Intellian Intelligent Mediator 8 Type 2

Web Service Specification

Disclaimer

The information in this user guide is subject to change without prior notice through a product life cycle. A printed version of the user guide is periodically updated and may contain inaccuracies or omissions compared to the recent product information. The most up-to-date information is available on our website at https://www.intelliantech

Revision History

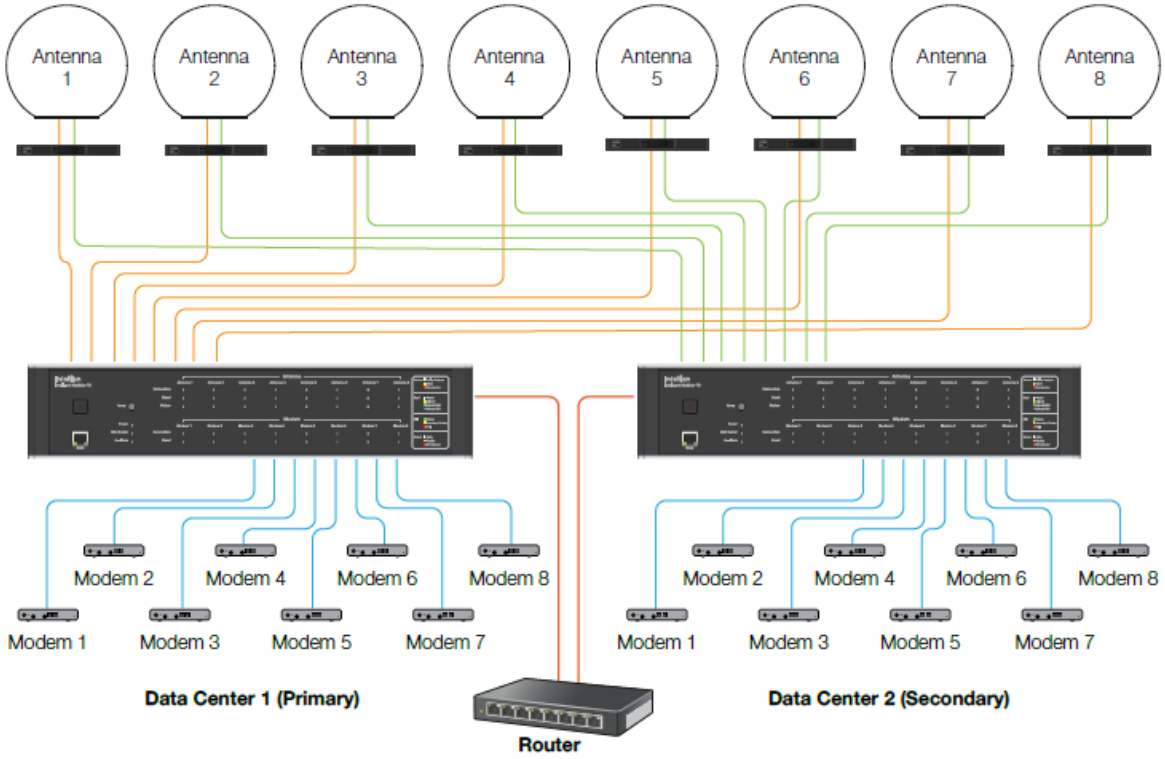
Revision	Date	Changes
1.0	Jun 23 2023	Initial Draft Release.
1.1	Apr 12 2024	Update
1.2	Apr 16 2025	1. Based on SW version : v9.3.4/v1.3.4 2. Update all APIs.

1. Introduction to the Intellian Web Service Interface

This chapter contains the following sections:

- Brief Introduction to Intellian Web Service Interface
- Locating the Web Service Endpoint

The Intellian Mediator provides a RESTful interface over HTTP/HTTPS. This interface abstracts many of the complexities of the Intellian Antenna control system, and presents a straightforward interface



[System Configuration]

1.1. Brief Introduction to Intellian Web Service Interface

The purpose of Intellian Web Service Interface is to provide a set of web service based APIs, which third party integrators can use to integrade Intellian Antenna Control System into an existing Network Management System for various purposes. Through the web services, users can monitor and control various aspects of Intellian Antenna Control System.

The web service interface is meant for machine to machine connections, and is not meant for direct use by human personnels through web browser except for occational debugging.

The web service interface is designed following RESTful principles, using HTTP methods as verbs.

Status Query is performed using HTTP GET method. Controlling is performed using HTTP PUT and POST methods. Most resources are described using Javascript Object Notation (JSON) encoding except some special occasions.

1.2. Locating the Web Service Endpoint

The Web Service endpoint will be running on the HTTP(port 80) and /HTTPS(port 443) ports on a well-known IP address assigned to Intellian Mediator.

1.3. Abbreviations and Acronyms

- ACU Antenna Control Unit
- ADU Above Deck Unit
- BDU Below Deck Unit
- BUC Block Upconverter
- LNB Low Noise Block Converter
- NBD Narrow Band Detector
- ACS Antenna Control System
- GPS Global Positioning System
- ICD Interface Control Document
- IP Internet Protocol
- kHz kiloHertz
- LAN Local Area Network
- MHz MegaHertz
- NMS Network Management System (Pulse or iVantage)
- OpenAMIP Open Antenna Modem Interface Protocol
- OpenBMIP Open BUC Modem Interface Protocol
- RF Radio Frequency
- TCP Transmission Control Protocol
- LEO — Low Earth Orbit
- MEO — Medium Earth Orbit
- GEO/GSO — Geosynchronous Equatorial and Geostationary Orbits
- LHCP Left hand circular polarization
- RHCP Right hand circular polarization

1.4. Related Documentation

2. Request Format

This chapter contains the following sections:

- Authentication
- Request Verbs
- Request Headers

The Web Service Interface expects REST requests in the following format.

<verb> /api/<resource_path> HTTP/1.1
<header1>
<header2>
...

2.1. Authentication

Clients authenticate to the Intellian Web Service using X-Auth-Token of API Key type. This X-Auth-Token should be passed in every request to the web service in the Authorization part of your HTTP(S) header

To send a request to Intellian web services, the client should perform the following steps to be authenticated

a. You have to your creadentials to request a token: Userid, Password (Refer to User login)

b. You can get “permission” and “token”.

c. Add this token to the request.

d. The authorization method and a space, i.e, “X-Auth-Token” is then put before the encoded string (X-Auth-Token 21d7d36217……).

This header should be sent in every request. A request without the header, or with an incorrect token, will be rejected.

2.2. Request Verbs

The table below shows all the REST verbs supported by the Intellian web services. Not every verb is applicable to every resource.

Name	Description
GET	Used to read/query a resource. Individual resources may be queried by specifying its unique id, or multiple instances may be queried using complex queries specified in the URI.
POST	Used with APIs involving a object or file. For exmaple, to uprade Antenna F/W, client may use antenna firmware upgrade API using POST and sending the upgrade package as a payload of POST in binary/octet stream.

PUT	Not allowed
DELETE	Used to delete a resource

2.3. Request Headers

The table below contains all the headers used in Intellian Web Services. Currently only headers regarding content body type are used.

Name	Description	Example
Content-type	In request, specifies the MIME type of the request body. Applicable to POST. In responses, specifies the MIME type of the response body.	application/json application/octet-stream depending on each service call text/plain application/json depending on each service call

3. Response Format

3.1. Response Status Codes

The Intellian Web Service can automatically respond with one of the following status codes. The status codes are divided into three categories:

- 2xx – success codes
- 4xx – errors on the client side (such as an invalid request)
- 5xx – errors on the server side

The table below lists all the currently used response codes.

Status Code	Description
200	Operation Success The request operation has succeeded. The exact success message depends on each request.
400	Bad Request There is a problem with request, such as invalid header field or missing parameter in JSON body.
401	Not Authenticated Authorization has failed because either Authorization header is missing or a given ID/password pair is not valid.
404	Not Found There is no matching service end porint for the given Request URI.
405	Not Allowed on the resource The given HTTP method is not allowed on the given request URI.
411	Invalid Request Necessary HTTP header field, such as content-length, is missing.
500	Server Error Intellian ACU server has encountered an error while processing request. The exact error text depends on each request

3.2. Response Body

Response body of a response to a request is specified in ***Content-type*** HTTP response header. The Content-type of a response to a request depends on each service being requested. Intellian Web Service uses two types of Content-type for responses.

- application/json
- text/plain

Most simple Antenna control & monitoring services adopts application/json to return control status or requested status data. Some serious Antenna control services like firmware upgrade use text/plain to return operation log in response body.

4. Common API

4.1. User Account

Method	URL	Descriptions
POST	/api/v1/auth/user/login	User login
GET	/api/v1/auth/user/logout	User logout
POST	/api/v1/auth/user/account/password	Change user password
POST	/api/v1/auth/user/account/reset	Reset user password
GET	/api/v1/auth/user/session_timeout	Get session timeout(in Minutes)
POST	/api/v1/auth/user/session_timeout	Set session timeout(in Minutes)

4.1.1. POST /api/v1/auth/user/login

Description

This API is used to login the Intellian Mediator system. User needs to specify parameters.

Request Field

Key	Type	Descriptions
id	String	User ID
pw	String	User Password
force	Boolean	Force logout of existing login account

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/auth/user/login

```
{
    "id": "intellian",
    "pw": "intellian1@1",
    "force": true
}
```

Response Field

Key / Field	Type	Descriptions
permission	String	User permission
token	String	Session token

Response Example

```
{
    "payload": {
        "permission": "ADMIN",
```

```
        "token": "cd8e434b1e6aa0246b4f846b12a8742e751831fc68fe1d08bbdcef5e35cd5d9bacda662435a8585d3355b66e98e749fb674815773ff9e4f7c50ddcdb8a165ac7"
    },
    "status": 1
}
```


4.1.2. GET /api/v1/auth/user/logout

Description

This API is used to logout the Intellian Mediator system. User needs to provide it as a bearer token in the Authorization part of your HTTP(S) header.

Request Field

none

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/auth/user/logout

Response Field

none

Response Example


```
{
    "payload": "success",
    "status": 1
}
```


4.1.3. POST /api/v1/auth/user/account/password

Description

This API is used to change password. User needs to specify parameters.

Request Field

Key / Field	Type	Descriptions
oldPassword	String	Current User Password
newPassword	String	New User Password

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/auth/user/account/password


```
{
    "oldPassword": "12345678",
    "newPassword": "intellian123"
```

```
}
```

Response Field

Key / Field	Type	Descriptions
oldPassword	String	Old User Password
newPassword	String	New User Password

Response Example

```
{
    "payload": {
        "new_pwd": "intellian123",
        "old_pwd": "12345678"
    },
    "status": 1
}
```

4.1.4. POST / api/v1/auth/user/account/reset

Description

This API is used to reset password to dafault. User needs to specify parameters.

Request Field

Key / Field	Type	Descriptions
userId	String	Account to reset

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/auth/user/account/reset

```
{
    "userId": "intellian"
}
```

Response Field

Key	Type	Descriptions
userId	String	Account to reset

Response Example

```
{
    "payload": {
```

```
        "userId": "intellian"
    },
    "status": 1
}
```


4.1.5. GET /api/v1/auth/user/session_timeout

Description

This API is used to get session timeout.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/auth/user/session_timeout

Response Field

Key	Type	Descriptions
timeout	Integer	Session timeout(in Minutes)

Response Example


```
{
    "payload": {
        "timeout": 60
    },
    "status": 1
}
```


4.1.6. POST /api/v1/auth/user/session_timeout

Description

This API is used to set session timeout.

Request Field

Key / Field	Type	Descriptions
timeout	Integer	Session timeout(in Minutes)

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/auth/user/session_timeout


```
{
    "timeout": 60
```

```
}
```

Response Field

Key	Type	Descriptions
timeout	Integer	Session timeout(in Minutes)

Response Example

```
{
    "payload": {
        "timeout": 60
    },
    "status": 1
}
```

4.2. System

Method	URL	Descriptions
GET	/api/v1/system/config/backup	Download mediator config backup
POST	/api/v1/system/config/restore	Restores mediator settings from backup file
POST	/api/v1/system/reboot	Reboot system
GET	/api/v1/status/system	Get status of system
GET	/api/v1/system/log/download	Download log
GET	/api/v1/library/satellite/list	Get satellite library list
GET	/api/v1/library/satellite/item/{index}	Get satellite library item
POST	/api/v1/library/satellite/item/{index}	Add/Edit satellite library item
DELETE	/api/v1/library/satellite/item/{index}	Delete satellite library item
GET	/api/v1/library/satellite/list/download	Download satellite library list
POST	/api/v1/library/satellite/list/upload	Upload satellite library list

4.2.1. GET /api/v1/system/config/backup

Description

This API is used to download mediator config backup files.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/system/config/backup

Response Field

Key	Type	Descriptions
path	String	Path of mediator config backup file to download

Response Example


```
{
    "payload": {
        "path": "/usr/local/download/backup/Mediator_Backup_M4-TB08M_MMD23040001_2025-04-08-072353.conf"
    },
    "status": 1
}
```


4.2.2. POST /api/v1/system/config/restore

Description

This API is used to restores the mediator settings using the setting files saved.

Request Field

Content-type : multipart/form-data

Key / Field	Type	Descriptions
file	String	Config backup file to be restored

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/system/config/restore

```
------WebKitFormBoundaryNepMTRfGBCMulf3W
Content-Disposition: form-data; name="file"; filename="Mediator_Backup_M4-TB08M_MMD23040001_2025-04-08-072353.conf"
Content-Type: application/octet-stream


------WebKitFormBoundaryNepMTRfGBCMulf3W--
```

Response Field

Key / Field	Type	Descriptions
payload	String	Response message
status	Integer	Response status code [0:Error, 1:Success]

Response Example


```
{
    "payload": "success",
    "status": 1
}
```


4.2.3. POST /api/v1/system/reboot

Description

This API is used to reboot system.

Request Field

Key / Field	Type	Descriptions
reset	Boolean	Reset or not [False: None, True: Reset]

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/system/reboot


```
{
    "reset": true
}
```


Response Field

Key	Type	Descriptions
reset	Boolean	Reset or not [False: None, True: Reset]

Response Example


```
{
    "payload": {
        "reset": true
    },
    "status": 1
}
```


4.2.4. GET /api/v1/status/system

Description

This API is used to get status of system.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/status/system

Response Field

Key	Type	Descriptions
currentAcuVersion	String	Current SW version
name	String	Model name
serialNumber	String	Serial number

Response Example

```
{
    "payload": {
        "currentAcuVersion": "v9.3.5",
        "name": "M4-TB08M",
        "serialNumber": "MMD23040001"
    },
    "status": 1
}
```

4.2.5. GET /api/v1/system/log/download

Description

This API is used to download logs, perform GET request to the URL according to the response path to download

URL : {http|https}://{ServerIP}:{port}/{path}

Request Field

Key / Field	Type	Descriptions
start_time	String	Start date of log to download
end_time	String	End date of log to download
includeAntBackup	Boolean	If true, download the antenna config files

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/system/log/download

```
{
    "start_time": "2025-04-07",
    "end_time": "2025-04-09",
    "includeAntBackup": false
```

```
}
```

Response Field

Key	Type	Descriptions
path	String	Download path

Response Example

```
{
    "payload": {
        "path": "/usr/local/download/backup/Mediator_M4_MMD23040001_log_2025-04-07_2025-04-09.tar"
    },
    "status": 1
}
```

4.2.6. GET /api/v1/library/satellite/list

Description

This API is used to get satellite library list.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/library/satellite/list

Response Field

Key	Type	Descriptions
date	String	Satellite library last modified date
library	Object Array	Satellite library
numSatellite	Integer	The number of satellite information in the Satellite library
version	String	Satellite library version

library Object

Key	Type	Descriptions
band	Integer	Satellite band [0:Ku, 1:C, 2:Ka, 3:X]
dvbFreq	Integer	DVB frequency (MHz)
dvbNid	Integer	DVB network id
dvbSymbol	Integer	DVB symbol rate (KSps)
dvbVerifyType	Integer	Satellite verification type [0:SIGNAL, 1:DVB_LOCK, 2:DVB_NID, 5:NBD]
identify	Integer	Satellite identifier [0:NBD, 1:DVB]

localFreq	Integer	Rx local frequency (MHz)
longitude	Number	Satellite longitude
nbdBandwidth	Integer	NBD bandwidth (kHz)
nbdFreq	Integer	NBD IF frequency (kHz)
rxPol	Integer	Rx polarization type [0:VERTICAL, 1:HORIZONTAL, 2:RHCP, 3:LHCP]
satName	String	Satellite name
skewOffset	Number	Satellite skew offset
txLocal	Integer	Tx local frequency (MHz)
txPol	Integer	Tx polarization type [0:VERTICAL, 1:HORIZONTAL, 2:RHCP, 3:LHCP]

Response Example


```
{
    "payload": {
        "date": "2024-06-17",
        "library": [
          {
            "band": 1,
            "dvbFreq": 100000,
            "dvbNid": 124,
            "dvbSymbol": 27500,
            "dvbVerifyType": 0,
            "identify": 0,
            "inclinedAngle": 0,
            "localFreq": 5150,
            "longitude": 115.5,
            "nbdBandwidth": 2000,
            "nbdFreq": 1070000,
            "rxPol": 1,
            "satName": "CHINA6_H",
            "skewOffset": 0.0,
            "tle": ["", ""],
            "txLocal": 0,
            "txPol": 0
          },
          {
            "band": 1,
            "dvbFreq": 100000,
            "dvbNid": 124,
            "dvbSymbol": 27500,
            "dvbVerifyType": 0,
            "identify": 0,
            "inclinedAngle": 0,
            "localFreq": 5150,
            "longitude": 115.5,
            "nbdBandwidth": 2000,
            "nbdFreq": 1070000,
            "rxPol": 0,
            "satName": "CHINA6_V",
```

```
            "skewOffset": 0.0,
            "tle": ["", ""],
            "txLocal": 0,
            "txPol": 1
          }
        ],
        "numSatellite": 2,
        "version": "00.01"
    },
    "status": 1
}
```


