Linksys Smart WiFi Platform Developer SDK API Documentation Release 1.5

October 30th, 2012

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS” WITH ALL FAULTS. BELKIN AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL BELKIN OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

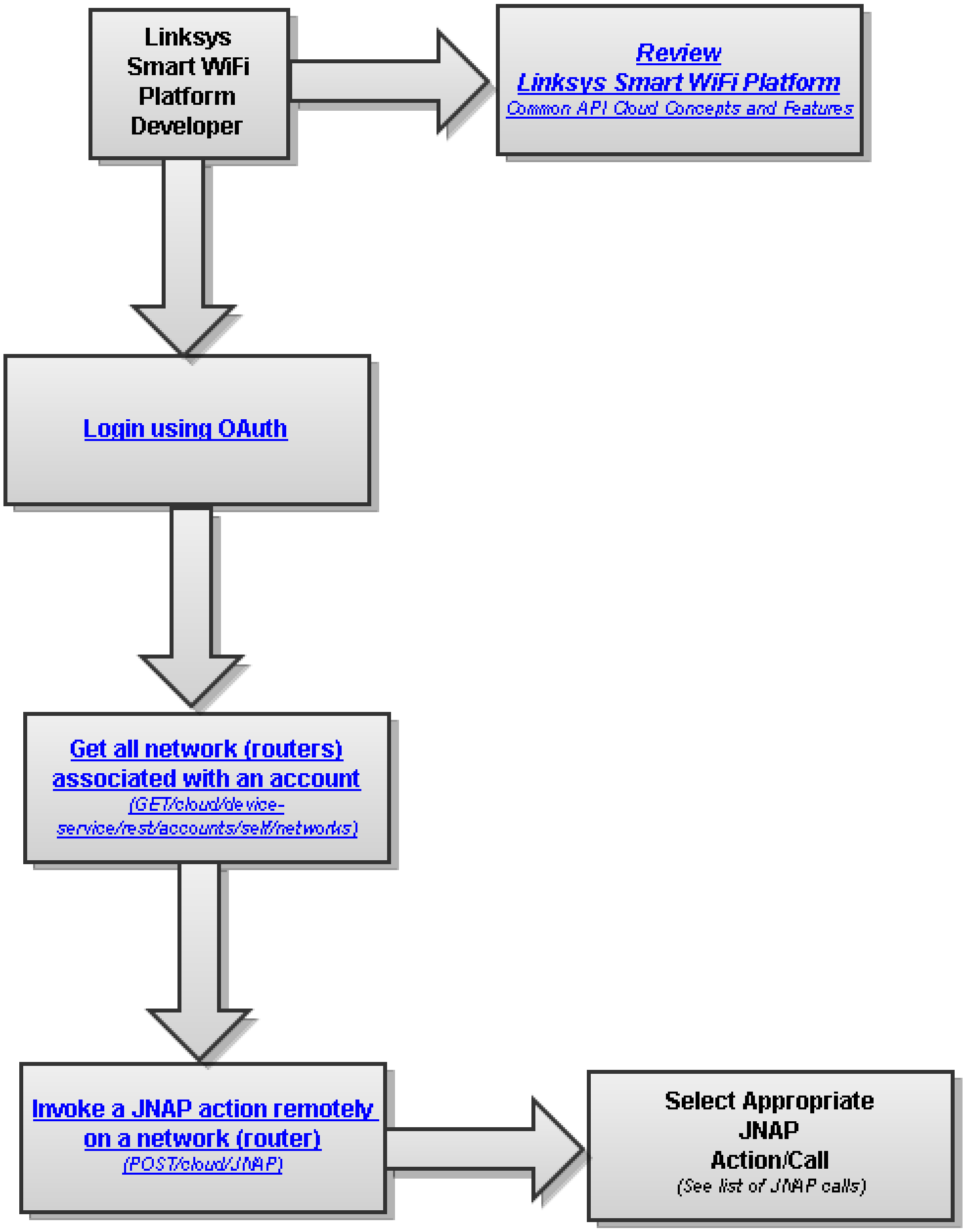
Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Modification History

Revision	Date	Originator	Major Revision	Comments
0.64	04/13/2012	Alex Roqueta	Y	Initial Pre Release DRAFT Document
0.68	04/20/2012	Alex Roqueta	N	Updated APIs
1.00	05/10/2012	Alex Roqueta	N	Added JNAP Matrix and APIs
1.01	05/18/2012	Alex Roqueta	N	Added OAuth documentation and APIs for: Get basic information of an account Get email address of an account
1.1	06/22/2012	Alex Roqueta	Y	Updated APIs to build 332 RC for Launch
1.2	08/11/2012	Alex Roqueta	Y	Updated Platform Overview to remove deprecated services
1.21	08/17/2012	Alex Roqueta	Y	Added DRAFT Port Forwarding APIs
1.22	08/20/2012	Alex Roqueta	Y	Updated DRAFT Device API notice
1.3	08/30/2012	Alex Roqueta	Y	Updated APIs to build 332 RC for Launch
1.4	10/04/2012	Alex Roqueta	Y	Updated Platform Overview to remove deprecated services
1.41	10/18/2012	Alex Roqueta	N	Added DRAFT APIs
1.42-1.46	10/30/2012	Alex Roqueta	N	Added and updated DRAFT APIs / Updated JNAP APIs
1.5	04/25/2013	Alex Roqueta	N	Branded to Belkin

Linksys Smart WiFi Developer SDK API Documentation

Platform Developer API Usage


Linksys Smart WiFi Platform Developer
Review
Linksys Smart WiFi Platform
Common API Cloud Concepts and Features
Login using OAuth
Get all network (routers) associated with an account
(GET/cloud/device-service/rest/accounts/self/networks)
Invoke a JNAP action remotely on a network (router)
(POST/cloud/JNAP)
Select Appropriate JNAP Action/Call
(See list of JNAP calls)

Linksys Smart WiFi Developer Platform Overview

Common API Concepts and Features

API Server Base URL

Production Environment

```
https://cloud.ciscoconnectcloud.com
```

Client Type ID

Every API request must include the Client Type ID.

The Client Type ID is a string that uniquely identifies a type of client/app making an API request.

Use the **X-Cisco-HN-Client-Type-Id** custom HTTP request header field to specify the Client Type ID as shown in the following example:

```
X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
```

Media Type

The Linksys Smart WiFi API supports JSON representation of the request/response data. The **Content-Type** and **Accept** HTTP header fields are used to specify the media type (MIME).

The client must use the **Content-Type** HTTP request header field to indicate the media type (JSON) of the entity-body sent to the server as described below.

```
Content-Type: application/json; charset=UTF-8
```

The client must use the **Accept** HTTP request header field to specify the media type (JSON) acceptable for the response as described below.

```
Accept: application/json
```

Pagination

A query or read API request may generate a result set that has hundreds or even thousands of items. It would not be a good idea to include all those items in a single API response; instead the Linksys Smart WiFi cloud supports the **startIndex** and the **maxItems** query parameters to determine which items to include in an API response.

startIndex
The **startIndex** parameter specifies the index of the first matching item that should be included in the API response. This parameter uses a zero-based index, meaning the first item is 0, the second item is 1 and so forth. This parameter works together with the **maxItems** parameter to determine which items to return. For example, to request the second set of 20 items -- i.e. items 21-40 -- set the **startIndex** parameter to 21 and the **maxItems** parameter to 20.

maxItems
The **maxItems** parameter specifies the maximum number of items that should be included in the API response. The default value and the maximum value allowed for this parameter depend on the specific API method.

Example:

```
https://cloud.ciscoconnectcloud.com/cloud/device-service/rest
/networks/network-id-goes-here/services/mediaservice/mediafeed/ALL_MUSIC?startIndex=0&maxItems=10
```

The API responses use the **totalItems**, **itemsReturned** and the **startIndex** fields to communicate the total number of items available in the result set, the number of items returned in the API response and the index of the first item returned:

totalItems
The **totalItems** field identifies the total number of items available in the result set.

itemsReturned
The **itemsReturned** field indicates the number of items returned in the API response.

startIndex
The **startIndex** field identifies the zero-based index of the first item returned.

The **totalItems**, **itemsReturned** and **startIndex** fields are wrapped in a parent object called **paginationResult** in the API responses as shown in the example below.

Example:
The following data indicate that an API response contains the 31st to 60th items of a total result set of 308 items.

```
"paginationResult":{
      "startIndex":0,
      "itemsReturned":3,
      "totalItems":3
   }
```

User Authentication (Login)

User authentication allows a client application to make the Linksys Smart WiFi API calls on behalf of a particular user account.

The end goal of the authentication process is to obtain a valid **access token** which can be used to make subsequent API calls.

A client can specify a **access token** in an API request using the **Authorization** HTTP request header field as shown in the example below:

```
Authorization: Bearer access-token-goes-here
```

A client uses the Linksys Smart WiFi OAuth 2.0 API to obtain an **access token**.

Error Response - Cloud API

When an API request fails, Linksys Smart WiFi API returns an HTTP 4xx or 5xx status code that generically identifies the failure along with a JSON response that provides more specific information about the error(s) that caused the failure.

For each **error**, the JSON response will include a **code** field and optionally a **message** field. An error response may contain more than one **error**.

Note: The content of the **message** field is for debugging purpose only and should not be displayed to the end user. The client is responsible for displaying an appropriate localized error meessage to the end user based on the error code returned.

If an API request is successful, no error related fields should be expected to be present in the response.

Example Response Body on Error

```
{
  "errors":[
    {
      "error":{
        "code":"UNKNOWN",
        "message":"unexpected error occurred",
        "parameters":[

        ]
      }
    }
  ]
}
```

Field Name	Description	Data Type	Required
code	A unique error code.	String	Yes
message	Error message	String	No
parameters	a list of parameter(s) associated with the error	List	No

Common Error Codes Returned - Cloud API

HTTP Status Code	Error Code	Error Code Meaning

400	INVALID_PARAMETER	Some of the element/query-parameter/header values specified in the request are not valid.
400	MISSING_PARAMETER	Some of the required element/query-parameter/header values are not specified (or missing) in the request.
401	INVALID_SESSION_TOKEN	The access token (aka session token) has expired or is not valid.
403	INVALID_CLIENT_TYPE	The client type ID is not valid.
403	ACCESS_DENIED	A resource could not be created / retrieved / updated / deleted because the client/account submitting the request was not authorized to do so.
500	UNKNOWN	An internal service/server error occurred. (e.g. database is down, etc.)
503	TEMPORARILY_UNAVAILABLE	The Linksys Smart WiFi is temporarily unavailable.

Error Response - JNAP Action

The HTTP status code and the format of the error response are different for a JNAP action (aks JNAP API call).

HTTP status code:
Always returns 200.

Error Response:
The error response conforms to the following format:

```
{
  "result":"error-code-goes-here",
  "error":"error-description-goes-here"
}
```

Note: The error (error description) returned is for debugging purpose only and should not be displayed to the end user. The client is responsible for displaying an appropriate localized error meessage to the end user based on the result (error code) returned.

Common Error Codes Returned - JNAP Action

HTTP Status Code	Error Code (result)	Error Code Meaning
200	_ErrorNetworkUnreachable	The target network (router) is not reachable.
200	_ErrorUnauthorized	The account making the request is not authorized to invoke the specified JNAP action on the specified network (router).
200	_ErrorUnexpected	An unexpected error occured.
200	_ErrorUnknownSession	The access token (aka session token) specified in the request header field is invalid or has expired.
200	_ErrorUnknownTarget	The target network (router) is not known.

Example Response Body on Error:

```
{
  "result":"_ErrorUnknownTarget",
  "error":"The target device/router is not known."
}
```

API Details

Account

Get basic information of an account

Purpose

Gets (aka reads or retrieves) the basic information (i.e. accountId, firstName, lastName, and middleName) of the currently logged-in account.

Request: URI and Headers

GET /cloud/user-service/rest/accounts/self/basicinfo

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, and **Accept** header fields are specified as shown above.

Here is an example of raw request headers:

```
GET /cloud/user-service/rest/accounts/self/basicinfo HTTP/1.1
X-Cisco-HN-Client-Type-Id: AA333BD4-72B2-5B4B-45EC-77AE486B17DE
Authorization: Bearer 27766B3ED7F444E9FDFD3849380291BCB46C24
Accept: application/json
Accept-Charset: utf-8
```

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the basic information of the currently logged-in account is retrieved successfully.

Response: Entity Body (on Success)

Returns a representation of the basic information of the currently logged-in account, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
accountId	A unique ID (system generated) of the account.	String	Yes
firstName	The first name of the user that the account belongs to.	String	Yes
lastName	The last name of the user that the account belongs to.	String	Yes
middleName	The middle name of the user that the account belongs to.	String	No

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success):

```
{
   "account":{
      "accountId":"36D3A139-C87E-4A34-94ED-F65BA53BDA68",
      "firstName":"John",
      "lastName":"Doe",
      "middleName":"L"
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:
See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"INVALID_SESSION_TOKEN",
        "message":"invalid access token",
        "parameters":[

        ]
      }
    }
  ]
}
```

Get email address of an account

Purpose

Gets (aka reads or retrieves) the email address of the currently logged-in account.

Request: URI and Headers

GET /cloud/user-service/rest/accounts/self/emailaddress

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, and **Accept** header fields are specified as shown above.

Here is an example of raw request headers:

```
GET /cloud/user-service/rest/accounts/self/emailaddress HTTP/1.1
X-Cisco-HN-Client-Type-Id: AA333BD4-72B2-5B4B-45EC-77AE486B17DE
Authorization: Bearer 27766B3ED7F444E9FDFD3849380291BCB46C24
Accept: application/json
Accept-Charset: utf-8
```

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the email address of the currently logged-in account is retrieved successfully.

Response: Entity Body (on Success)

Returns a representation of the email address of the currently logged-in account, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
accountId	A unique ID ((system generated) of the account.	String	Yes
emailAddress	The email address of the account.	String	Yes

Example Response Body (on Success):

```
{
   "account":{
      "accountId":"36D3A139-C87E-4A34-94ED-F65BA53BDA68",
      "emailAddress":"xyzpqrtest1@gmail.com"
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:
See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"INVALID_SESSION_TOKEN",
        "message":"invalid access token",
        "parameters":[

        ]
      }
    }
  ]
}
```

Network / Router

Get all networks (routers) associated with an account

Purpose

Gets all the networks (one network represents one router) associated with the currently logged-in account.
Note: There can only be one "default network" for an account.

Request: URI and Headers

GET /cloud/device-service/rest/accounts/self/networks

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, and **Accept** header fields are specified as shown above.

Here is an example of raw request headers:

```
GET /cloud/device-service/rest/accounts/self/networks HTTP/1.1
X-Cisco-HN-Client-Type-Id: AA333BD4-72B2-5B4B-45EC-77AE486B17DE
Authorization: Bearer 27766B3ED7F444E9FDFD3849380291BCB46C24
Accept: application/json
Accept-Charset: utf-8
```

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the networks associated with the account are retrieved successfully.

Response: Entity Body (on Success)

Returns a representation of the requested account's networks, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success):

```
{
  "networkAccountAssociations": [
    {
      "networkAccountAssociation": {
        "network": {
          "networkId": "network-2141C31D2EF440E1AF280674022DD0FF5E4987C5@ciscoconnectcloud.com",
          "friendlyName": "Home Network 1",
          "owner": {
            "accountId": "36D3A139-C87E-4A34-94ED-F65BA53BDA68",
            "firstName": "John",
            "lastName": "Doe",
            "alias": "John D",
          },
          "routerModelNumber": "E4200",
          "routerSerialNumber": "01C1360C105603"
        },
        "role": "ADMIN",
        "owner": true,
        "defaultNetwork": false
      }
    },
    {
      "networkAccountAssociation": {
        "network": {
          "networkId": "network-34E394A9A00A4C469BC2932C3B79940BFB73557D@ciscoconnectcloud.com",
          "friendlyName": "Home Network 2",
          "owner": {
            "accountId": "36D3A139-C87E-4A34-94ED-F65BA53BDA68",
            "firstName": "John",
            "lastName": "Doe",
            "alias": "John D",
          },
          "routerModelNumber": "E4200",
          "routerSerialNumber": "55C1360C105666"
        },
        "role": "ADMIN",
        "owner": true,
        "defaultNetwork": true
      }
    }
  ]
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:
See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"INVALID_SESSION_TOKEN",
        "message":"invalid access token",
        "parameters":[

        ]
      }
    }
  ]
}
```

Invoke a JNAP action remotely on a network (router)

Purpose

It invokes a JNAP action remotely on a network (router).

Request: URI and Headers

POST /cloud/JNAP

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
X-Cisco-HN-Network-Id: network-id-goes-here
X-JNAP-Action: jnap-action-goes-here
Accept: application/json
Content-Type: application/json; charset=UTF-8

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, **X-Cisco-HN-Network-Id**, **X-JNAP-Action**, **Accept** and **Content-Type** header fields are specified as shown above.

Here is an example of raw request headers:

```
POST /cloud/JNAP HTTP/1.1
X-Cisco-HN-Client-Type-Id: AA333BD4-72B2-5B4B-45EC-77AE486B17DE
Authorization: Bearer 27766B3ED7F444E9FDFD3849380291BCB46C24
X-Cisco-HN-Network-Id: network-34E394A9A00A4C469BC2932C3B79940BFB73557D@ciscoconnectcloud.com
X-JNAP-Action: http://cisco.com/jnap/core/GetDeviceInfo
Accept: application/json
Accept-Charset: utf-8
Content-Type: application/json; charset=UTF-8
Content-Length: 2
```

Request: Entity Body

The request body contains a JNAP request payload corresponding to the JNAP action mentioned in the **X-JNAP-Action** header field.

Example Request Body:

```
{}
```

Response: Status Codes

Always returns 200.

Response: Entity Body (on Success)

The response body contains a JNAP response payload returned by the router.

Example Response Body (on Success):

```
{
  "result":"OK",
  "output":{
    "services":[
      "http://cisco.com/jnap/core/Core",
      "http://cisco.com/jnap/debug/Debug",
      "http://cisco.com/jnap/devicelist/DeviceList",
      "http://cisco.com/jnap/diagnostics/Diagnostics",
      "http://cisco.com/jnap/firmwareupdate/FirmwareUpdate",
      "http://cisco.com/jnap/guestnetwork/GuestNetwork",
      "http://cisco.com/jnap/guestnetwork/GuestNetworkAuthentication",
      "http://cisco.com/jnap/locale/Locale",
      "http://cisco.com/jnap/macfilter/MACFilter",
      "http://cisco.com/jnap/networkconnections/NetworkConnections",
      "http://cisco.com/jnap/ownednetwork/OwnedNetwork",
      "http://cisco.com/jnap/router/Router",
      "http://cisco.com/jnap/routerleds/RouterLEDs",
      "http://cisco.com/jnap/routerlog/RouterLog",
      "http://cisco.com/jnap/wirelessap/WPSServer",
      "http://cisco.com/jnap/wirelessap/WirelessAP"
    ],
    "firmwareDate":"2012-04-05T11:52:00Z",
    "description":"Simultaneous Dual-Band Wireless-N Gigabit Router",
"manufacturer":"Cisco Systems, Inc.",
    "firmwareVersion":"2.1.37.131997",
    "serialNumber":"01C1360C105603",
    "modelNumber":"E4200",
    "hardwareVersion":"2"
  }
}
```

Response: Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - JNAP Action section for details.

Error codes returned:
See Common Error Codes Returned - JNAP Action.

Example Response Body (on Error):

```
{
  "result":"_ErrorSessionExpired ",
  "error":"the session has expired"
}
```

Device

Get all the devices in a network

Purpose

Gets (aka reads or retrieves) all the devices that are connected to the network (i.e. router) or have been connected to it at some time in the past. This operation can be performed by any account(s) associated with the network.

Request

GET /cloud/device-service/rest/networks/<networkId>/devices

The <networkId> is a value that uniquely identifies a network.

Here is the raw http request:

```
GET /cloud/device-service/rest/networks/networkId-goes-here/devices

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
Accept-Charset: utf-8
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is perfomred successfully.

Response: Entity Body (on Success)

Returns a representation of the requested network's devices, as a JSON document.

Elements/fields included for each device in the response body:

Name	Description	Data Type	Required?
deviceId	The unique ID associated with the device. The router generates this value the first time it discovers the device on the network.	String	Yes
macAddresses	A list of MAC addresses that are known to belong to the device's network adapters. Since a single physical device may have multiple network adapters, a device may appear as multiple devices in the device list. If the router is able to detect that the two "devices" are in fact the same physical device, it may automatically merge the device data.	List of **macAddress**	Yes
manufacturer	The manufacturer of the device.	String	No
modelNumber	The model number of the device.	String	No
serialNumber	The serial number of the device.	String	No
hardwareVersion	The hardware version of the device.	String	No
firmwareVersion	The version number of the device's firmware.	String	No
firmwareDate	The date and time associated with the device's firmware.	DateTime	No
type	The type of the device. It is **not** intended to be displayed to the end user. If this value is not present, the type of the device is unknown. The list of values currently in use: **Computer** - A generic computer **Camera** - A network-enabled camera **Phone** - A cell (mobile) phone **GameConsole** - A video game console, such as an XBOX **Printer** - A network-attached printer **MediaPlayer** - A network-attached media player, both audio and video, such as a TiVo or Sonos **Storage** - A network-attached storage device **Infrastructure** - A network infrastructure device, such as a router, wireless access point, bridge, etc	String	No
friendlyName	A user-specified friendly name for the device. if a user does not explicitly name the device, the value of this field will be same as the value of the **defaultFriendlyName** field.	String	No
defaultFriendlyName	A friendly name for the device, as determined by the router's internal identification mechanism. It can **not** be overwritten by the user.	String	No
description	A brief description of the device.	String	No

operatingSystem	The operating system running on the device.	String	No
authority	Whether the device is the authority (i.e. main router) on the local network.	Boolean (**true** or **false**)	Yes
layer3Connections	A list of the device's upstream connections to the network. If the device is currently offline, the list will be empty. If the device is online, the list will usually contain one item, since it is very rare for a device to be simultaneously connected to the same network through multiple network adapters.	List of **layer3Connection**	No
layer3Connection.macAddress	The MAC address of the connected network adapter on the device.	String	Yes (if **layer3Connection** element is present)
layer3Connection.ipAddress	The IPv4 address of the device's connection.	String	No
layer3Connection.ipv6Address	The IPv6 address of the device's connection.	String	No

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "devices": [
        {
            "device": {
                "deviceId":"0b6ec788-71a7-476f-85e3-9729d36bf844",
                "macAddresses": [ { "macAddress":"58:6D:8F:F5:55:B6" } ],
                "manufacturer":"Cisco Systems, Inc.",
                "modelNumber":"EA4500",
                "serialNumber":"01C1360C102979",
                "hardwareVersion":"2",
                "firmwareVersion":"2.1.38.138143",
                "firmwareDate":"2012-06-19T21:38:00Z",
                "type":"Infrastructure",
                "friendlyName":"My Home Router",
                "defaultFriendlyName":"Cisco2019",
                "description":"Simultaneous Dual-Band Wireless-N Gigabit Router",
                "authority":true,
                "layer3Connections": [
                    {
                        "layer3Connection": {
                            "macAddress":"58:6D:8F:F5:55:B6",
                            "ipAddress":"192.168.1.1"
                        }
                    }
                ]
            }
        },
        {
            "device": {
                "deviceId":"748517b7-fbc8-4454-9fab-c28019a12731",
                "macAddresses": [ { "macAddress":"44:1E:A1:CA:27:83" }, {
"macAddress":"00:25:9C:12:A8:41" } ],
                "type":"Computer",
                "friendlyName":"My Windows laptop",
                "defaultFriendlyName":"AJAIN-WS",
                "operatingSystem":"Windows 7",
                "authority":false
            }
        },
        {
            "device": {
                "deviceId":"852f1065-866d-4219-b4c7-b29ee50712d2",
                "macAddresses": [ { "macAddress":"90:27:E4:ED:75:42" } ],
                "manufacturer":"Apple",
                "modelNumber":"MacBook",
                "type":"Computer",
                "friendlyName":"My MacBook Pro",
                "defaultFriendlyName":"AJAIN's MacBook Pro",
                "operatingSystem":"OS X",
                "authority":false,
                "layer3Connections": [
                    {
                        "layer3Connection": {
                            "macAddress":"90:27:E4:ED:75:42",
                            "ipAddress":"192.168.1.102"
                        }
                    }
                ]
            }
        }
    ]
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	The network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Get a device in a network

Purpose

Gets (aka reads or retrieves) the information about a specific device in a network. This operation can be performed by any account(s) associated with the network.

Request

GET /cloud/device-service/rest/networks/<networkId>/devices/<deviceId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <deviceId> is a value that uniquely identifies a device within a network.

To get the device information of the router (representing the specified network), the <deviceId> can be replaced by **router** keyword in the URL as shown below.

GET /cloud/device-service/rest/networks/<networkId>/devices/router

Here is the raw http request:

```
GET /cloud/device-service/rest/networks/networkId-goes-here/devices/deviceId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
Accept-Charset: utf-8
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is perfomred successfully.

Response: Entity Body (on Success)

Returns a representation of the requested device, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
deviceId	The unique ID associated with the device. The router generates this value the first time it discovers the device on the network.	String	Yes
macAddresses	A list of MAC addresses that are known to belong to the device's network adapters. Since a single physical device may have multiple network adapters, a device may appear as multiple devices in the device list. If the router is able to detect that the two "devices" are in fact the same physical device, it may automatically merge the device data.	List of **macAddress**	Yes
manufacturer	The manufacturer of the device.	String	No
modelNumber	The model number of the device.	String	No
serialNumber	The serial number of the device.	String	No
hardwareVersion	The hardware version of the device.	String	No
firmwareVersion	The version number of the device's firmware.	String	No
firmwareDate	The date and time associated with the device's firmware.	DateTime	No
type	The type of the device. It is **not** intended to be displayed to the end user. If this value is not present, the type of the device is unknown. The list of values currently in use: **Computer** - A generic computer **Camera** - A network-enabled camera **Phone** - A cell (mobile) phone **GameConsole** - A video game console, such as an XBOX **Printer** - A network-attached printer **MediaPlayer** - A network-attached media player, both audio and video, such as a TiVo or Sonos **Storage** - A network-attached storage device **Infrastructure** - A network infrastructure device, such as a router, wireless access point, bridge, etc	String	No
friendlyName	A user-specified friendly name for the device. if a user does not explicitly name the device, the value of this field will be same as the value of the **defaultFriendlyName** field.	String	No
defaultFriendlyName	A friendly name for the device, as determined by the router's internal identification mechanism. It can **not** be overwritten by the user.	String	No
description	A brief description of the device.	String	No
operatingSystem	The operating system running on the device.	String	No
authority	Whether the device is the authority (i.e. main router) on the local network.	Boolean (**true** or **false**)	Yes
layer3Connections	A list of the device's upstream connections to the network. If the device is currently offline, the list will be empty. If the device is online, the list will usually contain one item, since it is very rare for a device to be simultaneously connected to the same network through multiple network adapters.	List of **layer3Connection**	No
layer3Connection.macAddress	The MAC address of the connected network adapter on the device.	String	Yes (if **layer3Connection** element is present)
layer3Connection.ipAddress	The IPv4 address of the device's connection.	String	No
layer3Connection.ipv6Address	The IPv6 address of the device's connection.	String	No

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "device": {
 "deviceId":"0b6ec788-71a7-476f-85e3-9729d36bf844",
 "macAddresses": [ { "macAddress":"58:6D:8F:F5:55:B6" } ],
 "manufacturer":"Cisco Systems, Inc.",
 "modelNumber":"EA4500",
 "serialNumber":"01C1360C102979",
 "hardwareVersion":"2",
 "firmwareVersion":"2.1.38.138143",
 "firmwareDate":"2012-06-19T21:38:00Z",
 "type":"Infrastructure",
 "friendlyName":"My Home Router",
        "defaultFriendlyName":"Cisco2019",
 "description":"Simultaneous Dual-Band Wireless-N Gigabit Router",
 "authority":true,
 "layer3Connections": [
     {
  "layer3Connection": {
      "macAddress":"58:6D:8F:F5:55:B6",
      "ipAddress":"192.168.1.1"
  }
     }
 ]
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
404	DEVICE_NOT_FOUND	A device with the specified device ID is not found in the specified network.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Update a device in a network

Purpose

Updates the information about a specific device in a network. This operation can be performed only by ADMIN account(s) associated with the network.

Request

PUT /cloud/device-service/rest/networks/<networkId>/devices/<deviceId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <deviceId> is a value that uniquely identifies a device within a network.

Here is the raw http request:

```
PUT /cloud/device-service/rest/networks/networkId-goes-here/devices/deviceId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Content-Type: application/json; charset=UTF-8
Accept: application/json
Accept-Charset: utf-8

json-content-goes-here
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, **Content-Type** and **Accept** header fields are specified as shown above.

Request: Entity Body

The request body contains a partial representation of the device to be updated, as a JSON document.

Elements/fields to be included in the request body:

Name	Description	Data Type	Required?
friendlyName	The user-specified friendly name of the device.	String	No

Example Request Body (JSON):

```
{
    "device": {
 "friendlyName":"My Home Router"
    }
}
```

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the updated device, as a JSON document.

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "device": {
 "deviceId":"0b6ec788-71a7-476f-85e3-9729d36bf844",
 "macAddresses": [ { "macAddress":"58:6D:8F:F5:55:B6" } ],
 "manufacturer":"Cisco Systems, Inc.",
 "modelNumber":"EA4500",
 "serialNumber":"01C1360C102979",
 "hardwareVersion":"2",
 "firmwareVersion":"2.1.38.138143",
 "firmwareDate":"2012-06-19T21:38:00Z",
 "type":"Infrastructure",
 "friendlyName":"My Home Router",
        "defaultFriendlyName":"Cisco2019",
 "description":"Simultaneous Dual-Band Wireless-N Gigabit Router",
 "authority":true,
 "layer3Connections": [
     {
  "layer3Connection": {
      "macAddress":"58:6D:8F:F5:55:B6",
      "ipAddress":"192.168.1.1"
  }
     }
 ]
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
404	DEVICE_NOT_FOUND	A device with the specified device ID is not found in the specified network.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Delete a device from a network

Purpose

Delete the inofrmation about a specific device from the device list of a network. This is useful for "cleaning up" after a device is permanently removed from the network (e.g., thrown away, returned to the store). Only devices that are not currently connected to the network can be deleted. Note that if a device is deleted and subsequently rejoins the network, the device ID that is assigned to it when it is rediscovered will not necessarily be the same device ID that it had before it was deleted.
This operation can be performed only by ADMIN account(s) associated with the network.

Request

DELETE /cloud/device-service/rest/networks/<networkId>/devices/<deviceId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <deviceId> is a value that uniquely identifies a device within a network.

Here is the raw http request:

```
DELETE /cloud/device-service/rest/networks/networkId-goes-here/devices/deviceId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
Accept-Charset: utf-8
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the deleted device, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "device": {
 "deviceId":"0b6ec788-71a7-476f-85e3-9729d36bf844"
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
404	DEVICE_NOT_FOUND	A device with the specified device ID is not found in the specified network.
400	DEVICE_CANNOT_BE_DELETED	The device cannot be deleted because it is currently connected to the network, or it is the router device that represents the specified network.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Port Forwarding

Single Port Forwarding Rules

Get the metadata about single-port forwarding rules

Purpose

Gets the metadata (the maximum number of rules that can exist simultaneously, the maximum length of the description field, etc.) about single-port forwarding rules for the specified router. This operation can be performed by any account(s) associated with the network.

Request

GET /cloud/device-service/rest/networks/<networkId>/singleportforwardingrulesmetadata

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
GET /cloud/device-service/rest/networks/networkId-goes-here/singleportforwardingrulesmetadata

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
Accept-Charset: utf-8
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is perfomred successfully.

Response: Entity Body (on Success)

Returns a representation of the metadata about single-port forwarding rules, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
maxDescriptionLength	The maximum length, in bytes, of the **description** field of a single-port forwarding rule.	Integer	Yes
maxRules	The maximum number of single-port forwarding rules that can exist simultaneously.	Integer	Yes

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "singlePortForwardingRulesMetadata": {
        "maxDescriptionLength":20,
        "maxRules":20
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Get all single-port forwarding rules

Purpose

Gets (aka reads or retrieves) the list of single-port forwarding rules currently set on the router. This operation can be performed by any account(s) associated with the network.

Request

GET /cloud/device-service/rest/networks/<networkId>/singleportforwardingrules

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
GET /cloud/device-service/rest/networks/networkId-goes-here/singleportforwardingrules

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
Accept-Charset: utf-8
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is perfomred successfully.

Response: Entity Body (on Success)

Returns a representation of the single-port forwarding rules, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
singlePortForwardingRules	The current list of single-port forwarding rules.	List of **singlePortForwardingRule**	Yes
singlePortForwardingRule	A rule describing a single external port that should be forwarded to a server on the LAN. Only one rule can exist for a specific external port/protocol combination at any given time.	Complex	No
Fields included in **singlePortForwardingRule**:			
ruleId	The unique rule ID.	String	Yes
enabled	Whether the rule is enabled.	Boolean	Yes
externalPort	The external port that should be forwarded. This value must be between 0 and 65535.	Integer	Yes
protocol	The protocol that should be forwarded. Valid values are: **TCP** - The TCP protocol **UDP** - The UDP protocol **Both** - Both the TCP and UDP protocols	String	Yes
internalServerIPAddress	The IP address of the destination server on the LAN. This value must be in the same subnet as the router's LAN host address (as defined by SetLANSettings JNAP call), but not equal to the router's LAN host address and not fall inside the following reserved ranges: **Address Block** **Description** **Reference**	String	Yes

192.168.1.0/24: **192.168.1.0**
Subnetwork ID
RFC 922, Section 7

	192.168.1.0/24: **192.168.1.255** Subnetwork Broadcast Address RFC 922, Section 7		
internalPort	The port number of the destination server on the LAN. This value must be between 0 and 65535.	Integer	Yes
description	A human-readable description of the rule. Typically used for capturing the name of the application the rule is set up for.	String	Yes

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "singlePortForwardingRules": [
        {
            "singlePortForwardingRule": {
                "ruleId":"TCP8000",
                "enabled":true,
                "externalPort":8000,
                "protocol":"TCP",
                "internalServerIPAddress":"192.168.1.140",
                "internalPort":8080,
                "description":"Testing Rule 1"
            }
        },
        {
            "singlePortForwardingRule": {
                "ruleId":"Both9000",
                "enabled":false,
                "externalPort":9000,
                "protocol":"Both",
                "internalServerIPAddress":"192.168.1.200",
                "internalPort":80,
                "description":"Testing Rule 2"
            }
        }
    ]
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	The network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Add a single-port forwarding rule

Purpose

Adds a new single-port forwarding rule on the router. This operation can be performed only by ADMIN account(s) associated with the network.

Request

POST /cloud/device-service/rest/networks/<networkId>/singleportforwardingrules

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
POST /cloud/device-service/rest/networks/networkId-goes-here/singleportforwardingrules

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Content-Type: application/json; charset=UTF-8
Accept: application/json
Accept-Charset: utf-8

json-content-goes-here
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, **Content-Type** and **Accept** header fields are specified as shown above.

Request: Entity Body

The request body contains a representation of the single-port forwarding rule to be added, as a JSON document.

Elements/fields to be included in the request body:

Name	Description	Data Type	Required?
singlePortForwardingRule	A rule describing a single external port that should be forwarded to a server on the LAN. Only one rule can exist for a specific external port/protocol combination at any given time.	Complex	Yes
Fields to be included in **singlePortForwardingRule** :			
enabled	Whether the rule is enabled.	Boolean	Yes
externalPort	The external port that should be forwarded. This value must be between 0 and 65535.	Integer	Yes
protocol	The protocol that should be forwarded. Valid values are: **TCP** - The TCP protocol **UDP** - The UDP protocol **Both** - Both the TCP and UDP protocols	String	Yes
internalServerIPAddress	The IP address of the destination server on the LAN. This value must be in the same subnet as the router's LAN host address (as defined by SetLANSettings JNAP call), but not equal to the router's LAN host address and not fall inside the following reserved ranges: **Address Block** **Description** **Reference**	String	Yes

192.168.1.0/24: **192.168.1.0**
Subnetwork ID
RFC 922, Section 7

	192.168.1.0/24: **192.168.1.255** Subnetwork Broadcast Address RFC 922, Section 7		
internalPort	The port number of the destination server on the LAN. This value must be between 0 and 65535.	Integer	Yes
description	A human-readable description of the rule. Typically used for capturing the name of the application the rule is set up for.	String	Yes

Example Request Body (JSON):

```
{
    "singlePortForwardingRule": {
        "enabled":true,
        "externalPort":8000,
        "protocol":"TCP",
        "internalServerIPAddress":"192.168.1.140",
        "internalPort":8080,
        "description":"Testing Rule 1"
    }
}
```

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a partial representation of the newly added single-port forwarding rule, as a JSON document.

In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "singlePortForwardingRule": {
        "ruleId":TCP8000,
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
409	RULE_OVERLAP	The single-port forwarding rule being added overlaps with an existing rule. In other words, only one rule can exist for a specific external port/protocol combination at any given time.
400	MAX_RULES_LIMIT_REACHED	A new single-port forwarding rule can not be added because the number of single-port forwarding rules already existing on the router is equal to the maximum allowed number of rules.
400	DESCRIPTION_TOO_LONG	The specified description is longer than the maximum allowed length.
400	INVALID_EXTERNAL_PORT	The specified external port was not between 0 and 65535.
400	INVALID_INTERNAL_PORT	The specified internal server port was not between 0 and 65535.
400	INVALID_INTERNAL_SERVER_IP_ADDRESS	The specified internal server IP addresses was invalid.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Get a single-port forwarding rule

toc

Purpose

Gets (aka reads or retrieves) a specific single-port forwarding rule set on the router. This operation can be performed by any account(s) associated with the network.

Request

GET /cloud/device-service/rest/networks/<networkId>/singleportforwardingrules/<ruleId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <ruleId> is a value that uniquely identifies a single-port forwarding rule set on a router.

Here is the raw http request:

```
GET /cloud/device-service/rest/networks/networkId-goes-here/singleportforwardingrules/ruleId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
Accept-Charset: utf-8
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is perfomred successfully.

Response: Entity Body (on Success)

Returns a representation of the requested single-port forwarding rule, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
singlePortForwardingRule	A rule describing a single external port that should be forwarded to a server on the LAN. Only one rule can exist for a specific external port/protocol combination at any given time.	Complex	Yes
Fields included in **singlePortForwardingRule** :			
ruleId	The unique rule ID.	String	Yes
enabled	Whether the rule is enabled.	Boolean	Yes
externalPort	The external port that should be forwarded. This value must be between 0 and 65535.	Integer	Yes
protocol	The protocol that should be forwarded. Valid values are: **TCP** - The TCP protocol **UDP** - The UDP protocol **Both** - Both the TCP and UDP protocols	String	Yes
internalServerIPAddress	The IP address of the destination server on the LAN. This value must be in the same subnet as the router's LAN host address (as defined by SetLANSettings JNAP call), but not equal to the router's LAN host address and not fall inside the following reserved ranges: **Address Block** **Description** **Reference**	String	Yes

192.168.1.0/24: **192.168.1.0**
Subnetwork ID
RFC 922, Section 7

	192.168.1.0/24: **192.168.1.255** Subnetwork Broadcast Address RFC 922, Section 7		
internalPort	The port number of the destination server on the LAN. This value must be between 0 and 65535.	Integer	Yes
description	A human-readable description of the rule. Typically used for capturing the name of the application the rule is set up for.	String	Yes

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "singlePortForwardingRule": {
        "ruleId":"TCP8000",
        "enabled":true,
        "externalPort":8000,
        "protocol":"TCP",
        "internalServerIPAddress":"192.168.1.140",
        "internalPort":8080,
        "description":"Testing Rule 1"
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
404	RULE_NOT_FOUND	A rule with the specified rule ID is not found on the specified router / network.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Update a single-port forwarding rule

Purpose

Updates a specific single-port forwarding rule set on the router. This operation can be performed only by ADMIN account(s) associated with the network.

Request

PUT /cloud/device-service/rest/networks/<networkId>/singleportforwardingrules/<ruleId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <ruleId> is a value that uniquely identifies a single-port forwarding rule set on a router.

Here is the raw http request:

```
PUT /cloud/device-service/rest/networks/networkId-goes-here/singleportforwardingrules/ruleId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Content-Type: application/json; charset=UTF-8
Accept: application/json
Accept-Charset: utf-8

json-content-goes-here
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, **Content-Type** and **Accept** header fields are specified as shown above.

