Certificate API

Update installed version of Wifi key pair and certificate if necessary

When the policy specifies that the wifi requires private key authentication, the client will check whether it needs a new key pair and certificate by submitting the version it received last time it got one, or an empty string if it has never downloaded a key and certificate before.
The server compares the version provided by the client with a version it calculates. If they differ, it generates a new private key and certificate and sends it to the client. Otherwise, it will return an empty response.
The key pair and certificate are serialized using pkcs12 format and then encoded into base64 into the certificate field of the response.
The key pair and certificate are not saved on the server.

Request

- **Method:** POST
- **Resource:** /api/v1/afw/spaces/{workspace-id}/wifi-certificate
- **JSON Data:** A JSON encoded WifiCertificateEntity.Update

Response (Key pair out of date)

- **Status:** 200 OK
- **JSON Data:** A JSON encoded WifiCertificateEntity

Response (Key pair up to date)

- **Status:** 204 No Content

Example

Request

```
POST /api/v1/afw/spaces/1d89e147-a845-4825-93bf-154592454c25/wifi-certificate HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net

{
  "version": "2a20455d639e31f282a535d6e43b4876c37a4eb2e79d111457c4acdf75be4e95"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 128

{
  "certificate": "05C6i6wC7MYTHVOZ3ciV0ukfE0DUFYdxK15wnCawLwMLX62kf01AYXLCAlyg15uX",
  "cert_alias": "pulsesecure.wifi.cert",
  "format": "pkcs12",
  "password": "apassword",
  "version": "481a9340edbcd23a222685466d8b5a4940d83a1281b92df8bf951cfc870b5369"
}
```

AFW - Policies

Retrieving Workspace Policy

Request a list of apps needed / allowed in the workspace.

Request

- **Method:** GET
- **Resource:** /api/v1/afw/spaces/<workspace-id>/policy

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a WorkspacePolicyEntity entity.
- **Errors:**
  - 40x - Errors will be propagated from Google if the token is invalid or the domain has already been enrolled somewhere.

Example

Request

```
GET /api/v1/afw/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/policy HTTP/1.1
Accept: application/json
Host: api-env.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1331

{
   "policy_version": "123123",
   "settings": {
      "server_name": "demo.pulsesecure.net",
      "user_name": "flast",
      "google_account" : "username@democorp.com",
      "passcode_set_date": "last modification date of passcode. determine if should get passcode",
      "location_enabled": true
   },
   "properties": {
      "usb_debugging": "allow",
      "vpn_enabled": true,
      "vpn_host": "https://workspace.acmegizmo.com/cert/",
      "vpn_use_scep_certificate": true,
      "vpn_force_update_scep_certificate": true,
      "wifi_use_scep_certificate": true,
      "wifi_force_update_scep_certificate": true,
      "activesync_use_scep_certificate": true,
      "activesync_force_update_scep_certificate": true
```

```
  },
  "app_rules": [
    {
      "package_name": "com.enterprise.myapp",
      "command": "add",
      "network_access": "direct",
      "group_tags": [ "browser" ],
      "source": "enterprise",
      "url": "http://myserver.com/download/myapp.apk",
      "icon_url": "https://s3.amazonaws.com/icons/myapp.png",
      "version_code": 1234,
      "title": "My App",
      "app_permissions": {
        "values": {
          "android.permission.ACCESS_COARSE_LOCATION": "denied",
          "android.permission.ACCESS_FINE_LOCATION": "default",
          "android.permission.ACCESS_NETWORK_STATE": "granted"
        }
      }
    },
    {
      "package_name": "net.pulsesecure.pulsesecure",
      "network_access": "direct",
      "source": "market",
      "title": "PulseSecure VPN",
      "app_restrictions": {
        "profile_name": "whatever",
        "url": "https://sj-vpn.pulsesecure.net",
        "action": "create",
        "username": "dtirosh",
        "vpn_default": true
      }
    }
  ],
  "email_configuration": {
    "schema_version": 1,
    "pkg_name": "csapp.com.android.email",
    "activesync_device_id": "c4d67b1147b3ff1e7d40d643e76584a7",
    "configuration_hash": "2d33919ea4d6fe900f2d83653c0e8a42",
    "display_name": "test_user",
    "email_address": "test@mobilespaces.com",
    "security_type": "ssl",
    "security_accept_all_certs": True,
    "server_address": "exch.mobilespaces.net",
    "server_port": 443,
    "service_type": "eas",
    "user_name": "testuser"
   }
}
```

AFW - GCM Message

When the server sends a GCM message to the client, the message is formatted as follows

- message - message envelope
- action - the GCM message action. any of:
  - check_in - request the client to refresh the policy. The client will issue a GET policy.
  - wipe - request the client to wipe. Client will issue a POST state flagged as "wiped"
  - lock - admin request to lock device. Client will issue a POST state flagged as "locked"
  - send_logs - request from the client to upload logs

```
{
   "message" : {
      "action" : "check_in"
   }
}
```

AFW - Workspace State

Update State

Request

- **Method:** POST
- **Resource:** /api/v1/afw/spaces/{workspace-id}/state
- **JSON Data:** Request data should be in the form of a JSON dictionary representing an WorkspaceState entity.

Note: workspace_state is optional. This field is sent by the client whenever there is a state change that the server needs to know about.

Response

- **Status:** 204 Server returns status code 400 if inputs such as package name, state, action are invalid. The maximum number of app states allowed in the request is 100. Server returns status code 400 if the number of app states pass the limit.

Example

Request

```
POST /api/v1/afw/spaces/f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b/state HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: customer.pulseworkspace.net

{
  "app_states": [
    {
      "action": "server_install",
      "package_name": "com.google.chrome",
      "state": "installed"
    },
    {
      "action": "server_enable",
      "package_name": "com.google.picasa",
      "state": "installing"
    },
    {
      "action": "no_action",
      "package_name": "com.google.camera",
      "state": "system_enabled"
    }
  ],
  "delta": false,
  "workspace_state": "locked"
}
```

Response

```
HTTP/1.1 204 No Content
```

AFW - VPN Certificate SCEP configurations API

Get SCEP configurations needed to create a SCEP certificate request.

When policy indicates client needs to use SCEP to request VPN certificate from external PKI server, client will get SCEP related configurations from PWS server. Important components of the SCEP configurations includes external SCEP server URL and challenge. Response from server will also include other information for example Subject Names which client can use when creating certificate requests.

Request

- **Method:** GET
- **Resource:** /api/v1/afw/spaces/{workspace-id}/scep-configuration

Response

- **Status:** 200 OK
- **JSON Data:** A JSON encoded CertificateScepRequestEntity

Example

Request

```
GET /api/v1/afw/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/scep-configuration HTTP/1.1
Accept: application/json
Host: api-env.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 206

{
  "ca_name": "Default_CA",
  "challenge": "dummy_scep_challenge",
  "key_size": 2048,
  "key_type": "RSA",
  "key_usage": 5,
  "scep_url": "http://scep.example.com",
  "subject_cn": "user_1",
  "subject_o": "domain_1.example.com",
  "subject_email": "user1@domain_1.example.com"
}
```

Android App Permissions Schema

This API provides the latest version of an applications permissions schema. This API is similar to the App Config Schema API. App permissions should be created and updated with a MobileApp using the app_permissions attribute and a AppPermissionsEntity data structure.
This Google API is documented under the Products API.

Get App Permissions Schema

Get the app permissions schema for an Android app.

Request

- **Method:** GET
- **Resource:** /api/v1/afw/apps/<package-name>/app-permissions-schema

Response

- **Status:** 200
- **JSON Data:** The response data will be an AppPermissionsSchemaEntity.
- **Errors:**
  - 404 Not Found - An app with that package-name could not be found.

Example

Request

```
GET /api/v1/afw/apps/com.cisco.webex/app-permissions-schema
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "version": "1e93e453ae80459d9a031d24f4f7a2d4",
  "schema":
  [
    {
      "name": "android.camera",
      "label": "android.camera",
      "choices": ["denied", "default", "granted"],
      "prop_type": "choice"
    },
```

```
    {
      "name": "android.contacts",
      "label": "android.contacts",
      "choices": ["denied", "default", "granted"],
      "prop_type": "choice"
    }
  ]
}
```

AFW - VPN Certificate API

Update installed version of VPN key pair and certificate if necessary

When the policy specifies that the vpn requires private key authentication, the client will check whether it needs a new key pair and certificate by submitting the version it received last time it got one, or an empty string if it has never downloaded a key and certificate before.
The server compares the version provided by the client with a version it calculates. If they differ, it generates a new private key and certificate and sends it to the client. Otherwise, it will return an empty response.
The key pair and certificate are serialized using pkcs12 format and then encoded into base64 into the certificate field of the response.
The key pair and certificate are not saved on the server.

Request

- **Method:** POST
- **Resource:** /api/v1/afw/spaces/{workspace-id}/vpn-certificate
- **JSON Data:** A JSON encoded VpnCertificateEntity.Update

Response (Key pair out of date)

- **Status:** 200 OK
- **JSON Data:** A JSON encoded VpnCertificateEntity

Response (Key pair up to date)

- **Status:** 204 No Content

Example

Request

```
POST /api/v1/afw/spaces/1d89e147-a845-4825-93bf-154592454c25/vpn-certificate HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net

{
  "version": "2a20455d639e31f282a535d6e43b4876c37a4eb2e79d111457c4acdf75be4e95"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 128

{
  "certificate": "05C6i6wC7MYTHVOZ3ciV0ukfE0DUFYdxK15wnCawLwMLX62kf01AYXLCAlyg15uX",
  "cert_alias": "pulsesecure.vpn.cert",
  "format": "pkcs12",
  "password": "apassword",
  "version": "481a9340edbcd23a222685466d8b5a4940d83a1281b92df8bf951cfc870b5369"
}
```

Google Play Store Custom App delegation API

API endpoints to manage Android custom apps API permission.
Google provides Google Play Custom App API so EMM provider can publish custom apps to specific enterprise in Google Play Store.
Before an EMM provide can do that, the Developer account of the enterprise need to delegate publishing rights to A Service Account of the EMM provider.

Get URL to delegate Custom App permission to Pulse

In Google Play Custom App Publishing API documents (https://developers.google.com/android/work/play/custom-app-api/get-started), there is this section:
“Obtain private app publishing rights As an EMM or third-party app developer, you need to embed a button (or similar feature) into your existing console that sends end users to the Play Console URL below, which guides them through the account delegation process.”

```
title Delegate Google Custom Apps upload permission
Browser -> UI: Admin clicks `delegate` button.
UI -> PWS: "POST https://api.domain.pulseone.net/api/v1/afw/domains/app-publishing-delegation-url\nnext: https://domain.pulseone.net/.../android-for-work\n"
PWS->UI: "url: https://play.google.com/apps/publish/delegatePrivateApp?\nservice_account=<Pulse service account>&\ncontinueUrl=https://api.domain.pulseone.net/api/v1/afw/domains/custom-app-delegation-callback?t=<tmp token id>"
UI -> Browser: Redirect to `https://play.google.com/apps...`
Browser -> Google: Admin login with Google developer account and follow delegation wizard.
Google -> Browser: redirect to "https://api.domain.pulseone.net/api/v1/afw/domains/custom-app-delegation-callback?\ndeveloperAccount=<developer account ID>&developerAccountLink=https://play.google.com/apps/publish?account=developerAccountId"
Browser -> PWS: "GET https://api.domain.pulseone.net/api/v1/afw/domains/custom-app-delegation-callback..."
Note over PWS: Save developer id in domain property:\ngoogle_enterprise_dev_account_for_custom_app
PWS -> Browser: Redirect to `https://domain.pulseone.net/.../android-for-work`
```

Request

- **Method:** POST
- **Resource:** /api/v1/afw/domains/app-publishing-delegation-url
- **JSON Data:**
- next: Console URL to return to on completion (e.g. the android-for-work settings page of the current domain)

Response

- **Status:** 200
- **JSON Data:**
  - url: URL to redirect the user to delegate Customer App rights.
- **Errors:**
  - 40x - Errors will be propagated from Google if there is a problem

Example

Request

```
POST /api/v1/afw/domains/app-publishing-delegation-url HTTP/1.1
Accept: application/json
Host: api.domain.pulseone.net
Content-Type: application/json
Content-Length: 75

{
  "next": "https://domain.pulseone.net/admin/settings/android-for-work"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 77

{
  "url": "https://play.google.com/apps/publish/delegatePrivateApp?service_account=<Pulse service
account>&continueUrl=https://api.domain.pulseone.net/api/v1/afw/domains/"
}
```

Google Custom Apps Delegation Callback

When we request Enterprise Admin to delegate Google Custom App uploading permission to Pulse Service account, Google will ask Admin to login with their Google Developer account, then grant The Dedicated Pulse Secure Custome Apps Service Account the permission to publish App to the Enterprise.
After the delegation process, Google will redirect the Browser to access the callback API endpoint on PWS server.
On success, this updates this domain property:

- google_enterprise_dev_account_for_custom_app : ID of the Google Developer account for the Enterprise.

Then PWS will redirect Browser back to console UI page.

Request

- **Method:** GET
- **Resource:** /api/v1/afw/domains/custom-app-delegation-callback
- **Query Parameters:**
  - developerAccount: ID of the Google Developer account for the Enterprise.
  - developerAccountLink: Link to the developer account info page.

Response

- **Status:** 303
- **Errors:**
  - 40x - Errors will be propagated from Google if there is a problem

Example

Request

```
GET /api/v1/afw/domains/custom-app-delegation-callback?developerAccount=<developer account
ID>&developerAccountLink=https://play.google.com/apps/publish?account=developerAccountId HTTP/1.1
Accept: application/json
Host: api.domain.pulseone.net
```

Response

```
HTTP/1.1 303 See Other
Location: https://domain.pulseone.net/admin/settings/android-for-work
```

Workspace Debug Data API

API endpoint to upload client debugging information visible as activity in the console

Upload Debug Data Dump Zip File

Client should upload data dump as a .zip archive.

Request

- **Method:** POST
- **Resource:** /api/v1/afw/spaces/{space-id}/debug-data
- **Content-Type:** application/zip
- **Body:** The binary contents of the .zip file

Response

- **Status:** 204

Example

Request

```
POST /api/v1/afw/spaces/b61bcc6e-e576-11e5-afdc-0242ac13000f/debug-data HTTP/1.1
Host: customer.api.pulseworkspace.net
Content-Type: application/zip
Content-Length: 4230132

BINARY-ZIP-FILE-DATA
```

Response

```
HTTP/1.1 204 OK
```

AFW - ActiveSync Certificate API

Update installed version of ActiveSync key and certificate pair if necessary

When the policy specifies that the ActiveSync requires private key authentication, the client will check whether it needs a new key pair and certificate by submitting the version it received last time it got one, or an empty string if it has never downloaded a key and certificate before.
The server compares the version provided by the client with a version it calculates. If they differ, it generates a new private key and certificate and sends it to the client. Otherwise, it will return an empty response.
The key pair and certificate are serialized using pkcs12 format and then encoded into base64 into the certificate field of the response.
The key pair and certificate are not saved on the server.

Request

- **Method:** POST
- **Resource:** /api/v1/afw/spaces/{workspace-id}/activesync-certificate
- **JSON Data:** A JSON encoded ActiveSyncCertificateEntity.Update

Response (Key pair out of date)

- **Status:** 200 OK
- **JSON Data:** A JSON encoded ActiveSyncCertificateEntity

Response (Key pair up to date)

- **Status:** 204 No Content

Example

Request

```
POST /api/v1/afw/spaces/1d89e147-a845-4825-93bf-154592454c25/activesync-certificate HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net

{
  "version": "2a20455d639e31f282a535d6e43b4876c37a4eb2e79d111457c4acdf75be4e95"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 128

{
  "certificate": "05C6i6wC7MYTHVOZ3ciV0ukfE0DUFYdxK15wnCawLwMLX62kf01AYXLCAlyg15uX",
  "cert_alias": "pulsesecure.activesync.cert",
  "format": "pkcs12",
  "password": "apassword",
  "version": "481a9340edbcd23a222685466d8b5a4940d83a1281b92df8bf951cfc870b5369"
}
```

AFW - Workspace Registration

Workspace registration for AFW devices can happen with either PIN or an Authentication header using the Bearer scheme. This document does not cover obtaining a PIN or Bearer token.

Registration With PIN

Exchange a registration code and user email for workspace UUID and credentials.
Returns HTTP status 403 if the user email and/or reg_key pair are not found in our database of pending registrations, either because they are entered incorrectly, made up, or already used.

Request

- **Method:** POST
- **Resource:** /api/v1/afw/register
- **JSON Data:** A JSON dictionary representing a WorkspaceRegistrationRequestEntity entity.

Response

- **JSON Data:** A JSON dictionary representing an AAfwWorkspaceRegistrationResponseEntity entity.
- **Errors:**
  - 403 - The user email and/or reg_key pair are not found in our database of pending registrations, either because they are entered incorrectly, made up, or already used.
  - 400 {code = 40013} - If the system is not enrolled with AFW.

Example

Request

```
POST /api/v1/afw/register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: register.pulseworkspace.net

{
  "reg_key": "12345678",
  "user_email": "user@example.com"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 515

{
  "acc": "2893475",
  "api_url": "https://afw.pulseworkspace.net",
  "client_type": "managed_client",
  "credentials": {
    "algorithm": "HS256",
    "expires": 1556657905,
    "id": "{alphanumeric string}",
    "renew": 1446657905,
    "secret": "{alphanumeric string}",
    "type": "hawk"
  },
  "id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
  "key": "atestpassword",
  "websocket_url": "wss://test.pulseworkspace.net",
  "sandbox_apk_url": "https://s3.amazonaws.com/cld-apps-mobilespaces/sandbox.apk",
  "google_sender_id": "508794832067",
  "afw_enterprise_type": "afw"
}
```

Registration With Bearer Token

Note: When authenticating with SAML, the server returns a temp-token which should be used as the Bearer token for registering for a workspace.
For obtaining a Bearer token after authenticating with SAML, please refer to SAML authentication workflow.

Request

- **Method:** POST
- **Resource:** /api/v1/afw/register
- **Headers:**
  - Authentication: This header should use the Bearer authentication scheme as discussed in RFC-6750 . The token value should be the temp-token provided when performing SAML authentication.

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing an AAfwWorkspaceRegistrationResponseEntity entity.
- **Errors:**
  - 403 - The Bearer token is not found or is otherwise invalid.
  - 400 {code = 40013} - If the system is not enrolled with AFW.

Example

Request

```
POST /api/v1/afw/register HTTP/1.1
Accept: application/json
Authorization: Bearer 89c43087dd8c80579e86bd6dc9c1ea010774a23dd737cbff090797a8a9c24e0c
Host: api.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 515

{
  "acc": "2893475",
  "api_url": "https://api.pulseworkspace.net",
  "client_type": "managed_device",
  "credentials": {
    "algorithm": "HS256",
    "expires": 1556657905,
    "id": "{alphanumeric string}",
    "renew": 1446657905,
    "secret": "{alphanumeric string}",
    "type": "hawk"
  },
  "id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
  "key": "atestpassword",
  "websocket_url": "wss://test.pulseworkspace.net",
  "sandbox_apk_url": "https://s3.amazonaws.com/cld-apps-mobilespaces/sandbox.apk",
  "google_sender_id": "508794832067",
  "afw_enterprise_type": "afw"
}
```

Registration With Session Token

Client needs to get the session token from AAA service before calling this API.
Since the API use a session token cookie, it must be protected from CSRF. CSRF is prevented using CSRF tokens.

Request

- **Method:** POST
- **Resource:** /api/v1/afw/register
- **Cookie:**
  - DSID: A valid session token.
- **Headers:**
  - X-XSRF-TOKEN: CSRF Token

Response

- **JSON Data:** A JSON dictionary representing an AAfwWorkspaceRegistrationResponseEntity entity.
- **Errors:**
  - 403 - If session token is invalid.
  - 400 {code = 40013} - If the system is not enrolled with AFW.

Example

Request

```
POST /api/v1/afw/register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: register.pulseworkspace.net
Cookie: DSID=foobar12345
X-XSRF-TOKEN: <XSRF-TOKEN-COOKIE-VALUE>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 515
Set-Cookie: XSRF-TOKEN=<XSRF-TOKEN-COOKIE-VALUE>

{
  "acc": "2893475",
  "api_url": "https://afw.pulseworkspace.net",
  "client_type": "managed_client",
  "credentials": {
    "algorithm": "HS256",
    "expires": 1556657905,
    "id": "{alphanumeric string}",
    "renew": 1446657905,
    "secret": "{alphanumeric string}",
    "type": "hawk"
  },
  "id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
  "key": "atestpassword",
  "websocket_url": "wss://test.pulseworkspace.net",
  "sandbox_apk_url": "https://s3.amazonaws.com/cld-apps-mobilespaces/sandbox.apk",
  "google_sender_id": "508794832067",
  "afw_enterprise_type": "afw"
}
```

Android For Work

Architecture Overview

To facilitate Android for Work device management, we will create a new afwapi Python application (wrapped in docker image) to communicate with the client. It will not have its own data store and only be able to communicate with the Policy Engine and Auth services pwsapi via REST. It will communicate with mobile devices with REST, and with back-end servers, via REST or the Google API.
Note: the afwapi only respond to **client requests**. Any communication from the backend to the client is done directly from the pws server using GCM notifications. Such notifications include

- notify client to pull latest policy (after a server policy change)
- admin operations on the client: wipe, lock
- debugging (send-logs)

Even though afwapi is a separate "server", all API requests go through the apibridge proxy at a URL such as https://api.pulseworkspace.net. apibridge will handle load balancing and redirecting request to afwapi as needed.

Glossary

- apibridge - Nginx proxy in front of afwapi and pwsapi that handles Hawk authentication and offloads HTTP, websockets, and large file uploads.
- afwapi - New API endpoint for AFW workspace client to talk to
- pwsapi - Existing API endpoint for workspace management
- authapi - Planned API endpoint for internal authentication

Email/Regkey Registration

registration.json

- The client enters his email/regkey in the registration screen in the client app.
- The client uses the first digit of the registration key to determine the server to send the registration code to.
- The server returns registration info, (workspace id, key, etc) - see messages details below.
- the returned id/key are used for authentication of further requests.

NOTE: registration is not AFW-specific: it is the same for workspace and afw. The client should use either the wss or REST URLs depending on client type.

Authentication

Authentication of pwsapi is based on Hawk HMAC authentication . We should do the same thing
for afwapi communication with the mobile client. We can be more relaxed with renewal and expire times for keys to reduce bandwidth usage for mobile devices.
Authentication is based on claims .

- workspace - This claim lets us know the request is coming from a workspace for a particular domain

```
"claim": {
  # The workspace this claim was granted for
  "aud": "workspace:<workspace-id>",
  # Provides workspace level access for specified domain
  "role": "workspace",
  "type": "domain",
  "id": "<domain-id>"
}
```

When a mobile client makes a requests to afwapi, it does it through apibridge. apibridge will handle authenticating the request from the client and will inject a X-Auth-Data-Token header into the request and pass it to afwapi. afwapi will need to pass this header along to pwsapi when making requests as the X-Auth-Data-Token tells pwsapi which database to authenticate to and how.

Authorization Flow

```
title Authorization Flow

Device->ApiBridge: GET /api/v1/req\nAuthorization: Hawk
ApiBridge->AuthAPI: GET /api/v1/auth/token\nAuthorization: Hawk
AuthAPI-->ApiBridge: X-Auth-Data-Token: token
ApiBridge->AfwAPI: GET /api/v1/req\nX-Auth-Data-Token: token
AfwAPI->PwsAPI: GET /api/v1/workspaces/...\nX-Auth-Data-Token: token
PwsAPI-->AfwAPI:
AfwAPI-->ApiBridge:
ApiBridge-->Device:

Device->ApiBridge: GET /api/v1/req2\nAuthorization: Hawk
ApiBridge->ApiBridge: check cache
ApiBridge->AfwAPI: GET /api/v1/req2\nX-Auth-Data-Token: token
AfwAPI->PwsAPI: GET /api/v1/workspaces/...\nX-Auth-Data-Token: token
PwsAPI-->AfwAPI:
AfwAPI-->ApiBridge:
ApiBridge-->Device:
```

Blocked by: SER-753

Policies

The afwapi will need to get policy information from the Policy Engine. Currently the Policy Engine is mixed up in multiple places throughout the cs.core code base. A new API for pwsapi needs to be created that can return a complete, flattened policy. A flattened policy should have the complete set of properties and apps that is required for a workspace. The API can present a way to get an Android view of the policy so that iOS related items are excluded.

