

# **WebSAM Network Management**

## **Web API 1.0**

### **リファレンスマニュアル**

---

# 著作権

本書に記載する内容の著作権は、日本電気株式会社に帰属します。本書の内容の一部、または、全部を日本電気株式会社の書面による許可なくコピー、改変することを禁止しています。

本書の内容には、日本電気株式会社が開示するすべての情報を掲載していない場合、または、他の方法で開示している情報と表現が異なっている場合があります。また、本書の内容は、将来、予告なしに変更または、廃止する場合がありますので、あらかじめご承知おきください。

本書を制作するにあたり、正確さを期するために万全の注意を払っておりますが、日本電気株式会社は、本書の内容に関し、その正確性、有用性、確実性、その他のいかなる保証もいたしません。また、日本電気株式会社は、本書の技術的、もしくは、編集上の間違いや欠落について、一切の責任を負いません。

# 商標


- NEC、NEC ロゴは、日本およびその他の国における日本電気株式会社の商標および登録商標です。
- Microsoft、Windows、Microsoft Edge、Internet Explorer は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。
- Google Chrome は、Google Inc. の登録商標または商標です。
- Firefox は、Mozilla Foundation の米国およびその他の国における登録商標または商標です。
- Linux は Linus Torvalds 氏の米国およびその他の国における登録商標です。
- Red Hat は Red Hat Software, Inc. の商標または登録商標です。
- Intel、Xeon、Intel Core は、Intel Corporation の米国ならびに他の国における商標または登録商標です。
- そのほかの会社名ならびに商標名は各社の商標または登録商標です。
- 本文中では™や®は明記していません。

# はじめに

このたびは、WebSAM のネットワーク運用管理製品をお買い求めいただき、誠にありがとうございます。WebSAM のネットワーク運用管理製品では、WebSAM Integrated Management Server (以下、IMS と示す) コンポーネントを活用することで、複数製品の情報、操作を 1 つの Web コンソールでシームレスに統合し、運用することができます。

本書では、IMS コンポーネントが提供する Web API 1.0 について説明しています。Web API を利用することで、Web コンソールで操作できるものと同じ情報を他のアプリケーションで活用することができます。Web API を利用する前に、本書をよくお読みください。

## 本書の構成

本書の構成は、以下の通りです。本書の対象者は、Web API のすべての利用者  となります。


本書の構成

タイトル	内容
「第 1 章 Web API について (1 ページ)」	Web API の概要や利用にあたってのシステム構成について説明します。
「第 2 章 Web API 共通の仕様 (4 ページ)」	Web API 利用におけるリクエスト、および、レスポンスの共通仕様について説明します。
「第 3 章 リソース形式 (19 ページ)」	各 API のリクエスト、および、レスポンスのメッセージボディにおけるリソース形式について説明します。
「第 4 章 Web API リファレンス (33 ページ)」	各 API の詳細な仕様について説明します。
「第 5 章 チュートリアル (61 ページ)」	Web API の基本的な使い方を具体的な運用例をもとに説明します。

## 本書の表記規則


本書では、注意すべき事項や補足事項について、以下の表記を用います。

注意補足事項の表記

表記	説明
 <b>注意</b>	機能設定、操作を行う上で守らなければならない事柄や特に注意すべき点を示します。
<b>ヒント</b>	知っておくと役に立つ便利な情報を示します。

本書では、以下の表記規則に従って記述しています。

## 表記規則

表記	説明	例
[item]	メニュー、項目名、ボタンなどの画面要素を示します。	 ダッシュボード]メニュー、 [OK]ボタン
<userinput>	ユーザー環境により変化する項目、および入力値を示します。	<% インストールパス %>、 <filepath>
configuration file	設定ファイルの記述内容を示します。	以下の値を設定します。  port = 27120
command line	コマンドライン操作を示します。	以下のコマンドを実行します。  > Setup.exe

本書では、以下の略称を用いて記述しています。

## 略称表現

正式表記	略称表現
WebSAM Integrated Management Server	Integrated Management Server、IMS
WebSAM NetvisorPro V	NetvisorPro
WebSAM Network Flow Analyzer	NFA
WebSAM NetvisorPro V Event Adapter	Event Adapter
WebSAM SystemManager G	SystemManager G

---

# 目次

<b>第 1 章 Web API について .....</b>	<b>1</b>
1.1 Web API で可能な操作 .....	2
1.2 Web API 利用時のシステム構成.....	2
<b>第 2 章 Web API 共通の仕様.....</b>	<b>4</b>
2.1 リクエスト形式.....	5
2.1.1 エンドポイント.....	5
2.1.2 認証方式 .....	5
2.1.3 HTTP バージョン .....	7
2.1.4 HTTP メソッド.....	8
2.1.5 HTTP リクエストヘッダー .....	8
2.1.6 クエリーパラメーターの指定 .....	9
2.2 リクエスト、レスポンスのデータ形式.....	13
2.3 レスポンス形式.....	14
2.3.1 HTTP レスポンスヘッダー .....	14
2.3.2 共通の HTTP ステータスコード .....	15
2.3.3 エラーレスポンスの形式.....	16
2.4 Web API のバージョン .....	17
<b>第 3 章 リソース形式.....</b>	<b>19</b>
3.1 構成管理のためのリソース .....	20
3.1.1 リージョンリソース (RegionObject).....	20
3.1.2 ノードリソース (NodeObject) .....	20
3.1.3 ネットワークインターフェイスリソース (NetworkInterfaceObject).....	24
3.2 性能管理のためのリソース .....	25
3.2.1 ウィジェットデータリソース (WidgetDataObject).....	25
3.2.2 統計情報リソース (StatisticsInfoObject) .....	27
3.2.3 統計項目リソース (StatisticItemObject) .....	28
3.2.4 統計データリソース (StatisticDataObject) .....	31
<b>第 4 章 Web API リファレンス .....</b>	<b>33</b>
4.1 構成管理 API.....	34
4.1.1 リージョングループ一覧取得 .....	34
4.1.2 ノード一覧取得.....	34
4.1.3 ノード詳細取得.....	37
4.1.4 ネットワークインターフェイス一覧取得 .....	38

---

4.1.5 ネットワークインターフェイス詳細取得 .....	41
4.2 障害管理 API .....	43
4.2.1 ノード稼働率 TopN の取得 .....	43
4.2.2 特定ノードの稼働率の取得 .....	45
4.3 性能管理 API .....	46
4.3.1 SNMP データ TopN 取得 .....	46
4.3.2 特定ノードの SNMP データ取得 .....	49
4.3.3 特定ネットワークインターフェイスの SNMP データ取得 .....	51
4.3.4 フローデータ TopN 取得 .....	52
4.3.5 特定ノードのフローデータ TopN 取得 .....	55
4.3.6 特定ネットワークインターフェイスのフローデータ TopN 取得 .....	57
<b>第 5 章 チュートリアル .....</b>	<b>61</b>
5.1 構成情報の操作 .....	62
5.1.1 特定シリーズのソフトウェアバージョンを確認する .....	62
5.1.2 物理インターフェイスの詳細情報を確認する .....	64
5.2 障害情報の操作 .....	67
5.2.1 障害が発生しているノードを確認する .....	67
5.2.2 複数ノードの稼働状況を確認する .....	69
5.3 性能情報の操作 .....	72
5.3.1 ルーターの CPU 負荷の状況を確認する .....	73
5.3.2 WAN インターフェイスの通信内容を確認する .....	76
<b>付録 A. レポート作成用サンプルマクロ .....</b>	<b>82</b>
A.1 サンプルマクロの概要 .....	82
A.2 サンプルマクロの使用方法 .....	83

---

# 第 1 章

## Web API について

IMS コンポーネントでは、管理情報をリモートから操作するための Web API を提供しています。本章では、Web API の概要や利用にあたってのシステム構成について説明します。

---

### 目次

1.1 Web API で可能な操作 .....	2
1.2 Web API 利用時のシステム構成.....	2

---

## 1.1 Web API で可能な操作

Web API を利用することで、以下の操作を行うことができます。

表 1-1 Web API で可能な操作

機能名	操作内容	説明
構成管理	リージョングループ一覧取得	監視対象のノード、および、ネットワークインターフェイスの管理情報を取得することができます。
	ノード一覧取得	
	ノード詳細取得	
	ネットワークインターフェイス一覧取得	
	ネットワークインターフェイス詳細取得	
障害管理	ノードの稼働率 TopN の取得	ノードの稼働率などの障害管理のための情報を取得することができます。
	特定ノードの稼働率の取得	
性能管理	SNMP データ TopN 取得	NetvisorPro が収集した SNMP データ(MIB 情報)や、NFA が受信したフローデータを取得することができます。
	特定ノードの SNMP データ取得	
	特定ネットワークインターフェイスの SNMP データ取得	
	フローデータ TopN 取得	
	特定ノードのフローデータ TopN 取得	
	特定ネットワークインターフェイスのフローデータ TopN 取得	

各 API の詳細については、「[第4章 Web API リファレンス \(33 ページ\)](#)」を参照してください。

## 1.2 Web API 利用時のシステム構成

Web API 利用時のシステム構成について説明します。

IMS コンポーネントにおいて、Web コンソールを利用できるようにセットアップすることで、Web API も併せて利用できる状態となります。

「[図 1-1 システム構成 \(3 ページ\)](#)」に示すように、Web API を利用するクライアントシステム(アプリケーション)は、IMS コンポーネントにアクセスして、NetvisorPro、または、NFA から取得した情報を操作します。



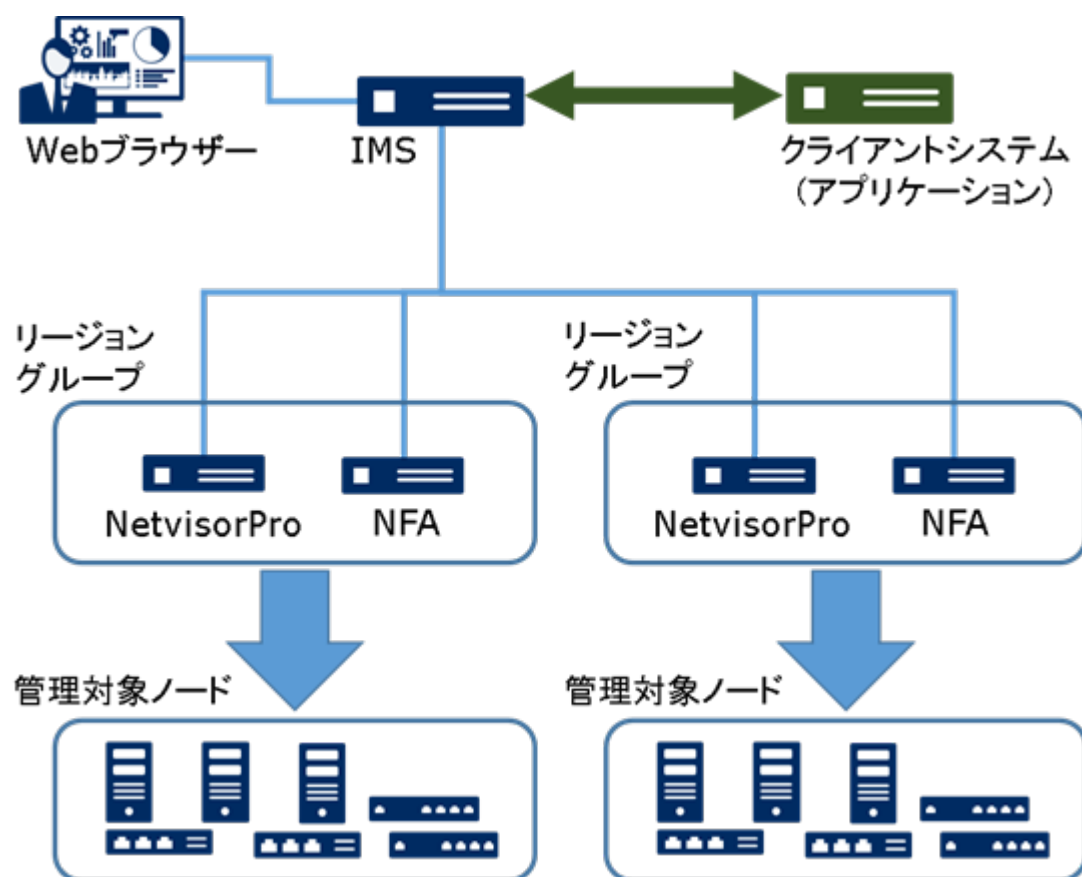


図 1-1 システム構成

## 第2章

# Web API 共通の仕様

本章では、Web API 利用におけるリクエストおよびレスポンスの共通仕様について説明します。

---

### 目次

2.1 リクエスト形式.....	5
2.2 リクエスト、レスポンスのデータ形式.....	13
2.3 レスポンス形式.....	14
2.4 Web API のバージョン .....	17

---

## 2.1 リクエスト形式

Web API に対するリクエストの形式について説明します。

### 2.1.1 エンドポイント

以下の URL を API のエンドポイントとして使用します。

- `http://<IMS サーバーのドメイン名(FQDN)>/<アプリケーション名>/api/v<バージョン>/`
- `https://<IMS サーバーのドメイン名(FQDN)>/<アプリケーション名>/api/v<バージョン>/`

<IMS サーバーのドメイン名(FQDN)>には、Web コンソールにアクセスする場合と同じドメイン名を指定します。

<アプリケーション名>には、以下のいずれかを指定します。

- `nvp`

SNMP データなどの NetvisorPro に関する情報の操作を行う場合に指定します。

- `nfa`

フローデータなどの NFA に関する情報の操作を行う場合に指定します。

- <指定なし>

IMS コンポーネントで管理している構成情報などの共通的な情報を操作する場合は、<アプリケーション名>部分の指定は行いません。

<バージョン>には、Web API のメジャーバージョンを指定します。

例えば、Web API バージョン 1.0 で提供されている共通的な情報を操作する API のに対しては、以下のようなエンドポイントとなります。

```
http://ims.nec.com:8080/api/v1/
```

#### ヒント

Web コンソールにアクセスする通信のポート番号を変更している場合は、上記の例のように、ドメイン名(FQDN)にポート番号を加えて指定します。

### 2.1.2 認証方式

IMS コンポーネントが提供するすべての Web API では、毎回のリクエスト時に認証を必要とします。

HTTP リクエストヘッダーに以下の認証情報を付与することで認証を行います。

```
Date: Date
Authorization: SharedKeyLite AccessKeyId:Signature
```

認証情報は、以下の手順で生成します。

1. Access Key ID、および、Secret Access Key を取得します。

Access Key ID とは、Web API を利用するユーザー固有の API キーのことを指します。


Secret Access Key とは、署名に使われる秘密鍵のことを指します。

- a. Web API を利用するユーザーで Web コンソールにログインします。

### ヒント

当該ユーザーの役割、および、権限の範囲で、Web API を利用することになります。

- b. プロファイル編集画面を表示します。

画面右上の[]アイコンをクリックして表示する[**プロフィール編集**]メニューを選択します。

- c. Web API アクセスキーを発行します。

[**Web API アクセスキー**]の[**発行**]ボタンをクリックします。

- d. Web API アクセスキーを表示します。

[**Web API アクセスキー**]の[**表示**]ボタンをクリックします。

当該ユーザーに対する Access Key ID、および、Secret Access Key が表示されます。

2. リクエスト日時(Date)を決定します。

```
Date: Fri, 21 Aug 2020 13:15:45 GMT
```

Date は、上記のように、IMF-fixdate と呼ばれる形式(RFC7231 参照)で指定します。また、グリニッジ標準時で指定する必要があります。

指定形式の詳細を以下に示します。

```
Date: <day-name>, <day> <month> <year> <hour>:<minute>:<second> GMT
```

### <day-name>

曜日を以下の形式で指定します。

"Mon"、"Tue"、"Wed"、"Thu"、"Fri"、"Sat"、"Sun"

### <day>

日を2桁の数字で指定します。例えば、"04" や"15"のように指定します。

### <month>

月を以下の形式で指定します。

"Jan"、"Feb"、"Mar"、"Apr"、"May"、"Jun"、"Jul"、"Aug"、"Sep"、"Oct"、"Nov"、"Dec"

### <year>

年を4桁の数字で指定します。例えば、"2020"のように指定します。

<hour>

時間を2桁の数字で指定します。"09"や"23"のように指定します。

<minute>

分を2桁の数字で指定します。"05"や"59"のように指定します。

<second>

秒を2桁の数字で指定します。"05"や"59"のように指定します。

### 3. [Authorization]で指定する *Signature* を生成します。

*Signature* は、リクエスト毎に以下の手順で生成する必要があります。

- a. 実行する API のパス(クエリー文字は除く)を決定します。

```
CanonicalizedResource = "/api/v1/nodes"
```

- b. リクエスト日時(Date)と API のパスを改行文字(CRLF)で連結し、認証対象の文字列を生成します。

```
CRLF = "\x0D\x0A"
StringToSign = Date + CRLF + CanonicalizedResource
```