4.2.7. GET /api/v1/library/satellite/item/{index}

Description

This API is used to get satellite library item, {index} : satellite library array index, starting at 1.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/library/satellite/item/{index}

Response Field

Key	Type	Descriptions
library	Object Array	Satellite library

library Object

Key	Type	Descriptions
band	Integer	Satellite band [0:Ku, 1:C, 2:Ka, 3:X]
dvbFreq	Integer	DVB frequency (MHz)
dvbNid	Integer	DVB network id
dvbSymbol	Integer	DVB symbol rate (KSps)
dvbVerifyType	Integer	Satellite verification type [0:SIGNAL, 1:DVB_LOCK, 2:DVB_NID, 5:NBD]
identify	Integer	Satellite identifier [0:NBD, 1:DVB]
localFreq	Integer	Rx local frequency (MHz)
longitude	Number	Satellite longitude
nbdBandwidth	Integer	NBD bandwidth (kHz)
nbdFreq	Integer	NBD IF frequency (kHz)
rxPol	Integer	Rx polarization type [0:VERTICAL, 1:HORIZONTAL, 2:RHCP, 3:LHCP]
satName	String	Satellite name
skewOffset	Number	Satellite skew offset
txLocal	Integer	Tx local frequency (MHz)
txPol	Integer	Tx polarization type

		[0:VERTICAL, 1:HORIZONTAL, 2:RHCP, 3:LHCP]

Response Example


```
{
    "payload": {
        "library": [
          {
            "band": 1,
            "dvbFreq": 100000,
            "dvbNid": 124,
            "dvbSymbol": 27500,
            "dvbVerifyType": 0,
            "identify": 0,
            "inclinedAngle": 0,
            "localFreq": 5150,
            "longitude": 115.5,
            "nbdBandwidth": 2000,
            "nbdFreq": 1070000,
            "rxPol": 1,
            "satName": "CHINA6_H",
            "skewOffset": 0.0,
            "tle": ["", ""],
            "txLocal": 0,
            "txPol": 0
          }
        ]
    },
    "status": 1
}
```


4.2.8. POST /api/v1/library/satellite/item/{index}

Description

This API is used to set satellite library item, {index} : satellite library array index, starting at 1.

Request Field

Key / Field	Type	Descriptions
index	Integer	Satellite index in library
band	Integer	Satellite band [0:Ku, 1:C, 2:Ka, 3:X]
dvbFreq	Integer	DVB frequency (MHz)
dvbNid	Integer	DVB network id
dvbSymbol	Integer	DVB symbol rate (KSps)
dvbVerifyType	Integer	Satellite verification type [0:SIGNAL, 1:DVB_LOCK, 2:DVB_NID, 5:NBD]
identify	Integer	Satellite identifier [0:NBD, 1:DVB]

localFreq	Integer	Rx local frequency (MHz)
longitude	Number	Satellite longitude
nbdBandwidth	Integer	NBD bandwidth (kHz)
nbdFreq	Integer	NBD IF frequency (kHz)
rxPol	Integer	Rx polarization type [0:VERTICAL, 1:HORIZONTAL, 2:RHCP, 3:LHCP]
satName	String	Satellite name
skewOffset	Number	Satellite skew offset
txLocal	Integer	Tx local frequency (MHz)
txPol	Integer	Tx polarization type [0:VERTICAL, 1:HORIZONTAL, 2:RHCP, 3:LHCP]
LNG_EW	Integer	0: east 1: west
inclinedAngle	Integer	Angle for calculating the differences between actual target and obsolete satellite.

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/library/satellite/item/{index}


```
{
    "index": 29,
    "satName": "CHINA6_H",
    "band": 1,
    "longitude": 115.5,
    "LNG_EW": 0,
    "localFreq": 5150,
    "skewOffset": 0.0,
    "rxPol": 1,
    "txPol": 0
    "txLocal": 0,
    "inclinedAngle": 0,
    "identify": 0,
    "nbdFreq": 1070000,
    "nbdBandwidth": 2000,
    "dvbVerifyType": 0,
    "dvbFreq": 100000,
    "dvbSymbol": 27500,
    "dvbNid": 124
}
```


Response Field

Key	Type	Descriptions
library	Object Array	Satellite library

library Object

Key	Type	Descriptions

band	Integer	Satellite band [0:Ku, 1:C, 2:Ka, 3:X]
dvbFreq	Integer	DVB frequency (MHz)
dvbNid	Integer	DVB network id
dvbSymbol	Integer	DVB symbol rate (KSps)
dvbVerifyType	Integer	Satellite verification type [0:SIGNAL, 1:DVB_LOCK, 2:DVB_NID, 5:NBD]
identify	Integer	Satellite identifier [0:NBD, 1:DVB]
localFreq	Integer	Rx local frequency (MHz)
longitude	Number	Satellite longitude
nbdBandwidth	Integer	NBD bandwidth (kHz)
nbdFreq	Integer	NBD IF frequency (kHz)
rxPol	Integer	Rx polarization type [0:VERTICAL, 1:HORIZONTAL, 2:RHCP, 3:LHCP]
satName	String	Satellite name
skewOffset	Number	Satellite skew offset
txLocal	Integer	Tx local frequency (MHz)
txPol	Integer	Tx polarization type [0:VERTICAL, 1:HORIZONTAL, 2:RHCP, 3:LHCP]

Response Example


```
{
    "payload": {
        "date": "2024-06-17",
        "library": [
          {
            "band": 1,
            "dvbFreq": 100000,
            "dvbNid": 124,
            "dvbSymbol": 27500,
            "dvbVerifyType": 0,
            "identify": 0,
            "inclinedAngle": 0,
            "localFreq": 5150,
            "longitude": 115.5,
            "nbdBandwidth": 2000,
            "nbdFreq": 1070000,
            "rxPol": 1,
            "satName": "CHINA6_H",
            "skewOffset": 0.0,
            "tle": ["", ""],
            "txLocal": 0,
            "txPol": 0
          },
          {
            "band": 1,
            "dvbFreq": 100000,
            "dvbNid": 124,
            "dvbSymbol": 27500,
            "dvbVerifyType": 0,
```

```
            "identify": 0,
            "inclinedAngle": 0,
            "localFreq": 5150,
            "longitude": 115.5,
            "nbdBandwidth": 2000,
            "nbdFreq": 1070000,
            "rxPol": 0,
            "satName": "CHINA6_V",
            "skewOffset": 0.0,
            "tle": ["", ""],
            "txLocal": 0,
            "txPol": 1
          }
        ],
        "numSatellite": 2,
        "version": "00.01"
    },
    "status": 1
}
```


4.2.9. DELETE /api/v1/library/satellite/item/{index}

Description

This API is used to delete satellite library item, {index} : satellite library array index, starting at 1.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/library/satellite/item/{index}

Response Field

Key	Type	Descriptions
library	Object Array	Satellite library

library Object

Key	Type	Descriptions
band	Integer	Satellite band [0:Ku, 1:C, 2:Ka, 3:X]
dvbFreq	Integer	DVB frequency (MHz)
dvbNid	Integer	DVB network id
dvbSymbol	Integer	DVB symbol rate (KSps)
dvbVerifyType	Integer	Satellite verification type [0:SIGNAL, 1:DVB_LOCK, 2:DVB_NID, 5:NBD]
identify	Integer	Satellite identifier [0:NBD, 1:DVB]
localFreq	Integer	Rx local frequency (MHz)

longitude	Number	Satellite longitude
nbdBandwidth	Integer	NBD bandwidth (kHz)
nbdFreq	Integer	NBD IF frequency (kHz)
rxPol	Integer	Rx polarization type [0:VERTICAL, 1:HORIZONTAL, 2:RHCP, 3:LHCP]
satName	String	Satellite name
skewOffset	Number	Satellite skew offset
txLocal	Integer	Tx local frequency (MHz)
txPol	Integer	Tx polarization type [0:VERTICAL, 1:HORIZONTAL, 2:RHCP, 3:LHCP]

Response Example


```
{
    "payload": {
        "date": "2024-06-17",
        "library": [
          {
            "band": 1,
            "dvbFreq": 100000,
            "dvbNid": 124,
            "dvbSymbol": 27500,
            "dvbVerifyType": 0,
            "identify": 0,
            "inclinedAngle": 0,
            "localFreq": 5150,
            "longitude": 115.5,
            "nbdBandwidth": 2000,
            "nbdFreq": 1070000,
            "rxPol": 1,
            "satName": "CHINA6_H",
            "skewOffset": 0.0,
            "tle": ["", ""],
            "txLocal": 0,
            "txPol": 0
          }
        ],
        "numSatellite": 1,
        "version": "00.01"
    },
    "status": 1
}
```


4.2.10. GET /api/v1/library/satellite/list/download

Description

This API is used to download satellite library list.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/library/satellite/list/download

Response Field

Key	Type	Descriptions
path	String	Path of satellite library file to download

Response Example

```
{
    "payload": {
        "path": "/usr/local/download/backup/SatelliteLibrary_backup_2025-04-09-
050434.isl"
    },
    "status": 1
}
```

4.2.11. POST /api/v1/library/satellite/list/upload

Description

This API is used to upload satellite library list.

Request Field

Content-type : multipart/form-data

Key / Field	Type	Descriptions
filename	String	Satellite library file to be uploaded

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/library/satellite/list/upload

```
------WebKitFormBoundarykatHsr7NVDJJJ6zA
Content-Disposition: form-data; name="filename"; filename="SatelliteLibrary_backup_2025-
04-09-050434.isl"
Content-Type: application/octet-stream


------WebKitFormBoundarykatHsr7NVDJJJ6zA--
```

Response Field

Key	Type	Descriptions
date	String	Satellite library last modified date
library	Object Array	Satellite library
numSatellite	Integer	The number of satellite information in the Satellite library
version	String	Satellite library version

library Object

Key	Type	Descriptions
band	Integer	Satellite band [0:Ku, 1:C, 2:Ka, 3:X]
dvbFreq	Integer	DVB frequency (MHz)
dvbNid	Integer	DVB network id
dvbSymbol	Integer	DVB symbol rate (KSps)
dvbVerifyType	Integer	Satellite verification type [0:SIGNAL, 1:DVB_LOCK, 2:DVB_NID, 5:NBD]
identify	Integer	Satellite identifier [0:NBD, 1:DVB]
localFreq	Integer	Rx local frequency (MHz)
longitude	Number	Satellite longitude
nbdBandwidth	Integer	NBD bandwidth (kHz)
nbdFreq	Integer	NBD IF frequency (kHz)
rxPol	Integer	Rx polarization type [0:VERTICAL, 1:HORIZONTAL, 2:RHCP, 3:LHCP]
satName	String	Satellite name
skewOffset	Number	Satellite skew offset
txLocal	Integer	Tx local frequency (MHz)
txPol	Integer	Tx polarization type [0:VERTICAL, 1:HORIZONTAL, 2:RHCP, 3:LHCP]

Response Example


```
{
    "payload": {
        "date": "2025-04-09",
        "library": [
          {
            "band": 1,
            "dvbFreq": 100000,
            "dvbNid": 124,
            "dvbSymbol": 27500,
            "dvbVerifyType": 0,
            "identify": 0,
            "inclinedAngle": 0,
            "localFreq": 5150,
            "longitude": 115.5,
            "nbdBandwidth": 2000,
            "nbdFreq": 1070000,
            "rxPol": 1,
            "satName": "CHINA6_H",
            "skewOffset": 0.0,
            "tle": ["", ""],
```

```
            "txLocal": 0,
            "txPol": 0
          },
          {
            "band": 1,
            "dvbFreq": 100000,
            "dvbNid": 124,
            "dvbSymbol": 27500,
            "dvbVerifyType": 0,
            "identify": 0,
            "inclinedAngle": 0,
            "localFreq": 5150,
            "longitude": 115.5,
            "nbdBandwidth": 2000,
            "nbdFreq": 1070000,
            "rxPol": 0,
            "satName": "CHINA6_V",
            "skewOffset": 0.0,
            "tle": ["", ""],
            "txLocal": 0,
            "txPol": 1
          }
        ],
        "numSatellite": 2,
        "version": "00.01"
    },
    "status": 1
}
```


5. Mediator API

5.1. Antenna

Method	URL	Descriptions
POST	/api/v1/asset/antenna/create	Create an Antenna
POST	/api/v1/asset/antenna/update	Update an Antenna
POST	/api/v1/asset/antenna/delete	Delete an Antenna
GET	/api/v1/asset/antenna/list	Getting information about all antennas
POST	/api/v1/asset/antenna/complex-blockages	Getting Complex blockage (Block1 Only)
POST	/api/v1/mediation/simultaneous_band/track	Setting Tracking band (Block1 Only)
GET	/api/v1/mediation/simultaneous_band/track	Getting Tracking band (Block1 Only)
POST	/api/v1/mediation/simultaneous_band/active	Setting Band activate (Block1 Only)
GET	/api/v1/mediation/simultaneous_band/active	Getting Band activate (Block1 Only)

5.1.1. POST /api/v1/asset/antenna/create

Description

This API is used to register new antenna which uses http/https protocol to communicate with Intellian mediator. Antenna port, id, name etc details should be specified for registering antenna.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port (1 ~ 8)
id	String	Asset ID
name	String	Antenna name (Up to 32 byte)
description	String	Antenna descriptions (Up to 200 byte)
enable	Integer	0: Disable 1: Enable
model	String	Model name of the antenna
modemCode	Integer	Model code of the antenna 0: Intellian antenna 5: V240MT 9: NX series 12: V240MT Gen 2 20: X Series 21: mPOWER Land Fixed 22: mPOWER NX 23: Block1 24: Block0 25: V240M1 26: V240M2
ip	String	IP address
ipPort	Integer	Intellian antenna port (it should be 4010)
supGeoBand	Integer	The bitmask of support GEO band [0:Ku, 1:C, 2:Ka, 3: X] * bitmask of band : Ku(0x01) | C(0x02) | Ka(0x04) | X(0x08)
supMeoBand	Integer	The bitmask of support MEO band [2:Ka] * bitmask of band : Ka(0x04)

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/antenna/create

```
{
    "connPort": 6,
    "id": "12345678",
    "name": "A6 G1",
    "description": "v240MT Gen I",
    "enable": 1,
    "model": "v240MT",
    "modemCode": 5,
    "ip": "172.22.1.39",
    "ipPort": 4010,
    "supGeoBand": 3,
    "supMeoBand": 4
```

```
}
```

Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port (1 ~ 8)
id	String	Asset ID
name	String	Antenna name (Up to 32 byte)
description	String	Antenna descriptions (Up to 200 byte)
enable	Integer	0: Disable 1: Enable
model	String	Model name of the antenna
modemCode	Integer	Model code of the antenna 0: Intellian antenna 5: V240MT 9: NX series 12: V240MT Gen 2 20: X Series 21: mPOWER Land Fixed 22: mPOWER NX 23: Block1 24: Block0 25: V240M1 26: V240M2
ip	String	IP address
ipPort	Integer	Intellian antenna port (it should be 4010)
supGeoBand	Integer	The bitmask of support GEO band [0:Ku, 1:C, 2:Ka, 3: X] * bitmask of band : Ku(0x01) | C(0x02) | Ka(0x04) | X(0x08)
supMeoBand	Integer	The bitmask of support MEO band [2:Ka] * bitmask of band : Ka(0x04)

Response Example


```
{
    "payload": {
        "connPort": 6,
        "id": "12345678",
        "name": "A6 G1",
        "description": "v240MT Gen I",
        "enable": 1,
        "model": "v240MT",
        "modemCode": 5,
        "ip": "172.22.1.39",
        "ipPort": 4010,
        "supGeoBand": 3,
        "supMeoBand": 4
    },
    "status": 1
}
```

5.1.2. POST /api/v1/asset/antenna/update

Description

This API is used to update information of registered antenna. Antenna ipaddress, name, model, support band and updated configuration parameters should be specified for updating existing configuration of antenna.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port (1 ~ 8)
id	String	Asset ID
name	String	Antenna name (Up to 32 byte)
description	String	Antenna descriptions (Up to 200 byte)
enable	Integer	0: Disable 1: Enable
model	String	Model name of the antenna
modemCode	Integer	Model code of the antenna 0: Intellian antenna 5: V240MT 9: NX series 12: V240MT Gen 2 20: X Series 21: mPOWER Land Fixed 22: mPOWER NX 23: Block1 24: Block0 25: V240M1 26: V240M2
ip	String	IP address
ipPort	Integer	Intellian antenna port (it should be 4010)
supGeoBand	Integer	The bitmask of support GEO band [0:Ku, 1:C, 2:Ka, 3: X] * bitmask of band : Ku(0x01) | C(0x02) | Ka(0x04) | X(0x8)
supMeoBand	Integer	The bitmask of support MEO band [2:Ka] * bitmask of band : Ka(0x04)

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/antenna/update

```
{
    "connPort": 6,
    "id": "12345678",
    "name": "A6 G1",
    "description": "v240MT Gen I",
    "enable": 0,
    "model": "v240MT",
    "modemCode": 5,
    "ip": "172.22.1.39",
    "ipPort": 0,
    "supGeoBand": 3,
```

```
    "supMeoBand": 4
}
```

Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port (1 ~ 8)
id	String	Asset ID
name	String	Antenna name (Up to 32 byte)
description	String	Antenna descriptions (Up to 200 byte)
enable	Integer	0: Disable 1: Enable
model	String	Model name of the antenna
modemCode	Integer	Model code of the antenna 0: Intellian antenna 5: V240MT 9: NX series 12: V240MT Gen 2 20: X Series 21: mPOWER Land Fixed 22: mPOWER NX 23: Block1 24: Block0 25: V240M1 26: V240M2
ip	String	IP address
ipPort	Integer	Intellian antenna port (it should be 4010)
supGeoBand	Integer	The bitmask of support GEO band [0:Ku, 1:C, 2:Ka, 3: X] * bitmask of band : Ku(0x01) | C(0x02) | Ka(0x04) | X(0x8)
supMeoBand	Integer	The bitmask of support MEO band [2:Ka] * bitmask of band : Ka(0x04)

Response Example

```
{
    "payload": {
        "connPort": 6,
        "id": "12345678",
        "name": "A6 G1",
        "description": "v240MT Gen I",
        "enable": 0,
        "model": "v240MT",
        "modemCode": 5,
        "ip": "172.22.1.39",
        "ipPort": 0,
        "supGeoBand": 3,
        "supMeoBand": 4
    },
    "status": 1
}
```

5.1.3. POST /api/v1/asset/antenna/delete

Description

This API is used to delete already registered antenna in Intellian Mediator. Registered Antenna connPort should be specified for deleting antenna.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port(1~8) want to delete

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/antenna/delete

```
{
    "connPort": 6
}
```

Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port(1~8)

Response Example

```
{
    "payload": {
        "connPort": 6"
    },
    "status": 1
}
```

5.1.4. GET /api/v1/asset/antenna/list

Description

This API is used to get the information about all antenna lists. General information about the antenna like IP Address, Version, Status etc information can be retrieved.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/antenna/list

Response Field

Key / Field	Type	Descriptions
acuConnectionState	Integer	Connection state between Mediator and ACU [0: Disconnected, 1: Connected]
avblBand	Integer	~~Available band, not used~~
avblGeoBand	Integer	The bitmask of available band for GEO [0:Ku, 1:C, 2:Ka, 3: X] * bitmask of band : Ku(0x01) | C(0x02) | Ka(0x04) | X(0x8)
avblMeoBand	Integer	The bitmask of available band for MEO [2:Ka] * bitmask of band : Ka(0x04)
avblOrbit	Integer	~~Available orbit,~~ not used
azAbs	Integer	Absolute azimuth angle of antenna
azRel	Integer	Relative azimuth angle of antenna
azTar	Integer	Target azimuth angle of antenna
blockages	Object array	Blockages list (Up to 20)
bowOffset	Integer	Offset of ship and antenna
buc	Object	Object for buc info
bucFault	Integer	0: Reset, 1: Set
buclp	Object String	BUC IP list [cBand, kuBand, kaBand, meoKaBand, xBand]
bucTxClock	Object Integer	BUC Reference Tx Clock [cBand, kuBand, kaBand, meoKaBand, xBand] [0: Off, 1: 10MHz, 2: 50MHz]
connPort	Integer	Port number of Mediator which the antenna is connected to(1~8)
connPort2	Integer	2nd Port number of Mediator which the antenna is connected to (1 ~ 8) (Block1 Only)
connectionState	Integer	Connection state between ACU and ANT [0: Disconnected, 1: Connected]
curBand	Integer	Current activated band [0:Ku, 1:C, 2:Ka, 3: X]
curOrbit	Integer	Current activated orbit [0:GEO, 1:MEO, 2:LEO]
demod	String	Demod status [lock|unlock/on|off-A|B], [fault] ex) “lock-on-A”
description	String	Description of antenna inforamtion
el	Integer	~~elevation angle of antenna, 0 ~ 90°~~ not used
elAbs	Integer	Elevation angle of antenna
elTar	Integer	Target elevation angle of antenna
enable	Integer	0: Disable 1: Enable
esno	Integer	Ratio of energy per symbol to noise power spectral density
ew	Integer	0: east 1: west

fault	Object array	Fault status of antenna
		{
		code
		message
		severity
		time
		}

heading	Integer	Ship’s heading information
id	String	Asset id

ifFreq	Integer	IF frequency (KHz)
inBlockage	Integer	Status of blocked
ip	String	Network IP information of the antenna
ipPort	Integer	Intellian antenna port (it should be 4010)
latitude	Integer	Latitude of the antenna position, 0 ~ 90 (degree)
longitude	Integer	Longitude of the antenna position, 0 ~ 180 (degree)
manufacturer	Integer	0: Intellian 1: Other vendor
model	String	Model name of the antenna
modelCode	Integer	Model code of antenna 0: Intellian antenna 5: V240MT 9: NX series 12: V240MT Gen 2 20: X Series 21: mPOWER Land Fixed 22: mPOWER NX 23: Block1 24: Block0 25: V240M1 26: V240M2
name	String	Name of the antenna registered by user
ns	Integer	2: south 3: north
operationState	Integer	The role of antenna 0: None 1: Primary 2: Secondary 3: Backup
pol	Integer	Pol(skew) angle, 0 ~ 90°
rfChain	Integer	Chain number of mPOWER modem to which the antenna is connected.
rxLocal	Integer	Rx Local frequency (MHz)
rxPol	Integer	Receive polarity 0:VERTICAL 1:HORIZONTAL 2:RHCP 3:LHCP
signalLevel	Integer	Antenna signal level
simultaneousBand	Object	Service band status information (Block1 Only) config(Object Array) : System activation status per band trackingBand : Current tracking band [0:Ku, 1:C, 2:Ka, 3: X]
supBand	Integer	~~Support band [1:Ka, 2:Ku, 4:C], not used~~
supGeoBand	Integer	The bitmask of support band for GEO [0:Ku, 1:C, 2:Ka, 3: X] * bitmask of band : Ku(0x01) | C(0x02) | Ka(0x04) | X(0x8)
supMeoBand	Integer	The bitmask of support band for MEO [2:Ka] * bitmask of band : Ka(0x04)
supOrbit	Integer	The bitmask of support orbit [0:GEO, 1:MEO, 2:LEO] * bitmask of orbit : GEO(0x01) | MEO(0x02) | LEO(0x04)
targetSatBand	Integer	Target band [0: Ku, 1: C, 2: Ka, 3: X]
targetSatLongitude	Integer	-180 ~ 180
targetSatName	String	Target satellite name
targetSatType	Integer	Target orbit

		[0: GEO, 1: MEO, 2 : LEO]
trackingStatus	Integer	Tracking status of antenna 0: INITIALIZE 1: UPGRADE 2: SETUP 3: DIAGNOSTIC 4: TEST 5: MANUAL 6: SEARCHING 7: TRACKING 8: DEMO 9: DFU 10: CALIBRATION 11: UNWRAP 12: ACU INITIALIZE 13: SLEEP 14: SEARCH 1 15: SEARCH 2 16: SEARCH 3 17: BLOCKZONE 18: COMMUNICATION ERROR 19: WAITING 20: STOP 21: STOW 22: PARK 23: STANDBY
txPol	Integer	transmit polarity 0:VERTICAL 1:HORIZONTAL 2:RHCP 3:LHCP

blockage object

Key / Field	Type	Descriptions
azEnd	Integer	Azimuth end of blockage
azStart	Integer	Azimuth start of blockage
el	Integer	Elevation angle of blockage
enable	Integer	Blockage control [0: Disable, 1: Enable]

buc object

BUC model	Key	Type	Descriptions
BUC Common	connPort	Integer	1~8
	satBand	Integer	0:BAND_NONE 1: KA 2: KU 4: C
	satOrbit	Integer	1: GEO 2: MEO 3: LEO
	model	Integer	0: General 1: NJRC

			2: COMTECH 3: TERRASAT 4: TELEDYNE 5: MMW (MISSION_MICROWAVE) 6: CPI (BCIP) 7: XCVR 8: REVGO 9 : MMW (BMIP) 10: LBAND_HPA 11: XCVR_Ku 12: REVGO_PRO 13: REVGO_HP 14: ADVANTECH

BUC model	Key	Type	Descriptions
0: General 1: NJRC 2: COMTECH 3: TERRASAT 4: TELEDYNE 5: MMW	txState	Integer	0: OFF 1: STANDBY 2: TRANSMIT
	attenuator	Integer	dBm
	rfOutput	Integer	power (dBm)
	ipAddress	String	IP of BUC
	temperature	Integer	℃

BUC model	Key	Type	Descriptions
6: CPI (BCIP)	txState	Integer	0: STANDBY 1: TRANSMIT 2: TRANSMITINHIBIT 3: WARMUP
	attenuator	Integer	dBm
	rfOutput	Integer	power (dBm)
	ipAddress	String	IP of BUC
	temperature	Integer	℃, (not support)
	controlMode	Integer	0: Local 1: Remote All 2: Remote Ethernet 3: Remote Serial 4: Remote Web
	rfUnit	String	ex) dBm
	powerMode	Integer	0: MANUAL 1: ALC
	serialNumber	String	serial number of the BUC, ex) "11295"
	swVersion	String	software version of BUC., ex) "02.01.18"
	ampName	String	name of the BUC, ex) "AMP_N%1"

	ampState	Integer	0: FAULT 1: POWER_ON 2: READY
	highRfOutAlarm	Integer	dBm
	highRfOutFault	Integer	dBm
	alcSetpoint	Integer	dBm
	manualRfSetpoint	Integer	dBm
	attenuatorSet	Integer	dBm
	disableAlcSetpoint	Integer	dBm
	unlockType	Integer	0: ALARM 1: FAULT
	bandSelect	Integer	0~3(no enum)
	ipAddress	String	IP address of BUC
	netmask	String	Network mask of BUC
	gateway	String	Gateway of BUC
	ntpServer	String	IP address of NTP server
	ntpEnable	Integer	0:disable 1:enable
	vendor	Integer	TBD

BUC model	Key	Type	Descriptions
7: XCVR 8: REVGO 9: MMW_BMIP	txState	Integer	0: STANDBY 1: TRANSMIT
	rfOutput	Integer	RF forward power (dBm)
	temperature	Integer	℃, BUC temperature
	attenuator	Integer	dB, BUC attenuator
	txLocalFreq	Integer	MHz, BUC local frequency
	txPLLStatus	Integer	Tx PLL Lock State 0 : Unlock 1 : Lock
	outputStatus	Integer	0 : Normal 1 : Overdriven 2 : Mute 3 : MuteInternal
	temperatureStatus	Integer	0 : Normal 1 : OverTemperature
	fanStatus	Integer	0 : Normal 1 : Standby 2 : Failed
	partNumber	String	Part number of BUC
	serialNumber	String	Serial number of BUC

	swVersion	String	Software version of BUC
	ipAddress	String	IP address of BUC
7: XCVR	txPolarity	Integer	0: VERTICAL, 1: HORIZONTAL, 2: RHCP, 3: LHCP
	rxPolarity	Integer	0: VERTICAL, 1: HORIZONTAL, 2: RHCP, 3: LHCP
	rxLocalFreq	Integer	MHz, rx local frequency
	rxPLLStatus	Integer	Rx PLL Lock State 0 : Unlock 1 : Lock

Response Example


```
{
    "payload": [
        {
            "acuConnectionState ": 1,
            "avblBand": 3,
            "avblGeoBand": 3,
            "avblMeoBand": 4,
            "avblOrbit": 24,
            "azAbs": 54.02,
            "azRel": 54.02,
            "azTar": 0.0,
            "blockages": [
                {
                    "azEnd": 210,
                    "azStart": 170,
                    "el": 90,
                    "enable": 0
                },
                {
                    "azEnd": 25,
                    "azStart": 15,
                    "el": 51,
                    "enable": 0
                },
                {
                    "azEnd": 175,
                    "azStart": 155,
                    "el": 90,
                    "enable": 0
                },
                {
                    "azEnd": 155,
```

```
                    "azStart": 141,
                    "el": 45,
                    "enable": 0
                }
            ],
            "bowOffset": 182,
            "buc": {
                "attenuator": 9.0,
                "connPort": 1,
                "fanStatus": 0,
                "ipAddress": "172.22.1.28",
                "model": 8,
                "outputStatus": 2,
                "partNumber": "RGUC-AT40-48RE",
                "rfOutput": 22.69,
                "satBand": 2,
                "satOrbit": 1,
                "serialNumber": "R210306650",
                "swVersion": "0.1.b272",
                "temperature": 31.10,
                "temperatureStatus": 0,
                "txLocalFreq": 26550,
                "txPLLStatus": 0,
                "txState": 1
            },
            "bucFault": 0,
            "bucIp": {
                "cBand": "172.22.1.28",
                "kaBand": "172.22.1.28",
                "kuBand": "172.22.1.28",
                "meoKaBand": "172.22.1.28"
            },
            "bucTxClock": {
                "c": "172.22.1.28",
                "ku": "172.22.1.28",
                "ka": "172.22.1.28",
                "meoKa": "172.22.1.28",
            },
            "connPort": 1,
            "connPort2": 0,
            "connectionState": 1,
            "curBand": 2,
            "curOrbit": 1,
            "demod": "lock/off-A",
            "description": "Intellian Antenna",
            "el": -53.11,
            "elAbs": -53.11,
            "elTar": -53.11,
            "enable": 1,
            "esno": 12.4,
```

```
            "ew": 0,
            "fault": [
                {
                    "code": 123,
                    "message": "ACU_COMMUNICATION",
                    "severity": 1,
                    "time": ""
                }
            ],
            "heading": 0.0,
            "id": "12345678",
            "ifFreq": 1416000,
            "inBlockage": false,
            "ip": "10.1.105.143",
            "ipPort": 4010,
            "latitude": 37.11,
            "longitude": 127.06,
            "manufacturer": 0,
            "model": "v240MT",
            "modelCode": 5,
            "name": "A1 G2",
            "ns": 3,
            "operationState": 1,
            "pol": 0.0,
            "rfChain": 1,
            "rxLocal": 16750,
            "rxPol": 3,
            "signalLevel": 220,
            "simultaneousBand": {
                "config": [
                    {
                        "band": 2,
                        "enable": true
                    },
                    {
                        "band": 3,
                        "enable": true
                    }
                ],
                "trackingBand": 2
            },
            "supBand": 3,
            "supGeoBand": 3,
            "supMeoBand": 4,
            "supOrbit": 24,
            "targetSatBand": 2,
            "targetSatLongitude": 267.26,
            "targetSatName": "O3B M017",
            "targetSatType": 1,
            "trackingStatus": 3,
```

```
            "txPol": 3
        },
        {
            "acuConnectionState ": 3,
            "avblBand": 3,
            "avblGeoBand": 3,
            "avblMeoBand": 4,
            "avblOrbit": 24,
            "azAbs": 54.02,
            "azRel": 54.02,
            "azTar": 0.0,
            "blockages": [
                {
                    "azEnd": 210,
                    "azStart": 170,
                    "el": 90,
                    "enable": 0
                },
                {
                    "azEnd": 25,
                    "azStart": 15,
                    "el": 51,
                    "enable": 0
                },
                {
                    "azEnd": 175,
                    "azStart": 155,
                    "el": 90,
                    "enable": 0
                },
                {
                    "azEnd": 155,
                    "azStart": 141,
                    "el": 45,
                    "enable": 0
                }
            ],
            "bowOffset": 182,
            "buc": {
                "attenuator": 9.0,
                "connPort": 1,
                "fanStatus": 0,
                "ipAddress": "172.22.1.28",
                "model": 8,
                "outputStatus": 2,
                "partNumber": "RGUC-AT40-48RE",
                "rfOutput": 22.69,
                "satBand": 2,
                "satOrbit": 1,
                "serialNumber": "R210306650",
```

```
            "swVersion": "0.1.b272",
            "temperature": 31.10,
            "temperatureStatus": 0,
            "txLocalFreq": 26550,
            "txPLLStatus": 0,
            "txState": 1
        },
        "bucFault": 0,
        "bucIp": {
            "cBand": "172.22.1.28",
            "kaBand": "172.22.1.28",
            "kuBand": "172.22.1.28",
            "meoKaBand": "172.22.1.28"
        },
        "bucTxClock": {
            "c": "172.22.1.28",
            "ku": "172.22.1.28",
            "ka": "172.22.1.28",
            "meoKa": "172.22.1.28",
        },
        "connPort": 2,
        "connectionState": 1,
        "curBand": 2,
        "curOrbit": 1,
        "demod": "lock/on-A",
        "description": "Intellian Antenna",
        "el": -53.11,
        "enable": 0,
        "esno": 12.4,
        "ew": 0,
        "fault": [],
        "heading": 0.0,
        "id": "12345678",
        "inBlockage": false,
        "ip": "10.1.105.143",
        "ipPort": 4010,
        "latitude": 37.11,
        "longitude": 127.06,
        "manufacturer": 0,
        "model": "v240MT",
        "modelCode": 5,
        "name": "A1 G2",
        "ns": 3,
        "operationState": 1,
        "pol": 0.0,
        "rxPol": 3,
        "signalLevel": 220,
        "simultaneousBand": {
            "config": [
                {
```

```
                        "band": 0,
                        "enable": false
                    },
                    {
                        "band": 0,
                        "enable": true
                    }
                ],
                "trackingBand": 0
            },
            "supBand": 3,
            "supGeoBand": 3,
            "supMeoBand": 4,
            "supOrbit": 24,
            "targetSatBand": 2,
            "targetSatLongitude": 267.26,
            "targetSatName": "O3B M017",
            "targetSatType": 1,
            "trackingStatus": 3,
            "txPol": 3
        }
    ],
    "status": 1
}
```


5.1.5. POST /api/v1/asset/antenna/complex-blockages (Block1 Only)

Description

This API is used to delete already registered antenna in Intellian Mediator. Registered Antenna connPort should be specified for deleting antenna.