Request: Entity Body

The request body contains a partial representation of the single-port forwarding rule to be updated, as a JSON document.

Elements/fields to be included in the request body:

Name	Description	Data Type	Required?

singlePortForwardingRule	A rule describing a single external port that should be forwarded to a server on the LAN. Only one rule can exist for a specific external port/protocol combination at any given time.	Complex	Yes
Fields to be included in **singlePortForwardingRule** :			
enabled	Whether the rule is enabled.	Boolean	No
externalPort	The external port that should be forwarded. This value must be between 0 and 65535.	Integer	No
protocol	The protocol that should be forwarded. Valid values are: **TCP** - The TCP protocol **UDP** - The UDP protocol **Both** - Both the TCP and UDP protocols	String	No
internalServerIPAddress	The IP address of the destination server on the LAN. This value must be in the same subnet as the router's LAN host address (as defined by SetLANSettings JNAP call), but not equal to the router's LAN host address and not fall inside the following reserved ranges: **Address Block** **Description** **Reference**	String	No

192.168.1.0/24: **192.168.1.0**
Subnetwork ID
RFC 922, Section 7

	192.168.1.0/24: **192.168.1.255** Subnetwork Broadcast Address RFC 922, Section 7		
internalPort	The port number of the destination server on the LAN. This value must be between 0 and 65535.	Integer	No
description	A human-readable description of the rule. Typically used for capturing the name of the application the rule is set up for.	String	No

Example Request Body (JSON):

```
{
    "singlePortForwardingRule": {
        "enabled":true,
        "externalPort":8000,
        "protocol":"TCP",
        "internalServerIPAddress":"192.168.1.140",
        "internalPort":8080,
        "description":"Testing Rule 1"
    }
}
```

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the updated single-port forwarding rule, as a JSON document.

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

⚠ The **ruleId** may change after a rule has been updated. The respone body contains the updated **ruleId**.

Example Response Body (on Success) (JSON):

```
{
    "singlePortForwardingRule": {
        "ruleId":"TCP8000",
        "enabled":true,
        "externalPort":8000,
        "protocol":"TCP",
        "internalServerIPAddress":"192.168.1.140",
        "internalPort":8080,
        "description":"Testing Rule 1"
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
404	RULE_NOT_FOUND	A rule with the specified rule ID is not found on the specified router / network.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
409	RULE_OVERLAP	The single-port forwarding rule being updated overlaps with an existing rule. In other words, only one rule can exist for a specific external port/protocol combination at any given time.
400	DESCRIPTION_TOO_LONG	The specified description is longer than the maximum allowed length.
400	INVALID_EXTERNAL_PORT	The specified external port was not between 0 and 65535.
400	INVALID_INTERNAL_PORT	The specified internal server port was not between 0 and 65535.
400	INVALID_INTERNAL_SERVER_IP_ADDRESS	The specified internal server IP addresses was invalid.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Delete a single-port forwarding rule

Purpose

Deletes a specific single-port forwarding rule set on the router. This operation can be performed only by ADMIN account(s) associated with the network.

Request

DELETE /cloud/device-service/rest/networks/<networkId>/singleportforwardingrules/<ruleId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <ruleId> is a value that uniquely identifies a single-port forwarding rule set on a router.

Here is the raw http request:

```
DELETE /cloud/device-service/rest
/networks/networkId-goes-here/singleportforwardingrules/ruleId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
Accept-Charset: utf-8
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a partial representation of the deleted single-port forwarding rule, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "singlePortForwardingRule": {
        "ruleId":TCP8000,
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
404	RULE_NOT_FOUND	A rule with the specified rule ID is not found on the specified router / network.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Port Range Forwarding Rules

Get the metadata about port range forwarding rules

Purpose

Gets the metadata (the maximum number of rules that can exist simultaneously, the maximum length of the description field, etc.) about port range forwarding rules for the specified router. This operation can be performed by any account(s) associated with the network.

Request

GET /cloud/device-service/rest/networks/<networkId>/portrangeforwardingrulesmetadata

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
GET /cloud/device-service/rest/networks/networkId-goes-here/portrangeforwardingrulesmetadata

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
Accept-Charset: utf-8
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is perfomred successfully.

Response: Entity Body (on Success)

Returns a representation of the metadata about port range forwarding rules, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
maxDescriptionLength	The maximum length, in bytes, of the **description** field of a port range forwarding rule.	Integer	Yes
maxRules	The maximum number of port range forwarding rules that can exist simultaneously.	Integer	Yes

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "portRangeForwardingRulesMetadata": {
        "maxDescriptionLength":20,
        "maxRules":20
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Get all port range forwarding rules

Purpose

Gets (aka reads or retrieves) the list of port range forwarding rules currently set on the router. This operation can be performed by any account(s) associated with the network.

Request

GET /cloud/device-service/rest/networks/<networkId>/portrangeforwardingrules

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
GET /cloud/device-service/rest/networks/networkId-goes-here/portrangeforwardingrules

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
Accept-Charset: utf-8
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is perfomred successfully.

Response: Entity Body (on Success)

Returns a representation of the port range forwarding rules, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
portRangeForwardingRules	The current list of port range forwarding rules.	List of **portRangeForwardingRule**	Yes
portRangeForwardingRule	A rule describing a range of external ports that should be forwarded to a server on the LAN. Only one rule can exist for a specific external port/protocol combination at any given time.	Complex	No
Fields included in **portRangeForwardingRule**:			
ruleId	The unique rule ID.	String	Yes
enabled	Whether the rule is enabled.	Boolean	Yes
firstExternalPort	The first external port in the range that should be forwarded. This value must be between 0 and 65535.	Integer	Yes
lastExternalPort	The last external port in the range that should be forwarded. This value must be between 0 and 65535.	Integer	Yes
protocol	The protocol that should be forwarded. Valid values are: **TCP** - The TCP protocol **UDP** - The UDP protocol **Both** - Both the TCP and UDP protocols	String	Yes
internalServerIPAddress	The IP address of the destination server on the LAN. This value must be in the same subnet as the router's LAN host address (as defined by SetLANSettings JNAP call), but not equal to the router's LAN host address and not fall inside the following reserved ranges: **Address Block** **Description** **Reference**	String	Yes

192.168.1.0/24: **192.168.1.0**
Subnetwork ID
RFC 922, Section 7

	192.168.1.0/24: **192.168.1.255** Subnetwork Broadcast Address RFC 922, Section 7		
description	A human-readable description of the rule. Typically used for capturing the name of the application the rule is set up for.	String	Yes

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "portRangeForwardingRules": [
        {
            "portRangeForwardingRule": {
                "ruleId":"TCP6888",
                "enabled":true,
                "firstExternalPort":6888,
                "lastExternalPort":6900,
                "protocol":"TCP",
                "internalServerIPAddress":"192.168.1.140",
                "description":"Testing Rule 1"
            }
        },
        {
            "portRangeForwardingRule": {
                "ruleId":"Both8090",
                "enabled":false,
                "firstExternalPort":8090,
                "lastExternalPort":8100,
                "protocol":"Both",
                "internalServerIPAddress":"192.168.1.200",
                "description":"Testing Rule 2"
            }
        }
    ]
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	The network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Add a port range forwarding rule

Purpose

Adds a new port range forwarding rule on the router. This operation can be performed only by ADMIN account(s) associated with the network.

Request

POST /cloud/device-service/rest/networks/<networkId>/portrangeforwardingrules

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
POST /cloud/device-service/rest/networks/networkId-goes-here/portrangeforwardingrules

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Content-Type: application/json; charset=UTF-8
Accept: application/json
Accept-Charset: utf-8

json-content-goes-here
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, **Content-Type** and **Accept** header fields are specified as shown above.

Request: Entity Body

The request body contains a representation of the port range forwarding rule to be added, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
portRangeForwardingRule	A rule describing a range of external ports that should be forwarded to a server on the LAN. Only one rule can exist for a specific external port/protocol combination at any given time.	Complex	Yes
Fields included in **portRangeForwardingRule** :			
enabled	Whether the rule is enabled.	Boolean	Yes
firstExternalPort	The first external port in the range that should be forwarded. This value must be between 0 and 65535.	Integer	Yes
lastExternalPort	The last external port in the range that should be forwarded. This value must be between 0 and 65535.	Integer	Yes
protocol	The protocol that should be forwarded. Valid values are: **TCP** - The TCP protocol **UDP** - The UDP protocol **Both** - Both the TCP and UDP protocols	String	Yes
internalServerIPAddress	The IP address of the destination server on the LAN. This value must be in the same subnet as the router's LAN host address (as defined by SetLANSettings JNAP call), but not equal to the router's LAN host address and not fall inside the following reserved ranges: **Address Block** **Description** **Reference**	String	Yes

192.168.1.0/24: **192.168.1.0**
Subnetwork ID
RFC 922, Section 7

	192.168.1.0/24: **192.168.1.255** Subnetwork Broadcast Address RFC 922, Section 7		
description	A human-readable description of the rule. Typically used for capturing the name of the application the rule is set up for.	String	Yes

Example Request Body (JSON):

```
{
    "portRangeForwardingRule": {
        "enabled":true,
        "firstExternalPort":6888,
        "lastExternalPort":6900,
        "protocol":"TCP",
        "internalServerIPAddress":"192.168.1.140",
        "description":"Testing Rule 1"
    }
}
```

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a partial representation of the newly added port range forwarding rule, as a JSON document.

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "portRangeForwardingRule": {
        "ruleId":"TCP6888",
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
409	RULE_OVERLAP	The port range forwarding rule being added overlaps with an existing rule. In other words, only one rule can exist for a single external port/protocol combination at any given time.
400	MAX_RULES_LIMIT_REACHED	A new port range forwarding rule can not be added because the number of port range forwarding rules already existing on the router is equal to the maximum allowed number of rules.
400	DESCRIPTION_TOO_LONG	The specified description is longer than the maximum allowed length.
400	INVALID_EXTERNAL_PORT	One of the specified external ports was not between 0 and 65535.
400	INVALID_EXTERNAL_PORT_RANGE	The specified external port ranges contained a first value that was greater than its last value.
400	INVALID_INTERNAL_SERVER_IP_ADDRESS	The specified internal server IP addresses was invalid.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Get a port range forwarding rule

Purpose

Gets (aka reads or retrieves) a specific port range forwarding rule set on the router. This operation can be performed by any account(s) associated with the network.

Request

GET /cloud/device-service/rest/networks/<networkId>/portrangeforwardingrules/<ruleId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <ruleId> is a value that uniquely identifies a port range forwarding rule set on a router.

Here is the raw http request:

```
GET /cloud/device-service/rest/networks/networkId-goes-here/portrangeforwardingrules/ruleId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
Accept-Charset: utf-8
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is perfomred successfully.

Response: Entity Body (on Success)

Returns a representation of the requested port range forwarding rule, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
portRangeForwardingRule	A rule describing a range of external ports that should be forwarded to a server on the LAN. Only one rule can exist for a specific external port/protocol combination at any given time.	Complex	Yes
Fields included in **portRangeForwardingRule**:			
ruleId	The unique rule ID.	String	Yes
enabled	Whether the rule is enabled.	Boolean	Yes
firstExternalPort	The first external port in the range that should be forwarded. This value must be between 0 and 65535.	Integer	Yes
lastExternalPort	The last external port in the range that should be forwarded. This value must be between 0 and 65535.	Integer	Yes
protocol	The protocol that should be forwarded. Valid values are: **TCP** - The TCP protocol **UDP** - The UDP protocol **Both** - Both the TCP and UDP protocols	String	Yes
internalServerIPAddress	The IP address of the destination server on the LAN. This value must be in the same subnet as the router's LAN host address (as defined by SetLANSettings JNAP call), but not equal to the router's LAN host address and not fall inside the following reserved ranges:	String	Yes

Address Block	**Description**	**Reference**
192.168.1.0/24: **192.168.1.0**	Subnetwork ID	RFC 922, Section 7

	192.168.1.0/24: **192.168.1.255** Subnetwork Broadcast Address RFC 922, Section 7		
description	A human-readable description of the rule. Typically used for capturing the name of the application the rule is set up for.	String	Yes

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "portRangeForwardingRule": {
        "ruleId":"TCP6888",
        "enabled":true,
        "firstExternalPort":6888,
        "lastExternalPort":6900,
        "protocol":"TCP",
        "internalServerIPAddress":"192.168.1.140",
        "description":"Testing Rule 1"
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
404	RULE_NOT_FOUND	A rule with the specified rule ID is not found on the specified router / network.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Update a port range forwarding rule

Purpose

Updates a specific port range forwarding rule set on the router. This operation can be performed only by ADMIN account(s) associated with the network.

Request

PUT /cloud/device-service/rest/networks/<networkId>/portrangeforwardingrules/<ruleId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <ruleId> is a value that uniquely identifies a port range forwarding rule set on a router.

Here is the raw http request:

```
PUT /cloud/device-service/rest/networks/networkId-goes-here/portrangeforwardingrules/ruleId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Content-Type: application/json; charset=UTF-8
Accept: application/json
Accept-Charset: utf-8

json-content-goes-here
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, **Content-Type** and **Accept** header fields are specified as shown above.

Request: Entity Body

The request body contains a partial representation of the port range forwarding rule to be updated, as a JSON document.

Elements/fields to be included in the request body:

Name	Description	Data Type	Required?
portRangeForwardingRule	A rule describing a range of external ports that should be forwarded to a server on the LAN. Only one rule can exist for a specific external port/protocol combination at any given time.	Complex	Yes

Fields included in **portRangeForwardingRule** :			
enabled	Whether the rule is enabled.	Boolean	No
firstExternalPort	The first external port in the range that should be forwarded. This value must be between 0 and 65535.	Integer	No
lastExternalPort	The last external port in the range that should be forwarded. This value must be between 0 and 65535.	Integer	No
protocol	The protocol that should be forwarded. Valid values are: **TCP** - The TCP protocol **UDP** - The UDP protocol **Both** - Both the TCP and UDP protocols	String	No
internalServerIPAddress	The IP address of the destination server on the LAN. This value must be in the same subnet as the router's LAN host address (as defined by SetLANSettings JNAP call), but not equal to the router's LAN host address and not fall inside the following reserved ranges: **Address Block** **Description** **Reference**	String	No

192.168.1.0/24: **192.168.1.0**
Subnetwork ID
RFC 922, Section 7

	192.168.1.0/24: **192.168.1.255** Subnetwork Broadcast Address RFC 922, Section 7		
description	A human-readable description of the rule. Typically used for capturing the name of the application the rule is set up for.	String	No

Example Request Body (JSON):

```
{
    "portRangeForwardingRule": {
        "enabled":true,
        "firstExternalPort":6888,
        "lastExternalPort":6900,
        "protocol":"TCP",
        "internalServerIPAddress":"192.168.1.140",
        "description":"Testing Rule 1"
    }
}
```

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the updated port range forwarding rule, as a JSON document.

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

⚠ The **ruleId** may change after a rule has been updated. The respone body contains the updated **ruleId**.

Example Response Body (on Success) (JSON):

```
{
    "portRangeForwardingRule": {
        "ruleId":"TCP6888",
        "enabled":true,
        "firstExternalPort":6888,
        "lastExternalPort":6900,
        "protocol":"TCP",
        "internalServerIPAddress":"192.168.1.140",
        "description":"Testing Rule 1"
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
404	RULE_NOT_FOUND	A rule with the specified rule ID is not found on the specified router / network.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
409	RULE_OVERLAP	The port range forwarding rule being updated overlaps with an existing rule. In other words, only one rule can exist for a single external port/protocol combination at any given time.
400	DESCRIPTION_TOO_LONG	The specified description is longer than the maximum allowed length.
400	INVALID_EXTERNAL_PORT	One of the specified external ports was not between 0 and 65535.
400	INVALID_EXTERNAL_PORT_RANGE	The specified external port ranges contained a first value that was greater than its last value.
400	INVALID_INTERNAL_SERVER_IP_ADDRESS	The specified internal server IP addresses was invalid.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Delete a port range forwarding rule

Purpose

Deletes a specific port range forwarding rule set on the router. This operation can be performed only by ADMIN account(s) associated with the network.

Request

DELETE /cloud/device-service/rest/networks/<networkId>/portrangeforwardingrules/<ruleId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <ruleId> is a value that uniquely identifies a port range forwarding rule set on a router.

Here is the raw http request:

```
DELETE /cloud/device-service/rest
/networks/networkId-goes-here/portrangeforwardingrules/ruleId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
Accept-Charset: utf-8
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a partial representation of the deleted port range forwarding rule, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
    "portRangeForwardingRule": {
        "ruleId":"TCP6888",
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
404	RULE_NOT_FOUND	A rule with the specified rule ID is not found on the specified router / network.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Network Traffic Statistics

Get network traffic statistics

Purpose

Gets (aka reads or retrieves) network traffic statistics for each device on the LAN. This includes all traffic between devices on the LAN and traffic from the LAN to the WAN. This operation can be performed by any account(s) associated with the network.

Request

GET /cloud/device-service/rest/networks/<networkId>/trafficstatistics

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
GET /cloud/device-service/rest/networks/networkId-goes-here/trafficstatistics

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
Accept-Charset: utf-8
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the traffic statistics as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
trafficStatistics	The list of traffic-statistic.	List of **trafficStatistic**	Yes
trafficStatistic	A traffic-statistic containing the network traffic stats of a device.	Complex	No
Fields included in **trafficStatistic**:			
device.deviceId	A value that uniquely identifies a device within a network	String	Yes

bytesSent	The total bytes sent for all currently open TCP/UDP connections between the WAN and a given device.	Long	Yes
bytesReceived	The total bytes received for all currently open TCP/UDP connections between the WAN and a given device.	Long	Yes

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
     "trafficStatistics": [
         {
             "trafficStatistic": {
   "device": {
       "deviceId":"0b6ec788-71a7-476f-85e3-9729d36bf844"
   },
                 "bytesSent":9223372036854775807,
                 "bytesReceived":9223372036854775807
             }
         },
         {
             "trafficStatistic": {
   "device": {
       "deviceId":"748517b7-fbc8-4454-9fab-c28019a12731"
   },
                 "bytesSent":9223372036854775777,
                 "bytesReceived":9223372036854775779
             }
         }
     ]
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	The network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Event Subscription

Create an event subscription for a network

Purpose

It creates an event subscription with specified information.

An event subscription for a network is uniquely identified as a unique combination of the client, account, network and event type (i.e. DEVICE_JOIEND_NETWORK or DEVICE_LEFT_NETWORK).

Request

POST /cloud/event-service/rest/clients/self/accounts/self/networks/<networkId>/eventsubscriptions

The <networkId> is the value that uniquely identifies a network.

Here is the raw http request:

```
POST /cloud/event-service/rest
/clients/self/accounts/self/networks/networkId-goes-here/eventsubscriptions

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Content-Type: application/json; charset=UTF-8
Accept: application/json

json-content-goes-here
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, **Content-Type** and **Accept** header fields are specified as shown above.

Request: Entity Body

The request body contains a representation of the event subscription being created, as a JSON document.

Elements/fields to be included in the request body:

Name	Description	Data Type	Required?
name	The friendly name of the event subscription.	String	No
eventType	The type of the event being subscribed. Allowed Values: **DEVICE_JOIEND_NETWORK**, **DEVICE_LEFT_NETWORK**	String	Yes
timeFilters	A list of one or more **timeFilter(s)**.	List	Yes
timeFilter	The schedule specifying when the event should be honored (i.e. notify the subscriber/client) and ignored during each calendar week within the specified date/time range. Multiple time filters can be specified for an event subscription. However, the date/time range should not overlap across multiple time filters.	Complex	Yes
timeFilter.startAt **timeFilter.endAt**	The date/time range when the filter / schedule is effective.	DateTime	Yes
timeFilter.monday **timeFilter.tuesday** **timeFilter.wednesday** **timeFilter.thursday** **timeFilter.friday** **timeFilter.saturday** **timeFilter.sunday**	Each string member represents a schedule for a day of the week. The string must be exactly 48 characters long. Each character represents a 30-minute interval during the day, beginning at midnight. A "0" character indicates that the event should be ignored during the interval; a "1" indicates that event should be honored (i.e. notify the subscriber/client). No other characters may appear in the string. For example, the following string indicates that the event should only be honored (i.e. notify the subscriber/client) between 9 AM and 9 PM: "000000000000000000111111111111111111111111000000"	String	No

Example Request Body (JSON):

```
{
   "eventSubscription":{
      "name":"firendly name of the event subscription",
      "eventType":"DEVICE_JOINED_NETWORK",
      "timeFilters":[
         {
            "timeFilter":{
               "startAt":"2012-06-19T20:38:00Z",
               "endAt":"2012-06-30T20:38:00Z",
               "monday":"000000000000000000111111111111111111111111000000",
               "tuesday":"000000000000000000111111111111111111111111000000",
               "wednesday":"000000000000000000111111111111111111111111000000",
               "thursday":"000000000000000000111111111111111111111111000000",
               "friday":"000000000000000000111111111111111111111111000000",
               "saturday":"000000000000000000111111111111111111111111000000",
               "sunday":"000000000000000000111111111111111111111111000000"
            }
         },
         {
            "timeFilter":{
               "startAt":"2012-07-19T20:38:00Z",
               "endAt":"2012-07-30T20:38:00Z",
               "monday":"000000000000000000111111111111111111111111000000",
               "tuesday":"000000000000000000111111111111111111111111000000",
               "wednesday":"000000000000000000111111111111111111111111000000",
               "thursday":"000000000000000000111111111111111111111111000000",
               "friday":"000000000000000000111111111111111111111111000000",
               "saturday":"000000000000000000111111111111111111111111000000",
               "sunday":"000000000000000000111111111111111111111111000000"
            }
         }
      ]
   }
}
```

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a partial representation of the newly created event subscription, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "eventSubscription":{
      "eventSubscriptionId":"A937F8C9-F379-440D-93E9-74A285E20C6F"
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.

504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
409	DUPLICATE_EVENT_SUBSCRIPTION	The event subscription could not be created because the submitted data would create a duplicate (non-unique) event subscription for the given client, account, network and the event type.
404	EVENT_SUBSCRIPTION_NOT_SUPPORTED	The specified network (router) does not support the event subscription for the specified event type.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Get all event subscriptions for a network

Purpose

Gets (aka reads or retrieves) all the event subscriptions associated with a specific network.

Request

GET /cloud/event-service/rest/clients/self/accounts/self/networks/<networkId>/eventsubscriptions

The <networkId> is the value that uniquely identifies a network.

Here is the raw http request:

```
GET /cloud/event-service/rest
/clients/self/accounts/self/networks/networkId-goes-here/eventsubscriptions

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the requested event subscriptions, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?

eventSubscriptionId	A unique ID of the event subscription.	String	Yes
name	The friendly name of the event subscription.	String	No
eventType	The type of the event being subscribed. Allowed Values: **DEVICE_JOIEND_NETWORK**, **DEVICE_LEFT_NETWORK**	String	Yes
timeFilters	A list of one or more **timeFilter(s)**.	List	Yes
timeFilter	The schedule specifying when the event should be honored (i.e. notify the subscriber/client) and ignored during each calendar week within the specified date/time range. Multiple time filters can be specified for an event subscription. However, the date/time range should not overlap across multiple time filters.	Complex	Yes
timeFilter.startAt **timeFilter.endAt**	The date/time range when the filter / schedule is effective.	DateTime	Yes
timeFilter.monday **timeFilter.tuesday** **timeFilter.wednesday** **timeFilter.thursday** **timeFilter.friday** **timeFilter.saturday** **timeFilter.sunday**	Each string member represents a schedule for a day of the week. The string must be exactly 48 characters long. Each character represents a 30-minute interval during the day, beginning at midnight. A "0" character indicates that the event should be ignored during the interval; a "1" indicates that event should be honored (i.e. notify the subscriber/client). No other characters may appear in the string. For example, the following string indicates that the event should only be honored (i.e. notify the subscriber/client) between 9 AM and 9 PM: "000000000000000000111111111111111111111111000000"	String	No

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "eventSubscriptions":[
      {
         "eventSubscription":{
            "eventSubscriptionId":"A937F8C9-F379-440D-93E9-74A285E20C6F",
            "name":"firendly name of the event subscription",
            "eventType":"DEVICE_JOINED_NETWORK",
            "timeFilters":[
               {
                  "timeFilter":{
                     "startAt":"2012-06-19T20:38:00Z",
                     "endAt":"2012-06-30T20:38:00Z",
                     "monday":"000000000000000000111111111111111111111111000000",
                     "tuesday":"000000000000000000111111111111111111111111000000",
                     "wednesday":"000000000000000000111111111111111111111111000000",
                     "thursday":"000000000000000000111111111111111111111111000000",
                     "friday":"000000000000000000111111111111111111111111000000",
                     "saturday":"000000000000000000111111111111111111111111000000",
                     "sunday":"000000000000000000111111111111111111111111000000"
                  }
               },
               {
                  "timeFilter":{
                     "startAt":"2012-07-19T20:38:00Z",
                     "endAt":"2012-07-30T20:38:00Z",
                     "monday":"000000000000000000111111111111111111111111000000",
                     "tuesday":"000000000000000000111111111111111111111111000000",
                     "wednesday":"000000000000000000111111111111111111111111000000",
                     "thursday":"000000000000000000111111111111111111111111000000",
                     "friday":"000000000000000000111111111111111111111111000000",
                     "saturday":"000000000000000000111111111111111111111111000000",
                     "sunday":"000000000000000000111111111111111111111111000000"
                  }
               }
            ]
         }
      },
      {
         "eventSubscription":{
            "eventSubscriptionId":"B80E49D8-776C-440E-A613-68D5F0FD65FE",
            "name":"firendly name of the event subscription",
            "eventType":"DEVICE_LEFT_NETWORK",
            "timeFilters":[
               {
                  "timeFilter":{
                     "startAt":"2012-06-19T20:38:00Z",
                     "endAt":"2012-06-30T20:38:00Z",
                     "monday":"000000000000000000000000000000000000000000000000",
                     "tuesday":"000000000000000000000000000000000000000000000000",
                     "wednesday":"000000000000000000000000000000000000000000000000",
                     "thursday":"000000000000000000000000000000000000000000000000",
                     "friday":"000000000000000000000000000000000000000000000000",
                     "saturday":"000000000000000000000000000001111111111000000000",
                     "sunday":"000000000000000000111111111111111111111111000000"
                  }
               }
            ]
         }
      }
   ]
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
404	EVENT_SUBSCRIPTION_NOT_SUPPORTED	The specified network (router) does not support the event subscription for the specified event type.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Create an event subscription for a device

Purpose

It creates an event subscription with specified information.

An event subscription for a device is uniquely identified as a unique combination of the client, account, network, device and event type (i.e. DEVICE_JOIEND_NETWORK or DEVICE_LEFT_NETWORK).

Request

POST /cloud/event-service/rest/clients/self/accounts/self/networks/<networkId>/devices/<deviceId>/eventsubscriptions

The <networkId> is the value that uniquely identifies a network.
The <deviceId> is the value that uniquely identifies a device within a network.

Here is the raw http request:

```
POST /cloud/event-service/rest
/clients/self/accounts/self/networks/networkId-goes-here/devices/deviceId-goes-here/eventsubscriptions

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Content-Type: application/json; charset=UTF-8
Accept: application/json

json-content-goes-here
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, **Content-Type** and **Accept** header fields are specified as shown above.

Request: Entity Body

The request body contains a representation of the event subscription to be created, as a JSON document.

Elements/fields to be included in the request body:

name	The friendly name of the event subscription.	String	No
eventType	The type of the event being subscribed. Allowed Values: **DEVICE_JOIEND_NETWORK**, **DEVICE_LEFT_NETWORK**	String	Yes

timeFilters	A list of one or more **timeFilter(s)**.	List	Yes
timeFilter	The schedule specifying when the event should be honored (i.e. notify the subscriber/client) and ignored during each calendar week within the specified date/time range. Multiple time filters can be specified for an event subscription. However, the date/time range should not overlap across multiple time filters.	Complex	Yes
timeFilter.startAt **timeFilter.endAt**	The date/time range when the filter / schedule is effective.	DateTime	Yes
timeFilter.monday **timeFilter.tuesday** **timeFilter.wednesday** **timeFilter.thursday** **timeFilter.friday** **timeFilter.saturday** **timeFilter.sunday**	Each string member represents a schedule for a day of the week. The string must be exactly 48 characters long. Each character represents a 30-minute interval during the day, beginning at midnight. A "0" character indicates that the event should be ignored during the interval; a "1" indicates that event should be honored (i.e. notify the subscriber/client). No other characters may appear in the string. For example, the following string indicates that the event should only be honored (i.e. notify the subscriber/client) between 9 AM and 9 PM: "000000000000000000111111111111111111111111000000"	String	No

Example Request Body (JSON):

```
{
   "eventSubscription":{
      "name":"firendly name of the event subscription",
      "eventType":"DEVICE_JOINED_NETWORK",
      "timeFilters":[
         {
            "timeFilter":{
               "startAt":"2012-06-19T20:38:00Z",
               "endAt":"2012-06-30T20:38:00Z",
               "monday":"000000000000000000111111111111111111111111000000",
               "tuesday":"000000000000000000111111111111111111111111000000",
               "wednesday":"000000000000000000111111111111111111111111000000",
               "thursday":"000000000000000000111111111111111111111111000000",
               "friday":"000000000000000000111111111111111111111111000000",
               "saturday":"000000000000000000111111111111111111111111000000",
               "sunday":"000000000000000000111111111111111111111111000000"
            }
         }
      ]
   }
}
```

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a partial representation of the newly created event subscription, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "eventSubscription":{
      "eventSubscriptionId":"BE931A56-B146-4F41-BEF1-8A66A21BCAB8"
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
404	DEVICE_NOT_FOUND	A device with the specified device ID is not found in the specified network.
409	DUPLICATE_EVENT_SUBSCRIPTION	The event subscription could not be created because the submitted data would create a duplicate (non-unique) event subscription for the given device and event type.
404	EVENT_SUBSCRIPTION_NOT_SUPPORTED	The specified network (router) does not support the event subscription for the specified event type.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Get all event subscriptions for a device

Purpose

Gets (aka reads or retrieves) all the event subscriptions associated with a specific device within a network.

Request

GET /cloud/event-service/rest/clients/self/accounts/self/networks/<networkId>/devices/<deviceId>/eventsubscriptions

The <networkId> is the value that uniquely identifies a network.
The <deviceId> is the value that uniquely identifies a device.

Here is the raw http request:

```
GET /cloud/event-service/rest
/clients/self/accounts/self/networks/networkId-goes-here/devices/deviceId-goes-here/eventsubscriptions

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the requested event subscriptions, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
eventSubscriptionId	A unique ID of the event subscription.	String	Yes
name	The friendly name of the event subscription.	String	No
eventType	The type of the event being subscribed. Allowed Values: **DEVICE_JOIEND_NETWORK**, **DEVICE_LEFT_NETWORK**	String	Yes
timeFilters	A list of one or more **timeFilter(s)**.	List	Yes
timeFilter	The schedule specifying when the event should be honored (i.e. notify the subscriber/client) and ignored during each calendar week within the specified date/time range. Multiple time filters can be specified for an event subscription. However, the date/time range should not overlap across multiple time filters.	Complex	Yes
timeFilter.startAt **timeFilter.endAt**	The date/time range when the filter / schedule is effective.	DateTime	Yes
timeFilter.monday **timeFilter.tuesday** **timeFilter.wednesday** **timeFilter.thursday** **timeFilter.friday** **timeFilter.saturday** **timeFilter.sunday**	Each string member represents a schedule for a day of the week. The string must be exactly 48 characters long. Each character represents a 30-minute interval during the day, beginning at midnight. A "0" character indicates that the event should be ignored during the interval; a "1" indicates that event should be honored (i.e. notify the subscriber/client). No other characters may appear in the string. For example, the following string indicates that the event should only be honored (i.e. notify the subscriber/client) between 9 AM and 9 PM: "000000000000000000111111111111111111111111000000"	String	No

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "eventSubscriptions":[
      {
         "eventSubscription":{
            "eventSubscriptionId":"96E5B04A-3500-45EA-8A47-B086129D1617",
            "name":"firendly name of the event subscription",
            "eventType":"DEVICE_JOINED_NETWORK",
            "timeFilters":[
               {
                  "timeFilter":{
                     "startAt":"2012-06-19T20:38:00Z",
                     "endAt":"2012-06-30T20:38:00Z",
                     "monday":"000000000000000000111111111111111111111111000000",
                     "tuesday":"000000000000000000111111111111111111111111000000",
                     "wednesday":"000000000000000000111111111111111111111111000000",
                     "thursday":"000000000000000000111111111111111111111111000000",
                     "friday":"000000000000000000111111111111111111111111000000",
                     "saturday":"000000000000000000111111111111111111111111000000",
                     "sunday":"000000000000000000111111111111111111111111000000"
                  }
               }
            ]
         }
      },
      {
         "eventSubscription":{
            "eventSubscriptionId":"BE931A56-B146-4F41-BEF1-8A66A21BCAB8",
            "name":"firendly name of the event subscription",
            "eventType":"DEVICE_LEFT_NETWORK",
            "timeFilters":[
               {
                  "timeFilter":{
                     "startAt":"2012-12-01T00:00:00Z",
                     "endAt":"2012-12-31T23:59:59Z",
                  }
               }
            ]
         }
      }
   ]
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
404	DEVICE_NOT_FOUND	A device with the specified device ID is not found in the specified network.
404	EVENT_SUBSCRIPTION_NOT_SUPPORTED	The specified network (router) does not support the event subscription for the specified event type.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Get an event subscription

Purpose

Gets (aka reads or retrieves) an event subscription.

Request

GET /cloud/event-service/rest/eventsubscriptions/<eventSubscriptionId>

The <eventSubscriptionId> is the value that uniquely identifies an event subscription.

Request

Here is the raw http request:

```
GET /cloud/event-service/rest/eventsubscriptions/eventSubscriptionId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the requested event subscription, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
eventSubscriptionId	A unique ID of the event subscription.	String	Yes
name	The friendly name of the event subscription.	String	No
eventType	The type of the event being subscribed. Allowed Values: **DEVICE_JOIEND_NETWORK**, **DEVICE_LEFT_NETWORK**	String	Yes
timeFilters	A list of one or more **timeFilter(s)**.	List	Yes

timeFilter	The schedule specifying when the event should be honored (i.e. notify the subscriber/client) and ignored during each calendar week within the specified date/time range. Multiple time filters can be specified for an event subscription. However, the date/time range should not overlap across multiple time filters.	Complex	Yes
timeFilter.startAt **timeFilter.endAt**	The date/time range when the filter / schedule is effective.	DateTime	Yes
timeFilter.monday **timeFilter.tuesday** **timeFilter.wednesday** **timeFilter.thursday** **timeFilter.friday** **timeFilter.saturday** **timeFilter.sunday**	Each string member represents a schedule for a day of the week. The string must be exactly 48 characters long. Each character represents a 30-minute interval during the day, beginning at midnight. A "0" character indicates that the event should be ignored during the interval; a "1" indicates that event should be honored (i.e. notify the subscriber/client). No other characters may appear in the string. For example, the following string indicates that the event should only be honored (i.e. notify the subscriber/client) between 9 AM and 9 PM: "000000000000000000111111111111111111111111000000"	String	No
device.deviceId	The unique ID of the device the event subscription is associated with.	String	No
network.networkId	The unique ID of the network the event subscription is associated with.	String	Yes
account.accountId	The unique ID of the account the event subscription is associated with.	String	Yes
client.clientId	The unique ID of the client the event subscription is associated with.	String	Yes

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON) (event subscription for a network):

```
{
    "eventSubscription":{
       "eventSubscriptionId":"0CE74CB4-4C05-47B8-AB54-3C59967F3C89",
       "name":"firendly name of the event subscription",
       "eventType":"DEVICE_JOINED_NETWORK",
       "timeFilters":[
          {
             "timeFilter":{
                "startAt":"2012-06-19T20:38:00Z",
                "endAt":"2012-06-30T20:38:00Z",
                "monday":"000000000000000000111111111111111111111111000000",
                "tuesday":"000000000000000000111111111111111111111111000000",
                "wednesday":"000000000000000000111111111111111111111111000000",
                "thursday":"000000000000000000111111111111111111111111000000",
                "friday":"000000000000000000111111111111111111111111000000",
                "saturday":"000000000000000000111111111111111111111111000000",
                "sunday":"000000000000000000111111111111111111111111000000"
             }
          },
          {
             "timeFilter":{
                "startAt":"2012-07-19T20:38:00Z",
                "endAt":"2012-07-30T20:38:00Z",
                "monday":"000000000000000000111111111111111111111111000000",
                "tuesday":"000000000000000000111111111111111111111111000000",
                "wednesday":"000000000000000000111111111111111111111111000000",
                "thursday":"000000000000000000111111111111111111111111000000",
                "friday":"000000000000000000111111111111111111111111000000",
                "saturday":"000000000000000000111111111111111111111111000000",
                "sunday":"000000000000000000111111111111111111111111000000"
             }
          }
       ],
       "network":{
          "networkId":"network-3535FA4746074CD3954EE3F6AF3AC8C652331EC9@ciscoconnectcloud.com"
       },
       "account":{
          "accountId":"AEFFC7F3-5644-45DB-A338-0B7D1CB9EBE7"
       },
       "client":{
          "clientId":"AA3466DB-722B-5BB4-45CE-77E7486B17ED"
       }
    }
}
```

Example Response Body (on Success) (JSON) (event subscription for a device within a network):

```
{
    "eventSubscription":{
       "eventSubscriptionId":"5E548B5D-C817-448C-843C-91FB456AE5CA",
       "name":"firendly name of the event subscription",
       "eventType":"DEVICE_JOINED_NETWORK",
       "timeFilters":[
          {
             "timeFilter":{
                "startAt":"2012-06-19T20:38:00Z",
                "endAt":"2012-06-30T20:38:00Z",
                "monday":"000000000000000000111111111111111111111111000000",
                "tuesday":"000000000000000000111111111111111111111111000000",
                "wednesday":"000000000000000000111111111111111111111111000000",
                "thursday":"000000000000000000111111111111111111111111000000",
                "friday":"000000000000000000111111111111111111111111000000",
                "saturday":"000000000000000000111111111111111111111111000000",
                "sunday":"000000000000000000111111111111111111111111000000"
             }
          }
       ],
       "device":{
          "deviceId":"0b6ec788-71a7-476f-85e3-9729d36bf844"
       },
       "network":{
          "networkId":"network-3535FA4746074CD3954EE3F6AF3AC8C652331EC9@ciscoconnectcloud.com"
       },
       "account":{
          "accountId":"AEFFC7F3-5644-45DB-A338-0B7D1CB9EBE7"
       },
       "client":{
          "clientId":"AA3466DB-722B-5BB4-45CE-77E7486B17ED"
       }
    }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	EVENT_SUBSCRIPTION_NOT_FOUND	An event subscription with the specified event subscription ID is not found.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"EVENT_SUBSCRIPTION_NOT_FOUND",
        "message":"An event subscription with the specified event subscription ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Delete an event subscription

Purpose

Deletes an event subscription.