- c. 認証対象の文字列と Secret Access Key から HMAC(SHA256) を生成し Base64 でエンコードします。

```
Signature = Base64(HmacSHA256(SecretAccessKey, StringToSign))
#=> "rpFv195+j7TUV+C1W4LpPb7sBskmjVyUV2jNTz69UfU="
```

### 4. HTTP ヘッダーを生成します。

*AccessKeyId* には、プロファイル編集画面で取得した Access Key ID を指定します。

*Signature* には、リクエスト毎に生成した値を指定します。

```
Date: Fri, 21 Aug 2020 13:15:45 GMT
Authorization: SharedKeyLite Ws/jdOKW1VBwbjwgNq49BS+sc5cLKs9Qh5LeafXDV
Sg=:rpFv195+j7TUV+C1W4LpPb7sBskmjVyUV2jNTz69UfU=
```

## 2.1.3 HTTP バージョン

サポートする HTTP バージョンの詳細を示します。

表 2-1 サポートする HTTP バージョン

プロトコル	HTTP/1.0	HTTP/1.1	HTTP/2
HTTP	—	○	—
HTTPS	—	○	—

## 2.1.4 HTTP メソッド

IMS コンポーネントの Web API では 2 種類の HTTP メソッドを使用します。

表 2-2 使用する HTTP メソッド一覧

メソッド	主な用途
GET	既存のリソースの一覧や詳細を取得するために使用します。
POST	既存のリソースに対し、条件を指定して詳細な情報を取得する場合に使用します。 本メソッドは、GET(一覧、詳細の取得)、PUT(編集)、DELETE(削除)に属さないリソースの操作を行う場合にも使用します。

本バージョンでは、HTTP メソッドの PUT(編集)、および、DELETE(削除)は使用しません。

## 2.1.5 HTTP リクエストヘッダー

Web API に対して HTTP リクエストを送信する際には HTTP ヘッダーが必要です。

o : 必須、\* : 省略可、- : 使用しない

表 2-3 必要な HTTP リクエストヘッダー一覧

ヘッダー	メソッド		説明
	GET	POST	
Content-Type	-	*	リクエスト時のデータフォーマットを指定します。 application/json を指定してください。 省略した場合、または、application/*、*/*を指定した場合は、application/json を指定したものとして処理します。
Content-Length	-	o	リクエストデータのサイズをバイト数で指定します。
Accept	*	*	レスポンス時に受信可能なデータフォーマットを指定します。 application/json を指定してください。 省略した場合、または、application/*、*/*を指定した場合は、application/json を指定したものとして処理します。
Accept-Language	*	*	レスポンス時に受信可能な言語コードを指定します。 ja(日本語)、または、en(英語)が指定可能です。 省略した場合は、en(英語)が指定されたものとして処理します。
Connection	*	*	TCP コネクションを継続利用するかどうかを指定します。 明示的に TCP 接続を終了させたい場合にのみ "close" を指定します。それ以外(ヘッダーの省略を含む)は、持続接続を要求したものとして処理します。
Host	o	o	リクエスト先のサーバのホスト名を指定します。
Date	o	o	リクエストの発行日時を指定します。 例) "Fri, 21 Aug 2020 08:12:31 GMT"

ヘッダー	メソッド		説明
	GET	POST	
Authorization	○	○	認証に使用する情報を指定します。 "SharedKeyLite AccessKeyId:Signature" の形式になります。 詳細は「 <a href="#">2.1.2 認証方式 (5 ページ)</a> 」を参照してください。

## 2.1.6 クエリーパラメーターの指定

Web API の詳細な動作を制御するためのパラメーターは、URL クエリー文字列として指定します。

URL クエリー文字列とは、リクエスト時の URL の後に **key=value** の形式で付与するパラメーターのことを指します。

URL の末尾にクエスチョンマーク(?)を付与し、以降に **key=value** 形式でパラメーターを表現します。複数の URL クエリーを利用する場合は、それぞれのパラメーターをアンパサンド(&)で連結します。

例:

```
GET /api/v1/nodes?sort=name,desc&size=200&page=2
```

利用可能な URL クエリー文字列は、以下の 4 種類に分類することができます。

- プロパティ制御 (props) :  
取得対象のリソースのプロパティを絞り込むクエリーパラメーター。
- フィルター指定 :  
取得対象を条件指定により絞り込むクエリーパラメーター。
- ページ操作 (page, size) :  
ページの概念を用いて、取得するデータ件数を絞り込むクエリーパラメーター。
- ソート (sort) :  
取得対象の一覧情報を指定の順番に並び替えるクエリーパラメーター。

各クエリーパラメーターの具体的な使い方を以下に示します。

### ヒント

各クエリーパラメーターは組み合わせて利用することができます。

## プロパティ制御 (props)

本パラメーターは、取得対象のプロパティを絞り込み、データ量を最小化したい場合に指定します。

指定形式は以下の通りです。

```
props=<プロパティ名>
```

#### <プロパティ名>

取得対象のプロパティ名。

プロパティ名は以下のように指定することができます。

- 階層構造のオブジェクトのプロパティを指定する場合は、上位のオブジェクト名を指定する、もしくは、階層構造をドット(.)区切りで表現して、プロパティ名を指定します。
- 複数のプロパティ名を指定する場合は、コンマ(,)区切りで指定することができます。
- プロパティ名の後ろにエクスクラメーションマーク(!)を付加した場合は、そのプロパティの情報のみ返却しないという意味になります。

例:

```
GET /api/v1/nodes?props=id,properties.name,properties.regionId
```

上記は、ノード一覧取得の Web API において、ノードの ID、ノード名、ノードが所属するリージョン ID の 3 つのプロパティのみのノードリソースの一覧を取得することを表しています。

## フィルター指定

本パラメーターは、特定条件に合致したリソースのみを取得したい場合に指定します。

指定形式は以下の通りです。

- 指定条件に合致したリソースを取得

```
<key>=<フィルター条件>
```

- 指定条件に合致したリソース以外を取得

```
!<Key>=<フィルター条件>
```

#### <key>

フィルター対象とする項目(プロパティ名)。

指定可能な項目は、Web API ごとに異なります。

#### <フィルター条件>

<key>で指定した項目の値に対する条件。

以下のように指定します。

- null と合致することを条件とする場合:



対象: すべての JSON 型

```
<key>=:null
```

- `true` と合致することを条件とする場合:

対象: `boolean` 型

```
<key>=:true
```

- 指定値<*value*>と等しいことを条件とする場合:

対象: `string` 型, `number` 型

```
<key>==<value>
```

- 指定値<*value*>を含む文字列であることを条件とする場合:

対象: `string` 型

```
<key>=~<value>
```

## ヒント

指定値<*value*>を含まない文字列であることを条件とする場合は、先頭にエクスクラメーションマーク(!)を付けて指定します。

```
!<key>=~<value>
```

- 指定値<*value*>から始まる文字列であることを条件とする場合:

対象: `string` 型

```
<key>=^<value>
```

- 指定値<*value*>で終わる文字列であることを条件とする場合:

対象: `string` 型

```
<key>=$<value>
```

- 指定値<*value*>の大小関係を条件とする場合:

対象: `number` 型, `date(iso8601)`型

- 指定値<*value*>以上の値

```
<key>=>=<value>
```

- 指定値<*value*>以下の値

```
<key>=<=<value>
```

## ヒント

指定値<value>よりも大きい、または、小さいという場合は、上記の<フィルター条件>の"="を削除して指定します。

例: 指定値<value>よりも大きい値

```
<key>=><value>
```

- 2つの指定値<value1>と<value2>の間(境界値含む)の値であること条件とする場合:  
対象: number 型, date(iso8601)型

```
<key>=[<value1>,<value2>]
```

複数指定を行った場合は以下のように処理します。

- 同一の<key>に対する条件を複数指定した場合は、"OR 条件"として処理します。
- 異なる<key>に対する条件を複数指定した場合は、"AND 条件"として処理します。

例:

```
GET /api/v1/nodes?properties@name=~switch&properties@name=~router
```

上記は、ノード一覧取得の Web API において、ノード名に"switch"、または、"router"という文字列が含まれているノードリソースのみを一覧で取得することを表しています。

## ページ操作 (page, size)

大量のデータを取り扱う場合がある一覧取得などの Web API においては、ページの実装を採用し、1 度に取得できるデータ件数を制限しています。

本パラメーターは、取得対象のページを指定する場合や 1 ページあたりのデータ件数を既定値から変更したい場合に指定します。

指定形式は以下の通りです。

- 取得するページを指定する：

```
page=<取得対象のページ番号>
```

- 1 ページあたりのデータ件数を既定値から変更する：

```
size=<データ件数>
```

指定可能なデータ件数の上限値は、Web API ごとに異なります。

2つのパラメーターは、同時に指定することも可能です。

例:

```
GET /api/v1/nodes?page=2&size=200
```

上記は、ノード一覧取得の Web API において、201 番目から 400 番目のノードリソースの一覧を取得することを表しています。

## ソート (sort)

本パラメーターは、取得対象の一覧情報を各プロパティ値でソートする場合に指定します。指定形式は以下の通りです。

```
sort=<プロパティ名>,[asc|desc]
```

### <プロパティ名>

ソート対象のプロパティ名。

ソート対象として指定できるプロパティは Web API ごとに制限があります。

#### asc

指定したプロパティの昇順に並び変えます。

#### desc

指定したプロパティの降順に並び変えます。

複数のプロパティに対し、ソートを指定することも可能です。

例:

```
GET /api/v1/nodes?sort=series,asc&sort=swVersion,desc
```

上記は、ノード一覧取得の Web API において、シリーズを昇順にソートし、かつ、ソフトウェアバージョンを降順にソートして、ノードリソースの一覧を取得することを表しています。

## クエリーパラメーターの指定に関する一般的な注意

- 「key=」という指定をした場合は、「key」という API パラメーターに空文字列を指定した扱いになります。多くの API において、指定しない場合と空文字列を指定した場合で動作が一致していますが、API によっては異なる動作になる場合もありますので、各 API の仕様を確認した上で使用してください。
- URL と同様に、URL クエリー文字列も URL エンコーディングの対象となります。

## 2.2 リクエスト、レスポンスのデータ形式

リクエスト・レスポンスのメッセージボディのデータ形式は、以下の通りです。

- 文字列エンコーディング

文字列のエンコーディングは、UTF-8 をサポートしています。

- メッセージボディの形式

RFC4627 で規定されている JSON 形式をサポートしています。

- 日時の形式

日時を表す文字列の指定は、以下で示す ISO8601 の拡張形式をサポートしています。

```
YYYY-MM-DDThh:mm:ssZ
```

- YYYY : 年
- MM : 月
- DD : 日
- hh : 時
- mm : 分
- ss : 秒
- Z : タイムゾーン

例 :

- 日本時刻の場合  
2020-10-15T09:00:00+09:00
- UTC の場合  
2020-10-15T00:00:00Z

## 2.3 レスpons形式

Web API のレスポンスの形式について説明します。

### 2.3.1 HTTP レスponsヘッダー

Web API の HTTP レスponsには、HTTP ヘッダーが付与されます。

o : 付与される、\* : 付与される場合がある

表 2-4 付与される HTTP レスponsヘッダー一覧

ヘッダー	メソッド		説明
	GET	POST	
Content-Type	o	o	レスpons時のデータフォーマットを返却します。 Content-Type: application/json; charset=utf-8
Content-Length	*	*	レスponsデータのサイズをバイト数で返却します。 チャンク転送エンコーディングによってレスponsが返却される場合には付与されません。

ヘッダー	メソッド		説明
	GET	POST	
Transfer-Encoding	*	*	転送に使用するエンコード形式を通知します。 チャンク転送エンコーディングによってレスポンスが返却される場合に付与されます。 Transfer-Encoding: chunked の場合は Content-Length ヘッダーは付与されません。
Connection	*	*	コネクションの状態 (Keep-Alive または close) を通知します。 Connection ヘッダーが付与されない場合は、Connection: Keep-Alive と同義です。 Connection: close を受信したクライアントは、その TCP 接続を再度利用することができないため、再接続が必要となります。

### ⚠ 注意

API の処理が失敗した場合には、「[表 2-4 付与される HTTP レスポンスヘッダー一覧 \(14 ページ\)](#)」の記載に関わらず、一部の HTTP ヘッダーが付与されない場合があります。

## 2.3.2 共通の HTTP ステータスコード

Web API 処理の成功・失敗は、HTTP ステータスコードで通知します。

API 共通の HTTP ステータスコードを「[表 2-5 API が返却する HTTP ステータスコード一覧 \(15 ページ\)](#)」に示します。各 API 固有の HTTP ステータスコードは、「[第 4 章 Web API リファレンス \(33 ページ\)](#)」を参照してください。エラーレスポンスの形式については「[2.3.3 エラーレスポンスの形式 \(16 ページ\)](#)」を参照してください。

表 2-5 API が返却する HTTP ステータスコード一覧

code	説明
400 Bad Request	リクエストの構文が誤っていることを示します。各 API で共通的な発生原因としては以下のことが考えられます。 <ul style="list-style-type: none"> <li>HTTP リクエストのメッセージボディが JSON 形式ではない。(リクエストのメッセージボディが指定された場合のみ)</li> </ul>
401 Unauthorized	ユーザー認証に失敗したことを示します。各 API で共通的な発生原因としては以下のことが考えられます。 <ul style="list-style-type: none"> <li>認証キーが指定されていない。</li> <li>認証キーの内容が不正。</li> </ul>
404 Not Found	指定したリソースが存在していないことを示します。各 API で共通的な発生原因としては以下のことが考えられます。 <ul style="list-style-type: none"> <li>URL の指定内容が不正。</li> </ul>
405 Method Not Allowed	許可されていない HTTP メソッドを指定したことを示します。各 API で共通的な発生原因としては以下のことが考えられます。 <ul style="list-style-type: none"> <li>API のリクエスト形式でサポートする HTTP メソッドを指定していない。</li> </ul>
406 Not Acceptable	リクエストの Accept 関連のヘッダーに受理できない内容が含まれていることを示します。各 API で共通的な発生原因としては以下のことが考えられます。

code	説明
	<ul style="list-style-type: none"> <li>• Accept の指定が、サーバーのサポートするレスポンス形式と一致しない。</li> <li>• Accept-Charset の指定が、サーバーのサポートする文字セットと一致しない。</li> </ul>
411 Length Required	<p>リクエストの Content-Length ヘッダーの不備でサーバーがアクセスを拒否したことを示します。各 API で共通的な発生原因としては以下のことが考えられます。</p> <ul style="list-style-type: none"> <li>• Content-Length ヘッダーの指定がない。</li> </ul>
413 Request Entity Too Large	<p>リクエストエンティティがサーバーの許容範囲を超えていることを示します。各 API で共通的な発生原因としては以下のことが考えられます。</p> <ul style="list-style-type: none"> <li>• リクエストのメッセージボディのサイズが大きすぎる。</li> </ul>
414 Request-URI Too Long	<p>リクエスト URI の長さがサーバーの許容範囲を超えていることを示します。各 API で共通的な発生原因としては以下のことが考えられます。</p> <ul style="list-style-type: none"> <li>• 指定した URL が長すぎる。</li> </ul>
415 Unsupported Media Type	<p>指定したメディアタイプがサーバーでサポートしていないことを示します。各 API で共通的な発生原因としては以下のことが考えられます。</p> <ul style="list-style-type: none"> <li>• リクエストの Content-Type ヘッダーの指定が、サーバーのサポートするメディアタイプに一致しない。</li> </ul>
500 Internal Server Error	<p>API 処理実行時に内部でエラーが発生したことを示します。各 API で共通的な発生原因としては以下のことが考えられます。</p> <ul style="list-style-type: none"> <li>• 予期しないエラーが発生した。</li> </ul>
503 Service Unavailable	<p>サービスが一時的に過負荷になり、使用不可能になったことを示します。各 API で共通的な発生原因としては以下のことが考えられます。</p> <ul style="list-style-type: none"> <li>• サービスの初期化中のため、リクエストを受け付けできなかった。</li> <li>• 大量のリクエストを処理中のため、一時的に新規リクエストを受け付けできなかった。</li> </ul>

### 2.3.3 エラーレスポンスの形式

API の実行がエラーとなった場合は、HTTP レスポンスのステータスラインに「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」に示した 4xx 系(Client Error)または 5xx 系(Server Error)のステータスコードを返却します。また、すべての API で、レスポンスのメッセージボディに、以下の JSON 形式でエラー内容の詳細を返す場合があります<sup>注1</sup>。

#### リソース形式

```
{
  "errorMessage" : string,
  "errorName" : string,
  "target" : string,
  "params" : array,
  "subErrors" : array
}
```

