Workspace ONE Notifications Service Guide

[V1, May 7, 2019]

Overview

The Workspace ONE Notifications Service is a robust, flexible cloud-hosted service designed to generate and serve actionable, real-time notifications in compliance with the Hero card specifications. It enables a wide variety of powerful use cases to provide an engaging user experience within the Workspace ONE Intelligent Hub. Some use cases are:

1. **New apps** - Out of the box with Intelligent Hub we provide New App Notifications. In this scenario, the end-user is notified when new apps are entitled to them by the admin. From the notification, the user can conveniently navigate to a list view of all their new apps in one place and can choose to view more details. This way the end user is made aware of new apps that have been assigned to them so they can start using them right away.
2. **Reminders** - Urgent reminders with or without a call to action. For example, we can create a notification to remind the user that their password is expiring soon and provide a link to reset it. Similarly, we can direct the user to surveys, Terms of Use updates etc.
3. **Emergency Alerts** - Emergency alerts such as natural disaster warnings nearby or other urgent messages.

Authentication

Creating a service client

You need to create a service client on Identity Manager for your app to get the access token that you need to create notifications. This is a one-time setup. In the VMware Identity Manager admin UI, go to `Catalog -> Settings` Click on the Remote App Access menu on the left-hand side and click `Create Client` * Select `Service Client Token` as the `Access Type` * Enter a `Client ID` * Click `Add`

The **client secret** is generated and is displayed on the UI.

Getting an access token

You will need to acquire a new access token whenever the current token expires. By default, an access token is valid for 6 hours but that can be configured to a different value too. To acquire an access token, we use the following API:

```
POST /SAAS/auth/oauthtoken
Host: acme.vmwareidentity.com
Body: grant_type=client_credentials
Authorization: Basic <token>
Content-Type: application/x-www-form-urlencoded
```

Here `<token>` is obtained by **base64** encoding of `client_id:client_secret` from the previous step.

For example: if, `client id = test-notification` and `client secret = QdW9baluUl9zUiadOjXuzu8F8fyE1Qs` Then the **base 64** encoding of `test-notification:dQdW9baluUl9zUiadOjXuzu8F8fyE1Qs` gives `dGVzdC1ub3RpZmljYXRpb246ZFFkVzliYWx1VWw5elVpYWRPalh1enU4RjhmeUUxUXPCoA==`

More information can be found here: (https://github.com/vmware/idm/wiki/Integrating-Client-Credentials-app-with-OAuth2)

Create a new notification

The notification service API can be used to create new notifications for a particular user on a specific device and/or tenant.

To get the user ID of a particular user, see here.

Create a notification for a user

To create a new notification for a user with user ID = {**targetUserId**}

```
POST: /ws1notifications/api/v1/users/{targetUserId}/notifications
Host: acme.vmwareidentity.com
Content-Type: application/json
Authorization: Bearer <Access-Token>
```

Body

See Guidelines for creating user friendly notifications

Response

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

{
    "created_at": "2018-04-02T08:03:07.71Z",
    "updated_at": "2018-04-02T08:03:07.71Z",
    "id": "2cad8515-c6db-446a-b543-390bac2e67fc",
    "tenant_id": "greenbox-ui",
    "user_id": "7cd85d9a-c742-43ef-90dc-2d48d3b85539",
    "creator_id": "70d95845-4e5f-4567-b515-e0333a5b3ce2",
    "device_id": null,
    "read_at": null,
    "last_action_id": null,
    "notification_card": {
        "header": {
            "title": "This is a title"
        },
        "body": {
            "description": "This is a description"
        }
    }
}
```

Create a notification for multiple users (Synchronous)

This is an API endpoint to create a new notification for multiple users. This endpoint synchronously creates notifications for up to 100 users. To create notifications for more than 100 users, we use the asynchronous endpoint.

For example, to create a notification for user1 with userID = "abc123" and user2 with userID = "xyz456", we use

```
POST: /ws1notifications/api/v1/distributed_notifications
Host: acme.vmwareidentity.com
Content-Type: application/json
Authorization: Bearer <Access-Token>
```

Body

```
{
  "notification_card": {
   "header":{
```

```
      "title":"This is a title"
   },
   "body":{
      "description":"This is a description"
   }
},
  "user_ids": [
    "abc123",
    "xyz456"
  ]
}
```

Response

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

{
    "abc123": {
        "status_code": "200",
        "notification_id": "023a5977-cf8e-4bdd-b120-ded901ec34a8"
    },
    "xyz456": {
        "status_code": "200",
        "notification_id": "105cfeac-9e4f-4219-b6ba-a72ede3e52d9"
    }
}
```

Create a notification for multiple users (Asynchronous)

This is an API endpoint to create a new notification for multiple users. This endpoint asynchronously creates notifications for a very large number of users.