Request

DELETE /cloud/event-service/rest/eventsubscriptions/<eventSubscriptionId>

The <eventSubscriptionId> is a value that uniquely identifies an event subscription.

Here is the raw http request:

```
DELETE /cloud/event-service/rest/eventsubscriptions/{color:#0000ff}<eventSubscriptionId>{color}

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns partial representation of the deleted event subscription, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "eventSubscription":{
      "eventSubscriptionId":"0CE74CB4-4C05-47B8-AB54-3C59967F3C89"
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	EVENT_SUBSCRIPTION_NOT_FOUND	The event subscription with the specified event subscription ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"EVENT_SUBSCRIPTION_NOT_FOUND",
        "message":"An event subscription with the specified event subscription ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Event Notification Callback

Event Notification Callback API Details

Purpose

The clients that subscribe to the events (**DEVICE_JOIEND_NETWORK**, **DEVICE_LEFT_NETWORK**) will register a callback URL (must be HTTPS) with Linksys Smart Wifi cloud server. Linksys Smart Wifi cloud server will post the event details to the callback URL when an event (that meets the client's subscription criteria) happens.

The event data will be posted to the callback URL in the format specified below.

Request

Here is the raw http callback request:

```
POST callback-url-goes-here

Content-Type: application/json; charset=UTF-8

json-content-goes-here
```

Request: Entity Body

The request body contains the following information needed by the client / third-party system to perform notification, as a JSON document.

Elements/fields to be included in the request body:

Name	Description	Data Type	Required?
event.eventId	A unique ID of the event.	String	Yes
event.eventType	The type of the event (**DEVICE_JOINED_NETWORK** or **DEVICE_LEFT_NETWORK**).	String	Yes
event.happendAt	The date and time (in UTC) when the event occurred.	DateTime	Yes
event..device.deviceId	The device which triggered the event.	String	Yes
event..network.networkId	The network where the event happend.	String	Yes
eventSubscription.eventSubscriptionId	A unique ID of the event subscription causing the callback.	String	Yes
eventSubscription.account.accountId	A unique ID of the account associated with the event subscription.	String	Yes
eventSubscription.client.clientId	A unique ID of the client associated with the event subscription.	String	Yes
eventSubscription.client.clientSecret	The secret of the client registered with Linksys Smart Wifi cloud server. The client should use this to verify that the call is indeed coming from the Linksys Smart Wifi cloud server.	String	Yes

In future, additional fields may be included in the request. The callback handler on the client side must be coded to ignore the fields it does not recognize.

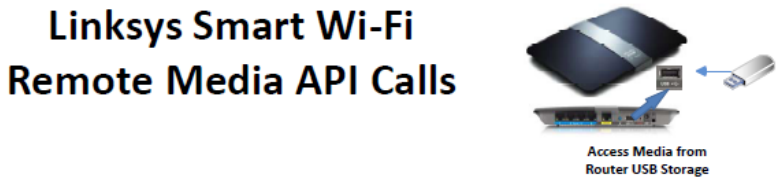
Example Request Body (JSON):

```
{
    "eventNotification":{
       "event":{
          "eventId":"0B6EC788-71A7-476F-85E3-9729D36BF844",
          "eventType":"DEVICE_JOINED_NETWORK",
          "happendAt":"2009-04-28T21:22:56Z",
          "device":{
             "deviceId":"0b6ec788-71a7-476f-85e3-9729d36bf844"
          },
          "network":{
             "networkId":"network-3535FA4746074CD3954EE3F6AF3AC8C652331EC9@ciscoconnectcloud.com"
          }
       },
       "eventSubscription":{
          "eventSubscriptionId":"5E548B5D-C817-448C-843C-91FB456AE5CA",
          "account":{
             "accountId":"BD2C93F8-EDED-4375-91BE-3619C63C4D2E"
          },
          "client":{
             "clientId":"AF2F93F8-ADEA-7132-19BE-7754C63C4D2E",
             "clientSecret":"122323223332DDDSSDDDSSDD"
          }
       }
    }
}
```

Media Service

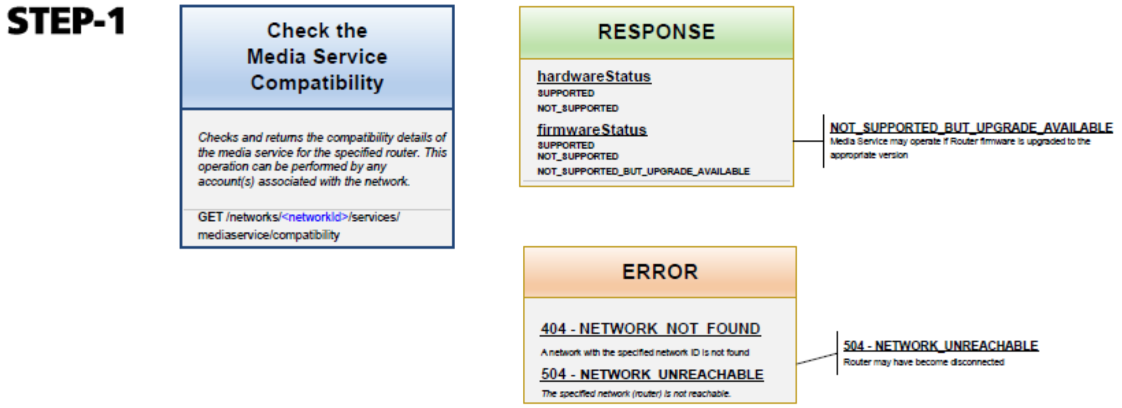
Remote Media Service Walk-through

The steps provided below will walk you through the process required to call the Remote Media Service APIs



Linksys Smart Wi-Fi
Remote Media API Calls

Access Media from
Router USB Storage



STEP-1

Check the Media Service Compatibility

Checks and returns the compatibility details of the media service for the specified router. This operation can be performed by any account(s) associated with the network.

GET /networks/<networkId>/services/mediaservice/compatibility

RESPONSE

hardwareStatus
SUPPORTED
NOT_SUPPORTED

firmwareStatus
SUPPORTED
NOT_SUPPORTED
NOT_SUPPORTED_BUT_UPGRADE_AVAILABLE

NOT_SUPPORTED_BUT_UPGRADE_AVAILABLE
Media Service may operate if Router firmware is upgraded to the appropriate version

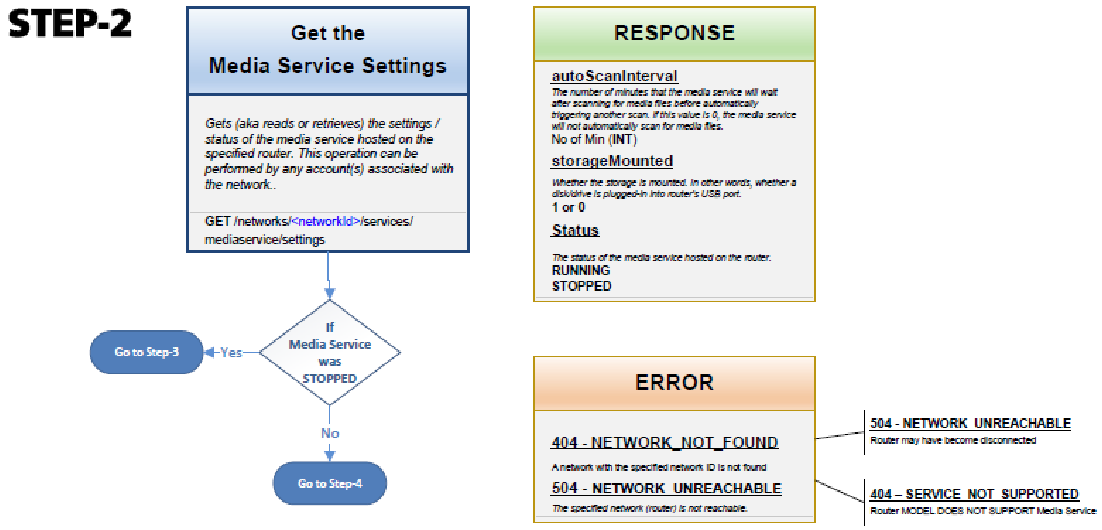
ERROR

404 - NETWORK_NOT_FOUND
A network with the specified network ID is not found

504 - NETWORK_UNREACHABLE
The specified network (router) is not reachable.

504 - NETWORK_UNREACHABLE
Router may have become disconnected

Linksys Smart Wi-Fi
Remote Media API Calls



STEP-2

Get the Media Service Settings

Gets (aka reads or retrieves) the settings / status of the media service hosted on the specified router. This operation can be performed by any account(s) associated with the network..

GET /networks/<networkId>/services/mediaservice/settings

RESPONSE

autoScanInterval

The number of minutes that the media service will wait after scanning for media files before automatically triggering another scan. If this value is 0, the media service will not automatically scan for media files.

No of Min (**INT**)

storageMounted

Whether the storage is mounted. In other words, whether a disk/drive is plugged-in into router's USB port.

1 or 0

Status

The status of the media service hosted on the router.

RUNNING

STOPPED

ERROR

404 - NETWORK_NOT_FOUND

A network with the specified network ID is not found

504 - NETWORK_UNREACHABLE

The specified network (router) is not reachable.

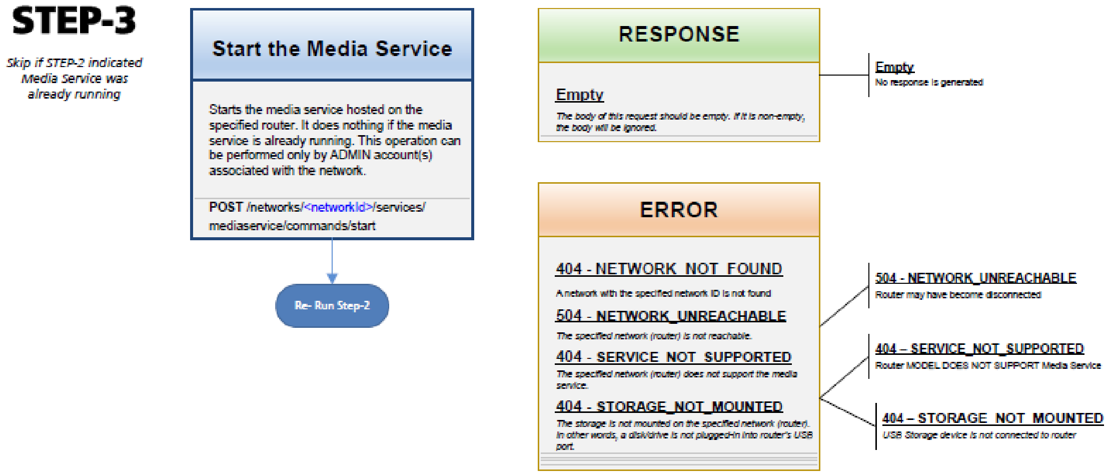
504 - NETWORK_UNREACHABLE

Router may have become disconnected

404 – SERVICE_NOT_SUPPORTED

Router MODEL DOES NOT SUPPORT Media Service

Linksys Smart Wi-Fi
Remote Media API Calls



STEP-3

Skip if STEP-2 indicated Media Service was already running

Start the Media Service

Starts the media service hosted on the specified router. It does nothing if the media service is already running. This operation can be performed only by ADMIN account(s) associated with the network.

POST /networks/<networkId>/services/mediaservice/commands/start

RESPONSE

Empty

The body of this request should be empty. If it is non-empty, the body will be ignored.

Empty

No response is generated

ERROR

404 - NETWORK_NOT_FOUND

A network with the specified network ID is not found

504 - NETWORK_UNREACHABLE

The specified network (router) is not reachable.

404 - SERVICE_NOT_SUPPORTED

The specified network (router) does not support the media service.

404 - STORAGE_NOT_MOUNTED

The storage is not mounted on the specified network (router). In other words, a disk/drive is not plugged-in into router's USB port.

504 - NETWORK_UNREACHABLE

Router may have become disconnected

404 – SERVICE_NOT_SUPPORTED

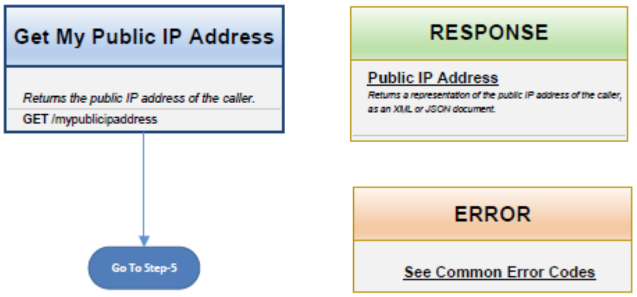
Router MODEL DOES NOT SUPPORT Media Service

404 – STORAGE_NOT_MOUNTED

USB Storage device is not connected to router

Linksys Smart Wi-Fi
Remote Media API Calls

STEP-4



Get My Public IP Address

Returns the public IP address of the caller.

GET /mypublicipaddress

RESPONSE

Public IP Address

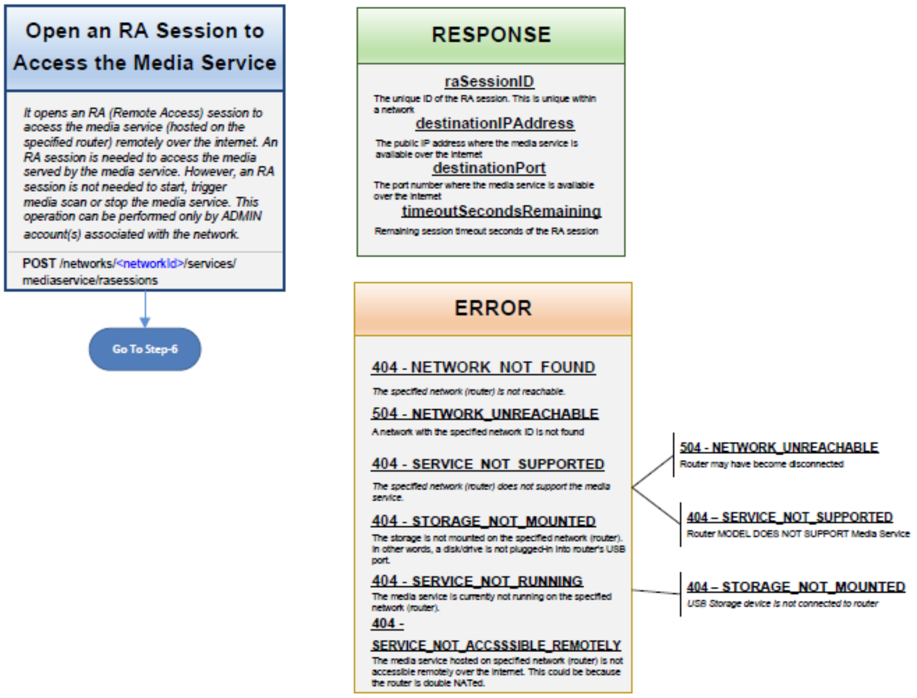
Returns a representation of the public IP address of the caller, as an XML or JSON document.

ERROR

See Common Error Codes

Linksys Smart Wi-Fi
Remote Media API Calls

STEP-5



Open an RA Session to Access the Media Service

It opens an RA (Remote Access) session to access the media service (hosted on the specified router) remotely over the internet. An RA session is needed to access the media served by the media service. However, an RA session is not needed to start, trigger media scan or stop the media service. This operation can be performed only by ADMIN account(s) associated with the network.

POST /networks/<networkId>/services/mediaservice/rasessions

RESPONSE

raSessionID

The unique ID of the RA session. This is unique within a network

destinationIPAddress

The public IP address where the media service is available over the internet

destinationPort

The port number where the media service is available over the internet

timeoutSecondsRemaining

Remaining session timeout seconds of the RA session

ERROR

404 - NETWORK_NOT_FOUND

The specified network (router) is not reachable.

504 - NETWORK_UNREACHABLE

A network with the specified network ID is not found

404 - SERVICE_NOT_SUPPORTED

The specified network (router) does not support the media service.

404 - STORAGE_NOT_MOUNTED

The storage is not mounted on the specified network (router). In other words, a disk/drive is not plugged-in into router's USB port.

404 - SERVICE_NOT_RUNNING

The media service is currently not running on the specified network (router).

404 - SERVICE_NOT_ACCSSSIBLE_REMOTELY

The media service hosted on specified network (router) is not accessible remotely over the internet. This could be because the router is double NATed.

504 - NETWORK_UNREACHABLE

Router may have become disconnected

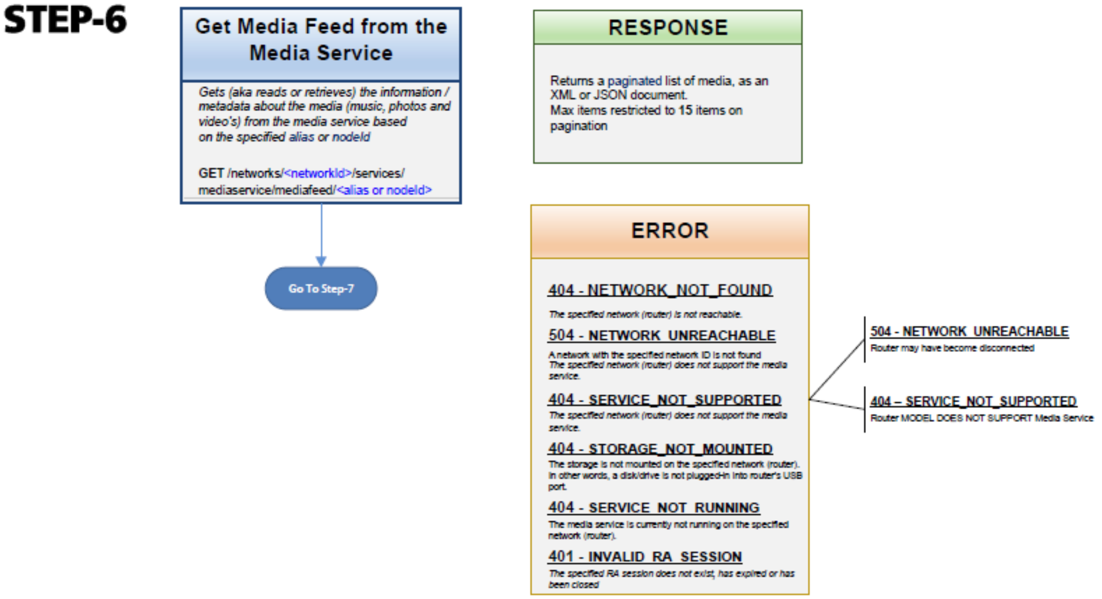
404 – SERVICE_NOT_SUPPORTED

Router MODEL DOES NOT SUPPORT Media Service

404 – STORAGE_NOT_MOUNTED

USB Storage device is not connected to router

Linksys Smart Wi-Fi
Remote Media API Calls



STEP-6

Get Media Feed from the Media Service

Gets (aka reads or retrieves) the information / metadata about the media (music, photos and video's) from the media service based on the specified alias or nodeId

GET /networks/<networkId>/services/mediaservice/mediafeed/<alias or nodeId>

RESPONSE

Returns a paginated list of media, as an XML or JSON document.
Max items restricted to 15 items on pagination

ERROR

404 - NETWORK_NOT_FOUND

The specified network (router) is not reachable.

504 - NETWORK_UNREACHABLE

A network with the specified network ID is not found
The specified network (router) does not support the media service.

404 - SERVICE_NOT_SUPPORTED

The specified network (router) does not support the media service.

404 - STORAGE_NOT_MOUNTED

The storage is not mounted on the specified network (router). In other words, a disk/drive is not plugged-in into router's USB port.

404 - SERVICE_NOT_RUNNING

The media service is currently not running on the specified network (router).

401 - INVALID_RA_SESSION

The specified RA session does not exist, has expired or has been closed

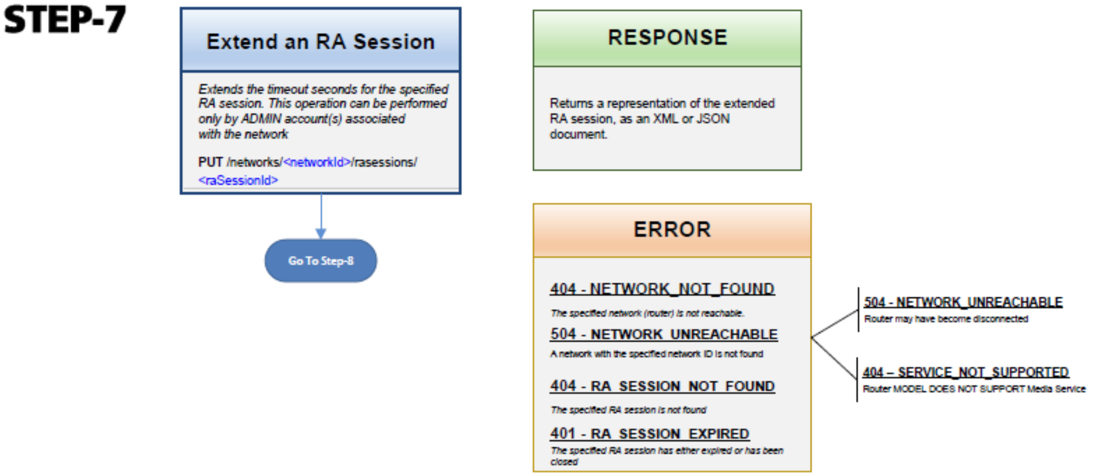
504 - NETWORK_UNREACHABLE

Router may have become disconnected

404 – SERVICE_NOT_SUPPORTED

Router MODEL DOES NOT SUPPORT Media Service

Linksys Smart Wi-Fi
Remote Media API Calls



STEP-7

Extend an RA Session

Extends the timeout seconds for the specified RA session. This operation can be performed only by ADMIN account(s) associated with the network

PUT /networks/<networkId>/rasessions/<raSessionId>

RESPONSE

Returns a representation of the extended RA session, as an XML or JSON document.

ERROR

404 - NETWORK_NOT_FOUND

The specified network (router) is not reachable.

504 - NETWORK_UNREACHABLE

A network with the specified network ID is not found

404 - RA_SESSION_NOT_FOUND

The specified RA session is not found

401 - RA_SESSION_EXPIRED

The specified RA session has either expired or has been closed

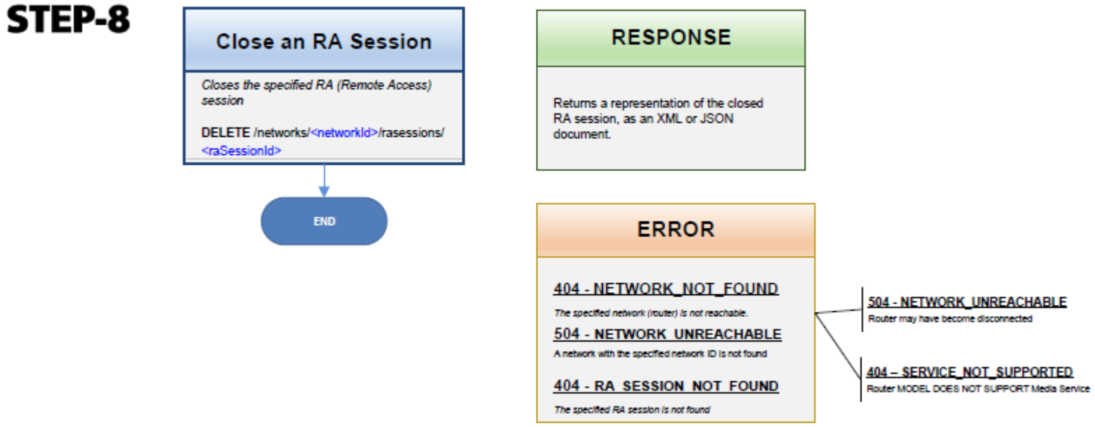
504 - NETWORK_UNREACHABLE

Router may have become disconnected

404 – SERVICE_NOT_SUPPORTED

Router MODEL DOES NOT SUPPORT Media Service

Linksys Smart Wi-Fi Remote Media API Calls



STEP-8

Close an RA Session

Closes the specified RA (Remote Access) session

DELETE /networks/<networkId>/rasessions/<raSessionId>

END

RESPONSE

Returns a representation of the closed RA session, as an XML or JSON document.

ERROR

404 - NETWORK_NOT_FOUND

The specified network (router) is not reachable.

504 - NETWORK_UNREACHABLE

A network with the specified network ID is not found

404 - RA_SESSION_NOT_FOUND

The specified RA session is not found

504 - NETWORK_UNREACHABLE

Router may have become disconnected

404 – SERVICE_NOT_SUPPORTED

Router MODEL DOES NOT SUPPORT Media Service

Check the media service compatibility

Purpose

Checks and returns the compatibility details of the media service for the specified router. This operation can be performed by any account(s) associated with the network.

Request

GET /cloud/device-service/rest/networks/<networkId>/services/mediaservice/compatibility

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
GET /cloud/device-service/rest/networks/networkId-goes-here/services/mediaservice/compatibility

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is perfomred successfully.

Response: Entity Body (on Success)

Returns a representation of the service compatibility, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
hardwareStatus	The status of the router hardware for supporting the media service . Valid values: **SUPPORTED** **NOT_SUPPORTED**	String	Yes

firmwareStatus	The status of the router firmware for supporting the media service . Valid values: **SUPPORTED** **NOT_SUPPORTED** **NOT_SUPPORTED_BUT_UPGRADE_AVAILABLE**	String	Yes

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "serviceCompatibility":{
      "hardwareStatus":"SUPPORTED",
      "firmwareStatus":"SUPPORTED"
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Get the media service settings

Purpose

Gets (aka reads or retrieves) the settings / status of the media service hosted on the specified router. This operation can be performed by any account(s) associated with the network.

Request

GET /cloud/device-service/rest/networks/<networkId>/services/mediaservice/settings

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
GET /cloud/device-service/rest/networks/networkId-goes-here/services/mediaservice/settings

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is perfomred successfully.

Response: Entity Body (on Success)

Returns a representation of the media service settings, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
autoScanInterval	The number of minutes that the media service will wait after scanning for media files before automatically triggering another scan. If this value is 0, the media service will not automatically scan for media files.	Integer	Yes
lastScannedAt	The last time that the media service scanned for media files. If no scan has been performed, this value will not be present	DateTime	No
storageMounted	Whether the storage is mounted. In other words, whether a disk/drive is plugged-in into router's USB port.	Boolean	Yes
status	The status of the media service hosted on the router. Valid values: **RUNNING** **STOPPED**	String	Yes

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "mediaServiceSettings":{
      "autoScanInterval":120,
      "lastScannedAt":"2012-10-25T00:03:45Z",
      "storageMounted":true,
      "status":"RUNNING"
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
404	SERVICE_NOT_SUPPORTED	The specified network (router) does not support the media service.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Update the media service settings

Purpose

Updates the settings of the media service hosted on the specified router. This operation can be performed only by ADMIN account(s) associated with the network.

Request

PUT /cloud/device-service/rest/networks/<networkId>/services/mediaservice/settings

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
PUT /cloud/device-service/rest/networks/networkId-goes-here/services/mediaservice/settings

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Content-Type: application/json; charset=UTF-8
Accept: application/json

json-content-goes-here
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, **Content-Type** and **Accept** header fields are specified as shown above.

Request: Entity Body

The request body contains a partial representation of the media service settings to be updated, as a JSON document.

Elements/fields to be included in the request body:

Name	Description	Data Type	Required?
autoScanInterval	The number of minutes that the media service will wait after scanning for media files before automatically triggering another scan. If this value is 0, the media service will not automatically scan for media files.	Integer	No

Example Request Body (JSON):

```
{
   "mediaServiceSettings":{
      "autoScanInterval":360
   }
}
```

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the updated media service settings, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "mediaServiceSettings":{
      "autoScanInterval":360
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
404	SERVICE_NOT_SUPPORTED	The specified network (router) does not support the media service.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Start the media service

Purpose

Starts the media service hosted on the specified router. It does nothing if the media service is already running. This operation can be performed only by ADMIN account(s) associated with the network.

Request

POST /cloud/device-service/rest/networks/<networkId>/services/mediaservice/commands/start

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
POST /cloud/device-service/rest/networks/networkId-goes-here/services/mediaservice/commands/start

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the executed service command, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "serviceCommand":{
      "name":"start"
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
404	SERVICE_NOT_SUPPORTED	The specified network (router) does not support the media service.
404	STORAGE_NOT_MOUNTED	The storage is not mounted on the specified network (router). In other words, a disk/drive is not plugged-in into router's USB port.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Stop the media service

toc

Purpose

Stops the media service hosted on the specified router. It does nothing if the media service is already stopped. This operation can be performed only by ADMIN account(s) associated with the network.

Request

POST /cloud/device-service/rest/networks/<networkId>/services/mediaservice/commands/stop

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
POST /cloud/device-service/rest/networks/networkId-goes-here/services/mediaservice/commands/stop

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the executed service command, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "serviceCommand":{
      "name":"stop"
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
404	SERVICE_NOT_SUPPORTED	The specified network (router) does not support the media service.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Trigger media scan

toc

Purpose

It instructs the media service hosted on the specified router to trigger a scan of the media files present on the mounted storage (USB drive or stick). This operation can be performed by any account(s) associated with the network.

Request

POST /cloud/device-service/rest/networks/<networkId>/services/mediaservice/commands/scan

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
POST /cloud/device-service/rest/networks/networkId-goes-here/services/mediaservice/commands/scan

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the executed service command, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "serviceCommand":{
      "name":"scan"
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
404	SERVICE_NOT_SUPPORTED	The specified network (router) does not support the media service.
404	STORAGE_NOT_MOUNTED	The storage is not mounted on the specified network (router). In other words, a disk/drive is not plugged-in into router's USB port.
404	SERVICE_NOT_RUNNING	The media service is currently not running on the specified network (router).

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Get media feed from the media service

Purpose

Gets (aka reads or retrieves) the metadata about the media (music, photos and videos) from the media service hosted on the specified router.

It returns a list of nodes contained in the specified node. A node represents either a real media (music, photo or video) file OR a container that contains other nodes.

An alias is a well-known / predefined node id.

List of supported Alias(s):

Alias	Description
MUSIC	music files
MUSIC_ALL	list of all music files.
MUSIC_PLAYLISTS	music by playlists
MUSIC_GENRE	music by genre

MUSIC_ARTISTS	music by artists
MUSIC_ALBUMS	music by albums
MUSIC_FOLDERS	music by folders
PICTURE	picture files
PICTURE_ALL	list of all picture files
PICTURE_PLAYLISTS	picture by playlists
PICTURE_FOLDERS	picture by folders
PICTURE_ALBUMS	picture by albums
VIDEO	video files
VIDEO_ALL	list of all videos
VIDEO_PLAYLISTS	video by playlists
VIDEO_GENRE	video by genre
VIDEO_ACTORS	video by actors
VIDEO_ALBUMS	video by albums
VIDEO_SERIES	video by series
VIDEO_FOLDERS	video by folders

Request

GET /cloud/device-service/rest/networks/<networkId>/services/mediaservice/mediafeed/<alias or nodeId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <nodeId> or <alias> is a value that uniquely identifies a node. A node represents either a real media (music, photo or video) file OR a container that contains other nodes.

Here is the raw http request:

```
GET /cloud/device-service/rest
/networks/networkId-goes-here/services/mediaservice/mediafeed/alias-or-nodeId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
X-Cisco-HN-RA-Session-Id: ra-session-id-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, **X-Cisco-HN-RA-Session-Id** and **Accept** header fields are specified as shown above.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a paginated list of nodes, as a JSON document. Max number of nodes returned in a response is currently capped at **15**.

Elements/fields included in the response body:

Name	Description	DataType	Required
paginationResult	The pagination details of the response returned.	String	Yes
title	The friendly name of the node specified in the request URI.	String	Yes
nodes	A list of nodes contained in the node specified in the request URI.	List of **node**	No

Fields included in each **node**			
nodeId	The unique id (assigned by media service) of the node. Example: 0$1$8I778	String	Yes
nodeType	The type of the node. Valid values are: **LEAF** - A real media (music, photo or video) file. **BRANCH** - A container that contains other nodes.	String	Yes
mediaType	The type of the media. Valid values are: **MUSIC**, **VIDEO**, **PICTURE**. This field is returned only for the **LEAF** node.	String	No
title	The friendly name of the node / media.	String	Yes
album	The name of the album the media belongs to. This field is returned only for the **LEAF** node.	String	No
artist	The name of the media artist. This field is returned only for the **LEAF** node. Applies for **MUSIC** and **VIDEO** only.	String	No
genre	The genre of the media. This field is returned only for the **LEAF** node. Applies for **MUSIC** and **VIDEO** only.	String	No
duration	The duration of the media. This field is returned only for the **LEAF** node. Applies for **MUSIC** and **VIDEO** only.	String	No
contentSize	The size of the media file in bytes. This field is returned only for the **LEAF** node.	String	No
resolution	The resolution of the media. This field is returned only for the **LEAF** node. Applies for **PICTURE** and **VIDEO** only.	String	No
mimeType	The mime type of the media file. This field is returned only for the **LEAF** node.	String	No
year	The year when the media was created or published. This field is returned only for the **LEAF** node.	String	No
thumbnailUrl	The thumbnail url of the media. This field is returned only for the **LEAF** node. This URL is valid as long as the specified RA session is valid.	String	No
url	The url of the media. This field is returned only for the **LEAF** node. This URL is valid as long as the specified RA session is valid.	String	No
count	The number of child nodes in this node. This field is returned only for the **BRANCH** node.	Integer	No

Example Response Body (on Success) (VIDEO_ALL) (JSON):

```
{
   "mediaFeed":{
      "paginationResult":{
         "startIndex":0,
         "itemsReturned":1,
         "totalItems":1
      },
      "title":"All Videos",
      "nodes":[
         {
            "node":{
               "nodeId":"0$3$28I1546",
               "nodeType":"LEAF",
               "mediaType":"VIDEO",
               "title":"Wildlife in HD",
               "album":"videos",
               "artist":"",
               "genre":"Unknown",
               "duration":"0:00:30.093",
               "contentSize":"26246026",
               "resolution":"1280x720",
               "mimeType":"video/x-ms-wmv",
               "year":"2009",

"thumbnailUrl":"http://64.101.110.133:62432/eurl/7f1b6a9c-1fb5-11e2-b10b-c0c1c06415e9/4VteYpgQBruf5Bjqujx
"url":"http://64.101.110.133:62432/eurl/7f1b6a9c-1fb5-11e2-b10b-c0c1c06415e9/4VteYpgQBruf5BjqujxIXKGGVABt
}
         }
      ]
   }
}
```

Example Response Body (on Success) (MUSIC_ALL) (JSON):

```
{
   "mediaFeed":{
      "paginationResult":{
         "startIndex":0,
         "itemsReturned":3,
         "totalItems":3
      },
      "title":"All Tracks",
      "nodes":[
         {
            "node":{
               "nodeId":"0$1$8I1290",
               "nodeType":"LEAF",
               "mediaType":"MUSIC",
               "title":"Kalimba",
               "album":"Ninja Tuna",
               "artist":"Mr. Scruff",
               "genre":"Electronic",
               "duration":"0:05:48",
               "contentSize":"8414449",
               "mimeType":"audio/mpeg",
               "year":"2008",

"thumbnailUrl":"http://64.101.110.133:62432/eurl/7f1b6a9c-1fb5-11e2-b10b-c0c1c06415e9/4VteYpgQBruf5Bjqujx
"url":"http://64.101.110.133:62432/eurl/7f1b6a9c-1fb5-11e2-b10b-c0c1c06415e9/4VteYpgQBruf5BjqujxIXKGGVABt
}
         },
         {
            "node":{
               "nodeId":"0$1$8I1802",
               "nodeType":"LEAF",
               "mediaType":"MUSIC",
               "title":"Maid with the Flaxen Hair",
               "album":"Fine Music, Vol. 1",
               "artist":"Richard Stoltzman/Slovak Radio Symphony Orchestra",
               "genre":"Classical",
               "duration":"0:02:50",
               "contentSize":"4113874",
               "mimeType":"audio/mpeg",
               "year":"2008",

"thumbnailUrl":"http://64.101.110.133:62432/eurl/7f1b6a9c-1fb5-11e2-b10b-c0c1c06415e9/4VteYpgQBruf5Bjqujx
"url":"http://64.101.110.133:62432/eurl/7f1b6a9c-1fb5-11e2-b10b-c0c1c06415e9/4VteYpgQBruf5BjqujxIXKGGVABt
}
         },
         {
            "node":{
               "nodeId":"0$1$8I1034",
               "nodeType":"LEAF",
               "mediaType":"MUSIC",
               "title":"Sleep Away",
               "album":"Bob Acri",
               "artist":"Bob Acri",
               "genre":"Jazz",
               "duration":"0:03:21",
               "contentSize":"4842585",
               "mimeType":"audio/mpeg",
               "year":"2004",

"thumbnailUrl":"http://64.101.110.133:62432/eurl/7f1b6a9c-1fb5-11e2-b10b-c0c1c06415e9/4VteYpgQBruf5Bjqujx
"url":"http://64.101.110.133:62432/eurl/7f1b6a9c-1fb5-11e2-b10b-c0c1c06415e9/4VteYpgQBruf5BjqujxIXKGGVABt
}
         }
      ]
   }
}
```

Example Response Body (on Success) (PICTURE_ALL) (JSON):

```
{
   "mediaFeed":{
      "paginationResult":{
         "startIndex":0,
         "itemsReturned":2,
         "totalItems":2
      },
      "title":"All Photos",
      "nodes":[
         {
            "node":{
               "nodeId":"0$2$20I522",
               "nodeType":"LEAF",
               "mediaType":"PICTURE",
               "title":"photo",
               "album":"mypics",
               "contentSize":"2180207",
               "resolution":"2592x1936",
               "mimeType":"image/jpeg",
               "year":"2012",

"thumbnailUrl":"http://64.101.110.133:62432/eurl/7f1b6a9c-1fb5-11e2-b10b-c0c1c06415e9/4VteYpgQBruf5Bjqujx
"url":"http://64.101.110.133:62432/eurl/7f1b6a9c-1fb5-11e2-b10b-c0c1c06415e9/4VteYpgQBruf5BjqujxIXKGGVABt
}
         },
         {
            "node":{
               "nodeId":"0$2$20I266",
               "nodeType":"LEAF",
               "mediaType":"PICTURE",
               "title":"photo1",
               "album":"mypics",
               "contentSize":"1923654",
               "resolution":"2592x1936",
               "mimeType":"image/jpeg",
               "year":"2012",

"thumbnailUrl":"http://64.101.110.133:62432/eurl/7f1b6a9c-1fb5-11e2-b10b-c0c1c06415e9/4VteYpgQBruf5Bjqujx
"url":"http://64.101.110.133:62432/eurl/7f1b6a9c-1fb5-11e2-b10b-c0c1c06415e9/4VteYpgQBruf5BjqujxIXKGGVABt
}
         }
      ]
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	The network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
404	SERVICE_NOT_SUPPORTED	The specified network (router) does not support the media service.
404	STORAGE_NOT_MOUNTED	The storage is not mounted on the specified network (router). In other words, a disk/drive is not plugged-in into router's USB port.
404	SERVICE_NOT_RUNNING	The media service is currently not running on the specified network (router).
401	INVALID_RA_SESSION	The specified RA session does not exist, has expired or has been closed.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"INVALID_RA_SESSION",
        "message":"The specified RA session does not exist, has expired or has been closed.",
        "parameters":[

        ]
      }
    }
  ]
}
```

RA (Remote Access) Session

Get my public IP address

Purpose

It returns the public IP address of the caller.