Flattened Policy

Policies will need to be modified so that the entire state of the policy can be serialized into JSON. This will allow us to send the entire policy to the client and let the client decide how to satisfy the policy. This means rules that are currently decided by the server need to be conveyed in its entirety to the client.
Note: Flattened policy app rules will only have the complete set of packages required/allowed for that policy. So there is no need for "add" or "remove" rules because this will be determined when flattening the policy so that the end result is just "add" rules.
Application installation logic is described below, in "Policy Processing Rules" section.

Workflow

AFW

```
title AFW
participant "Dpc Client" as Device
participant "Dpc Client\ninside Profile" as pDevice

note left of Device: Registration depends on\n the customer domain\n already being enrolled.

note over Device: Register:\nemail  : [___]\nregkey: [___]\n
Device->S3: GET register.json
note over Device: find reg.server based on regkey

# Registration
Device->AfwAPI: POST /api/v1/afw/register\n(user_email, reg_key)
AfwAPI-->-Device: id,key,api_url,credentials,afw_enterprise_type

opt client_type == sandbox
    note over Device: Switch to sandbox workflow
end


Device->+AfwAPI: PUT /api/v1/afw/spaces/<workspace-id>/device\nclient_type,model,os_number,...
note over AfwAPI: Policy is calculated for Workspace
AfwAPI-->-Device:
Device->+AfwAPI: GET /api/v1/afw/spaces/<workspace-id>/policy
note over AfwAPI:
AfwAPI may parse/modify the policy
before replying to device
end note
AfwAPI-->-Device: Flattened policy
note over Device: Validate compliance\n(against policy/properties)
```

```
opt connect to google account (if not using EMM-managed accounts)
# Google User Creation
Device->AfwAPI: GET /api/v1/afw/spaces/<workspace-id>/google-account
AfwAPI->GoogleAPI: get_account(google_account)
GoogleAPI->AfwAPI: google_account, exists
AfwAPI->Device: { google_account: email, exists: true/false }

opt create google account (if missing)
note over Device: Create google\naccount for email\npassword: [___]
Device->+AfwAPI: POST /api/v1/afw/spaces/<workspace_id>/google-account\n { "password": "something",
"hash_function": "SHA-1" }

AfwAPI->GoogleAPI: Create Google User Account
note right of AfwAPI: What happens if\nthis fails or we dont\n have domain credentials?
GoogleAPI-->AfwAPI:
AfwAPI-->-Device: HTTP 204

note over Device: User logs into Google Account\nto provision workspace


end

end

opt request authentication token if using EMM-managed AfW accounts

Device-->AfwAPI: POST /api/v1/afw/spaces/<workspace_id>/authentication-token
AfwAPI-->Device: authentication_token

note over Device: Token used to authenticate with Google services

end

alt FUTURE: passcode not implemented in this release.

    Note over Device: prompt for passcode
    Device->+AfwAPI: PUT /api/v1/afw/spaces/<workspace-id>/passcode\n(clear-text pwd)
    AfwAPI-->-Device: HTTP 204
else policy/settings contains passcode_set_date
    Device->+AfwAPI: GET /api/v1/afw/spaces/<workspace-id>/passcode
    AfwAPI-->-Device: (hash, length)
end


Note over Device: Provision Workspace,\nEncrypt, reboot if needed

Device->*pDevice: Switch to Profile,\nMigrate policy,\ncredentials
pDevice->GoogleAPI: register with GCM
destroy Device
GoogleAPI-->pDevice: gcm_id
Note over pDevice: Login to GMS\n(google_account from policy/settings)
pDevice->+AfwAPI: PUT /api/v1/afw/spaces/<workspace-id>/device\ngoogle_device_id, gcm_id
AfwAPI-->-pDevice:
alt policy require vpn cert
```

```
pDevice->AfwAPI: POST /api/v1/afw/spaces/<workspace-id>/certificate\ncert-version
Note over AfwAPI: [re]-generate cert\nif version changed.
AfwAPI-->pDevice: cert, cert-version
end
Note over pDevice: Process policy\n(against actual apps)
pDevice->AfwAPI: PUT /api/v1/afw/spaces/<workspace-id>/state
activate AfwAPI
pDevice->pDevice: pkgs with client_enable:\nenableApp\n \n(system apps. no server\ninteraction required)
note over AfwAPI: AFW sends Google commands to\ninstall/remove apps on the device
AfwAPI->GoogleAPI: pkgs with server_install:\ninstall
GoogleAPI-->pDevice: INSTALL_APP
AfwAPI->AfwAPI: Save workspace state
AfwAPI-->-pDevice:

loop for each app installed (system or market) in the profile:
Device->pDevice: PACKAGE_INSTALLED
pDevice->+AfwAPI: PUT /api/v1/afw/<workspace-id>/state
AfwAPI-->-pDevice:
end
```

Policy Processing Rules

The client follows the following logic while processing the policy. as an outcome, it generates the state structure:

- If the app is already in the workspace, then no action is needed (action = no_action)
- If source is system: action = client_enable. **Note:** in the future we might have a screen to hide system apps and thus have a different meaning for add and optional.
- If source is market:
- If command is add or add_optional: action = server_install
- If command is optional: action = server_enable
- Special processing of apps with a group_tag:
  - All apps with the same group_tag belong to the same group.
  - The first non-system app in a group is the default_package.
  - A missing system package with a group_tag is silently ignored.
  - If any system package exists (and thus client_enabled), the action for all market or enterprise packages in the group is "downgraded" to server_enable instead of server_install.
  - The default_package is installed only if none of the system packages of the same group_tag are available.

Notes:

- With the add_optional command, the client should allow uninstallation of the app by the user.
- **Special case**: If an app with source = market is found as a SystemApp on the device (e.g. Chrome), action will show server_install, but the client will also enable it locally (until server pushes market update).

Android App Config Schema

Note: This API differs from the Android Products API so that it mirrors the iOS App Config Schema API.
Note: Android app config schema is *not* versioned by Google.

Get App Config Schema

Get the app config schema for an Android app.

Request

- **Method:** GET
- **Resource:** /api/v1/afw/apps/<package-name>/app-config-schema

Response

- **Status:** 200
- **JSON Data:** The response data will be an AppConfigSchemaEntity.
- **Errors:**
  - 404 Not Found - An app with that package-name could not be found.

Example

Request

```
GET /api/v1/afw/apps/com.cisco.webex/app-config-schema
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "version": "1e93e453ae80459d9a031d24f4f7a2d4",
  "schema":
  [
    {
      "description": "VPN Connection Name (Required)",
      "restriction_type": "string",
      "key": "profile_name",
      "title": "Connection Name"
    },
    {
      "description": "VPN Connection URL (Required)",
      "restriction_type": "string",
      "key": "url",
      "title": "URL"
    },
```

```
  {
    "description": "VPN Action: Create = 0 or Delete = 1 or Set Default = 2",
    "title": "VPN Action",
    "entry_value": ["0", "1", "2"],
    "key": "action",
    "entry": ["create", "delete", "setdefault"],
    "restriction_type": "choice"
  }
 ]
}
```

AFW - Google User Accounts

Get Google Account for workspace

This endpoint is used by the client to found out the Google Account for the workspace, and whether it already exists.

Request

- **Method:** GET
- **Resource:** `/api/v1/afw/spaces//google-account

Response

- **Status:** 200
- **JSON Data**
  - google_account: (**string**) The google account name for this workspace-id. null if the domain uses EMM-managed AfW accounts.
  - exists: (**bool**) A boolean describing whether or not a google user account associated with this workspace email exists. Should always be true if the domain uses EMM-managed AfW accounts.
- **Errors:**
  - 400 {code = 40005} - If the enterprise uses EMM-managed AfW accounts.
  - 404: Not Found - If a workspace with the id <workspace-id> cannot be found.

Example

Request

```
GET /api/v1/afw/239c28c8-4f5a-4c53-9289-f1bea5ba4cdd/google-account/exists HTTP/1.1
Accept: application/json
Host: api-env.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 20

{
  "google_account" : "user@domain.com",
  "exists": true
}
```

Creation Of Google User Account

This endpoint can be used by the client to create a Google User Account for the user_email specified at workspace registration time.

Request

- **Method:** POST
- **Resource:** `/api/v1/afw/spaces//google-account
- **JSON Data**
  - google_account: (**string**) The google account name to create for this workspace-id
  - password (**str**) A hashed password for the google user account. This value is hashed by the client.
  - hash_function (**str**) The hash algorithm used for the password. This key matches the Google definition. See hashFunction here for details on available algorithms and the format of the password: https://developers.google.com/admin-sdk/directory/v1/reference/users
    Example: Set to SHA-1 and provide an sha1 hash as hex as the password.

Response

- **Status:** 204
- **Errors:**
  - 404 Not Found : a workspace with the id <workspace-id> cannot be found.
  - 409 Conflict : the user already exists; their password will not be updated
  - 400 Bad Request : the inputs are invalid, or the creation attempt was rejected by the google servers for some other reason

Example

Request

```
POST /api/v1/afw/239c28c8-4f5a-4c53-9289-f1bea5ba4cdd/google-account HTTP/1.1
Accept: application/json
Host: api-env.pulseworkspace.net

{
  "google_account" : "user@domain.com",
  "hash_function": "SHA-1",
  "password": "ab87d24bdc7452e55738deb5f868e1f16dea5ace"
}
```

Response

```
HTTP/1.1 204 OK
HTTP/1.1 406 Conflict
Content-Type: application/json
Content-Length: 139

{
    "message": "test@example.com already has a Google Apps account",
    "code": 40901,
    "id": "61ec6a95-5669-45e7-bf27-f08641bf34e1"
}
```

PWS based safetyNet Implementation API contract

The purpose of this document is to define the Google SafetyNet implementation for Pulse Workspace. The flow will include standard Google API's with Google Play as well as proprietary API's between PWS client-server. The SafetyNet guide is: [http://developer.android.com/training/safetynet/index.html]

flow

chart diagram:

```
title SafetyNet flow
participant GoogleService

PWS_Client -> PWS_Server : 1. getNonce()
PWS_Server -> PWS_Server : 2. generate nonce for this session
PWS_Server -> PWS_Client : 3. nonce
PWS_Client -> GoogleService : 4.SafetyNet attestation request with nonce
GoogleService -> PWS_Client : 5.attestation Result Response (JWS)
PWS_Client -> PWS_Server : 6. REST request with JWS
PWS_Server  ->  GoogleService : 7. signature validation of JWS
GoogleService  ->  PWS_Server : 8. signature validation result
PWS_Server -> PWS_Server : 9. validate nonce (and Workspace ID match)
PWS_Server -> PWS_Server : 10. validate that ctsProfileMatch is True
PWS_Server -> PWS_Client : 11. REST response
PWS_Client -> PWS_Client : 12. self validation
```

The workflow is described below:

1. The MobileApp requests a nonce from PWS server using the REST request. This nonce is used to prevent replay attacks.
2. The PWS server generates a nonce securely
3. The server returns nonce to MobileApp.
4. The MobileApp performs an attest() request containing this nonce. This API is provided by the Google Play Services SDK, bundled into the application.
5. GoogleServices respond with the attestation result, which is in JSON Web Signature (JWS) format – a base64-encoded and signed JSON object.
6. The MobileApp performs the REST request to the PWS server with the JWS AttestationResult.
7. At this point, the PWS server must first verify that the JWS object is indeed signed by Google by asking Google to verify it on your behalf. This is done by sending the JWS to https://www.googleapis.com/androidcheck/v1/attestations/verify. The response contains the attribute isValidSignature and it should be True if the validation was successful.

8. PWS server further validates that the payload data of the JWS object match the original compatibility check request. The AttestationResult payload is a dictionary, populated by GoogleServices, that includes at least the following:

    - Nonce
    - apkCertificateDigestSha256
    - timestamp
    - apkDigestSha256
    - apkPackageName
    - ctsProfileMatch

    For example:

    {
    "nonce":
"eyJleHBpcmVzIjogMTQ1OTkzMzk0OS4xNTE2NzYsICJzYWx0IjogImQ3YzEyZiIsICJ3b3Jrc3BhY2VfaWQi
OiAiNDAyMDhhZmItMDIzNi00YTBjLWI1MWQtMDYxMmQ4Y2Y2OGEwIn3VM-
zpkzoVb_30J2xBUeyST9XQIA56qClFnarctxfdRQ==",
    "timestampMs": 9860437986543,
    "apkCertificateDigestSha256": ["base64 encoded, SHA-256 hash of the certificate used to sign requesting
app"],
    "apkDigestSha256": "base64 encoded, SHA-256 hash of the app's APK",
    "ctsProfileMatch": true
    }

    - Verify that the received nonce is valid and not too old (timestamp check is implicit here).
    - Verify that the apkPackageName field matches the expected package name of the MobileApp
    - Verify that the ctsProfileMatch field is true.

9. If all checks succeed, An "OK" response to the client app’s REST request is returned. If checks fails, the server knows that this device was tampered with and can take appropriate action as configured in the policy. I suggest that this response from PWS server to Client will be signed by the **PWS** attestation key. this is done to prevent an attack where the device is tempered and the attacker falsifies the server response that everything is OK. the is done with our own PWS public key.

10. The MobileApp checks the response from server and takes the appropriate action accordingly.

APIs

Get Nonce

Request

- **Method:** GET
- **Resource:** /api/v1/afw/spaces/<workspace-id>/device/safetynet

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following key:

  JSON Data:
  {
  "nonce": "<nonce>"
  }

- nonce: string with at least 16 bytes

Example

JSON Request

```
GET /api/v1/afw/spaces/450d84d9-28ed-43a6-877d-1571f878e120/device/nonce HTTP/1.1
Accept: application/json
Host: api-env.api.pulseworkspace.net
Authorization: Hawk mac="wH/XbuM1MxCUhCUD3AmLY8zT7am485ZOZ8vTDzHjs6g=", [snip]...
```

JSON Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 202w
{

"nonce":"eyJleHBpcmVzIjogMTQ1OTkzMzk0OS4xNTE2NzYsICJzYWx0IjogImQ3YzEyZiIsICJ3b3Jrc3BhY2VfaW
QiOiAiNDAyMDhhZmItMDIzNi00YTBjLWI1MWQtMDYxMmQ4Y2Y2OGEwIn3VM-
zpkzoVb_30J2xBUeyST9XQIA56qClFnarctxfdRQ=="
}
```

Send server compatibility check response for verification

Request

- **Method:** POST
- **Resource:** /api/v1/afw/spaces/<workspace-id>/device/safetynet
- **JSON Data:** A dictionary with the key jws and the value of the JWS entity returned by the the attest call of the Google SafetyNet API.

The request contains a Google-generated JWS object that has these entities in the payload:

- nonce: The nonce provided in the request
- apkCertificateDigestSha256: SHA-256 hash of the certificate used to sign the app wrapped in list
- timestampMs: Timestamp (ms resolution)
- apkDigestSha256: SHA-256 hash of the app's APK
- apkPackageName: Name of the package
- ctsProfileMatch: Result of the verification (true or false)

There are additional undocumented fields in the response fields in the response, for example the extension field.


```
{
   "nonce":"1avepxeaJ/jNZThuc6RaOZzzbFfJyOEqfofnJOA6fh8=",
   "timestampMs":1459849913399,
   "apkPackageName":"com.cigital.safetynetplayground",
   "apkDigestSha256":"uKTKZ5sJGPcgKqE831eexJDhJMXRcoD+GPSbPia+vT8=",
   "ctsProfileMatch":true,
   "extension":"CYDWmfZPFCWO",
   "apkCertificateDigestSha256":["5OVuFtjJciXQAHhwZDPdg7QFz7sg7DeDnI6qNho5724="]
}
```


Response

- **Status:** 200
- **JSON Data:** A JSON dictionary with the key:
- attestation_ok and true or false.
- **Errors:**
- 500 Internal Server Error - There was a problem with verifying the JWS

Example

Request

```
POST /api/v1/afw/spaces/450d84d9-28ed-43a6-877d-1571f878e120/device/safetynet HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: foobar.pulseworkspace.net

{
    "jws": "eyJhbGciOiJSUzI1NiIsIng1YyI6WyJNSUlFZmpDQ0EyYWdBd0lCQWdJSVZaeDlNZDVhb[lots of data]"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 189

{
    "attestation_ok": false
}
```

AFW - AfW Account Authentication Token

Request Authentication Token

This endpoint can be used by the DPC to request an authentication token which it can use to authenticate a workspace to an EMM-managed AfW account.
Note that token must be used within a few minutes or it will expire, in which case a new token must be requested.

Request

- **Method:** POST
- **Resource:** `/api/v1/afw/spaces//authentication-token

Response

- **Status:** 200
- **JSON Data**
  - authentication_token: Authentication token for the DPC to use for workspace provisioning.
- **Errors:**
  - 404 Not Found : a workspace with the id <workspace-id> cannot be found.
  - 400 Bad Request { code = 40005 }: the domain does not use EMM-managed AfW accounts.
  - 400 Bad Request : the inputs are invalid, or the request was rejected by the google servers for some other reason

Example

Request

```
POST /api/v1/afw/239c28c8-4f5a-4c53-9289-f1bea5ba4cdd/authentication-token HTTP/1.1
Accept: application/json
Host: api-env.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 42

{ "authentication_token": "AD45HEJAS432" }
```

AFW - Device Info

Update Device Info

Sent to the server whenever essential info is set. Once set, items do not change.

Request

- **Method:** PUT
- **Resource:** /api/v1/afw/spaces/<workspace-id>/device
- **JSON Data:** Request data should be in the form of a JSON dictionary representing a [Device] (../workspaces/entities/core.md#device] entity.

Response

- **Status:** 204
- **Errors:**
  - 202 Not Found - The google device id can't be verified by Google.
  - 202 - The google device is not found by Google AfW service. Server returns status code 202 in the case that a device is still not fully registered in Google play store although we have a valid google_device_id. This is due to a google bug (https://dev.pulsesecure.net/jira/browse/PWS-2097).

Example

Request

```
PUT /api/v1/afw/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/device
Content-Type: application/json
Content-Length: 505

{
  "gcm_id" : "example-gcm-id",
  "google_device_id": "example-google-device-id",
  "client_type: "afw",
  "manufacturer" : "LG",
  "os_version" : "5.1",
  "model" : "LG-G3",
  "imei": "99000344570074",
  "imsi": "310006062122684",
  "carrier": "Sprint-us",
  "fingerprint": "samsung/jfltespr/jfltespr:5.0.1/LRX22C/L720VPUGOH1:user/release-keys",
  "client_versioncode": 414539,
  "client_pub_version": "1.3.3.1542.1",
  "client_version_string": "20151021-171044-r3.1542.1",
  "supported_features": [],
  "owner_mode": 1
}
```

Response

```
HTTP/1.1 204 No Content
```

Retrieving Device Info

Request

- **Method:** GET
- **Resource:** /api/v1/afw/spaces/<workspace-id>/device

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a [Device] (../workspaces/entities.md#device] entity.
- **Errors:**
  - 404 Not Found - The workspace ID is invalid

Example

Request

```
GET /api/v1/afw/spaces/<workspace-id>/device
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "client_type": "afw",
  "google_device_id": "example_google_device_id",
  "gcm_id": "example_gcm_id",
  "policy_id" : "current policy ID published on this space"
}
```

AFW - Domains

Enrolling A Domain

A customer who wants to use Google Apps for their user may enroll an existing Google Apps domain with Pulse as their EMM provider. During domain enrollment, PWS will save the customer's enterprise id as a domain property.
For a information on generating an EMM token for a domain, see:
https://support.google.com/work/android/answer/6095370
For more information about enterprise enrollment, see: https://developers.google.com/android/work/play/emm-api/v1/enterprises#resource
Note that the enterprise_id returned from this is equal to the "customer ID" used by the Google Apps Admin SDK.
On success, this updates these domain properties:

- afw_enterprise_type : "google"
- afw_enterprise_id : ID of the enterprise
- afw_enterprise_primary_domain : the primary domain name of the enterprise

Request

- **Method:** POST
- **Resource:** /api/v1/afw/domains
- **JSON Data:**
  - token: Domain enrollment token

Response

- **Status:** 204
- **Errors:**
  - 40x - Errors will be propagated from Google if the token is invalid or the domain has already been enrolled somewhere.

Example

Request

```
POST /api/v1/afw/domains HTTP/1.1
Accept: application/json
Host: api-env.pulseworkspace.net
Content-Type: application/json
Content-Length: 58

{
  "token": "8843d7f92416211de9ebb963ff4ce28125932878"
}
```

Response

```
HTTP/1.1 204 No Content
```

Un-enroll Google AFW domain

Request

- **Method:** DELETE
- **Resource:** /api/v1/afw/domain
- **Request Data:** No Request data

Response

- **Status:** 204
- **Errors:**
  - 404 - The domain isn't currently enrolled.

Example

Request

```
DELETE /api/v1/afw/domain HTTP/1.1
Host: api-env.pulseworkspace.net
```

Response

```
HTTP/1.1 204 No Content
```

Setup an AfW Accounts Enterprise

A customer who does not wish to use Google Apps can setup an AfW Accounts domain, whose users accounts are managed by the EMM and used for EMM purposes only.
To initiate setup, the console requests a setup URL from the server to redirect the user to in order to do the AfW Accounts enterprise signup.
Note that this URL should only be requested when a user initiates the flow. Requesting this URL initiates the signup process and stores a "completion token" for later use to complete the signup process, under the assumption the user will immediately be redirected to signup. Requesting a URL will probably invalidate any previously requested URL.
On completion of the flow, the enterprise ID of the created enterprise is stored in a domain property and the user is returned to the AfW settings page in the console.

Flow overview

This web sequence diagram shows the overall flow of setting up and enrolling a new AfW Accounts (aka EMM-managed) enterprise.

