

Troubleshoot Actions on IMM Servers Through Intersight API Requests

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[API Requests from Intersight SaaS or Intersight Appliance Account](#)

[Overview Steps](#)

[Decommission/Recommission a Server](#)

[Unassign Server Profile](#)

[Remove Server](#)

[Troubleshoot Actions Through API Explorer in the Device Console](#)

[Reboot CIMC Management Controller of a Server](#)

[Reboot an I/O Module \(IOM\)](#)

[Related Information](#)

Introduction

This document describes API requests that can be useful at contention times when certain actions on servers cannot be performed through the UI.

Contributed by Luis Uribe Rojas, Cisco TAC Engineer and Justin Pierce, Technical Leader.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Intersight
- Unified Computing System (UCS) Servers
- Intersight Managed Mode (IMM)
- Application Programming Interface (API)

Components Used

The information in this document is based on these software and hardware versions:

- Cisco UCS 6454 Fabric Interconnect, firmware 4.2(1m)
- UCSB-B200-M5 blade server, firmware 4.2(1a)
- Intersight software as a service (SaaS)

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

Cisco Intersight provides a cloud-based RESTful API to manage Intersight-connected targets across multiple Data Centers. Cisco Intersight infrastructure services include the deployment, monitoring, management, and support for physical and virtual infrastructures.

In situations where certain actions on Intersight Managed Servers cannot be performed through the user interface (UI) of Intersight, either because options are grayed out or access to the UI is not available, API requests can be a useful alternative.

API Requests from Intersight SaaS or Intersight Appliance Account

Overview Steps

The next examples adhere to a consistent structure, although the specific parameters and values used can vary. This is a brief summary of the steps involved:

Log in to the Intersight account.

For a SaaS environment, in a browser, navigate to [API Reference](#) and log in with your account.

Note: For an appliance environment, such as Intersight Connected Virtual Appliance (CVA) or Intersight Private Virtual Appliance (PVA) In a browser navigate to <https://<Appliance-Hostname>/apidocs/apirefs> and log in with the Appliance credentials.

1. Look for the API request that you need and use a GET call filtered with known field value(s), such as Serial Number, Server Profile, Server Name, Device Moid, and so on.
2. Use the **PATCH** call with the correspondent Action to perform the required task

Tip: On Query Parameters, ensure to use the same exact letters for Key and Value Examples to avoid errors.

In the API Reference guide, it is useful to review the Response Model tab for the proper syntax and all the supported actions that can be used in the payload of a call. For example, from `/api/v1/compute/BladeIdentities/`, the supported AdminAction are **None**, **Decommission**, **Recommission**, **Reack**, **Remove**, and **Replace**. This model is used throughout this document.

Service: Intersight

API Reference v1.0.11-11265

compute/BladeIdentities

REST Client:

PATCH /api/v1/compute/BladeIdentities/{Moid}

Parameters Request Model Response Model

MO and potentially properties of Ancestor MOs. Displaynames are intended as a way to provide a normalized user appropriate name for an MO, especially for MOs which do not have a 'Name' property, which is the case for much of the inventory discovered from managed targets. There are a limited number of keys, currently 'short' and 'hierarchical'. The value is an array and clients should use the first element of the array.

AdminAction: string Updated by UI/API to trigger specific action type.

- * 'None' - No operation value for maintenance actions on an equipment.
- * 'Decommission' - Decommission the equipment and temporarily remove it from being managed by Intersight.
- * 'Recommission' - Recommission the equipment.
- * 'Reack' - Reacknowledge the equipment and discover it again.
- * 'Remove' - Remove the equipment permanently from Intersight management.
- * 'Replace' - Replace the equipment with the other one.

AdminActionState: string (Read Only) The state of Maintenance Action performed. This will have three states. Applying - Action is in progress. Applied - Action is completed and applied. Failed - Action has failed.

- * 'None' - Nil value when no action has been triggered by the user.
- * 'Applied' - User configured settings are in applied state.
- * 'Applying' - User settings are being applied on the target server.
- * 'Failed' - User configured settings could not be applied.

Identifier: integer Numeric Identifier assigned by the management system to the equipment. Identifier can only be changed if it has been PATCHED with the AdminAction property set to 'Recommission'.

Lifecycle: string (Read Only) The equipment's lifecycle status.

- * 'None' - Default state of an equipment. This should be an initial state when no state is defined for an equipment.
- * 'Active' - Default Lifecycle State for a physical entity.
- * 'Decommissioned' - Decommission Lifecycle state.

Decommission/Recommission a Server

In the Intersight API Reference document, look for the compute/BladeIdentities request, select the first GET call, and then enter the required Query Parameters.

