

# **Intel® Endpoint Management Assistant (Intel® EMA)**

Version 2 Application Programming Interface Guide

Intel® EMA Version 1.3.2

October 2019

## Legal Disclaimer

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses. Check with your system manufacturer or retailer or learn more at <http://www.intel.com/technology/vpro>.

Intel, Intel vPro, Intel EMA, Intel AMT, and the Intel logo, are trademarks of Intel Corporation in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2019 Intel Corporation.

# Contents

|                                                                         |          |
|-------------------------------------------------------------------------|----------|
| <b>INTRODUCTION .....</b>                                               | <b>1</b> |
| <i>Solution Overview.....</i>                                           | <i>1</i> |
| <b>1 API INTRODUCTION .....</b>                                         | <b>1</b> |
| <b>2 AUTHENTICATION AND AUTHORIZATION.....</b>                          | <b>1</b> |
| 2.1 Authentication.....                                                 | 1        |
| 2.2 Authorization.....                                                  | 2        |
| <b>3 INTEL® EMA API URI DETAILS .....</b>                               | <b>2</b> |
| 3.1 802.1X SETUPS.....                                                  | 3        |
| GET /api/v2/802_1XSetups .....                                          | 3        |
| POST /api/v2/802_1XSetups .....                                         | 3        |
| GET /api/v2/802_1XSetups/{_802_1XSetupId}.....                          | 6        |
| PUT /api/v2/802_1XSetups/{_802_1XSetupId}.....                          | 9        |
| DELETE /api/v2/802_1XSetups/{_802_1XSetupId}.....                       | 12       |
| GET /api/v2/802_1XSetups/CertificateAuthorities.....                    | 12       |
| 3.2 ACCESS TOKENS .....                                                 | 13       |
| GET /api/v2/accessTokens/getUsingExistingToken.....                     | 13       |
| GET /api/v2/accessTokens/getUsingWindowsCredentials .....               | 14       |
| 3.3 AGENTS.....                                                         | 15       |
| GET /api/v2/agents/getWin32Console.....                                 | 15       |
| GET /api/v2/agents/getWin32Service.....                                 | 15       |
| GET /api/v2/agents/getWin64Console.....                                 | 15       |
| GET /api/v2/agents/getWin64Service.....                                 | 16       |
| 3.4 AJAX COOKIES.....                                                   | 16       |
| GET /api/v2/ajaxCookies.....                                            | 16       |
| 3.5 INTEL® AMT CREDENTIALS.....                                         | 17       |
| GET /api/v2/amtCredentials/{endpointId} .....                           | 17       |
| 3.6 INTEL® AMT DISCOVERIES .....                                        | 17       |
| POST /api/v2/amtDiscoveries/discoverByIP .....                          | 17       |
| POST /api/v2/amtDiscoveries/discoverByIPAutomatically.....              | 18       |
| GET /api/v2/amtDiscoveries/{amtDiscoveryId}.....                        | 19       |
| GET /api/v2/amtDiscoveries.....                                         | 20       |
| POST /api/v2/amtDiscoveries/{amtDiscoveryId}/cancel .....               | 22       |
| POST /api/v2/amtDiscoveries/discoverBySubnet.....                       | 22       |
| POST /api/v2/amtDiscoveries/discoverBySubnetAutomatically.....          | 23       |
| POST /api/v2/amtDiscoveries/discoverByIPRange.....                      | 24       |
| POST /api/v2/amtDiscoveries/discoverByIPRangeAutomatically.....         | 24       |
| POST /api/v2/amtDiscoveries/discoverByLocalBroadcast.....               | 25       |
| POST /api/v2/amtDiscoveries/discoverByLocalBroadcastAutomatically ..... | 26       |
| 3.7 INTEL® AMT PROFILES.....                                            | 27       |
| GET /api/v2/amtProfiles.....                                            | 27       |
| POST /api/v2/amtProfiles.....                                           | 27       |
| GET /api/v2/amtProfiles/{amtProfileId}.....                             | 31       |
| PUT /api/v2/amtProfiles/{amtProfileId}.....                             | 34       |
| DELETE /api/v2/amtProfiles/{amtProfileId} .....                         | 35       |
| 3.8 INTEL® AMT PROFILE TO ENDPOINT GROUPS ASSIGNMENTS .....             | 35       |
| GET /api/v2/amtProfileToEndpointGroupsAssignments/{amtProfileId} .....  | 35       |
| 3.9 INTEL® AMT PROVISIONING CERTIFICATES.....                           | 36       |
| GET /api/v2/amtProvisioningCertificates.....                            | 36       |
| GET /api/v2/amtProvisioningCertificates/{amtCertificateId}.....         | 37       |
| DELETE /api/v2/amtProvisioningCertificates/{amtCertificateId} .....     | 38       |
| GET /api/v2/amtProvisioningCertificates/{amtCertificateId}/getFile..... | 39       |
| POST /api/v2/amtProvisioningCertificates/uploadPfx .....                | 39       |
| POST /api/v2/amtProvisioningCertificates/upload .....                   | 40       |

|                                                                              |     |
|------------------------------------------------------------------------------|-----|
| 3.10 INTEL® AMT SETUPS .....                                                 | 42  |
| GET /api/v2/amtSetups/endpoints/{amtSetupId} .....                           | 42  |
| DELETE /api/v2/amtSetups/endpoints/{amtSetupId} .....                        | 44  |
| GET /api/v2/amtSetups/auto/{amtSetupId} .....                                | 44  |
| DELETE /api/v2/amtSetups/auto/{amtSetupId} .....                             | 46  |
| POST /api/v2/amtSetups/endpoints/provision .....                             | 47  |
| POST /api/v2/amtSetups/endpoints/unprovision .....                           | 49  |
| POST /api/v2/amtSetups/auto/set .....                                        | 50  |
| 3.11 ENDPOINT GROUPS .....                                                   | 52  |
| GET /api/v2/endpointGroups .....                                             | 52  |
| POST /api/v2/endpointGroups .....                                            | 53  |
| GET /api/v2/endpointGroups/{endpointGroupId} .....                           | 55  |
| PUT /api/v2/endpointGroups/{endpointGroupId} .....                           | 57  |
| DELETE /api/v2/endpointGroups/{endpointGroupId} .....                        | 58  |
| GET /api/v2/endpointGroups/{endpointGroupId}/getMshFile .....                | 58  |
| GET /api/v2/endpointGroups/{endpointGroupId}/getUserAccessRights .....       | 59  |
| 3.12 ENDPOINT IB OPERATIONS .....                                            | 59  |
| POST /api/v2/endpointIBOperations/reboot .....                               | 59  |
| POST /api/v2/endpointIBOperations/SLEEP .....                                | 61  |
| POST /api/v2/endpointIBOperations/hibernate .....                            | 63  |
| POST /api/v2/endpointIBOperations/shutdown .....                             | 64  |
| POST /api/v2/endpointIBOperations/alert .....                                | 66  |
| 3.13 ENDPOINT OOB OPERATIONS .....                                           | 68  |
| POST /api/v2/endpointOOBOperations/Multiple/PowerOn .....                    | 68  |
| POST /api/v2/endpointOOBOperations/Multiple/Sleep/Light .....                | 70  |
| POST /api/v2/endpointOOBOperations/Multiple/Sleep/Deep .....                 | 71  |
| POST /api/v2/endpointOOBOperations/Multiple/PowerCycle/OffSoft .....         | 73  |
| POST /api/v2/endpointOOBOperations/Multiple/PowerOff/Hard .....              | 75  |
| POST /api/v2/endpointOOBOperations/Multiple/Hibernate .....                  | 76  |
| POST /api/v2/endpointOOBOperations/Multiple/PowerOff/Soft .....              | 78  |
| POST /api/v2/endpointOOBOperations/Multiple/PowerCycle/OffHard .....         | 80  |
| POST /api/v2/endpointOOBOperations/Multiple/MasterBusReset .....             | 81  |
| POST /api/v2/endpointOOBOperations/Multiple/PowerOff/SoftGraceful .....      | 83  |
| POST /api/v2/endpointOOBOperations/Multiple/PowerOff/HardGraceful .....      | 84  |
| POST /api/v2/endpointOOBOperations/Multiple/MasterBusReset/Graceful .....    | 86  |
| POST /api/v2/endpointOOBOperations/Multiple/PowerCycle/OffSoftGraceful ..... | 88  |
| POST /api/v2/endpointOOBOperations/Multiple/PowerCycle/OffHardGraceful ..... | 90  |
| POST /api/v2/endpointOOBOperations/Single/PowerOn .....                      | 91  |
| POST /api/v2/endpointOOBOperations/Single/Sleep/Light .....                  | 92  |
| POST /api/v2/endpointOOBOperations/Single/Sleep/Deep .....                   | 93  |
| POST /api/v2/endpointOOBOperations/Single/PowerCycle/OffSoft .....           | 93  |
| POST /api/v2/endpointOOBOperations/Single/PowerOff/Hard .....                | 94  |
| POST /api/v2/endpointOOBOperations/Single/Hibernate .....                    | 94  |
| POST /api/v2/endpointOOBOperations/Single/PowerOff/Soft .....                | 95  |
| POST /api/v2/endpointOOBOperations/Single/PowerCycle/OffHard .....           | 96  |
| POST /api/v2/endpointOOBOperations/Single/MasterBusReset .....               | 96  |
| POST /api/v2/endpointOOBOperations/Single/PowerOff/SoftGraceful .....        | 97  |
| POST /api/v2/endpointOOBOperations/Single/PowerOff/HardGraceful .....        | 98  |
| POST /api/v2/endpointOOBOperations/Single/MasterBusReset/Graceful .....      | 98  |
| POST /api/v2/endpointOOBOperations/Single/PowerCycle/OffSoftGraceful .....   | 99  |
| POST /api/v2/endpointOOBOperations/Single/PowerCycle/OffHardGraceful .....   | 99  |
| 3.14 ENDPOINTS .....                                                         | 100 |
| GET /api/v2/endpoints/{endpointId} .....                                     | 100 |
| GET /api/v2/endpoints/{endpointId}/HardwareInfoFromAmt .....                 | 102 |
| DELETE /api/v2/endpoints/{endpointId} .....                                  | 104 |
| GET /api/v2/endpoints .....                                                  | 104 |
| 3.15 NETWORK INTERFACES .....                                                | 106 |
| GET /api/v2/networkInterfaces .....                                          | 106 |
| 3.16 ROLES .....                                                             | 107 |

|                                                                                                    |            |
|----------------------------------------------------------------------------------------------------|------------|
| GET /api/v2/roles .....                                                                            | 107        |
| GET /api/v2/roles/{roleId} .....                                                                   | 107        |
| 3.17 SERVER SETTINGS .....                                                                         | 108        |
| GET /api/v2/serverSettings/amtDiscoverySettings .....                                              | 108        |
| 3.18 TENANTS .....                                                                                 | 109        |
| GET /api/v2/tenants.....                                                                           | 109        |
| POST /api/v2/tenants.....                                                                          | 110        |
| GET /api/v2/tenants/{tenantId} .....                                                               | 111        |
| PUT /api/v2/tenants/{tenantId} .....                                                               | 112        |
| DELETE /api/v2/tenants/{tenantId}.....                                                             | 113        |
| 3.19 USER GROUP MEMBERSHIPS.....                                                                   | 113        |
| GET /api/v2/userGroupMemberships/{userGroupId}.....                                                | 113        |
| POST /api/v2/userGroupMemberships/{userGroupId}/addMembers .....                                   | 114        |
| POST /api/v2/userGroupMemberships/{userGroupId}/removeMembers .....                                | 115        |
| 3.20 USER GROUPS .....                                                                             | 116        |
| GET /api/v2/userGroups.....                                                                        | 116        |
| POST /api/v2/userGroups.....                                                                       | 117        |
| GET /api/v2/userGroups/{userGroupId}.....                                                          | 118        |
| PUT /api/v2/userGroups/{userGroupId}.....                                                          | 119        |
| DELETE /api/v2/userGroups/{userGroupId}.....                                                       | 121        |
| 3.21 USER GROUP TO ENDPOINT GROUP ASSIGNMENTS .....                                                | 121        |
| GET /api/v2/userGroupToEndpointGroupAssignments/{endpointGroupId}.....                             | 121        |
| POST /api/v2/userGroupToEndpointGroupAssignments/{endpointGroupId}.....                            | 122        |
| DELETE /api/v2/userGroupToEndpointGroupAssignments/{endpointGroupId}/userGroup/{userGroupId} ..... | 123        |
| 3.22 USERS.....                                                                                    | 123        |
| GET /api/v2/users.....                                                                             | 123        |
| POST /api/v2/users.....                                                                            | 124        |
| GET /api/v2/users/{userId}.....                                                                    | 126        |
| PUT /api/v2/users/{userId}.....                                                                    | 127        |
| DELETE /api/v2/users/{userId} .....                                                                | 128        |
| GET /api/v2/users/getUserByName .....                                                              | 128        |
| POST /api/v2/users/{userId}/resetPassword .....                                                    | 129        |
| 3.23 WIFI SETUPS .....                                                                             | 130        |
| GET /api/v2/wifiSetups.....                                                                        | 130        |
| POST /api/v2/wifiSetups.....                                                                       | 131        |
| GET /api/v2/wifiSetups/{wifiSetupId}.....                                                          | 132        |
| PUT /api/v2/wifiSetups/{wifiSetupId} .....                                                         | 133        |
| DELETE /api/v2/wifiSetups/{wifiSetupId} .....                                                      | 134        |
| <b>4 TROUBLESHOOTING .....</b>                                                                     | <b>136</b> |
| <b>5 HTTP STATUS CODES .....</b>                                                                   | <b>137</b> |
| 400 Bad Request Errors .....                                                                       | 137        |
| 401 Method Not Allowed Errors .....                                                                | 141        |
| 403 Forbidden Errors .....                                                                         | 141        |
| 404 Not Found Errors.....                                                                          | 141        |
| 405 Method Not Allowed Errors .....                                                                | 142        |
| 409 Conflict Errors .....                                                                          | 142        |
| 415 Unsupported Media Type Errors.....                                                             | 144        |
| <b>APPENDIX - 802.1X SETUP USING INTEL® EMA WEB API.....</b>                                       | <b>145</b> |
| EXAMPLE 802.1X SETUP .....                                                                         | 145        |
| WiFi Setup .....                                                                                   | 146        |
| Intel® AMT Profiles .....                                                                          | 147        |
| Endpoint Group .....                                                                               | 148        |
| Install Intel® EMA Agents .....                                                                    | 148        |
| Test connections.....                                                                              | 149        |
| Add certificates to Intel® EMA database .....                                                      | 150        |
| 802.1X SETUP SETTINGS.....                                                                         | 152        |



# Introduction

## Solution Overview

This document provides an outline for developers about the Intel® Endpoint Management Assistant (Intel® EMA) Application Programming Interface (API) version 2.0.

## 1 API Introduction

The Intel® EMA API details are available in this document, as well as displayed by the installed Intel® EMA application itself. The web-based version of the API documentation is accessible at [https://<your\\_ema\\_url>/swagger](https://<your_ema_url>/swagger).

Since the API documentation is implemented with the Open API (or Swagger) specification, the detailed version of the documentation in the Swagger JSON format, viewable with a Swagger compliant editor, e.g., <https://swagger.io/swagger-editor>, and can be obtained at [https://<your\\_ema\\_url>/swagger/docs/v2](https://<your_ema_url>/swagger/docs/v2).

Code samples on how to use the API are available in the folder [Intel EMA installation package folder] \Samples.



**IMPORTANT!** These samples should **never** be hosted in a production environment.

For hosting in a test environment for development purposes, copy the Samples folder to the Intel® EMA website root folder (e.g., C:\inetpub\wwwroot\).

## 2 Authentication and Authorization

Authentication and authorization are a commonly known term for a framework that executes access control to ensure a secure environment and effective network management.

### 2.1 Authentication

All requests made to the Intel® EMA REST API are met with a bearer token challenge. The token can be obtained via the OAuth2 Resource Owner Password Credentials flow in the token path [https://<your\\_ema\\_url>/api/token](https://<your_ema_url>/api/token).

To obtain a token, you must generate an HTTP POST request to the token path using the following parameters in the message body:

- grant\_type: It must be the value **password**.
- username: The resource owner's username.
- password: The resource owner's password.

The following example illustrates a curl command line tool using HTTP POST to obtain a bearer token via the HTTP POST, replacing the placeholder values <in brackets>:

- `$ curl -k -d grant\_type=password&username=user@yourdomain.com&password=<password>`

To use the bearer token to access an Intel® EMA Uniform Resource Identifier (URI), set the token in the request header as depicted in the following example with curl:

- `curl -H "Authorization: Bearer <token>" https://<your\_ema\_url>/api/v2/<endpoint>`

In this release the token lifetime is 1 hour.

## 2.2 Authorization

To use the REST API, callers must be in a specific role required by the URI. Role-based security supports authorization by making authorization decisions based on the user's identity or role membership.

The authorization process determines whether a specific user has the necessary permissions to enforce specific commands or operations.

A role is a set of principles that are under the same umbrella of privileges, security-wise. Thus, in the case of Intel® EMA, the system uses role membership to determine whether a user is authorized to perform a requested action.

These roles are:

- **Global Administrator:** This role performs user management, tenant management, and server management. The Global Administrator does not perform endpoint management and does not (and cannot) belong to any endpoint group. The Global Administrator's control spans all tenants in a single Intel® EMA server installation instance.
- **Tenant Administrator:** This role is specific to a particular tenant and can perform all operations (user management, endpoint management, Intel® AMT Discovery) under that tenant. Therefore, the Tenant Administrator does not (and cannot) belong to any user group in its tenant. A Tenant Administrator user cannot manage a Global Administrator user.
- **Account Manager:** This role is specific to a particular tenant, and can perform user management only. However, an Account Manager cannot manage users with higher-level roles (e.g., a Tenant Administrator or Global Administrator). Account Managers cannot perform endpoint management, and therefore cannot belong to any user group.
- **Endpoint Group Creator:** This role is specific to a particular tenant. It can perform endpoint management, as well as create new endpoint groups and manage Intel® AMT Profiles. An Endpoint Group Creator can be a member of multiple user groups and can manage all groups to which they belong. Endpoint Group Creators cannot perform user management. However, they can see the list of all user groups and the list of all Endpoint Group Creators and Endpoint Group Users in that tenant (i.e., user roles in that tenant that are equal or lower in the user role hierarchy; they cannot see Account Managers, Tenant Administrators, or Global Administrators).
- **Endpoint Group User:** This role is specific to a particular tenant, and can perform endpoint management only. Endpoint Group Users can be members of multiple user groups, but they cannot perform user management, and can only view their own user information.

See the *Intel® EMA Administration and Usage Guide* document for further information about the user roles.

Authorization enables you to make more granular choices when it comes to granting access to specific resources. There are authorization filters that are triggered before an action is requested to verify if the requesting user has the necessary privileges to perform the action. If the request is not authorized, the filter returns an error message and the action is not executed.

## 3 Intel® EMA API URI Details

The subsequent sections (3.1 – 3.23) include content that is automatically generated for the Intel® EMA API. It lists multiple API calls along with its descriptions, parameters, and other relevant details to help developers understand how to leverage them.

Online help for the API is implemented via the Swagger API framework. This section is an accurate depiction of the information contained in Swagger about the API.



## 3.1 802.1X Setups

### GET /api/v2/802\_1XSetups

The **GET /api/v2/802\_1XSetups** call gets a collection of 802.1X Setups (**802.1XSetups\_GetSetupsCollection**)

- Role required: Tenant Administrator, Endpoint Group Creator, or Endpoint Group User
- Return type: **\_802\_1XSetupSummary**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "SetupId": 0,
    "SetupName": "string",
    "TenantId": "string"
  }
]
```

- XML

```
<?xml version="1.0"?>
<Inline Model>
  <SetupId>1</SetupId>
  <SetupName>string</SetupName>
  <TenantId>string</TenantId>
</Inline Model>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request
  - 403 : Forbidden
  - 500 : Internal Server Error

### POST /api/v2/802\_1XSetups

The **POST /api/v2/802\_1XSetups** call creates an 802.1X Setup to be used by WifiSetup and Wired802.1X in Intel® AMT Profiles. (**802.1XSetups\_AddSetup**)

- Role required: Tenant Administrator
- Setup properties:
  - TenantId: This field is not required for this operation. If provided, it will be ignored.
  - PSK: This value, if provided, must be an OctetString (convertible to a byte array).
  - ProtectedAccessCredential: This value, if provided, must be an OctetString (convertible to a byte array).
  - AuthenticationProtocol: Currently, only the 0=EAP\_TLS option is fully supported.

- ClientAuthenticationSettings.DesignatedUsernameCN: The default value for this property is 4=UserPrincipalName. This is the user name required by the Network Policy Server (NPS), which is the RADIUS implementation by Microsoft.
- The request body is **\_802\_1XSetupDTO** (required). This is the 802.1X Setup object to create.

```
{
  "TenantId": "000000000-0000-0000-0000-000000000000",
  "SetupId": 0,
  "SetupName": "string",
  "Data": {
    "Enabled": "enabled",
    "PSK": "string",
    "ProtectedAccessCredential": "string",
    "Domain": "string",
    "Username": "string",
    "ServerCertificateNameComparisonOption": "other",
    "ServerCertificateName": "string",
    "UseRoamingIdentity": true,
    "RoamingIdentity": "string",
    "AuthenticationProtocol": "eAP_TLS",
    "PXETimeoutInSeconds": 0,
    "AvailableInS0": true,
    "ADIntegrationSettings": {
      "OrganizationUnit": "string",
      "SecurityGroups": [
        "string"
      ]
    },
    "ClientAuthenticationSettings": {
      "ClientCertificateSettings": {
        "Source": "none",
        "Thumbprint": "string",
        "CAName": "string",
        "Template": "string"
      },
      "CommonNamesOption": "default",
      "CommonNames": [
        "dNSFQDN"
      ],
      "DesignatedSubjectCN": "dNSFQDN",
      "DesignatedUsernameCN": "dNSFQDN"
    },
    "RootCertificateSettings": {
      "Source": "none",
      "Thumbprint": "string",
      "CAName": "string"
    }
  }
}
```

- Return type: **\_802\_1XSetupSummary**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "SetupId": 0,
  "SetupName": "string",
  "Data": {
    "Enabled": "enabled",
    "PSK": "string",
    "ProtectedAccessCredential": "string",
    "Domain": "string",
    "Username": "string",
    "ServerCertificateNameComparisonOption": "other",
    "ServerCertificateName": "string",
    "UseRoamingIdentity": true,
    "RoamingIdentity": "string",
    "AuthenticationProtocol": "eAP_TLS",
    "PXETimeoutInSeconds": 0,
    "AvailableInS0": true,
    "ADIntegrationSettings": {
      "OrganizationUnit": "string",
      "SecurityGroups": [
        "string"
      ]
    },
    "ClientAuthenticationSettings": {
      "ClientCertificateSettings": {
        "Source": "none",
        "Thumbprint": "string",
        "CAName": "string",
        "Template": "string"
      },
      "CommonNamesOption": "default",
      "CommonNames": [
        "dNSFQDN"
      ],
      "DesignatedSubjectCN": "dNSFQDN",
      "DesignatedUsernameCN": "dNSFQDN"
    },
    "RootCertificateSettings": {
      "Source": "none",
      "Thumbprint": "string",
      "CAName": "string"
    }
  }
}
```

- XML

```
<?xml version="1.0"?>
<_802_1XSetupDTO>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <SetupId>1</SetupId>
  <SetupName>string</SetupName>
  <Data>
    <Enabled>enabled</Enabled>
```

```

    <PSK>string</PSK>

    <ProtectedAccessCredential>string</ProtectedAccessCredential>
    <Domain>string</Domain>
    <Username>string</Username>

    <ServerCertificateNameComparisonOption>other</ServerCertificateNameComparisonOption>
    <ServerCertificateName>string</ServerCertificateName>
    <UseRoamingIdentity>true</UseRoamingIdentity>
    <RoamingIdentity>string</RoamingIdentity>
    <AuthenticationProtocol>eAP_TLS</AuthenticationProtocol>
    <PXETimeoutInSeconds>1</PXETimeoutInSeconds>
    <AvailableInS0>true</AvailableInS0>
    <ADIntegrationSettings>
    <OrganizationUnit>string</OrganizationUnit>
    <SecurityGroups>string</SecurityGroups>
    </ADIntegrationSettings>
    <ClientAuthenticationSettings>
    <ClientCertificateSettings>
    <Source>none</Source>
    <Thumbprint>string</Thumbprint>
    <CAName>string</CAName>
    <Template>string</Template>
    </ClientCertificateSettings>
    <CommonNamesOption>default</CommonNamesOption>
    <CommonNames>dNSFQDN</CommonNames>
    <DesignatedSubjectCN>dNSFQDN</DesignatedSubjectCN>
    <DesignatedUsernameCN>dNSFQDN</DesignatedUsernameCN>
    </ClientAuthenticationSettings>
    <RootCertificateSettings>
    <Source>none</Source>
    <Thumbprint>string</Thumbprint>
    <CAName>string</CAName>
    </RootCertificateSettings>
    </Data>
</_802_1XSetupDTO>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request **\_802\_1XSetupSummary**
  - 400 : Invalid ModelState
  - 403 : Forbidden
  - 409 : 802.1X Setup name already exists
  - 500 : Internal Server Error

## GET /api/v2/802\_1XSetups/{\_802\_1XSetupId}

The **GET /api/v2/802\_1XSetups/{\_802\_1XSetupId}** call gets an 802.1X Setup by ID.  
(802.1XSetups\_GetSetupById)

- Role required: Tenant Administrator, Endpoint Group Creator, or Endpoint Group User
- Return type: **\_802\_1XSetupDTO**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "SetupId": 0,
  "SetupName": "string",
  "Data": {
    "Enabled": "enabled",
    "PSK": "string",
    "ProtectedAccessCredential": "string",
    "Domain": "string",
    "Username": "string",
    "ServerCertificateNameComparisonOption": "other",
    "ServerCertificateName": "string",
    "UseRoamingIdentity": true,
    "RoamingIdentity": "string",
    "AuthenticationProtocol": "eAP_TLS",
    "PXETimeoutInSeconds": 0,
    "AvailableInS0": true,
    "ADIntegrationSettings": {
      "OrganizationUnit": "string",
      "SecurityGroups": [
        "string"
      ]
    },
    "ClientAuthenticationSettings": {
      "ClientCertificateSettings": {
        "Source": "none",
        "Thumbprint": "string",
        "CAName": "string",
        "Template": "string"
      },
      "CommonNamesOption": "default",
      "CommonNames": [
        "dNSFQDN"
      ],
      "DesignatedSubjectCN": "dNSFQDN",
      "DesignatedUsernameCN": "dNSFQDN"
    },
    "RootCertificateSettings": {
      "Source": "none",
      "Thumbprint": "string",
      "CAName": "string"
    }
  }
}
```

- XML

```
<?xml version="1.0"?>
<_802_1XSetupDTO>
```

```

<TenantId>00000000-0000-0000-0000-000000000000</TenantId>
<SetupId>1</SetupId>
<SetupName>string</SetupName>
<Data>
  <Enabled>enabled</Enabled>
  <PSK>string</PSK>

  <ProtectedAccessCredential>string</ProtectedAccessCredential>
    <Domain>string</Domain>
    <Username>string</Username>

  <ServerCertificateNameComparisonOption>other</ServerCertificate
NameComparisonOption>
    <ServerCertificateName>string</ServerCertificateName>
    <UseRoamingIdentity>true</UseRoamingIdentity>
    <RoamingIdentity>string</RoamingIdentity>
    <AuthenticationProtocol>eAP_TLS</AuthenticationProtocol>
    <PXETimeoutInSeconds>1</PXETimeoutInSeconds>
    <AvailableInS0>true</AvailableInS0>
    <ADIntegrationSettings>
      <OrganizationUnit>string</OrganizationUnit>
      <SecurityGroups>string</SecurityGroups>
    </ADIntegrationSettings>
    <ClientAuthenticationSettings>
      <ClientCertificateSettings>
        <Source>none</Source>
        <Thumbprint>string</Thumbprint>
        <CAName>string</CAName>
        <Template>string</Template>
      </ClientCertificateSettings>
      <CommonNamesOption>default</CommonNamesOption>
      <CommonNames>dNSFQDN</CommonNames>
      <DesignatedSubjectCN>dNSFQDN</DesignatedSubjectCN>
      <DesignatedUsernameCN>dNSFQDN</DesignatedUsernameCN>
    </ClientAuthenticationSettings>
    <RootCertificateSettings>
      <Source>none</Source>
      <Thumbprint>string</Thumbprint>
      <CAName>string</CAName>
    </RootCertificateSettings>
  </Data>
</_802_1XSetupDTO>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **\_802\_1XSetupDTO**
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal Server Error

## PUT /api/v2/802\_1XSetups/{\_802\_1XSetupId}

The **PUT /api/v2/802\_1XSetups/{\_802\_1XSetupId}** call updates an 802.1X Setup by ID.  
(802.1XSetups\_UpdateSetup)