表 2-6 プロパティ

プロパティ名	JSON 型	説明
errorMessage	string	エラーの内容を表すエラーメッセージ。

プロパティ名	JSON 型	説明
その他	-	エラーの補足情報や内部で利用するデータ。

## 注

1. エラー内容によっては、JSON 形式以外(HTML 形式等)のメッセージボディが返却される場合があります。

## 2.4 Web API のバージョン

IMS コンポーネントでは、API バージョンによって製品バージョン間での互換性をコントロールしています。

古い API バージョンを指定して API を呼び出した場合には、指定された API バージョンでの動作をサポートします。

### 対応 API 一覧

- 1.0

以下の情報取得のための API をサポートします。

- 構成管理 API
  - \* 「[4.1.1 リージョングループ一覧取得 \(34 ページ\)](#)」
  - \* 「[4.1.2 ノード一覧取得 \(34 ページ\)](#)」
  - \* 「[4.1.3 ノード詳細取得 \(37 ページ\)](#)」
  - \* 「[4.1.4 ネットワークインターフェース一覧取得 \(38 ページ\)](#)」
  - \* 「[4.1.5 ネットワークインターフェース詳細取得 \(41 ページ\)](#)」
- 障害管理 API
  - \* 「[4.2.1 ノード稼働率 TopN の取得 \(43 ページ\)](#)」
  - \* 「[4.2.2 特定ノードの稼働率の取得 \(45 ページ\)](#)」
- 性能管理 API
  - \* 「[4.3.1 SNMP データ TopN 取得 \(46 ページ\)](#)」
  - \* 「[4.3.2 特定ノードの SNMP データ取得 \(49 ページ\)](#)」
  - \* 「[4.3.3 特定ネットワークインターフェースの SNMP データ取得 \(51 ページ\)](#)」
  - \* 「[4.3.4 フローデータ TopN 取得 \(52 ページ\)](#)」
  - \* 「[4.3.5 特定ノードのフローデータ TopN 取得 \(55 ページ\)](#)」

- \* 「4.3.6 特定ネットワークインターフェ이스のフローデータ TopN 取得 (57 ページ)」

## IMS バージョンとの対応一覧

IMS バージョン	API バージョン
2.0	1.0



# 第3章

## リソース形式

本章では、各 API のリクエスト・レスポンスのメッセージボディのリソース形式について説明します。

---

### 目次

3.1 構成管理のためのリソース .....	20
3.2 性能管理のためのリソース .....	25

---

## 3.1 構成管理のためのリソース

構成管理のための各リソースの形式について説明します。

### 3.1.1 リージョンリソース (RegionObject)

リージョングループの情報を表すリソースの形式を以下に示します。

#### リソース形式

```
{
  "regionId" : {
    "name" : string
  }
}
```

#### プロパティ

プロパティ名	JSON 型	説明
<i>regionId</i>	string	リージョングループを識別するための ID。
name	string	プロパティ名の <i>regionId</i> に対するリージョングループの名前。

### 3.1.2 ノードリソース (NodeObject)

ノードを表すリソースの形式を以下に示します。

#### リソース形式

```
{
  "id" : string,
  "revision" : number,
  "properties" : NodePropertyObject,
  "severity" : "FATAL" | "CRITICAL" | "ERROR" | "WARNING" | "UNKNOWN" |
               "NORMAL",
  "onMap" : true | false,
  "type" : "NODE"
}
```

#### プロパティ

プロパティ名	JSON 型	説明
id	string	ノードを識別するための ID。
revision	number	ノード情報に対するリビジョン。
properties	object	ノードのプロパティ情報(NodePropertyObject)。

プロパティ名	JSON 型	説明
		詳細は、「 <a href="#">NodePropertyObject (21 ページ)</a> 」を参照してください。
severity	string	ノードの現在の状態(重要度)。 以下のいずれかとなります。 "FATAL"、"CRITICAL"、"ERROR"、"WARNING"、 "UNKNOWN"、"NORMAL"
onMap	boolean	トポロジーマップへの登録状況。 <ul style="list-style-type: none"> <li>• true: トポロジーマップにノードが登録されていることを示します。</li> <li>• false: トポロジーマップにノードが登録されていないことを示します。</li> </ul>
type	string	オブジェクトのタイプ。 ノードであることを示す"NODE"となります。

## NodePropertyObject

- NodePropertyObject のオブジェクト形式

```
{
  "regionId" : string,
  "regionName" : string,
  "name" : string,
  "dnsName" : string | null,
  "sysName" : string | null,
  "ipv4Address" : string | null,
  "ipv6Address" : string | null,
  "virtualIpv4Address" : string | null,
  "nodeType" : "Access Point" | "Firewall" | "IP Phone" |
    "L2 Switch" | "L3 Switch" | "Load Balancer" |
    "Network Appliance" | "Network Client" |
    "Network Device" | "Router" | "Server" | "Storage" |
    "VM" | "Wireless LAN Controller",
  "vendor" : string,
  "series" : string,
  "serialNumber" : string | null,
  "sysDescr" : string | null,
  "sysObjectId" : string | null,
  "swVersion" : string | null,
  "administrator" : string | null,
  "location" : string | null,
  "memo" : string | null,
  "url" : string | null,
  "snmpVersion" : "1" | "2c" | "3" | null,
  "snmpPort" : number | null,
  "snmpCommunityGet" : string | null,
  "snmpCommunitySet" : string | null,
  "snmpUserName" : string | null,
  "snmpEngineId" : string | null,
  "snmpSecurityLevel" : "NoAuth/NoPriv" | "Auth/NoPriv" |
    "Auth/Priv",
  "snmpAuthenticationProtocol" : "MD5" | "SHA1" | null,
```

```

    "snmpPrivacyProtocol" : "DES" | null,
    "dpid" : string | null,
    "fexId" : number | null
  }

```

- NodePropertyObject のプロパティ

プロパティ名	JSON 型	説明
regionId	string	ノードが所属するリージョングループを識別するための ID。
regionName	string	ノードが所属するリージョングループ名。
name	string	ノードの表示名。 255 文字以内の文字列で示します。
dnsName	string	DNS(Domain Name System)に問い合わせ取得したホスト名 (FQDN)。
sysName	string	ノードの MIB(1.3.6.1.2.1.1.5)から SNMP を用いて取得したホスト名。 255 文字以内の文字列で示します。
ipv4Address	string	ノードの IPv4 アドレス。 192.168.0.1 のように、IPv4 アドレス形式で示します。
ipv6Address	string	ノードの IPv6 アドレス。 0～9、a～e、A～E、コロン(:)の組み合わせのグローバルユニキャストアドレスで示します。
virtualIpv4Address	string	ノードの仮想 IPv4 アドレス。 192.168.0.1 のように、IPv4 アドレス形式で示します。
nodeType	string	ノードの装置種別。 以下のいずれかとなります。 "Access Point"、"Firewall"、"IP Phone"、"L2 Switch"、 "L3 Switch"、"Load Balancer"、"Network Appliance"、 "Network Client"、"Network Device"、"Router"、 "Server"、"Storage"、"VM"、"Wireless LAN Controller"
vendor	string	ベンダー名。
series	string	装置のシリーズ名。
serialNumber	string	シリアル番号。 255 文字以内の文字列で示します。
sysDescr	string	ノードの MIB(1.3.6.1.2.1.1.1)から SNMP を用いて取得した装置情報。
sysObjectId	string	ノードの MIB(1.3.6.1.2.1.1.2)から SNMP を用いて取得した機種種の識別 ID。
swVersion	string	ノードのソフトウェア(ファームウェア)のバージョン情報。 255 文字以内の文字列で示します。
administrator	string	ノードの管理者情報。 255 文字以内の文字列で示します。
location	string	ノードの設置場所。 255 文字以内の文字列で示します。
memo	string	ノードに対するメモ情報。 255 文字以内の文字列で示します。

プロパティ名	JSON 型	説明
url	string	ノードの管理情報として登録されている URL。 2083 文字以内の文字列で示します。
snmpVersion	string	ノードにアクセス可能な SNMP のバージョン。 以下のいずれかとなります。 "1"、"2c"、"3"
snmpPort	number	SNMP による Get、および、Set で利用するポート番号。 1～65535 の値で示します。 このプロパティ値が null の場合は、SNMP 通信時に 161 番ポートを利用して処理します。
snmpCommunityGet	string	SNMP による Get 用のコミュニティ名。 255 文字以下の文字列で示します。 プロパティ値に空文字列を指定している場合は、空文字列をコミュニティ名として SNMP 通信を行うことを示します。
snmpCommunitySet	string	SNMP による Set 用のコミュニティ名。 255 文字以下の文字列で示します。 プロパティ値に空文字列を指定している場合は、空文字列をコミュニティ名として SNMP 通信を行うことを示します。
snmpUserName	string	SNMPv3 のユーザー名。 印字可能な ASCII の範囲の 1 文字以上 32 文字以内の文字列で示します。
snmpEngineId	string	SNMPv3 のエージェントを識別するためのエンジン ID。 先頭に 0x を付加し、2 桁の値をコロン(:)区切りで示します。 例: 0x11:22:33:44:55
snmpSecurityLevel	string	SNMPv3 通信で使用するセキュリティレベル。 以下のいずれかとなります。 <ul style="list-style-type: none"> <li>• "NoAuth/NoPriv" : 認証、暗号化を行わない</li> <li>• "Auth/NoPriv" : 認証を行い、暗号化は行わない</li> <li>• "Auth/Priv" : 認証、暗号化を行う</li> </ul>
snmpAuthenticationProtocol	string	SNMPv3 で使用する認証プロトコル。 以下のいずれかとなります。 "MD5"、"SHA1"
snmpPrivacyProtocol	string	SNMPv3 通信で利用する暗号化プロトコル。 以下の文字列となります。 "DES"
dpid	string	OpenFlow 対応装置の Datapath ID。 "0000-0000-0000-0000"～"ffff-ffff-ffff-ffff"の範囲で示します。
fexId	number	Cisco Nexus2000 シリーズを識別するための ID。 0～4294967295 の範囲の数字で示します。

### 3.1.3 ネットワークインターフェイスリソース (NetworkInterfaceObject)

ネットワークインターフェイスリソースを表すリソースの形式を以下に示します。

#### リソース形式

```
{
  "id" : string,
  "revision" : number,
  "properties" : InterfacePropertyObject,
  "nodeId" : string,
  "type" : "NETWORK_INTERFACE"
}
```

#### プロパティ

プロパティ名	JSON 型	説明
id	string	ネットワークインターフェイスを識別するための ID。
revision	number	ネットワークインターフェイス情報に対するリビジョン。
properties	object	ネットワークインターフェイスのプロパティ情報 (InterfacePropertyObject)。 詳細は、「 <a href="#">InterfacePropertyObject (24 ページ)</a> 」を参照してください。
nodeId	string	ネットワークインターフェイスを実装しているノードを識別するための ID。
type	string	オブジェクトのタイプ。 ネットワークインターフェイスであることを示す "NETWORK_INTERFACE" となります。

#### InterfacePropertyObject

- InterfacePropertyObject のオブジェクト形式

```
{
  "regionId" : string,
  "regionName" : string,
  "ifIndex" : number,
  "name" : string,
  "ifName" : string | null,
  "description" : string | null,
  "type" : string,
  "speed" : number,
  "macAddress" : string | null,
  "ipv4Addresses" : [
    {
      "ipv4Address" : string,
      "ipv4Subnetmask" : number,
    },
  ],
}
```

```

    ...
  ]
}

```

- InterfacePropertyObject のプロパティ

プロパティ名	JSON 型	説明
regionId	string	ネットワークインターフェイスを実装しているノードが所属するリージョングループを識別するための ID。
regionName	string	ネットワークインターフェイスを実装しているノードが所属するリージョングループ名。
ifIndex	number	ネットワークインタフェースのインデックス。
name	string	ネットワークインターフェイスの表示名。
ifName	string	ノードの MIB(1.3.6.1.2.1.31.1.1.1.1)から SNMP を用いて取得したネットワークインターフェイスの名前。
description	string	ノードの MIB(ifDescr: 1.3.6.1.2.1.2.2.1.2)から SNMP を用いて取得したネットワークインターフェイスの説明情報。
type	string	ネットワークインターフェイスのタイプ。
speed	number	ネットワークインターフェイスの帯域幅。 値の単位は、Kbps です。
macAddress	string	ネットワークインターフェイスの MAC アドレス。
ipv4Addresses	object	ネットワークインターフェイスに紐づく IPv4 アドレスの配列。 プロパティの詳細は、「ipv4Addresses のプロパティ」を参照してください。 IPv4 アドレスが割り当てられていない場合は、空の配列となります。

- ipv4Addresses のプロパティ

プロパティ名	JSON 型	説明
ipv4Address	string	ネットワークインターフェイスの IPv4 アドレス。 192.168.0.1 のように、IPv4 アドレス形式で示します。
ipv4Subnetmask	number	ネットワークインターフェイスのサブネットマスク。 CIDR (Classless InterDomain Routing) 表記の数値で示します。 例: 24 (255.255.255.0)

## 3.2 性能管理のためのリソース

性能管理のための各リソースの形式について説明します。

### 3.2.1 ウィジェットデータリソース (WidgetDataObject)

ウィジェットが利用している統計情報を表すリソースの形式を以下に示します。

## リソース形式

```
{
  "statisticsWidgetData" : {
    "LINE" : StatisticsInfoObject,
    "PIE" : StatisticsInfoObject,
    "NONE" : StatisticsInfoObject
  },
  "dnsResolving" : HostsObject | null
}
```

## プロパティ

プロパティ名	JSON 型	説明
statisticsWidgetData	object	指定範囲における全体の統計情報を保持するオブジェクト。
statisticsWidgetData.LINE	object	指定範囲における折れ線グラフ用の統計情報を保持するオブジェクト。 フローデータの場合にのみ定義されます。 詳細は、「 <a href="#">3.2.2 統計情報リソース (StatisticsInfoObject)</a> (27 ページ)」を参照してください。
statisticsWidgetData.PIE	object	指定範囲における円グラフ用の統計情報を保持するオブジェクト。 フローデータにおける以下のデータ種別(data_type)の場合にのみ定義されます。 <ul style="list-style-type: none"> <li>• アプリケーション (application)</li> <li>• IP プロトコル (ip-protocol)</li> <li>• DSCP (dscp)</li> </ul> 詳細は、「 <a href="#">3.2.2 統計情報リソース (StatisticsInfoObject)</a> (27 ページ)」を参照してください。
statisticsWidgetData.NONE	object	指定範囲における汎用チャート用の統計情報を保持するオブジェクト。 ノードの稼働率に関するデータにのみ定義されます。 詳細は、「 <a href="#">3.2.2 統計情報リソース (StatisticsInfoObject)</a> (27 ページ)」を参照してください。
dnsResolving	object	統計情報の中に含まれている IPv4 アドレスに対するホスト名情報のオブジェクト。 詳細は、「 <a href="#">HostsObject</a> (26 ページ)」を参照してください。

## HostsObject

- HostsObject のオブジェクト形式

```
{
  "IPv4 Address" : string,
  ...
}
```

- HostsObject のプロパティ



プロパティ名	JSON 型	説明
<i>IPv4 Address</i>	string	プロパティ名の IPv4 アドレスに対するホスト名。 プロパティの <b>key</b> には、統計データ中に含まれている IPv4 アドレスで、かつ、DNS による名前解決が行えた IPv4 アドレスが列挙されます。

- 例:

```
{
  "192.168.10.1" : "host1.co.jp",
  "192.168.10.2" : "host2.co.jp",
  "192.168.10.3" : "host3.co.jp"
}
```

## 3.2.2 統計情報リソース (StatisticsInfoObject)

統計情報を表すリソースの形式を以下に示します。

### リソース形式

```
{
  "rows" : [ StatisticItemObject ],
  "data" : [ StatisticDataObject ],
  "otherData" : null | [
    {
      "value" : number
    }
  ]
}
```

### プロパティ

プロパティ名	JSON 型	説明
rows	object[]	統計項目(StatisticItemObject)の配列。 詳細は、「 <a href="#">3.2.3 統計項目リソース (StatisticItemObject) (28 ページ)</a> 」を参照してください。
data	object[]	統計データ(StatisticDataObject)の配列。 詳細は、「 <a href="#">3.2.4 統計データリソース (StatisticDataObject) (31 ページ)</a> 」を参照してください。
otherData	object[]	指定したランキング(Top N)におけるその他のデータの配列。 NetvisorPro が収集した SNMP データ(性能データ)を示す場合は、定義されません。 NFA が受信したフローデータを示す場合は、以下となります。 <ul style="list-style-type: none"> <li>• 折れ線グラフ用データ(LINE)の場合は、必ず、<b>null</b> となります。</li> <li>• 円グラフ用データ(PIE)の場合で、かつ、その他のデータが存在する場合に、要素数 1 の配列となります。その他のデータが存在しない場合は、<b>null</b> となります。</li> </ul> その他の統計情報を示す場合は、必ず、 <b>null</b> となります。

プロパティ名	JSON 型	説明
otherData[0].value	number	指定範囲におけるランキング外データの合計値。

### 3.2.3 統計項目リソース (StatisticItemObject)

統計項目を表すリソースの形式を以下に示します。

#### リソース形式

```
{
  "id" : string,
  "statisticsValue" : number | null,
  Other properties
}
```

#### プロパティ

プロパティ名	JSON 型	説明
id	string	統計項目を識別するための ID。
statisticsValue	number	折れ線グラフ用データの場合は、指定範囲での平均値。 円グラフ用データの場合は、指定範囲の合計値。 汎用チャート用データの場合は、指定範囲での対象項目の値。 指定範囲にデータが存在しない場合は、null となります。
Other properties	-	データ種別ごとの追加プロパティ。 データ種別ごとに追加されるプロパティの詳細は「 <a href="#">Other properties (28 ページ)</a> 」に示します。