Request Field

Key	Type	Descriptions
force	Boolean	Request Complex blockzone [False: send response only when value changes True : send response unconditionally]

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/antenna/complex-blockages


```
{
    "force": false
}
```


Response Field

Key	Type	Descriptions
blockageMode	Integer	Blockzone mode 0: Blockzone not in use 1: Normal blockzone 2: Complex blockzone
complexBlockages	Array Number array	Complex blockzone data array (Number array x 360) Number Array: [Blockzone start azimuth angle, Blockzone first start elevation angle, Blockzone first end elevation angle, Blockzone second start elevation angle, Blockzone second end elevation angle]
connPort	Integer	Antenna connetion port(1~8)
dc	Integer	Data center ID

Response Example


```
{
    "payload": [
    {
        "blockageMode": 2,
        "complexBlockages": [
            [0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0],
            …
        ],
        "connPort": 1,
        "dc": 1
    },
    {
        "blockageMode": 2,
        "complexBlockages": [],
        "connPort": 2,
        "dc": 1
    },
    {
        "blockageMode": 1,
        "complexBlockages": [],
        "connPort": 3,
        "dc": 1
    }
    ],
    "status": 1
}
```

5.1.6. POST /api/v1/mediation/simultaneous_band/track (Block1 Only)

Description

This API is used to set current tracking band.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port(1~8)
trackingBand	Integer	Tracking target band [0:Ku, 1:C, 2:Ka, 3:X]

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediation/simultaneous_band/track

```
{
    "connPort": 1,
    "trackingBand": 2
}
```

Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port(1~8)
trackingBand	Integer	Tracking target band [0:Ku, 1:C, 2:Ka, 3:X]

Response Example

```
{
    "payload": {
        "connPort": 1,
        "trackingBand": 2
    },
    "status": 1
}
```

5.1.7. GET /api/v1/mediation/simultaneous_band/track (Block1 Only)

Description

This API is used to get current tracking band.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediation/simultaneous_band/track

Response Field

Key/Field	Type	Descriptions
connPort	Integer	Antenna connetion port(1~8)
trackingBand	Integer	Tracking target band [0:Ku, 1:C, 2:Ka, 3:X]

Response Example

```
{
    "payload": {
        "connPort": 1,
        "trackingBand": 2
    },
    "status": 1
}
```

5.1.8. POST /api/v1/mediation/simultaneous_band/active (Block1 Only)

Description

This API is used to enable/disable band service.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port(1~8)
band	Integer	Target band [0:Ku, 1:C, 2:Ka, 3:X]
enable	Boolean	Enable/Disable band service [False: Disable, True: Enable]

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediation/simultaneous_band/active

```
{
    "connPort": 1,
    "band": 3,
    "enable": true
}
```

Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port(1~8)
band	Integer	Target band

		[0:Ku, 1:C, 2:Ka, 3:X]
enable	Boolean	Enable/Disable band service [False: Disable, True: Enable]

Response Example


```
{
    "payload": {
        "connPort": 1,
        "band": 3,
        "enable": true
    },
    "status": 1
}
```


5.1.9. GET /api/v1/mediation/simultaneous_band/active (Block1 Only)

Description

This API is used to enable/disable band service.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediation/simultaneous_band/active

Response Field

Key/Field	Type	Descriptions
connPort	Integer	Antenna connetion port(1~8)
band	Integer	Target band [0:Ku, 1:C, 2:Ka, 3:X]
enable	Boolean	Enable/Disable band service [False: Disable, True: Enable]

Response Example


```
{
    "payload": {
        "connPort": 1,
        "band": 3,
        "enable": true
    },
    "status": 1
}
```

5.2. Modem

Method	URL	Descriptions
POST	/api/v1/asset/modem/create	Create a modem
POST	/api/v1/asset/modem/update	Update a modem
POST	/api/v1/asset/modem/delete	Delete a modem
GET	/api/v1/asset/modem/list	Getting information about all modems
POST	/api/v1/asset/modem/target_satellite	Setting information about modem's taget satellite
GET	/api/v1/asset/modem/target_satellite	Getting information about modem's taget satellite
POST	/api/v1/control/target_satellite/tle_upload	Upload TLE for target satellite
GET	/api/v1/control/target_satellite/tle_download	Download TLE for target satellite

5.2.1. POST /api/v1/asset/modem/create

Description

This API is used to register new modem which uses http/https protocol or serial to communicate with Intellian mediator. Modem port, id, name etc details should be specified for registering modem.

Request Field

Key	Type	Descriptions
connPort	Integer	Port number of Mediator which the modem is connected to (1 ~ 8)
connPort2	Integer	2nd port for modem
connPortType	Integer	Type of connection port type 0: None 1: Primary port 2: Secondary port
id	String	Asset ID
name	String	Name of the modem registered by user
description	String	Description of modem inforamtion
enable	Integer	0: Disable 1: Enable
model	Integer	Refer to [List of model]
modemType	Integer	Refer to [List of modemType]
modemPort	Integer	0: RS-232 1: RS-422 2: ETHERNET
protocol	Integer	Refer to [List of protocol]
gpsOutProtocol	Integer	0: GPGLL 1: GPGGA 2: SIMPLE GPGGA
useTxMute	Integer	0: No (Disable) 1: Yes (Enable)
useModemLock	Integer	0: No (Disable) 1: Yes (Enable)
txMute	Integer	0: Low 1: High
modemLock	Integer	0: Low

		1: High
curBand	Integer	Current activated band [0:Ku, 1:C, 2:Ka, 3: X]
curOrbit	Integer	Current activated orbit [0:GEO, 1:MEO, 2:LEO]
supGeoBand	Integer	The bitmask of support band for GEO [0:Ku, 1:C, 2:Ka, 3: X] * bitmask of band : Ku(0x01) | C(0x02) | Ka(0x04) | X(0x8)
supMeoBand	Integer	The bitmask of support band for MEO [2:Ka] * bitmask of band : Ka(0x04)
ip	String	Network IP of the modem
subnetMask	String	Network subnet mask of the modem
gateway	String	Network gateway of the modem
dns	String	Network DNS of the modem
tcpPort	Integer	TCP communication port number of modem
tcpPort2	Integer	TCP communication 2nd port number of modem
udpPort	Integer	UDP communication port number of modem (If the value is 0, use the same port as tcpPort.)
udpPort2	Integer	UDP communication 2nd port number of modem (If the value is 0, use the same port as tcpPort2.)
commIp	String	Network IP of the modem for communication
commSubnet	String	Network subnet mask of the modem for communication
commGateway	String	Network gateway of the modem for communication

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/modem/create

```
{
    "connPort": 6,
    "connPort2": 7,
    "id": "12345678",
    "name": "Gilat",
    "description": "Intellian Modem",
    "enable": 0,
    "model": 0,
    "modemType": 2,
    "modemPort": 16,
    "protocol": 12,
    "gpsOutProtocol": 0,
    "useTxMute": 1,
    "useModemLock": 1,
    "txMute": 0,
    "modemLock": 0,
    "curBand": 0,
    "curOrbit": 0,
    "supGeoBand": 7,
    "supMeoBand": 4,
    "ip": "172.22.1.39",
    "subnetMask": "255.255.255.0",
    "gateway": "172.22.1.1",
    "dns": "192.168.2.1",
```

```
    "tcpPort": 5001,
    "tcpPort2": 5002,
    "udpPort": 0,
    "udpPort2": 0,
    "commIp": "172.22.1.40",
    "commSubnet": "255.255.255.0",
    "commGateway": "172.22.1.1"
}
```


Response Field

Key	Type	Descriptions
connPort	Integer	Port number of Mediator which the modem is connected to (1 ~ 8)
connPort2	Integer	2nd port for modem
connPortType	Integer	Type of connection port type 0: None 1: Primary port 2: Secondary port
id	String	Asset ID
name	String	Name of the modem registered by user
description	String	Description of modem inforamtion
enable	Integer	0: Disable 1: Enable
model	Integer	Refer to [List of model]
modemType	Integer	Refer to [List of modemType]
modemPort	Integer	0: RS-232 1: RS-422 2: ETHERNET
protocol	Integer	Refer to [List of protocol]
gpsOutProtocol	Integer	0: GPGLL 1: GPGGA 2: SIMPLE GPGGA
useTxMute	Integer	0: No (Disable) 1: Yes (Enable)
useModemLock	Integer	0: No (Disable) 1: Yes (Enable)
txMute	Integer	0: Low 1: High
modemLock	Integer	0: Low 1: High
curBand	Integer	Current activated band [0:Ku, 1:C, 2:Ka, 3: X]
curOrbit	Integer	Current activated orbit [0:GEO, 1:MEO, 2:LEO]
supGeoBand	Integer	The bitmask of support band for GEO [0:Ku, 1:C, 2:Ka, 3: X] * bitmask of band : Ku(0x01) | C(0x02) | Ka(0x04) | X(0x8)
supMeoBand	Integer	The bitmask of support band for MEO [2:Ka] * bitmask of band : Ka(0x04)
ip	String	Network IP of the modem
subnetMask	String	Network subnet mask of the modem
gateway	String	Network gateway of the modem
dns	String	Network DNS of the modem
tcpPort	Integer	TCP communication port number of modem

tcpPort2	Integer	TCP communication 2nd port number of modem
udpPort	Integer	UDP communication port number of modem (If the value is 0, use the same port as tcpPort.)
udpPort2	Integer	UDP communication 2nd port number of modem (If the value is 0, use the same port as tcpPort2.)
commIp	String	Network IP of the modem for communication
commSubnet	String	Network subnet mask of the modem for communication
commGateway	String	Network gateway of the modem for communication

Response Example


```
{
    "payload": {
        "connPort": 6,
        "connPort2": 7,
        "id": "12345678",
        "name": "Gilat",
        "description": "Intellian Modem",
        "enable": 0,
        "model": 0,
        "modemType": 2,
        "modemPort": 16,
        "protocol": 12,
        "gpsOutProtocol": 0,
        "useTxMute": 1,
        "useModemLock": 1,
        "txMute": 0,
        "modemLock": 0,
        "curBand": 0,
        "curOrbit": 0,
        "supGeoBand": 7,
        "supMeoBand": 4,
        "ip": "172.22.1.39",
        "subnetMask": "255.255.255.0",
        "gateway": "172.22.1.1",
        "dns": "192.168.2.1",
        "tcpPort": 5001,
        "tcpPort2": 5002,
        "udpPort": 0,
        "udpPort2": 0,
        "commIp": "172.22.1.40",
        "commSubnet": "255.255.255.0",
        "commGateway": "172.22.1.1"
    },
    "status": 1
}
```


5.2.2. POST /api/v1/asset/modem/update

Description

This API is used to update information of registered modem. Modem ipaddress, name, support band and updated configuration parameters should be specified for updating existing configuration of modem.

Request Field

Key	Type	Descriptions
connPort	Integer	Port number of Mediator which the modem is connected to (1 ~ 8)
connPort2	Integer	2nd port for modem
connPortType	Integer	Type of connection port type 0: None 1: Primary port 2: Secondary port
id	String	Asset ID
name	String	Name of the modem registered by user
description	String	Description of modem inforamtion
enable	Integer	0: Disable 1: Enable
model	Integer	Refer to [List of model]
modemType	Integer	Refer to [List of modemType]
modemPort	Integer	0: RS-232 1: RS-422 2: ETHERNET
protocol	Integer	Refer to [List of protocol]
gpsOutProtocol	Integer	0: GPGLL 1: GPGGA 2: SIMPLE GPGGA
useTxMute	Integer	0: No (Disable) 1: Yes (Enable)
useModemLock	Integer	0: No (Disable) 1: Yes (Enable)
txMute	Integer	0: Low 1: High
modemLock	Integer	0: Low 1: High
curBand	Integer	Current activated band [0:Ku, 1:C, 2:Ka, 3: X]
curOrbit	Integer	Current activated orbit [0:GEO, 1:MEO, 2:LEO]
supGeoBand	Integer	The bitmask of support band for GEO [0:Ku, 1:C, 2:Ka, 3: X] * bitmask of band : Ku(0x01) | C(0x02) | Ka(0x04) | X(0x8)
supMeoBand	Integer	The bitmask of support band for MEO [2:Ka] * bitmask of band : Ka(0x04)
ip	String	Network IP of the modem
subnetMask	String	Network subnet mask of the modem
gateway	String	Network gateway of the modem
dns	String	Network DNS of the modem
tcpPort	Integer	TCP communication port number of modem
tcpPort2	Integer	TCP communication 2nd port number of modem
udpPort	Integer	UDP communication port number of modem (If the value is 0, use the same port as tcpPort.)
udpPort2	Integer	UDP communication 2nd port number of modem (If the value is 0, use the same port as tcpPort2.)
commIp	String	Network IP of the modem for communication
commSubnet	String	Network subnet mask of the modem for communication
commGateway	String	Network gateway of the modem for communication

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/modem/update


```
{
    "connPort": 6,
    "connPort2": 7,
    "id": "12345678",
    "name": "Gilat",
    "description": "Intellian Modem",
    "enable": 0,
    "model": 0,
    "modemType": 2,
    "modemPort": 16,
    "protocol": 12,
    "gpsOutProtocol": 0,
    "useTxMute": 1,
    "useModemLock": 1,
    "txMute": 0,
    "modemLock": 0,
    "curBand": 0,
    "curOrbit": 0,
    "supGeoBand": 7,
    "supMeoBand": 4,
    "ip": "172.22.1.39",
    "subnetMask": "255.255.255.0",
    "gateway": "172.22.1.1",
    "dns": "192.168.2.1",
    "tcpPort": 5001,
    "tcpPort2": 5002,
    "udpPort": 0,
    "udpPort2": 0,
    "commIp": "172.22.1.40",
    "commSubnet": "255.255.255.0",
    "commGateway": "172.22.1.1"
}
```


Response Field

Key	Type	Descriptions
connPort	Integer	Port number of Mediator which the modem is connected to (1 ~ 8)
connPort2	Integer	2nd port for modem
connPortType	Integer	Type of connection port type 0: None 1: Primary port 2: Secondary port
id	String	Asset ID
name	String	Name of the modem registered by user
description	String	Description of modem inforamtion

enable	Integer	0: Disable 1: Enable
model	Integer	Refer to [List of model]
modemType	Integer	Refer to [List of modemType]
modemPort	Integer	0: RS-232 1: RS-422 2: ETHERNET
protocol	Integer	Refer to [List of protocol]
gpsOutProtocol	Integer	0: GPGLL 1: GPGGA 2: SIMPLE GPGGA
useTxMute	Integer	0: No (Disable) 1: Yes (Enable)
useModemLock	Integer	0: No (Disable) 1: Yes (Enable)
txMute	Integer	0: Low 1: High
modemLock	Integer	0: Low 1: High
curBand	Integer	Current activated band [0:Ku, 1:C, 2:Ka, 3: X]
curOrbit	Integer	Current activated orbit [0:GEO, 1:MEO, 2:LEO]
supGeoBand	Integer	The bitmask of support band for GEO [0:Ku, 1:C, 2:Ka, 3: X] * bitmask of band : Ku(0x01) | C(0x02) | Ka(0x04) | X(0x8)
supMeoBand	Integer	The bitmask of support band for MEO [2:Ka] * bitmask of band : Ka(0x04)
ip	String	Network IP of the modem
subnetMask	String	Network subnet mask of the modem
gateway	String	Network gateway of the modem
dns	String	Network DNS of the modem
tcpPort	Integer	TCP communication port number of modem
tcpPort2	Integer	TCP communication 2nd port number of modem
udpPort	Integer	UDP communication port number of modem (If the value is 0, use the same port as tcpPort.)
udpPort2	Integer	UDP communication 2nd port number of modem (If the value is 0, use the same port as tcpPort2.)
commIp	String	Network IP of the modem for communication
commSubnet	String	Network subnet mask of the modem for communication
commGateway	String	Network gateway of the modem for communication

Response Example


```
{
    "payload": {
        "connPort": 6,
        "connPort2": 7,
        "id": "12345678",
        "name": "Gilat",
        "description": "Intellian Modem",
        "enable": 0,
        "model": 0,
        "modemType": 2,
```

```
        "modemPort": 16,
        "protocol": 12,
        "gpsOutProtocol": 0,
        "useTxMute": 1,
        "useModemLock": 1,
        "txMute": 0,
        "modemLock": 0,
        "curBand": 0,
        "curOrbit": 0,
        "supGeoBand": 7,
        "supMeoBand": 4,
        "ip": "172.22.1.39",
        "subnetMask": "255.255.255.0",
        "gateway": "172.22.1.1",
        "dns": "192.168.2.1",
        "tcpPort": 5001,
        "tcpPort2": 5002,
        "udpPort": 0,
        "udpPort2": 0,
        "commIp": "172.22.1.40",
        "commSubnet": "255.255.255.0",
        "commGateway": "172.22.1.1"
    },
    "status": 1
}
```


5.2.3. POST /api/v1/asset/modem/delete

Description

This API is used to delete already registered modem in Intellian Mediator. Registered Modem connPort should be specified for deleting modem.