- Role required: Tenant Administrator
- Setup properties:
  - TenantId: This field is not required for this operation. If provided, it will be ignored.
  - PSK: This value, if provided, must be an OctetString (convertible to a byte array).
  - ProtectedAccessCredential: This value, if provided, must be an OctetString (convertible to a byte array).
  - AuthenticationProtocol: Currently, only the 0=EAP\_TLS option is fully supported.
  - ClientAuthenticationSettings.DesignatedUsernameCN: The default value for this property is 4=UserPrincipalName. This is the user name required by the Network Policy Server (NPS), which is the RADIUS implementation by Microsoft.
- The path parameter is **\_802\_1XSetupId** (required) int32 format.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **\_802\_1XSetupDTO** (required). This is the 802.1X Setup object to update.

```
{
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "SetupId": 0,
  "SetupName": "string",
  "Data": {
    "Enabled": "enabled",
    "PSK": "string",
    "ProtectedAccessCredential": "string",
    "Domain": "string",
    "Username": "string",
    "ServerCertificateNameComparisonOption": "other",
    "ServerCertificateName": "string",
    "UseRoamingIdentity": true,
    "RoamingIdentity": "string",
    "AuthenticationProtocol": "eAP_TLS",
    "PXETimeoutInSeconds": 0,
    "AvailableInS0": true,
    "ADIntegrationSettings": {
      "OrganizationUnit": "string",
      "SecurityGroups": [
        "string"
      ]
    },
  },
  "ClientAuthenticationSettings": {
    "ClientCertificateSettings": {
      "Source": "none",
      "Thumbprint": "string",
      "CAName": "string",
      "Template": "string"
    }
  }
}
```

```

    },
    "CommonNamesOption": "default",
    "CommonNames": [
        "dNSFQDN"
    ],
    "DesignatedSubjectCN": "dNSFQDN",
    "DesignatedUsernameCN": "dNSFQDN"
},
"RootCertificateSettings": {
    "Source": "none",
    "Thumbprint": "string",
    "CAName": "string"
}
}
}

```

- Return type: **\_802\_1XSetupDTO**.
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "TenantId": "000000000-0000-0000-0000-000000000000",
  "SetupId": 0,
  "SetupName": "string",
  "Data": {
    "Enabled": "enabled",
    "PSK": "string",
    "ProtectedAccessCredential": "string",
    "Domain": "string",
    "Username": "string",
    "ServerCertificateNameComparisonOption": "other",
    "ServerCertificateName": "string",
    "UseRoamingIdentity": true,
    "RoamingIdentity": "string",
    "AuthenticationProtocol": "eAP_TLS",
    "PXETimeoutInSeconds": 0,
    "AvailableInS0": true,
    "ADIntegrationSettings": {
      "OrganizationUnit": "string",
      "SecurityGroups": [
        "string"
      ]
    },
  },
  "ClientAuthenticationSettings": {
    "ClientCertificateSettings": {
      "Source": "none",
      "Thumbprint": "string",
      "CAName": "string",
      "Template": "string"
    },
    "CommonNamesOption": "default",
    "CommonNames": [
      "dNSFQDN"
    ],
  },
}

```



```

        "DesignatedSubjectCN": "dNSFQDN",
        "DesignatedUsernameCN": "dNSFQDN"
    },
    "RootCertificateSettings": {
        "Source": "none",
        "Thumbprint": "string",
        "CAName": "string"
    }
}
}

```

o XML

```

<?xml version="1.0"?>
<_802_1XSetupDTO>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <SetupId>1</SetupId>
  <SetupName>string</SetupName>
  <Data>
    <Enabled>enabled</Enabled>
    <PSK>string</PSK>
    <ProtectedAccessCredential>string</ProtectedAccessCredential>
    <Domain>string</Domain>
    <Username>string</Username>

    <ServerCertificateNameComparisonOption>other</ServerCertificateName
ComparisonOption>
    <ServerCertificateName>string</ServerCertificateName>
    <UseRoamingIdentity>true</UseRoamingIdentity>
    <RoamingIdentity>string</RoamingIdentity>
    <AuthenticationProtocol>eAP_TLS</AuthenticationProtocol>
    <PXETimeoutInSeconds>1</PXETimeoutInSeconds>
    <AvailableInS0>true</AvailableInS0>
    <ADIntegrationSettings>
      <OrganizationUnit>string</OrganizationUnit>
      <SecurityGroups>string</SecurityGroups>
    </ADIntegrationSettings>
    <ClientAuthenticationSettings>
      <ClientCertificateSettings>
        <Source>none</Source>
        <Thumbprint>string</Thumbprint>
        <CAName>string</CAName>
        <Template>string</Template>
      </ClientCertificateSettings>
      <CommonNamesOption>default</CommonNamesOption>
      <CommonNames>dNSFQDN</CommonNames>
      <DesignatedSubjectCN>dNSFQDN</DesignatedSubjectCN>
      <DesignatedUsernameCN>dNSFQDN</DesignatedUsernameCN>
    </ClientAuthenticationSettings>
    <RootCertificateSettings>
      <Source>none</Source>
      <Thumbprint>string</Thumbprint>
      <CAName>string</CAName>
    </RootCertificateSettings>
  </Data>
</_802_1XSetupDTO>

```

```
</Data>
</_802_1XSetupDTO>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request
  - 400 : Invalid ModelState
  - 403 : Forbidden
  - 404 : Not found
  - 409 : 802.1X Setup name already exists or SetupId mismatch
  - 500 : Internal Server Error

## DELETE /api/v2/802\_1XSetups/{\_802\_1XSetupId}

The **DELETE /api/v2/802\_1XSetups/{\_802\_1XSetupId}** call deletes an 802.1X Setup by ID. (802.1XSetups\_DeleteSetup)

- Role required: Tenant Administrator
- The path parameter is **\_802\_1XSetupId** (required) int32 format.
- Response types:
  - 204 : Successful request, no content
  - 403 : Forbidden
  - 404 : Not found
  - 409 : Cannot delete 802.1X Setup if it is still associated with Intel® AMT Profile or WiFi Setup
  - 500 : Internal Server Error

## GET /api/v2/802\_1XSetups/CertificateAuthorities

The **GET /api/v2/802\_1XSetups/CertificateAuthorities** call gets a collection of Certificate Authorities found in the current Windows domain. (802.1XSetups\_GetCAInDomain)

- Role required: Tenant Administrator
- Return type: **CertificateAuthoritySummary**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "CAFullName": "string",
    "CertificateTemplates": [
      "string"
    ],
    "IsRoot": true
  }
]
```

- XML

```
<?xml version="1.0"?>
<Inline Model>
  <CAFullName>string</CAFullName>
  <CertificateTemplates>string</CertificateTemplates>
  <IsRoot>true</IsRoot>
</Inline Model>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request **CertificateAuthoritySummary**
  - 403 : Forbidden
  - 500 : Internal Server Error

## 3.2 Access Tokens

### GET /api/v2/accessTokens/getUsingExistingToken

The **GET /api/v2/accessTokens/getUsingExistingToken** call gets a new Access Token.  
(**AccessTokens\_GetUsingExistingToken**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User.
- This method provides Users the means to obtain a new Access Token before their current one expires.
- Return type: **AccessToken**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "access_token": "string",
  "token_type": "string",
  "expires_in": 0,
  "userName": "string",
  "issued": "string",
  "expires": "string"
}
```

- XML

```
<?xml version="1.0"?>
<AccessToken>
  <access_token>string</access_token>
  <token_type>string</token_type>
  <expires_in>1</expires_in>
  <userName>string</userName>
  <issued>string</issued>
  <expires>string</expires>
</AccessToken>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AccessToken**
  - 403 : Forbidden
  - 500 : Internal Server Error

## GET /api/v2/accessTokens/getUsingWindowsCredentials

The **GET /api/v2/accessTokens/getUsingWindowsCredentials** call gets a new Access Token using Windows Authentication credentials. (**AccessTokens\_getUsingWindowsCredentials**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User
- This method will only grant an Access Token under the following conditions:
  1. Intel® EMA must be installed with Windows Domain Authentication mode.
  2. The Intel® EMA server must be joined to an Active Directory domain.
  3. The caller must have an Intel® EMA account with a username that maps to a User Principal Name in Active Directory.
- Return type: **AccessToken**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "access_token": "string",
  "token_type": "string",
  "expires_in": 0,
  "userName": "string",
  "issued": "string",
  "expires": "string"
}
```

- XML

```
<?xml version="1.0"?>
<AccessToken>
  <access_token>string</access_token>
  <token_type>string</token_type>
  <expires_in>1</expires_in>
  <userName>string</userName>
  <issued>string</issued>
  <expires>string</expires>
</AccessToken>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml

- Response types:
  - 200 : Successful request, content returned **AccessToken**
  - 401 : Unauthorized
  - 405 : Method Not Allowed
  - 500 : Internal Server Error

## 3.3 Agents

### GET /api/v2/agents/getWin32Console

The **GET /api/v2/agents/getWin32Console** call gets the latest agent executable for Win32Console (**Agents\_GetWin32Console**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User
- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, attachment returned
  - 500 : Internal Server Error

### GET /api/v2/agents/getWin32Service

The **GET /api/v2/agents/getWin32Service** call gets the latest agent installer for Win32Service (**Agents\_GetWin32Service**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User
- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, attachment returned
  - 500 : Internal Server Error

### GET /api/v2/agents/getWin64Console

The **GET /api/v2/agents/getWin64Console** call gets the latest agent executable for Win64Console (**Agents\_GetWin64Console**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User
- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, attachment returned

- 500 : Internal Server Error

## GET /api/v2/agents/getWin64Service

The **GET /api/v2/agents/getWin64Service** call gets the latest agent installer for Win64Service (**Agents\_GetWin64Service**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User
- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, attachment returned
  - 500 : Internal Server Error

## 3.4 Ajax Cookies

### GET /api/v2/ajaxCookies

The **GET /api/v2/ajaxCookies** call Gets an Ajax cookie. (**AjaxCookies\_GetAjaxCookie**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User
- Return type: **AjaxCookie**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "cookie": "string",
  "expireTime": "2018-10-23T17:09:08.756Z"
}
```

- XML

```
xml version="1.0"?>
<AjaxCookie>
  <cookie>string</cookie>
  <expireTime>1970-01-01T00:00:00.001Z</expireTime>
</AjaxCookie>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AjaxCookie**
  - 401 : Unauthorized
  - 500 : Internal Server Error

## 3.5 Intel® AMT Credentials

### GET /api/v2/amtCredentials/{endpointId}

The **GET /api/v2/amtCredentials/{endpointId}** call returns the Intel AMT admin credentials or Intel MEBx password, based on credential type defined in the query string, for a provisioned endpoint if its associated AmtSetup record is found in the database (**AmtCredentials\_GetById**)

- Roles required: Tenant Administrator, Endpoint Group Creator or Endpoint Group User with execute right to the endpoint  
When returning the Intel® MEBx password, the username field in Intel® AMT credentials is always null
- Return type: **AmtCredentialsDTO**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "userName": "string",
  "password": "string"
}
```

- XML

```
<?xml version="1.0"?>
<AmtCredentialsDTO>
  <userName>string</userName>
  <password>string</password>
</AmtCredentialsDTO>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtCredentialsDTO**
  - 400 : Bad request
  - 403 : Forbidden
  - 404 : Not found
  - 409 : Resource conflict due to not able to get the credential or password
  - 500 : Internal server error

## 3.6 Intel® AMT Discoveries

### POST /api/v2/amtDiscoveries/discoverByIP

The **POST /api/v2/amtDiscoveries/discoverByIP** call submits a request to perform remote Intel® AMT Discovery by IP address. (**AmtDiscoveries\_DiscoveryByIp**)

- Role required: Tenant Administrator
- IP corresponds to the IP address of the Intel® AMT computer to discover.  
MAC corresponds to the MAC address of the server's network interface to be used for the discovery.  
The format is 6 octets separated by colons (:). For example:

```
{
  ...
  "MAC": "00:1F:C6:9B:D6:67"
}
```

- The request body is (required). This is the Intel® AMT Discoveries object to create
- Return type: **AmtDiscoveryProcessId**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "amtDiscoveryId" : "amtDiscoveryId"
}
```

- XML

```
<?xml version="1.0"?>
<AmtDiscoveryProcessId>
  <amtDiscoveryId>string</amtDiscoveryId>
</AmtDiscoveryProcessId>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtDiscoveryProcessId**
  - 400 : Bad request
  - 500 : Internal server error

## POST /api/v2/amtDiscoveries/discoverByIPAutomatically

The **POST /api/v2/amtDiscoveries/discoverByIPAutomatically** call submits a request to perform remote Intel® AMT Discovery automatically by IP address. (**AmtDiscoveries\_DiscoveryByIpAutomatically**)

- Role required: Tenant Administrator
  - Interval refers to the minutes between each automatic discovery execution. Interval 0 turns off automatic discoveries.
  - RestartManageabilityServer indicates whether to restart the Manageability Server or not, as changes to its configuration are applied only when the server is restarted.
  - IP corresponds to the IP address of the Intel® AMT computer to discover.
  - MAC corresponds to the MAC address of the server's network interface to be used for the discovery. The format is 6 octets separated by colons (:).
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **SingleIPAddressAutomatic** (required). This is the IP address object.



```
{
  "Interval": 0,
  "RestartManageabilityServer": true,
  "IP": "string",
  "MAC": "string"
}
```

- Response types:
  - 204 : Successful request, no content
  - 400 : Bad request
  - 500 : Internal server error

## GET /api/v2/amtDiscoveries/{amtDiscoveryId}

The **GET /api/v2/amtDiscoveries/{amtDiscoveryId}** call gets the result of an Intel® AMT Discovery process by ID. (**AmtDiscoveries\_GetAmtDiscoveryResult**)

- Role required: Tenant Administrator
- Return type: **AmtDiscoveryResult**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "AmtDiscoveryId": "00000000-0000-0000-0000-000000000000",
  "Title": "string",
  "Request": {
    "TenantId": "00000000-0000-0000-0000-000000000000",
    "NetInterfaceMacAddress": "string",
    "DiscoveryType": "singleIP",
    "StartIPAddress": "string",
    "EndIPRange": "string",
    "SubnetMask": "string"
  },
  "StartDateStr": "string",
  "EndDateStr": "string",
  "LastUpdatedStr": "string",
  "ExecutionType": "manual",
  "ExecutionTypeStr": "string",
  "DiscoveryTypeStr": "string",
  "Status": "requested",
  "StatusStr": "string",
  "AddressesQuantity": 0,
  "Found": 0,
  "Results": [
    {
      "IPAddress": "string",
      "AmtVersion": "string",
      "PortStatus": "waitingForSetup",
      "PortStatusDescription": "string",
      "ProvisioningState": "string",
      "DetailId": 0,
      "ErrorMessage": "string",
      "AddressesQuantity": 0,
      "Hostname": "string"
    }
  ]
}
```

```
]
}
```

o XML

```
<?xml version="1.0"?>
<AmtDiscoveryResult>
  <AmtDiscoveryId>00000000-0000-0000-0000-000000000000</AmtDiscoveryId>
  <Title>string</Title>
  <Request>
    <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
    <NetInterfaceMacAddress>string</NetInterfaceMacAddress>
    <DiscoveryType>singleIP</DiscoveryType>
    <StartIPAddress>string</StartIPAddress>
    <EndIPRange>string</EndIPRange>
    <SubnetMask>string</SubnetMask>
  </Request>
  <StartDateStr>string</StartDateStr>
  <EndDateStr>string</EndDateStr>
  <LastUpdatedStr>string</LastUpdatedStr>
  <ExecutionType>manual</ExecutionType>
  <ExecutionTypeStr>string</ExecutionTypeStr>
  <DiscoveryTypeStr>string</DiscoveryTypeStr>
  <Status>requested</Status>
  <StatusStr>string</StatusStr>
  <AddressesQuantity>1</AddressesQuantity>
  <Found>1</Found>
  <Results>
    <IPAddress>string</IPAddress>
    <AmtVersion>string</AmtVersion>
    <PortStatus>waitingForSetup</PortStatus>
    <PortStatusDescription>string</PortStatusDescription>
    <ProvisioningState>string</ProvisioningState>
    <DetailId>1</DetailId>
    <ErrorMessage>string</ErrorMessage>
    <AddressesQuantity>1</AddressesQuantity>
    <Hostname>string</Hostname>
  </Results>
</AmtDiscoveryResult>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **AmtDiscoveryResult**
  - o 404 : Not found
  - o 500 : Internal server error

## GET /api/v2/amtDiscoveries

The **GET /api/v2/amtDiscoveries** call Gets a collection of Intel® AMT Discovery summaries.  
(**AmtDiscoveries\_GetAmtDiscoveryStatus**)

- Role required: Tenant Administrator
- The result of an Intel® AMT Discovery is referenced by its process identifier.
- To get the full list of Intel® AMT Discovery processes use `discoveryStatus = null`.
- Return type: **AmtDiscoveryResultSummary**.
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "AmtDiscoveryId": "00000000-0000-0000-0000-000000000000",
    "Title": "string",
    "Request": {
      "TenantId": "00000000-0000-0000-0000-000000000000",
      "NetInterfaceMacAddress": "string",
      "DiscoveryType": "singleIP",
      "StartIPAddress": "string",
      "EndIPRange": "string",
      "SubnetMask": "string"
    },
    "StartDateStr": "string",
    "EndDateStr": "string",
    "LastUpdatedStr": "string",
    "ExecutionType": "manual",
    "ExecutionTypeStr": "string",
    "DiscoveryTypeStr": "string",
    "Status": "requested",
    "StatusStr": "string",
    "AddressesQuantity": 0,
    "Found": 0
  }
]
```

- XML

```
<?xml version="1.0"?>
<Inline Model>
  <AmtDiscoveryId>00000000-0000-0000-0000-000000000000</AmtDiscoveryId>
  <Title>string</Title>
  <Request>
    <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
    <NetInterfaceMacAddress>string</NetInterfaceMacAddress>
    <DiscoveryType>singleIP</DiscoveryType>
    <StartIPAddress>string</StartIPAddress>
    <EndIPRange>string</EndIPRange>
    <SubnetMask>string</SubnetMask>
  </Request>
  <StartDateStr>string</StartDateStr>
  <EndDateStr>string</EndDateStr>
  <LastUpdatedStr>string</LastUpdatedStr>
```

```

    <ExecutionType>manual</ExecutionType>
    <ExecutionTypeStr>string</ExecutionTypeStr>
    <DiscoveryTypeStr>string</DiscoveryTypeStr>
    <Status>requested</Status>
    <StatusStr>string</StatusStr>
    <AddressesQuantity>1</AddressesQuantity>
    <Found>1</Found>
  </Inline Model>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtDiscoveryResultSummary**
  - 500 : Internal server error

## POST /api/v2/amtDiscoveries/{amtDiscoveryId}/cancel

The **POST /api/v2/amtDiscoveries/{amtDiscoveryId}/cancel** call Submits a request to cancel a running remote Intel® AMT Discovery process by ID. (**AmtDiscoveries\_CancelAmtDiscoveryProcess**)

- Role required: Tenant Administrator
- The path parameter is **amtDiscoveryId** (required). Intel® AMT Discovery process ID. Canonical GUID form: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx (string)
- Response types:
  - 204 : Successful request, no content
  - 404 : Not found
  - 500 : Internal server error

## POST /api/v2/amtDiscoveries/discoverBySubnet

The **POST /api/v2/amtDiscoveries/discoverBySubnet** call Submits a request to perform remote Intel® AMT Discovery in a subnet. It can take from minutes to hours, depending on the number of usable hosts in the given subnet. (**AmtDiscoveries\_DiscoverBySubnet**)

- Role required: Tenant Administrator
- StartIPAddress corresponds to any IP address of an Intel® AMT computer in a subnet, where the discovery is going to be executed.
- SubnetMask can be given in dot-decimal notation or CIDR suffix.
- MAC corresponds to the MAC address of the server's network interface to be used for the discovery. The format is 6 octets separated by colons (:). For example:

```

{
  ...
  "SubnetMask": "255.255.255.0", or "SubnetMask": "24",
  "MAC": "00:1F:C6:9B:D6:67"
}

```

**Note:** SubnetMask "255.255.255.254" is considered invalid, since its Host Address Range is 0.

- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json

- application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **Subnet**. This is the Subnet object.
- Return type: **AmtDiscoveryProcessId**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.
  - XML
- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtDiscoveryProcessId**
  - 400 : Bad request
  - 500 : Internal server error

## POST /api/v2/amtDiscoveries/discoverBySubnetAutomatically

The **POST /api/v2/amtDiscoveries/discoverBySubnetAutomatically** call submits a request to perform remote Intel® AMT Discovery automatically in a subnet. It can take from minutes to hours, depending on the number of usable hosts in the given subnet. (**AmtDiscoveries\_DiscoverBySubnetAutomatically**)

- Role required: Tenant Administrator
- Interval refers to the minutes between each automatic discovery execution. Interval 0 turns off automatic discoveries.
- RestartManageabilityServer indicates whether to restart the Manageability Server or not, as changes to its configuration are applied only when the server is restarted.
- StartIPAddress corresponds to any IP address of an Intel® AMT computer in a subnet, where the discovery is going to be executed.
- SubnetMask can be given in dot-decimal notation or CIDR suffix.
- Note: SubnetMask "255.255.255.254" is considered invalid, since its Host Address Range is 0.
- MAC corresponds to the MAC address of the server's network interface to be used for the discovery. The format is 6 octets separated by colons (:).
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **SubnetAutomatic** (required). This is the Subnet object.

```
{
  "Interval": 0,
  "RestartManageabilityServer": true,
  "StartIPAddress": "string",
  "SubnetMask": "string",
  "MAC": "string"
}
```

- Response types:
  - 204 : Successful request, no content
  - 400 : Bad request

- 500 : Internal server error

## POST /api/v2/amtDiscoveries/discoverByIPRange

The **POST /api/v2/amtDiscoveries/discoverByIPRange** call Submits a request to perform remote Intel® AMT Discovery within an IP address range. (**AmtDiscoveries\_DiscoverByIPRange**)

- Role required: Tenant Administrator
- StartIPAddress corresponds to any IP address of an Intel® AMT computer in a subnet, where the discovery is going to be executed.
- EndIPRange cannot be lower than the StartIPAddress.
- MAC corresponds to the MAC address of the server's network interface to be used for the discovery. The format is 6 octets separated by colons (:). For example:

```
{
  ...
  "MAC": "00:1F:C6:9B:D6:67" \
}
```

- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is (required). This is the Intel® AMT Discoveries object to create
- Return type: **AmtDiscoveryProcessId**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "amtDiscoveryId": "string"
}
```

- XML

```
<?xml version="1.0"?>
<AmtDiscoveryProcessId>
  <amtDiscoveryId>string</amtDiscoveryId>
</AmtDiscoveryProcessId>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtDiscoveryProcessId**
  - 400 : Bad request
  - 500 : Internal server error

## POST /api/v2/amtDiscoveries/discoverByIPRangeAutomatically

The **POST /api/v2/amtDiscoveries/discoverByIPRangeAutomatically** call Submits a request to perform remote Intel® AMT Discovery automatically within an IP address range.

**(AmtDiscoveries\_DiscoverByIPRangeAutomatically)**

- Role required: Tenant Administrator
- Interval refers to the minutes between each automatic discovery execution. Interval 0 turns off automatic discoveries.
- RestartManageabilityServer indicates whether to restart the Manageability Server or not, as changes to its configuration are applied only when the server is restarted.
- StartIPAddress corresponds to any IP address of an Intel® AMT computer in a subnet, where the discovery is going to be executed.
- EndIPRange cannot be lower than the StartIPAddress.
- MAC corresponds to the MAC address of the server's network interface to be used for the discovery. The format is 6 octets separated by colons (:).
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **IPRangeDiscoveryAutomatic** (required). This is the IP address range object

```
{
  "Interval": 0,
  "RestartManageabilityServer": true,
  "StartIPAddress": "string",
  "EndIPRange": "string",
  "MAC": "string"
}
```

- Response types:
  - 204 : Successful request, no content
  - 400 : Bad request
  - 500 : Internal server error

## POST /api/v2/amtDiscoveries/discoverByLocalBroadcast

The **POST /api/v2/amtDiscoveries/discoverByLocalBroadcast** call submits a request to perform remote Intel® AMT Discovery by local broadcast. **(AmtDiscoveries\_DiscoveryByLocalBroadcast)**

- Role required: Tenant Administrator
- MAC corresponds to the MAC address of the server's network interface to be used for the discovery. The format is 6 octets separated by colons (:). For example:

```
{
  ...
  "MAC": "00:1F:C6:9B:D6:67"
}
```

- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is (required). This is the Intel® AMT Discoveries object to create

- Return type: **AmtDiscoveryProcessId**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "amtDiscoveryId": "string"
}
```

- XML

```
<?xml version="1.0"?>
<AmtDiscoveryProcessId>
  <amtDiscoveryId>string</amtDiscoveryId>
</AmtDiscoveryProcessId>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtDiscoveryProcessId**
  - 400 : Bad request
  - 500 : Internal server error

## POST /api/v2/amtDiscoveries/discoverByLocalBroadcastAutomatically

The **POST /api/v2/amtDiscoveries/discoverByLocalBroadcastAutomatically** call Submits a request to perform remote Intel® AMT Discovery automatically by local broadcast.

**(AmtDiscoveries\_DiscoveryByLocalBroadcastAutomatically)**

- Role required: Tenant Administrator
- Interval refers to the minutes between each automatic discovery execution. Interval 0 turns off automatic discoveries.
- RestartManageabilityServer indicates whether to restart the Manageability Server or not, as changes to its configuration are applied only when the server is restarted.
- MAC corresponds to the MAC address of the server's network interface to be used for the discovery. The format is 6 octets separated by colons (:).
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **LocalBroadcastAutomatic**. This is the Local Broadcast Automatic object.

```
{
  "Interval": 0,
  "RestartManageabilityServer": true,
  "MAC": "string"
}
```

- Response types:
  - 204 : Successful request, no content
  - 400 : Bad request



- 500 : Internal server error

## 3.7 Intel® AMT Profiles

### GET /api/v2/amtProfiles

The **GET /api/v2/amtProfiles** call gets a list of Intel® AMT Profile summaries. (**AmtProfiles\_GetCollection**)

- Role required: Tenant Administrator, Endpoint Group Creator, or Endpoint Group User
- Return type: **AmtProfileSummary**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "AmtProfileId": 0,
    "Name": "string",
    "Description": "string",
    "TenantId": "string"
  }
]
```

- XML

```
<?xml version="1.0"?>
<Inline Model>
  <AmtProfileId>1</AmtProfileId>
  <Name>string</Name>
  <Description>string</Description>
  <TenantId>string</TenantId>
</Inline Model>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtProfileSummary**
  - 500 : Internal Server Error

### POST /api/v2/amtProfiles

The **POST /api/v2/amtProfiles** call creates an Intel® AMT Profile. (**AmtProfiles\_Post**)

- Role required: Tenant Administrator or Endpoint Group Creator
- The following defines the profile object's optional parameters and their default values. If no argument is provided for an optional parameter, then its default value is used.
  - TlsAuthType: 1 = TlsNoAuth. If this is set to NoTLS, you need to specify the CIRASettings.
  - PowerPackageSettings.PowerActiveOn: 1 = AlwaysOnInAc.
  - PowerPackageSettings.PowerStatIdleTimeoutInMinutes: 1.
  - ManagementInterfacesSettings.WebUIServiceEnabledState: 2 = Enabled.
  - ManagementInterfacesSettings.KVMInterfaceState: 2 = Enabled.
  - ManagementInterfacesSettings.UserConsentRequired: 4294967295 = All.