```
title AFW Accounts Sign Up Flow

Browser->PWS: "POST afw/v1/signup-url\nnext: https://console/.../android-for-work\n"
note over PWS: Generate temp token ID
PWS->"Google": "POST /enterprises/signupUrl\ncallbackUrl: https://api/v1/afw/signup-callback?token_id=...\n"
"Google"->PWS: "completionToken:...\nsignUpUrl:https://google.com/afw-accounts/signUp?...\n"
note over PWS: Save auth data, next and completionToken \nin redis with temp token
PWS->Browser: "url: https://google.com/afw-accounts/signUp?...\n(Same URL returned by Google)\n"
Browser->Google: "GET /afw-accounts/signUp?..."
Google->Browser: "HTML Sign up form"
note over Browser: Admin logs in using a \npersonal Google Account
Browser->Google: "POST Form Data\n(Might be multiple steps)\n"
note over Google: Creates new enterprise
Google->Browser: "301 Redirect\nLocation: https://api/v1/afw/signup-callback?token_id=...&enterpriseToken=...\n"
Browser->PWS: "GET /v1/afw/signup-callback?token_id=...&enterpriseToken=...\n"
PWS->Google: "POST /androidenterprise/v1/enterprises/completeSignup\nenterpriseToken: ...\ncompletionToken:
...\n"
Google->PWS: "id: ...\nname: ...\nadministrators: [{email: ...}]\n"
PWS->Google: "PUT /v1/enterprises/<enterpriseId>/account\naccountEmail: <ESA email>\n"
note over Google: Sets the ESA for the enterprise
Google->PWS: "OK"
note over PWS: Set domain props
PWS->Browser: "301 Redirect\nLocation: https://console/.../android-for-work"
Browser->PWS: "GET /.../android-for-work"
note over Browser: User is back\nin the console
```

Request

- **Method:** POST
- **Resource:** /api/v1/afw/domains/signup-url
- **JSON Data:**
- next: Console URL to return to on completion (e.g. the android-for-work settings page of the current domain)

Response

- **Status:** 200
- **JSON Data:**
- setup_url: URL to redirect the user to in order to sign up a new enterprise
- **Errors:**
- 40x - Errors will be propagated from Google if there is a problem

Example

Request

```
POST /api/v1/afw/domains/signup-url HTTP/1.1
Accept: application/json
Host: api-env.pulseworkspace.net
Content-Type: application/json
Content-Length: 75

{
  "next": "https://test.unity.dev/admin/settings/android-for-work"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 77

{
  "url": "https://google.com/afw-domain-setup?t=123kjlk1j2lk3j123"
}
```

AfW Enterprise Setup Callback

When we request an enterprise setup URL from the Google EMM API, we provide a callback URL to redirect the user to with an enterprise token that we use to complete the sign up process and enroll the new enterprise to our EMM. We can put any query parameters we wish onto this URL to maintain the state of the flow; Google's EMM API will add an additional query parameter enterpriseToken which we may exchange for information about the newly created enterprise.
This request does not use Hawk authentication. Instead, a temp token ID is passed as a query parameter, which is used to authorize the user and also fetch the "completion token" and "next" URL from the initial call to signup-url.
The provided enterprise token and a completion token returned by google when the signup process was initiated are submitted to the [Enterprises.completeSignUp|https://developers.google.com/android/work/play/emm-api/v1/enterprises/completeSignup] API call for validation.
On success, this updates these domain properties:

- afw_enterprise_type : "afw"
- afw_enterprise_id : The ID of the newly created enterprise
- afw_enterprise_primary_domain : Set empty, as AfW domains have no associated domain name
- afw_domain_admin_user : The Google Account email address of the user who did the setup flow

Request

- **Method:** GET
- **Resource:** /api/v1/afw/domains/signup-callback
- **Query Parameters:**
  - enterpriseToken: Token provided by google to complete signup
  - token_id: Temp token ID used to track the console URL to return to and the authentication data of the user who initiated the request.

Response

- **Stat**us: 303
- Error**s:**
  - 40x - Errors will be propagated from Google if there is a problem

Example

Request

```
GET /api/v1/afw/domains/signup-callback?enterpriseToken=abc123def&next=https://test.unity.dev/admin/settings/android-for-work HTTP/1.1
Accept: application/json
Host: api-env.pulseworkspace.net
```

Response

```
HTTP/1.1 303 See Other
Location: https://test.unity.dev/admin/settings/android-for-work
```

Get DPC-specific Token

This token is specific for Device owner mode in the Managed Google Play Accounts provisioning method.
During device setup, the user enters a special DPC-specific token when they are prompted to add an account. A token is in the format “afw#DPC_IDENTIFIER.” For an EMM named ACME, “afw#acme” would install the default DPC of the ACME EMM. Each EMM must request a specific DPC identifier from Google before they can use it in the provisioning process.

Request

- **Method:** GET
- **Resource:** /api/v1/afw/dpc-token

Response

- **Status:** 200
- **JSON Data:**
  - token: The DPC specific token

Example

Request

```
GET /api/v1/afw/dpc-token HTTP/1.1
Accept: application/json
Host: api-env.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 77

{
  "token": "afw#acme"
}
```

Workspace iOS Mobile Apps API

API endpoints to manage iOS mobile apps in the app catalog.
Note: This API mirrors the Android API of the same name, except iOS app bundle upload.

Create Mobile App

Request

- **Method:** POST
- **Resource:** /api/v1/ios/apps
- **JSON Data:** A MobileAppEntity.Update entity.

Response

- **Status:** 200

Example

Request

```
POST /api/v1/ios/apps HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net

{
  "package_name": "com.cisco.webex",
  "title": "Cisco Webex",
  "tags": [
    "productivity"
  ],
  "app_config": {
    "schema": {
      "schema": ...,
      "version": ....
    },
    "values": ...
  },
  "category": "Internet",
  "creator": "Interwebs",
  "version": "1.0.1",
  "is_manual": true,
  "hosting_type": "external",
  "download_url": "https://example.com/app.ipa",
  "is_required": true,
  "network_access": "direct",
  "country_code": "US"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 575

{
  "id": "c54677e8-b0f5-11e6-9ee1-0242c0aa0007",
  "platform": "ios",
  "package_name": "com.cisco.webex",
  "title": "Cisco Webex",
  "tags": [
    "productivity"
  ],
  "app_config": {
    "schema": {
      "schema": ...,
      "version": ....
    },
    "values": ...
  },
  "category": "Internet",
  "creator": "Interwebs",
  "version": "1.0.1",
  "is_manual": true,
  "hosting_type": "external",
  "download_url": "https://example.com/app.ipa",
  "icon_url": "https://s3.amazonaws.com/unitydev-io-icons/ios-com.cisco.webex.png",
  "licenses": "not licensed",
  "vpp_total": 0,
  "vpp_used": 0
  "source": "appstore",
  "is_required": true,
  "network_access": "direct",
  "country_code": "US",
  "modified_on": "2016-11-22T20:53:50",
  "created_on": "2016-11-22T20:53:50",
}
```

Upload App Bundle

Request

- **Method:** POST
- **Resource:** /api/v1/ios/apps/{package-name}/bundle
- **multipart/form-data:** Form data field named version which contains package version, also file contains the ipa bundle file.

Response

- **Status:** 200
  - **JSON Data:** Response will be a MobileAppEntity.UploadResult.

Example

Request

```
POST /api/v1/ios/apps/com.cisco.webex/bundle HTTP/1.1
Content-Length: <length>
Content-Type: multipart/form-data; boundary=723e82b4e41f4d0d9c66db24f035a326
Host: api.pulseone.net

--723e82b4e41f4d0d9c66db24f035a326
Content-Disposition: form-data; name="version"

1.0.1
--723e82b4e41f4d0d9c66db24f035a326
Content-Disposition: form-data; name="file"; filename="com.cisco.webex.1.0.1.ipa"
Content-Type: application/octet-stream

--723e82b4e41f4d0d9c66db24f035a326--
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: <length>

{
  "package_name": "com.cisco.webex",
  "platform": "ios",
  "title": "Cisco Webex",
  "icon_url": "https://s3.amazonaws.com/unitydev-io-icons/ios-com.cisco.webex.png",
  "version": "1.0.1",
  "download_url": "https://api.pulseone.net/api/v1/ios/apps/com.cisco.webex/plist/1.0.1"
  "md5": "636007ed84a3d2b55297e77683c77c91",
}
```

Get Mobile App

Retrieve a Mobile App by package name.

Request

- **Method:** GET
- **Resource:** /api/v1/ios/apps/{package-name}

Response

- **Status:** 200
- **JSON Data:** Response will be a MobileAppEntity.

Example

Request

```
GET /api/v1/ios/apps/com.cisco.webex /1.1
Accept: application/json
Host: customer.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 575

{
  "id": "c54677e8-b0f5-11e6-9ee1-0242c0aa0007",
  "platform": "ios",
  "package_name": "com.cisco.webex",
  "title": "Cisco Webex",
  "tags": [
    "productivity"
  ],
  "app_config": {
    "schema": {
      "schema": ...,
      "version": ....
    },
    "values": ...
  },
  "category": "Internet",
  "creator": "Interwebs",
  "version": "1.0.1",
  "is_manual": true,
  "hosting_type": "external",
  "download_url": "https://example.com/app.ipa",
  "icon_url": "https://s3.amazonaws.com/unitydev-io-icons/ios-com.cisco.webex.png",
  "licenses": "not licensed",
  "vpp_total": 0,
  "vpp_used": 0
  "source": "appstore",
  "is_required": true,
  "network_access": "direct",
  "country_code": "US",
  "modified_on": "2016-11-22T20:53:50",
  "created_on": "2016-11-22T20:53:50",
}
```

Update Mobile App

Request

- **Method:** PUT
- **Resource:** /api/v1/ios/apps/{package-id}
- **JSON Data:** A MobileAppEntity.Update object.

Response

- **Status:** 204

Example

Request

```
PUT /api/v1/ios/apps/com.sec.android.cloudagent HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net

{
  "package_name": "com.cisco.webex",
  "title": "Super Awesome Cisco Webex",
  "tags": [
    "meetings"
  ],
  "app_config": {
    "schema": {
      "schema": ...,
      "version": ....
    },
    "values": ...
  },
  "category": "Intranet",
  "creator": "Interwebs",
  "version": "1.0.1",
  "is_manual": true,
  "hosting_type": "external",
  "download_url": "https://apps.example.com/app.ipa",
  "is_required": false,
  "network_access": "direct",
  "country_code": "US"
}
```

Response

```
HTTP/1.1 204 No Content
```

Delete Mobile App

An endpoint for deleting mobile apps from the app catalog.
The server will reject the request if the mobile app belongs to any app rules on any policies. The server will return a list of policy entities with id and name. The name is suitable for displaying a helpful list to the IT Admin so that they know what policies must be edited to remove the app rule for the mobile app being deleted. The id is useful for linking directly to the policy.

Request

- **Method:** delete
- **Resource:** /api/v1/ios/apps/{package-id}

Response

- **Status:** 204 - The mobile app has been deleted.
- **Status:** 400 - The mobile app belongs to policies and cannot be deleted.
- **JSON Data:** Response will be a dictionary with a key of items that will contain a list of PolicyEntity. The policy items will contain id and name.

Example

Request

```
DELETE /api/v1/ios/apps/com.cisco.webex HTTP/1.1
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 204 No Content
```

OR

```
HTTP/1.1 400 Bad Request

{
  "items": [
    {
      "id": 1,
      "name": "Example Policy"
    }
  ]
}
```

List Apps

In order to list all apps for this platform, please use the (Workspaces Mobile Apps Search API)[/api/unity/workspaces/mobile-apps.md] and specify the platform.

Searching For Mobile apps

Use Apple's iTunes API to find iOS products.

Request

- **Method:** GET
- **Resource:** /api/v1/ios/apps/search
- **Parameters:**
- q: *required* Search keyword.
- country_code: *optional* Two-letter country code. Default is "US".
- start: Number of results to skip (default 0)
- limit: Maximum number of results to return (default 10, maximum 100)

Response

- **Status:** 200
- **JSON Data:**
  - total (int): Total number of results available
  - items : A list of MobileAppSearchResult.
- **Errors:**
  - 50201: raised when the iTunes API is not returning expected results. This is expected to be temporary.

Example

```
GET /api/v1/ios/apps/search?q=WebEx
Host: api.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "total": 2910
  "items": [
    {
      "title": "Cisco WebEx Meetings",
      "package_name": "com.cisco.webex.meetings",
      "image": "https://lh3.ggpht.com/MmfNO-
OC2YuTrxTYhhPypVgxPIPLe1uDAW5HlcUWbvawzIAevWqqa2_dMOgjWAJTuy8=w300",
      "url": "https://itunes.apple.com/store/apps/details?id=com.cisco.webex.meetings",
      "description": "Take your Web meetings anywhere!  Join any web conference...",
      "creator": "Cisco",
      "category": "Productivity",
      "source": "appstore"
    },
    ...
  ]
}
```

iOS Enrollment APIs

Get enrollment URL

Clients call this endpoint to get the enrollment URL associated with the hawk credentials of a workspace. The clients will open Safari with the enrollment URL. As hawk credentials cannot be shared between Safari and the client application, temporary token authentication is used. The call to the endpoint creates a temporary token associated with the workspace id and adds it as a query param to the enrollment-url.

Request

- **METHOD:** POST
- **RESOURCE:** /api/v1/ios/enrollment-url

Response

- **Status:** 200
- **JSON Data:** A json dictionary with key as enrollment_url and URL as the value
- **Errors:**
  - 401 Unauthorized - When the hawk credentials don't belong to a workspace

Example

Request

```
POST /api/v1/ios/enrollment-url
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "enrollment_url": "http://example.com/api/v1/ios/enroll?t=aS4wmtaqksp"
}
```

GET Profile Service payload

Safari browser calls this endpoint to get the profile service payload, it is **not Hawk authenticated**. The temporary token generated in /api/v1/ios/enrollment-url is used for authentication. It returns a Profile Service plist with the URL for subsequent call and a list of device attributes to send in the subsequent call.

Request

- **METHOD:** GET
- **RESOURCE:** /api/v1/ios/enroll
- **Query string parameters:**
  - t: (str) The id of the temporary token.

Response

- **STATUS:** 200
- **Content-Type:** application/x-apple-aspen-config
- **Body:** signed pkcs7 profile service payload
- **Errors:**
  - 400 Invalid token

Example

Request

```
GET /api/v1/ios/enroll?t=<token_id>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/x-apple-aspen-config
Body: (bytes) signed profile service payload
```

Certificate Configuration Payload / MDM Configuration Payload

End point is called by ios MDM agent twice. This call is **not hawk authenticated**. The response of this call depends on the workspace state. Signed Certificate configuration payload is returned when workspace state is SPACE_UNREGISTERED, MDM configuration payload when it is STATE_ENROLLING

Request

- **METHOD:** POST
- **RESOURCE:** /api/v1/ios/profile
- **Query string parameters:**
- t: (str) The id of the temporary token.

Response

- **STATUS:** 200
- **Content-Type:** application/x-apple-aspen-config
- **Body:** (bytes)
  - Workspace state is STATE_UNREGISTERED: signed Certificate Configuration Payload
  - Workspace state is STATE_ENROLLING: signed MDM Configuration Payload
- **Errors:**
  - 400 Missing parameter: 't'
  - 50002 iOS MDM push certificate not configured
  - 403 Invalid value for parameter 't'
  - 415 Unsupported media type

Example

Request

```
POST /api/v1/ios/profile?t=<token_id>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/x-apple-aspen-config
Body: (bytes) first call: signed Configuration Payload with device certificates
              second call: signed Configuration Payload with MDM profile
```

Un-enroll an iOS device

Client calls this endpoint to un-enroll the iOS device.

Request

- **Method:** POST
- **Resource:** /api/v1/ios/spaces/<workspace-id>/unenroll

Request Parameters

- workspace_id: (*UUID*) The space id.

Response

- **Status:** 204
- **Errors:**
  - 404 - The workspace is not found.
  - 401 Unauthorized - When the hawk credentials don't belong to a workspace

Example

Request

```
POST /api/v1/ios/spaces/9ed4cea1-2f7b-4832-b5c1-82505548807e/unenroll HTTP/1.1
```

Response

```
HTTP/1.1 204 No Content

123 com.myCompany.myApp true 20 25 15 20 25 30 1.50 20.5 20.0 20.1 20.2 20.3 20.4 20.5 blue blue red green
2048 1024 2048 4096 1.5 1.2 1.3 1.4 1.5 2015-07-21T16:29:30Z 1
Critial Settings Enabled Click this checkbox to enable.
Connection Timeout The connection timeout in seconds.
Email Address Please enter the user's email address.
Start Date Please enter a start date.
```

iOS - Certificate SCEP configurations API

Get SCEP configurations needed to create a SCEP certificate request.

When policy indicates manged iOS client needs to use SCEP to request VPN certificate from external PKI server, client will get SCEP related configurations from PWS server. Important components of the SCEP configurations includes external SCEP server URL and challenge. Response from server will also include other information for example Subject Names which client can use when creating certificate requests.

Reqeust

- **Method:** GET
- **Resource:** /api/v1/ios/spaces/{workspace-id}/scep-configuration

Response

- **Status:** 200 OK
- **JSON Data:** A JSON encoded CertificateScepRequestEntity

Example

Request

```
GET /api/v1/ios/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/scep-configuration HTTP/1.1
Accept: application/json
Host: api-env.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 206

{
  "ca_name": "Default_CA",
  "challenge": "dummy_scep_challenge",
  "key_size": 2048,
  "key_type": "RSA",
  "key_usage": 5,
  "scep_url": "http://scep.example.com",
  "subject_cn": "user_1",
  "subject_o": "domain_1.example.com",
  "subject_email": "user1@domain_1.example.com"
}
```

iOS - VPN Certificate API

Update installed version of VPN key pair and certificate if necessary

When the policy specifies that the vpn requires private key authentication, the client will check whether it needs a new key pair and certificate by submitting the version it received last time it got one, or an empty string if it has never downloaded a key and certificate before.
The server compares the version provided by the client with a version it calculates. If they differ, it generates a new private key and certificate and sends it to the client. Otherwise, it will return an empty response.
The key pair and certificate are serialized using pkcs12 format and then encoded into base64 into the certificate field of the response.

Request

- **Method:** POST
- **Resource:** /api/v1/ios/spaces/{workspace-id}/vpn-certificate
- **JSON Data:** A JSON encoded VpnCertificateEntity.Update

Response (Key pair out of date)

- **Status:** 200 OK
- **JSON Data:** A JSON encoded VpnCertificateEntity

Response (Key pair up to date)

- **Status:** 204 No Content

Example

Request

```
POST /api/v1/ios/spaces/1d89e147-a845-4825-93bf-154592454c25/vpn-certificate HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net

{
  "version": "2a20455d639e31f282a535d6e43b4876c37a4eb2e79d111457c4acdf75be4e95"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 128

{
  "certificate": "05C6i6wC7MYTHVOZ3ciV0ukfE0DUFYdxK15wnCawLwMLX62kf01AYXLCAlyg15uX",
  "cert_alias": "pulsesecure.vpn.cert",
  "format": "pkcs12",
  "password": "apassword",
  "version": "481a9340edbcd23a222685466d8b5a4940d83a1281b92df8bf951cfc870b5369"
}
```

iOS - Workspace Registration

Workspace registration for iOS devices can happen with either PIN or an Authentication header using the Bearer scheme. This document does not cover obtaining a PIN or Bearer token.

Registration With PIN

Exchange a PIN and user email for workspace UUID and credentials.
The user email and/or reg_key pair are not found in our database of pending registrations, either because they are entered incorrectly, made up, or already used.

Request

- **Method:** POST
- **Resource:** /api/v1/ios/register
- **JSON Data:** A JSON dictionary representing a WorkspaceRegistrationRequestEntity entity.

Response

- **JSON Data:** A JSON dictionary representing a IosWorkspaceRegistrationResponseEntity entity.
- **Errors:**
  - 403 - The user email and/or reg_key pair are not found in our database of pending registrations, either because they are entered incorrectly, made up, or already used.
  - 50002 - The MDM push certificate is not configured for the domain

Example

Request

```
POST /api/v1/ios/register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: api.pulseworkspace.net

{
  "reg_key": "12345678",
  "user_email": "user@example.com"
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 189

{
  "api_url": "https://api.pulseworkspace.net",
  "client_type": "managed_device",
  "credentials": {
    "algorithm": "HS256",
    "expires": 1556657905,
    "id": "{alphanumeric string}",
    "renew": 1446657905,
    "secret": "{alphanumeric string}",
    "type": "hawk"
  },
  "id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
  "key": "atestpassword",
}
```

Registration With Bearer Token

Note: When authenticating with SAML, the server returns a temp-token which should be used as the Bearer token for registering for a workspace.
For obtaining a Bearer token after authenticating with SAML, please refer to SAML authentication workflow.

Request

- **Method:** POST
- **Resource:** /api/v1/ios/register
- **Headers:**
  - Authentication: This header should use the Bearer authentication scheme as discussed in RFC-6750 . The token value should be the temp-token provided when performing SAML authentication.

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a IosWorkspaceRegistrationResponseEntity entity.
- **Errors:**
  - 403 - The Bearer token is not found or is otherwise invalid.
  - 50002 - The MDM push certificate is not configured for the domain

Example

Request

```
POST /api/v1/ios/register HTTP/1.1
Accept: application/json
Authorization: Bearer 89c43087dd8c80579e86bd6dc9c1ea010774a23dd737cbff090797a8a9c24e0c
Host: api.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 189

{
  "api_url": "https://api.pulseworkspace.net",
  "client_type": "managed_client",
  "credentials": {
    "algorithm": "HS256",
    "expires": 1556657905,
    "id": "{alphanumeric string}",
    "renew": 1446657905,
    "secret": "{alphanumeric string}",
    "type": "hawk"
  },
  "id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
  "key": "atestpassword"
}
```

Registration With Session Token

Client needs to get the session token from AAA service before calling this API.
Since the API use a session token cookie, it must be protected from CSRF. CSRF is prevented using CSRF tokens.

Request

- **Method:** POST
- **Resource:** /api/v1/ios/register
- **Cookie:**
  - DSID: A valid session token.
- **Headers:**
  - X-XSRF-TOKEN: CSRF Token

Response

- **JSON Data:** A JSON dictionary representing a IosWorkspaceRegistrationResponseEntity entity.
- **Errors:**
  - 403 - If session token is invalid.
  - 50002 - The MDM push certificate is not configured for the domain.

Example

Request

```
POST /api/v1/ios/register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: register.pulseworkspace.net
Cookie: DSID=foobar12345
X-XSRF-TOKEN: <XSRF-TOKEN-COOKIE-VALUE>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 515
Set-Cookie: XSRF-TOKEN=<XSRF-TOKEN-COOKIE-VALUE>

{
  "api_url": "https://api.pulseworkspace.net",
  "client_type": "managed_client",
  "credentials": {
    "algorithm": "HS256",
    "expires": 1556657905,
    "id": "{alphanumeric string}",
    "renew": 1446657905,
    "secret": "{alphanumeric string}",
    "type": "hawk"
  },
  "id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
  "key": "atestpassword"
}
```

iOS App Config Schema

Create App Config Schema

This API allows an IT Admin to provide an updated version of the app config schema XML for an app, thereby updating the app's app config schema. This differs from the Android App Config Schema API because Android app configs are provided programmatically by Google, so there is no need for IT Admins to upload the app config schema.

Request

- **Method:** POST
- **Resource:** /api/v1/ios/apps/<package-name>/app-config-schema
- **Body:** An XML file containing app config schema.

Response

- **Status:** 200
- **JSON Data:** The response data will be an AppConfigSchemaEntity.
- **Errors:**
  - 404 Not Found - An app with that package-name could not be found.

Example

Request

Shortened for conciseness.

```
POST /api/v1/ios/apps/com.cisco.webex/app-config-schema HTTP/1.1
Content-Length: 568
Content-Type: multipart/form-data; boundary=723e82b4e41f4d0d9c66db24f035a326
Host: api.pulseone.net

--723e82b4e41f4d0d9c66db24f035a326
Content-Disposition: form-data; name="app-config-schema"; filename="com-cisco-webex.xml"

<?xml version="1.0" encoding="UTF-8"?>

<managedAppConfiguration>
  <version>123</version>
  <bundleId>com.cisco.webex</bundleId>
  <dict>
  ...
</managedAppConfiguration>
```

Response

Shortened for conciseness.

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "version": "6c512339f7fd40ddb10108d6988b9591",
  "schema": [
    {
      "managedAppConfiguration": {
        "bundleId": "com.myCompany.myApp",
        "dict": {
          "boolean": {
            "@keyName": "boolkey1",
            "defaultValue": {
              "value": "true"
            }
            ...
          }
        ...
      }
      ...
    }
  ]
}
```

Apple Device Enrollment Profile

Download Certificate

PWS server generates the certificate which is used to create Apple DEP account

Request

- **Method:** GET
- **Resource:** /api/v1/ios/dep/certificate

Response

- **Status:** 200
- **Body:** A PEM encoded certificate file.

Example

Request

```
GET /api/v1/ios/dep/certificate HTTP/1.1
```

Response

Shortened for conciseness.

```
HTTP/1.1 200 OK
Content-Length: 568
Content-Type: multipart/form-data; boundary=723e82b4e41f4d0d9c66db24f035a326
Host: api.pulseone.net

--723e82b4e41f4d0d9c66db24f035a326
Content-Disposition: form-data; name="certificate"; filename="certificate.pem"

-----BEGIN CERTIFICATE-----
MIIC4TCCAcmgAwIBAgIRANq0pGZ5eUtDufQcAe0hz2swDQYJKoZIhvcNAQELBQAw
...
-----END CERTIFICATE-----
```

Upload Apple Server Token

Upload the S/MIME encrypted Server Token which is from the Apple DEP web portal

Request

- **Method:** POST
- **Resource:** /api/v1/ios/dep/server-token
- **Body:** A S/MIME encrypted server token.

Response

- **Status:** 204
- **Errors:**
  - 400 Bad Request - If the server token is invalid.

Example

Request

Shortened for conciseness.

```
POST /api/v1/ios/dep/server-token HTTP/1.1
Content-Length: 568
Content-Type: multipart/form-data; boundary=723e82b4e41f4d0d9c66db24f035a326
Host: api.pulseone.net

--723e82b4e41f4d0d9c66db24f035a326
Content-Disposition: form-data; name="server-token"; filename="server_token.p7m"

Content-Type: application/pkcs7-mime; name="smime.p7m"; smime-type=enveloped-data
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7m"
Content-Description: S/MIME Encrypted Message

MIAGCSqGSIb3DQEHA6CAMIACAQAxggFdMIIBWQIBADBBMCwxFDASBgNVBAoMC1B1bHNlU2VjdXJl
...
```

Response

```
HTTP/1.1 204 No Content
```

Retrieve Apple Device Enrolment Program account info

Request

- **Method:** GET
- **Resource:** /api/v1/ios/dep/account

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a [DepAccountEntity] (../entities.md] entity.
- **Errors:**
  - 404 Not Found - The DEP account does not exist.