Request Field

Key	Type	Descriptions
connPort	Integer	Modem connetion port(1~8) want to delete

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/modem/delete


```
{
    "connPort": 6
}
```


Response Field

Key	Type	Descriptions

connPort	Integer	Modem connetion port(1~8)

Response Example

```
{
    "payload": {
        "connPort": 6
    },
    "status": 1
}
```

5.2.4. GET /api/v1/asset/modem/list

Description

This API is used to get the information about all modem lists. General information about the modem like IP Address, Version, Status etc information can be retrieved.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/modem/list

Response Field

Key/Field	Type	Descriptions
connPort	Integer	Port number of Mediator which the modem is connected to(1~8)
connPort2	Integer	2nd port for modem
id	String	Asset ID
name	String	Name of the modem registered by user
description	String	Description of modem inforamtion
enable	Integer	0: Disable 1: Enable]
model	Integer	Refer to [List of model]
modemType	Integer	Refer to [List of modemType]
modemPort	Integer	0: RS-232 1: RS-422 2: ETHERNET
protocol	Integer	Refer to [List of protocol]
gpsOutProtocol	Integer	0: GPGLL 1: GPGGA 2: SIMPLE GPGGA
useTxMute	Integer	0: No (Disable) 1: Yes (Enable)
useModemLock	Integer	0: No (Disable) 1: Yes (Enable)
txMute	Integer	0: Low 1: High
modemLock	Integer	0: Low 1: High

curBand	Integer	Current activated band [0:Ku, 1:C, 2:Ka, 3: X]
curOrbit	Integer	Current activated orbit [0:GEO, 1:MEO, 2:LEO]
supGeoBand	Integer	The bitmask of support band for GEO [0:Ku, 1:C, 2:Ka, 3: X] * bitmask of band : Ku(0x01) | C(0x02) | Ka(0x04) | X(0x8)
supMeoBand	Integer	The bitmask of support band for MEO [2:Ka] * bitmask of band : Ka(0x04)
ip	String	Network IP of the modem
subnetMask	String	Network subnet mask of the modem
gateway	String	Network gateway of the modem
dns	String	Network DNS of the modem
tcpPort	Integer	TCP communication port number of modem
tcpPort2	Integer	TCP communication 2nd port number of modem
udpPort	Integer	UDP communication port number of modem (If the value is 0, use the same port as tcpPort.)
udpPort2	Integer	UDP communication 2nd port number of modem (If the value is 0, use the same port as tcpPort2.)
commIp	String	Network IP of the modem for communication
commSubnet	String	Network subnet mask of the modem for communication
commGateway	String	Network gateway of the modem for communication
avblGeoBand	Integer	The bitmask of available band for GEO [0:Ku, 1:C, 2:Ka, 3: X] * bitmask of band : Ku(0x01) | C(0x02) | Ka(0x04) | X(0x8)
avblMeoBand	Integer	The bitmask of available band for MEO [2:Ka] * bitmask of band : Ka(0x04)
connPortType	Integer	Type of connection port type 0: None 1: Primary port 2: Secondary port
connectionState	Integer	Connection state of modem [0: Disconnected, 1: Connected]
targetSatellite	Object array	[Refer to 5.2.6]
modemInterfaceEnable	Integer	Option for Modem interface 0: Disable (Modem interface IP/GW/Subnet can't be edited) 1: Enable (Modem interface IP/GW/Subnet can be edited)

[List of modemType]
0 User Setting
1 iDirect I/O
2 iDirect OpenAMIP
3 Comtech I/O
4 Comtech ROAM
5 HUGHES
6 Satlink Serial
7 Satlink Vacp
8 Elektrikom AMIP
9 GILAT SE II
10 IPSTAR
12 NEWTEC AMIP
13 Viasat (MEO)
14 Comtech (MEO)
15 HUGHES AMIP

16 OpenAMIP v1.17
17 Viasat TMR
99 Modem Emulation

[List of protocol]
0 IO Console
1 OpenAMIP
2 Serial GPS
3 ROAM
4 VACP
5 Elektrikom AMIP
6 GILAT
7 SOTM
8 G5
9 NEWTEC AMIP
10 VIASAT (MEO only)
11 HUGHES AMIP
12 OpenAMIP v1.17

[List of model]
0 Others
1 iDirect X7
2 iDirect 9350
3 Comtech H Pro
4 Comtech 760
5 NewTect MDM6000
6 Test modem

Response Example


```
{
    "payload": [
        {
            "connPort": 6,
            "connPort2": 7,
            "connPortType": 1,
            "connectionState": 1,
            "id": "12345678",
            "name": "Gilat",
            "description": "Intellian Modem",
            "enable": 1,
            "model": 0,
            "modemType": 16,
            "modemPort": 2,
            "protocol": 12,
            "gpsOutProtocol": 0,
            "useTxMute": 1,
            "useModemLock": 1,
            "txMute": 0,
            "modemLock": 0,
            "curBand": 2,
            "curOrbit": 1,
            "supGeoBand": 7,
```

```
            "supMeoBand": 4,
            "avblGeoBand": 0,
            "avblMeoBand": 4,
            "ip": "172.22.1.39",
            "subnetMask": "255.255.255.0",
            "gateway": "172.22.1.1",
            "dns": "192.168.2.1",
            "tcpPort": 5001,
            "tcpPort2": 5002,
            "udpPort": 0,
            "udpPort2": 0,
            "targetSatellite": [],
            "commIp": "172.22.1.40",
            "commSubnet": "255.255.255.0",
            "commGateway": "172.22.1.1",
            "modemInterfaceEnable": 0
        },
        {
            "connPort": 1,
            "connPort2": 2,
            "connPortType": 1,
            "connectionState": 1,
            "id": "12345678",
            "name": "Viasat",
            "description": "Intellian Modem",
            "enable": 0,
            "model": 0,
            "modemType": 13,
            "modemPort": 2,
            "protocol": 10,
            "gpsOutProtocol": 0,
            "useTxMute": 1,
            "useModemLock": 1,
            "txMute": 0,
            "modemLock": 0,
            "curBand": 0,
            "curOrbit": 0,
            "supGeoBand": 7,
            "supMeoBand": 4,
            "avblGeoBand": 0,
            "avblMeoBand": 4,
            "ip": "172.22.1.39",
            "subnetMask": "255.255.255.0",
            "gateway": "172.22.1.1",
            "dns": "192.168.2.1",
            "tcpPort": 5001,
            "tcpPort2": 5002,
            "udpPort": 0,
            "udpPort2": 0,
            "targetSatellite": [],
```

```
            "commIp": "172.22.1.40",
            "commSubnet": "255.255.255.0",
            "commGateway": "172.22.1.1"
            "modemInterfaceEnable": 0
        }
    ],
    "status": 1
}
```


5.2.5. POST /api/v1/asset/modem/target_satellite

Description

This API is used to set the modem's target satellite information. User needs to specify satellite information as a parameter.

Request Field

Key	Type	Descriptions
band	Integer	[0:Ku, 1:C, 2:Ka, 3: X]
bandwidth	Integer	0 ~ 65535
ifFreq	Integer	IF frequency (KHz) 950000 ~ 2150000
inclinedAngle	Integer	Angle for calculating the differences between actual target and obsolete satellite.
longitude	Integer	-180 ~ 180
rxLocalFreq	Integer	Rx local frequency (MHz) C band : < 6000 X band : < 9000 KU band: < 15000 KA band: < 23000 MEO KA band : < 18250
rxPol	Integer	Receive polarity 0: VERTICAL, 1: HORIZONTAL, 2: RHCP, 3: LHCP
satelliteInfoType	Integer	Tracking method 0: Longitude 1: TLE
skewOffset	Integer	-90° ~ 90°
targetSatName	String	Target satellite name
tle	Array	TLE string, Line 1, Line2
tleSatName	String	TLE satellite name
txLocalFreq	Integer	Tx local frequency (MHz)
txPol	Integer	Transmit polarity 0: VERTICAL, 1: HORIZONTAL, 2: RHCP, 3: LHCP

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/modem/target_satellite


```
{
    "band": 1,
    "bandwidth": 0,
    "ifFreq": 1064000,
    "inclinedAngle": 0,
    "longitude": -41,
    "rxLocalFreq": 5150,
    "rxPol": 3,
    "satelliteInfoType": 0,
    "skewOffset": 0,
    "targetSatName": "ses-6",
    "tle": ["", ""],
    "tleSatName": "",
    "txLocalFreq": 0,
    "txPol": 2
}
```


Response Field

Key	Type	Descriptions
band	Integer	[0:Ku, 1:C, 2:Ka, 3: X]
bandwidth	Integer	0 ~ 65535
ifFreq	Integer	IF frequency (KHz) 950000 ~ 2150000
inclinedAngle	Integer	Angle for calculating the differences between actual target and obsolete satellite.
longitude	Integer	-180 ~ 180
rxLocalFreq	Integer	Rx local frequency (MHz) C band : < 6000 X band : < 9000 KU band: < 15000 KA band: < 23000 MEO KA band : < 18250
rxPol	Integer	Receive polarity 0: VERTICAL, 1: HORIZONTAL, 2: RHCP, 3: LHCP
satelliteInfoType	Integer	Tracking method 0: Longitude 1: TLE
skewOffset	Integer	-90° ~ 90°
targetSatName	String	Target satellite name
tle	Array	TLE string, Line 1, Line2
tleSatName	String	TLE satellite name
txLocalFreq	Integer	Tx local frequency (MHz)
txPol	Integer	Transmit polarity 0: VERTICAL, 1: HORIZONTAL, 2: RHCP, 3: LHCP

Response Example


```
{
    "payload": {
        "band": 1,
        "bandwidth": 0,
        "ifFreq": 1064000,
        "inclinedAngle": 0,
        "longitude": -41,
        "rxLocalFreq": 5150,
        "rxPol": 3,
        "satelliteInfoType": 0,
        "skewOffset": 0,
        "targetSatName": "ses-6",
        "tle": ["", ""],
        "tleSatName": "",
        "txLocalFreq": 0,
        "txPol": 2
    },
    "status": 1
}
```


5.2.6. GET /api/v1/asset/modem/target_satellite

Description

This API is used to get the modem's target satellite information. User needs to specify connPort as a parameter.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/modem/target_satellite

Response Field

Key/Field	Type	Descriptions
band	Integer	[0:Ku, 1:C, 2:Ka, 3: X]
bandwidth	Integer	0 ~ 65535
ifFreq	Integer	IF frequency (KHz) 950000 ~ 2150000
inclinedAngle	Integer	Angle for calculating the differences between actual target and obsolete satellite.
longitude	Number	-180 ~ 180
rxLocalFreq	Integer	Rx local frequency (MHz) C band : < 6000 X band : < 9000 KU band: < 15000 KA band: < 23000

		MEO KA band : < 18250
rxPol	Integer	Receive polarity 0: VERTICAL, 1: HORIZONTAL, 2: RHCP, 3: LHCP
satelliteInfoType	Integer	Tracking method 0: Longitude 1: TLE
skewOffset	Integer	-90° ~ 90°
targetSatName	String	Target satellite name
tle	Array	TLE string, Line 1, Line2
tleSatName	String	TLE satellite name
txLocalFreq	Integer	Tx local frequency (MHz)
txPol	Integer	Transmit polarity 0: VERTICAL, 1: HORIZONTAL, 2: RHCP, 3: LHCP

Response Example


```
{
    "payload": [
        {
            "band": 1,
            "bandwidth": 0,
            "ifFreq": 1064000,
            "inclinedAngle": 0,
            "longitude": -41,
            "rxLocalFreq": 5150,
            "rxPol": 3,
            "satelliteInfoType": 0,
            "skewOffset": 0,
            "targetSatName": "ses-6",
            "tle": ["", ""],
            "tleSatName": "",
            "txLocalFreq": 0,
            "txPol": 2
        }
    ],
    "status": 1
}
```


5.2.7. POST /api/v1/control/target_satellite/tle_upload

Description

This API is used to upload TLE files for target satellite information. User needs to specify connPort, band as a parameter.

Request Field

Content-type : multipart/form-data

Key / Field	Type	Descriptions
filename	String	TLE file to be uploaded
connPort	Integer	Port number of Mediator which the modem is connected to(1~8)
band	Integer	[0:Ku, 1:C, 2:Ka, 3: X]

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/control/target_satellite/tle_upload

```
------WebKitFormBoundaryR9UmRWkqj8OWtC4U
Content-Disposition: form-data; name="filename"; filename="SES_17.txt"
Content-Type: text/plain


------WebKitFormBoundaryR9UmRWkqj8OWtC4U
Content-Disposition: form-data; name="connPort"

3
------WebKitFormBoundaryR9UmRWkqj8OWtC4U
Content-Disposition: form-data; name="band"

2
------WebKitFormBoundaryR9UmRWkqj8OWtC4U--
```

Response Field

Key / Field	Type	Descriptions
tleSatName	String	Satellite name of TLE file
tle	Array	TLE string, Line 1, Line2

Response Example


```
{
    "payload": {
        "tle": [
          "1 49332U 21095A   23189.06253312 -.00000277  00000+0  00000+0 0  9998",
          "2 49332   0.0168  45.3462 0001144  63.0698 132.7498  1.00269266  7052"
        ],
        "tleSatName": "SES-17"
    },
    "status": 1
}
```


5.2.8. GET /api/v1/control/target_satellite/tle_download

Description

This API is used to download the TLE file of target satellite information. User needs to specify connPort, band as a parameter

Request Field

Key / Field	Type	Descriptions
connPort	Integer	Port number of Mediator which the modem is connected to(1~8)
band	Integer	[0:Ku, 1:C, 2:Ka, 3: X]

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/control/target_satellite/tle_download

```
{
    "band ": 2,
    "connPort": 3
}
```

Response Field

Key	Type	Descriptions
path	String	Path of the TLE file to download

Response Example

```
{
    "payload": {
        "path": "/usr/local/download/system-data/tle_3_ka"
    },
    "status": 1
}
```

5.3. Link(RF Path)

Method	URL	Descriptions
POST	/api/v1/mediation/operation/links	Create & Update links
GET	/api/v1/mediation/operation/links	Getting information about all links
DELETE	/api/v1/mediation/operation/link	Delete a link
POST	/api/v1/mediation/operation/state	Set operation mode
POST	/api/v1/mediation/operation/service/switch	Service switch for Block1 (Block1 Only)

5.3.1. POST /api/v1/mediation/operation/links

Description

This API is used to set link. id, modemPort, antennaPort, role, dc, type etc details should be specified

for set link.

Request Field

Key	Type	Descriptions
id	Integer	Link ID, 1~8(Single)/1~16(Dual Date Center)
antPort	Integer	Antenna port number, 1~8
mdmPort	Integer	Modem port number, 1~8
role	Integer	Antenna's Role 0: None 1: Primary 2: Secondary 3: Backup
dc	Integer	DC number, 1~2
type	Integer	Link type [0: Normal, 1: Fixed]

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediation/operation/links

```
{
    "links": [
        {
            "id": 1,
            "antPort": 1,
            "mdmPort": 1,
            "role": 1,
            "dc": 1,
            "type": 1
        },
        {
            "id": 2,
            "antPort": 2,
            "mdmPort": 1,
            "role": 2,
            "dc": 1,
            "type": 0
        }
    ]
}
```

Response Field

Key	Type	Descriptions
id	Integer	Link ID, 1~8(Single)/1~16(Dual Date Center)
antPort	Integer	Antenna port number, 1~8
mdmPort	Integer	Modem port number, 1~8
role	Integer	Antenna's Role 0: None 1: Primary

		2: Secondary 3: Backup
dc	Integer	DC number, 1~2
type	Integer	Link type [0: Normal, 1: Fixed]

Response Example


```
{
    "payload": {
        "links": [
            {
                "id": 1,
                "antPort": 1,
                "mdmPort": 1,
                "role": 1,
                "dc": 1,
                "type": 1
            },
            {
                "id": 2,
                "antPort": 2,
                "mdmPort": 1,
                "role": 2,
                "dc": 1,
                "type": 0
            }
        ]
    },
    "status": 1
}
```


5.3.2. GET /api/v1/mediation/operation/links

Description

This API is used to get the link information.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediation/operation/links

Response Field

Key/Field	Type	Descriptions
id	Integer	Link ID, 1~8(Single)/1~16(Dual Date Center)

antPort	Integer	Antenna port number, 1~8
mdmPort	Integer	Modem port number, 1~8
role	Integer	Antenna's Role 0: None 1: Primary 2: Secondary 3: Backup
dc	Integer	DC number, 1~2
type	Integer	Link type [0: Normal, 1: Fixed]

Response Example


```
{
    "payload": [
        {
            "id": 1,
            "antPort": 1,
            "mdmPort": 1,
            "role": 1,
            "dc": 1,
            "type": 1
        },
        {
            "id": 2,
            "antPort": 2,
            "mdmPort": 1,
            "role": 2,
            "dc": 1,
            "type": 0
        }
    ],
    "status": 1
}
```


5.3.3. DELETE /api/v1/mediation/operation/link

Description

This API is used to delete already registered link in Intellian Mediator. Registered link's id should be specified for deleting link.

Request Field

Key	Type	Descriptions
id	Integer	Link ID want to delete

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediation/operation/link

```
{
    "id": 2
}
```

Response Field

Key	Type	Descriptions
id	Integer	Link ID want to delete

Response Example

```
{
    "payload": {
        "id": 2
    },
    "status": 1
}
```

5.3.4. POST /api/v1/mediation/operation/state

Description

This API is used to set operation mode of mediator.