- ManagementInterfacesSettings.SOLEnabled: true.
- ManagementInterfacesSettings.IDEREnabled: true.
- ManagementInterfacesSettings.RedirectionServiceState: SOLEnabled=true and [IDEREnabled=true: 32771(IdleOnSolOn); IDEREnabled=false: 32770(IdleOffSolOn)]; SOLEnabled=false and [IDEREnabled=true: 32769(IdleOnSolOff); IDEREnabled=false: 32768(Disabled)].
- FqdnSettings.FqdnSource: 0 = SharedWithHostOS.
- IpSettings.DHCPEnabled: true.
- IpSettings.SharedStaticIp: false.
- IpSettings.Source: 0 = DHCP.
- WiFiConnectionSettings.WiFiSetups: Empty collection (it is optional only if WiFiConnectionSettings.WiFiConnectionEnabledConfiguration=0)
- Wired802\_1XSettings: An object containing \_802\_1Setup\_DBLookupKey=null.
- CIRASettings (the whole object): null.
- CIRASettings.CIRATunnel: false.
- CIRASettings.CIRAProxies (the whole object): Empty collection.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AmtProfileNew**. This is the Intel AMT Profile New object.

```
{
  "Name": "string",
  "Description": "string",
  "TlsAuthType": "noTLS",
  "PowerPackageSettings": {
    "PowerActiveOn": "onInS0",
    "PowerStateIdleTimeoutInMinutes": 0
  },
  "ManagementInterfacesSettings": {
    "WebUIServiceEnabledState": "enabled",
    "KVMInterfaceState": "enabled",
    "UserConsentRequired": "none",
    "UserConsentDisplayTimeout": 0,
    "SOLEnabled": true,
    "IDEREnabled": true,
    "RedirectionServiceState": "disabled"
  },
  "FqdnSettings": {
    "FqdnSource": "sharedWithHostOS"
  },
  "IpSettings": {
    "DHCPEnabled": true,
    "SharedStaticIp": true,
    "Source": "dHCP",
    "IP": "string",
    "SubnetMask": "string",
    "DefaultGateway": "string",
    "PrimaryDNS": "string",
    "SecondaryDNS": "string"
  },
}
```

```

"WiFiConnectionSettings": {
  "AMTHostWiFiSyncEnabled": "disabled",
  "WiFiConnectionEnabledConfiguration": "noSetup",
  "WiFiEnabledInPowerState": "s0Only",
  "WiFiSetups": [
    0
  ]
},
"Wired802_1XSettings": {
  "_802_1Setup_DBLookupKey": 0
},
"CIRASettings": {
  "CIRATunnel": true,

  "CIRAProxies": [
    {
      "AccessInfo": "string",
      "InfoFormat": "iPv4",
      "Port": 0,
      "DNSSuffix": "string"
    }
  ]
}
}

```

- Return type: **AmtProfile**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "AmtProfileId": 0,
  "Name": "string",
  "Description": "string",
  "TlsAuthType": "noTLS",
  "PowerPackageSettings": {
    "PowerActiveOn": "onInS0",
    "PowerStateIdleTimeoutInMinutes": 0
  },
  "ManagementInterfacesSettings": {
    "WebUIServiceEnabledState": "enabled",
    "KVMInterfaceState": "enabled",
    "UserConsentRequired": "none",
    "UserConsentDisplayTimeout": 0,
    "SOLEnabled": true,
    "IDEREnabled": true,
    "RedirectionServiceState": "disabled"
  },
  "FqdnSettings": {
    "FqdnSource": "sharedWithHostOS"
  },
  "IpSettings": {
    "DHCPEnabled": true,
    "SharedStaticIp": true,
    "Source": "dHCP",

```

```

        "IP": "string",
        "SubnetMask": "string",
        "DefaultGateway": "string",
        "PrimaryDNS": "string",
        "SecondaryDNS": "string"
    },
    "WiFiConnectionSettings": {
        "AMTHostWiFiSyncEnabled": "disabled",
        "WiFiConnectionEnabledConfiguration": "noSetup",
        "WiFiEnabledInPowerState": "s0Only",
        "WiFiSetups": [
            0
        ]
    },
    "Wired802_1XSettings": {
        "_802_1Setup_DBLookupKey": 0
    },
    "CIRASettings": {
        "CIRATunnel": true,

        "CIRAProxies": [
            {
                "AccessInfo": "string",
                "InfoFormat": "iPv4",
                "Port": 0,
                "DNSSuffix": "string"
            }
        ]
    },
    "TenantId": "string"
}

```

o XML

```

<?xml version="1.0"?>
<AmtProfile>
  <AmtProfileId>1</AmtProfileId>
  <Name>string</Name>
  <Description>string</Description>
  <TlsAuthType>noTLS</TlsAuthType>
  <PowerPackageSettings>
    <PowerActiveOn>onInS0</PowerActiveOn>

    <PowerStateIdleTimeoutInMinutes>1</PowerStateIdleTimeoutInMinutes>
  </PowerPackageSettings>
  <ManagementInterfacesSettings>

  <WebUIServiceEnabledState>enabled</WebUIServiceEnabledState>
    <KVMInterfaceState>enabled</KVMInterfaceState>
    <UserConsentRequired>none</UserConsentRequired>
    <UserConsentDisplayTimeout>1</UserConsentDisplayTimeout>
    <SOLEnabled>true</SOLEnabled>
    <IDEREEnabled>true</IDEREEnabled>

```

```

    <RedirectionServiceState>disabled</RedirectionServiceState>
  </ManagementInterfacesSettings>
  <FqdnSettings>
    <FqdnSource>sharedWithHostOS</FqdnSource>
  </FqdnSettings>
  <IpSettings>
    <DHCPEnabled>true</DHCPEnabled>
    <SharedStaticIp>true</SharedStaticIp>
    <Source>dHCP</Source>
    <IP>string</IP>
    <SubnetMask>string</SubnetMask>
    <DefaultGateway>string</DefaultGateway>
    <PrimaryDNS>string</PrimaryDNS>
    <SecondaryDNS>string</SecondaryDNS>
  </IpSettings>
  <WiFiConnectionSettings>
    <AMTHostWiFiSyncEnabled>disabled</AMTHostWiFiSyncEnabled>

  <WiFiConnectionEnabledConfiguration>noSetup</WiFiConnectionEnabledConfiguration>
    <WiFiEnabledInPowerState>s0Only</WiFiEnabledInPowerState>
    <WiFiSetups>1</WiFiSetups>
  </WiFiConnectionSettings>
  <Wired802_1XSettings>
    <_802_1Setup_DBLookupKey>1</_802_1Setup_DBLookupKey>
  </Wired802_1XSettings>
  <CIRASettings>
    <CIRATunnel>true</CIRATunnel>

    <CIRAProxies>
      <AccessInfo>string</AccessInfo>
      <InfoFormat>iPv4</InfoFormat>
      <Port>1</Port>
      <DNSSuffix>string</DNSSuffix>
    </CIRAProxies>
  </CIRASettings>
  <TenantId>string</TenantId>
</AmtProfile>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtProfile**
  - 400 : Bad Request
  - 409 : AMT Profile name already exists
  - 500 : Internal Server Error

## GET /api/v2/amtProfiles/{amtProfileId}

The **GET /api/v2/amtProfiles/{amtProfileId}** call gets an Intel® AMT Profile by ID. (**AmtProfiles\_GetAmtProfile**)

- Role required: Tenant Administrator, Endpoint Group Creator, or Endpoint Group User
- Return type: **AmtProfile**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "AmtProfileId": 0,
  "Name": "string",
  "Description": "string",
  "TlsAuthType": "noTLS",
  "PowerPackageSettings": {
    "PowerActiveOn": "onInS0",
    "PowerStateIdleTimeoutInMinutes": 0
  },
  "ManagementInterfacesSettings": {
    "WebUIServiceEnabledState": "enabled",
    "KVMInterfaceState": "enabled",
    "UserConsentRequired": "none",
    "UserConsentDisplayTimeout": 0,
    "SOLEnabled": true,
    "IDEREnabled": true,
    "RedirectionServiceState": "disabled"
  },
  "FqdnSettings": {
    "FqdnSource": "sharedWithHostOS"
  },
  "IpSettings": {
    "DHCPEnabled": true,
    "SharedStaticIp": true,
    "Source": "dHCP",
    "IP": "string",
    "SubnetMask": "string",
    "DefaultGateway": "string",
    "PrimaryDNS": "string",
    "SecondaryDNS": "string"
  },
  "WiFiConnectionSettings": {
    "AMTHostWiFiSyncEnabled": "disabled",
    "WiFiConnectionEnabledConfiguration": "noSetup",
    "WiFiEnabledInPowerState": "s0Only",
    "WiFiSetups": [
      0
    ]
  },
  "Wired802_1XSettings": {
    "_802_1Setup_DBLookupKey": 0
  },
  "CIRASettings": {
    "CIRATunnel": true,
    "CIRAProxies": [
      {
        "AccessInfo": "string",
        "InfoFormat": "IPv4",
        "Port": 0,

```

```

        "DNSSuffix": "string"
    }
]
},
"TenantId": "string"
}

```

- XML

```

<?xml version="1.0"?>
<AmtProfile>
  <AmtProfileId>1</AmtProfileId>
  <Name>string</Name>
  <Description>string</Description>
  <TlsAuthType>noTLS</TlsAuthType>
  <PowerPackageSettings>
    <PowerActiveOn>onInS0</PowerActiveOn>

  <PowerStateIdleTimeoutInMinutes>1</PowerStateIdleTimeoutInMinutes>
  </PowerPackageSettings>
  <ManagementInterfacesSettings>

  <WebUIServiceEnabledState>enabled</WebUIServiceEnabledState>
    <KVMInterfaceState>enabled</KVMInterfaceState>
    <UserConsentRequired>none</UserConsentRequired>
    <UserConsentDisplayTimeout>1</UserConsentDisplayTimeout>
    <SOLEnabled>true</SOLEnabled>
    <IDEREnabled>true</IDEREnabled>
    <RedirectionServiceState>disabled</RedirectionServiceState>
  </ManagementInterfacesSettings>
  <FqdnSettings>
    <FqdnSource>sharedWithHostOS</FqdnSource>
  </FqdnSettings>
  <IpSettings>
    <DHCPEnabled>true</DHCPEnabled>
    <SharedStaticIp>true</SharedStaticIp>
    <Source>dHCP</Source>
    <IP>string</IP>
    <SubnetMask>string</SubnetMask>
    <DefaultGateway>string</DefaultGateway>
    <PrimaryDNS>string</PrimaryDNS>
    <SecondaryDNS>string</SecondaryDNS>
  </IpSettings>
  <WiFiConnectionSettings>
    <AMTHostWiFiSyncEnabled>disabled</AMTHostWiFiSyncEnabled>

  <WiFiConnectionEnabledConfiguration>noSetup</WiFiConnectionEnabledConfiguration>
    <WiFiEnabledInPowerState>s0Only</WiFiEnabledInPowerState>
    <WiFiSetups>1</WiFiSetups>
  </WiFiConnectionSettings>
  <Wired802_1XSettings>
    <_802_1Setup_DBLookupKey>1</_802_1Setup_DBLookupKey>

```

```

</Wired802_1XSettings>
<CIRASettings>
  <CIRATunnel>true</CIRATunnel>

  <CIRAProxies>
    <AccessInfo>string</AccessInfo>
    <InfoFormat>iPv4</InfoFormat>
    <Port>1</Port>
    <DNSSuffix>string</DNSSuffix>
  </CIRAProxies>
</CIRASettings>
<TenantId>string</TenantId>
</AmtProfile>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtProfile**
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal Server Error

## PUT /api/v2/amtProfiles/{amtProfileId}

The **PUT /api/v2/amtProfiles/{amtProfileId}** call updates an Intel® AMT Profile by ID. (**AmtProfiles\_Put**)

- Role required: Tenant Administrator or Endpoint Group Creator
- The properties AmtProfileId and TenantId are readonly and cannot be updated.
- The following list defines the default values that would be used for any optional parameters if they are omitted from the profile object:
  - TlsAuthType: 1 = TlsNoAuth. If this is set to NoTLS, you need to specify the CIRASettings.
  - PowerPackageSettings.PowerActiveOn: 1 = AlwaysOnInAc.
  - PowerPackageSettings.PowerStateIdleTimeoutInMinutes: 1.
  - ManagementInterfacesSettings.WebUIServiceEnabledState: 2 = Enabled.
  - ManagementInterfacesSettings.KVMInterfaceState: 2 = Enabled.
  - ManagementInterfacesSettings.UserConsentRequired: 4294967295 = All.
  - ManagementInterfacesSettings.SOLEnabled: true.
  - ManagementInterfacesSettings.IDEREnabled: true.
  - ManagementInterfacesSettings.RedirectionServiceState: SOLEnabled=true and [IDEREnabled=true: 32771(IdentifierOnSolOn); IDEREnabled=false: 32770(IdentifierOffSolOn)]; SOLEnabled=false and [IDEREnabled=true: 32769(IdentifierOnSolOff); IDEREnabled=false: 32768(Disabled)].
  - FqdnSettings.FqdnSource: 0 = SharedWithHostOS.
  - IpSettings.DHCPEnabled: true.
  - IpSettings.SharedStaticIp: false.
  - IpSettings.Source: 0 = DHCP.
  - WiFiConnectionSettings.WiFiSetups: Empty collection (it is optional only if WiFiConnectionSettings.WiFiConnectionEnabledConfiguration=0)
  - Wired802\_1XSettings: An object containing \_802\_1Setup\_DBLookupKey=null.
  - CIRASettings (the whole object): null.



- CIRASettings.CIRATunnel: false.
  - CIRASettings.CIRAProxies (the whole object): Empty collection.
- The path parameter is amtProfileId. The Intel® AMT Profiles.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **amtProfileId**. This is the Intel AMT Profile Id object.
- Return type: **AmtProfile**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.
  - XML
- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned AmtProfile
  - 400 : Bad request
  - 404 : Not found
  - 409 : AMT profile name already exists
  - 500 : Internal Server Error

## DELETE /api/v2/amtProfiles/{amtProfileId}

The **DELETE /api/v2/amtProfiles/{amtProfileId}** call deletes an Intel® AMT Profile by ID. (AmtProfiles\_Delete)

- Role required: Tenant Administrator or Endpoint Group Creator
- The path parameter is amtProfileId. The Intel® AMT Profiles.
- Response types:
  - 204 : Successful request, no content
  - 404 : Not found
  - 409 : Cannot be deleted since the AMT Profile is still linked to at least one Endpoint Group
  - 500 : Internal Server Error

## 3.8 Intel® AMT Profile To Endpoint Groups Assignments

### GET /api/v2/amtProfileToEndpointGroupsAssignments/{amtProfileId}

The **GET /api/v2/amtProfileToEndpointGroupsAssignments/{amtProfileId}** call gets a list of Endpoint Groups associated with a given AmtProfileId. (**AmtProfileToEndpointGroupsAssignments\_GetEndpointGroups**)

- Role required: Tenant Administrator, Endpoint Group Creator, or Endpoint Group User
- 404 will be returned if AmtProfileId is not found.
- Return type: **AmtProfileToEndpointGroupsAssignments**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "EndpointGroupCount": 0,
  "AccessibleEndpointGroups": [
    {
```

```

        "EndpointGroupId": "string",
        "Name": "string"
    }
]
}

```

- o XML

```

<?xml version="1.0"?>
<AmtProfileToEndpointGroupsAssignments>
  <EndpointGroupCount>1</EndpointGroupCount>
  <AccessibleEndpointGroups>
    <EndpointGroupId>string</EndpointGroupId>
    <Name>string</Name>
  </AccessibleEndpointGroups>
</AmtProfileToEndpointGroupsAssignments>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **AmtProfileToEndpointGroupsAssignments**
  - o 403 : Forbidden
  - o 404 : Not found
  - o 500 : Internal Server Error

## 3.9 Intel® AMT Provisioning Certificates

### GET /api/v2/amtProvisioningCertificates

The **GET /api/v2/amtProvisioningCertificates** call gets a collection certificates required for Intel® AMT provisioning (**AmtProvisioningCertificates\_GetCollection**)

- Role required: Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User
- Return type: **AmtCertificate**
- Example data: Content-Type = Application/JSON and application/XML.
  - o JSON.

```

[
  {
    "AmtCertificateId": 0,
    "Name": "string",
    "CommonName": "string",
    "NotBefore": "2019-08-01T20:05:00.491Z",
    "NotAfter": "2019-08-01T20:05:00.492Z",
    "HasPrivateKey": true,
    "Thumbprint": "string",
    "IsAmtProvisioningCert": true,
    "InUseByAmtProfile": true,
    "Pem": "string",
    "Enabled": true,
  }
]

```

```

    "TenantId": "00000000-0000-0000-0000-000000000000",
    "CreatedOn": "2019-08-01T20:05:00.492Z",
    "CreatedBy": "00000000-0000-0000-0000-000000000000",
    "ModifiedOn": "2019-08-01T20:05:00.492Z",
    "ModifiedBy": "00000000-0000-0000-0000-000000000000"
  }
]

```

- XML

```

<?xml version="1.0"?>
<Inline Model>
  <AmtCertificateId>1</AmtCertificateId>
  <Name>string</Name>
  <CommonName>string</CommonName>
  <NotBefore>1970-01-01T00:00:00.001Z</NotBefore>
  <NotAfter>1970-01-01T00:00:00.001Z</NotAfter>
  <HasPrivateKey>true</HasPrivateKey>
  <Thumbprint>string</Thumbprint>
  <IsAmtProvisioningCert>true</IsAmtProvisioningCert>
  <InUseByAmtProfile>true</InUseByAmtProfile>
  <Pem>string</Pem>
  <Enabled>true</Enabled>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
</Inline Model>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtCertificate**
  - 401 : Unauthorized
  - 403 : Forbidden
  - 500 : Internal server error

## GET /api/v2/amtProvisioningCertificates/{amtCertificateId}

The **GET /api/v2/amtProvisioningCertificates/{amtCertificateId}** call gets a certificate used for Intel® AMT provisioning by ID (**AmtProvisioningCertificates\_GetById**).

- Role required: Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User
- The path parameter is **amtCertificateId**.
- Return type: **AmtCertificate**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
```

```

"AmtCertificateId": 0,
"Name": "string",
"CommonName": "string",
"NotBefore": "2019-08-01T20:05:00.508Z",
"NotAfter": "2019-08-01T20:05:00.508Z",
"HasPrivateKey": true,
"Thumbprint": "string",
"IsAmtProvisioningCert": true,
"InUseByAmtProfile": true,
"Pem": "string",
"Enabled": true,
"TenantId": "00000000-0000-0000-0000-000000000000",
"CreatedOn": "2019-08-01T20:05:00.508Z",
"CreatedBy": "00000000-0000-0000-0000-000000000000",
"ModifiedOn": "2019-08-01T20:05:00.508Z",
"ModifiedBy": "00000000-0000-0000-0000-000000000000"
}

```

- XML

```

<?xml version="1.0"?>
<AmtCertificate>
  <AmtCertificateId>1</AmtCertificateId>
  <Name>string</Name>
  <CommonName>string</CommonName>
  <NotBefore>1970-01-01T00:00:00.001Z</NotBefore>
  <NotAfter>1970-01-01T00:00:00.001Z</NotAfter>
  <HasPrivateKey>true</HasPrivateKey>
  <Thumbprint>string</Thumbprint>
  <IsAmtProvisioningCert>true</IsAmtProvisioningCert>
  <InUseByAmtProfile>true</InUseByAmtProfile>
  <Pem>string</Pem>
  <Enabled>true</Enabled>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
</AmtCertificate>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtCertificate**
  - 401 : Unauthorized
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal server error

**DELETE /api/v2/amtProvisioningCertificates/{amtCertificateId}**

The **DELETE /api/v2/amtProvisioningCertificates/{amtCertificateId}** call deletes a certificate used for Intel® AMT provisioning, as well as the next ones in the chain serially if they are not a part of another chain. (**AmtProvisioningCertificates\_DeleteById**).

- Role required: Tenant Administrator
- The path parameter is **amtCertificateId**, The Intel® AMT Provisioning Certificates.
- Response types:
  - 204 : Successful request, no content
  - 401 : Unauthorized
  - 403 : Forbidden
  - 404 : Not found
  - 409 : Conflict: certificate is in use by AMT profile
  - 500 : Internal server error

## GET /api/v2/amtProvisioningCertificates/{amtCertificateId}/getFile

The **GET /api/v2/amtProvisioningCertificates/{amtCertificateId}/getFile** call gets the .CER file of an Intel® AMT provisioning certificate, or a certificate in its chain, by ID (**AmtProvisioningCertificates\_GetFile**)

- Role required: Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User
- The filename of the .CER file will be specified in the "filename" parameter of the "Content-Disposition" header.
- The path parameter is **amtCertificateId**, The Intel® AMT Provisioning Certificates.
- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
- Response types:
  - 200 : Successful request, attachment returned
  - 401 : Unauthorized
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal server error

## POST /api/v2/amtProvisioningCertificates/uploadPfx

The **POST /api/v2/amtProvisioningCertificates/uploadPfx** call upload .PFX of Intel® AMT provisioning certificate (**AmtProvisioningCertificates\_UploadPfx**)

- Role required: Tenant Administrator
- To import an Intel® AMT provisioning certificate, POST with enctype "multipart/form-data", the .PFX file attached, and the following parameters:  
<br /><b>name:</b> certificate nickname<br /><b>password:</b> password for the .PFX file<br />
- Return type: **AmtCertificate**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "AmtCertificateId": 0,
    "Name": "string",
    "CommonName": "string",
    "NotBefore": "2019-08-01T20:05:00.522Z",
    "NotAfter": "2019-08-01T20:05:00.522Z",
    "HasPrivateKey": true,
    "Thumbprint": "string",
    "IsAmtProvisioningCert": true,
    "InUseByAmtProfile": true,
```

```

    "Pem": "string",
    "Enabled": true,
    "TenantId": "00000000-0000-0000-0000-000000000000",
    "CreatedOn": "2019-08-01T20:05:00.522Z",
    "CreatedBy": "00000000-0000-0000-0000-000000000000",
    "ModifiedOn": "2019-08-01T20:05:00.522Z",
    "ModifiedBy": "00000000-0000-0000-0000-000000000000"
  }
]

```

- o XML

```

<?xml version="1.0"?>
<Inline Model>
  <AmtCertificateId>1</AmtCertificateId>
  <Name>string</Name>
  <CommonName>string</CommonName>
  <NotBefore>1970-01-01T00:00:00.001Z</NotBefore>
  <NotAfter>1970-01-01T00:00:00.001Z</NotAfter>
  <HasPrivateKey>true</HasPrivateKey>
  <Thumbprint>string</Thumbprint>
  <IsAmtProvisioningCert>true</IsAmtProvisioningCert>
  <InUseByAmtProfile>true</InUseByAmtProfile>
  <Pem>string</Pem>
  <Enabled>true</Enabled>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
</Inline Model>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **AmtCertificate**
  - o 400 : Invalid file
  - o 401 : Unauthorized
  - o 403 : Forbidden
  - o 404 : Not found
  - o 409 : Conflict: certificate thumbprint already exists
  - o 415 : Unsupported media type
  - o 500 : Internal Server Error

## POST /api/v2/amtProvisioningCertificates/upload

The **POST /api/v2/amtProvisioningCertificates/upload** call upload .PFX general certificates (**AmtProvisioningCertificates\_Upload**)

- Role required: Tenant Administrator

- To import an certificate, POST with enctype "multipart/form-data", the .PFX file attached, and the following parameters:<br /><b>name:</b> certificate nickname<br /><b>password:</b> password for the .PFX file<br />
- Return type: **AmtCertificate**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "AmtCertificateId": 0,
    "Name": "string",
    "CommonName": "string",
    "NotBefore": "2019-08-01T20:05:00.534Z",
    "NotAfter": "2019-08-01T20:05:00.534Z",
    "HasPrivateKey": true,
    "Thumbprint": "string",
    "IsAmtProvisioningCert": true,
    "InUseByAmtProfile": true,
    "Pem": "string",
    "Enabled": true,
    "TenantId": "00000000-0000-0000-0000-000000000000",
    "CreatedOn": "2019-08-01T20:05:00.535Z",
    "CreatedBy": "00000000-0000-0000-0000-000000000000",
    "ModifiedOn": "2019-08-01T20:05:00.535Z",
    "ModifiedBy": "00000000-0000-0000-0000-000000000000"
  }
]
```

- XML

```
<?xml version="1.0"?>
<Inline Model>
  <AmtCertificateId>1</AmtCertificateId>
  <Name>string</Name>
  <CommonName>string</CommonName>
  <NotBefore>1970-01-01T00:00:00.001Z</NotBefore>
  <NotAfter>1970-01-01T00:00:00.001Z</NotAfter>
  <HasPrivateKey>true</HasPrivateKey>
  <Thumbprint>string</Thumbprint>
  <IsAmtProvisioningCert>true</IsAmtProvisioningCert>
  <InUseByAmtProfile>true</InUseByAmtProfile>
  <Pem>string</Pem>
  <Enabled>true</Enabled>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
</Inline Model>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json

- application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtCertificate**
  - 400 : Invalid file
  - 401 : Unauthorized
  - 403 : Forbidden
  - 404 : Not found
  - 415 : Unsupported media type
  - 500 : Internal Server Error

## 3.10 Intel® AMT Setups

### GET /api/v2/amtSetups/endpoints/{amtSetupId}

- The **GET /api/v2/amtSetups/endpoints/{amtSetupId}** call gets the AmtSetup (or provisioning record) of an Endpoint (**AmtSetups\_GetManualProvisioningRecordById**).
- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with View right to the resource.
- The path parameter is **amtSetupId**. Use value of EndpointId. Hex string with 64 characters and without the 0x prefix
- Return type: **AmtSetup[String]**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "AmtSetupId": "string",
  "Type": "auto",
  "PID": "string",
  "Creation": "2019-08-01T20:05:00.548Z",
  "SetsRandomMebxPassword": true,
  "Profile": {
    "UsesTLS": true,
    "UsesCIRA": true,
    "UsesEmaAccount": true,
    "CiraIntranetSuffix": "string",
    "AdminPassword": "string",
    "MebxPasswordState": "doNotSetMebxPassword",
    "ProvisionCertificateHash": "string",
    "ProvisioningDnsSuffix": "string",
    "PPS": "string"
  },
  "State": 0,
  "StateString": "string",
  "ExtraAmtInfo": {
    "LastUpdated": "2019-08-01T20:05:00.549Z",
    "HECIDriver": {
      "Name": "string",
      "Status": true,
      "Details": "string"
    },
    "CorporateDNS": {
      "Name": "string",
      "Status": true,
      "Details": "string"
    }
  }
}
```



```

    },
    "CorporateVPN": {
        "Name": "string",
        "Status": true,
        "Details": "string"
    },
    "IntelNic": {
        "Name": "string",
        "Status": true,
        "Details": "string"
    }
},
"AmtProfileId": 0
}

```

○ XML

```

<?xml version="1.0"?>
<AmtSetup[String]>
  <AmtSetupId>string</AmtSetupId>
  <Type>auto</Type>
  <PID>string</PID>
  <Creation>1970-01-01T00:00:00.001Z</Creation>
  <SetsRandomMebxPassword>true</SetsRandomMebxPassword>
  <Profile>
    <UsesTLS>true</UsesTLS>
    <UsesCIRA>true</UsesCIRA>
    <UsesEmaAccount>true</UsesEmaAccount>
    <CiraIntranetSuffix>string</CiraIntranetSuffix>
    <AdminPassword>string</AdminPassword>
    <MebxPasswordState>doNotSetMebxPassword</MebxPasswordState>
    <ProvisionCertificateHash>string</ProvisionCertificateHash>
    <ProvisioningDnsSuffix>string</ProvisioningDnsSuffix>
    <PPS>string</PPS>
  </Profile>
  <State>1</State>
  <StateString>string</StateString>
  <ExtraAmtInfo>
    <LastUpdated>1970-01-01T00:00:00.001Z</LastUpdated>
    <HECIDriver>
      <Name>string</Name>
      <Status>true</Status>
      <Details>string</Details>
    </HECIDriver>
    <CorporateDNS>
      <Name>string</Name>
      <Status>true</Status>
      <Details>string</Details>
    </CorporateDNS>
    <CorporateVPN>
      <Name>string</Name>
      <Status>true</Status>
      <Details>string</Details>
    </CorporateVPN>
  </ExtraAmtInfo>
</AmtSetup>

```

```

    <IntelNic>
      <Name>string</Name>
      <Status>true</Status>
      <Details>string</Details>
    </IntelNic>
  </ExtraAmtInfo>
  <AmtProfileId>1</AmtProfileId>
</AmtSetup[String]>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtSetup[String]**
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal server error

## DELETE /api/v2/amtSetups/endpoints/{amtSetupId}

The **DELETE /api/v2/amtSetups/endpoints/{amtSetupId}** call deletes the AmtSetup (or provisioning record) of an Endpoint (**AmtSetups\_DeleteManualProvisioningRecordById**).

- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with Execute right to the endpoint.
- The path parameter is **amtSetupId**. Use value of EndpointId. Hex string with 64 characters and without the 0x prefix
- Response types:
  - 204 : Successful request, no content
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal server error

## GET /api/v2/amtSetups/auto/{amtSetupId}

The **GET /api/v2/amtSetups/auto/{amtSetupId}** call gets the AmtSetup (or auto-provisioning setting) of an Endpoint Group (**AmtSetups\_GetAutoProvisioningRecordById**)

- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with View right to the resource.
- The path parameter is **amtSetupId**. Use value of EndpointGroupId. Hex string with 64 characters and without the 0x prefix
- Return type: **AmtSetup[String]**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "AmtSetupId": "string",
  "Type": "auto",
  "PID": "string",
  "Creation": "2019-08-01T20:05:00.590Z",
  "SetsRandomMebxPassword": true,
  "Profile": {

```

```

    "UsesTLS": true,
    "UsesCIRA": true,
    "UsesEmaAccount": true,
    "CiraIntranetSuffix": "string",
    "AdminPassword": "string",
    "MebxPasswordState": "doNotSetMebxPassword",
    "ProvisionCertificateHash": "string",
    "ProvisioningDnsSuffix": "string",
    "PPS": "string"
  },
  "State": 0,
  "StateString": "string",
  "ExtraAmtInfo": {
    "LastUpdated": "2019-08-01T20:05:00.590Z",
    "HECIDriver": {
      "Name": "string",
      "Status": true,
      "Details": "string"
    },
    "CorporateDNS": {
      "Name": "string",
      "Status": true,
      "Details": "string"
    },
    "CorporateVPN": {
      "Name": "string",
      "Status": true,
      "Details": "string"
    },
    "IntelNic": {
      "Name": "string",
      "Status": true,
      "Details": "string"
    }
  },
  "AmtProfileId": 0
}

```

- XML

```

<?xml version="1.0"?>
<AmtSetup[String]>
  <AmtSetupId>string</AmtSetupId>
  <Type>auto</Type>
  <PID>string</PID>
  <Creation>1970-01-01T00:00:00.001Z</Creation>
  <SetsRandomMebxPassword>true</SetsRandomMebxPassword>
  <Profile>
    <UsesTLS>true</UsesTLS>
    <UsesCIRA>true</UsesCIRA>
    <UsesEmaAccount>true</UsesEmaAccount>
    <CiraIntranetSuffix>string</CiraIntranetSuffix>
    <AdminPassword>string</AdminPassword>
    <MebxPasswordState>doNotSetMebxPassword</MebxPasswordState>
  </Profile>
</AmtSetup[String]>

```

```

    <ProvisionCertificateHash>string</ProvisionCertificateHash>
    <ProvisioningDnsSuffix>string</ProvisioningDnsSuffix>
    <PPS>string</PPS>
  </Profile>
  <State>1</State>
  <StateString>string</StateString>
  <ExtraAmtInfo>
    <LastUpdated>1970-01-01T00:00:00.001Z</LastUpdated>
    <HECIDriver>
      <Name>string</Name>
      <Status>true</Status>
      <Details>string</Details>
    </HECIDriver>
    <CorporateDNS>
      <Name>string</Name>
      <Status>true</Status>
      <Details>string</Details>
    </CorporateDNS>
    <CorporateVPN>
      <Name>string</Name>
      <Status>true</Status>
      <Details>string</Details>
    </CorporateVPN>
    <IntelNic>
      <Name>string</Name>
      <Status>true</Status>
      <Details>string</Details>
    </IntelNic>
  </ExtraAmtInfo>
  <AmtProfileId>1</AmtProfileId>
</AmtSetup[String]>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtSetup[String]**
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal server error

## DELETE /api/v2/amtSetups/auto/{amtSetupId}

The **DELETE /api/v2/amtSetups/auto/{amtSetupId}** call deletes the AmtSetup (or auto-provisioning setting) of an Endpoint Group (**AmtSetups\_DeleteAutoProvisioningRecordById**).

- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with Execute right to the endpoint.
- The path parameter is **amtSetupId**. Use value of EndpointGroupId. Hex string with 64 characters and without the 0x prefix
- Response types:
  - 204 : Successful request, no content

- 403 : Forbidden
- 404 : Not found
- 500 : Internal server error

## POST /api/v2/amtSetups/endpoints/provision

The **POST /api/v2/amtSetups/endpoints/provision** call submits a request to provision an Intel® AMT Endpoint (**AmtSetups\_Provision**).

- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with Execute right to the endpoint.
- Certificate provisioning will be requested only if an **AMTProvisioningCertificateId** is provided, otherwise host-based-provisioning will be requested.
- **AdminCredential.Password** requirements:
  - between 8 and 32 characters
  - contains at least one number
  - contains both lowercase and uppercase alpha characters
  - contains at least one special character: '!', '@', '#', '\$', '%', '^', '&', '\*', '(', ')', '-', '+'
- **UsesTLS** and **UsesCira** cannot be both true or both false. You need to choose one.
- The path parameter is **ManualProvisioningRequest**. AMT Setup request object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **ManualProvisioningRequest**. This is the Intel® AMT Setups object.