Example

Request

```
GET /api/v1/ios/dep/account
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "97577199-bd17-4102-8ab1-11008e8cb21b",
  "server_name": "Pulse Secure MDM",
  "org_name": "Pulse Secure",
  "org_email": "pws@pulsesecure.net"
  "token_expiry_on": "2023-11-20T02:53:09",
  "created_on": "2015-11-20T02:53:09"
}
```

Sync Apple Device Enrollment Program account info with Apple DEP portal

Request

- **Method:** POST
- **Resource:** /api/v1/ios/dep/account/sync

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a [DepAccountEntity] (../entities.md] entity.
- **Errors:**
  - 404 Not Found - The DEP account does not exist.

Example

Request

```
POST /api/v1/ios/dep/account/sync
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "97577199-bd17-4102-8ab1-11008e8cb21b",
  "server_name": "Pulse Secure MDM",
  "org_name": "Pulse Secure",
  "org_email": "pws@pulsesecure.net"
  "token_expiry_on": "2023-11-20T02:53:09",
  "created_on": "2015-11-20T02:53:09"
}
```

Delete an Apple Device Enrollment Program account

Request

- **Method:** DELETE
- **Resource:** /api/v1/ios/dep/account

A body is not required for deleting an account.

Response

- **Status:** 204

Example

Request

```
DELETE /api/v1/ios/dep/account HTTP/1.1
Accept: application/json
Host: customer.pulseworkspace.net
```

Response

```
HTTP/1.1 204 No Content
```

Update Apple Device Enrollment Program profile

Request

- **Method:** POST
- **Resource:** /api/v1/ios/dep/profile
- **JSON Data:** Request data should be in the form of a JSON dictionary representing a [DepProfileEntity] (../entities.md] entity.

Response

- **Status:** 204

Example

Request

```
POST /api/v1/ios/dep/profile
Content-Type: application/json
Content-Length: 505

{
  "department": "engineering",
  "support_phone_number": "123-456-7890",
  "is_supervised": true,
  "is_mdm_removable": true,
  "setup_items": {
    "passcode": true,
    "location": true,
    "restore": false,
    "apple_id": true,
    "tos": true,
    "biometric": true,
    "payment": false,
    "zoom": true,
    "siri": true,
    "diagnostics": false
  }
}
```

Response

```
HTTP/1.1 204 No Content
```

Retrieve Apple Device Enrolment Program profile

Request

- **Method:** GET
- **Resource:** /api/v1/ios/dep/profile

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a [DepProfileEntity] (../entities.md] entity.
- **Errors:**
  - 404 Not Found - The DEP profile does not exist.

Example

Request

```
GET /api/v1/ios/dep/profile
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 505

{
  "department": "engineering",
  "support_phone_number": "123-456-7890",
  "is_supervised": true,
  "is_mdm_removable": true,
  "setup_items": {
    "passcode": true,
    "location": true,
    "restore": false,
    "apple_id": true,
    "tos": true,
    "biometric": true,
    "payment": false,
    "zoom": true,
    "siri": true,
    "diagnostics": false
  }
}
```

Get Realms

Returns the realms defined in PCS. The API is available only when using SAML authentication.

Request

- **Method:** GET
- **Resource:** /api/v1/ios/dep/realms

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary with the following keys:
  - items: (*list*) A list of DepRealmEntitys.
- **Errors:**
  - 400 Bad Request - If using SAML authentication is disabled.

Example

Request

```
GET /api/v1/ios/dep/realms
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 505

{
  "items": [
    {
      "realm": "CertRealm+Local",
      "selected": False
    },
    {
      "realm": "PWS-User",
      "selected": False
    },
    {
      "realm": "System Local",
      "selected": True
    },
    {
      "realm": "LDAP Realm",
      "selected": False
    }
  ]
}
```

Update Realm used to authenticate users in PCS using SAML

Request

- **Method:** PUT
- **Resource:** /api/v1/ios/dep/realm
- **JSON Data:** Request data should be in the form of a JSON dictionary with the following keys.
  - realm: (*str*) The name of a realm.

Response

- **Status:** 204

Example

Request

```
PUT /api/v1/ios/dep/realm
Content-Type: application/json
Content-Length: 505

{
  "realm": "System Local"
}
```

Get PCS Sign-In URL

Request

- **Method:** GET
- **Resource:** /api/v1/ios/dep/pcs-signin-url

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary with the following keys:
  - url: (*str*) The PCS sign in URL
- **Errors:**
  - 400 Bad Request - If using PIN registration is enabled.

Example

Request

```
http
GET /api/v1/ios/dep/pcs-signin-url
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 50

{
  "url": "https://se3.acmegizmo.com/pcs-signin"
}
```

Update PCS Sign-in URL

Request

- **Method:** PUT
- **Resource:** /api/v1/ios/dep/pcs-signin-url
- **JSON Data:** Request data should be in the form of a JSON dictionary with the following keys.
  - url: (*str*) The PCS sign in URL.

Response

- **Status:** 204

Example

Request

```
http
PUT /api/v1/ios/dep/realm
Content-Type: application/json
Content-Length: 50

{
  "url": "https://se3.acmegizmo.com/pcs-signin"
}
```

Response

```
HTTP/1.1 204 No Content
```

Registration Workspace of DEP device

Register the PulseSecure unified client which was installed during DEP device enrollment. The API returns the HAWK credentials.

Request

- **Method:** POST
- **Resource:** /api/v1/ios/dep/register
- **Query Parameters:**
  - t: (*str*) The registration token generated by PWS server.

Response

- **JSON Data:** A JSON dictionary representing a IosWorkspaceRegistrationResponseEntity entity.
- **Errors:**
  - 400 - The token is missing in the query parameter.
  - 403 - The token is invalid.

Example

Request

```
POST /api/v1/ios/dep/register?t=123488ebcdd8fb0c4623a66b35ce39a958a3 HTTP/1.1
Accept: application/json
Host: api.pulseworkspace.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 189

{
  "api_url": "https://api.pulseworkspace.net",
  "credentials": {
    "algorithm": "HS256",
    "expires": 1556657905,
    "id": "{alphanumeric string}",
    "renew": 1446657905,
    "secret": "{alphanumeric string}",
    "type": "hawk"
  },
  "id": "f4287ae0-63b7-4c3c-b7e1-d3b3421ace0b",
  "key": "atestpassword"
}
```

{ "managedAppConfiguration": { "bundleId": "com.myCompany.myApp", "dict": { "boolean": { "@keyName": "boolkey1", "defaultValue": { "value": "true" } }, "date": { "@keyName": "datekey1", "defaultValue": { "value": "2015-07-21T16:29:30Z" } }, "float": [{ "@keyName": "floatkey1", "constraint": { "@max": "2", "@min": "1" }, "defaultValue": { "value": "1.50" } }, { "@keyName": "floatkey2", "constraint": { "values": { "value": ["20.0", "20.1", "20.2", "20.3", "20.4", "20.5"] } }, "defaultValue": { "value": "20.5" } }], "floatArray": { "@keyName": "floatarraykey1", "constraint": { "values": { "value": ["1.2", "1.3", "1.4", "1.5"] } }, "defaultValue": { "value": "1.5" } }, "integer": [{ "@keyName": "intkey1", "constraint": { "@max": "100", "@min": "1", "@nullable": "true" }, "defaultValue": { "value": "20" } }, { "@keyName": "intkey2", "constraint": { "values": { "value": ["15", "20", "25", "30"] } }, "defaultValue": { "value": "25" } }, { "@keyName": "hiddenkey1", "defaultValue": { "value": "1" } }], "integerArray": { "@keyName": "intarraykey1", "constraint": { "values": { "value": ["1024", "2048", "4096"] } }, "defaultValue": { "value": "2048" } }, "string": [{ "@keyName": "strkey1", "constraint": { "@nullable": "true", "@pattern": "CN=.*" }, "defaultValue": { "userVariable": { "@value": "cn" } } }, { "@keyName": "strkey2", "constraint": { "values": { "value": ["blue", "red", "green"] } }, "defaultValue": { "value": "blue" } }], "stringArray": { "@keyName": "strarraykey1", "constraint": { "@pattern": "[A-Za-z]@company.com" }, "defaultValue": { "userVariable": { "@value": "emailAddress" } } } }, "presentation": { "@defaultLocale": "en-US", "field": [{ "@keyName": "strarraykey1", "@type": "input", "description": { "language": { "#text": "Please enter the user's email address.", "@value": "en-US" } }, "label": { "language": { "#text": "Email Address", "@value": "en-US" } } }, { "@keyName": "datekey1", "@type": "datetime", "description": { "language": { "#text": "Please enter a start date.", "@value": "en-US" } }, "label": { "language": { "#text": "Start Date", "@value": "en-US" } } }, { "@keyName": "hiddenkey1", "@type": "hidden" }], "fieldGroup": { "field": [{ "@keyName": "boolkey1", "@type": "checkbox", "description": { "language": { "#text": "Click this checkbox to enable.", "@value": "en-US" } }, "label": { "language": { "#text": "Enabled", "@value": "en-US" } } }, { "@keyName": "intkey2", "@type": "select", "description": { "language": { "#text": "The connection timeout in seconds.", "@value": "en-US" } }, "label": { "language": { "#text": "Connection Timeout", "@value": "en-US" } }, "options": { "option": [{ "@value": "15", "language": { "#text": "15", "@value": "en-US" } }, { "@value": "20", "language": { "#text": "20", "@value": "en-US" } }, { "@value": "25", "language": { "#text": "25", "@value": "en-US" } }, { "@value": "30", "language": { "#text": "30", "@value": "en-US" } }] } }], "name": { "language": { "#text": "Critial Settings", "@value": "en-US" } } } }, "version": "123" } }

iOS - Policies

Retrieving Workspace Policy

Request the policy for an iOS device.

Request

- **Method:** GET
- **Resource:** /api/v1/ios/spaces/<workspace-id>/policy

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a WorkspacePolicyEntity entity.

Example

Request

```
GET /api/v1/ios/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/policy HTTP/1.1
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1331

{
   "policy_version": "123123",
   "settings": {
      "server_name": "demo.pulsesecure.net",
      "user_name": "flast",
      "google_account" : "username@democorp.com",
      "location_enabled": true
   },
   "properties": {
      "usb_debugging": "allow",
      "vpn_enabled": true,
      "vpn_host": "https://workspace.acmegizmo.com/cert/",
      "vpn_use_scep_certificate": true,
      "vpn_force_update_scep_certificate": true,
      "wifi_use_scep_certificate": true,
      "wifi_force_update_scep_certificate": true,
      "activesync_use_scep_certificate": true,
      "activesync_force_update_scep_certificate": true
   },
```

```
  "app_rules": [
     {
        "package_name": "com.enterprise.myapp",
        "command": "add",
        "network_access": "direct",
        "group_tags": [ "browser" ],
        "source": "enterprise",
        "url": "http://myserver.com/download/myapp.ipa",
        "icon_url": "https://s3.amazonaws.com/icons/myapp.png",
        "version_code": 1234,
        "title": "My App"
     },
     {
        "package_name": "net.pulsesecure.pulsesecure",
        "network_access": "direct",
        "source": "market",
        "title": "PulseSecure VPN",
     }
  ],
  "email_configuration": {
     "schema_version": 1,
     "pkg_name": "csapp.com.android.email",
     "activesync_device_id": "c4d67b1147b3ff1e7d40d643e76584a7",
     "configuration_hash": "2d33919ea4d6fe900f2d83653c0e8a42",
     "display_name": "test_user",
     "email_address": "test@mobilespaces.com",
     "security_type": "ssl",
     "security_accept_all_certs": true,
     "server_address": "exch.mobilespaces.net",
     "server_port": 443,
     "service_type": "eas",
     "user_name": "testuser"
  },
  "vpn_ondemand": {
     "enabled": true,
     "items": [
        {
           "id": "eb493232-be43-466b-bf14-af0f26c4e876",
           "name": "My Cool Rule",
           "seq": 1,
           "description": "list of cool domains",
           "criteria": {
              "criteria_type": "DNSDomainMatch",
              "value": [
                 "example.com",
              ],
           },
           "action": {
              "value": "EvaluateConnection",
              "action_parameters": {
                 "domains": [
                    "example.net"
                 ],
```

```
                "domain_action": "ConnectIfNeeded",
                "required_dns_servers": [],
                "required_url_probe": ""
            }
        }
    }
  ]
 }
}
```


Retrieving Workspace policy settings

Request workspace policy settings.

Request

- **Method:** GET
- **Resource:** /api/v1/ios/spaces/<workspace-id>/policy/settings

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a PolicySettingsEntity entity.

Example

Request


```
GET /api/v1/ios/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/policy/settings HTTP/1.1
Accept: application/json
Host: api.pulseone.net
```


Response


```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 265

{
   "server_name": "demo.pulsesecure.net",
   "user_name": "flast",
   "google_account" : "username@democorp.com",
   "location_enabled": true,
   "is_locked": false,
   "is_wiped": false,
   "full_device_wipe_allowed": true,
   "location_enabled": true
}
```

Retrieving Workspace VPN OnDemand configuration.

Request workspace VPN OnDemand configuration.

Request

- **Method:** GET
- **Resource:** /api/v1/ios/spaces/<workspace-id>/policy/vpn-ondemand

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary representing a VpnOnDemandConfigurationEntity entity.

Example

Request

```
GET /api/v1/ios/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/policy/vpn-ondemand HTTP/1.1
Accept: application/json
Host: api.pulseone.net
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 265

{
   "enabled": true,
   "items": [
      {
         "id": "eb493232-be43-466b-bf14-af0f26c4e876",
         "name": "My Cool Rule",
         "seq": 1,
         "description": "list of cool domains",
         "criteria": [
            {
               "criteria_type": "DNSDomainMatch",
               "value": [
                  "example.net",
                  "example.com"
               ]
            },
            {
               "criteria_type": "DNSServerAddressMatch",
               "value": [
                  "216.3.128.12",
                  "216.3.128.13"
               ]
            }
         ],
```

```
            "action": {
                "value": "EvaluateConnection",
                "action_parameters": {
                    "domains": [
                        "example.net"
                    ],
                    "domain_action": "ConnectIfNeeded",
                    "required_dns_servers": [],
                    "required_url_probe": ""
                }
            }
        }
    ]
}
```

iOS - iOS MDM Lost Mode

iOS MDM Protocol provides three commands — EnableLostMode, DisableLostMode, and DeviceLocation — to let the MDM server help locate supervised devices when they are lost or stolen. A fourth command, PlayLostModeSound, plays a loud sound on the lost device. These commands may be used only in supervised mode. The first three commands are available in iOS 9.3 and later and the fourth in iOS 10.3.
To put a device in supervised mode, customer could chose to use Apple Deployment Program, Configurator 2, or Apple School Manager to set up the devices.
Reference: https://developer.apple.com/enterprise/documentation/MDM-Protocol-Reference.pdf

Enable/disable iOS MDM Lost Mode

Put iOS device into or out of iOS MDM Lost Mode, this API will send notification to the MDM agent in iOS for it to check-in. When device does check-in, the server will send the MDM command to toggle MDM Lost Mode. Note: This only works when device is in supervised mode, when this is called on a device that does not support MDM Lost Mode, this API will still return ok. But we will log Error activity when device return Error for MDM commands.

Request

- **Method:** PUT
- **Resource:** /api/v1/ios/spaces/<workspace-id>/lost-mode
- **JSON Data:** A JSON dictionary representing a LostModeRequestEntity entity.

Response

- **Status:** 204
- **Errors:**
  - 404 Not Found - If the workspace is not found.
  - 400 Bad Request - If device platform is not iOS.

Example

Request

```
PUT /api/v1/ios/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/lost-mode
Content-Type: application/json
Content-Length: 123
{
  "lost": true,
  "message": "Please return this device to where it’s found, thanks!",
  "phone_number": "123-456-7890",
}
```

Response

```
HTTP/1.1 204 No Content
```

Example

Request

```
PUT /api/v1/ios/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/lost-mode
Content-Type: application/json
Content-Length: 20
{
  "lost": false,
}
```

Response

```
HTTP/1.1 204 No Content
```

iOS MDM Lost Mode actions

Ask iOS device that is in iOS MDM Lost Mode to perform lost mode actions, this API will send notification to the MDM agent in iOS for it to check-in. When device does check-in, the server will send the MDM command to ask it to perform supported actions, for example send location or play sound. Note: This only works when device is in supervised mode, when this is called on a device that does not support MDM Lost Mode, this API will still return ok. But we will log Error activity when device return Error for MDM commands.

Request

- **Method:** POST
- **Resource:** /api/v1/ios/spaces/<workspace-id>/lost-mode/actions
- **JSON Data:** A JSON dictionary representing a LostModeActionRequestEntity entity.

Response

- **Status:** 204
- **Errors:**
  - 404 Not Found - If the workspace is not found.
  - 400 Bad Request - If device platform is not iOS. Or ask device to perform lost mode actions while it is not in lost mode.

Example

Request

```
POST /api/v1/ios/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/lost-mode/actions
Content-Type: application/json
Content-Length: 32
{
  "action": "send_location",
}
```

Response

```
HTTP/1.1 204 No Content
```

Example

Request

```
POST /api/v1/ios/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/lost-mode/actions
Content-Type: application/json
Content-Length: 29
{
  "action": "play_sound",
}
```

Response

```
HTTP/1.1 204 No Content
```

iOS Enterprise App Store

iOS Enterprise App Store Flow

Below is a sequence diagram that describes the flow of iOS Enterprise App Store.

```
title iOS Enterprise App Store Flow


participant UC
participant MDM Agent
participant PWS Server
participant JS App
participant Safari
participant User


UC->PWS Server: /api/v1/ios/app-store-url
PWS Server->UC: "app_store_url": "/api/v1/ios/app-store?s={encrypted token}"
UC->Safari: GET /api/v1/ios/app-store?s={encrypted token}
Safari->PWS Server: GET /api/v1/ios/app-store?s={encrypted token}
note over PWS Server: The HTML/JS returned here contains \nthe encrypted realm + workspace id that the JS App can \nuse to get HAWK credentials for \nthis workspace.
PWS Server->Safari: HTML/JS
Safari->Safari: Load JS App
JS App->PWS Server: GET /api/v1/ios/app-store/credentials?s={encrypted token}
PWS Server->JS App: Provides HAWK credentials
JS App->PWS Server: (HAWK) GET /api/v1/ios/spaces/{space-id}/apps
PWS Server->JS App: Apps for app store to display
JS App->Safari: Render app store
User->Safari: Install app
JS App->PWS Server: POST /api/v1/ios/spaces/{space-id}/apps \n{ package_name: foo }
PWS Server->JS App: 204 No Content
PWS Server->PWS Server: Queue installation of optional app
PWS Server->MDM Agent: Check In.
MDM Agent->PWS Server: Check In.
PWS Server->MDM Agent: Install App.
```

Get the Enterprise App Store URL

Clients call this endpoint to get the app store url associated with the hawk credentials of a workspace. The clients will open Safari with the app store url. As hawk credentials cannot be shared between Safari and the client application, realm authentication is used.

Request

- **METHOD:** GET
- **RESOURCE:** /api/v1/ios/app-store-url

Response

- **Status:** 200
- **Json Data:** A json dictionary with key as app_store_url and URL as the value
- **Errors:**
  - 401 Unauthorized - When the hawk credentials don't belong to a workspace

Example

Request

GET /api/v1/ios/app-store-url

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "app_store_url": "http://example.com/api/v1/ios/app-store?s=16417bca0242ac13000c"
}
```

Get the Enterprise App Store UI HTML

Returns the HTML page of the app store.

Request

- **METHOD:** GET
- **RESOURCE:** /api/v1/ios/app-store?s={encrypted token}

Response

- **Status:** 200
- **HTML:** HTML contains the app store UI.
- **Errors:**
  - 401 Unauthorized - When the realm auth does not belong to a workspace

Example

Request

```
GET /api/v1/ios/app-store?s=16417bca0242ac13000c
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/html

<html>
   ...
</html>
```

Get the HAWK credentials

Safari browsers call this endpoint to get the hawk credentials of a workspace. The hawk credentials will be used by the Safari browser to make API calls.

Request

- **METHOD:** GET
- **RESOURCE:** /api/v1/ios/app-store/credentials?s={encrypted token}

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following keys:
  - id: Space ID
  - api_url: A URL to use for v1 API requests
  - credentials: The credentials to use for HAWK authentication for future requests. This data structure is described in more detail in the Auth documentation.
- **Errors:**
  - 401 Unauthorized - When the realm auth does not belong to a workspace

Example

Request

GET /api/v1/ios/app-store/credentials?s=16417bca0242ac13000c

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "16417bca-efef-11e5-8e5a-0242ac13000c",
  "api_url": "https://api.pulseworkspace.net",
  "credentials": {
    "algorithm": "HS256",
    "expires": 1556657905,
    "id": "{alphanumeric string}",
    "renew": 1446657905,
    "secret": "{alphanumeric string}",
    "type": "hawk"
  }
}
```

Get the apps to be displayed on the enterprise app store

Returns all the apps that will be displayed on the app store UI page.

Request

- **METHOD:** GET
- **RESOURCE:** /api/v1/ios/spaces/{space-id}/apps

Response

- **Status:** 200
- **JSON Data:** Response data will be in the form of a JSON dictionary representing a list of WorkspaceAppDetailsEntity entities.
- **Errors:**
  - 401 Unauthorized - When the hawk credentials don't belong to a workspace

Example

Request

GET /api/v1/ios/spaces/16417bca-efef-11e5-8e5a-0242ac13000c/apps

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "items": [
    {
      "package_name": "com.enterprise.myapp",
      "command": "add",
      "network_access": "direct",
      "group_tags": [ "browser" ],
      "source": "enterprise",
      "url": "http://myserver.com/download/myapp.apk",
      "icon_url": "https://s3.amazonaws.com/icons/myapp.png",
      "version_code": 1234,
      "title": "Chrome",
      "installation_status": "installed",
      "installed_on": "2016-03-22T05:29:45",
      "description": "app description"
    },
    ...
  ]
}
```

Post the apps which need to be installed

Request

- **METHOD:** POST
- **RESOURCE:** /api/v1/ios/spaces/{space-id}/apps

Response

- **Status:** 204
- **JSON Data:** Response data will be in the form of a JSON dictionary with the following keys:
  - package_names: The package names of the apps which are needed to be installed.
- **Errors:**
  - 401 Unauthorized - When the hawk credentials don't belong to a workspace

Example

Request

```
POST /api/v1/ios/spaces/16417bca-efef-11e5-8e5a-0242ac13000c/apps
Content-Type: application/json

{
  "package_names": ["com.enterprise.app", ...]
}
```

Response

```
HTTP/1.1 204 No Content
```

Apple VPP

Upload Apple Token

Upload the base64 encoded Apple VPP Token which is from the Apple deploy web portal

Request

- **Method:** POST
- **Resource:** /api/v1/ios/vpp/token
- **Body:** A base64 encoded server token.

Response

- **Status:** 204
- **Errors:**
  - 400 Bad Request - If the token is invalid.

Example

Request

Shortened for conciseness.