Request Field

Key	Type	Descriptions
state	Integer	Operation mode of Mediator 0: None 1: Setup mode 2: Automatic mode

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediation/operation/state

```
{
    "state": 1
}
```

Response Field

Key	Type	Descriptions
state	Integer	Operation mode of Mediator 0: None 1: Setup mode 2: Automatic mode

Response Example

```
{
    "payload": {
        "state": 1
    },
    "status": 1
}
```

5.4. BUC

Method	URL	Descriptions
POST	/api/v1/asset/buc/info	Getting information about specific BUC
POST	/api/v1/asset/buc/config	Setting BUC basic configurations
POST	/api/v1/asset/buc/config/tx_state	Setting BUC Tx
POST	/api/v1/asset/buc/config/amp_info	Setting BUC Amp name (CPI BUC Only)
POST	/api/v1/asset/buc/config/setpoint	Setting BUC setpoint (CPI BUC Only)
POST	/api/v1/asset/buc/config/meo	Setting BUC unlockType, bandSelect (CPI BUC Only)
POST	/api/v1/asset/buc/config/ip	Setting BUC IP (CPI BUC Only)
POST	/api/v1/asset/buc/config/clock	Setting NTP configurations (CPI BUC Only)
POST	/api/v1/asset/buc/fault/reset	Resetting BUC fault (CPI BUC Only)
POST	/api/v1/asset/buc/config/tx_clock	Setting Reference Tx Clock(10MHz/50MHz)

5.4.1. POST /api/v1/asset/buc/info

Description

This API is used to request information to BUC.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/buc/info

```
{
    "connPort": 1
}
```

Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)

Response Example

```
{
    "payload": {
        "connPort": 1
    },
    "status": 1
}
```

5.4.2. POST /api/v1/asset/buc/config

Description

This API is used to set the BUC configuration.

Request Field

OpenBMIP and GEO BUC(General, NJRC, Comtech, Terrasat, Teledyne, MMW)

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
txState	Integer	0: STANDBY 1: TRANSMIT
attenuator	Integer	GEO BUC Only 0 ~ 15.5 (step 0.25) (General) 0 ~ 15.5 (step 0.5) (1: NJRC) 0 ~ 20.0 (step 0.25) (2: COMTECH) 0 ~ 20.0 (step 0.1) (4: TELEDYNE) 0 ~ 30 (step 0.1) (5: MMW)
txLoFreqIndex	Enum	0~3 (MMW only)
muteIoPolarity	Enum	0, 1 (MMW only)

CPI BUC(BCIP)

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
txState	Integer	0: STANDBY 1: TRANSMIT 2: TRANSMITINHIBIT 3: WARMUP
powerMode	Enum	0: Manual 1: ALC

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/buc/config

```
{
    "connPort": 1,
    "txState": 1,
    "attenuator": 0,
    "txLoFreqIndex": 0,
    "muteIoPolarity": 0
}
```


Response Field

OpenBMIP and GEO BUC(General, NJRC, Comtech, Terrasat, Teledyne, MMW)

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
txState	Integer	0: STANDBY 1: TRANSMIT
attenuator	Integer	GEO BUC Only 0 ~ 15.5 (step 0.25) (General) 0 ~ 15.5 (step 0.5) (1: NJRC) 0 ~ 20.0 (step 0.25) (2: COMTECH) 0 ~ 20.0 (step 0.1) (4: TELEDYNE) 0 ~ 30 (step 0.1) (5: MMW)
txLoFreqIndex	Enum	0~3 (MMW only)
muteIoPolarity	Enum	0, 1 (MMW only)

CPI BUC(BCIP)

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
txState	Integer	0: STANDBY 1: TRANSMIT 2: TRANSMITINHIBIT 3: WARMUP
powerMode	Enum	0: Manual 1: ALC

Response Example


```
{
    "payload": {
        "connPort": 1
        "txState": 1,
        "attenuator": 0,
        "txLoFreqIndex": 0,
        "muteIoPolarity": 0
    },
    "status": 1
}
```

5.4.3. POST /api/v1/asset/buc/config/tx_state

Description

This API is used to set BUC tx on/off.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
txState	Integer	0: STANDBY 1: TRANSMIT

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/buc/config/tx_state

```
{
    "connPort": 1,
    "txState": 1
}
```

Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
txState	Integer	0: STANDBY 1: TRANSMIT

Response Example

```
{
    "payload": {
        "connPort": 1,
        "txState": 1
    },
    "status": 1
}
```

5.4.4. POST /api/v1/asset/buc/config/amp_info

Description

This API is used to set the BUC Amp information in CPI BUC.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)

ampName	String	Amp name

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/buc/config/amp_info

```
{
    "connPort": 1,
    "ampName": "Amp"
}
```

Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
ampName	String	Amp name

Response Example

```
{
    "payload": {
        "connPort": 1,
        "ampName": "Amp"
    },
    "status": 1
}
```

5.4.5. POST /api/v1/asset/buc/config/setpoint

Description

This API is used to set the BUC configuration in CPI BUC.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
highRfOutAlarm	Integer	0 ~ 90
highRfOutFault	Integer	0 ~ 90
alcSetpoint	Integer	0 ~ 90
manualRfSetpoint	Integer	0 ~ 90
attenuatorSet	Integer	0 ~ 90
disableAlcSetpoint	Integer	0 ~ 90

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/buc/config/setpoint

```
{
    "connPort": 1,
    "highRfOutAlarm": 44,
    "highRfOutFault": 45,
    "alcSetpoint": 41,
    "manualRfSetpoint": 38.9,
    "attenuatorSet": 0.9,
    "disableAlcSetpoint": 19.9
}
```


Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
highRfOutAlarm	Integer	0 ~ 90
highRfOutFault	Integer	0 ~ 90
alcSetpoint	Integer	0 ~ 90
manualRfSetpoint	Integer	0 ~ 90
attenuatorSet	Integer	0 ~ 90
disableAlcSetpoint	Integer	0 ~ 90

Response Example


```
{
    "payload": {
        "connPort": 1,
        "highRfOutAlarm": 44,
        "highRfOutFault": 45,
        "alcSetpoint": 41,
        "manualRfSetpoint": 38.9,
        "attenuatorSet": 0.9,
        "disableAlcSetpoint": 19.9
    },
    "status": 1
}
```


5.4.6. POST /api/v1/asset/buc/config/meo

Description

This API is used to set the BUC configuration in CPI BUC.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
unlockType	Integer	0: Alarm 1: Fault
bandSelect	Integer	0 ~ 3

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/buc/config/meo

```
{
    "connPort": 1,
    "unlockType": 0,
    "bandSelect": 0
}
```

Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
unlockType	Integer	0: Alarm 1: Fault
bandSelect	Integer	0 ~ 3

Response Example

```
{
    "payload": {
        "connPort": 1,
        "unlockType": 0,
        "bandSelect": 0
    },
    "status": 1
}
```

5.4.7. POST /api/v1/asset/buc/config/ip

Description

This API is used to set the BUC IP in CPI BUC.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
ipAddress	String	IP address of CPI BUC
netmask	String	Netmask
gateway	String	Gateway

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/buc/config/ip

```
{
    "connPort": 1,
    "ipAddress": "172.22.1.38",
    "netmask": "255.255.255.128",
    "gateway": "172.22.1.1"
}
```


Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
ipAddress	String	IP address of CPI BUC
netmask	String	Netmask
gateway	String	Gateway

Response Example


```
{
    "payload": {
        "connPort": 1,
        "ipAddress": "172.22.1.38",
        "netmask": "255.255.255.128",
        "gateway": "172.22.1.1"
    },
    "status": 1
}
```


5.4.8. POST /api/v1/asset/buc/config/clock

Description

This API is used to set the NTP server in CPI BUC.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
ntpServer	String	IP address of NTP server
ntpEnable	Integer	0: Disable 1: Enable

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/buc/config/clock


```
{
    "connPort": 1,
    "ntpServer": "192.43.244.18",
```

```
    "ntpEnable": 0
}
```

Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)
ntpServer	String	IP address of NTP server
ntpEnable	Integer	0: Disable 1: Enable

Response Example

```
{
    "payload": {
        "connPort": 1,
        "ntpServer": "192.43.244.18",
        "ntpEnable": 0
    },
    "status": 1
}
```

5.4.9. POST /api/v1/asset/buc/fault/reset

Description

This API is used to set the BUC fault reset in CPI BUC.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/buc/fault/reset

```
{
    "connPort": 1
}
```

Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connection port(1~8)

Response Example


```
{
    "payload": {
        "connPort": 1
    },
    "status": 1
}
```


5.4.10. POST /api/v1/asset/buc/config/tx_clock

Description

This API is used to set BUC tx reference clock configuration.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna port number, 1~8
band	Integer	[0:Ku, 1:C, 2:Ka, 3: X]
orbit	Integer	[0:GEO, 1:MEO, 2:LEO]
txClock	Integer	Tx Reference clock [0: Off, 1: 10MHz, 2: 50MHz]

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/buc/config/tx_clock


```
{
    "band": 2,
    "connPort": 1,
    "orbit": 1,
    "txClock": 1
}
```


Response Field

Key	Type	Descriptions
connPort	Integer	Antenna port number, 1~8
band	Integer	[0:Ku, 1:C, 2:Ka, 3: X]
orbit	Integer	[0:GEO, 1:MEO, 2:LEO]
txClock	Integer	Tx Reference clock [0: Off, 1: 10MHz, 2: 50MHz]

Response Example


```
{
    "payload": {
        "band": 2,
```

```
        "connPort": 1,
        "orbit": 1,
        "txClock": 1
    },
    "status": 1
}
```


5.5. DDC(Dual Data Center)

Method	URL	Descriptions
POST	/api/v1/mediator/ddc/config	Setting information about Dual Data Center
GET	/api/v1/mediator/ddc/config	Getting information about Dual Data Center

5.5.1. POST /api/v1/mediator/ddc/config

Description

This API is used to configure dual data center.

Request Field

Key	Type	Descriptions
connection	Integer	Connection status with peer dc [0: Peer DC is disconnected 1: Peer DC is connected]
mode	Integer	Dual data center mode [0: Disable, 1: Active-Standby, 2: Dual-active]
description	String	Data center description
name	String	Data center name
peerDcIp	String	IP address of peer dc
role	Integer	Role of Data center [0: None, 1: Host, 2: Remote]

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/ddc/config


```
{
    "connection": 1,
    "mode": 1,
    "description": "Dual Data Center",
    "name": "DC1",
    "peerDcIp": "10.1.105.228",
    "role": 1
}
```


Response Field

Key/Field	Type	Descriptions
connection	Integer	Connection status with peer dc [0: Peer DC is disconnected 1: Peer DC is connected]
mode	Integer	Dual data center mode [0: Disable, 1: Active-Standby, 2: Dual-active]
description	String	Data center description
name	String	Data center name
peerDcIp	String	IP address of peer dc
role	Integer	Role of Data center [0: None, 1: Host, 2: Remote]

Response Example


```
{
    "payload": {
        "connection": 1,
        "mode": 1,
        "description": "Dual Data Center",
        "name": "DC1",
        "peerDcIp": "10.1.105.228",
        "role": 1
    },
    "status": 1
}
```


5.5.2. GET /api/v1/mediator/ddc/config

Description

This API is used to get the dual data center information. General information about the dual data center like device status, role and peerIp can be retrieved.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/ddc/config

Response Field

Key/Field	Type	Descriptions
connection	Integer	Connection status with peer dc [0: Peer DC is disconnected 1: Peer DC is connected]
mode	Integer	Dual data center mode [0: Disable, 1: Active-Standby, 2: Dual-active]
description	String	Data center description
name	String	Data center name
peerDcIp	String	IP address of peer dc

role	Integer	Role of Data center [0: None, 1: Host, 2: Remote]

Response Example


```
{
    "payload": {
        "connection": 1,
        "mode": 1,
        "description": "Dual Data Center",
        "name": "DC1",
        "peerDcIp": "10.1.105.228",
        "role": 1
    },
    "status": 1
}
```


5.6. Information

Method	URL	Descriptions
GET	/api/v1/asset/info/all	Getting information about all asset

5.6.1. GET /api/v1/asset/info/all

Description

This API is used to retrieve all information about the mediator.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/asset/info/all

Response Field

Key / Field	Type	Descriptions
antenna	Object Array	[Refer to 5.1.4]
modem	Object	[Refer to 5.2.4]
group	Object	Currently NOT USED. Empty array
link	Object	[Refer to 5.3.2]
mediator	Object	Information for Mediator
modemAggregation	Object	[Refer to 5.7.22]
peerInfo	Object	Information for Remote DC (Host DC Only)

mediator Object

Key / Field	Type	Descriptions
dualDataCenter	Object	Object for Dual Data Center
handoverSchedule	Object array	Object for MEO Handover Schedule
headingDevice	Object	Object for Heading Device
mediatorMode	Object	Object for Mediator Mode
systemInfo	Object	Object for Temperature,Pan,Power of System
systemTime	Object	Object for System Time

dualDtaCenter Object

Key / Field	Type	Descriptions
connection	Integer	Connection status with peer dc [0: Peer DC is disconnected 1: Peer DC is connected]
dcActivate	Integer	0: deactivate, 1: activate [Currently NOT USED]
description	String	Data center description
id	Integer	ID of Data center
mode	Integer	Dual data center mode [0: Disable, 1: Active-Standby, 2: Dual-active]
name	String	Data center name
peerDcId	Integer	ID of Peer Data center
peerDcIp	String	IP address of peer dc
peerDcName	String	Peer Data center name
role	Integer	Role of Data center [0: None, 1: Host, 2: Remote]
autoRoleSwitch	Integer	Currently NOT USED
autoRoleSwitchCountdown	Integer	Currently NOT USED
autoRoleSwitchTimer	Integer	Currently NOT USED

handoverSchedule Object

Key / Field	Type	Descriptions
connPort	Integer	Modem port number(1~8)
countdown	String	Remaining time to handover, format "hh:mm:ss"
handoverTime	String	handover time, format "HH:mm:ss yyyy-MM-dd"
satelliteName	String	Satellite Name which is handover to. In mPOWER modem, after HO time, display "Awaiting Next Handover" until the next HO.
startTime	String	start of handover operation, format "HH:mm:ss yyyy-MM-dd"
lastHandoverTime	Integer	The most recent Handover time. (Displayed only on mPOWER modem.)

headingDevice Object

Key / Field	Type	Descriptions
angle	Number	0 ~ 360 (step 0.1)
device	Integer	0: None 1: NMEA

		7: Static 8: NMEA2000
speed	Integer	Baud rate of heading device 0: 4800 1: 9600 2: 19200 3: 38400
status	Integer	Connection status of heading device 0: Disconnect 1: Connect

mediatorMode Object

Key / Field	Type	Descriptions
medMesMode	Integer	[0: MES mode, 1: Standalone mode] Currently NOT USED
medMesStatus	Integer	Currently NOT USED
operationState	Integer	Operation mode of Mediator 0: None 1: Setup mode 2: Automatic mode

systemInfo Object

Key / Field	Type	Descriptions
mcuTemperature1	Integer	Temperature of Main board
mcuTemperature2	Integer	Temperature of Main board
panMonitoring	Integer	0: Pan is off 1: Pan is on
powerVoltage1	Integer	Voltage of Power 1
powerVoltage2	Integer	Voltage of Power 2
rxTemperature	Integer	Temperature of Rx RF board
txTemperature	Integer	Temperature of Tx RF board

systemTime Object

Key / Field	Type	Descriptions
dateTime	String	System time, hh:mm:ss YYYY-MM-DD format
timestamp	Integer	System time, unit time format

Response Example


```
{
    "payload": {
        "antenna": [],
        "modem": [],
        "group": [],
        "link": [],
```

```
        "modemAggregation": [],
        "mediator": {
            "dualDataCenter": {
                "connection": 1,
                "dcActivate": 1,
                "description": "Dual Data Center",
                "mode": 2,
                "name": "DC1",
                "id": 1,
                "peerDcId": 2,
                "peerDcIp": "10.1.105.228",
                "peerDcName": "DC2",
                "role": 1,
                "autoRoleSwitch": 0,
                "autoRoleSwitchCountdown": 0,
                "autoRoleSwitchTimer": 300
            },
            "handoverSchedule": [
                {
                    "connPort": 1,
                    "countdown": "00:15:48",
                    "handoverTime": "00:49:10 2025-04-10",
                    "satelliteName": "O3B M008",
                    "startTime": "00:48:10 2025-04-10"
                },
                {
                    "connPort": 2,
                    "countdown": "00:15:48",
                    "handoverTime": "00:49:10 2025-04-10",
                    "satelliteName": "O3B M008",
                    "startTime": "00:48:10 2025-04-10"
                },
                {
                    "connPort": 3,
                    "countdown": "",
                    "handoverTime": "",
                    "satelliteName": "",
                    "startTime": ""
                },
                {
                    "connPort": 4,
                    "countdown": "",
                    "handoverTime": "",
                    "lastHandoverTime": "07:45:00 2025-01-15",
                    "satelliteName": "Awaiting Next Handover",
                    "startTime": ""
                },
                {
                    "connPort": 5,
                    "countdown": "",
```

```
                "handoverTime": "",
                "lastHandoverTime": "07:45:00 2025-01-15",
                "satelliteName": "Awaiting Next Handover",
                "startTime": ""
            },
            {
                "connPort": 6,
                "countdown": "",
                "handoverTime": "",
                "satelliteName": "",
                "startTime": ""
            },
            {
                "connPort": 7,
                "countdown": "",
                "handoverTime": "",
                "lastHandoverTime": "",
                "satelliteName": "Awaiting Next Handover",
                "startTime": ""
            },
            {
                "connPort": 8,
                "countdown": "",
                "handoverTime": "",
                "lastHandoverTime": "",
                "satelliteName": "Awaiting Next Handover",
                "startTime": ""
            },
        ],
        "headingDevice": {
            "angle": 38.29,
            "device": 1,
            "speed": 0,
            "status": 0
        },
        "mediatorMode": {
            "medMesMode": 1,
            "medMesStatus": 1,
            "operationState": 1,
        },
        "systemInfo": {
            "mcuTemperature1": 32.5,
            "mcuTemperature2": 31.5,
            "panMonitoring": 1,
            "powerVoltage1": 23.76,
            "powerVoltage2": 0.01,
            "rxTemperature": 31.5,
            "txTemperature": 31.5,
        },
        "systemTime": {
```

```
                "dateTime": "00:27:08 2023-06-14",
                "timestamp": 1686702428
            }
        },
    },
    "status": 1
}
```