```
{
  "EndpointId": "string",
  "UsesTls": true,
  "UsesCira": true,
  "SetsRandomMebxPassword": true,
  "UsesEmaAccount": true,

  "AdminCredential": {
    "Password": "string"
  },
  "AmtCertificateId": 0
}
```

- Return type: **AmtSetup[String]**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "AmtSetupId": "string",
  "Type": "auto",
  "PID": "string",
  "Creation": "2019-08-01T20:05:00.600Z",
  "SetsRandomMebxPassword": true,
  "Profile": {
    "UsesTLS": true,
    "UsesCIRA": true,
    "UsesEmaAccount": true,
  }
}
```

```

    "AdminPassword": "string",
    "MebxPasswordState": "doNotSetMebxPassword",
    "ProvisionCertificateHash": "string",
    "ProvisioningDnsSuffix": "string",
    "PPS": "string"
  },
  "State": 0,
  "StateString": "string",
  "ExtraAmtInfo": {
    "LastUpdated": "2019-08-01T20:05:00.600Z",
    "HECIDriver": {
      "Name": "string",
      "Status": true,
      "Details": "string"
    },
    "CorporateDNS": {
      "Name": "string",
      "Status": true,
      "Details": "string"
    },
    "CorporateVPN": {
      "Name": "string",
      "Status": true,
      "Details": "string"
    },
    "IntelNic": {
      "Name": "string",
      "Status": true,
      "Details": "string"
    }
  },
  "AmtProfileId": 0
}

```

- XML

```

<?xml version="1.0"?>
<AmtSetup[String]>
  <AmtSetupId>string</AmtSetupId>
  <Type>auto</Type>
  <PID>string</PID>
  <Creation>1970-01-01T00:00:00.001Z</Creation>
  <SetsRandomMebxPassword>true</SetsRandomMebxPassword>
  <Profile>
    <UsesTLS>true</UsesTLS>
    <UsesCIRA>true</UsesCIRA>
    <UsesEmaAccount>true</UsesEmaAccount>

    <AdminPassword>string</AdminPassword>
    <MebxPasswordState>doNotSetMebxPassword</MebxPasswordState>
    <ProvisionCertificateHash>string</ProvisionCertificateHash>
    <ProvisioningDnsSuffix>string</ProvisioningDnsSuffix>
    <PPS>string</PPS>
  </Profile>
</AmtSetup[String]>

```

```

</Profile>
<State>1</State>
<StateString>string</StateString>
<ExtraAmtInfo>
  <LastUpdated>1970-01-01T00:00:00.001Z</LastUpdated>
  <HECIDDriver>
    <Name>string</Name>
    <Status>true</Status>
    <Details>string</Details>
  </HECIDDriver>
  <CorporateDNS>
    <Name>string</Name>
    <Status>true</Status>
    <Details>string</Details>
  </CorporateDNS>
  <CorporateVPN>
    <Name>string</Name>
    <Status>true</Status>
    <Details>string</Details>
  </CorporateVPN>
  <IntelNic>
    <Name>string</Name>
    <Status>true</Status>
    <Details>string</Details>
  </IntelNic>
</ExtraAmtInfo>
<AmtProfileId>1</AmtProfileId>
</AmtSetup[String]>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, result returned **AmtSetup[String]**
  - 400 : Bad request
  - 403 : Forbidden
  - 404 : Not found
  - 409 : Endpoint's AMT is already provisioned
  - 500 : Internal server error

## POST /api/v2/amtSetups/endpoints/unprovision

The **POST /api/v2/amtSetups/endpoints/unprovision** call submits a request to unprovision an Intel® AMT capable endpoint (**AmtSetups\_Unprovision**)

- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with Execute right to the endpoint.
- **Warning: This operation will reset the Intel® AMT device to factory settings!**
- The Intel® AMT admin password is optional and should only be used to unprovision an Intel® AMT endpoint that was provisioned outside of Intel® EMA, or that has no AmtSetup (or Provisioning Record) in the database.

- If the Intel® AMT capable endpoint was provisioned with Intel® EMA, and the corresponding AmtSetup record is in the database, then the Intel® AMT admin password is unnecessary.
- AdminCredential.Password requirements:
  - between 8 and 32 characters
  - contains at least one number
  - contains both lowercase and uppercase alpha characters
  - contains at least one special character
- The path parameter is **ManualUnprovisioningRequest**. An unprovision Intel® AMT request.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **ManualUnprovisioningRequest**. This is the Intel® AMT Setups object.

```
{
  "EndpointId": "string",
  "AdminCredential": {
    "Password": "string"
  }
}
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
- Response types:
  - 204 : Successful request, no content
  - 400 : Bad request
  - 403 : Forbidden
  - 404 : Not found
  - 409 : AMT endpoint not provisioned, or TCPRELAY not enabled by policy
  - 500 : Internal server error

## POST /api/v2/amtSetups/auto/set

The **POST /api/v2/amtSetups/auto/set** call sets Intel® AMT auto-provisioning for an Endpoint Group. (**AmtSetups\_SetAutoProvisioningForEndpointGroup**)

- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with Execute right to the endpoint.
- For certificate provisioning, provide an AmtCertificateId in the request, otherwise Host-based-provisioning will be set.
- To unset auto-provisioning, delete the AmtSetup record using the EndpointGroupId as AmtSetupId.
- The path parameter is **AutoProvisioningRequest**. The Intel® AMT Setups.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AutoProvisioningRequest**. This is the Intel® AMT Setups object.

```
{
  "EndpointGroupId": "string",
  "AmtProfileId": 0,
  "SetsRandomMebxPassword": true,
}
```



```

"AdminCredential": {
  "Password": "string"
},
"UsesEmaAccount": true,
"AmtCertificateId": 0
}

```

- Return type: **AmtSetup[String]**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "AmtSetupId": "string",
  "Type": "auto",
  "PID": "string",
  "Creation": "2019-08-01T20:05:00.637Z",
  "SetsRandomMebxPassword": true,
  "Profile": {
    "UsesTLS": true,
    "UsesCIRA": true,
    "UsesEmaAccount": true,
    "CiraIntranetSuffix": "string",
    "AdminPassword": "string",
    "MebxPasswordState": "doNotSetMebxPassword",
    "ProvisionCertificateHash": "string",
    "ProvisioningDnsSuffix": "string",
    "PPS": "string"
  },
  "State": 0,
  "StateString": "string",
  "ExtraAmtInfo": {
    "LastUpdated": "2019-08-01T20:05:00.637Z",
    "HECIDriver": {
      "Name": "string",
      "Status": true,
      "Details": "string"
    },
    "CorporateDNS": {
      "Name": "string",
      "Status": true,
      "Details": "string"
    },
  },
}

```

- XML

```

<?xml version="1.0"?>
<AmtSetup[String]>
  <AmtSetupId>string</AmtSetupId>
  <Type>auto</Type>
  <PID>string</PID>
  <Creation>1970-01-01T00:00:00.001Z</Creation>
  <SetsRandomMebxPassword>true</SetsRandomMebxPassword>
  <Profile>
    <UsesTLS>true</UsesTLS>
  </Profile>
</AmtSetup[String]>

```

```

<UsesCIRA>true</UsesCIRA>
<UsesEmaAccount>true</UsesEmaAccount>
<CiraIntranetSuffix>string</CiraIntranetSuffix>
<AdminPassword>string</AdminPassword>
<MebxPasswordState>doNotSetMebxPassword</MebxPasswordState>
<ProvisionCertificateHash>string</ProvisionCertificateHash>
<ProvisioningDnsSuffix>string</ProvisioningDnsSuffix>
<PPS>string</PPS>
</Profile>
<State>1</State>
<StateString>string</StateString>
<ExtraAmtInfo>
<LastUpdated>1970-01-01T00:00:00.001Z</LastUpdated>
<HECIDriver>
<Name>string</Name>
<Status>true</Status>
<Details>string</Details>
</HECIDriver>
<CorporateDNS>
<Name>string</Name>
<Status>true</Status>
<Details>string</Details>
</CorporateDNS>
<CorporateVPN>
<Name>string</Name>
<Status>true</Status>
<Details>string</Details>
</CorporateVPN>
<IntelNic>
<Name>string</Name>
<Status>true</Status>
<Details>string</Details>
</IntelNic>
</ExtraAmtInfo>
<AmtProfileId>1</AmtProfileId>
</AmtSetup[String]>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **AmtSetup[String]**
  - 400 : Bad Request
  - 403 : Forbidden
  - 404 : Not found
  - 409 : Conflict
  - 500 : Internal server error

## 3.11 Endpoint Groups

### GET /api/v2/endpointGroups

The **GET /api/v2/endpointGroups** call gets a collection of Endpoint Group summaries.  
(**EndpointGroups\_GetSummariesCollection**)

- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with Read right to the resource.
- Return type: **EndpointGroupSummary**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "EndpointGroupId": "string",
    "Name": "string",
    "EndpointCount": 0
  }
]
```

- XML

```
<?xml version="1.0"?>
<Inline Model>
  <EndpointGroupId>string</EndpointGroupId>
  <Name>string</Name>
  <EndpointCount>1</EndpointCount>
</Inline Model>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointGroupSummary**
  - 500 : Internal Server Error

## POST /api/v2/endpointGroups

The **POST /api/v2/endpointGroups** call creates a new Endpoint Group (**EndpointGroups\_Post**)

- Role required: Tenant Administrator or Endpoint Group Creator.
- Password is needed when the endpoint group's policy needs to be modified. Currently, the modification of the policy is not supported yet.
- Password requirements:
  - at least 8 characters
  - contains at least one number
  - contains both lowercase and uppercase alpha characters
  - contains at least one special character
- UserConsentKVM\_Timeout requirements:
  - timeout value must fall between 1 and 65
  - timeout in seconds
  - this value will be taken if the AllowUserConsentKVM web permission is enabled
- The path parameter is **NewEndpointGroup**.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json

- application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **NewEndpointGroup**. This is the Endpoint Groups object.

```
{
  "Name": "string",
  "Description": "string",
  "Password": "string",
  "UserConsentKVM_Timeout": 0,
  "Permissions": {
    "AllowWakeup": true,
    "AllowSleep": true,
    "AllowReset": true,
    "AllowTcpCommunication": true,
    "AllowAlert": true,
    "AllowConsole": true,
    "AllowKvm": true,
    "AllowFileAccess": true,
    "AllowWmi": true,
    "AllowLocation": true,
    "AllowP2P": true,
    "AllowUserConsentKVM": true
  }
}
```

- Return type: **EndpointGroup**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Name": "string",
  "Description": "string",
  "EndpointGroupId": "string",
  "TenantId": "string",
  "LastUpdated": "2019-08-01T20:05:00.657Z",
  "UserConsentKVM_Timeout": 0,
  "Permissions": {
    "AllowWakeup": true,
    "AllowSleep": true,
    "AllowReset": true,
    "AllowTcpCommunication": true,
    "AllowAlert": true,
    "AllowConsole": true,
    "AllowKvm": true,
    "AllowFileAccess": true,
    "AllowWmi": true,
    "AllowLocation": true,
    "AllowP2P": true,
    "AllowUserConsentKVM": true
  }
}
```

- XML

```
<?xml version="1.0"?>
<EndpointGroup>
  <Name>string</Name>
  <Description>string</Description>
  <EndpointGroupId>string</EndpointGroupId>
  <TenantId>string</TenantId>
  <LastUpdated>1970-01-01T00:00:00.001Z</LastUpdated>
  <UserConsentKVM_Timeout>1</UserConsentKVM_Timeout>
  <Permissions>
    <AllowWakeup>true</AllowWakeup>
    <AllowSleep>true</AllowSleep>
    <AllowReset>true</AllowReset>
    <AllowTcpCommunication>true</AllowTcpCommunication>
    <AllowAlert>true</AllowAlert>
    <AllowConsole>true</AllowConsole>
    <AllowKvm>true</AllowKvm>
    <AllowFileAccess>true</AllowFileAccess>
    <AllowWmi>true</AllowWmi>
    <AllowLocation>true</AllowLocation>
    <AllowP2P>true</AllowP2P>
    <AllowUserConsentKVM>true</AllowUserConsentKVM>
  </Permissions>
</EndpointGroup>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointGroup**
  - 400 : Invalid ModelState
  - 409 : Conflict
  - 500 : Internal Server Error

## GET /api/v2/endpointGroups/{endpointGroupId}

The **GET /api/v2/endpointGroups/{endpointGroupId}** call gets the details of a Endpoint Group by ID. (**EndpointGroups\_GetEndpointGroupById**)

- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with Read right to the resource.
- The path parameter is **endpointGroupId**. Hex string with 64 characters and without the 0x prefix
- Return type: **EndpointGroup**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Name": "string",
  "Description": "string",
  "EndpointGroupId": "string",
  "TenantId": "string",
  "LastUpdated": "2019-08-01T20:05:00.674Z",
  "UserConsentKVM_Timeout": 0,
```

```

    "Permissions": {
      "AllowWakeup": true,
      "AllowSleep": true,
      "AllowReset": true,
      "AllowTcpCommunication": true,
      "AllowAlert": true,
      "AllowConsole": true,
      "AllowKvm": true,
      "AllowFileAccess": true,
      "AllowWmi": true,
      "AllowLocation": true,
      "AllowP2P": true,
      "AllowUserConsentKVM": true
    }
  }
}

```

- XML

```

<?xml version="1.0"?>
<EndpointGroup>
  <Name>string</Name>
  <Description>string</Description>
  <EndpointGroupId>string</EndpointGroupId>
  <TenantId>string</TenantId>
  <LastUpdated>1970-01-01T00:00:00.001Z</LastUpdated>
  <UserConsentKVM_Timeout>1</UserConsentKVM_Timeout>
  <Permissions>
    <AllowWakeup>true</AllowWakeup>
    <AllowSleep>true</AllowSleep>
    <AllowReset>true</AllowReset>
    <AllowTcpCommunication>true</AllowTcpCommunication>
    <AllowAlert>true</AllowAlert>
    <AllowConsole>true</AllowConsole>
    <AllowKvm>true</AllowKvm>
    <AllowFileAccess>true</AllowFileAccess>
    <AllowWmi>true</AllowWmi>
    <AllowLocation>true</AllowLocation>
    <AllowP2P>true</AllowP2P>
    <AllowUserConsentKVM>true</AllowUserConsentKVM>
  </Permissions>
</EndpointGroup>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointGroup**
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal Server Error

## PUT /api/v2/endpointGroups/{endpointGroupId}

The **PUT /api/v2/endpointGroups/{endpointGroupId}** call updates an Endpoint Group. (**EndpointGroups\_Put**)

- Role required: Tenant Administrator or Endpoint Group Creator.
- The only property that can be updated is Description.
- The path parameter is **EndpointGroupUpdate**. The Endpoint Groups.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointGroupUpdate**. This is the Endpoint Groups object.

```
{
  "Description": "string",
  "EndpointGroupId": "string"
}
```

- Return type: **EndpointGroup**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Name": "string",
  "Description": "string",
  "EndpointGroupId": "string",
  "TenantId": "string",
  "LastUpdated": "2019-08-01T20:05:00.683Z",
  "UserConsentKVM_Timeout": 0,
  "Permissions": {
    "AllowWakeup": true,
    "AllowSleep": true,
    "AllowReset": true,
    "AllowTcpCommunication": true,
    "AllowAlert": true,
    "AllowConsole": true,
    "AllowKvm": true,
    "AllowFileAccess": true,
    "AllowWmi": true,
    "AllowLocation": true,
    "AllowP2P": true,
    "AllowUserConsentKVM": true
  }
}
```

- XML

```
<?xml version="1.0"?>
<EndpointGroup>
  <Name>string</Name>
  <Description>string</Description>
  <EndpointGroupId>string</EndpointGroupId>
  <TenantId>string</TenantId>
```

```

<LastUpdated>1970-01-01T00:00:00.001Z</LastUpdated>
<UserConsentKVM_Timeout>1</UserConsentKVM_Timeout>
<Permissions>
  <AllowWakeup>true</AllowWakeup>
  <AllowSleep>true</AllowSleep>
  <AllowReset>true</AllowReset>
  <AllowTcpCommunication>true</AllowTcpCommunication>
  <AllowAlert>true</AllowAlert>
  <AllowConsole>true</AllowConsole>
  <AllowKvm>true</AllowKvm>
  <AllowFileAccess>true</AllowFileAccess>
  <AllowWmi>true</AllowWmi>
  <AllowLocation>true</AllowLocation>
  <AllowP2P>true</AllowP2P>
  <AllowUserConsentKVM>true</AllowUserConsentKVM>
</Permissions>
</EndpointGroup>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointGroup**
  - 400 : Bad Request
  - 403 : Forbidden
  - 404 : Not found
  - 409 : Conflict
  - 500 : Internal Server Error

## DELETE /api/v2/endpointGroups/{endpointGroupId}

The **DELETE /api/v2/endpointGroups/{endpointGroupId}** call deletes an Endpoint Group. (**EndpointGroups\_Delete**).

- Role required: Tenant Administrator or Endpoint Group Creator.
- The path parameter is **endpointGroupId**. Hex string with 64 characters and without the 0x prefix
- Response types:
  - 204 : Successful request, no content
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal server error

## GET /api/v2/endpointGroups/{endpointGroupId}/getMshFile

The **GET /api/v2/endpointGroups/{endpointGroupId}/getMshFile** call gets the .msh file associated with an Endpoint Group. (**EndpointGroups\_GetMshFile**).

- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with Read right to the resource.
- The path parameter is **endpointGroupId**. Hex string with 64 characters and without the 0x prefix This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json



- text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 403 : Forbidden
  - 404 : Not found

## GET /api/v2/endpointGroups/{endpointGroupId}/getUserAccessRights

The **GET /api/v2/endpointGroups/{endpointGroupId}/getUserAccessRights** call gets the highest access right for a User to an Endpoint Group. (**EndpointGroups\_GetMaxUserRightsByEndpointGroup**)

- Role required: Tenant Administrator, Endpoint Group Creator or Endpoint Group User.
- The access rights that a User has to an Endpoint Group are obtained by way of membership in User Group(s) assigned to that Endpoint Group.
- The path parameter is **endpointGroupId**. Hex string with 64 characters and without the 0x prefix
- Return type: **UserAccessRights**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "AccessRightsId": 0,
  "AccessRights": "string"
}
```

- XML

```
<?xml version="1.0"?>
<UserAccessRights>
  <AccessRightsId>1</AccessRightsId>
  <AccessRights>string</AccessRights>
</UserAccessRights>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request **UserAccessRights**
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal server error

## 3.12 Endpoint IB Operations

### POST /api/v2/endpointIBOperations/reboot

The **POST /api/v2/endpointIBOperations/reboot** call submits batch request to reboot Endpoints inband (**EndpointIBOperations\_Reboot**)

- Roles required: Tenant Administrator, Endpoint Group Creator or Endpoint Group User
- The action must be allowed by the Endpoint's Endpoint Group.

- With an HTTP 200, the response list will contain only those EndpointIds for which a request was put on the message bus.
- EndpointIds for which a request was not put on the message bus for one reason or another will not appear in the response list.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointGroupUpdate**.

```
[
  {
    "EndpointId": "string"
  }
]
```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
      "EndpointId": "string"
    }
  ]
}
```

- XML

```
<?xml version="1.0"?>
<EndpointOperationsList>
```

```

    <Accepted>
      <EndpointId>string</EndpointId>
    </Accepted>
    <Forbidden>
      <EndpointId>string</EndpointId>
    </Forbidden>
    <NotFound>
      <EndpointId>string</EndpointId>
    </NotFound>
    <BadRequest>
      <EndpointId>string</EndpointId>
    </BadRequest>
    <InternalServerError>
      <EndpointId>string</EndpointId>
    </InternalServerError>
  </EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

## POST /api/v2/endpointIBOperations/sleep

The **POST /api/v2/endpointIBOperations/sleep** call submits batch request to sleep Endpoints inband (EndpointIBOperations\_Sleep)

- Roles required: Tenant Administrator, Endpoint Group Creator or Endpoint Group User
- The action must be allowed by the Endpoint's Endpoint Group.
- With an HTTP 200, the response list will contain only those EndpointIds for which a request was put on the message bus.
- EndpointIds for which a request was not put on the message bus for one reason or another will not appear in the response list.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointGroupUpdate**. This is the Endpoint IB Operations object.

```

[
  {
    "EndpointId": "string"
  }
]

```

- Return type: **EndpointOperationsList**

- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
      "EndpointId": "string"
    }
  ]
}
```

- XML

```
<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
  <InternalServerError>
    <EndpointId>string</EndpointId>
  </InternalServerError>
</EndpointOperationsList>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header

- application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

## POST /api/v2/endpointIBOperations/hibernate

The **POST /api/v2/endpointIBOperations/hibernate** call submits batch request to hibernate Endpoints inband (**EndpointIBOperations\_Hibernate**)

- Roles required: Tenant Administrator, Endpoint Group Creator or Endpoint Group User
- The action must be allowed by the Endpoint's Endpoint Group.
- With an HTTP 200, the response list will contain only those EndpointIds for which a request was put on the message bus.
- EndpointIds for which a request was not put on the message bus for one reason or another will not appear in the response list.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointGroupUpdate**. This is the Endpoint IB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
}
```

```

    "BadRequest": [
      {
        "EndpointId": "string"
      }
    ],
    "InternalServerError": [
      {
        "EndpointId": "string"
      }
    ]
  }
}

```

- XML

```

<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
  <InternalServerError>
    <EndpointId>string</EndpointId>
  </InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

## POST /api/v2/endpointIBOperations/shutdown

The **POST /api/v2/endpointIBOperations/shutdown** call submits batch request to shutdown Endpoints inband (**EndpointIBOperations\_Shutdown**)

- Roles required: Tenant Administrator, Endpoint Group Creator or Endpoint Group User
- The action must be allowed by the Endpoint's Endpoint Group.
- With an HTTP 200, the response list will contain only those EndpointIds for which a request was put on the message bus.

- EndpointIds for which a request was not put on the message bus for one reason or another will not appear in the response list.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointGroupUpdate**. This is the Endpoint IB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
      "EndpointId": "string"
    }
  ]
}
```

- XML

```
<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
```

```

</Accepted>
<Forbidden>
  <EndpointId>string</EndpointId>
</Forbidden>
<NotFound>
  <EndpointId>string</EndpointId>
</NotFound>
<BadRequest>
  <EndpointId>string</EndpointId>
</BadRequest>
<InternalServerError>
  <EndpointId>string</EndpointId>
</InternalServerError>
</EndpointOperationsList>

```

- 
- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

## POST /api/v2/endpointIBOperations/alert

The **POST /api/v2/endpointIBOperations/alert** call submits batch request to alert Endpoints inband (**EndpointIBOperations\_Alert**)

- Roles required: Tenant Administrator, Endpoint Group Creator or Endpoint Group User
- The action must be allowed by the Endpoint's Endpoint Group.
- With an HTTP 200, the response list will contain only those EndpointIds for which a request was put on the message bus.
- EndpointIds for which a request was not put on the message bus for one reason or another will not appear in the response list.
- On receiving an alert, an Endpoint's desktop will popup a MessageBox that closes after a specified duration.
- The path parameter is **AlertRequest**. An object containing a list of Endpoints to which an alert will popup with indicated message and for a specified duration.
  - **Message**: Allowed characters: space, enter, 0-9, a-z, A-Z. Maximum length: 510.
  - **Duration**(seconds): A number between 0 and 300, with 0 for indefinite.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint IB Operations object.

