



Overview

API Structure

This document explains the application programming interface (API) and all of its properties that can be used to maintain and operate Cisco UCS Invicta Series Solid State Systems.

UCS Invicta Series Solid State Systems API was designed and implemented as RESTful Web Service. All API methods are accessible over HTTPS request using the URL that consists of the following parts:

- The domain: The IP of the Cisco UCS Invicta Series Solid State Systems
- The URI: /restapi/<API_version>/<token>/<methodName>/<URI_reminder>

A URI reminder is not always required but is essential to any methods that are requested over PUT or DELETE.

API Versions

The <API_version> in the URI (Example: 16) is important for consistent API support and compatibility. As the future enhancements and updates to the API methods are implemented, support for earlier versions is accessible. Each API method will preserve its signature as well as the response data structure under the version it was released, while all enhancements and changes is available under a new API version number.

The <token> URI element is included as a security measure to provide basic level of protection against unauthorized API use. See authenticate method section for all details.

JSON

Required data envelope on input (for POST and PUT request raw data) is JSON object notation labeled 'data'.

The response object structure consists of envelope_version, status, message and data. Please note that the data portion could be empty if there is no data to return.

Here is a sample output of the raw JSON string following an API method call to find:

```
{"status":1,"message":"OK","data":{"unitName":"devMachine","apiVersions":["15","16"]},"envelope_version":"15"}
```

cURL Examples

For cURL usage, please follow the examples below. Note that the `<ip_address>` and `<token>` entities should be replaced with the actual IP address and token, respectively.

- **GET**

```
curl -k -X GET --url https://<ip_address>/restapi/16/<token>/raidshealth
-H "Accept: application/json"
```

- **POST**

```
curl -X POST -d data='{ "name": "lun1", "vg": "ssd1", "size": 2, "strip-
ing": true }' -k --url https://<ip_address>/restapi/16/<token>/lun -H
"Accept: application/json"
```

- **PUT**

```
curl -X PUT -d data='{ "size": 3 }' -k --url https://<ip_ad-
dress>/restapi/16/<token>/lun/lun1 -H "Accept: application/json"
```

- **DELETE**

```
curl -X DELETE -d data='{ "name": "bond0:1" }' -k --url https://<ip_ad-
dress>/restapi/16/<token>/virtualinterface/0 -H "Accept: applica-
tion/json"
```

API Access/Permissions

A user account must be assigned to a role that has been granted API access permission. Permissions for individual API methods are dependent and directly related to a user's assigned roles and its associated permissions in the UI. For example, if a user is not assigned to a role that has permission to access the LUNs settings page in the UI, that user will also not have access to any of the LUN settings that are related to the LUN setting API methods. Similarly, if a user was assigned a role in the UI that has permission to add and edit LUNs but does not have LUN delete permission, the same restriction will apply in the API.

Supported HTTP Methods

Version	GET /list	POST	PUT	DELETE
1.5	X	X	X	X
1.6	X	X	X	X

Supported Incoming Data Formats

Version	JSON	XML
1.5	X	
1.6	X	