5.7. Mediator Configuration

Method	URL	Descriptions
GET	/api/v1/mediator/search/rule/mg_adjust	Get MEO/GEO option
POST	/api/v1/mediator/search/rule/mg_adjust	Set MEO/GEO option
GET	/api/v1/mediator/search/rule/change_role	Get role option
POST	/api/v1/mediator/search/rule/change_role	Set role option
GET	/api/v1/mediator/group/if_test_mode	Get buc test mode settings
POST	/api/v1/mediator/group/if_test_mode	Set buc test mode
GET	/api/v1/sdb/file/download/ephemeris	Download ephemeris of SDB files
GET	/api/v1/sdb/file/download/spacecraft	Download spacecraft of SDB files
GET	/api/v1/sdb/file/download/schedule	Download schedule of SDB files
GET	/api/v1/sdb/file/download/channels	Download channels of SDB files
POST	/api/v1/sdb/file/upload/ephemeris	Upload ephemeris of SDB files
POST	/api/v1/sdb/file/upload/spacecraft	Upload spacecraft of SDB files
POST	/api/v1/sdb/file/upload/schedule	Upload schedule of SDB files
POST	/api/v1/sdb/file/upload/channels	Upload channels of SDB files
GET	/api/v1/status/sdb	Get status of SDB files
GET	/api/v1/mediator/ifl/gain/user	Get gain values of IFL balancing
POST	/api/v1/mediator/ifl/gain/user	Set gain value of IFL balancing
GET	/api/v1/mediator/ifl/gain/user/download	Download IFL gain config file
POST	/api/v1/mediator/ifl/gain/user/upload	Restore from IFL gain config file
GET	/api/v1/mediation/policy/service/priority	Getting information about modem service priority
POST	/api/v1/mediation/policy/service/priority	Setting information about modem service priority
GET	/api/v1/mediation/policy/modem/aggregation	Getting information about modem aggregation
POST	/api/v1/mediation/policy/modem/aggregation	Setting information about modem aggregation
GET	/api/v1/mediator/reference_out/config	Get external reference out configuration
POST	/api/v1/mediator/reference_out/config	Set external reference out configuration
GET	/api/v1/ship/heading	Get heading device settings
POST	/api/v1/ship/heading	Set heading device setting
GET	/api/v1/mediator/network/ip	Get network IP settings of mediator
POST	/api/v1/mediator/network/ip	Set network IP settings of mediator

GET	/api/v1/mediator/network/management_interface	Get management interface configure
POST	/api/v1/mediator/network/management_interface	Set management interface configure
GET	/api/v1/mediator/network/service	Get network servfice configuration
POST	/api/v1/mediator/network/service	Set network servfice configuration

5.7.1. GET /api/v1/mediator/search/rule/mg_adjust

Description

This API is used to get MEO/GEO option values.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/search/rule/mg_adjust

Response Field

Key	Type	Descriptions
esnoThreshold	Integer	Threshold of EsNo delta for role switch
switchBackTimer	Integer	Timeout for switch back
switchOutTimer	Integer	Timeout for switch out
priorityModemPort	Array	Currently NOT USED in IM8 T2

Response Example


```
{
    "payload": {
        "esnoThreshold": 5,
        "switchBackTimer": 60,
        "switchOutTimer": 60,
        "priorityModemPort": []
    },
    "status": 1
}
```


5.7.2. POST /api/v1/mediator/search/rule/mg_adjust

Description

This API is used to set MEO/GEO option.

Request Field

Key	Type	Descriptions
esnoThreshold	Integer	Threshold of EsNo delta for role switch

switchBackTimer	Integer	Timeout for switch back
switchOutTimer	Integer	Timeout for switch out
groupId	Integer	Always “1” in IM8 T2
priorityModemPort	Array	Currently NOT USED in IM8 T2

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/search/rule/mg_adjust

```
{
    "esnoThreshold": 5,
    "switchBackTimer": 60,
    "switchOutTimer": 60,
    "groupId": 1,
    "priorityModemPort": []
}
```

Response Field

Key	Type	Descriptions
esnoThreshold	Integer	Threshold of EsNo delta for role switch
switchBackTimer	Integer	Timeout for switch back
switchOutTimer	Integer	Timeout for switch out
priorityModemPort	Array	Currently NOT USED in IM8 T2

Response Example

```
{
    "payload": {
        "esnoThreshold": 5,
        "switchBackTimer": 60,
        "switchOutTimer": 60,
        "priorityModemPort": []
    },
    "status": 1
}
```

5.7.3. GET /api/v1/mediator/search/rule/change_role

Description

This API is used to get role option values.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/search/rule/change_role

Response Field

Key	Type	Descriptions
modemLockBroadcast	Integer	Modem lock broadcast [0: Off, 1: On]
roleSwitchByUnlock	Integer	Current NOT USED
roleSwitchByUnlockStatus	Integer	Current NOT USED
signalThreshold	Integer	Threshold for role switch by signal difference
timeoutRole	Integer	Timeout for role switch
useRoleSwitchCondition	Object	Role switch options for mPOWER modem

useRoleSwitchCondition Object

Key	Type	Descriptions
antennaEvent	Boolean	Option for role switch by antenna is not tracking [True: Enable, False: Disable]
antennaFailure	Boolean	Option for role switch by antenna fault [True: Enable, False: Disable]
bucFailure	Boolean	Option for role switch by BUC fault [True: Enable, False: Disable]
demodUnlock	Boolean	Option for role switch by Demod unlocked [True: Enable, False: Disable]
esno	Boolean	Option for role switch by EsNo delta [True: Enable, False: Disable]
inBlockage	Boolean	Option for role switch by Known blockage [True: Enable, False: Disable]

Response Example


```
{
    "payload": {
        "timeoutRole": 5,
        "signalThreshold": 60,
        "modemLockBroadcast": 60,
        "roleSwitchByUnlock": 60,
        "roleSwitchByUnlockStatus": 60,
        "useRoleSwitchCondition": {
            "antennaEvent": true,
            "antennaFailure": true,
            "bucFailure": true,
            "demodUnlock": true,
            "esno": true,
            "inBlockage": true
        }
    },
    "status": 1
}
```

5.7.4. POST /api/v1/mediator/search/rule/change_role

Description

This API is used to set role option.

Request Field

Key	Type	Descriptions
modemLockBroadcast	Integer	Modem lock broadcast [0: Off, 1: On]
roleSwitchByUnlock	Integer	Current NOT USED
roleSwitchByUnlockStatus	Integer	Current NOT USED
signalThreshold	Integer	Threshold for role switch by signal difference
timeoutRole	Integer	Timeout for role switch
useRoleSwitchCondition	Object	Role switch options for mPOWER modem

useRoleSwitchCondition Object

Key	Type	Descriptions
antennaEvent	Boolean	Option for role switch bye antenna is not tracking [True: Enable, False: Disable]
bucFailure	Boolean	Option for role switch by BUC fault [True: Enable, False: Disable]
demodUnlock	Boolean	Option for role switch by Demod unlocked [True: Enable, False: Disable]
esno	Boolean	Option for role switch by EsNo delta [True: Enable, False: Disable]
inBlockage	Boolean	Option for role switch by Known blockage [True: Enable, False: Disable]

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/search/rule/change_role

```
{
    "timeoutRole": 5,
    "signalThreshold": 30,
    "modemLockBroadcast": 0,
    "roleSwitchByUnlock": 0,
    "roleSwitchByUnlockStatus": 0,
    "groupId": 1,
    "useRoleSwitchCondition": {
        "antennaEvent": true,
        "inBlockage": true,
        "demodUnlock": true,
        "esno": true,
        "bucFailure": true
    }
}
```

Response Field

Key	Type	Descriptions
modemLockBroadcast	Integer	Modem lock broadcast [0: Off, 1: On]
roleSwitchByUnlock	Integer	Current NOT USED
roleSwitchByUnlockStatus	Integer	Current NOT USED
signalThreshold	Integer	Threshold for role switch by signal difference
timeoutRole	Integer	Timeout for role switch
useRoleSwitchCondition	Object	Role switch options for mPOWER modem

useRoleSwitchCondition Object

Key	Type	Descriptions
antennaEvent	Boolean	Option for role switch by antenna is not tracking [True: Enable, False: Disable]
antennaFailure	Boolean	Option for role switch by antenna fault [True: Enable, False: Disable]
bucFailure	Boolean	Option for role switch by BUC fault [True: Enable, False: Disable]
demodUnlock	Boolean	Option for role switch by Demod unlocked [True: Enable, False: Disable]
esno	Boolean	Option for role switch by EsNo delta [True: Enable, False: Disable]
inBlockage	Boolean	Option for role switch by Known blockage [True: Enable, False: Disable]

Response Example


```
{
    "payload": {
        "timeoutRole": 5,
        "signalThreshold": 60,
        "modemLockBroadcast": 60,
        "roleSwitchByUnlock": 60,
        "roleSwitchByUnlockStatus": 60,
        "useRoleSwitchCondition": {
            "antennaEvent": true,
            "antennaFailure": true,
            "bucFailure": true,
            "demodUnlock": true,
            "esno": true,
            "inBlockage": true
        }
    },
    "status": 1
}
```


5.7.5. GET /api/v1/mediator/group/if_test_mode

Description

This API is used to get the buc test mode settings.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/group/if_test_mode

Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port(1~8) If BUC test mode is 0(off), this value should be 0
meoBucTestMode	Integer	BUC test mode 0: Off 1: Auto 2: Manual 3: Unmute 4: Mute

Response Example


```
{
    "payload": {
        "connPort": 0,
        "meoBucTestMode": 0
    },
    "status": 1
}
```


5.7.6. POST /api/v1/mediator/group/if_test_mode

Description

This API is used to set the buc test mode.

Request Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port(1~8) If BUC test mode is 0(off), this value should be 0
meoBucTestMode	Integer	BUC test mode 0: Off 1: Auto 2: Manual 3: Unmute 4: Mute

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/group/if_test_mode

```
{
    "connPort": 1,
    "meoBucTestMode": 3
}
```


Response Field

Key	Type	Descriptions
connPort	Integer	Antenna connetion port(1~8) If BUC test mode is 0(off), this value should be 0
meoBucTestMode	Integer	BUC test mode 0: Off 1: Auto 2: Manual 3: Unmute 4: Mute

Response Example


```
{
    "payload": {
        "connPort": 1,
        "meoBucTestMode": 3
    },
    "status": 1
}
```


5.7.7. GET /api/v1/sdb/file/download/ephemeris

Description

This API is used to download the ephemeris of SDB files for the O3b classic service.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/sdb/file/download/ephemeris

Response Field

Key	Type	Descriptions
path	String	Path of the SDB file to download

Response Example


```
{
```

```
    "payload": {
        "path": "/usr/local/download/system-data/ephemeris"
    },
    "status": 1
}
```

5.7.8. GET /api/v1/sdb/file/download/spacecraft

Description

This API is used to download the spacecraft of SDB files for the O3b classic service.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/sdb/file/download/spacecraft

Response Field

Key	Type	Descriptions
path	String	Path of the SDB file to download

Response Example

```
{
    "payload": {
        "path": "/usr/local/download/system-data/spacecraft"
    },
    "status": 1
}
```

5.7.9. GET /api/v1/sdb/file/download/schedule

Description

This API is used to download the schedule of SDB files for the O3b classic service.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/sdb/file/download/schedule

Response Field

Key	Type	Descriptions
path	String	Path of the SDB file to download

Response Example

```
{
    "payload": {
        "path": "/usr/local/download/system-data/schedule"
    },
    "status": 1
}
```

5.7.10. GET /api/v1/sdb/file/download/channels

Description

This API is used to download the channels of SDB files for the O3b classic service.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/sdb/file/download/channels

Response Field

Key	Type	Descriptions
path	String	Path of the SDB file to download

Response Example

```
{
    "payload": {
        "path": "/usr/local/download/system-data/channels"
    },
    "status": 1
}
```

5.7.11. POST /api/v1/sdb/file/upload/ephemeris

Description

This API is used to upload the ephemeris of SDB files for the O3b classic service.

Request Field

Content-type : multipart/form-data

Key / Field	Type	Descriptions
filename	String	SDB file

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/sdb/file/upload/ephemeris

```
------WebKitFormBoundary6fmMcpJSu6jidtrU
Content-Disposition: form-data; name="filename"; filename="ephemeris"
Content-Type: application/octet-stream


------WebKitFormBoundary6fmMcpJSu6jidtrU--
```

Response Field

Key / Field	Type	Descriptions
payload	String	Response message
status	Integer	Response status code [0:Error, 1:Success]

Response Example

```
{
    "payload": "success",
    "status": 1
}
```

5.7.12. POST /api/v1/sdb/file/upload/spacecraft

Description

This API is used to upload the spacecraft of SDB files for the O3b classic service.

Request Field

Content-type : multipart/form-data

Key / Field	Type	Descriptions
filename	String	SDB file

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/sdb/file/upload/spacecraft

```
------WebKitFormBoundaryB6L4VO1S7tNJViVL
Content-Disposition: form-data; name="filename"; filename="spacecraft"
```

```
Content-Type: application/octet-stream



------WebKitFormBoundaryB6L4VO1S7tNJViVL--
```

Response Field

Key / Field	Type	Descriptions
payload	String	Response message
status	Integer	Response status code [0:Error, 1:Success]

Response Example

```
{
    "payload": "success",
    "status": 1
}
```

5.7.13. POST /api/v1/sdb/file/upload/schedule

Description

This API is used to upload the schedule of SDB files for the O3b classic service.

Request Field

Content-type : multipart/form-data

Key / Field	Type	Descriptions
filename	String	SDB file

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/sdb/file/upload/schedule

```
------WebKitFormBoundaryTbjzYbtEBF2Sxqid
Content-Disposition: form-data; name="filename"; filename="schedule"
Content-Type: application/octet-stream



------WebKitFormBoundaryTbjzYbtEBF2Sxqid--
```

Response Field

Key / Field	Type	Descriptions
payload	String	Response message

status	Integer	Response status code [0:Error, 1:Success]

Response Example

```
{
    "payload": "success",
    "status": 1
}
```

5.7.14. POST /api/v1/sdb/file/upload/channels

Description

This API is used to upload the channels of SDB files for the O3b classic service.

Request Field

Content-type : multipart/form-data

Key / Field	Type	Descriptions
filename	String	SDB file

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/sdb/file/upload/channels

```
------WebKitFormBoundaryxHczPXp3MbaFOSb9
Content-Disposition: form-data; name="filename"; filename="channels"
Content-Type: application/octet-stream


------WebKitFormBoundaryxHczPXp3MbaFOSb9--
```

Response Field

Key / Field	Type	Descriptions
payload	String	Response message
status	Integer	Response status code [0:Error, 1:Success]

Response Example

```
{
    "payload": "success",
    "status": 1
}
```

5.7.15. GET /api/v1/status/sdb

Description

This API is used to get the status of an SDB files.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/status/sdb

Response Field

Key	Type	Descriptions
date	String	Date the SDB files was uploaded
name	String	Name of SDB files
size	String	Size of SDB files (Bytes)

Response Example


```
{
    "payload": [
        {
            "date": "2025/04/14",
            "name": "ephemeris",
            "size": "3232"
        },
        {
            "date": "2025/04/14",
            "name": "spacecraft",
            "size": "3350"
        },
        {
            "date": "2025/04/14",
            "name": "schedule",
            "size": "4678"
        },
        {
            "date": "2025/04/14",
            "name": "channels",
            "size": "1055"
        },
    ],
    "status": 1
}
```

5.7.16. GET /api/v1/mediator/ifl/gain/user

Description

This API is used to get the Tx, Rx gain values of IFL balancing.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/ifl/gain/user

Response Field

Key	Type	Descriptions
rxGains	Array	Rx gain per antenna
rxGainsEach	Array	The Rx gain value is organized in an 8(Antenna) x 8(Modem) matrix format.
txGains	Array	Tx gain per antenna
txGainsEach	Array	The Tx gain value is organized in an 8(Antenna) x 8(Modem) matrix format.

Response Example

```
{
    "payload": {
        "rxGains": [0, 0, 0, 0, 0, 0, 0, 0],
        "rxGainsEach": [
            [0, 0, 0, 0, 0, 0, 0, 0],   // ANT1: Modem1 ~ Modem8
            [0, 0, 0, 0, 0, 0, 0, 0],   // ANT2
            [0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0]
        ],
        "txGains": [0, 0, 0, 0, 0, 0, 0, 0]
        "txGainsEach": [
            [0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0]
        ]
    },
    "status": 1
```

```
}
```

5.7.17. POST /api/v1/mediator/ifl/gain/user

Description

This API is used to set the Tx, Rx gain value of IFL balancing.