```

{
  "EndpointIds": [

```



```

    {
      "EndpointId": "string"
    }
  ],
  "Message": "string",
  "Duration": 0
}

```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
      "EndpointId": "string"
    }
  ]
}

```

- XML

```

<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>

```

```

        <EndpointId>string</EndpointId>
    </BadRequest>
    <InternalServerError>
        <EndpointId>string</EndpointId>
    </InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

### 3.13 Endpoint OOB Operations

#### POST /api/v2/endpointOOBOperations/Multiple/PowerOn

The **POST /api/v2/endpointOOBOperations/Multiple/PowerOn** call submits a batch request to perform an out of band PowerOn operation on multiple endpoints (**EndpointOOBOperations\_MultiplePowerOn**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- It will send also a Wake on LAN (WOL) request to all listed endpoints, but only Intel(R) vPro(TM) endpoints are officially supported.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```

[
  {
    "EndpointId": "string"
  }
]

```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "Accepted": [
    {

```

```

        "EndpointId": "string"
    }
],
"Forbidden": [
    {
        "EndpointId": "string"
    }
],
"NotFound": [
    {
        "EndpointId": "string"
    }
],
"BadRequest": [
    {
        "EndpointId": "string"
    }
],
"InternalServerError": [
    {
        "EndpointId": "string"
    }
]
]
}

```

- o XML

```

<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
  <InternalServerError>
    <EndpointId>string</EndpointId>
  </InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **EndpointOperationsList**

- 400 : Invalid ModelState
- 403 : User not allowed to execute operation
- 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/Sleep/Light

The **POST /api/v2/endpointOOBOperations/Multiple/Sleep/Light** call submits a batch request to perform an out of band Sleep light operation on multiple endpoints (**EndpointOOBOperations\_MultipleSleepLight**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
```

```

    {
      "EndpointId": "string"
    }
  ]
}

```

- XML

```

<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
  <InternalServerError>
    <EndpointId>string</EndpointId>
  </InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/Sleep/Deep

The **POST /api/v2/endpointOOBOperations/Multiple/Sleep/Deep** call submits a batch request to perform an out of band Sleep Deep operation on multiple endpoints (**EndpointOOBOperations\_MultipleSleepDeep**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml

- application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
      "EndpointId": "string"
    }
  ]
}
```

- XML

```
<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
```

```

<BadRequest>
  <EndpointId>string</EndpointId>
</BadRequest>
<InternalServerError>
  <EndpointId>string</EndpointId>
</InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/PowerCycle/OffSoft

The **POST /api/v2/endpointOOBOperations/Multiple/PowerCycle/OffSoft** call submits a batch request to perform an out of band Power Cycle (Off soft) operation on multiple endpoints (**EndpointOOBOperations\_MultiplePowerCycleOffSoft**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```

[
  {
    "EndpointId": "string"
  }
]

```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ]
}

```

```

],
"Forbidden": [
  {
    "EndpointId": "string"
  }
],
"NotFound": [
  {
    "EndpointId": "string"
  }
],
"BadRequest": [
  {
    "EndpointId": "string"
  }
],
"InternalServerError": [
  {
    "EndpointId": "string"
  }
]
}

```

- XML

```

<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
  <InternalServerError>
    <EndpointId>string</EndpointId>
  </InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation



- 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/PowerOff/Hard

The **POST /api/v2/endpointOOBOperations/Multiple/PowerOff/Hard** call submits a batch request to perform an out of band Power off - Hard operation on multiple endpoints

(EndpointOOBOperations\_MultiplePowerOffHard)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
```

```

        "EndpointId": "string"
    }
]
}

```

- o XML

```

<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
  <InternalServerError>
    <EndpointId>string</EndpointId>
  </InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **EndpointOperationsList**
  - o 400 : Invalid ModelState
  - o 403 : User not allowed to execute operation
  - o 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/Hibernate

The **POST /api/v2/endpointOOBOperations/Multiple/Hibernate** call submits a batch request to perform an out of band Hibernate operation on multiple endpoints (**EndpointOOBOperations\_MultipleHibernate**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
  - o application/x-www-form-urlencoded

- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
      "EndpointId": "string"
    }
  ]
}
```

- XML

```
<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
```

```

        <EndpointId>string</EndpointId>
    </BadRequest>
    <InternalServerError>
        <EndpointId>string</EndpointId>
    </InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/PowerOff/Soft

The **POST /api/v2/endpointOOBOperations/Multiple/PowerOff/Soft** call submits a batch request to perform an out of band Power Off-Soft operation on multiple endpoints  
(**EndpointOOBOperations\_MultiplePowerOffSoft**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```

[
  {
    "EndpointId": "string"
  }
]

```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],

```

```

    "Forbidden": [
      {
        "EndpointId": "string"
      }
    ],
    "NotFound": [
      {
        "EndpointId": "string"
      }
    ],
    "BadRequest": [
      {
        "EndpointId": "string"
      }
    ],
    "InternalServerError": [
      {
        "EndpointId": "string"
      }
    ]
  ]
}

```

- XML

```

<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
  <InternalServerError>
    <EndpointId>string</EndpointId>
  </InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/PowerCycle/OffHard

The **POST /api/v2/endpointOOBOperations/Multiple/PowerCycle/OffHard** call submits a batch request to perform an out of band Power Cycle (Off hard) operation on multiple endpoints

(**EndpointOOBOperations\_MultiplePowerCycleOffHard**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
      "EndpointId": "string"
    }
  ]
}
```

```
]
}
```

- o XML

```
<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
  <InternalServerError>
    <EndpointId>string</EndpointId>
  </InternalServerError>
</EndpointOperationsList>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned EndpointOperationsList
  - o 400 : Invalid ModelState
  - o 403 : User not allowed to execute operation
  - o 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/MasterBusReset

The **POST /api/v2/endpointOOBOperations/Multiple/MasterBusReset** call submits a batch request to perform an out of band Master Bus Reset operation on multiple endpoints

**(EndpointOOBOperations\_MultipleMasterBusReset)**

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
  - o application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
      "EndpointId": "string"
    }
  ]
}
```

- XML

```
<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
```



```

<InternalServerError>
  <EndpointId>string</EndpointId>
</InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/PowerOff/SoftGraceful

The **POST /api/v2/endpointOOBOperations/Multiple/PowerOff/SoftGraceful** call submits a batch request to perform an out of band Power Off - Soft Graceful operation on multiple endpoints

**(EndpointOOBOperations\_MultiplePowerOffSoftGraceful)**

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```

[
  {
    "EndpointId": "string"
  }
]

```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {

```

```

        "EndpointId": "string"
    }
],
"Not Found": [
    {
        "EndpointId": "string"
    }
],
"BadRequest": [
    {
        "EndpointId": "string"
    }
],
"InternalServerError": [
    {
        "EndpointId": "string"
    }
]
]
}

```

- XML

```

<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
  <InternalServerError>
    <EndpointId>string</EndpointId>
  </InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/PowerOff/HardGraceful

The **POST /api/v2/endpointOOBOperations/Multiple/PowerOff/HardGraceful** call submits a batch request to perform an out of band Power Off - Hard Graceful operation on multiple endpoints (**EndpointOOBOperations\_MultiplePowerOffHardGraceful**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
      "EndpointId": "string"
    }
  ]
}
```

```
}
```

- o XML

```
<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
  <InternalServerError>
    <EndpointId>string</EndpointId>
  </InternalServerError>
</EndpointOperationsList>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **EndpointOperationsList**
  - o 400 : Invalid ModelState
  - o 403 : User not allowed to execute operation
  - o 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/MasterBusReset/Graceful

The **POST /api/v2/endpointOOBOperations/Multiple/MasterBusReset/Graceful** call submits a batch request to perform an out of band Master Bus Reset Graceful operation on multiple endpoints (**EndpointOOBOperations\_MultipleMasterBusResetGraceful**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
  - o application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
      "EndpointId": "string"
    }
  ]
}
```

- XML

```
<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
```

```

<InternalServerError>
  <EndpointId>string</EndpointId>
</InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/PowerCycle/OffSoftGraceful

The **POST /api/v2/endpointOOBOperations/Multiple/PowerCycle/OffSoftGraceful** call submits a batch request to perform an out of band Power Cycle (Off - Soft Graceful) operation on multiple endpoints (**EndpointOOBOperations\_MultiplePowerCycleOffSoftGraceful**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.  
With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```

[
  {
    "EndpointId": "string"
  }
]

```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {

```

```

        "EndpointId": "string"
    }
],
"Not Found": [
    {
        "EndpointId": "string"
    }
],
"BadRequest": [
    {
        "EndpointId": "string"
    }
],
"InternalServerError": [
    {
        "EndpointId": "string"
    }
]
}

```

- XML

```

{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "Not Found": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
      "EndpointId": "string"
    }
  ]
}

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json

- application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointOperationsList**
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Multiple/PowerCycle/OffHardGraceful

The **POST /api/v2/endpointOOBOperations/Multiple/PowerCycle/OffHardGraceful** call submits a batch request to perform an out of band Power Cycle (Off - Hard Graceful) operation on multiple endpoints (**EndpointOOBOperations\_MultiplePowerCycleOffHardGraceful**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- With an HTTP 200, the response list will contain those EndpointIds for which a request was put on the message bus in Accepted list.
- EndpointIds for which a request was not put on the message bus will appear in the Forbidden, NotFound, BadRequest or InternalServerError list, depending on the case.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **AlertRequest**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- Return type: **EndpointOperationsList**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "Accepted": [
    {
      "EndpointId": "string"
    }
  ],
  "Forbidden": [
    {
      "EndpointId": "string"
    }
  ],
  "NotFound": [
    {
      "EndpointId": "string"
    }
  ],
  "BadRequest": [
```



```

    {
      "EndpointId": "string"
    }
  ],
  "InternalServerError": [
    {
      "EndpointId": "string"
    }
  ]
}

```

- o XML

```

<?xml version="1.0"?>
<EndpointOperationsList>
  <Accepted>
    <EndpointId>string</EndpointId>
  </Accepted>
  <Forbidden>
    <EndpointId>string</EndpointId>
  </Forbidden>
  <NotFound>
    <EndpointId>string</EndpointId>
  </NotFound>
  <BadRequest>
    <EndpointId>string</EndpointId>
  </BadRequest>
  <InternalServerError>
    <EndpointId>string</EndpointId>
  </InternalServerError>
</EndpointOperationsList>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **EndpointOperationsList**
  - o 400 : Invalid ModelState
  - o 403 : User not allowed to execute operation
  - o 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/PowerOn

The **POST /api/v2/endpointOOBOperations/Single/PowerOn** call submits a request to perform an out of band PowerOn operation (**EndpointOOBOperations\_SinglePowerOn**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO** EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - o application/json
  - o text/json

- application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/Sleep/Light

The **POST /api/v2/endpointOOBOperations/Single/Sleep/Light** call submits a request to perform an out of band Sleep Light operation (**EndpointOOBOperations\_SingleSleepLight**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation

- 409 : Endpoint not ready to execute operation yet
- 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/Sleep/Deep

The **POST /api/v2/endpointOOBOperations/Single/Sleep/Deep** call submits a request to perform an out of band Sleep Deep operation (**EndpointOOBOperations\_SingleSleepDeep**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/PowerCycle/OffSoft

The **POST /api/v2/endpointOOBOperations/Single/PowerCycle/OffSoft** call submits a request to perform an out of band Power Cycle (Off Soft) operation (**EndpointOOBOperations\_SinglePowerCycleOffSoft**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
```

```

    "EndpointId": "string"
  }
]

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/PowerOff/Hard

The **POST /api/v2/endpointOOBOperations/Single/PowerOff/Hard** call submits a request to perform an out of band Power Off - Hard operation (**EndpointOOBOperations\_SinglePowerOffHard**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```

[
  {
    "EndpointId": "string"
  }
]

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/Hibernate

The **POST /api/v2/endpointOOBOperations/Single/Hibernate** call submits a request to perform an out of band Hibernate operation (**EndpointOOBOperations\_SingleHibernate**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/PowerOff/Soft

The **POST /api/v2/endpointOOBOperations/Single/PowerOff/Soft** call submits a request to perform an out of band Power Off - Soft operation (**EndpointOOBOperations\_SinglePowerOffSoft**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json

- application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/PowerCycle/OffHard

The **POST /api/v2/endpointOOBOperations/Single/PowerCycle/OffHard** call submits a request to perform an out of band Power Cycle (Off Hard) operation (**EndpointOOBOperations\_SinglePowerCycleOffHard**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/MasterBusReset

The **POST /api/v2/endpointOOBOperations/Single/MasterBusReset** call submits a request to perform an out of band Master Bus Reset operation (**EndpointOOBOperations\_SingleMasterBusReset**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml

- application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/PowerOff/SoftGraceful

The **POST /api/v2/endpointOOBOperations/Single/PowerOff/SoftGraceful** call submits a request to perform an out of band Power Off - Soft Graceful operation (**EndpointOOBOperations\_SinglePowerOffSoftGraceful**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/PowerOff/HardGraceful

The **POST /api/v2/endpointOOBOperations/Single/PowerOff/HardGraceful** call submits a request to perform an out of band Power Off - Hard Graceful operation

(**EndpointOOBOperations\_SinglePowerOffHardGraceful**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/MasterBusReset/Graceful

The **POST /api/v2/endpointOOBOperations/Single/MasterBusReset/Graceful** call submits a request to perform an out of band Master Bus Reset Graceful operations

(**EndpointOOBOperations\_SingleMasterBusResetGraceful**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```



```
}
]
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/PowerCycle/OffSoftGraceful

The **POST /api/v2/endpointOOBOperations/Single/PowerCycle/OffSoftGraceful** call submits a request to perform an out of band Power Cycle (Off - Soft Graceful) operations (**EndpointOOBOperations\_SinglePowerCycleOffSoftGraceful**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## POST /api/v2/endpointOOBOperations/Single/PowerCycle/OffHardGraceful

The **POST /api/v2/endpointOOBOperations/Single/PowerCycle/OffHardGraceful** call submits a request to perform an out of band Power Cycle (Off - Hard Graceful) operation (**EndpointOOBOperations\_SinglePowerCycleOffHardGraceful**)

- Roles required: Tenant Administrator, user in a UserGroup with Execute privileges associated to target EndpointGroup.
- The path parameter is **EndpointIdDTO**. EndpointId object
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **EndpointIdDTO**. This is the Endpoint OOB Operations object.

```
[
  {
    "EndpointId": "string"
  }
]
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned
  - 400 : Invalid ModelState
  - 403 : User not allowed to execute operation
  - 409 : Endpoint not ready to execute operation yet
  - 500 : Internal Server Error

## 3.14 Endpoints

### GET /api/v2/endpoints/{endpointId}

The **GET /api/v2/endpoints/{endpointId}** call gets the details of an Endpoint by ID (**Endpoints\_GetById**)

- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with Read right to the resource.
- The path parameter is **endpointId**. Hex string with 64 characters and without the 0x prefix
- Return type: **Endpoint**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "EndpointId": "string",
  "EndpointGroupId": "string",
  "EndpointGroupName": "string",
  "LastUpdate": "2019-08-01T20:05:01.006Z",
  "ComputerName": "string",
  "PlatformType": 0,
  "AgentVersion": "string",
  "AgentType": "string",
}
```

```

"PowerState": 0,
"PowerStateUpdate": "2019-08-01T20:05:01.006Z",
"IsConnected": true,
"IsCiraConnected": true,
"NodeIdentity": 0,
"OperatingSystem": "string",
"NeighborsCount": 0,
"AgentLocalAdminMode": "unknown",
"NetworkInterfaces": [
  {
    "IPv4Address": "string",
    "IPv6Address": "string",
    "Subnet": "string",
    "Gateway": "string",
    "DnsSuffix": "string",
    "Mac": "string",
    "GatewayMac": "string",
    "Description": "string"
  }
],
"MEInfo": {
  "VersionString": "string",
  "IsAmtEnabled": true,

  "CiraEnabled": true,
  "AmtProvisioningState": "pre",
  "AmtProvisioningMode": "adminControlMode",
  "AmtControlMode": "notprovisioned",
  "MEVersion": "string",
  "MEFWBuildNumber": 0,
  "IsEHBC": true,
  "IsAmtVersionValid": true,
  "Version": 0
}
}

```

- o XML

```

<?xml version="1.0"?>
<Endpoint>
  <EndpointId>string</EndpointId>
  <EndpointGroupId>string</EndpointGroupId>
  <EndpointGroupName>string</EndpointGroupName>
  <LastUpdate>1970-01-01T00:00:00.001Z</LastUpdate>
  <ComputerName>string</ComputerName>
  <PlatformType>1</PlatformType>
  <AgentVersion>string</AgentVersion>
  <AgentType>string</AgentType>
  <PowerState>1</PowerState>
  <PowerStateUpdate>1970-01-01T00:00:00.001Z</PowerStateUpdate>
  <IsConnected>true</IsConnected>
  <IsCiraConnected>true</IsCiraConnected>
  <NodeIdentity>1</NodeIdentity>
  <OperatingSystem>string</OperatingSystem>

```

```

<NeighborsCount>1</NeighborsCount>
<AgentLocalAdminMode>unknown</AgentLocalAdminMode>
<NetworkInterfaces>
  <IPv4Address>string</IPv4Address>
  <IPv6Address>string</IPv6Address>
  <Subnet>string</Subnet>
  <Gateway>string</Gateway>
  <DnsSuffix>string</DnsSuffix>
  <Mac>string</Mac>
  <GatewayMac>string</GatewayMac>
  <Description>string</Description>
</NetworkInterfaces>
<MEInfo>
  <VersionString>string</VersionString>
  <IsAmtEnabled>true</IsAmtEnabled>

  <CiraEnabled>true</CiraEnabled>
  <AmtProvisioningState>pre</AmtProvisioningState>
  <AmtProvisioningMode>adminControlMode</AmtProvisioningMode>
  <AmtControlMode>notprovisioned</AmtControlMode>
  <MEVersion>string</MEVersion>
  <MEFWBuildNumber>1</MEFWBuildNumber>
  <IsEHBC>true</IsEHBC>
  <IsAmtVersionValid>true</IsAmtVersionValid>
  <Version>1</Version>
</MEInfo>
</Endpoint>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **Endpoint**
  - 400 : Bad Request
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal server error

## GET /api/v2/endpoints/{endpointId}/HardwareInfoFromAmt

The **GET /api/v2/endpoints/{endpointId}** call gets the Hardware information of an Endpoint using Id and AMT connection (**Endpoints\_GetById**)

- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with Read right to the resource.
- The Return code Not Found means endpoint doesn't exist or hardware information for it is not found as the endpoint doesn't support AMT requests.
- Return type: **Endpoint**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
```

```

"AmtPlatformInfo": {
  "ComputerModel": "string",
  "ManufacturerName": "string",
  "SerialNumber": "string",
  "VersionNumber": "string",
  "SystemId": "00000000-0000-0000-0000-000000000000"
},
"AmtBaseBoardInfo": {
  "ManufacturerName": "string",
  "ProductName": "string",
  "VersionNumber": "string",
  "SerialNumber": "string",
  "AssetTag": "string",
  "IsReplaceable": true
},
"AmtBiosInfo": {
  "ManufacturerName": "string",
  "VersionNumber": "string",
  "ReleaseDate": "string"
},
"AmtMemoryModuleInfo": [
  {
    "BankLabel": "string",
    "ManufacturerName": "string",
    "SerialNumber": "string",
    "Size": 0,
    "FormFactor": "string",
    "MemoryType": "string",
    "AssetTag": "string",
    "PartNumber": "string"
  }
]
}

```

#### o XML

```

<?xml version="1.0"?>
<EndpointAmtHardwareInfo>
  <AmtPlatformInfo>
    <ComputerModel>string</ComputerModel>
    <ManufacturerName>string</ManufacturerName>
    <SerialNumber>string</SerialNumber>
    <VersionNumber>string</VersionNumber>
    <SystemId>00000000-0000-0000-0000-000000000000</SystemId>
  </AmtPlatformInfo>
  <AmtBaseBoardInfo>
    <ManufacturerName>string</ManufacturerName>
    <ProductName>string</ProductName>
    <VersionNumber>string</VersionNumber>
    <SerialNumber>string</SerialNumber>
    <AssetTag>string</AssetTag>
    <IsReplaceable>true</IsReplaceable>
  </AmtBaseBoardInfo>
  <AmtBiosInfo>

```

```

    <ManufacturerName>string</ManufacturerName>
    <VersionNumber>string</VersionNumber>
    <ReleaseDate>string</ReleaseDate>
  </AmtBiosInfo>
  <AmtMemoryModuleInfo>
    <BankLabel>string</BankLabel>
    <ManufacturerName>string</ManufacturerName>
    <SerialNumber>string</SerialNumber>
    <Size>1</Size>
    <FormFactor>string</FormFactor>
    <MemoryType>string</MemoryType>
    <AssetTag>string</AssetTag>
    <PartNumber>string</PartNumber>
  </AmtMemoryModuleInfo>
</EndpointAmtHardwareInfo>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **Endpoint**
  - 400 : Bad Request
  - 403 : Forbidden
  - 404 : Not found
  - 409: Conflict
  - 500 : Internal server error

## DELETE /api/v2/endpoints/{endpointId}

The **DELETE /api/v2/endpoints/{endpointId}** call deletes an Endpoint by ID (**Endpoints\_Delete**)

- Role required: Tenant Administrator, or Endpoint Group Creator or Endpoint Group User with Delete right to the resource.
- The path parameter is **endpointId**. Hex string with 64 characters and without the 0x prefix
- Response types:
  - 204 : Successful request, no content
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal server error

## GET /api/v2/endpoints

The **GET /api/v2/endpoints** call get a collection of Endpoint summaries, filterable by EndpointGroupId. (**Endpoints\_GetSummariesCollection**)

- Role required: Tenant Administrator, Endpoint Group Creator, or Endpoint Group User.
- If endpointGroupId is not specified or empty, get a collection of Endpoint summaries for all endpoint groups that this user has access to.
- The path parameter is **endpointGroupId**. Endpoint Group ID
- Return type: **EndpointSummary**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "EndpointId": "string",
    "EndpointGroupId": "string",
    "EndpointGroupName": "string",
    "LastUpdate": "2019-08-01T20:05:01.017Z",
    "MEVersion": "string",
    "ComputerName": "string",
    "PlatformType": 0,
    "AgentVersion": 0,
    "AgentIdentifier": 0,
    "MEFWBuildNumber": 0,
    "PowerState": 0,
    "PowerStateUpdate": "2019-08-01T20:05:01.017Z",
    "IsConnected": true,
    "NodeIdentity": 0,
    "IsAmtVersionValid": true,
    "AmtControlMode": "notprovisioned",
    "AmtProvisioningState": "pre",
    "AmtProvisioningMode": "adminControlMode",
    "IsCiraConnected": true
  }
]
```

- XML

```
<?xml version="1.0"?>
<Inline Model>
  <EndpointId>string</EndpointId>
  <EndpointGroupId>string</EndpointGroupId>
  <EndpointGroupName>string</EndpointGroupName>
  <LastUpdate>1970-01-01T00:00:00.001Z</LastUpdate>
  <MEVersion>string</MEVersion>
  <ComputerName>string</ComputerName>
  <PlatformType>1</PlatformType>
  <AgentVersion>1</AgentVersion>
  <AgentIdentifier>1</AgentIdentifier>
  <MEFWBuildNumber>1</MEFWBuildNumber>
  <PowerState>1</PowerState>
  <PowerStateUpdate>1970-01-01T00:00:00.001Z</PowerStateUpdate>
  <IsConnected>true</IsConnected>
  <NodeIdentity>1</NodeIdentity>
  <IsAmtVersionValid>true</IsAmtVersionValid>
  <AmtControlMode>notprovisioned</AmtControlMode>
  <AmtProvisioningState>pre</AmtProvisioningState>
  <AmtProvisioningMode>adminControlMode</AmtProvisioningMode>
  <IsCiraConnected>true</IsCiraConnected>
</Inline Model>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml

- text/xml
- Response types:
  - 200 : Successful request, content returned **EndpointSummary**
  - 404 : Not found
  - 500 : Internal server error

## 3.15 Network Interfaces

### GET /api/v2/networkInterfaces

The **GET /api/v2/networkInterfaces** call gets a collection of network interfaces of the Intel® EMA server.  
(**NetworkInterfaces\_GetNetworkInterfacesInfo**)

- Role required: Global Administrator or Tenant Administrator
- Return type: **NetworkInterface**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "IPv4Address": "string",
    "IPv6Address": "string",
    "Subnet": "string",
    "Gateway": "string",
    "DnsSuffix": "string",
    "Mac": "string",
    "GatewayMac": "string",
    "Description": "string"
  }
]
```

- XML

```
<?xml version="1.0"?>
<Inline Model>
  <IPv4Address>string</IPv4Address>
  <IPv6Address>string</IPv6Address>
  <Subnet>string</Subnet>
  <Gateway>string</Gateway>
  <DnsSuffix>string</DnsSuffix>
  <Mac>string</Mac>
  <GatewayMac>string</GatewayMac>
  <Description>string</Description>
</Inline Model>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **NetworkInterface**
  - 500 : Internal server error



## 3.16 Roles

### GET /api/v2/roles

The **GET /api/v2/roles** call gets a collection of Roles. (**Roles\_GetAll**)

- Role required: Global Administrator, Tenant Administrator, or Account Manager.
- Return type: **Role**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "RoleId": 0,
    "Name": "string",
    "Description": "string",
    "LastUpdated": "2019-08-01T20:05:01.030Z",
    "Enabled": true,
    "TenantId": "00000000-0000-0000-0000-000000000000"
  }
]
```

- XML

```
<?xml version="1.0"?>
<Inline Model>
  <RoleId>1</RoleId>
  <Name>string</Name>
  <Description>string</Description>
  <LastUpdated>1970-01-01T00:00:00.001Z</LastUpdated>
  <Enabled>true</Enabled>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
</Inline Model>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **Role**
  - 500 : Internal Server Error

### GET /api/v2/roles/{roleId}

The **GET /api/v2/roles/{roleId}** call gets the details of a Role by ID. (**Roles\_GetRole**)

- Role required: Global Administrator, Tenant Administrator, or Account Manager.
- The path parameter is roleId. Role ID
- Return type: **Role**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
```

```

    "RoleId": 0,
    "Name": "string",
    "Description": "string",
    "LastUpdated": "2019-08-01T20:05:01.035Z",
    "Enabled": true,
    "TenantId": "00000000-0000-0000-0000-000000000000"
  }

```

- o XML

```

<?xml version="1.0"?>
<Role>
  <RoleId>1</RoleId>
  <Name>string</Name>
  <Description>string</Description>
  <LastUpdated>1970-01-01T00:00:00.001Z</LastUpdated>
  <Enabled>true</Enabled>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
</Role>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **Role**
  - o 403 : Forbidden
  - o 404 : Not found
  - o 500 : Internal Server Error

## 3.17 Server Settings

### GET /api/v2/serverSettings/amtDiscoverySettings

The **GET /api/v2/serverSettings/amtDiscoverySettings** call gets a the automatic Intel® AMT discovery settings of the Manageability server. (**ServerSettings\_GetAmtDiscoverySettings**)

- Role required: Global Administrator or Tenant Administrator.
- TenantId corresponds to the Tenant Administrator who submitted the automatic Intel® AMT discovery request.
- Return type: **AutomaticAMTDiscoverySettings**
- Example data: Content-Type = Application/JSON and application/XML.
  - o JSON.

```

{
  "Interval": 0,
  "Type": "singleIP",
  "TypeStr": "string",
  "MAC": "string",
  "FirstIPAddress": "string",
  "EndRangeIPAddress": "string",
  "SubnetMask": "string",
  "TenantId": "string"
}

```

```
}
```

- o XML

```
<?xml version="1.0"?>
<AutomaticAMTDiscoverySettings>
  <Interval>1</Interval>
  <Type>singleIP</Type>
  <TypeStr>string</TypeStr>
  <MAC>string</MAC>
  <FirstIPAddress>string</FirstIPAddress>
  <EndRangeIPAddress>string</EndRangeIPAddress>
  <SubnetMask>string</SubnetMask>
  <TenantId>string</TenantId>
</AutomaticAMTDiscoverySettings>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **AutomaticAMTDiscoverySettings**
  - o 500 : Internal server error

## 3.18 Tenants

### GET /api/v2/tenants

The **GET /api/v2/tenants** call gets a collection of Tenants . (**Tenants\_GetAll**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User.
- Return type: **Tenant**
- Example data: Content-Type = Application/JSON and application/XML.
  - o JSON.

```
[
  {
    "CreatedBy": "00000000-0000-0000-0000-000000000000",
    "CreatedOn": "2019-08-01T20:05:01.090Z",
    "TenantId": "00000000-0000-0000-0000-000000000000",
    "ModifiedBy": "00000000-0000-0000-0000-000000000000",
    "ModifiedOn": "2019-08-01T20:05:01.091Z",
    "Description": "string",
    "Name": "string"
  }
]
```

- o XML

```
<?xml version="1.0"?>
<Inline Model>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
```

```

    <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
    <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
    <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
    <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
    <Description>string</Description>
    <Name>string</Name>
  </Inline Model>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **Tenant**
  - 404 : Not found
  - 500 : Internal Server Error

## POST /api/v2/tenants

The **POST /api/v2/tenants** call creates a Tenant. (**Tenants\_Post**)

- Role required: Global Administrator
- The path parameter is **Tenant**.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **Tenant**. This is the Tenants object.

```

{
  "CreatedBy": "00000000-0000-0000-0000-000000000000",
  "CreatedOn": "2019-08-01T20:05:01.099Z",
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "ModifiedBy": "00000000-0000-0000-0000-000000000000",
  "ModifiedOn": "2019-08-01T20:05:01.099Z",
  "Description": "string",
  "Name": "string"
}

```

- Return type: **Tenant**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "CreatedBy": "00000000-0000-0000-0000-000000000000",
  "CreatedOn": "2019-08-01T20:05:01.095Z",
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "ModifiedBy": "00000000-0000-0000-0000-000000000000",
  "ModifiedOn": "2019-08-01T20:05:01.095Z",
  "Description": "string",
  "Name": "string"
}

```

```
}
```

- o XML

```
<?xml version="1.0"?>
<Tenant>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <Description>string</Description>
  <Name>string</Name>
</Tenant>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **Tenant**
  - o 400 : Bad request
  - o 409 : Conflict
  - o 500 : Internal Server Error

## GET /api/v2/tenants/{tenantId}

The **GET /api/v2/tenants/{tenantId}** call gets the details of a Tenant by ID. (**Tenants\_GetById**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User.
- The path parameter is **tenantId**. Canonical GUID form: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
- Return type: **Tenant**
- Example data: Content-Type = Application/JSON and application/XML.
  - o JSON.

```
{
  "CreatedBy": "00000000-0000-0000-0000-000000000000",
  "CreatedOn": "2019-08-01T20:05:01.109Z",
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "ModifiedBy": "00000000-0000-0000-0000-000000000000",
  "ModifiedOn": "2019-08-01T20:05:01.109Z",
  "Description": "string",
  "Name": "string"
}
```

- o XML

```
<?xml version="1.0"?>
<Tenant>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
```

```

    <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
    <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
    <Description>string</Description>
    <Name>string</Name>
  </Tenant>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **Tenant**
  - 400 : Bad Request Error
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal Server Error

## PUT /api/v2/tenants/{tenantId}

The **PUT /api/v2/tenants/{tenantId}** call updates a Tenant. (**Tenants\_Put**)

- Role required: Global Administrator
- Only the fields Name and Description can be updated.
- The path parameter is **tenantId**. Canonical GUID form: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **tenantId**. This is the Tenants object.