```
POST /api/v1/ios/vpp/token HTTP/1.1
Content-Length: 568
Content-Type: multipart/form-data; boundary=723e82b4e41f4d0d9c66db24f035a326
Host: api.pulseone.net

--723e82b4e41f4d0d9c66db24f035a326

Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="token.txt"

MIAGCSqGSIb3DQEHA6CAMIACAQAxggFdMIIBWQIBADBBMCwxFDASBgNVBAoMC1B1bHNlU2VjdXJl
...
```

Response

```
HTTP/1.1 204 No Content
```

Retrieve Apple VPP token info

Request

- **Method:** GET
- **Resource:** /api/v1/ios/vpp/token

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary with the following keys:
  - id: (*str*) ID of the token.
  - organization: (*str*) Name of an organization.
  - expire_on: (*str*) RFC 3339 timestamp when the token will expire.
- **Errors:**
  - 404 Not Found - The VPP token does not exist.

Example

Request

```
GET /api/v1/ios/vpp/token
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "97577199-bd17-4102-8ab1-11008e8cb21b",
  "organization": "Pulse Secure MDM",
  "expire_on": "2023-11-20T02:53:09"
}
```

Delete Apple VPP token

Request

- **Method:** Delete
- **Resource:** /api/v1/ios/vpp/token

Response

- **Status:** 204

Example

Request

```
DELETE /api/v1/ios/vpp/token
```

Response

```
HTTP/1.1 204 No Content
```

Retrieve a list of workspaces which installed a VPP app

Request

- **Method:** GET
- **Resource:** /api/v1/ios/vpp/<app-id>/spaces

Response

- **Status:** 200
- **JSON Data:** A JSON dictionary with the following keys:
- items: A list of dictionary with the following keys:
  - space_id : (*str*) The workspace id of the device.
  - user_id : (*str*) The user id that was provided by the server.
  - device_id : (*int*) The id generated by the server during the registration of device.
  - manufacturer: (*str*) The device manufacturer
  - model: (*str*) The device model
  - os_version: (*str*) The device OS version
  - serial_number: (*str*) The serial number of the device
  - user_name: (*str*) The workspace user's username.
  - workspace_email : (*str*) The email address used for registering with workspace
- total: (*int*) Total number of spaces.
- **Errors:**
  - 404 Not Found - The VPP app does not exist.

Example

Request

```
GET /api/v1/ios/vpp/97577199-bd17-4102-8ab1-11008e8cb21b/spaces
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "total": 2,
  "items": [
    {
      "id" : "97577199-bd17-4102-8ab1-11008e8cb21b",
      "user_id" : "88ebcdd8-fb0c-4623-a66b-35ce39a958a3",
      "device_id" : "1",
      "os_version" : "7.1.2",
      "serial_number" : "88ebcdd8fb0c",
      "carrier" : "AirTel",
      "manufacturer" : "Apple",
      "model" : "iPhone7,2",
      "username": "paul",
      "workspace_email": "paul@example.com"
    },
    {
      "space_id" : "6ec6a624-f10e-4898-8c61-330efaa7a929",
      "user_id" : "0a65125e-edce-4af0-880b-7fd48b37f219",
      "device_id" : "2",
      "os_version" : "7.1.2",
      "serial_number" : "0a65125eedce",
      "carrier" : "AirTel",
      "manufacturer" : "Apple",
      "model" : "iPhone7,2",
      "username": "alex",
      "workspace_email": "alex@example.com"
    }
  ]
}
```

iOS - Device Info

Update Device Info

Sent to the server whenever essential info is set.
For managed iOS devices there is only one field need to be set in the request body ios_apns_device_token.
For managed iOS clients, we need all fields as long as they are available to the client.

Request

- **Method:** PUT
- **Resource:** /api/v1/ios/spaces/<workspace-id>/device
- **JSON Data:** a JSON encoded IosManagedClientDeviceInfoEntity

Response

- **Status:** 204
- **Errors:**
  - 404 Not Found - If the workspace is not found.
  - 400 Bad Request - If required fields are missing in the request.

Example

Request

For managed iOS device:

```
PUT /api/v1/ios/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/device
Content-Type: application/json
Content-Length: 54
Authorization: Hawk id="...", ...

{
  "ios_apns_device_token": "94e4bd6064ab5255c31aca4d1b664f1a8d4bc83a44ac205652182a31a1cb278c"
}
```

Response

```
HTTP/1.1 204 No Content
```

For managed iOS client:

```
PUT /api/v1/ios/spaces/61ec6a95-5669-45e7-bf27-f08641bf34e1/device
Content-Type: application/json
Content-Length: 256
Authorization: Hawk id="...", ...

{
  "ios_apns_device_token": "94e4bd6064ab5255c31aca4d1b664f1a8d4bc83a44ac205652182a31a1cb278c",
  "carrier": "AT&T",
  "manufacturer": "Apple",
  "model": "iphone 9.1",
  "name": "Pulse user1's iPhone",
  "os_version": "12.0.1",
  "os_type": "ios",
  "client_version_string": "12.1.2.3",
}
```

Report Generation API

This API documents the way to generate the PDF Graph reports of various Dashboards in Pulse One like Overall Dashboard, UEBA Dashboard, User Activities Dashboard, Workspace Dashboard , Profiled Devices Dashboard and e-mailing the same to subscribed recipients or downloading the same.

Download reports

Generates and streams selected dashboard category report in PDF format.

This API takes the dashboard-category as input. dashboard-category can take any of the following values:

- overviewdash: for getting overall dashboard report
- profilerdash: for getting profiled devices dashboard report
- workspacesdash: for getting workspaces dashboard report
- cloudsecuredash: for getting cloud secure dashboard report
- uebadashboard: for getting ueba dashboard report
- useractivitiesdash: for getting user activities dashboard report.

Request

- Method: POST
- Resource: /api/v1/reports/{dashboard-category}/reports
- JSON Data: Request data will be in the form of ReportPropertiesEntity

Response

- Status: 200
- Content-Type: application/octet-stream
- Body: The binary contents of PDF report.
- Errors:
  - 400 Bad Request - If any of the fields in ReportPropertiesEntity are missing.
  - 404 Page not found - If dashboard-category in the url is incorrect.

Example

Request

POST /api/v1/reports/config
Accept: application/json
Host: api.pulseone.net


```
{
   "download_requested": true,
   "email_addresses": null,
   "config": [
      {
         "name": "frequent_user_logins",
         "title": "FREQUENT USER LOGINS",
         "graph_type": "bar",
         "filter": {
```

```
            "time_frame": "now-30d/s",
            "realm": "Users"
         }
      },
      {
         "name": "vpn_realm_usage",
         "title": "Name",
         "graph_type": "horizontal_bar",
         "filter": {
            "time_frame": "now-1d/s"
      }
      },
      ...
   ]
}
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/octet-stream
Content-Length: <length>

Content of the dashboard report (Pdf)
...
```

Email reports

Generates and mails selected dashboard category report in PDF format to the specified recipients.

This API takes the dashboard-category as input. dashboard-category can take any of the following values only:

- overviewdash: for getting overall dashboard report
- profilerdash: for getting profiled devices dashboard report
- workspacesdash: for getting workspaces dashboard report
- cloudsecuredash: for getting cloud secure dashboard report
- uebadashboard: for getting ueba dashboard report
- useractivitiesdash: for getting user activities dashboard report

Request

- Method: POST
- Resource: /api/v1/reports/{dashboard-category}/reports
- JSON Data: Request data will be in the form of ReportPropertiesEntity

Response

- Status: 204
- Content-Type: application/json
- Errors:
  - 400 Bad Request - If any of the fields in ReportPropertiesEntity are missing or if invalid email addresses are provided.
  - 404 Page not found - If dashboard-category in the url is incorrect.

Example

Request

```
POST /api/v1/reports/config
Accept: application/json
Host: api.pulseone.net

{
   "download_requested": false,
   "email_addresses": [test1@comapany.com, test2@company.com],
   "config": [
      {
         "name": "frequent_user_logins",
         "title": "FREQUENT USER LOGINS",
         "graph_type": "bar",
         "filter": {
            "time_frame": "now-30d/s",
            "realm": "Users"
         }
      },
      {
         "name": "vpn_realm_usage",
         "title": "Name",
         "graph_type": "horizontal_bar",
         "filter": {
            "time_frame": "now-1d/s"
      }
      },
      ...
   ]
}
```

Response

```
HTTP/1.1 204 No Content
```

Locations API

Get Location Database Version Status

Get the location database (GeoLite2-City) details. If the database is uploaded and whether current version is up-to date or not.

Request

- Method: GET
- Resource: /api/v1/locations/database

Response

- Status: 200
- Content-Type: application/json
- JSON Data:
  - LocationDbInfoEntity: Containing Location Database details

Example

Request

```
GET /api/v1/locations/database
Accept: application/json
Host: pulseone.example.com
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: <length>

{
  "available": true,
  "latest_version": true,
  "published": "2020-05-30T20:48:31Z",
  "sha256_hash": "6a22d0f099ef482d9269feb61471678d17ced90a",
  "uploaded": "2020-05-30T20:48:31Z"
}

If there is no database uploaded:

HTTP/1.1 204 No Content
```

Upgrade Location Database

Upload the location database first time or upgrade the existing location database (GeoLite2-City) by uploading a newer location database package.

This API can be used by UI to upload a location database upgrade package content_hash and content, by making a multipart/form-data request.

SHA256 hash (content_hash) field of the content would be entered by admin while uploading the location database upload package via console UI (i.e., it is not calculated by browser).

Request

- Method: PUT
- Resource: /api/v1/locations/database
- Body: Two parts (multipart),
  - content_hash: String containing SHA256 hash of the content.
  - content: Binary contents of location database upgrade package.

Response

- Status: 204
- Errors:
  - 400 Bad Request - If database could not be uploaded due to SHA1 hash mismatch or invalid package.

Example

Request

```
PUT /api/v1/locations/database

Content-Disposition: multipart/form-data; name="content"; filename="GeoLite2-City.pkg"
content: <<binary-content of file GeoLite2-City.pkg>>
Host: pulseone.example.com

{
  "content": <<binary-content of file GeoLite2-City.pkg>>,
  "content_hash": "6a22d0f099ef482d9269feb61471678d17ced90a"
}
```

Response

If database is successfully uploaded:

```
HTTP/1.1 204 Created
```

If database could not be uploaded due to SHA1 hash mismatch or invalid package.

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: <length>
```

Entities

This section contains all the entities that are mentioned in the document.

ActivityEntity

This entity defines all possible properties for an activity represented in Unity.

- message: (*str*) It is like MSG in RFC 5424: A free-form message.
- time: (*str*) Timestamp per RFC 3339 , in UTC, with second precision.
- severity: (*str*) MUST be one of the following. The severity symbols are from RFC 5424. In order of decreasing severity:
- emergency: System is unusable
- alert: Action must be taken immediately
- critical: Critical conditions
- error: Error conditions
- warning: Warning conditions
- notice: Normal but significant condition
- informational: Informational messages
- debug: Debug-level messages
- activity_type: (*str*) A symbol of our choosing (like MSGID in RFC 5424):
  - requested_operation.started
  - requested_operation.succeeded
  - requested_operation.not_performed * system.event.logged
- reference.type: (*str*) The type of the entity referenced by this activity.
- reference.id: (*str*) The id of the entity referenced by this activity.
- params: (*FlexEntity*) A dictionary of parameters similar to STRUCTURED-DATA in RFC 5424. The parameters depend on the activity_type. As we define keys we will add them here. So far we have:
- conflicting_operation: value can be rsa_key or upgrade
- actor: (*RefKey*) A value describing the acting entity that caused the activity to be created. This can be a user, the system, a notification, etc.
- target: (*RefKey*) A value describing the target entity of an activity. This can be a workspace, an appliance, a user, etc.

ActivityEntity.Reference

A representation of an activity that references a particular entity.

- message: **Required.**
- timestamp: **Required.**
- severity: **Required.**
- activity_type: **Required.**
- reference.type: **Required.**
- reference.id: **Required.**
- params: **Optional.**

ActivityEntity.Update

A representation of an activity used during creation or update of an activity.

- message: **Required.**
- timestamp: **Required.**
- severity: **Required.**
- activity_type: **Required.**
- params: **Optional.**

ActiveSyncCertificateEntity

- certificate: (*str*) The base64 encoded pkcs12 certificate for ActiveSync connection
- cert_alias: (*str*) The alias for the ActiveSync certificate
- password: (*str*) The password for pkcs12 certificate
- format: (*str*) The format of the certificate. Currently, it is "pkcs12"
- version: (*str*) The version of the certificate in sha256 hash format

ActiveSyncCertificateEntity.Update

- version: (*str*) The version of certificate which is currently installed in client

AddEsapTaskCreationEntity

Entity sent when creating Add ESAP task for an appliance

- params: (*dict*) Parameters with following keys:
- firmware_id: (*str*) MD5 hash of the ESAP package (firmware) content
- activate: (*boolean*) To activate ESAP package.
  - True Activate ESAP.
  - False Don't activate ESAP.

AfwWorkspaceRegistrationResponseEntity

- id: (*str*) The id of the workspace. This is a UUID.
- key: (*str*) To prove identity to access the PWS server.
- api_url: (*str*) The hostname and port of the AfW REST APIs.
- client_type: (*str*) One of: managed_client, managed_device.
- credentials: (*Credentials*) The credentials to use for HAWK authentication for future requests. This data structure is described in more detail in the Auth entities documentation.
- acc: (*str*) A static string needed by the Android client.
- websocket_url: (*str*) The hostname and port for the websocket connection.
- sandbox_apk_url: (*str*) The download URL for sandbox.apk for non-AfW Android clients
- afw_enterprise_type: (afw|google) If this domain uses EMM-managed AfW Accounts, this will be "afw". If the domain uses Google Apps managed Google Accounts, this will be "google" or null.
- google_sender_id: (*str*) A unique value created when you configure your API project (given as "Project Number" in the Google Developers Console). The sender ID is used in the registration process to identify an app server that is permitted to send messages to the client app.

AppConfigSchemaEntity

- version (*str*): A version string of this schema. The version string is a hash of the most current version of the schema known to the server. This can be used to determine if the current schema is different from the existing schema. The value is otherwise opaque to the client.
- schema (*list of mixed*): The app config schema for this app, according to the platform of the app.

ApplianceActivityEntity(ActivityEntity)

Extends ActivityEntity to define an activity for an appliance.

- appliance: (*json*) JSON object containing the following:
- id: (*UUID*) The id of the appliance.
- name: (*str*) The name of the appliance.

ApplianceAuthMechanismStats

This entity defines properties related to the authentication mechanism statistics for an appliance for a defined range of time. This statistic is PPS specific.

- auth_mechanism: Nested stat key for authentication mechanism
- l3: (*int*) Number of L3 Authentications within a given time range
- mac: (*int*) Number of Mac Auth Bridge authentications within a given time range
- eap: (*int*) Number of EAP Authentications within a given time range
- other: (*int*) All other authentications including browser authentications within a given time range

ApplianceAuthStats

- auth_results: Nested stat key for authentication statistics
- success: (*int*) Number of successful authentications within the defined time range
- failure: (*int*) Number of failed authentications within the defined time range

ApplianceBackupMetadataGetEntity

Entity that defines metadata of a managed appliance's backup in GET request

- backup_id: (*str*) UUID of the backup
- name: (*str*) Name of the backup
- description: (*str*) Tag or description of the backup
- active_esap_version: (*str*) Version of ESAP package that is active on the appliance at the time of backup.
- active_pulse_desktop_version: (*str*) Version of Pulse Secure Desktop client package that is active on the appliance at the time of backup.
- installed_esap_versions: (*list of strings*) Versions of all *ESAP packages* installed (other than active version) on the appliance at the time of backup.
- installed_pulse_desktop_versions: (*list of strings*) Versions of all *Pulse Secure Desktop client* packages installed (other than active version) on the appliance at the time of backup.
- content_hash: (*str*) SHA1 hash of this backup's content
- content_size: (*str*) Size of this backup's content in bytes
- created: (*str*) RFC 3339 timestamp of when this backup's content was created.
- appliance_id: (*str*) UUID of the appliance from which this backup was created.
- type: (*str*) Type of the appliance from which this backup was created. Possible values for this field:
- VPN: Pulse Connect Secure
- NAC: Pulse Policy Secure

ApplianceBackupMetadataUpdateEntity

Entity that defines metadata of a managed appliance's backup in PUT request

- active_esap_version
- active_pulse_desktop_version
- installed_esap_versions
- installed_pulse_desktop_versions
- content_hash
- content_size

Refer to ApplianceBackupMetadataGetEntity section for definition of above fields.

ApplianceBackupTaskCreationEntity

Entity sent when creating backup task for an appliance

- params: (*dict*) Parameters with following keys:
- name: (*str*) Name of the backup to be created
- description: (*str*) Tag or description of the backup to be created

ApplianceClusterEntity

Defines information about a cluster.

- id: (*UUID*) UUID generated by Pulse One.
- intrinsic_id: (*str*) ID of the cluster provided by the appliance.
- name: (*str*) The cluster's name
- type: (*str*) Cluster type. Possible values active-active/active-passive

ApplianceComplianceStats

This entity defines properties related to compliance results statistics for an appliance for a defined range of time.

- compliance: Nested stat key for compliance statistics
- compliant: (*int*) Number of Compliant Health Checks within a given time range
- failed: (*int*) Number of Failed Health Checks within a given time range
- remediated: (*int*) Number of Remediated Health Checks within a given time range
- not_assessed: (*int*) Number of logins whose Health was not assessed within a given time range

ApplianceConfigBlockChangeEntity

Defines a change between two blocks.

- change: (str) Operation type (modified, added, deleted)
- title: (str) Title of the block.
- from: (ApplianceConfigBlockDiffEntity)
- to: (ApplianceConfigBlockDiffEntity)

ApplianceConfigBlockDiffEntity

Defines a set of fields necessary to determine block differences.

- id: (UUID) The id of the block.
- date: (str) RFC-3339 formatted timestamp of the time of modification/creation.
- content_id: (UUID) The id of the related block content.
- content_size: (int) Size of the block content in bytes.

ApplianceConfigBlockTypeDiffEntity

A list of changed blocks for a block type.

- type: (str) The block type
- blocks: (list of ApplianceConfigBlockChangeEntity)

ApplianceConfigCommitEntity.Commits

A list of commits for an appliance.

- id : (_UUID_) The id of the commit.
- date : (_str_) RFC-3339 formatted timestamp of the time of modification / creation.

ApplianceConfigHeadCommitEntity

Defines information about the appliance's actual and prescribed HEAD commits.

- commit: (Sha1) Sha1 hash of commit representing the appliance's actual configuration. If this value is empty in PUT operation, 204 No Content error is returned.
- prescribed_commit: (Sha1) Sha1 hash of commit representing prescribed configuration for an appliance. If this value is not set after a GET, appliance can assume that is in sync with Pulse One.

ApplianceDeviceHealthCheckFailureReasonStats

This entity defines properties related to health check failure reason statistics for an appliance for a defined range of time.

- hc_failure_reason: Nested stat key for healthcheck failure reasons
- <failure reason>: (int) Number of users that failed for the reason define by the key within a given time period. The failure reason MUST be one of the following:
  - hc_failure_0: Generic failure
  - hc_failure_1: Anti-virus not installed
  - hc_failure_2: Anti-virus not running
  - hc_failure_3: Anti-virus not up to date
  - hc_failure_4: Anti-virus scan time check failed
  - hc_failure_5: Firewall not installed
  - hc_failure_6: Firewall not running
  - hc_failure_7: Anti-malware not installed
  - hc_failure_8: Anti-spyware not installed
  - hc_failure_9: Anti-spyware not running
  - hc_failure_10: Unsupported OS
  - hc_failure_11: Restricted ports open
  - hc_failure_12: Required ports not open
  - hc_failure_13: Detected restricted processes
  - hc_failure_14: Required processes not detected
  - hc_failure_15: Detected restricted files
  - hc_failure_16: Required files missing
  - hc_failure_17: Incorrect registry settings
  - hc_failure_18: Detected restricted NetBIOS names
  - hc_failure_19: Required NetBIOS names not found
  - hc_failure_20: Detected restricted MAC address
  - hc_failure_21: Required MAC address not present
  - hc_failure_22: Certificate missing
  - hc_failure_23: Patches missing
  - hc_failure_24: Cache cleaner failed
  - hc_failure_25: Connected from non-SVW
  - hc_failure_26: Remote IMV failure
  - hc_failure_27: Enhanced Endpoint Security failed
  - hc_failure_28: 3rd party sub policy failed
  - hc_failure_29: Detected rooted devices
  - hc_failure_30: Detected jail broken devices
  - hc_failure_31: Mobile security suite not active
  - hc_failure_32: HDEncryption software not installed
  - hc_failure_33: Detected Unencrypted drives
  - hc_failure_34: Drives are missing
  - hc_failure_35: Unsupported client for HDEncryption check
  - hc_failure_36: Patch Management software not installed
  - hc_failure_37: Detected missing patches
  - hc_failure_38: Unsupported client for PatchMgmt check
  - hc_failure_39: Deprecated patch assessment rule
  - hc_failure_40: Deprecated EES policy
  - hc_failure_41: Deprecated SVW policy

ApplianceDeviceOSLoginStats

This entity defines properties related to the OS of the devices logging into an appliance for a defined range of time.

- device_os: Nested stat key for device OS statistics
- <device OS>: (*int*) Number of logins for a particular OS defined by the Key within a given time range. Device OS can be defined: unknown, android, ios, blackberry, windows_xp, windows_vista, windows_7 windows_8, windows_10, linux, chrome_os, others

ApplianceDeviceUserRolesStats

This entity defines properties related to the user role statistics for an appliance for a defined range of time.

- user_role: Nested stat key for user roles
- <role id>: (*int*) Number of users assigned to the user role defined by the Key within a given time period.

ApplianceEndpointStatsEntity

This entity defines properties related to endpoint statistics for an appliance for a defined given time period. Not all stats are required but when a stat is included a count MUST be given for all nested values. The failure to give a count means that piece of data is not available for the defined time period.

- timestamp_end: **Required.** (*str*) An RFC 3339 timestamp in UTC of when these statistics were collected. This timestamp provides at least one-second precision.
- timestamp_start: **Required.** (*str*) An RFC 3339 timestamp in UTC of the start time of the collected statistics. This timestamp provides at least one-second precision.

ApplianceFirmwareMetadataCollectionEntity

Represents a list of ApplianceFirmwareMetadataEntity entities

- total: **required** (*int*) Total number of ApplianceFirmwareMetadataEntity entities
- items: **required** (*array*) JSON array of ApplianceFirmwareMetadataEntity entities

ApplianceFirmwareMetadataEntity

Entity that defines metadata of a managed appliance firmware in GET request

- firmware_id: (*str*) MD5 hash of the firmware content
- description: (*str*) Description of the firmware
- version: (*str*) Version of the firmware
- type: (*str*) Type of the firmware. Possible values for this field:
- VPN: Pulse Connect Secure
- NAC: Pulse Policy Secure
- ESAP: End-point Security Appliance Plugin
- content_size: (*str*) Size of this firmware's content in bytes
- created: (*str*) RFC 3339 timestamp of when this firmware's content was created in Pulse One.

ApplianceFirmwareMetadataUpdateEntity

Entity that defines metadata of a managed appliance firmware in PUT request

- description: (*str*) Description of the firmware
- version: (*str*) Version of the managed appliance firmware in following format. <major_version>.<minor_version>R<maintaince_version>[[-HF<hotfix-number>],[-B<beta-number>],[EA<early-access-number]] Ex: '5.2R6', '8.1R7', '9.0R1:HF2', '9.0R1:B2', '5.2R3.1', '8.1R8:EA2'
- type: (*str*) Type of the firmware. Possible values for this field:
- VPN: Pulse Connect Secure
- NAC: Pulse Policy Secure
- ESAP: End-point Security Appliance Plugin
- content_hash: (*str*) MD5 hash of the firmware package content

ApplianceGroupConfigsSettingsEntity

- master_appliance_id: (UUID) The id of the master appliance of the group. Master Appliance id is always id of an appliance and not the cluster. This in contrast to Target Appliance that has a type of either appliance or cluster
- block_types: (list of str) The list of config block types associated with the ApplianceGroup. Only these block types are going to be replicated to Group's Target Appliances
- config_state: (str) Configuration state of the group regarding Target Appliances' configurations. For the further understanding of config_state, see "State Transition Diagram for the Group" of Appliance Groups. Possible states are:
- in_sync
- publish_required
- publish_in-progress
- publish_conflict
- publish_failed
- group_rendering

ApplianceGroupConfigsSettingsEntity.Update

- master_appliance_id: Required
- block_types: Required

ApplianceGroupEntity

- id: (UUID) The id of the group.
- name: (str) A name for the group. Limited to 255 characters.
- description: (str) A description of the group. Limited to 255 characters.
- master_appliance_id: (ApplianceGroupConfigsSettingsEntity.masterapplianceid) Group's Config Master Appliance ID
- config_state: (ApplianceGroupConfigsSettingsEntity.config_state) Group's config state
- members: (list of ApplianceGroupTargetEntity)

ApplianceGroupEntity.Create

- name: Required
- description: Optional

ApplianceGroupEntity.Update

- name: Required
- description: Optional

ApplianceGroupMemberListEntity

Describes group's members.

- id: (UUID) The id of the group.
- members: (list of ApplianceGroupTargetEntity)

ApplianceGroupMembersEntity

- item: (list of UUID) The list of IDs of members subscribed to this group that will be used as an operand of the operation. If empty, all subscribed appliances are considered as operand set. Appliance that are already cluster members will not be allowed to subscribe to a group.