Request

GET /cloud/device-service/rest/mypublicipaddress

Here is the raw http request:

```
GET /cloud/device-service/rest/mypublicipaddress

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the public IP address of the caller, as a JSON document.

Example Response Body (on Success) (JSON):

```
{
   "ipAddress":{
      "ipv4":"64.101.110.61"
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"INVALID_CLIENT_TYPE",
        "message":"The client type ID is not valid.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Open an RA session to access the media service

Purpose

It opens an RA (Remote Access) session to access the media service (hosted on the specified router) remotely over the internet. An RA session is needed to access the media served by the media service. However, an RA session is not needed to start, trigger media scan or stop the media service. This operation can be performed only by ADMIN account(s) associated with the network.

Request

POST /cloud/device-service/rest/networks/<networkId>/services/mediaservice/rasessions

The <networkId> is a value that uniquely identifies a network (A network represents a router).

Here is the raw http request:

```
POST /cloud/device-service/rest/networks/networkId-goes-here/services/mediaservice/rasessions

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Content-Type: application/json; charset=UTF-8
Accept: application/json

xml-or-json-content-goes-here
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization**, **Content-Type** and **Accept** header fields are specified as shown above.

Request: Entity Body

The request body contains a representation of the RA session to be opened, as a JSON document.

Elements/fields to be included in the request body:

Name	Description	Data Type	Required?
sourceIPAddress	The public IP address of the client which will access the media served by the media service remotely over the internet. The router will allow streaming/downloading media only from this IP address. If the public IP address of the client changes then the client should close any existing RA session and open a new one.	String	Yes

Example Request Body (JSON):

```
{
  "raSession": {
    "sourceIPAddress": "173.36.196.7"
  }
}
```

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the newly opened RA session, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
raSessionId	The unique ID of the RA session. This is unique within a network.	String	Yes
destinationIPAddress	The public IP address where the media service is available over the internet.	String	Yes
destinationPort	The port number where the media service is available over the internet.	Integer	Yes
timeoutSecondsRemaining	Remaining session timeout seconds of this RA session.	String	Yes

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "raSession":{
      "raSessionId":"D46D6F02-1FD0-11E2-A158-C0C1C06415E9",
      "destinationIPAddress":"64.101.110.133",
      "destinationPort":40288,
      "timeoutSecondsRemaining":1200
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
404	SERVICE_NOT_SUPPORTED	The specified network (router) does not support the media service.
404	STORAGE_NOT_MOUNTED	The storage is not mounted on the specified network (router). In other words, a disk/drive is not plugged-in into router's USB port.
404	SERVICE_NOT_RUNNING	The media service is currently not running on the specified network (router).

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Get an RA session

Purpose

Gets (aka reads or retrieves) the specified RA (Remote Access) session. This operation can be performed only by ADMIN account(s) associated with the network.

Request

GET /cloud/device-service/rest/networks/<networkId>/rasessions/<raSessionId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <raSessionId> is a value that uniquely identifies an RA session within a network.

Here is the raw http request:

```
GET /cloud/device-service/rest/networks/networkId-goes-here/rasessions/raSessionId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the requested RA session, as a JSON document.

Elements/fields included in the response body:

Name	Description	Data Type	Required?
raSessionId	The unique ID of the RA session. This is unique within a network.	String	Yes
destinationIPAddress	The public IP address where the media service is available.	String	Yes
destinationPort	The port number where the media service is available.	Integer	Yes
timeoutSecondsRemaining	Remaining session timeout seconds of this RA session.	String	Yes

⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "raSession":{
      "raSessionId":"D46D6F02-1FD0-11E2-A158-C0C1C06415E9",
      "destinationIPAddress":"64.101.110.133",
      "destinationPort":40288,
      "timeoutSecondsRemaining":912
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
401	INVALID_RA_SESSION	The specified RA session does not exist, has expired or has been closed.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Extend an RA session

Purpose

Extends the timeout seconds for the specified RA session. This operation can be performed only by ADMIN account(s) associated with the network.

Request

PUT /cloud/device-service/rest/networks/<networkId>/rasessions/<raSessionId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <raSessionId> is a value that uniquely identifies an RA session within a network.

Here is the raw http request:

```
PUT /cloud/device-service/rest/networks/networkId-goes-here/rasessions/raSessionId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Authorization: Bearer access-token-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id**, **Authorization** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the extended RA session, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "raSession":{
      "raSessionId":"D46D6F02-1FD0-11E2-A158-C0C1C06415E9",
      "destinationIPAddress":"64.101.110.133",
      "destinationPort":40288,
      "timeoutSecondsRemaining":1200
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
401	INVALID_RA_SESSION	The specified RA session does not exist, has expired or has been closed.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Close an RA session

Purpose

Closes the specified RA (Remote Access) session.

Request

DELETE /cloud/device-service/rest/networks/<networkId>/rasessions/<raSessionId>

The <networkId> is a value that uniquely identifies a network (A network represents a router).
The <raSessionId> is a value that uniquely identifies an RA session within a network.

Here is the raw http request:

```
DELETE /cloud/device-service/rest/networks/networkId-goes-here/rasessions/raSessionId-goes-here

X-Cisco-HN-Client-Type-Id: client-type-id-goes-here
Accept: application/json
```

Make sure that the **X-Cisco-HN-Client-Type-Id** and **Accept** header fields are specified as shown above.

Request: Entity Body

The body of this request should be empty. If it is non-empty, the body will be ignored.

Response: Status Codes (on Success)

Returns 200 if the operation is performed successfully.

Response: Entity Body (on Success)

Returns a representation of the closed RA session, as a JSON document.

> ⚠ In future, additional fields may be included in the response. The client must be coded to ignore the fields it does not recognize.

Example Response Body (on Success) (JSON):

```
{
   "raSession":{
      "raSessionId":"D46D6F02-1FD0-11E2-A158-C0C1C06415E9"
   }
}
```

Response: Status Codes & Entity Body (on Error)

Returns a JSON document containing a description of the error. See Error Response - Cloud API section for details.

Status / Error codes returned:

HTTP Status Code	Error Code	Error Code Meaning
404	NETWORK_NOT_FOUND	A network with the specified network ID is not found.
504	NETWORK_UNREACHABLE	The specified network (router) is not reachable.
401	INVALID_RA_SESSION	The specified RA session does not exist, has expired or has been closed.

See Common Error Codes Returned - Cloud API.

Example Response Body (on Error):

```
{
  "errors":[
    {
      "error":{
        "code":"NETWORK_NOT_FOUND",
        "message":"A network with the specified network ID is not found.",
        "parameters":[

        ]
      }
    }
  ]
}
```

Linksys Smart WiFi Platform OAuth 2.0 Documentation

Overview

Linksys Smart WiFi Platform uses OAuth 2.0 open-standard protocol to allow a user to authorize an application (aka client) to access his/her home networks (routers) and/or data stored in the Linksys Smart WiFi, after he/she has been authenticated (logged-in).

A third-party application (aka client) often requires limited access to a user's home networks (routers) and/or data stored in the Linksys Smart WiFi. To ensure that access to user's home networks (routers) and/or data stored in the Linksys Smart WiFi are not abused, all requests for access must be approved by the user.

OAuth 2.0 is a relatively simple protocol and a developer can integrate with Linksys Smart WiFi Platform's OAuth 2.0 endpoints without too much effort. In a nutshell, a developer registers his/her application (aka client) with Linksys Smart WiFi, redirects a browser to a URL, parses a token from the response, and uses the token to invoke a Linksys Smart WiFi API.

Linksys Smart WiFi Platform supports the OAuth 2.0 protocol with bearer tokens for web and native (aka installed) applications.

Applications (aka Clients) follow the same basic steps when accessing a Linksys Smart WiFi API using OAuth 2.0. At a high level, using OAuth 2.0 to access a Linksys Smart WiFi API consists of the following four steps:

1 Register Application (aka Client)

All applications (aka clients) that access a Linksys Smart WiFi API must be registered. The result of this registration process is a set of values (e.g. client_id, client_secret, redirect_uri, list of APIs to be accessed, etc.) that are known to both Linksys Smart WiFi and the application. The set of values needed varies based on the type of application being built. For example a JavaScript application does not require a secret, but a server-side web application or a native application does.

2 Obtain an Access Token from the Linksys Smart WiFi authorization service

Before an application (aka client) can access Linksys Smart WiFi APIs, it must obtain an access token that grants access to the APIs.

There are several ways to make this request, and they vary based on the type of application.

The request requires the user to login to Linksys Smart WiFi. After logging in, the user will see the permissions (list of APIs to be accessed) requested by the application and is asked if he/she is willing to grant the application those permissions. This process is called "user consent".

If the user grants permission to the application, the application will be sent an access token or an authorization code (which is used to obtain an access token). If the user does not grant permission to the application, the Linksys Smart WiFi Authorization Service returns an error.

3 Send Access Token to an API

After an application (aka client) has obtained an access token, it may send the access token in a request to a Linksys Smart WiFi API. The

Access token is sent to a Linksys Smart WiFi API in the HTTP Authorization header as shown below.

```
Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA
```

4 Refresh the Access Token (optional)

Access tokens have a limited lifetime and, in some cases, an application (aka client) needs access to Linksys Smart WiFi APIs beyond the lifetime of a single access token. When this is the case, the application can obtain what is called a refresh token. A refresh token allows the application to obtain new access tokens.

Authorization Flows

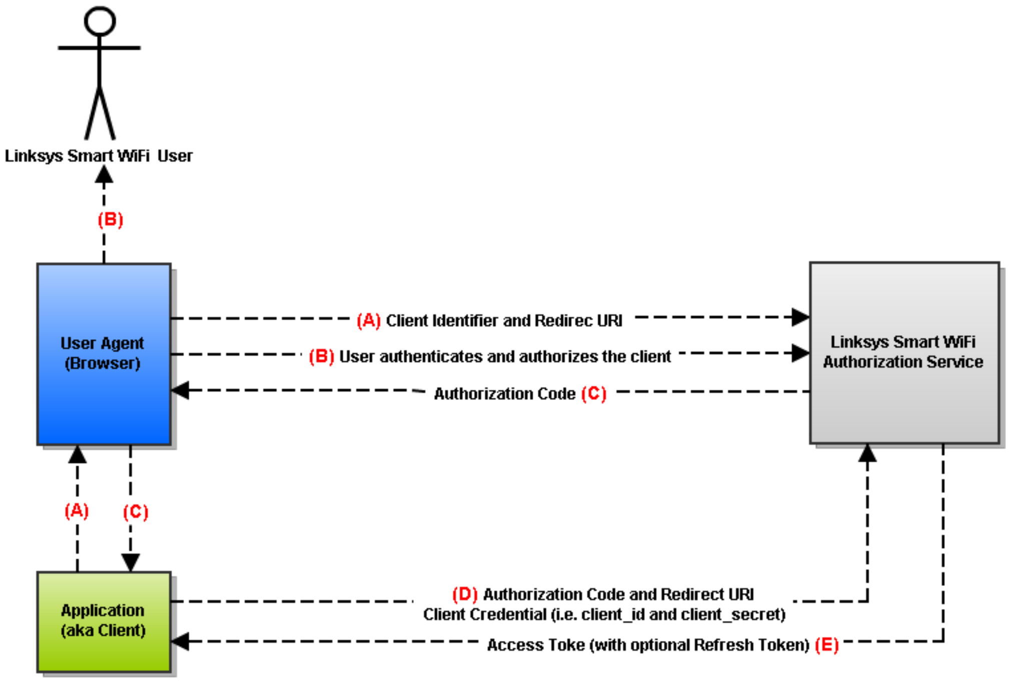
Linksys Smart WiFi Platform supports the following OAuth 2.0 authorization flows:

- 1 Authorization Code Grant Flow
  Server-Side Web Applications and **Native Applications** (i.e. iPhone / iPad apps, Android apps) should use this flow.
- 2 Implicit Grant Flow
  Browser-based **Client-Side Web Applications** (using JavaScript, Flash, etc.) should use this flow.

1 Authorization Code Grant Flow

1.0 Overview

Server-Side Web Applications and **Native Applications** (i.e. iPhone / iPad apps, Android apps) should use this flow.


Linksys Smart WiFi User
(B)
User Agent (Browser)
(A) Client Identifier and Redirec URI
(B) User authenticates and authorizes the client
Authorization Code (C)
Linksys Smart WiFi Authorization Service
(A)
(C)
Application (aka Client)
(D) Authorization Code and Redirect URI
Client Credential (i.e. client_id and client_secret)
Access Toke (with optional Refresh Token) (E)

You need to upgrade your Gliffy Plugin License. Your license entitles you to 500 users but you currently have a Confluence license for 2000 users. Please upgrade your license promptly.

Step (A): The application (aka client) initiates this flow by directing user's browser to Linksys Smart WiFi's **authorization endpoint** URL. The application (aka client) includes its client identifier, local state, and a redirection URI to which the Linksys Smart WiFi authorization service will send the browser back once access is granted (or denied).

⚠ In order for an application (aka client) to be approved, it must implement the OAuth workflow by launching the device’s **default browser, external to the application**. (The use of an embedded browser is not acceptable.)

Step (B): The Linksys Smart WiFi authorization service authenticates the user (via the browser). After authentication (logging in), the user will see the permissions (list of APIs to be accessed) requested by the application and is asked if he/she is willing to grant the application those permissions.

Step (C): Assuming the user grants access, the Linksys Smart WiFi authorization service redirects the browser back to the application (aka client) using the redirection URI provided earlier. The redirection URI includes an authorization code and any local state provided by the application (aka client) earlier.

Step (D): The application (aka client) requests an access token from the Linksys Smart WiFi's **token endpoint** by including the authorization code received in the previous step. When making the request, the application (aka client) authenticates with the Linksys Smart WiFi authorization service by including its credential (i.e. client_id and client_secret). The application (aka client) also includes the redirection URI used to obtain the authorization code for verification.

Step (E): The Linksys Smart WiFi authorization service authenticates the application (aka client), validates the authorization code, and ensures the redirection URI received matches the URI used to redirect the application (aka client) in step (C). If valid, the Linksys Smart WiFi authorization service responds back with an access token and optionally, a refresh token. If a refresh token is present in the response, then the application (aka client) may use it to obtain new access tokens at any time.

The application (aka client) can invoke/access a Linksys Smart WiFi API after it receives the access token.

1.1 Direct the user's browser to Linksys Smart WiFi's authorization endpoint URL

The application (aka client) initiates this flow by directing user's browser to Linksys Smart WiFi's **authorization endpoint** URL with a set of query string parameters.

⚠ In order for an application (aka client) to be approved, it must implement the OAuth workflow by launching the device’s **default browser, external to the application**. (The use of an embedded browser is not acceptable.)

The set of query string parameters supported by the Linksys Smart WiFi authorization service for Authorization Code Grant Flow are:

Parameter Name	Parameter Value	Description	Required?
response_type	**code**	For Authorization Code Grant Flow, the value of this parameter must be **code**.	Yes
client_id	The client identifier issued to the application (aka client) during the application / client registration process.	Indicates the application (aka client) that is making the request.	Yes
redirect_uri	One of the redirect_uri values registered during the application / client registration process.	Determines where the response is sent. The value of this parameter must exactly match one of the values registered during the application / client registration process (including the http or https schemes, case, and trailing '/').	Yes
state	any string	An opaque value used by the application (aka client) to maintain state between the request and callback. The Linksys Smart WiFi authorization service includes this value when redirecting the browser back to the application (aka client). The parameter should be used for preventing cross-site request forgery as described in OAuth Section 10.12	Yes
access_type	**online** or **offline**	Indicates if the application (aka client) needs to access a Linksys Smart WiFi API when the user is not present at the browser. This parameter defaults to **online**. If an application (aka client) needs to refresh access tokens when the user is not present at the browser, then use **offline**. This will result in client obtaining a refresh token the first time the client exchanges an authorization code for a user.	No
approval_prompt	**force** or **auto**	Indicates if the user should be re-prompted for consent. The default is **auto**, so a given user should only see the consent page the first time through the sequence. If the value is **force**, then the user sees a consent page even if they have previously given consent to the client.	No

An example URL is shown below.

Production Environment:

```
https://cloud.ciscoconnectcloud.com/cloud/authorization-service/oauth/authorize?
response_type=code
    &client_id=YOUR_CLIENT_ID
    &redirect_uri=YOUR_REDIRECT_URI
    &state=SOME_ARBITRARY_BUT_UNIQUE_STRING
```

For security, the value of **redirect_uri** parameter must exactly match one of the values registered during the application (aka client) registration process (including the http or https schemes, case, and trailing '/').

The **state** parameter should be set to some arbitrary string you generate uniquely for each auth request. This value will be passed back as a parameter to the **redirect_uri** once the user has authorized the application (aka client) and the application should check that the returned value matched the value it passed in at the start of the flow. This guards against Cross-site Request Forgery by ensuring the incoming redirect is part of the auth flow initiated by the application (aka client).

1.2 The user is prompted to authorize the application (aka client)

If the user is not logged-in into Linksys Smart WiFi, He/She will be prompted (via the browser) to log-in.

If the user has not already authorized the application, He/She will be prompted (via the browser) to authorize the application. This process is called "user consent".

1.3 The user's browser is redirected back to the application (aka client)

If the user grants the access request, the Linksys Smart WiFi authorization service issues an authorization code and delivers it to the application (aka client) by adding the following parameters to the query component of the redirection URI using the "application/x-www-form-urlencoded" format:

Parameter Name	Description / Parameter Value	Required?
code	The authorization code generated / issued by the Linksys Smart WiFi authorization service. The authorization code expires shortly (max lifetime of 10 minutes) after it is issued to mitigate the risk of leaks. The application (aka client) must not use the authorization code more than once.	Yes
state	The exact value received from the application (aka client).	Yes if the **state** parameter was included in the authorization request

For example, the Linksys Smart WiFi authorization service redirects the browser by sending the following HTTP response:

```
HTTP/1.1 302 Found
Location: YOUR_REDIRECT_URI?code=AUTHORIZATION_CODE_GENERATED_BY_CISCO_CONNECT&state=YOUR_STATE_VALUE
```

⚠ In future, additional parameters/fields may be included in the response. The client must be coded to ignore the parameters/fields it does not recognize.

If the request fails due to a missing, invalid, or mismatching redirection URI, or if the client identifier is missing or invalid, the Linksys Smart WiFi authorization service will inform the user of the error, and will not automatically redirect the browser to the invalid redirection URI.

If the user denies the access request or if the request fails for reasons other than a missing or invalid redirection URI, the Linksys Smart WiFi authorization service informs the application (aka client) by adding the following parameters to the query component of the redirection URI using the "application/x-www-form-urlencoded" format:

Parameter Name	Description / Parameter Value	Required?

error	A single error code from the following: **invalid_request** The request is missing a required parameter, includes an invalid parameter value, or is otherwise malformed. **unauthorized_client** The application (aka client) is not authorized to request an authorization code using this method. **access_denied** The user or Linksys Smart WiFi authorization service denied the request. **unsupported_response_type** The Linksys Smart WiFi authorization service does not support the specified response type. **server_error** The Linksys Smart WiFi authorization service encountered an unexpected condition which prevented it from fulfilling the request. **temporarily_unavailable** The Linksys Smart WiFi authorization service is currently unable to handle the request due to a temporary overloading or maintenance of the server.	Yes
error_description	A human-readable UTF-8 encoded text providing additional information, used to assist the application (aka client) developer in understanding the error that occurred.	No
state	The exact value received from the application (aka client).	Yes if a **state** parameter was included in the authorization request.

For example, the Linksys Smart WiFi authorization service redirects the browser by sending the following HTTP response:

```
HTTP/1.1 302 Found
   Location: YOUR_REDIRECT_URI?error=ERROR_CODE_RETURNED_BY_CISCO_CONNECT&state=YOUR_STATE_VALUE
```

1.4 Exchange the authorization code for an access token

After the application (aka client) receives the authorization code, it should make an HTTPs POST request to Linksys Smart WiFi's **token endpoint** to exchange the authorization code for an access token and a refresh token.

The application (aka client) makes a request to Linksys Smart WiFi's **token endpoint** by adding the following parameters using the "application/x-www-form-urlencoded" format in the HTTP request entity-body:

Parameter Name	Description / Parameter Value	Required?
grant_type	As defined in the OAuth 2.0 specification, the value of this parameter must be set to **authorization_code**.	Yes
code	The authorization code generated / issued by the Linksys Smart WiFi authorization service.	Yes
redirect_uri	The **redirect_uri** parameter that was included in the initial request (for authorization code), and their values must be identical.	Yes

The application (aka client) must specify its credentials (i.e. client id/client secret issued during application/client registration) using HTTP Basic authentication scheme as defined in RFC2617. The client identifier is used as the username, and the client secret is used as the password.

An example request:

Production Environment:

```
POST /cloud/authorization-service/oauth/token HTTP/1.1
Host: cloud.ciscoconnectcloud.com
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Accept: application/json
Accept-Charset: utf-8

grant_type=authorization_code&
code=AUTHORIZATION_CODE_GENERATED_BY_CISCO_CONNECT&
redirect_uri=YOUR_REDIRECT_URI
```

If the access token request is valid and authorized, the Linksys Smart WiFi authorization service issues an access token and optional refresh token, and constructs the response by adding the following parameters to the entity body of the HTTP response with a 200 (OK) status code:

Parameter Name	Description / Parameter Value	Required?
access_token	The access token generated / issued by the Linksys Smart WiFi authorization service.	Yes
token_type	Indicates the type of token returned. At this time, this field will always have the value **Bearer**. The Value is case insensitive.	Yes
expires_in	The lifetime in seconds of the access token. For example, the value "3600" denotes that the access token will expire in one hour from the time the response was generated.	Yes
refresh_token	A refresh token which can be used to obtain a new access token. Refresh tokens are valid until the user revokes access. This field is only present if **access_type=offline** is included in the initial request (for authorization code)	No

The parameters are included in the entity body of the HTTP response using the "application/json" media type as defined by RFC4627. The parameters are serialized into a JSON structure by adding each parameter at the highest structure level. Parameter names and string values are included as JSON strings. Numerical values are included as JSON numbers. The order of parameters does not matter and can vary.

The Linksys Smart WiFi authorization service includes the HTTP "Cache-Control" response header field RFC2616 with a value of "no-store" as well as the "Pragma" response header field RFC2616 with a value of "no-cache".

An example successful response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
    "access_token":"ACESS_TOKEN_GENERATED_BY_CISCO_CONNECT",
    "token_type":"Bearer",
    "expires_in":NUMBER_OF_SECONDS_UNTIL_ACCESS_TOKEN_EXPIRES,
    "refresh_token":"REFRESH_TOKEN_GENERATED_BY_CISCO_CONNECT"
}
```

⚠ In future, additional parameters/fields may be included in the response. The client must be coded to ignore the parameters/fields it does not recognize.

If the request is invalid or application (aka client) authentication failed, the Linksys Smart WiFi authorization service responds with an HTTP 400 (Bad Request) status code and includes the following parameters with the response:

Parameter Name	Description / Parameter Value	Required?
error	A single error code from the following: **invalid_request** The request is missing a required parameter, includes an unsupported parameter value (other than grant type), repeats a parameter, includes multiple credentials, utilizes more than one mechanism for authenticating the client, or is otherwise malformed. **invalid_client** Client authentication failed (e.g. unknown client, no client authentication included, or unsupported authentication method). **invalid_grant** The provided authorization grant (e.g. authorization code, resource owner credentials) or refresh token is invalid, expired, revoked, does not match the redirection URI used in the authorization request, or was issued to another client. This error code is also returned if the account is disabled. **unauthorized_client** The authenticated application (aka client) is not authorized to use the specified grant type. **unsupported_grant_type** The specified grant type is not supported by the Linksys Smart WiFi authorization service. **server_error** The Linksys Smart WiFi authorization service encountered an unexpected condition which prevented it from fulfilling the request. **temporarily_unavailable** The Linksys Smart WiFi authorization service is currently unable to handle the request due to a temporary overloading or maintenance of the server.	Yes

error_description	A human-readable UTF-8 encoded text providing additional information, used to assist the application (aka client) developer in understanding the error that occurred.	No

An example error response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
"error":"ERROR_CODE_RETURNED_BY_CISCO_CONNECT"
}
```

1.5 Calling a Linksys Smart WiFi API

After the application (aka client) has obtained an access token, the application can access a Linksys Smart WiFi API by including it in an **Authorization: Bearer** HTTP header as shown below:

```
Authorization: Bearer ACCESS_TOKEN_GOES_HERE
```

1.6 Using a Refresh Token

If the Linksys Smart WiFi authorization service issued a refresh token to the application (aka client), the application (aka client) can use the refresh token to obtain a new access token any time until the the refresh token is revoked.

The application (aka client) makes an HTTPs POST request to Linksys Smart WiFi's **token endpoint** by adding the following parameters using the "application/x-www-form-urlencoded" format in the HTTP request entity-body:

Parameter Name	Description / Parameter Value	Required?
grant_type	As defined in the OAuth 2.0 specification, the value of this parameter must be set to **refresh_token**.	Yes
refresh_token	The refresh token issued to the application (aka client).	Yes

The application (aka client) must specify its credentials (i.e. client id/client secret issued during client registration) using HTTP Basic authentication scheme as defined in RFC2617. The client identifier is used as the username, and the client secret is used as the password.

An example request:

Production Environment:

```
POST /cloud/authorization-service/oauth/token HTTP/1.1
Host: cloud.ciscoconnectcloud.com
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Accept: application/json
Accept-Charset: utf-8

grant_type=refresh_token&
refresh_token=REFRESH_TOKEN_GENERATED_BY_CISCO_CONNECT
```

If the request is valid and authorized, the Linksys Smart WiFi authorization service issues an access token, and constructs the response by adding the following parameters to the entity body of the HTTP response with a 200 (OK) status code:

Parameter Name	Description / Parameter Value	Required?
access_token	The access token generated / issued by the Linksys Smart WiFi authorization service.	Yes
token_type	Indicates the type of token returned. At this time, this field will always have the value **Bearer**. The Value is case insensitive.	Yes
expires_in	The lifetime in seconds of the access token. For example, the value "3600" denotes that the access token will expire in one hour from the time the response was generated.	Yes

The parameters are included in the entity body of the HTTP response using the "application/json" media type as defined by RFC4627. The parameters are serialized into a JSON structure by adding each parameter at the highest structure level. Parameter names and string values are included as JSON strings. Numerical values are included as JSON numbers. The order of parameters does not matter and can vary.

The Linksys Smart WiFi authorization service includes the HTTP "Cache-Control" response header field RFC2616 with a value of "no-store" as well as the "Pragma" response header field RFC2616 with a value of "no-cache".

An example successful response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
    "access_token":"ACESS_TOKEN_GENERATED_BY_CISCO_CONNECT",
    "token_type":"Bearer",
    "expires_in":NUMBER_OF_SECONDS_UNTIL_ACCESS_TOKEN_EXPIRES
}
```

⚠ In future, additional parameters/fields may be included in the response. The client must be coded to ignore the parameters/fields it does not recognize.

If the request is invalid or application (aka client) authentication failed, the Linksys Smart WiFi authorization service responds with an HTTP 400 (Bad Request) status code and includes the following parameters with the response:

Parameter Name	Description / Parameter Value	Required?
error	A single error code from the following: **invalid_request** The request is missing a required parameter, includes an unsupported parameter value (other than grant type), repeats a parameter, includes multiple credentials, utilizes more than one mechanism for authenticating the application (aka client), or is otherwise malformed. **invalid_client** Application (aka client) authentication failed (e.g. unknown client, no client authentication included, or unsupported authentication method). **invalid_grant** The provided authorization grant (e.g. authorization code, resource owner credentials) or refresh token is invalid, expired, revoked, does not match the redirection URI used in the authorization request, or was issued to another client. This error code is also returned if the account is disabled. **unauthorized_client** The authenticated application (aka client) is not authorized to use the specified grant type. **unsupported_grant_type** The specified grant type is not supported by the Linksys Smart WiFi authorization service. **server_error** The Linksys Smart WiFi authorization service encountered an unexpected condition which prevented it from fulfilling the request. **temporarily_unavailable** The Linksys Smart WiFi authorization service is currently unable to handle the request due to a temporary overloading or maintenance of the server.	Yes
error_description	A human-readable UTF-8 encoded text providing additional information, used to assist the application (aka client) developer in understanding the error that occurred.	No

An example error response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
"error":"ERROR_CODE_RETURNED_BY_CISCO_CONNECT"
}
```

1.7 Revoking an Access Token or a Refresh Token

An application (aka client) uses the Linksys Smart WiFi's **revocation endpoint** to revoke an access token or a refresh token (i.e. refresh token and all related access tokens).

Developers may use this feature when configuring a "Log Out" button in their application.

The application (aka client) makes a request to the Linksys Smart WiFi's **revocation endpoint** by adding "one" of following parameters using the "application/x-www-form-urlencoded" format in the HTTP request entity-body:

Parameter Name	Description / Parameter Value	Required?
access_token	The access token to be revoked.	One of the **access_token** or **refresh_token** is required.
refresh_token	The refresh token to be revoked. In this case, all related access tokens are revoked as well.	One of the **access_token** or **refresh_token** is required.

An example request for revoking an access token:

Production Environment:

```
POST /cloud/authorization-service/oauth/revoke HTTP/1.1
Host: cloud.ciscoconnectcloud.com
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Accept: application/json
Accept-Charset: utf-8

access_token=ACCESS_TOKEN_GENERATED_BY_CISCO_CONNECT
```

An example request for revoking a refresh token:

Production Environment:

```
POST /cloud/authorization-service/oauth/revoke HTTP/1.1
Host: cloud.ciscoconnectcloud.com
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Accept: application/json
Accept-Charset: utf-8

refresh_token=REFRESH_TOKEN_GENERATED_BY_CISCO_CONNECT
```

The Linksys Smart WiFi authorization service indicates successful processing of the request by returning an HTTP status code 200 with the access/refresh token that has been revoked successfully.

An example response if the specified access token has been revoked successfully:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
    "access_token":"ACCESS_TOKEN_GENERATED_BY_CISCO_CONNECT"
}
```

An example response if the specified refresh token has been revoked successfully:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
    "refresh_token":"REFRESH_TOKEN_GENERATED_BY_CISCO_CONNECT"
}
```

In future, additional parameters/fields may be included in the response. The client must be coded to ignore the parameters/fields it does not recognize.

For all error conditions, a status code 400 is used along with one of the following error responses.

Parameter Name	Description / Parameter Value	Required?
error	A single error code from the following: **invalid_request** The request is missing a required parameter, includes an unsupported parameter value, repeats a parameter, includes multiple tokens, or is otherwise malformed. **server_error** The Linksys Smart WiFi authorization service encountered an unexpected condition which prevented it from fulfilling the request. **temporarily_unavailable** The Linksys Smart WiFi authorization service is currently unable to handle the request due to a temporary overloading or maintenance of the server.	Yes
error_description	A human-readable UTF-8 encoded text providing additional information, used to assist the application (aka client) developer in understanding the error that occurred.	No

An example error response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
"error":"ERROR_CODE_RETURNED_BY_CISCO_CONNECT"
}
```

2 Implicit Grant Flow

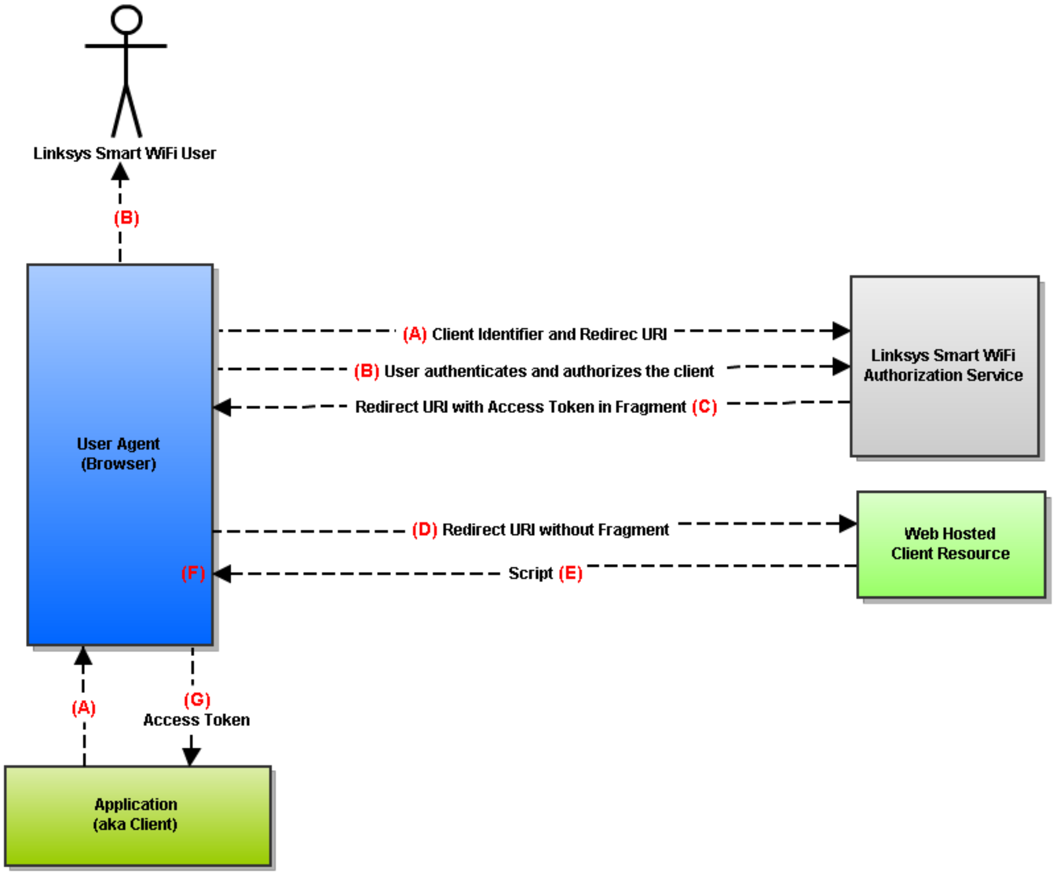
2.0 Overview

Browser-based **Client-Side Web Applications** (using JavaScript, Flash, etc.) should use this flow.

The Implicit Grant Flow is used to obtain access tokens (it does not support the issuance of refresh tokens) and is optimized for applications (aka clients) which are typically implemented in a browser using a scripting language such as JavaScript.

Unlike the Authorization Code Grant Flow in which the application (aka client) makes separate requests for authorization code and access token, the client receives the access token as the result of the authorization request.

The Implicit Grant Flow does not include application (aka client) authentication, and relies on the presence of the Linksys Smart WiFi user and the registration of the redirection URI.


Linksys Smart WiFi User
(B)
User Agent (Browser)
(A) Client Identifier and Redirec URI
(B) User authenticates and authorizes the client
Redirect URI with Access Token in Fragment (C)
Linksys Smart WiFi Authorization Service
(D) Redirect URI without Fragment
Script (E)
(F)
Web Hosted Client Resource
(A)
(G) Access Token
Application (aka Client)

You need to upgrade your Gliffy Plugin License. Your license entitles you to 500 users but you currently have a Confluence license for 2000 users. Please upgrade your license promptly.

Step (A): The application (aka client) initiates this flow by directing user's browser to Linksys Smart WiFi's **authorization endpoint** URL. The application (aka client) includes its client identifier, local state, and a redirection URI to which the Linksys Smart WiFi authorization service will send the browser back once access is granted (or denied).

⚠ In order for an application (aka client) to be approved, it must implement the OAuth workflow by launching the device's **default browser, external to the application**. (The use of an embedded browser is not acceptable.)

Step (B): The Linksys Smart WiFi authorization service authenticates the user (via the browser). After authentication (logging in), the user will see the permissions (list of APIs to be accessed) requested by the application and is asked if he/she is willing to grant the application those permissions.

Step (C): Assuming the user grants access, the Linksys Smart WiFi authorization service redirects the browser back to the application (aka client) using the redirection URI provided earlier. The redirection URI includes an access token and any local state provided by the application (aka client) earlier in the URI fragment.

Step (D): The browser follows the redirection instructions by making a request to the web-hosted client resource (which does not include the fragment per RFC2616). The browser retains the fragment information locally.

Step (E): The web-hosted client resource returns a web page (typically an HTML document with an embedded script) capable of accessing the full redirection URI including the fragment retained by the browser, and extracting the access token (and other parameters) contained in the fragment.

Step (F): The browser executes the script provided by the web-hosted client resource locally, which extracts the access token and passes it to the application (aka client).

The application (aka client) can invoke/access a Linksys Smart WiFi API after it receives the access token.

2.1 Direct the user's browser to Linksys Smart WiFi's authorization endpoint URL

The application (aka client) initiates this flow by directing user's browser to Linksys Smart WiFi's **authorization endpoint** URL with a set of query string parameters.

> ⚠ In order for an application (aka client) to be approved, it must implement the OAuth workflow by launching the device's **default browser, external to the application**. (The use of an embedded browser is not acceptable.)

The set of query string parameters supported by the Linksys Smart WiFi authorization service for Implicit Grant Flow are:

Parameter Name	Parameter Value	Description	Required?
response_type	**token**	For Implicit Grant Flow, the value of this parameter must be **token**.	Yes
client_id	The client identifier issued to the application (aka client) during the application / client registration process.	Indicates the application (aka client) that is making the request.	Yes
redirect_uri	One of the redirect_uri values registered during the application / client registration process.	Determines where the response is sent. The value of this parameter must exactly match one of the values registered during the application / client registration process (including the http or https schemes, case, and trailing '/').	Yes
state	any string	An opaque value used by the application (aka client) to maintain state between the request and callback. The Linksys Smart WiFi authorization service includes this value when redirecting the browser back to the application (aka client). The parameter should be used for preventing cross-site request forgery as described in OAuth Section 10.12	Yes
approval_prompt	**force** or **auto**	Indicates if the user should be re-prompted for consent. The default is **auto**, so a given user should only see the consent page the first time through the sequence. If the value is **force**, then the user sees a consent page even if they have previously given consent to the client.	No

An example URL is shown below.

Production Environment:

```
https://cloud.ciscoconnectcloud.com/cloud/authorization-service/oauth/authorize?
response_type=token
    &client_id=YOUR_CLIENT_ID
    &redirect_uri=YOUR_REDIRECT_URI
    &state=SOME_ARBITRARY_BUT_UNIQUE_STRING
```

For security, the value of **redirect_uri** parameter must exactly match one of the values registered during the application (aka client) registration process (including the http or https schemes, case, and trailing '/').

The **state** parameter should be set to some arbitrary string you generate uniquely for each auth request. This value will be passed back as a parameter to the **redirect_uri** once the user has authorized the application (aka client) and the application should check that the returned value matched the value it passed in at the start of the flow. This guards against Cross-site Request Forgery by ensuring the incoming redirect is part of the auth flow initiated by the application (aka client).

2.2 The user is prompted to authorize the application (aka client)

If the user is not logged-in into Linksys Smart WiFi, He/She will be prompted (via the browser) to log-in.

If the user has not already authorized the application, He/She will be prompted (via the browser) to authorize the application. This process is called "user consent".

2.3 The user's browser is redirected back to the application (aka client)

If the user grants the access request, the Linksys Smart WiFi authorization service issues an access token and delivers it to the application (aka client) by adding the following parameters to the fragment component of the redirection URI using the "application/x-www-form-urlencoded" format:

Parameter Name	Description / Parameter Value	Required?
access_token	The access token generated / issued by the Linksys Smart WiFi authorization service.	Yes

token_type	Indicates the type of token returned. At this time, this field will always have the value **Bearer**. The Value is case insensitive.	Yes
expires_in	The lifetime in seconds of the access token. For example, the value "3600" denotes that the access token will expire in one hour from the time the response was generated.	Yes
state	The exact value received from the application (aka client).	Yes if the **state** parameter was included in the authorization request

Since a fragment is not returned by the browser to the application (aka client), client-side script must parse the fragment and extract the value of the access_token parameter.