```

{
  "CreatedBy": "00000000-0000-0000-0000-000000000000",
  "CreatedOn": "2019-08-01T20:05:01.125Z",
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "ModifiedBy": "00000000-0000-0000-0000-000000000000",
  "ModifiedOn": "2019-08-01T20:05:01.125Z",
  "Description": "string",
  "Name": "string"
}

```

- Return type: **Tenant**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "CreatedBy": "00000000-0000-0000-0000-000000000000",
  "CreatedOn": "2019-08-01T20:05:01.120Z",
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "ModifiedBy": "00000000-0000-0000-0000-000000000000",
  "ModifiedOn": "2019-08-01T20:05:01.120Z",
  "Description": "string",
}

```

```
"Name": "string"
}
```

- o XML

```
<?xml version="1.0"?>
<Tenant>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <Description>string</Description>
  <Name>string</Name>
</Tenant>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **Tenant**
  - o 400 : Bad request
  - o 404 : Not found
  - o 500 : Internal server error

## DELETE /api/v2/tenants/{tenantId}

The **DELETE /api/v2/tenants/{tenantId}** call deletes a Tenant By ID (**Tenants\_Delete**)

- Role required: Global Administrator
- The path parameter is **tenantId**. Canonical GUID form: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
- Response types:
  - o 204 : Successful request, no content
  - o 400 : Bad request
  - o 404 : Not found
  - o 500 : Internal server error

## 3.19 User Group Memberships

### GET /api/v2/userGroupMemberships/{userGroupId}

The **GET /api/v2/userGroupMemberships/{userGroupId}** call gets a collection of names of User Group members. (**UserGroupMemberships\_GetNames**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, or Endpoint Group Creator.
- The path parameter is **userGroupId**. User Group identifier
- Return type: **UserGroupMember**
- Example data: Content-Type = Application/JSON and application/XML.
  - o JSON.

```
[
{
```

```

    "UserName": "string"
  }
]

```

- o XML

```

<?xml version="1.0"?>
<Inline Model>
  <UserName>string</UserName>
</Inline Model>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **UserGroupMember**
  - o 404 : Not found
  - o 500 : Internal Server Error

## POST /api/v2/userGroupMemberships/{userGroupId}/addMembers

The **POST /api/v2/userGroupMemberships/{userGroupId}/addMembers** call add members to a User Group. (**UserGroupMemberships\_Post**)

- Role required: Global Administrator, Tenant Administrator, or Account Manager.
- With an HTTP 200, the response list will contain only those members that got added to the User Group. Members not added for one reason or another will not appear in the response list.
- This API call consumes the following media types via the Content-Type request header.
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
  - o application/x-www-form-urlencoded
- The request body is **tenantId**. This is the User Group Memberships object.

```

[
  {
    "UserName": "string"
  }
]

```

- Return type: **UserGroupMember**
- Example data: Content-Type = Application/JSON and application/XML.
  - o JSON.

```

[
  {
    "UserName": "string"
  }
]

```

- o XML



```
<?xml version="1.0"?>
<Inline Model>
  <UserName>string</UserName>
</Inline Model>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **UserGroupMember**
  - 400 : Bad request
  - 404 : Not found
  - 500 : Internal Server Error

## POST /api/v2/userGroupMemberships/{userGroupId}/removeMembers

The **POST /api/v2/userGroupMemberships/{userGroupId}/removeMembers** call removes members from a User Group. (**UserGroupMemberships\_RemoveMembers**)

- Role required: Global Administrator, Tenant Administrator, or Account Manager.
- With an HTTP 200, the response list will contain only those members removed from the User Group. Members not removed for one reason or another will not appear in the response list.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **tenantId**. This is the User Group Memberships object.

```
[
  {
    "UserName": "string"
  }
]
```

- Return type: **UserGroupMember**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "UserName": "string"
  }
]
```

- XML

```
<?xml version="1.0"?>
<Inline Model>
  <UserName>string</UserName>
```

```
</Inline Model>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **UserGroupMember**
  - 400 : Bad request
  - 404 : Not found
  - 500 : Internal server error

## 3.20 User Groups

### GET /api/v2/userGroups

The **GET /api/v2/userGroups** call gets a collection of User Groups. (**UserGroups\_GetAll**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, or Endpoint Group Creator.
- Return type: **UserGroup**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "UserGroupId": 0,
    "Name": "string",
    "TenantId": "00000000-0000-0000-0000-000000000000",
    "Description": "string",
    "CreatedOn": "2019-08-01T20:05:01.149Z",
    "CreatedBy": "00000000-0000-0000-0000-000000000000",
    "ModifiedOn": "2019-08-01T20:05:01.149Z",
    "ModifiedBy": "00000000-0000-0000-0000-000000000000",
    "RoleId": 0,
    "AccessRightsId": "execute",
    "AccessRights": "string"
  }
]
```

- XML

```
[
  {
    "UserGroupId": 0,
    "Name": "string",
    "TenantId": "00000000-0000-0000-0000-000000000000",
    "Description": "string",
    "CreatedOn": "2019-08-01T20:05:01.149Z",
    "CreatedBy": "00000000-0000-0000-0000-000000000000",
    "ModifiedOn": "2019-08-01T20:05:01.149Z",
```

```

    "ModifiedBy": "00000000-0000-0000-0000-000000000000",
    "RoleId": 0,
    "AccessRightsId": "execute",
    "AccessRights": "string"
  }
]

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **UserGroup**
  - 500 : Internal server error

## POST /api/v2/userGroups

The **POST /api/v2/userGroups** call creates a new User Group having Execute or View right.  
(**UserGroups\_InsertUserGroup**)

- Role required: Global Administrator, Tenant Administrator, or Account Manager.
- For AccessRightsId, enter one of the two following values:
  - Execute
  - View
- The roleId value will be automatically populated by the system
- The path parameter is **UserGroup**. User Group object to create
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **UserGroup**. This is the User Groups object.

```

{
  "UserGroupId": 0,
  "Name": "string",
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "Description": "string",
  "CreatedOn": "2019-08-01T20:05:01.156Z",
  "CreatedBy": "00000000-0000-0000-0000-000000000000",
  "ModifiedOn": "2019-08-01T20:05:01.156Z",
  "ModifiedBy": "00000000-0000-0000-0000-000000000000",
  "RoleId": 0,
  "AccessRightsId": "execute",
  "AccessRights": "string"
}

```

- Return type: **UserGroup**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{

```

```

    "UserGroupId": 0,
    "Name": "string",
    "TenantId": "00000000-0000-0000-0000-000000000000",
    "Description": "string",
    "CreatedOn": "2019-08-01T20:05:01.153Z",
    "CreatedBy": "00000000-0000-0000-0000-000000000000",
    "ModifiedOn": "2019-08-01T20:05:01.153Z",
    "ModifiedBy": "00000000-0000-0000-0000-000000000000",
    "RoleId": 0,
    "AccessRightsId": "execute",
    "AccessRights": "string"
  }

```

- XML

```

<?xml version="1.0"?>
<UserGroup>
  <UserGroupId>1</UserGroupId>
  <Name>string</Name>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <Description>string</Description>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
  <RoleId>1</RoleId>
  <AccessRightsId>execute</AccessRightsId>
  <AccessRights>string</AccessRights>
</UserGroup>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **UserGroup**
  - 400 : Bad request
  - 409 : Conflict, User Group name already exists
  - 500 : Internal Server Error

## GET /api/v2/userGroups/{userGroupId}

The **GET /api/v2/userGroups/{userGroupId}** call gets a User Group by ID. (**UserGroups\_GetUserGroup**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, or Endpoint Group Creator.
- The path parameter is **userGroupId**. User Group identifier
- Return type: **UserGroup**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "UserGroupId": 0,

```

```

    "Name": "string",
    "TenantId": "00000000-0000-0000-0000-000000000000",
    "Description": "string",
    "CreatedOn": "2019-08-01T20:05:01.164Z",
    "CreatedBy": "00000000-0000-0000-0000-000000000000",
    "ModifiedOn": "2019-08-01T20:05:01.165Z",
    "ModifiedBy": "00000000-0000-0000-0000-000000000000",
    "RoleId": 0,
    "AccessRightsId": "execute",
    "AccessRights": "string"
  }

```

o XML

```

<?xml version="1.0"?>
<UserGroup>
  <UserGroupId>1</UserGroupId>
  <Name>string</Name>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <Description>string</Description>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
  <RoleId>1</RoleId>
  <AccessRightsId>execute</AccessRightsId>
  <AccessRights>string</AccessRights>
</UserGroup>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **UserGroup**
  - o 400 : Bad request
  - o 403 : Forbidden
  - o 404 : Not found
  - o 500 : Internal server error

## PUT /api/v2/userGroups/{userGroupId}

The **PUT /api/v2/userGroups/{userGroupId}** call updates a User Group by ID.  
(**UserGroups\_UpdateUserGroup**)

- Role required: Global Administrator, Tenant Administrator, or Account Manager.
- The only fields that can be updated are Name, Description and AccessRights.
- For AccessRightsId, enter one of the two following values:
  - o Execute
  - o View
- The roleId value will be automatically populated by the system
- The path parameter is **userGroupId**. The User Group identifier
- This API call consumes the following media types via the Content-Type request header.

- application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **userGroupId**. This is the User Groups object.

```
{
  "UserGroupId": 0,
  "Name": "string",
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "Description": "string",
  "CreatedOn": "2019-08-01T20:05:01.174Z",
  "CreatedBy": "00000000-0000-0000-0000-000000000000",
  "ModifiedOn": "2019-08-01T20:05:01.174Z",
  "ModifiedBy": "00000000-0000-0000-0000-000000000000",
  "RoleId": 0,
  "AccessRightsId": "execute",
  "AccessRights": "string"
}
```

- Return type: **UserGroup**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "UserGroupId": 0,
  "Name": "string",
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "Description": "string",
  "CreatedOn": "2019-08-01T20:05:01.170Z",
  "CreatedBy": "00000000-0000-0000-0000-000000000000",
  "ModifiedOn": "2019-08-01T20:05:01.170Z",
  "ModifiedBy": "00000000-0000-0000-0000-000000000000",
  "RoleId": 0,
  "AccessRightsId": "execute",
  "AccessRights": "string"
}
```

- XML

```
<?xml version="1.0"?>
<UserGroup>
  <UserGroupId>1</UserGroupId>
  <Name>string</Name>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <Description>string</Description>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
  <RoleId>1</RoleId>
  <AccessRightsId>execute</AccessRightsId>
  <AccessRights>string</AccessRights>
</UserGroup>
```

```
</UserGroup>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : OK: Return content. **UserGroup**
  - 400 : Bad request
  - 403 : Forbidden
  - 404 : Not found
  - 409 : Conflict, User Group name already exists
  - 500 : Internal Server Error

## DELETE /api/v2/userGroups/{userGroupId}

The **DELETE /api/v2/userGroups/{userGroupId}** call deletes a User Group by ID.  
(UserGroups\_DeleteUserGroup)

- Role required: Global Administrator, Tenant Administrator, or Account Manager.
- The path parameter is **userGroupId**. The User Group identifier.
- Response types:
  - 204 : Successful request, no content.
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal Server Error

## 3.21 User Group To Endpoint Group Assignments

### GET /api/v2/userGroupToEndpointGroupAssignments/{endpointGroupId}

The **GET /api/v2/userGroupToEndpointGroupAssignments/{endpointGroupId}** call gets a collection of User Groups assigned to an Endpoint Group (UserGroupToEndpointGroupAssignments\_GetUserGroups)

- Role required: Tenant Administrator, or Endpoint Group Creator.
- The path parameter is **endpointGroupId**. Endpoint Group ID. Hex string with 64 characters and without the 0x prefix.
- Return type: **UserGroupIdentifier**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "UserGroupId": 0,
    "Name": "string",
    "Description": "string",
    "AccessRightsId": "execute",
    "AccessRights": "string"
  }
]
```

- XML

```
<?xml version="1.0"?>
```

```
<Inline Model>
  <UserGroupId>1</UserGroupId>
  <Name>string</Name>
  <Description>string</Description>
  <AccessRightsId>execute</AccessRightsId>
  <AccessRights>string</AccessRights>
</Inline Model>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : OK: Return content. **UserGroupIdentifier**
  - 400 : Bad request
  - 403 : Forbidden: If the caller does not have the required rights.
  - 500 : Internal Server Error: Exceptions thrown.

## POST /api/v2/userGroupToEndpointGroupAssignments/{endpointGroupId}

The **POST /api/v2/userGroupToEndpointGroupAssignments/{endpointGroupId}** call assigns a User Group to an Endpoint Group (**UserGroupToEndpointGroupAssignments\_AssignUserGroup**)

- Role required: Tenant Administrator, or Endpoint Group Creator of a suitable User Group.
- The path parameter is **endpointGroupId**. EndPoint Group ID. Hex string with 64 characters and without the 0x prefix.
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **endpointGroupId**. This is the User Group To Endpoint Group Assignments object.

```
{
  "UserGroupId": 0
}
```

- Return type: **UserGroupIdentifierDTO**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "UserGroupId": 0
}
```

- XML

```
<?xml version="1.0"?>
<UserGroupIdentifierDTO>
  <UserGroupId>1</UserGroupId>
```



```
</UserGroupIdentifierDTO>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : OK: Return content. **UserGroupIdentifierDTO**
  - 400 : Bad request: If request body is not in right format.
  - 403 : Forbidden: If the caller does not have the required rights.
  - 404 : Not found: If {userGroupId} does not exist.
  - 409 : Conflict: If {userGroupId} and {endpointGroupId} are already associated.
  - 500 : Internal Server Error: Exceptions thrown.

## DELETE

### /api/v2/userGroupToEndpointGroupAssignments/{endpointGroupId}/userGroup/{userGroupId}

The **DELETE**

**/api/v2/userGroupToEndpointGroupAssignments/{endpointGroupId}/userGroup/{userGroupId}** call disassociates an User Group from a Endpoint Group  
(**UserGroupToEndpointGroupAssignments\_DisassociateUserGroup**)

- Role required: Tenant Administrator, or Endpoint Group Creator of a suitable User Group.
- The path parameter is **endpointGroupId**. Endpoint Group ID. Hex string with 64 characters and without the 0x prefix.
- Response types:
  - 204 : Successful request, no content.
  - 400 : Bad request: If URI is not formatted right or If {endpointGroupId} and {userGroupId} are already associated.
  - 403 : Forbidden: If the caller does not have the required rights.
  - 404 : Not found: If {endpointGroupId} or {userGroupId} does not exist.
  - 500 : Internal Server Error: Exceptions thrown.

## 3.22 Users

### GET /api/v2/users

The **GET /api/v2/users** call gets a collection of Users. (**Users\_GetSummariesCollection**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, or Endpoint Group Creator.
- Return type: **UserSummary**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
[
  {
    "UserId": "string",
    "Username": "string",
    "Enabled": true,
    "TenantId": "00000000-0000-0000-0000-000000000000",
    "Description": "string",
```

```

    "RoleId": 0,
    "SysRole": "none"
  }
]

```

- o XML

```

<?xml version="1.0"?>
<Inline Model>
  <UserId>string</UserId>
  <Username>string</Username>
  <Enabled>true</Enabled>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <Description>string</Description>
  <RoleId>1</RoleId>
  <SysRole>none</SysRole>
</Inline Model>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **UserSummary**
  - o 401 : Unauthorized
  - o 500 : Internal server error

## POST /api/v2/users

The **POST /api/v2/users** call creates a User (**Users\_CreateUser**)

- Role required: Global Administrator, Tenant Administrator, or Account Manager.
- TenantId is not required when a Global Administrator creates another Global Administrator.
- If Intel® EMA installed in Windows Domain Authentication mode, the Username used for registering a new user must be its User Principle Name (UPN).
- Username must be in an email address format.
- Creating a 'Locked' user is not allowed. The 'Enabled' property should always be passed as 'true'
- Password is required only when Intel® EMA is not installed in Windows Domain Authentication mode.
- Password Requirements
  - o at least 8 characters
  - o contains at least one number
  - o contains both lowercase and uppercase alpha characters
  - o contains at least one special character
- The path parameter is **NewUser**. User object to be created
- This API call consumes the following media types via the Content-Type request header.
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
  - o application/x-www-form-urlencoded
- The request body is **NewUser**. This is the Users object.

```

{

```

```

    "UserId": "00000000-0000-0000-0000-000000000000",
    "Username": "string",
    "Enabled": true,
    "TenantId": "00000000-0000-0000-0000-000000000000",
    "Description": "string",
    "Password": "string",
    "RoleId": 0
  }

```

- Return type: **User**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "UserId": "00000000-0000-0000-0000-000000000000",
  "Username": "string",
  "CreatedOn": "2019-08-01T20:05:01.205Z",
  "CreatedBy": "00000000-0000-0000-0000-000000000000",
  "ModifiedOn": "2019-08-01T20:05:01.205Z",
  "ModifiedBy": "00000000-0000-0000-0000-000000000000",
  "Enabled": true,
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "Description": "string",
  "RoleId": 0,
  "SysRole": "none"
}

```

- XML

```

<?xml version="1.0"?>
<User>
  <UserId>00000000-0000-0000-0000-000000000000</UserId>
  <Username>string</Username>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
  <Enabled>true</Enabled>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <Description>string</Description>
  <RoleId>1</RoleId>
  <SysRole>none</SysRole>
</User>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **User**
  - 400 : Bad request
  - 401 : Unauthorized

- 409 : Conflict
- 500 : Internal server error

## GET /api/v2/users/{userId}

The **GET /api/v2/users/{userId}** call gets the details of a User by ID. (**Users\_GetById**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User.
- Endpoint Group Users can see only their own information.
- The path parameter is **userId**. User ID. Canonical GUID form: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
- Return type: **User**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "UserId": "00000000-0000-0000-0000-000000000000",
  "Username": "string",
  "CreatedOn": "2019-08-01T20:05:01.217Z",
  "CreatedBy": "00000000-0000-0000-0000-000000000000",
  "ModifiedOn": "2019-08-01T20:05:01.217Z",
  "ModifiedBy": "00000000-0000-0000-0000-000000000000",
  "Enabled": true,
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "Description": "string",
  "RoleId": 0,
  "SysRole": "none"
}
```

- XML

```
<?xml version="1.0"?>
<User>
  <UserId>00000000-0000-0000-0000-000000000000</UserId>
  <Username>string</Username>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
  <Enabled>true</Enabled>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <Description>string</Description>
  <RoleId>1</RoleId>
  <SysRole>none</SysRole>
</User>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **User**
  - 401 : Unauthorized

- 403 : Forbidden
- 404 : Not found
- 500 : Internal server error

## PUT /api/v2/users/{userId}

The **PUT /api/v2/users/{userId}** call updates a User by ID. (**Users\_UpdateUser**)

- Role required: Global Administrator, Tenant Administrator, or Account manager.
- Only Description, RoleId, and Enabled can be updated, except for TenantId if converting a user from a Global Admin to a Tenant User. All other fields are ignored.
- Manually locking a user by changing the 'Enabled' property to 'false' is not currently supported.
- The path parameter is userId. User ID. Canonical GUID form: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **userId**. This is the Users object.
- Return type: **User**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "UserId": "00000000-0000-0000-0000-000000000000",
  "Username": "string",
  "CreatedOn": "2019-08-01T20:05:01.263Z",
  "CreatedBy": "00000000-0000-0000-0000-000000000000",
  "ModifiedOn": "2019-08-01T20:05:01.263Z",
  "ModifiedBy": "00000000-0000-0000-0000-000000000000",
  "Enabled": true,
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "Description": "string",
  "RoleId": 0,
  "SysRole": "none"
}
```

- XML

```
<?xml version="1.0"?>
<User>
  <UserId>00000000-0000-0000-0000-000000000000</UserId>
  <Username>string</Username>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
  <Enabled>true</Enabled>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <Description>string</Description>
  <RoleId>1</RoleId>
  <SysRole>none</SysRole>
</User>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **User**
  - 400 : Bad request
  - 401 : Unauthorized
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal server error

## DELETE /api/v2/users/{userId}

The **DELETE /api/v2/users/{userId}** call deletes a User by ID. (**Users\_DeleteUser**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User
- All users can delete their own accounts.
- Global Administrators can delete any user account from any Tenant.
- Tenant Administrators can delete any user account from the same Tenant.
- Account Managers can delete any user having the primary role of Account Manager, Endpoint Group Creator, or Endpoint Group User from the same Tenant.
- The path parameter is **userId**. Canonical GUID form: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
- Response types:
  - 204 : Successful request, no content
  - 401 : Unauthorized
  - 403 : Forbidden
  - 404 : Not found
  - 500 : Internal server error

## GET /api/v2/users/getUserByName

The **GET /api/v2/users/getUserByName** call gets the details of a User by username. (**Users\_GetByName**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User.
- Endpoint Group **Users** can see only their own information.
- The path parameter is **username**.
- Return type: **User**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "UserId": "00000000-0000-0000-0000-000000000000",
  "Username": "string",
  "CreatedOn": "2019-08-01T20:05:01.273Z",
  "CreatedBy": "00000000-0000-0000-0000-000000000000",
  "ModifiedOn": "2019-08-01T20:05:01.273Z",
  "ModifiedBy": "00000000-0000-0000-0000-000000000000",
  "Enabled": true,
  "TenantId": "00000000-0000-0000-0000-000000000000",
  "Description": "string",
  "RoleId": 0,
```

```
"SysRole": "none"
}
```

- o XML

```
<?xml version="1.0"?>
<User>
  <UserId>00000000-0000-0000-0000-000000000000</UserId>
  <Username>string</Username>
  <CreatedOn>1970-01-01T00:00:00.001Z</CreatedOn>
  <CreatedBy>00000000-0000-0000-0000-000000000000</CreatedBy>
  <ModifiedOn>1970-01-01T00:00:00.001Z</ModifiedOn>
  <ModifiedBy>00000000-0000-0000-0000-000000000000</ModifiedBy>
  <Enabled>true</Enabled>
  <TenantId>00000000-0000-0000-0000-000000000000</TenantId>
  <Description>string</Description>
  <RoleId>1</RoleId>
  <SysRole>none</SysRole>
</User>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request, content returned **User**
  - o 400 : Bad request
  - o 401 : Unauthorized
  - o 403 : Forbidden
  - o 404 : Not found
  - o 500 : Internal server error

## POST /api/v2/users/{userId}/resetPassword

The **POST /api/v2/users/{userId}/resetPassword** call set a User's password. (**Users\_UpdatePassword**)

- Role required: Global Administrator, Tenant Administrator, Account Manager, Endpoint Group Creator, or Endpoint Group User.
- Global Administrators can reset the passwords of all Users without providing their current passwords.
- Tenant Administrators can reset the passwords of Users in the same Tenant without providing their current passwords.
- All other users need to provide their current passwords to reset them.
- Password is required only when Intel® EMA is not installed in Windows Domain Authentication mode.
- Password Requirements
  - o at least 8 characters
  - o contains at least one number
  - o contains both lowercase and uppercase alpha characters
  - o contains at least one special character
- The path parameter is **userId**. User ID. Canonical GUID form: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
- This API call consumes the following media types via the Content-Type request header.
  - o application/json
  - o text/json
  - o application/xml

- o text/xml
  - o application/x-www-form-urlencoded
- The request body is **userId**. This is the Users object.

```
{
  "CurrentPassword": "string",
  "NewPassword": "string"
}
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
- Response types:
  - o 204 : Successful request, no content
  - o 400 : Bad request
  - o 401 : Unauthorized
  - o 403 : Forbidden
  - o 404 : Not found
  - o 500 : Internal server error

## 3.23 Wifi Setups

### GET /api/v2/wifiSetups

The **GET /api/v2/wifiSetups** call gets a collection of WifiSetups. (**WifiSetups\_GetCollection**)

- Role required: Tenant Administrator, Endpoint Group Creator, or Endpoint Group User
- Return type: **WifiSetup**
- Example data: Content-Type = Application/JSON and application/XML.
  - o JSON.

```
[
  {
    "WifiSetupId": 0,
    "SetupName": "string",
    "Priority": 0,
    "SSID": "string",
    "EncryptionAlgorithm": "tKIP",
    "KeyManagementProtocol": "wPAPSK",
    "Passphrase": "string",
    "TenantId": "string",
    "_802_1_SetupLookupKey": 0
  }
]
```

- o XML

```
<?xml version="1.0"?>
<Inline Model>
  <WifiSetupId>1</WifiSetupId>
  <SetupName>string</SetupName>
  <Priority>1</Priority>
  <SSID>string</SSID>
  <EncryptionAlgorithm>tKIP</EncryptionAlgorithm>
  <KeyManagementProtocol>wPAPSK</KeyManagementProtocol>
  <Passphrase>string</Passphrase>
```



```
<TenantId>string</TenantId>
<_802_1_SetupLookupKey>1</_802_1_SetupLookupKey>
</Inline Model>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request **WifiSetup**
  - 500 : Internal Server Error

## POST /api/v2/wifiSetups

The **POST /api/v2/wifiSetups** call creates a WifiSetup to be used by Intel® AMT Profiles. (**WifiSetups\_Post**)

- Role required: Tenant Administrator
- When the KeyManagementProtocol is WPAPSK or WPA2PSK, Passphrase is required, and \_802\_1\_SetupLookupKey is not needed and will be set to null.
- When the KeyManagementProtocol is WPAIEEE802\_1 or WPA2IEEE802\_1, \_802\_1\_SetupLookupKey is required, and Passphrase is not needed and will be set to null.
- The path parameter is **WifiSetupNew**. WifiSetup object to create
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **WifiSetupNew**. This is the Wifi Setups object.

```
{
  "SetupName": "string",
  "Priority": 0,
  "SSID": "string",
  "EncryptionAlgorithm": "tKIP",
  "KeyManagementProtocol": "wPAPSK",
  "Passphrase": "string",
  "TenantId": "string",
  "_802_1_SetupLookupKey": 0
}
```

- Return type: **WifiSetup**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```
{
  "WifiSetupId": 0,
  "SetupName": "string",
  "Priority": 0,
  "SSID": "string",
  "EncryptionAlgorithm": "tKIP",
  "KeyManagementProtocol": "wPAPSK",
  "Passphrase": "string",
}
```

```
"TenantId": "string",
"_802_1_SetupLookupKey": 0
```

- o XML

```
<?xml version="1.0"?>
<WifiSetup>
  <WifiSetupId>1</WifiSetupId>
  <SetupName>string</SetupName>
  <Priority>1</Priority>
  <SSID>string</SSID>
  <EncryptionAlgorithm>tKIP</EncryptionAlgorithm>
  <KeyManagementProtocol>WPAPSK</KeyManagementProtocol>
  <Passphrase>string</Passphrase>
  <TenantId>string</TenantId>
  <_802_1_SetupLookupKey>1</_802_1_SetupLookupKey>
</WifiSetup>
```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - o application/json
  - o text/json
  - o application/xml
  - o text/xml
- Response types:
  - o 200 : Successful request **WifiSetup**
  - o 400 : Invalid ModelState
  - o 409 : WifiSetup name already exists
  - o 500 : Internal Server Error

## GET /api/v2/wifiSetups/{wifiSetupId}

The GET /api/v2/wifiSetups/{wifiSetupId} call gets a WiFiSetup by ID. (WifiSetups\_GetById)

- Role required: Tenant Administrator, Endpoint Group Creator, or Endpoint Group User
- The path parameter is **wifiSetupId**. WifiSetup ID
- Return type: **WifiSetup**
- Example data: Content-Type = Application/JSON and application/XML.
  - o JSON.

```
{
  "WifiSetupId": 0,
  "SetupName": "string",
  "Priority": 0,
  "SSID": "string",
  "EncryptionAlgorithm": "tKIP",
  "KeyManagementProtocol": "WPAPSK",
  "Passphrase": "string",
  "TenantId": "string",
  "_802_1_SetupLookupKey": 0
}
```

- o XML

```
<?xml version="1.0"?>
```

```

<WifiSetup>
  <WifiSetupId>1</WifiSetupId>
  <SetupName>string</SetupName>
  <Priority>1</Priority>
  <SSID>string</SSID>
  <EncryptionAlgorithm>tKIP</EncryptionAlgorithm>
  <KeyManagementProtocol>WPAPSK</KeyManagementProtocol>
  <Passphrase>string</Passphrase>
  <TenantId>string</TenantId>
  <_802_1_SetupLookupKey>1</_802_1_SetupLookupKey>
</WifiSetup>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Successful request, content returned **WifiSetup**
  - 404 : Not found
  - 500 : Internal Server Error

## PUT /api/v2/wifiSetups/{wifiSetupId}

The **PUT /api/v2/wifiSetups/{wifiSetupId}** call updates a WiFiSetup by ID. (**WifiSetups\_Put**)

- Role required: Tenant Administrator
- When the KeyManagementProtocol is WPAPSK or WPA2PSK and the original KeyManagementProtocol is WPAPSK or WPA2PSK, Passphrase is optional, and \_802\_1\_SetupLookupKey is not needed and will be set to null internally. If Passphrase is not provided or empty, the original Passphrase will be used.
- When the KeyManagementProtocol is WPAPSK or WPA2PSK and the original KeyManagementProtocol is WPAIEEE802\_1 or WPA2IEEE802\_1, Passphrase is required, and \_802\_1\_SetupLookupKey is not needed and will be set to null internally.
- When the KeyManagementProtocol is WPAIEEE802\_1 or WPA2IEEE802\_1, \_802\_1\_SetupLookupKey is required, and Passphrase is not needed and will be set to null internally.
- The path parameter is **wifiSetupId**. WifiSetup ID
- This API call consumes the following media types via the Content-Type request header.
  - application/json
  - text/json
  - application/xml
  - text/xml
  - application/x-www-form-urlencoded
- The request body is **wifiSetupId**. This is the Wifi Setups object.