Request Field

Key	Type	Descriptions
direction	Integer	0: RX 1: TX
gains	Array	Object of gains

gains Object

Key	Type	Descriptions
antennaPort	Integer	Antenna connetion port(1~8)
gain	Integer	Gain value Rx Range : -20.0 ~ 10 (Step 0.5) Tx Range : -16.5 ~ 15 (Step 0.5)
modemPort	Integer	Modem connetion port(1~8)

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/ifl/gain/user

```
{
    "direction": 1,
    "gains": [
        {
            "modemPort": 1,
            "antennaPort": 1,
            "gain": 2
        }
    ]
}
```

Response Field

Key	Type	Descriptions
direction	Integer	0: RX 1: TX
gains	Array	Object of gains

gains Object

Key	Type	Descriptions
antennaPort	Integer	Antenna connetion port(1~8)
gain	Integer	Gain value Rx Range : -20.0 ~ 10 (Step 0.5)

		Tx Range : -16.5 ~ 15 (Step 0.5)
modemPort	Integer	Modem connetion port(1~8)

Response Example


```
{
    "payload": {
        "direction": 1,
        "gains": [
            {
                "modemPort": 1,
                "antennaPort": 1,
                "gain": 2
            }
        ]
    },
    "status": 1
}
```


5.7.18. GET /api/v1/mediator/ifl/gain/user/download

Description

This API is used to download the Tx, Rx gain values of IFL balancing.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/ifl/gain/user/download

Response Field

Key	Type	Descriptions
path	String	Path of the IFL gain file to download

Response Example


```
{
    "payload": {
        "path": "/usr/local/download/backup/ifl_gain.cfg"
    },
    "status": 1
}
```

5.7.19. POST /api/v1/mediator/ifl/gain/user/upload

Description

This API is used to restores ifl gains using the files saved.

Request Field

Content-type : multipart/form-data

Key / Field	Type	Descriptions
filename	String	IFL gain file

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/ifl/gain/user/upload

```
------WebKitFormBoundaryS0SQlByT3ujoBZuF
Content-Disposition: form-data; name="filename"; filename="ifl_gain.cfg"
Content-Type: application/octet-stream


------WebKitFormBoundaryS0SQlByT3ujoBZuF--
```

Response Field

Key / Field	Type	Descriptions
payload	String	Response message
status	Integer	Response status code [0:Error, 1:Success]

Response Example

```
{
    "payload": "success",
    "status": 1
}
```

5.7.20. GET /api/v1/mediation/policy/service/priority

Description

This API is used to get the modem service priority.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediation/policy/service/priority

Response Field

Key/Field	Type	Descriptions
connPort	Integer	Modem port number (1~8)
band	Integer	[0:Ku, 1:C, 2:Ka, 3: X]
dc	Integer	Data center number
minConnect	Integer	Minimum count of antennas that must be connected for modem service
orbit	Integer	[0:GEO, 1:MEO, 2:LEO]
priority	Integer	Service priority

Response Example


```
{
    "payload": {
        "priority": [
            {
                "band": 2,
                "connPort": 3,
                "dc": 1,
                "minConnect": 2,
                "orbit": 1,
                "priority": 0
            },
            {
                "band": 2,
                "connPort": 3,
                "dc": 1,
                "minConnect": 1,
                "orbit": 0,
                "priority": 1
            }
        ]
    },
    "status": 1
}
```


5.7.21. POST /api/v1/mediation/policy/service/priority

Description

This API is used to set the modem service priority.

Request Field

Key	Type	Descriptions
connPort	Integer	Modem port number (1~8)
band	Integer	[0:Ku, 1:C, 2:Ka, 3: X]

dc	Integer	Data center number
minConnect	Integer	Minimum count of antennas that must be connected for modem service
orbit	Integer	[0:GEO, 1:MEO, 2:LEO]
priority	Integer	Service priority

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediation/policy/service/priority

```
{
    "priority": [
        {
            "band": 2,
            "connPort": 3,
            "dc": 1,
            "minConnect": 2,
            "orbit": 1,
            "priority": 0
        },
        {
            "band": 2,
            "connPort": 3,
            "dc": 1,
            "minConnect": 1,
            "orbit": 0,
            "priority": 1
        }
    ]
}
```

Response Field

Key	Type	Descriptions
connPort	Integer	Modem port number (1~8)
band	Integer	[0:Ku, 1:C, 2:Ka, 3: X]
dc	Integer	Data center number
minConnect	Integer	Minimum count of antennas that must be connected for modem service
orbit	Integer	[0:GEO, 1:MEO, 2:LEO]
priority	Integer	Service priority

Response Example

```
{
    "payload": {
        "priority": [
            {
                "band": 2,
```

```
                "connPort": 3,
                "dc": 1,
                "minConnect": 2,
                "orbit": 1,
                "priority": 0
            },
            {
                "band": 2,
                "connPort": 3,
                "dc": 1,
                "minConnect": 1,
                "orbit": 0,
                "priority": 1
            }
        ]
    },
    "status": 1
}
```


5.7.22. GET /api/v1/mediation/policy/modem/aggregation

Description

This API is used to get the modem aggregation information.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediation/policy/modem/aggregation

Response Field

Key/Field	Type	Descriptions
id	Integer	Index of Modem aggregation configuration
main	Integer	Main Modem port number
mainDc	Integer	Data center id of Main Modem
mainName	String	Name of Main Modem
sub	Number	Sub Modem port number
subDc	Integer	Data center id of Sub Modem
subName	String	Name of Sub Modem
desc	String	Description of Modem aggregation

Response Example


```
{
    "payload": {
        "modemAggregation": [
```

```
            {
                "id": 1,
                "main": 1,
                "mainDc": 1,
                "mainName": "Gilat A",
                "sub": 1,
                "subDc": 1,
                "subName": "Gilat B",
                "desc": "Modem Aggregation"
            }
        ]
    },
    "status": 1
}
```


5.7.23. POST /api/v1/mediation/policy/modem/aggregation

Description

This API is used to set the modem aggregation. User needs to set two modem for aggregation.

Request Field

Key	Type	Descriptions
id	Integer	Index of Modem aggregation configuration
main	Integer	Main Modem port number
mainDc	Integer	Data center id of Main Modem
mainName	String	Name of Main Modem
sub	Integer	Sub Modem port number
subDc	Integer	Data center id of Sub Modem
subName	String	Name of Sub Modem
desc	String	Description of Modem aggregation

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediation/policy/modem/aggregation


```
{
    "modemAggregation": [
        {
            "id": 1,
            "main": 1,
            "mainDc": 1,
            "mainName": "Gilat A",
            "sub": 3,
            "subDc": 1,
            "subName": "Gilat B",
            "desc": "Modem Aggregation"
        }
    ]
```

```
}
```

Response Field

Key	Type	Descriptions
id	Integer	Index of Modem aggregation configuration
main	Integer	Main Modem port number
mainDc	Integer	Data center id of Main Modem
mainName	String	Name of Main Modem
sub	Integer	Sub Modem port number
subDc	Integer	Data center id of Sub Modem
subName	String	Name of Sub Modem
desc	String	Description of Modem aggregation

Response Example


```
{
    "payload": {
        "modemAggregation": [
            {
                "id": 1,
                "main": 1,
                "mainDc": 1,
                "mainName": "Gilat A",
                "sub": 1,
                "subDc": 1,
                "subName": "Gilat B",
                "desc": "Modem Aggregation"
            }
        ]
    },
    "status": 1
}
```


5.7.24. GET /api/v1/mediator/reference_out/config

Description

This API is used to get external reference out configuration.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/reference_out/config

Response Field

Key/Field	Type	Descriptions
out1	Integer	REF OUT1 [0: Off, 1: 10MHz, 2: 50MHz]
out2	Integer	REF OUT2 [0: Off, 1: 10MHz, 2: 50MHz]

Response Example

```
{
    "payload": {
        "out1": 0,
        "out2": 0
    },
    "status": 1
}
```

5.7.25. POST /api/v1/mediator/reference_out/config

Description

This API is used to set external reference out configuration.

Request Field

Key	Type	Descriptions
out1	Integer	REF OUT1 [0: Off, 1: 10MHz, 2: 50MHz]
out2	Integer	REF OUT2 [0: Off, 1: 10MHz, 2: 50MHz]

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/reference_out/config

```
{
    "out1": 0,
    "out2": 0
}
```

Response Field

Key	Type	Descriptions
out1	Integer	REF OUT1 [0: Off, 1: 10MHz, 2: 50MHz]
out2	Integer	REF OUT2 [0: Off, 1: 10MHz, 2: 50MHz]

Response Example

```
{
    "payload": {
        "out1": 0,
        "out2": 0
    },
    "status": 1
}
```


5.7.26. GET /api/v1/ship/heading

Description

This API is used to get the heading device settings.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/ship/heading

Response Field

Key/Field	Type	Descriptions
angle	Integer	Heading angle
device	Integer	Heading device type 0: None 1: NMEA 0183 7: Static 8: NMEA 2000
sentence	String	Heading sentence
speed	Integer	Baud rate of heading device 0: 4800 1: 9600 2: 19200 3: 38400
status	Integer	Connection status of heading device [0: Disconnect, 1: Connect]

Response Example


```
{
    "payload": {
        "angle": 0,
        "device": 1,
        "sentence": "",
        "speed": 0,
        "status": 0
    },
    "status": 1
```

```
}
```

5.7.27. POST /api/v1/ship/heading

Description

This API is used to set the heading device.

Request Field

Key	Type	Descriptions
device	Integer	Heading device type 0: None 1: NMEA 0183 7: Static 8: NMEA 2000
speed	Integer	Baud rate of heading device 0: 4800 1: 9600 2: 19200 3: 38400
angle	Integer	Heading angle

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/ship/heading

```
{
    "device": 1,
    "speed": 0,
    "angle": 0
}
```

Response Field

Key/Field	Type	Descriptions
angle	Integer	Heading angle
device	Integer	Heading device type 1: NMEA 0183 8: NMEA 2000
speed	Integer	Baud rate of heading device 0: 4800 1: 9600 2: 19200 3: 38400

Response Example

```
{
    "payload": {
```

```
        "angle": 0,
        "device": 1,
        "speed": 0
    },
    "status": 1
}
```


5.7.28. GET /api/v1/mediator/network/ip

Description

This API is used to get the network IP settings of the mediator.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/network/ip

Response Field

Key/Field	Type	Descriptions
dnsserver	String	DNS server
gateway	String	Gateway of Mediator
ip	String	IP address of Mediator
subnetMask	String	Subnet mask of Mediator
tcpModemPort	Integer	TCP port (Currently NOT USED)
udpModemPort	Integer	UDP port (Currently NOT USED)

Response Example


```
{
    "payload": {
        "dnsserver": "168.126.63.1",
        "gateway": "10.1.105.254",
        "ip": "10.1.105.245",
        "subnetMask": "255.255.255.0",
        "tcpModemPort": 4001,
        "udpModemPort": 49184
    },
    "status": 1
}
```


5.7.29. POST /api/v1/mediator/network/ip

Description

This API is used to set the network IP settings of the mediator.

Request Field

Key	Type	Descriptions
dnsserver	String	DNS server
gateway	String	Gateway of Mediator
ip	String	IP address of Mediator
subnetMask	String	Subnet mask of Mediator
tcpModemPort	Integer	TCP port (Currently NOT USED)
udpModemPort	Integer	UDP port (Currently NOT USED)

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/network/ip

```
{
    "ip": "10.1.105.245",
    "subnetMask": "255.255.255.0",
    "gateway": "10.1.105.254",
    "dnsserver": "168.126.63.1",
    "tcpModemPort": 4001,
    "udpModemPort": 49184

}
```

Response Field

Key/Field	Type	Descriptions
dnsserver	String	DNS server
gateway	String	Gateway of Mediator
ip	String	IP address of Mediator
subnetMask	String	Subnet mask of Mediator
tcpModemPort	Integer	TCP port (Currently NOT USED)
udpModemPort	Integer	UDP port (Currently NOT USED)

Response Example

```
{
    "payload": {
        "dnsserver": "168.126.63.1",
        "gateway": "10.1.105.254",
        "ip": "10.1.105.245",
        "subnetMask": "255.255.255.0",
        "tcpModemPort": 4001,
        "udpModemPort": 49184
    },
    "status": 1
}
```

5.7.30. GET /api/v1/mediator/network/management_interface

Description

This API is used to get the management interface configuration.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/network/management_interface

Response Field

Key/Field	Type	Descriptions
ip	String	IP address of Management interface
leaseEndAddr	String	DHCP lease end IP address
leaseStartAddr	String	DHCP lease start IP address
leaseTime	Integer	Lease time (seconds)
subnetMask	String	Subnet mask of Management interface

Response Example

```
{
    "payload": {
        "ip": "192.168.2.1",
        "leaseEndAddr": "192.168.2.200",
        "leaseStartAddr": "192.168.2.101",
        "subnetMask": "255.255.255.0",
        "leaseTime": 604800
    },
    "status": 1
}
```

5.7.31. POST /api/v1/mediator/network/management_interface

Description

This API is used to set the management interface configuration.

Request Field

Key	Type	Descriptions
ip	String	IP address of Management interface
leaseEndAddr	String	DHCP lease end IP address
leaseStartAddr	String	DHCP lease start IP address
leaseTime	Integer	Lease time (seconds)
subnetMask	String	Subnet mask of Management interface

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/network/management_interface


```
{
    "ip": "192.168.2.1",
    "leaseEndAddr": "192.168.2.200",
    "leaseStartAddr": "192.168.2.101",
    "subnetMask": "255.255.255.0",
    "leaseTime": 604800

}
```


Response Field

Key/Field	Type	Descriptions
ip	String	IP address of Management interface
leaseEndAddr	String	DHCP lease end IP address
leaseStartAddr	String	DHCP lease start IP address
leaseTime	Integer	Lease time (seconds)
subnetMask	String	Subnet mask of Management interface

Response Example


```
{
    "payload": {
        "ip": "192.168.2.1",
        "leaseEndAddr": "192.168.2.200",
        "leaseStartAddr": "192.168.2.101",
        "subnetMask": "255.255.255.0",
        "leaseTime": 604800
    },
    "status": 1
}
```


5.7.32. GET /api/v1/mediator/network/service

Description

This API is used to get the network service configuration.

Request Field

None

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/network/service

Response Field

Key/Field	Type	Descriptions
httpEnable	Integer	HTTP enable 0 : HTTP connection not possible & no redirects to HTTPS 1 : HTTP redirects to HTTPS (DEFAULT)
httpsPort	Integer	HTTPS port (DEFAULT = 443)
icmpEnable	Integer	ICMP 0 : Ping not working 1 : Ping is working (DEFAULT)
ntpServerEnable	Integer	NTP Server [0 or 1(DEFAULT)]
rsyncEnable	Integer	Rsync [0 or 1(DEFAULT)]
sshPort	Integer	SSH port (DEFAULT = 22)
sshService	Integer	SSH Service 0 : SSH connection impossible 1 : SSH connection possible (DEFAULT)

Response Example


```
{
    "payload": {
        "httpEnable": 1,
        "httpsPort": 443,
        "icmpEnable": 1,
        "ntpServerEnable": 1,
        "rsyncEnable": 1,
        "sshPort": 22,
        "sshService": 1
    },
    "status": 1
}
```


5.7.33. POST /api/v1/mediator/network/service

Description

This API is used to set the network service configuration.

Request Field

Key	Type	Descriptions
httpEnable	Integer	HTTP enable 0 : HTTP connection not possible & no redirects to HTTPS 1 : HTTP redirects to HTTPS (DEFAULT)
httpsPort	Integer	HTTPS port (DEFAULT = 443)
icmpEnable	Integer	ICMP 0 : Ping not working 1 : Ping is working (DEFAULT)
ntpServerEnable	Integer	NTP Server [0 or 1(DEFAULT)]
rsyncEnable	Integer	Rsync [0 or 1(DEFAULT)]
sshPort	Integer	SSH port (DEFAULT = 22)

sshService	Integer	SSH Service 0 : SSH connection impossible 1 : SSH connection possible (DEFAULT)

Request Example

URL : {http|https}://{ServerIP}:{port}/api/v1/mediator/network/service

```
{
    "httpEnable": 1,
    "httpsPort": 443,
    "icmpEnable": 1,
    "ntpServerEnable": 1,
    "rsyncEnable": 1,
    "sshPort": 22,
    "sshService": 1
}
```

Response Field

Key/Field	Type	Descriptions
httpEnable	Integer	HTTP enable 0 : HTTP connection not possible & no redirects to HTTPS 1 : HTTP redirects to HTTPS (DEFAULT)
httpsPort	Integer	HTTPS port
icmpEnable	Integer	ICMP 0 : Ping not working 1 : Ping is working (DEFAULT)
ntpServerEnable	Integer	NTP Server [0 or 1(DEFAULT)]
rsyncEnable	Integer	Rsync [0 or 1(DEFAULT)]
sshPort	Integer	SSH port
sshService	Integer	SSH Service 0 : SSH connection impossible 1 : SSH connection possible (DEFAULT)

Response Example

```
{
    "payload": {
        "httpEnable": 1,
        "httpsPort": 443,
        "icmpEnable": 1,
        "ntpServerEnable": 1,
        "rsyncEnable": 1,
        "sshPort": 22,
        "sshService": 1
    },
    "status": 1
```

```
}
```