This example uses these parameters:

Key	Value	Usage
\$filter	Serial Eq 'FLM2402001A'	To filter output to the server with the Serial Number provided.
\$select	Moid	To select the values to display from that object. Value displayed is the Server Moid.

The screenshot displays the Cisco Intersight Developer Center interface. On the left, the 'API Reference v1.0.11-11360' section is active, with a search bar containing 'blade'. Below the search bar, a list of API endpoints is shown, with 'compute/BladeIdentities' selected. The main content area shows the details for the GET endpoint `/api/v1/compute/BladeIdentities`. The endpoint description states: 'Filter criteria for the resources to return. A URI with a \$filter query option identifies a subset of the entries from the Collection of Entries. The subset is determined by selecting only the Entries that satisfy the predicate expression specified by the \$filter option. The expression language that is used in \$filter queries supports references to properties and literals. The literal values can be strings enclosed in single quotes, numbers and boolean values (true or false).' Below the description, several query parameters are listed with their types and descriptions:

- `$filter (string)`: Filter criteria for the resources to return.
- `$orderby (string)`: Determines what properties are used to sort the collection of resources.
- `$top (integer)`: Specifies the maximum number of resources to return in the response.
- `$skip (integer)`: Specifies the number of resources to skip in the response.
- `$select (string)`: Specifies a subset of properties to return.
- `$expand (string)`: Specify additional attributes or related resources to return in addition to the primary resources.

On the right side, a 'REST Client' panel is visible, showing a 'Send' button and a 'Response Text' section with a JSON response snippet.

Apply the PATCH call with the action required. This example uses:

```
{"AdminAction": "Decommission"}
```

The screenshot displays the Cisco Intersight Developer Center interface. The main content area shows the API endpoint `/api/v1/compute/BladeIdentities/{Moid}` with a `PATCH` method. The `Moid` parameter is described as "The unique Moid identifier of a resource instance." The `If-Match` header is described as a mechanism to prevent lost updates by checking the `ModTime` property of the resource. A REST Client panel on the right shows a JSON response snippet with fields like `"Moid"`, `"Object"`, `"ClassI"`, `"Create"`, `"ModTim"`, `"Tags"`, `"Owners"`, `"Scc1"`, `"6111"`, `]`, `"Share"`, `"Accoun"`, `"Domain"`, `"Ancest"`, `"Displa"`, `"hier"`, `"ch"`, and `]`.

Tip: If Recommission is required, use `{"AdminAction":"Recommission"}`.

Unassign Server Profile

Look for the *Server/Profiles* request and select the first **GET** call, then enter the required Query Parameters.

This example uses these parameters:

Key	Value	Usage
\$filter	Name Eq 'UCSX-Server-boot-SAN'	To filter output to server profile that has the name entered.
\$select	Moid,Name	To select the value(s) to display from that object. In this case

The screenshot shows the Cisco Intersight Developer Center interface. At the top, there are navigation links for Guides, API Reference, Downloads, Code Repo, and Support. The main content area is titled "GET /api/v1/server/Profiles" and includes a "REST Client" toggle. Below the endpoint name, there are tabs for "Parameters" and "Response Model". The "Parameters" tab is active, showing a list of query parameters: \$filter (string), \$orderby (string), \$stop (integer), \$skip (integer), \$select (string), and \$expand (string). Each parameter has a description of its function. On the left side, there is a "Service" dropdown set to "Intersight" and an "API Reference v1.0.11-11265" section with a search bar containing "server/Profiles". Below the search bar, a list of actions is shown, with the "PATCH" action highlighted. A "REST Client" panel is visible on the right side of the page, showing a "Send" button and a "Response Text" section with a JSON response.

Apply the **PATCH** call with the action required. This example uses:

```
{"Action": "Unassign"}
```

The screenshot shows the Cisco Intersight Developer Center API Reference page. The service is set to 'Intersight' and the API Reference version is 'v1.0.11-11265'. The search results for 'server/Profiles' are shown, with the 'PATCH' method for updating a resource highlighted. The main content area displays the endpoint `/api/v1/server/Profiles/{Moid}` and the `Moid` path parameter, which is described as 'The unique Moid identifier of a resource instance.' It also shows the `If-Match` header parameter, which is used to prevent lost updates by checking the resource's `ModTime` property. A REST Client sidebar is visible on the right, showing a request body with a `{Moid}` value.

Remove Server

In the Intersight API Reference document, look for `compute/BladeIdentities` request and select the first GET call, then enter the required Query Parameters.

This example uses these parameters:

Key	Value	Usage
\$filter	Serial Eq 'FLM2402001A'	To filter output to only server with Serial Number provided.
\$select	Moid	To select the values to display from that object. Value displayed is the Server Moid.