For example, the Linksys Smart WiFi authorization service redirects the browser by sending the following HTTP response:

```
HTTP/1.1 302 Found
Location: YOUR_REDIRECT_URI#access_token=ACESS_TOKEN_GENERATED_BY_CISCO_CONNECT&token_type=Bearer
            &expires_in=NUMBER_OF_SECONDS_UNTIL_ACCESS_TOKEN_EXPIRES&state=YOUR_STATE_VALUE
```

⚠ In future, additional parameters/fields may be included in the response. The client must be coded to ignore the parameters/fields it does not recognize.

If the request fails due to a missing, invalid, or mismatching redirection URI, or if the client identifier is missing or invalid, the Linksys Smart WiFi authorization service will inform the user of the error, and will not automatically redirect the browser to the invalid redirection URI.

If the user denies the access request or if the request fails for reasons other than a missing or invalid redirection URI, the Linksys Smart WiFi authorization service informs the application (aka client) by adding the following parameters to the fragment component of the redirection URI using the "application/x-www-form-urlencoded" format:

Parameter Name	Description / Parameter Value	Required?
error	A single error code from the following: **invalid_request** The request is missing a required parameter, includes an invalid parameter value, or is otherwise malformed. **unauthorized_client** The application (aka client) is not authorized to request an access token using this method. **access_denied** The user or Linksys Smart WiFi authorization service denied the request. **unsupported_response_type** The Linksys Smart WiFi authorization service does not support the specified response type. **server_error** The Linksys Smart WiFi authorization service encountered an unexpected condition which prevented it from fulfilling the request. **temporarily_unavailable** The Linksys Smart WiFi authorization service is currently unable to handle the request due to a temporary overloading or maintenance of the server.	Yes
error_description	A human-readable UTF-8 encoded text providing additional information, used to assist the application (aka client) developer in understanding the error that occurred.	No
state	The exact value received from the application (aka client).	Yes if a **state** parameter was included in the authorization request.

For example, the Linksys Smart WiFi authorization service redirects the browser by sending the following HTTP response:

```
HTTP/1.1 302 Found
Location: YOUR_REDIRECT_URI#error=ERROR_CODE_RETURNED_BY_CISCO_CONNECT&state=YOUR_STATE_VALUE
```

2.4 Calling a Linksys Smart WiFi API

After the application (aka client) has obtained an access token, the application can access a Linksys Smart WiFi API by including it in an **Authorization: Bearer** HTTP header as shown below:

```
Authorization: Bearer ACCESS_TOKEN_GOES_HERE
```

2.5 Revoking an Access Token

An application (aka client) uses the Linksys Smart WiFi's **revocation endpoint** to revoke an access token.

Developers may use this feature when configuring a "Log Out" button in their application.

The application (aka client) makes a request to the Linksys Smart WiFi's **revocation endpoint** by adding the following parameter using the "application/x-www-form-urlencoded" format in the HTTP request entity-body:

Parameter Name	Description / Parameter Value	Required?
access_token	The access token to be revoked.	Yes

An example request for revoking an access token:

Production Environment:

```
POST /cloud/authorization-service/oauth/revoke HTTP/1.1
Host: cloud.ciscoconnectcloud.com
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
Accept: application/json
Accept-Charset: utf-8

access_token=ACCESS_TOKEN_GENERATED_BY_CISCO_CONNECT
```

The Linksys Smart WiFi authorization service indicates successful processing of the request by returning an HTTP status code 200 with the access token that has been revoked successfully.

An example response if the specified access token has been revoked successfully:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
    "access_token":"ACCESS_TOKEN_GENERATED_BY_CISCO_CONNECT"
}
```

⚠ In future, additional parameters/fields may be included in the response. The client must be coded to ignore the parameters/fields it does not recognize.

For all error conditions, a status code 400 is used along with one of the following error responses.

Parameter Name	Description / Parameter Value	Required?
error	A single error code from the following: **invalid_request** The request is missing a required parameter, includes an unsupported parameter value, repeats a parameter, includes multiple tokens, or is otherwise malformed. **server_error** The Linksys Smart WiFi authorization service encountered an unexpected condition which prevented it from fulfilling the request. **temporarily_unavailable** The Linksys Smart WiFi authorization service is currently unable to handle the request due to a temporary overloading or maintenance of the server.	Yes
error_description	A human-readable UTF-8 encoded text providing additional information, used to assist the application (aka client) developer in understanding the error that occurred.	No

An example error response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
"error":"ERROR_CODE_RETURNED_BY_CISCO_CONNECT"
}
```

Linksys Smart WiFi - JNAP Calls

Service Name	JNAP API Name	Description
Core Service		**This service exposes common device-level functionality and settings**
Core Service	GetDeviceInfo	This service provides access to basic device properties and settings. This action returns basic information about the device. Any device that exposes a JNAP server **must** implement this action.
DDNS Service		**This service provides access to a device's DDNS settings.**
DDNS Service	GetDDNSSettings	This action gets the device's current DDNS settings.
DDNS Service	GetDDNSStatus	This action gets the device's current DDNS status.
DDNS Service	SetDDNSSettings	This action sets the devices's DDNS settings
Device List Service	Device List Service	**This service provides information about the devices that are connected to the network, as well as those that are not currently online but have been connected to the network in the past.**
Device List Service	DeleteDevice	This action deletes a device from the device list. This is useful for "cleaning up" after a device is permanently removed from the network (e.g., thrown away, returned to the store). Only devices that are not currently connected to the network can be deleted. Note that if a device is deleted and subsequently rejoins the network, the device ID that is assigned to it when it is rediscovered will not necessarily be the same device ID that it had before it was deleted.
Device List Service	GetDevices	This action returns information about some or all of the devices that are connected to the network or have been connected to it at some time in the past. The caller can optionally specify the revision number that was returned from a previous call, in order to query for changes that have occurred since that call.
Device List Service		
Device List Service	SetDeviceProperties	This action creates or modifies custom device properties.
Diagnostic Service		**This service provides access to diagnostic tests**
Diagnostic Service	GetPingStatus	This action gets the status of the ping test.
Diagnostic Service	GetTracerouteStatus	This action gets the status of the traceroute test.
Diagnostic Service	StartPing	This service provides access to diagnostic tests. This action starts a ping test.
Diagnostic Service	StartTraceroute	This action starts a traceroute test.
Diagnostic Service	StopPing	This action stops the currently running ping test, if any.
Diagnostic Service	StopTraceroute	This action stops the currently running traceroute test, if any.
Guest Network Service		**This service provides access to guest network settings on a wireless access point.**
Guest Network Service	GetGuestNetworkSettings	This action gets the wireless access point's guest network settings.
Guest Network Service	SetGuestNetworkSettings	This action sets the wireless access point's guest network settings
Guest Network Authentication	Guest Network Authentication	**This service provides a mechanism to authenticate a device for WAN access via the guest network interface**
Guest Network Authentication	GuestNetworkAuthentication	This service provides a mechanism to authenticate a device for WAN access via the guest network interface
Locale Service	Locale Service	**This service provides access to locale-related device properties and settings**
Locale Service	GetTimeSettings	This action gets the device's time settings.
Locale Service	SetTimeSettings	This action sets the device's time settings.

Linksys Smart WiFi - JNAP Calls

Service Name	JNAP API Name	Description
MAC Filter Service		**This service provides access to MAC address filter settings on a router or other network infrastructure device.**
MAC Filter Service	GetMACFilterSettings	This action gets the MAC address filter settings.
MAC Filter Service	SetMACFilterSettings	This action sets the MAC address filter settings
Network Connection Service		**This service returns information about the network connections between client devices and the network infrastructure device.**
Network Connection Service	GetNetworkConnections	This action returns information about the connections between the network infrastructure device which implements this action and the devices which have a direct (single-hop) wired or wireless network connection to it.
Parental Control Service		**This service provides access to a router's parental controls settings.**
Parental Control Service	GetParentalControlSettings	This action gets the router's parental control settings
Parental Control Service	SetParentalControlSettings	This action sets the router's parental control settings.
QOS Service		**This service provides access to Quality of Service settings on a router or other network infrastructure device.**
QOS Service	GetLANQoSSettings	This action gets the QoS settings related to prioritizing wired LAN traffic.
QOS Service	GetQoSSettings	This action gets the QoS settings related to all traffic passing handled by this network infrastructure device.
QOS Service	GetWLANQoSSettings	This action gets the QoS settings related to prioritizing wireless LAN traffic
QOS Service	SetLANQoSSettings	This action sets the QoS settings related to prioritizing wired LAN traffic.
QOS Service	SetQoSSettings	This action sets the QoS settings
QOS Service	SetWLANQoSSettings	This action sets the QoS settings related to prioritizing wireless LAN traffic.
QOS Service	UpdateAutoAssignedRules	This action refreshes the QoS auto-assigned rule set. It will not overwrite any existing rules. Note: This method makes a call to a web-service API and may take some time to complete, depending on the WAN connection.
Router Service		**This service provides access to basic properties and settings of a router.**
Router Service	ConnectPPPWAN	This action causes the router to connect to PPP.
Router Service	DisconnectPPPWAN	This action causes the router to disconnect from PPP.
Router Service	GetDHCPSettings	This action gets router settings related to the DHCP server.
Router Service	GetEthernetPortConnections	This action gets information about the router's Ethernet port connections.
Router Service	GetIPv6Settings	This action gets router settings related to IPv6.
Router Service	GetLANSettings	This action gets router settings related to LAN management.

Linksys Smart WiFi - JNAP Calls		
Service Name	**JNAP API Name**	**Description**
Router Service	GetMACAddressCloneSettings	This action gets router settings related to the WAN MAC address.
Router Service	GetRoutingSettings	This action gets router settings related to routing.
Router Service	GetStaticRoutingTable	This action gets the static routing table.
Router Service	GetWANSettings	This action gets router settings related to the WAN connection.
Router Service	GetWANStatus	This action gets the current status of the router's WAN connection.
Router Service	SetIPv6Settings	This action sets router settings related to IPv6.
Router Service	SetLANSettings	This action sets router settings related to LAN management.
Router Service	SetMACAddressCloneSettings	This action sets router settings related to the WAN MAC address.
Router Service	SetRoutingSettings	This action sets router settings related to routing.
Router Service	SetWANSettings	This action sets router settings related to the WAN connection.
Router LED Service		**This service provides access to a router's LED settings.**
Router LED Service	GetRouterLEDSettings	This service provides access to a router's LED settings. This action gets the router's LED settings.
Router LED Service	SetRouterLEDSettings	This action sets the router's LED settings.
Router UPnP Services		**This service provides access to a router's UPnP settings.**
Router UPnP Services	GetUPnPSettings	This service provides access to a router's UPnP settings. This action gets the router's current UPnP settings.
Router UPnP Services	SetUPnPSettings	This action sets the router's UPnP settings.
Storage Service		**This service provides access to a device's mounted drives. This service provides access to a device's mounted drives.**
FTP Server		**This service provides access to a device's FTP folders.**
FTP Server	CreateFTPFolder	This action creates a new FTP folder on the device.
FTP Server	DeleteFTPFolder	This action deletes an FTP folder on the device.
FTP Server	EditFTPFolder	This action edits information about an FTP folder on the device.
FTP Server	GetFTPFolders	This action gets a list of the device's FTP folders.
FTP Server	GetFTPServerSettings	This action gets the device's FTP server settings.

Linksys Smart WiFi - JNAP Calls

Service Name	JNAP API Name	Description
FTP Server	SetFTPServerSettings	This action sets the device's FTP server settings.
SMB Server		**This service provides access to a device's SMB folders.**
SMB Server	CreateSMBFolder	This action creates a new SMB folder on the device.
SMB Server	DeleteSMBFolder	This action deletes a SMB folder on the device.
SMB Server	EditSMBFolder	This action edits information about a SMB folder on the device.
SMB Server	GetSMBFolders	This action gets a list of the device's SMB folders.
SMB Server	GetSMBServerSettings	This action gets the device's SMB server settings.
SMB Server	SetSMBServerSettings	This action sets the device's SMB server settings.
Storage		**This service provides access to a device's mounted partitions.**
Storage	CreateDirectory	This action creates a directory on a mounted partition.
Storage	CreateGroup	his action creates a new user group of the device.
Storage	CreateUser	This action creates a new user of the device.
Storage	DeleteGroup	This action deletes a group of the device.
Storage	DeleteUser	This action deletes a user of the device.
Storage	EditGroup	This action edits information about a group of the device.
Storage	EditUser	This action edits information about a user of the device.
Storage	GetGroups	This action gets a list of the device's groups.
Storage	GetMountedPartitions	This action gets information about the device's mounted partitions.
Storage	GetUsers	This action gets a list of the device's users.
Storage	ListSubdirectories	This action gets the list of subdirectories in a directory on a mounted partition.
Storage	UnmountPartition	This action unmounts a device's mounted partition.
UPnP Media Server		**This service provides access to a device's UPnP media folders**
UPnP Media Server	CreateUPnPMediaFolder	This action creates a new UPnP media folder on the device.

Linksys Smart WiFi - JNAP Calls		
Service Name	**JNAP API Name**	**Description**
UPnP Media Server	DeleteUPnPMediaFolder	This action deletes a UPnP media folder on the device.
UPnP Media Server	EditUPnPMediaFolder	This action edits information about a UPnP media folder on the device.
UPnP Media Server	GetUPnPMediaFolders	This action gets a list of the device's UPnP media folders.
UPnP Media Server	TriggerUPnPMediaFolderScan	This action triggers a scan of the UPnP media folders on the device.
Wireless AP Services		**This service provides access to properties and settings of the 802.11 wireless access point.**
WPS Server		**This service allows a client to start and stop WPS sessions on the wireless access point using a "soft" pushbutton method rather than a physical button on the AP.**
WPS Server	GetWPSServerSessionStatus	This action gets the status of the WPS session, if any, that is currently in progress on the wireless access point.
WPS Server	StartWPSServerSession	This action starts a WPS session on the wireless access point
WPS Server	StopWPSServerSession	This action stops the current WPS session on the wireless access point.
WPS Server 2		**This service extends the WPSServer service, allowing a client to provision the WPS server settings on the wireless access point.**
WPS Server 2	GetWPSServerSessionStatus	This action gets the status of the WPS session, if any, that is currently in progress on the wireless access point.
WPS Server 3	GetWPSServerSettings	This action gets the WPS server settings.
WPS Server 4	SetWPSServerSettings	This action sets the WPS server settings.
WPS Server 5	StartWPSServerSession	This action starts a WPS session on the wireless access point.
WPS Server 6	StopWPSServerSession	This action stops the current WPS session on the wireless access point.
Wireless AP		**This service provides access to properties and settings of the 802.11 wireless access point**
Wireless AP	GetAdvancedRadioInfo	This action gets advanced settings for all of the wireless access point's wireless radios.
Wireless AP	GetRadioInfo	This action gets information and settings for all of the wireless access point's wireless radios.
Wireless AP	SetAdvancedRadioSettings	This action sets advanced settings for one or more of the wireless access point's wireless radios.
Wireless AP	SetRadioSettings	This action sets information and settings for one or more of the wireless access point's wireless radios.

JNAP - Core Service – Developer

Core Service

Copyright Notice

Contents

Services

Core

`http://cisco.com/jnap/core/Core`

This service exposes common device-level functionality and settings.

Service Actions

- CheckAdminPassword
- FactoryReset
- GetAdminPasswordRestrictions
- GetDeviceInfo
- IsAdminPasswordDefault
- Reboot
- SetAdminPassword

Actions

CheckAdminPassword

`http://cisco.com/jnap/core/CheckAdminPassword`

This action checks the device's admin password.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/core/CheckAdminPasswordResult`

Value	Description
`OK`	Success.
`Error`	
`ErrorCannotAuthenticateWithSessionToken`	A session token was specified with the request. In order to check the admin password, no session token must be specified.
`_ErrorAbortedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	

`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/core/CheckAdminPassword"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

FactoryReset

`http://cisco.com/jnap/core/FactoryReset`

This action causes the device to reset its settings to their factory- default values and reboot the device.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/core/FactoryResetResult`

Value	Description
OK	Success.
Error	
ErrorDeviceBusy	The device is busy and cannot reboot now.
ErrorDisallowedRemoteCall	This is being called remotely, which is not allowed
_ErrorAbortedAction	
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/core/FactoryReset"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]
```

```
{
  "result": "OK"
}
```

GetAdminPasswordRestrictions

`http://cisco.com/jnap/core/GetAdminPasswordRestrictions`

This action gets the restrictions that the device imposes on its admin password.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
`minLength`	`int`	no	The minimum length of the admin password, in bytes.
`maxLength`	`int`	no	The maximum length of the admin password, in bytes.
`allowedCharacters`	`UnicodeRange[]`	no	A list of Unicode codepoint ranges, the union of which represents the complete set of characters that are allowed in the admin password. This array will always contain at least one item.

Result

`http://cisco.com/jnap/core/GetAdminPasswordRestrictionsResult`

Value	Description
`OK`	Success.

Error	
_ErrorAbortedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/core/GetAdminPasswordRestrictions"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "minLength": 8,
    "maxLength": 16,
    "allowedCharacters": [
      {
        "lowCodepoint": 32,
        "highCodepoint": 126
      }
    ]
  }
}
```

GetDeviceInfo

`http://cisco.com/jnap/core/GetDeviceInfo`

This service provides access to basic device properties and settings. This action returns basic information about the device. Any device that exposes a JNAP server **must** implement this action.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
`manufacturer`	`string`	no	The manufacturer of the device.
`modelNumber`	`string`	no	The model number of the device.
`hardwareVersion`	`string`	no	The hardware version of the device.
`description`	`string`	no	A brief description of the device.
`serialNumber`	`string`	no	The serial number of the device.
`firmwareVersion`	`string`	no	The version number of the device's firmware.
`firmwareDate`	`DateTime`	no	The date and time associated with the device's firmware.
`services`	`string[]`	no	The JNAP services exposed by the device.

Result

`http://cisco.com/jnap/core/GetDeviceInfoResult`

Value	Description
`OK`	Success.
`Error`	
`_ErrorAbortedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/core/GetDeviceInfo"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "manufacturer": "Linksys, Inc.",
    "modelNumber": "EX9000",
    "hardwareVersion": "1.2.0003",
    "description": "Linksys EX9000 Example Device",
    "serialNumber": "XXXX-XXXXXXX-12345",
    "firmwareVersion": "3.4.0005",
    "firmwareDate": "2011-07-12T00:00:00Z",
    "services": [
      "example string"
    ]
  }
}
```

IsAdminPasswordDefault

`http://cisco.com/jnap/core/IsAdminPasswordDefault`

This action returns whether the device's current admin password is the default value.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
`isAdminPasswordDefault`	`bool`	no	Whether the device's current admin password is the default value.

Result

`http://cisco.com/jnap/core/IsAdminPasswordDefaultResult`

Value	Description
`OK`	Success.

Error	
_ErrorAbortedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/core/IsAdminPasswordDefault"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "isAdminPasswordDefault": true
  }
}
```

Reboot

`http://cisco.com/jnap/core/Reboot`

This action causes the device to reboot.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/core/RebootResult`

Value	Description
`OK`	Success.
`Error`	
`ErrorDeviceBusy`	The device is busy and cannot reboot now.
`_ErrorAbortedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/core/Reboot"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

SetAdminPassword

`http://cisco.com/jnap/core/SetAdminPassword`

This action sets the device's admin password.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
`adminPassword`	`string`	no	The new admin password of the device. Devices may restrict the length and content of their admin password as they see fit. Clients can invoke the GetAdminPasswordRestrictions action to determine the specific restrictions imposed by the device.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/core/SetAdminPasswordResult`

Value	Description
OK	Success.
Error	
ErrorInvalidAdminPassword	The specified admin password is not valid for this device.
_ErrorAbortedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/core/SetAdminPassword"

{
  "adminPassword": "letmeIN!"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

Structures

UnicodeRange

`http://cisco.com/jnap/unicode/UnicodeRange`

Enumerations and structures related to Unicode. This structure represents a range of Unicode codepoints.

Structure Members

Member Name	Type	Optional	Description
`lowCodepoint`	`int`	no	The lowest Unicode codepoint in the range.
`highCodepoint`	`int`	no	The highest Unicode codepoint in the range.

JNAP - DDNS Service

DDNS Service

Contents

Services

DDNS

http://cisco.com/jnap/ddns/DDNS

This service provides access to a device's DDNS settings.

Service Actions

- GetDDNSSettings
- GetDDNSStatus
- SetDDNSSettings

Actions

GetDDNSSettings

http://cisco.com/jnap/ddns/GetDDNSSettings

This action gets the device's current DDNS settings.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
ddnsProvider	DDNSProvider	no	The current DDNS provider.
dynDNSSettings	DynDNSSettings	yes	The current DynDNS provider settings. This value will be present if and only if the value of the ddnsProvider parameter is DynDNS.
tzoSettings	TZOSettings	yes	The current TZO provider settings. This value will be

			present if and only if the value of the ddnsProvider parameter is TZO.

Result

http://cisco.com/jnap/ddns/GetDDNSSettingsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action: "http://cisco.com/jnap/ddns/GetDDNSSettings"

     {
     }
```

```
HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK",
       "output": {
         "ddnsProvider": "DynDNS",
         "dynDNSSettings": {
           "username": "bob",
           "password": "iambob",
           "hostName": "kingdomofbob",
           "isWildcardEnabled": false,
           "mode": "Dynamic",
           "isMailExchangeEnabled": true,
           "mailExchangeSettings": {
             "hostName": "bobsmail",
             "isBackup": true
           }
         }
       }
     }
```

GetDDNSStatus

http://cisco.com/jnap/ddns/GetDDNSStatus

This action gets the device's current DDNS status.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
status	DDNSStatus	no	The current DDNS status.

Result

http://cisco.com/jnap/ddns/GetDDNSStatusResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	

_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
   Host: 192.168.1.1
   Content-Type: application/json; charset=utf-8
   Content-Length: [number of octets in request body]
   X-JNAP-Action: "http://cisco.com/jnap/ddns/GetDDNSStatus"

   {
   }
```

```
HTTP/1.1 200 OK
   Content-Type: application/json; charset=utf-8
   Content-Length: [number of octets in response body]

   {
    "result": "OK",
    "output": {
      "status": "Success"
    }
   }
```

SetDDNSSettings

http://cisco.com/jnap/ddns/SetDDNSSettings

This action sets the devices's DDNS settings.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
ddnsProvider	DDNSProvider	no	The desired DDNS provider.
dynDNSSettings	DynDNSSettings	yes	The current DynDNS provider settings. This value must be present if and only if the value of the ddnsProvider parameter is DynDNS.
tzoSettings	TZOSettings	yes	The current TZO provider settings. This value must be present if and only if the value of the ddnsProvider parameter is TZO.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/ddns/SetDDNSSettingsResult

Value	Description

OK	Success.
Error	
ErrorInvalidHostName	The specified host name or mail exchange host name was invalid.
ErrorInvalidPassword	The specified password was invalid.
ErrorInvalidUsername	The specified username was invalid.
ErrorMissingDynDNSSettings	The DDNS provider was specified as DynDNS, but no DynDNS settings were specified.
ErrorMissingMailExchangeSettings	Mail exchange was specified as enabled, but no mail exchange settings were specified.
ErrorMissingTZOSettings	The DDNS provider was specified as TZO, but no TZO settings were specified.
ErrorSuperfluousDynDNSSettings	DynDNS settings were specified even though the specified DDNS provider was not DynDNS.
ErrorSuperfluousMailExchangeSettings	Mail exchange settings were specified even though mail exchange was specified as disabled.
ErrorSuperfluousTZOSettings	TZO settings were specified even though the specified DDNS provider was not TZO.
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	

_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action: "http://cisco.com/jnap/ddns/SetDDNSSettings"

     {
       "ddnsProvider": "DynDNS",
       "dynDNSSettings": {
         "username": "bob",
         "password": "iambob",
         "hostName": "kingdomofbob",
         "isWildcardEnabled": false,
         "mode": "Dynamic",
         "isMailExchangeEnabled": true,
         "mailExchangeSettings": {
           "hostName": "bobsmail",
           "isBackup": true
         }
       }
     }
```

```
HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK"
     }
```

Structures

DynDNSMailExchangeSettings

http://cisco.com/jnap/ddns/DynDNSMailExchangeSettings

DynDNS mail exchange settings.

Structure Members

Member Name	Type	Optional	Description
hostName	string	no	The mail exchange host name. This value cannot be an empty string.
isBackup	bool	no	Whether the mail exchange is a backup MX. If this value is false, the mail exchange is the primary mail relay.

DynDNSSettings

http://cisco.com/jnap/ddns/DynDNSSettings

DynDNS settings.

Structure Members

Member Name	Type	Optional	Description
username	string	no	The username of the DynDNS account. This value cannot be an empty

			string.
password	string	no	The password of the DynDNS account. This value cannot be an empty string.
hostName	string	no	The DNS host name that is mapped to the device.
isWildcardEnabled	bool	no	Whether *.hostname is mapped to the device.
mode	DynDNSMode	no	The DynDNS mode.
isMailExchangeEnabled	bool	no	Whether DynDNS also maps a mail exchange hostname.
mailExchangeSettings	DynDNSMailExchangeSettings	yes	The DynDNS mail exchange settings. This member must be present if and only if the value of the isMailExchangeEnabled member is true.

TZOSettings

http://cisco.com/jnap/ddns/TZOSettings

TZO settings.

Structure Members

Member Name	Type	Optional	Description
username	string	no	The username of the TZO account. This value cannot be an empty string.
password	string	no	The password of the TZO account. This value cannot be an empty string.
	string	no	The DNS host name that is mapped to the device.

hostName			

Enumerations

DDNSProvider

http://cisco.com/jnap/ddns/DDNSProvider

This service provides access to a device's DDNS settings. Possible DDNS providers.

Enumeration Values

Value	Description
None	There is no DDNS provider.
DynDNS	DynDNS.org is the DDNS provider.
TZO	TZO.com is the DDNS provider.

DDNSStatus

http://cisco.com/jnap/ddns/DDNSStatus

Possible DDNS status.

Enumeration Values

Value	Description

Connecting	Connecting.
Success	Success.
AuthenticationFailed	Authentication failed.
Failed	Failed.
NotEnabled	Not enabled.

DynDNSMode

http://cisco.com/jnap/ddns/DynDNSMode

Possible DynDNS modes.

Enumeration Values

Value	Description
Dynamic	Dynamic mode.
Static	Static mode.
Custom	Custom mode.

JNAP - Diagnostics

Diagnostics Service

Contents

Services

Diagnostics

http://cisco.com/jnap/diagnostics/Diagnostics

This service provides access to diagnostic tests.

Service Actions

- GetPingStatus
- GetTracerouteStatus
- StartPing
- StartTraceroute
- StopPing
- StopTraceroute

Actions

GetPingStatus

http://cisco.com/jnap/diagnostics/GetPingStatus
This action gets the status of the ping test.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
isRunning	bool	no	Whether the ping test is currently running.
pingLog	string	no	The output of the pending or most recently completed ping test.

Result

http://cisco.com/jnap/diagnostics/GetPingStatusResult

Value	Description
OK	Success.

Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action: "http://cisco.com/jnap/diagnostics/GetPingStatus"

    {
    }


HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]

    {
      "result": "OK",
      "output": {
        "isRunning": true,
        "pingLog": "PING 192.168.10.1: 32 data bytes"
      }
    }
```

GetTracerouteStatus

http://cisco.com/jnap/diagnostics/GetTracerouteStatus
This action gets the status of the traceroute test.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
isRunning	bool	no	Whether the traceroute test is currently running.
tracerouteLog	string	no	The output of the pending or most recently completed traceroute test.

Result

http://cisco.com/jnap/diagnostics/GetTracerouteStatusResult

Value	Description

OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action:
"http://cisco.com/jnap/diagnostics/GetTracerouteStatus"

    {
    }


HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]

    {
      "result": "OK",
      "output": {
        "isRunning": true,
        "tracerouteLog": "traceroute to 192.168.10.1, 30 hops max, 38 byte
packets"
      }
    }
```

StartPing

http://cisco.com/jnap/diagnostics/StartPing

This service provides access to diagnostic tests. This action starts a ping test.

Input Parameters

Name	Type	Optional	Description
host	string	no	The IP or URL to ping. This must be a valid IPv4 address in dot-decimal notation or a valid host name as defined in RFCs 952 and 1123.
packetSizeBytes	int	no	The ping packet size in bytes. This value must be between 32 and 65500.
pingCount	int	yes	The number of times to ping the host. If this value is not specified, the number of pings is unlimited and the test will continue until StopPing is called. If specified, this value must be greater than 0.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/diagnostics/StartPingResult

Value	Description
OK	Success.
Error	
ErrorInvalidHost	The specified host is invalid.
ErrorInvalidPacketSizeBytes	The specified packet size is invalid.
ErrorInvalidPingCount	The specified ping count is invalid.
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action: "http://cisco.com/jnap/diagnostics/StartPing"

     {
       "host": "192.168.10.1",
       "packetSizeBytes": 32,
       "pingCount": 10
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK"
     }
```

StartTraceroute

http://cisco.com/jnap/diagnostics/StartTraceroute

This action starts a traceroute test.

Input Parameters

Name	Type	Optional	Description
host	string	no	The IP or URL to traceroute. This must be a valid IPv4 address in dot-decimal notation or a valid host name as defined in RFCs 952 and 1123.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/diagnostics/StartTracerouteResult

Value	Description
OK	Success.
Error	
ErrorInvalidHost	The specified host is invalid.
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action: "http://cisco.com/jnap/diagnostics/StartTraceroute"

     {
       "host": "192.168.10.1"
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK"
     }
```

StopPing

http://cisco.com/jnap/diagnostics/StopPing
This action stops the currently running ping test, if any.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/diagnostics/StopPingResult

Value	Description

OK	Success.
Error	
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action: "http://cisco.com/jnap/diagnostics/StopPing"

    {
    }
```

```
HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]

    {
      "result": "OK"
    }
```

StopTraceroute

http://cisco.com/jnap/diagnostics/StopTraceroute

This action stops the currently running traceroute test, if any.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/diagnostics/StopTracerouteResult

Value	Description
OK	Success.
Error	
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	

_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action: "http://cisco.com/jnap/diagnostics/StopTraceroute"

    {
    }


HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]

    {
      "result": "OK"
    }
```

JNAP - Guest Network Service

Guest Network Service

Contents

Services

GuestNetwork

http://cisco.com/jnap/guestnetwork/GuestNetwork
This service provides access to guest network settings on a wireless access point.

Service Actions

- GetGuestNetworkSettings
- SetGuestNetworkSettings

Actions

GetGuestNetworkSettings

http://cisco.com/jnap/guestnetwork/GetGuestNetworkSettings

This action gets the wireless access point's guest network settings.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
isGuestNetworkEnabled	bool	no	Whether the wireless access point's guest network feature is currently enabled.
broadcastGuestSSID	bool	no	Whether SSID broadcast for the guest network is currently enabled.
guestSSID	string	no	The guest network SSID.
	string	no	The guest network password.

guestPassword			
maxSimultaneousGuests	int	no	The maximum number of users that are allowed on the guest network simultaneously.
canEnableGuestNetwork	bool	no	Whether the wireless access point's guest network feature can be enabled, given the wireless access point's current configuration. This information is necessary, for example, if the wireless access point is a dual-band router that only implements the guest network on one of its radios, and that radio is currently disabled. Note that this value will always be true when isGuestNetworkEnabled is true.
guestPasswordRestrictions	GuestPasswordRestrictions	no	The restrictions that the wireless access point imposes on the guest password.
maxSimultaneousGuestsLimit	int	no	The maximum value that the maxSimultaneousGuests parameter can ever be set to.

Result

http://cisco.com/jnap/guestnetwork/GetGuestNetworkSettingsResult

Value	Description
OK	Success.
Error	

_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action:
"http://cisco.com/jnap/guestnetwork/GetGuestNetworkSettings"

     {
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK",
       "output": {
```

```
    "isGuestNetworkEnabled": true,
    "broadcastGuestSSID": true,
    "guestSSID": "HappyNet-guest",
    "guestPassword": "iamaGUEST!",
    "maxSimultaneousGuests": 4,
    "canEnableGuestNetwork": true,
    "guestPasswordRestrictions": {
      "minLength": 4,
      "maxLength": 32,
      "allowedCharacters": [
        {
          "lowCodepoint": 32,
          "highCodepoint": 126
        }
      ]
    },
    "maxSimultaneousGuestsLimit": 100
  }
}
```

SetGuestNetworkSettings

http://cisco.com/jnap/guestnetwork/SetGuestNetworkSettings
This action sets the wireless access point's guest network settings.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
isGuestNetworkEnabled	bool	no	Whether the wireless access point's guest network feature should be enabled.
broadcastGuestSSID	bool	no	Whether SSID broadcast for the guest network should be enabled.

guestSSID	string	no	The new guest network SSID.
guestPassword	string	no	The new guest network password.
maxSimultaneousGuests	int	no	The new maximum number of users that should be allowed on the guest network simultaneously.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/guestnetwork/SetGuestNetworkSettingsResult

Value	Description
OK	Success.
Error	
ErrorCannotEnableGuestNetwork	The guest network feature cannot be enabled due to the current configuration of the wireless access point. Clients should invoke the GetGuestNetworkSettings action and verify that the canEnableGuestNetwork output parameter is true before trying to enable the guest network.
ErrorGuestSSIDConflict	The specified guest network SSID conflicts with another SSID on the wireless access point.
ErrorInvalidGuestPassword	The specified guest network password is too short, too long, or contains characters that are not allowed.
ErrorInvalidGuestSSID	The specified guest network SSID is too short, too long, or contains characters that are not allowed.
ErrorInvalidMaxSimultaneousGuests	The specified maximum number of simultaneous guests is less than 1, or greater than the maximum

	allowed.
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action:
"http://cisco.com/jnap/guestnetwork/SetGuestNetworkSettings"

     {
       "isGuestNetworkEnabled": true,
       "broadcastGuestSSID": true,
       "guestSSID": "HappyNet-guest",
       "guestPassword": "iamaGUEST!",
       "maxSimultaneousGuests": 4
     }
```

```
HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]

    {
      "result": "OK"
    }
```

Structures

GuestPasswordRestrictions

http://cisco.com/jnap/guestnetwork/GuestPasswordRestrictions
This service provides access to guest network settings on a wireless access point. Restrictions that the wireless access point imposes on guest passwords.

Structure Members

Member Name	Type	Optional	Description
minLength	int	no	The minimum length of the guest password, in bytes.
maxLength	int	no	The maximum length of the guest password, in bytes.
allowedCharacters	UnicodeRange[]	no	A list of Unicode codepoint ranges, the union of which represents the complete set of characters that are allowed in the guest password. This array will always contain at least one item.

UnicodeRange

http://cisco.com/jnap/unicode/UnicodeRange

Enumerations and structures related to Unicode. This structure represents a range of Unicode codepoints.

Structure Members

Member Name	Type	Optional	Description
lowCodepoint	int	no	The lowest Unicode codepoint in the range.
highCodepoint	int	no	The highest Unicode codepoint in the range.

JNAP - Locale Service

Locale Service

Contents

Services

Locale

http://cisco.com/jnap/locale/Locale
This service provides access to locale-related device properties and settings.

Service Actions

- GetLocale
- GetTimeSettings
- SetLocale
- SetTimeSettings

Actions

GetLocale

http://cisco.com/jnap/locale/GetLocale
This action gets the device's locale.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
supportedLocales	string[]	no	The list of locales supported by the device, in RFC 3066 format.
locale	string	no	The current locale of the device, in RFC 3066 format.

Result

http://cisco.com/jnap/locale/GetLocaleResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action: "http://cisco.com/jnap/locale/GetLocale"

    {
    }
```

```
HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK",
       "output": {
         "supportedLocales": [
           "example string"
         ],
         "locale": "en-US"
       }
     }
```

GetTimeSettings

http://cisco.com/jnap/locale/GetTimeSettings

This action gets the device's time settings.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
timeZoneID	string	no	The ID of the device's current time zone.
autoAdjustForDST	bool	no	Whether the device adjusts its time zone automatically to account for Daylight Savings Time.
supportedTimeZones	TimeZone[]	no	The list of supported offsets from UTC (in minutes) that the device's local time can be set to.
currentTime	DateTime	no	The current time according to the device's local clock.

Result

http://cisco.com/jnap/locale/GetTimeSettingsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	

_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action: "http://cisco.com/jnap/locale/GetTimeSettings"

    {
    }


HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]

    {
      "result": "OK",
      "output": {
        "timeZoneID": "PST8",
        "autoAdjustForDST": true,
        "supportedTimeZones": [
          {
            "timeZoneID": "PST8",
            "utcOffsetMinutes": -480,
            "observesDST": true,
            "description": "(GMT-08:00) Pacific Time”
          }
        ],
        "currentTime": "2010-09-31T14:30:59Z"
      }
}
```

SetLocale

http://cisco.com/jnap/locale/SetLocale
This action sets the device's locale.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
locale	string	no	The desired locale of the device, in RFC 3066 format.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/locale/SetLocaleResult

Value	Description
OK	Success.
Error	
	The specified locale is not supported by the device.

ErrorUnsupportedLocale	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action: "http://cisco.com/jnap/locale/SetLocale"

     {
       "locale": "en-US"
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
```

```
  "result": "OK"
}
```

SetTimeSettings

http://cisco.com/jnap/locale/SetTimeSettings
This action sets the device's time settings.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
timeZoneID	string	no	The ID of the desired time zone for the device.
autoAdjustForDST	bool	no	Whether the device should adjust its time zone automatically to account for Daylight Savings Time. If the specified time zone does not observe DST, this value must be false.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/locale/SetTimeSettingsResult

Value	Description
	Success.