ApplianceGroupTargetEntity(Entity):

A description of a single member of the group.

- appliance_id: (UUID) An ID of SecurityApplianceEntity being a member of the group.
- config_state: (str) Configuration state of this SecurityApplianceEntity.

ApplianceInfoEntity

- appliance_version: **optional** (*str*) Release Version of currently running on appliance format major.minor[.HF]-build_number. ie: 8.1R5.1-19456 or 8.1R5-21345
- boot_timestamp: **optional** (*str*) An RFC 3339 timestamp in UTC of the last time the appliance was rebooted.
- if_map: **optional** (*str*) A code to indicate IF-MAP configuration Possible values none/client/server
- concurrent_user_licenses **optional** (*int*) Number of concurrent user licenses installed on the appliance. For compatibility with older Pulse One servers, this should be the sum of the breakdown.
- concurrent_user_license_breakdown: (*dict*) (*Since May 2017*) Any subset of the following keys:
- access_licenses: (*int*) Number of ACCESS licenses, which can be used on PCS or PPS
- consec_licenses: (*int*) Number of CONSEC licenses, which can be used on PCS
- polsec_licenses: (*int*) Number of POLSEC licenses, which can be used on PPS
- license_mode: **optional** (*str*) License management mode of appliance. Possible values concurrent_users/named_users. This field will not be sent from appliances that are functioning as license-servers.
- named_user_licenses: **optional** (*dict*) Information about named user licenses
- installed: (*int*) Number of locally *installed* named user licenses on this appliance.
- consumed: (*int*) Number of *consumed* named user licenses (currently in use) on this appliance. This field will not be sent from license server. This consumed license count can be greater than installed count, if appliance is acting as license-client and some leased_in licenses are also in use.
- leased_in: (*int*) Number of named user licenses *leased* from license-server. This field will be sent only from appliances acting as license-client.
- leased_out: (*int*) Number of named user licenses *leased* out to all license-clients. This field will be sent only from appliances acting as license-server.
- license_role **optional** (*str* or *null*) (*Since May 2017*)
  - client if this appliance is a client of a license server
  - server if this appliance is a virtual or physical license server
  - standalone if this appliance is a standalone (from licensing perspective)
- license_server **optional** (*bool*) True if this appliance is a client of a license server (really!), false if not
- cluster: **optional** (*json*) This is a json object. This key will have value as null, if appliance is operating in standalone mode. If appliance is operating as a cluster member, then this json object contains the following:
- id: **Required.** (*str*) The cluster's permanent unique ID
- name: **Required.** (*str*) The cluster's name
- type: **Required.** (*str*) Cluster type. Possible values active-active/active-passive
- node_name: **Required.** (*str*) The current node name in cluster
- leader_node **Required.** (*boolean*) The role of the current node in cluster, whether acting as leader node or not.
- active_node **optional** (*boolean*) This key will be present only if cluster type is active-passive. Value *true* indicates that this appliance is active node in A/P cluster, false indicates that this appliance is passive node in A/P cluster.
- admin_url: **optional** (*str*) Appliance's Admin WebUI URL. It can be used by Pulse One to launch it so that IT Admin can configure this appliance using its native web interface (e.g., configure Master Appliance of the Group)
- model: **optional** (*str*) A value designating the Appliance platform type.

ApplianceRegistrationInfoEntity

Defines attributes of a registration info exchanged between Pulse One and an appliance during the registration handshake.

- registration_code: (*str*) A registration code that is configured in the Appliance by the Appliance admin.
- type: (*str*) A three letter designation of the Appliance functional type. Possible, case insensitive, values are:
  - VPN
  - NAC
- model: (*str*) A value designating the Appliance platform type.
- serial_number: (*str*) A code assigned for identification of a single unity of Appliance. Uniqueness of this value is not guaranteed.
- appliance_version: (*str*) A version number of software running on the Appliance at the time of the registration handshake.
- client_certificate_csr: (*str*) CSR of the gateway client certificate. Required by new PZT gateway registration. CSR common name should be the appliance ID.
- server_certificate_csr: **Optional** (*str*) CSR of the gateway server certificate. CSR common name should be the appliance ID.

ApplianceRestoreTaskCreationEntity

Entity sent when creating restore task for an appliance

- params: (*dict*) Parameters with following keys:
- backup_id: (*str*) UUID of backup which needs to be restored

ApplianceStatsAggregationEntity

Defines important attributes of a stat aggregation.

- timestamp_start: (*str*) RFC-3339 formatted timestamp of the start time of the duration.
- timestart_end: (*str*) RFC-3339 formatted timestamp of the end time of the duration.
- resolution: (*str*) Resolution of the data returned.
- limit: (*int*) Number of days of stats returned.
- latest_stats: (*SaHealthStatsEntity* or *ApplianceEndpointStatsEntity*) which contains an aggregate over the last resolution. ie: resolution of hour and now() is 9:15, then the result will contain an aggregation from 8:15 to 9:15.
- stats: (list of *Entity*) List of either SaHealthStatsEntity or ApplianceEndpointStatsEntity

Note: the list of stats will be grouped by the resolution, but modulo the resolution. ie: hour resolution will result in aggregates that are rounded to the hour. 9:15 - 9:00 will be one result and 9:00 to 8:00 will be another result.

ApplianceStatsInfoEntity

Defines overall appliance information for a customer domain.

- clusters: (*int*) Number of deployed clusters.
- appliances: (list *json*) List of JSON objects containing the following:
- type: (*str*) Appliance type.
- count: (*int*) Number of appliances in the domain of this type.
- concurrent_users: (*int*) Number of concurrent users connected or licenses consumed.
- concurrent_user_licenses: (*int*) Number of concurrent user licenses
- concurrent_user_license_breakdown: (*dict*) (*Since May 2017*) Any subset of the following keys:
  - access_licenses: (*int*) Number of ACCESS licenses, which can be used on PCS or PPS
  - consec_licenses: (*int*) Number of CONSEC licenses, which can be used on PCS
  - polsec_licenses: (*int*) Number of POLSEC licenses, which can be used on PPS
- license_mode: (*dict*) (*Since May 2017*) Any subset of the following keys: _ concurrent_users: (_int_) Number of PCS/PPS appliances running in concurrent_users licensing mode _ named_users: (_int_) Number of PCS/PPS appliances running in named_users licensing mode
- named_user_licenses: (*dict*) (*Since May 2017*) Any subset of the following keys. (Refer to ApplianceInfoEntity section for more details on each of these counts from any single PCS/PPS appliance) _ installed: (_int_) Number of installed named-user licenses in all PCS/PPS appliances _ consumed: (_int_) Number of consumed named-user licenses in all PCS/PPS appliances _ leased_in: (_int_) Number of named-user licenses leased by registered PCS/PPS appliances _ leased_out: (_int_) Number of named-user licenses leased out by registered license servers
- license_role: (*dict*) (*Since May 2017*) Any subset of the following keys:
  - client: (*int*) Number of PCS or PPS appliances acting as license clients
  - server: (*int*) Number of PCS or PPS appliances acting as license servers
  - standalone: (*int*) Number of PCS or PPS appliances acting as standalone (from licensing perspective)
- appliance_versions: (*json*) JSON object containing the following key-value pairs:
  - <appliance_version>: (*int*) Number of appliances with this version.
- *appliance_version: format major.minor[.HF].build_number ie: 8.1R4.30456 or 8.1R4.1.32789*

ApplianceStatsThresholdEntity

Defines the attributes returned from an appliance stats threshold query.

- appliance: (*json*) JSON object containing the following:
- id: (*UUID*) The id of the appliance.
- name: (*str*) Name of the appliance.
- value: (*float*) Value of the metric which exceeded the threshold.
- timestamp: (*str*) RFC-3339 formatted timestamp of the returned value.

ApplianceTaskCollectionEntity

Represents a list of ApplianceTaskEntity entities

- total: **required** (*int*) Total number of ApplianceTaskEntity entities
- items: **required** (*array*) JSON array of ApplianceTaskEntity entities

ApplianceTaskEntity

Entity returned when fetching an appliance's task details.

- id: (*str*) UUID of the task
- appliance_id: (*str*) UUID of the appliance
- type: (*str*) Type of task in hierarchical namespace representation. Possible values:
  - system.operations.appliance.config.backup: Take a configuration backup and upload to PulseOne.
  - system.operations.appliance.config.restore: Restore configuration from a specific configuration backup stored in PulseOne.
  - system.operations.appliance.config.xml_import: Import given XML configuration
- params: (*dict*) Parameters or entity data specific to above mentioned *type* parameter
- created: (*str*) RFC 3339 timestamp of when this task was created.
- status : (*str*) This field can have one of the values described below.
  - pending The appliance has not started working on the task yet.
  - in_progress_cancellable The appliance is working on the task. The appliance can cancel the task at this point in time.
  - in_progress_not_cancellable The appliance is working on the task. The appliance cannot cancel the task at this point in time.
  - failed The appliance finished working on the task. The task execution resulted in one or more errors.
  - success The appliance finished working on the task successfully.
  - cancelling The appliance has started working on cancelling the task.
  - cancelled The appliance has cancelled the task successfully.
- completed: (*str*) RFC 3339 timestamp of when this task was completed (either *success* or *cancelled* or *failed*). This field will be null, if this task has not completed yet.

ApplianceTaskSummaryEntity

Summary of an appliance's task

For definitions of below fields see ApplianceTaskEntity entity above.

- id
- type
- status

AppPermissionEntity

Resembles a policy property.

- name: (*str*) The name of the permission. This is set by Google.
- label: (*str*) The label of the permission.
- description: (*str*) The description of the permission.
- choices: (*list of str*) The choices available for this permission property. Will contain: denied, default, granted.
- prop_type: (*str*) The type of this permission. Will be choice.

AppPermissionsEntity

- schema (*AppPermissionsSchemaEntity*): The app permission schema.
- values (*dict of mixed*): The app permission values. The keys for this dictionary should be the unique name of the app permission entity. The value should be whatever value the IT Admin sets for this property.

AppPermissionsSchemaEntity

- version (*str*): A version string of this schema. The version string is a hash of the most current version of the schema known to the server. This can be used to determine if the current schema is different from the existing schema. The value is otherwise opaque to the client.
- schema (*list of AppPermissionEntity*): The app permissions schema for this app.

AppRestrictionsSchemaRestrictionEntity

Note: The Google API documents a value field which will be present, but should be ignored.

- key: (*str*) The unique key that the product uses to identify the restriction, e.g. "com.google.android.gm.fieldname".
- title: (*str*) The name of the restriction for display to the user.
- restriction_type: (*str*) The type of the restriction; this determines the UI to use for editing the restriction in addition to the acceptable values for the restriction. One of:
  - bool
  - bundle
  - bundleArray
  - choice
  - hidden
  - integer
  - multiselect
  - string
- description: (*str*) A longer description of the restriction, giving more detail of what it affects.
- entry: (*list of str*) For choice or multiselect restrictions, the list of possible entries' human-readable names.
- entry_value: (*list of str*) For choice or multiselect restrictions, the list of possible entries' machine-readable values.
- default_value: (*JSON*) The default value of the restriction; the value type depends on the type of the restriction.
- nestedRestriction: (*list of self*) Nested child properties related to this property.

ApproveProductEntity

- url: (*str*) A URL that displays a product's permissions and that can also be used to approve the product. In another words, the "approve" request (from the "Approve" button) will pass in the URL that is used to display the iframe with permissions to approve.The URL was returned by the GenerateProductApprovalUrlEntity.

CertificateScepRequestEntity

The entity has all the fields related to SCEP configurations needed for Android client to create a SCEP request to get certificates from external PKI server which supports SCEP.

- scep_url: (*str*) External PKI server SCEP endpoint URL.
- ca_name: (*str*) External PKI SCEP Server CA name.
- subject_o: (*str*) Certificate Subject O string.
- sublect_cn: (*str*) Certificate Subject CN string.
- subject_email: (*str*) Certificate Subject emailAddress string.
- challenge: (*str*) SCEP challenge which needs to be put in SCEP cerficiate request.
- key_size: (*int*) Key size for the private key.
- key_type: (*str*) Private key type.
- RSA RSA is the default key type we will indicate client to use.
- key_usage: (*int*) Certificate key usage.
- 5 5 means both digitalSignature and keyEncipherment

CloudSecureEndpointStatsEntity

This entity captures details about the endpoint that is accessing the cloud applications through CloudSecure. All top level keys for CloudSecureEndpointStatsEntity are optional except the id and session_start_time keys. Combination of id and session_start_time keys are used as unique identifier for endpoint's cloudsecure stats in Pulse One.

- cloud_secure_record_id: (*str*) Identifier for cloud secure record details. This is a unique identifier the cloud secure record entry. This is created on the appliance with combination of an integer and the machine id of the appliance.
- session_start_time: (*str*) RFC-3339 formatted timestamp for endpoint session start time on appliance.
- session_update_time: (*str*) RFC-3339 formatted timestamp of the endpoint's last update time.
- session_end_time: (*str*) RFC-3339 formatted timestamp for endpoint session end time on appliance.
- auth_result: (*bool*) Status of endpoint session authentication.
- os_type: (*int*) Identifer of the endpoint OS type.
- os_version: (*int*) Identifier of the endpoint OS version.
- compliance: (*str*) Identifier of the endpoint compliance status.
- endpoint_model: (*int*) Identifier of the endpoint model.
- endpoint_platform: (*int*) Identifier of the endpoint platform.
- user_access_record_id: (*str*) Identifier for the access history record. We will not be using this currently in Pulse One. But could be useful in future to tie with existing endpoint stats details.
- failed_applications: (*array*) List of failed applications identifiers. Each application identifier is an integer.
- success_applications: (*array*) List of successful application identifiers. Each application identifier is an integer.
- roles: (*array*) List of role identifiers assigned to endpoint. Each role identifier is a string.

CloudSecureMappingsEntity

Defines information about various dictionaries that captures mapping between identifiers and corresponding display names used in Cloud Secure data.

- os_types: (*array*) JSON array of IdValueEntity entities, which provides mapping between device os type identifier (int) to OS type of the device (string)
- os_versions: (*array*) JSON array of IdValueEntity entities, which provides mapping between device os version identifier (int) to device os version (string)
- endpoint_models: (*array*) JSON array of IdValueEntity entities, which provides mapping between device model identifier (int) to device model name (string)
- endpoint_platforms: (*array*) JSON array of IdValueEntity entities, which provides mapping between device platform identifier (int) to device platform name (string)
- applications: (*array*) JSON array of IdValueEntity entities, which provides mapping between application identifier (int) to application name (string)

CloudSecureStatsAggregationEntity

This entity captures aggregated CloudSecure stats and details.

- timestamp_start: (*str*) RFC-3339 formatted timestamp of the start time of the duration.
- timestart_end: (*str*) RFC-3339 formatted timestamp of the end time of the duration.
- resolution: (*str*) Resolution of the data returned.
- limit: (*int*) Number of entries for which data returned.
- duration: (*int*) Duration for which data returned.
- devices: (*json*) JSON object containing list of device names and corresponding counters.
- compliance: (*json*) JSON object containing list of compliance status and corresponding counters. Contains counters for Compliant, Non-Compliant, Remediated, Not-Assessed for compliance status keys
- roles: (*json*) JSON object containing list of role names and corresponding counters.
- applications: (*json*) JSON object containing below details
- successful: (*json*) JSON object containing list of successful application names and corresponding counters
- failed: (*json*) JSON object containing list of failed application names and corresponding counters
- total_successful: (*int*) Total count of successful applications This captures the count of all the applications, not restricted to the limit parameter passed.
- total_failed: (*int*) Total count of failed applications This captures the count of all the applications, not restricted to the limit parameter passed.
- stats: (*array*) JSON array containing application stat for given duration with resolution window. Contains below details
    - apps_count: (*json*) JSON object containing list of application names and corresponding counters.
    - timestamp: (*str*) RFC-3339 formatted timestamp indicating the time for the application stats.

ClusterHistoryEntity

- mode: (*str*) Cluster mode.
- trigger: (*str*) Represent who triggered transition to the current cluster mode.
    - manual
    - auto_failover
    - null: Should only be null for the first state of the system. All other times it is manual or auto_failover
- time: (*str*) Timestamp of when the cluster got into the current mode.

ClusterInfoEntity

- node: NodeStatusEntity
- cluster: ClusterStatusEntity

ClusterStatusEntity

- nodes: (*list* of *str*) A list of all the cluster node IPs.
- health: (*str*) Health status of the cluster node.
    - red: Unable to reach passive node.
    - yellow: Cluster status is changing (Should turn to green after few minutes)
    - green: Able to reach passive node
    - null
- auto_failover: (*int*) Auto failover timeout interval. (0 means auto failover is disabled.)

ConfigStatusEntity

- upload Optional (json) If this key is present, the value is a dictionary containing:
- status required (str) One of:
- idle: No upload is in progress or pending.
- pending: The appliance is waiting to export.
- exporting: The appliance is exporting its configuration.
- preparing: The appliance is preparing the exported configuration for uploading.
- uploading: The appliance is uploading its configuration to Pulse One.
- download Optional (json) If this key is present, the value is a dictionary containing:
- status required (str) One of:
- idle: No download is in progress or pending.
- pending: The appliance is waiting to download.
- downloading: The appliance is downloading its prescribed configuration from Pulse One.
- preparing: The appliance is preparing to apply the configuration.
- applying: The appliance is importing the configuration.
- rejected: the most recent import attempt failed.

The rejected status need not be persistent. On occasion the appliance MAY try again to download and import the configuration. The status codes generally happen in the sequence they are listed, beginning and ending with idle. There are other transitions, including these:

- For uploading:
  - The appliance can find that Pulse One's view of its configuration is already up to date. The upload status goes from preparing to idle.
  - While exporting, the appliance can find that it needs to restart the export. The status can go from exporting to pending.
- For downloading:
  - The pending status might not be implemented.
  - Pulse One could prescribe a configuration identical to what the appliance already has. The download status would go from preparing to idle.

DepAccountEntity

- id: (*str*) The unique id to identify the account.
- server_name: (*str*) An identifiable name for the PWS MDM server. It is the server name entered during applying for DEP account in Apple DEP portal.
- org_name: (*str*) The customer organization name. It is the organization name entered during applying for DEP account in Apple DEP portal.
- org_email: (*str*) The customer organization email address. It is the email address entered during applying for DEP account in Apple DEP portal.
- created_on: (*str*) An RFC 3339 timestamp in UTC when the account was created.
- token_expiry_on: (*str*) An RFC 3339 timestamp in UTC when the token will expire.

DepProfileEntity

- department: (*str*) The user-defined department or location name.
- support_phone_number: (*str*) A support phone number for the organization..
- is_supervised: (*bool*) If true, the device must be supervised.
- is_mdm_removable: (*bool*) If false, the MDM payload delivered by the configuration URL cannot be removed by the user via the user interface on the device; that is, the MDM payload is locked onto the device. This key can be set to false only if is_supervised is set to true.
- setup_items: A JSON dictionary with the following keys:
- passcode: (*bool*) If false, hides and disables the passcode pane.
- location: (*bool*) If false, disables Location Services.
- restore: (*bool*) If false, disables restoring from backup.
- apple_id: (*bool*) If false, disables signing in to Apple ID and iCloud.
- tos: (*bool*) If false, Skips Terms and Conditions.
- biometric: (*bool*) If false, skips Touch ID setup.
- payment: (*bool*) If false, skips Apple Pay setup.
- zoom: (*bool*) If false, skips zoom setup.
- siri: (*bool*) If false, disables Siri.
- diagnostics: (*bool*) If false, disables automatically sending diagnostic information.

DepRealmEntity

- realm: (*str*) The name of a realm.
- selected: (*bool*) Whether the realm is configured for DEP users to authenticate in PCS.

DomainProperty

- id: (*int*) Unique identifier of the property.
- read_only: (*boolean*) Whether the property is editable or not
- name: (*str*) The name of the property. This needs to be unique across all domain properties.
- max_value: (*int*) The maximum allowed value of the property.
- min_value: (*int*) The minimum allowed value of the property.
- value: (*int*, *boolean*, or *str*) The value of the property.
- label: (*str*) The label of the property. This needs to be unique across all domain properties.
- prop_type: (*str*) The type of the property such as "date", "int", "string", and "bool".
- group: (*str*) The group that the property belongs to.
- choices: (*list* of *str*) A list of possible values for the property. *NOTE*: A special case for this attribute *for domain properties* is the value SAs. This value indicates that the choices for this property should be populated with the current list of registered appliances. This special case value is not currently used with policy properties.
- created_on: (*str*) Timestamp of property creation date in rfc3339 format.
- modified_on: (*str*) Timestamp of property modification date in rfc3339 format.
- sensitive: (*boolean*) Whether the value of the properties is sensitive and should be masked, for example in logs and in UI.

EmailDomainEntity

- id: (*UUID*) The id of the email domain.
- email_domain: (*str*) An email domain such as example.com. Limited to 100 characters.
- domain_id: (*int*) The id of a domain.
- created: (*str) RFC 3339 Timestamp when the email domain was created.

EmailDomainEntity.Create

- email_domain: **Required**

Eula Entity

- content : (_str_) Markdown encoded content of End User License Agreement (EULA)
- id : (_str_) A unique id of this EULA in UUID format.
- created : (_str_) RFC 3339 Timestamp when this EULA has been created in Pulse One
- signed : (_str_) RFC 3339 Timestamp of at the time when the user signed this EULA. If this Timestamp is not present, this EULA has never been signed by the user.
- agrees : (_boolean_) Boolean indicator of user's acceptance of EULA

EulaEntity.id

A minimal representation of a EulaEntity entity with the following fields :

- id : (_str_) A unique id of this EULA in UUID format.

FirmwareUpgradeTaskCreation Entity

This particular entity could be system.operations.appliance.firmware.stage or. system.operations.appliance.firmware.install

FlexEntity

An entity that is a dictionary with no other defined structure.

GenerateProductApprovalUrlEntity

- url: (*str*) A URL that can be rendered in an iframe to display the permissions (if any) of a product. This URL can be used to approve the product only once and only within 24 hours of being generated. If the product is currently unapproved and has no permissions, this URL will point to an empty page. If the product is currently approved, a URL will only be generated if that product has added permissions since it was last approved, and the URL will only display those new permissions that have not yet been accepted.

GraphEntity

Captures the information about the data representation in the particular Graph.

- name : required (str) name of the Graph data.
- title : (str) title of the Graph data.
- graph_type : required (str) type of the Graph. it could be bar, pie, doughnut, strip, horizontal_bar, horizontal_double_bar.
- filter : (dict) JSON body of the REST call for a particular graph data.

IdValueEntity

Defines information about mapping between an integer identifier to the corresponding string value. Used by appliance to send the different CloudSecure mapping details.

- id: (*int*) identifier of the entity
- value: (*str*) value of the entity corresponding to the identifier

IosWorkspaceRegistrationResponseEntity

- id: (*str*) The id of the workspace. This is a UUID.
- key: (*str*) To prove identity to access the PWS server.
- api_url: (*str*) The hostname and port of the Pulse One API.
- client_type: (*str*) One of: managed_client, managed_device.
- credentials: (*Credentials*) The credentials to use for HAWK authentication for future requests. This data structure is described in more detail in the Auth entities documentation.

LocationDbInfoEntity

Represents GeoLite2-City database's information entity

- available: (boolean) If the database is available or not.
- latest_version: (boolean) If the database is of the latest version or not.
- sha256_hash: (str) SHA256 hash of the location database file.
- uploaded: (str) uploaded datetime of the database in RFC3339 format.
- published: (str) published datetime of the database in RFC3339 format.

LocationDetailsEntity

Captures location details

- city: **required** (*str*) City name of the location
- country: **required** (*str*) Country name of the location
- continent: **required** (*str*) Continent name of the location
- postal_code: **required** (*str*) Postal code for the location
- latitude: **required** (*str*) Latitude value for the location
- longitude: **required** (*str*) Longitude value for the location
- timezone: **required** (*str*) Timezone of the location

LostModeActionRequestEntity

- action: (*str*) Action to take when device is already in lost mode. Possible values:
- send_location - Ask iOS device which is in iOS MDM Lost Mode to send location to server. Supported since iOS 9.3
- play_sound - Ask iOS device which is in iOS MDM Lost Mode to play a sound. Supported since iOS 10.3

LostModeRequestEntity

- lost: (*bool*) If true, put device into lost mode. If false, put device out of lost mode.
- message: (*str*) Optional, This will be displayed on the device after it is in Lost Mode. If it is omitted, system will use a default text configured by Admin. Only applies when lost has value of true.