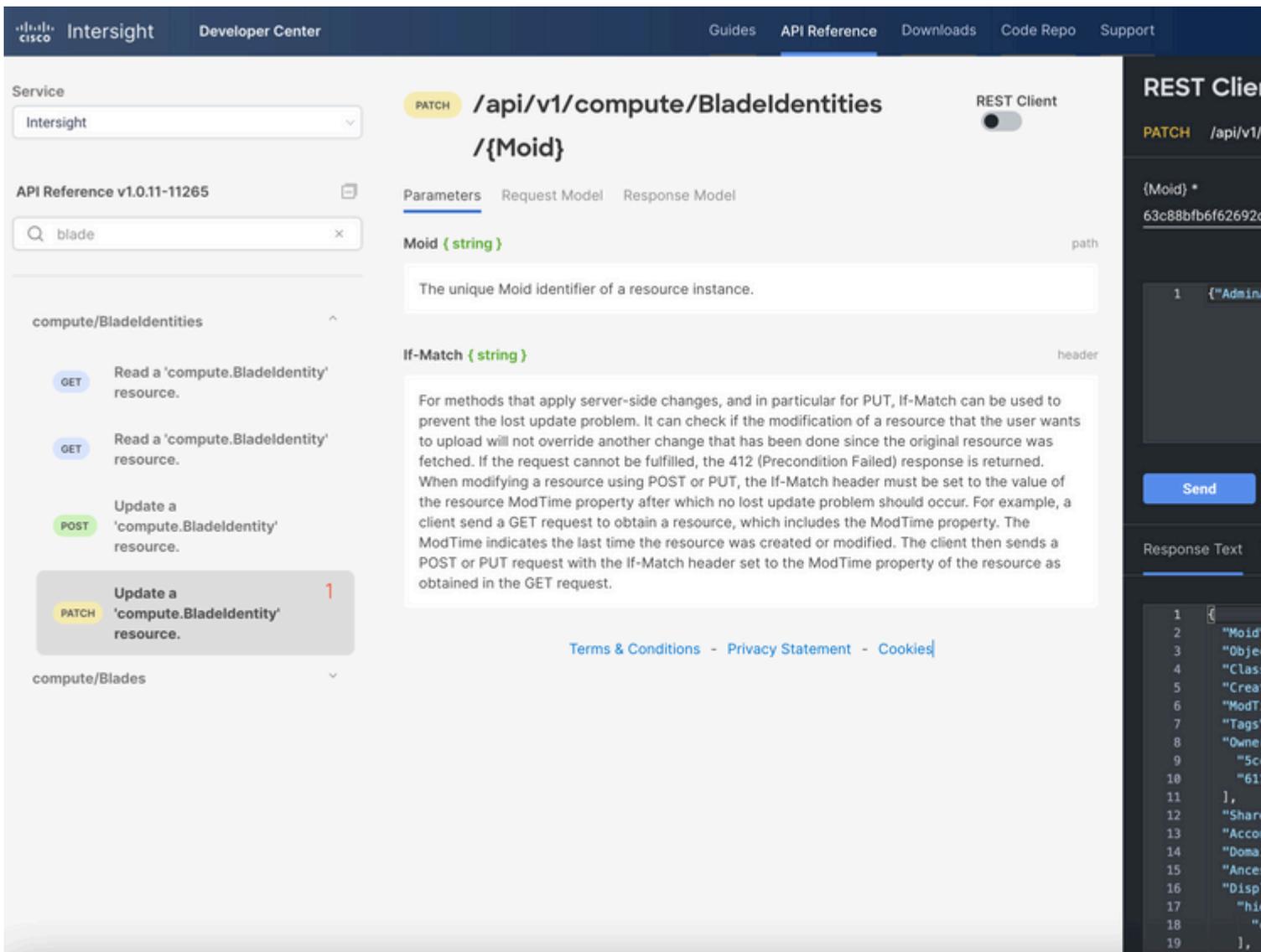
The screenshot shows the Cisco Intersight Developer Center API Reference page for the endpoint `/api/v1/compute/BladeIdentities`. The page is divided into several sections:

- Service:** Intersight
- API Reference v1.0.11-11265:** Search for "blade" (indicated by a red '1').
- compute/BladeIdentities:** List of API methods (indicated by a red '2'). The PATCH method is highlighted.
- GET /api/v1/compute/BladeIdentities:** Details for the GET method, including a description of the `$filter` query parameter and its usage.
- REST Client:** Panel on the right showing the response body for the GET request, including a "Send" button and a "Response Text" tab.

