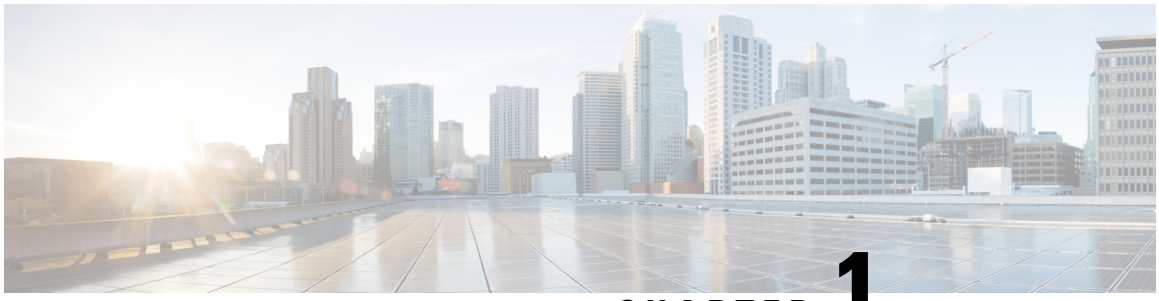
North Bound API User Guide for Cisco IoT Field Network Director, Release 4.x

First Published: 2021-12-17

Last Modified: 2024-01-29

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Overview of the IoT FND North Bound API

Overview

IoT FND maintains a database of inventory information about network devices, groups, properties, metrics, and events. You can use NB API to retrieve network statistics and properties for deployed networked devices. You can also access the database using the IoT FND NB API (Database Queries).

The IoT FND NB API is a Simple Object Access Protocol (SOAP) API that provides methods for:

- Read-only access to the IoT FND database
- Push-based event reporting
- Invoking management operations such as mesh firmware updates, rule creation, and mesh migration

Many APIs return lists of identifiers or objects. Because these lists could be very long in a large network, every method has three optional arguments: queryId, count, and offset. NB APIs use the Web Services Description Language (WSDL) to define network services.

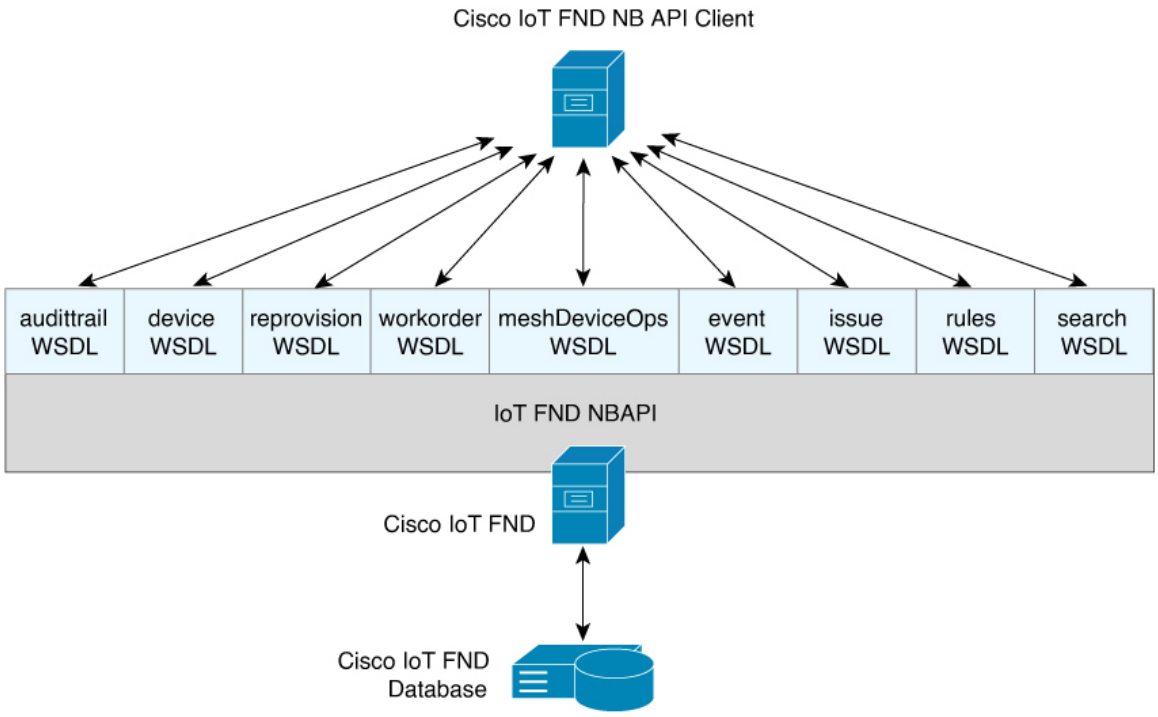
When no argument is included, the call returns the first count list items. The maximum count is 1000 items.

To iterate through the full contents of a list, select a queryId, which is a random string. Then repeat the call using the same queryId, and increment the offset by *count* for each call, starting from 0. When the call returns an empty list, the iteration is complete, and the queryId is invalidated on the server. Reusing queryId starts the request from the beginning. If a particular queryId is not used for 10 minutes, it is cleared to conserve resources.

The API uses HTTPS and HTTP Basic Authentication for username and password authorization, and for sending event data.

Note IoT FND Release 2.1.1-54 and later do not support TLSv1.0 or TLSv1.1 based connections. Only TLS1.2 based connections are supported.

Figure 1: Database Queries



Document Conventions

This document uses the following conventions:

Conventions	Indication
bold font	Commands and keywords and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
courier font	Terminal sessions and information the system displays appear in courier font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means *reader take note* . Notes contain helpful suggestions or references to material not covered in the manual.

Caution Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

Danger **IMPORTANT SAFETY INSTRUCTIONS Means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.**

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at: http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html .

Subscribe to *What's New in Cisco Product Documentation* , which lists all new and revised Cisco technical documentation as an RSS feed and delivers content directly to your desktop using a reader application. The RSS feeds are a free service.

IOT FND NB API Modules

IoT FND defines the following API modules:

API Module	WSDL URL	Method Calls
audittrail Audit Trail API	https://<*server_address*> /nbapi/audittrail?wsdl	deleteAuditTrailsByTime getAuditTrailsByOperation getAuditTrailsByTime getAuditTrailsByUser getAuditTrailsByUserAndIp getAuditTrailsByUserAndOperation getAuditTrailsByUserIpAndOperation

API Module	WSDL URL	Method Calls
device Device Management API	https://<*server_address*> /nbapi/device?wsdl	addDevices exportDevices getJob removeDevices setDevices updateDevices
reprovision Mesh Firmware Migration API	https://<*server_address*> /nbapi/reprovision?wsdl	cancelReprovision showReprovisionStatus startReprovisionByEidList startReprovisionByEidListAbridged startReprovisionByGroup startReprovisionByGroupAbridged
meshDeviceOps Firmware Upgrade API	https://<*server_address*> /nbapi/meshDeviceOps?wsdl	getFirmwareUploadStatus getFirmwareImageInfoList scheduleReload setBackupFirmwareImage startUpload stopUpload
Note Firmware upgrades are only supported on Cisco IOS.		
workorder Workorders API	https://<*server_address*> :portnumber/nbapi/workorder	RequestUserAuthentication RequestSignedAuthorization UploadServiceReport
event Event API	https://<*server_address*> /nbapi/event?wsdl	searchEvents subscribeForCgmeshOutage subscribeForEvents unSubscribeForCgmeshOutage unSubscribeForEvents
issue Issues API	https://<*server_address*> /nbapi/issue?wsdl	searchIssues

API Module	WSDL URL	Method Calls
rules Rules API	https://<*server_address*> /nbapi/rules?wsdl	activateRule createRule deactivateRule dropRule findRulesByName findRulesByUsername
search Search API	https://<*server_address*> /nbapi/search?wsdl	getDeviceDetails getDeletedDevices getUpdatedDeviceDetails getGroups getMetricHistory searchDevices findEidsForIpAddress findEidsForIpAddressByDeviceType

To view the WSDL of the API in a Web browser, use this URL format:

https://<*server_address* >/nbapi/<*api*> ?wsdl

For example:

https://10.27.167.19/nbapi/event?wsdl

Query Syntax

The IoT FND NB API supports the following simple query language:

- Search := filter ?[filter ...]
- Filter := Filter := <fieldName><operator><value>
- Operator := < | <= | > | >= | <> | = | :

Search query examples:

```
"deviceType:cgr1000 uptime>=100 uplinkTxDrops<-50"
"deviceType:cgmesh uptime>=100"
"" (search everything)
"uptime>=100 status:up"
"eid:xyz"
"xyz"
```

Property Field Names for All Devices

The following table describes the property field names available to all devices. Field names are case sensitive.

Table 1: Property Field Names for All Devices

Field Name	Type	Description	Example
eid	string	Unique identifier for the device. For routers and communication modules, this is the string representation of the X.500 distinguished name subject ID contained in the devices X.509 certificate. This field name is obtained from the notice-of-shipment file.	CGR1240/K9 CGR1120/K9
deviceType	enum	Identifier for the device type that indicates which IoT FND module coordinates communications with the device. • Cisco Connected Grid Routers 1000 series are **cgr1000** • Cisco Aggregation Services Routers 1000 series are **asr1000** • Cisco Integrated Services Routers 3900 series are **isr3900** • Cisco 8000 Series Routers are **c8000** • Cisco 800 Series Integrated Services Routers (ISR 800s) are **c800** • Cisco 800 Series Integrated Services Routers (IR 800s) are **ir800** • Cisco 800 Series Access Points are **ap800** • Cisco Catalyst IR8100 Heavy Duty Series Routers are **ir8100** • Cisco 500 Series Wireless Personal Area Network (WPAN) Industrial Routers (IR 500) are **ir500** • Communications modules are **cgmesh** • The IoT FND database is **db** This identifier is obtained from the notice-of-shipment file.	cgmesh cgr1000
ip	string	Primary IP address that IoT FND uses to contact the device. Can be IPv4 or IPv6. This address is obtained when the device registers with IoT FND.	1.1.1.1
lat	decimal	Latitude of the device obtained from manual CSV import.	10
lng	decimal	Longitude of the device obtained from manual CSV import.	-11.1
alt	decimal	Altitude of the device obtained from manual CSV import.	10
mapLevel	decimal integer	Minimum map zoom level at which the device is displayed on the map. This is useful for viewing large networks. This value is obtained from manual CSV import and is an optional field.	16
geoHash	string	String hash latitude and longitude values that are used for automatic hierarchical grid-based clustering of the devices.	s1qeg9spc8n95rrw1dww7

Field Name	Type	Description	Example
lastHeard	timestamp	Time when the device was last heard from or contacted. This is used as the primary determiner of device activity level and status. This value is automatically set when the device reports to the IoT FND or the IoT FND contacts the device.	2011-05-02 00:00:00
status	enum	Current status of the device automatically set by IoT FND. Values are limited to up, down, and unheard.	up
certC	string	X.500 country name from the certificate subject, if one exists. This value is obtained from the notice-of-shipment file, as are all “cert” properties.	US
certST	string	X.500 state or province name, if one exists.	CA
certL	string	X.500 locality name, if one exists.	San Jose
certO	string	X.500 organization name, if one exists.	Cisco Systems, Inc.
certOU	string	X.500 organizational unit name, if one exists.	Operations Department
certCN	string	X.500 common name, if one exists.	App Client
certSN	string	X.500 serial number, if one exists.	12:34:A4:B9
pid	string	Product ID for the device. For routers and communication modules, this is the Cisco Secure Unique Device Identifier (SUDI) product ID from the certificate.	CGR1240/K9
vid	string	Version ID for the device. Obtained from the SUDI for routers and communication modules.	1.0
sn	string	Serial number for the device.	JAF1741ALPA

Property Field Names for Supported Routers

The following table describes the property field names available to supported Connected Grid routers. The field names are case sensitive.

Table 2: Property Field Names for Supported Routers

Field Name	Type	Description
activeLinkType	string	Physical link type over which device communicates with other devices, including IoT FND.
endUserIPv6Prefix	string	End user IPv6 address for basic mapping rule for the device.
endUserIPv6PrefixLen	string	Prefix length for the end user IPv6 address.
mapTipv6Address	string	Map-T settings IPv6 address.
mapTipv4Address	string	Map-T settings IPv4 address.

Field Name	Type	Description
mapTpsid	string	Map-T status PSID.
meshAddress	string	IP address of the mesh link automatically assigned by IoT FND during registration.
meshLocalAddress	string	Local WPAN address of the mesh link automatically assigned by IoT FND during registration.
meshPrefix	string	Subnet prefix address.
meshPrefixLength	string	Subnet prefix address length.
meshPanid	string	Subnet Private Area Network (PAN) ID.

Properties Field Names for Range Extenders

Property Field Names for Range Extenders table describes the property field names available to range extenders. The field names are case sensitive.

Table 3: Property Field Names for Range Extenders

Field Name	Type	Description
batteryState	string	Determine the current battery state.
bbuManufacturer	string	The manufacturer of the Battery Backup Unit (BBU).
bbuPid	string	The physical model name of the BBU.
bbuPresent	string	The BBU hardware is present.
bbuReady	string	The BBU is ready.
bbuSn	string	The serial number of the BBU.
bbuVid	string	The physical hardware version of the BBU.
powerSource	string	Determine if the device is running on AC or DC power.

Property Field Names for Communications Modules

The following table describes the property field names available to communications modules. The field names are case sensitive.

Table 4: Property Field Names for Communications Modules

Field Name	Type	Description
meshAddress	string	The IP address of the mesh link automatically assigned by IoT FND during registration.

Field Name	Type	Description
meshLocalAddress	string	Local WPAN address of the mesh link automatically assigned by IoT FND during registration.
meshPrefix	string	Subnet prefix address.
meshPrefixLength	string	Subnet prefix address length.
meshPanid	string	Subnet PAN ID.

Metrics Field Names

Metrics collected by IoT FND are defined per device type, and maintained in a XML file specific to each device type. IoT FND locates the XML files after it loads the boot strap image.

Note Do not use metrics defined for interfaces for searches.

Metrics for Communication Modules

The following table describes the metrics for communication modules.

Table 5: Metrics for Communication Modules

Metric Name	Unit	Min	Max	Description
uptime	sec	0	31536000	Amount of time, in seconds, that the module has been running since last boot.
meshTxSpeed	b/s	0	76800	Current speed of data transmission over the uplink network interface, measured in bits per second, averaged over a short element-specific time period.
meshTxDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the uplink interface because the outbound queue was full.
meshRxSpeed	b/s	0	76800	Rate of data received by the uplink network interface, measured in bits per second, averaged over a short element-specific time period.
meshRxReassemblyDrops	drops/sec	0	1	Rate of incoming packet fragments dropped because there was no space in the reassembly buffer.
meshHops	hops	1	8	Number of hops the element is from the root of its RPL routing tree.
meshLinkCost	–	1	3	RPL cost value for the link between the element and its uplink neighbor.
meshPathCost	–	1	24	RPL path-cost value between the element and the root of the routing tree.
meshRssi	dBm	-80	20	Measured RSSI value of the primary mesh RF uplink.
meshReverseRssi	dBm	-80	20	RSSI value measured by the mesh uplink neighbor.

Metrics for Communication Module Loopback Interface

The following table describes the metrics for communication modules loopback interface.

Table 6: Metrics for Communication Modules Loopback Interface

Metric Name	Unit	Min	Max	Description
txSpeed	b/s	0	76800	Current speed of data transmission over the interface, measured in bits per second, averaged over a short element-specific time period (for example, an hour).
txDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the interface because the outbound queue was full.
rxSpeed	b/s	0	76800	Rate of data received by the network interface, measured in bits per second, averaged over a short element-specific time period (for example, an hour).
txUnicastPackets	packets/sec	0	76800	Current packet receive rate over the interface, measured in packets per second, averaged over a short element-specific time period (for example, an hour).

Metrics for WPAN Module Interfaces

The following table describes the metrics for WPAN module interfaces.

Table 7: Metrics for the WPAN Module Interfaces

Metric Name	Unit	Min	Max	Description
txSpeed	b/s	0	76800	Current speed of data transmission over the interface, measured in bits per second, averaged over a short element-specific time period (for example, an hour).
txDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the interface because the outbound queue was full.
txSpeed	b/s	0	76800	Rate of data received by the network interface, measured in bits per second, averaged over a short element-specific time period (for example, an hour).
txUnicastPackets	packets/sec	0	76800	Current packet send rate over the interface, measured in packets per second, averaged over a short element-specific time period (for example, an hour).
rxUnicastPackets	packets/sec	0	76800	Current packet receive rate over the interface, measured in packets per second, averaged over a short element-specific time period (for example, an hour).

Metrics for PPP Interfaces

The following table describes the metrics for PPP interfaces.

Table 8: Metrics for PPP Interfaces

Metric Name	Unit	Min	Max	Description
txSpeed	b/s	0	76800	Current speed of data transmission over the interface, measured in bits per second, averaged over a short element-specific time period.

Metric Name	Unit	Min	Max	Description
txDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the interface because the outbound queue was full.
rxSpeed	b/s	0	76800	Rate of data received by the network interface, measured in bits per second, averaged over a short element-specific time period.
txUnicastPackets	packets/sec	0	76800	Current packet send rate over the interface, measured in packets per second, averaged over a short element-specific time period.
rxUnicastPackets	packets/sec	0	76800	Current packet receive rate over the interface, measured in packets per second, averaged over a short element-specific time period.

Metrics for RPL Interfaces

The following table describes the metrics for Routing Protocol for Low Power and Lossy Networks (RPL) interfaces..

Table 9: Metrics for RPL Interfaces

Metric Name	Unit	Min	Max	Description
hops	hops	1	8	Number of hops the element is from the root RPL tree.
linkCost	–	1	3	RPL cost value for the link between the element and its uplink neighbor.
pathCost	–	1	24	RPL path cost value between the element and the root of the routing tree.
rssi	dBm	-80	20	Measured RSSI value of the primary mesh RF uplink.
reverseRSSI	dBm	-80	20	RSSI value measured by the element's mesh uplink neighbor.

Metrics for Supported Connected Grid Routers

The following table describes the metrics for supported routers.

Table 10: Metrics for Supported Routers

Metric Name	Unit	Min	Max	Description
batteryLevel	%	0	100	Percentage of charge remaining in the first battery.
batteryLevel2	%	0	100	Percentage of charge remaining in the second battery.
batteryRuntime	minutes	0	65535	Runtime remaining on the first battery.
batteryRuntime2	minutes	0	65535	Runtime remaining on the second battery.
cellRSSI	dBm	-100	0	Cell Received Signal Strength Indicator (RSSI).
chassisTemp	Celsius	0	100	Internal temperature of the device.

Metric Name	Unit	Min	Max	Description
meshEndpointCount	devices	0	10000	Number of active mesh endpoints connected to this element.
meshRoutes	entries	0	1000	Number of entries that a given router has in its source-route table. This is a method to measure the number of elements in a given PAN.
meshRxReassemblyDrops	drops/sec	0	1	Rate of incoming packet fragments dropped because there was no space in the reassembly buffer.
meshRxSpeed	b/s	0	76800	Rate of data received by the uplink network interface, measured in bits per second, averaged over a short element-specific time period.
meshTxDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the uplink interface because the outbound queue was full.
meshTxSpeed	b/s	0	76800	Current speed of data transmission over the uplink network interface, measured in bits per second, averaged over a short element-specific time period.
serial0rxSpeed	b/s	0	100000000	Receive rate for the Serial 0 interface.
serial0txSpeed	b/s	0	100000000	Transmit rate for Serial 0 interface.
serial1rxSpeed	b/s	0	100000000	Receive rate for the Serial 1 interface.
serial1txSpeed	b/s	0	100000000	Transmit rate for Serial 1 interface.
uplinkRssi	dBm	-100	-50	Measured RSSI value of the primary RF uplink used for all RF uplinks.
uplinkRxDrops	drops/sec	0	1	Rate of packets received on the uplink interface, but then dropped because the inbound queue was full.
uplinkRxSpeed	b/s	0	3000000	Rate of data received by the uplink network interface, measured in bits per second, averaged over a short element-specific time period.
uplinkTxDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the uplink interface because the outbound queue was full.
uplinkTxSpeed	b/s	0	500000	Current speed of data transmission over the uplink network interface, measured in bits per second, averaged over a short element-specific time period.
uptime	sec	0	31536000	Amount of time, in seconds, that the element has been running since last boot.

Metrics for VPN Interfaces

The following table describes the metrics for the VPN interfaces.

Table 11: Metrics for VPN Interfaces

Metric Name	Unit	Min	Max	Description
txSpeed	b/s	0	76800	Current speed of data transmission over the interface, measured in bits per second, averaged over a short element-specific time period.

Metric Name	Unit	Min	Max	Description
txDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the interface because the outbound queue was full.
txSpeed	b/s	0	76800	Rate of data received by the network interface, measured in bits per second, averaged over a short element-specific time period.

Metrics for 3G Interfaces

The following table describes the metrics for 3G interfaces.

Table 12: Metrics for 3G Interfaces

Metric Name	Unit	Min	Max	Description
txSpeed	b/s	0	76800	Current speed of data transmission over the interface, measured in bits per second, averaged over a short element-specific time period.
txDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the interface because the outbound queue was full.
txSpeed	b/s	0	76800	Rate of data received by the network interface, measured in bits per second, averaged over a short element-specific time period.

Metrics for WiMax Interfaces

The following table describes the metrics for WiMAX interfaces.

Table 13: Metrics for WiMAX Module Interfaces

Metric Name	Unit	Min	Max	Description
txSpeed	b/s	0	76800	Current speed of data transmission over the interface, measured in bits per second, averaged over a short element-specific time period.
txDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the interface because the outbound queue was full.
txSpeed	b/s	0	76800	Rate of data received by the network interface, measured in bits per second, averaged over a short element-specific time period.

Metrics for WPAN Interfaces

The following table describes the metrics for the WPAN interfaces.

Table 14: Metrics for WPAN Interfaces

Metric Name	Unit	Min	Max	Description
txSpeed	b/s	0	76800	Current speed of data transmission over the interface, measured in bits per second, averaged over a short element-specific time period.
txDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the interface because the outbound queue was full.
txSpeed	b/s	0	76800	Rate of data received by the network interface, measured in bits per second, averaged over a short element-specific time period.

Metrics for Management Interfaces

The following table describes the metrics for management interfaces.

Table 15: Metrics for Management Interfaces

Metric Name	Unit	Min	Max	Description
txSpeed	b/s	0	76800	Current speed of data transmission over the interface, measured in bits per second, averaged over a short element-specific time period.
txDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the interface because the outbound queue was full.
txSpeed	b/s	0	76800	Rate of data received by the network interface, measured in bits per second, averaged over a short element-specific time period.

Metrics for Ethernet Interfaces

The following table describes the metrics for Ethernet interfaces.

Table 16: Metrics for Ethernet Interfaces

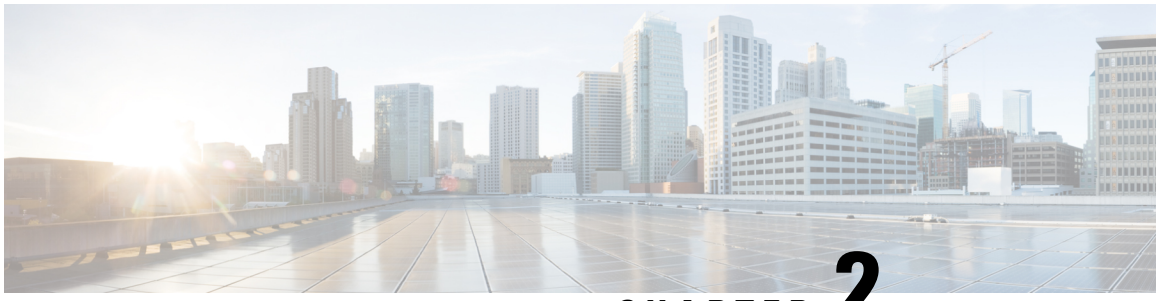
Metric Name	Unit	Min	Max	Description
txSpeed	b/s	0	76800	Current speed of data transmission over the interface, measured in bits per second, averaged over a short element-specific time period.
txDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the interface because the outbound queue was full.
txSpeed	b/s	0	76800	Rate of data received by the network interface, measured in bits per second, averaged over a short element-specific time period.

Metrics for Serial Interfaces

The following table describes the metrics for the serial interfaces.

Table 17: Metrics for Serial Interfaces

Metric Name	Unit	Min	Max	Description
txSpeed	b/s	0	76800	Current speed of data transmission over the interface, measured in bits per second, averaged over a short element-specific time period.
txDrops	drops/sec	0	1	Rate of packets dropped while trying to transmit on the interface because the outbound queue was full.
txSpeed	b/s	0	76800	Rate of data received by the network interface, measured in bits per second, averaged over a short element-specific time period.



CHAPTER 2

Audit Trail API

This chapter describes the Audit Trail API.

Using the Audit API

In your IoT FND NB API client application, use this IoT FND server URL to access the Audit Trail API WSDL:

http://<*server_address*> /nbapi/audittrail?wsdl

The time service used is Linux Epoch, see:

http://www.epochconverter.com

Audit API Method Calls

This section contains the following topics:

getAuditTrailsByTime

This call lets the client retrieve the audit trail for a specified time range.

Prototype

```
 <aud:getAuditTrailsByTime
>
  <startTimeInMs
>1329163991000</startTimeInMs>
  <endTimeInMs
>1329166091000</endTimeInMs>
 </aud:getAuditTrailsByTime>
```

Parameters

The following table describes the parameters in the interface.

Table 18: getAuditTrailsByTime Request

Parameter	Description
startTimeInMs	Audit trail entries start time for the specified time range in Epoch time format (for example, midnight on September 17, 2004 is specified as 109537920000).
endTimeInMs	Audit trail entries end time for the specified time range in Epoch time format.

getAuditTrailsByTime SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aud="http://audittrail.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <aud:getAuditTrailsByTime>
         <startTimeInMs>1329163991000</startTimeInMs>
         <endTimeInMs>1329166091000</endTimeInMs>
      </aud:getAuditTrailsByTime>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header>
      <seam:conversationId
xmlns:seam="http://www.jboss.org/seam/webservice">54</seam:conversationId>
   </env:Header>
   <env:Body>
     <ns2:getAuditTrailsByTimeResponse xmlns:ns2="http://audittrail.nbapi.cgms.cisco.com/">

         <audit_trail>
            <details>N/A</details>
            <generatedAt>1329166073474</generatedAt>
            <id>2001000</id>
            <ipAddrNum>2130706433</ipAddrNum>
            <ipAddrStr>127.0.0.1</ipAddrStr>
            <operation>Login</operation>
            <status>Success</status>
            <userName>root</userName>
         </audit_trail>
         <audit_trail>
            <details>N/A</details>
            <generatedAt>1329164689460</generatedAt>
            <id>1001000</id>
            <ipAddrNum>2130706433</ipAddrNum>
            <ipAddrStr>127.0.0.1</ipAddrStr>
            <operation>Login</operation>
            <status>Success</status>
            <userName>root</userName>
         </audit_trail>
         <audit_trail>
            <details>N/A</details>
            <generatedAt>1329164073521</generatedAt>
            <id>1002</id>
            <ipAddrNum>2130706433</ipAddrNum>
            <ipAddrStr>127.0.0.1</ipAddrStr>
            <operation>Login</operation>
```

```
            <status>Success</status>
            <userName>root</userName>
         </audit_trail>
         <audit_trail>
            <details>N/A</details>
            <generatedAt>1329164069521</generatedAt>
            <id>1001</id>
            <ipAddrNum>2130706433</ipAddrNum>
            <ipAddrStr>127.0.0.1</ipAddrStr>
            <operation>Password changed</operation>
            <status>Success</status>
            <userName>root</userName>
         </audit_trail>
         <audit_trail>
            <details>N/A</details>
            <generatedAt>1329164057605</generatedAt>
            <id>1000</id>
            <ipAddrNum>2130706433</ipAddrNum>
            <ipAddrStr>127.0.0.1</ipAddrStr>
            <operation>Login</operation>
            <status>Success</status>
            <userName>root</userName>
         </audit_trail>
      </ns2:getAuditTrailsByTimeResponse>
   </env:Body>
</env:Envelope>
```

getAuditTrailsByUser

This call retrieves audit trail entries for a specified user and time range.

Prototype

```
 <aud:getAuditTrailsByUser
>
  <userName
>endpoint_oper</userName>
  <startTimeInMs
>1329163991000</startTimeInMs>
  <endTimeInMs
>1329174551000</endTimeInMs>
 </aud:getAuditTrailsByUser>
```

Parameters

The following table describes the parameters in the interface.

Table 19: getAuditTrailsbyUser Request

Parameter	Description
userName	Audit trail entries for the specified user name.
startTimeInMs	Audit trail entries start time for the specified time range.
endTimeInMs	Audit trail entries end time for the specified time range.

getAuditTrailsByUser SOAP XML Request Format Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aud="http://audittrail.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <aud:getAuditTrailsByUser>
         <!-Optional:->
         <userName>endpoint_oper</userName>
         <startTimeInMs>1329163991000</startTimeInMs>
         <endTimeInMs>1329174551000</endTimeInMs>
      </aud:getAuditTrailsByUser>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header>
      <seam:conversationId
xmlns:seam="http://www.jboss.org/seam/webservice">85</seam:conversationId>
   </env:Header>
   <env:Body>
     <ns2:getAuditTrailsByUserResponse xmlns:ns2="http://audittrail.nbapi.cgms.cisco.com/">

         <audit_trail>
            <details>N/A</details>
            <generatedAt>1329174503078</generatedAt>
            <id>2001003</id>
            <ipAddrNum>2130706433</ipAddrNum>
            <ipAddrStr>127.0.0.1</ipAddrStr>
            <operation>Logout</operation>
            <status>Success</status>
            <userName>endpoint_oper</userName>
         </audit_trail>
         <audit_trail>
            <details>N/A</details>
            <generatedAt>1329174494253</generatedAt>
            <id>2001002</id>
            <ipAddrNum>2130706433</ipAddrNum>
            <ipAddrStr>127.0.0.1</ipAddrStr>
            <operation>Login</operation>
            <status>Success</status>
            <userName>endpoint_oper</userName>
         </audit_trail>
      </ns2:getAuditTrailsByUserResponse>
   </env:Body>
</env:Envelope>
```

getAuditTrailsByUserAndIp

This call retrieves audit trail entries for a specified user, IP address, and time range.

Prototype

```
 <aud:getAuditTrailsByUserAndIp
>
  <userName
>endpoint_oper</userName>
```

```
  <ipAddr
>127.0.0.1</ipAddr>
  <startTimeInMs
>1329163991000</startTimeInMs>
  <endTimeInMs
>1329174551000</endTimeInMs>
 </aud:getAuditTrailsByUserAndIp>
```

Parameters

The following table describes the parameters in the interface.

Table 20: getAuditTrailsByUserandIP Request

Parameter	Description
userName	Audit trail entries for the specified user name.
ipAddr	Audit trail entries for the specified IP address.
startTimeInMs	Audit trail entries start time for the specified time range.
endTimeInMs	Audit trail entries end time for the specified time range.

getAuditTrailsByUserAndIp SOAP XML Request Format Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aud="http://audittrail.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <aud:getAuditTrailsByUserAndIp>
         <userName>endpoint_oper</userName>
         <ipAddr>127.0.0.1</ipAddr>
         <startTimeInMs>1329163991000</startTimeInMs>
         <endTimeInMs>1329174551000</endTimeInMs>
      </aud:getAuditTrailsByUserAndIp>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header>
      <seam:conversationId
xmlns:seam="http://www.jboss.org/seam/webservice">86</seam:conversationId>
   </env:Header>
   <env:Body>
      <ns2:getAuditTrailsByUserAndIpResponse
xmlns:ns2="http://audittrail.nbapi.cgms.cisco.com/">
         <audit_trail>
            <details>N/A</details>
            <generatedAt>1329174503078</generatedAt>
            <id>2001003</id>
            <ipAddrNum>2130706433</ipAddrNum>
            <ipAddrStr>127.0.0.1</ipAddrStr>
            <operation>Logout</operation>
            <status>Success</status>
            <userName>endpoint_oper</userName>
```

```
            </audit_trail>
            <audit_trail>
               <details>N/A</details>
               <generatedAt>1329174494253</generatedAt>
               <id>2001002</id>
               <ipAddrNum>2130706433</ipAddrNum>
               <ipAddrStr>127.0.0.1</ipAddrStr>
               <operation>Login</operation>
               <status>Success</status>
               <userName>endpoint_oper</userName>
            </audit_trail>
         </ns2:getAuditTrailsByUserAndIpResponse>
   </env:Body>
</env:Envelope>
```

getAuditTrailsByOperation

This call retrieves audit trail entries for a specified operation type and time range.

Prototype

```
 <aud:getAuditTrailsByOperation
>
  <operation
>password changed</operation>
  <startTimeInMs
>1329163991000</startTimeInMs>
  <endTimeInMs
>1329174551000</endTimeInMs>
 </aud:getAuditTrailsByOperation>
```

Parameters

The following table describes the parameters in the interface.

Table 21: getAuditTrailsByOperation Request

Parameter	Description
operation	Audit trail entries by operation type.
startTimeInMs	Audit trail entries start time for the specified time range.
endTimeInMs	Audit trail entries end time for the specified time range.

getAuditTrailsByOperation SOAP XML Request Format

```
 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aud="http://audittrail.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <aud:getAuditTrailsByOperation>
         <operation>password changed</operation>
         <startTimeInMs>1329163991000</startTimeInMs>
         <endTimeInMs>1329174551000</endTimeInMs>
      </aud:getAuditTrailsByOperation>
```

```
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header>
      <seam:conversationId
xmlns:seam="http://www.jboss.org/seam/webservice">88</seam:conversationId>
   </env:Header>
   <env:Body>
      <ns2:getAuditTrailsByOperationResponse
xmlns:ns2="http://audittrail.nbapi.cgms.cisco.com/">
         <audit_trail>
            <details>N/A</details>
            <generatedAt>1329164069521</generatedAt>
            <id>1001</id>
            <ipAddrNum>2130706433</ipAddrNum>
            <ipAddrStr>127.0.0.1</ipAddrStr>
            <operation>Password changed</operation>
            <status>Success</status>
            <userName>root</userName>
         </audit_trail>
      </ns2:getAuditTrailsByOperationResponse>
   </env:Body>
</env:Envelope>
```

getAuditTrailsByUserAndOperation

This call retrieves audit trail entries for a specified user, operation type, and time range.

Prototype

```
 <aud:getAuditTrailsByUserAndOperation
>
  <userName
>endpoint_oper</userName>
  <operation
>login</operation>
  <startTimeInMs
>1329163991000</startTimeInMs>
  <endTimeInMs
>1329174551000</endTimeInMs>
 </aud:getAuditTrailsByUserAndOperation>
```

Parameters

The following table describes the parameters in the interface.

Table 22: getAuditTrailsByUserAndOperation Request

Parameter	Description
userName	Audit trail entries for the specified user.
operation	Audit trail entries for the specified operation type.

Parameter	Description
startTimeInMs	Audit trail entries start time for the specified time range.
endTimeInMs	Audit trail entries end time for the specified time range.

getAuditTrailsByUserAndOperation SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aud="http://audittrail.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <aud:getAuditTrailsByUserAndOperation>
         <userName>endpoint_oper</userName>
         <operation>login</operation>
         <startTimeInMs>1329163991000</startTimeInMs>
         <endTimeInMs>1329174551000</endTimeInMs>
      </aud:getAuditTrailsByUserAndOperation>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header>
      <seam:conversationId
xmlns:seam="http://www.jboss.org/seam/webservice">89</seam:conversationId>
   </env:Header>
   <env:Body>
      <ns2:getAuditTrailsByUserAndOperationResponse
xmlns:ns2="http://audittrail.nbapi.cgms.cisco.com/">
         <audit_trail>
            <details>N/A</details>
            <generatedAt>1329174494253</generatedAt>
            <id>2001002</id>
            <ipAddrNum>2130706433</ipAddrNum>
            <ipAddrStr>127.0.0.1</ipAddrStr>
            <operation>Login</operation>
            <status>Success</status>
            <userName>endpoint_oper</userName>
         </audit_trail>
      </ns2:getAuditTrailsByUserAndOperationResponse>
   </env:Body>
</env:Envelope>
```

getAuditTrailsByUserIpAndOperation

This call retrieves audit trail entries for a specified user, IP address, operation type, and time range.

Prototype

```
 <aud:getAuditTrailsByUserIpAndOperation
>
  <ipAddr
>127.0.0.1</ipAddr>
  <userName
```

```
>endpoint_oper</userName>
  <operation
>login</operation>
  <startTimeInMs
>1329163991000</startTimeInMs>
  <endTimeInMs
>1329174551000</endTimeInMs>
 </aud:getAuditTrailsByUserIpAndOperation>
```

Parameters

The following table describes the parameters in the interface.

Table 23: getAuditTrailsByUserIPAndOperation Request

Parameter	Description
ipAddr	Audit trail entries for the specified IP address.
userName	Audit trail entries for the specified user.
operation	Audit trail entries for the specified operation type.
startTimeInMs	Audit trail entries start time for the specified time range.
endTimeInMs	Audit trail entries end time for the specified time range.

getAuditTrailsByUserIpAndOperation SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aud="http://audittrail.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <aud:getAuditTrailsByUserIpAndOperation>
         <ipAddr>127.0.0.1</ipAddr>
         <userName>endpoint_oper</userName>
         <operation>login</operation>
         <startTimeInMs>1329163991000</startTimeInMs>
         <endTimeInMs>1329174551000</endTimeInMs>
      </aud:getAuditTrailsByUserIpAndOperation>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header>
      <seam:conversationId
xmlns:seam="http://www.jboss.org/seam/webservice">90</seam:conversationId>
   </env:Header>
   <env:Body>
      <ns2:getAuditTrailsByUserIpAndOperationResponse
xmlns:ns2="http://audittrail.nbapi.cgms.cisco.com/">
         <audit_trail>
            <details>N/A</details>
            <generatedAt>1329174494253</generatedAt>
            <id>2001002</id>
```

```
            <ipAddrNum>2130706433</ipAddrNum>
            <ipAddrStr>127.0.0.1</ipAddrStr>
            <operation>Login</operation>
            <status>Success</status>
            <userName>endpoint_oper</userName>
         </audit_trail>
      </ns2:getAuditTrailsByUserIpAndOperationResponse>
   </env:Body>
</env:Envelope>
```

deleteAuditTrailsByTime

This call removes audit trail entries by the specified time.

Prototype

```
 <aud:deleteAuditTrailsByTime
>
  <startTimeInMs
>1329163991000</arg0>
  <endTimeInMs
>1329174551000</arg1>
 </aud:deleteAuditTrailsByTime>
```

Parameters

The following table describes the parameters in the interface.

Table 24: deleteAuditTrailsByTime Request

Parameter	Description
startTimeInMs	Delete audit trail entries start time by specified time range.
endTimeInMs	Delete audit trail entries end time by specified time range.

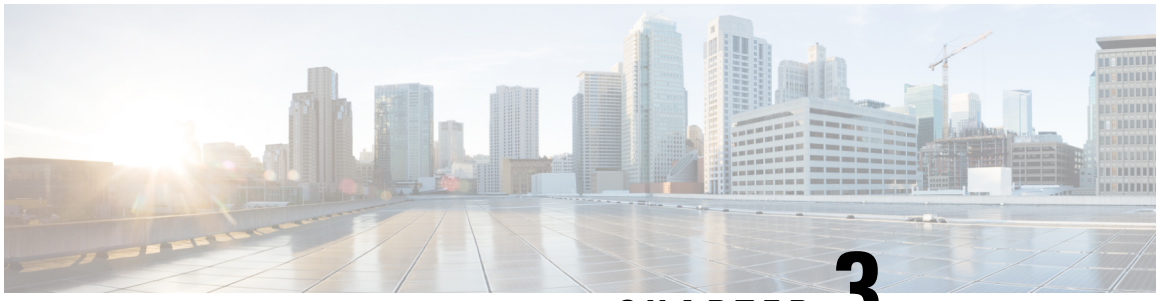
deleteAuditTrailsByTime SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aud="http://audittrail.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <aud:deleteAuditTrailsByTime>
         <startTimeInMs>1329163991000</arg0>
         <endTimeInMs>1329174551000</arg1>
      </aud:deleteAuditTrailsByTime>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header>
      <seam:conversationId
xmlns:seam="http://www.jboss.org/seam/webservice">91</seam:conversationId>
```

```
   </env:Header>
   <env:Body>
      <ns2:deleteAuditTrailsByTimeResponse
xmlns:ns2="http://audittrail.nbapi.cgms.cisco.com/">
         <delete_response>Successfully deleted audit trails.</delete_response>
      </ns2:deleteAuditTrailsByTimeResponse>
   </env:Body>
</env:Envelope>
```



CHAPTER 3

Device Management API

This chapter describes the Device Management API.

Using the Device Management API

In your IoT FND NB API client application, use this IoT FND server URL to access the Device API WSDL:

http://<*server_address*> /nbapi/device?wsdl

Using the device-client Script

The IoT FND distribution provides a Device Management API client (device-client script) in the cgms-tools-*version* .x86_64.rpm package. The client is wrapped by a shell script (/opt/cgms-tools/bin/device-client) to directly communicate with the IoT FND API.

```
[root@localhost bin]# ./device-client

usage:
    ./bin/device-client add <endpoint URL> <filename>
    ./bin/device-client remove <endpoint URL> <filename>
    ./bin/device-client set <endpoint URL> <filename> <status>
    ./bin/device-client update <endpoint URL> <filename>
    ./bin/device-client export <endpoint URL> <query>
    ./bin/device-client job <endpoint URL> <job id>
```

The following table lists device-client script commands:

Table 25: device-client Commands

Command	Description
add	Calls the addDevices API.
remove	Calls the removeDevices API.

Command	Description
set	Calls the setDevices API to change status.
update	Calls the updateDevices API to change device properties.
export	Calls the exportDevices API.
job	Calls the getJob API to get the status of the job for one of the commands above.

Except for export and job, all other commands return a job ID used to query the job execution status.

Device Management API Method Calls

This section contains the following topics:

addDevices

This call lets the client add the devices specified in a CSV file to a managed device list in the IoT FND database. It is an asynchronous call that returns a job ID used to query the status of the add device job using the getJob call (see getJob, on page 36).

Prototype

```
 <dev:addDevices
>
  <file
>
  <data
>eid:564620131674</data>
  <filename
>H:\SGBU\csv_loading\test_for_device_api_2.csv</filename>
  <username
>root</username>
  </file>
 </dev:addDevices>
```

Parameters

The following table describes the parameters in the interface.

Table 26: addDevices Parameters

Parameter	Description

file data	A structure describing the CSV file to import. This file contains the list and details of devices to add. The data type is base 64 binary.
	This example CSV file specifies new values for the mapLevel, latitude, and longitude properties of the specified devices:
	`eid,mapLevel,lat,lng`
	`CA06001400100,2,-22.229488,37.860659 CA06001400100-0,15,-22.22887230524572,37.85273990917842`
	`CA06001400100-1,15,-22.21759741145942,37.85962556788922`

Results

The following table describes the parameters in the response.

Table 27: addDevices Response

Parameter	Type	Description
jobId	int	Use the job ID to get more information about the status of this call. For more information, see getJob, on page 36.

addDevice SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:dev="http://devicemgmt.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <dev:addDevices>
         <!--Optional:-->
         <file>
            <!--Optional:-->
            <data>eid:564620131674</data>
            <!--Optional:-->
            <filename>H:\SGBU\csv_loading\test_for_device_api_2.csv</filename>
            <!--Optional:-->
            <username>root</username>
         </file>
      </dev:addDevices>
   </soapenv:Body>
</soapenv:Envelope>
```

removeDevices

This call lets the client remove the devices specified in a CSV file from a managed devices list stored in the IoT FND database. It is an asynchronous call that returns a job ID used to query the status of this job using the getJob call (see getJob, on page 36).

Prototype

```
<dev:removeDevices
```

```
>
  <file
>
  <data
>eid:136162421901</data>
  <filename
>h:\SGBU|csv_loading\remove_device_Notepad_format.csv</filename>
  <
username
>root</username>
  </file>
 </dev:removeDevices>
```

Parameters

The following table describes the parameters in the interface.

Table 28: removeDevices Parameters

Parameter	Options	Description

file	data
	filename
	username
		A structure describing the CSV file to import. This file contains the list of devices to remove. The data type is base 64 binary.
		This example CSV file specifies new values for the mapLevel, latitude, and longitude properties of the specified devices:
		
		`eid,mapLevel,lat,lng`
		`CA06001400100,2,-22.229488,37.860659 CA06001400100-0,15,-22.22887230524572,37.85273990917842`
		
		`CA06001400100-1,15,-22.21759741145942,37.85962556788922`

Results

The following table describes the parameters in the response.

Table 29: removeDevices Response

Parameter	Type	Description
jobId	int	The job ID used to get more information about the status of this call. For more information, see getJob, on page 36.

removeDevices SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:dev="http://devicemgmt.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <dev:removeDevices>
         <!--Optional:-->
         <file>
            <!--Optional:-->
            <data>eid:136162421901</data>
            <!--Optional:-->
            <filename>h:\SGBU|csv_loading\remove_device_Notepad_format.csv</filename>
```

```
            <!--Optional:-->
            <username>root</username>
         </file>
      </dev:removeDevices>
   </soapenv:Body>
</soapenv:Envelope>
```

setDevices

This call sets the properties of the specified devices as defined in a CSV file.

Parameters

The following table describes the parameters in the interface.

Table 30: setDevices Parameters

Description

A structure describing the CSV file to import. This file contains the list of devices and corresponding properties to set. The data type is base 64 binary.
This example CSV file specifies new values for the mapLevel, latitude, and longitude properties of the specified devices:
`eid,mapLevel,lat,lng`
`CA06001400100,2,-22.229488,37.860659 CA06001400100-0,15,-22.22887230524572,37.85273990917842`
`CA06001400100-1,15,-22.21759741145942,37.85962556788922`

Results

The following table describes the parameters in the response.

Table 31: setDevices Response

Parameter	Type	Description
jobId	int	The job ID used to get more information about the status of this call. For more information, see getJob, on page 36.

updateDevices

This call updates the properties of the specified devices with new values provided in CSV file.

Prototype

```
 <dev:updateDevices
>
  <file
>
  <data
>eid:568127286335</data>
  <filename
```

```
>?</filename>
  <username
>?</username>
  </file>
 </dev:updateDevices>
```

Parameters

The following table describes the parameters in the interface.

Table 32: updateDevice Parameters

Parameter	Type	Description

file	data
	filename
	username
		This structure describes the CSV file to import. This file contains the list of devices and corresponding properties to updat
		This example CSV file specifies new values for the mapLevel, latitude, and longitude properties of the specified devices:
		`eid,mapLevel,lat,lng`
		`CA06001400100,2,-22.229488,37.860659CA06001400100-0,15,-22.22887230524572,37.85273990917842CA060014001`

Results

The following table describes the parameters in the response.

Table 33: updateDevice Response

Parameter	Type	Description
jobId	int	The job ID used to get more information about the status of this call. For more information, see getJob, on page 36.

updateDevice SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:dev="http://devicemgmt.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <dev:updateDevices>
         <!--Optional:-->
         <file>
            <!--Optional:-->
            <data>eid:568127286335</data>
            <!--Optional:-->
            <filename>?</filename>
            <!--Optional:-->
            <username>?</username>
         </file>
      </dev:updateDevices>
   </soapenv:Body>
</soapenv:Envelope>
```

exportDevices

This call returns a list of devices found by the query. This is an asynchronous call. Unlike other device management APIs, this call is blocking until all results are consumed by the caller.

Prototype

```
 <dev:exportDevices
>
  <username
>root</username>
  <query
>cgr1000</query>
 </dev:exportDevices>
```

Parameters

The following table describes the parameters in the interface.

Table 34: exportDevices Parameters

Parameter	Type	Description
username	string	IoT FND username used as part of the job creation.
query	string	The query that determines the criteria of the devices to export.

Results

The following table describes the parameters in the response.

Table 35: exportDevices Response

Parameter	Type	Description
data	base64Binary	The job ID used to get more information about the status of this call. For more information, see getJob, on page 36.
date	dateTime	The date of the query (UTC time).
query	string	The query used to find the devices to export.
status	string	Status of the job.

exportDevices SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:dev="http://devicemgmt.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <dev:exportDevices>
         <!--Optional:-->
         <username>root</username>
```

```
         <!--Optional:-->
         <query>cgr1000</query>
      </dev:exportDevices>
   </soapenv:Body>
</soapenv:Envelope>
```

getJob

This call returns a job report describing the detailed execution status of the specified job.

Prototype

```
 <dev:getJob
>
         <jobId
>110</jobId>
 </dev:getJob>
```

Parameters

The following table describes the parameters in the interface.

Table 36: getJob Parameters

Parameter	Type	Description
jobId	int	The job ID to return information about.

Results

The following table describes the parameters in the response.

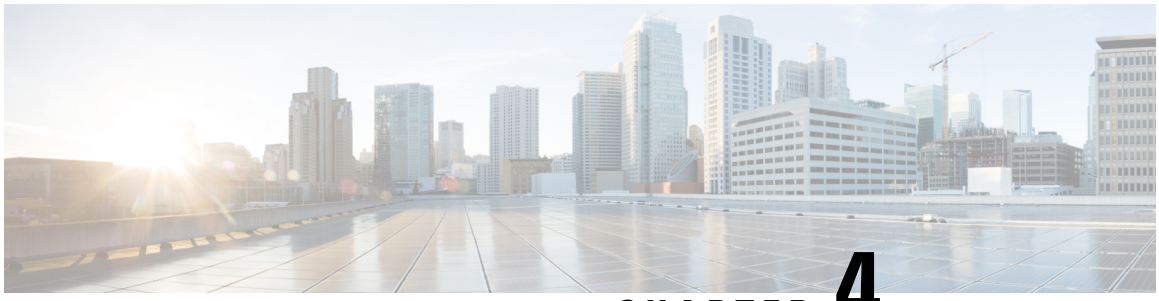
Table 37: getJob Results

Parameter	Type	Description
id	integer	Job ID.
username	string	The username of the job submitter.
filename	string	The filename of the job.
type	string	The job type.
totalCount	integer	Total count includes successes and failures.
successCount	integer	Success count represents the number of devices successfully updated.
failureCount	integer	Failure count summarizes the number of devices not successfully updated.
status	string	The job execution status.
submittedAt	date	The job submission date (UTC time).

Parameter	Type	Description
updatedAt	date	The job updating date (UTC time).
logs	string	The logged messages of the job execution.

exportJob SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:dev="http://devicemgmt.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <dev:getJob>
         <jobId>110</jobId>
      </dev:getJob>
   </soapenv:Body>
</soapenv:Envelope>
```



CHAPTER 4

Event API

This chapter describes the Event API.

Using the Event API

In your IoT FND NB API client application, use this IoT FND server URL to access the Event API WSDL:

http://<*server_address*> /nbapi/event?wsdl

For example:

http://10.27.167.19/nbapi/event?wsdl

Event API Method Calls

This section contains the following topics:

searchEvents

This call searches for events based on device type, event name, event time, and event severity. This API returns only the events of the domain that the user belongs to.

Prototype

```
<even:searchEvents>
<query>deviceType:cgmesh eventName:up</query>
<count>4</count>
<offset>0</offset>
</even:searchEvents>
```

Parameters

Table 38: searchEvents Parameters

Parameter	Type	Description
query	string	Search query string.
count	integer	Number of results to retrieve.
offset	integer	Position of the first result.

Use the parameters in Table 39: Query Parameters, on page 40 and the options listed in Table 40: eventName Query Options , on page 41 in the query.

Table 39: Query Parameters

Parameter	Delimiters	Options	Description
deviceType	=	asr1000 cgmesh cgnms cgr1000 db c800 ir800 ir500 ap800	Device type.
eventName	=	Options are listed in Table 40: eventName Query Options , on page 41 .	User-defined name.
eventSeverity	=	CRITICAL MAJOR MINOR INFO	Severity level.
eventTime	>|<|>=|<=	User defined.	UTC date and time in the format: yyyy-MM-dd HH:mm:ss:SSS.

Table 40: eventName Query Options

Option	Device	Description
aaaFailure	cgr1000 c800 ir800	The AAA server returned an error or was unreachable while attempting to authenticate a meter.
act2lFailure	cgr1000 c800 ir800	The system rebooted after a hardware ACT2 failure, and the ACT2 process is being invoked.
archiveLogModeDisabled	db	Database archive log mode is disabled. Hot backups are not permitted.
archiveLogModeEnabled	db	Database archive log mode is enabled.
batteryFailure	cgr1000 c800 ir800	The battery failed.
batteryLow	cgr1000 c800 ir800	The battery charge is below the normal range.
batteryNormal	cgr1000 c800 ir800	The battery charge is in the normal range.
bbuConfigFailure	cgr1000 c800 ir800	The battery back up (BBU) configuration failed.
bbuFirmwareDownloadFailed	cgr1000 c800 ir800	The BBU firmware download failed.
bbuFirmwareDownloadPassed	cgr1000 c800 ir800	The BBU firmware download passed.
bbuFirmwareMismatchFound	cgr1000 c800 ir800	A BBU firmware mismatch was found.

Option	Device	Description
bbuFirmwareUpgradeFailure	cgr1000 c800 ir800 ir500	The BBU firmware upgrade failed.
bbuLockOut	ir500	The BBU is locked out.
bbuPowerOff	cgr1000 c800 ir800 ir500	The battery backup unit is not powered.
cd11IfRogueApDetectedNotif	ap800	-
cd11IfStationSwitchOverNotif	ap800	-
ciscoIetfDot11QosExtChangeNotif	ap800	-
ciscoWlanVlanWepChangeNotif	ap800	-
coldBoot	cgmesh cgr1000 c800 ir800 ap800	A cold boot occurred or a mesh node registered due to cold boot.
configPushed	cgr1000 c800 ir800	(Industrial Operations Kit only) The group configuration pushed to CGRs.
configRollback	cgr1000 c800 ir800 ap800	Configuration rollback required.
criticallyLowFRASpace	db	The Flash Recovery Area (FRA) free space is critically low. Run database backup immediately or risk IoT FND and database failure.
criticallyLowTableSpace	db	Database “USERS” table space is critically low. Contact your DBA immediately to add more space or risk IoT FND and database failure.
defaultRouteLost	cgmesh	The mesh node lost the default route.

Option	Device	Description
deviceAdded	cgr1000 c800 ir800	(Industrial Operations Kit only) The EID of the new device.
deviceLocChanged	cgr1000 c800 ir800	This event occurs when the GPS location changed in relation to the configured Interval and Distance thresholds.
deviceRemoved	cgr1000 c800 ir800	(Industrial Operations Kit only) The EID of the removed device.
deviceUnknown	IoT FND	Unknown device detected by NMS.
doorClose	cgr1000 c800 ir800	The device door was closed.
doorOpen	cgr1000 c800 ir800	The device door is open.
dot11AuthenticateFail	ap800	Dot11 authentication failed for the access point
dot11Deauthenticate	ap800	Dot11 deauthentication frame was detected.
dot11Disassociate	ap800	Dot11 disassociation frame was detected.
dot1xAuthFailed	cgmesh	Dot1x authentication failed for the meter.
dot1xAuthFailure	cgmesh	A Dot1x authentication failure was detected.
dot1xAuthFlood	cgmesh	A Dot1x authentication flood was detected.
dot1xReauth	cgmesh	Multiple attempts to send the mesh key to the meter failed. Re-authentication is in progress.

Option	Device	Description
down	asr1000 cgmesh IoT FND cgr1000 c800 ir800 db ir500 ap800	The specified device is down.
hardwareInsertion	cgr1000 c800 ir800	A new piece of hardware was inserted into the chassis.
hardwareRemoval	cgr1000 c800 ir800	Hardware was removed from the chassis.
HSMdown	IoT FND	The Hardware Security Module (HSM) is down.
HSMup	IoT FND	The HSM is down.
interfaceDown	asr1000 c800 ir800	The device interface is down.
interfaceUp	asr1000 c800 ir800	The device interface is up.
linecardFailure	cgr1000 c800 ir800	Linecard failure detected.
linePowerFailure	cgr1000 c800 ir800	The line power supply for the device failed. This is different from the device being turned off, as happens in a power outage.

Option	Device	Description
linePowerRestored	cgr1000 c800 ir800	The line power supply for the device is restored.
lowBattery	cgr1000 c800 ir800 ir500	The device battery backup unit charge is low or below the specified threshold.
lowFlashSpace	cgr1000 c800 ir800	The device is nearly out of memory on the flash partition.
lowFlashSpaceOk	cgr1000 c800 ir800	The CGR available flash memory is within the specified threshold.
lowFRASpace	db	Low database CGR space detected.
lowMemory	cgr1000 c800 ir800	Available memory is below the specified threshold.
lowMemoryOk	cgr1000 c800 ir800	Available memory is above the specified threshold.
manualCloseEvent	asr1000 cgmesh cgr1000 c800 ir800	The issue state changed by admin to closed.
manualNMSAddrChange	cgmesh	The mesh node registered due to a manual NMS address change.
manualReRegistration	cgmesh	The mesh node registered due to manual registration
meshConnectivityLost	cgmesh	The mesh node lost all connectivity.
meshLinkKeyTimeout	cgmesh	The mesh node link key timed out.
meshUpgradeSuccess	cgmesh	The mesh module firmware upgrade was successful.

Option	Device	Description
meterCertChange	cgmesh	The mesh node registered due to a certificate change.
metricRetrievalFailure	asr1000 cgr1000 c800 ir800	Metric retrieval failed.
migratedToBetterPAN	cgmesh	The mesh node migrated to a better PAN.
modemTemperatureColdAlarm	cgr1000 c800 ir800	The temperature of the modem module fell below specified levels.
modemTemperatureColdAlarmRecovery	cgr1000 c800 ir800	The modem alarm reset.
modemTemperatureWarmAlarm	cgr1000 c800 ir800	The temperature of the modem module fell above specified levels.
modemTemperatureWarmAlarmRecovery	cgr1000 c800 ir800	The modem alarm reset.
nmsAddrChange	cgmesh	The mesh node registered due to an NMS address change.
nmsError	cgmesh	The mesh node registered due to an NMS error.
normalFRASpace	db	Database FRA space is normal.
normalTableSpace	db	Database table space is normal.
outage	cgmesh cgr1000 c800 ir800 ir500	An outage was detected for this device.
portDown	cgr1000 c800 ir800 ap800	Ethernet interface {0}/{1} is down.

Option	Device	Description
portFailure	cgr1000 c800 ir800 ap800	Syslog message corresponding to the port facility was generated.
portUp	cgr1000 c800 ir800 ap800	Ethernet interface {0}/{1} is up.
powerSourceNormal	cgr1000 c800 ir800	The input power source is equal to or better than when system started.
powerSourceWarning	cgr1000 c800 ir800	One or more input power source is not connected.
refreshMeshKeyFailed	asr1000 cgmesh	A refresh of the mesh key failed.
registered	asr1000 cgmesh	The event received is registered with NMS.
registrationFailure	cgr1000 c800 ir800 ap800	Device registration failed.
registrationRequest	cgr1000 c800 ir800 ap800	A registration request from a device was received.
registrationSuccess	cgr1000 c800 ir800 ap800	Device registration was successful.
rejoinedWithNewIP	cgmesh	The mesh node registered with a new IP address.

Option	Device	Description
restoration	cgmesh	The device was restored from outage.
restorationRegistration	cgmesh	The mesh node registered after an outage.
rplTreeReset	cgr1000 c800 ir800	The RPL tree version was reset to 2. Because the RPL tree is updated with data, versions increment. A value of 2 signifies a reset to its original initial state.
rplTreeSizeCritical	cgr1000 c800 ir800	More than the maximum number of mesh nodes joined the RPL tree.
rplTreeSizeCriticalClear	cgr1000 c800 ir800	Less than the maximum number of mesh nodes detected in the RPL tree.
rplTreeSizeMajor	cgr1000 c800 ir800	More than the expected mesh nodes joined the RPL tree were detected.
rplTreeSizeMajorClear	cgr1000 c800 ir800	The expected number of mesh nodes in the RPL tree were detected.
rplTreeVersionReset	cgr1000 c800 ir800	The RPL tree version was reset to 2. Because the RPL tree is updated with data, versions increment. A value of 2 signifies a reset to its original initial state.
ruleEvent	asr1000 cgmesh IoT FND cgr1000 c800 ir800 db	This is a rule-generated event.
sdcardRemovalAlarm	cgr1000	SD card removal detected, and an alarm sent.

Option	Device	Description
signatureFailure	asr1000 cgmesh cgr1000 IoT FND c800 ir800 db ir500	Invalid signature reported by mesh nodes. If this event occurs, you must verify the certificate setup and that the mesh node and IoT FND are time synchronized.
softwareCrash	cgr1000 c800 ir800	Software process failed with a stateless restart, indicating an interruption of a service. Messages are processed for crashes on supervisor modules and line cards.
systemSwInconsistent	cgr1000 c800 ir800	Inconsistency detected in software or file system.
temperatureMajorAlarm	cgr1000 c800 ir800	A thermal sensor indicates that the temperature has reached the operating major threshold.
temperatureMajorRecovery	cgr1000 c800 ir800	A major temperature alarm has recovered.
temperatureMinor	cgr1000 c800 ir800	The device temperature reached the minor threshold.
temperatureMinorRecovery	cgr1000 c800 ir800	The minor temperature alarm has recovered.
timeMismatch	cgmesh	The NMS server time does not match the device local time.
timeMismatchResolved	cgmesh	The NMS server time matches the device local time.
tunnelDown	cgr1000 c800 ir800	The tunnel is down.

Option	Device	Description
tunnelProvFailure	cgr1000 c800 ir800	Tunnel provisioning failed.
tunnelProvRequest	cgr1000 c800 ir800	A tunnel provisioning request was received from a device.
tunnelProvSuccess	cgr1000 c800 ir800	Tunnel provisioning was successful.
tunnelUp	cgr1000 c800 ir800	The tunnel is up.
unknown	asr1000 cgmesh IoT FND cgr1000 c800 ir800 ir500 ap800 db	The event received is not registered with NMS.
unknownRegReason	cgmesh	The mesh node registered for an unknown reason.
unknownWPANChange	cgmesh	The mesh node changed its WPAN for an unknown reason.
unsupported	cgr1000 c800 ir800 ap800	Unsupported device detected.

Option	Device	Description
up	asr1000 cgmesh IoT FND cgr1000 c800 ir800 ap800 db	The specified device is up.
veryLowFRASpace	db	Very low database FRA space detected.
wpanWatchdogReload	cgr1000 c800 ir800	The WatchDog reloaded the WPAN module. The bridge was unresponsive for more than 5 minutes and the WatchDog is enabled.

Results

Table 41: searchEvents Results

Field	Type	Description
subscriptionid	long	Subscription ID used by the listener to identify the subscription response origin.
events	List<EventDetail>	Details about the event.

searchEvents SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://event.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <even:searchEvents>
         <!--Optional:-->
         <query>deviceType:cgmesh eventName:up</query>
         <!--Optional:-->
         <count>4</count>
         <!--Optional:-->
         <offset>0</offset>
      </even:searchEvents>
   </soapenv:Body>
</soapenv:Envelope>
```

subscribeForEvents

This call streams a set of events to the API listener based on the query. Event subscriptions are based on device type, event name, or severity. Listener registers the URL and specifies the push window. After every configured eventPushWindowSec event push window, all new events that are received for the subscribed domain in this window are delivered to the registered URL. Subscription-based events notification uses the same query language as searchEvents, on page 39, except that the eventTime attribute-based queries cannot be subscribed to and return unsuccessful subscription errors.

Note You can ONLY subscribe or unsubscribe to events of the domain that you belong to. Only root users have the privilege to subscribe or unsubscribe to events of other domains.

Prototype

```
<even:subscribeForEvents>
<soapEndPointUrl>http://customer.network.com:11001/Process/Service/ProcessCellRouterStates/ReceiveEvents/EventPushService?wsdl</soapEndPointUrl>

<query>deviceType:cgmesh eventName:registered</query>
<eventPushWindowSec>21</eventPushWindowSec>
<domain>root</domain>
</even:subscribeForEvents>
```

Parameters

Table 42: subscribeForEvents Parameters

Parameter	Type	Description

soapEndPointUrl	string	The address of the WSDL file–as implemented by your client–that receives event notifications from the IoT FND NB API. For more information, see Handling Event Notifications On the Client Side, on page 59.
		http://<*server_address*> :<*port number*> /<*path*><*api*> ?wsdl
		For example:
		http://localhost:8445/event?wsdl
		soapEndPointUrl must point to the WSDL document that describes the listener application receiving the events. Ensure that the *target namespace* and *service name* parameters in the Web Services Description Language (WSDL) document conform to these default values:
		`target namespace "http://pushevent.nbapi.cgms.cisco.com/" service name "EventPushService"`
		Note Release 2.1 and later installations support https for soapEndPointUrl.
		Functionality in IoT FND 1.1.3 and later installations
		Some applications cannot set the default values for these parameters when generating the WSDL file. If this is the case, in IoT FND 1.1.3 and later installations you can set the following properties in the server/cgms/conf/cgms.properties file to match the values in the generated WSDL document:
		`eventSubscriberNamespace ="http://event.nbapi.cgms.mydomain.com/" eventSubscriberServicename ="MyEventService"`
		The *target namespace* and *service name* parameters must match those specified in the cgms.properties file. If they are not specified in the cgms.properties file, they must match the default values.
		Note Only one set of these values is allowed in the cgms.properties file. If there are multiple subscribers, they must use the *target namespace* and *service name* parameter values specified in all WSDL documents pointed to in the **soapEndPointUrl** of the subscriptions.

query	string	The query string. **Note** The query language eventTime field cannot be inside the subscription.
eventPushWindowSec	integer	The event push window time, in seconds. The query executes after x seconds, and the results are pushed to the listener endpoint specified in the **soapEndPointUrl** WSDL file.
Domain	string	Specify the domain name to subscribe to events.

Results

Table 43: subscribeforEvents Response

Field	Type	Description
subscriptionId	long	Subscription ID

The QueryResult Status field indicates if the subscription succeeded or failed.

subscribeForEvents SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://event.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <even:subscribeForEvents>
         <!--Optional:-->
         <soapEndPointUrl>http://localhost</soapEndPointUrl>
         <!--Optional:-->
         <query>deviceType:cgmesh eventName:registered</query>
        <!--Optional:-->
         <eventPushWindowSec>60</eventPushWindowSec>
         <!--Mandatory field for new subscriptions from 4.9.0-->
         <!--To update subscriptions prior to 4.9.0, this field should be empty-->
         <domain>root</domain>
      </even:subscribeForEvents>
   </soapenv:Body>
</soapenv:Envelope>
```

unSubscribeForEvents

This call unsubscribes the defined listener event query.

Prototype

```
<even:unSubscribeForEvents>
<query>deviceType:cgmesh eventName:registered</query>
<domain>root</domain>
</even:unSubscribeForEvents>
```

Parameters

Table 44: unSubscribeForEvents Parameters

Parameter	Type	Description
soapEndPointUrl	string	Address where the EventNbapiService WSDL is located. http://<*server_address*> :<*port number*> /<*path*><*api*> ?wsdl
query	string	Query string. **Note** Query language eventTime field cannot be inside the subscription.

Parameter	Type	Description
Domain	string	Specify the domain name to unsubscribe to events.

Results

Table 45: unSubscribeForEvents Results

Field	Type	Description
subscriptionId	long	Subscription ID

The QueryResult Status field indicates if the subscription succeeded or failed.

unSubscribeForEvents SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://event.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <even:unSubscribeForEvents>
         <!--Optional:-->
         <soapEndPointUrl>http://localhost</soapEndPointUrl>
         <!--Optional:-->
         <query>deviceType:cgmesh eventName:registered</query>
         <!--Mandatory field for new subscriptions from 4.9.0-->
         <!--To update subscriptions prior to 4.9.0, this field should be empty-->
         <domain>root</domain>
      </even:unSubscribeForEvents>
   </soapenv:Body>
</soapenv:Envelope>
```

subscribeForCgmeshOutage

This call is similar to subscribeForEvents, except that it is for outage and restoration events. Up to 10 subscribers (listeners) at a time can register for these events.

Prototype

```
<even:subscribeForCgmeshOutage>
<soapEndPointUrl>http://128.107.109.98:8456/nbapi/pushevent?wsdl</soapEndPointUrl>
<domain>root</domain>
</even:subscribeForCgmeshOutage>
```

Parameters

Table 46: subscribeForCgmeshOutage Parameters

Parameter	Type	Description
soapEndPointUrl	string	Address of the WSDL, implemented by your client that receives outage notifications from the IoT FND NB API. For more information, see Handling Event Notifications On the Client Side, on page 59. http://<*server_address*> :<*port number*> /<*path*><*api*> ?wsdl For example: http://localhost:8445/outage?wsdl **Note** Release 2.1 and later installations support https for soapEndPointUrl.
Domain	string	Specify the domain name to subscribe to cgmesh outage events.

To configure the amount of time, in seconds, after which IoT FND pushes batches of outage events and Restoration Events to all subscribers, set the value of the event-Outage-push-sec parameter in the /opt/cgms/conf/cgms.properties file. For example, to set event-Outage-push-sec to 30, add this line to the file:

```
event-Outage-push-sec=30
```

When IoT FND pushes outage events to subscribers, only subscribers that are up receive the events. The subscribers that are down (they do not respond) do not receive these events even after they come back online, but they receive the next outage event push.

For a very fast outage event push, set event-Outage-push-sec to a value as low as one second. If you set the push value to one second, IoT FND executes a job to find and push new events in the queue.

Results

Table 47: Subscribe for CG-Mesh Outage Response

Parameter	Type	Description
subscriptionId	long	Subscription ID which can be used by the listener to identify for which subscription they are getting the response from.

The QueryResult Status field indicates if the subscription succeeded or failed.

subscribeForCgmeshOutage SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://event.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <even:subscribeForCgmeshOutage>
         <!--Optional:-->
         <soapEndPointUrl>http:/localhost</soapEndPointUrl>
         <!--Mandatory field for new subscriptions from 4.9.0 -->
         <!-- To update subscriptions prior to 4.9.0, this field should be empty-->
```

```
        <domain>root</domain>
      </even:subscribeForCgmeshOutage>
   </soapenv:Body>
</soapenv:Envelope>
```

unSubscribeForCgmeshOutage

This call unsubscribes the defined listener.

Prototype

```
<even:unSubscribeForCgmeshOutage>
<soapEndPointUrl>http://128.107.109.98:8456/nbapi/pushevent?wsdl</soapEndPointUrl>
<domain>root</domain>
</even:unSubscribeForCgmeshOutage>
```

Parameters

The following table describes the parameters in the request.

Table 48: unSubscribeForCgmeshOutage Parameters

Parameter	Type	Description
soapEndPointUrl	string	Address of the Event WSDL service. The soapEndPointUrl identifies the subscription. http://<*server_address*> :<*port number*> /<*path*><*api*> ?wsdl **Note** Release 2.1 and later installations support HTTPS for soapEndPointUrl.
Domain	string	Specify the domain name to unsubscribe to cgmesh outage events.

Results

Table 49: unSubscribeForCgmeshOutage Results

Parameter	Type	Description
subscriptionId	long	Subscription ID

The QueryResult Status field indicates if the subscription succeeded or failed.

unSubscribeForCgmeshOutage SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://event.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <even:unSubscribeForCgmeshOutage>
         <!--Optional:-->
         <soapEndPointUrl>http:/localhost</soapEndPointUrl>
         <!--Mandatory field for new subscriptions from 4.9.0-->
         <!--To update subscriptions prior to 4.9.0, this field should be empty-->
         <domain>root</domain>
```

```
      </even:unSubscribeForCgmeshOutage>
   </soapenv:Body>
</soapenv:Envelope>
```

getAllEventSubscriptions

This API gets all the events or outage events for which you have subscribed in IoT FND.

Based on the value defined in the `outageSubscription` field, the API provides information on either all events or only outage events:

- Set the value to `"true"` to get information on all the outage events.
- Set the value to `"false"` to get information on all the events.

Prototype

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:even="http://event.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <even:getAllEventSubscriptions>
         <!--Optional:-->
         <outageSubscription>false</outageSubscription>
          </even:getAllEventSubscriptions>
   </soapenv:Body>
</soapenv:Envelope>
```

Parameters

Table 50: getAllEventSubscriptions Parameters

Parameter	Type	Description
outageSubscription	boolean	• Set the value to "true" to get the outage events. • Set the value to false to get all events.
Domain	string	Specify the domain name to subscribe to all events or only outage events.

getAllEventSubscriptions SOAP XML Request Format

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"/>
   <soap:Body>
     <ns2:getAllEventSubscriptionsResponse xmlns:ns2="http://event.nbapi.cgms.cisco.com/">

         <eventSubscriptionListResult>
            <queryStatus>SUCCEEDED</queryStatus>
            <eventSubscriptions>
               <eventPushWindowSec>60</eventPushWindowSec>
               <id>1000</id>
               <isOutageSubscription>false</isOutageSubscription>
               <lastPushTime>0</lastPushTime>
               <queryString>deviceType:ir800</queryString>
               <soapEndPointUrl>http:/localhost</soapEndPointUrl>
               <domain>root</domain>
```

```
            </eventSubscriptions>
            <numberOfRecords>1</numberOfRecords>
            <showOutageSubscriptionOnly>false</showOutageSubscriptionOnly>
         </eventSubscriptionListResult>
      </ns2:getAllEventSubscriptionsResponse>
   </soap:Body>
</soap:Envelope>
```

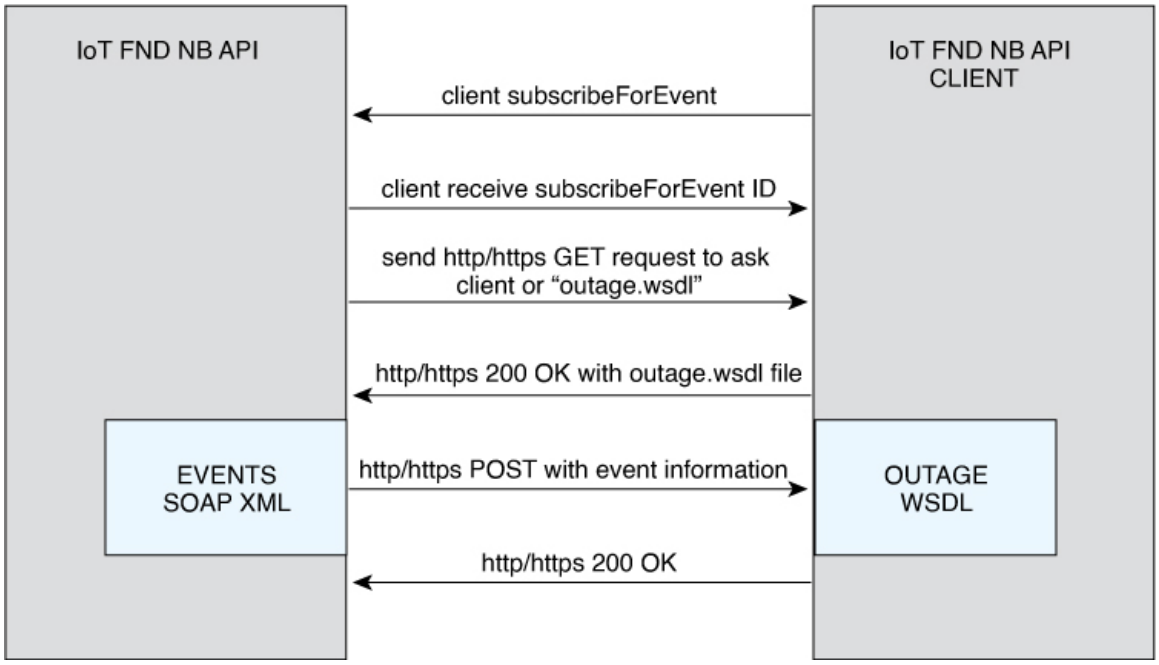
Handling Event Notifications On the Client Side

When subscribing for an event type, your IoT FND NB API client must implement a Web Service that implements the WSDL for handling event notifications sent by the IoT FND NB API. The WSDL you must provide the receiveEvents() method, which the IoT FND NB API uses to send event notifications to your client.

Example

In the following figure, the IoT FND NB API client implements the Outage WSDL. When the client subscribes for outage events, IoT FND uses the IoT FND NB API to call the method receiveEvents() on the IoT FND NB API client.

Figure 2: Event Notification Handling



This is the Event XML that your client must implement to receive outage notifications from IoT FND.

Events and Outage Notification Handling WSDL (Client Side)

The clients must implement the WSDL file to receive notification on events and power outages. The WSDL file varies for different releases of IoT FND.

- For IoT FND releases prior to 4.9.x, the WSDL file is same for both events and outage notifications.
- For IoT FND release 4.9.x to 4.10.x, the WSDL file is different for events and outage notifications.
  - Events
  - Outages

WSDL File for Event and Outage Notification

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:tns="http://pushevent.nbapi.cgms.cisco.com/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="eventPush"
 targetNamespace="http://pushevent.nbapi.cgms.cisco.com/">
 <wsdl:types>
  <xsd:schema targetNamespace="http://pushevent.nbapi.cgms.cisco.com/">
   <xsd:element name="receiveEvents" type="tns:receiveEvents" />
   <xsd:complexType name="receiveEvents">
    <xsd:sequence>
     <xsd:element minOccurs="0" name="eventQueryResult"
      type="tns:eventQueryResult" />
    </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="eventQueryResult">
    <xsd:complexContent>
     <xsd:extension base="tns:queryResult">
      <xsd:sequence>
       <xsd:element maxOccurs="unbounded" minOccurs="0"
        name="events" nillable="true" type="tns:eventDetail" />
       <xsd:element name="subscriptionId" type="xsd:long" />
      </xsd:sequence>
     </xsd:extension>
    </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType abstract="true" name="queryResult">
    <xsd:sequence>
     <xsd:element minOccurs="0" name="queryId" type="xsd:string" />
     <xsd:element minOccurs="0" name="queryStatus" type="xsd:string" />
    </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="eventDetail">
    <xsd:sequence>
     <xsd:element minOccurs="0" name="eid" type="xsd:string" />
     <xsd:element minOccurs="0" name="eventMessage" type="xsd:string" />
     <xsd:element minOccurs="0" name="eventSeverity" type="xsd:string" />
     <xsd:element minOccurs="0" name="eventTime" type="xsd:long" />
     <xsd:element minOccurs="0" name="eventTypeName" type="xsd:string" />
     <xsd:element minOccurs="0" name="meterId" type="xsd:string" />
    </xsd:sequence>
   </xsd:complexType>
  </xsd:schema>
 </wsdl:types>
 <wsdl:message name="receiveEvents">
  <wsdl:part element="tns:receiveEvents" name="receiveEvents" />
 </wsdl:message>
 <wsdl:portType name="EventPushService">
  <wsdl:operation name="receiveEvents">
   <wsdl:input message="tns:receiveEvents" />
  </wsdl:operation>
 </wsdl:portType>
```

```
 <wsdl:binding name="EventPushServiceBinding" type="tns:EventPushService">
  <soap:binding style="document"
   transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="receiveEvents">
   <soap:operation soapAction="http://pushevent.nbapi.cgms.cisco.com/receiveEvents" />
   <wsdl:input>
    <soap:body use="literal" />
   </wsdl:input>
  </wsdl:operation>
 </wsdl:binding>
 <wsdl:service name="EventPushService">
  <wsdl:port binding="tns:EventPushServiceBinding" name="EventPushService">
   <soap:address location="http://127.0.0.1:8008" />
  </wsdl:port>
 </wsdl:service>
</wsdl:definitions>
```

WSDL File for Outage Notification

Note From IoT FND release 4.9.x to 4.10.x, the domain is specified in the request.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:tns="http://pushevent.nbapi.cgms.cisco.com/"
 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="eventPush"
targetNamespace="http://pushevent.nbapi.cgms.cisco.com/">
 <wsdl:types>
  <xsd:schema targetNamespace="http://pushevent.nbapi.cgms.cisco.com/">
   <xsd:element name="receiveEvents" type="tns:receiveEvents" />
   <xsd:complexType name="receiveEvents">
    <xsd:sequence>
     <xsd:element minOccurs="0" name="eventQueryResult" type="tns:eventQueryResult" />
    </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="eventQueryResult">
    <xsd:complexContent>
     <xsd:extension base="tns:queryResult">
      <xsd:sequence>
       <xsd:element maxOccurs="unbounded" minOccurs="0" name="events" nillable="true"
type="tns:eventDetail" />
       <xsd:element name="subscriptionId" type="xsd:long" />
      </xsd:sequence>
     </xsd:extension>
    </xsd:complexContent>
   </xsd:complexType>
   <xsd:complexType abstract="true" name="queryResult">
    <xsd:sequence>
     <xsd:element minOccurs="0" name="queryId" type="xsd:string" />
     <xsd:element minOccurs="0" name="queryStatus" type="xsd:string" />
    </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="eventDetail">
    <xsd:sequence>
     <xsd:element minOccurs="0" name="domain" type="tns:domain"/>
     <xsd:element minOccurs="0" name="eid" type="xsd:string" />
     <xsd:element minOccurs="0" name="eventMessage" type="xsd:string" />
     <xsd:element minOccurs="0" name="eventSeverity" type="xsd:string" />
     <xsd:element minOccurs="0" name="eventTime" type="xsd:long" />
     <xsd:element minOccurs="0" name="eventTypeName" type="xsd:string" />
```

```
      <xsd:element minOccurs="0" name="meterId" type="xsd:string" />
     </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="domain">
     <xsd:sequence>
      <xsd:element minOccurs="0" name="id" type="xsd:long"/>
      <xsd:element minOccurs="0" name="name" type="xsd:string"/>
     </xsd:sequence>
    </xsd:complexType>
   </xsd:schema>
  </wsdl:types>
  <wsdl:message name="receiveEvents">
   <wsdl:part element="tns:receiveEvents" name="receiveEvents" />
  </wsdl:message>
  <wsdl:portType name="EventPushService">
   <wsdl:operation name="receiveEvents">
    <wsdl:input message="tns:receiveEvents" />
   </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="EventPushServiceBinding" type="tns:EventPushService">
   <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
   <wsdl:operation name="receiveEvents">
    <soap:operation soapAction="http://pushevent.nbapi.cgms.cisco.com/receiveEvents" />
    <wsdl:input>
     <soap:body use="literal" />
    </wsdl:input>
   </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="EventPushService">
   <wsdl:port binding="tns:EventPushServiceBinding" name="EventPushService">
    <soap:address location="http://10.88.202.38:33005/FNDIncomingEvent" />
   </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Push Mechanisms

Push mechanisms work only when the NMS server has successfully completed the subscription, as defined in the subscribeForEvents, on page 52 and subscribeForCgmeshOutage, on page 55 API methods.

A successful subscription leads to generation of the Subscription ID that is sent to the subscriber. The subscriber uses the Subscription ID to track the event push.

IoT FND runs a Scheduled Job every x seconds. Seconds are configurable during the subscription by using the event-Outage-push-sec global parameter defined in the /opt/cgms/conf/cgms.properties file. After every x seconds, IoT FND generates an EventList and pushes it to the subscribers defined in the soapEndPointUrl.

The web service to implement on the NMS side is:

```
public void receiveEvents(EventQueryResult eventQueryResult) throws java.rmi.RemoteException;
```

QueryResult

The QueryResult Status field indicates if the subscription succeeded or failed. The following table describes the parameters in the response.

Table 51: EventQueryResult Response

Field	Type	Description
subscriptionId	long	Subscription ID used by the listener to identify which subscription the response is from.
events	List<*EventDetail*>	Details about the event.

The following table describes the parameters in the EventDetail results.

Table 52: EventDetail Response

Field	Type	Description
eid	string	Serial number for the CGR and MAC address for the mesh endpoint.
eventMessage	string	Message related to the event.
eventTime	long	Time in milliseconds.
eventTypeName	string	Type of event
meterId	string	Meter ID corresponding to the EID. For this value to return, it must be included in the import file when importing meters in IoT FND.

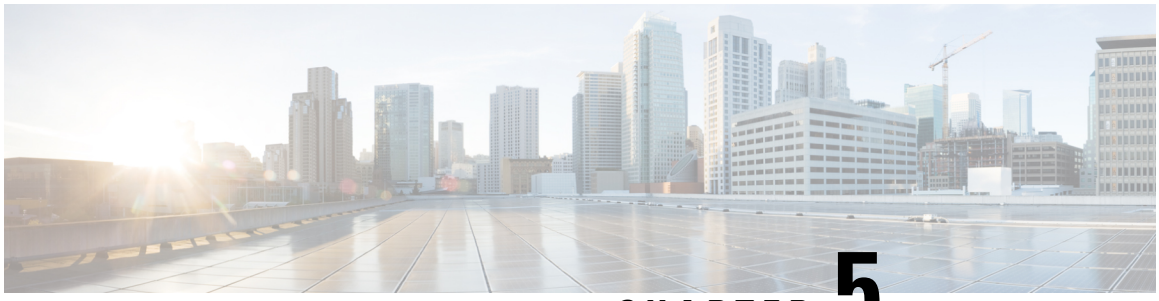
Example

This is an example of the XML content that the subscriber receives:

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header>
      <seam:conversationId
xmlns:seam="http://www.jboss.org/seam/webservice">110</seam:conversationId>
   </env:Header>
   <env:Body>
      <ns2:searchEventsResponse xmlns:ns2="http://event.nbapi.cgms.cisco.com/">
         <eventQueryResult>
            <queryId></queryId>
            <queryStatus>SUCCEEDED</queryStatus>
            <events>
               <eid>NE01</eid>
               <eventMessage>Device is Up</eventMessage>
               <eventSeverity>INFO</eventSeverity>
               <eventTime>1314656731899</eventTime>
               <eventTypeName>up</eventTypeName>
               <meterId>Sjc123</meterId>
            </events>
            <events>
               <eid>NE01</eid>
               <eventMessage>Outage detected on this device</eventMessage>
               <eventSeverity>CRITICAL</eventSeverity>
               <eventTime>1314656731908</eventTime>
               <eventTypeName>outage</eventTypeName>
               <meterId>Sjc123</meterId>
            </events>
            <events>
```

```
                <eid>NE01</eid>
                <eventMessage>Device has been Restored from Outage</eventMessage>
                <eventSeverity>INFO</eventSeverity>
                <eventTime>1314656771923</eventTime>
                <eventTypeName>restoration</eventTypeName>
                <meterId>Sjc123</meterId>
             </events>
             <events>
                <eid>NE01</eid>
                <eventMessage>Device is Up</eventMessage>
                <eventSeverity>INFO</eventSeverity>
                <eventTime>1314656771933</eventTime>
                <eventTypeName>up</eventTypeName>
                <meterId>Sjc123</meterId>
             </events>
             <subscriptionId>2</subscriptionId>
          </eventQueryResult>
       </ns2:searchEventsResponse>
   </env:Body>
</env:Envelope>
```

The subscriptionId XML element (<subscriptionId>2</subscriptionId>) tells the receiver that this push is for the subscription ID equal to 2.



CHAPTER 5

Firmware Upgrade API

Note This API only applies to mesh devices.

Note Firmware upgrades are only supported on Cisco IOS.

This chapter describes the Firmware Upgrade API.

Using the Firmware Upgrade API

In your IoT FND NB API client application, use this IoT FND server URL to access the Firmware Upgrade API WSDL:

http://<*server_address*> /nbapi/meshDeviceOps?wsdl

Firmware Upgrade API Method Calls

The following are the Firmware Upgrade API method calls:

The Firmware Upgrade API module allows client applications to start and stop firmware uploads to a group of mesh devices, check their firmware upload status, obtain firmware information by firmware group, setup a backup firmware image, and schedule a firmware reload.

Specified input parameters are not case sensitive, and are validated before further processing by IoT FND. All input parameters are character string. The utcDateAndTime format is “YYYY-MM-DD*HH:MM:SS.” The following table lists return codes sent by the Firmware Upgrade module.

Table 53: Firmware Upgrade API Return Codes

Code	Definition
0	Success.
–1	Unsupported device type.
–2	Firmware group not found.
–3	Firmware image not found.
–4	Invalid date and time format (scheduleReload, on page 71).
–5	Invalid firmware image.
–6	Cannot perform operation; firmware group is active.
–7	Cannot find a device in the up state in the firmware group.
–8	Cannot perform operation; empty firmware group.
–9	Cannot stop firmware update on inactive firmware group.
–10	Invalid load date specified.
–11	Simultaneous firmware group operations over maximum threshold.
–12	Cannot perform operation; firmware group file image only present on the nodes.
–99	Indicates a lower-layers system error, including: • System busy • Invalid user permissions (Role Based Access Control) • Another operation is pending for the group A text message returns with the return code.

startUpload

This call initiates the firmware upgrade on the specified firmware group with the specified firmware image.

Prototype

```
startUpload (devType, firmwareGroup, firmwareImageName)
```

Parameters

The following table describes the parameters in the interface.

Table 54: startUpload Parameters

Parameter	Options	Description
devType	cgmesh	The device to receive the firmware upgrade.
firmwareGroup	string	The firmware group of devices to receive the firmware upgrade.
firmwareImageName	string	The firmware image name.

Results

The following table describes the parameters in the response.

Table 55: startUpload Results

Value	Description
0	Success.
-1	devType not found or is not supported.
-2	firmwareGroup not found or is invalid.
-3	firmwareImageName not found or is invalid.
-99	System error.

SOAP XML Request Format

```
startUpload
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mes="http://meshDeviceOps.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:startUpload>
         <firmwareGroup>?</firmwareGroup>
         <firmwareImageName>?</firmwareImageName>
      </mes:startUpload>
   </soapenv:Body>
</soapenv:Envelope>
```

stopUpload

This call stops the firmware upgrade in progress on the specified firmware group.

Prototype

```
stopUpload (devType, firmwareGroup)
```

Parameters

The following table describes the parameters in the interface.

Table 56: stopUpload Parameters

Parameter	Options	Description
devType	cgmesh	The device to receive the firmware upgrade.
firmwareGroup	string	The firmware group of devices to receive the firmware upgrade.

Results

The following table describes the parameters in the response.

Table 57: stopUpload Results

Value	Description
0	Success.
-1	devType not found or is not supported.
-2	firmwareGroup not found or is invalid.
-99	System error.

SOAP XML Request Format

```
stopUpload
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mes="http://meshDeviceOps.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:startUpload>
         <firmwareGroup>?</firmwareGroup>
         <firmwareImageName>?</firmwareImageName>
      </mes:startUpload>
   </soapenv:Body>
</soapenv:Envelope>
```

getFirmwareUploadStatus

This call returns a list of value pairs indicating the number of devices in the specified firmware group with a partial firmware upload.

Prototype

```
getFirmwareUploadStatus (devType, firmwareGroup)
```

Parameters

The following table describes the parameters in the interface.

Table 58: getFirmwareUploadStatus Parameters

Parameter	Options	Description
devType	cgmesh	The device to receive the firmware upgrade.
firmwareGroup	string	The firmware group of devices to receive the firmware upgrade.

Results

The following table describes the parameters in the response.

Table 59: getFirmwareUploadStatus Results

Value	Description
0	Success.
-1	devType not found or is not supported.
-2	firmwareGroup not found or is invalid.
-99	System error.

SOAP XML Request Format

```
getFirmwareUploadStatus
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mes="http://meshDeviceOps.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:getFirmwareUploadStatus>
         <firmwareGroup>?</firmwareGroup>
      </mes:getFirmwareUploadStatus>
   </soapenv:Body>
</soapenv:Envelope>
```

getFirmwareImageInfoList

This call returns a list with the name, version and number of devices that are running, uploaded and have a backup with this firmware version. The action status and scheduled reload date and time also return.

Prototype

```
getFirmwareImageInfoList (devType, firmwareGroup)
```

Parameters

The following table describes the parameters in the interface.

Table 60: getFirmwareImageListInfo Parameters

Parameter	Options	Description
devType	cgmesh	The device to receive the firmware upgrade.
firmwareGroup	string	The firmware group of devices to receive the firmware upgrade.

Results

The following table describes the parameters in the response.

Table 61: getFirmwareImageListInfo Results

Value	Description
-1	devType not found or is not supported.
-2	firmwareGroup not found or is invalid.
-99	System error.

SOAP XML Request Format

```
setBackupFirmwareImage
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mes="http://meshDeviceOps.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:setBackupFirmwareImage>
         <firmwareGroup>?</firmwareGroup>
         <firmwareImageName>?</firmwareImageName>
      </mes:setBackupFirmwareImage>
   </soapenv:Body>
</soapenv:Envelope>
```

setBackupFirmwareImage

This call sets the specified firmware image as the backup for the specified firmware group.

Prototype

```
setBackupFirmwareImage (devType, firmwareGroup, firmwareImageName)
```

Parameters

The following table describes the parameters in the interface.

Table 62: setBackupFirmwareImage Parameters

Parameter	Options	Description
devType	cgmesh	The device to receive the firmware upgrade.
firmwareGroup	string	The firmware group of devices to receive the firmware upgrade.
firmwareImageName	string	The firmware image name.

Results

The following table describes the parameters in the response.

Table 63: setBackupFirmwareImage Results

Value	Description
0	Success.
-1	devType not found or is not supported.
-2	firmwareGroup not found or is invalid.
-3	firmwareImageName not found or is invalid.
-99	System error.

SOAP XML Request Format

```
setBackupFirmwareImage
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mes="http://meshDeviceOps.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:setBackupFirmwareImage>
         <firmwareGroup>?</firmwareGroup>
         <firmwareImageName>?</firmwareImageName>
      </mes:setBackupFirmwareImage>
   </soapenv:Body>
</soapenv:Envelope>
```

scheduleReload

This call sets the date and time for a firmware upgrade of the specified firmware image and group.

Prototype

```
scheduleReload (devType, firmwareGroup, firmwareImageName, utcDateAndTime)
```

Parameters

The following table describes the parameters in the interface.

Table 64: scheduleReload Parameters

Parameter	Options	Description
devType	cgmesh	The device to receive the firmware upgrade.
firmwareGroup	string	The firmware group of devices to receive the firmware upgrade.
firmwareImageName	string	The firmware image name.
reloadGmtTime	string	The format is “yyyy-MM-ddTHH:mm:ss.”
utcDateAndTime	string	The format is “YYYY-MM-DD*HH:MM:SS.”

Results

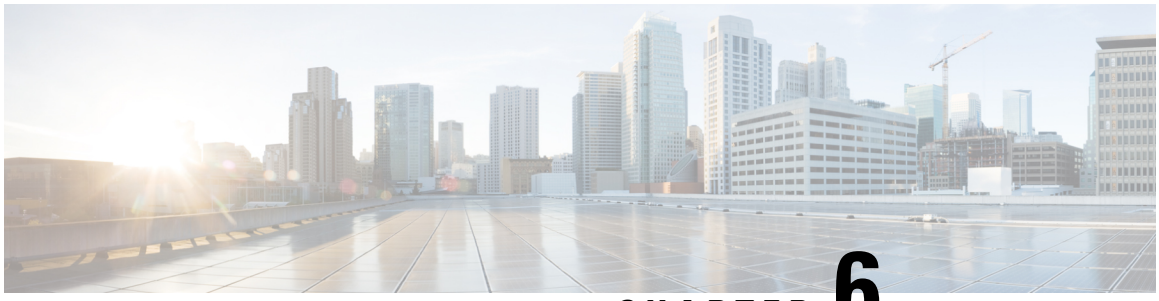
The following table describes the parameters in the response.

Table 65: scheduleReload Results

Value	Description
0	Success.
-1	devType not found or is not supported.
-2	firmwareGroup not found or is invalid.
-3	firmwareImageName not found or is invalid.
-4	utcDateAndTime format is invalid.
-5	Bad firmware image.
-99	System error.

SOAP XML Request Format

```
scheduleReload
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mes="http://meshDeviceOps.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <mes:scheduleReload>
         <firmwareGroup>?</firmwareGroup>
         <firmwareImageName>?</firmwareImageName>
         <reloadGmtTime>?</reloadGmtTime>
      </mes:scheduleReload>
   </soapenv:Body>
</soapenv:Envelope>
```



CHAPTER 6

Issues API

This chapter describes the Issues API.

Using the Issues API

In your IoT FND NB API client application, use this IoT FND server URL to access the Issues API WSDL:

http://<*server_address*> /nbapi/issue?wsdl

Issues API Method Calls

searchIssues

This call searches for issues in the IoT FND database based on the provided criteria and returns a list of those issues.

Prototype

```
 <even:searchIssues
>
  <query>deviceType:cgmesh issue:lowBattery</query>
  <count>4</count>
  <offset>0</offset>
 </even:searchIssues>
```

Parameters

Table 66: searchIssues Parameters

Parameter	Type	Option	Description
query	string	deviceType name issue issueStatus lastUpdateTime occurTime issueSeverity	Search query string. Table 67: Query Option Parameters , on page 74 lists valid parameters for these options.
count	integer	–	Number of results to retrieve.
offset	integer	–	Position of the first result.

Table 67: Query Option Parameters

Option Name	Device	Parameter	Description
deviceType	db asr1000 cgmesh IoT FND cgr1000 ir500 c800 ir800	db asr1000 cgmesh cgnms cgr1000 ir500 c800 ir800	The type of device.
name		User-defined entry.	
issue	db	criticallyLowFRASpace	The database Flash Recovery Area (FRA) free space is critically low. Run database backup immediately or risk IoT FND and database failure.
	db	criticallyLowTableSpace	Database “USERS” tablespace is critically low. Contact your DBA immediately to add more space or risk IoT FND and database failure.

Option Name	Device	Parameter	Description
	asr1000 cgmesh cgr1000 c800 ir800 ir500	deviceDown	The device is down.
	cgmesh c800 ir800 ir500	deviceTimeMismatch	The device time and NMS time are not in sync.
	cgr1000 c800 ir800	doorOpen	The device door was tampered with. Check corresponding event for details.
	asr1000 cgmesh cgr1000 c800 ir800 ir500	down	The device is down.
	cgr1000 c800 ir800	linePowerFailure	The line power supply of the CGR failed.
	cgr1000 c800 ir800	linecardFailure	Linecard failure detected.
	cgr1000 c800 ir800	lowBattery	The device is running on a low battery.
	cgr1000 c800 ir800	lowFlashSpace	The device is low on flash space.

Option Name	Device	Parameter	Description
	cgr1000 c800 ir800	lowMemory	The device is low on memory space.
	cgr1000 c800 ir800	majorTemperature	Major temperature alarm triggered.
	cgr1000 c800 ir800	minorTemperature	Minor temperature alarm triggered.
	cgr1000 c800 ir800	portDown	The Ethernet interface {0}/{1} is down.
	cgr1000 c800 ir800	portFailure	Device port failure detected.
	cgr1000 c800 ir800	powerSourceAlarm	One or more input power sources are not connected. Check the related event for details.
	cgr1000 c800 ir800	rplTreeSizeMajor	More than the expected nodes joined the RPL tree.
	cgr1000 c800 ir800	rplTreeSizeCritical	More than the maximum number of nodes joined the RPL tree.
	asr1000 cgmesh cgr1000 c800 ir800 ir500	signatureFailure	Invalid signature reported by mesh nodes. If this issue occurs, you must verify the certificate setup and that the mesh node and IoT FND are time synchronized.

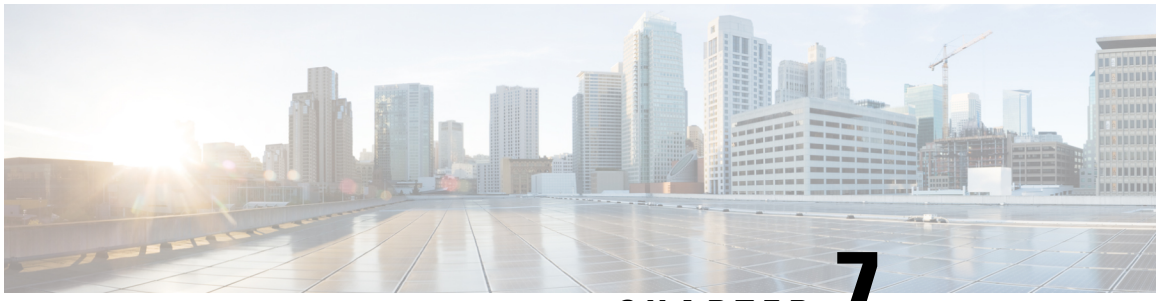
Option Name	Device	Parameter	Description
	asr1000 cgmesh IoT FND cgr1000 c800 ir800 ir500	unknown	The issue does not have a defined issue type.
	cgr1000 c800 ir800	wpanWatchdogReload	The WatchDog is reloading the WPAN module. The bridge was unresponsive for more than 5 minutes and the WatchDog is enabled.
issueStatus	N/A	OPEN CLOSED	
lastUpdateTime	N/A	UTC date and time in the format: yyyy-MM-dd HH:mm:ss:SSS. Delimiters are: >|<|>=|<=	
occurTime	N/A	UTC date and time in the format: yyyy-MM-dd HH:mm:ss:SSS. Delimiters are: >|<|>=|<=	
issueSeverity	db asr1000 cgmesh IoT FND cgr1000 c800 ir800 ir500	CRITICAL MAJOR MINOR INFO	

Results

This call returns a list of issueDetail structures.

Table 68: searchIssues Results

Name	Type	Description
eid	string	Device ID.
issueLastUpdateTime	dateTime	Last time the issue was updated.
issueMessage	string	Issue description.
issueOccurTime	dateTime	Issue occurrence time.
issueStatus	string	Issue status.
issueTypeName	string	Issue type.
meterId	string	Meter ID.



CHAPTER 7

Mesh Firmware Migration API

Note This API only applies to CGRs running CG-OS software.

This chapter describes the Firmware Migration API.

Using the Mesh Firmware Migration API

In your IoT FND NB API client application, use the following IoT FND server URL to access the Mesh Firmware Migration API WSDL:

http://<*server_address*> /nbapi/reprovision?wsdl

Mesh Firmware Migration API Method Calls

IoT FND allows you to update earlier versions of CGR firmware to allow Cisco mesh networking using the following APIs:

The mesh firmware migration process also requires editing of the Router Configuration and FAR Addition templates in IoT FND. See the IoT FND User Guide.

cancelReprovision

This call cancels a scheduled reprovisioning operation. Devices are queued in batches of 12 in FIFO order. As soon as the top reprovisioning operation completes, reprovisioning begins on the next device in the queue. When a reprovisioning operation is canceled, reprovisioning operations in progress complete. If the operation is scheduled for the future, then this call cancels the entire operation.

Prototype

```
cancelReprovision
(String uid)
```

Parameters

The following table describes the parameters in the interface.

Table 69: cancelReprovision Parameters

Parameter	Type	Description
uid	string	The UID of the scheduled reprovision operation.

Results

If the uid parameter references an unknown UID, the UID of an operation in progress, or a UID of a device that completed the reprovisioning process, the resultStatus is FAILED, and an appropriate errorDetails message is set. If the resultStatus is SUCCESS, the operation was canceled and will not execute.

The following table describes the parameters in the response.

Table 70: cancelReprovision Results

Parameter	Type	Description
resultStatus	string	SUCCESS or FAILED. FAILED returns only on a bad UID. For all other results such as the UID of an operation in progress or UID of a device that completed the reprovisioning process, resultStatus is SUCCESS and the cancel operation did not run.
errorDetails	string	Error description if resultStatus is FAILED.

showReprovisionStatus

This call retrieves the status of the reprovision operation correlating to the specified UID.

Prototype

```
showReprovisionStatus
(String uid)
```

Parameters

The following table describes the parameters in the interface.

Table 71: showReprovisionStatus Parameters

Parameter	Type	Description
uid	string	The UID of the scheduled reprovision operation.

Results

```
ShowReprovisionStatusReport showReprovisionStatus(String uid)
```

A ShowReprovisionStatusReport always returns. resultStatus is ERROR if the UID does not reference a known operation, which happens if the UID was incorrect or if the UID referenced an operation that completed and was cleaned up by the automatic pruning logic. Data retention time defaults to 7 days after an operation completes, which is configurable in IoT FND.

The following table describes the parameters in the response.

Table 72: showReprovisionStatus Results

Parameter	Type	Description
resultStatus	string	SUCCESS or FAILED.
errorDetails	string	Error description if resultStatus is FAILED.
uid	string	The UID of the reprovision operation correlating to this status.
operationStatus	string	The ReprovisionOperationStatus value correlating to this operation.
scheduledFor	int	Date specified in startReprovisionByEidList, on page 82, startReprovisionByEidListAbridged, on page 83, startReprovisionByGroup, on page 84, or startReprovisionByGroupAbridged, on page 85 for operation execution.
submittedAt	int	Date when this operation was submitted to startReprovisionByEidList, on page 82.
totalCount	int	Total number of devices specified in the list of EIDs in startReprovisionByEidList, on page 82, startReprovisionByEidListAbridged, on page 83, startReprovisionByGroup, on page 84, or startReprovisionByGroupAbridged, on page 85.
processedAt	int	Date when the operation began.
completedAt	int	Date when the operation finished processing all devices listed in startReprovisionByEidList, on page 82, startReprovisionByEidListAbridged, on page 83, startReprovisionByGroup, on page 84, or startReprovisionByGroupAbridged, on page 85 if COMPLETED or FAILED; otherwise this parameter is null.
successCount	int	Number of devices successfully processed.
failedCount	int	Number of items that failed the processing operation.
itemReports	int	List of ReprovisionItemReport objects. Each object defines the status for a single EID specified in startReprovisionByEidList, on page 82, startReprovisionByEidListAbridged, on page 83, startReprovisionByGroup, on page 84, or startReprovisionByGroupAbridged, on page 85.

For a ShowReprovisionStatusReport object with an operationStatus of SCHEDULED or CANCELED, ReprovisionItemReport is null because no information is available for those objects until processing begins. For all other operationStatus values, a ReprovisionItemReport object will be returned for each EID that is to be processed.

The following table describes the parameters in the ReprovisionItemReport for the specified device.

Table 73: showReprovisionStatus ReprovisionItemReport Results

Parameter	Type	Description
eid	int	The EID of the device.
itemStatus	string	The ReprovisionItemStatus value related to this device.
errorDetails	string	Detailed error message if operationStatus is ERROR.
processedAt	int	Date when the operation began.
completedAt	int	Date when the operation finished processing the device.

startReprovisionByEidList

This call is general for all reprovisioning actions, not just for mesh migration. Some input parameters are not applicable to mesh migration operations. This call schedules an operation to execute at a future date.

Note For startReprovisionByEidList and startReprovisionByGroup, on page 84, specify the interface name and interface type if all FARs have the same interface name and type. This is normally used for internal testing.

Prototype

```
startReprovisionByEidList
(String action, List<String> eidList, String interfaceName, String interfaceType, Date
executionDate)
```

Parameters

The following table describes the parameters in the interface.

Table 74: startReprovisionByEidList Parameters

Parameter	Type	Description
action	string	The name of the reprovisioning action to run. For mesh migration, this is MESH_FIRMWARE_ACTIVATION.
eidList	string	List of EIDs to perform the action on. **Note** The list is limited to a maximum of 1000 IP addresses. An error returns and the operation is not scheduled if more than 1000 are specified.
interfaceName	string	(For test purposes only.) The name of the interface involved in the action. **Note** Use the abridged calls for groups of mesh devices.
interfaceType	string	(For test purposes only.) The type of the interface involved in the action **Note** Use the abridged calls for groups of mesh devices.

Parameter	Type	Description
executionDate	date	Date to execute the action. This is a required parameter. Specify the current time or a time in the past to execute immediately.

Results

This method always returns a StartReprovisionReport object. If the operation failed to execute due to invalid parameters or if the EID list was determined invalid, FAILED returns. An operation is only scheduled and the UID returns if resultStatus is SUCCESS.

The following table describes the parameters in the response.

Table 75: startReprovisionByEidList Results

Parameter	Type	Description
resultStatus	int	SUCCESS or FAILED.
errorDetails	string	Error description if resultStatus is FAILED.
uid	string	The UID of the reprovision operation correlating to this status.

startReprovisionByEidListAbridged

This call schedules an operation to execute at a future date.

Prototype

```
startReprovisionByEidListAbridged
(String action, List<String> eidList, String interfaceName, String interfaceType, Date
executionDate)
```

Parameters

The following table describes the parameters in the interface.

Table 76: startReprovisionByEidListAbridged Parameters

Parameter	Type	Description
action	string	The name of the reprovisioning action to run. For mesh migration, this is “Mesh Activation.”
eidList	string	List of EIDs to perform the action on. **Note** The list is limited to a maximum of 1000 IP addresses. An error returns and the operation is not scheduled if more than 1000 are specified.
executionDate	date	Date to execute the action. This is a required parameter. Specify the current time or a time in the past to execute immediately.

Results

This method always returns a StartReprovisionReport object. If the operation failed to execute due to invalid parameters or if the EID list was determined invalid, FAILED returns. An operation is only scheduled and the UID returns if resultStatus is SUCCESS.

The following table describes the parameters in the response.

Table 77: startReprovisionByEidListAbridged Results

Parameter	Type	Description
resultStatus	int	SUCCESS or FAILED.
errorDetails	string	Error description if resultStatus is FAILED.
uid	string	The UID of the reprovision operation correlating to this status.

startReprovisionByGroup

This call executes the reprovisioning operation on the specified group.

Prototype

```
startReprovisionByGroup
(String action, String groupName, String interfaceName, String interfaceType, Date
executionDate)
```

Parameters

The following table describes the parameters in the interface.

Table 78: startReprovisionByGroup Parameters

Parameter	Type	Description
action	string	The name of the reprovisioning action to run. For mesh migration, this is “Mesh Activation.”
groupName	string	The name of the tunnel provisioning group.
executionDate	date	Date to execute the action. This is a required parameter. Specify the current time or a time in the past to execute immediately.

Results

This method always returns a StartReprovisionReport object. If the operation failed to execute due to invalid parameters or if the EID list was determined invalid, FAILED returns. An operation is only scheduled and the UID returns if resultStatus is SUCCESS.

The following table describes the parameters in the response.

Table 79: startReprovisionByGroup Results

Parameter	Type	Description
resultStatus	int	SUCCESS or FAILED.
errorDetails	string	Error description if resultStatus is FAILED.
uid	string	The UID of the reprovision operation correlating to this status.

startReprovisionByGroupAbridged

This call executes the reprovisioning operation on the specified group.

Prototype

```
startReprovisionByGroupAbridged
(String action, String groupName, Date executionDate)
```

Parameters

The following table describes the parameters in the interface.

Table 80: startReprovisionByGroupAbridged Parameters

Parameter	Type	Description
action	string	**Note** For mesh migration, this parameter is "Mesh Activation." The name of the reprovisioning action to run.
groupName	string	The name of the tunnel provisioning group.
executionDate	int	Date to execute the action. This is a required parameter. Specify the current time or a time in the past to execute immediately.

Results

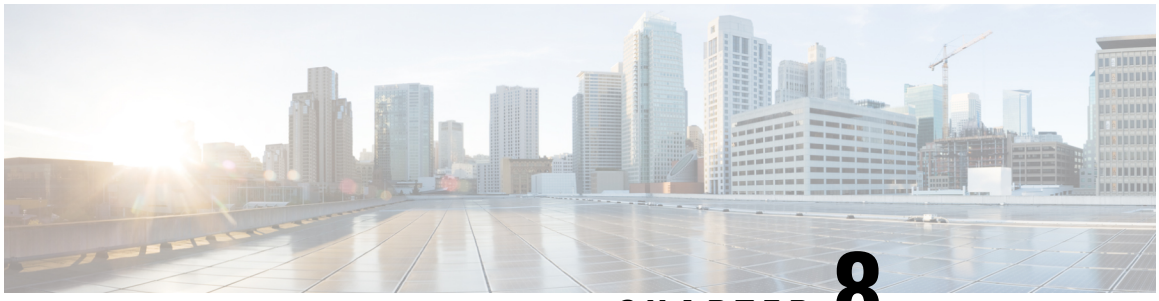
This method always returns a StartReprovisionReport object. If the operation failed to execute due to invalid parameters or if the EID list was determined invalid, FAILED returns. An operation is only scheduled and the UID returns if resultStatus is SUCCESS.

The following table describes the parameters in the response.

Table 81: startReprovisionByGroupAbridged Results

Parameter	Type	Description
resultStatus	int	SUCCESS or FAILED.
errorDetails	string	Error description if resultStatus is FAILED.

Parameter	Type	Description
uid	string	The UID of the reprovision operation correlating to this status.



CHAPTER 8

Rules API

This chapter describes the Rules API.

Using the Rules API

The Rules API provides basic functionality to add, remove, activate, deactivate, and find rules. Each rule is identified by a name and a user. A user cannot have two rules with the same name, but different users can have rules with the same name.

When creating new rules, specify the rule name. To create a new rule, use the same syntax used in the Search API.

After creating a rule, the IoT FND Rules API returns a rule ID, which you can use to refer to the rule for removing, activating, and deactivating, operations. By default, IoT FND activates rules when created with actions.

In your IoT FND NB API client application, use this IoT FND server URL to access the Rules API WSDL:

http://<*server_address*> /nbapi/rules?wsdl

Example (Python-SUDS Client)

```
import sys
from suds.client import Client
import logging
logging.basicConfig(level=logging.INFO)
logging.getLogger('suds.client').setLevel(logging.DEBUG)
url = "http://localhost/nbapi/rules?wsdl"
cl = Client(url, username='root', password='Tree123!')
#print cl
#parse the properties of a rule and return a dictionary struct
def parseRule(r):
    rule = {}
    rule['id'] = r.id
    if (r.properties != None and r.properties != ""):
      #print r.properties
      for p in r.properties.entry:
        if (p.key == "literal"):
   rule['rule']=p.value
```

```
 else:
   rule[p.key]=p.value
    for a in r.actions:
      rule['action'] = a.type
      rule['label'] = a.parameter
    #print rule
    return rule
# output a CSV list of rules
users = ['root']
properties = ['username','status', 'lastUpdate', 'name', 'literal']
print "username,rulename,id,status,lastUpdate,rule,action,label"
for u in users:
  #print u
  result = cl.service.findRulesByUsername(u)
  #print result
  for r in result:
    rule = parseRule(r)
    print rule['username']+",",
    print rule['name']+",",
    print str(rule['id'])+",",
    print rule['status']+",",
    print rule['lastUpdate']+",",
    print rule['rule']+",",
    print rule['action']+",",
    print rule['label']
# create two new rules
names = ['My New Rule 1', 'My New Rule 2']
myAction1 = cl.factory.create('ruleAction')
myAction1.parameter = 'My Label'
myAction1.type = 'ADD_LABEL'
myAction2 = cl.factory.create('ruleAction')
myAction2.parameter = 'My Label'
myAction2.type = 'REMOVE_LABEL'
myActions = [myAction1,myAction2]
myRule1 = cl.factory.create("createRule")
myRule1.name = names[0]
myRule1.username = users[0]
myRule1.literal = 'deviceType:cgmesh status:down meshRssi<-90 meshHops>5'
myRule1.actions = myAction1
print myRule1
myRule2 = cl.factory.create("createRule")
myRule2.name = names[1]
myRule2.username = users[0]
myRule2.literal = 'label:"My Label" deviceType:cgmesh status:up'
myRule2.actions = myAction2
print myRule2
r = cl.service.createRule(myRule1)
myr1 = parseRule(r)
print myr1
r = cl.service.createRule(myRule2)
myr2 = parseRule(r)
print myr2
# toggle rules by name
for n in names:
  #print n
  result = cl.service.findRulesByName(n)
  if result != None:
    #print result
    for r in result:
      rule = parseRule(r)
      if (rule['status'] == "ACTIVATED"):
   r = cl.service.deactivateRule(rule['id'])
      else:
   r = cl.service.activateRule(rule['id'])
```

```
      rule = parseRule(r)
      print rule
sys.exit()
# delete the rules we created
r = dropRule(myr1["id"])
r = dropRule(myr2["id"])
```

Rules API Method Calls

activateRule

This call activates a deactivated rule. By default, when you create a rule with actions, IoT FND activates it. Only activated rules receive trigger events on rule matching.

Prototype

```
 <rul:activateRule
>
  <id
>?</id>
 </rul:activateRule>
```

Parameters

Table 82: activateRule Parameters

Parameter	Type	Description
id	long	Rule ID.

Results

This call returns the rule object if activation is successful.

Table 83: activateRule Results

Field	Type	Description
id	long	Rule ID.
properties	Map<String,String>	Rule properties such as name, username, status, and lastUpdate.
actions	List<RuleAction>	List of actions the rule performs.

activateRule SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
```

```
      <rul:activateRule>
         <!--Optional:-->
         <id>?</id>
      </rul:activateRule>
   </soapenv:Body>
</soapenv:Envelope>
```

createRule

This call creates a new rule with the specified action and name. The rule language is the same as the search language. The action can be one of the following: create label, remove label, or add event.

Prototype

```
 <rul:createRule
>
  <name
>?</name>
  <username
>?</username>
  <literal
>?</literal>
  <actions
>
  <parameter
>?</parameter>
  <type
>create_label
</type>
  </actions>
 </rul:createRule>
```

Parameters

Table 84: createRule Parameters

Parameter	Type	Description
name	string	The name of the rule.
username	string	The IoT FND username of the rule creator.
literal	string	The rule syntax.
actions	ruleAction	A list of rule actions. For every action, you must provide two strings: • parameter—Action label. • type—Action type. Valid options are: • **create_label** • **remove_label** • **add_event**

Results

Table 85: createRule Results

Field	Type	Description
id	long	Rule ID.
properties	Map<String,String>	Rule properties such as name, username, status, and lastUpdate.
actions	List<RuleAction>	List of actions the rule performs.

createRule SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <rul:createRule>
         <!--Optional:-->
         <name>?</name>
         <!--Optional:-->
         <username>?</username>
         <!--Optional:-->
         <literal>?</literal>
         <!--Zero or more repetitions:-->
         <actions>
            <!--Optional:-->
            <parameter>?</parameter>
            <!--Optional:-->
            <type>?</type>
         </actions>
      </rul:createRule>
   </soapenv:Body>
</soapenv:Envelope>
```

Sample createRule Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
    <soapenv:Header/>
    <soapenv:Body>
       <rul:createRule>
          <!--Optional:-->
          <name>2016-03-09-addEvent-3</name>
          <!--Optional:-->
          <username>root</username>
          <!--Optional:-->
          <literal>deviceCategory:router issue:lowMemory</literal>
          <!--Zero or more repetitions:-->
          <actions xmlns=''>

<parameter>{"eventName":"testINFO","eventSeverity":"INFO","eventMsg":"test_INFO"}</parameter>

             <type>add_event</type>
             </actions>
              <actions xmlns=''>
             <parameter>AndersonLabel</parameter>
```

```
            <type>add_label</type>
            </actions>
             <actions xmlns=''>
            <parameter>AndersonLabel</parameter>
            <type>remove_label</type>
         </actions>
      </rul:createRule>
   </soapenv:Body>
</soapenv:Envelope>
```

Sample createRule Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"/>
   <soap:Body>
      <ns2:createRuleResponse xmlns:ns2="http://rules.nbapi.cgms.cisco.com/">
         <rule>
            <actions>

<parameter>{"eventName":"testINFO","eventSeverity":"INFO","eventMsg":"test_INFO"}</parameter>

               <type>ADD_EVENT</type>
            </actions>
            <actions>
               <parameter>AndersonLabel</parameter>
               <type>ADD_LABEL</type>
            </actions>
            <actions>
               <parameter>AndersonLabel</parameter>
               <type>REMOVE_LABEL</type>
            </actions>
            <id>20001141</id>
            <properties>
               <entry>
                  <key>username</key>
                  <value>root</value>
               </entry>
               <entry>
                  <key>status</key>
                  <value>DEACTIVATED</value>
               </entry>
               <entry>
                  <key>lastUpdate</key>
                  <value>2016-03-09 17:28:25.771</value>
               </entry>
               <entry>
                  <key>name</key>
                  <value>2016-03-09-addEvent-3</value>
               </entry>
               <entry>
                  <key>literal</key>
                  <value>deviceCategory:router issue:lowMemory</value>
               </entry>
            </properties>
         </rule>
      </ns2:createRuleResponse>
   </soap:Body>
</soap:Envelope>
```

deactiveRule

This call deactivates the specified rule and returns the rule object if deactivation is successful.

Prototype

```
 <rul:deactiveRule
>
  <id
>?</id>
 </rul:deactiveRule>
```

Parameters

Table 86: deactivateRule Parameters

Parameter	Type	Description
id	long	Rule ID.

Results

This call returns the rule object if deactivation is successful.

Table 87: deactivateRule Results

Field	Type	Description
id	long	Rule ID.
properties	Map<String,String>	Rule properties such as name, username, status, and lastUpdate.
actions	List<RuleAction>	List of actions the rule performs.

deactivateRule SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <rul:createRule>
         <!--Optional:-->
         <name>?</name>
         <!--Optional:-->
         <username>?</username>
         <!--Optional:-->
         <literal>?</literal>
         <!--Zero or more repetitions:-->
         <actions>
            <!--Optional:-->
            <parameter>?</parameter>
            <!--Optional:-->
            <type>?</type>
         </actions>
```

```
      </rul:createRule>
   </soapenv:Body>
</soapenv:Envelope>
```

dropRule

This call removes the rule specified in the *id* parameter.

Prototype

```
 <rul:dropRule
>
  <id
>?</id>
 </rul:dropRule>
```

Parameters

Table 88: dropRule Parameters

Parameter	Type	Description
id	long	ID of rule to remove.

Results

Table 89: dropRule Results

Field	Type	Description
id	long	ID of removed rule.
properties	Map<String,String>	Rule properties such as name, username, status, and lastUpdate.
actions	List<RuleAction>	List of actions the rule performs.

dropRule SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <rul:createRule>
         <!--Optional:-->
         <name>?</name>
         <!--Optional:-->
         <username>?</username>
         <!--Optional:-->
         <literal>?</literal>
         <!--Zero or more repetitions:-->
         <actions>
            <!--Optional:-->
            <parameter>?</parameter>
```

```
            <!--Optional:-->
            <type>?</type>
         </actions>
      </rul:createRule>
   </soapenv:Body>
</soapenv:Envelope>
?
dropRule SOAP XML request format
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <rul:dropRule>
         <!--Optional:-->
         <id>?</id>
      </rul:dropRule>
   </soapenv:Body>
</soapenv:Envelope>
```

findRulesByName

This call returns all rules specified in the *name* parameter.

Prototype

```
 <rul:findRulesByName
>
  <name
>?</name>
 </rul:findRulesByName>
```

Parameters

Table 90: findRulesByName Parameters

Parameter	Type	Description
name	string	Rule name.

Results

This call returns a list of rules matching the name string.

Table 91: findRulesByName Results

Field	Type	Description
id	long	Rule ID.
properties	Map<String,String>	Rule properties such as name, username, status, and lastUpdate.
actions	List<RuleAction>	List of actions the rule performs.

findRulesByName SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <rul:findRulesByName>
         <!--Optional:-->
         <name>?</name>
      </rul:findRulesByName>
   </soapenv:Body>
</soapenv:Envelope>
```

findRulesByUsername

This call returns a list of rules created by the user defined in *username* .

Prototype

```
 <rul:findRulesByUsername
>
  <username
>?</username>
 </rul:findRulesByUsername>
```

Parameters

Table 92: findRulesByUsername Parameters

Parameter	Type	Description
username	string	The IoT FND username of the creator of the rules.

Results

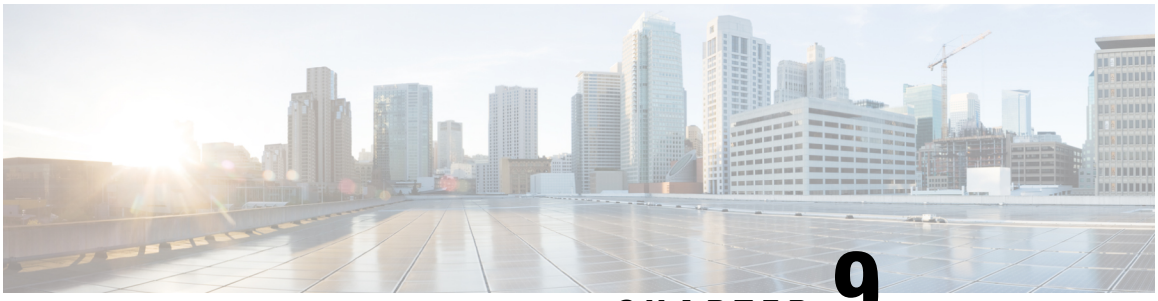
Table 93: findRulesByUsername Results

Field	Type	Description
id	long	Rule ID.
properties	Map<String,String>	Rule properties such as name, username, status, and lastUpdate.
actions	List<RuleAction>	List of actions the rule performs.

findRulesByUsername SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <rul:findRulesByUsername>
```

```
         <!--Optional:-->
         <username>?</username>
      </rul:findRulesByUsername>
   </soapenv:Body>
</soapenv:Envelope>
```



CHAPTER 9

Search API

This chapter describes the Search API.

Overview

All search interfaces return a queryStatus, in addition to the actual search results. Search interfaces with count and offset arguments also return a queryId. The queryId is useful for retrieving more results from the same search. The queryStatus contains an integer status code and a string message about the code.

The search API provides the following interfaces:

- DeviceDetailQueryResult getDeviceDetails(string query, list<string> deviceIds, string queryId, long count, long offset)
- DeletedDeviceQueryResult getDeletedDevices(string deviceType, date startTime, date endTime)
- DeviceQueryResult searchDevices(string query, list<string> fieldNames, string queryId, long count, long offset)
- EidsForIpAddressesResult findEidsForIpAddressesByDeviceType(string deviceType, list<string> ipAddresses)
- EidsForIpAddressesResult findEidsForIpAddresses(list<string> ipAddresses)
- GroupQueryResult getGroups(string groupType)
- MetricHistoryQueryResult getMetricHistory(string query, list<string> deviceIds, date startTime, date endTime, List<string> metricIds, string rollupInterval, string rollupFunction, string queryId, long count, long offset)
- UpdatedDeviceDetailQueryResult getUpdatedDeviceDetails(string query, date startTime, date endTime, string queryId, long count, long offset)

Using the Search API

Use this IoT FND server URL to access the Search API WSDL:

http://<*server_address*> /nbapi/search?wsdl

Provided are these examples of client applications using the Search API.

Example 1 (Python-SUDS Client)

```
from suds.client import Client
import logging
logging.basicConfig(level=logging.INFO)
logging.getLogger('suds.client').setLevel(logging.INFO)
url = "http://localhost/nbapi/search?wsdl"
cl = Client(url, username='root', password='Tree123!')
globalMetrics = ['uptime']
cgmeshMetrics =
['meshHops','meshLinkCost','meshPathCost','meshRssi','meshReverseRssi','meshRxSpeed','meshTxSpeed']
cgr1kStatus =
['doorStatus','numBBU','battery0Level','battery0Runtime','battery1Level','battery1Runtime','battery2Level','battery2Runtime']
cgr1kMetrics = ['chassisTemp','ethernetTxDrops','ethernetRxSpeed','ethernetTxSpeed']
cgr1kCellIfMetrics = ['cellularRSSI','cellularRxSpeed','cellularTxSpeed']
cgr1kWiMAXIfMetrics = ['wimaxRSSI','wimaxRxSpeed','wimaxTxSpeed']
cgr1kMeshIfMetrics = ['meshEndpointCount','meshRxSpeed','meshTxSpeed']
# search FAN router devices
devices = []
result = cl.service.searchDevices('deviceType:cgr1000
status:down',['lng','lat','ip'],'',1000,0)
#print result
if result.queryStatus == "SUCCEEDED":
  print "eid,lng,lat,ip,"
  for d in result.devices:
    devices.append(d.eid)
    print str(d.eid)+",",
    for f in d.fields.entry:
      print str(f.value)+",",
    print ""
```

Example 2 (Python-SUDS Client)

```
from suds.client import Client
import logging
logging.basicConfig(level=logging.INFO)
logging.getLogger('suds.client').setLevel(logging.INFO)
url = "http://localhost/nbapi/search?wsdl"
cl = Client(url, username='root', password='Tree123!')
#print cl
# search mesh end devices
devices = []
result = cl.service.searchDevices('deviceType:cgmesh',['lng','lat','ip'],'',10,0)
#print result
if result.queryStatus == "SUCCEEDED":
  print "eid,lng,lat,ip,"
  for d in result.devices:
    devices.append(d.eid)
    print str(d.eid)+",",
```

```
    for f in d.fields.entry:
      print str(f.value)+",",
    print ""
globalMetrics = ['uptime']
cgmeshMetrics =
['meshHops','meshLinkCost','meshPathCost','meshRssi','meshReverseRssi','meshRxSpeed','meshTxSpeed']
cgr1kStatus =
['doorStatus','numBBU','battery0Level','battery0Runtime','battery1Level','battery1Runtime','battery2Level','battery2Runtime']
cgr1kMetrics = ['chassisTemp','ethernetTxDrops','ethernetRxSpeed','ethernetTxSpeed']
cgr1kCellIfMetrics = ['cellularRSSI','cellularRxSpeed','cellularTxSpeed']
cgr1kWiMAXIfMetrics = ['wimaxRSSI','wimaxRxSpeed','wimaxTxSpeed']
cgr1kMeshIfMetrics = ['meshEndpointCount','meshRxSpeed','meshTxSpeed']
# get device metric history
result =
cl.service.getMetricHistory('status:up',[str(devices[0]),str(devices[1]),str(devices[2])],'2012-04-01
 00:00:00','2012-04-01 23:59:59',cgmeshMetrics,'day','avg','',2,0)
#print result
if (result.queryStatus == "SUCCEEDED"):
  try:
    result.metricValues
  except:
    print "no metrics returned"
  else:
    print "eid,metric,value,timestamp,"
    for m in result.metricValues:
        print str(m.eid), ",", str(m.metricId),",", str(m.value),",", str(m.timestamp)
# search FAN router devices
devices = []
result = cl.service.searchDevices('deviceType:cgr1000',['lng','lat','ip'],'',10,0)
#print result
if result.queryStatus == "SUCCEEDED":
  print "eid,lng,lat,ip,"
  for d in result.devices:
    devices.append(d.eid)
    print str(d.eid)+",",
    for f in d.fields.entry:
      print str(f.value)+",",
    print ""
# get device metric history
for f in devices:
  result = cl.service.getMetricHistory('status:up',f,'2012-04-01 00:00:00','2012-04-01
23:59:59',cgr1kMetrics+cgr1kCellIfMetrics+cgr1kWiMAXIfMetrics+cgr1kMeshIfMetrics,'day','avg','',0,0)

  #print result
  if (result.queryStatus == "SUCCEEDED"):
    try:
      result.metricValues
    except:
      print "no metrics returned"
    else:
      print "eid,metric,value,timestamp,"
      for m in result.metricValues:
        print str(m.eid), ",", str(m.metricId),",", str(m.value),",", str(m.timestamp)
```

Example 3 (Python-SUDS Client)

```
import sys
from suds.client import Client
import logging
logging.basicConfig(level=logging.INFO)
logging.getLogger('suds.client').setLevel(logging.DEBUG)
url = "http://localhost/nbapi/search?wsdl"
```

```
cl = Client(url, username='root', password='Tree123!')
# get config groups
groups =
['DeviceType','Status','ConfigGroup','FirmwareGroup','TunnelProvisioningGroup','Label']
for g in groups:
  result = cl.service.getGroups(g)
  #print result
  if result.queryStatus == "SUCCEEDED":
    print g
    for r in result.groups:
      print r.name,
      if (r.properties != None and r.properties != ""):
        for p in r.properties.entry:
          if (p.key == "description" or p.key == "deviceType"):
     print p.value,
      if r.memberCount != None:
        print "x " + str(r.memberCount) + ",",
      print ""
    print ""
# search devices
devices = []
result = cl.service.searchDevices('status:up',['lng','lat','ip'],'',10,0)
#print result
if result.queryStatus == "SUCCEEDED":
  print "eid,lng,lat,ip,"
  for d in result.devices:
    devices.append(d.eid)
    print str(d.eid)+",",
    for f in d.fields.entry:
      print str(f.value)+",",
    print ""
# fetch mesh endpoint node device details
metrics =
("uptime","nodeLocalTime","meshHops","meshRssi","meshReverseRssi","meshLinkCost","meshPathCost","meshRxSpeed","meshTxSpeed","meshTxDrops")
properties =
("s1","deviceType","pid","lng","lat","alt","gateway","hopLevel","lastHeard","status","configSync","runningFirmwareVersion","hardwareId","vid","ent3N","ent1","ent0","ent5N","ent5","ent0","ent0")
devices = []
result = cl.service.searchDevices('deviceType:cgmesh','','',10,0)
if result.queryStatus == "SUCCEEDED":
  for d in result.devices:
    devices.append(d.eid)
  print devices
if len(devices) != 0:
  fn = "./cgmeshDevDetails.csv"
  f = open(fn, 'w')
  #fetch device details
  print >>f, "eid,configGroup,firmwareGroup,",
  for p in properties:
    print >>f, p+",",
  for m in metrics:
    print >>f, m+",",
  print >>f, "interfaces"
  for d in devices:
    result = cl.service.getDeviceDetails('',d,'',0,0)
    #print result
    if result.queryStatus == "SUCCEEDED":
      print >>f,
d+","+result.deviceDetails[0].configGroup+","+result.deviceDetails[0].firmwareGroup+",",
      for p in properties:
        for j in result.deviceDetails[0].properties.entry:
   if j.key == p:
     try:
       j.value
     except:
```

```
        pass
      else:
        if j.value != "null":
          print >>f, j.value,
        print >>f, ",",
       for m in metrics:
         for j in result.deviceDetails[0].metrics.entry:
    if j.key == m:
      print >>f, j.value,
         print >>f, ",",
       for i in result.deviceDetails[0].interfaces:
         print >>f, i.name,
       print >>f, ""
# fetch FAN router details
metrics =
["uptime","chassisTemp","ethernetTxSpeed","ethernetRxSpeed","ethernetTxDrops","meshEndpointCount","meshRxSpeed","meshTxSpeed"]
properties =
[illegible]
devices = []
result = cl.service.searchDevices('deviceType:cgr1000','','',10,0)
#print result
if result.queryStatus == "SUCCEEDED":
  for d in result.devices:
    devices.append(d.eid)
  print devices
if len(devices) != 0:
  fn = "./cgr1kDevDetails.csv"
  f = open(fn, 'w')
  #fetch device details
  print >>f, "eid,configGroup,firmwareGroup,",
  for p in properties:
    print >>f, p+",",
  for m in metrics:
    print >>f, m+",",
  print >>f, "interfaces"
  for d in devices:
    result = cl.service.getDeviceDetails('',d,'',0,0)
    #print result
    if result.queryStatus == "SUCCEEDED":
      print >>f,
d+","+result.deviceDetails[0].configGroup+","+result.deviceDetails[0].firmwareGroup+",",
       for p in properties:
         for j in result.deviceDetails[0].properties.entry:
    if j.key == p:
      try:
        j.value
      except:
        pass
      else:
        if j.value != "null":
          print >>f, j.value,
        print >>f, ",",
       for m in metrics:
         for j in result.deviceDetails[0].metrics.entry:
    if j.key == m:
      print >>f, j.value,
         print >>f, ",",
       for i in result.deviceDetails[0].interfaces:
         print >>f, i.name,
       print >>f, ""
```

Using the search-client Script

The IoT FND distribution provides a Search API client (search-client script) in the cgms-tools-*version* .x86_64.rpm package. The client is wrapped by a shell script (/opt/cgms-tools/bin/search-client) for communicating with the IoT FND API.

```
[root@localhost bin]# ./search-client

usage:
./search-client device <endpoint URL> <query> <field names> <count> <offset>
    ./search-client device <endpoint URL> <query Id> <field names>
    ./search-client deviceDetail <endpoint URL> <query> <count> <offset>
    ./search-client deviceDetail <endpoint URL> <query Id>
    ./search-client deviceDetail <endpoint URL> <device eid list>
   ./search-client updatedDeviceDetail <endpoint URL> <query> <startTime> <endTime> <count>
 <offset>
    ./search-client updatedDeviceDetail <endpoint URL> <queryId>
    ./search-client deletedDevice <endpoint URL> <deviceType> <startTime> <endTime>
    ./search-client group <endpoint URL> <group type>
            <group type> can be deviceType, label, status, configGroup, firmwareGroup,
tunnelGroup or
    subnetGroup
   ./search-client metricHistory <endpoint URL> <query> <startTime> <endTime> <field names>

         <rollupInterval> <rollupFunction> <count> <offset>
    ./search-client metricHistory <endpoint URL> <queryId>
    ./search-client metricHistory <endpoint URL> <device eid list> <startTime> <endTime>
<field names>
         <rollupInterval> <rollupFunction>
    ./search-client eidByAddresses <endPoint URL> <deviceIps>
   ./search-client ReprovisionByEids <endPoint URL> <groupName> <deviceEids> <interfaceName>

          <interfaceType>
```

The following table lists search-client script commands.

Table 94: search-client Commands

Command	Description
device	Calls the searchDevices API.
deviceDetail	Calls the getDeviceDetails API.
deletedDevice	Calls the getDeletedDevices API.
eidByAddresses	Calls the findEidsForIpAddresses API.
event	Calls the getUpdatedDeviceDetails API.
group	Calls the getGroups API.
issues	Calls the getUpdatedDeviceDetails API.
meshFirmwareUpgrade	Calls the getUpdatedDeviceDetails API.
metricHistory	Calls the getMetricHistory API.

Command	Description
updatedDeviceDetail	Calls the getUpdatedDeviceDetails API.
work order	Calls the getUpdatedDeviceDetails API.

Search API Method Calls

searchDevices

This call lets the client provide a search query string (using the query language) and returns a list of device details (properties or metrics).

Prototype

```
DeviceQueryResult searchDevices
(string query, list<string> fieldNames, string queryId, long count, long offset)
```

Parameters

Table 95: searchDevices Parameters

Parameter	Type	Description
query	string	Search query string.
fieldNames	string	List of property or metric names to retrieve for each device. For information about properties and metrics, see Property Field Names for All Devices and Metrics Field Names.
queryId	string	Query ID returned by the call. If available results for the query is more than count, the caller can use the returned queryId to call the same API repeatedly. When queryId is provided, all other parameters are ignored.
count	long	Number of results to retrieve. Valid range is 1–40000.
offset	long	Position of the first result. Valid values are >= 0.

Results

The following table lists device details returned by this call.

Table 96: searchDevices Results

Name	Type	Description
eid	string	Device ID.
fields	map<string.string>	A list of fieldname-value pairs.

Name	Type	Description
key	long	A long value associated with the device.

searchDevices SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sear="http://search.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <sear:searchDevices>
         <!--Optional:-->
         <query>deviceType:cgr1000</query>
         <!--Zero or more repetitions:-->
         <fieldNames></fieldNames>
         <!--Optional:-->
         <queryId></queryId>
         <!--Optional:-->
         <count>10</count>
         <!--Optional:-->
         <offset>0</offset>
   </sear:searchDevices> </soapenv:Body>
</soapenv:Envelope>
```

getGroups

This call lets the client retrieve the following:

- List of groups for a specific group type within the system
- Current number of members in each group
- Properties assigned to the group

Group types are any one of the following strings:

- deviceType
- status
- label
- configGroup
- firmwareGroup
- subnetGroup
- tunnelProvisioningGroup

Prototype

```
GroupQueryResult getGroups
(string groupType)
```

Parameters

Table 97: getGroups Parameters

Parameter	Type	Description
groupType	string	Group type strings from the above list.

Results

The following table lists groups returned by this call.

Table 98: getGroups Results

Name	Type	Description
name	string	Name of the group.
memberCount	long	Number of devices in the group.
properties	map<string, string>	Property name-value pairs.

getGroups SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sear="http://search.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <sear:getGroups>
         <!--Optional:-->
         <groupType>devicetype</groupType>
      </sear:getGroups>
   </soapenv:Body>
</soapenv:Envelope>
```

getDeviceDetails

This call lets the client retrieve the following information:

- Address list information:
  - Address
  - addressType
  - Key
  - prefixLength
- Assets list information:
  - Key
  - Name

    - Asset metrics
    - Asset properties
- ConfigGroup as a string
- EID as a string
- FirmwareGroup as a string
- Interface list information:
    - Interface key
    - Interface name
    - Interface index
    - Interface addresses list
    - Interface metrics
    - Interface properties
- Key as a long integer
- Labels list
- Metrics list
- Properties list
- SubnetGroup as a string
- Route list information:
    - Index
    - Key
    - Route metrics
    - Route properties

Devices are retrieved by specifying a query or a list. For example, the Route properties key-value pair information retrieved is:

```
destAddressType:2
nextHopAddressType:4
nextHopAddress : fe80:0:0:0:207:8108:3c:270b
destAddress:0:0:0:0:0:0:0:0
prefixLength:0
```

Prototype

```
DeviceDetailQueryResult getDeviceDetails
(string query, list<string> deviceIds, string queryId,long count, long offset)
```

Parameters

Table 99: getDeviceDetails Parameters

Parameter	Type	Description
query	string	Search query.
deviceIds	list<string>	List of device EIDs to retrieve. When deviceIds is provided, the query parameter is ignored.
queryId	string	Query ID returned by the call. If available results for the query is more than count, the caller can use the returned queryId to call the same API repeatedly. When queryId is provided, all other parameters are ignored.
count	long	Number of results to retrieve. Valid range is 1–40000
offset	long	Position of the first result. Valid values are >= 0.

Results

Parameters returned by this call are listed in the following table.

Table 100: getDeviceDetails Results

Parameter	Type	Description
addresses	list<address>	A list of addresses known to the device. See Table 101: Address Attributes, on page 110.
assets	list<asset>	A list of assets installed on the device. See Table 104: Asset Attributes , on page 110.
configGroup	string	The name of the configuration group assigned to the device.
eid	string	The string ID of the device.
firmwareGroup	string	The name of the firmware group assigned to the device.
interfaces	list<interface>	A list of interfaces of the device. See Table 102: Interface Attributes , on page 110.
labels	list<string>	The list of labels assigned to the device.
key	long	A long value associated with the device.
metrics	map<string, double>	Metric type-value pairs. See Metrics Field Names .
properties	map<string, string>	Property name-value pairs. See Property Field Names for All Devices .
routes	list<route>	A list of routes known to the device. See Table 103: Route Attributes , on page 110.
subnetGroup	string	The name of the subnet group assigned to the device.

The following table describes the Address attributes.

Table 101: Address Attributes

Attribute	Type	Description
key	long	A long value associated with the address.
addressType	string	Value is one of these address types: IPV4, IPV6, IPV4Z, IPV6Z or UNKNOWN.
address	string	The IP address.
prefixLength	integer	Subnet prefix length of the address.

The following table describes the Interface attributes.

Table 102: Interface Attributes

Attribute	Type	Description
key	long	A long value associated with the interface.
index	integer	Index value of the interface.
name	string	The name of the interface.
addresses	list<address>	List of Address (see Table 101: Address Attributes, on page 110) associated with the interface.
properties	map<string, string>	Properties of the interface in name-value pairs.
metrics	map<string, double>	Metrics of the interface.

The following table describes the Route attributes.

Table 103: Route Attributes

Attribute	Type	Description
key	long	A long value associated with the route.
index	integer	Index value of the route.
properties	map<string, string>	Properties of the route in name-value pairs.
metrics	map<string, double>	Metrics of the route.

The following table describes the Asset attribute parameters.

Table 104: Asset Attributes

Attribute	Type	Description
key	long	A long value associated with the asset.
name	string	The name of the asset.

Attribute	Type	Description
properties	map<string, string>	Properties of the asset in name-value pairs.
metrics	map<string, double>	Metrics of the asset.

getDeviceDetails SOAP XML Request Format for a FAR

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sear="http://search.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <sear:getDeviceDetails>
         <!--Optional:-->
         <query>far_test2</query>
         <!--Zero or more repetitions:-->
         <deviceIds>+JSJ1522003G</deviceIds>
         <!--Optional:-->
         <queryId></queryId>
         <!--Optional:-->
         <count>5</count>
         <!--Optional:-->
         <offset>0</offset>
      </sear:getDeviceDetails>
   </soapenv:Body>
</soapenv:Envelope>
```

getDeviceDetails SOAP XML Request Format for a Mesh Device

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sear="http://search.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <sear:getDeviceDetails>
         <!--Optional:-->
         <query>deviceType:cgmesh status:up</query>
         <!--Zero or more repetitions:-->
         <deviceIds></deviceIds>
         <!--Optional:-->
         <queryId></queryId>
         <!--Optional:-->
         <count>10</count>
         <!--Optional:-->
         <offset>0</offset>
      </sear:getDeviceDetails>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header>
      <seam:conversationId
xmlns:seam="http://www.jboss.org/seam/webservice">1125</seam:conversationId>
   </env:Header>
   <env:Body>
      <ns2:getDeviceDetailsResponse xmlns:ns2="http://search.nbapi.cgms.cisco.com/">
         <deviceDetailQueryResult>
```

```
            <queryId>E6FDA0F0D3E0ADFFF69E334462D1EF6A</queryId>
            <queryStatus>SUCCEEDED</queryStatus>
         </deviceDetailQueryResult>
      </ns2:getDeviceDetailsResponse>
   </env:Body>
</env:Envelope>
```

getDeletedDevices

This call retrieves information of devices deleted within a time period.

Prototype

```
DeletedDeviceQueryResult getDeletedDevices
(string deviceType, date startTime, date endTime)
```

Parameters

Table 105: getDeletedDevices Parameters

Parameter	Type	Description
deviceType	string	The name of the device type to filter the results by. For example, cgr1000 only matches deleted devices against CGR devices.
startTime	date	Starting UTC date and time for the deletion interval. This is a mandatory parameter. It cannot be null. Format is YYYY-MM-DDThh:mm:ss (for example, 2015-03-25T00:00:00).
endTime	date	Ending UTC date and time for the deletion interval. This is a mandatory parameter. It cannot be null. Use it as the startTime for the next call without retrieving repeated records. Format is YYYY-MM-DDThh:mm:ss.

Results

The following table lists device details returned by this call.

Table 106: getDeletedDevices Results

Name	Type	Description
deleteDate	date	The date and time of device deletion.
deviceType	string	The device type of the deleted device.
eid	string	Device ID.

search-client Script Example

```
search-client deletedDevice https://kml-lnx2/nbapi/search?wsdl "cgr1000" "2015-03-25 12:00:00"
 "2015-03-26 12:00:00"
```

search-client Script Results

```
URL: https://kml-lnx2/nbapi/search?wsdl
Mar 25, 2015 5:16:43 PM org.apache.cxf.service.factory.ReflectionServiceFactoryBean
buildServiceFromWSDL
INFO: Creating Service {http://search.nbapi.cgms.cisco.com/}SearchWebService from WSDL:
https://kml-lnx2/nbapi/search?wsdl
Mar 25, 2015 5:16:44 PM org.apache.cxf.service.factory.ReflectionServiceFactoryBean
buildServiceFromWSDL
INFO: Creating Service {http://search.nbapi.cgms.cisco.com/}SearchWebService from WSDL:
https://kml-lnx2/nbapi/search?wsdl
{"deleteDate":"Wed Mar 25 16:00:44 PDT
2015","deviceType":"cgr1000","eid":"CGR1240/K9+JSJ200201"}
{"deleteDate":"Wed Mar 25 16:00:44 PDT
2015","deviceType":"cgr1000","eid":"CGR1240/K9+JSJ200202"}

Elapsed Time: [169] ms
Status: Query succeeded
Rows: 2
```

SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sear="http://search.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <sear:getDeletedDevices>
         <deviceType>cgr1000</deviceType>
         <startTime>2015-03-25T00:00:00</startTime>
         <endTime>2015-03-26T00:00:00</endTime>
      </sear:getDeletedDevices>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"/>
   <soap:Body>
      <ns2:getDeletedDevicesResponse xmlns:ns2="http://search.nbapi.cgms.cisco.com/">
         <deletedDeviceQueryResult>
            <queryStatus>SUCCEEDED</queryStatus>
            <deletedDevices>
               <deleteDate>2015-03-25T23:00:44Z</deleteDate>
               <deviceType>cgr1000</deviceType>
               <eid>CGR1240/K9+JSJ200201</eid>
            </deletedDevices>
            <deletedDevices>
               <deleteDate>2015-03-25T23:00:44Z</deleteDate>
               <deviceType>cgr1000</deviceType>
               <eid>CGR1240/K9+JSJ200202</eid>
            </deletedDevices>
         </deletedDeviceQueryResult>
      </ns2:getDeletedDevicesResponse>
   </soap:Body>
</soap:Envelope>
```

Example Python Request

```
#!/usr/local/bin/python
from suds.transport.https import HttpAuthenticated
from suds.client import Client
import logging
from datetime import timedelta,date,datetime,tzinfo
import requests
from requests.auth import HTTPBasicAuth
import datetime

#transport = HttpAuthenticated(username='root',password='PeterChen123!')
transport = HttpAuthenticated(username='root',password='Private123!')
#WSDL_URL= "https://172.27.126.110/nbapi/search/?wsdl"
WSDL_URL= "https://kml-lnx2/nbapi/search/?wsdl"
client = Client(WSDL_URL, faults=False,cachingpolicy=1,location=WSDL_URL,transport=transport)


print(client)
deviceType="cgr1000"
string_start_date = "2015-03-25 12:00:00.78200"
string_end_date = "2016-03-26 12:00:00.78200"
startTime = datetime.datetime.strptime(string_start_date, "%Y-%m-%d %H:%M:%S.%f")
endTime= datetime.datetime.strptime(string_end_date, "%Y-%m-%d %H:%M:%S.%f")
count=5
offset =0
searchthis = client.service.getDeletedDevices(deviceType,startTime,endTime)
print(searchthis)
```

Results

```
Suds ( https://fedorahosted.org/suds/ )  version: 0.6

Service ( SearchWebService ) tns="http://search.nbapi.cgms.cisco.com/"
   Prefixes (1)
      ns0 = "http://search.nbapi.cgms.cisco.com/"
   Ports (1):
      (SearchWebServicePort)
         Methods (8):
            findEidsForIpAddresses(xs:string[] ipAddresses)
           findEidsForIpAddressesByDeviceType(xs:string deviceType, xs:string[] ipAddresses)

            getDeletedDevices(xs:string deviceType, xs:dateTime startTime, xs:dateTime
endTime)
            getDeviceDetails(xs:string query, xs:string[] deviceIds, xs:string queryId,
xs:long count, xs:long offset)
            getGroups(xs:string groupType)
           getMetricHistory(xs:string query, xs:string[] deviceIds, xs:dateTime startTime,
 xs:dateTime endTime, xs:string[] metricIds, xs:string rollupInterval, xs:string
rollupFunction, xs:string queryId, xs:long count, xs:long offset)
            getUpdatedDeviceDetails(xs:string query, xs:dateTime startTime, xs:dateTime
endTime, xs:string queryId, xs:long count, xs:long offset)
          searchDevices(xs:string query, xs:string[] fieldNames, xs:string queryId, xs:long
 count, xs:long offset)
         Types (36):
            Exception
            address
            asset
            deletedDevice
            deletedDeviceQueryResult
            device
            deviceDetail
```

```
            deviceDetailQueryResult
            deviceQueryResult
            eidsForIpAddressesResult
            findEidsForIpAddresses
            findEidsForIpAddressesByDeviceType
            findEidsForIpAddressesByDeviceTypeResponse
            findEidsForIpAddressesResponse
            getDeletedDevices
            getDeletedDevicesResponse
            getDeviceDetails
            getDeviceDetailsResponse
            getGroups
            getGroupsResponse
            getMetricHistory
            getMetricHistoryResponse
            getUpdatedDeviceDetails
            getUpdatedDeviceDetailsResponse
            group
            groupQueryResult
            interface
            metricHistoryQueryResult
            metricValue
            queryResult
            queryStatus
            route
            searchDevices
            searchDevicesResponse
            updatedDeviceDetail
            updatedDeviceDetailQueryResult


(200, (deletedDeviceQueryResult){
   queryStatus = "SUCCEEDED"
   deletedDevices[] =
      (deletedDevice){
         deleteDate = 2015-03-25 23:00:44+00:00
         deviceType = "cgr1000"
         eid = "CGR1240/K9+JSJ200201"
      },
      (deletedDevice){
         deleteDate = 2015-03-25 23:00:44+00:00
         deviceType = "cgr1000"
         eid = "CGR1240/K9+JSJ200202"
      },
})
```

getUpdatedDeviceDetails

This call lets the client retrieve the following information of devices updated within a specified time period:

- key
- eid
- configGroup
- event
- firmwareGroup
- issues

- Labels list
- meshDeviceOps
- workorder
- Updated Properties list
- Updated Metrics list
- Updated Interfaces list
- Updated Routes list
- Updated Addresses list
- Deleted Interfaces list
- Deleted Routes list
- Deleted Addresses list

Prototype

```
UpdatedDeviceDetailQueryResult getUpdatedDeviceDetails
(string query, date startTime, date endTime, string queryId, long count, long offset)
```

Parameters

Table 107: getUpdatedDeviceDetails Parameters

Parameter	Type	Description
query	string	Search query.
startTime	date	Starting UTC date and time for the updated interval. This is a mandatory parameter. It cannot be null. Format is YYYY-MM-DDThh:mm:ss (for example, 2015-03-25T00:00:00).
endTime	date	Ending UTC date and time for the updated interval. This is a mandatory parameter. It cannot be null. Format is YYYY-MM-DDThh:mm:ss. endTime is exclusive. It can be used as the startTime for the next call without retrieving repeated records.
queryId	string	The ID of the initial query used in subsequent calls to retrieve remaining records.
count	long	Number of results to retrieve. Valid range is 1–40000.
offset	long	Position of the first result. Valid values are >= 0.

This call returns the records of devices that match the specified deviceType and have delete time >= startTime and < endTime.

Results

The following table lists the device details returned by this call. The return message, UpdatedDeviceDetailQueryResult, is a list of UpdatedDeviceDetail records.

Table 108: getUpdatedDeviceDetails Results

Parameter	Type	Description
configGroup	string	The name of the configuration group assigned to the device.
deletedAddresses	list<string>	Deleted IP addresses.
deletedInterfaces	list<string>	Deleted interface names.
deletedRoutes	list<string>	Deleted route names.
eid	string	Device string ID.
event	string	Search for events based on device type, event name, event time, and event severity.
firmwareGroup	string	The name of the firmware group assigned to the device.
issues	string	Call searches for issues in the IoT FND database based on the provided criteria that return a list of those issues.
key	long	A long value associated with the device.
meshDeviceOps	string	Allows client applications to start and stop firmware uploads to a group of mesh devices, check their firmware upload status, obtain firmware information by firmware group, set up a backup firmware image, and schedule a firmware reload.
workorder	string	Provides seven primary action calls to cloud-based management services.
updatedAddresses	list<address>	The list of addresses of the device updated during the specified time interval.
updatedInterfaces	list<interface>	The interface of the device updated during the specified time interval.
updatedMetrics	map<string, double>	The metrics of the device updated during the specified time interval.
updatedProperties	map<string, string>	The properties of the device updated during the specified time interval.
updatedRoutes	list<route>	The routes of the device updated during the specified time interval.

See Table 101: Address Attributes, on page 110, Table 102: Interface Attributes , on page 110, and Table 103: Route Attributes , on page 110 for information on the Address, Interface, and Route attributes.

Example

```
search-client updatedDeviceDetails https://kml-lnx2/nbapi/search?wsdl "deviceCategory=router"
 "2015-03-25 12:00:00" "2015-03-26 12:00:00" 5 0
search-client updatedDeviceDetails https://kml-lnx2/nbapi/search?wsdl "deviceCategory=router"
 "2015-03-26 00:00:00" "2015-03-26 00:20:00" 2 0
```

Example SOAP Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sear="http://search.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
```

```
      <sear:getUpdatedDeviceDetails>
         <query>deviceCategory=router</query>
         <startTime>2015-03-25T23:00:00</startTime>
         <endTime>2015-03-26T00:00:00</endTime>
         <count>1</count>
         <offset>0</offset>
      </sear:getUpdatedDeviceDetails>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"/>
   <soap:Body>
     <ns2:getUpdatedDeviceDetailsResponse xmlns:ns2="http://search.nbapi.cgms.cisco.com/">

         <updatedDeviceDetailQueryResult>
            <queryId>E1EA07CE9BE5536A883469707FCA602E</queryId>
            <queryStatus>SUCCEEDED</queryStatus>
            <updatedDeviceDetails>
               <configGroup>default-c800</configGroup>
               <eid>C819HGW-S-A-K9+FTX174685VB</eid>
               <firmwareGroup>default-c800</firmwareGroup>
               <key>270000</key>
               <updatedInterfaces>
                  <index>1</index>
                  <key>290009</key>
                  <metrics>
                     <entry>
                        <key>rxSpeed</key>
                        <value>3790.8790113658024</value>
                     </entry>
                     <entry>
                        <key>utilBytes</key>
                        <value>8.40631367E8</value>
                     </entry>
                     <entry>
                        <key>inBytes</key>
                        <value>7.97480016E8</value>
                     </entry>
                     <entry>
                        <key>txSpeed</key>
                        <value>210.51087003645023</value>
                     </entry>
                     <entry>
                        <key>outBytes</key>
                        <value>4.3151351E7</value>
                     </entry>
                     <entry>
                        <key>txDrops</key>
                        <value>0.0</value>
                     </entry>
                  </metrics>
                  <name>GigabitEthernet0</name>
                  <properties>
                     <entry>
                        <key>physAddress</key>
                        <value>c025.5c08.e3f5</value>
                     </entry>
                  </properties>
               </updatedInterfaces>
                        .
```

```
            .
            .
<updatedMetrics>
   <entry>
      <key>cellularBandwidth</key>
      <value>0.0</value>
   </entry>
   <entry>
      <key>cellularTxSpeed</key>
      <value>0.0</value>
   </entry>
   <entry>
      <key>ethernetTxSpeed</key>
      <value>516.6779073252414</value>
   </entry>
   <entry>
      <key>cellularBwPerCycle</key>
      <value>0.0</value>
   </entry>
   <entry>
      <key>ethernetTxDrops</key>
      <value>0.0</value>
   </entry>
   <entry>
      <key>uptime</key>
      <value>1650960.0</value>
   </entry>
   <entry>
      <key>cellularRxSpeed</key>
      <value>0.0</value>
   </entry>
   <entry>
      <key>cellularEcio</key>
      <value>-11.0</value>
   </entry>
   <entry>
      <key>ethernetRxSpeed</key>
      <value>3790.8790113658024</value>
   </entry>
   <entry>
      <key>cellConnectTime</key>
      <value>93.0</value>
   </entry>
   <entry>
      <key>cellularRssi</key>
      <value>-117.0</value>
   </entry>
</updatedMetrics>
<updatedProperties>
   <entry>
      <key>runningFirmwareImageId</key>
      <value>null</value>
   </entry>
   <entry>
      <key>lastUpdate</key>
      <value>2015-03-25 23:29:19.0</value>
   </entry>
   <entry>
      <key>lng</key>
      <value>-26.9007</value>
   </entry>
   <entry>
      <key>reloadFirmwareVersion</key>
```

```
   <value>null</value>
</entry>
<entry>
   <key>reloadDate</key>
   <value>null</value>
</entry>
<entry>
   <key>backupFirmwareVersion</key>
   <value>null</value>
</entry>
<entry>
   <key>slot4FirmwareImageId</key>
   <value>null</value>
</entry>
<entry>
   <key>deviceType</key>
   <value>c800</value>
</entry>
<entry>
   <key>alt</key>
   <value>21</value>
</entry>
<entry>
   <key>name</key>
   <value>C819HGW-S-A-K9+FTX174685VB</value>
</entry>
<entry>
   <key>configInSync</key>
   <value>false</value>
</entry>
<entry>
   <key>runningFirmwareVersion</key>
   <value>15.5(0.23)T</value>
</entry>
<entry>
   <key>hardwareId</key>
   <value>null</value>
</entry>
<entry>
   <key>vid</key>
   <value>null</value>
</entry>
<entry>
   <key>lat</key>
   <value>26.0197</value>
</entry>
<entry>
   <key>slot5FirmwareImageId</key>
   <value>null</value>
</entry>
<entry>
   <key>sn</key>
   <value>FTX174685VB</value>
</entry>
<entry>
   <key>backupFirmwareImageId</key>
   <value>null</value>
</entry>
<entry>
   <key>status</key>
   <value>up</value>
</entry>
<entry>
   <key>hostname</key>
```

```
                        <value>kit-819</value>
                     </entry>
                     <entry>
                        <key>pid</key>
                        <value>C819HGW-S-A-K9</value>
                     </entry>
                     <entry>
                        <key>lastHeard</key>
                        <value>2015-03-25 23:34:18.0</value>
                     </entry>
                     <entry>
                        <key>mapLevel</key>
                        <value>1</value>
                     </entry>
                     <entry>
                        <key>ip</key>
                        <value>172.27.161.82</value>
                     </entry>
                     <entry>
                        <key>downloadFirmwareVersion</key>
                        <value>null</value>
                     </entry>
                     <entry>
                        <key>slot6FirmwareImageId</key>
                        <value>null</value>
                     </entry>
                     <entry>
                        <key>geoHash</key>
                        <value>eksu5bex8r8hrfk9942g4</value>
                     </entry>
                     <entry>
                        <key>downloadFirmwareImageId</key>
                        <value>null</value>
                     </entry>
                  </updatedProperties>
               </updatedDeviceDetails>
            </updatedDeviceDetailQueryResult>
         </ns2:getUpdatedDeviceDetailsResponse>
      </soap:Body>
</soap:Envelope>
```

Example SOAP Request using event

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"xmlns:even="http://event.nbapi.cgms.cisco.com/">

 <soapenv:Header/>
 <soapenv:Body>
  <even:searchEvents>
   <!--Optional:-->
   <query>?</query>
   <!--Optional:-->
   <count>?</count>
   <!--Optional:-->
   <offset>?</offset>
  </even:searchEvents>
 </soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Request using issues

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"xmlns:iss="http://issues.nbapi.cgms.cisco.com/">

 <soapenv:Header/>
 <soapenv:Body>
  <iss:searchIssues>
   <!--Optional:-->

   <query>deviceType:cgr1000 issueStatus:open</query>
   <!--Optional:-->
   <count>2</count>
   <!--Optional:-->
   <offset>0</offset>
  </iss:searchIssues>
 </soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Request using meshDeviceOps

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mes="http://meshDeviceOps.nbapi.cgms.cisco.com/">
 <soapenv:Header/>
 <soapenv:Body>
  <mes:getFirmwareImageInfoList>
   <!--Optional:-->
   <firmwareGroup>?</firmwareGroup>
  </mes:getFirmwareImageInfoList>
 </soapenv:Body>
</soapenv:Envelope>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mes="http://meshDeviceOps.nbapi.cgms.cisco.com/">
 <soapenv:Header/>
 <soapenv:Body>
  <mes:getFirmwareUploadStatus>
   <!--Optional:-->
   <firmwareGroup>?</firmwareGroup>
  </mes:getFirmwareUploadStatus>
 </soapenv:Body>
</soapenv:Envelope>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mes="http://meshDeviceOps.nbapi.cgms.cisco.com/">
 <soapenv:Header/>
 <soapenv:Body>
  <mes:scheduleReload>
   <!--Optional:-->
   <firmwareGroup>?</firmwareGroup>
   <!--Optional:-->
   <firmwareImageName>?</firmwareImageName>
   <!--Optional:-->
   <reloadGmtTime>?</reloadGmtTime>
  </mes:scheduleReload>

 </soapenv:Body>
</soapenv:Envelope>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mes="http://meshDeviceOps.nbapi.cgms.cisco.com/">
 <soapenv:Header/>
 <soapenv:Body>
  <mes:setBackupFirmwareImage>
   <!--Optional:-->
```

```
   <firmwareGroup>?</firmwareGroup>
   <!--Optional:-->
   <firmwareImageName>?</firmwareImageName>
  </mes:setBackupFirmwareImage>
 </soapenv:Body>
</soapenv:Envelope>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mes="http://meshDeviceOps.nbapi.cgms.cisco.com/">
 <soapenv:Header/>
 <soapenv:Body>
  <mes:startUpload>
   <!--Optional:-->
   <firmwareGroup>?</firmwareGroup>
   <!--Optional:-->
   <firmwareImageName>?</firmwareImageName>
  </mes:startUpload>
 </soapenv:Body>
</soapenv:Envelope>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mes="http://meshDeviceOps.nbapi.cgms.cisco.com/">
 <soapenv:Header/>
 <soapenv:Body>
  <mes:stopUpload>
   <!--Optional:-->
   <firmwareGroup>?</firmwareGroup>
  </mes:stopUpload>
 </soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Request using queryId

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sear="http://search.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <sear:getUpdatedDeviceDetails>
         <queryId>E1EA07CE9BE5536A883469707FCA602E</queryId>
      </sear:getUpdatedDeviceDetails>
   </soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Request using workorder

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wor="http://workorder.nbapi.cgms.cisco.com/">
 <soapenv:Header/>
 <soapenv:Body>
  <wor:RequestSignedAuthorization>
   <!--Optional:-->
   <technicianUserName>?</technicianUserName>
   <!--Optional:-->
   <workOrderNumber>?</workOrderNumber>
   <!--Optional:-->
   <appVersion>?</appVersion>
  </wor:RequestSignedAuthorization>
 </soapenv:Body>
</soapenv:Envelope>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wor="http://workorder.nbapi.cgms.cisco.com/">
 <soapenv:Header/>
 <soapenv:Body>
```

```
  <wor:RequestSignedAuthorizationWithClientKey>
   <!--Optional:-->
   <technicianUserName>?</technicianUserName>
   <!--Optional:-->
   <workOrderNumber>?</workOrderNumber>
   <!--Optional:-->

   <appVersion>?</appVersion>
   <!--Optional:-->
   <appKey>?</appKey>
  </wor:RequestSignedAuthorizationWithClientKey>
 </soapenv:Body>
</soapenv:Envelope>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wor="http://workorder.nbapi.cgms.cisco.com/">
 <soapenv:Header/>
 <soapenv:Body>
  <wor:RequestUserAuthentication>
   <!--Optional:-->
   <userAuthInfo>
    <!--Optional:-->
    <appVersion>?</appVersion>
    <!--Optional:-->
    <scriptVersion>?</scriptVersion>
   </userAuthInfo>
  </wor:RequestUserAuthentication>
 </soapenv:Body>
</soapenv:Envelope>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wor="http://workorder.nbapi.cgms.cisco.com/"> <soapenv:Header/> <soapenv:Body>
<wor:UploadServiceReport>
   <!--Zero or more repetitions:-->   <serviceStatusReport>    <!--Optional:-->
    <orderNumber>?</orderNumber>
    <!--Optional:-->
    <deviceId>?</deviceId>
    <!--Optional:-->
    <technicianUserName>?</technicianUserName>
    <!--Optional:-->
    <status>?</status>
   </serviceStatusReport>
  </wor:UploadServiceReport>
 </soapenv:Body>
</soapenv:Envelope>
```

Example Python Request

IoT FND 3.0 and NB API 3.0 work with both TLS1.0 and TLS1.2.

Note If you use TLS1.2 with SUDS package, you must use the ActivePython package available at:

http://www.activestate.com/blog/2015/11/activepython-vs-open-source-python-whats-difference

Note If you use TLS1.0 with SUDS package, you must use the regular Python package available at:

https://www.python.org/downloads/

```
#!/usr/local/bin/python
from suds.transport.https import HttpAuthenticated
from suds.client import Client
import logging
from datetime import timedelta,date,datetime,tzinfo
import requests
from requests.auth import HTTPBasicAuth
import datetime

transport = HttpAuthenticated(username='root',password='PeterChen123!')
WSDL_URL= "https://172.27.126.110/nbapi/search/?wsdl"
client = Client(WSDL_URL, faults=False,cachingpolicy=1,location=WSDL_URL,transport=transport)


print(client)
#query="deviceType:cgr1000"
query="deviceType:cgmesh"
#fieldNames = ["uptime","uplinkTxSpeed","uplinkTxDrops"]
string_start_date = "2014-09-28 20:30:55.78200"
string_end_date = "2015-03-25 20:30:55.78200"
startTime = datetime.datetime.strptime(string_start_date, "%Y-%m-%d %H:%M:%S.%f")
endTime= datetime.datetime.strptime(string_end_date, "%Y-%m-%d %H:%M:%S.%f")
#deviceId=["00173bab003c3500", "00173bab003c3501"]
count=5
offset =0
searchthis = client.service.getUpdatedDeviceDetails(query,startTime,endTime, "",count,
offset)
print(searchthis)
```

Results

```
Suds ( https://fedorahosted.org/suds/ )  version: 0.6

Service ( SearchWebService ) tns="http://search.nbapi.cgms.cisco.com/"
   Prefixes (1)
      ns0 = "http://search.nbapi.cgms.cisco.com/"
   Ports (1):
      (SearchWebServicePort)
         Methods (8):
            findEidsForIpAddresses(xs:string[] ipAddresses)
           findEidsForIpAddressesByDeviceType(xs:string deviceType, xs:string[] ipAddresses)

            getDeletedDevices(xs:string deviceType, xs:dateTime startTime, xs:dateTime
endTime)
            getDeviceDetails(xs:string query, xs:string[] deviceIds, xs:string queryId,
xs:long count, xs:long offset)
            getGroups(xs:string groupType)
           getMetricHistory(xs:string query, xs:string[] deviceIds, xs:dateTime startTime,
 xs:dateTime endTime, xs:string[] metricIds, xs:string rollupInterval, xs:string
rollupFunction, xs:string queryId, xs:long count, xs:long offset)
            getUpdatedDeviceDetails(xs:string query, xs:dateTime startTime, xs:dateTime
endTime, xs:string queryId, xs:long count, xs:long offset)
          searchDevices(xs:string query, xs:string[] fieldNames, xs:string queryId, xs:long
 count, xs:long offset)
         Types (36):
            Exception
            address
            asset
            deletedDevice
            deletedDeviceQueryResult
            device
            deviceDetail
```

```
                deviceDetailQueryResult
                deviceQueryResult
                eidsForIpAddressesResult
                findEidsForIpAddresses
                findEidsForIpAddressesByDeviceType
                findEidsForIpAddressesByDeviceTypeResponse
                findEidsForIpAddressesResponse
                getDeletedDevices
                getDeletedDevicesResponse
                getDeviceDetails
                getDeviceDetailsResponse
                getGroups
                getGroupsResponse
                getMetricHistory
                getMetricHistoryResponse
                getUpdatedDeviceDetails
                getUpdatedDeviceDetailsResponse
                group
                groupQueryResult
                interface
                metricHistoryQueryResult
                metricValue
                queryResult
                queryStatus
                route
                searchDevices
                searchDevicesResponse
                updatedDeviceDetail
                updatedDeviceDetailQueryResult


(200, (updatedDeviceDetailQueryResult){
   queryId = "60EF7F6ACFD8EE823AC2B8F83E545BC9"
   queryStatus = "SUCCEEDED"
   updatedDeviceDetails[] =
      (updatedDeviceDetail){
         configGroup = "default-cgmesh"
         eid = "00000000001"
         firmwareGroup = "default-cgmesh"
         key = 261628
         updatedMetrics = ""
         updatedProperties =
            (updatedProperties){
               entry[] =
                  (entry){
                     key = "runningFirmwareImageId"
                     value = "null"
                  },
                  (entry){
                     key = "lastUpdate"
                     value = "2015-03-12 21:28:48.0"
                  },
                  (entry){
                     key = "lng"
                     value = "1000.0"
                  },
                  (entry){
                     key = "reloadFirmwareVersion"
                     value = "null"
                  },
                  (entry){
                     key = "reloadDate"
                     value = "null"
                  },
```

```
(entry){
   key = "function"
   value = "meter"
},
(entry){
   key = "backupFirmwareVersion"
   value = "null"
},
(entry){
   key = "slot4FirmwareImageId"
   value = "null"
},
(entry){
   key = "deviceType"
   value = "cgmesh"
},
(entry){
   key = "alt"
   value = "null"
},
(entry){
   key = "name"
   value = "00000000001"
},
(entry){
   key = "configInSync"
   value = "false"
},
(entry){
   key = "runningFirmwareVersion"
   value = "null"
},
(entry){
   key = "hardwareId"
   value = "null"
},
(entry){
   key = "vid"
   value = "null"
},
(entry){
   key = "lat"
   value = "1000.0"
},
(entry){
   key = "slot5FirmwareImageId"
   value = "null"
},
(entry){
   key = "sn"
   value = "null"
},
(entry){
   key = "backupFirmwareImageId"
   value = "null"
},
(entry){
   key = "status"
   value = "unheard"
},
(entry){
   key = "hostname"
   value = "null"
},
```

```
                (entry){
                   key = "pid"
                   value = "null"
                },
                (entry){
                   key = "lastHeard"
                   value = "null"
                },
                (entry){
                   key = "mapLevel"
                   value = "16"
                },
                (entry){
                   key = "ip"
                   value = "2.2.2.11"
                },
                (entry){
                   key = "downloadFirmwareVersion"
                   value = "null"
                },
                (entry){
                   key = "slot6FirmwareImageId"
                   value = "null"
                },
                (entry){
                  key = "geoHash"
                   value = "null"
                },
                (entry){
                   key = "downloadFirmwareImageId"
                   value = "null"
                },
          }
},
(updatedDeviceDetail){
   configGroup = "default-cgmesh"
   eid = "00000000002"
   firmwareGroup = "default-cgmesh"
   key = 261627
   updatedMetrics = ""
   updatedProperties =
      (updatedProperties){
         entry[] =
            (entry){
               key = "runningFirmwareImageId"
               value = "null"
            },
            (entry){
               key = "lastUpdate"
               value = "2015-03-12 21:28:48.0"
            },
            (entry){
               key = "lng"
               value = "1000.0"
            },
            (entry){
               key = "reloadFirmwareVersion"
               value = "null"
            },
            (entry){
               key = "reloadDate"
               value = "null"
            },
            (entry){
```

```
      key = "function"
      value = "meter"
   },
   (entry){
      key = "backupFirmwareVersion"
      value = "null"
   },
   (entry){
      key = "slot4FirmwareImageId"
      value = "null"
   },
   (entry){
      key = "deviceType"
      value = "cgmesh"
   },
   (entry){
      key = "alt"
      value = "null"
   },
   (entry){
      key = "name"
      value = "00000000002"
   },
   (entry){
      key = "configInSync"
      value = "false"
   },
   (entry){
      key = "runningFirmwareVersion"
      value = "null"
   },
   (entry){
      key = "hardwareId"
      value = "null"
   },
   (entry){
      key = "vid"
      value = "null"
   },
   (entry){
      key = "lat"
      value = "1000.0"
   },
   (entry){
      key = "slot5FirmwareImageId"
      value = "null"
   },
   (entry){
      key = "sn"
      value = "null"
   },
   (entry){
      key = "backupFirmwareImageId"
      value = "null"
   },
   (entry){
      key = "status"
      value = "unheard"
   },
   (entry){
      key = "hostname"
      value = "null"
   },
   (entry){
```

```
                  key = "pid"
                  value = "null"
               },
               (entry){
                  key = "lastHeard"
                  value = "null"
               },
               (entry){
                  key = "mapLevel"
                  value = "16"
               },
               (entry){
                  key = "ip"
                  value = "2.2.2.12"
               },
               (entry){
                  key = "downloadFirmwareVersion"
                  value = "null"
               },
               (entry){
                  key = "slot6FirmwareImageId"
                  value = "null"
               },
               (entry){
                  key = "geoHash"
                  value = "null"
               },
               (entry){
                  key = "downloadFirmwareImageId"
                  value = "null"
               },
         }
},
(updatedDeviceDetail){
   configGroup = "default-cgmesh"
   eid = "0007810800a80bfe"
   firmwareGroup = "default-cgmesh"
   key = 20127
   updatedAddresses[] =
      (address){
         address = "fe80:0:0:0:0:0:0:1"
         addressType = "IPV6"
         key = 20001004
         prefixLength = 64
      },
      (address){
         address = "2010:dead:beef:cafe:0:8108:a8:bfe"
         addressType = "IPV6"
         key = 20001005
         prefixLength = 64
      },
      (address){
         address = "fe80:0:0:0:207:8108:a8:bfe"
         addressType = "IPV6"
         key = 20001006
         prefixLength = 64
      },
      (address){
         address = "0:0:0:0:0:0:0:1"
         addressType = "IPV6"
         key = 20001007
         prefixLength = 128
      },
   updatedInterfaces[] =
```

```
   (interface){
      addresses[] =
         (address){
            address = "0:0:0:0:0:0:0:1"
            addressType = "IPV6"
            key = 20001007
            prefixLength = 128
         },
      index = 1
      key = 50004
      metrics = ""
      name = "lo"
      properties = ""
   },
   (interface){
      addresses[] =
         (address){
            address = "2010:dead:beef:cafe:0:8108:a8:bfe"
            addressType = "IPV6"
            key = 20001005
            prefixLength = 64
         },
         (address){
            address = "fe80:0:0:0:207:8108:a8:bfe"
            addressType = "IPV6"
            key = 20001006
            prefixLength = 64
         },
      index = 2
      key = 50005
      metrics = ""
      name = "lowpan"
      properties =
         (properties){
            entry[] =
               (entry){
                  key = "physAddress"
                  value = "0007810800a80bfe"
               },
         }
   },
   (interface){
      addresses[] =
         (address){
            address = "fe80:0:0:0:0:0:0:1"
            addressType = "IPV6"
            key = 20001004
            prefixLength = 64
         },
      index = 3
      key = 50006
      metrics = ""
      name = "ppp"
      properties =
         (properties){
            entry[] =
               (entry){
                 key = "physAddress"
                  value = "0007810800a80bfe"
               },
         }
   },
updatedMetrics = ""
updatedProperties =
```

```
(updatedProperties){
   entry[] =
      (entry){
         key = "runningFirmwareImageId"
         value = "null"
      },
      (entry){
         key = "lastUpdate"
         value = "2015-03-12 21:28:48.0"
      },
      (entry){
         key = "meshPanid"
         value = "26930"
      },
      (entry){
         key = "lng"
         value = "-88.66548868"
      },
      (entry){
         key = "reloadFirmwareVersion"
         value = "null"
      },
      (entry){
         key = "activeLinkType"
         value = "RF"
      },
      (entry){
         key = "reloadDate"
         value = "1970-01-01 00:00:00.0"
      },
      (entry){
         key = "backupFirmwareVersion"
         value = "null"
      },
      (entry){
         key = "slot4FirmwareImageId"
         value = "null"
      },
      (entry){
         key = "deviceType"
         value = "cgmesh"
      },
      (entry){
         key = "alt"
         value = "null"
      },
      (entry){
         key = "name"
         value = "0007810800a80bfe"
      },
      (entry){
         key = "configInSync"
         value = "true"
      },
      (entry){
         key = "runningFirmwareVersion"
         value = "5.5.62"
      },
      (entry){
         key = "hardwareId"
         value = "null"
      },
      (entry){
         key = "vid"
```

```
                  value = "3.1"
               },
               (entry){
                  key = "lat"
                  value = "43.56901132"
               },
               (entry){
                  key = "slot5FirmwareImageId"
                  value = "null"
               },
               (entry){
                  key = "sn"
                  value = "0007810800A80BFE"
               },
               (entry){
                  key = "backupFirmwareImageId"
                  value = "null"
               },
               (entry){
                  key = "status"
                  value = "outage"
               },
               (entry){
                  key = "hostname"
                  value = "null"
               },
               (entry){
                  key = "pid"
                  value = "OWCM"
               },
               (entry){
                  key = "lastHeard"
                  value = "2015-02-24 18:59:52.0"
               },
               (entry){
                  key = "mapLevel"
                  value = "16"
               },
               (entry){
                  key = "ip"
                  value = "2010:dead:beef:cafe:0:8108:a8:bfe"
               },
               (entry){
                  key = "downloadFirmwareVersion"
                  value = "null"
               },
              (entry){
                  key = "slot6FirmwareImageId"
                  value = "null"
               },
               (entry){
                  key = "geoHash"
                  value = "dp8zy77zbk7u02v6jxzgn"
               },
               (entry){
                  key = "downloadFirmwareImageId"
                  value = "null"
               },
               (entry){
                  key = "previousMeshPanid"
                  value = "2015"
               },
      }
   updatedRoutes[] =
```

```
        (route){
           index = 1
           key = 100005
           metrics = ""
           properties =
              (properties){
                 entry[] =
                    (entry){
                       key = "destAddressType"
                       value = "2"
                    },
                    (entry){
                       key = "nextHopAddressType"
                       value = "4"
                    },
                    (entry){
                       key = "nextHopAddress"
                       value = "fe80:0:0:0:207:8108:bf:a053"
                    },
                    (entry){
                       key = "destAddress"
                       value = "0:0:0:0:0:0:0:0"
                    },
                    (entry){
                       key = "prefixLength"
                       value = "0"
                    },
              }
        },
},
(updatedDeviceDetail){
   configGroup = "default-cgmesh"
   eid = "0007810800a80d40"
   firmwareGroup = "default-cgmesh"
   key = 20126
   updatedAddresses[] =
      (address){
         address = "2010:dead:beef:cafe:0:8108:a8:d40"
         addressType = "IPV6"
         key = 20001009
         prefixLength = 64
      },
      (address){
         address = "fe80:0:0:0:0:0:0:1"
         addressType = "IPV6"
         key = 20001010
         prefixLength = 64
      },
      (address){
         address = "0:0:0:0:0:0:0:1"
         addressType = "IPV6"
         key = 20001011
         prefixLength = 128
      },
      (address){
         address = "fe80:0:0:0:207:8108:a8:d40"
         addressType = "IPV6"
         key = 20001008
         prefixLength = 64
      },
   updatedInterfaces[] =
      (interface){
         addresses[] =
            (address){
```

```
                address = "0:0:0:0:0:0:0:1"
                addressType = "IPV6"
               key = 20001011
                prefixLength = 128
             },
          index = 1
          key = 50007
          metrics = ""
          name = "lo"
          properties = ""
       },
       (interface){
          addresses[] =
             (address){
                address = "2010:dead:beef:cafe:0:8108:a8:d40"
                addressType = "IPV6"
                key = 20001009
                prefixLength = 64
             },
             (address){
                address = "fe80:0:0:0:207:8108:a8:d40"
                addressType = "IPV6"
                key = 20001008
                prefixLength = 64
             },
          index = 2
          key = 50008
          metrics = ""
          name = "lowpan"
          properties =
             (properties){
                entry[] =
                   (entry){
                      key = "physAddress"
                      value = "0007810800a80d40"
                   },
             }
       },
       (interface){
          addresses[] =
             (address){
                address = "fe80:0:0:0:0:0:0:1"
                addressType = "IPV6"
                key = 20001010
                prefixLength = 64
             },
          index = 3
          key = 50009
          metrics = ""
          name = "ppp"
          properties =
             (properties){
                entry[] =
                   (entry){
                      key = "physAddress"
                      value = "0007810800a80d40"
                   },
             }
       },
    updatedMetrics = ""
    updatedProperties =
       (updatedProperties){
          entry[] =
             (entry){
```

```
      key = "runningFirmwareImageId"
      value = "null"
   },
   (entry){
      key = "lastUpdate"
      value = "2015-03-12 21:28:48.0"
   },
   (entry){
      key = "meshPanid"
      value = "26930"
   },
   (entry){
      key = "lng"
      value = "-88.96548868"
   },
   (entry){
      key = "reloadFirmwareVersion"
      value = "null"
   },
   (entry){
      key = "activeLinkType"
      value = "RF"
   },
   (entry){
      key = "reloadDate"
      value = "1970-01-01 00:00:00.0"
   },
   (entry){
      key = "backupFirmwareVersion"
      value = "null"
   },
   (entry){
      key = "slot4FirmwareImageId"
      value = "null"
   },
   (entry){
      key = "deviceType"
      value = "cgmesh"
   },
   (entry){
      key = "alt"
      value = "null"
   },
   (entry){
      key = "name"
      value = "0007810800a80d40"
   },
   (entry){
      key = "configInSync"
      value = "true"
   },
   (entry){
      key = "runningFirmwareVersion"
      value = "5.5.42"
   },
   (entry){
      key = "hardwareId"
      value = "null"
   },
   (entry){
      key = "vid"
      value = "3.1"
   },
   (entry){
```

```
                  key = "lat"
                  value = "43.46901132"
               },
               (entry){
                  key = "slot5FirmwareImageId"
                  value = "null"
               },
               (entry){
                  key = "sn"
                  value = "0007810800A80D40"
               },
               (entry){
                  key = "backupFirmwareImageId"
                  value = "null"
               },
               (entry){
                  key = "status"
                  value = "outage"
               },
               (entry){
                  key = "hostname"
                  value = "null"
               },
               (entry){
                  key = "pid"
                  value = "OWCM"
               },
               (entry){
                  key = "lastHeard"
                  value = "2015-02-23 23:27:00.0"
               },
               (entry){
                  key = "mapLevel"
                  value = "16"
               },
               (entry){
                  key = "ip"
                  value = "2010:dead:beef:cafe:0:8108:a8:d40"
               },
               (entry){
                  key = "downloadFirmwareVersion"
                  value = "5.5.59"
               },
               (entry){
                  key = "slot6FirmwareImageId"
                  value = "null"
               },
               (entry){
                  key = "geoHash"
                  value = "dp8xr9333g58c96cp1ptr"
               },
               (entry){
                  key = "downloadFirmwareImageId"
                  value = "null"
               },
               (entry){
                  key = "previousMeshPanid"
                  value = "2015"
               },
      }
   updatedRoutes[] =
      (route){
         index = 1
         key = 100006
```

```
            metrics = ""
            properties =
               (properties){
                  entry[] =
                     (entry){
                        key = "destAddressType"
                        value = "2"
                     },
                     (entry){
                        key = "nextHopAddressType"
                        value = "4"
                     },
                     (entry){
                        key = "nextHopAddress"
                        value = "fe80:0:0:0:207:8108:cc:e50b"
                     },
                     (entry){
                        key = "destAddress"
                        value = "0:0:0:0:0:0:0:0"
                     },
                     (entry){
                        key = "prefixLength"
                        value = "0"
                     },
               }
         },
   },
   (updatedDeviceDetail){
      configGroup = "default-cgmesh"
      eid = "0007810800bf38e7"
      firmwareGroup = "default-cgmesh"
      key = 20130
      updatedAddresses[] =
         (address){
            address = "fe80:0:0:0:207:8108:bf:38e7"
            addressType = "IPV6"
            key = 104001023
            prefixLength = 64
         },
         (address){
            address = "fe80:0:0:0:0:0:0:1"
            addressType = "IPV6"
            key = 104001024
            prefixLength = 64
         },
         (address){
            address = "2011:dead:beef:cafe:0:8108:bf:38e7"
            addressType = "IPV6"
            key = 104001025
            prefixLength = 64
         },
         (address){
            address = "0:0:0:0:0:0:0:1"
            addressType = "IPV6"
            key = 104001026
            prefixLength = 128
         },
      updatedInterfaces[] =
         (interface){
            addresses[] =
               (address){
                  address = "0:0:0:0:0:0:0:1"
                  addressType = "IPV6"
                  key = 104001026
```

```
                prefixLength = 128
             },
          index = 1
          key = 250016
          metrics = ""
          name = "lo"
          properties = ""
       },
       (interface){
          addresses[] =
             (address){
                address = "fe80:0:0:0:207:8108:bf:38e7"
                addressType = "IPV6"
                key = 104001023
                prefixLength = 64
             },
             (address){
                address = "2011:dead:beef:cafe:0:8108:bf:38e7"
                addressType = "IPV6"
                key = 104001025
                prefixLength = 64
             },
          index = 2
          key = 250017
          metrics = ""
          name = "lowpan"
          properties =
             (properties){
                entry[] =
                   (entry){
                      key = "physAddress"
                      value = "0007810800bf38e7"
                   },
             }
       },
       (interface){
          addresses[] =
             (address){
                address = "fe80:0:0:0:0:0:0:1"
                addressType = "IPV6"
                key = 104001024
                prefixLength = 64
             },
          index = 3
          key = 250018
          metrics = ""
          name = "ppp"
          properties =
             (properties){
                entry[] =
                   (entry){
                      key = "physAddress"
                      value = "0007810800bf38e7"
                   },
             }
       },
    updatedMetrics =
       (updatedMetrics){
          entry[] =
             (entry){
                key = "nodeLocalTime"
                value = 1427113644.0
             },
       }
```

```
updatedProperties =
   (updatedProperties){
      entry[] =
         (entry){
            key = "runningFirmwareImageId"
            value = "null"
         },
         (entry){
            key = "lastUpdate"
            value = "2015-03-23 12:27:24.0"
         },
         (entry){
            key = "meshPanid"
            value = "118"
         },
         (entry){
            key = "lng"
            value = "-87.71548868"
         },
         (entry){
            key = "reloadFirmwareVersion"
            value = "null"
         },
         (entry){
            key = "reloadDate"
            value = "null"
         },
         (entry){
            key = "backupFirmwareVersion"
            value = "null"
         },
         (entry){
            key = "slot4FirmwareImageId"
            value = "null"
         },
         (entry){
            key = "deviceType"
            value = "cgmesh"
         },
         (entry){
            key = "alt"
            value = "null"
         },
         (entry){
            key = "name"
            value = "0007810800bf38e7"
         },
         (entry){
            key = "configInSync"
            value = "false"
         },
         (entry){
            key = "runningFirmwareVersion"
            value = "5.5.42"
         },
         (entry){
            key = "hardwareId"
            value = "null"
         },
         (entry){
            key = "vid"
            value = "3.1"
         },
         (entry){
```

```
   key = "lat"
   value = "42.16901532"
},
(entry){
   key = "slot5FirmwareImageId"
   value = "null"
},
(entry){
   key = "sn"
   value = "0007810800BF38E7"
},
(entry){
   key = "backupFirmwareImageId"
   value = "null"
},
(entry){
   key = "status"
   value = "down"
},
(entry){
   key = "hostname"
   value = "null"
},
(entry){
   key = "pid"
   value = "OWCM"
},
(entry){
   key = "lastHeard"
   value = "2015-03-23 12:26:00.0"
},
(entry){
   key = "mapLevel"
   value = "16"
},
(entry){
   key = "ip"
   value = "2011:dead:beef:cafe:0:8108:bf:38e7"
},
(entry){
   key = "downloadFirmwareVersion"
   value = "null"
},
(entry){
   key = "meshSsid"
   value = "sjklin-internal"
},
(entry){
   key = "slot6FirmwareImageId"
   value = "null"
},
(entry){
   key = "meterCert"
   value = "host/smartmeter"
},
(entry){
   key = "geoHash"
   value = "dp3xguxs2fpne6gfgtjyw"
},
(entry){
   key = "downloadFirmwareImageId"
   value = "null"
},
(entry){
```

```
                        key = "meshTxPower"
                        value = "-34"
                     },
                     (entry){
                        key = "previousMeshPanid"
                        value = "null"
                     },
                     (entry){
                        key = "meshSecMode"
                        value = "1"
                     },
                  }
      },
})
```

getMetricHistory

This call lets the client retrieve the historical metric values saved in the IoT FND database by specifying the following:

- Single device
- List of devices
- Query that returns devices

You can also specify a rollup interval from the following:

- none
- hour
- day

And a rollup function from the following:

- avg
- max
- min

Prototype

```
MetricHistoryQueryResult getMetricHistory
(string query, list<string> deviceIds, date startTime,
            date endTime, list<string> metricIds, string rollupInterval,
            string rollupFunction, string queryId, long count,
            long offset)
```

Parameters

Table 109: getMetricHistory Parameters

Parameter	Type	Description
query	string	Search query string

Parameter	Type	Description
deviceIds	list<string>	List of device EIDs to retrieve.
startTime	date	Starting UTC date and time for the metric history interval; use null if not defined.
endTime	date	Ending UTC date and time for the metric history interval; use null if not defined.
metricIds	list<string>	List of metric type names to retrieve for each device.
rollupInterval	string	Valid values are none, hour, or day.
rollupFunction	string	Valid values are avg, max, or min.
queryId	string	If available results for the *query* is more than *count* , the caller can use the returned *queryId* to call the same API repeatedly. When *queryId* is provided, all other parameters are ignored.
count	long	Number of results to retrieve. Valid range is 1–40000
offset	long	Position of the first result. Valid values are >= 0.

Results

This interface returns a list of metric values, as defined in Table 110: getMetricHistory Results, on page 143.

Table 110: getMetricHistory Results

Parameter	Type	Description
eid	string	String ID of the device.
metricId	string	Metric type name.
timestamp	date	UTC timestamp of the metric value.
value	double	Value of the metric.

getMetricHistory SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sear="http://search.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <sear:getMetricHistory>
         <!--Optional:-->
         <query>deviceType:cgr1000</query>
         <!--Zero or more repetitions:-->
        <deviceIds>far_test3</deviceIds>
         <!--Optional:-->
         <startTime>null</startTime>
         <!--Optional:-->
         <endTime>null</endTime>
         <!--Zero or more repetitions:-->
         <metricIds>1</metricIds>
         <!--Optional:-->
```

```
            <rollupInterval>none</rollupInterval>
            <!--Optional:-->
            <rollupFunction>min</rollupFunction>
            <!--Optional:-->
            <queryId></queryId>
            <!--Optional:-->
            <count>5</count>
            <!--Optional:-->
            <offset>1</offset>
         </sear:getMetricHistory>
      </soapenv:Body>
   </soapenv:Envelope>
```

findEidsForIpAddresses

This call retrieves the EID of devices based on the IP address, and allows the Collection Engine to communicate to IoT FND which devices to migrate.

Prototype

```
EidsForIpAddressesResult findEidsForIpAddresses
(list<string> ipAddresses)
```

Parameters

Table 111: findEidsForIpAddresses Parameters

Parameter	Type	Description
ipAddresses	list<string>	The list of IP addresses with EIDs corresponding to devices to search for. **Note** The list is limited to a maximum of 1000 IP addresses. An error returns if more than 1000 are specified. IoT FND does not support inputs as GRE Tunnel IPv6 addresses.

Results

This call always returns a response of type EidsForIpAddressesResult. The queryStatus value defines any errors.

On success, the eidMap contains correlations between the IP address and the EID of the device with that IP address. If an IP address is not mapped to a device, those IP addresses are included in the invalidMappings value.

The following table describes the parameters in the response.

Table 112: findEidsForIpAddresses Results

Parameter	Type	Description
eidMap	map<string,string>	Map of IP addresses-to-EIDs, where the address maps to a single CGR.
invalidMappings	list<string>	List of IP addresses that did not correspond to any CGR.

findEidsForIpAddressesByDeviceType

This call retrieves the EID of devices based on the device type, and allows the Collection Engine to communicate to IoT FND which devices to migrate.

Prototype

```
EidsForIpAddressesResult findEidsForIpAddressesByDeviceType
(string deviceType, list<string> ipAddresses)
```

Parameters

Table 113: findEidsForIpAddressesByDeviceType Parameters

Parameter	Type	Description
deviceType	string	The name of the device type with EIDs corresponding to devices to search for. For example, cgr1000 only matches IP addresses against CGR devices.
ipAddresses	list<string>	Returns a list of uplink IP addresses the collection engine uses to track CGR devices. WAN or IPSEC Tunnel. **Note** The list is limited to a maximum of 1000 IP addresses. An error returns if more than 1000 are specified. IoT FND does not support inputs as GRE Tunnel IPv6 addresses.

Results

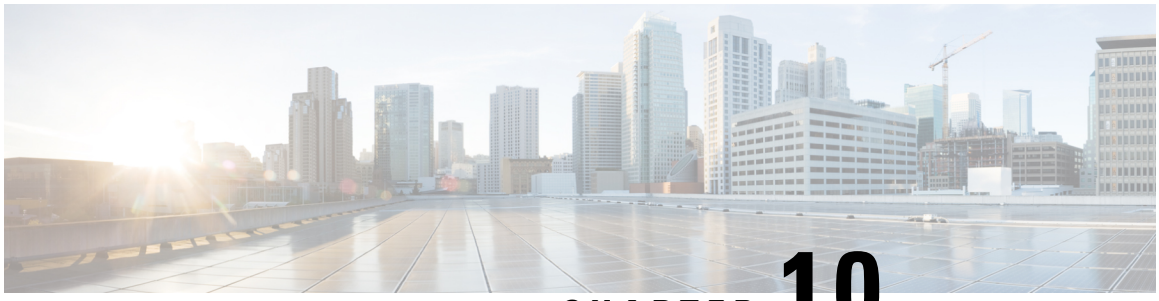
This call always returns a response of type EidsForIpAddressesResult. The queryStatus value defines any errors.

On success, the eidMap contains correlations between the IP address and the EID of the device with that IP address. If an IP address is not mapped to a device of the specified deviceType, those IP addresses are included in the invalidMappings value.

The following table describes the parameters in the response.

Table 114: findEidsForIpAddressesByDeviceType Results

Parameter	Type	Description
eidMap	map<string,string>	Map of IP addresses-to-EIDs, where the address maps to a single CGR.
invalidMappings	list<string>	List of IP addresses that did not correspond to any CGR.



CHAPTER 10

Workorders API

This chapter describes the Workorders API.

Using the Workorders API

These APIs talk with any cloud-based management services. It contains seven primary action calls sent using the SOAP envelope over HTTP(s):

https://<*server_address*> :portnumber/nbapi/workorder

Audit API Method Calls

RequestUserAuthentication

This call requests user authentication with the cloud service. The utility field technician must sign in to IoT FND before they can use the field tool. The username and password is encrypted and cached in the device for remote connections. Below is the definition for this action based on the Cisco HSL language:

Struct UserAuthInfo

```
String  appVersion ## IoT-FND application version for version control
String  scriptVersion ## IoT-FND button script version for version control
```

Enum UserAuthResult

```
“AUTH_OK” ## Authentication OK
“AUTH_OK_NO_PERMISSION” ## Authentication OK but no permission to access resources
“AUTH_OK_VER_MISMATCH” ## Authentication Fail - User/Pwd may be ok but the version is
## not expected
"AUTH_FAIL" ## NOTE:  This may never appear in the SOAP response as some
## web servers will return http status code 40x if auth fails.
## Both the 40x status code and AUTH_FAIL in the SOAP response
## are handled.
```

Struct UserAuthResponseInfo

```
UserAuthResult userAuthResult ## Authentication result
String expectAppVer ## [Optional] If the current app version
## needs to be upgraded
String expectScriptVer ## If the current script version needs to be
## upgraded
```

Struct UserAuthResponse

Input:

```
UserAuthInfo          userAuthInfo
```

Output:

```
UserAuthResponse  RequestUserAuthenticationResponse
```

Action RequestUserAuthentication

```
UserAuthResponseInfo userAuthResponse  ## Authentication result
```

SOAP XML Request Format

```
POST  /nms/fieldtool   HTTP/1.1
Host: sample.cisco.com:8080
Content-Length: nnn
Authorization: Basic bW2eFaHU5K+
SOAPAction: http://workorder.nbapi.cgms.cisco.com/RequestUserAuthentication
Connection: close
Content-type: text/xml; charset=”UTF-8”
User-Agent: Cisco Device Management Application
<?xml version=”1.0” encoding=”utf-8”?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=”http://schemas.xmlsoap.org/soap/envelope/”
xmlns:m="http://workorder.nbapi.cgms.cisco.com/">
    <SOAP-ENV:Body>
        <m:RequestUserAuthentication>
            <userAuthInfo>
                <appVersion>1.2.23.128</appVersion>
                <scriptVersion>2.3.4</scriptVersion>
            </userAuthInfo>
        </m:RequestUserAuthentication>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
HTTP/1.1   200 OK
Content-type: text/xml; charset=”UTF-8”
Content-Length: nnn
<?xml version=”1.0” encoding=”utf-8”?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=”http://schemas.xmlsoap.org/soap/envelope/”>
    <SOAP-ENV:Body>
       <m:RequestUserAuthenticationResponse xmlns:m="http://workorder.nbapi.cgms.cisco.com/">

            <userAuthResponse>
                 <userAuthResult>AUTH_OK</userAuthResult>
```

```
                <expectAppVer>1.2.25.02</expectAppVer>
                <expectScriptVer>2.3.5</expectScriptVer>
            </userAuthResponse>
        </m:RequestUserAuthenticationResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

RequestSignedAuthorization

This call retrieves work authorizations from the cloud service. For security, the utility technician requests work authorizations from the management cloud service before they can complete the work order.

Enum UserRole

```
"ADMIN" ## Unlimited commands
"TECH" ## Most commands and interface configure
"VIEWER" ## Read only commands
```

Struct DeviceInfo

```
String deviceId ## Device unique ID such as serial number
String ssid  ## WiFi SSID for a specific FAR device
String passphrase ## WiFi Passphrase for a specific FAR device
String firmwareVersion ## [Optional] Firmware version for this device. If present,
## IoT-FND needs to request it from NMS/Cloud
String configurationId ## [Optional] Unique configuration profile id for this
## device. If present, Bahamas needs to request it from
## NMS/Cloud
```

Attribute CanonicalizationTheMethod

```
String Algorithm
```

Attribute SignatureTheMethod

```
String Algorithm
```

Attribute TransformTheMethod

```
String Algorithm
```

Attribute DigestTheMethod

```
String Algorithm
```

Struct TransformsInfo

```
TransformTheMethod TransformMethod
```

Struct SignatureReference

```
TransformsInfo Transforms
DigestTheMethod DigestMethod
String DigestValue
```

Struct SignatureSignedInfo

```
CanonicalizationTheMethod CanonicalizationMethod
SignatureTheMethod SignatureMethod
SignatureReference Reference
## Signed info reference
```

Struct SignatureX509Data

```
String X509SubjectName
String X509Certificate
```

Struct SignatureKeyInfo

```
SignatureX509Data X509Data
```

Struct XMLSignature

```
SignatureSignedInfo SignedInfo
String SignatureValue
SignatureKeyInfo
KeyInfo
```

Struct SignedAuth

```
String version ## Version of the schema for signed authorization
String authId ## UID in the string format
String authCreated ## Date String in UTC format
String deviceId ## Device unique ID such as serial number
String technician ## Name of the technician requesting the authorization
UserRole userRole ## What role to manage the FAR
String startDateTime ## Date in UTC string to indicate valid-from
String endDateTime ## Date in UTC string to indicate valid-to
XmlSignature Signature ## The signature info to be passed to FAR
```

Struct SignedAuthorization

```
SignedAuth signedAuth
DeviceInfo deviceInfo ##[Optional]
```

Struct SignedAuthorizationResponse

```
SignedAuthorization[] signedAuthorization
```

Action RequestSignedAuthorization

Input:

```
String technicianUserName ## The technician username registered with NMS
String workOrderNumber ## If the technician knows a specific name, he can
## put it here; otherwise put null string will
## retrieve all the orders assigned to him
```

Output:

```
SignedAuthorizationResponse RequestSignedAuthorizationResponse
```

SOAP XML Request Format Request

```
POST  /nbapi/workorder   HTTP/1.1
Host: 128.154.157.111:8443
Content-Length: nnn
SOAPAction: http://workorder.nbapi.cgms.cisco.com/RequestSignedAuthorization
Authorization: Basic eWfqD54DefCdEf4B3aA=
Connection: close
Content-type: text/xml; charset="UTF-8"
User-Agent: Cisco Device Management Application
<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:m="http://workorder.nbapi.cgms.cisco.com/">

    <SOAP-ENV:Body>
        <m:RequestSignedAuthorization>
            <technicianUserName>bob</technicianUserName>
            <workOrderNumber>abcd-efgh-hijk</workOrderNumber>
        </m:RequestSignedAuthorization>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
HTTP/1.1   200 OK
Content-type: text/xml; charset="UTF-8"
Content-Length: nnn
<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
        <m:RequestSignedAuthorizationResponse
xmlns:m="http://workorder.nbapi.cgms.cisco.com/">
            <signedAuthorization>
                <signedAuth>
                   <version>1</version>
                   <orderNumber>9429C098-495A-402C-9456-57F3CC7475BF</orderNumber>
                   <authCreated>2011-09-20T03:48:28Z</authCreated>
                   <userRole>tech</userRole>
                   <technicianUserName>bob</technicianUserName>
                   <deviceId>AB193AQ</deviceId>
                  <startDate>2011-09-20T03:48:28Z</startDate>
                   <endDate>2011-09-20T09:48:28Z</endDate>
                   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
                         <SignedInfo>
                               <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
                               <SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
                               <Reference URI="">
                                     <Transforms>
                                          <TransformMethod
```

```
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                                      </Transforms>
                                      <DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                                   <DigestValue>CpJhSOAalu8QJeUX2wiaGG0ZGEk=</DigestValue>

                                </Reference>
                             </SignedInfo>
                             <SignatureValue>
                                lgkxDOwRdaAUOvKTVyzbdsLL0c4ONB51LTmmOdWXCYMErwMSy53nWUencU

                                h14cmR4rPsumdYWJzCqJik2oci6p8uMnLDCv8=
                             </SignatureValue>
                             <KeyInfo>
                                <X509Data>
                                    <X509SubjectName>
                                        CN=My Name,O=Test Certificates Inc.,C=US
                                    </X509SubjectName>
                                    <X509Certificate>

MIIB9zCCAWCgAwIBAgIERZwdkzANBgkqhkiG9w0BAQUFADBAMQswCQYD

VQQGEwJVUzEfMB0GA1UEChMWVGVzdCBDZXJ0aWZpY2F0ZXMgSW5jLjE
                                     MA4GA1UEAxMHTXkgTmFtZTAeFw0wNzAxMDMyMTE4MTFaFw0zMTA4M

                                        ...
                                    </X509Certificate>
                                </X509Data>
                             </KeyInfo>
                          </Signature>
                    </signedAuth>
                    <deviceInfo>
                        <deviceId>AB193AQ</deviceId>
                        <ssid>testfar</ssid>
                        <passphrase>cisco123</passphrase>
                        <firmwareVersion>1.2.3</firmwareVersion>
                        <configurationId>123.234</configurationId>
                    </deviceInfo>
                </signedAuthorization>
                <signedAuthorization>
                    ...  ...  ...  ...  ...
                <signedAuthorization>
            </m:RequestSignedAuthorizationResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

UploadServiceReport

This call uploads the service report to the cloud service. IoT FND tracks the service status for a work order, and prompts the utility technician to mark status field as job complete, job expired, job incomplete, and so on. Once work order status is reported, the work order can be safely removed from IoT FND. Work orders are archived on the NMS server.

Enum ServiceReportResult

```
“REPORT_OK”
“REPORT_FAIL”
```

Enum ServiceStatus

```
"Completed"     # The work is done
"Incomplete"     # The work can not be completed due to some reasons
"Expired"     # The work authorization is expired
```

Struct ServiceStatusReport

```
String    orderNumber
String    deviceId
String    technicianUserName
ServiceStaus     status
```

Struct ServiceStatusResponse

```
ServiceReportResult      result
String      comment    ## [Optional]
```

Struct ServiceReportResponse

```
ServiceStatusResponse      serviceReportResponse
```

Action UploadServiceReport

Input:

```
ServiceStatusReport []       serviceStatusReport
```

Output:

```
ServiceReportResponse       UploadServiceReportResponse
```

SOAP XML Request Format

```
POST  /nms/fieldtool   HTTP/1.1
Host: sample.cisco.com:8080
Content-Length: nnn
Authorization: Basic bW2eFaHdD3K
SOAPAction: [http://workorder.nbapi.cgms.cisco.com/UploadServiceReport
] Connection: close
Content-type: text/xml; charset="UTF-8"
User-Agent: Cisco Device Management Application
<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:m="http://workorder.nbapi.cgms.cisco.com/">
    <SOAP-ENV:Body>
        <m:UploadServiceReport>
            <serviceStatusReport>
                <orderNumber>9429C098-495A-402C-9456-57F3CC7475BF </orderNumber>
                <deviceId>AB193BQ</deviceId>
                <technicianUserName>Bob</technicianUserName>
                <status>Completed</status>
            </serviceStatusReport>
            <serviceStatusReport>
                <orderNumber>8688C098-495A-402C-2356-57F3CC7475BF </orderNumber>
                <deviceId>CC210EQ</deviceId>
```

```
                <technicianUserName>Alex</technicianUserName>
                <status>Expired</status>
            </serviceStatusReport>
        </m:UploadServiceReport>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Response

```
HTTP/1.1   200 OK
Content-type: text/xml; charset=”UTF-8”
Content-Length: nnn
<?xml version=”1.0” encoding=”utf-8”?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=”http://schemas.xmlsoap.org/soap/envelope/”>
    <SOAP-ENV:Body>
        <m:UploadServiceReportResponse xmlns:m="http://workorder.nbapi.cgms.cisco.com/">
            <serviceReportResponse>
                <result>REPORT_FAIL</result>
                <comment>Syntax Error</comment>
            </serviceReportResponse>
        </m:UploadServiceReportResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```