#### Other properties

データ種別ごとに追加されるプロパティを以下に示します。

- ノードの稼働率

ノードの稼働率に対する追加プロパティを以下に示します。

プロパティ名	JSON 型	説明
name	string	取得した稼働率に対するノードの表示名。

- SNMP データ

NetvisorPro が収集した SNMP データの統計項目に対する追加プロパティを以下に示します。

プロパティ名	JSON 型	説明
nodeId	string	統計データを取得したノードを識別するための ID。 「 <a href="#">3.1.2 ノードリソース (NodeObject) (20 ページ)</a> 」の "id" となります。

プロパティ名	JSON 型	説明
		IMS コンポーネントと NetvisorPro との構成情報の同期が行われていない場合は、null となります。
nodeName	string	統計データを取得したノードの表示名。 IMS コンポーネントと NetvisorPro との構成情報の同期が行われていない場合は、null となります。
instanceId	string	本項目のインスタンス ID。 ネットワークインターフェイスの場合は、ネットワークインターフェイスを識別するための「 <a href="#">3.1.3 ネットワークインターフェイスリソース (NetworkInterfaceObject) (24 ページ)</a> 」の "id" となります。 ネットワークインターフェイスではない場合は、null となります。また、ネットワークインターフェイスの場合であっても IMS コンポーネントと NetvisorPro との構成情報の同期が行われていない場合は、null となります。
instanceName	string	本項目のインスタンス名。 ネットワークインターフェイスの場合は、ネットワークインターフェイス名となります。 ネットワークインターフェイスではない場合は、自動的に付加された文字列となります。 ネットワークインターフェイスの場合で、かつ、IMS コンポーネントと NetvisorPro との構成情報の同期が行われていない場合は、null となります。

- エクスポーター (フローデータ)

NFA が受信したフローデータにおけるエクスポーターの通信量に対する追加プロパティを以下に示します。

プロパティ名	JSON 型	説明
nodeId	string	統計データを取得したノードを識別するための ID。 「 <a href="#">3.1.2 ノードリソース (NodeObject) (20 ページ)</a> 」の "id" となります。 IMS コンポーネントと NFA との構成情報の同期が行われていない場合は、null となります。
nodeName	string	統計データを取得したノードの表示名。 IMS コンポーネントと NFA との構成情報の同期が行われていない場合は、null となります。
originalExporterId	string	統計データを取得したノードの NFA 上でのエクスポーター ID。
originalExporterName	string	統計データを取得したノードの NFA 上でのエクスポーター表示名。

- 入力インターフェイス、出力インターフェイス (フローデータ)

NFA が受信したフローデータにおける入出力インターフェイスの通信量に対する追加プロパティを以下に示します。

プロパティ名	JSON 型	説明
nodeId	string	統計データを取得したノードを識別するための ID。 「 <a href="#">3.1.2 ノードリソース (NodeObject) (20 ページ)</a> 」の "id" となります。

プロパティ名	JSON 型	説明
		IMS コンポーネントと NFA との構成情報の同期が行われていない場合は、null となります。
nodeName	string	統計データを取得したノードの表示名。 IMS コンポーネントと NFA との構成情報の同期が行われていない場合は、null となります。
interfaceId	string	統計データを取得したネットワークインターフェイスの ID。 IMS コンポーネントと NFA との構成情報の同期が行われていない場合は、null となります。
interfaceName	string	統計データを取得したネットワークインターフェイス名。 IMS コンポーネントと NFA との構成情報の同期が行われていない場合は、null となります。
originalExporterId	string	統計データを取得したノードの NFA 上でのエクスポーター ID。
originalExporterName	string	統計データを取得したノードの NFA 上でのエクスポーター表示名。
originalInterfaceId	string	統計データを取得したネットワークインターフェイスの NFA 上での ID。
originalInterfaceName	string	統計データを取得したネットワークインターフェイスの NFA 上での名前。

- アプリケーション (フローデータ)

NFA が受信したフローデータにおけるアプリケーションの通信量に対する追加プロパティを以下に示します。

プロパティ名	JSON 型	説明
name	string	取得した統計データに対するアプリケーション名。
entityId	string	取得した統計データに対するアプリケーションの NFA 上での ID。

- IP プロトコル (フローデータ)

NFA が受信したフローデータにおける IP プロトコルの通信量に対する追加プロパティを以下に示します。

プロパティ名	JSON 型	説明
name	string	取得した統計データに対する IP プロトコル名。
entityId	string	取得した統計データに対する IP プロトコル番号。

- DSCP (フローデータ)

NFA が受信したフローデータにおける DSCP の通信量に対する追加プロパティを以下に示します。

プロパティ名	JSON 型	説明
name	string	取得した統計データに対する DSCP 表示名。
entityId	string	取得した統計データに対する DSCP 値。 10 進数表記で示します。

- カンバセーション (フローデータ)

NFA が受信したフローデータにおけるカンバセーション(2 点間)の通信量に対する追加プロパティを以下に示します。

プロパティ名	JSON 型	説明
srcIpAddress	string	取得した統計データに対する送信元 IPv4 アドレス。
dstIpAddress	string	取得した統計データに対する宛先 IPv4 アドレス。

- 送信元 IP アドレス、宛先 IP アドレス (フローデータ)

NFA が受信したフローデータにおける送信元、または、宛先 IP アドレスの通信量に対する追加プロパティを以下に示します。

プロパティ名	JSON 型	説明
ipAddress	string	取得した統計データに対する送信元、または、宛先 IPv4 アドレス。

- 送信元エンドポイントグループ、宛先エンドポイントグループ (フローデータ)

NFA が受信したフローデータにおける送信元、または、宛先エンドポイントグループの通信量に対する追加プロパティを以下に示します。

プロパティ名	JSON 型	説明
name	string	取得した統計データに対する送信元、または、宛先エンドポイントグループの表示名。
entityId	string	取得した統計データに対する送信元、または、宛先エンドポイントグループの ID。

- 送信元 AS、宛先 AS (フローデータ)

NFA が受信したフローデータにおける送信元、または、宛先 AS の通信量に対する追加プロパティを以下に示します。

プロパティ名	JSON 型	説明
name	string	取得した統計データに対する送信元、または、宛先 AS 番号。10 進数表記で示します。

### 3.2.4 統計データリソース (StatisticDataObject)

統計データを表すリソースの形式を以下に示します。

#### リソース形式

```
{
  "id" : string,
  "value" : number,
  "time" : iso8601 | null
}
```

## プロパティ

プロパティ名	JSON 型	説明
id	string	この統計データに対する統計項目を識別するための ID。 「 <a href="#">3.2.3 統計項目リソース (StatisticItemObject) (28 ページ)</a> 」の "id" と一致します。
value	number	"id" が一致する統計項目に対する統計値。
time	iso8601	統計データに対応する時刻情報。 ISO8601 の拡張形式の UTC (協定世界時 : Coordinated universal time) 時刻で示します。

# 第4章

## Web API リファレンス

本章では、各 API の詳細な仕様について説明します。

---

### 目次

4.1 構成管理 API.....	34
4.2 障害管理 API.....	43
4.3 性能管理 API.....	46

---

## 4.1 構成管理 API

ノードやネットワークインターフェ이스の構成情報を操作するための Web API の仕様について説明します。

### 4.1.1 リージョングループ一覧取得

リージョングループの情報を一覧で取得します。

#### 必要な権限

オブザーバー以上

#### リクエスト形式

```
GET /api/v1/regions
```

- 利用可能なクエリーパラメーター  
なし。
- ボディ  
なし。

#### レスポンス形式

- ステータスコード

code	意味	説明
200	成功	一覧の取得に成功したことを示します。

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

```
{  
  "data" : { RegionObject }  
}
```

各プロパティの詳細は、「[3.1.1 リージョンリソース \(RegionObject\) \(20 ページ\)](#)」を参照してください。

### 4.1.2 ノード一覧取得

ノードの情報を一覧で取得します。



クエリーパラメーターを指定することで、条件に合致したノードのみの情報を取得することができます。

## 必要な権限

オブザーバー以上

## リクエスト形式

GET /api/v1/nodes

- 利用可能なクエリーパラメーター

パラメーター名	省略可否	説明
props	省略可	<p>取得するノードのプロパティ情報を絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">プロパティ制御 (props) (9 ページ)</a>」を参照してください。</p> <p>例えば、ノードを特定し、ノードに対する ID を取得したい場合は、以下のように指定します。</p> <pre>props=id,properties.name,properties.regionId</pre>
page	省略可	<p>取得対象のページを指定するためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">ページ操作 (page, size) (12 ページ)</a>」を参照してください。</p> <p>1 ページあたりのデータ量は、既定値で、100 ノード分の情報となっています。本パラメーターを指定しない場合は、1 ページ目の情報を返却します。</p> <p>例えば、101 番目から 200 番目のノード情報を取得する場合は、以下のように指定します。</p> <pre>page=2</pre>
size	省略可	<p>1 ページあたりのデータ量を変更するためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">ページ操作 (page, size) (12 ページ)</a>」を参照してください。</p> <p>1~250 の範囲で指定することができます。</p> <p>例えば、1 ページあたり 200 ノード分の情報を返却するように設定する場合は、以下のように指定します。</p> <pre>size=200</pre> <p>指定内容が不正、または、本パラメーターを指定しなかった場合は、1 ページあたり 100 ノード分の情報を返却します。</p>
sort	省略可	<p>ノード情報を指定条件でソートして取得するためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">ソート (sort) (13 ページ)</a>」を参照してください。</p> <p>ソート対象のプロパティとしては、以下を指定することができます。</p> <ul style="list-style-type: none"> <li>name</li> </ul>

パラメーター名	省略可否	説明
		<p>ノード名。</p> <ul style="list-style-type: none"> <li>• <b>regionName</b> ノードが所属するリージョングループの名前。</li> <li>• <b>ipv4Address</b> ノードの IPv4 アドレス。</li> <li>• <b>severity</b> ノードの現在の状態(重要度)。</li> <li>• <b>nodeType</b> ノードの装置種別。</li> <li>• <b>vendor</b> 装置のベンダー名。</li> <li>• <b>series</b> 装置のシリーズ名。</li> <li>• <b>swVersion</b> ソフトウェア(ファームウェア)のバージョン情報。</li> </ul> <p>例えば、ノード名の降順でソートする場合は、以下のように指定します。</p> <p><code>sort=name,desc</code></p>
properties@name	省略可	<p>取得対象をノード名で絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">フィルター指定 (10 ページ)</a>」を参照してください。</p>
properties@nodeType	省略可	<p>取得対象をノードの装置種別で絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">フィルター指定 (10 ページ)</a>」を参照してください。</p>
properties@vendor	省略可	<p>取得対象を装置のベンダー名で絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">フィルター指定 (10 ページ)</a>」を参照してください。</p>
properties@series	省略可	<p>取得対象を装置のシリーズ名で絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">フィルター指定 (10 ページ)</a>」を参照してください。</p>
properties@swVersion	省略可	<p>取得対象を装置のソフトウェア(ファームウェア)のバージョン情報で絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">フィルター指定 (10 ページ)</a>」を参照してください。</p>
properties@regionName	省略可	<p>取得対象をノードが所属するリージョン名で絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">フィルター指定 (10 ページ)</a>」を参照してください。</p>
severity	省略可	<p>取得対象をノードの現在の状態(重要度)で絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">フィルター指定 (10 ページ)</a>」を参照してください。</p>
ipv4Address	省略可	<p>取得対象をノードの IPv4 アドレスで絞り込むためのパラメーター。</p>

パラメーター名	省略可否	説明
		指定形式の詳細は、「 <a href="#">フィルター指定 (10 ページ)</a> 」を参照してください。

## ヒント

本 Web API を利用する際、レスポンス時のデータ量を最小にするため、幾つかのクエリーパラメーターを指定することを推奨します。

- ボディ  
なし。

## レスポンス形式

- ステータスコード

code	意味	説明
200	成功	一覧の取得に成功したことを示します。 結果が 0 件の場合も 200 を返却します。
400	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>クエリーパラメーターの <code>severity</code> に指定した重要度値が不正。</li> <li>クエリーパラメーターの <code>ipv4Address</code> に指定した値が IPv4 アドレス形式ではない。</li> <li>クエリーパラメーターの <code>sort</code> に指定した項目名が不正。</li> </ul>

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

```
{
  "totalCount" : number,
  "data" : [ NodeObject ]
}
```

プロパティ名	JSON 型	説明
totalCount	number	クエリーパラメーターで指定した条件に合致するノードの総数。  size で指定した値、または、100 よりも大きい値の場合は、page を用いた取得が必要になります。
data	object	ノードのリソース情報。 各プロパティの詳細は、「 <a href="#">3.1.2 ノードリソース (NodeObject) (20 ページ)</a> 」を参照してください。

### 4.1.3 ノード詳細取得

指定したノードの詳細なプロパティを取得します。

## 必要な権限

オブザーバー以上

## リクエスト形式

GET /api/v1/nodes/{node\_id}

- API パラメーター

パラメーター名	説明
node_id	取得対象のノードの ID。 「 <a href="#">3.1.2 ノードリソース (NodeObject) (20 ページ)</a> 」の id を指定します。

- 利用可能なクエリーパラメーター  
なし。
- ボディ  
なし。

## レスポンス形式

- ステータスコード

code	意味	説明
200	成功	プロパティの取得に成功したことを示します。
404	失敗	指定した id のノードが存在しないことを示します。

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

```
{
  "data" : NodeObject
}
```

プロパティ名	JSON 型	説明
data	object	ノードのリソース情報。 各プロパティの詳細は、「 <a href="#">3.1.2 ノードリソース (NodeObject) (20 ページ)</a> 」を参照してください。

### 4.1.4 ネットワークインターフェース一覧取得

ネットワークインターフェースの情報を一覧で取得します。

クエリーパラメーターを指定することで、条件に合致したネットワークインターフェイスのみの情報を取得することができます。

## 必要な権限

オブザーバー以上

## リクエスト形式

```
GET /api/v1/nodes/{node_id}/interfaces
```

- API パラメーター

パラメーター名	説明
node_id	取得対象のノードの ID。 「 <a href="#">3.1.2 ノードリソース (NodeObject) (20 ページ)</a> 」の id を指定します。

- 利用可能なクエリーパラメーター

パラメーター名	省略可否	説明
props	省略可	取得するネットワークインターフェイスのプロパティ情報を絞り込むためのパラメーター。 指定形式の詳細は、「 <a href="#">プロパティ制御 (props) (9 ページ)</a> 」を参照してください。 例えば、ネットワークインターフェイスを特定し、ネットワークインターフェイスに対する ID を取得したい場合は、以下のように指定します。  props=id,properties.ifIndex,properties.name
page	省略可	取得対象のページを指定するためのパラメーター。 指定形式の詳細は、「 <a href="#">ページ操作 (page, size) (12 ページ)</a> 」を参照してください。 1 ページあたりのデータ量は、既定値で、20 インターフェイス分の情報となっています。本パラメーターを指定しない場合は、1 ページ目の情報を返却します。 例えば、21 番目から 40 番目のノード情報を取得する場合は、以下のように指定します。  page=2
size	省略可	1 ページあたりのデータ量を変更するためのパラメーター。 指定形式の詳細は、「 <a href="#">ページ操作 (page, size) (12 ページ)</a> 」を参照してください。 1~2,000 の範囲で指定することができます。 例えば、1 ページあたり 100 インターフェイス分の情報を返却するように設定する場合は、以下のように指定します。  size=100

パラメーター名	省略可否	説明
		指定内容が不正、または、本パラメーターを指定しなかった場合は、1 ページあたり 20 インターフェイス分の情報を返却します。
sort	省略可	<p>ネットワークインターフェイス情報を指定条件でソートして取得するためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">ソート (sort) (13 ページ)</a>」を参照してください。</p> <p>ソート対象のプロパティとしては、以下を指定することができます。</p> <ul style="list-style-type: none"> <li>• name ネットワークインターフェイス名。</li> <li>• type ネットワークインターフェイスのタイプ。</li> <li>• ifIndex ネットワークインターフェイスのインデックス。</li> <li>• speed ネットワークインターフェイスの帯域幅。</li> <li>• macAddress ネットワークインターフェイスの MAC アドレス。</li> </ul> <p>例えば、ネットワークインターフェイスの帯域幅の降順でソートする場合は、以下のように指定します。</p> <pre>sort=speed, desc</pre>
properties@name	省略可	<p>取得対象をネットワークインターフェイス名で絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">フィルター指定 (10 ページ)</a>」を参照してください。</p>
properties@type	省略可	<p>取得対象をネットワークインターフェイスのタイプで絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">フィルター指定 (10 ページ)</a>」を参照してください。</p>
properties@ifIndex	省略可	<p>取得対象をネットワークインターフェイスのインデックスで絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">フィルター指定 (10 ページ)</a>」を参照してください。</p>
properties@speed	省略可	<p>取得対象をネットワークインターフェイスの帯域幅で絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">フィルター指定 (10 ページ)</a>」を参照してください。</p>
properties@macAddress	省略可	<p>取得対象をネットワークインターフェイスの MAC アドレスで絞り込むためのパラメーター。</p> <p>指定形式の詳細は、「<a href="#">フィルター指定 (10 ページ)</a>」を参照してください。</p>

## ヒント

本 Web API を利用する際、レスポンス時のデータ量を最小にするため、幾つかのクエリーパラメーターを指定することを推奨します。

- ボディ  
なし。

## レスポンス形式

- ステータスコード

code	意味	説明
200	成功	一覧の取得に成功したことを示します。 結果が 0 件の場合も 200 を返却します。
400	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>クエリーパラメーターの <code>properties@ifIndex</code> に指定した値が数値ではない。</li> <li>クエリーパラメーターの <code>properties@speed</code> に指定した値が数値ではない。</li> <li>クエリーパラメーターの <code>sort</code> に指定した項目名が不正。</li> </ul>
404	失敗	指定した <code>id</code> のノードが存在しないことを示します。

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

```
{
  "totalCount" : number,
  "data" : [ NetworkInterfaceObject ]
}
```

プロパティ名	JSON 型	説明
totalCount	number	クエリーパラメーターで指定した条件に合致するノードの総数。  size で指定した値、または、20 よりも大きい値の場合は、page を用いた取得が必要になります。
data	object	ネットワークインターフェイスのリソース情報。 各プロパティの詳細は、「 <a href="#">3.1.3 ネットワークインターフェイスリソース (NetworkInterfaceObject) (24 ページ)</a> 」を参照してください。

### 4.1.5 ネットワークインターフェイス詳細取得

指定したネットワークインターフェイスの詳細なプロパティを取得します。

## 必要な権限

オブザーバー以上

## リクエスト形式

```
GET /api/v1/nodes/{node_id}/interfaces/{if_id}
```

- API パラメーター

パラメーター名	説明
node_id	取得対象のネットワークインターフェイスを持っているノードの ID。 「 <a href="#">3.1.2 ノードリソース (NodeObject) (20 ページ)</a> 」の id を指定します。
if_id	取得対象のネットワークインターフェイスの ID。 「 <a href="#">3.1.3 ネットワークインターフェイスリソース (NetworkInterfaceObject) (24 ページ)</a> 」の id を指定します。

- 利用可能なクエリーパラメーター

なし。

- ボディ

なし。

## レスポンス形式

- ステータスコード

code	意味	説明
200	成功	プロパティの取得に成功したことを示します。
404	失敗	指定した id のネットワークインターフェイスが存在しないことを示します。

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

```
{
  "data" : NetworkInterfaceObject
}
```

プロパティ名	JSON 型	説明
data	object	ネットワークインターフェイスのリソース情報。 各プロパティの詳細は、「 <a href="#">3.1.3 ネットワークインターフェイスリソース (NetworkInterfaceObject) (24 ページ)</a> 」を参照してください。



## 4.2 障害管理 API

イベントやノードの稼働率などの障害情報を操作するための Web API の仕様について説明します。

### 4.2.1 ノード稼働率 TopN の取得

稼働率の低いノードの TopN を取得します。

#### 必要な権限

オブザーバー以上

#### リクエスト形式

POST /api/v1/nodes/availabilities/search

- 利用可能なクエリーパラメーター  
なし。
- ボディ

```
{
  "nodeIds" : [ string ] | null,
  "startDate" : iso8601,
  "endDate" : iso8601,
  "topN" : number | null,
  "property" : string | null,
  "direction" : string | null
}
```

プロパティ名	JSON 型	省略可否	説明
nodeIds	string[]	省略可	取得対象ノードを示す ID の配列。 取得対象として、複数の ID を指定することができ、指定した ID に合致するノードの稼働率のみを取得します。 ID は、最大 250 件まで指定することができます。 省略時は、すべてのノードが取得対象となります。
startDate	iso8601	必須	稼働率の計算対象とする期間の開始時刻。 ISO8601 の拡張形式で指定します。
endDate	iso8601	必須	稼働率の計算対象とする期間の終了時刻。 ISO8601 の拡張形式で指定します。
topN	number	省略可	稼働率の低い上位ランキングのデータ件数(TopN)。 1~100 の範囲で指定することができます。 省略時はすべてのノードの稼働率を取得します。
property	string	省略可	ソート対象のプロパティ。 以下のいずれかを指定することができます。

プロパティ名	JSON 型	省略可否	説明
			<ul style="list-style-type: none"> <li>• <b>name</b> ノード名。</li> <li>• <b>statisticsValue</b> 稼働率の値。</li> </ul>
direction	string	省略可	ソート順を示すパラメーター。 <b>ASC</b> 指定したプロパティの昇順に並び変えます。 <b>DESC</b> 指定したプロパティの降順に並び変えます。 <b>property</b> を指定している場合は、省略することはできません。

## ヒント

- 本 Web API を利用する際、レスポンス時のデータ量を小にするため、**nodeIds**、または、**topN** を指定することを推奨します。
- **property**、および、**direction** の指定を省略した場合は、以下を指定した場合と同様に処理します。

```
"property" : "statisticsValue",
"direction" : "ASC"
```

## レスポンス形式

- ステータスコード

code	意味	説明
200	成功	データの取得に成功したことを示します。 メッセージボディで指定した条件に合致するデータが存在しなかった場合も 200 を返却します。
400	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>• メッセージボディの形式が不正、または、指定がない。</li> <li>• メッセージボディで指定した <b>startDate</b>、または、<b>endDate</b> が ISO8601 拡張形式ではない。</li> <li>• メッセージボディで指定した <b>startDate</b>、および、<b>endDate</b> での期間が不正。</li> <li>• メッセージボディで指定した <b>topN</b> の指定値が不正。</li> <li>• メッセージボディで指定した <b>property</b> の指定値が不正。</li> <li>• メッセージボディで指定した <b>direction</b> の指定値が不正。</li> </ul>

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

```
WidgetDataObject
```

各プロパティの詳細は、「[3.2.1 ウィジェットデータリソース \(WidgetDataObject\) \(25 ページ\)](#)」を参照してください。

## 4.2.2 特定ノードの稼働率の取得

指定したノードにおける稼働率を取得します。

### 必要な権限

オブザーバー以上

### リクエスト形式

POST /api/v1/nodes/{node\_id}/availability

- API パラメーター

パラメーター名	説明
node_id	取得対象のノードの ID。 「 <a href="#">3.1.2 ノードリソース (NodeObject) (20 ページ)</a> 」の id を指定します。

- 利用可能なクエリーパラメーター

パラメーター名	省略可否	説明
startDate	省略不可	稼働率の計算対象とする期間の開始時刻。 ISO8601 の拡張形式で指定します。
endDate	省略不可	稼働率の計算対象とする期間の終了時刻。 ISO8601 の拡張形式で指定します。

- ボディ  
なし。

### レスポンス形式

- ステータスコード

code	意味	説明
200	成功	データの取得に成功したことを示します。 メッセージボディで指定した条件に合致するデータが存在しなかった場合も 200 を返却します。
400	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>クエリーパラメーターで startDate、または、endDate の指定がない。</li> <li>クエリーパラメーターで指定した startDate、または、endDate が ISO8601 拡張形式ではない。</li> <li>クエリーパラメーターで指定した startDate、および、endDate での期間が不正。</li> </ul>

code	意味	説明
404	失敗	URL 中で指定した <code>{node_id}</code> に該当するノードが存在していないことを示します。

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

```
{
  "data" : {
    "id" : string,
    "name" : string,
    "availability" : number
  }
}
```

プロパティ名	JSON 型	説明
data	object	ノードの稼働率を示すオブジェクト。
id	string	稼働率を取得したノードの ID。 「 <a href="#">3.1.2 ノードリソース (NodeObject) (20 ページ)</a> 」の id と一致します。
name	string	稼働率を取得したノードの表示名。
availability	number	指定期間における稼働率。

## 4.3 性能管理 API

NetvisorPro が収集した SNMP データ(MIB 情報)や、NFA が受信したフローデータを操作するための Web API の仕様について説明します。

### 4.3.1 SNMP データ TopN 取得

指定したリージョングループ内における TopN の SNMP データ(MIB 情報)を取得します。

#### 必要な権限

オブザーバー以上

#### リクエスト形式

```
POST /nvp/api/v1/regions/{region_id}/performance/{data_type}/search
```

- API パラメーター

パラメーター名	説明
region_id	取得対象のリージョングループの ID。

パラメーター名	説明
	「3.1.1 リージョンリソース (RegionObject) (20 ページ)」の <code>regionId</code> を指定します。
<code>data_type</code>	<p>取得対象の統計データに対するデータ種別。 以下のデータ種別を指定することができます。</p> <ul style="list-style-type: none"> <li>• <code>ifInUtil</code> 入力インターフェイス使用率</li> <li>• <code>ifOutUtil</code> 出力インターフェイス使用率</li> <li>• <code>ifInDiscards</code> 入力パケット損失率</li> <li>• <code>ifOutDiscards</code> 出力パケット損失率</li> <li>• <code>ifInErrors</code> 入力パケットエラー率</li> <li>• <code>ifOutErrors</code> 出力パケットエラー率</li> <li>• <code>cpuUtil</code> CPU 使用率</li> <li>• <code>memoryUtil</code> メモリ使用率</li> </ul>

- 利用可能なクエリーパラメーター

なし。

- ボディ

```
{
  "startDate" : iso8601,
  "endDate" : iso8601,
  "top" : number,
  "filterType" : "node" | "interface" | null,
  "filters" : [ string ] | null
}
```

プロパティ名	JSON 型	省略可否	説明
<code>startDate</code>	<code>iso8601</code>	必須	取得対象期間の開始時刻。 ISO8601 の拡張形式で指定します。
<code>endDate</code>	<code>iso8601</code>	必須	取得対象期間の終了時刻。 ISO8601 の拡張形式で指定します。
<code>top</code>	<code>number</code>	必須	取得対象の上位ランキングのデータ件数(TopN)。 1~100 の範囲で指定することができます。
<code>filterType</code>	<code>string</code>	省略可	<p>フィルター対象の種別。 取得対象を絞り込む種別を以下のいずれかで指定します。</p> <p><b>node</b> ノードの ID で取得対象のデータを絞り込みます。</p> <p><b>interface</b></p>

プロパティ名	JSON 型	省略可否	説明
			<p>ネットワークインターフェ이스の ID で取得対象のデータを絞り込みます。</p> <p>URL 中の <code>{data_type}</code> にネットワークインターフェイス関連のデータ種別を指定している場合にのみ指定することができます。</p> <p><code>filters</code> を指定している場合は、省略することはできません。</p>
<code>filters</code>	<code>string[]</code>	省略可	<p>取得対象のノード、または、ネットワークインターフェ이스の ID の配列。</p> <p>省略した場合は、取得対象の絞り込みは行いません。</p> <p>取得対象として、複数の ID を指定することができ、指定した ID に合致するデータのみを取得対象とします。</p> <p>ID は、最大 250 件まで指定することができます。</p> <p><code>filterType</code> の指定値により、指定した ID の種別を判断します。例えば、<code>filterType</code> に <code>node</code> を指定した場合は、指定値をノード ID として処理します。</p>

## レスポンス形式

- ステータスコード

code	意味	説明
200	成功	<p>データの取得に成功したことを示します。</p> <p>メッセージボディで指定した条件に合致するデータが存在しなかった場合も 200 を返却します。</p>
400	失敗	<p>以下のいずれかであることを示します。</p> <ul style="list-style-type: none"> <li>メッセージボディの形式が不正、または、指定がない。</li> <li>メッセージボディで指定した <code>startDate</code>、または、<code>endDate</code> が ISO8601 拡張形式ではない。</li> <li>メッセージボディで指定した <code>startDate</code>、および、<code>endDate</code> での期間が不正。</li> <li>メッセージボディで指定した <code>top</code> の指定値が不正、または、<code>top</code> の指定がない。</li> <li>メッセージボディで <code>filters</code> を指定しているが、<code>filterType</code> を指定していない。</li> <li>メッセージボディで指定した <code>filters</code> の指定値が不正。</li> </ul>
404	失敗	<p>以下のいずれかであることを示します。</p> <ul style="list-style-type: none"> <li>URL 中で指定した <code>{region_id}</code> に該当するリージョングループが存在していない。</li> <li>URL 中の <code>{data_type}</code> 指定内容が不正。</li> </ul>
503	失敗	<p>ファイアウォールやサービス停止などの影響で、IMS コンポーネントと NetvisorPro との通信が正常に行えない状況であることを示します。</p>

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

StatisticsInfoObject

rows 配列は、"id"の昇順となります。data 配列は、"id"の昇順、および、"time"の昇順となります。

各プロパティの詳細は、「[3.2.2 統計情報リソース \(StatisticsInfoObject\) \(27 ページ\)](#)」を参照してください。

## 4.3.2 特定ノードの SNMP データ取得

指定したノードにおける SNMP データ(MIB 情報)を取得します。

### 必要な権限

オブザーバー以上

### リクエスト形式

```
POST /nvp/api/v1/nodes/{node_id}/performance/{data_type}/search
```

- API パラメーター

パラメーター名	説明
node_id	取得対象のノードの ID。 「 <a href="#">3.1.2 ノードリソース (NodeObject) (20 ページ)</a> 」の id を指定します。
data_type	取得対象の統計データに対するデータ種別。 以下のデータ種別を指定することができます。 <ul style="list-style-type: none"><li>ifInUtil 入力インターフェイス使用率</li><li>ifOutUtil 出力インターフェイス使用率</li><li>ifInDiscards 入力パケット損失率</li><li>ifOutDiscards 出力パケット損失率</li><li>ifInErrors 入力パケットエラー率</li><li>ifOutErrors 出力パケットエラー率</li><li>cpuUtil CPU 使用率</li><li>memoryUtil メモリ使用率</li></ul>

- 利用可能なクエリーパラメーター  
なし。
- ボディ

```
{
  "startDate" : iso8601,
  "endDate" : iso8601,
  "top" : number | null
}
```

プロパティ名	JSON 型	省略可否	説明
startDate	iso8601	必須	取得対象期間の開始時刻。 ISO8601 の拡張形式で指定します。
endDate	iso8601	必須	取得対象期間の終了時刻。 ISO8601 の拡張形式で指定します。
top	number	必須	取得対象の上位ランキングのデータ件数(TopN)。 1~2147483647 の範囲で指定することができます。

## レスポンス形式

- ステータスコード

code	意味	説明
200	成功	データの取得に成功したことを示します。 メッセージボディで指定した条件に合致するデータが存在しなかった場合も 200 を返却します。
400	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>メッセージボディの形式が不正、または、指定がない。</li> <li>メッセージボディで指定した startDate、または、endDate が ISO8601 拡張形式ではない。</li> <li>メッセージボディで指定した startDate、および、endDate での期間が不正。</li> <li>メッセージボディで指定した top の指定値が不正。</li> </ul>
404	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>URL 中で指定した {node_id} に該当するノードが存在していない。</li> <li>URL 中の {data_type} 指定内容が不正。</li> </ul>
503	失敗	ファイアウォールやサービス停止などの影響で、IMS コンポーネントと NetvisorPro との通信が正常に行えない状況であることを示します。

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

StatisticsInfoObject

rows 配列は、"id" の昇順となります。data 配列は、"id" の昇順、および、"time" の昇順となります。

各プロパティの詳細は、「[3.2.2 統計情報リソース \(StatisticsInfoObject\) \(27 ページ\)](#)」を参照してください。



### 4.3.3 特定ネットワークインターフェイスの SNMP データ取得

指定した単一のネットワークインターフェイスに対する SNMP データ(MIB 情報)を取得します。

#### 必要な権限

オブザーバー以上

#### リクエスト形式

```
POST /nvp/api/v1/interfaces/{if_id}/performance/{data_type}/search
```

- API パラメーター

パラメーター名	説明
if_id	取得対象のネットワークインターフェイスの ID。 「 <a href="#">3.1.3 ネットワークインターフェイスリソース (NetworkInterfaceObject) (24 ページ)</a> 」の id を指定します。
data_type	取得対象の統計データに対するデータ種別。 以下のデータ種別を指定することができます。 <ul style="list-style-type: none"> <li>ifInUtil 入力インターフェイス使用率</li> <li>ifOutUtil 出力インターフェイス使用率</li> <li>ifInDiscards 入力パケット損失率</li> <li>ifOutDiscards 出力パケット損失率</li> <li>ifInErrors 入力パケットエラー率</li> <li>ifOutErrors 出力パケットエラー率</li> </ul>

- 利用可能なクエリーパラメーター

なし。

- ボディ

```
{
  "startDate" : iso8601,
  "endDate" : iso8601
}
```

プロパティ名	JSON 型	省略可否	説明
startDate	iso8601	必須	取得対象期間の開始時刻。 ISO8601 の拡張形式で指定します。
endDate	iso8601	必須	取得対象期間の終了時刻。

プロパティ名	JSON 型	省略可否	説明
			ISO8601 の拡張形式で指定します。

## レスポンス形式

- ステータスコード

code	意味	説明
200	成功	データの取得に成功したことを示します。
400	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>メッセージボディの形式が不正、または、指定がない。</li> <li>メッセージボディで指定した <code>startDate</code>、または、<code>endDate</code> が ISO8601 拡張形式ではない。</li> <li>メッセージボディで指定した <code>startDate</code>、および、<code>endDate</code> での期間が不正。</li> </ul>
404	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>URL 中で指定した <code>{if_id}</code> に該当するネットワークインターフェイスが存在していない。</li> <li>URL 中の <code>{data_type}</code> 指定内容が不正。</li> </ul>

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

```
StatisticsInfoObject
```

`rows` 配列は、`"id"` の昇順となります。`data` 配列は、`"id"` の昇順、および、`"time"` の昇順となります。

各プロパティの詳細は、「[3.2.2 統計情報リソース \(StatisticsInfoObject\) \(27 ページ\)](#)」を参照してください。

### 4.3.4 フローデータ TopN 取得

指定したリージョングループ内における TopN のフローデータを取得します。

## 必要な権限

オブザーバー以上

## リクエスト形式

```
POST /nfa/api/v1/regions/{region_id}/spdata/{data_type}/search
```

- API パラメーター

パラメーター名	説明
region_id	取得対象のリージョングループの ID。 「 <a href="#">3.1.1 リージョンリソース (RegionObject) (20 ページ)</a> 」の regionId を指定します。
data_type	取得対象の統計データに対するデータ種別。 以下のデータ種別を指定することができます。 <ul style="list-style-type: none"> <li>• in-interface 通信量の多い入力インターフェイス。</li> <li>• out-interface 通信量の多い出力インターフェイス。</li> <li>• application 通信量の多いアプリケーション。</li> <li>• ip-protocol 通信量の多い IP プロトコル。</li> <li>• dscp 通信量の多い DSCP 値。</li> <li>• conversation 通信量の多いカンバセーション(2 点間)。</li> <li>• src-ipaddress 通信量の多い送信元 IP アドレス。</li> <li>• dst-ipaddress 通信量の多い宛先 IP アドレス。</li> <li>• src-endpointgrp 通信量の多い送信元エンドポイントグループ。</li> <li>• dst-endpointgrp 通信量の多い宛先エンドポイントグループ。</li> <li>• src-as 通信量の多い送信元 AS(Autonomous System)。</li> <li>• dst-as 通信量の多い宛先 AS(Autonomous System)。</li> </ul>

- 利用可能なクエリーパラメーター  
なし。
- ボディ

```
{
  "startDate" : iso8601,
  "endDate" : iso8601,
  "topN" : number,
  "valueUnit" : "bytes" | "bps" | "packets" | "pps" | null,
  "filterType" : "node" | "interface" | null,
  "filters" : [ string ] | null
}
```

プロパティ名	JSON 型	省略可否	説明
startDate	iso8601	必須	取得対象期間の開始時刻。 ISO8601 の拡張形式で指定します。

プロパティ名	JSON 型	省略可否	説明
endDate	iso8601	必須	取得対象期間の終了時刻。 ISO8601 の拡張形式で指定します。
topN	number	必須	取得対象の上位ランキングのデータ件数(TopN)。 1~100 の範囲で指定することができます。
valueUnit	string	省略可	取得対象のデータの単位。 bytes と bps は同じ意味として処理します。 packets と pps も同じ意味として処理します。 省略時は、bytes (bps) を指定した場合と同様に処理します。
filterType	string	省略可	フィルター対象の種別。 取得対象を絞り込む種別を以下のいずれかで指定します。 <b>node</b> ノードの ID で取得対象のデータを絞り込みます。 <b>interface</b> ネットワークインターフェイスの ID で取得対象のデータを絞り込みます。 URL 中の {data_type} に exporter を指定している場合は指定することができません。 filters を指定している場合は、省略することはできません。
filters	string[]	省略可	取得対象のノード、または、ネットワークインターフェイスの ID の配列。 省略した場合は、取得対象の絞り込みは行いません。 取得対象として、複数の ID を指定することができ、指定した ID に合致するデータのみを取得対象とします。 ID は、最大 250 件まで指定することができます。 filterType の指定値により、指定した ID の種別を判断します。例えば、filterType に node を指定した場合は、指定値をノード ID として処理します。

## レスポンス形式

- ステータスコード

code	意味	説明
200	成功	データの取得に成功したことを示します。 メッセージボディで指定した条件に合致するデータが存在しなかった場合も 200 を返却します。
400	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>メッセージボディの形式が不正、または、指定がない。</li> <li>メッセージボディで指定した startDate、または、endDate が ISO8601 拡張形式ではない。</li> <li>メッセージボディで指定した startDate、および、endDate での期間が不正。</li> <li>メッセージボディで指定した topN の指定値が不正、または、topN の指定がない。</li> </ul>

code	意味	説明
		<ul style="list-style-type: none"> <li>メッセージボディで指定した <code>valueUnit</code> の指定値が不正。</li> <li>メッセージボディで <code>filters</code> を指定しているが、<code>filterType</code> を指定していない。</li> <li>メッセージボディで指定した <code>filters</code> の指定値が不正。</li> </ul>
404	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>URL 中で指定した <code>{region_id}</code> に該当するリージョングループが存在していない。</li> <li>URL 中の <code>{data_type}</code> 指定内容が不正。</li> </ul>
503	失敗	ファイアウォールやサービス停止などの影響で、IMS コンポーネントと NFA との通信が正常に行えない状況であることを示します。

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

WidgetDataObject

`rows` 配列は、"id"の昇順となります。`data` 配列は、"id"の昇順、および、"time"の昇順となります。

各プロパティの詳細は、「[3.2.1 ウィジェットデータリソース \(WidgetDataObject\) \(25 ページ\)](#)」を参照してください。

### 4.3.5 特定ノードのフローデータ TopN 取得

指定したノード(エクスポーター)における TopN のフローデータを取得します。

#### 必要な権限

オブザーバー以上

#### リクエスト形式

POST /nfa/api/v1/nodes/{node\_id}/flowdata/{data\_type}/search

- API パラメーター

パラメーター名	説明
node_id	取得対象のノードの ID。 <a href="#">「3.1.2 ノードリソース (NodeObject) (20 ページ)」</a> の id を指定します。
data_type	取得対象の統計データに対するデータ種別。 以下のデータ種別を指定することができます。 <ul style="list-style-type: none"> <li>in-interface 通信量の多い入力インターフェイス。</li> <li>out-interface</li> </ul>

パラメーター名	説明
	<p>通信量の多い出力インターフェイス。</p> <ul style="list-style-type: none"> <li>• application 通信量の多いアプリケーション。</li> <li>• ip-protocol 通信量の多い IP プロトコル。</li> <li>• dscp 通信量の多い DSCP 値。</li> <li>• conversation 通信量の多いカンバセーション(2 点間)。</li> <li>• src-ipaddress 通信量の多い送信元 IP アドレス。</li> <li>• dst-ipaddress 通信量の多い宛先 IP アドレス。</li> <li>• src-endpointgrp 通信量の多い送信元エンドポイントグループ。</li> <li>• dst-endpointgrp 通信量の多い宛先エンドポイントグループ。</li> <li>• src-as 通信量の多い送信元 AS(Autonomous System)。</li> <li>• dst-as 通信量の多い宛先 AS(Autonomous System)。</li> </ul>

- 利用可能なクエリーパラメーター  
なし。
- ボディ

```
{
  "startDate" : iso8601,
  "endDate" : iso8601,
  "topN" : number,
  "valueUnit" : "bytes" | "bps" | "packets" | "pps" | null
}
```

プロパティ名	JSON 型	省略可否	説明
startDate	iso8601	必須	取得対象期間の開始時刻。 ISO8601 の拡張形式で指定します。
endDate	iso8601	必須	取得対象期間の終了時刻。 ISO8601 の拡張形式で指定します。
topN	number	必須	取得対象の上位ランキングのデータ件数(TopN)。 1~100 の範囲で指定することができます。
valueUnit	string	省略可	取得対象のデータの単位。 bytes と bps は同じ意味として処理します。 packets と pps も同じ意味として処理します。 省略時は、bytes (bps) を指定した場合と同様に処理します。

## レスポンス形式

- ステータスコード

code	意味	説明
200	成功	データの取得に成功したことを示します。 メッセージボディで指定した条件に合致するデータが存在しなかった場合も 200 を返却します。
400	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>メッセージボディの形式が不正、または、指定がない。</li> <li>メッセージボディで指定した <code>startDate</code>、または、<code>endDate</code> が ISO8601 拡張形式ではない。</li> <li>メッセージボディで指定した <code>startDate</code>、および、<code>endDate</code> での期間が不正。</li> <li>メッセージボディで指定した <code>topN</code> の指定値が不正、または、<code>topN</code> の指定がない。</li> <li>メッセージボディで指定した <code>valueUnit</code> の指定値が不正。</li> </ul>
404	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>URL 中で指定した <code>{node_id}</code> に該当するノードが存在していない。</li> <li>URL 中の <code>{data_type}</code> 指定内容が不正。</li> </ul>
503	失敗	ファイアウォールやサービス停止などの影響で、IMS コンポーネントと NFA との通信が正常に行えない状況であることを示します。

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

`WidgetDataObject`

`rows` 配列は、"id"の昇順となります。`data` 配列は、"id"の昇順、および、"time"の昇順となります。

各プロパティの詳細は、「[3.2.1 ウィジェットデータリソース \(WidgetDataObject\) \(25 ページ\)](#)」を参照してください。

### 4.3.6 特定ネットワークインターフェイスのフローデータ TopN 取得

指定したネットワークインターフェイスにおける TopN のフローデータを取得します。

#### 必要な権限

オブザーバー以上

## リクエスト形式

```
POST /nfa/api/v1/interfaces/{if_id}/flowdata/{data_type}/search
```

- API パラメーター

パラメーター名	説明
if_id	取得対象のネットワークインターフェイスの ID。 「 <a href="#">3.1.3 ネットワークインターフェイスリソース (NetworkInterfaceObject) (24 ページ)</a> 」の id を指定します。
data_type	取得対象の統計データに対するデータ種別。 以下のデータ種別を指定することができます。 <ul style="list-style-type: none"> <li><b>in-interface</b> 通信量の多い入力インターフェイス。</li> <li><b>out-interface</b> 通信量の多い出力インターフェイス。</li> <li><b>application</b> 通信量の多いアプリケーション。</li> <li><b>ip-protocol</b> 通信量の多い IP プロトコル。</li> <li><b>dscp</b> 通信量の多い DSCP 値。</li> <li><b>conversation</b> 通信量の多いカンバセーション(2 点間)。</li> <li><b>src-ipaddress</b> 通信量の多い送信元 IP アドレス。</li> <li><b>dst-ipaddress</b> 通信量の多い宛先 IP アドレス。</li> <li><b>src-endpointgrp</b> 通信量の多い送信元エンドポイントグループ。</li> <li><b>dst-endpointgrp</b> 通信量の多い宛先エンドポイントグループ。</li> <li><b>src-as</b> 通信量の多い送信元 AS(Autonomous System)。</li> <li><b>dst-as</b> 通信量の多い宛先 AS(Autonomous System)。</li> </ul>

- 利用可能なクエリーパラメーター  
なし。
- ボディ

```
{
  "startDate" : iso8601,
  "endDate" : iso8601,
  "topN" : number,
  "valueUnit" : "bytes" | "bps" | "packets" | "pps" | null
}
```



プロパティ名	JSON 型	省略可否	説明
startDate	iso8601	必須	取得対象期間の開始時刻。 ISO8601 の拡張形式で指定します。
endDate	iso8601	必須	取得対象期間の終了時刻。 ISO8601 の拡張形式で指定します。
topN	number	必須	取得対象の上位ランキングのデータ件数(TopN)。 1~100 の範囲で指定することができます。
valueUnit	string	省略可	取得対象のデータの単位。 bytes と bps は同じ意味として処理します。 packets と pps も同じ意味として処理します。 省略時は、bytes (bps) を指定した場合と同様に処理します。

## レスポンス形式

- ステータスコード

code	意味	説明
200	成功	データの取得に成功したことを示します。 メッセージボディで指定した条件に合致するデータが存在しなかった場合も 200 を返却します。
400	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>メッセージボディの形式が不正、または、指定がない。</li> <li>メッセージボディで指定した startDate、または、endDate が ISO8601 拡張形式ではない。</li> <li>メッセージボディで指定した startDate、および、endDate での期間が不正。</li> <li>メッセージボディで指定した topN の指定値が不正、または、topN の指定がない。</li> <li>メッセージボディで指定した valueUnit の指定値が不正。</li> </ul>
404	失敗	以下のいずれかであることを示します。 <ul style="list-style-type: none"> <li>URL 中で指定した {if_id} に該当するネットワークインターフェイスが存在していない。</li> <li>URL 中の {data_type} 指定内容が不正。</li> </ul>
503	失敗	ファイアウォールやサービス停止などの影響で、IMS コンポーネントと NFA との通信が正常に行えない状況であることを示します。

その他の発生する可能性のあるステータスコードについては、「[2.3.2 共通の HTTP ステータスコード \(15 ページ\)](#)」を参照してください。

- ボディ

WidgetDataObject

rows 配列は、"id" の昇順となります。data 配列は、"id" の昇順、および、"time" の昇順となります。

各プロパティの詳細は、「[3.2.1 ウィジェットデータリソース \(WidgetDataObject\) \(25 ページ\)](#)」を参照してください。

# 第5章

## チュートリアル

本章では、Web API の基本的な使い方を具体的な運用例をもとに説明します。

---

### 目次

5.1 構成情報の操作.....	62
5.2 障害情報の操作.....	67
5.3 性能情報の操作.....	72

---

## 5.1 構成情報の操作

各 API を利用して、構成情報を操作する方法について説明します。

### 5.1.1 特定シリーズのソフトウェアバージョンを確認する

特定シリーズのノードにおいてソフトウェアバージョンを確認する手順について説明します。

本操作は、同一シリーズの機種において、ソフトウェアバージョンに差がないかを確認する場合に行います。

ここでは、以下の操作例を用いて手順を説明します。

#### 操作例：

管理対象となっている 10 ノードの QX-S6600 シリーズに対し、ソフトウェアバージョンに差がないかを確認する。

#### ヒント

リクエスト、および、レスポンスの内容は、可読性向上のため、改行とインデントを加えて記載しています。

1. 条件に合致するノード一覧を取得します。

ノード一覧取得の API を以下のクエリーパラメーターを付加して実行します。

- シリーズ名による絞り込み:

```
properties@series==QX-S6600 series
```

- 取得対象のノード情報:

ノード名、ソフトウェアバージョンに加え、IP アドレス、シリーズ名を取得します。

```
props=id,properties.name,properties.ipv4Address,
properties.series,properties.swVersion
```

API の詳細は、「[4.1.2 ノード一覧取得 \(34 ページ\)](#)」を参照してください。

具体的な Web API のリクエストとレスポンスの内容は以下の通りです。

- リクエスト:

```
GET /api/v1/nodes?properties%40series=%3DQX-S6600+series&props=id%
2Cproperties.name%2Cproperties.ipv4Address%2Cproperties.series%2Cp
roperties.swVersion HTTP/1.1
Date: Thu, 01 Oct 2020 01:12:11 GMT
Host: ims.nec.com
Authorization: SharedKeyLite Mkh0Av4xJI0p+egYhDYLq6b0hTB1MDQLwqn8
atgWPI=:ej3yLD+ta2sdTWoLJe9+0N7sQks11QsMQpi42OK6DlM=
```

- レスポンス:

```

HTTP/1.1 200 OK
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Transfer-Encoding: chunked
Expires: 0
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Thu, 01 Oct 2020 01:12:11 GMT
X-Frame-Options: DENY
Content-Type: application/json; charset=UTF-8

{
  "totalCount": 10,
  "data": [
    {
      "id": "2d68c165-3043-4101-8ff6-981eb0d16d0e",
      "properties": {
        "ipv4Address": "172.28.1.180",
        "series": "QX-S6600 series",
        "swVersion": "7.1.3",
        "name": "HO_Center_SW001"
      }
    },
    {
      "id": "025e26d5-7657-4c16-8aa1-c832603f03d2",
      "properties": {
        "ipv4Address": "172.28.1.181",
        "series": "QX-S6600 series",
        "swVersion": "7.1.3",
        "name": "HO_Center_SW002"
      }
    },
    {
      "id": "025e26d5-7657-4c16-8aa1-c832603f03d2",
      "properties": {
        "ipv4Address": "172.28.2.180",
        "series": "QX-S6600 series",
        "swVersion": "7.2.8",
        "name": "B1_Center_SW003"
      }
    },
    :
    (略)
    :
    {
      "id": "1cac0684-173f-4665-8869-9a872495983f",
      "properties": {
        "ipv4Address": "172.28.5.181",
        "series": "QX-S6600 series",
        "swVersion": "7.2.8",
        "name": "B4_Center_SW004"
      }
    }
  ]
}

```

2. 取得したソフトウェアバージョンを確認します。

各ノードに対する `properties.swVersion` の値に差がないかを確認します。

バージョンが低いノードが見つかった場合は、予防保守の観点で、ソフトウェアのバージョンアップ計画を検討します。

## 5.1.2 物理インターフェイスの詳細情報を確認する

指定ノードの物理インターフェイスの情報を確認する手順について説明します。

本操作は、指定ノードに実装されている物理インターフェイスのインデックス (`ifIndex`) や回線速度などを確認する場合に行います。

ここでは、以下の操作例を用いて手順を説明します。

### 操作例：

管理対象となっているノード「`sys_sw01`」に対し、物理インターフェイスの詳細を確認する。

### ヒント

リクエスト、および、レスポンスの内容は、可読性向上のため、改行とインデントを加えて記載しています。

1. ノード「`sys_sw01`」の ID を調べます。

ノード一覧取得の API を以下のクエリーパラメーターを付加して実行します。

- ノード名による絞り込み:

```
properties@name==sys_sw01
```

- 取得対象のノード情報:

確実にノードを特定するために、ID に加えて、ノード名、リージョングループ名を取得します。

```
props=id,properties.name,properties.regionName
```

API の詳細は、「[4.1.2 ノード一覧取得 \(34 ページ\)](#)」を参照してください。

具体的な Web API のリクエストとレスポンスの内容は以下の通りです。

- リクエスト:

```
GET /api/v1/nodes?properties%40name=%3Dsys_sw01&props=id%2Cproperties.name%2Cproperties.regionName HTTP/1.1
Date: Thu, 01 Oct 2020 01:18:28 GMT
Host: ims.nec.com
Authorization: SharedKeyLite MkhnoAv4xJI0p+egYhDYLq6bOhTB1MDQLwqn8atgWPI=:KY8iNmKctVreozEHFUfox+SHVq8PJpdc/4/1HkIQM=
```

- レスポンス:

```
HTTP/1.1 200 OK
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Transfer-Encoding: chunked
Expires: 0
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Thu, 01 Oct 2020 01:18:28 GMT
X-Frame-Options: DENY
Content-Type: application/json; charset=UTF-8

{
  "totalCount": 1,
  "data": [
    {
      "id": "a1e70177-877e-4ad3-8403-317f4a3ee291",
      "properties": {
        "name": "sys_sw01",
        "regionName": "Tokyo"
      }
    }
  ]
}
```

取得したノード情報のノード名、リージョングループ名から意図したノードであることを確認し、ノードの ID を取得します。

- 対象ノードのネットワークインターフェイスの情報を取得します。

ネットワークインターフェイス一覧取得の API を以下のクエリーパラメーターを付加して実行します。

- ネットワークインターフェイスのタイプによる絞り込み:

```
properties@type==ethernetCsmacd(6)
```

API の詳細は、「[4.1.4 ネットワークインターフェイス一覧取得 \(38 ページ\)](#)」を参照してください。

具体的な Web API のリクエストとレスポンスの内容は以下の通りです。

- リクエスト:

```
GET /api/v1/nodes/a1e70177-877e-4ad3-8403-317f4a3ee291/interfaces?
properties%40type=%3DethernetCsmacd%286%29 HTTP/1.1
Date: Thu, 01 Oct 2020 01:23:15 GMT
Host: ims.nec.com
Authorization: SharedKeyLite MkhnoAv4xJI0p+egYhDYLq6bOhTB1MDQLwqn8
atgWPI=:MyqUQ/IGXX7IYiwhE0X5aosfxvK7/RMg/7kXxx2QE80=
```

- レスポンス:

```
HTTP/1.1 200 OK
X-XSS-Protection: 1; mode=block
```

```

X-Content-Type-Options: nosniff
Transfer-Encoding: chunked
Expires: 0
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Thu, 01 Oct 2020 01:23:15 GMT
X-Frame-Options: DENY
Content-Type: application/json;charset=UTF-8

{
  "totalCount": 16,
  "data": [
    {
      "nodeId": "ale70177-877e-4ad3-8403-317f4a3ee291",
      "properties": {
        "macAddress": "00:e0:fc:69:06:10",
        "name": "Ethernet0/1",
        "speed": 1000000000,
        "description": "Ethernet0/1",
        "regionId": "default",
        "ifIndex": 2050,
        "ifName": "Ethernet0/1",
        "type": "ethernetCsmacd(6)",
        "regionName": "Tokyo"
      },
      "type": "NETWORK_INTERFACE",
      "id": "05f26f88-d638-43ae-a221-a60fa890c2f2",
      "revision": 1
    },
    :
    (略)
    :
    {
      "nodeId": "ale70177-877e-4ad3-8403-317f4a3ee291",
      "properties": {
        "macAddress": "00:e0:fc:69:06:10",
        "name": "Ethernet0/16",
        "speed": 1000000000,
        "description": "Ethernet0/16",
        "regionId": "default",
        "ifIndex": 3458,
        "ifName": "Ethernet0/16",
        "type": "ethernetCsmacd(6)",
        "regionName": "Tokyo"
      },
      "type": "NETWORK_INTERFACE",
      "id": "3d9105b5-ec25-4806-a53c-e031e7259854",
      "revision": 1
    }
  ]
}

```

## ヒント

- totalCount の値が 20 以上だった場合は、クエリーパラメーターの page を付加して、続きのネットワークインターフェイスの情報を取得します。



- `page` を付加した情報取得において、続きのネットワークインターフェースの情報が存在しなかった場合でも Web API は 200(成功)を返却します。

## 5.2 障害情報の操作

各 API を利用して、障害情報を操作する方法について説明します。

### 5.2.1 障害が発生しているノードを確認する

障害が発生しているノードの情報を取得する手順について説明します。

本操作は、障害が発生しているノードの状態を確認したい場合に行います。

ここでは、以下の操作例を用いて手順を説明します。

#### 操作例：

状態(重要度)が **Warning** 以上のノードを障害が発生しているノードとして検索を行い、ノード情報としてノード名、IP アドレス、シリーズ名と状態(重要度)を取得する。

#### ヒント

リクエスト、および、レスポンスの内容は、可読性向上のため、改行とインデントを加えて記載しています。

1. 条件に合致するノード一覧を取得します。

ノード一覧取得の API を以下のクエリーパラメーターを付加して実行します。

- 状態(重要度)による絞り込み:

```
severity==FATAL&severity==CRITICAL&severity==ERROR
&severity==WARNING
```

- 取得対象のノード情報(ノード名、IP アドレス、シリーズ名、状態):

```
props=id,properties.name,properties.ipv4Address,
properties.series,severity
```

API の詳細は、「[4.1.2 ノード一覧取得 \(34 ページ\)](#)」を参照してください。

具体的な Web API のリクエストとレスポンスの内容は以下の通りです。

- リクエスト:

```
GET /api/v1/nodes?severity=%3DFATAL&severity=%3DCRITICAL&severity=
%3DERROR&severity=%3DWARNING&props=id%2Cproperties.name%2Cproperti
es.ipv4Address%2Cproperties.series%2Cseverity HTTP/1.1
Date: Thu, 01 Oct 2020 01:33:10 GMT
Host: ims.nec.com
```

```
Authorization: SharedKeyLite MkhnoAv4xJI0p+egYhDYLq6bOhTB1MDQLwqn8
atgWPI=:bDnGI4FS8062gMHvM4udhPbLKceQS9Tkkh5LGKSRJWo=
```

- レスポンス:

```
HTTP/1.1 200 OK
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Transfer-Encoding: chunked
Expires: 0
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Thu, 01 Oct 2020 01:33:10 GMT
X-Frame-Options: DENY
Content-Type: application/json; charset=UTF-8

{
  "totalCount": 3,
  "data": [
    {
      "properties": {
        "ipv4Address": "172.28.1.200",
        "series": "Windows series",
        "name": "H4_Server005"
      },
      "severity": "FATAL",
      "id": "48fd3a29-3ca1-40eb-9efb-ae415b97a4b9"
    },
    {
      "properties": {
        "ipv4Address": "172.28.5.10",
        "series": "QX-S3400F series",
        "name": "B4_switch005"
      },
      "severity": "WARNING",
      "id": "505b6d90-f3df-434e-945b-0f27b6957ef1"
    },
    {
      "properties": {
        "ipv4Address": "172.28.2.7",
        "series": "IX3000 series",
        "name": "B1_router003"
      },
      "severity": "WARNING",
      "id": "6be706f9-b8be-4b6d-9cd1-be360b0c771d"
    }
  ]
}
```

2. 取得内容 (レスポンスボディ) を確認します。

## ヒント

- Warning 以上のノードが存在しなかった場合でも Web API は 200(成功)を返却します。
- totalCount の値が 100 以上だった場合は、クエリーパラメーターの page を付加して、続きのノードの情報を取得します。

## 5.2.2 複数ノードの稼働状況を確認する

複数ノードの稼働状況を確認する手順について説明します。

本操作は、役割や機種が同じノードに対し、稼働状況を確認したい場合に行います。

ここでは、以下の操作例を用いて手順を説明します。

### 操作例：

本社1階に配置した15台のネットワークカメラ「h1\_camera01~15」に対し、2020年9月20日~26日の7日間における稼働率の状況を確認する。

### ヒント

リクエスト、および、レスポンスの内容は、可読性向上のため、改行とインデントを加えて記載しています。

1. 名前が「h1\_camera」から始まるノードのIDを調べます。

ノード一覧取得のAPIを以下のクエリーパラメーターを付加して実行します。

- ノード名による絞り込み:

```
properties@name=^h1_camera
```

- 取得対象のノード情報:

確実にノードを特定するために、IDに加えて、ノード名、リージョングループ名を取得します。

```
props=id,properties.name,properties.regionName
```

APIの詳細は、「[4.1.2 ノード一覧取得 \(34 ページ\)](#)」を参照してください。

具体的なWeb APIのリクエストとレスポンスの内容は以下の通りです。

- リクエスト:

```
GET /api/v1/nodes?properties%40name=%5Eh1_camera&props=id%2Cproperties.name%2Cproperties.regionName HTTP/1.1
Date: Thu, 01 Oct 2020 01:46:38 GMT
Host: ims.nec.com
Authorization: SharedKeyLite MkhnoAv4xJI0p+egYhDYLq6bOhTB1MDQLwqn8atgWPI=:mCtlu3j/nbpIkXD8k5nVM9XT43xjIE/aBgoVuPkD8Vo=
```

- レスポンス:

```
HTTP/1.1 200 OK
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Transfer-Encoding: chunked
Expires: 0
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Thu, 01 Oct 2020 01:46:38 GMT
```

```

X-Frame-Options: DENY
Content-Type: application/json;charset=UTF-8

{
  "totalCount": 15,
  "data": [
    {
      "id": "d70afb4f-ea6d-401b-95e4-4fff8bd59e3a",
      "properties": {
        "name": "h1_camera12",
        "regionName": "Tokyo"
      }
    },
    {
      "id": "e631ea51-3a28-460f-9555-27a8a76d777c",
      "properties": {
        "name": "h1_camera08",
        "regionName": "Tokyo"
      }
    },
    :
    (略)
    :
    {
      "id": "fafa3e0f-1db1-4d18-a013-97ebc3d4a333",
      "properties": {
        "name": "h1_camera13",
        "regionName": "Tokyo"
      }
    }
  ]
}

```

取得したノード情報のノード名、リージョングループ名から意図したノードであることを確認し、ノードの ID を取得します。

2. 対象ノード内での稼働率の低い Top5 の情報を取得します。

ノード稼働率 TopN の取得 API を適切にリクエストボディを指定して実行します。

API の詳細は、「[4.2.1 ノード稼働率 TopN の取得 \(43 ページ\)](#)」を参照してください。

具体的な Web API のリクエストとレスポンスの内容は以下の通りです。

- リクエスト:

```

POST /api/v1/nodes/availabilities/search HTTP/1.1
Content-Length: 758
Date: Thu, 01 Oct 2020 01:55:32 GMT
Host: ims.nec.com
Content-Type: application/json; charset=utf-8
Authorization: SharedKeyLite MkhnoAv4xJI0p+egYhDYLq6bOhTB1MDQLwqn8
atgWPI=:KArZAT/XYUdE75hWJCZcqH15tzKOisUdODhASz7kbs=

{
  "nodeIds": [
    "d70afb4f-ea6d-401b-95e4-4fff8bd59e3a",
    "e631ea51-3a28-460f-9555-27a8a76d777c",

```

```

      :
      (略)
      :
      "fafe3e0f-1db1-4d18-a013-97ebc3d4a333"
    ],
    "startDate": "2020-09-20T00:00:00+09:00",
    "endDate": "2020-09-26T23:59:59+09:00",
    "topN": 5,
    "property": "statisticsValue",
    "direction": "ASC"
  }

```

- レスポンス:

```

HTTP/1.1 200 OK
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Transfer-Encoding: chunked
Expires: 0
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Thu, 01 Oct 2020 01:55:32 GMT
X-Frame-Options: DENY
Content-Type: application/json; charset=UTF-8

{
  "dnsResolving": null,
  "statisticsWidgetData": {
    "NONE": {
      "rows": [
        {
          "id":
            "0fad4250-f29f-4ab9-9af9-1d2c70c579d4",
          "name": "h1_camera10",
          "statisticsValue": 0.6785709261894226
        },
        {
          "id":
            "110fa066-d6f2-409d-972d-8f6290ceefda",
          "name": "h1_camera15",
          "statisticsValue": 0.6999184531385121
        },
        {
          "id":
            "5cd14942-38c7-4927-8f9d-03ec1d456f71",
          "name": "h1_camera14",
          "statisticsValue": 0.8576132991273826
        },
        {
          "id":
            "605b1844-0cfc-4281-8758-2ac5177c8eec",
          "name": "h1_camera03",
          "statisticsValue": 0.9204882796728816
        },
        {
          "id":
            "6673d979-a6ca-4ec0-80f4-02e9ce6bf7e5",
          "name": "h1_camera02",

```

```

        "statisticsValue": 1
    },
    ],
    "data": [
        {
            "id":
                "0fad4250-f29f-4ab9-9af9-1d2c70c579d4",
            "value": 0.6785709261894226,
            "time": null
        },
        {
            "id":
                "110fa066-d6f2-409d-972d-8f6290ceefda",
            "value": 0.6999184531385121,
            "time": null
        },
        {
            "id":
                "5cd14942-38c7-4927-8f9d-03ec1d456f71",
            "value": 0.8576132991273826,
            "time": null
        },
        {
            "id":
                "605b1844-0cfc-4281-8758-2ac5177c8eec",
            "value": 0.9204882796728816,
            "time": null
        },
        {
            "id":
                "6673d979-a6ca-4ec0-80f4-02e9ce6bf7e5",
            "value": 1,
            "time": null
        }
    ],
    "otherData": null
}

```

3. 取得した稼働率の状況を確認します。

Top5 の中で稼働率が 100%ではないノードを確認します。5 ノードともに 100%ではない場合は、TopN の範囲広げて再度、稼働率を取得します。5 ノードともに稼働率が 100%の場合は、すべてのノードが稼働率 100%だと判断できます。

稼働率の状況から故障し易い機器を判断し、機器交換の計画を検討します。

## 5.3 性能情報の操作

各 API を利用して、性能情報を操作する方法について説明します。

### 5.3.1 ルーターの CPU 負荷の状況を確認する

管理対象となっているルーターの CPU 使用率の状況を確認する手順について説明します。

本操作は、ルーターの CPU 負荷の状況を確認したい場合に行います。

ここでは、以下の操作例を用いて手順を説明します。

#### 操作例：

リージョングループ「東日本」と「西日本」で管理するすべてのルーターの 2020 年 9 月 28 日の CPU 使用率の全体状況を確認する。

#### ヒント

リクエスト、および、レスポンスの内容は、可読性向上のため、改行とインデントを加えて記載しています。

1. 装置種別がルーターとなっているノードの ID を調べます。

ノード一覧取得の API を以下のクエリーパラメーターを付加して実行します。

- 装置種別による絞り込み:

```
properties@nodeType==Router
```

- 取得対象のノード情報:

ノードの ID に加えて、ノード名、リージョングループの ID、名前 を取得します。

```
props=id,properties.name,properties.regionId,properties.regionName
```

- 取得情報をリージョングループ名でソート:

```
sort=regionName,desc
```

API の詳細は、「[4.1.2 ノード一覧取得 \(34 ページ\)](#)」を参照してください。

具体的な Web API のリクエストとレスポンスの内容は以下の通りです。

- リクエスト:

```
GET /api/v1/nodes?sort=regionName%2Cdesc&properties%40nodeType=%3D
Router&props=id%2Cproperties.name%2Cproperties.regionId%2Cproperti
es.regionName HTTP/1.1
Date: Thu, 01 Oct 2020 02:22:28 GMT
Host: ims.nec.com
Authorization: SharedKeyLite MkhnOAv4xJI0p+egYhDYLq6bOhTB1MDQLwqn8
atgWPI=:jx9+VexFbWua05Ga41peFNTibuHAIxFxflsD3934T8qs=
```

- レスポンス:

```
HTTP/1.1 200 OK
X-XSS-Protection: 1; mode=block
```

```

X-Content-Type-Options: nosniff
Transfer-Encoding: chunked
Expires: 0
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Thu, 01 Oct 2020 02:22:28 GMT
X-Frame-Options: DENY
Content-Type: application/json;charset=UTF-8

{
  "totalCount": 40,
  "data": [
    {
      "id": "d70afb4f-ea6d-401b-95e4-4fff8bd59e3a",
      "properties": {
        "regionId": "east",
        "regionName": "東日本"
        "name": "h1_router001"
      }
    },
    {
      "id": "e631ea51-3a28-460f-9555-27a8a76d777c",
      "properties": {
        "regionId": "east",
        "regionName": "東日本"
        "name": "h2_router002"
      }
    },
    :
    (略)
    :
    {
      "id": "0a01aa11-a4e2-476d-8781-f5c6426bc543",
      "properties": {
        "regionId": "west",
        "regionName": "西日本"
        "name": "b10_router010"
      }
    }
  ]
}

```

取得したノード情報を確認し、ノードの ID および、リージョングループの ID を取得します。

2. リージョングループ毎にルーターの CPU 使用率の情報を取得します。

SNMP データ TopN 取得 の API を適切にリクエストボディを指定して実行します。

API の詳細は、「[4.3.1 SNMP データ TopN 取得 \(46 ページ\)](#)」を参照してください。

リージョングループ「東日本」に対する具体的な Web API のリクエストとレスポンスの内容は以下の通りです。

- リクエスト:

```

POST /nvp/api/v1/regions/east/performance/cpuUtil/search HTTP/1.1
Content-Length: 928

```



```

Date: Thu, 01 Oct 2020 04:56:44 GMT
Host: ims.nec.com
Content-Type: application/json; charset=utf-8
Authorization: SharedKeyLite MkhnOAv4xJI0p+egYhDYLq6bOhTB1MDQLwqn8
atgWPI=:6DC1wnA3pWuLhXRCz3jKOsJoTEkj3M2keGdvp5OLi9c=

{
  "startDate": "2020-09-28T00:00:00+09:00",
  "endDate": "2020-09-28T23:59:59+09:00",
  "top": 10,
  "filterType": "node",
  "filters": [
    "d70afb4f-ea6d-401b-95e4-4fff8bd59e3a",
    "e631ea51-3a28-460f-9555-27a8a76d777c",
    :
    (略)
    :
    "1bbf4600-d85f-465f-b14b-4251a4c67a1c"
  ]
}

```

- レスポンス:

```

HTTP/1.1 200 OK
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Transfer-Encoding: chunked
Expires: 0
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Thu, 01 Oct 2020 04:56:44 GMT
X-Frame-Options: DENY
Content-Type: application/json; charset=UTF-8

{
  "dnsResolving": null,
  "statisticsWidgetData": {
    "LINE": {
      "rows": [
        {
          "nodeName": "h1_router001",
          "statisticsValue": 83.3,
          "instanceId": null,
          "nodeId":
            "d70afb4f-ea6d-401b-95e4-4fff8bd59e3a",
          "instanceName": "System.0",
          "id": "0"
        },
        :
        (略)
        :
        {
          "nodeName": "b2_router002",
          "statisticsValue": 42.0,
          "instanceId": null,
          "nodeId":
            "fad6a1ce-0785-4bcb-b164-bfd14f9055c5",
          "instanceName": "Slot.0.1",

```

```

        "id": "9"
    }
],
"data": [
    { "id": "0", "value": 92.4,
      "time": "2020-09-27T15:00:00Z" },
    { "id": "0", "value": 71.5,
      "time": "2020-09-27T16:00:00Z" },
    { "id": "0", "value": 94.9,
      "time": "2020-09-27T17:00:00Z" },
      :
      (略)
      :
    { "id": "0", "value": 74.3,
      "time": "2020-09-28T11:00:00Z" },
    { "id": "0", "value": 75.6,
      "time": "2020-09-28T12:00:00Z" },
    { "id": "0", "value": 72.6,
      "time": "2020-09-28T13:00:00Z" },
    { "id": "0", "value": 84.3,
      "time": "2020-09-28T14:00:00Z" },
      :
      (略)
      :
    { "id": "9", "value": 44.9,
      "time": "2020-09-28T14:00:00Z" }
],
"otherData": null
}
}
}

```

同様に、リージョングループ「西日本」に対してもルーターの CPU 使用率を取得します。

### 3. 取得した CPU 使用率の状況を確認します。

それぞれのリージョングループで取得した Top10 の CPU 使用率から、全体の CPU 使用率の傾向を確認します。例えば、リージョングループ「東日本」で一位の CPU 使用率が、50%程度だった場合は、リージョングループ「東日本」のルーターは CPU 使用率が最大 50%程度であり、全体として CPU 負荷が高くないと判断できます。

また、2つのリージョングループから取得した CPU 使用率の情報を組み合わせることで、2つのリージョングループを合わせた CPU 使用率 Top10 などを調べることも可能です。

## 5.3.2 WAN インターフェイスの通信内容を確認する

WAN に接続しているネットワークインターフェイスの通信内容を確認する手順について説明します。

本操作は、ネットワークインターフェイスの負荷の原因を調べたい場合などに行います。

ここでは、以下の操作例を用いて手順を説明します。

**操作例：**

ルーター「h1\_rt01」の WAN と接続するネットワークインターフェイス「GigaEthernet0」に対し、2020 年 9 月 20 日から 26 日の期間で発生したアプリケーション通信の内容を確認する。

**ヒント**

リクエスト、および、レスポンスの内容は、可読性向上のため、改行とインデントを加えて記載しています。

1. ノード「h1\_rt01」の ID を調べます。

ノード一覧取得の API を以下のクエリーパラメーターを付加して実行します。

- ノード名による絞り込み:

```
properties@name==h1_rt01
```

- 取得対象のノード情報:

確実にノードを特定するために、ID に加えて、ノード名、リージョングループ名を取得します。

```
props=id,properties.name,properties.regionName
```

API の詳細は、「[4.1.2 ノード一覧取得 \(34 ページ\)](#)」を参照してください。

具体的な Web API のリクエストとレスポンスの内容は以下の通りです。

- リクエスト:

```
GET /api/v1/nodes?properties%40name=%3Dh1_rt01&props=id%2Cproperties.name%2Cproperties.regionName HTTP/1.1
Date: Fri, 02 Oct 2020 00:21:24 GMT
Host: ims.nec.com
Authorization: SharedKeyLite MkhnoAv4xJI0p+egYhDYLq6bOhTB1MDQLwqn8atgWPI=:rSyEnCWw9UBFdy0zS9vQvp3FsToOM4G2XP4AjrSDGic=
```

- レスポンス:

```
HTTP/1.1 200 OK
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Transfer-Encoding: chunked
Expires: 0
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Fri, 02 Oct 2020 00:21:24 GMT
X-Frame-Options: DENY
Content-Type: application/json; charset=UTF-8

{
  "totalCount": 1,
  "data": [
    {
```

```

        "id": "a1e70177-877e-4ad3-8403-317f4a3ee291",
        "properties": {
            "name": "h1_rt01",
            "regionName": "Tokyo"
        }
    }
]
}

```

取得したノード情報のノード名、リージョングループ名から意図したノードであることを確認し、ノードの ID を取得します。

## 2. ネットワークインターフェイス「GigaEthernet0」の ID を調べます。

ネットワークインターフェイス一覧取得の API を以下のクエリーパラメーターを付加して実行します。

- ネットワークインターフェイス名による絞り込み:

```
properties@name==GigaEthernet0
```

- 取得対象のネットワークインターフェイス情報:

ネットワークインターフェイスの ID に加えて、ifIndex 値、ネットワークインターフェイス名を取得します。

```
props=id,properties.ifIndex,properties.name
```

API の詳細は、「[4.1.4 ネットワークインターフェイス一覧取得 \(38 ページ\)](#)」を参照してください。

具体的な Web API のリクエストとレスポンスの内容は以下の通りです。

- リクエスト:

```

GET /api/v1/nodes/a1e70177-877e-4ad3-8403-317f4a3ee291/interfaces?
properties%40name=%3DGigaEthernet0&props=id%2Cproperties.ifIndex%2
Cproperties.name HTTP/1.1
Date: Fri, 02 Oct 2020 00:25:57 GMT
Host: ims.nec.com
Authorization: SharedKeyLite MkhnoAv4xJI0p+egYhDYLq6bOhTBlMDQLwqn8
atgWPI=:xRUCo92TNFWQZfS9yCJmrXDywxRinzHdlSanmkzD4EA=

```

- レスポンス:

```

HTTP/1.1 200 OK
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Transfer-Encoding: chunked
Expires: 0
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Fri, 02 Oct 2020 00:25:57 GMT
X-Frame-Options: DENY
Content-Type: application/json; charset=UTF-8

```

```
{
  "totalCount": 1,
  "data": [
    {
      "id": "6192a2f2-223d-4bc9-811a-5b190f96dcccb",
      "properties": {
        "ifIndex": 1,
        "name": "GigaEthernet0"
      }
    }
  ]
}
```

取得したネットワークインターフェイス情報を確認し、ネットワークインターフェイスの ID を取得します。

3. 対象のネットワークインターフェイスに対するアプリケーション通信の情報を取得します。

特定ネットワークインターフェイスのフローデータ TopN 取得 の API を適切にリクエストボディを指定して実行します。

API の詳細は、「[4.3.6 特定ネットワークインターフェイスのフローデータ TopN 取得 \(57 ページ\)](#)」を参照してください。

ここでは、通信量の多いアプリケーション Top50 を取得します。

- リクエスト:

```
POST /nfa/api/v1/interfaces/6192a2f2-223d-4bc9-811a-5b190f96dcccb/flowdata/application/search HTTP/1.1
Content-Length: 116
Date: Fri, 02 Oct 2020 01:21:16 GMT
Host: ims.nec.com
Content-Type: application/json; charset=utf-8
Authorization: SharedKeyLite MkhnoAv4xJI0p+egYhDYLq6bOhTB1MDQLwqn8atgWPI=:kT7vV8C4qkYDGB14Sc8lfN5NPtUZvi6SZEkw0NrSpY=

{
  "startDate": "2020-09-11T00:00:00+09:00",
  "endDate": "2020-09-26T23:59:59+09:00",
  "topN": 50,
  "valueUnit": "bytes"
}
```

- レスポンス:

```
HTTP/1.1 200 OK
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Transfer-Encoding: chunked
Expires: 0
Pragma: no-cache
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Date: Fri, 02 Oct 2020 01:21:16 GMT
X-Frame-Options: DENY
Content-Type: application/json; charset=UTF-8
```

```

{
  "dnsResolving": null,
  "statisticsWidgetData": {
    "LINE": {
      "rows": [
        {
          "id": "1",
          "entityId": "9",
          "name": "ftp-data",
          "statisticsValue": 94793.2
        },
        {
          "id": "2",
          "entityId": "53",
          "name": "http",
          "statisticsValue": 92242.2
        },
        :
        (略)
        :
        {
          "id": "50",
          "entityId": "887",
          "name": "univ-appserver",
          "statisticsValue": 1893.2
        }
      ],
      "data": [
        { "id": "1", "value": 90071, "time": "2020-09-19T15:00:00Z" },
        { "id": "2", "value": 89920, "time": "2020-09-19T15:00:00Z" },
        :
        (略)
        :
        { "id": "50", "value": 1939, "time": "2020-09-19T15:00:00Z" },
        { "id": "1", "value": 96144, "time": "2020-09-19T21:00:00Z" },
        { "id": "2", "value": 96593, "time": "2020-09-19T21:00:00Z" },
        :
        (略)
        :
        { "id": "50", "value": 1923, "time": "2020-10-17T09:00:00Z" }
      ],
      "otherData": null
    },
    "PIE": {
      "rows": [
        {
          "id": "1",
          "entityId": "9",
          "name": "ftp-data",
          "statisticsValue": 7166372400
        },

```

```

        {
            "id": "2",
            "entityId": "53",
            "name": "http",
            "statisticsValue": 6973511400
        },
        :
        (略)
        :
        {
            "id": "50",
            "entityId": "887",
            "name": "univ-appserver",
            "statisticsValue": 143132400
        }
    ],
    "data": [
        { "id": "1", "value": 7166372400,
          "time": null },
        { "id": "2", "value": 6973511400,
          "time": null },
        :
        (略)
        :
        { "id": "50", "value": 143132400,
          "time": null }
    ],
    "otherData": null
}
}
}

```

4. 取得したアプリケーション通信の情報を確認します。

折れ線グラフ用の統計情報からは、各アプリケーションの時系列での通信量の変化を確認することができます。

円グラフ用の統計情報からは、アプリケーション毎の通信量の指定期間における合計値を確認することができます。

# 付録 A. レポート作成用サンプルマクロ

IMS コンポーネントは、Microsoft Excel を用いて性能情報のレポートを作成するためのサンプルマクロを提供しています。

ここでは、提供するサンプルマクロの概要や使用方法について説明します。

サンプルマクロは、インストールメディアの以下のパスに格納しています。

- \IMS\tools\report\IMS\_SampleReport\_V1.0.xlsm

## 注意

サンプルマクロを利用するためには、利用環境において、.NET Framework 3.5 をインストール、または、機能を有効にする必要があります。

Windows 10 においては、以下の手順で機能を有効にすることができます。

1. [コントロール パネル]から[プログラムと機能]を選択します。
2. [Windows の機能の有効化または無効化]を選択します。  
Windows の機能ダイアログが表示されます。
3. [.NET Framework 3.5 (.NET 2.0 および 3.0 を含む)]チェックボックスをオンにします。  
再起動を求められた場合は、コンピュータの再起動を実施します。

## A.1 サンプルマクロの概要

提供するサンプルマクロの概要について説明します。

提供するサンプルマクロでは、Web API を用いて指定期間における SNMP データ、または、フローデータを取得し、グラフの作成、データの出力を行います。

取得可能なデータ種別は、以下の通りです。

- SNMP データ (NetvisorPro 利用時)
  - CPU 使用率
  - メモリ使用率
  - 入力インターフェイス使用率
  - 出力インターフェイス使用率
  - 入力パケット損失率
  - 出力パケット損失率
  - 入力パケットエラー率
  - 出力パケットエラー率
- フローデータ (NFA 利用時)



- 通信量の多い入力インターフェイス
- 通信量の多い出力インターフェイス
- 通信量の多いアプリケーション (線グラフ、または、円グラフ)
- 通信量の多い IP プロトコル (線グラフ、または、円グラフ)
- 通信量の多い DSCP (線グラフ、または、円グラフ)
- 通信量の多いカンバセーション (2 点間)
- 通信量の多い送信元 IP アドレス
- 通信量の多い宛先 IP アドレス
- 通信量の多い送信元エンドポイントグループ
- 通信量の多い宛先エンドポイントグループ
- 通信量の多い送信元 AS (Autonomous System)
- 通信量の多い宛先 AS (Autonomous System)

データの取得対象とするノード、または、インターフェイスは最大で 10 個までを同時に指定することができます。

対象件数 (TopN) の指定は、最大で 100 まで指定することができます。ただし、対象件数 (TopN) に 20 を超える値を指定した場合は、グラフの作成は行わず、グラフ作成用のデータの出力のみ行います。

## A.2 サンプルマクロの使用方法

提供するサンプルマクロの使用方法について説明します。

サンプルマクロは、インストールメディアから任意の端末にコピーして利用します。

サンプルマクロを Microsoft Excel で開いた際、必ず、マクロの有効化(コンテンツの有効化)を実施してください。

具体的な操作手順は以下の通りです。

1. **[基本設定]**シートの Web API アクセスキーを入力します。

**[Access Key ID]**、および、**[Secret Access Key]**は、Web コンソールのプロファイル編集画面から発行、確認することができます。

2. **[基本設定]**シートの IMS サーバーの情報を入力します。

Web コンソールにアクセスする場合と同じ**[ドメイン名(FQDN)]**、**[ポート番号]**、**[プロトコル]**を指定します。

3. **[条件入力]**シートに取得対象データの条件を指定します。

単一のノード、または、インターフェイスを指定する場合と複数のノードおよびインターフェイスを指定する場合とは、入力するシートが異なります。

4. グラフ作成を実行します。

[条件入力]シートの右上に配置する[**グラフ作成**]アイコンをクリックするとマクロが実行され、グラフ作成処理が開始されます。

処理が完了するとグラフおよびデータを出力するシートが作成されます。

5. 作成されたシートの内容を確認します。

上記の操作を繰り返し実施することで、様々なデータを取得することができます。

サンプルマクロが作成するグラフでは運用に支障がある場合は、得られたデータから、Microsoft Excel を用いて集計、グラフ作成等の作業を実施してください。

---

**WebSAM Network Management**  
**Web API 1.0**  
**リファレンスマニュアル**

**IMS00AJ0100-01**

**2020 年 11 月 第 1 版 発行**

**日本電気株式会社**

---

**© NEC Corporation 2020**