OK	
Error	
ErrorTimeZoneDoesNotObserveDST	The value of the autoAdjustForDST parameter was specified as true, but the specified time zone does not observe DST.
ErrorUnknownTimeZone	The specified time zone ID does not correspond to any of the time zones supported by the device.
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action: "http://cisco.com/jnap/locale/SetTimeSettings"

    {
```

```
    "timeZoneID": "PST8",
    "autoAdjustForDST": true
  }


HTTP/1.1 200 OK
  Content-Type: application/json; charset=utf-8
  Content-Length: [number of octets in response body]

  {
    "result": "OK"
  }
```

Structures

TimeZone

http://cisco.com/jnap/locale/TimeZone

This service provides access to locale-related device properties and settings. A time zone supported by the device.

Structure Members

Member Name	Type	Optional	Description
timeZoneID	string	no	A unique identifier for the time zone.
utcOffsetMinutes	int	no	The number of minutes that the time zone is offset from Universal Coordinated Time (UTC).
observesDST	bool	no	Whether the time zone observes Daylight Savings Time (DST).
description	string	no	A human-readable description of the time zone.

JNAP - MAC Filter Service

MAC Filter Service

Contents

Services

MACFilter

http://cisco.com/jnap/macfilter/MACFilter
This service provides access to MAC address filter settings on a router or other network infrastructure device.

Service Actions

- GetMACFilterSettings
- SetMACFilterSettings

Actions

GetMACFilterSettings

http://cisco.com/jnap/macfilter/GetMACFilterSettings

This action gets the MAC address filter settings.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
macFilterMode	MACFilterMode	no	The current MAC address filter mode.
macAddresses	MACAddress[]	no	The list of MAC addresses that the filter mode applies to. If the value of the isMACFilterEnabled parameter is false, this value has no effect.
maxMACAddresses	int	no	The maximum number of MAC addresses that can be filtered.

Result

http://cisco.com/jnap/macfilter/GetMACFilterSettingsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
```

```
     X-JNAP-Action:
"http://cisco.com/jnap/macfilter/GetMACFilterSettings"

     {
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK",
       "output": {
         "macFilterMode": "Allow",
         "macAddresses": [
           "00:22:5F:A1:73:C1"
         ],
         "maxMACAddresses": 32
       }
     }
```

SetMACFilterSettings

http://cisco.com/jnap/macfilter/SetMACFilterSettings
This action sets the MAC address filter settings.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
macFilterMode	MACFilterMode	no	The desired MAC address filter mode.
	MACAddress[]	no	The list of MAC addresses that the filter mode applies to.

macAddresses			

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/macfilter/SetMACFilterSettingsResult

Value	Description
OK	Success.
Error	
ErrorDuplicateMACAddresses	The specified MAC address list contains duplicates.
ErrorInvalidMACAddress	The specified MAC address list contains an invalid address.
ErrorTooManyMACAddresses	The specified MAC address list contains too many addresses.
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	

_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action:
"http://cisco.com/jnap/macfilter/SetMACFilterSettings"

     {
       "macFilterMode": "Allow",
       "macAddresses": [
         "00:22:5F:A1:73:C1"
       ]
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK"
     }
```

Enumerations

MACFilterMode

http://cisco.com/jnap/macfilter/MACFilterMode

This service provides access to MAC address filter settings on a router or other network infrastructure device. Possible MAC address filter modes.

Enumeration Values

Value	Description
Disabled	The MAC filter is disabled.
Allow	Only MAC addresses in the list are allowed to join the network.
Deny	All MAC addresses *except* those in the list are allowed to join the network.

JNAP - Network Connections Service

Network Connections Service

Copyright Notice

This document is provided for evaluation purposes only and is not an option, grant or license from Cisco to any potential partner. The disclosure by Cisco of this Confidential Information will not result in any obligation on the part of Cisco to enter into any further agreement with any potential partner with respect to this document, the program or otherwise, nor shall disclosure of this document by Cisco give any potential partner any right to, directly or indirectly, develop, manufacture or sell any product derived from or which uses any of the Confidential Information. References to "partner" in this document are prospective in nature and a potential partner should not assume that receiving this document creates any partnership arrangement with Cisco. This document may be modified from time to time by Cisco, and potential partners should not assume that any part of this document will be included in the final program.

Contents

Services

NetworkConnections

http://cisco.com/jnap/networkconnections/NetworkConnections
This service returns information about the network connections between client devices and the network infrastructure device.

Service Actions

- GetNetworkConnections

Actions

GetNetworkConnections

http://cisco.com/jnap/networkconnections/GetNetworkConnections
This action returns information about the connections between the network infrastructure device which implements this action and the devices which have a direct (single-hop) wired or wireless network connection to it.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
macAddresses	MACAddress[]	yes	An optional list of MAC addresses. If this value is specified, information will be returned **only** for wireless connections in which the wireless client's MAC address appears in the list. If this value is not specified, the returned list of wireless connections will not be filtered by MAC address. Note that if an empty array is specified for this value, the returned list of wireless connections will always be empty.

Output Parameters

Name	Type	Optional	Description
connections	Layer2Connection[]	no	The list of devices, if any, that satisfy the specified filter criteria.

Result

http://cisco.com/jnap/networkconnections/GetNetworkConnectionsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	

_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action:
"http://cisco.com/jnap/networkconnections/GetNetworkConnections"

     {
       "macAddresses": [
         "00:22:5F:A1:73:C1"
       ]
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK",
       "output": {
         "connections": [
           {
             "macAddress": "00:22:5F:A1:73:C1",
             "negotiatedMbps": 123,
             "wireless": {
               "bssid": "00:22:6B:62:B0:0E",
               "isGuest": true,
               "band": "2.4GHz",
               "signalDecibels": -72
             }
           }
         ]
       }
     }
```

Structures

Layer2Connection

http://cisco.com/jnap/networkconnections/Layer2Connection
A connection between a device and its upstream network infrastructure device.

Structure Members

Member Name	Type	Optional	Description
macAddress	MACAddress	no	The MAC address of the connected network adapter on the device.
negotiatedMbps	int	no	The negotiated speed of the connection, in megabits per second.
wireless	WirelessConnection	yes	Information about the wireless connection. This value is only present if the connection is wireless.

WirelessConnection

http://cisco.com/jnap/networkconnections/WirelessConnection
Information about a device's wireless connection to a network infrastructure device.

Structure Members

Member Name	Type	Optional	Description

bssid	MACAddress	no	The BSSID of the wireless access point that the device is connected to.
isGuest	bool	no	Whether the device is connected to the guest network.
band	WirelessBand	no	The wireless band used for the connection.
signalDecibels	int	no	The signal strength as detected by the wireless access point.

Enumerations

WirelessBand

http://cisco.com/jnap/networkconnections/WirelessBand
 This service returns information about the network connections between client devices and the network infrastructure device. Wireless frequency bands.

Enumeration Values

Value	Description
2.4GHz	The 2.4GHz frequency band.
5GHz	The 5GHz frequency band.

JNAP - Owned Network Service

Owned Network Service

Table of Contents

Services

OwnedNetwork

`http://cisco.com/jnap/ownednetwork/OwnedNetwork`

This service provides information about the "owned network" associated with the router.

Service Actions

- GetOwnedNetworkID

Actions

GetOwnedNetworkID

```
http://cisco.com/jnap/ownednetwork/GetOwnedNetworkID
```

This action returns the unique ID of the owned network associated with the router.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
ownedNetworkID	string	yes	The unique ID of the owned network associated with the router. This value will be present if and only if the router is associated with an owned network.

Result

```
http://cisco.com/jnap/ownednetwork/GetOwnedNetworkIDResult
```

Value	Description
OK	Success.
Error	
_ErrorAbortedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	

_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/ownednetwork/GetOwnedNetworkID"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "ownedNetworkID": "SAMPLE_NETWORK"
  }
```

JNAP - Parental Control Service

Parental Control Service

Contents

Services

ParentalControl

http://cisco.com/jnap/parentalcontrol/ParentalControl
This service provides access to a router's parental controls settings.

Service Actions

- GetParentalControlSettings
- SetParentalControlSettings

Actions

GetParentalControlSettings

http://cisco.com/jnap/parentalcontrol/GetParentalControlSettings
This action gets the router's parental control settings.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
isParentalControlEnabled	bool	no	Whether parental control features are currently enabled.
rules	ParentalControlRule[]	no	The current list of parental control rules.
maxRuleDescriptionLength	int	no	The maximum length, in bytes, of a rule's description.
maxRuleMACAddresses	int	no	The maximum number of MAC addresses that can be specified by a

			rule.
maxRuleBlockedURLLength	int	no	The maximum length, in bytes, of a blocked URL specified by a rule.
maxRuleBlockedURLs	int	no	The maximum number of blocked URLs that a rule can specify.
maxRules	int	no	The maximum number of rules that can exist simultaneously.

Result

http://cisco.com/jnap/parentalcontrol/GetParentalControlSettingsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	

_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action:
"http://cisco.com/jnap/parentalcontrol/GetParentalControlSettings"

     {
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK",
       "output": {
         "isParentalControlEnabled": true,
         "rules": [
           {
             "isEnabled": true,
             "description": "no Internet after 9pm",
             "macAddresses": [
               "00:22:5F:A1:73:C1"
             ],
             "wanSchedule": {
               "sunday":
"000000000000000000111111111111111111111111000000",
               "monday":
"000000000000000000111111111111111111111111000000",
               "tuesday":
"000000000000000000111111111111111111111111000000",
               "wednesday":
"000000000000000000111111111111111111111111000000",
               "thursday":
"000000000000000000111111111111111111111111000000",
               "friday":
"000000000000000000111111111111111111111111000000",
               "saturday":
"000000000000000000111111111111111111111111000000"
             },
             "blockedURLs": [
               "example string"
             ]
           }
         ],
         "maxRuleDescriptionLength": 32,
         "maxRuleMACAddresses": 10,
         "maxRuleBlockedURLLength": 32,
         "maxRuleBlockedURLs": 10,
         "maxRules": 14
       }
     }
```

SetParentalControlSettings

http://cisco.com/jnap/parentalcontrol/SetParentalControlSettings
This action sets the router's parental control settings.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
isParentalControlEnabled	bool	no	Whether parental control features should be enabled.
rules	ParentalControlRule[]	no	The desired list of parental control rules.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/parentalcontrol/SetParentalControlSettingsResult

Value	Description
OK	Success.
Error	
ErrorBlockedURLTooLong	One of the blocked URLs of one of the specified rules was longer than the maximum allowed length.
ErrorDescriptionTooLong	The description of one of the specified rules was longer than the maximum allowed length.
ErrorInvalidMACAddress	One of the specified rules contains an invalid MAC address.
ErrorInvalidWANSchedule	One of the specified rules contains an invalid WAN schedule.
ErrorRulesOverlap	A MAC address was specified more than once in the list of rules.
ErrorTooManyBlockedURLs	One of the specified rules contains more than the maximum allowed number of blocked URLs.
ErrorTooManyMACAddresses	One of the specified rules contains more than the maximum allowed number of MAC addresses.
ErrorTooManyRules	The specified list of rules contains more than the maximum allowed number of rules.
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	

_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action:
"http://cisco.com/jnap/parentalcontrol/SetParentalControlSettings"

     {
       "isParentalControlEnabled": true,
       "rules": [
         {
           "isEnabled": true,
           "description": "no Internet after 9pm",
           "macAddresses": [
             "00:22:5F:A1:73:C1"
           ],
           "wanSchedule": {
             "sunday":
"000000000000000000111111111111111111111111000000",
             "monday":
"000000000000000000111111111111111111111111000000",
             "tuesday":
"000000000000000000111111111111111111111111000000",
             "wednesday":
"000000000000000000111111111111111111111111000000",
             "thursday":
"000000000000000000111111111111111111111111000000",
             "friday":
"000000000000000000111111111111111111111111000000",
             "saturday":
"000000000000000000111111111111111111111111000000"
           },
           "blockedURLs": [
             "example string"
           ]
         }
       ]
     }
```

```
HTTP/1.1 200 OK
   Content-Type: application/json; charset=utf-8
   Content-Length: [number of octets in response body]

   {
     "result": "OK"
   }
```

Structures

ParentalControlRule

http://cisco.com/jnap/parentalcontrol/ParentalControlRule
A rule for blocking or allowing access to the WAN or certain websites from specific devices.

Structure Members

Member Name	Type	Optional	Description
isEnabled	bool	no	Whether the rule enabled.
description	string	no	A description of the rule.
	MACAddress[]	no	The MAC addresses that are governed

macAddresses			by the rule. A given MAC address can only appear once in the list of parental control rules.
wanSchedule	WANSchedule	no	The schedule specifying when WAN access should be allowed and blocked during each calendar week.
blockedURLs	string[]	no	The list of URLs that should be blocked by the rule.

WANSchedule

http://cisco.com/jnap/parentalcontrol/WANSchedule

This service provides access to a router's parental control settings. A schedule specifying when WAN access should be enabled during a calendar week.

Each string member represents a WAN access schedule for a day of the week. The string must be exactly 48 characters long. Each character represents a 30-minute interval during the day, beginning at midnight. A "0" character indicates that WAN access should be blocked during the interval; a "1" indicates that WAN access should be allowed. No other characters may appear in the string. For example, the following string indicates that WAN access should only be allowed between 9 AM and 9 PM:

"000000000000000000111111111111111111111111000000"

Structure Members

Member Name	**Type**	**Optional**	**Description**
sunday	string	no	
monday	string	no	
tuesday	string	no	
wednesday	string	no	
thursday	string	no	

friday	string	no	
saturday	string	no	

JNAP - QOS Service

QoS Service

Copyright Notice

This document is provided for evaluation purposes only and is not an option, grant or license from Cisco to any potential partner. The disclosure by Cisco of this Confidential Information will not result in any obligation on the part of Cisco to enter into any further agreement with any potential partner with respect to this document, the program or otherwise, nor shall disclosure of this document by Cisco give any potential partner any right to, directly or indirectly, develop, manufacture or sell any product derived from or which uses any of the Confidential Information. References to "partner" in this document are prospective in nature and a potential partner should not assume that receiving this document creates any partnership arrangement with Cisco. This document may be modified from time to time by Cisco, and potential partners should not assume that any part of this document will be included in the final program.

Contents

Services

QoS

http://cisco.com/jnap/qos/QoS
This service provides access to Quality of Service settings on a router or other network infrastructure device.

Service Actions

- GetLANQoSSettings
- GetQoSSettings
- GetWLANQoSSettings
- SetLANQoSSettings
- SetQoSSettings
- SetWLANQoSSettings
- UpdateAutoAssignedRules

Actions

GetLANQoSSettings

http://cisco.com/jnap/qos/GetLANQoSSettings
This action gets the QoS settings related to prioritizing wired LAN traffic.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
ethernetPortPriorities	QoSPriority[]	no	The priorities of the Ethernet switch ports. The first array entry corresponds to the first switch port, the second entry to the second switch port and so on.

Result

http://cisco.com/jnap/qos/GetLANQoSSettingsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action: "http://cisco.com/jnap/qos/GetLANQoSSettings"

    {
    }
```

```
HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK",
       "output": {
         "ethernetPortPriorities": [
           "Low"
         ]
       }
     }
```

GetQoSSettings

http://cisco.com/jnap/qos/GetQoSSettings

This action gets the QoS settings related to all traffic passing handled by this network infrastructure device.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
isQoSEnabled	bool	no	Whether QoS is currently enabled.
isQoSAutoPrioritizingEnabled	bool	no	Whether QoS auto-prioritizing is currently enabled. QoS auto-prioritizing automatically assigns QoS priorities for devices from values returned from a web-service. When enabled, device priorities are automatically updated as device becomes more accurately identified.
upstreamBandwidthKbps	int	no	The upstream bandwidth, in kilobits per second. This value is used effectively allocate bandwidth between all data streams. The closer this value is to the actual value, the more efficient the QoS system will operate. If this value is 0, the upstream bandwidth is determined automatically.
downstreamBandwidthKbps	int	no	The downstream bandwidth, in kilobits per second. This value is used effectively allocate bandwidth between all data streams. The closer this value is to the actual value, the more efficient the QoS system will operate. If this value is 0, the

			downstream bandwidth is determined automatically.
deviceRules	QoSDeviceRule[]	no	The list of rules that apply to specific devices. This value is returned regardless of the value of *isQoSEnabled*.
applicationRules	QoSApplicationRule[]	no	The list of rules that apply to specific applications. This value is returned regardless of the value of *isQoSEnabled*.
maxDescriptionLength	int	no	The maximum allowed length of a rule description.
maxApplicationRules	int	no	The maximum number of QoS application rules that can exist simultaneously.
maxPortRanges	int	no	The maximum number of port ranges allowed per application rule.
maxDeviceRules	int	no	The maximum number of QoS device rules that can exist simultaneously.

Result

http://cisco.com/jnap/qos/GetQoSSettingsResult

Value	Description
OK	Success.
Error	

_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action: "http://cisco.com/jnap/qos/GetQoSSettings"

     {
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK",
       "output": {
         "isQoSEnabled": true,
         "isQoSAutoPrioritizingEnabled": true,
  "upstreamBandwidthKbps": 0,
  "downstreamBandwidthKbps": 0,
  "deviceRules": [
   {
    "macAddress": "00:22:5F:A1:73:C1",
    "priority": "Medium",
    "trafficType": "Video",
    "description": "DVR"
   }
  ],
  "applicationRules": [
   {
    "portRanges": [
     {
      "protocol": "TCP",
      "firstPort": 27015,
      "lastPort": 27016,
      "priority": "Low"
     }
    ],
    "trafficType": "Generic",
    "description": "MSN Messenger"
   }
  ],
  "maxDescriptionLength": 32,
  "maxApplicationRules": 15,
  "maxPortRanges": 75,
  "maxDeviceRules": 15
 }
}
```

GetWLANQoSSettings

http://cisco.com/jnap/qos/GetWLANQoSSettings

This action gets the QoS settings related to prioritizing wireless LAN traffic.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
isWMMEnabled	bool	no	Whether Wireless Multimedia (WMM) is currently enabled.
isWirelessAcknowledgementEnabled	bool	no	Whether wireless acknowledgement is currently enabled.

Result

http://cisco.com/jnap/qos/GetWLANQoSSettingsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	

_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action: "http://cisco.com/jnap/qos/GetWLANQoSSettings"

     {
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK",
       "output": {
         "isWMMEnabled": true,
         "isWirelessAcknowledgementEnabled": true
       }
     }
```

SetLANQoSSettings

http://cisco.com/jnap/qos/SetLANQoSSettings

This action sets the QoS settings related to prioritizing wired LAN traffic.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
ethernetPortPriorities	QoSPriority[]	no	The priorities of the Ethernet switch ports. The first array entry corresponds to the first switch port, the second entry to the second switch port and so on. The length of this array cannot exceed to number of ports on the device.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/qos/SetLANQoSSettingsResult

Value	Description
OK	Success.
Error	
ErrorInvalidPortCount	The length of the priority array exceeded the number of ports on the device.

_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action: "http://cisco.com/jnap/qos/SetLANQoSSettings"

    {
      "ethernetPortPriorities": [
        "Low"
      ]
    }


HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]

    {
```

```
  "result": "OK"
}
```

SetQoSSettings

http://cisco.com/jnap/qos/SetQoSSettings
This action sets the QoS settings.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
isQoSEnabled	bool	no	Whether QoS should be enabled.
isQoSAutoPrioritizingEnabled	bool	no	Whether QoS auto-prioritizing is currently enabled.
upstreamBandwidthKbps	int	no	The upstream bandwidth, in kilobits per second. This value is used effectively allocate bandwidth between all data streams. The closer this value is to the actual value, the more efficient the QoS system will operate. If this value is 0, the upstream bandwidth is determined automatically.
downstreamBandwidthKbps	int	no	The downstream bandwidth, in kilobits per second. This value is used effectively

			allocate bandwidth between all data streams. The closer this value is to the actual value, the more efficient the QoS system will operate. If this value is 0, the downstream bandwidth is determined automatically.
deviceRules	QoSDeviceRule[]	no	The list of rules that apply to specific devices. If the value of the *isQoSEnabled* parameter is false, the new rules are saved but have no immediate effect.
applicationRules	QoSApplicationRule[]	no	The list of rules that apply to specific applications. If the value of the *isQoSEnabled* parameter is false, the rules are saved but have no immediate no effect.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/qos/SetQoSSettingsResult

Value	Description
OK	Success.
Error	
	Two or more of the application rules contain

ErrorConflictingApplicationRules	conflicting port ranges.
ErrorConflictingDeviceRules	Two or more of the device rules contain conflicting MAC addresses.
ErrorDescriptionTooLong	One of the specified rule descriptions was longer than the maximum allowed length.
ErrorInvalidDownstreamBandwidth	The specified downstream bandwidth was less than 0.
ErrorInvalidPortRange	One of the application rule port ranges contained an invalid port range.
ErrorInvalidUpstreamBandwidth	The specified upstream bandwidth was less than 0.
ErrorTooManyApplicationRules	The number of application rules exceeded the maximum allowed.
ErrorTooManyDeviceRules	The number of device rules exceeded the maximum allowed.
ErrorTooManyPortRanges	One of the application rule port ranges contained more than the maximum allowed number.
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action: "http://cisco.com/jnap/qos/SetQoSSettings"

     {
       "isQoSEnabled": true,
       "isQoSAutoPrioritizingEnabled": true,
       "upstreamBandwidthKbps": 0,
       "downstreamBandwidthKbps": 0,
       "deviceRules": [
         {
           "macAddress": "00:22:5F:A1:73:C1",
           "priority": "Medium",
           "trafficType": "Video",
           "description": "DVR"
         }
       ],
       "applicationRules": [
         {
           "portRanges": [
             {
               "protocol": "TCP",
               "firstPort": 27015,
               "lastPort": 27016,
               "priority": "Low"
             }
           ],
           "trafficType": "Generic",
           "description": "MSN Messenger"
         }
       ]
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK"
     }
```

SetWLANQoSSettings

http://cisco.com/jnap/qos/SetWLANQoSSettings
This action sets the QoS settings related to prioritizing wireless LAN traffic.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
isWMMEnabled	bool	no	Whether Wireless Multimedia (WMM) should be enabled.
isWirelessAcknowledgementEnabled	bool	no	Whether wireless acknowledgement should be enabled.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/qos/SetWLANQoSSettingsResult

Value	Description
OK	Success.
Error	

_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action: "http://cisco.com/jnap/qos/SetWLANQoSSettings"

     {
       "isWMMEnabled": true,
       "isWirelessAcknowledgementEnabled": true
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK"
}
```

UpdateAutoAssignedRules

http://cisco.com/jnap/qos/UpdateAutoAssignedRules

This action refreshes the QoS auto-assigned rule set. It will not overwrite any existing rules. Note: This method makes a call to a web-service API and may take some time to complete, depending on the WAN connection.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/qos/UpdateAutoAssignedRulesResult

Value	Description
OK	Success.
Error	
ErrorCloudUnavailable	The auto-assigned rules could not be updated because the cloud service is currently unavailable.
ErrorTooManyDeviceRules	Applying the auto-assigned would exceed the maximum allowed.
_ErrorDisallowedAction	
_ErrorInvalidInput	

_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action:
"http://cisco.com/jnap/qos/UpdateAutoAssignedRules"

    {
    }


HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]

    {
      "result": "OK"
    }
```

Structures

QoSApplicationPortRange

http://cisco.com/jnap/qos/QoSApplicationPortRange
A QoS port range for an application rule.

Structure Members

Member Name	Type	Optional	Description
protocol	QoSProtocol	no	The protocol of the port range.
firstPort	int	no	The first port in the port range. This value must be between 0 and 65335
lastPort	int	yes	The last port in the port range. This value must be between 0 and 65335 and greater than or equal to *firstPort*. If not present, the value of *firstPort*will be used.
priority	QoSPriority	no	The priority of the traffic matching this port range.

QoSApplicationRule

http://cisco.com/jnap/qos/QoSApplicationRule
A QoS rule for prioritizing network traffic to a set of port ranges used by an application.

Structure Members

Member Name	Type	Optional	Description

portRanges	QoSApplicationPortRange[]	no	The set of port ranges which define this application rule.
trafficType	QoSTrafficType	no	The type of traffic the application will be producing or consuming.
description	string	no	A human-readable description of the rule.

QoSDeviceRule

http://cisco.com/jnap/qos/QoSDeviceRule
A QoS rule for prioritizing network traffic to a specific MAC address. A given MAC address can be associated with at most one rule.

Structure Members

Member Name	Type	Optional	Description
macAddress	MACAddress	no	The MAC address whose traffic is governed by the rule.
priority	QoSPriority	no	The priority assigned to the traffic governed by the rule.
trafficType	QoSTrafficType	no	The type of traffic the device will be typically producing or consuming.
description	string	no	A human-readable description of the rule.

Enumerations

QoSPriority

http://cisco.com/jnap/qos/QoSPriority
This service provides access to Quality of Service settings on a router or other network infrastructure device. Priorities that can be assigned to QoS rules.

Enumeration Values

Value	Description
Low	Low priority.
Normal	Normal (default) priority.
Medium	Medium priority.
High	High priority.

QoSProtocol

http://cisco.com/jnap/qos/QoSProtocol
Protocols supported for QoS application rules.

Enumeration Values

Value	Description
TCP	TCP protocol
UDP	UDP protocol
Both	Both TCP and UDP protocols

QoSTrafficType

http://cisco.com/jnap/qos/QoSTrafficType
Types of traffic understood by QoS rules. The traffic type is used to optimize the QoS system based on certain characteristics specific to the type of data stream.

Enumeration Values

Value	Description
Background	Background traffic. This type should be used for non-urgent types of network traffic, such as FTP.
Generic	Generic/undefined traffic. This type should be used when the traffic type is unknown or does not resemble one of the other supported traffic types.
Voice	Voice traffic. This type should be specified to optimize for traffic with characteristics of voice data.
Video	Video traffic. This type should be specified to optimize for traffic with characteristics of video data.

JNAP - Router LEDs Service

Router LEDs Service

Copyright Notice

This document is provided for evaluation purposes only and is not an option, grant or license from Cisco to any potential partner. The disclosure by Cisco of this Confidential Information will not result in any obligation on the part of Cisco to enter into any further agreement with any potential partner with respect to this document, the program or otherwise, nor shall disclosure of this document by Cisco give any potential partner any right to, directly or indirectly, develop, manufacture or sell any product derived from or which uses any of the Confidential Information. References to "partner" in this document are prospective in nature and a potential partner should not assume that receiving this document creates any partnership arrangement with Cisco. This document may be modified from time to time by Cisco, and potential partners should not assume that any part of this document will be included in the final program.

Contents

Services

RouterLEDs

http://cisco.com/jnap/routerleds/RouterLEDs
This service provides access to a router's LED settings.

Service Actions

- GetRouterLEDSettings
- SetRouterLEDSettings

Actions

GetRouterLEDSettings

http://cisco.com/jnap/routerleds/GetRouterLEDSettings
This service provides access to a router's LED settings. This action gets the router's LED settings.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
isSwitchportLEDEnabled	bool	no	Whether the router's switchport LED is currently enabled.

Result

http://cisco.com/jnap/routerleds/GetRouterLEDSettingsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action:
"http://cisco.com/jnap/routerleds/GetRouterLEDSettings"

     {
     }
```

```
HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK",
       "output": {
         "isSwitchportLEDEnabled": true
       }
     }
```

SetRouterLEDSettings

http://cisco.com/jnap/routerleds/SetRouterLEDSettings

This action sets the router's LED settings.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
isSwitchportLEDEnabled	bool	no	Whether the router's switchport LED should be enabled.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/routerleds/SetRouterLEDSettingsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action:
"http://cisco.com/jnap/routerleds/SetRouterLEDSettings"

     {
       "isSwitchportLEDEnabled": true
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK"
     }
```

JNAP - Router Service

Router Service

Copyright Notice

Table of Contents

Services

Router

`http://cisco.com/jnap/router/Router`

This service provides access to basic properties and settings of a router.

Service Actions

- ConnectPPPWAN
- DisconnectPPPWAN
- GetEthernetPortConnections
- GetIPv6Settings
- GetLANSettings
- GetMACAddressCloneSettings
- GetRoutingSettings
- GetStaticRoutingTable
- GetWANSettings
- GetWANStatus
- Reconnect6rdTunnel
- ReleaseDHCPIPv6WANLease
- ReleaseDHCPWANLease
- RenewDHCPIPv6WANLease
- RenewDHCPWANLease
- SetIPv6Settings
- SetLANSettings
- SetMACAddressCloneSettings
- SetRoutingSettings
- SetWANSettings

Actions

ConnectPPPWAN

`http://cisco.com/jnap/router/ConnectPPPWAN`

This action causes the router to connect to PPP.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/router/ConnectPPPWANResult`

Value	Description
OK	Success.
Error	
ErrorInvalidWANType	The current WAN type is not PPPoE, PPTP, or L2TP.
_ErrorAbortedAction	
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	

`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/ConnectPPPWAN"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

DisconnectPPPWAN

`http://cisco.com/jnap/router/DisconnectPPPWAN`

This action causes the router to disconnect from PPP.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/router/DisconnectPPPWANResult`

Value	**Description**
`OK`	Success.

Error	
ErrorInvalidWANType	The current WAN type is not PPPoE, PPTP, or L2TP.
_ErrorAbortedAction	
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/DisconnectPPPWAN"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

GetEthernetPortConnections

`http://cisco.com/jnap/router/GetEthernetPortConnections`

This action gets information about the router's Ethernet port connections.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
`wanPortConnection`	`EthernetPortConnection`	no	The current state of the WAN Ethernet port.
`lanPortConnections`	`EthernetPortConnection[]`	no	The current state of the LAN Ethernet ports.

Result

`http://cisco.com/jnap/router/GetEthernetPortConnectionsResult`

Value	Description
`OK`	Success.
`Error`	
`_ErrorAbortedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	

`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/GetEthernetPortConnections"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "wanPortConnection": "100Mbps",
    "lanPortConnections": [
      "None"
    ]
  }
}
```

GetIPv6Settings

`http://cisco.com/jnap/router/GetIPv6Settings`

This action gets router settings related to IPv6.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
isIPv6AutomaticEnabled	bool	no	Whether the router current uses IPv6 for all Internet connections.
ipv6rdTunnelMode	IPv6rdTunnelMode	yes	The router's current 6rd tunnel mode. This value will not be present if the value of the isIPv6AutomaticEnabled parameter is true.
ipv6rdTunnelSettings	IPv6rdTunnelSettings	yes	The router's 6rd tunnel settings. This value will not be present unless the value of the isIPv6AutomaticEnabled parameter is false and the value of the ipv6rdTunnelMode parameter is Manual.
duid	string	no	The DUID used by the router for its DHCPv6 transactions.

Result

`http://cisco.com/jnap/router/GetIPv6SettingsResult`

Value	Description
OK	Success.
Error	
_ErrorAbortedAction	

_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/GetIPv6Settings"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "isIPv6AutomaticEnabled": false,
    "ipv6rdTunnelMode": "Automatic",
    "duid": "00:02:03:09:05:05:00:25:9C:12:7B:00"
  }
}
```

GetLANSettings

`http://cisco.com/jnap/router/GetLANSettings`

This action gets router settings related to LAN management.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
`ipAddress`	`IPAddress`	no	The IP address of the router on the LAN. If *isDHCPEnabled* is **true**, this is also the DHCP server's IP address.
`networkPrefixLength`	`int`	no	The network prefix length of the LAN.
`minNetworkPrefixLength`	`int`	no	The minimum supported network prefix length of the LAN.
`maxNetworkPrefixLength`	`int`	no	The maximum supported network prefix length of the LAN.
`hostName`	`string`	no	The desired host name of the router on the LAN.
`minAllowedDHCPLeaseMinutes`	`int`	no	The minimum allowed length, in minutes, of a DHCP lease.
`maxAllowedDHCPLeaseMinutes`	`int`	yes	The maximum allowed

			length, in minutes, of a DHCP lease. If this value is not present, there is no upper limit on the DHCP lease time.
maxDHCPReservationDescriptionLength	int	no	The maximum length, in bytes, of the description member of a DHCP reservation.
isDHCPEnabled	bool	no	Whether the router is currently acting as a DHCP server for other devices on the LAN.
dhcpSettings	DHCPSettings	no	Configurable settings of the router's DHCP server.

Result

http://cisco.com/jnap/router/GetLANSettingsResult

Value	**Description**
OK	Success.
Error	
_ErrorAbortedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	

_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/GetLANSettings"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "ipAddress": "192.0.2.1",
    "networkPrefixLength": 24,
    "minNetworkPrefixLength": 8,
    "maxNetworkPrefixLength": 30,
    "hostName": "myrouter",
    "minAllowedDHCPLeaseMinutes": 123,
    "maxAllowedDHCPLeaseMinutes": 123,
    "maxDHCPReservationDescriptionLength": 15,
    "isDHCPEnabled": true,
    "dhcpSettings": {
      "leaseMinutes": 1440,
      "firstClientIPAddress": "192.0.2.100",
      "lastClientIPAddress": "192.0.2.150",
      "dnsServer1": "203.0.113.103",
      "reservations": [
        {
          "macAddress": "00:22:5F:A1:73:C1",
          "ipAddress": "192.0.2.99",
          "description": "webcam"
        }
      ]
    }
  }
}
```

GetMACAddressCloneSettings

`http://cisco.com/jnap/router/GetMACAddressCloneSettings`

This action gets router settings related to the WAN MAC address.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
`isMACAddressCloneEnabled`	`bool`	no	Whether the router is currently using a cloned MAC address for its WAN interface.
`macAddress`	`MACAddress`	yes	The MAC address of the router's interface to the WAN. This value will be present if and only if the value of the isMACAddressCloneEnabled parameter is true.

Result

`http://cisco.com/jnap/router/GetMACAddressCloneSettingsResult`

Value	Description
`OK`	Success.
`Error`	

_ErrorAbortedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/GetMACAddressCloneSettings"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "isMACAddressCloneEnabled": true,
    "macAddress": "00:22:6B:62:B0:0D"
  }
}
```

GetRoutingSettings

`http://cisco.com/jnap/router/GetRoutingSettings`

This action gets router settings related to routing.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
`isNATEnabled`	`bool`	no	Whether NAT is enabled on the router.
`isDynamicRoutingEnabled`	`bool`	no	Whether Routing Information Protocol (RIP) is enabled on the router. This value can be true if and only if the value of the isNATEnabled member is false.
`entries`	`NamedStaticRouteEntry[]`	no	The static routing entries to other networks or network segments.
`maxStaticRouteEntries`	`int`	no	The maximum number of static routing entries.

Result

`http://cisco.com/jnap/router/GetRoutingSettingsResult`

Value	Description
`OK`	Success.
`Error`	

_ErrorAbortedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/GetRoutingSettings"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "isNATEnabled": true,
    "isDynamicRoutingEnabled": false,
    "entries": [
      {
        "name": "Router1",
        "settings": {
          "interface": "LAN",
          "destinationLAN": "192.0.2.50",
          "networkPrefixLength": 24,
          "gateway": "192.0.2.1"
        }
      }
    ],
    "maxStaticRouteEntries": 20
  }
}
```

GetStaticRoutingTable

`http://cisco.com/jnap/router/GetStaticRoutingTable`

This action gets the static routing table.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
`table`	`StaticRouteEntry[]`	no	The static routing table.

Result

`http://cisco.com/jnap/router/GetStaticRoutingTableResult`

Value	Description
`OK`	Success.
`Error`	
`_ErrorAbortedAction`	
`_ErrorInvalidInput`	

_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/GetStaticRoutingTable"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "table": [
      {
        "interface": "LAN",
        "destinationLAN": "192.0.2.50",
        "networkPrefixLength": 24,
        "gateway": "192.0.2.1"
      }
    ]
  }
}
```

GetWANSettings

`http://cisco.com/jnap/router/GetWANSettings`

This action gets router settings related to the WAN connection.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
wanType	WANType	no	The router's current configured WAN type.
pppoeSettings	PPPoESettings	yes	The router's current PPPoE settings. This value will only be present if the value of the wanType parameter is PPPoE.
tpSettings	TPSettings	yes	The router's current PPTP/L2TP settings. This value will only be present if the value of the wanType parameter is PPTP or L2TP.
telstraSettings	TelstraSettings	yes	The router's current Telstra settings. This value will only be present if the value of the wanType parameter is Telstra.
staticSettings	StaticSettings	yes	The router's current static settings. This value will only be present if the value of the wanType parameter is Static.
bridgeSettings	BridgeSettings	yes	The router's current bridge-mode settings. This value will only be present if the value of the wanType parameter is Bridge.
dsliteSettings	DSLiteSettings	yes	The router's current DS-Lite settings. This value will only be present if the value of the wanType parameter is DSLite.
domainName	string	yes	The current domain name of the router on the WAN, if any. The domain name is sometimes used by ISPs to provide reverse-lookup on the

			WAN IP address of the router. This value will be assigned by the upstream router unless specified in the staticSettings.
`mtu`	`int`	no	The current maximum packet size (maximum transmission unit), in octets, of the WAN connection. If this value is 0, the MTU is determined automatically by the router. If the value of the wanType parameter is Bridge, this value will be 0.

Result

`http://cisco.com/jnap/router/GetWANSettingsResult`

Value	Description
`OK`	Success.
`Error`	
`_ErrorAbortedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
```

```
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/GetWANSettings"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "wanType": "Bridge",
    "bridgeSettings": {
      "useStaticSettings": true,
      "staticSettings": {
        "ipAddress": "192.0.2.110",
        "networkPrefixLength": 24,
        "gateway": "192.0.2.1",
        "dnsServer1": "203.0.113.120",
        "dnsServer2": "203.0.113.183",
        "domainName": "cisco.com"
      }
    },
    "domainName": "cisco.com",
    "mtu": 1500
  }
}
```

GetWANStatus

`http://cisco.com/jnap/router/GetWANStatus`

This action gets the current status of the router's WAN connection.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
`supportedWANTypes`	`WANType[]`	no	The list of WAN types supported by the router.
`isDetectingWANType`	`bool`	no	Whether the router is currently attempting to determine the WAN type that it is connected to.
`detectedWANType`	`WANType`	yes	The WAN type that the router believes it is currently connected to. This value can be present only if the value of the isDetectingWANType member is false.
`wanStatus`	`WANStatus`	no	The router's current WAN connection status.
`wanConnection`	`WANConnectionInfo`	yes	Information about the router's current WAN connection. If the value of the wanStatus parameter is not Connected, this value will not be present.
`state`	`PPPConnectionState`	yes	The PPP connection establishment state. This value must be present if and only if the value of the detectedWANType member is PPPoE.
`wanIPv6Status`	`WANStatus`	no	The router's current WAN IPv6 connection status.
`linkLocalIPv6Address`	`IPv6Address`	yes	The router's current WAN IPv6 link-local IPv6 address. This value will not be present if IPv6 is not enabled or supported on

			the WAN interface.
wanIPv6Connection	WANIPv6ConnectionInfo	yes	Information about the router's current WAN IPv6 connection. If the value of the wanIPv6Status parameter is not Connected, this value will not be present.
macAddress	MACAddress	no	The MAC address on the WAN interface.

Result

`http://cisco.com/jnap/router/GetWANStatusResult`

Value	Description
OK	Success.
Error	
_ErrorAbortedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
```

```
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/GetWANStatus"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "supportedWANTypes": [
      "DHCP"
    ],
    "isDetectingWANType": false,
    "detectedWANType": "DHCP",
    "wanStatus": "Connected",
    "wanConnection": {
      "wanType": "DHCP",
      "ipAddress": "198.51.100.110",
      "networkPrefixLength": 24,
      "gateway": "198.51.100.1",
      "mtu": 1500,
      "dhcpLeaseMinutes": 1440,
      "dnsServer1": "203.0.113.120",
      "dnsServer2": "203.0.113.183"
    },
    "state": "Connected",
    "wanIPv6Status": "Connected",
    "linkLocalIPv6Address": "2001:0DB8:85A3:0000:0000:8A2E:0370:7334",
    "wanIPv6Connection": {
      "wanType": "DHCPv6",
      "networkInfo": {
        "ipAddress": "2001:0DB8:85A3:0000:0000:8A2E:0370:7334",
        "gateway": "2001:0DB8:85A3:0000:0000:8A2E:0370:7334",
        "dhcpLeaseMinutes": 1440,
        "dnsServer1": "2001:0DB8:85A3:0000:0000:8A2E:0370:7334",
        "dnsServer2": "2001:0DB8:85A3:0000:0000:8A2E:0370:7334",
        "dnsServer3": "2001:0DB8:85A3:0000:0000:8A2E:0370:7334"
      }
    },
    "macAddress": "00:22:5F:A1:73:C1"
  }
}
```

Reconnect6rdTunnel

`http://cisco.com/jnap/router/Reconnect6rdTunnel`

This action causes the router to begin reconnecting the 6rd tunnel.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/router/Reconnect6rdTunnelResult`

Value	Description
`OK`	Success.
`Error`	
`ErrorInvalidIPv6WANType`	The current WAN IPv6 connection type is not 6rd tunnel.
`_ErrorAbortedAction`	
`_ErrorDisallowedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	

`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/Reconnect6rdTunnel"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

ReleaseDHCPIPv6WANLease

`http://cisco.com/jnap/router/ReleaseDHCPIPv6WANLease`

This action causes the router to release its DHCP IPv6 lease.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/router/ReleaseDHCPIPv6WANLeaseResult`

Value	Description
`OK`	Success.
`Error`	

ErrorInvalidIPv6WANType	The current WAN IPv6 connection type is not automatic.
_ErrorAbortedAction	
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/ReleaseDHCPIPv6WANLease"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

ReleaseDHCPWANLease

`http://cisco.com/jnap/router/ReleaseDHCPWANLease`

This action causes the router to release its DHCP lease.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/router/ReleaseDHCPWANLeaseResult`

Value	Description
`OK`	Success.
`Error`	
`ErrorInvalidWANType`	The current WAN type is not DHCP.
`_ErrorAbortedAction`	
`_ErrorDisallowedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
```

```
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/ReleaseDHCPWANLease"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

RenewDHCPIPv6WANLease

`http://cisco.com/jnap/router/RenewDHCPIPv6WANLease`

This action causes the router to begin renewing its DHCP IPv6 lease. If a DHCP IPv6 lease already exists, it will be released first.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/router/RenewDHCPIPv6WANLeaseResult`

Value	**Description**
`OK`	Success.
`Error`	
`ErrorInvalidIPv6WANType`	The current WAN IPv6 connection type is not automatic.
`_ErrorAbortedAction`	

`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/RenewDHCPIPv6WANLease"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

RenewDHCPWANLease

`http://cisco.com/jnap/router/RenewDHCPWANLease`

This action causes the router to begin renewing its DHCP lease. If a DHCP lease already exists, it will be released first.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/router/RenewDHCPWANLeaseResult`

Value	Description
`OK`	Success.
`Error`	
`ErrorInvalidWANType`	The current WAN type is not DHCP.
`_ErrorAbortedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/RenewDHCPWANLease"
```

```
{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

SetIPv6Settings

`http://cisco.com/jnap/router/SetIPv6Settings`

This action sets router settings related to IPv6.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
`isIPv6AutomaticEnabled`	`bool`	no	Whether the router should use IPv6 for all Internet connections.
`ipv6rdTunnelMode`	`IPv6rdTunnelMode`	yes	The desired 6rd tunnel mode for the router. This value must be specified if and only if the value of the isIPv6AutomaticEnabled parameter is false.
`ipv6rdTunnelSettings`	`IPv6rdTunnelSettings`	yes	The desired 6rd tunnel settings for the router. This value must be present if and only if the value of the isIPv6AutomaticEnabled parameter is false and the value of the ipv6rdTunnelMode parameter

			is Manual.

Output Parameters

This action does not have any output parameters.

Result

```
http://cisco.com/jnap/router/SetIPv6SettingsResult
```

Value	Description
`OK`	Success.
`Error`	
`ErrorInvalidBorderRelay`	The specified 6rd tunnel border relay was 0.0.0.0.
`ErrorInvalidBorderRelayPrefixLength`	The specified 6rd tunnel border relay prefix length was invalid.
`ErrorInvalidPrefix`	The specified 6rd tunnel prefix was invalid.
`ErrorInvalidPrefixLength`	The specified 6rd tunnel prefix length was invalid.
`ErrorMissingIPv6rdTunnelMode`	IPv6 automatic was specified as disabled, but no 6rd tunnel mode was specified.
`ErrorMissingIPv6rdTunnelSettings`	The 6rd tunnel mode was specified as Manual, but no tunnel settings were specified.
`ErrorSuperfluousIPv6rdTunnelMode`	A 6rd tunnel mode was specified, even though IPv6 automatic was specified as enabled.
`ErrorSuperfluousIPv6rdTunnelSettings`	6rd tunnel settings were specified, even though 6rd tunnel mode was not specified as Manual.
`_ErrorAbortedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	

_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/SetIPv6Settings"

{
  "isIPv6AutomaticEnabled": false,
  "ipv6rdTunnelMode": "Automatic"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

SetLANSettings

`http://cisco.com/jnap/router/SetLANSettings`

This action sets router settings related to LAN management.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description

<table>
<tr><td></td><td></td><td></td><td></td></tr>
<tr>
<td>ipAddress</td>
<td>IPAddress</td>
<td>no</td>
<td>The desired IP address of the router on the LAN.
The following restrictions from RFC 5735 are applied to the router's LAN host address. The router's LAN host address must not fall inside the following reserved ranges:
<table>
<tr><th>Address Block</th><th>Description</th><th>Reference</th></tr>
<tr><td><em>0.0.0.0/8</em></td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>127.0.0.0/8</em></td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>169.254.0.0/16</em></td><td>Link Local</td><td>RFC 3927</td></tr>
<tr><td><em>224.0.0.0/4</em></td><td>Multicast</td><td>RFC 3171</td></tr>
<tr><td><em>240.0.0.0/4</em></td><td>Reserved for Future Use</td><td>RFC 1112, Section 4</td></tr>
<tr><td><em>255.255.255.255/32</em></td><td>Limited Broadcast</td><td>RFC 919, Section 7 RFC 922, Section 7</td></tr>
<tr><td><em>192.168.1.0/24</em>: <strong>192.168.1.0</strong></td><td>Subnetwork ID</td><td>RFC 922, Section 7</td></tr>
<tr><td><em>192.168.1.0/24</em>: <strong>192.168.1.255</strong></td><td>Subnetwork Broadcast Address</td><td>RFC 922, Section 7</td></tr>
</table>
Additionally it must not conflict with the guest LAN subnetwork as returned by <em>http://cisco.com/jnap/guestnetwork/GetGuestNetworkSettings2</em>.</td>
</tr>
<tr>
<td>networkPrefixLength</td>
<td>int</td>
<td>no</td>
<td>The desired network prefix length of the LAN. This value must be between the <em>minNetworkPrefixLength</em> and <em>maxNetworkPrefixLength</em> values returned by <em>GetLANSettings</em>.</td>
</tr>
<tr>
<td>hostName</td>
<td>string</td>
<td>no</td>
<td>The desired host name of the router on the LAN. This value</td>
</tr>
</table>

			must be between 1 and 15 characters and otherwise follow the format restrictions defined in RFC 952.
`isDHCPEnabled`	`bool`	no	Whether the router is currently acting as a DHCP server for other devices on the LAN.
`dhcpSettings`	`DHCPSettings`	yes	Configurable settings of the router's DHCP server. This value may be omitted if *isDHCPEnabled* is false.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/router/SetLANSettingsResult`

Value	**Description**
`OK`	Success.
`Error`	
`ErrorInvalidFirstClientIPAddress`	The specified first client IP address in the allowed range is not valid.
`ErrorInvalidHostName`	The specified host name is not valid.
`ErrorInvalidIPAddress`	The specified router IP address is not valid.
`ErrorInvalidLastClientIPAddress`	The specified last client IP address in the allowed range is not valid.
`ErrorInvalidLeaseMinutes`	The specified DHCP lease length is outside the allowed range.
`ErrorInvalidNetworkPrefixLength`	The specified network prefix is invalid or is not supported by the router.
`ErrorInvalidPrimaryDNSServer`	The specified primary DNS server IP addresses is not invalid.
`ErrorInvalidReservationIPAddress`	The IP address of one of the specified DHCP reservations is not valid.
`ErrorInvalidReservationMACAddress`	The MAC address of one of the specified DHCP reservations

	was 00:00:00:00:00:00.
ErrorInvalidSecondaryDNSServer	The specified secondary DNS server IP addresses is not invalid.
ErrorInvalidTertiaryDNSServer	The specified tertiary DNS server IP addresses is not invalid.
ErrorInvalidWINSServer	The specified WINS server IP address is not invalid.
ErrorMissingDHCPSettings	The DHCP server was specified as enabled, but no DHCP settings were specified.
ErrorReservationDescriptionInvalid	The description value of one of the specified DHCP reservations was invalid.
ErrorReservationDescriptionTooLong	The description of one of the specified DHCP reservations was longer than the maximum allowed length.
ErrorReservationsOverlap	The specified list of DHCP reservations contained more than one reservation for a single IP or MAC address.
_ErrorAbortedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
```

```
X-JNAP-Action: "http://cisco.com/jnap/router/SetLANSettings"

{
  "ipAddress": "192.0.2.1",
  "networkPrefixLength": 24,
  "hostName": "myrouter",
  "isDHCPEnabled": true,
  "dhcpSettings": {
    "leaseMinutes": 1440,
    "firstClientIPAddress": "192.0.2.100",
    "lastClientIPAddress": "192.0.2.150",
    "dnsServer1": "203.0.113.103",
    "reservations": [
      {
        "macAddress": "00:22:5F:A1:73:C1",
        "ipAddress": "192.0.2.99",
        "description": "webcam"
      }
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

SetMACAddressCloneSettings

`http://cisco.com/jnap/router/SetMACAddressCloneSettings`

This action sets router settings related to the WAN MAC address.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
`isMACAddressCloneEnabled`	`bool`	no	Whether the router should use a cloned

			MAC address for its WAN interface.
`macAddress`	`MACAddress`	yes	The desired MAC address of the router's interface to the WAN. This value must be present if and only if the value of the isMACAddressCloneEnabled parameter is true. This value cannot be 0:0:0:0:0:0.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/router/SetMACAddressCloneSettingsResult`

Value	**Description**
`OK`	Success.
`Error`	
`ErrorInvalidMACAddress`	The specified MAC address was 0:0:0:0:0:0.
`ErrorMissingMACAddress`	The value of the isMACAddressCloneEnabled parameter was true but a MAC address was not specified.
`ErrorSuperfluousMACAddress`	The value of the isMACAddressCloneEnabled parameter was false but a MAC address was specified.
`_ErrorAbortedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	

`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/SetMACAddressCloneSettings"

{
  "isMACAddressCloneEnabled": true,
  "macAddress": "00:22:6B:62:B0:0D"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

SetRoutingSettings

`http://cisco.com/jnap/router/SetRoutingSettings`

This action sets router settings related to routing.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
`isNATEnabled`	`bool`	no	Whether NAT is enabled on the router.
`isDynamicRoutingEnabled`	`bool`	no	Whether Routing Information Protocol (RIP) is enabled on the

			router. This value can be true if and only if the value of the isNATEnabled member is false.
entries	NamedStaticRouteEntry[]	no	The static routing entries to other networks or network segments.

Output Parameters

This action does not have any output parameters.

Result

```
http://cisco.com/jnap/router/SetRoutingSettingsResult
```

Value	Description
OK	Success.
Error	
ErrorDuplicateEntryName	More than one static route entry has the same name.
ErrorInvalidDynamicRoutingEnabled	The value of the isNATEnabled parameter is true but isDynamicRoutingEnabled is true.
ErrorInvalidGateway	A static route entry has an invalid gateway IP address or is not within the same subnetwork as the IP address.
ErrorInvalidIPAddress	A static route entry has an invalid IP address.
ErrorInvalidNetworkPrefixLength	A static route entry has an invalid network prefix length.
ErrorTooManyEntries	The specified list of entries contains more than the maximum allowed number of entries.
_ErrorAbortedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	

_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/SetRoutingSettings"

{
  "isNATEnabled": true,
  "isDynamicRoutingEnabled": false,
  "entries": [
    {
      "name": "Router1",
      "settings": {
        "interface": "LAN",
        "destinationLAN": "192.0.2.50",
        "networkPrefixLength": 24,
        "gateway": "192.0.2.1"
      }
    }
  ]
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

SetWANSettings

`http://cisco.com/jnap/router/SetWANSettings`

This action sets router settings related to the WAN connection.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
`wanType`	`WANType`	no	The desired type of the router's WAN connection.
`pppoeSettings`	`PPPoESettings`	yes	The desired PPPoE settings for the router. If the value of the wanType parameter is not PPPoE, this value is ignored if present.
`tpSettings`	`TPSettings`	yes	The desired PPTP/L2TP settings for the router. If the value of the wanType parameter is not PPTP or L2TP, this value is ignored if present.
`telstraSettings`	`TelstraSettings`	yes	The desired Telstra settings for the router. If the value of the wanType parameter is not Telstra, this value is ignored if present.
`staticSettings`	`StaticSettings`	yes	The desired static settings for the router. If the value of the wanType parameter is not Static, this value is ignored if present.
`bridgeSettings`	`BridgeSettings`	yes	The desired bridge-mode settings for the router. If the value of the wanType parameter is not Bridge, this value is ignored if present.
`dsliteSettings`	`DSLiteSettings`	yes	The desired DS-Lite settings for the router. If the value of the wanType parameter is not DS-Lite, this value is ignored if present.
`mtu`	`int`	no	The desired maximum packet size (maximum transmission unit), in octets, of the WAN connection. If this value is 0, the router will determine the MTU of the WAN connection automatically. Otherwise, the allowed values

<table>
<tr><td></td><td></td><td></td><td>depend on the specified WAN type:
<table>
<tr><th>WAN type</th><th>Allowed MTU values</th></tr>
<tr><td>DHCP</td><td>0, 576-1500</td></tr>
<tr><td>PPPoE</td><td>0, 576-1492</td></tr>
<tr><td>PPTP</td><td>0, 576-1460</td></tr>
<tr><td>L2TP</td><td>0, 576-1460</td></tr>
<tr><td>Telstra</td><td>0</td></tr>
<tr><td>DSLite</td><td>0</td></tr>
<tr><td>Static</td><td>0, 576-1500</td></tr>
<tr><td>Bridge</td><td>0</td></tr>
</table>
</td></tr>
</table>

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/router/SetWANSettingsResult`

Value	Description
`OK`	Success.
`Error`	
`ErrorInvalidDomainName`	The specified domain name of the static settings is not valid.
`ErrorInvalidGateway`	The specified gateway IP address is not valid, is not within the same subnetwork as the IP address, or is equal to the IP address.
`ErrorInvalidIPAddress`	The specified IP address is not valid.
`ErrorInvalidMTU`	The specified MTU is not valid.

`ErrorInvalidMaxIdleMinutes`	The specified max idle time is invalid.
`ErrorInvalidNetworkPrefixLength`	The specified network prefix length is not valid.
`ErrorInvalidPassword`	The PPP password is not valid.
`ErrorInvalidPrimaryDNSServer`	The specified primary DNS server IP addresses is not invalid.
`ErrorInvalidReconnectAfterSeconds`	The specified reconnect time is invalid.
`ErrorInvalidSecondaryDNSServer`	The specified secondary DNS server IP addresses is not invalid.
`ErrorInvalidServer`	The specified server IP address is not valid.
`ErrorInvalidServiceName`	The PPPoE service name is not valid.
`ErrorInvalidTertiaryDNSServer`	The specified tertiary DNS server IP addresses is not invalid.
`ErrorInvalidUsername`	The PPP username is not valid.
`ErrorMissingBridgeSettings`	The WAN type was specified as Bridge, but no bridge settings were specified.
`ErrorMissingPPPoESettings`	The WAN type was specified as PPPoE, but no PPPoE settings were specified.
`ErrorMissingStaticSettings`	The WAN type was specified as Static, but no static settings were specified.
`ErrorMissingTPSettings`	The WAN type was specified as PPTP or L2TP, but no TP settings were specified.
`ErrorMissingTelstraSettings`	The WAN type was specified as Telstra, but no Telstra settings were specified.
`ErrorSuperfluousBridgeSettings`	Bridge settings were specified even though the specified WAN type was not Bridge.
`ErrorSuperfluousPPPoESettings`	PPPoE settings were specified even though the specified WAN type was not PPPoE.
`ErrorSuperfluousStaticSettings`	Static settings were specified even though the specified WAN type was not Static.

`ErrorSuperfluousTPSettings`	TP settings were specified even though the specified WAN type was not PPTP or L2TP.
`ErrorSuperfluousTelstraSettings`	Telstra settings were specified even though the specified WAN type was not Telstra.
`ErrorUnsupportedWANType`	The specified WAN type is not supported by the router.
`_ErrorAbortedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/router/SetWANSettings"

{
  "wanType": "PPPoE",
  "pppoeSettings": {
    "username": "catlover21",
    "password": "ilovecats",
    "serviceName": "",
    "behavior": "ConnectOnDemand",
    "maxIdleMinutes": 15,
    "reconnectAfterSeconds": 30
  },
  "mtu": 1492
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

Structures

AFTRSettings

`http://cisco.com/jnap/router/AFTRSettings`

AFTR settings.

Structure Members

Member Name	Type	Optional	Description
`aftrURL`	`string`	yes	ISP Server (AFTR) URL. Either aftrURL or aftrAddress will be set.
`aftrAddress`	`IPv6Address`	yes	ISP Server (AFTR) IPv6 address. Either aftrURL or aftrAddress will be set.

BridgeSettings

`http://cisco.com/jnap/router/BridgeSettings`

Bridge-mode settings.

Structure Members

Member Name	Type	Optional	Description
`useStaticSettings`	`bool`	no	Whether the router's WAN IP address and other settings are statically specified.
`staticSettings`	`StaticSettings`	yes	The static settings used for the connection. This value must be present if and only if the value of the useStaticSettings member is true.

DHCPLease

`http://cisco.com/jnap/router/DHCPLease`

A DHCP client lease.

Structure Members

Member Name	Type	Optional	Description
`macAddress`	`MACAddress`	no	The client's MAC address.
`ipAddress`	`IPAddress`	no	The DHCP-assigned IP address.
`expiration`	`DateTime`	no	The expiration time of the DHCP lease.
`hostName`	`string`	yes	The client's host name, if one was provided in the client's DHCP request.
`clientID`	`string`	yes	The DHCP client identifier, if one was provided in the client's DHCP request.

DHCPReservation

`http://cisco.com/jnap/router/DHCPReservation`

A DHCP reservation. A given MAC address can reserve at most one IP address from the DHCP server.

Structure Members

<table>
<tr><th>Member Name</th><th>Type</th><th>Optional</th><th>Description</th></tr>
<tr><td><code>macAddress</code></td><td><code>MACAddress</code></td><td>no</td><td>The MAC address that the DHCP reservation is for.</td></tr>
<tr><td><code>ipAddress</code></td><td><code>IPAddress</code></td><td>no</td><td>The client IP address that is reserved for the specified MAC address. This value must be in the same subnet as the router's LAN host address (as defined by SetLANSettings), but not equal to the router's LAN host address and not fall inside the following reserved ranges:
<table>
<tr><td>Address Block</td><td>Description</td><td>Reference</td></tr>
<tr><td><i>192.168.1.0/24</i>: <b>192.168.1.0</b></td><td>Subnetwork ID</td><td>RFC 922, Section 7</td></tr>
<tr><td><i>192.168.1.0/24</i>: <b>192.168.1.255</b></td><td>Subnetwork Broadcast Address</td><td>RFC 922, Section 7</td></tr>
</table></td></tr>
<tr><td><code>description</code></td><td><code>string</code></td><td>no</td><td>An optional host name the DHCP server will use to track</td></tr>
</table>

			the client instead of the client-supplied host name in the DHCP client table. Specifying the empty string will use the client-supplied host name for the DHCP table entry. Otherwise this value must be less than the maxDHCPReservationDescriptionLength member of GetLANSettings and follow the format restrictions defined in RFC 952.

DHCPSettings

`http://cisco.com/jnap/router/DHCPSettings`

User-configurable DHCP server settings

Structure Members

<table>
<tr><th>Member Name</th><th>Type</th><th>Optional</th><th>Description</th></tr>
<tr><td><code>leaseMinutes</code></td><td><code>int</code></td><td>no</td><td>The number of minutes for which a new lease issued by DHCP server should be valid. This value must be betwe the minAllowedLeaseMinutes and maxAllowedLeaseMinutes members of the DHCPSetting action.</td></tr>
<tr><td><code>firstClientIPAddress</code></td><td><code>IPAddress</code></td><td>no</td><td>The first client IP address in the range of addresses that should be allocated by the DHCP server. This value mus be in the same subnet as the router's LAN host address defined by SetLANSettings) and not fall inside the follow reserved ranges:
<table>
<tr><th>Address Block</th><th>Description</th><th>Reference</th></tr>
<tr><td><i>192.168.1.0/24</i>: <b>192.168.1.255</b></td><td>Subnetwork Broadcast Address</td><td>RFC 922, Section 7</td></tr>
</table>
This range of DHCP-assignable client IP addresses may included the router's LAN IP address, though it will neve assigned to a client.</td></tr>
<tr><td><code>lastClientIPAddress</code></td><td><code>IPAddress</code></td><td>no</td><td>The last client IP address in the range of addresses that should be allocated by the DHCP server. This value must be greater than or equal to the firstClientIPAddress valu and not fall inside the following reserved ranges:</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>Address Block</td><td>Description</td><td>Reference</td></tr>
<tr><td><i>192.168.1.0/24</i>: <b>192.168.1.255</b></td><td>Subnetwork Broadcast Address</td><td>RFC 922, Section 7</td></tr>
</table>
</td></tr>
<tr><td><code>dnsServer1</code></td><td><code>IPAddress</code></td><td>yes</td><td>
The desired IP address of the primary DNS server, if an The following restrictions from RFC 5735 are applied to primary DNS server address. The primary DNS server m not fall inside the following reserved ranges:
<table>
<tr><td>Address Block</td><td>Description</td><td>Reference</td></tr>
<tr><td><i>0.0.0.0/8</i></td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><i>127.0.0.0/8</i></td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><i>169.254.0.0/16</i></td><td>Link Local</td><td>RFC 392</td></tr>
<tr><td><i>224.0.0.0/4</i></td><td>Multicast</td><td>RFC 317</td></tr>
<tr><td><i>240.0.0.0/4</i></td><td>Reserved for Future Use</td><td>RFC 1112. Section</td></tr>
<tr><td><i>255.255.255.255/32</i></td><td>Limited Broadcast</td><td>RFC 919 Section 7 RFC 922, Section</td></tr>
</table>
</td></tr>
<tr><td><code>dnsServer2</code></td><td><code>IPAddress</code></td><td>yes</td><td>
The desired IP address of the secondary DNS server, if The following restrictions from RFC 5735 are applied to secondary DNS server address. The secondary DNS se must not fall inside the following reserved ranges:
<table>
<tr><td>Address Block</td><td>Description</td><td>Reference</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td><em>0.0.0.0/8</em></td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>127.0.0.0/8</em></td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>169.254.0.0/16</em></td><td>Link Local</td><td>RFC</td></tr>
<tr><td><em>224.0.0.0/4</em></td><td>Multicast</td><td>RFC</td></tr>
<tr><td><em>240.0.0.0/4</em></td><td>Reserved for Future Use</td><td>RFC 1112, Section</td></tr>
<tr><td><em>255.255.255.255/32</em></td><td>Limited Broadcast</td><td>RFC Section 7 RFC 922, Section</td></tr>
</table>
</td></tr>
<tr><td><code>dnsServer3</code></td><td><code>IPAddress</code></td><td>yes</td><td>
The desired IP address of the tertiary DNS server, if any The following restrictions from RFC 5735 are applied to tertiary DNS server address. The tertiary DNS server mu not fall inside the following reserved ranges:
<table>
<tr><td>Address Block</td><td>Description</td><td>Reference</td></tr>
<tr><td><em>0.0.0.0/8</em></td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>127.0.0.0/8</em></td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>169.254.0.0/16</em></td><td>Link Local</td><td>RFC 392</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td><i>224.0.0.0/4</i></td><td>Multicast</td><td>RFC 317</td></tr>
<tr><td><i>240.0.0.0/4</i></td><td>Reserved for Future Use</td><td>RFC 1112, Section</td></tr>
<tr><td><i>255.255.255.255/32</i></td><td>Limited Broadcast</td><td>RFC 919 Section 7 RFC 922, Section</td></tr>
</table>
</td></tr>
<tr><td><code>winsServer</code></td><td><code>IPAddress</code></td><td>yes</td><td>
The desired IP address of the WINS server, if any. The following restrictions from RFC 5735 are applied to WINS server address. The WINS server must not fall ins the following reserved ranges:
<table>
<tr><th>Address Block</th><th>Description</th><th>Reference</th></tr>
<tr><td><i>0.0.0.0/8</i></td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><i>127.0.0.0/8</i></td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><i>169.254.0.0/16</i></td><td>Link Local</td><td>RFC 392</td></tr>
<tr><td><i>224.0.0.0/4</i></td><td>Multicast</td><td>RFC 317</td></tr>
<tr><td><i>240.0.0.0/4</i></td><td>Reserved for Future Use</td><td>RFC 1112, Section 4</td></tr>
<tr><td><i>255.255.255.255/32</i></td><td>Limited Broadcast</td><td>RFC 919 Section 7 RFC 922, Section 7</td></tr>
</table>
</td></tr>
</table>

reservations	DHCPReservation[]	no	The desired list of DHCP reservations.

DSLiteSettings

`http://cisco.com/jnap/router/DSLiteSettings`

DS-Lite-mode settings.

Structure Members

Member Name	Type	Optional	Description
useManualSettings	bool	no	Whether the router's AFTR address is manually specified.
manualSettings	AFTRSettings	yes	The manual settings used for the connection. This value must be present if and only if the value of the useManualSettings member is true.

IPv6NetworkInfo

`http://cisco.com/jnap/router/IPv6NetworkInfo`

The IPv6 network information.

Structure Members

Member Name	Type	Optional	Description
ipAddress	IPv6Address	no	The IP address of the router on the WAN.
gateway	IPv6Address	yes	The IP address of the WAN gateway.
dhcpLeaseMinutes	int	yes	The number of minutes in the DHCP lease. If the value of the wanType parameter is not *DHCPv6*, this value will not be present.
dnsServer1	IPv6Address	yes	The IP address of the primary DNS server.
dnsServer2	IPv6Address	yes	The IP address of the secondary DNS server, if any.
dnsServer3	IPv6Address	yes	The IP address of the tertiary DNS server, if any.

IPv6rdTunnelSettings

`http://cisco.com/jnap/router/IPv6rdTunnelSettings`

IPv6 rapid deployment tunnel settings.

Structure Members

Member Name	Type	Optional	Description
`prefix`	`IPv6Address`	no	The IPv6 prefix, specified as an IPv6 address with all bits after the prefix set to 0.
`prefixLength`	`int`	no	The IPv6 prefix length. This value must be between 0 and 64.
`borderRelay`	`IPAddress`	no	The IPv4 address of the border relay.
`borderRelayPrefixLength`	`int`	no	The IPv4 border relay network prefix length. This value must be between 0 and 32. The value (prefixLength - borderRelayPrefixLength) must be less than or equal to 32.

NamedStaticRouteEntry

`http://cisco.com/jnap/router/NamedStaticRouteEntry`

Named static routing entry settings.

Structure Members

Member Name	Type	Optional	Description
`name`	`string`	no	The name of the entry.
`settings`	`StaticRouteEntry`	no	The location of the Destination LAN IP address.

PPPoESettings

`http://cisco.com/jnap/router/PPPoESettings`

PPPoE WAN connection settings.

Structure Members

Member Name	Type	Optional	Description
`username`	`string`	no	The username to be used for the connection. This value must be between 0 and 255 characters.
`password`	`string`	no	The password to be used for the connection. This value must be between 0 and 255 characters.
`serviceName`	`string`	no	The service name to be used for the connection. This value may be an empty string for some service providers. This value must be between 0 and 255 characters.
`behavior`	`PPPConnectionBehavior`	no	The connection behavior.
`maxIdleMinutes`	`int`	yes	The maximum number of minutes that the connection can be idle before it is automatically disconnected. This value is required if the value of the behavior member is ConnectOnDemand. This value must be between 1 and 9999.
`reconnectAfterSeconds`	`int`	yes	The number of seconds to wait before automatically reconnecting after the connection is disconnected. This value is required if the value of the behavior member is KeepAlive. This value must be between 20 and 180.

StaticRouteEntry

`http://cisco.com/jnap/router/StaticRouteEntry`

Static routing entry settings.

Structure Members

<table>
<tr><th>Member Name</th><th>Type</th><th>Optional</th><th>Description</th></tr>
<tr><td>interface</td><td>DestinationInterface</td><td>no</td><td>The interface the route applies to.</td></tr>
<tr><td>destinationLAN</td><td>IPAddress</td><td>no</td><td>The IP address of the remote network of the static rou The following restrictions from RFC 5735 are applied static route address. The static route address must no inside the following reserved ranges:
<table>
<tr><th>Address Block</th><th>Description</th><th>Refere</th></tr>
<tr><td><em>0.0.0.0/8</em></td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>127.0.0.0/8</em></td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>169.254.0.0/16</em></td><td>Link Local</td><td>RFC 3</td></tr>
<tr><td><em>224.0.0.0/4</em></td><td>Multicast</td><td>RFC 3</td></tr>
<tr><td><em>240.0.0.0/4</em></td><td>Reserved for Future Use</td><td>RFC 1112, Section</td></tr>
<tr><td><em>255.255.255.255/32</em></td><td>Limited Broadcast</td><td>RFC 9 Section 7 RFC 922, Section</td></tr>
</table>
</td></tr>
<tr><td>networkPrefixLength</td><td>int</td><td>no</td><td>The network prefix length for the Destination LAN IP address. This value must be between 8 and 30.</td></tr>
<tr><td>gateway</td><td>IPAddress</td><td>yes</td><td>The IP address of the gateway server that enables communication with the remote network. This value m omitted if the <em>interface</em> member is set to <strong>WAN</strong>, in whic the default gateway of the WAN interface will be used</td></tr>
</table>

The following restrictions from RFC 5735 are applied static route gateway address. This value must be in th same subnet as the router's LAN host address (as de by SetLANSettings) for static routes on the *LAN* interf and must not be in the same subnet as the router's LA address for static routes on the *Internet* interface. Additionally it must not fall inside the following reserve ranges:

Address Block	Description	Ref
0.0.0.0/8	"This" Network	RFC 112 Sectiion 3.2.1.
127.0.0.0/8	Loopback	RFC 112 Section 3.2.1.
169.254.0.0/16	Link Local	RFC 3
224.0.0.0/4	Multicast	RFC 3
240.0.0.0/4	Reserved for Future Use	RFC 111 Section
255.255.255.255/32	Limited Broadcast	RFC 9 Section 7 RF 922, Section
192.168.1.0/24: **192.168.1.0**	Subnetwork ID	RFC 9 Section
192.168.1.0/24: **192.168.1.255**	Subnetwork Broadcast Address	RFC 9 Section

StaticSettings

`http://cisco.com/jnap/router/StaticSettings`

Static IP WAN connection settings.

Structure Members

<table>
<tr><th>Member Name</th><th>Type</th><th>Optional</th><th>Description</th></tr>
<tr><td>ipAddress</td><td>IPAddress</td><td>no</td><td>The host IP address of the router on the WAN.
The following restrictions from RFC 5735 are applied to the WAN host address. The WAN host address must not fall inside the following reserved ranges:
<table>
<tr><td>Address Block</td><td>Description</td><td>Reference</td></tr>
<tr><td>*0.0.0.0/8*</td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td>*127.0.0.0/8*</td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td>*169.254.0.0/16*</td><td>Link Local</td><td>RFC 3927</td></tr>
<tr><td>*224.0.0.0/4*</td><td>Multicast</td><td>RFC 3171</td></tr>
<tr><td>*240.0.0.0/4*</td><td>Reserved for Future Use</td><td>RFC 1112, Section 4</td></tr>
<tr><td>*255.255.255.255/32*</td><td>Limited Broadcast</td><td>RFC 919, Section 7 RFC 922, Section 7</td></tr>
<tr><td>*192.168.1.0/24*: **192.168.1.0**</td><td>Subnetwork ID</td><td>RFC 922, Section 7</td></tr>
<tr><td>*192.168.1.0/24*: **192.168.1.255**</td><td>Subnetwork Broadcast Address</td><td>RFC 922, Section 7</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>networkPrefixLength</td><td>int</td><td>no</td><td>The WAN network prefix length. This value must be between 8 and 30.</td></tr>
<tr><td>gateway</td><td>IPAddress</td><td>no</td><td>The IP address of the WAN gateway.
The following restrictions from RFC 5735 are applied to the WAN gateway address. The WAN gateway address must not fall inside the following reserved ranges:
<table>
<tr><td>Address Block</td><td>Description</td><td>Reference</td></tr>
<tr><td>*0.0.0.0/8*</td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td>*127.0.0.0/8*</td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td>*169.254.0.0/16*</td><td>Link Local</td><td>RFC 3927</td></tr>
<tr><td>*224.0.0.0/4*</td><td>Multicast</td><td>RFC 3171</td></tr>
<tr><td>*240.0.0.0/4*</td><td>Reserved for Future Use</td><td>RFC 1112, Section 4</td></tr>
<tr><td>*255.255.255.255/32*</td><td>Limited Broadcast</td><td>RFC 919, Section 7 RFC 922, Section 7</td></tr>
<tr><td>*192.168.1.0/24*: **192.168.1.0**</td><td>Subnetwork ID</td><td>RFC 922, Section 7</td></tr>
<tr><td>*192.168.1.0/24*: **192.168.1.255**</td><td>Subnetwork Broadcast Address</td><td>RFC 922, Section 7</td></tr>
</table>
</td></tr>
<tr><td>dnsServer1</td><td>IPAddress</td><td>no</td><td>The IP address of the primary DNS server.
The following restrictions from RFC 5735 are applied to the primary DNS server address. The primary DNS server</td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>address must not fall inside the following reserved ranges:
<table>
<tr><th>Address Block</th><th>Description</th><th>Reference</th></tr>
<tr><td><em>0.0.0.0/8</em></td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>127.0.0.0/8</em></td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>169.254.0.0/16</em></td><td>Link Local</td><td>RFC 3927</td></tr>
<tr><td><em>224.0.0.0/4</em></td><td>Multicast</td><td>RFC 3171</td></tr>
<tr><td><em>240.0.0.0/4</em></td><td>Reserved for Future Use</td><td>RFC 1112, Section 4</td></tr>
<tr><td><em>255.255.255.255/32</em></td><td>Limited Broadcast</td><td>RFC 919, Section 7 RFC 922, Section 7</td></tr>
</table>
</td>
</tr>
<tr>
<td><code>dnsServer2</code></td>
<td><code>IPAddress</code></td>
<td>yes</td>
<td>The IP address of the secondary DNS server, if any.
The following restrictions from RFC 5735 are applied to the secondary DNS server address. The secondary DNS server address must not fall inside the following reserved ranges:
<table>
<tr><th>Address Block</th><th>Description</th><th>Reference</th></tr>
<tr><td><em>0.0.0.0/8</em></td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>127.0.0.0/8</em></td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td><em>169.254.0.0/16</em></td><td>Link Local</td><td>RFC 3927</td></tr>
<tr><td><em>224.0.0.0/4</em></td><td>Multicast</td><td>RFC 3171</td></tr>
<tr><td><em>240.0.0.0/4</em></td><td>Reserved for Future Use</td><td>RFC 1112, Section 4</td></tr>
<tr><td><em>255.255.255.255/32</em></td><td>Limited Broadcast</td><td>RFC 919, Section 7 RFC 922, Section 7</td></tr>
</table>
</td>
</tr>
<tr>
<td><code>dnsServer3</code></td>
<td><code>IPAddress</code></td>
<td>yes</td>
<td>
The IP address of the tertiary DNS server, if any.
The following restrictions from RFC 5735 are applied to the tertiary DNS server address. The tertiary DNS server address must not fall inside the following reserved ranges:
<table>
<tr><td>Address Block</td><td>Description</td><td>Reference</td></tr>
<tr><td><em>0.0.0.0/8</em></td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>127.0.0.0/8</em></td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><em>169.254.0.0/16</em></td><td>Link Local</td><td>RFC 3927</td></tr>
<tr><td><em>224.0.0.0/4</em></td><td>Multicast</td><td>RFC 3171</td></tr>
<tr><td><em>240.0.0.0/4</em></td><td>Reserved for Future Use</td><td>RFC 1112, Section 4</td></tr>
<tr><td><em>255.255.255.255/32</em></td><td>Limited Broadcast</td><td>RFC 919, Section 7 RFC 922,</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td></td><td><table><tr><td></td><td></td><td>Section 7</td></tr></table></td></tr>
<tr><td>domainName</td><td>string</td><td>yes</td><td>The desired domain name of the router on the WAN. This must be a valid host name as defined in RFC 952 and RFC 1123. Omitting this value will clear the currently configured domain name.</td></tr>
</table>

TPSettings

`http://cisco.com/jnap/router/TPSettings`

PPTP/L2TP WAN connection settings.

Structure Members

<table>
<tr><th>Member Name</th><th>Type</th><th>Optional</th><th>Description</th></tr>
<tr><td>useStaticSettings</td><td>bool</td><td>no</td><td>Whether the router's WAN IP address and other s statically specified.</td></tr>
<tr><td>staticSettings</td><td>StaticSettings</td><td>yes</td><td>The static settings used for the connection. This would be present if and only if the value of the useStatic member is true.</td></tr>
<tr><td>server</td><td>IPAddress</td><td>no</td><td>The IP address of the L2TP server.
The following restrictions from RFC 5735 are app L2TP server address. The L2TP server address m fall inside the following reserved ranges:<table><tr><td>Address Block</td><td>Description</td><td>Ref</td></tr><tr><td><i>0.0.0.0/8</i></td><td>"This" Network</td><td>RF 11 Sec 3.2</td></tr><tr><td><i>127.0.0.0/8</i></td><td>Loopback</td><td>RF 11 Sec 3.2</td></tr><tr><td><i>169.254.0.0/16</i></td><td>Link Local</td><td>RF</td></tr></table></td></tr>
</table>

			224.0.0.0/4 Multicast RF *240.0.0.0/4* Reserved for Future Use RF 11 Sec *255.255.255.255/32* Limited Broadcast RF Sec 7 92 Sec
`username`	`string`	no	The username to be used for the connection. This must be between 0 and 255 characters.
`password`	`string`	no	The password to be used for the connection. This must be between 0 and 255 characters.
`behavior`	`PPPConnectionBehavior`	no	The connection behavior.
`maxIdleMinutes`	`int`	yes	The maximum number of minutes that the connec be idle before it is automatically disconnected. Th required if the value of the behavior member is ConnectOnDemand. This value must be between 9999.
`reconnectAfterSeconds`	`int`	yes	The number of seconds to wait before automatica reconnecting after the connection is disconnected value is required if the value of the behavior mem KeepAlive. This value must be between 20 and 1

TelstraSettings

`http://cisco.com/jnap/router/TelstraSettings`

Telstra-mode settings.

Structure Members

Member Name	Type	Optional	Description
`server`	`IPAddress`	no	The IP address of the Telstra server.

<table>
<tr><td></td><td></td><td></td><td>The following restrictions from RFC 5735 are applied to the Telstra server address. The Telstra server address must not fall inside the following reserved ranges:
<table>
<tr><th>Address Block</th><th>Description</th><th>Reference</th></tr>
<tr><td><i>0.0.0.0/8</i></td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><i>127.0.0.0/8</i></td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><i>169.254.0.0/16</i></td><td>Link Local</td><td>RFC 3927</td></tr>
<tr><td><i>224.0.0.0/4</i></td><td>Multicast</td><td>RFC 3171</td></tr>
<tr><td><i>240.0.0.0/4</i></td><td>Reserved for Future Use</td><td>RFC 1112, Section 4</td></tr>
<tr><td><i>255.255.255.255/32</i></td><td>Limited Broadcast</td><td>RFC 919, Section 7 RFC 922, Section 7</td></tr>
</table>
</td></tr>
<tr><td><code>username</code></td><td><code>string</code></td><td>no</td><td>The username to be used for the connection. This value must be between 0 and 255 characters.</td></tr>
<tr><td><code>password</code></td><td><code>string</code></td><td>no</td><td>The password to be used for the connection. This value must be between 0 and 255 characters.</td></tr>
</table>

WANConnectionInfo

`http://cisco.com/jnap/router/WANConnectionInfo`

Information about the router's current WAN connection.

Structure Members

Member Name	**Type**	**Optional**	**Description**
`wanType`	`WANType`	no	The type of the WAN connection.

ipAddress	IPAddress	no	The IP address of the router on the WAN.
networkPrefixLength	int	no	The WAN network prefix length.
gateway	IPAddress	no	The IP address of the WAN gateway.
mtu	int	no	The maximum packet size (maximum transmission unit), in octets, of the WAN connection.
dhcpLeaseMinutes	int	yes	The number of minutes in the DHCP lease. If the value of the wanType parameter is not *DHCP*, this value will not be present.
dnsServer1	IPAddress	no	The IP address of the primary DNS server.
dnsServer2	IPAddress	yes	The IP address of the secondary DNS server, if any.
dnsServer3	IPAddress	yes	The IP address of the tertiary DNS server, if any.

WANIPv6ConnectionInfo

`http://cisco.com/jnap/router/WANIPv6ConnectionInfo`

Information about the router's current WAN IPv6 connection.

Structure Members

Member Name	Type	Optional	Description
wanType	WANIPv6Type	no	The type of the WAN connection.
networkInfo	IPv6NetworkInfo	yes	IPv6 network info. This is set if and only if the type is not 6rd Tunnel.

Enumerations

DestinationInterface

`http://cisco.com/jnap/router/DestinationInterface`

Possible static route entry destination interface locations.

Enumeration Values

Value	Description
`LAN`	LAN.
`Internet`	Internet.

EthernetPortConnection

`http://cisco.com/jnap/router/EthernetPortConnection`

This service provides access to basic properties and settings of a router. Possible Ethernet port connection states.

Enumeration Values

Value	Description
`None`	Nothing is plugged in to the port.
`10Mbps`	The port has a connection with a theoretical maximum throughput of 10 Mbps.
`100Mbps`	The port has a connection with a theoretical maximum throughput of 100 Mbps.
`1Gbps`	The port has a connection with a theoretical maximum throughput of 1 GBps.

IPv6rdTunnelMode

`http://cisco.com/jnap/router/IPv6rdTunnelMode`

Possible modes of the router's IPv6 rapid deployment tunnel.

Enumeration Values

Value	Description
`Disabled`	The 6rd tunnel feature is disabled.
`Automatic`	The 6rd tunnel feature is enabled determines the tunnel settings automatically.
`Manual`	The 6rd tunnel feature is enabled and uses user-specified tunnel settings.

PPPConnectionBehavior

`http://cisco.com/jnap/router/PPPConnectionBehavior`

Types of connection-maintenance behavior used by PPPoE, PPTP, and L2TP.

Enumeration Values

Value	Description
`ConnectOnDemand`	Automatically disconnect after the connection has been idle for a specified amount of time.
`KeepAlive`	Automatically reconnect when the connection is disconnected.

PPPConnectionState

`http://cisco.com/jnap/router/PPPConnectionState`

Possible states of the router's PPP connection establishment.

Enumeration Values

Value	Description
`Connecting`	The router has not finished communicating with the upstream device.
`AuthenticationFailure`	The router failed to authenticate with the upstream device.
`Connected`	The router is connected to the upstream device.

WANIPv6Type

`http://cisco.com/jnap/router/WANIPv6Type`

Types of WAN IPv6 connection supported by the router.

Enumeration Values

Value	Description
`Static`	Static
`Bridge`	Bridge
`6rd Tunnel`	6rd Tunnel
`SLAAC`	**S**tate **l**ess **a**ddress **a**uto **c**onfiguration.
`DHCPv6`	DHCPv6

WANStatus

`http://cisco.com/jnap/router/WANStatus`

Possible statuses of the router's WAN connection.

Enumeration Values

Value	Description
`Indeterminate`	The router cannot determine the status of its WAN connection.
`Disconnected`	The router's WAN port is not connected at layer 2, and the router is not in the process of negotiating connectivity.
`LimitedConnection`	The router's WAN port is connected at layer 2, but the router cannot send or receive IP packets and is not in the process of negotiating IP connectivity.
`Connecting`	The router's WAN port is in the process of negotiating IP connectivity (for example, using DHCP).
`Connected`	The router can send and receive IP packets on the WAN.

WANType

`http://cisco.com/jnap/router/WANType`

Types of WAN connection supported by the router.

Enumeration Values

Value	Description
`DHCP`	A DHCP WAN connection.
`PPPoE`	A DHCP PPPoE WAN connection.
`PPTP`	A PPTP WAN connection.
`L2TP`	A L2TP WAN connection.
`Telstra`	A Telstra WAN connection.
`Static`	A static IP WAN connection.
`Bridge`	The router is in bridge mode.
`DSLite`	A DS-Lite WAN connection.

JNAP - Router UPnP Service

Router UPnP Service

Contents

Services

RouterUPnP

http://cisco.com/jnap/routerupnp/RouterUPnP
This service provides access to a router's UPnP settings.

Service Actions

- GetUPnPSettings
- SetUPnPSettings

Actions

GetUPnPSettings

http://cisco.com/jnap/routerupnp/GetUPnPSettings
This service provides access to a router's UPnP settings. This action gets the router's current UPnP settings.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
isUPnPEnabled	bool	no	Whether UPnP is currently enabled on the router.
canUsersConfigure	bool	no	Whether users are currently allowed to configure UPnP.
canUsersDisableWANAccess	bool	no	Whether users are currently allowed to disable WAN access.

Result

http://cisco.com/jnap/routerupnp/GetUPnPSettingsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
```

```
     X-JNAP-Action:
"http://cisco.com/jnap/routerupnp/GetUPnPSettings"

     {
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK",
       "output": {
         "isUPnPEnabled": true,
         "canUsersConfigure": true,
         "canUsersDisableWANAccess": true
       }
     }
```

SetUPnPSettings

http://cisco.com/jnap/routerupnp/SetUPnPSettings
This action sets the router's UPnP settings.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
isUPnPEnabled	bool	no	Whether UPnP should be enabled on the router.
canUsersConfigure	bool	no	Whether users should be allowed to configure UPnP.
	bool	no	Whether users should be allowed

canUsersDisableWANAccess			to disable WAN access.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/routerupnp/SetUPnPSettingsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action:
"http://cisco.com/jnap/routerupnp/SetUPnPSettings"

    {
      "isUPnPEnabled": true,
      "canUsersConfigure": true,
      "canUsersDisableWANAccess": true
    }


HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]

    {
      "result": "OK"
    }
```

JNAP - Storage Service

Storage Service

Contents

Services

Storage

`http://cisco.com/jnap/storage/Storage`

This service provides access to a device's mounted partitions.

Service Actions

- CreateDirectory
- CreateGroup
- CreateUser
- DeleteGroup
- DeleteUser
- EditGroup
- EditUser
- GetGroups
- GetMountedPartitions
- GetUsers
- ListSubdirectories
- RemoveStorageDevice

Actions

CreateDirectory

`http://cisco.com/jnap/storage/CreateDirectory`

This action creates a directory on a mounted partition.

Input Parameters

Name	Type	Optional	Description
`partitionName`	`string`	no	The name of the mounted partition to create the directory on.
`path`	`string`	no	The name of the directory to create. This can include parent directories which will be created if needed. Each directory must must conform to the following requirements: • must be between 1 and 22 characters long • must not begin or end with a space • must not begin with a / • must not be . or .. • must not contain any ASCII control characters (U+0000 - U+001F) • must not contain any of the following characters: \ ? * : | < > " • must not be an existing non-directly file Relative paths are interpreted as relative to the root directory of the partition.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/storage/CreateDirectoryResult`

Value	Description
`OK`	Success.

Error	
ErrorInvalidPath	The specified directory is not valid.
ErrorUnknownPartition	The specified partition does not correspond to any mounted partition.
_ErrorAbortedAction	
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/storage/CreateDirectory"

{
  "partitionName": "/dev/sdb1",
  "path": "parent/child"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

CreateGroup

`http://cisco.com/jnap/storage/CreateGroup`

This action creates a new user group of the device.

Input Parameters

Name	Type	Optional	Description
`name`	`string`	no	The name of the group. This value must be between 1 and 12 characters long and may only contain characters a-z, A-Z, 0-9, dash(-), and underscore (_). Group names must be unique.
`description`	`string`	no	The description of the group. This value must be between 0 and 63 characters long and may only contain characters a-Z, A-Z, 0-9, dash (-), and space ( ).
`hasWritePermissions`	`bool`	no	Whether the group has write permissions.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/storage/CreateGroupResult`

Value	Description
`OK`	Success.
`Error`	
`ErrorGroupExists`	A group with the specified name already exists.
`ErrorInvalidDescription`	The specified description is invalid.
`ErrorInvalidName`	The specified name is invalid.

`ErrorTooManyGroups`	The group cannot be created because the maximum number of groups has already been reached.
`_ErrorAbortedAction`	
`_ErrorDisallowedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/storage/CreateGroup"

{
  "name": "friends",
  "description": "",
  "hasWritePermissions": false
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

CreateUser

`http://cisco.com/jnap/storage/CreateUser`

This action creates a new user of the device.

Input Parameters

Name	Type	Optional	Description
`name`	`string`	no	The name of the user. This value must be between 1 and 20 characters long and may only contain characters a-z, A-Z, 0-9, dash(-), and underscore (_). User names must be unique.
`fullName`	`string`	no	The full name of the user. This value must be between 0 and 63 characters long and may only contain characters a-Z, A-Z, 0-9, dash (-), and space ( ).
`description`	`string`	no	The description of the user. This value must be between 0 and 63 characters long and may only contain characters a-Z, A-Z, 0-9, dash (-), and space ( ).
`memberOfGroup`	`string`	no	The name of the group that the user is a member of.
`isEnabled`	`bool`	no	Whether the user's account is currently enabled.
`password`	`string`	no	The user's password. This value must be between 4 and 64 characters long and may only contain characters a-Z, A-Z, 0-9.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/storage/CreateUserResult`

Value	Description
`OK`	Success.
`Error`	

`ErrorInvalidDescription`	The specified description is invalid.
`ErrorInvalidFullName`	The specified full name is invalid.
`ErrorInvalidName`	The specified name is invalid.
`ErrorInvalidPassword`	The specified password is invalid.
`ErrorTooManyUsers`	The user cannot be created because the maximum number of user accounts has already been reached.
`ErrorUnknownMemberOfGroup`	The specified group does not exist.
`ErrorUserExists`	A user with the specified name already exists.
`_ErrorAbortedAction`	
`_ErrorDisallowedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/storage/CreateUser"

{
  "name": "john",
  "fullName": "John Smith",
  "description": "",
  "memberOfGroup": "friends",
  "isEnabled": true,
  "password": "iamjohn"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

DeleteGroup

`http://cisco.com/jnap/storage/DeleteGroup`

This action deletes a group of the device.

Input Parameters

Name	Type	Optional	Description
`name`	`string`	no	The name of the group.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/storage/DeleteGroupResult`

Value	Description
`OK`	Success.
`Error`	
`ErrorCannotDeleteGroup`	The specified group is not deletable.
`ErrorUnknownGroup`	There is no group with the specified name.

_ErrorAbortedAction	
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/storage/DeleteGroup"

{
  "name": "friends"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

DeleteUser

`http://cisco.com/jnap/storage/DeleteUser`

This action deletes a user of the device.

Input Parameters

Name	Type	Optional	Description
name	string	no	The name of the user.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/storage/DeleteUserResult`

Value	Description
OK	Success.
Error	
ErrorCannotDeleteUser	The specified user is not deletable.
ErrorUnknownUser	There is no user with the specified name.
_ErrorAbortedAction	
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/storage/DeleteUser"

{
  "name": "john"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

EditGroup

`http://cisco.com/jnap/storage/EditGroup`

This action edits information about a group of the device.

Input Parameters

Name	Type	Optional	Description
name	string	no	The name of the group.
description	string	no	The description of the group. This value must be between 0 and 63 characters long and may only contain characters a-Z, A-Z, 0-9, dash (-), and space ( ).
hasWritePermissions	bool	no	Whether the group has write permissions.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/storage/EditGroupResult`

Value	Description
OK	Success.
Error	
ErrorCannotEditGroup	The specified group is not editable.
ErrorInvalidDescription	The specified description is invalid.
ErrorUnknownGroup	There is no group with the specified name.
_ErrorAbortedAction	
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/storage/EditGroup"

{
  "name": "friends",
  "description": "",
  "hasWritePermissions": false
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

EditUser

`http://cisco.com/jnap/storage/EditUser`

This action edits information about a user of the device.

Input Parameters

Name	Type	Optional	Description
`name`	`string`	no	The name of the user.
`fullName`	`string`	no	The full name of the user. This value must be between 0 and 63 characters long and may only contain characters a-Z, A-Z, 0-9, dash (-), and space ( ).
`description`	`string`	no	The description of the user. This value must be between 0 and 63 characters long and may only contain characters a-Z, A-Z, 0-9, dash (-), and space ( ).
`memberOfGroup`	`string`	no	The name of the group that the user is a member of.
`isEnabled`	`bool`	no	Whether the user's account is currently enabled.
`password`	`string`	no	The user's password. This value must be between 4 and 64 characters long and may only contain characters a-Z, A-Z, 0-9.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/storage/EditUserResult`

Value	Description
OK	Success.
Error	
ErrorCannotEditUser	The specified user is not editable.
ErrorInvalidDescription	The specified description is invalid.
ErrorInvalidFullName	The specified full name is invalid.
ErrorInvalidPassword	The specified password is invalid.
ErrorUnknownMemberOfGroup	The specified group does not exist.
ErrorUnknownUser	There is no user with the specified name.
_ErrorAbortedAction	
_ErrorDisallowedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/storage/EditUser"
```

```
{
  "name": "john",
  "fullName": "John Smith",
  "description": "",
  "memberOfGroup": "friends",
  "isEnabled": true,
  "password": "iamjohn"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

GetGroups

`http://cisco.com/jnap/storage/GetGroups`

This action gets a list of the device's groups.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
`groups`	`Group[]`	no	The list of groups.
`maxGroups`	`int`	no	The maximum number of groups that can exist simultaneously.

Result

`http://cisco.com/jnap/storage/GetGroupsResult`

Value	Description
`OK`	Success.
`Error`	
`_ErrorAbortedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	
`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/storage/GetGroups"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "groups": [
      {
        "name": "admin",
        "description": "administrators",
        "hasWritePermissions": true,
        "isEditable": false,
        "isDeletable": false
      }
    ],
    "maxGroups": 10
  }
}
```

GetMountedPartitions

`http://cisco.com/jnap/storage/GetMountedPartitions`

This action gets information about the device's mounted partitions.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
`partitions`	`Partition[]`	no	The list of the device's mounted partitions.

Result

`http://cisco.com/jnap/storage/GetMountedPartitionsResult`

Value	Description
`OK`	Success.
`Error`	

_ErrorAbortedAction	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/storage/GetMountedPartitions"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "partitions": [
      {
        "partitionName": "/dev/sdb1",
        "storageDeviceName": "/dev/sdb",
        "label": "MYDRIVE",
        "fileSystem": "FAT32",
        "usedKB": 38816,
        "availableKB": 994880
      }
    ]
  }
}
```

GetUsers

`http://cisco.com/jnap/storage/GetUsers`

This action gets a list of the device's users.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
`users`	`User[]`	no	The list of users.
`maxUsers`	`int`	no	The maximum number of users that can exist simultaneously.

Result

`http://cisco.com/jnap/storage/GetUsersResult`

Value	Description
`OK`	Success.
`Error`	
`_ErrorAbortedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	

_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/storage/GetUsers"

{
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "users": [
      {
        "name": "admin",
        "fullName": "administrator",
        "description": "the administrator",
        "memberOfGroup": "admin",
        "isEnabled": true,
        "isEditable": false,
        "isDeletable": false
      }
    ],
    "maxUsers": 10
  }
}
```

ListSubdirectories

`http://cisco.com/jnap/storage/ListSubdirectories`

This action gets the list of subdirectories in a directory on a mounted partition.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
`partitionName`	`string`	no	The name of the mounted partition.
`path`	`string`	no	The path to the parent directory whose subdirectories should be listed. Relative paths are interpreted as relative to the root directory of the partition.

Output Parameters

Name	Type	Optional	Description
`subdirectories`	`string[]`	no	The list of subdirectories in the specified directory. The items in the list are directory names, not paths. The path to a subdirectory can be computed by prefixing its name with the value of the path parameter.

Result

`http://cisco.com/jnap/storage/ListSubdirectoriesResult`

Value	Description
`OK`	Success.
`Error`	
`ErrorPathDoesNotExist`	The specified path does not exist.
`ErrorUnknownPartition`	The specified partition does not correspond to any mounted partition.
`_ErrorAbortedAction`	

_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionVerification	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/storage/ListSubdirectories"

{
  "partitionName": "/dev/sdb1",
  "path": "/"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK",
  "output": {
    "subdirectories": [
      "example string"
    ]
  }
}
```

RemoveStorageDevice

`http://cisco.com/jnap/storage/RemoveStorageDevice`

This action removes a physical storage device, unmounting all partitions on the device and making it safe for removal.

Input Parameters

Name	Type	Optional	Description
`storageDeviceName`	`string`	no	The name of the physical storage device. This value is associated with one or more partitions and can be retrieved using the GetMountedPartitions method.

Output Parameters

This action does not have any output parameters.

Result

`http://cisco.com/jnap/storage/RemoveStorageDeviceResult`

Value	Description
`OK`	Success.
`Error`	
`ErrorUnknownStorageDevice`	The specified storage device does not exist on the system.
`_ErrorAbortedAction`	
`_ErrorDisallowedAction`	
`_ErrorInvalidInput`	
`_ErrorInvalidOutput`	
`_ErrorNotReady`	
`_ErrorSessionVerification`	
`_ErrorUnauthorized`	
`_ErrorUnexpected`	
`_ErrorUnknownAction`	

`_ErrorUnknownSession`	

Sample Transaction

```
POST [request-uri] HTTP/1.1
Host: 192.168.1.1
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in request body]
X-JNAP-Action: "http://cisco.com/jnap/storage/RemoveStorageDevice"

{
  "storageDeviceName": "/dev/sdb"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: [number of octets in response body]

{
  "result": "OK"
}
```

Structures

Group

`http://cisco.com/jnap/storage/Group`

Information about a group.

Structure Members

Member Name	Type	Optional	Description
`name`	`string`	no	The name of the group. This value must be between 1 and 12 characters long. Group names must be unique.
`description`	`string`	no	The description of the group. This value must be between 0 and 63 characters long.
`hasWritePermissions`	`bool`	no	Whether members of the group have read-write permission. If this value is false, members of the group have read-only permission.
`isEditable`	`bool`	no	Whether the group can be edited. Some groups such as admin and guest cannot be edited.
`isDeletable`	`bool`	no	Whether the group can be deleted. Some groups such as admin and guest cannot be deleted.

Partition

`http://cisco.com/jnap/storage/Partition`

This service provides access to a device's mounted partitions. Information about and current settings of a partition.

Structure Members

Member Name	Type	Optional	Description
`partitionName`	`string`	no	The name the partition. This value has is merely a unique identifier for a specific partition, to be used as an input argument to various methods.

storageDeviceName	string	no	The name of the physical storage device on which this partition is located. This value is merely a unique identifier for a specific physical storage device, to be used as an input argument to various methods.
label	string	no	The label of the partition.
fileSystem	string	no	The file system used on the partition.
usedKB	long	no	The amount of space currently used on the partition, in kilobytes.
availableKB	long	no	The amount of free space currently available on the partition, in kilobytes.

User

`http://cisco.com/jnap/storage/User`

Information about a user.

Structure Members

Member Name	Type	Optional	Description
name	string	no	The name of the user. This value must be between 1 and 20 characters long. User names must be unique.
fullName	string	no	The full name of the user. This value must be between 0 and 63 characters long.
description	string	no	The description of the user. This value must be between 0 and 63 characters long.
memberOfGroup	string	no	The name of the group that the user is a member of.
isEnabled	bool	no	Whether the user's account is currently enabled.
isEditable	bool	no	Whether the user can be edited. Some users such as admin cannot be edited.
isDeletable	bool	no	Whether the user can be deleted. Some users such as admin and guest cannot be deleted.

JNAP - Wireless AP Service

Wireless AP Service

Contents

Services

WPSServer

http://cisco.com/jnap/wirelessap/WPSServer
This service allows a client to start and stop WPS sessions on the wireless access point using a "soft" pushbutton method rather than a physical button on the AP.

Service Actions

- GetWPSServerSessionStatus
- StartWPSServerSession
- StopWPSServerSession

WPSServer2

http://cisco.com/jnap/wirelessap/WPSServer2

This service extends the WPSServer service, allowing a client to provision the WPS server settings on the wireless access point.

Service Actions

- GetWPSServerSessionStatus
- GetWPSServerSettings
- SetWPSServerSettings
- StartWPSServerSession
- StopWPSServerSession

WirelessAP

http://cisco.com/jnap/wirelessap/WirelessAP
This service provides access to properties and settings of the 802.11 wireless access point.

Service Actions

- GetRadioInfo

- SetRadioSettings

Actions

GetRadioInfo

http://cisco.com/jnap/wirelessap/GetRadioInfo
This action gets information and settings for all of the wireless access point's wireless radios.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
radios	RadioInfo[]	no	The list of information and settings for each radio.

Result

http://cisco.com/jnap/wirelessap/GetRadioInfoResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action: "http://cisco.com/jnap/wirelessap/GetRadioInfo"

     {
     }
```

```
HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]

    {
      "result": "OK",
      "output": {
        "radios": [
          {
            "radioID": "RADIO_2.4GHz",
            "physicalRadioID": "wdev0",
            "bssid": "00:22:5F:A1:73:C1",
            "band": "2.4GHz",
            "supportedModes": [
              "802.11a"
            ],
            "supportedChannels": [
              123
            ],
            "supportedWideChannels": [
              123
            ],
            "supportedSecurityTypes": [
              "None"
            ],
            "maxRADIUSSharedKeyLength": 64,
            "settings": {
              "isEnabled": true,
              "mode": "802.11bgn",
              "ssid": "Grilled-Cheese",
              "broadcastSSID": true,
              "channelWidth": "Standard",
              "channel": 0,
              "security": "WPA2-Personal",
              "wpaPersonalSettings": {
                "passphrase": "grilled cheese is the new internet"
              },
              "wpaEnterpriseSettings": {
                "radiusServer": "192.168.1.200",
                "radiusPort": 1812,
                "sharedKey": "no grilled cheese in my enterprise, thanks"
              }
            }
          }
        ]
      }
    }
```

GetWPSServerSessionStatus

http://cisco.com/jnap/wirelessap/GetWPSServerSessionStatus
This action gets the status of the WPS session, if any, that is currently in progress on the wireless access point.

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
isWPSSessionInProgress	bool	no	Whether a WPS session is currently in progress.
serverPIN	string	no	The WPS server's PIN number.
lastResult	WPSSessionResult	yes	The result of the last WPS session, if present.

Result

http://cisco.com/jnap/wirelessap/GetWPSServerSessionStatusResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action:
"http://cisco.com/jnap/wirelessap/GetWPSServerSessionStatus"

    {
    }


HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]

    {
      "result": "OK",
      "output": {
        "isWPSSessionInProgress": true,
        "serverPIN": "04840954",
        "lastResult": "Failed"
      }
    }
```

GetWPSServerSettings

http://cisco.com/jnap/wirelessap/GetWPSServerSettings

Note:

This action does not require HTTP basic authentication.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

Name	Type	Optional	Description
enabled	bool	no	Whether the WPS server is enabled.

Result

http://cisco.com/jnap/wirelessap/GetWPSServerSettingsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
      Host: 192.168.1.1
      Content-Type: application/json; charset=utf-8
      Content-Length: [number of octets in request body]
      X-JNAP-Action:
"http://cisco.com/jnap/wirelessap/GetWPSServerSettings"

      {
      }


HTTP/1.1 200 OK
      Content-Type: application/json; charset=utf-8
      Content-Length: [number of octets in response body]

      {
        "result": "OK",
        "output": {
          "enabled": true
        }
      }
```

SetRadioSettings

http://cisco.com/jnap/wirelessap/SetRadioSettings
This action sets information and settings for one or more of the wireless access point's wireless radios.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
radios	NewRadioSettings[]	no	The list of settings that should be set. Radios that are not included in this list will not have their settings changed unless they share a physical radio with one of the specified radios.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/wirelessap/SetRadioSettingsResult

Value	Description
OK	Success.
Error	
ErrorInvalidKey	One of the specified WEP keys was not valid.
ErrorInvalidPassphrase	One of the specified WPA passphrases was not valid.
ErrorInvalidRADIUSPort	One of the specified RADIUS port values was not valid.
ErrorInvalidRADIUSServer	One of the specified RADIUS servers was not valid.
ErrorInvalidSSID	One of the specified SSIDs was not between 1 and 32 bytes long.
ErrorInvalidSharedKey	One of the specified WPA-Enterprise shared keys was not valid.
	One of the specified WEP TX key values was not between 1 and 4.

ErrorInvalidTXKey	
ErrorMissingWEPSettings	One of the specified wireless security types was WEP, but no WEP settings were specified for that radio.
ErrorMissingWPAEnterpriseSettings	One of the specified wireless security types was a WPA-Enterprise variant, but the no WPA-Enterprise settings were specified for that radio.
ErrorMissingWPAPersonalSettings	One of the specified wireless security types was a WPA-Personal variant, but no WPA-Personal settings were specified for that radio.
ErrorUnknownRadio	One of the specified radio IDs does not correspond to an actual radio in the access point.
ErrorUnsupportedChannel	One of the specified wireless channels is not supported by that radio.
ErrorUnsupportedMode	One of the specified radios does not support the wireless mode that was specified for it.
ErrorUnsupportedSecurity	One of the specified radios does not support the wireless security type that was specified for it.
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	

_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action:
"http://cisco.com/jnap/wirelessap/SetRadioSettings"

     {
       "radios": [
         {
           "radioID": "RADIO_2.4GHz",
           "settings": {
             "isEnabled": true,
             "mode": "802.11bgn",
             "ssid": "Grilled-Cheese",
             "broadcastSSID": true,
             "channelWidth": "Standard",
             "channel": 0,
             "security": "WPA2-Personal",
             "wpaPersonalSettings": {
               "passphrase": "grilled cheese is the new internet"
             },
             "wpaEnterpriseSettings": {
               "radiusServer": "192.168.1.200",
               "radiusPort": 1812,
               "sharedKey": "no grilled cheese in my enterprise, thanks"
             }
           }
         }
       ]
     }


HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK"
     }
```

SetWPSServerSettings

http://cisco.com/jnap/wirelessap/SetWPSServerSettings

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
enabled	bool	no	Whether to enable the WPS server.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/wirelessap/SetWPSServerSettingsResult

Value	Description
OK	Success.
Error	
_ErrorInvalidInput	

_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action:
"http://cisco.com/jnap/wirelessap/SetWPSServerSettings"

    {
      "enabled": true
    }


HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]

    {
      "result": "OK"
    }
```

StartWPSServerSession

http://cisco.com/jnap/wirelessap/StartWPSServerSession

This action starts a WPS session on the wireless access point.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

Name	Type	Optional	Description
clientPIN	string	yes	The client-supplied PIN, if any, to use for the WPS session. If this value is not present, the WPS session will be in pushbutton mode and will not use a client PIN. If this value is present, it must be a valid 4 or 8 digit WPS PIN. 8-digit PINs will be verified using the checksum algorithm defined in the WPS specification.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/wirelessap/StartWPSServerSessionResult

Value	Description
OK	Success.
Error	
	The specified client PIN is not valid.

ErrorInvalidClientPIN	
ErrorWPSServerNotEnabled	The WPS server is not enabled.
ErrorWPSSessionAlreadyInProgress	A WPS session is already in progress.
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
     Host: 192.168.1.1
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in request body]
     X-JNAP-Action:
"http://cisco.com/jnap/wirelessap/StartWPSServerSession"

     {
       "clientPIN": "07121976"
     }
```

```
HTTP/1.1 200 OK
     Content-Type: application/json; charset=utf-8
     Content-Length: [number of octets in response body]

     {
       "result": "OK"
     }
```

StopWPSServerSession

http://cisco.com/jnap/wirelessap/StopWPSServerSession

This action stops the current WPS session on the wireless access point.

Note:

This action is safe to call within the context of a transaction.

Input Parameters

This action does not have any input parameters.

Output Parameters

This action does not have any output parameters.

Result

http://cisco.com/jnap/wirelessap/StopWPSServerSessionResult

Value	Description
OK	Success.
Error	

ErrorWPSSessionNotInProgress	No WPS session is currently in progress.
_ErrorInvalidInput	
_ErrorInvalidOutput	
_ErrorNotReady	
_ErrorSessionExpired	
_ErrorTargetUnreachable	
_ErrorUnauthorized	
_ErrorUnexpected	
_ErrorUnknownAction	
_ErrorUnknownSession	
_ErrorUnknownTarget	

Sample Transaction

```
POST [request-uri] HTTP/1.1
    Host: 192.168.1.1
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in request body]
    X-JNAP-Action:
"http://cisco.com/jnap/wirelessap/StopWPSServerSession"

    {
    }


HTTP/1.1 200 OK
    Content-Type: application/json; charset=utf-8
    Content-Length: [number of octets in response body]
```

```
{
  "result": "OK"
}
```

Structures

NewRadioSettings

http://cisco.com/jnap/wirelessap/NewRadioSettings
Desired settings for a wireless radio. This structure is used to change radio settings with the SetRadioSettings action.

Structure Members

Member Name	Type	Optional	Description
radioID	string	no	The unique identifier of the radio.
settings	RadioSettings	no	The radio's desired settings.

RadioInfo

http://cisco.com/jnap/wirelessap/RadioInfo
Information about and current settings of a wireless radio. This structure is returned from the GetRadioInfo action.

Structure Members

Member Name	Type	Optional	Description
radioID	string	no	The unique identifier of the radio. This value never changes for a particular radio, and no two radios is the same wireless access point will have the same radio ID.
physicalRadioID	string	no	The unique identifier of the physical radio hardware. This value never changes for a particular radio. However, multiple radios within the wireless access point may have the same physical radio ID if the access point has a selectable-band radio. In this case, radios that have the same physical radio ID represent mutually-exclusive configurations of the same physical radio, and modifying settings of one radio will cause the settings of the other radio to be modified as well. Only one of the radios that share a physical radio can be enabled at any given time; enabling such a radio will cause the other radios to be disabled.
bssid	MACAddress	no	The radio's BSSID.
band	WirelessBand	no	The radio's wireless frequency band.
supportedModes	WirelessMode[]	no	The list of wireless modes that the radio supports.
supportedChannels	int[]	no	The list of standard (20 MHz) wireless channels that the radio supports

			The presence of the value 0 in this list indicates that the radio supports channel auto-selection. Note that the list of supported channels depends on the access point's configured locale, since certain channels are only supported in certain regions.
supportedWideChannels	int[]	no	The list of wide (40 MHz) wireless channels that the radio supports The presence of the value 0 in this list indicates that the radio supports wide channel auto-selection. This list may be empty if the radio does not support wide channels. Note that the list of supported wide channels depends on the access point's configured locale, since certain wide channels are only supported in certain regions.
supportedSecurityTypes	WirelessSecurity[]	no	The list of wireless security types that the radio supports.
maxRADIUSSharedKeyLength	int	no	The maximum allowed length, in bytes, of a RADIUS shared key value.
settings	RadioSettings	no	The radio's current settings.

RadioSettings

http://cisco.com/jnap/wirelessap/RadioSettings
A wireless radio's settings.

Structure Members

Member Name	Type	Optional	Description
isEnabled	bool	no	Whether the radio is enabled.
mode	WirelessMode	no	The radio's mode.
ssid	string	no	The radio's SSID. The maximum length of this value is 32 bytes.
broadcastSSID	bool	no	Whether SSID broadcast is enabled.
channelWidth	WirelessChannelWidth	no	The wireless channel width that the radio operates on.
channel	int	no	The wireless channel the radio should operate on. The value 0 indicates that the wireless access point should auto-select the channel.
security	WirelessSecurity	no	The radio's security type.
wepSettings	WEPSettings	yes	The radio's WEP settings. When getting wireless radio settings, this value will be present if and only if the value of the security member is WEP. When setting wireless radio settings, this value is required if the value of the security member is WEP; if the security is not WEP, this value is ignored if present.
wpaPersonalSettings	WPAPersonalSettings	yes	The radio's WPA-Personal settings. When getting wireless radio settings, this value will be present if and only if the value of the security member is a WPA-Personal variant. When setting wireless radio settings, this value is required if the value of the security member is a WPA-Personal variant; if the security is not a WPA-Personal variant, this value is ignored if present.
wpaEnterpriseSettings	WPAEnterpriseSettings	yes	The radio's WPA-Enterprise settings. When getting wireless radio settings, this value will be present if and only if the value of the security member is a WPA-Enterprise variant. When setting wireless radio settings, this value is required if the value of the security member is a WPA-Enterprise variant; if the security is not a WPA-Enterprise variant, this value is ignored if present.

WEPSettings

http://cisco.com/jnap/wirelessap/WEPSettings

Settings used when a wireless radio's security type is WEP.

Structure Members

Member Name	Type	Optional	Description
encryption	WEPEncryption	no	The WEP encryption type.
key1	string	no	The first WEP key value. If the value of the encryption member is WEP-64, this value must be exactly 10 hexadecimal digits or an empty string. If the value of the encryption member is WEP-128, this value must be exactly 26 hexadecimal digits or an empty string. Specifying the empty string will clear the key if it was previously set.
key2	string	no	The second WEP key value. If the value of the encryption member is WEP-64, this value must be exactly 10 hexadecimal digits or an empty string. If the value of the encryption member is WEP-128, this value must be exactly 26 hexadecimal digits or an empty string. Specifying the empty string will clear the key if it was previously set.
key3	string	no	The third WEP key value. If the value of the encryption member is WEP-64, this value must be exactly 10 hexadecimal digits or an empty string. If the value of the encryption member is WEP-128, this value must be exactly 26 hexadecimal digits or an empty string. Specifying the empty string will clear the key if it was previously set.
key4	string	no	The fourth WEP key value. If the value of the encryption member is WEP-64, this value must be exactly 10 hexadecimal digits or an empty string. If the value of the encryption member is WEP-128, this value must be exactly 26 hexadecimal digits or an empty string. Specifying the empty string will clear the key if it was previously set.
	int	no	The index specifying which key value to use for

txKey			encryption. The value of the corresponding key cannot be an empty string. This value must be between 1 and 4.

WPAEnterpriseSettings

http://cisco.com/jnap/wirelessap/WPAEnterpriseSettings
Settings used when a wireless radio's security type is a WPA-Enterprise variant.

Structure Members

<table>
<tr><th>Member Name</th><th>Type</th><th>Optional</th><th>Description</th></tr>
<tr><td>radiusServeR</td><td>IPAddress</td><td>no</td><td>The IP address of the RADIUS server.
The following restrictions from RFC 5735 are applied to the RADIUS server address.
<table>
<tr><td>Address Block</td><td>Description</td><td>Reference</td></tr>
<tr><td><i>0.0.0.0/8</i></td><td>"This" Network</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><i>127.0.0.0/8</i></td><td>Loopback</td><td>RFC 1122, Section 3.2.1.3</td></tr>
<tr><td><i>169.254.0.0/16</i></td><td>Link Local</td><td>RFC 3927</td></tr>
<tr><td><i>224.0.0.0/4</i></td><td>Multicast</td><td>RFC 3171</td></tr>
<tr><td><i>240.0.0.0/4</i></td><td>Reserved for Future Use</td><td>RFC 1112, Section 4</td></tr>
<tr><td><i>255.255.255.255/32</i></td><td>Limited Broadcast</td><td>RFC 919, Section 7
RFC 922, Section 7</td></tr>
</table></td></tr>
<tr><td>radiusPort</td><td>int</td><td>no</td><td>The port number of the RADIUS server.</td></tr>
<tr><td>sharedKey</td><td>string</td><td>no</td><td>The RADIUS shared secret. This value must be composed entirely of charact in the range U+0032 - U+007E (printable ASCII) and cannot be an empty string.</td></tr>
</table>

WPAPersonalSettings

http://cisco.com/jnap/wirelessap/WPAPersonalSettings
Settings used when a wireless radio's security type is a WPA-Personal variant.

Structure Members

Member Name	Type	Optional	Description
passphrase	string	no	The WPA passphrase. This value must either be a 8-63 characters in the range U+0032 - U+007E (printable ASCII) or exactly 64 ASCII hexadecimal digits.

Enumerations

WEPAuthentication

http://cisco.com/jnap/wirelessap/WEPAuthentication
The WEP authentication type. This is the authentication type used only for WEP-based security.

Enumeration Values

Value	Description
Auto	Automatically choose the authentication type for each wireless client station.
Open	Use open-system authentication. Wireless client stations will not be required to authenticate to complete the 802.11 association process.
SharedKey	Use shared key authentication. Wireless clients must prove ownership of the WEP shared secret as part of the 802.11 association process.

WEPEncryption

http://cisco.com/jnap/wirelessap/WEPEncryption
Types of WEP encryption.

Enumeration Values

Value	Description
WEP-64	64-bit WEP encryption with a 40-bit key. This value should only be used when the security type is WEP.
WEP-128	128-bit WEP encryption with a 104-bit key. This value should only be used when the security type is WEP.

WPSSessionResult

http://cisco.com/jnap/wirelessap/WPSSessionResult
WPS session result

Enumeration Values

Value	Description
Failed	The last WPS session failed.
Succeeded	The last WPS session succeeded.

WirelessBand

http://cisco.com/jnap/wirelessap/WirelessBand

This service provides access to properties and settings of the 802.11 wireless access point. Wireless frequency bands.

Enumeration Values

Value	Description
2.4GHz	The 2.4GHz frequency band.
5GHz	The 5GHz frequency band.

WirelessBasicTransmissionRate

http://cisco.com/jnap/wirelessap/WirelessBasicTransmissionRate

The allowed set of basic wireless transmission rates. This value is used primarily to allow the radio to support legacy 802.11b wireless client stations.

Enumeration Values

Value	Description
Default	Use the default rate set for the currently configured wireless mode. This should be used to ensure compatibility with the widest range of devices.
All	Allow all data rates supported by the radio.
802.11bCompatible	Use only data rates supported by 802.11b devices (i.e. 1-2Mbps). This should only be used for backwards compatibility with legacy 802.11b devices.

WirelessChannelWidth

http://cisco.com/jnap/wirelessap/WirelessChannelWidth

Possible wireless channel widths.

Enumeration Values

Value	Description
Auto	The wireless access point will automatically determine the wireless channel width.
Standard	The wireless channel is 20 MHz wide.
Wide	The wireless channel is 40 MHz wide.

WirelessMode

http://cisco.com/jnap/wirelessap/WirelessMode

Modes representing types of wireless traffic that a wireless radio will accept.

Enumeration Values

Value	Description
802.11a	The radio only accepts 802.11a traffic.
802.11b	The radio only accepts 802.11b traffic.
802.11g	The radio only accepts 802.11g traffic.
802.11n	The radio only accepts 802.11n traffic.
802.11an	The radio only accepts 802.11a and 802.11n traffic.
802.11bg	The radio only accepts 802.11b and 802.11g traffic.
802.11bn	The radio only accepts 802.11b and 802.11n traffic.

802.11gn	The radio only accepts 802.11g and 802.11n traffic.
802.11bgn	The radio only accepts 802.11b, 802.11g, and 802.11n traffic.

WirelessSecurity

http://cisco.com/jnap/wirelessap/WirelessSecurity

Wireless security types supported by a wireless radio. Each security type represents a combination of an 802.11 wireless protocol, authentication, and encryption. All possible combinations are not represented, only those that are reasonable options for a modern wireless access point to expose.

Enumeration Values

Value	Description
None	No authentication or encryption; commonly referred to as "open".
WEP	WEP protocol with no authentication and WEP encryption using a preshared key; commonly referred to as "WEP shared".
WPA-Personal	WPA protocol with preshared-key authentication and TKIP encryption.
WPA-Enterprise	WPA protocol with RADIUS authentication and TKIP encryption.
WPA2-Personal	WPA2 protocol with preshared-key authentication and AES encryption.
WPA2-Enterprise	WPA2 protocol with RADIUS authentication and AES encryption.
WPA-Mixed-Personal	Either WPA-Personal or WPA2-Personal.
WPA-Mixed-Enterprise	Either WPA-Enterprise or WPA2-Enterprise.

WirelessTransmissionPower

http://cisco.com/jnap/wirelessap/WirelessTransmissionPower
Wireless transmission power.

Enumeration Values

Value	Description
Low	Low transmission power.
Medium	Medium transmission power.
High	High transmission power.