```

{
  "WifiSetupId": 0,
  "SetupName": "string",
  "Priority": 0,
  "SSID": "string",
  "EncryptionAlgorithm": "tKIP",
  "KeyManagementProtocol": "WPAPSK",
  "Passphrase": "string",
  "TenantId": "string",

```

```

    "_802_1_SetupLookupKey": 0
  }

```

- Return type: **WifiSetup**
- Example data: Content-Type = Application/JSON and application/XML.
  - JSON.

```

{
  "WifiSetupId": 0,
  "SetupName": "string",
  "Priority": 0,
  "SSID": "string",
  "EncryptionAlgorithm": "tKIP",
  "KeyManagementProtocol": "wPAPSK",
  "Passphrase": "string",
  "TenantId": "string",
  "_802_1_SetupLookupKey": 0
}

```

- XML

```

<?xml version="1.0"?>
<WifiSetup>
  <WifiSetupId>1</WifiSetupId>
  <SetupName>string</SetupName>
  <Priority>1</Priority>
  <SSID>string</SSID>
  <EncryptionAlgorithm>tKIP</EncryptionAlgorithm>
  <KeyManagementProtocol>wPAPSK</KeyManagementProtocol>
  <Passphrase>string</Passphrase>
  <TenantId>string</TenantId>
  <_802_1_SetupLookupKey>1</_802_1_SetupLookupKey>
</WifiSetup>

```

- This API call produces the following media types according to the Accept request header; the mediatype will be conveyed by the Content-Type response header
  - application/json
  - text/json
  - application/xml
  - text/xml
- Response types:
  - 200 : Returns updated WifiSetup **WifiSetup**
  - 400 : Bad request
  - 404 : Not found
  - 409 : WifiSetup name already exists
  - 500 : Internal Server Error

## DELETE /api/v2/wifiSetups/{wifiSetupId}

The **DELETE /api/v2/wifiSetups/{wifiSetupId}** call deletes a WiFiSetup by ID. (**WifiSetups\_Delete**)

- Role required: Tenant Administrator
- The path parameter is **wifiSetupId**. WifiSetup ID
- Response types:
  - 204 : Successful request, no content

- 404 : Not found
- 409 : Cannot delete Wifisetup if it is still associated with AMT Profile
- 500 : Internal Server Error

## 4 Troubleshooting

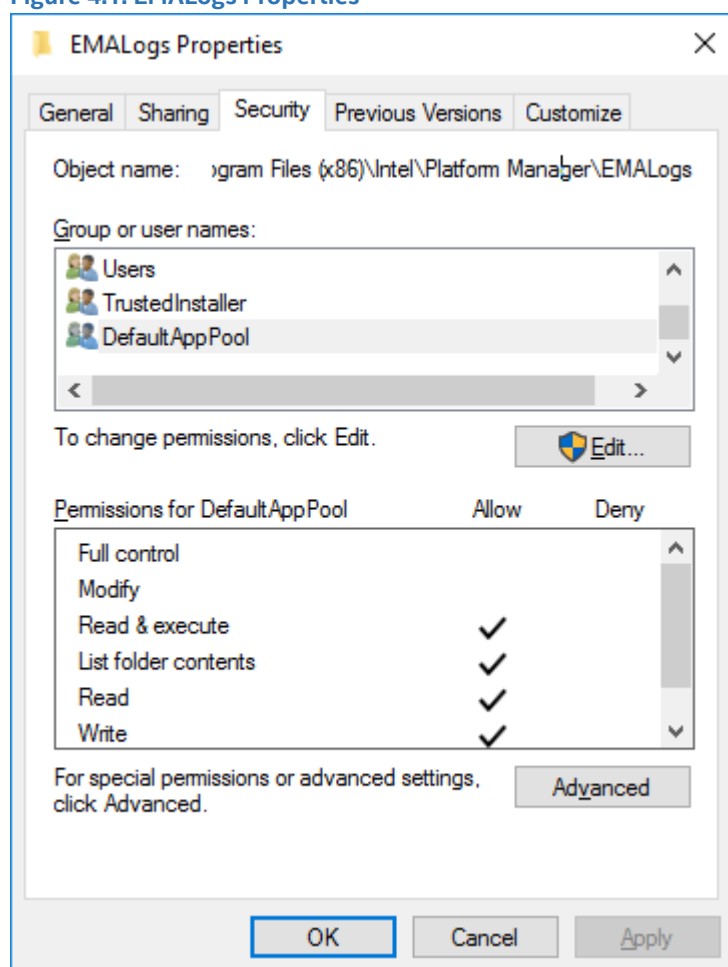
The Intel® EMA Server uses the NLog to log Intel® EMA API errors and debug information. The configuration file is located in **c:\inetpub\wwwroot\NLog.config**.

The default directory where the troubleshooting logs are written is **C:\Program Files (x86)\Intel\Platform Manager\EMALogs**.

Troubleshooting logs are written in the subdirectory EmaWebApiLogs within, which is a directory for each tenant.

For this release, the write permission to **C:\Program Files (x86)\Intel\Platform Manager\EMALogs** must be configured for the system account **IIS AppPool\DefaultAppPool**.

Figure 4.1: EMALogs Properties



## 5 HTTP Status Codes

The Intel® EMA API uses HTTP status codes that generally follow REST conventions where 2xx indicates success, 4xx indicates client errors, and 5xx indicates server errors. The following error codes are commonly used in the Intel® EMA API.

- **200 OK:** Successful request with content returned.
- **204 No Content:** Successful request with no content returned.
- **400 Bad Request:** bad request from the client.
- **401 Unauthorized:** the client is not authenticated.
- **403 Forbidden:** the client does not have the correct permissions to access the resource.
- **404 Not Found:** the requested resource is not found.
- **405 Method Not Allowed:** the requested method is not supported by the resource.
- **409 Conflict:** the request could not be completed do to a conflict with the current state of the resource.
- **415 Unsupported Media Type:** the resource does not support the media type from the request.
- **500 Internal Server Error:** the server encountered an unexpected error.

In some cases the error messages will be returned as the Intel® EMA API Extended Errors in following format:

```
{
  "Message": {
    "ExtendedCode": string,
    "ExtendedMessage": "string"
  }
}
```

### 400 Bad Request Errors

The following is a table listing the Intel® EMA API Extended Errors for error code 400.

**Table 5.1: 400 Extended Intel® EMA API Errors**

| Extended Code | Error Type                            | Extended Message                     |
|---------------|---------------------------------------|--------------------------------------|
| 1000          | OFFSET_PARAMETER_INVALID_400          | Offset parameter is invalid          |
| 1001          | PAGESIZE_PARAMETER_INVALID_400        | pageSize parameter is invalid        |
| 1002          | POWERSTATE_PARAMETER_INVALID_400      | powerState parameter is invalid      |
| 1003          | CONNECTIONSTATE_PARAMETER_INVALID_400 | connectionState parameter is invalid |
| 1004          | STARTDATE_PARAMETER_INVALID_400       | startDate parameter is invalid       |
| 1005          | ENDDATE_PARAMETER_INVALID_400         | endDate parameter is invalid         |
| 1006          | ENABLED_PARAMETER_INVALID_400         | enabled parameter is invalid         |
| 1007          | TENANTID_PARAMETER_INVALID_400        | tenantId parameter is invalid        |
| 1008          | ROLEID_PARAMETER_INVALID_400          | roleId parameter is invalid          |
| 1009          | SEARCHTYPE_PARAMETER_INVALID_400      | searchType parameter is invalid      |

|             |                                                      |                                                                         |
|-------------|------------------------------------------------------|-------------------------------------------------------------------------|
| <b>1010</b> | SEARCHVALUE_PARAMETER_INVALID_400                    | searchValue parameter is invalid                                        |
| <b>1011</b> | SEARCH_PARAMETERS_INVALID_400                        | both search parameters are required                                     |
| <b>1012</b> | UPDATE_PARAMETERS_INVALID_400                        | update parameters are required                                          |
| <b>1013</b> | ENDPOINTGROUPID_PARAMETER_INVALID_400                | endpointGroupId parameter is invalid                                    |
| <b>1014</b> | USERGROUP DISSOCIATION_INVALID_400                   | userGroup parameter is invalid                                          |
| <b>1015</b> | IP_ADDRESS_INVALID_400                               | IP address is invalid                                                   |
| <b>1016</b> | START_IP_ADDRESS_INVALID_400                         | Start IP address is invalid                                             |
| <b>1017</b> | END_IP_ADDRESS_INVALID_400                           | End IP address is invalid                                               |
| <b>1018</b> | SUBNETMASK_INVALID_400                               | Subnetmask is invalid                                                   |
| <b>1019</b> | AMT_NOT_SUPPORTED_400                                | Endpoint does not have Intel® AMT support                               |
| <b>1020</b> | AMT_FW_INVALID_400                                   | Endpoint Intel® ME version is invalid for Intel® AMT provisioning       |
| <b>1021</b> | AMT_ONLY_EITHER_TLS_OR_CIRA_CAN_BE_SELECTED_400      | Only either TLS or CIRA can be selected for Intel AMT provisioning      |
| <b>1022</b> | AMT_USB_PROVISIONING_NOT_SUPPORTED_400               | Endpoint does not support Intel® AMT provisioning using USB             |
| <b>1023</b> | AMT_HBP_PROVISIONING_NOT_SUPPORTED_400               | Endpoint does not support Intel® AMT host based provisioning            |
| <b>1024</b> | AMT_PKI_PROVISIONING_NOT_SUPPORTED_400               | Endpoint does not support Intel® AMT PKI certificate based provisioning |
| <b>1025</b> | AMT_PROVISIONING_CERT_HASH_TYPE_UNKNOWN_400          | Intel® AMT provisioning certificate hash type is unknown                |
| <b>1026</b> | AMT_CIRA_NOT_SUPPORTED_400                           | Endpoint does not support CIRA                                          |
| <b>1027</b> | CURRENT_PASSWORD_CANNOT_BE_NULL_OR_EMPTY_400         | Current password cannot be null or empty                                |
| <b>1028</b> | CALLERID_PARAMETER_INVALID_400                       | callerId parameter is invalid                                           |
| <b>1029</b> | WIFISSETUP_ENABLED_BUT_NO_WIFISSETUPID_SELECTED_400  | Wifi Connection Setup enabled, but no WifiSetupId selected              |
| <b>1030</b> | WIFISSETUP_NOT_ENABLED_BUT_WIFISSETUPID_SELECTED_400 | Wifi Connection Setup not enabled, but WifiSetupId(s) selected          |
| <b>1031</b> | ENDPOINTGROUP_AMTPROFILEID_NOT_EXISTS_400            | Cannot update Endpoint Group since AmtProfile ID not exists             |
| <b>1032</b> | PASSWORD_RESET_FAILED_DUE_TO_BAD_CREDENTIALS_400     | Password reset failed due to bad credentials                            |



|             |                                                                |                                                                                   |
|-------------|----------------------------------------------------------------|-----------------------------------------------------------------------------------|
| <b>1033</b> | WIFISSETUP_IDS_NOT_IN_DATABASE_400                             | WiFiSetupIds(s) associated with Intel® AMT Profile were not found in the Database |
| <b>1034</b> | MODEL_CANNOT_BE_NULL_400                                       | Model cannot be null                                                              |
| <b>1035</b> | CALLER_NOT_PERMITTED_TO_CREATE_USER_HAVING_SELECTED_ROLE_400   | Caller not permitted to create a new user with selected role                      |
| <b>1036</b> | CALLER_NOT_PERMITTED_TO_CREATE_USER_WITH_SELECTED_TENANTID_400 | Caller not permitted to create a new user with a different tenant                 |
| <b>1037</b> | CALLER_NOT_PERMITTED_TO_CREATE_USER_WITHOUT_TENANTID_400       | Caller not permitted to create a new user without a tenant                        |
| <b>1038</b> | CALLER_NOT_PERMITTED_TO_UPDATE_USER_WITH_SELECTED_ROLE_ID_400  | Caller not permitted to update user with selected role                            |
| <b>1039</b> | NAME_ONLY_ONE_PER_REQUEST_400                                  | Only one name per request allowed                                                 |
| <b>1040</b> | PASSWORD_ONLY_ONE_PER_REQUEST_400                              | Only one password per request allowed                                             |
| <b>1041</b> | FILE_ONLY_ONE_PER_REQUEST_400                                  | Only one file per request allowed                                                 |
| <b>1042</b> | FILE_INVALID_400                                               | Uploaded file is invalid                                                          |
| <b>1043</b> | CERTIFICATE_NAME_INVALID_400                                   | Certificate name is invalid                                                       |
| <b>1044</b> | CERTIFICATE_PASSWORD_INVALID_400                               | Certificate password is invalid                                                   |
| <b>1045</b> | CERTIFICATE_IMPORT_FAILED_400                                  | Certificate import failed. Please check that the .PFX file and password are valid |
| <b>1046</b> | UPLOADED_FILE_NOT_AMT_PROVISIONING_CERTIFICATE_400             | Uploaded file is not an Intel® AMT provisioning certificate                       |
| <b>1047</b> | CERTIFICATE_ID_INVALID_400                                     | Certificate ID is invalid                                                         |
| <b>1048</b> | CERTIFICATE_IS_NOT_FOR_AMT_PROVISIONING_400                    | Certificate is not for Intel® AMT provisioning                                    |
| <b>1049</b> | AMTSETUPID_SPECIFIED_IN_REQUEST_IS_INVALID_400                 | The AmtSetupId specified in the request is invalid                                |
| <b>1050</b> | AMTPROFILE_ID_INVALID_400                                      | Intel® AMT Profile ID is invalid                                                  |
| <b>1051</b> | CERTIFICATE_HAS_EXPIRED_400                                    | Certificate has expired                                                           |
| <b>1052</b> | CIRA_INTRANET_SUFFIX_INVALID_400                               | CIRA Intranet Suffix is invalid                                                   |
| <b>1053</b> | CIRA_LIMIT_EXCEEDED                                            | CIRA Proxies limit exceeded                                                       |
| <b>1054</b> | CREATE_DISABLED_USER_NOT_ALLOWED_400                           | Creating new user with enabled set to false is not permitted                      |
| <b>1055</b> | _802_1X_SETUP_ROOT_CERTIFICATE_REQUIRED_400                    | A root certificate is required for the specified authentication protocol          |

|             |                                                                     |                                                                                                 |
|-------------|---------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <b>1056</b> | _802_1X_SETUP_CLIENT_CERTIFICATE_REQUIRED_400                       | A client certificate is required for the specified authentication protocol                      |
| <b>1057</b> | _802_1X_SETUP_INVALID_CONFIGURATION_SETTINGS_REQUIRED_400           | Invalid configuration settings for certificate (required), please review                        |
| <b>1058</b> | _802_1X_SETUP_INVALID_CONFIGURATION_SETTINGS_400                    | Invalid configuration settings for certificate, please review                                   |
| <b>1059</b> | _802_1X_SETUP_PSK_PARAMETER_INVALID_400                             | The field PSK is not a valid OctetString                                                        |
| <b>1060</b> | _802_1X_SETUP_PROTECTED_ACCESS_CREDENTIAL_PARAMETER_INVALID_400     | The field ProtectedAccessCredential is not a valid OctetString                                  |
| <b>1061</b> | _802_1X_SETUP_SERVER_CERTIFICATE_NAME_COMPARISON_OPTION_INVALID_400 | The field ServerCertificateNameComparisonOption is not a valid number                           |
| <b>1062</b> | MAX_FILE_SIZE_EXCEEDED_400                                          | The file size exceeds the maximum allowed                                                       |
| <b>1063</b> | ENDPOINTS_LIST_CANNOT_BE_EMPTY_400                                  | Endpoints list cannot be empty                                                                  |
| <b>1064</b> | _802_1X_SETUP_PROTOCOL_INVALID_400                                  | The authentication protocol is not supported.                                                   |
| <b>1065</b> | AMT_GLOBALLY_DISABLED_400                                           | Intel® AMT in Endpoint is globally disabled and cannot be provisioned.                          |
| <b>1066</b> | MEBX_PASS_CHANGE_HBP_NOT_SUPPORTED_400                              | Intel® MEBx password change is not supported during Host Based Provisioning                     |
| <b>2022</b> | INVALID_USERNAME_FORMAT_400                                         | Invalid username format                                                                         |
| <b>3002</b> | RESOURCEID_ON_PATH_AND_MODEL_DO_NOT_MATCH_400                       | Resource Id value on the path to the controller and Resource Id in the input model do not match |
| <b>3003</b> | ENDPOINTID_PARAMETER_INVALID_400                                    | Input endpoint Id parameter is invalid                                                          |
| <b>3004</b> | USERGROUPID_PARAMETER_INVALID_400                                   | Input endpoint Id parameter is invalid                                                          |
| <b>3005</b> | TENANTID_OF_REQUESTED_ROLE_AND_TARGET_USER_DO_NOT_MATCH_400         | TenantId of requested role and target user do not match                                         |
| <b>3008</b> | MAC_ADDRESS_INVALID_400                                             | MAC address is invalid or does not exist                                                        |
| <b>3009</b> | IP_RANGE_INVALID_400                                                | End IP Address cannot be lower than Start IP Address                                            |
| <b>3011</b> | INVALID_PASSWORD_FORMAT_400                                         | Invalid password format                                                                         |

|             |                                       |                                                    |
|-------------|---------------------------------------|----------------------------------------------------|
| <b>3012</b> | UPN_IS_NOT_IN_DOMAIN_400              | UPN used is not registered in this domain          |
| <b>3017</b> | MANUAL_USER_LOCKING_NOT_SUPPORTED_400 | Manually locking user is not currently supported   |
| <b>3018</b> | POWER_OP_NOT_SUPPORTED_IN_FW_400      | Endpoint does not support the power operation sent |
| <b>3020</b> | ENDPOINT_NOT_ROUTABLE_400             | Endpoint is not routable                           |
| <b>3021</b> | BAD_PROVISIONING_STATE_400            | Endpoint provisioning state is not correct         |
| <b>3022</b> | POWER_OP_NOT_SUPPORTED_400            | Power operation sent is not supported              |
| <b>3026</b> | INVALID_AMT_CREDENTIAL_TYPE_400       | Input Intel® AMT credential type is invalid        |

## 401 Method Not Allowed Errors

The following is a table listing the Intel® EMA API Extended Errors for error code 401.

**Table 5.2: 401 Intel® EMA API Extended Errors**

| <b>Extended Code</b> | <b>Error Type</b>                    | <b>Extended Message</b>                     |
|----------------------|--------------------------------------|---------------------------------------------|
| <b>3007</b>          | UNAUTHORIZED_USER_NOT_REGISTERED_401 | User is not registered in Intel® EMA system |

## 403 Forbidden Errors

The following is a table listing the Intel® EMA API Extended Errors for error code 403.

**Table 5.3: 403 Intel® EMA API Extended Errors**

| <b>Extended Code</b> | <b>Error Type</b>                       | <b>Extended Message</b>                                         |
|----------------------|-----------------------------------------|-----------------------------------------------------------------|
| <b>3006</b>          | USER_LOCKED_OR_DELETED_403              | The user is locked or doesn't exist                             |
| <b>3015</b>          | AMT_PROVISION_RECORD_RETRIEVE_FORBIDDEN | User has insufficient rights to retrieve Intel® AMT credentials |
| <b>3019</b>          | POWER_OP_NOT_ALLOWED_403                | Endpoint is not allowed to execute this power operation         |
| <b>3023</b>          | POWER_OP_USER_FORBIDDEN_403             | User not allowed to execute power operations                    |

## 404 Not Found Errors

The following is a table listing the Intel® EMA API Extended Errors for error code 404.

**Table 5.4: 404 Intel® EMA API Extended Errors**

| Extended Code | Error Type                              | Extended Message                                          |
|---------------|-----------------------------------------|-----------------------------------------------------------|
| <b>3013</b>   | AMT_PROVISION_RECORD_NOT_FOUND          | Intel® AMT Provisioning Record does not exist in database |
| <b>3014</b>   | UNABLE_TO_RETRIEVE_AMT_PROVISION_RECORD | Intel® AMT Provisioning Record does not exist in databas  |
| <b>3016</b>   | ENDPOINTID_RECORD_NOT_FOUND             | Endpoint record does not exist in database                |
| <b>3025</b>   | UNABLE_TO_RETRIEVE_MEBX_PASSWORD        | Intel® MEBx password doesn't exist in the database        |

## 405 Method Not Allowed Errors

The following is a table listing the Intel® EMA API Extended Errors for error code 405.

**Table 5.5: 405 Intel® EMA API Extended Errors**

| Extended Code | Error Type                             | Extended Message                                      |
|---------------|----------------------------------------|-------------------------------------------------------|
| <b>3010</b>   | METHOD_NOT_ALLOWED_WRONG_AUTH_MODE_405 | Method not allowed due to current authentication mode |

## 409 Conflict Errors

The following is a table listing the Intel® EMA API Extended Errors for error code 409.

**Table 5.6: 409 Extended Intel® EMA API Errors**

| Extended Code | Error Type                                          | Extended Message                                              |
|---------------|-----------------------------------------------------|---------------------------------------------------------------|
| <b>2001</b>   | USER_GROUP_NAME_CONFLICT_409                        | UserGroup.Name already exists                                 |
| <b>2002</b>   | ENDPOINT_GROUP_NAME_CONFLICT_409                    | EndpointGroup.Name already exists                             |
| <b>2003</b>   | USER_NAME_CONFLICT_409                              | User.Name already exists                                      |
| <b>2004</b>   | TENANT_NAME_CONFLICT_409                            | Tenant.Name already exists                                    |
| <b>2005</b>   | USER_GROUP_TO_ENDPOINT_GROUP_CONFLICT_409           | UserGroup and EndpointGroup are already associated            |
| <b>2006</b>   | AMT_NOT_PROVISIONED_UNPROVISIONING_NOT_POSSIBLE_409 | Unprovision is not possible since endpoint is not provisioned |
| <b>2007</b>   | AMT_ALREADY_PROVISIONED_409                         | Intel® AMT is already provisioned                             |

|             |                                                       |                                                                                                       |
|-------------|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| <b>2008</b> | WIFISSETUP_NAME_CONFLICT_409                          | WiFiSetup.SetupName already exists                                                                    |
| <b>2009</b> | AMTPROFILE_NAME_CONFLICT_409                          | AMTProfile.Name already exists                                                                        |
| <b>2010</b> | AMTPROFILE_STILL_LINKED_TO_ENDPOINTGROUP_CONFLICT_409 | Cannot delete Intel AMT Profile since it is still linked to at least one Endpoint Group               |
| <b>2011</b> | WIFISSETUP_STILL_LINKED_TO_AMTPROFILE_CONFLICT_409    | Cannot delete Wifi Setup since it is still linked to Intel AMT Profile                                |
| <b>2012</b> | REQUEST_NOT_POSSIBLE_TCPRELAY_DISABLED_IN_POLICY_409  | Request is not possible because TCPRELAY is disabled in the Policy Group                              |
| <b>2013</b> | AMTCERTIFICATE_PART_OF_ANOTHER_CHAIN_409              | Intel® AMT certificate could not be deleted, since it is part of another Intel AMT Certificate Chain  |
| <b>2014</b> | AMTCERTIFICATE_IN_USE_BY_PROFILE_409                  | Intel® AMT certificate could not be deleted, since it is use in an Intel AMT Profile for provisioning |
| <b>2015</b> | CERTIFICATE_THUMBPRINT_ALREADY_EXISTS_409             | Certificate could not be imported, since its thumbprint already exists in the database                |
| <b>2016</b> | CERTIFICATE_NAME_ALREADY_EXISTS_409                   | Certificate not imported, since its name is already in use                                            |
| <b>2017</b> | _802_1X_SETUP_NAME_CONFLICT_409                       | 802_1XSetup.SetupName already exists                                                                  |
| <b>2018</b> | _802_1X_SETUP_ID_CONFLICT_409                         | SetupId in model and from request don't match                                                         |
| <b>2019</b> | _802_1X_SETUP_STILL_LINKED_TO_WIFISSETUP_CONFLICT_409 | Cannot delete 802.1x Setup since it is still linked to WiFi Seup                                      |
| <b>2020</b> | _802_1X_SETUP_STILL_LINKED_TO_AMTPROFILE_CONFLICT_409 | Cannot delete 802.1x Setup since it is still linked to Intel AMT Profile                              |

|             |                                                       |                                                                                                 |
|-------------|-------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <b>2021</b> | AMTPROFILE_STILL_LINKED_TO_INTELAMTSETUP_CONFLICT_409 | Cannot delete Intel AMT Profile since it is still linked to at least one Intel AMT Setup record |
| <b>2023</b> | ENDPOINT_NOT_ROUTABLE_CONFLICT_409                    | Endpoint must be CIRA connected or have neighbors                                               |
| <b>2024</b> | AMT_PROVISION_STATE_CONFLICT_409                      | Intel® AMT must be in post-provisioning state                                                   |
| <b>2025</b> | AMT_CONNECTION_CONFLICT_409                           | Intel® AMT connection problem                                                                   |
| <b>3024</b> | FW_NOT_READY_409                                      | Endpoint is not ready to execute this operation yet, please wait and retry                      |

## 415 Unsupported Media Type Errors

The following is a table listing the Intel® EMA API Extended Errors for error code 415.

**Table 5.7: 405 Intel® EMA API Extended Errors**

| <b>Extended Code</b> | <b>Error Type</b>                     | <b>Extended Message</b>        |
|----------------------|---------------------------------------|--------------------------------|
| <b>3000</b>          | BAD_MEDIA_ONLY_TEXT_FILE_ACCEPTED_415 | Only the text file is accepted |

## Appendix - 802.1x Setup Using Intel® EMA Web API



**Note:** You must be logged in as a Tenant Administrator user to perform the steps in this section.

The sample files are in the folder [Intel® EMA installation package folder] \Samples. These files are not automatically hosted on the Intel® EMA website during installation. These sample files are implemented using bare-minimum code to demonstrate how to use the API and do not use secure coding practices to guard against security concerns like cross-site scripting.



**IMPORTANT!** These samples should *never* be hosted in a production environment.

For hosting in a test environment for development purposes, copy the Samples folder to the Intel® EMA website root folder (e.g., C:\inetpub\wwwroot\).

### Example 802.1X Setup

This example shows how to create an 802.1X setup to be used by both network connection types. This example is intended for learning purposes and is not intended for use “as is” in production environments where security is a concern.

1. Access Intel® EMA Web API sample 802\_1XSetup.html and Get Token. (You can also use the Intel® EMA website GUI to perform this.
2. Go to Get the Certificate Authorities in the Domain section and click the **Get CAs** button.



**Note:** This endpoint will return a list of the Certification Authorities in the network and its Certificate Templates. These values will be used below to create the 802.1X setup.

3. Go to the POST 802.1X Setup section. Specify a **Name** for the setup, then specify the following settings to use EAP-TLS protocol and request the client certificate to the Certification Authority (refer to the 802.1X Setup Settings section for a detailed description of each property):

```
{
  "UseRoamingIdentity": false,
  "AuthenticationProtocol": 0,
  "AvailableInS0": true,
  "PSK": null,
  "PACPassword": null,
  "ProtectedAccessCredential": null,
  "Domain": null,
  "Password": null,
  "Username": null,
  "ServerCertificateName": "WIN-GUVUHKBNQ69.vprodemo.com",
  "ServerCertificateNameComparisonOption": 2,
  "RoamingIdentity": null,
  "PXETimeoutInSeconds": 300,
  "ADIntegrationSettings":
  {
    "OrganizationUnit": "VProDevComputers",
    "SecurityGroups": ["CN=vPro8021XComputers,DC=VPRODEMO,DC=COM"],
    "DomainName": "vprodemo.com"
  },
  "ClientAuthenticationSettings":
  {
    "ClientCertificateSettings":
    {
      "Source": 1,
      "Thumbprint": null,
      "CAName": "192.168.1.2\\VPRODEMO-WIN-GUVUHKBNQ69-CA",
      "Template": "AMTMachine"
    },
    "CommonNamesOption": 0,
    "CommonNames": null,
    "DesignatedSubjectCN": 4
  },
  "RootCertificateSettings":
  {
    "Source": 1,
    "Thumbprint": null,
    "CAName": "192.168.1.2\\VPRODEMO-WIN-GUVUHKBNQ69-CA"
  }
}
```

4. Click **Post 802.1X Setup** to create the setup and get its corresponding ID.

## WiFi Setup

Create a WiFi Setup to be used by the Intel® AMT Profile for wireless connection.

1. Access Intel® EMA Web API sample WifiSetup.html and Get Token. (You can also use the Intel® EMA website GUI to perform this. See the *Intel® EMA Administration and Usage Guide* for further details.)
2. Go to the POST WiFi Setup section, and specify a **Name** for the setup, and the following values:

- SSID Name: "NETGEAR45"
- Key Management Protocol: WiFi Protected Access II Enterprise (WPA2-802.1X)
- Encryption Algorithm: Counter mode CBC MAC Protocol
- 802.1X Setup ID: The ID obtained in the 802.1X Setup subsection



**Note:** In case the WPA encryption mode was set in the Wireless Connection subsection, the values combination must be "KMP=WiFi Protected Access Enterprise (WPA-802.1X); EA=Temporal Key Integrity Protocol"

3. Click **Post WiFi Setup** to create the setup and get its corresponding ID.



## Intel® AMT Profiles

Create two profiles, one for each type of network connection.

### Wired Connection

1. Access the Intel® EMA Web API sample AMTProfiles.html and Get Token. (You can also use the Intel® EMA website GUI to perform this. See the *Intel® EMA Administration and Usage Guide* for further details.)
2. Go to the POST AMT Profile section, set a **Name** and **Description** for the Profile and the following values:

- Setup: Setup TLS security
- Power Active On: 1
- Power State Idle Timeout: 1
- Management Interfaces Settings:

```
{
  "WebUIServiceEnabledState": 2,
  "KVMInterfaceState": 2,
  "UserConsentRequired": 1,
  "UserConsentDisplayTimeout": 60,
  "SOLEnabled": true,
  "IDEREnabled": true,
  "RedirectionServiceState": 32771
}
```

- Fqdn Settings:

```
{
  "FqdnSource": 0
}
```

- IP Settings:

```
{
  "DHCPEnabled": true,
  "SharedStaticIp": false,
  "Source": 0,
  "IP": "1.1.1.1",
  "SubnetMask": "255.255.255.0",
  "DefaultGateway": "1.1.1.1",
  "PrimaryDNS": "1.1.1.1",
  "SecondaryDNS": "1.1.1.1"
}
```

- WiFi Connection Settings:

```
{
  "AMTHostWiFiSyncEnabled": 0,
  "WiFiConnectionEnabledConfiguration": 0,
  "WiFiEnabledInPowerState": 2,
  "WiFiSetups": []
}
```

- Wired 802.1X Settings. Set the ID obtained in the 802.1X Setup subsection:

```
{
  "_802_1Setup_DBLookupKey": 1
}
```

3. Click **Post AMT Profile** to create the profile and get its corresponding ID.

### Wireless Connection

This profile requires the WiFi Setup ID created in the previous subsection.

1. Access the Intel® EMA Web API sample AMTProfiles.html and Get Token. (You can also use the Intel® EMA website GUI to perform this.)
2. Go to the POST AMT Profile section, set a **Name** and **Description** for the Profile and the following values:

- Setup: Setup TLS security
- Power Active On: 1
- Power State Idle Timeout: 1
- Management Interfaces Settings:

```
{
  "WebUIServiceEnabledState": 2,
  "KVMInterfaceState": 2,
  "UserConsentRequired": 1,
  "UserConsentDisplayTimeout": 60,
  "SOLEnabled": true,
  "IDEREnabled": true,
  "RedirectionServiceState": 32771
}
```

- Fqdn Settings:

```
{
  "FqdnSource": 0
}
```

- IP Settings:

```
{
  "DHCPEnabled": true,
  "SharedStaticIp": false,
  "Source": 0,
  "IP": "1.1.1.1",
  "SubnetMask": "255.255.255.0",
  "DefaultGateway": "1.1.1.1",
  "PrimaryDNS": "1.1.1.1",
  "SecondaryDNS": "1.1.1.1"
}
```

- WiFi Connection Settings. In "WiFiSetups" set the ID obtained in WiFi Setup creation:

```
{
  "AMTHostWiFiSyncEnabled": 1,
  "WiFiConnectionEnabledConfiguration": 1,
  "WiFiEnabledInPowerState": 1,
  "WiFiSetups": [1]
}
```

- Wired 802.1X Settings:

```
{
  "_802_1Setup_DBLookupKey": null
}
```

3. Click **Post AMT Profile** to create the profile and get its corresponding ID.

## Endpoint Group

Create two endpoint groups, each enabled with Intel® AMT autosetup, using the recently created two Intel® AMT Profiles, respectively. We used "Host-based provision" as the provision approach.

## Install Intel® EMA Agents

Use two Intel® AMT endpoints, which are all not-provisioned. Follow the normal Intel® EMA steps to register each endpoint to each endpoint group created early, respectively.



**Note:** The device must be connected to a regular port (no 802.1X authentication) during installation in order for the provisioning can be carried out.

## Test connections

### Wired connection

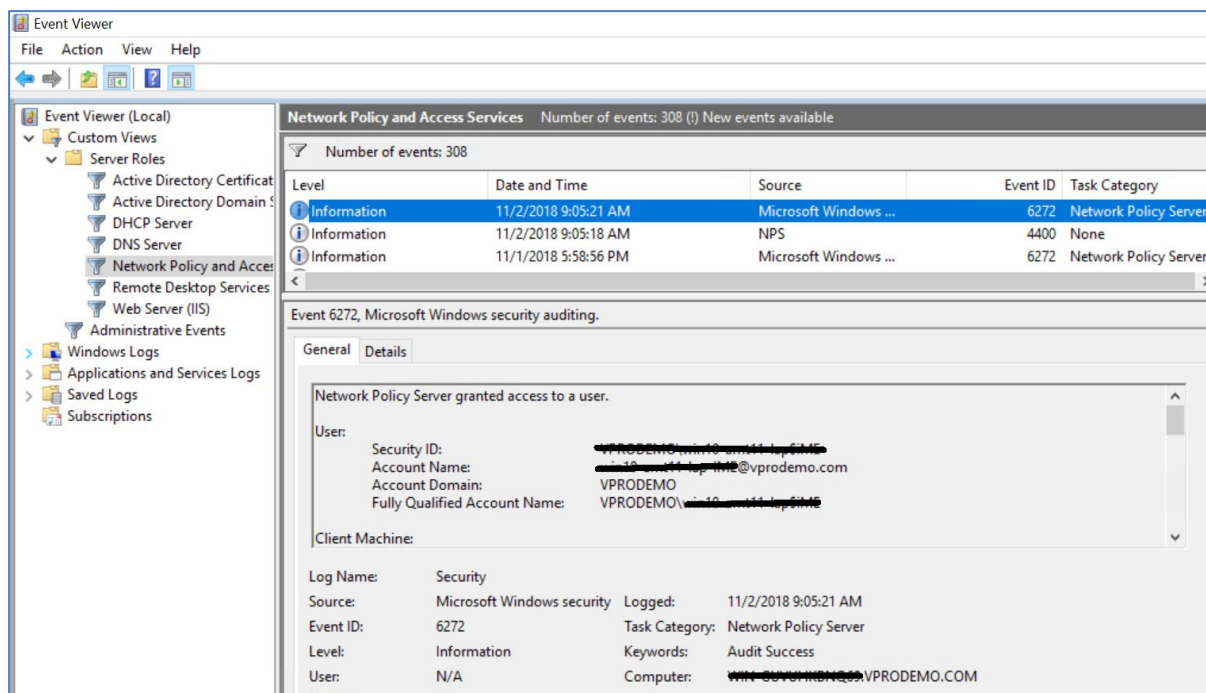
1. At the wired Ethernet switch, change the Ethernet cable to the port configured to use 802.1X authentication in the Wired Connection subsection.
2. Restart the endpoint to the BIOS Setup
3. On another computer (e.g., the server) open the MeshCommander Remote Management Console ([download](#))
4. Create an entry by clicking **Add Computer...**
  - a. Hostname: Set the IP Address for the Ethernet interface (this is the same address as in a no-802.1X connection)
  - b. Auth/Security: Digest/TLS
  - c. Username: admin
  - d. Password: The password set in the Intel® AMT Profile subsection
5. Click **Connect** for the entry. This should connect to Intel® AMT and load its properties. A Remote Desktop connection is also possible from this tool.

### Wireless connection

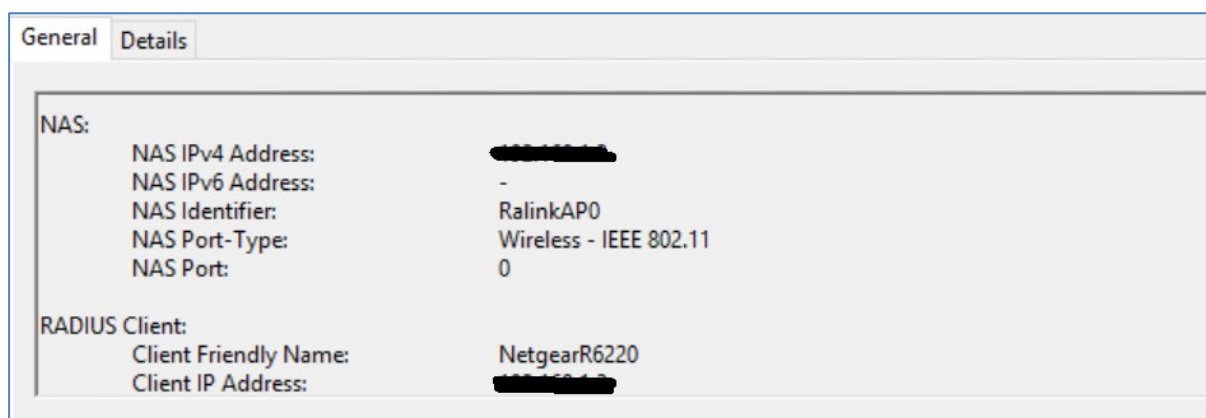
1. Unplug the Ethernet cable and activate the device's Wi-Fi interface without being connected to any network
2. Restart the endpoint to the BIOS Setup
3. In another computer (e.g., the server) open the MeshCommander Remote Management Console ([download](#))
4. Create an entry clicking **Add Computer...**
  - a. Hostname: Set the IP Address for the Wi-Fi interface (this is the same address as in a no-802.1X connection)
  - b. Auth/Security: Digest/TLS
  - c. Username: admin
  - d. Password: The password set in the Intel® AMT Profile subsection
5. Click **Connect** for the entry. This should connect to Intel® AMT and load its properties. A Remote Desktop connection is also possible from this tool.

To verify the access granted by the RADIUS server, go to the Windows Server's Event Viewer in **Custom Views > Server Roles > Network Policy and Access Services**.

The latest event should be the recent connection, indicating a user was granted, as shown in the figure below.



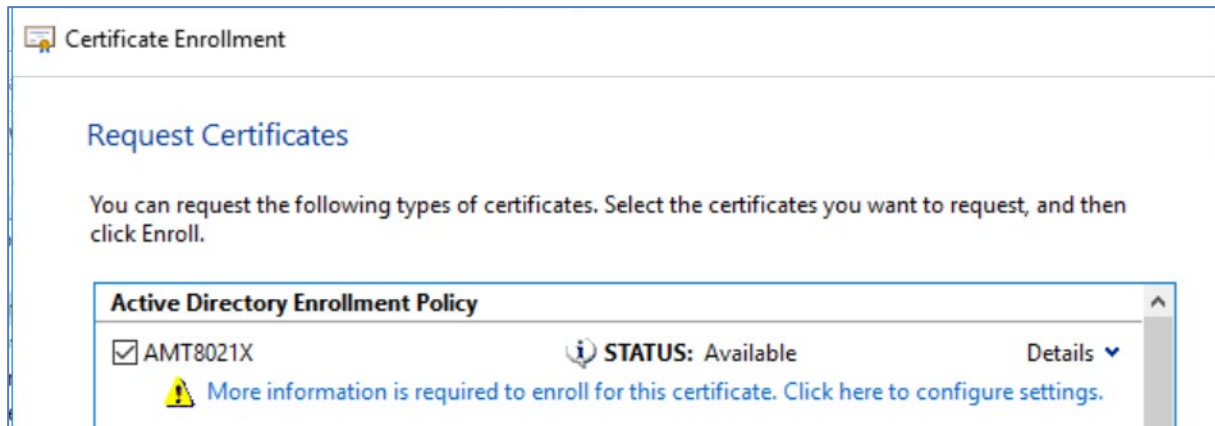
Also, the requester RADIUS client can be seen in the event shown in the figure below.



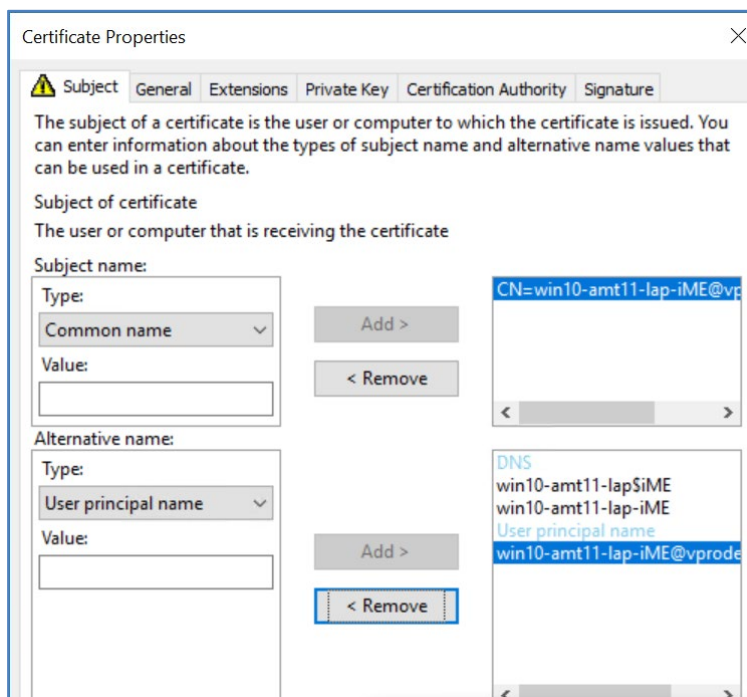
## Add certificates to Intel® EMA database

The following instructions indicate how to manually get both root and client certificates to be uploaded to Intel® EMA's database in order to be used by 802.1X Setup. This is required if the certificates will be obtained from Intel® EMA database.

1. Open the **Windows Certificates – Local Computer** console and go to the **Personal\Certificates** folder.
2. Right click on the Certificates folder and select **All Tasks > Request New Certificate**.
3. Click **Next** in the first two panels. Then select the Certificate Template created for 802.1X during environment setup and click the blue message link below it to supply more information.



4. On the **Subject** tab of **Certificate Properties**, add the following values:
  - Subject name:
    - Type: Common name; Value: User Principal Name of the Intel® AMT computer (e.g., win10-amt11-lap-iME@vprodemo.com). Note that you need to append -iME.
  - Alternative name:
    - Type: DNS; Value: SAMAccount Name (e.g., win10-amt11-lap-\$iME). Note that you need to append \$iME.
    - Type: DNS; Value: Common Name (e.g., win10-amt11-lap-iME). Note that you need to append -iME.
    - Type: User principal name; Value: User Principal Name (e.g., win10-amt11-lap-iME@vprodemo.com). Note that you need to append -iME.



5. Save the properties and click **Enroll** to create the certificate.
6. Find the certificate, right-click it, and select **All Tasks > Export**.
7. Click **Next** in the first panel and select **Yes, export the private key**.

8. In the Export File Format panel, the **Personal Information Exchange (.PFX)** option is selected by default. Under that, select both **Include all certificates in the certification path if possible** and **Export all extended properties**.
9. In the next panel, check the **Password** option and indicate a password to protect the private key. Remember this password as must be indicated when the certificate is uploaded to Intel® EMA.
10. Indicate the name and path to save the certificate and click **Finish** to complete the export.
11. Follow Intel® EMA Usage Guide to upload the certificate via Server Settings user interface. This operation will add two certificates to the database: the client certificate we created by enrollment and the root certificate of the Certificate Authority that provided it.
12. Use the same user interface to see the new certificates that were put into the database. The root certificate has the CA name in Name and Common Name properties. Then, you can download the certificate and check its thumbprint value, to be used later.

## 802.1X Setup Settings



**Note:** You must be logged in as a Tenant Administrator user to perform the steps in this section.

This section describes all of the properties that are set in the Intel® EMA Web API 802.1X Setup sample.

Only EAP-TLS Authentication protocol is fully supported. Each property has an annotation indicating if it is needed for this protocol and the possible values.

| Setting                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PSK                       | <p>A pre-shared key used for pre-shared key EAP types such as EAP-PSK, EAP-SIM, and EAP-AKA.</p> <p>Value type: OctetString (will be converted to byte array internally)</p> <p>EAP-TLS Protocol: This property is not evaluated, can be null or not present.</p>                                                                                                                                                                               |
| ProtectedAccessCredential | <p>A credential used by the supplicant and AAA server to establish a mutually authenticated encrypted tunnel for confidential user authentication.</p> <p>This field is relevant for EAP-FAST only. It is not required if the server is configured for "PAC provisioning".</p> <p>Value type: OctetString (will be converted to byte array internally)</p> <p>EAP-TLS Protocol: This property is not evaluated, can be null or not present.</p> |

| Setting                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Domain                                | <p>The domain (also known as realm) within which Username is unique.</p> <p>The Domain string shouldn't contain the suffix, so the user name (Domain\user) will be correct.</p> <p>If the Domain string contains a suffix (e.g. Domain = intel.com), the user trying to authenticate will be of the form intel.com\user (instead of intel\user) and thus authentication will fail.</p> <p>Value type: String</p> <p>EAP-TLS Protocol: This property is not evaluated, can be null or not present.</p>                                                                                                                                   |
| Username                              | <p>Identifies the user requesting access to the network.</p> <p>Value type: String</p> <p>EAP-TLS Protocol: This property is not evaluated, can be null or not present.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| ServerCertificateNameComparisonOption | <p>The comparison algorithm that shall be used by the server to validate the subject name field of the certificate presented by the AAA server against the value of the ServerCertificateName property.</p> <p>This field is mandatory if ServerCertificateName is defined.</p> <p>Values (Int):</p> <ul style="list-style-type: none"> <li>• 2 = FullName</li> <li>• 3 = DomainSuffix</li> </ul> <p>EAP-TLS Protocol: This property is evaluated if ServerCertificateName is defined.</p>                                                                                                                                              |
| ServerCertificateName                 | <p>The name that shall be compared against the subject name field in the certificate provided by the AAA server. Shall contain either the fully qualified domain name of the AAA server, in which case ServerCertificateNameComparison shall contain "FullName", or the domain suffix of the AAA server, in which case ServerCertificateNameComparison shall contain "DomainSuffix".</p> <p>This field is optional. If not defined, the name is not checked. The authenticity of the certificate is always verified.</p> <p>Value type: String</p> <p>EAP-TLS Protocol: This property is evaluated, but can be null or not present.</p> |

| Setting                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RoamingIdentity        | <p>A string presented to the authentication server in 802.1x protocol exchange. The AAA server determines the format of this string. Formats supported by AAA servers include: &lt;domain&gt;\&lt;username&gt;, &lt;username&gt;@&lt;domain&gt;.</p> <p>This string, if defined, is sent in response to 802.1x "request identity" as clear text. If empty, the username is sent. When UseRoamingIdentity is false, this should be null or not present.</p> <p>Value type: String</p> <p>EAP-TLS Protocol: This property is not evaluated, can be null or not present.</p>                                |
| AuthenticationProtocol | <p>AuthenticationProtocol shall indicate the desired EAP (Extensible Authentication Protocol) type.</p> <p>EAP-TLS (0): shall indicate that the desired EAP type is the Transport Layer Security EAP type specified in RFC 2716. If AuthenticationProtocol contains 0, Username should not be null, ServerCertificateName and ServerCertificateNameComparison may be null or not null, and RoamingIdentity, Password, Domain, ProtectedAccessCredential, PACPassword, and PSK should be null.</p> <p>Values (Int):</p> <ul style="list-style-type: none"> <li>0 = EAP-TLS</li> </ul>                     |
| PXETimeoutInSeconds    | <p>Timeout in seconds, in which the Intel® AMT will hold an authenticated 802.1X session. During the defined period, Intel® AMT manages the 802.1X negotiation while a PXE boot takes place. After the timeout, control of the negotiation passes to the host. This setting applies to Wired Connections.</p> <p>The maximum value is 86400 seconds (one day); A value of 0 disables the feature; If this optional value is omitted, Intel® AMT sets a default value of 120 seconds.</p> <p>Value type: Int</p> <p>EAP-TLS Protocol: This property is evaluated. Suggested 120 as the default value.</p> |



| Setting                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AvailableInS0                           | <p>Indicates the activity setting of the 802.1X module in S0 state. The default value for this property is 'true'. This setting applies to Wired Connections.</p> <p>Functionality:</p> <ul style="list-style-type: none"> <li>FALSE, AMT is not accessible (over 802.1x enabled port) in case the host is in S0 but fails to authenticate to the server.</li> <li>TRUE, AMT handles the authentication in this case (but the host still can't be accessed until it authenticates successfully).</li> </ul> <p>If 802.1X is not configured, this API may still succeed as the setting may be stored for future use.</p> <p>Value type: Boolean</p> <p>EAP-TLS Protocol: This property is evaluated</p> |
| Enabled                                 | <p>Indicates whether the 802.1x profile is enabled. If this property is no provided the default value is Enabled. This setting applies to Wired Connections.</p> <p>Values (Int):</p> <ul style="list-style-type: none"> <li>2 = Enabled</li> <li>3 = Disabled</li> </ul> <p>For Disabled status, the whole process is performed (creation of: AD object, certificate, 802.1X profile) but the 802.1x profile is disabled so it will not be used on authentication.</p> <p>EAP-TLS Protocol: This property is evaluated, but may not be present.</p>                                                                                                                                                   |
| UseRoamingIdentity                      | <p>If this is true, the user will connect to the RADIUS server with an identity defined at RoamingIdentity. If this is false, RoamingIdentity should be null or not present.</p> <p>Value type: Boolean</p> <p>EAP-TLS Protocol: This property is not evaluated. Set it to false.</p>                                                                                                                                                                                                                                                                                                                                                                                                                  |
| ADIntegrationSettings                   | <p>Group settings used to integrate Intel® AMT with the security infrastructure of the domain's Active Directory (AD)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <div></div> <div>OrganizationUnit</div> | <p>Organization Unit in AD which the computer object for AMT will be created, must be in distinguished name (DN) format</p> <p>Value type: String</p> <p>EAP-TLS Protocol: This property is evaluated</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

| Setting                      |                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                              | SecurityGroups            | <p>The AD Object created for the Intel® AMT device is by default automatically added to the AD Security group named "Domain Computers". If necessary, it is also possible to define additional Security groups to which the object will be added in case the RADIUS server requires objects to be members of a specific Security group.</p> <p>Value type: List of Strings. Each one should be the Distinguished Name of that security group.</p> <p>EAP-TLS Protocol: This property is evaluated</p>                                                                                                                                                                                                                                                                               |
| ClientAuthenticationSettings |                           | Group settings related to client certificate generation/retrieval for use in authentication.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                              | ClientCertificateSettings | Group settings to specify the information and entities involved in requesting and issuing client certificates to be used during client authentication.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|                              | Source                    | <p>For protocols that support it, allows specifying how to obtain the certificate used to authenticate to the server. Required for EAP_TLS protocol.</p> <p>Values (Int):</p> <ul style="list-style-type: none"> <li>0 = None</li> <li>1 = Request from CA (the CName setting is required for this type)</li> <li>2 = Pick from database (the Thumbprint and Template settings are required for this type)</li> </ul> <p>EAP-TLS Protocol: This property is evaluated.</p>                                                                                                                                                                                                                                                                                                          |
|                              | Thumbprint                | <p>Indicates the thumbprint property of the certificate that is already in the Intel® EMA database.</p> <p>These certificates are uploaded to the Intel® EMA's database through the Intel® EMA Web API sample <a href="#">AmtProvisioningCertificates.html</a> using the Upload AMT General Certificate PFX endpoint. The certificate will be uploaded to the Tenant's certificates and will not be marked as provisioning certificate.</p> <p>It is important to be sure the indicated certificate is of the type required by RADIUS.</p> <p>This is required only if <code>ClientAuthenticationSettings.ClientCertificateSettings.Source = 2</code>.</p> <p>Value type: String</p> <p>EAP-TLS Protocol: This property is evaluated if you use to use the certificate from DB.</p> |

| Setting |  |                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------|--|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         |  | CAName            | <p>The name of the Certification Authority to be contacted.</p> <p>It must be in the format: hostname\CA Name</p> <p>This is required only if<br/>ClientAuthenticationSettings.ClientCertificateSettings.Source = 1.</p> <p>Value type: String</p> <p>EAP-TLS Protocol: This property is evaluated if you use to use the certificate from CA.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|         |  | Template          | <p>The name of the Certificate Template to be required.</p> <p>It must be available and comply with EMA requirements for requesting and retrieving client certificates.</p> <p>This is required only if<br/>ClientAuthenticationSettings.ClientCertificateSettings.Source = 1.</p> <p>Value type: String</p> <p>EAP-TLS Protocol: This property is evaluated if you use to use the certificate from CA.</p>                                                                                                                                                                                                                                                                                                                                                                                                            |
|         |  | CommonNamesOption | <p>Indicates if the requested Certificate's Alternative Names will be defined by default or user defined.</p> <p>This is required only if<br/>ClientAuthenticationSettings.ClientCertificateSettings.Source = 1.</p> <p>Values (Int):</p> <ul style="list-style-type: none"> <li>• 0 = Default. When selected, the client certificate's Subject Alternative Name is configured as follows: <ul style="list-style-type: none"> <li>○ User Principal Name</li> <li>○ DNS Host Name (FQDN)</li> <li>○ Host Name</li> <li>○ SAM Account Name</li> <li>○ UUID of the new AD object representing the AMT system</li> <li>○ DistinguishedName</li> </ul> </li> <li>• 1 = User Defined</li> </ul> <p>EAP-TLS Protocol: This property is evaluated, if you request certificate from Microsoft* CA. Both ways are supported.</p> |

| Setting              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CommonNames          | <p>If the CommonNamesOption setting is set to User Defined, the common names indicated in this setting are added as the certificate's Subject Alternative Name.</p> <p>This is required only if<br/>ClientAuthenticationSettings.ClientCertificateSettings.Source = 1.</p> <p>Allowed CommonName items:</p> <ul style="list-style-type: none"> <li>• 0 = DNSFQDN</li> <li>• 1 = Hostname</li> <li>• 2 = SAMAccountName</li> <li>• 3 = UUID</li> <li>• 4 = UserPrincipalName</li> <li>• 5 = DistinguishedName</li> <li>• 6 = AMTRMCP</li> <li>• 7 = AMTSecureRMCP</li> <li>• 8 = AMTHHTTP</li> <li>• 9 = AMTHHTTPS</li> <li>• 10 = AMTTCPRedirect</li> <li>• 11 = AMTTLSRedirect</li> <li>• 12 = AMTVNCKVM</li> </ul> <p>Value type: List of CommonName ID (Int)</p> <p>EAP-TLS Protocol: This property is evaluated, if you request certificate from Microsoft* CA.</p> |
| DesignatedSubjectCN  | <p>Indicates the requested certificate's Subject Name. If this setting is not specified, the default value is 4 = UserPrincipalName.</p> <p>This is required only if<br/>ClientAuthenticationSettings.ClientCertificateSettings.Source = 1.</p> <p>Value type: Int. A CommonName item's ID.</p> <p>EAP-TLS Protocol: This property is evaluated, if you request certificate from Microsoft* CA.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| DesignatedUsernameCN | <p>Optional. If specified, the system will output the value of the designated common name into the Username field of the 802.1X setup. If this setting is not specified, the default value is 4 = UserPrincipalName.</p> <p>Value type: Int. A CommonName item's ID.</p> <p><b>Note:</b> For NPS, the confirmed value that works is the User Principal Name.</p> <p>EAP-TLS Protocol: This property is evaluated, but may not be present.</p>                                                                                                                                                                                                                                                                                                                                                                                                                           |

| Setting                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RootCertificateSettings | <p>Group settings for obtaining the trusted root certificate to be installed in Intel® AMT.</p> <p>If the CA name is not known beforehand, it can be obtained using the Intel® EMA Web API sample 802_1XSetup.html in the Get the Certificate Authorities In the Domain section, which returns a list of the CA found in the domain and its respective Certificate Templates.</p> <p>The user must be sure the indicated CA is the correct root CA that provides the trusted root certificate for the 802.1X authentication.</p> <p>Required for all authentication protocols except GTC.</p>                                                                             |
| Source                  | <p>Designates where the trusted root certificate used to authenticate the server is obtained from.</p> <p>Values (Int):</p> <ul style="list-style-type: none"> <li>0 = None</li> <li>1 = Request from CA (the CAName setting is required for this type)</li> <li>2 = Pick from database (the Thumbprint setting is required for this type)</li> </ul> <p>EAP-TLS Protocol: This property is evaluated.</p>                                                                                                                                                                                                                                                                |
| Thumbprint              | <p>Specifies the thumbprint property of the certificate that is already in the Intel® EMA database.</p> <p>These certificates are uploaded to the Intel® EMA's database through the Intel® Web API sample AmtProvisioningCertificates.html using the Upload AMT General Certificate PFX endpoint. The certificate will be uploaded to the Tenant's certificates and will not be marked as provisioning certificate.</p> <p>It is important to be sure the indicated certificate is of the type required by RADIUS.</p> <p>This is only required if RootCertificateSettings.Source = 2.</p> <p>Value type: String</p> <p>EAP-TLS Protocol: This property is evaluated.</p> |
| CAName                  | <p>The name of the root Certification Authority to be contacted.</p> <p>This is only required if RootCertificateSettings.Source = 1.</p> <p>EAP-TLS Protocol: This property is evaluated.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |