

Junos[®] Space

Network Director API

Published
2022-09-01

RELEASE
6.2

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Junos® Space Network Director API

6.2

Copyright © 2022 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <https://support.juniper.net/support/eula/>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide | iv

1

Overview

Network Director API | 2

Overview of Network Director API | 2

2

Fault and Monitoring APIs

Overview | 7

Fault and Monitoring APIs | 7

API Elements and Sample Output Files | 10

Fault and Monitoring API Elements and Sample Output Files | 10

3

API Reference

API Reference | 28

Junos Space Network Director API Reference | 28

About This Guide

Use this guide to refer to APIs provided by Junos Space Network Director for automating provisioning and management of services on EX Series and QFX Series devices.

1

PART

Overview

[Network Director API](#) | 2

CHAPTER 1

Network Director API

IN THIS CHAPTER

- [Overview of Network Director API | 2](#)

Overview of Network Director API

IN THIS SECTION

- [Prerequisites | 3](#)
- [Supported Devices and Services | 3](#)

The Junos Space Network Director API application runs on the Junos Space Network Management Platform, and is exposed by the Network Director orchestration services.

The Network Director API is a set of Representational State Transfer (REST) APIs that enable network management functions, including:

- Plugin and integration with CloudStack
- Virtualization of cloud and data center operations
- Provisioning of secure multitenant networks in a shared network infrastructure
- Automation of tenant services in the data center
- Support for Layer 2, Layer 3, security, and Internet services
- Provision of a single point of integration with external cloud and data center orchestration tools
- Support for obtaining alarm, monitoring and fault, and wireless client information.

Only the Super Administrator and Monitor Admin can access the Network Director APIs.

This topic describes:

Prerequisites

Before you can use the Network Director API, you must first install the following software:

- Junos Space Network Management Platform Release 20.3R1.
- Network Director API and Network Director GUI Release 4.1R1. These two applications are packaged together and need to be installed.
- REST HTTP client, which can be browser-based or script-based.
- (Optional) Python 2.7, 3.1, or 3.4 programming language software for running the sample API scripts included in the Network Director API software.

Supported Devices and Services

[Table 1 on page 3](#) lists the Juniper Networks devices that Network Director orchestration services and API support:

Table 1: Supported Platforms and the Software Versions for Network Director API

Supported Platforms	Qualified Junos OS, MSS, or the ESXi Releases
EX Series Switches	
EX2200 and EX2200-C (standalone and Virtual Chassis) and EX3200 EX3300 (standalone and Virtual Chassis) EX4200 (standalone and Virtual Chassis) EX4500 (standalone and Virtual Chassis) EX4550 (standalone and Virtual Chassis) Mixed EX4200, EX4500, and EX4550 Virtual Chassis EX6200 EX8200 (standalone and Virtual Chassis)	Junos OS Releases 11.4, 12.1, 12.2, 12.3, and 13.2X50-D10.2,
EX Series Switches with ELS	
EX9200 (standalone and Virtual Chassis)	Junos OS Releases 13.2R1, 13.2R2.4, 13.3R2, 16.1R1, 17.1R1.8, 17.2R1.13, 17.4R1.16, 18.1R1.9, and 18.2R1

Table 1: Supported Platforms and the Software Versions for Network Director API (Continued)

Supported Platforms	Qualified Junos OS, MSS, or the ESXi Releases
QFX Series Switches and Datacenter Fabrics	
QFX3500 (non-ELS) QFX3600 (non-ELS)	Junos OS Release 12.3X50-D40
QFX3500 with ELS (standalone and Virtual Chassis) QFX3600 with ELS (standalone and Virtual Chassis) QFX5100-48S with ELS (standalone and Virtual Chassis) QFX5100-24Q with ELS (standalone and Virtual Chassis) QFX5100-96S with ELS (standalone and Virtual Chassis) QFX5110	Junos OS Release 13.2X51-D20 Junos OS Release 17.4R1.16 and 18.1R3.3 for QFX5110
QFabric systems (QFX3000-G and QFX3000-M)	Junos OS Releases 13.1X52-D10 and 13.1X50-D15
Virtual Chassis Fabric	Junos OS Releases 13.1X52-D10 and 13.1X50-D15
VMware vCenter Server VMware Host	VMware ESX versions 4.0 and 4.1 VMware ESXi versions 5.0, 5.1, 5.5, 6.0, and 6.5

The type of network services supported depends on the device and the topology of the network.

[Table 2 on page 5](#) describes the supported devices and topologies, and configurations.

Table 2: Supported Devices, Topologies, and Services

Supported Devices and Topologies	Services	Configuration
<ul style="list-style-type: none"> EX Series switches—EX4200 and EX4550 switches QFabric systems—QFX 3000-G and QFX3000-M QFabric systems QFX Series—QFX3500 and QFX3600 switches 	Layer 2	VLANs on the EX Series switch, QFX Series, or QFabric system
<ul style="list-style-type: none"> EX Series switch or QFX Series connected to an MX Series router QFabric system connected to an MX Series router 	<ul style="list-style-type: none"> Layer 2 Layer 2 and Layer 3 Layer 2 and Layer 3 with Internet access 	<ul style="list-style-type: none"> VLANs on the EX Series switch, QFX Series, or QFabric system Layer 3 interfaces on the MX Series router BGP static configuration on the MX Series router
<ul style="list-style-type: none"> EX Series switch or QFX Series connected to an SRX Series Services Gateway QFabric system connected to an SRX Series Services Gateway 	<ul style="list-style-type: none"> Layer 2 Layer 2 and Layer 3 Layer 2 and Layer 3 with firewall 	<ul style="list-style-type: none"> VLANs on the EX Series switch, QFX Series, or QFabric system Layer 3 interfaces on the SRX Series Services Gateway Security policies on the SRX Series Services Gateway

RELATED DOCUMENTATION
[Fault and Monitoring APIs](#)
[Fault and Monitoring API Elements and Sample Output Files | 10](#)
[Junos Space Network Director API Reference | 28](#)

2

PART

Fault and Monitoring APIs

[Overview](#) | 7

[API Elements and Sample Output Files](#) | 10

CHAPTER 2

Overview

IN THIS CHAPTER

- [Fault and Monitoring APIs | 7](#)

Fault and Monitoring APIs

IN THIS SECTION

- [Monitoring APIs | 8](#)
- [Client APIs | 8](#)
- [Alarm APIs | 8](#)

The fault and monitoring APIs provide monitoring information, alarm notifications, and client information.

Network Director management APIs are a set of Representational State Transfer (REST) APIs that enable network management functions, including:

- Exporting monitoring and fault data for location, logical, and virtualization hierarchy to a third party management system
- Providing client data regarding location, equipment, and domains, as well information for wireless connected and unconnected clients
- Using notification profiles for forwarding events for alarms relating to all categories and severities
- Querying alarm history and filters based on time, type, and severity
- Allowing clients to specify host and NIC (network interface card) information

This topic describes:

Monitoring APIs

Monitoring APIs provide inventory information for managed devices in the network in both summarized and detailed levels. At the detailed level, the APIs provide data about the device's operational and environmental status, as well as the operational status of ports on the devices.

The monitoring APIs provide the following types of device information:

- Junos devices
- QFabric systems
- Virtual Chassis
- Wireless LAN controllers
- Access points
- Radios
- Power supplies
- Fans
- Ports
- Virtual machines (VMs)

See the Network Director API reference documentation for more details regarding specific APIs.

Client APIs

Client APIs provide information about wired and wireless clients and sessions on the network. A client is any device that is connected to the network through a wireless access point or through an access port on a switch that is an 802.1X authenticator port. Examples of clients include VoIP phones, laptops, printers, security cameras, and so on. Examples of session information includes number of sessions, session activity, information about each client, such as MAC address, IP address, username, client VLAN, port or wireless access point to which client is connected to, and so on.

See the Network Director API reference documentation for more details regarding specific APIs.

Alarm APIs

Network Director alarm APIs collect SNMP notifications and system alerts from devices and systems that are monitored. Depending on the notification or alert, an alarm is triggered. Client applications are required to receive alarms that are forwarded. The client application must use the same login credential as the one used for Network Director UI.

The alarm APIs provide the following types of alarm information:

- Name of alarm
- ID number of alarm
- Alarm type
- Category of alarm
- Severity of alarm
- State of alarm
- Acknowledgment of alarm
- Entity ID of alarm
- Source address of device that triggered alarm
- Date and time when alarm was created and updated
- Party that updated alarm

See the Network Director API reference documentation for more details regarding specific APIs.

CHAPTER 3

API Elements and Sample Output Files

IN THIS CHAPTER

- [Fault and Monitoring API Elements and Sample Output Files | 10](#)

Fault and Monitoring API Elements and Sample Output Files

IN THIS SECTION

- [Managed Devices API Sample Output | 11](#)
- [Managed Devices API Sample Output Using Paging Option | 12](#)
- [Managed Devices API Sample Output Using Detail Option | 14](#)
- [Resource Utilization and Alarm Count Trending APIs Sample Output | 15](#)
- [Port APIs Sample Output | 19](#)
- [Port APIs Sample Output Using Detail Option | 20](#)
- [Port Traffic Trend Data APIs Sample Output | 22](#)
- [Client Sessions APIs Sample Output | 23](#)
- [Alarm APIs Sample Output | 25](#)
- [Alarm APIs Sample Output Using Detail Option | 26](#)

This topic provides the following sample API requests and associated output:

- Resource utilization
- Alarms and alarm count trend data
- Port traffic trend data
- Client sessions

Options on how to display the output are also provided. Options include viewing the output based on a range of pages (pagination) and the amount of detail (detail).

This topic provides the following information:

Managed Devices API Sample Output

All APIs require that you provide an accept header in the API request, otherwise the output data will be presented by default.

Here is an example of an API request, followed by sample output. The sample output is in JSON format:

```
https://10.94.45.84/api/juniper/nd/monitoring/devices
Authorization: Basic c3VwZXI6anVuaXBlcjEyMw==
Accept: application/vnd.juniper.nd.devices+json;version=2;q=0.01
{
  "managedDeviceList": {
    "@total": "3",
    "device": [
      {
        "@uri": "/api/juniper/nd/monitoring/devices/qfx/622592",
        "resourceType": "JUNOS_QFX",
        "instanceId": 622592,
        "hostname": "qfx5100-01",
        "platform": "QFX5100-96S-8Q",
        "ipaddr": "192.168.48.168",
        "serial": "VB3113470018",
        "osVersion": "13.2X51-D20.2",
        "family": "JUNOS_QFX",
        "type": "NORMAL",
        "connStatus": "UP",
        "configState": "OUT_OF_SYNC"
      },
      {
        "@uri": "/api/juniper/nd/monitoring/devices/qfx/622593",
        "resourceType": "JUNOS_QFX",
        "instanceId": 622593,
        "hostname": "qfx5100-02",
        "platform": "QFX5100-48S-6Q",
        "ipaddr": "192.168.55.88",
        "serial": "TA3714141203",
        "osVersion": "13.2X51-D20.2",
        "family": "JUNOS_QFX",
```

```

        "type": "NORMAL",
        "connStatus": "UP",
        "configState": "OUT_OF_SYNC"
    },
    {
        "@uri": "/api/juniper/nd/monitoring/devices/ex/622611",
        "resourceType": "JUNOS_MX",
        "instanceId": 622611,
        "hostname": "EX9200-01",
        "platform": "MX240",
        "ipaddr": "10.204.245.71",
        "serial": "IBM888",
        "osVersion": "13.3R2.7",
        "family": "JUNOS",
        "type": "NORMAL",
        "connStatus": "DOWN",
        "configState": "SYNCHRONIZING"
    }
]
}
}

```

Managed Devices API Sample Output Using Paging Option

You can use the paging option to apply filters on the output. For example, you can specify how many entries to display and from which entry in the output set. In this example, the API request specifies output for three entries, starting from the second entry.

Here is an example API request, followed by sample output. The sample output is in JSON format:

```

https://10.94.45.84/api/juniper/nd/monitoring/devices?paging=(start eq 2, limit eq 3)
{
    "managedDeviceList": {
        "@total": "3",
        "device": [
            {
                "@uri": "/api/juniper/nd/monitoring/devices/qfx/622592",
                "resourceType": "JUNOS_QFX",
                "instanceId": 622592,
                "hostname": "QFX5100-01",
                "platform": "QFX5100-96S-8Q",
                "ipaddr": "192.168.48.168",

```



```

        "serial": "VB3113470018",
        "osVersion": "13.2X51-D20.2",
        "family": "JUNOS_QFX",
        "type": "NORMAL",
        "connStatus": "UP",
        "configState": "OUT_OF_SYNC"
    },
    {
        "@uri": "/api/juniper/nd/monitoring/devices/qfx/622593",
        "resourceType": "JUNOS_QFX",
        "instanceId": 622593,
        "hostname": "QFX5100-04",
        "platform": "QFX5100-48S-6Q",
        "ipaddr": "192.168.55.88",
        "serial": "TA3714141203",
        "osVersion": "13.2X51-D20.2",
        "family": "JUNOS_QFX",
        "type": "NORMAL",
        "connStatus": "UP",
        "configState": "OUT_OF_SYNC"
    },
    {
        "@uri": "/api/juniper/nd/monitoring/devices/ex/622611",
        "resourceType": "JUNOS_MX",
        "instanceId": 622611,
        "hostname": "EX9200-06",
        "platform": "MX240",
        "ipaddr": "10.204.245.71",
        "serial": "IBM888",
        "osVersion": "13.3R2.7",
        "family": "JUNOS",
        "type": "NORMAL",
        "connStatus": "DOWN",
        "configState": "SYNCHRONIZING"
    }
]
}
}

```

Managed Devices API Sample Output Using Detail Option

You can request detailed information for a device by using the query parameter in each device summary block.

Here is an example API request, followed by sample output. This sample output is in XML format.

```
https://10.94.45.84/api/juniper/nd/monitoring/devices/ex/622611/
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<junosDeviceM0 uri="/api/juniper/nd/monitoring/devices/ex/622611">
  <resourceType>JUNOS_MX</resourceType>
  <instanceId>622611</instanceId>
  <hostname>EX9200-01</hostname>
  <platform>MX240</platform>
  <ipaddr>10.204.245.71</ipaddr>
  <serial>IBM888</serial>
  <osVersion>13.3R2.7</osVersion>
  <family>JUNOS</family>
  <type>NORMAL</type>
  <connStatus>DOWN</connStatus>
  <configState>SYNCHRONIZING</configState>
  <usedMacCount>0</usedMacCount>
  <usedVlanCount>0</usedVlanCount>
  <lastChanged>Tue Jun 24 05:08:34 +0000 2014</lastChanged>
  <tempRange>avg=48, max=62 (FPC 1 XM 0 Chip)</tempRange>
  <totalPortUps>32</totalPortUps>
  <totalPortDowns>0</totalPortDowns>
  <totalPortFrees>32</totalPortFrees>
  <totalPortUsed>0</totalPortUsed>
  <psStatus>
    <name>PEM 0</name>
    <status>ABSENT</status>
  </psStatus>
  <psStatus>
    <name>PEM 1</name>
    <status>ABSENT</status>
  </psStatus>
  <psStatus>
    <name>PEM 2</name>
    <status>ABSENT</status>
  </psStatus>
  <psStatus>
```

```

        <name>PEM 3</name>
        <status>OK</status>
    </psStatus>
    <fanStatus>
        <name>Front Fan</name>
        <status>OK</status>
    </fanStatus>
    <fanStatus>
        <name>Middle Fan</name>
        <status>OK</status>
    </fanStatus>
    <fanStatus>
        <name>Rear Fan</name>
        <status>OK</status>
    </fanStatus>
    <uptime>3d:18h:14m:41s</uptime>
</junosDeviceM0>

```

Resource Utilization and Alarm Count Trending APIs Sample Output

Resource utilization and alarm count trend data are supported only on EX Series devices, Access Points, Wireless LAN Controllers, and QFabric systems.

Here is an example API request, followed by sample output. This sample output is in XML format.

```

https://10.94.45.84/api/juniper/nd/monitoring/devices/ex/622611/resource-utilization-trend?
hours=2

```

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<resourceUtilizationTrend total="0"/>

```

```

https://10.94.45.84/api/juniper/nd/monitoring/devices/ex/622611/alarm-count-trend?hours=1

```

Output:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<alarmCountTrend total="12">
    <stat>
        <collectionTimestamp>1403815500</collectionTimestamp>
        <allInfoAlarm>0.0</allInfoAlarm>
        <allMinorAlarm>0.0</allMinorAlarm>
        <allMajorAlarm>1.0</allMajorAlarm>
        <allCriticalAlarm>0.0</allCriticalAlarm>
        <newInfoAlarm>0.0</newInfoAlarm>
        <newMinorAlarm>0.0</newMinorAlarm>
    </stat>

```

```

    <newMajorAlarm>0.0</newMajorAlarm>
    <newCriticalAlarm>0.0</newCriticalAlarm>
  </stat>
  <stat>
    <collectionTimestamp>1403815800</collectionTimestamp>
    <allInfoAlarm>0.0</allInfoAlarm>
    <allMinorAlarm>0.0</allMinorAlarm>
    <allMajorAlarm>1.0</allMajorAlarm>
    <allCriticalAlarm>0.0</allCriticalAlarm>
    <newInfoAlarm>0.0</newInfoAlarm>
    <newMinorAlarm>0.0</newMinorAlarm>
    <newMajorAlarm>0.0</newMajorAlarm>
    <newCriticalAlarm>0.0</newCriticalAlarm>
  </stat>
  <stat>
    <collectionTimestamp>1403816100</collectionTimestamp>
    <allInfoAlarm>0.0</allInfoAlarm>
    <allMinorAlarm>0.0</allMinorAlarm>
    <allMajorAlarm>1.0</allMajorAlarm>
    <allCriticalAlarm>0.0</allCriticalAlarm>
    <newInfoAlarm>0.0</newInfoAlarm>
    <newMinorAlarm>0.0</newMinorAlarm>
    <newMajorAlarm>0.0</newMajorAlarm>
    <newCriticalAlarm>0.0</newCriticalAlarm>
  </stat>
  <stat>
    <collectionTimestamp>1403816400</collectionTimestamp>
    <allInfoAlarm>0.0</allInfoAlarm>
    <allMinorAlarm>0.0</allMinorAlarm>
    <allMajorAlarm>1.0</allMajorAlarm>
    <allCriticalAlarm>0.0</allCriticalAlarm>
    <newInfoAlarm>0.0</newInfoAlarm>
    <newMinorAlarm>0.0</newMinorAlarm>
    <newMajorAlarm>0.0</newMajorAlarm>
    <newCriticalAlarm>0.0</newCriticalAlarm>
  </stat>
  <stat>
    <collectionTimestamp>1403816700</collectionTimestamp>
    <allInfoAlarm>0.0</allInfoAlarm>
    <allMinorAlarm>0.0</allMinorAlarm>
    <allMajorAlarm>1.0</allMajorAlarm>
    <allCriticalAlarm>0.0</allCriticalAlarm>
    <newInfoAlarm>0.0</newInfoAlarm>

```

```

    <newMinorAlarm>0.0</newMinorAlarm>
    <newMajorAlarm>0.0</newMajorAlarm>
    <newCriticalAlarm>0.0</newCriticalAlarm>
</stat>
<stat>
    <collectionTimestamp>1403817000</collectionTimestamp>
    <allInfoAlarm>0.0</allInfoAlarm>
    <allMinorAlarm>0.0</allMinorAlarm>
    <allMajorAlarm>1.0</allMajorAlarm>
    <allCriticalAlarm>0.0</allCriticalAlarm>
    <newInfoAlarm>0.0</newInfoAlarm>
    <newMinorAlarm>0.0</newMinorAlarm>
    <newMajorAlarm>0.0</newMajorAlarm>
    <newCriticalAlarm>0.0</newCriticalAlarm>
</stat>
<stat>
    <collectionTimestamp>1403817300</collectionTimestamp>
    <allInfoAlarm>0.0</allInfoAlarm>
    <allMinorAlarm>0.0</allMinorAlarm>
    <allMajorAlarm>1.0</allMajorAlarm>
    <allCriticalAlarm>0.0</allCriticalAlarm>
    <newInfoAlarm>0.0</newInfoAlarm>
    <newMinorAlarm>0.0</newMinorAlarm>
    <newMajorAlarm>0.0</newMajorAlarm>
    <newCriticalAlarm>0.0</newCriticalAlarm>
</stat>
<stat>
    <collectionTimestamp>1403817600</collectionTimestamp>
    <allInfoAlarm>0.0</allInfoAlarm>
    <allMinorAlarm>0.0</allMinorAlarm>
    <allMajorAlarm>1.0</allMajorAlarm>
    <allCriticalAlarm>0.0</allCriticalAlarm>
    <newInfoAlarm>0.0</newInfoAlarm>
    <newMinorAlarm>0.0</newMinorAlarm>
    <newMajorAlarm>0.0</newMajorAlarm>
    <newCriticalAlarm>0.0</newCriticalAlarm>
</stat>
<stat>
    <collectionTimestamp>1403817900</collectionTimestamp>
    <allInfoAlarm>0.0</allInfoAlarm>
    <allMinorAlarm>0.0</allMinorAlarm>
    <allMajorAlarm>1.0</allMajorAlarm>
    <allCriticalAlarm>0.0</allCriticalAlarm>

```

```

    <newInfoAlarm>0.0</newInfoAlarm>
    <newMinorAlarm>0.0</newMinorAlarm>
    <newMajorAlarm>0.0</newMajorAlarm>
    <newCriticalAlarm>0.0</newCriticalAlarm>
  </stat>
  <stat>
    <collectionTimestamp>1403818200</collectionTimestamp>
    <allInfoAlarm>0.0</allInfoAlarm>
    <allMinorAlarm>0.0</allMinorAlarm>
    <allMajorAlarm>1.0</allMajorAlarm>
    <allCriticalAlarm>0.0</allCriticalAlarm>
    <newInfoAlarm>0.0</newInfoAlarm>
    <newMinorAlarm>0.0</newMinorAlarm>
    <newMajorAlarm>0.0</newMajorAlarm>
    <newCriticalAlarm>0.0</newCriticalAlarm>
  </stat>
  <stat>
    <collectionTimestamp>1403818500</collectionTimestamp>
    <allInfoAlarm>0.0</allInfoAlarm>
    <allMinorAlarm>0.0</allMinorAlarm>
    <allMajorAlarm>1.0</allMajorAlarm>
    <allCriticalAlarm>0.0</allCriticalAlarm>
    <newInfoAlarm>0.0</newInfoAlarm>
    <newMinorAlarm>0.0</newMinorAlarm>
    <newMajorAlarm>0.0</newMajorAlarm>
    <newCriticalAlarm>0.0</newCriticalAlarm>
  </stat>
  <stat>
    <collectionTimestamp>1403818800</collectionTimestamp>
    <allInfoAlarm>0.0</allInfoAlarm>
    <allMinorAlarm>0.0</allMinorAlarm>
    <allMajorAlarm>1.0</allMajorAlarm>
    <allCriticalAlarm>0.0</allCriticalAlarm>
    <newInfoAlarm>0.0</newInfoAlarm>
    <newMinorAlarm>0.0</newMinorAlarm>
    <newMajorAlarm>0.0</newMajorAlarm>
    <newCriticalAlarm>0.0</newCriticalAlarm>
  </stat>
</alarmCountTrend>

```

Port APIs Sample Output

You can use port API requests to query managed ports for summary information for each port based on query parameters. If the query parameter value is omitted, summary information for all ports will be displayed.

Here is an example without any query parameters:

```
https://10.94.45.84/api/juniper/nd/monitoring/ports
```

Output is not included.

You can use the paging option to apply filters on the output. For example, you can specify how many entries to display and from which entry in the output set. In this example, the API request specifies output for three entries, starting from the tenth entry.

Here is an example API request, followed by sample output. The sample output is in XML format:

```
https://10.94.45.84/api/juniper/nd/monitoring/ports?paging=(start eq 10, limit eq 3)
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ports uri="/api/juniper/nd/monitoring/ports">
  <port uri="/api/juniper/nd/monitoring/ports/ex/VB3113470018_xe-0.0.90">
    <instanceId>VB3113470018_xe-0.0.90</instanceId>
    <portId>VB3113470018_xe-0.0.90</portId>
    <portName>xe-0/0/90</portName>
    <portMac>88:e0:f3:1d:6b:5d</portMac>
    <hostSerial>VB3113470018</hostSerial>
    <hostname>analytics-qfx5100-01</hostname>
    <portSpeed>10000</portSpeed>
    <portType>10-Gigabit</portType>
    <adminStatus>UP</adminStatus>
    <operStatus>UP</operStatus>
    <lastFlap>2 Days 5 Hrs 15 Mins 24 Secs ago</lastFlap>
  </port>
  <port uri="/api/juniper/nd/monitoring/ports/ex/VB3113470018_xe-0.0.91">
    <instanceId>VB3113470018_xe-0.0.91</instanceId>
    <portId>VB3113470018_xe-0.0.91</portId>
    <portName>xe-0/0/91</portName>
    <portMac>88:e0:f3:1d:6b:5e</portMac>
    <hostSerial>VB3113470018</hostSerial>
    <hostname>analytics-qfx5100-01</hostname>
```

```

    <portSpeed>10000</portSpeed>
    <portType>10-Gigabit</portType>
    <adminStatus>UP</adminStatus>
    <operStatus>UP</operStatus>
    <lastFlap>2 Days 5 Hrs 15 Mins 24 Secs ago</lastFlap>
  </port>
  <port uri="/api/juniper/nd/monitoring/ports/ex/VB3113470018_xe-0.0.92">
    <instanceId>VB3113470018_xe-0.0.92</instanceId>
    <portId>VB3113470018_xe-0.0.92</portId>
    <portName>xe-0/0/92</portName>
    <portMac>88:e0:f3:1d:6b:5f</portMac>
    <hostSerial>VB3113470018</hostSerial>
    <hostname>analytics-qfx5100-01</hostname>
    <portSpeed>10000</portSpeed>
    <portType>10-Gigabit</portType>
    <adminStatus>UP</adminStatus>
    <operStatus>UP</operStatus>
    <lastFlap>2 Days 5 Hrs 15 Mins 23 Secs ago</lastFlap>
  </port>
</ports>

```

Port APIs Sample Output Using Detail Option

You can request detailed information for a port by using the query parameter in each port summary block.

Here is an example API request, followed by sample output. The sample output is in XML format:

```

https://10.94.45.84/api/juniper/nd/monitoring/ports/ex/VB3113470018_xe-0.0.91
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<portTraffic uri="/api/juniper/nd/monitoring/ports/ex/VB3113470018_xe-0.0.91">
  <instanceId>VB3113470018_xe-0.0.91</instanceId>
  <portId>VB3113470018_xe-0.0.91</portId>
  <portName>xe-0/0/91</portName>
  <portMac>88:e0:f3:1d:6b:5e</portMac>
  <hostSerial>VB3113470018</hostSerial>
  <hostname>analytics-qfx5100-01</hostname>
  <portSpeed>10000</portSpeed>
  <portType>10-Gigabit</portType>
  <adminStatus>UP</adminStatus>
  <operStatus>UP</operStatus>
  <lastFlap>2 Days 5 Hrs 25 Mins 18 Secs ago</lastFlap>

```



```

<trafficInOut>
  <collectionTimestamp>1403819400</collectionTimestamp>
  <unicastInRate>0.0</unicastInRate>
  <unicastOutRate>0.0</unicastOutRate>
  <broadcastInRate>0.0</broadcastInRate>
  <broadcastOutRate>0.0</broadcastOutRate>
  <multicastInRate>0.0</multicastInRate>
  <multicastOutRate>0.0</multicastOutRate>
</trafficInOut>
<trafficError>
  <collectionTimestamp>0</collectionTimestamp>
  <dropsInErrs>0.0</dropsInErrs>
  <framingErrsIn>0.0</framingErrsIn>
  <runtInErr>0.0</runtInErr>
  <discardsInErr>0.0</discardsInErr>
  <l3IncompletesIn>0.0</l3IncompletesIn>
  <l2ChannelErrorsIn>0.0</l2ChannelErrorsIn>
  <l2MismatchTmeout>0.0</l2MismatchTmeout>
  <fifoErrsIn>0.0</fifoErrsIn>
  <resourceErrsIn>0.0</resourceErrsIn>
  <carrierTransOut>0.0</carrierTransOut>
  <collisionOut>0.0</collisionOut>
  <dropsOut>0.0</dropsOut>
  <agedPktsOut>0.0</agedPktsOut>
  <mtuErrsOut>0.0</mtuErrsOut>
  <hsLinkCrcErr>0.0</hsLinkCrcErr>
  <fifoErrsOut>0.0</fifoErrsOut>
  <resourceErrsOut>0.0</resourceErrsOut>
  <crcErrsIn>0.0</crcErrsIn>
  <crcErrsOut>0.0</crcErrsOut>
  <oversizedFrmIn>0.0</oversizedFrmIn>
  <oversizedFrmOut>0.0</oversizedFrmOut>
  <jabberFrmIn>0.0</jabberFrmIn>
  <fragFrmIn>0.0</fragFrmIn>
  <codeViolationIn>0.0</codeViolationIn>
</trafficError>
</portTraffic>

```

Port Traffic Trend Data APIs Sample Output

You can use Port traffic trend data API requests to query ports based on the port ID as well as the time range given by the parameters. You must provide either start time and end time of the query or the last number of hours.

Here is an example API request, followed by sample output. The sample output is in XML format:

```
https://0.94.45.84/api/juniper/nd/monitoring/ports/ex/
VB3113470018_xe-0.0.91/port-traffic-trend?
startTime=2014-03-03T01:02:00Z&endTime=2014-03-14T10:00:00Z
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<portTrafficTrend total="12" uri="/api/juniper/nd/monitoring/ports/ex/VB3113470018_xe-0.0.91">
  <instanceId>VB3113470018_xe-0.0.91</instanceId>
  <portId>VB3113470018_xe-0.0.91</portId>
  <portName>xe-0/0/91</portName>
  <portMac>88:e0:f3:1d:6b:5e</portMac>
  <hostSerial>VB3113470018</hostSerial>
  <hostname>analytics-qfx5100-01</hostname>
  <portSpeed>10000</portSpeed>
  <portType>10-Gigabit</portType>
  <adminStatus>UP</adminStatus>
  <operStatus>UP</operStatus>
  <lastFlap>2 Days 5 Hrs 25 Mins 18 Secs ago</lastFlap>
  <trafficInOut>
    <collectionTimestamp>1403816100</collectionTimestamp>
    <unicastInRate>0.0</unicastInRate>
    <unicastOutRate>0.0</unicastOutRate>
    <broadcastInRate>0.0</broadcastInRate>
    <broadcastOutRate>0.0</broadcastOutRate>
    <multicastInRate>0.0</multicastInRate>
    <multicastOutRate>0.0</multicastOutRate>
  </trafficInOut>
  <trafficInOut>
    <collectionTimestamp>1403816400</collectionTimestamp>
    <unicastInRate>0.0</unicastInRate>
    <unicastOutRate>0.0</unicastOutRate>
    <broadcastInRate>0.0</broadcastInRate>
    <broadcastOutRate>0.0</broadcastOutRate>
    <multicastInRate>0.0</multicastInRate>
    <multicastOutRate>0.0</multicastOutRate>
  </trafficInOut>
```

```

<trafficInOut>
  <collectionTimestamp>1403816700</collectionTimestamp>
  <unicastInRate>0.0</unicastInRate>
  <unicastOutRate>0.0</unicastOutRate>
  <broadcastInRate>0.0</broadcastInRate>
  <broadcastOutRate>0.0</broadcastOutRate>
  <multicastInRate>0.0</multicastInRate>
  <multicastOutRate>0.0</multicastOutRate>
</trafficInOut>
<trafficInOut>
  <collectionTimestamp>1403817000</collectionTimestamp>
  <unicastInRate>0.0</unicastInRate>
  <unicastOutRate>0.0</unicastOutRate>
  <broadcastInRate>0.0</broadcastInRate>
  <broadcastOutRate>0.0</broadcastOutRate>
  <multicastInRate>0.0</multicastInRate>
  <multicastOutRate>0.0</multicastOutRate>
</trafficInOut>
...
</portTrafficTrend>

```

Client Sessions APIs Sample Output

You can use the client sessions APIs to query client sessions based on search parameters, such as username, MAC address, IPv4 address, IPv6 address, and so on. If you do not specify a search parameter, an exception error is issued.

Here is an example API request, followed by sample output. The sample output is in XML format:

```

https://10.94.45.84/api/juniper/nd/monitoring/client-sessions?search=18:26:66:72:81:62
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<clientSessions total="1" uri="/api/juniper/nd/monitoring/client-sessions">
  <session uri="/api/juniper/nd/monitoring/client-sessions/wireless/18:26:66:72:81:62">
    <instanceId>18:26:66:72:81:62</instanceId>
    <resourceType>WIRELESSCLIENT</resourceType>
    <macAddress>18:26:66:72:81:62</macAddress>
    <userName>last-resort-AWN-2105</userName>
    <ipv4>192.168.34.175</ipv4>
    <ipv6>[]</ipv6>
    <linkLocal>fe80::1a26:66ff:fe72:8162</linkLocal>
  </session>
</clientSessions>

```

```

    </session>
  </clientSessions>

```

You can also search using partial search data:

Here is an example API request, followed by sample output. The sample output is in XML format:

```

https://10.94.45.84/api/juniper/nd/monitoring/client-sessions?search=BB
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<clientSessions total="3" uri="/api/juniper/nd/monitoring/client-sessions">
  <session uri="/api/juniper/nd/monitoring/client-sessions/wireless/1c:b0:94:4c:bb:09">
    <instanceId>1c:b0:94:4c:bb:09</instanceId>
    <resourceType>WIRELESSCLIENT</resourceType>
    <macAddress>1c:b0:94:4c:bb:09</macAddress>
    <userName>last-resort-AWN-2095</userName>
    <ipv4>192.168.33.197</ipv4>
    <ipv6>[]</ipv6>
    <linkLocal>fe80::1eb0:94ff:fe4c:bb09</linkLocal>
  </session>
  <session uri="/api/juniper/nd/monitoring/client-sessions/wireless/34:bb:26:df:ec:05">
    <instanceId>34:bb:26:df:ec:05</instanceId>
    <resourceType>WIRELESSCLIENT</resourceType>
    <macAddress>34:bb:26:df:ec:05</macAddress>
    <userName>last-resort-AWN-2124</userName>
    <ipv4>192.168.32.95</ipv4>
    <ipv6>[]</ipv6>
    <linkLocal>fe80::36bb:26ff:fedf:ec05</linkLocal>
  </session>
  <session uri="/api/juniper/nd/monitoring/client-sessions/wireless/58:a2:b5:d0:49:bb">
    <instanceId>58:a2:b5:d0:49:bb</instanceId>
    <resourceType>WIRELESSCLIENT</resourceType>
    <macAddress>58:a2:b5:d0:49:bb</macAddress>
    <userName>last-resort-AWN-2262</userName>
    <ipv4>192.168.33.3</ipv4>
    <ipv6>[]</ipv6>
    <linkLocal>fe80::5aa2:b5ff:fed0:49bb</linkLocal>
  </session>
</clientSessions>

```

Alarm APIs Sample Output

You can use alarm API requests to query alarms based on the query parameters. If you omit the query parameter, information for all alarms will be provided.

Here is an example API request, followed by sample output. The sample output is in XML format:

```
https://10.94.45.84/api/juniper/nd/monitoring/alarms?paging=(start eq 10, limit eq 3)
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<alarms uri="/api/juniper/nd/monitoring/alarms" total="93">
  <alarm uri="/api/juniper/nd/monitoring/alarms/11">
    <instanceId>11</instanceId>
    <alarmId>11</alarmId>
    <name>Port High Non-Unicast Traffic</name>
    <category>Threshold</category>
    <severity>MAJOR</severity>
    <state>ACTIVE</state>
    <ack>false</ack>
    <entityId>Port:LX0212521598.ge-0/0/2</entityId>
    <source>10.94.45.171</source>
    <creationTime>Thu Jun 12 18:48:23 UTC 2014</creationTime>
    <lastUpdated>Mon Jun 23 19:48:23 UTC 2014</lastUpdated>
    <updatedBy>System</updatedBy>
  </alarm>
  <alarm uri="/api/juniper/nd/monitoring/alarms/12">
    <instanceId>12</instanceId>
    <alarmId>12</alarmId>
    <name>Device High CPU Utilization</name>
    <category>Threshold</category>
    <severity>MAJOR</severity>
    <state>ACTIVE</state>
    <ack>false</ack>
    <entityId>Device:LX0212521598</entityId>
    <source>10.94.45.171</source>
    <creationTime>Thu Jun 12 19:48:23 UTC 2014</creationTime>
    <lastUpdated>Thu Jun 26 17:54:16 UTC 2014</lastUpdated>
    <updatedBy>System</updatedBy>
  </alarm>
  <alarm uri="/api/juniper/nd/monitoring/alarms/13">
    <instanceId>13</instanceId>
    <alarmId>13</alarmId>
    <name>Port High Utilization</name>
```

```

    <category>Threshold</category>
    <severity>MAJOR</severity>
    <state>ACTIVE</state>
    <ack>false</ack>
    <entityId>Port:P5228.xe-0/0/30</entityId>
    <source>192.168.48.246</source>
    <creationTime>Thu Jun 12 22:18:23 UTC 2014</creationTime>
    <lastUpdated>Thu Jun 26 17:54:17 UTC 2014</lastUpdated>
    <updatedBy>System</updatedBy>
  </alarm>
</alarms>

```

Alarm APIs Sample Output Using Detail Option

You can use alarm API requests to query alarms based on query parameters. If you omit the query parameter, information for all alarms will be provided.

Here is an example API request, followed by sample output. The sample output is in XML format:

```

https://10.94.45.84/api/juniper/nd/monitoring/alarms/10
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<alarm uri="/api/juniper/nd/monitoring/alarms/10">
  <instanceId>10</instanceId>
  <alarmId>10</alarmId>
  <name>Port High Non-Unicast Traffic</name>
  <category>Threshold</category>
  <severity>MAJOR</severity>
  <state>ACTIVE</state>
  <ack>false</ack>
  <entityId>Port:PG3713400425.ge-0/0/23</entityId>
  <source>10.204.247.50</source>
  <creationTime>Thu Jun 12 18:48:23 UTC 2014</creationTime>
  <lastUpdated>Thu Jun 26 17:54:16 UTC 2014</lastUpdated>
  <updatedBy>System</updaedBy>
</alarm>

```

RELATED DOCUMENTATION

[Overview of Network Director API | 2](#)

3

PART

API Reference

API Reference | 28

API Reference

IN THIS CHAPTER

- [Junos Space Network Director API Reference](#) | 28

Junos Space Network Director API Reference

Junos Space Network Director APIs are based on the Representational State Transfer (REST) standards. REST defines a set of principles for defining Web services, including how a system's resource states are transferred over HTTP. Clients can be written in any language that sends HTTP requests.

[Network Director RESTful API Reference](#) describes the Junos Space Network Director REST APIs and how to use them for automating Network Director operations.