- phone_number: (*str*) Optional, This will be the phone number that device will display on home screen. Only applies when lost has value of true.

MobileAppEntity

An entity defining a MobileApp for the app catalog in Pulse Workspace. A mobile app is an app added to our system that can then be used in policies by creating an AppRule.

Note: Currently license data only applies to apps on the `android` platform.

- id (*UuidField*): Undocumented.
- platform: (str) The platform of the app. One of ios or android.
- package_name: (str) The package name of the app.
- title: (str) The title of the app.
- tags: (JSON) A list of tags for this app.
- app_config: (*AppConfigEntity*) The app config for this app, containing both schema and values.
- app_permissions: (*AppPermissionsEntity*) The app permissiosn for this app, containing both schema and values.
- category (*str*): Undocumented.
- creator (*str*): Undocumented.
- version (_str__): The version of the app.
- is_manual (*bool*): Whether the app is manually added or added from Store.
- hosting_type (*str*): Where the app bundle file was hosted, one of:
- external: Bundle file hosted by customer
- internal: Bundle file uploaded/hosted through PWS server
- download_url: (str) The download location of this app, if the app does not come from an app store.
- icon_url: (str) The url of the app icon.
- license_type: (mixed) The type of license for this app, if any. One of:
- not_licensed: An app is not licensed if Google does not provide any license data for this app.
- free: An app is free if it can be obtained for free.
- licensed: An app is considered licensed if it is not free and Google reports license data for the app.
- licenses_purchased: (int) The number of licenses purchased.
- licenses_provisioned: (int) The number of licenses provisioned.
- source (*str*): The source for this app, one of:
- appstore: The Apple App Store
- googleplay: The Google Play Store
- enterprise: An app provided by the enterprise.
- nostore: An app provided by the OS on the device. This typically only applies to native Android apps.
- country_code: (str) Two-letter country code. Default is "US".
- is_required: (bool) Whether the app is required or optional.
- network_access (str): Network access mode. It should be same as the one in app rule. One of blocked, direct, optional_vpn, per_app_vpn, or require_vpn
- modified_on (*datetime*): Undocumented.
- created_on (*datetime*): Undocumented.

MobileAppEntity.Update

This entity represents a Mobile App during creation.

- package_name
- title
- tags
- app_config
- app_permissions
- category
- creator
- version
- is_manual
- hosting_type
- download_url
- source
- is_required
- network_access
- country_code

MobileAppEntity.UploadResult

This entity represents the meta data of an uploaded iOS app bundle ipa file.

- package_name (*str*): Package name
- platform: (str) The platform of the app. One of ios or android.
- title: (str) The title of the app.
- icon_url: (str) The url of the app icon.
- version (*str*): Version of the package
- download_url: (str) The plist download location of this app, will be a PWS endpoint.
- md5 (*str*): md5 sum of the file

NodeStatusEntity

- ip: (*str*) IP of the cluster node.
- id: (*str*) ID of the cluster node.
- mode: (*str*) Cluster mode.
  - standalone
  - active
- hostname: Hostname of the cluster node.
- history: (*list* of *ClusterHistoryEntity*) A list of ClusterHistoryEntity. The last item in the list is the current node state. Limited to 10 items at most.

OrchestrationApplianceConfigEntity

This entity defines the properties for orchestrated appliance config.

Appliance configs that are needed by both AWS and vSphere.

- admin_username: (*str*) Admin username of the appliance.
- admin_password: (*str*) Admin password of the appliance.
- secondary_admin_username: (*str*) Username of the appliance used for REST access. (Generated by server.)
- secondary_admin_password: (*str*) Password of the appliance used for REST access. (Generated by server.)
- internal_fqdn: (*str*) Internal FQDN of the appliance.
- external_fqdn: (*str*) External FQDN of the appliance.
- auth_code: (*str*) Auth code used to request the license from licensing server.
- company_name: (*str*) Company name.
- primary_dns: (*str*) Primary DNS of the appliance.
- secondary_dns: (*str*) Secondary DNS of the appliance.
- private_domain_name: (*str*) Private domain name of the appliance.
- public_domain_name: (*str*) Public domain name of the appliance.
- instance_type: (*str*) Appliance instance type. (Admin doesn't need to provider this during the create process at the moment.)

Appliance configs that are needed by vSphere.

- internal_ip_address: (*str*) IP address of the appliance internal interface.
- internal_subnet: (*str*) Subnet of the appliance internal interface.
- internal_gateway: (*str*) Gateway of the appliance internal interface.
- internal_vlan: (*str*) Vlan of the appliance internal interface.
- external_ip_address: (*str*) IP address of the appliance external interface.
- external_subnet: (*str*) Subnet of the appliance external interface.
- external_gateway: (*str*) Gateway of the appliance external interface.
- external_vlan: (*str*) Vlan of the appliance external interface.
- management_ip_address: (*str*) IP address of the appliance management interface.
- management_subnet: (*str*) Subnet of the appliance management interface.
- management_gateway: (*str*) Gateway of the appliance management interface.
- management_vlan: (*str*) Vlan of the appliance management interface. Appliance configs that are needed by AWS.
- management_fqdn: (*str*) Management FQDN of the appliance.
- management_domain_name: (*str*) Management domain name.

OrchestrationAwsConfigEntity

This entity defines the properties for AWS config.

- ami: (*str*) Amazon machine image ID.
- region: (*str*) Region of the AWS server.
- vpc_id: (*str*) Virtual private cloud ID.
- public_subnet_id: (*str*) ID of the public subnet.
- private_subnet_id: (*str*) ID of the private subnet.
- managment_subnet_id: (*str*) ID of the management subnet. (Management Subnet ID is optional. Hence if it is not filled in, Management FQDN and Management Domain Name too are not needed. However if Management Subnet ID is filled in, Management FQDN and Management Domain Name are required.)
- deployment_key_name: (*str*) Name of the deployment key.

OrchestrationConfigEntity

This entity defines the properties for orchestration config.

- appliance_id: (*UUID*) ID of the orchestrated appliance.
- service_account_id: (*UUID*) ID of the service account.
- appliance_config: *OrchestrationApplianceConfigEntity*
- deployment_config: *OrchestrationDeploymentConfigEntity*

OrchestrationDeploymentConfigEntity

This entity defines the properties for service account orchestration config.

- id: (*UUID*) ID of the service account config.
- aws_config: *OrchestrationAwsConfigEntity*
- vsphere_config: *OrchestrationVsphereConfigEntity*

OrchestrationVsphereConfigEntity

This entity defines the properties for vSphere config.

- datacenter_name: (*str*) Name of the data center where the appliance is deployed in.
- datastore_name_ (_str*) Name of the data store where the appliance is deployed in.
- resource_pool_name: (*str*) Name of the resource pool where the appliance is deployed in.
- internal_network_name: (*str*) Name of the network that is used as the appliance's internal network.
- external_network_name: (*str*) Name of the network that is used as the appliance's external network.
- management_network_name: (*str*) Name of the network that is used as the appliance's management network.
- appliance_master_template_name: (*str*) Appliance master template defined in the vSphere client.

Policy

- id: (*int*) The id of the policy.
- created_on: (*str*) RFC 3339 timestamp of when this policy was created.
- edited: (*int*) The state of the policy. Possible states are:
  - 0 = published: a policy has been published to all workspaces
  - 1 = edited: a policy has been edited but not yet published
  - 2 = publishing: a policy is actively publishing
  - 3 = deleted: a policy has been marked for deletion
  - 4 = deleting: a policy is actively being deleted
- modified_on: (*str*) RFC 3339 timestamp of the last time this policy was modified.
- name: (*str*) The name of the policy.
- query: (*str*) The query of the policy to create. A query is used to search elasticsearch and determine which Workspaces this policy should be applied to.
- ldap_groups: (_list of _LdapGroups*) A list of LdapGroup entities.
- install_inside_geofencing_area: (*bool*) Only install the policy when a device is inside a list of geofencing area.
- geofencing_area: (_list of _GeofencingArea*) A list of GeofencingArea entities.
- seq: (*int*) The sequence of this policy.
- state: (*str*) A state string that is human-friendly. See 'edited'.

PolicyProperty

Required attributes:

- id: (*int*) Unique identifier of the property.
- policy_group_id: (*int*) The id of the policy group that the property belongs to.
- policy_name: (*str*) The name of the policy group that the property belongs to.
- name: (*str*) The name of the property. This needs to be unique across all policy properties in a policy group.
- group: (*str*) The group that this property belongs to.
- label: (*str*) The label of the property. This needs to be unique across all domain properties.
- prop_type: (*str*) The type of the property.
- created_on: (*str*) Timestamp of property creation date in rfc3339 format.
- modified_on: (*str*) Timestamp of property modification date in rfc3339 format.

Optional attributes:

- platform: (*str*) The platform where the property is used. The possible values are all, android, and ios. The default is all .
- sensitive: (*boolean*) Whether the property contains sensitive information such as a password.
- hidden: (*boolean*) Whether or not this property is hidden. The default is False.
- value: (*mixed*) The value of the property.
- max_value: (*int*) The maximum allowed value of the property.
- min_value: (*int*) The minimum allowed value of the property.
- choices: (*list* of *str*) A list of possible values for the property.

PolicyRequestEntity

- ldap_groups: (*list* of *LdapGroup.Id*) A list of LdapGroup.Id entities that this policy should be applied to.
- name: (*str*) The name of the policy to create.
- query: (*str*) The query of the policy to create. A query is used to search elasticsearch and determine which Workspaces this policy should be applied to.
- type: (*str*) The policy type. One of:
  - flattened
  - apps
  - rules
  - connections
- device_owner_mode: (*int*) Device owner mode that the policy applies to. One of:
  - 0: BYO
  - 1: Corporate Owned
  - 99: All (BYO and Corporate Owned). This is the default.
- install_inside_geofencing_area: (*bool*) Only install the policy when a device is inside a list of geofencing area.
- geofencing_area: (*list* of *GeofencingArea.Id*) Only install the policy when a device is inside this list of geofencing area.

Policies must have a name. For types apps, rules, and connections, the name must be unique within their type. Policies of type flattened do not need to have a unique name, but they must have a unique combination of query and ldap_groups.

Type flattened is deprecated. New policies should use the other policy types.

ProfilerBriefEndpointEntity

This entity defines minimal set of endpoint properties for reporting purposes.

- appliance_id: **required** (*UUID*) Appliance which has profiled this endpoint recently.
- fist_seen: **required** (*str*) An RFC 3339 timestamp in UTC when the endpoint has been first profiled.
- last_seen: **required** (*str*) An RFC 3339 timestamp in UTC when the endpoint has been most recently profiled.
- macaddr: **required** (*str*) MAC address of the endpoint.
- manufacturer: **optional** (*str*) Manufacturer of the endpoint endpoint based on MAC OUI.
- manufacturer_id: **optional** (*int*) Manufacturer ID based on FingerBank DB.
- os: **optional** (*str*) Operating system of the profiled endpoint.
- category: **optional** (*str*) Category of the profiled endpoint.
- previous_os: **optional** (*str*) Previous OS, if the endpoint has changed its profile.
- previous_category: **optional** (*str*) Previous Category, if the endpoint has changed its profile.
- sid: **optional** (*str*) Session ID of the endpoint that is maintained with appliance.
- ip: **optional** (*str*) IP Address of the endpoint.
- hostname: **optional** (*str*) Hostname of the endpoint.

ProfilerEndpointEntity

This entity defines properties that are available for a profiled endpoint.

- appliance_id: **optional** (*UUID*) Appliance which has profiled this endpoint recently.
- fist_seen: **required** (*str*) An RFC 3339 timestamp in UTC when the endpoint has been first profiled.
- last_seen: **required** (*str*) An RFC 3339 timestamp in UTC when the endpoint has been most recently profiled.
- macaddr: **required** (*str*) MAC address of the endpoint.
- manufacturer: **optional** (*str*) Manufacturer of the endpoint endpoint based on MAC OUI.
- manufacturer_id: **optional** (*int*) Manufacturer ID based on FingerBank DB.
- os: **optional** (*str*) Operating system of the profiled endpoint.
- category: **optional** (*str*) Category of the profiled endpoint.
- previous_os: **optional** (*str*) Previous OS, if the endpoint has changed its profile.
- previous_category: **optional** (*str*) Previous Category, if the endpoint has changed its profile.
- sid: **optional** (*str*) Session ID of the endpoint that is maintained with appliance.
- ip: **optional** (*str*) IP Address of the endpoint.
- hostname: **optional** (*str*) Hostname of the endpoint.
- ports: **optional** (*json*) Nmap scanned ports
- tcp: **optional** (*json*) TCP ports that are scanned by Nmap
  - closed: **optional** (*array*) list of closed TCP ports as per Nmap scan results
  - filtered: **optional** (*array*) list of filtered TCP ports as per Nmap scan results
  - open: **optional** (*array*) list of opened TCP ports as per Nmap scan results
- udp: **optional** (*json*)
  - closed: **optional** (*array*) list of closed UDP ports as per Nmap scan results
  - filtered: **optional** (*array*) list of filtered UDP ports as per Nmap scan results
  - open: **optional** (*array*) list of opened UDP ports as per Nmap scan results
- dhcp: **optional** (*json*) data collected by DHCP collector
- type: **required** (*str*) Indicates the message type.
- mac_addr: **required** (*str*) Mac address of the endpoint
- timestamp: **required** (*float*) epoch timestamp indicates when this section has last updated.
- classified_os: **optional** (*str*) Classified OS based on DHCP data alone.
- classified_category: **optional** (*str*) Classified Category based on DHCP data alone.
- combination_id: **optional** (*int*) DHCP classification id based on FingerBank.
- hostname: **optional** (*str*) Hostname of the endpoint that is retrived from DHCP message.

- message_type: **optional** (*int*) Message type from the recieved DHCP message.
- parameter_request_list: **optional** (*str*) DHCP options list sent in DHCP message.
- requested_ip: **optional** (*str*) IP address assigned by DHCP server to endpoint.
- src_mac: **optional** (*str*) MAC address of the network endpoint from where appliance received DHCP message.
- vendor_class: **optional** (*str*) Vendor class that is retrieved from DHCP message.
- snmp: **optional** (*json*) data collected by SNMP and Trap Collectors
- type: **required** (*str*) Indicates the message type.
- mac_addr: **required** (*str*) Mac address of the endpoint.
- timestamp: **required** (*float*) epoch timestamp indicates when this section has last updated.
- classified_os: **optional** (*str*) Classified OS based on the SNMP and Trap collectors data.
- classified_category: **optional** (*str*) Classified category based on the SNMP and Trap collectors data.
- endpoints_count: **optional** (*int*) Number of MACs learned for port in CAM of table of the switch.
- ip: **optional** (*str*) IP Address of the endpoint detected using ARP table query.
- is_cdp_lldp: **optional** (*bool*) Indicates whether it is discovered from CDP/LLDP entries.
- port: **optional** (*str*) Port on which the endpoint is connected.
- snmp_version: **optional** (*str*) SNMP version that we have used to fetch this data.
- switch_ip: **optional** (*str*) IP address of the switch to which endpoint is connected.
- switch_name: **optional** (*str*) Name that is used while adding switch to appliance.
- switch_support: **optional** (*str*) SNMP SysSupport value.
- switch_vendor: **optional** (*str*) Manufacturer of the switch.
- valid_port: **optional** (*bool*) Indicates whether the port details mentioned in this section is valid or not.
- poll: **optional** (*json*) Data that is collected by SNMP periodic polling
    - type: **required** (*str*) Indicates the message type.
    - mac_addr: **required** (*str*) Mac address of the endpoint.
    - timestamp: **required** (*float*) epoch timestamp indicates when this section has last updated.
    - endpoints_count: **optional** (*int*) Number of MACs learned for port in CAM of table of the switch.
    - ip: **optional** (*str*) IP Address of the endpoint detected using ARP table query.
    - is_cdp_lldp: **optional** (*bool*) Indicates whether it is discovered from CDP/LLDP entries.
    - port: **optional** (*str*) Port on which the endpoint is connected.
    - snmp_version: **optional** (*str*) SNMP version that we have used to fetch this data.
    - switch_ip: **optional** (*str*) IP address of the switch to which endpoint is connected.
    - switch_name: **optional** (*str*) Name that is used while adding switch to appliance.
    - switch_support: **optional** (*str*) SNMP SysSupport value.
    - switch_vendor: **optional** (*str*) Manufacturer of the switch.
    - valid_port: **optional** (*bool*) Indicates whether the port details mentioned in this section is valid or not.
- trap: **optional** (*json*) Data collected by Trap collector
    - type: **required** (*str*) Indicates the message type.
    - mac_addr: **required** (*str*) Mac address of the endpoint.
    - timestamp: **required** (*float*) epoch timestamp indicates when this section has last updated.
    - discon_time: **optional** (*float*) epoch timestamp indicates when is last time endpoint disconnected from switch.
    - conn_time: **optional** (*float*) epoch timestamp indicates when is last time endpoint connected switch.
    - ifindex: **optional** (*str*) Switch interface index on which Switch is connected
    - port: **optional** (*str*) Port on which the endpoint is connected.
    - snmp_version: **optional** (*str*) SNMP version that we have used to fetch this data.
    - switch_ip: **optional** (*str*) IP address of the switch to which endpoint is connected.
    - switch_name: **optional** (*str*) Name that is used while adding switch to appliance.
    - switch_support: **optional** (*str*) SNMP SysSupport value.
    - switch_vendor: **optional** (*str*) Manufacturer of the switch.
    - valid_port: **optional** (*bool*) Indicates whether the port details mentioned in this section is valid or not.
    - vendor: **optional** (*str*) Vendor of the switch
    - vlan: **optional** (*str*) VLAN in which endpoint is connected.

- cdp: **optional** (*json*)
  - capability: **optional** (*int*) Capabilities enabled on endpoint.
  - classified_category: **optional** (*str*) Classified OS based on CDP data.
  - classified_os: **optional** (*str*) Classified Category based on CDP data.
  - duplex: **optional** (*str*) Indicate whether half/full duplex enabled.
  - ip: **optional** (*str*) IP address of the endpoint.
  - platform: **optional** (*str*) Platform of the endpoint provided in CDP message.
  - sys_name: **optional** (*str*) SNMP sys-descr's sys-name.
  - type: **optional** (*str*) Indicates the name of the section.
  - version: **optional** (*str*) Platform version of the endpoint.
- lldp: **optional** (*json*) LLDP data collected from supported switch by SNMP collector.
  - type: **optional** (*str*) Indicates the type of the section.
  - mac_addr: **optional** (*str*) Mac address of the endpoint.
  - capability_e: **optional** (*int*) Capabilities enabled on endpoint.
  - capability_s: **optional** (*int) Capabilities supported by the endpoint.
  - classified_os: **optional** (*str*) Classified OS based on the LLDP data.
  - classified_category: **optional** (*str*) Classified category based on the LLDP data.
  - sys_name: **optional** (*str*) SNMP sys descr name of the endpoint.
  - version: **optional** (*str*) OS Version that came in LLDP message.
- nmap: **optional** (*json*) data collected by Nmap Collector
- type: **required** (*str*) Indicates the message type.
- mac_addr: **required** (*str*) Mac address of the endpoint.
- timestamp: **required** (*float*) epoch timestamp indicates when this section has last updated.
- classified_os: **optional** (*str*) Classified OS based on the Nmap collectors data.
- classified_category: **optional** (*str*) Classified category based on the Nmap collectors data.
- ip: **optional** (*str*) IP address of the endpoint.
- host_name: **optional** (*str*) Hostname of the endpoint.
- snmp_sysdescr: **optional** (*str*) SNMP sysdescriptor of the endpoint if it has SNMP agent running.
- status: **optional** (*str*) Indicates whether endpoint is up or down.
- os_fingerprinted: **optional** (*bool*) Indicates wether Nmap is able to detect OS of endpoint or not.
- distance: **optional** (*int*) Indicates how many hops away from the appliance.
- ports: **optional** (*json*) Nmap scanned ports
  - tcp: **optional** (*json*) TCP ports that are scanned by Nmap
  - closed: **optional** (*array*) list of closed TCP ports as per Nmap scan results
  - filtered: **optional** (*array*) list of filtered TCP ports as per Nmap scan results
  - open: **optional** (*array*) list of opened TCP ports as per Nmap scan results
  - udp: **optional** (*json*)
  - closed: **optional** (*array*) list of closed UDP ports as per Nmap scan results
  - filtered: **optional** (*array*) list of filtered UDP ports as per Nmap scan results
  - open: **optional** (*array*) list of opened UDP ports as per Nmap scan results
- os_matches: **optional** (*array*) Nmap detected OS details.
  - classes: **optional** (*array*)
  - accuracy: **optional** (*int*) Nmap detected accuracy score.
  - cpe_list: **optional** (*array*) cpe list based on detected OS.
  - osfamily: **optional** (*str*) Name of the OS.
  - osgen: **optional** (*str*) Version of the OS.
  - type: **optional** (*str*) Category of the endpoint.
  - vendor: **optional** (*str*) Vendor of the endpoint.
  - line: **optional** (*int*) line number of Nmap output from which this has been parsed.
  - name: **optional** (*str*) Name of the Nmap detected OS.
- wmi: **optional** (*json*) data collected by WMI Collector
- type: **required** (*str*) Indicates the message type.
- mac_addr: **required** (*str*) Mac address of the endpoint.
- timestamp: **required** (*float*) epoch timestamp indicates when this section has last updated.
- category: **optional** (*str*) Category of the endpoint.

- status: **optional** (*str*) Status of the endpoint. Indicates whether it is up/down.
- classified_os: **optional** (*str*) Classified OS based on the WMI collectors data.
- classified_category: **optional** (*str*) Classified category based on the WMI collectors data.
- hostname: **optional** (*str*) Hostnmae of the endpoint.
- os: **optional** (*str*) Operating System of the endpoint.
- username: **optional** (*str*) Currently logged in user of endpoint.
- history: **optional** (*json*) Endpoint history that provides details on profile, ip and session change details. It is a JSON object with appliance id as key, and endpoint history with corresponding appliance as value.
- <appliance_id>: **optional** (*json*)
    - ip_address: **optional** (*array*) Represents IP address change history of an endpoint.
    - collector_type: **optional** (*str*) Represent which collector triggered this change.
    - detected_time: **optional** (*str*) An RFC 3339 timestamp in UTC when this change got triggered.
    - ip: **optional** (*str*) IP address of the endpoint.
    - host_name: **optional** (*str*) Hostname of the endpoint.
    - profile: **optional** (*array*) Represents OS & Category change history of an endpoint.
    - collector_type: **optional** (*str*) Represent which collector triggered this change.
    - detected_time: **optional** (*str*) An RFC 3339 timestamp in UTC when this change got triggered.
    - os: **optional** (*str*) Operating System.
    - category: **optional** (*str*) Category of the endpoint.
    - session_details: **optional** (*array*) Represents session change history of an endpoint.
    - collector_type: **optional** (*str*) Represent which collector triggered this change.
    - detected_time: **optional** (*str*) An RFC 3339 timestamp in UTC when this change got triggered.
    - session.login_host: **optional** (*str*) Appliance IP on which this session is established.
    - session.session_type: **optional** (*str*) Represent whether local or remote session.
    - session.state: **optional** (*str*) Is active session or not.
- session: **optional** (*json*) data that is collected from appliance local/remote session.
- login_host: **optional** (*str*) Appliance hostname on which session is established.
- login_host_addr: **optional** (*str*) Appliance IP on which session is established.
- mac_addr: **optional** (*str*) MAC address of the endpoint.
- session_type: **optional** (*str*) Local/Remote session.
- sid: **optional** (*str*) Session ID.
- source_ip: **optional** (*str*) IP address of the endpoint.
- state: **optional** (*str*) Indicates whether session is active or not.
- switch_ip: **optional** (*str*) IP address of the switch on which endpoint is connected.
- type: **optional** (*str*) Indicates section type.
- user_agent: **optional** (*str*) User agent that is collected from the session.
- MDM: **optional** (*json*) Data that id collected from MDM server.
- CompromisedStatus: **optional** (*str*) Indicates whether endpoint is compromised or not.
- EnrollmentStatus: **optional** (*str*) Indicates endpoint enrollment status with MDM server.
- IMEI: **optional** (*str*) IEMI number of the endpoint, if it is a mobile.
- IsPasscodeCompliant: **optional** (*str*) Indicates endpoint pass code compliant status.
- IsPasscodePresent: **optional** (*str*) Indicates whether endpoint has pass code or not.
- LastComplianceCheckOn: **optional** (*str*) Timestamp of last compliance check.
- LastCompromisedCheckOn: **optional** (*str*) Timestamp of last compromised check.
- LocationGroupName: **optional** (*str*) Indicates the location group (Local/Remote)
- Manufacturer: **optional** (*str*) Manufacturer of then endpoint.
- UDID: **optional** (*str*) MDM server assigned unique ID for the endpoint.
- classified_category: **optional** (*str*) Classified category based on MDM data.
- classified_os: **optional** (*str*) Classified OS based on MDM data.
- complianceReason: **optional** (*str*) Reason of compliance status.
- endpointName: **optional** (*str*) Name of the endpoint.
- isCompliant: **optional** (*str*) Indicates whether endpoint is compliant or not.
- isCompromised: **optional** (*str*) Indicates whether endpoint is compromised or not.
- isEnrolled: **optional** (*str*) Indicates enrollment status of the endpoint.