Apply the **PATCH** call with the action required. This example uses:

```
{"AdminAction": "Remove"}
```

Warning: This request results in the removal of the server from the Inventory. To add the server back into the inventory of the domain, a new discovery is required. This can be triggered through a physical reset of the server or by a chassis rediscovery task.



Troubleshoot Actions Through API Explorer in the Device Console

The Device Console allows you to monitor the health of your devices, and the status of their connection to Intersight. You can generate Tech Support bundles that contain diagnostic information to troubleshoot and analyze issues. In addition, the device console includes the ability to launch the API Explorer to perform Redfish-based operations on servers.

In the event that connectivity is lost between Intersight or the Appliance, the API Explorer in the Device Console can also be used to perform some basic troubleshoot actions.

1) Open the Device console, navigate to one of the Fabric Interconnect IP addresses, and select the Inventory tab.

2) Navigate to the specific device that needs to be troubleshoot, select the three dots to the right of it and select Launch API Explorer. The API Explorer is launched only for that device, and no others.

Reboot CIMC Management Controller of a Server

Launch the API Explorer for the Server:

The screenshot shows the Cisco Device Console interface for device F340-24-21-IMM-1. The 'INVENTORY' tab is selected, and the 'Servers' sub-tab is active. A table displays the following data:

Name	Health	Status	PID	Serial
F340-24-21-IMM-1-1-3	Healthy	Active	UCSB-B480-M5	FLM224
F340-24-21-IMM-1-1-5	Healthy	Active	UCSB-B200-M5	FCH214
F340-24-21-IMM-1-1-6	Healthy	Active	UCSB-B200-M5	FLM233
F340-24-21-IMM-1-2-1	Healthy	Decommissioned	UCSB-B200-M6	FCH243
F340-24-21-IMM-1-2-5	Healthy	Active	UCSB-B200-M5	FCH224
F340-24-21-IMM-1-3-1	Healthy	Active	UCSX-210C-M6	FCH251
F340-24-21-IMM-1-3-3	Healthy	Active	UCSX-210C-M6	FCH244

Type `CIMC` in `{ManagerID}` and apply a **POST** `Managers/{ManagerId}/Actions/Manager.Reset` call and add the type of reset.

This example uses:

```
{"ResetType": "ForceRestart"}
```

The screenshot shows the Cisco API Explorer interface for device F340-24-21-IMM-1-1-3 (Server). The 'POST' method is selected for the endpoint `/redfish/v1/Managers/{ManagerId}/Actions/Manager.Reset`. The 'Parameters' tab is active, showing a parameter `ManagerId { string }` with a description: 'The value of the Id property of the Manager resource'. The 'REST Client' tab is also visible, showing the request body: `{ "Target": ["/redfish/v1/M`.

Reboot an I/O Module (IOM)

Launch the API Explorer of the IOM:

Name	ID	Status	Model
F340-24-21-IMM-1-1	chassis-1	Active	UCSB-5108-AC2
F340-24-21-IMM-1-2	chassis-2	Active	N20-C6508
F340-24-21-IMM-1-3	chassis-3	Active	UCSX-9508

Type CMC in {ManagerID} and apply a **POST** *Managers/{ManagerId}/Actions/Manager.Reset* call and add the Reset Type. This example uses:

```
{"ResetType": "ForceRestart"}
```

API Explorer interface showing the REST Client for the endpoint `/redfish/v1/Managers/{ManagerId}/Actions/Manager.Reset`. The REST Client panel shows the request body:

```
1 {"ResetType": "ForceRestart"}
```

To reboot a peer IOM in the {ManagerID} field, enter PeerCMC with the same call as before.

CISCO API EXPLORER IoCard-1-1 (Chassis) Guides API Reference

API Reference v2019.2

POST /redfish/v1/Managers/{ManagerId}/Actions/Manager.Reset REST Client

Parameters Request Model Response Model

Managers

- GET Managers
- GET Managers/{ManagerId}
- PATCH Managers/{ManagerId}
- PUT Managers/{ManagerId}
- POST Managers/{ManagerId}/Actions/Manager.ForceFailover
- POST Managers/{ManagerId}/Actions/Manager.ModifyRedundancySet
- POST Managers/{ManagerId}/Actions/Manager.Reset

error: object

- @Message.ExtendedInfo: object *An array of messages describing one or more error messages.*
- Message: string (Read Only) *The human-readable message, if provided.*
- MessageArgs: array (Read Only) *This array of message arguments are substituted for the arguments in the message when looked up in the Message Registry.*
- Object: object

REST Client

POST /redfish/v1/Managers/{ManagerId}/Actions/Manager.Reset

{ManagerId} *

PeerCMC

```
1 {"ResetType": "ForceRestart"}
```

Related Information

[Intersight API Overview](#)

[Device Console Overview](#)