For example, to create a notification for user1 with userID = "abc123" and user2 with userID = "xyz456", we use

```
POST: /ws1notifications/api/v1/distributed_notifications_async
Host: acme.vmwareidentity.com
Content-Type: application/json
Authorization: Bearer <Access-Token>
```

Body

```
{
  "notification_card": {
   "header":{
      "title":"This is a title"
   },
```

```
   "body":{
      "description":"This is a description"
   }
},
  "user_ids": [
    "abc123",
    "xyz456"
  ]
}
```

Response

```
HTTP/1.1 200 OK
```

Create a notification for a user on a specific device

To create a new notification for a user with user ID = {**targetUserId**} and device ID = {**targetDeviceId**}

To get a specific device ID of a particular user, see here.

```
POST
/ws1notifications/api/v1/users/{targetUserId}/devices/{targetDeviceId}/notifi
cations
Host: acme.vmwareidentity.com
Content-Type: application/json
Authorization: Bearer <Access-Token>
```

Body

See Guidelines for creating user friendly notifications

Response

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

Guidelines for creating user friendly notifications

The notification card (the body of the create notification request) should be in accordance with the Hero card specifications as well as some conventions that are specific to Intelligent Hub as described below.

Here are some guidelines for creating effective notifications that enable a great user experience on Intelligent Hub!

Atleast one of the following attributes are **required** on the cards:

1. **header.title**

2. **header.subtitle**
3. **body.description**

This is a valid (albeit not very useful) notification.

```
{
    "header":{
       "title":"A notification with a header"
    }
}
```

1. Write short titles that catch the user's attention. The description message should be clear and concise.
2. At most three actions should be present on the notification card. One or two actions is the ideal user experience. Any more than three and the user experience degrades as the individual buttons will not be wide enough to accomodate even medium length action labels.
3. Keep action labels short to avoid truncation.
4. If we want the user to be able to repeat the action, we set `"allow_repeated" : true` else `"allow_repeated" : false` if we want the user to be able to perform that action only once.
5. The action buttons are rendered as either primary or secondary actions. We can set `"primary": true` if we want the action to be rendered as a primary action or `"primary": false` if we want it to be a secondary action as shown:

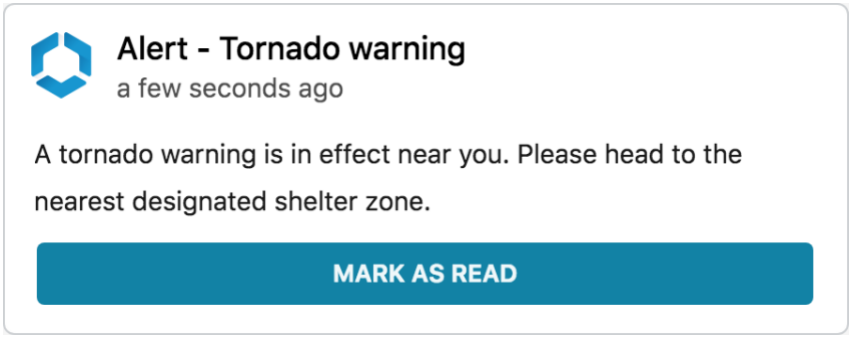
Shown below are notifications illustrating a few common use cases:

No-action announcement or alert

To create an unactionable announcement, notice or alert, we structure the notification body as follows:

```
{
    "header":{
       "title":"Alert - Tornado warning"
    },
    "body":{
       "description":"A tornado warning is in effect near you. Please head to
the nearest designated shelter zone."
       }
}
```

which looks like



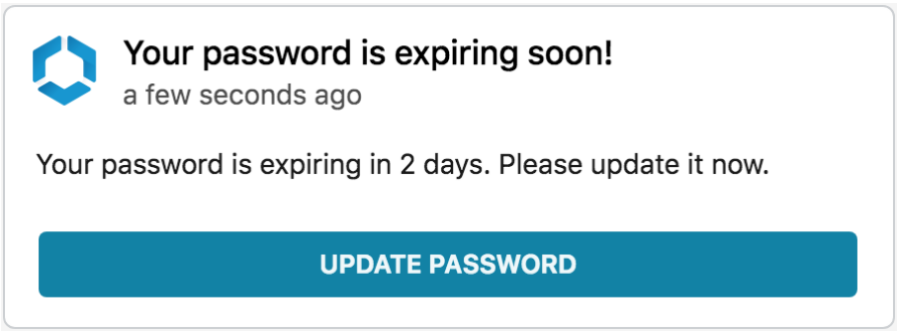
Single action notification

To create a notification with a single action such as a prompt for the user to change their password, we structure the body as follows:

```
{
    "header":{
       "title":"Your password is expiring soon!"
    },
    "body":{
       "description":"Your password is expiring in 2 days. Please update it
now."
    },
    "actions":[{
          "id":"a8406d97-2f46-45a8-9e3d-f68dac0cdf18",
          "label":"Update password",
          "completed_label": "Password updated",
          "type":"POST",
          "primary": true,
          "allow_repeated": false,
          "url":{
             "href":"https://password-change-url.vmwareidentity.com"
```

```
            },
            "action_key":"OPEN_IN"
        }]
}
```

which looks like



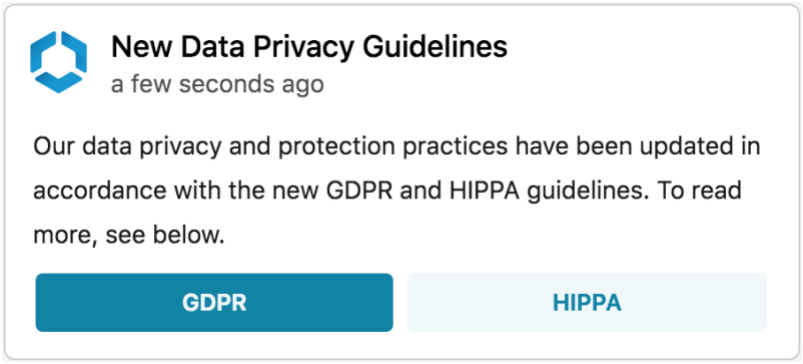
Multiple action notification

To create a notification with multiple actions, we structure the body as follows:

```
{
    "header":{
       "title":"New Data Privacy Guidelines"
    },
    "body":{
       "description":"Our data privacy and protection practices have been
updated in accordance with the new GDPR and HIPPA guidelines. To read more,
see below."
    },
    "actions":[{
          "id":"a8406d97-2f46-45a8-9e3d-f68dac0cdf18",
          "label":"GDPR",
          "type":"POST",
          "primary": true,
          "allow_repeated": false,
          "completed_label": "Done",
          "url":{
             "href":"https://gdpr.com"
          },
          "action_key":"OPEN_IN"
       },{
          "id":"b8406d97-2f46-45a8-9e3d-f68dac0cdf18",
          "label":"HIPPA ",
          "type":"POST",
          "primary": false,
          "allow_repeated": false,
          "completed_label": "Done",
          "url":{
```

```
            "href":"https://hippa.com"
         },
         "action_key":"OPEN_IN"
      }]
}
```

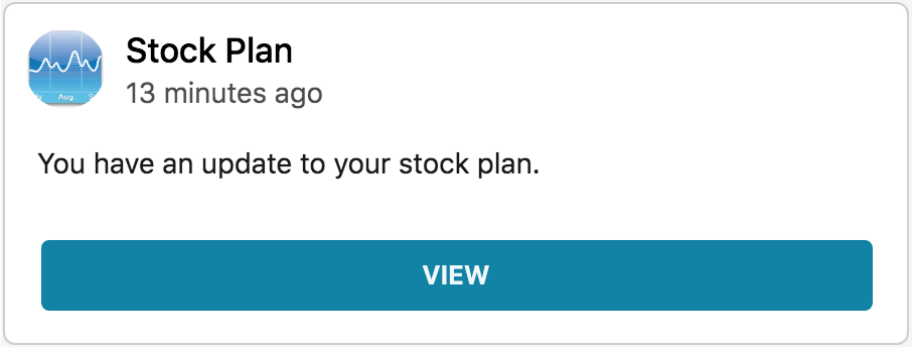
which looks like



Notification with a custom icon

To create a notification with a custom image as the icon, we add an 'image' property to the card.

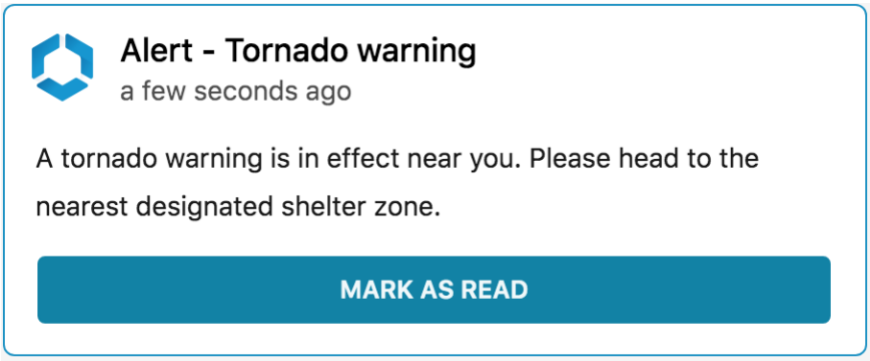
```
{
    "header":{
       "title":"Stock Plan"
    },
    "body":{
       "description":"You have an update to your stock plan."
    },

    "image":{
       "href":"https://acme.com/stock.png"
    },
    "actions":[{
          "id":"a8406d97-2f46-45a8-9e3d-f68dac0cdf11",
          "label":"View",
          "type":"POST",
          "primary": true,
          "allow_repeated": true,
          "completed_label": "View",
          "url":{
             "href":"https://stock-plan.acme.com"
          },
          "action_key":"OPEN_IN"
       }]
}
```



Priority notifications

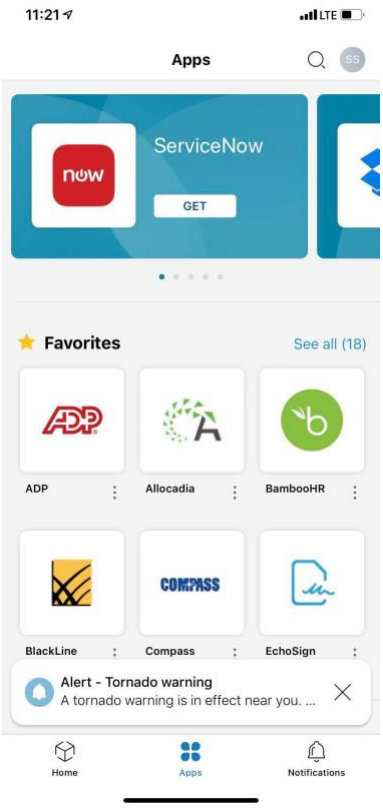
To mark a notification as high priority, we add an 'importance' property in the body of the card and give it a value of 1 as follows:

```
{
    "header":{
       "title":"Alert - Tornado warning"
    },
    "importance" : 1,
    "body":{
       "description":"A tornado warning is in effect near you. Please head to
the nearest designated shelter zone."
       }
}
```



By marking a card as priority, the user sees it in the priority section of the notifications feed with a highlighted border and the notification count does not reduce until the priority card is marked as read or acted on. In addition, if the user is on a different view of Intelligent Hub and a priority notification is received, they are alerted in the form of a message near

the bottom of the screen which brings the user's attention to the incoming priority notification.



View a notification

To view the details of a notification with ID = {**notification-id**}

```
GET /ws1notifications/api/v1/notifications/{notification-id}
Host: acme.vmwareidentity.com
Authorization: Bearer <Access-Token>
```

Response

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

{
    "created_at": "2018-07-13T07:25:26.854Z",
    "updated_at": "2018-07-13T07:25:26.854Z",
    "id": "bd6d0c3d-5152-489b-b0b0-11168133f634",
    "tenant_id": "acme",
    "user_id": "51e01723-9a10-416e-82f3-e151cabc3770",
    "creator_id": "69561a9e-6968-4e1d-8611-6622d0a4ae92",
```

```
    "device_id": null,
    "read_at": null,
    "last_action_id": null,
    "notification_card": {
        "header": {
            "title": "This is a title"
        },
        "body": {
            "description": "This is a description"
        }
    }
}
```

Update a notification

The Notifications API supports updating a notification with new content.

To update a notification with ID = {**notification-id**}

```
POST /ws1notifications/api/v1/notifications/{notification-id}
Host: acme.vmwareidentity.com
Content-Type: application/json
Authorization: Bearer <Access-Token>
```

Body:

```
{
   "header":{
      "title":"This is the updated title"
   },
   "body":{
      "description":"This is the updated description"
   }
}
```

Response:

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

Below is the response to update a notification with notification ID = 2cad8515-c6db-446a-b543-390bac2e67fc

```
{
    "created_at": "2018-04-02T08:03:07.71Z",
    "updated_at": "2018-04-02T08:11:08.165Z",
    "id": "2cad8515-c6db-446a-b543-390bac2e67fc",
    "tenant_id": " acme ",
    "user_id": "7cd85d9a-c742-43ef-90dc-2d48d3b85539",
    "creator_id": "70d95845-4e5f-4567-b515-e0333a5b3ce2",
    "device_id": null,
    "read_at": null,
```

```
    "last_action_id": null,
    "notification_card": {
        "header": {
            "title": "This is the updated title"
        },
        "body": {
            "description": "This is the updated description"
        }
    }
}
```

Delete a notification

The Notifications API supports deleting a notification. To delete a notification with ID = {**notification-id**}

```
DELETE /ws1notifications/api/v1/notifications/{notification-id}
Host: acme.vmwareidentity.com
Authorization: Bearer <Access-Token>
```

Response: `HTTP/1.1 204 No Content`

Additional Information

Getting the user id of a specific user

To get the user ID of a user with username = {**userName**}

```
GET
/SAAS/jersey/manager/api/scim/Users?filter=userName%20eq%20%22{userName}%22
Host: acme.vmwareidentity.com
Authorization: Bearer <Access-Token>
Accept: application/json
```

Response

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

{
    "totalResults": 1,
    "itemsPerPage": 1,
    "startIndex": 1,
    "schemas": [
        "urn:scim:schemas:core:1.0",
        "urn:scim:schemas:extension:workspace:1.0",
        "urn:scim:schemas:extension:enterprise:1.0",
        "urn:scim:schemas:extension:workspace:mfa:1.0",
        "urn:scim:schemas:extension:workspace:tenant:greenbox-ui:1.0"
    ],
    "Resources": [
```

```
        {
            "active": true,
            "userName": "user-name",
            "id": "51e01723-9a10-416e-82f3-e151as342bc3770",
            "meta": {
                "created": "2018-07-10T18:17:55.335Z",
                "lastModified": "2018-07-10T18:19:21.995Z",
                "location":
"https://acme.vmwareidentity.com/SAAS/jersey/manager/api/scim/Users/51e01723-
9a10-416e-82f3-e151as342bc3770",
                "version": "W/\"1531246761995\""
            },
            "name": {
                "givenName": "First Name",
                "familyName": "Last Name"
            },
            "emails": [
                {
                    "value": "useremail@domain.com"
                }
            ],
            "groups": [
                {
                    "value": "337f36b2-7550-49a5-a1d0-121da939d94b",
                    "type": "direct",
                    "display": "ALL USERS"
                }
            ],
            "roles": [
                {
                    "value": "04f574c0-06ac-4063-85a7-2ad0635b5c6d",
                    "display": "User"
                }
            ],
            "urn:scim:schemas:extension:workspace:1.0": {
                "internalUserType": "LOCAL",
                "userStatus": "1",
                "domain": "System Domain",
                "userStoreUuid": "487faa2c-e9d1-4939-bc70-26f79faf7f29"
            }
        }
    ]
}
```


Getting the device id of a specific user

This Airwatch API can be used to get the device IDs of a user with username = **userName**

The auth token is the base64 encoding of `admin-username:admin-password`

For example if the Airwatch admin username for the tenant is 'admin' and the admin password is 'password', then the auth `Token` is the base64 encoding of admin:password which is `YWRtaW46cGFzc3dvcmQ=`

We also need an admin API key of the Airwatch tenant which can be found in the Airwatch console under `Settings > System > Advanced > API`

```
GET /api/mdm/devices/search?user=userName
Host: <airwatch-tenant-url>
Authorization: Basic <Token>
Accept: application/json
aw-tenant-code: <tenant-admin-api-key>
```

Similary using this API endpoint we can also get all the device IDs of a certain device **model** (iPhone, Galaxy) or **platform** (iOS, Android)

To search by model or platfrom, we use '?model=iPhone' or '?platform=Android' instead of the '?user=userName' query string parameter at the end of the url.

Response

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

{
  "Devices": [
    {
      "EasIds": {
        "EasId": [
          "Text value"
        ]
      },
      "Udid": "027BE1C5AEC05C378C61C44103E9D3FCB2EC354D",
      "SerialNumber": "R51G844T90R",
      "MacAddress": "485A3F880798",
      "Imei": "356766060039613",
      "EasId": "6Q93UFOQ7H0K39JPMFPTEMJQ3K",
      "AssetNumber": "827BE1C5AEC05C378C61C44103E9D3FCB2EC354D",
      "DeviceFriendlyName": "users iPhone iOS 10.3.2 ",
      "LocationGroupId": {
        "Id": {
          "Value": 1
        },
        "Uuid": "5741a4a4-ece3-4f18-b0b0-28a3b2017b2e",
        "Name": "Text value"
      },
      "LocationGroupName": "locationgroup1",
      "UserId": {},
      "UserName": "userName",
      "UserEmailAddress": "userName@vmware.com",
      "Ownership": "C",
```

```
      "PlatformId": {},
      "Platform": "Apple",
      "ModelId": {},
      "Model": "iPhone",
      "OperatingSystem": "10.3.2",
      "PhoneNumber": "+14045550100",
      "LastSeen": "2018-07-16T11:41:11.3254194-04:00",
      "EnrollmentStatus": "Enrolled",
      "ComplianceStatus": "Compliant",
      "CompromisedStatus": true,
      "LastEnrolledOn": "2018-07-16T11:41:11.3254194-04:00",
      "LastComplianceCheckOn": "2018-07-16T11:41:11.3254194-04:00",
      "LastCompromisedCheckOn": "2018-07-16T11:41:11.3254194-04:00",
      "ComplianceSummary": {
        "DeviceCompliance": [
          {
            "CompliantStatus": true,
            "PolicyName": "application list compliance policy",
            "PolicyDetail": "compliance policy for device compromised status
including application list contains rule",
            "LastComplianceCheck": "2018-07-16T11:41:11.3364487-04:00",
            "NextComplianceCheck": "2018-07-16T11:41:11.3364487-04:00",
            "ActionTaken": [
              {
                "ActionType": 0
              }
            ],
            "Id": {
              "Value": 0
            },
            "Uuid": "74f8db15-ecca-4782-afaf-3b590907b7a7"
          }
        ]
      },
      "IsSupervised": true,
      "DeviceMCC": {
        "SIMMCC": "404",
        "CurrentMCC": "310"
      },
      "IsRemoteManagementEnabled": "abcd",
      "DataEncryptionYN": "Y",
      "AcLineStatus": 1,
      "VirtualMemory": 2,
      "OEMInfo": "abcd",
      "DeviceCapacity": 1,
      "AvailableDeviceCapacity": 1,
      "LastSystemSampleTime": "2018-07-16T11:41:11.3434692-04:00",
      "IsDeviceDNDEnabled": true,
      "IsDeviceLocatorEnabled": true,
      "IsCloudBackupEnabled": true,
```

```
      "IsActivationLockEnabled": true,
      "IsNetworkTethered": true,
      "BatteryLevel": "abcd",
      "IsRoaming": true,
      "LastNetworkLANSampleTime": "2018-07-16T11:41:11.344471-04:00",
      "LastBluetoothSampleTime": "2018-07-16T11:41:11.344471-04:00",
      "SystemIntegrityProtectionEnabled": true,
      "ProcessorArchitecture": 5,
      "UserApprovedEnrollment": true,
      "EnrolledViaDEP": true,
      "TotalPhysicalMemory": 3,
      "AvailablePhysicalMemory": 4,
      "Id": {
        "Value": 0
      },
      "Uuid": "fa1451be-9b6e-4e15-85fe-b2e884cf556a"
    }
  ],
  "Page": 1,
  "PageSize": 2,
  "Total": 3
}
```