- lastSeen: **optional** (*str*) Timestamp when MDM server has seen this endpoint last time.
- macAddress: **optional** (*str*) MAC address of the endpoint.
- manufacturer: **optional** (*str*) Manufacturer of the endpoint.
- model: **optional** (*str*) Model of the endpoint.
- osVersion: **optional** (*str*) Operating System Version.
- ownership: **optional** (*str*) Enterprise/User owned endpoint.
- platform: **optional** (*str*) Platform name.
- serialNumber: **optional** (*str*) Serial number of the endpoint.
- type: **optional** (*str*) Section name, usually it is 'wmi'.
- userEmail: **optional** (*str*) Logged in users email ID.
- userName: **optional** (*str*) Logged in username.
- user_agent: **optional** (*json*)
- type: **required** (*str*) Indicates the message type.
- classified_os: **optional** (*str*) Classified OS based on the user agent collectors data.
- classified_category: **optional** (*str*) Classified category based on the user agent collectors data.
- user_agent: **optional** (*str*) User agent collected from the session details.

ProfilerEndpointsEntity

Represents JSON object that is returned by endpoints API.

- total: **required** (*int*) Total number of the endpoint that can be returned by this API request.
- data: **required** (*array*) JSON array of ProfilerBriefEndpointEntity .

ProfilerProfileChangeStatsEntity

This entity represents profile change stats (number of endpoints changed it profile vs no profile change) of endpoints.

- profile_changed: **required** (_int_)
- others: **required** (*int*)

ProfilerSessionStatsEntity

This entity represents session stats (number of active sessions vs no sessions) of endpoints.

- sessions: **required** (*int*)
- no_sessions: **required** (_int_)

ProfilerStateStatsEntity

This entity represents with category and without category stats of endpoints.

- profiled: **required** (*int*) Number of endpoints with category.
- not_profiled: **required** (_int_) Number of endpoints without category.

Property Types

The possibly types of properties are:

- bool: True/False value.
- choice: A choice of options for a value.
- int: An integer (alias).
- integer: An integer.
- ipaddr: An IP Address.
- multi-text: A multi-line text value.
- string: A text value.

RefKey

A RefKey value is a compound value that contains a type and an id. RefKey should only be used in situations where the type of a reference is ambiguous.
Example of appropriate usage would be using RefKey for an actor or target on an activity:

- notification-abka7890v
- security_appliance-196c883a-2ad9-4ff8-80b5-f35551f8a0b5
- user-2c7728b4-2d4f-43ca-adc2-2e1f08ccfaf7

ReportPropertiesEntity

Captures the information about the data representation in a particular category of Dashboard report.

- download_requested : (bool) Ignores email_addresses field and downloads the report if set as True.
- email_addresses : (array of string) : An array of email addresses of the recipients of the dashboard report.
- config : (array of objects) array of GraphEntity

RoleEntity

- id: (UUID) The UUID of the role.
- name: (str) The display name of the role.
- is_system: (bool) True if the role is a system role.
- permissions: (RolePermissionsEntity) Permissions this role grants the user.

RoleEntity.Id

- id: (UUID) The UUID of the role.

RolePermissionsEntity

- action_map: (dict{str: list[str]}) Immutable Dictionary that maps actions names to a list of more actions. Non-recursive. When determining the actions allowed for a namespace, each action granted to the namespace should be checked in the action_map first. If the action exists in the map, the list of actions should be granted for the namespace instead. For example, if the action_map is {"WRITE_ACTIONS": ["READ", "CREATE"]} and the namespace has the "WRITE" action granted, then the namespace should actually be granted "READ" and "CREATE" actions.
- namespaces: (dict{str: list[str]}) Dictionary that maps namespaces to a list of actions the user is granted to perform on that namespace.

For example:

```
{
  "action_map": {
     "WRITE_ACTIONS": ["CREATE", "READ", "UPDATE"],
     "DELETE_ACTIONS": ["CREATE", "READ", "UPDATE", "DELETE"],
   },
  "namespaces": {
     "admin.*": ["READ"],
     "admin.settings.*": ["READ"], \
     "admin.appliances.appliance": ["DELETE_ACTIONS"], \
     "admin.appliances.configdist": ["WRITE_ACTIONS"],
     "admin.appliances.appliance.operations": ["WRITE_ACTIONS"],
     "admin.appliances.log_aggregator": ["READ"],
     "admin.users.user": ["READ"]
  }
}
```

SaHealthStatsEntity

This entity defines properties related to health statistics for an appliance for a specific moment in time.

- concurrent_users: (*int*) Number of concurrent users connected
- activesync_users: (*int*) Number of Active Sync users connected
- cpu_utilization: (*float*) Percentage of CPU utilization
- disk_utilization: (*float*) Percentage of disk utilization
- memory_utilization: (*float*) Percentage of memory utilization
- network_throughput: (*int*) Network Throughput
- timestamp: **Required.** (*str*) An RFC 3339 timestamp in UTC of when these statistics were collected. This timestamp provides at least one-second precision.

ScheduledTaskEntity

Entity sent when creating scheduled task. Refer ScheduledTaskUpdateEntity for missing definitions for some of the below fields.

- id: (*str*) Unique identifier (ID) of this scheduled task.
- enabled: (*boolean*)
  - True Task is scheduled for next run.
  - False Task is not scheduled for next run, as the task has been completed as per its schedule or explicitly disabled by admin.
- appliance_id
- cluster_id
- group_id
- once
- task
- type
- params
- comments
- last_task: (*dict*) Details of the last run of this scheduled task. (Optional)
- status: (*str*) Status of the last run for this scheduled task. Possible values:
  - scheduled This task is yet to run at least once
  - in_progress This run of the scheduled task has started and in-progress
  - successful Last run of task completed successfully.
  - failed Last run of this task has failed. Pulse One activity logs will contain error details.
  - partial_success This task has been completed, but partially successful. Ex: When task is for a group, then the task is successfully completed on some of the appliances under the group. Pulse One activity logs will contain more details.
- completed: (*str*) RFC 3339 timestamp of when this task was last run.

ScheduledTasksCollectionEntity

Represents a list of ScheduledTaskEntity entities.

- total: **required** (*int*) Total number of ScheduledTaskEntity entities
- items: **required** (*array*) JSON array of ScheduledTaskEntity entities

ScheduledTaskUpdateEntity

Entity sent when creating or updating a scheduled task.

- appliance_id: (*str*) UUID of the appliance on which this scheduled task should run. (or)
- cluster_id: (*str*) UUID of the cluster on which this scheduled task should run. (or)
- group_id: (*str*) UUID of the group on which this scheduled task should run.
- once (dict) : (*str*) RFC 3339 timestamp of when this task is expected to run once.
- task (dict) - Parameters and type of the task being scheduled
- type: (*str*) Type of the task being scheduled. Possible values: system.operations.appliance.firmware.stage - To schedule staging of firmware on selected target identified by appliance_id or cluster_id or group_id system.operations.appliance.firmware.install - To schedule installation of staged firmware on selected target identified by appliance_id or cluster_id or group_id
- params: (*dict*) Creation params specific to above task type. Mapping from type to params as follows: system.operations.appliance.firmware.stage - FirmwareUpgradeTaskCreationEntity system.operations.appliance.firmware.install - FirmwareUpgradeTaskCreationEntity
- comments: (*str*) *Optional* comments about this task being scheduled

SecurityAppliance

Defines all fields within the security appliance.

- id: (UUID) The id of the appliance.
- reg_id: (str) The registration id of the appliance. This is used in generating appliance registration codes.
- name: (str) The name of the appliance.
- state: (str) The state of the appliance. Possible states are:
  - unregistered
  - registered
  - deleted
- created: (str) A timestamp of when the appliance was created.
- updated: (str) A timestamp of when the appliance was last updated.
- group_id: (UUID) ID of the group of which this appliance is a member.
- appliance_commit_id: (UUID) The id of the ApplianceConfigCommit this appliance is currently associated with. In general, that defines current configuration of the appliance.
- pending_commit_id: (UUID) The id of the ApplianceConfigCommit this appliance has as a pending commit to be published. The pending commit is a configuration that has been generated by Pulse One and is pending IT Admin's approval to become a prescribed configuration for the appliance.
- type: (str) A three letter designation of the Appliance functional type. Possible, case insensitive, values are:
  - VPN
  - NAC
- model: (str) A value designating the Appliance platform type.
- serial_number: (str) A code assigned for identification of a single unity of Appliance. Uniqueness of this value is not guaranteed.
- config_size: (int) The overall size of uncompressed configuration referenced by ApplianceConfigCommit that is currently associated with the appliance.
- config_created: (*str) RFC 3339 Timestamp when the data was collected
- appliance_version: (str) A version number of software running on the Appliance at the time of the registartion handshake. Format of the version number is as follows: major.minor[.HF]- build_number.
- cluster: JSON object containing the following information about the cluster.
- id: (UUID) Pulse One generated UUID for the cluster.
- config_state: (str) The configuration state of the appliance. For the further understanding of config_state, see "State Transition Diagram for the Group" of Appliance Groups. Possible states are:
  - in_sync
  - rendering_group
  - rendering
  - publish_required
  - publishing
  - conflict
  - ignore
  - publish_failed
  - unknown
  - notification_channel_status: (str) The connection state of the appliance. If state of the appliance is unregistered, the connection state will always be offline. Possible states are:
  - online
  - offline

SecurityAppliance.Get

Entity returned when retrieving details about a single security appliance. Undefined fields are defined in the SecurityAppliance entity above.

- id
- name
- state
- created
- updated
- group_id
- appliance_commit_id
- pending_commit_id
- type
- model
- serial_number
- config_size
- config_created
- appliance_version
- cluster
- id
- config_state
- notification_channel_status

SecurityAppliance.GetAll

Entity returned when retrieving a list of security appliances. Undefined fields are defined in the SecurityAppliance entity above.

- id
- reg_id
- name
- state
- created
- updated
- group_id
- appliance_commit_id
- type
- model
- serial_number
- appliance_version
- config_size
- cluster
- config_state
- notification_channel_status

UserAccessRecordEntity

This entity defines properties of a user access record of a managed appliance. Here user could include a regular user or a device (mac-user) when logins performed by a device's (Ex: IoT) credentials.

- user_access_record_id: **required** (*str*) Unique ID for this user access entity
- login_time: **required** (*str*) An RFC 3339 timestamp in UTC of the time at which user performed login attempt
- logout_time: **optional** (*str* or *null*) An RFC 3339 timestamp in UTC of the time at which the user's session ended. A user's session can end for other reasons besides the user performing a logout. A session can end due to idle timeout, loss of heartbeat, session expiration, admin terminates session, and more.
- authentication_mechanism: **required** (*str*) Authentication mechanism used to perform login Possible values are: - `Other_Method` - `L3_Auth` - `Auth_802_1X` - `MAC_Auth`
- authentication_succeeded: **required** (*boolean*) Result of authentication during login. Possible values are: ```
  - True - Authentication successful
  - False - Authentication failed ```
- authentication_failure_reason: **required** (*str* or *null*) The reason for authentication failure, if authentication_succeeded is False. If authentication_succeeded is True, then this value would be null. Current possible values are given below. More could be added to this list in future.
  - null,
  - Failed
  - No Auth
  - Wrong Certificate
  - Admin Only
  - Admin Recovery
  - Feature Unlicensed
  - Max Sessions
  - Short Password
  - Account Disabled
  - Account LockedOut
  - Account Expired
  - No Roles
  - Too Many Sessions
  - Revoked Certificate
  - IP Denied
  - UA Denied
  - IP Blocked
  - No Certificate
  - Compat Authentication
  - Compat Directory
  - Netegrity Page
  - Radius Page
  - Realm Remediate
  - Role Remediate
  - OCSP Failure
  - No Assertion
  - Connect Error
  - SignIn Notification Decline
  - Chassis SSO Failed
  - Login Cancel
  - Too Many EES
  - Too Many PRM
  - Token Or OTP
  - Invalid Assertion
  - Empty Assertion
  - SPNEGO_SSO
  - Max Session Per User

    - Empty User Name
    - Password Change required but Password Management disabled
    - FIPS Client Required
    - Needs SAML Authentication
    - No Realm
    - Maximum Onboard Devices ```
- roles: **optional** (*list of str*) List of role-names that user has been assigned in this login session.
- username: **optional** (*str*) The username of the user who performed login attempt. This could be empty for device logins.
- realm: **optional** (*str*) The realm through which user performed login attempt.
- authentication_server_name: **required** (*str*) The name of primary authentication server through which user has performed login attempt.
- source_ip: **optional** (*str*) Source IP of the device through which login was attempted. This can be either IPv4 or fully expanded IPv6 address.
- mac_address: **required** (*str*) MAC address of the device through which login was attempted.
- device_id: **optional** (*str*) Specifies a unique identifier to identify the endpoint.
- device_os: **optional** (*str*) Operating system of device through which login was attempted. Current possible values are given below. More could be added to this list in future.
    - Unknown
    - Android
    - iOS
    - Blackberry
    - Windows_XP
    - Windows_Vista
    - Windows_7
    - Windows_8
    - Mac_OS
    - Linux
    - Other_OS
    - Windows_8_1
    - Windows_10
    - Chrome_OS
    - Windows
    - Mac_OS_10_8
    - Mac_OS_10_9
    - Mac_OS_10_10
    - Mac_OS_10_11
    - Mac_OS_10_12
    - Mac_OS_10_13 ```
- compliance: **optional** (*str* or *null*) The result of compliance check. This value can be updated more than once between a session start to end. Possible values are: ```
    - COMPLIANT_YES
    - COMPLIANT_NO
    - COMPLIANT_REMEDIATED
    - COMPLIANT_NOT_ASSESSED ```
- first_host_check_succeeded: **optional** (*boolean*) Specifies whether host check was in success state at the time of login attempt or session establishment. This value does not change during a session, even if compliance changes after session establishment. Possible values are: ```
    - True - Host check status successful
    - False - Host check status failed ```
- first_host_check_time: **optional** (*str* or *null*) An RFC 3339 timestamp in UTC of the time at which first host check was performed for this user session. This value does not change during a session.
- first_host_check_failed_policies: **optional** (*list of str* or *null*) List of failed host-checker policy names when first host check was performed for this user session. This value does not change during a session.
- first_host_check_failure_reasons: **optional** (*list of str* or *null*) List of host-checker failure reasons when first host check was performed for this user session. This value does not change during a session.

UserAccessRecordsEntity

Represents a list of UserAccessRecordEntity JSON objects

- total: **required** (*int*) Total number of UserAccessRecordEntity entities
- items: **required** (*array*) JSON array of UserAccessRecordEntity entities

UserAccessSummariesEntity

Represents a list of UserAccessSummaryEntity entities

- total: **required** (*int*) Total number of UserAccessSummaryEntity entities
- items: **required** (list of *UserAccessSummaryEntity*)

UserAccessSummaryEntity

This entity defines properties of an appliance user's sign-in summary

- username: **required** (*str*) Username of the user whose sign-in summary is represented by this entity.
- last_appliance_id: **required** (*UUID*) ID of *SecurityApplianceEntity* from which user has last logged-in or attempted to login.
- last_login_roles: **optional** (*list of str*) Roles that were assigned to this user during last login.
- last_login_realm: **optional** (*str*) The realm through which user performed last login attempt.
- last_login_time: **required** (*str*) An RFC 3339 timestamp in UTC of the time at which user performed last login.
- last_login_ip: **required** (*str*) Source IP of the device through which last login was attempted. This can be either IPv4 or fully expanded IPv6 address.
- successful_logins: **required** (*int*) Number of successful logins by this user.
- failed_logins: **required** (*int*) Number of failed logins by this user.
- compliant_sessions: **required** (*int*) Number of compliant sessions from this user.
- noncompliant_sessions: **required** (*int*) Number of non-compliant sessions from this user.
- remediated_sessions: **required** (*int*) Number of remediated sessions from this user.
- total_sessions_length: **required** (*int*) Total length of sessions by this user (in seconds)
- average_sessions_length: **required** (*int*) Average length of sessions by this user (in seconds)

UserEntity

A user may have a workspace, use the console (as an administrator), or both. Common fields for all users:

- id: (str) The user id. This is a UUID.
- created_on: (str) Timestamp of user creation date.
- full_name: (str) The full name of the user.
- locked: (boolean) Whether or not the user is locked.
- modified_on: (*str) Timestamp of user modification date.
- roles: (list of RoleEntity.List) The roles this user is associated with. Currently only one role is supported at a time so this list will have exactly one role.
- username: (str) The username of the user.
- workspace_email: (str) The primary email associated with the user. While this says "workspace", this email is used when sending emails to the user for other purposes such as password reset.
- sign_in_method: (str) How the user is authenticated when signing into the console. local_password or saml.

UserSigninHistoryRecordEntity

This entity defines properties of sign-in history of a specific user who might have signed in to different appliances at different times. The properties of this entity includes all properties of UserAccessRecordEntity, with *addition* of following fields:

- appliance_id: **required** (*UUID*) Unique ID of the appliance from which user performed sign-in attempt
- device_profiled_status: **required** (*boolean*) Indicates whether the device (MAC address) from which user has attempted sign-in was profiled by any registered PPS (profiler) or not. Possible values are: ```
  - True - Pulse One has a profile of this device from a profiler
  - False - Pulse One does not have profile of this device from any profiler `

UserSigninHistoryRecordsEntity

Represents a list of UserSigninHistoryRecordEntity entities

- total: **required** (*int*) Total number of UserSigninHistoryRecordEntity entities
- items: **required** (list of *UserSigninHistoryRecordEntity*)

VpnCertificateEntity

- certificate: (*str*) The base64 encoded pkcs12 certificate for VPN connection
- cert_alias: (*str*) The alias for the VPN certificate
- password: (*str*) The password for pkcs12 certificate
- format: (*str*) The format of the certificate. Currently, it is "pkcs12"
- version: (*str*) The version of the certificate in sha256 hash format

VpnCertificateEntity.Update

- version: (*str*) The version of certificate which is currently installed in client

WorkspaceAppDetailsEntity

The entity has all the fields that AppRuleEntity has plus the following fields:

- installation_status: (*str*) Installation status of an app. One of:
- installed: The app has been installed.
- available: The app is available to be installed.
- pending: The app has been requested to be installed.
- installed_on: (*str*) An RFC 3339 timestamp in UTC of when the app was installed. The field will be not included if the app has not been installed.
- description: (*str*) The description of the app. If the app was added before Del Mar release, the entity does not have this field.

WorkspacePolicyEntity

- app_rules: (*array of AppRuleEntity*) A list of app rule definitions.
- policy_version: (*str*) Unique identifier of this policy. Used to determine if there was any change.
- properties: (*FlexEntity*) Set of policy property name/value pairs which can be set in the console.
- settings: (*PolicySettingsEntity*) Server generated settings.
- email_configuration: (*EmailConfigurationEntity*) Email configuration.
- vpn_ondemand: (*VpnOnDemandConfigurationEntity*) VPN OnDemand configuration.

WorkspaceRegistrationRequestEntity

- user_email: (*str*) The email used for registering workspace.
- reg_key: (*str*) The registration key used for registering workspace.

WorkspaceState

- workspace_state: (*str*) The provisioning state of a workspace.
- enrolling: The device is registering and enrolling in the system.
- policy_pending: The workspace is pending a new policy, but has not started installing it yet.
- policy_installing: The workspace is installing a new policy.
- policy_current: The workspace has fully applied a policy.
- wiping: The workspace is responding to a command to wipe itself.
- wiped: The workspace is done wiping.
- locking: The workspace is locking itself.
- locked: The workspace has completed locking itself.
- unlocking: The workspace is unlocking itself. After unlocking, the workspace will be CURRENT.
- pw_failed_lock: The workspace has locked because of the number of failed password attempts.
- delta: (*boolean*)
- True: This message only lists modified apps.
- False: This message lists all known apps in the policy and in the workspace.
- app_states: (*list of WorkspaceAppState*)

API Errors

Each error will be accompanied by an Error Code. Errors via HTTP will also include an HTTP Status Code in the HTTP status header.

The Error Code is comprised of a signed integer that starts with its corresponding HTTP Status Code followed by a unique number (example: int('403' + '01') = 40301).

Below, each error includes the HTTP Status Code, Error Code and Description. In some cases, the Error Code simply has "##" to indicate a range of status codes where "##" will be some two-digit number. These are category error codes that match up with standard HTTP Status Codes.

General Errors

- 400, 400## - Bad Request -- Indicates that something was wrong with the request format.
- 400, 40001 - Too Many Items -- Too many items were provided in this request. The request can be repeated with a smaller number of items.
- 500, 500## - Internal Server Error -- There was an error on the server, retry the request with an exponential back-off.

Authentication Errors

- 401, 400## - Unauthorized -- Indicates that the resource requires authentication but no Authentication header was provided.
- 401, 40101 - Credential Renewal Required -- The provided credentials must be renewed.
- 403, 403## - Forbidden -- Indicates that the supplied credentials are not authorized to access this resource.
- 403, 40301 - Signature Does Not Match -- The request signature we calculated does not match the signature you provided.
- 403, 40302 - Request Time Too Skewed -- The difference between the request time and the server's time is too large.
- 403, 40303 - Credentials Expired -- The provided API credentials have exceeded their expiration time and grace period.
- 403, 40304 - Registration Code Invalid -- The provided registration code is invalid. It may be intended for a different domain, or expired.
- 403, 40305 - Invalid Credential Claim -- The provided credentials have an invalid claim.
- 403, 40309 - Activesync token Error -- Activesync token does not match with the one calculated from data.

Configuration Errors

- 400, 40002 - Hash Mismatch Error -- The sha1 id does not match with the one calculated from data.
- 400, 40003 - Group Configuration Error -- Master Appliance conflict.
- 400, 40004 - Group Configuration Error -- Target Appliance conflict.
- 404, 40401 - Config Block Not Found -- A block(s) in the commit is invalid.
- 409, 40901 - Config Commit Conflict - A commit is in conflict with the actual commit and cannot be promoted to the pending commit.

AfW Errors

- 404, 40410 - The google device id is not found by Google AfW service.
- 500, 50001 - The Google SafetyNet verification call failed
- 502, 50202 - The Google API is not available.
- 502, 50203 - Unexpected Google Response, try again later. This happens occasionally and we don't know why. Typically this will be a scenario where the Google library makes a request and returns a None response, which is not what we expect. The Google library should be returning a Response or an error. We cannot do much in this case other than retry later. We don't get any information about why the request failed or didn't happen.
- 400, 40005 - Attempt to use the google-account/authentication-token API with the wrong type of enterprise

iOS Errors

- 500, 50002 - Mdm push certificate is not configured for the domain
- 502, 50201 - The iTunes API is not available.
- User Settings Errors
- 409, 40902 - Version Conflict - New settings' previous version value is different from the current settings' version. They need to match for new settings to replace the current settings.
- 409, 40908 - IP Out of Range - In order to prevent unintended lockouts, we check that the current user's IP is within the whitelisted IP address ranges.
- 400, 40014 - Invalid IP Address Range - Not a valid IPv4 or IPv6 address range.

Requesting Technical Support

Technical product support is available through the Pulse Secure Global Support Center (PSGSC). If you have a support contract, then file a ticket with PSGSC.

- Product warranties—for product warranty information, visit https://www.pulsesecure.net.