



Flowmon

Networks



FlowMon REST API 8.01.01

Developer Guide

June 3, 2016

Contents

1	Introduction	6
2	Authorization	7
2.1	Obtaining authentication token	7
2.2	Getting Rest API information	8
2.2.1	Getting list of available methods	8
2.2.2	Getting current version of FlowMon	8
3	Parameters passing	10
3.0.3	GET Method	10
3.0.4	POST Method	10
3.0.5	PUT Method	11
3.0.6	DELETE Method	12
4	General Data Types	12
4.0.7	Validation Response	12
4.0.8	Dependency Object	14
5	FlowMon Configuration Center	15
5.1	Overview	16
5.1.1	Data Types	16
5.1.2	Methods	18
5.1.2.1	Full Device Info	18
5.1.2.2	Basic Device Info	20
5.1.2.3	Get CPU Usage	20
5.1.2.4	Get RAM Usage	21
5.1.2.5	Get Data Disk Usage	21
5.1.2.6	Get System Disk Usage	22
5.1.2.7	Get Boot Disk Usage	22
5.1.2.8	Get Interfaces	23
5.1.2.9	Get Interfaces Types	24
5.1.2.10	Get Exporters	25
5.1.2.11	Get Exporter States	26
5.1.2.12	Get Exporter Protocol	27
6	FlowMon Monitoring Center	28
6.1	Profiles	29
6.1.1	Data Types	29

6.1.2	Get All Profiles	30
6.1.3	Get Profile Detail	32
6.1.4	Creating New Profile	33
6.1.5	Modify Existing Profile	36
6.1.6	Delete Existing Profile	37
6.1.7	Get Profile Dependencies	38
6.2	Analysis	40
6.2.1	General Data Types	40
6.2.2	Chart Data Types	41
6.2.3	Analysis Data Types	42
6.2.4	General Analysis Methods	46
6.2.4.1	Get Profile Channels	46
6.2.4.2	Get Protocols	47
6.2.4.3	Get Columns Types	48
6.2.4.4	Get Columns	49
6.2.4.5	Get All Stored Filters	50
6.2.4.6	Get List Flows Outputs Definitions	50
6.2.5	Getting Chart Data Methods	52
6.2.5.1	Getting Chart Data Method	52
6.2.5.2	Getting Multiple Profiles Chart Data Method	53
6.2.6	Flows Search Related Methods	56
6.2.6.1	Get Flows Order Bys	56
6.2.6.2	Get Flows Listing	57
6.2.7	Statistics Search Related	58
6.2.7.1	Get Statistics Limit Types	58
6.2.7.2	Get Allowed Limit Operators	59
6.2.7.3	Get Statistics Methods	59
6.2.7.4	Get Statistics Order By Values	60
6.2.7.5	Get Aggregation Fileds	60
6.2.7.6	Get Statistics	61
6.2.8	Asynchronous Results	63
6.2.8.1	Get All Results	63
6.2.8.2	Get Result Data	64
6.3	Reports	66
6.3.1	Data Types	66
6.3.1.1	Chapter And Related Data Types	66
6.3.1.2	Getting Chapters Data Types	68
6.3.1.3	Report	70
6.3.2	Methods	71

6.3.2.1	Business Hours	71
6.3.2.1.1	Get List of All Business Hours	71
6.3.2.1.2	Get One Business Hours Detail	72
6.3.2.1.3	Create New Bussines Hours	73
6.3.2.1.4	Modify Existing Bussines Hours	74
6.3.2.1.5	Delete Existing Bussines Hours	76
6.3.2.2	Blacklists	76
6.3.2.2.1	Get Blacklists	76
6.3.2.2.2	Get One Blacklist Detail	77
6.3.2.2.3	Create New Blacklist	78
6.3.2.2.4	Modify Existing Blacklists	78
6.3.2.2.5	Delete Existing Blacklist	79
6.3.2.2.6	Get Dependencies	79
6.3.2.3	Chapters	80
6.3.2.3.1	Get Chapters	80
6.3.2.3.2	Get One Chapter Detail	81
6.3.2.3.3	Create New Chapters	82
6.3.2.3.4	Modify Existing Chapter	84
6.3.2.3.5	Delete Existing Chapter	86
6.3.2.3.6	Get Dependencies	86
6.3.2.4	Reports	87
6.3.2.4.1	Get Reports	87
6.3.2.4.2	Get One Report Detail	89
6.3.2.4.3	Create New Reports	90
6.3.2.4.4	Modify Existing Report	92
6.3.2.4.5	Delete Existing Report	94
6.3.2.4.6	Get Dependencies	95
6.3.2.5	Email Reports	96
6.3.2.5.1	Get All Email Reports	96
6.3.2.5.2	Get One Email Report Detail	97
6.3.2.5.3	Create New Email Report	97
6.3.2.5.4	Modify Existing Email Reports	99
6.3.2.5.5	Delete Existing Report	100
6.3.2.6	Get Chapter Data	100
6.4	Sources	103
6.4.1	Data Types	103
6.4.2	Methods	103
6.4.2.1	Get Sources List	103
6.5	Active Devices	105

6.5.1	Data Types	105
6.5.2	Methods	106
6.5.2.1	Get Active Devices Data	106
7	FlowMon Dashboard	108
7.1	Flowmon Dashboard	109
7.1.1	Dashboard Data Types	109
7.1.2	Dashboard Methods	111
7.1.2.1	Get Dashboards Method	111
7.1.3	Dashboard Widget Methods	112
7.1.3.1	Get Dashboard Widgets Method	112
7.1.4	Get Widget Data Method	113
7.1.4.1	Get Widget Data Method	113
8	System	115
8.1	System Messages	116
8.1.1	System Messages Types	116
8.1.2	System Messages Methods	116
8.1.2.1	Get Message Severity Method	116
8.1.2.2	Get Message Method	117
9	Examples For Javascript In Web Page	119
10	Examples For CURL	121
11	REST API changes	122

1 Introduction

This document describes usage of REST API for FlowMon devices which is available in Invea OS version 7.01.00 and newer.

The next chapter describes authorization process for REST API access. Following chapters contains description and usage of data types and methods.

2 Authorization

The API for interception management uses the OAuth 2.0 authorization. An application, which want to use API methods, has to request for granting access token using following HTTPS request (change only the values of the username and password properties).

2.1 Obtaining authentication token

New access token request

```
POST /resources/oauth/token
FORM-DATA grant_type=password
           client_id=invea-tech
           username=<user_login>
           password=<user_password>
```

The server will verify username and password and returns following response:

```
{
  "access_token": "AnlSCIWYbchsCc5sdc5ac4caca8a2",
  "token_type": "bearer",
  "expires_in": 3600,
  "refresh_token": "DS6SA512ADCVa51adc54VDS51VD5"
}
```

Now the application has to add this access token to every call of API method. The token has to be inserted in the **Authorization** header field of HTTPS requests and value of this field has to be "<token_type> <access_token>" from **/resources/oauth/token** response. So for the example the **Authentication** header value will look like "**bearer AnlSCIWYbchsCc5sdc5ac4caca8a2**". The access token is valid for <expires_in> seconds. Then the application can request new access token using previous HTTPS request or it can use <refresh_token> field to acquire new token without supplying of username and password.

Example of refresh token request:

```
POST /resources/oauth/token
FORM-DATA grant_type=refresh_token
           refresh_token=DS6SA512ADCVa51adc54VDS51VD5
           client_id=invea-tech
```

2.2 Getting Rest API information

There is new possibility of getting REST API available methods since Invea OS version 7.2.0.

2.2.1 Getting list of available methods

- Returns list of available methods

Example Usage

```
GET /rest/methods
```

```
RETURNS:
```

```
[
  {
    "url": "/rest/fcc/device",
    "method": "GET"
  },
  {
    "url": "/rest/fcc/device/info",
    "method": "GET"
  },
  {
    "url": "/rest/fcc/device/cpu",
    "method": "GET"
  },
  {
    "url": "/rest/fcc/device/ram",
    "method": "GET"
  },
  ...
]
```

2.2.2 Getting current version of FlowMon

- Returns Current version of FlowMon device.
- @returns 200 { "version": String("Version") }

Example Usage

```
GET /rest/version
```

```
RETURNS:
```

```
[  
  {  
    "version": "7.01.00"  
  }  
]
```

3 Parameters passing

This chapter describes supported ways of parameters passing for different HTTP methods.

3.0.3 GET Method

Generally GET method is used for getting data from server. Parameters specifying this reading are passed as URL parameters.

Plain HTTP request example

```
GET /rest/fmc/profiles?fields=["id","channels.id"] HTTP/1.1
Host: 192.168.51.78
Authorization: bearer 731489f1d74916b05cab6dbf43b9bd4d727a106c1e23b8df5e26aaf6c8594bdd
Cache-Control: no-cache
```

See **fields** parameter passed as URL parameter in the example.

3.0.4 POST Method

POST method is used for creating new objects. It can be also used for retrieving data in cases where passing parameters via URL will reach 2048 characters (URL length limit in browsers).

Supported way of parameters passing is in the HTTP request body as **multipart/form-data** content type (more at <https://en.wikipedia.org/wiki/MIME-Form-Data>).

Plain HTTP request example

```
POST /rest/fmc/profiles HTTP/1.1
Host: 192.168.51.78
Authorization: bearer 45208fa6ee2e6ec057e7680edd882fa9b0272681341c5e536ec852987cb99234
Cache-Control: no-cache
Postman-Token: 2ffa0b77-8455-1dad-32be-c9ce1879d183
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="entity"
```

```
{ "name": "mail", "parent": "live", "start": "2015-11-02 20:00", "type": "real",
  "status": "OK", "history": false, "description": "Mail protocols", "maxsize": 1024,
  "channels": [ { "id": "smtp", "name": "smtp", "sign": "+",
    "color": "#FF6248", "filter": "(proto tcp or proto udp) and port 25",
    "sources": [ ] } ] }
---WebKitFormBoundary7MA4YWxkTrZu0gW
```

3.0.5 PUT Method

HTTP PUT method is used for updating existing objects.

Supported way of parameters passing is in the HTTP request body as **x-www-form-urlencoded** content type (more at <https://en.wikipedia.org/wiki/Percent-encoding>).

Plain HTTP request example

```
PUT /rest/fmc/profiles HTTP/1.1
Host: 192.168.51.78
Authorization: bearer 45208fa6ee2e6ec057e7680edd882fa9b0272681341c5e536ec852987cb99234
Cache-Control: no-cache
Postman-Token: 4fcec97a-0460-58ae-aff8-84f54faf3bc6
Content-Type: application/x-www-form-urlencoded

entity=%7B++++%22id%22%3A+%22myGroup%2Fmynice-a4b2bd%22%2C++++%22name%22%3A+%22my+nice+profile%22%2C++++%22parent%22%3A+%22live%22%2C++++%22description%22%3A+%22this+is+my+profile%22%2C++++%22start%22%3A+%222015-05-23+12%3A10%22%2C++++%22type%22%3A+%22shadow%22%2C++++%22status%22%3A+%22new%22%2C++++%22history%22%3A+false%2C++++%22group%22%3A+%22myGroup%22%2C++++%22end%22%3A+%222015-06-01+10%3A15%22%2C++++%22maxsize%22%3A+100%2C++++%22channels%22%3A+%5B++++%7B++++%22id%22%3A+%22second-976978%22%2C++++%22name%22%3A+%22second+channel%22%2C++++%22sign%22%3A+%22-%22%2C++++%22color%22%3A+%22%23f44100%22%2C++++%22parent%22%3A+%5B+%22p3000-bb5174b%22+%5D%2C++++%22filter%22%3A+%22any%22%2C++++%22sources%22%3A+%5B+%5D++++%7D%2C++++%7B++++%22id%22%3A+%22firstc-9dcd29%22%2C++++%22name%22%3A+%22first+channel%22%2C++++%22sign%22%3A+%22%2B%22%2C++++%22color%22%3A+%22%232466f4%22%2C++++%22parent%22%3A+%5B+%22p3000-bb5174b%22+%5D%2C++++%22filter%22%3A+%22any%22%2C++++%22sources%22%3A+%5B+%5D++++%7D++++%5D+%7D
```

3.0.6 DELETE Method

For removing existing objects use DELETE HTTP method. There is usually one parameter with identification of existing object. Supported way of parameters passing is via URL parameters.

Plain HTTP request example

```
DELETE /rest/fmc/chapters/239 HTTP/1.1
Host: 192.168.51.78
Authorization: bearer 45208fa6ee2e6ec057e7680edd882fa9b0272681341c5e536ec852987cb99234
Cache-Control: no-cache
Postman-Token: 472628e7-d0d5-04c6-6a88-f7c011ce3ead
Content-Type: application/x-www-form-urlencoded
```

Parameter value **239** indentifying chapter is part of URL.

4 General Data Types

4.0.7 Validation Response

Validation Response is used when validation of given input fails. Validation Response consists of array of **ValidationError** objects and **checkedData** object.

ValidationError contains description of validation error, field identificator and optional parameters.

Validation Response Definition

```
ValidationError = {
  field: String("Dot separated path to invalid field within checked data"),
  message: String("Validation error message"),
  params: Array(String("Validation message parameters"))
}

ValidationResult = {
  validationResults: Array(ValidationError),
  checkedData: Object("Complete data that has been validated")
}
```

Example:

```
{
  validationResults:
    [{
      field: "name",
      message: "Name not unique",
      params: []
    },
    {
      field: "letters",
      message: "Too many values",
      params: []
    },
    {
      field: "innerObject.value",
      message: "Value must be integer in ranges.",
      params: [1, 100] //allowed range limits
    },
    {
      field: "innerArray.1.value",
      message: "Value too big.",
      params: [40] //allowed maximal value
    }
  ],

  checkedData: {
    id: null,
    name: "fd",
    type: "EventCategory",
    letters: ["a", "b", "c"],

    innerObject: {
      value: 123
    },

    innerArray: [
      {name: "t1", value: 32},
      {name: "t2", value: 45}
    ]
  }
}
```

4.0.8 Dependency Object

Dependency object tells which other objects rely on this object. For example report chapter rely on its profile. If you want to delete profile it will have influence on its report chapter. If you need to find out which objects are connected to you object use proper REST API method. This method will return Dependency object with following definition:

Dependency Object Definition

```
Dependencies = {  
  action: Action("Action the items are checked against. e.g. delete"),  
  items: [objects("Checked objects, may be in short form.")],  
  
  dependent: [  
    {  
      // Action that would be performed in case checked action was performed  
      action: String("delete|update"),  
      type: String("Data type of objects"),  
      items: [Object]  
    },  
    ...  
  ]  
}
```

5 FlowMon Configuration Center

5.1 Overview

5.1.1 Data Types

Following data structures are used in query parameters or results.

Asterisk sign [?] means that this field is optional.

CPU Load

```
CpuLoad = {  
    // Lower number means smaller CPU utilization.  
    "avg1Min": Number(Float, "Average CPU load in last minute."),  
    "avg5Min": Number(Float, "Average CPU load in last 5 minutes."),  
    "avg15Min": Number(Float, "Average CPU load in last 15 minutes."),  
}
```

Disc Usage

```
DiskUsage = {  
    "total": Integer("Bytes"),  
    "used": Integer("Sum of all used bytes"),  
    ?"consumers": Array(SpaceConsumer)  
}
```

Space Consumer

```
SpaceConsumer = {  
    "name": String("Name of application that consumes disk space."),  
    "used": Integer("Bytes used by this consumer")  
}
```

Device Information

```
DeviceInfo = {  
    "deviceName": String,  
    "onlineTime": Integer("Number of days the device is online."),  
    "disks": Array(DiskInfo)  
}
```

Disc Information

```
DiskInfo = {  
    "name": String("Disk name"),  
    "isSmart": Boolean,  
    "smartError": Boolean  
}
```


Device Full Information

```
FullDeviceInfo = {  
  "deviceInfo": DeviceInfo,  
  "cpuLoad": CpuLoad,  
  "ramUsage": DiskUsage, // In Megabytes  
  "systemDiskUsage": DiskUsage,  
  "bootDiskUsage": DiskUsage,  
  "dataDiskUsage": DiskUsage  
}
```

Network Interface Types

```
InterfaceType = {  
  "id": Integer,  
  "label": String  
} << InterfaceTypes;
```

Network Interface Information

```
InterfaceInfo = {  
  "id": String,  
  "type": Integer(InterfaceType.id),  
  "linkDetected": Integer(Boolean),  
  "mac": MacAddress,  
  "ip": String("IP address"),  
  "speed": Integer("Interface speed in Mb/s"),  
  "rxBytes": Integer("Recieved bytes"),  
  "txBytes": Integer("Transferred bytes"),  
  "mtu": Integer("MTU value in bytes")  
}
```

Exporting Protocol Type

```
ExportProtocol = {  
  "id": String,  
  "label": String  
} << ExportProtocols;
```

Target Of Exporting

```
ExporterTarget = {  
  "id": Integer,  
  "ip": String(IP),  
  "port": Integer(Port),  
  "exportProtocol": String(ExportProtocol.id)  
}
```

Exporter State

```
ExporterState = {
```

```
"id": Integer,  
"label": String  
} << ExporterStates  
  
ExporterStates = [  
  {"id": 0, "label": "running"},  
  {"id": 1, "label": "stopped"},  
  {"id": 2, "label": "error_stopped"},  
  {"id": 3, "label": "error_running"},  
  {"id": 4, "label": "not_licensed"}  
]
```

Exporter Information

```
ExporterInfo = {  
  "id": Integer,  
  "interface": String,  
  "state": ExporterState,  
  "targets": Array(ExporterTarget),  
  "activeTimeout": Integer("Seconds")  
  "inactiveTimeout": Integer("Seconds")  
}
```

5.1.2 Methods

5.1.2.1 Full Device Info

GET /rest/fcc/device

- Returns device info including CPU load and all disks usages.
- @returns 200 FullDeviceInfo (5.1.1)

Example Usage

```
GET /rest/fcc/device
```

RETURNS:

```
{  
  "deviceInfo": {  
    "deviceName": "localhost",  
    "onlineTime": 6,
```

```
    "disks": [
      {
        "name": "/dev/sda",
        "isSmart": 0,
        "smartError": 0
      }
    ]
  },
  "cpuLoad": {
    "avg1Min": "0.13",
    "avg5Min": "0.08",
    "avg15Min": "0.07"
  },
  "ramUsage": {
    "total": 3017,
    "used": 2115
  },
  "systemDiskUsage": {
    "total": 5165522944,
    "used": 1862746112
  },
  "bootDiskUsage": {
    "total": 254671872,
    "used": 146906112
  },
  "dataDiskUsage": {
    "total": 1073741824,
    "used": 4726534144,
    "consumers": [
      {
        "name": "Monitoring center",
        "used": 1946300784
      },
      {
        "name": "Plugins",
        "used": 0
      },
      {
        "name": "Others",
        "used": 2780233360
      }
    ]
  }
}
```

5.1.2.2 Basic Device Info

GET /rest/fcc/device/info

- Returns basic device info
- @returns 200 DeviceInfo (5.1.1)

Example Usage

```
GET /rest/fcc/device/info
```

```
RETURNS:
```

```
{
  "deviceName": "localhost",
  "onlineTime": 6,
  "disks": [
    {
      "name": "/dev/sda",
      "isSmart": 0,
      "smartError": 0
    }
  ]
}
```

5.1.2.3 Get CPU Usage

GET /rest/fcc/device/cpu

- Returns CPU load information.
- @returns 200 CpuLoad (5.1.1)

Example Usage

```
GET /rest/fcc/device/cpu
```

```
RETURNS:
```

```
{
  "avg1Min": "0.03",
  "avg5Min": "0.09",
  "avg15Min": "0.12"
}
```

5.1.2.4 Get RAM Usage

GET /rest/fcc/device/ram

- Returns current RAM usage
- @returns 200 DiskUsage (5.1.1)

Example Usage

```
GET /rest/fcc/device/ram
```

```
RETURNS:
```

```
{
  "total": 3017,
  "used": 2131
}
```

5.1.2.5 Get Data Disk Usage

GET /rest/fcc/device/datadisk

- Returns current data disk usage. Array of space consumers should be included.
- @returns 200 DiskUsage (5.1.1)

Example Usage

```
GET /rest/fcc/device/datadisk
```

```
RETURNS:
```

```
{
  "total": 1073741824,
  "used": 4790169600,
  "consumers": [
    {
      "name": "Monitoring center",
      "used": 2018890096
    },
    {
      "name": "Plugins",
      "used": 0
    }
  ]
}
```

```
    },  
    {  
      "name": "Others",  
      "used": 2771279504  
    }  
  ]  
}
```

5.1.2.6 Get System Disk Usage

GET /rest/fcc/device/systemdisk

- Returns current system disk usage
- @returns 200 DiskUsage (5.1.1)

Example Usage

```
GET /rest/fcc/device/systemdisk
```

RETURNS:

```
{  
  "total": 5165522944,  
  "used": 1920598016  
}
```

5.1.2.7 Get Boot Disk Usage

GET /rest/fcc/device/bootdisk

- Returns current boot disk usage
- @returns 200 DiskUsage (5.1.1)

Example Usage

```
GET /rest/fcc/device/bootdisk
```

RETURNS:

```
{
```

```
"total": 254671872,  
"used": 146906112  
}
```

5.1.2.8 Get Interfaces

GET /rest/fcc/interfaces

- Returns collection of all interfaces and its states
- @returns 200 [InterfaceInfo] (5.1.1)

Example Usage

GET /rest/fcc/interfaces

RETURNS:

```
[  
  {  
    "id": "eth0",  
    "type": 1,  
    "linkDetected": 1,  
    "mac": "00:0C:29:0D:DD:08",  
    "ip": "192.168.51.145",  
    "speed": 1000,  
    "rxBytes": 616767480,  
    "txBytes": 125958402,  
    "mtu": 1500  
  },  
  {  
    "id": "eth1",  
    "type": 3,  
    "linkDetected": 1,  
    "mac": "00:0C:29:0D:DD:12",  
    "ip": "",  
    "speed": 1000,  
    "rxBytes": 2182014814,  
    "txBytes": 1930,  
    "mtu": 1500  
  },  
  {  
    "id": "eth2",  
    "type": 2,  
    "linkDetected": 1,  

```

```
    "mac": "00:50:56:9E:E3:AE",
    "ip": "",
    "speed": 1000,
    "rxBytes": 4429527536973,
    "txBytes": 0,
    "mtu": 1500
  },
  {
    "id": "eth3",
    "type": 2,
    "linkDetected": 1,
    "mac": "00:50:56:9E:1B:3D",
    "ip": "",
    "speed": 1000,
    "rxBytes": 4428979216548,
    "txBytes": 0,
    "mtu": 1500
  }
]
```

5.1.2.9 Get Interfaces Types

GET /rest/fcc/interfaces/types

- Returns all interface types
- @returns 200 [InterfaceType] (5.1.1)

Example Usage

GET /rest/fcc/interfaces/types

RETURNS:

```
[
  {
    "id": 1,
    "label": "management"
  },
  {
    "id": 2,
    "label": "monitoring"
  },
  {
    "id": 3,
```



```
"label": "export"
}
]
```

5.1.2.10 Get Exporters

GET /rest/fcc/exporters

- Returns all exporter and its states - since version 7.1.1
- @returns 200 [ExporterInfo] (5.1.1)

Example Usage

GET /rest/fcc/exporters

RETURNS:

```
[
  {
    "id": 1,
    "interface": "eth2",
    "state": {
      "id": 0,
      "label": "running"
    },
    "targets": [
      {
        "id": 0,
        "ip": "localhost",
        "port": 3000,
        "exportProtocol": "v9"
      }
    ],
    "activeTimeout": 300,
    "inactiveTimeout": 30
  },
  {
    "id": 2,
    "interface": "eth3",
    "state": {
      "id": 0,
      "label": "running"
    },
    "targets": [
```

```
[
  {
    "id": 0,
    "ip": "localhost",
    "port": 3001,
    "exportProtocol": "v5"
  },
  {
    "id": 1,
    "ip": "localhost",
    "port": 3001,
    "exportProtocol": "v5"
  }
],
"activeTimeout": 300,
"inactiveTimeout": 30
}
```

5.1.2.11 Get Exporter States

GET /rest/fcc/exporters/states

- @returns 200 [ExporterState] (5.1.1)

Example Usage

GET /rest/fcc/exporters/states

RETURNS:

```
[
  {
    "id": 0,
    "label": "running"
  },
  {
    "id": 1,
    "label": "stopped"
  },
  {
    "id": 2,
    "label": "error_stopped"
  },
  {
    "id": 3,
```

```
    "label": "error_running"
  },
  {
    "id": 4,
    "label": "not_licensed"
  }
]
```

5.1.2.12 Get Exporter Protocol

GET /rest/fcc/exporters/protocols

- @returns 200 [ExportProtocol] (5.1.1)

Example Usage

```
GET /rest/fcc/exporters/protocols
```

RETURNS:

```
[
  {
    "id": "ipfix",
    "label": "IPFIX"
  },
  {
    "id": "v9",
    "label": "NetFlow v9"
  },
  {
    "id": "v5",
    "label": "NetFlow v5"
  }
]
```

6 FlowMon Monitoring Center

6.1 Profiles

6.1.1 Data Types

These data types are available at Invea OS version 7.3.0 and newer.

Asterisk sign [?] means that this field is optional.

Profiles Representation

```
Profile = {  
    // id is optional only in case of creating new profile  
    ?id: String("Profile Id"),  
    name: String("Profile name"),  
  
    // if profile is not at any group this key is missing  
    ?group: String("Group identifier")  
  
    // if not defined live profile will be selected as parent  
    ?parent: String("Profile id")  
    ?description: String("")  
  
    // if missing profile will start from now  
    ?start: String(Timestamp)  
  
    // if end missing during creating -> profile will be continuous  
    ?end: String(Timestamp)  
  
    // if type is real this property is required  
    ?maxsize: Int("Maximal profile size (MB)")  
  
    // if not specified, no expiration will be set  
    ?expire: Int("Number of hours to keep")  
    type: String("real|shadow")  
  
    // only for reading, during creating or updating is not required  
    ?status: String("profile current status")  
  
    // if profile is continuous, this field is false ...  
    ?history: Boolean  
    channels: [ProfileChannel]  
}
```

Profile Channel Representation

```
ProfileChannel = {  
    // id is optional only in case of creating new channel  
    ?id: String("Profile Id"),  
    name: String("Channel name (translated)")  
    sign: String("+|-")  
    color: String("Hexa color definition")  
  
    // array of parent profile channels  
    parent: "*" | [Channel.id] // array of parent profile channels or string containing "*" all channels wildcard  
    filter: String("Filter")  
}
```

6.1.2 Get All Profiles

- **GET /rest/fmc/profiles**
- **POST /rest/fmc/getprofiles**
- **In case of using profiles paramter we recommend method POST /rest/fmc/getprofiles to avoid truncation because of GET method**
- Returns array of Profile objects describing details of user visible profiles.
- @params fields [String("Profile or channel key, e.g. "type" or "channel.color" ")] - if not mentioned, all keys will be delivered. Getting all object fields (especialy channels.filter and description) is time consuming operation. In case of large amout of profiles we recommend to specify only some fields (e.g. id and name) to get much faster response.
- @params profiles [Profile.id] - List of profiles to display, if not mentioned all profiles will be displayed
- @returns 204 No data "No profiles defined."
- @returns 200 [Profile] (6.1.1)

Example Usage

```
GET /rest/fmc/profiles
```

RETURNS:

```
[
  {
    "id": "Sources/AllSou-fe36e4",
    "name": "All Sources",
    "parent": "live",
    "description": "All Sources",
    "start": "2015-06-01 09:25",
    "type": "shadow",
    "status": "OK",
    "history": true,
    "group": "Sources",
    "end": "2015-06-01 09:25",
    "channels": [
      {
        "id": "localh-045279",
        "name": "localhost (192.168.3.155)",
        "sign": "+",
        "color": "#f44100",
        "parent": [
          "p3000-bb5174b",
          "p3000-d1c75f2",
          "cucori-e2e527",
          "zuzu-dba3c23c"
        ],
        "filter": "router ip 192.168.3.155"
      },
      {
        "id": "localh-1fd198",
        "name": "localhost (192.168.3.188)",
        "sign": "+",
        "color": "#ffa216",
        "parent": [
          "p3000-bb5174b",
          "zuzu-dba3c23c",
          "cucori-e2e527",
          "p3000-d1c75f2"
        ],
        "filter": "router ip 192.168.3.188"
      },
      {
        "id": "localh-cee4d4",
        "name": "localhost (192.168.51.145)",
        "sign": "+",
        "color": "#1bb500",
        "parent": [
          "p3000-bb5174b",
          "zuzu-dba3c23c",

```

```
        "cucori-e2e527",  
        "p3000-d1c75f2"  
    ],  
    "filter": "router ip 192.168.51.145"  
  }  
]  
},  
...  
]
```

6.1.3 Get Profile Detail

GET /rest/fmc/profiles/id?id=Profile.id

- Returns profile detail
- @params fields [String("Profile or channel key, e.g. "type" or "channel.color" ")] - if not mentioned, all keys will be delivered
- @returns 204 No data "No profiles defined."
- @returns 200 [Profile] (6.1.1)

Example Usage

```
GET /rest/fmc/profiles/id?id=Sources/AllSou-fe36e4
```

RETURNS:

```
{  
  "id": "Sources/AllSou-fe36e4",  
  "name": "All Sources",  
  "parent": "live",  
  "description": "All Sources",  
  "start": "2015-06-01 09:25",  
  "type": "shadow",  
  "status": "OK",  
  "history": true,  
  "group": "Sources",  
  "end": "2015-06-01 09:25",  
  "channels": [  
    {  
      "id": "localh-045279",  
      "name": "localhost (192.168.3.155)",
```



```
"sign": "+",
"color": "#f44100",
"parent": [
    "p3000-bb5174b",
    "p3000-d1c75f2",
    "cucori-e2e527",
    "zuzu-dba3c23c"
],
"filter": "router ip 192.168.3.155"
},
{
    "id": "localh-1fd198",
    "name": "localhost (192.168.3.188)",
    "sign": "+",
    "color": "#ffa216",
    "parent": [
        "p3000-bb5174b",
        "zuzu-dba3c23c",
        "cucori-e2e527",
        "p3000-d1c75f2"
    ],
    "filter": "router ip 192.168.3.188"
},
{
    "id": "localh-cee4d4",
    "name": "localhost (192.168.51.145)",
    "sign": "+",
    "color": "#1bb500",
    "parent": [
        "p3000-bb5174b",
        "zuzu-dba3c23c",
        "cucori-e2e527",
        "p3000-d1c75f2"
    ],
    "filter": "router ip 192.168.51.145"
}
]
}
```

6.1.4 Creating New Profile

POST /rest/fmc/profiles?entity=Profile

- Return newly created profile
- @params entity Profile (6.1.1)
- @returns 201 Profile (6.1.1) - with profile.Id and its Channels.Ids
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example of using

```
POST /rest/fmc/profiles?entity={
  "name": "my nice profile",
  "group": "myGroup",
  "parent": "live",
  "description": "this is my profile",
  "start": "2015-05-23 14:10",
  "maxsize": 100,
  "type": "real",
  "channels": [
    {
      "name": "first channel",
      "sign": "+",
      "color": "#2466f4",
      "parent": ["p3000-bb5174b"],
      "filter": "any"
    },
    {
      "name": "second channel",
      "sign": "-",
      "color": "#f44100",
      "parent": ["p3000-bb5174b"],
      "filter": "any"
    }
  ]
}
```

RETURNS:

```
{
  "id": "myGroup/mynice-d1a0f5",
  "name": "my nice profile",
  "parent": "live",
  "start": "2015-05-23 13:10",
  "type": "real",
  "status": "built 0",
  "history": false,
  "description": "this is my profile",
  "group": "myGroup",
  "end": "2016-02-05 14:00",
  "maxsize": 100,
```

```
"channels": [  
  {  
    "id": "firstc-eb451b",  
    "name": "first channel",  
    "sign": "+",  
    "color": "#2466f4",  
    "parent": [ "p3000-bb5174b" ],  
    "filter": "any"  
  },  
  {  
    "id": "second-a0021e",  
    "name": "second channel",  
    "sign": "-",  
    "color": "#f44100",  
    "parent": [ "p3000-bb5174b" ],  
    "filter": "any"  
  }  
]  
}
```

There is a restriction for channels order. During profile creating or editing all channels with (sign) + should be defined before channels with (sign) -. For example following channels order is invalid:

Listing:

```
{  
  ...  
  "channels": [  
    {  
      "name": "first channel",  
      "sign": "-",  
      "color": "#2466f4",  
      "parent": [ "p3000-bb5174b" ],  
      "filter": "any"  
    },  
    {  
      "name": "second channel",  
      "sign": "+",  
      "color": "#f44100",  
      "parent": [ "p3000-bb5174b" ],  
      "filter": "any"  
    },  
    {  
      "name": "third channel",  
      "sign": "-",  
      "color": "#f44100",  
      "parent": [ "p3000-bb5174b" ],  
      "filter": "any"  
    }  
  ]  
}
```

```
        "filter": "any"
      }
    ]
  }
}
```

6.1.5 Modify Existing Profile

PUT /rest/fmc/profiles?entity=Profile

- Modifies existing profile and return new Profile object. If moving profile to different group, profile id will be changed, because profile ID contains group name.
- @params entity Profile (6.1.1)
- @returns 201 Modified profile object
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example of using

```
PUT /rest/fmc/profiles?entity={
  "id": "myGroup/mynice-a4b2bd",
  "name": "my nice profile",
  "parent": "live",
  "description": "this is my profile",
  "start": "2015-05-23 12:10",
  "type": "shadow",
  "status": "new",
  "history": false,
  "group": "myGroup",
  "end": "2015-06-01 10:15",
  "maxsize": 100,
  "channels": [
    {
      "id": "second-976978",
      "name": "second channel",
      "sign": "-",
      "color": "#f44100",
      "parent": [ "p3000-bb5174b" ],
      "filter": "any"
    },
    {
      "id": "firstc-9dcd29",
```

```
        "name": "first channel",
        "sign": "+",
        "color": "#2466f4",
        "parent": [ "p3000-bb5174b" ],
        "filter": "any"
    }
]
}
```

RETURNS:

```
{
  "id": "myGroup/mynice-a4b2bd",
  "name": "my nice profile",
  "parent": "live",
  "description": "this is my profile",
  "start": "2015-05-23 12:10",
  "type": "shadow",
  "status": "OK",
  "history": false,
  "group": "myGroup",
  "end": "2015-06-01 15:15",
  "channels": [
    {
      "id": "second-976978",
      "name": "second channel",
      "sign": "-",
      "color": "#f44100",
      "parent": [ "p3000-bb5174b" ],
      "filter": "any"
    },
    {
      "id": "firstc-9dcd29",
      "name": "first channel",
      "sign": "+",
      "color": "#2466f4",
      "parent": [ "p3000-bb5174b" ],
      "filter": "any"
    }
  ]
}
```

6.1.6 Delete Existing Profile

DELETE /rest/fmc/profiles/id/?id=id | [[id, ...]

- Delete existing profile or several profiles.
- @params id Profile.id | | [Profile.id,...] (6.1.1)
- @returns 204
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example

```
DELETE /rest/fmc/profiles/id?id=myGroup/mynice-a4b2bd
```

RETURNS:

Returns 204

Example of using multiple delete

```
DELETE /rest/fmc/profiles/id?id=["myGroup/mynice-a4b2bd", "myGroup/mynice-bc35b5"]
```

RETURNS:

Returns 204

6.1.7 Get Profile Dependencies

@url GET /rest/fmc/dependencies/profiles/id?id=Profile.id

- Returns dependencies object describing what other objects depends on this entity. At every dependent is shown action which will be performed when entity will be deleted.
- @returns 200 Dependencies (4.0.8)
- @throws 401 header("X-FM-Message": "No permission to manipulate profiles")
- @throws 404 header("X-FM-Message": "Profile does not exist")

Example of using

Example:

```
GET /rest/fmc/dependencies/profiles/id?id=mail
```

Returns 200

```
{
  "action": "delete",
  "items": [ "mail"],
  "dependent": [
    {
      "action": "delete",
      "type": "chapter",
      "items": [
        {
          "id": 239,
          "name": "REST test chapter name"
        },
        {
          "id": 2626,
          "name": "Mail chapter"
        }
      ]
    }
  ],
  {
    "action": "delete",
    "type": "profile",
    "items": [
      {
        "id": "mailch-3786dc",
        "name": "mail-child"
      },
      {
        "id": "vcxzvc-d6d177",
        "name": "vcxzvcxz"
      }
    ]
  },
  {
    "action": "delete",
    "type": "alert",
    "items": [
      {
        "id": "mailAlert",
        "name": "mailAlert"
      }
    ]
  }
]
```

6.2 Analysis

Following data structures are used in query parameters or results.

Asterisk sign [?] means that this field is optional.

6.2.1 General Data Types

These data types are used in statistic, flows or chart queries.

Profile Channel

```
Channel = {  
  "id": String("Channel identification string"),  
  "name": String("Human readable channel name")  
}
```

Profile Representation

```
Profile = {  
  "id": String("Profile/source identification string"),  
  "name": String("Profile/source human readable name"),  
  // "Channels that belongs to given profile"  
  "?channels": Array(substringFromMostRightSlashToEnd(Channel.id))  
}
```

Note Profiles created on FlowMon versions 5.10 and older has Profile.id equal to Profile.name.

Function **substringFromMostRightSlashToEnd(Channel.id)** provides text from most right slash to the end of in Channel.id. For example if Channel.id is "Sources/AllSou-f122e7/127-0-0-1" this function returns "127-0-0-1".

IP Protocol Data Type

```
Protocol = {  
  "id": Integer,  
  "name": String  
} << Protocols
```

Base Search Object

```
BasicAnalysisSearch = {
```



```
"from": Timestamp,  
"to": Timestamp,  
"profile": String(Profile.id)  
}
```

- All **profile** values can be obtained via method at section 6.1.2.

6.2.2 Chart Data Types

Data types used with chart queries:

Chart Search Object

```
AnalysisChartSearch = {  
    // "Channels that belongs to given profile"  
    ?"channels": Array(substringFromMostRightSlashToEnd(Channel.id))  
    "chart": {  
        "measure": String("Measures") << ["flows", "packets", "bytes"],  
        "protocol": Integer(Protocol.id) // values are at url GET /rest/fmc/protocols  
    }  
} extends BasicAnalysisSearch
```

- All **channels** values can be obtained via method at section 6.2.4.1.
- All **protocol** values can be obtained via method at section 6.2.4.2.

Batch Chart Search Object

```
AnalysisChartBatchSearch = {  
    ?showOnly: Integer // Maximal count of returned values. If not specified used default 800  
} extends AnalysisChartSearch
```

Chart Point Data Representation

```
ChartDataRecord = [  
    Integer("Unix timestamp"),  
    Number("Required value (depends on given measure and protocol)")  
]
```

Chart data

```
ChannelChartData = {
```

```
"channel": Channel,  
"values": Array(ChartDataRecord)  
}  
  
AnalysisChartData = Array(ChannelChartData);
```

6.2.3 Analysis Data Types

Stored Filter Representation

```
AnalysisFilter = {  
  "name": String,  
  "definition": String  
}
```

Possible Column Value Types

```
ColumnType = {  
  "id": String,  
  "description": String,  
  "givenAs": String("Name of inner javascript type the value is given as")  
} << ColumnTypes
```

- Method listing all column types is listed in section 6.2.4.2.

Result Columns

```
NfColumn = {  
  "id": String("Identification of the column"),  
  "name": String("Full name of the column"),  
  "group": String("Name of the group column belongs to.'NBAR' for instance."),  
  "output": Boolean("1 if this column can be used in output definition."),  
  "typeHint": String(ColumnType.id, "Hint for client how to treat values of the  
    column (how to format them, or sort them in client.)")  
}
```

- All NfColumn values can be listed via method in section 6.2.4.4.

Saved Output

```
SavedOutput = {  
  "name": String,  
  "columns": Array(String(NfColumn[output=1].id))  
} << SavedOutputs
```

Asynchronous Result Data Type

```
AsyncSearchResult = {
  "id": Integer("Result id"),
  ?"name": String("Saved result name"),

  // states meanings : 0 - computing, 1 - finished, 2 - failed
  ?state: Integer("Result state id") << [0, 1, 2]

  ?"search": Object("Search object") << [FlowsListingAnalysisSearch,
                                           StatisticsAnalysisSearch]
}
```

IP Mask Data Type

```
IpMask = {
  "ipVersion": Integer("Ip version") << [4,6],
  "mask": Integer("IP mask")
}
```

List Flows

List Flows Query Object

```
FlowsListingAnalysisSearch = {
  // "Channels of selected profile to search data in"
  "channels": Array(substringFromMostRightSlashToEnd(Cahnnel.id)),
  ?"listing": {
    // values at method GET /rest/fmc/analysis/aggregations
    ?"aggregateBy": Array(String(AnalysisAggregation.id)), "Aggregate by")
    ?"sourceIpMask": IpMask,
    ?"targetIpMask": IpMask,
    ?"sortBy": String(AnalysisFlowsOrderBy.id) // values at GET /rest/fmc/analysis/flows/orderbys
  },

  ?"filter": String("Analysis filter settings. Saved
                    filters are expanded and added by client.")
} extends BasicAnalysisSearch
```

- All **AnalysisAggregation** values can be listed via method in section 6.2.7.5.

Sortable List Flows Fields

```
AnalysisFlowsOrderBy = {
  "id": String,
  "name": String
} << AnalysisFlowsOrderBys;
```

- All **AnalysisFlowsOrderBy** values can be listed via method in section 6.2.6.1.

Result Data Row

```
TableResultRow = {  
  {NfColumn.id}: {ColumnType.givenAs},  
}  
  
AnalysisTableData = Array(TableResultRow)
```

Statistics

Statistic Query Object

```
AnalysisStatistic = {  
  "id": String,  
  "name": String,  
  "group": String("Group name statistics belongs to"),  
}
```

- All **AnalysisStatistic** values can be listed via method in section 6.2.7.3.

Sortable Statistics Fields

```
AnalysisStatisticsOrderBy = {  
  "id": String,  
  "name": String  
} << AnalysisStatisticsOrderBys;
```

- All **AnalysisStatisticsOrderBy** values can be listed via method in section 6.2.7.4.

Field Which Can Be Used For Aggregation

```
AnalysisAggregation = {  
  "id": String,  
  "name": String  
} << AnalysisAggregations;
```

- All **AnalysisAggregation** values can be listed via method in section 6.2.7.5.

Statistic Limitation Object

```
LimitedValueType = {
    "id": String,
    "name": String
} << LimitedValueTypes;

LimitOperator = String("Allowed limit operator") << ["<", ">"];
```

- All **LimitedValueType** values can be listed via method in section 6.2.7.1.

Statistic Query Object

```
StatisticsAnalysisSearch {
    // "Channels of selected profile to search data in"
    "channels": Array(substringFromMostRightSlashToEnd(Cahnnel.id)),
    "statistics": {
        "statistics": String(AnalysisStatistic.id, "Id of statistic to be used"),
        // values at GET /rest/fmc/analysis/statistics/methods

        "orderBy": String(AnalysisStatisticsOrderBy.id)
        // values at GET /rest/fmc/analysis/statistics/orderbys
    },

    ?"specific": {
        ?"bidirectional": Boolean,
        ?"aggregateBy": Array(String(AnalysisAggregation.id), "Aggregate results by") // values at
        // GET /rest/fmc/analysis/aggregations
        ?"sourceIpMask": IpMask,
        ?"targetIpMask": IpMask
    },

    ?limit: {
        "onColumn": String(LimitedValueType.id),
        // values at GET /rest/fmc/analysis/statistics/limitvaluetypes

        "operator": String(LimitOperator),
        // values at GET /rest/fmc/analysis/statistics/limitoperators

        "value": Integer
    },

    ?"filter": String("Analysis filter settings.
        Saved filters are expanded and added by client."),
} extends BasicAnalysisSearch
```

Result Not Completed Object Yet

```
ResultNotCompletedYet {  
  "code":202,  
  "status":"error",  
  "message":{"_\"Requested result is not available yet.\""}  
}
```

6.2.4 General Analysis Methods

6.2.4.1 Get Profile Channels

GET /rest/fmc/profiles/channels

- Returns list of all chanel for given profile. Profile id is given by parameter, profileId can contain "/".
- @param profileId Profile.id (6.1.1)
- @throws 404 Not found in case none or invalid profileId is given
- @returns 204 No data
- @returns 200 [Channel] (6.1.1)

Example Usage

```
GET /rest/fmc/profiles/channels?profileId=mail
```

RETURNS:

```
[  
  {  
    "id": "mail/pop3",  
    "name": "pop3"  
  },  
  {  
    "id": "mail/spop3",  
    "name": "spop3"  
  },  
  {  
    "id": "mail/imap",  
    "name": "imap"  
  },  
]
```

```
{
  "id": "mail/imap",
  "name": "imap"
},
{
  "id": "mail/smtp",
  "name": "smtp"
}
]
```

6.2.4.2 Get Protocols

. GET /rest/fmc/protocols

- Returns list of supported IP protocols
- @returns 200 [Protocol] (6.2.1)

Example Usage

GET /rest/fmc/protocols

RETURNS:

```
[
  {
    "id": 0,
    "name": "All"
  },
  {
    "id": 1,
    "name": "ICMP"
  },
  {
    "id": 2,
    "name": "IGMP"
  },
  {
    "id": 4,
    "name": "IPv4"
  },
  {
    "id": 6,
    "name": "TCP"
  },
]
```

```
[
  {
    "id": 17,
    "name": "UDP"
  },
  {
    "id": 41,
    "name": "IPv6"
  },
  {
    "id": 58,
    "name": "Ipv6-ICMP"
  },
  {
    "id": 255,
    "name": "Other"
  }
]
```

6.2.4.3 Get Columns Types

GET /rest/fmc/columnTypes

- Returns list of all defined and used columns types.
- @returns 200 [ColumnType] (6.2.3)

Example Usage

GET /rest/fmc/columnTypes

RETURNS:

```
[
  {
    "id": "Bytes",
    "description": "Number of bytes",
    "givenAs": "Number"
  },
  {
    "id": "Bits",
    "description": "Number of bits",
    "givenAs": "Number"
  },
  {
    "id": "Packets",
```



```
    "description": "Number of bits",
    "givenAs": "Number"
  },
  {
    "id": "Dir",
    "description": "Number of bits",
    "givenAs": "String"
  },
  ...
]
```

6.2.4.4 Get Columns

GET /rest/fmc/analysis/nfcolumns

- Returns list of all defined and used net flow columns.
- @returns 200 [NfColumn] (6.2.3)

Example Usage

GET /rest/fmc/analysis/nfcolumns

RETURNS:

```
[
  {
    "id": "ts",
    "name": "Start Time - first seen",
    "group": "Date and time",
    "aggregate": 0,
    "output": 1,
    "typeHint": "Datetime"
  },
  {
    "id": "te",
    "name": "End Time - last seen",
    "group": "Date and time",
    "aggregate": 0,
    "output": 1,
    "typeHint": "Datetime"
  },
  {
    "id": "td",
    "name": "Duration",
```

```
    "group": "Date and time",
    "aggregate": 0,
    "output": 1,
    "typeHint": "TimeInterval"
  },
  ...
]
```

6.2.4.5 Get All Stored Filters

GET /rest/fmc/analysis/filters

- Returns list of saved analysis filters
- @returns 200 [AnalysisFilter] (6.2.3)

Example Usage

```
GET /rest/fmc/analysis/filters
```

RETURNS:

```
[
  {
    "name": "SYN flags only",
    "definition": "proto tcp and flags S~and not flags ARFPU"
  }
]
```

6.2.4.6 Get List Flows Outputs Definitions

GET /rest/fmc/analysis/outputdefinitions

- Returns all saved outputs
- @returns 200 [SavedOutput] (6.2.3)

Example Usage

```
GET /rest/fmc/analysis/outputdefinitions
```

RETURNS:

```
[
  {
    "name": "default",
    "columns": [
      "ts",
      "td",
      "pr",
      "sa",
      "sp",
      "da",
      "dp",
      "flg",
      "tos",
      "pkt",
      "byt",
      "fl"
    ]
  },
  {
    "name": "extended",
    "columns": [
      "ts",
      "td",
      "pr",
      "sa",
      "sp",
      "da",
      "dp",
      "flg",
      "tos",
      "pkt",
      "byt",
      "pps",
      "bps",
      "bpp",
      "fl"
    ]
  },
  ...
]
```

6.2.5 Getting Chart Data Methods

6.2.5.1 Getting Chart Data Method

GET|POST /rest/fmc/analysis/chart

- Returns array of chart values according to given search object.
- @param search AnalysisChartSearch (6.2.2)
- @param showonly Integer("Maximal count of returned values.") Only for FlowMon version 7.1.0. This parameter is required.
- @returns 200 [AnalysisChartData] (6.2.2)

Example Usage

```
GET /rest/fmc/analysis/chart?search={
  "from": "2014-11-04 00:05",
  "to": "2014-11-04 14:20",
  "profile": "live",
  "chart": {
    "measure": "traffic",
    "protocol": 0
  }
}&showonly=700
```

RETURNS:

```
[
  {
    "channel": {
      "id": "live/p9996",
      "name": "NetFlow-port9996"
    },
    "values": [
      [ 1415059800000, 0 ],
      [ 1415060100000, 0 ],
      [ 1415060400000, 0 ],
      ...
    ]
  },
  {
    "channel": {
      "id": "live/p3000",
      "name": "NetFlow-port3000"
    }
  }
]
```

```
    },
    "values": [
      [ 1415059800000, 69226947.86664 ],
      [ 1415060100000, 66770998.45336 ],
      [ 1415060400000, 73834643.33336 ],
      ...
    ]
  },
  {
    "channel": {
      "id": "live/p2055",
      "name": "NetFlow-port2055"
    },
    "values": [
      [ 1415059800000, 0 ],
      [ 1415060100000, 0 ],
      [ 1415060400000, 0 ],
      ...
    ]
  }
]
```

6.2.5.2 Getting Multiple Profiles Chart Data Method

POST /rest/fmc/analysis/charts

- Returns array of chart values according to given array of search objects.
- @param search [AnalysisChartBatchSearch] (6.2.2)
- @returns 200 [AnalysisChartData] (6.2.2)
- @returns 204 No data
- @throws 405 ValidationResult in case search parameters are invalid

Example Usage

```
POST /rest/fmc/analysis/charts?search=[
  {
    "from": "2015-05-01 12:00",
    "to": "2015-05-02 12:00",
    "profile": "mail",
    "chart": {
```

```
        "measure": "packets",
        "protocol": 6
    },
    {
        "from": "2015-05-01 12:00",
        "to": "2015-05-02 12:00",
        "profile": "routers",
        "chart": {
            "measure": "packets",
            "protocol": 6
        }
    },
    {
        "from": "2015-05-01 12:00",
        "to": "2015-05-02 12:00",
        "profile": "live",
        "chart": {
            "measure": "packets",
            "protocol": 6
        }
    }
}
```

RETURNS:

```
[
  [
    {
      "channel": {
        "id": "mail/smtps",
        "name": "smtps"
      },
      "values": [
        [ 1430481900000, 0.29333 ],
        ...
      ]
    },
    {
      "channel": {
        "id": "mail/spop3",
        "name": "spop3"
      },
      "values": [
        [ 1430481900000, 0.65333 ],
        ...
      ]
    }
  ]
]
```

```
],
[
  {
    "channel":{
      "id":"routers/gre",
      "name":"gre"
    },
    "values":[
      [ 1430481900000, 0 ],
      ...
    ]
  }
],
[
  {
    "channel":{
      "id":"live/p3000-bb5174b",
      "name":"p3000"
    },
    "values":[
      [ 1430481900000, 4515.45 ],
      ...
    ]
  },
  {
    "channel":{
      "id":"live/zuzu-dba3c23c",
      "name":"zuzu"
    },
    "values":[
      [ 1430481900000, 0 ],
      ...
    ]
  },
  {
    "channel":{
      "id":"live/cucori-e2e527",
      "name":"cucorietka"
    },
    "values":[
      [ 1430481900000, 0 ],
      ...
    ]
  },
  {
    "channel":{
      "id":"live/p3000-d1c75f2",
      "name":"p3000"
    },
    "values":[
      [ 1430481900000, 0 ],
      ...
    ]
  }
]
```

```
    },
    "values": [
        [ 1430481900000, 0 ],
        ...
    ]
}
]
```

6.2.6 Flows Search Related Methods

6.2.6.1 Get Flows Order Bys

GET | POST /rest/fmc/analysis/flows/orderBys

- Returns all allowed order bys used for listing flows.
- @returns 200 [AnalysisFlowsOrderBy] (6.2.3)

Example Usage

```
GET /rest/fmc/analysis/flows/orderbys
```

RETURNS:

```
[
  {
    "id": "tstart",
    "name": "start time of flows"
  },
  {
    "id": "tend",
    "name": "end time of flows"
  },
  {
    "id": "nrtt",
    "name": "Round Trip Time Avg"
  },
  {
    "id": "nrttmin",
    "name": "Round Trip Time Min"
  },
  {
```



```
    "id": "nrttmax",
    "name": "Round Trip Time Max"
  },
  ...
]
```

6.2.6.2 Get Flows Listing

GET | POST /rest/fmc/analysis/flows

- Returns id of stored flows listing for given search object
- @param search FlowsListingAnalysisSearch (6.2.3)
- @param showonly Integer("Maximal count of rows to be returned. Default 10. Maximal value is 10000")
- @param output [NfColumn.id] (6.2.3) - array of columns to show
- @returns 200 [AsyncSearchResult] (6.2.3)

Example Usage With Aggregation By Source and Destination IPv4 Mask

POST /rest/fmc/analysis/flows

FORM-DATA

```
search={
  "from": "2014-11-01 10:00",
  "to": "2014-11-06 14:00",
  "profile": "mail",
  "channels": [ "pop3", "spop3", "smtp" ],
  "listing": {
    "aggregateBy": [ ],
    "sourceIpMask": {
      "ipVersion": 4,
      "mask": 24
    },
    "targetIpMask": {
      "ipVersion": 4,
      "mask": 16
    }
  },
  "filter": "any"
}
showonly=700
```

```
output=["ts","td","pr","sa","sp","da","dp","flg","pkt","byt","fl"]
```

RETURNS:

```
{
  "id": 797
}
```

Example Usage With Aggregation By Protocol And Source IP Sorted By Start Time

POST /rest/fmc/analysis/flows

FORM-DATA

```
search={
  "from": "2014-11-01 10:00",
  "to": "2014-11-06 14:00",
  "profile": "mail",
  "channels": [ "pop3", "spop3", "smtp" ],
  "listing": {
    "aggregateBy": [ "srcip", "proto" ],
    "sortBy": "tstart"
  },
  "filter": "any"
}
showonly=70
output=["ts","td","pr","sa","pr","pkt","byt","fl"]
```

RETURNS:

```
{
  "id": 804
}
```

6.2.7 Statistics Search Related

6.2.7.1 Get Statistics Limit Types

GET /rest/fmc/analysis/statistics/limitvaluetypes

- @returns zoo [LimitedValueType] (6.2.3)

Example Usage

GET /rest/fmc/analysis/statistics/limitvaluetypes

```
RETURNS:
[
  {
    "id": "packets",
    "name": "Packets"
  },
  {
    "id": "bytes",
    "name": "Bytes"
  }
]
```

6.2.7.2 Get Allowed Limit Operators

GET /rest/fmc/analysis/statistics/limitoperators

- @returns 200 [String]

Example Usage

```
GET /rest/fmc/analysis/statistics/limitoperators
```

```
RETURNS:
[
  "<",
  ">"
]
```

6.2.7.3 Get Statistics Methods

GET /rest/fmc/analysis/statistics/methods

- @returns 200 [AnalysisStatistic] (6.2.3)

Example Usage

```
GET /rest/fmc/analysis/statistics/methods
```

```
RETURNS:
```

```
[
  {
    "id": "record",
    "name": "IP Conversations",
    "group": "IP statistics"
  },
  {
    "id": "ip",
    "name": "Any IP address",
    "group": "IP statistics"
  },
  ...
]
```

6.2.7.4 Get Statistics Order By Values

GET /rest/fmc/analysis/statistics/orderbys

- Returns list of all allowed order bys.
- @returns 200 [AnalysisStatisticsOrderBy] (6.2.3)

Example Usage

GET /rest/fmc/analysis/statistics/orderbys

RETURNS:

```
[
  {
    "id": "bytes",
    "name": "bytes"
  },
  {
    "id": "packets",
    "name": "packets"
  },
  ...
]
```

6.2.7.5 Get Aggregation Fileds

GET /rest/fmc/analysis/aggregations

- Returns list of all allowed aggregations.
- @returns 200 [AnalysisAggregation] (6.2.3)

Example Usage

```
GET /rest/fmc/analysis/aggregations
```

RETURNS:

```
[
  {
    "id": "proto",
    "name": "IP protocol",
    "group": "IP statistics"
  },
  {
    "id": "srcip",
    "name": "Source IP address",
    "group": "IP statistics"
  },
  ...
]
```

6.2.7.6 Get Statistics

GET | POST /rest/fmc/analysis/statistics

- Returns statistics for given search object.
- @param search StatisticsAnalysisSearch (6.2.3)
- @param showonly Integer("Top-n limit") Default 10. Maximal value is 10000.
- @returns 200 [AsyncSearchResult] (6.2.3)

Example Usage For Statistic by IP Addresses Ordered By Flows

```
POST /rest/fmc/analysis/statistics
```

FORM-DATA

```
search={
  "from": "2014-11-05 00:00",
  "to": "2014-11-06 13:00",
  "profile": "mail",
  "channels": [ "smtp", "imaps" ],
```

```
    "statistics": {
      "statistics": "ip",
      "orderBy": "flows"
    },
    "filter": "any"
  }
  showonly=20
```

RETURNS:

```
{
  "id": 809
}
```

Example Usage For IP Conversations Aggregated By Source IP Address

POST /rest/fmc/analysis/statistics

FORM-DATA

```
search={
  "from": "2014-11-05 00:00",
  "to": "2014-11-06 13:00",
  "profile": "mail",
  "channels": [ "smtp", "imaps" ],
  "statistics": {
    "statistics": "record",
    "orderBy": "bytes"
  },
  "specific": {
    "forStatistics": "record",
    "bidirectional": 0,
    "aggregateBy": [ "srcip" ]
  },
  "filter": "any"
}
showonly=10
```

RETURNS:

```
{
  "id": 810
}
```

Example Usage For IP Conversation Aggregated By Protocol And IPv4 Masks

POST /rest/fmc/analysis/statistics

FORM-DATA

```
search={
  "from": "2014-11-05 00:00",
  "to": "2014-11-06 13:00",
  "profile": "mail",
  "channels": [ "smtp", "imaps" ],
```

```
"statistics": {
  "statistics": "record",
  "orderBy": "bytes"
},
"specific": {
  "forStatistics": "record",
  "bidirectional": 0,
  "aggregateBy": [ "proto"],
  "sourceIpMask": {
    "ipVersion": 4,
    "mask": 24
  },
  "targetIpMask": {
    "ipVersion": 4,
    "mask": 16
  }
},
"filter": "any"
}
showonly=10
```

RETURNS:

```
{
  "id": 810
}
```

6.2.8 Asynchronous Results

6.2.8.1 Get All Results

GET /rest/fmc/analysis/results

- Returns list of latest async results (available to the user)
- @returns 200 [AsyncResult] (6.2.3)

Example Usage For Statistic by IP Addresses Ordered By Flows

```
GET /rest/fmc/analysis/results
```

RETURNS:

```
[
```

```
{
  "id": 813,
  "name": "2014-11-06 13:50:55",
  "state": 1,
  "search": {
    "from": "2014-11-05 00:00",
    "to": "2014-11-06 13:00",
    "profile": "mail",
    "channels": [
      "smtp",
      "imaps"
    ],
    "statistics": {
      "statistics": "record",
      "orderBy": "bytes"
    },
    "specific": {
      "forStatistics": "record",
      "bidirectional": 0,
      "aggregateBy": [
        "proto"
      ],
      "sourceIpMask": {
        "ipVersion": 4,
        "mask": 24
      },
      "targetIpMask": {
        "ipVersion": 4,
        "mask": 16
      }
    },
    "filter": "any"
  }
},
...
]
```

6.2.8.2 Get Result Data

GET /rest/fmc/analysis/results/<id>

- Returns result data of specific query
- @param id AsyncSearchResult.id Id of requested result

- @returns 202 ResultNotCompletedYet (6.2.3) - in case of still computing result
- @returns 200 [AnalysisTableData] (6.2.3)

Example Usage

```
GET /rest/fmc/analysis/results/811
```

RETURNS:

```
[
  {
    "ts": "2014-11-06 13:35:11.885",
    "td": "554.458",
    "srcip": "192.168.9.2",
    "pkt": "995028",
    "byt": "1.3 G",
    "fl": "1091"
  },
  {
    "ts": "2014-11-06 13:36:34.796",
    "td": "460.223",
    "srcip": "192.168.9.101",
    "pkt": "204223",
    "byt": "264.0 M",
    "fl": "24"
  },
  ...
]
```

6.3 Reports

6.3.1 Data Types

These data types are available at Invea OS version 7.2.0 and newer.

Asterisk sign [?] means that this field is optional.

6.3.1.1 Chapter And Related Data Types

Blacklisted IPs Object

```
Blacklist = {  
  ?id: Int,  
  name: String("Human readable name"),  
  records: Array(String("IP address"))  
}
```

Bussines Hours Object Definition

```
BusinessHours = {  
  ?id: Integer,  
  label: String  
  intervals: [WorkingHoursInterval],  
  days: WorkingDays  
}  
  
WorkingHoursInterval = {  
  from: Int  
  to: Int  
}  
  
WorkingDays: {  
  monday: Boolean  
  tuesday: Boolean  
  wednesday: Boolean  
  thursday: Boolean  
  friday: Boolean  
  saturday: Boolean  
  sunday: Boolean  
}
```

Localization Object

```
LocalizedValues = {  
  cz: String("Czech language value"),  
  en: String("English language value"),  
  jp: String("Japanese language value"),  
}
```

Chapter Base Definition

```
ChapterBase = {  
  id: Integer,  
  // chapter name in current user's language localization -> read only parameter  
  // kept for backward compatibility reason  
  ?name: String,  
  // chapter name in current user's language localization -> read only parameter  
  // kept for backward compatibility reason  
  ?description: String("Chapter description"),  
  type: String << ["top", "traffic"]  
  localization: {  
    name: LocalizedValues,  
    description: LocalizedValues  
  },  
  profile: Profile.id // values at GET /rest/fmc/profiles  
  channels: "*" | [Channel.id] // values at GET /rest/fmc/profiles  
}
```

Top Chapter Definition

```
TopChapter extends ChapterBase = {  
  top: Integer("Top-n limit"),  
  statsBy: AnalysisStatistic.id,  
  // values at GET /rest/fmc/analysis/statistics/methods  
  
  orderBy: AnalysisStatisticsOrderBy.id,  
  // value at GET /rest/fmc/analysis/statistics/orderbys  
  
  ?specific: {  
    bidirectional: 0, // 0 as false ... 1 as true  
    ?aggregateBy: [AnalysisAggregation.Id],  
    // values at GET /rest/fmc/analysis/aggregations  
  
    ?sourceIpMask: IpMask,  
    ?targetIpMask: IpMask  
  },  
  
  // wheter is chapter disabled or enabled
```

```
status: Integer(Boolean),

// read only value - ingored when used for adding or modifying
outputColumns: Array(NfColumn.id), "Array of all column ids of the chapter."),

// set value columns between statistics and order by column, max. 5 columns
// values at GET /rest/fmc/analysis/statistics/orderbys
middleOutputColumns: Array(AnalysisStatisticsOrderBy.id),

?filter: String("BPF format filter"),

?blacklist: Int(Blacklist.id),
// values at GET /rest/fmc/blacklists

// write only value
?recompute: Int("Number of days to recompute")
}
```

Traffic Chapter Definition

```
TrafficChapter = {
  // read only value
  channelColumnName: String("Name of the first (channel) column."),

  // value according which will be channels sorted
  channelsOrderBy: String << [ "ibyt", "bps", "mbps" ],

  // Array of all columns (including 'channel' column) -> read only value
  // values at GET /rest/fmc/analysis/nfcolumns
  outputColumns: Array(String(NfColumn.id)),

  localization: {
    name: LocalizedValues,
    description: LocalizedValues,
    channelColumnName: LocalizedValues
  }
  showPercentile: Boolean // wether to show or not 95 percentil value
}
```

Chapter = TopChapter or TrafficChapter

6.3.1.2 Getting Chapters Data Types

Chapter Data Search Object

```
ChapterDataSearch = {
  "from": String("Time representation in yyyy-mm-dd hh:mm' format"),
  "to": String("Time representation in yyyy-mm-dd hh:mm' format"),
  ?"businessHours": Integer(BusinessHours.id, "Optional business hours id.")
}
```

Top Chapter Search Result Object

```
TopChapterDataRecord = {
  {NfColumnType.id}: String("Column value"),
  ...
}
```

Top Chapter Summary Object

```
SummaryRecord: {
  type: String() << ["total", "blacklisted"],
  values: Array({
    {NfColumnType.id}: String("Field summary value"),
    ...
  })
}
```

Top Chapter Data Object

```
TopChapterData = {
  data: Array(TopChapterDataRecord, "Array of data records"),
  summary: Array(SummaryRecord, "Summaries")
}
```

Traffic Chapter Search Channels Results

```
ChartDataRecordsFull = [
  Integer("Unix timestamp"),
  Number("Bites per second"),
  Number("Packets per second"),
  Number("Flows per second")
]
```

Traffic Chapter Search Channels Results

```
ChannelsChartData = {
  "channel": String("Name of the channel"),
  "values": Array(ChartDataRecordsFull)
}
```

Traffic Chapter Search Results

```
TrafficChapterData = {  
  "chart": Array(ChannelsChartData)  
  "table": TopChapterData  
}
```

6.3.1.3 Report

Report Representation

```
Report = {  
  id: Integer,  
  // name in users language - read only value  
  ?name: String,  
  // location in users language - read only value  
  ?location: String,  
  // device in users language - read only values  
  ?device: String,  
  localization: {  
    name: LocalizedValues,  
    location: LocalizedValues,  
    device: LocalizedValues  
  },  
  owner: String("Username, owner of the report"),  
  isDefault: Integer(Boolean),  
  chapters: Array(ChapterRelation)  
  // values at GET /rest/fmc/chapters  
}
```

```
ChapterRelation = {  
  chapterId: Int, // id of chapter linked into report  
  ?trafficChart: { // is required only for traffic chapter  
    // display this chapter chart as stacked  
    stacked: Boolean,  
    // display this chapter chart with logarithmic y axis  
    logarithmic: Boolean  
  }  
}
```

Email Reports Representation

```
EmailReport = {  
  id: Int,  
  report: Report.id,  
  lang: String("Language identifier ... e.g. 'en' "),  
  period: {
```

```
id: String("day|week|month|custom"),  
    // required only if "custom" is selected  
?from: String("hh:mm"),  
    // required only if "custom" is selected  
?to: String("hh:mm")  
},  
?businessHours: BusinessHours.id,  
days: {  
    monday: Boolean,  
    tuesday: Boolean,  
    wednesday: Boolean,  
    thursday: Boolean,  
    friday: Boolean,  
    saturday: Boolean,  
    sunday: Boolean  
},  
active: Boolean  
}
```

6.3.2 Methods

6.3.2.1 Business Hours

6.3.2.1.1 Get List of All Business Hours

GET /rest/fmc/businessHours

- Returns list of all defined business hours definitions
- @params fields [String("BH key")] - if not mentioned, all keys will be delivered
- @returns 204 No data "No business hours defined."
- @returns 200 [BusinessHours] (6.3.1.1)

Example Usage

GET /rest/fmc/businessHours

RETURNS:

```
[  
  {  
    "id": 1,
```

```
[{"label": "test",
  "intervals": [
    {
      "from": 7,
      "to": 11
    },
    {
      "from": 12,
      "to": 17
    }
  ],
  "days": {
    "sunday": false,
    "monday": true,
    "tuesday": true,
    "wednesday": true,
    "thursday": true,
    "friday": true,
    "saturday": false
  }
},
...
]
```

6.3.2.1.2 Get One Business Hours Detail

GET /rest/fmc/businessHours/id

- Returns specified business hours definition
- @params fields [String("BH key")] - if not mentioned, all keys will be delivered
- @returns 204 No data "Specified business hours not found."
- @returns 200 BusinessHours (6.3.1.1)

Example Usage

GET /rest/fmc/businessHours/2

RETURNS:

```
{
  "id": 2,
  "label": "test2",
```



```
"intervals":[
  {
    "from":7,
    "to":11
  },
  {
    "from":12,
    "to":17
  }
],
"days":{
  "sunday":false,
  "monday":true,
  "tuesday":true,
  "wednesday":true,
  "thursday":true,
  "friday":true,
  "saturday":false
}
}
```

6.3.2.1.3 Create New Bussines Hours

POST /rest/fmc/businessHours?entity=BusinessHours

- Creates new bussines hours instance
- @params entity BusinessHours (6.3.1.1)
- @returns 200 BusinessHours (6.3.1.1)
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
POST /rest/fmc/businessHours/?entity={
  "label":"new BH",
  "intervals":[
    {
      "from":7,
      "to":11
    },
    {
      "from":12,
      "to":17
    }
  ],
  "days":{
    "sunday":false,
    "monday":true,
    "tuesday":true,
    "wednesday":true,
    "thursday":true,
    "friday":true,
    "saturday":false
  }
}
```

```
{
  "from":12,
  "to":17
},
],
"days":{
  "sunday":false,
  "monday":true,
  "tuesday":true,
  "wednesday":true,
  "thursday":true,
  "friday":true,
  "saturday":false
}
}
RETURNS:
{
  "id":4,
  "label":"new BH",
  "intervals":[
    {
      "from":7,
      "to":11
    },
    {
      "from":12,
      "to":17
    }
  ],
  "days":{
    "sunday":false,
    "monday":true,
    "tuesday":true,
    "wednesday":true,
    "thursday":true,
    "friday":true,
    "saturday":false
  }
}
```

6.3.2.1.4 Modify Existing Bussines Hours

PUT /rest/fmc/businessHours?entity=BusinessHours

- Modify existing bussines hours instance
- @params entity BusinessHours (6.3.1.1)
- @returns 200 BusinessHours (6.3.1.1)
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
PUT /rest/fmc/businessHours/?entity={
  "id": 2,
  "label": "new BH name",
  "intervals": [
    {
      "from": 6,
      "to": 10
    },
    {
      "from": 12,
      "to": 18
    }
  ],
  "days": {
    "sunday": true,
    "monday": false,
    "tuesday": false,
    "wednesday": false,
    "thursday": false,
    "friday": false,
    "saturday": true
  }
}
RETURNS:
{
  "id": 2,
  "label": "new BH name",
  "intervals": [
    {
      "from": 6,
      "to": 10
    },
    {
      "from": 12,
      "to": 18
    }
  ],
}
```

```
"days":{
  "sunday":true,
  "monday":false,
  "tuesday":false,
  "wednesday":false,
  "thursday":false,
  "friday":false,
  "saturday":true
}
```

6.3.2.1.5 Delete Existing Bussines Hours

DELETE /rest/fmc/businessHours/<id>|<id>,<id>,... - single id, or list of ids separated by ,

- Removes Bussines Hours
- @params id BussinesHour.Id
- @returns 204
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
DELETE /rest/fmc/businessHours/6
```

```
Returns 204
```

6.3.2.2 Blacklists

6.3.2.2.1 Get Blacklists

GET /rest/fmc/blacklists

- Returns list of all defined blacklists

- @params fields [String("Blacklist key")] - if not mentioned, all keys will be delivered
- @returns 204 No data "No blacklists defined."
- @returns 200 200 [Blacklist] (6.3.1.1)

Example Usage

```
GET /rest/fmc/blacklists
```

RETURNS:

```
[
  {
    "id": 1,
    "name": "lol",
    "records": [
      "192.168.3.0"
    ]
  },
  ...
]
```

6.3.2.2.2 Get One Blacklist Detail

GET /rest/fmc/blacklists/id

- Returns list of all defined blacklists definitions
- @params fields [String("Blacklist key")] - if not mentioned, all keys will be delivered
- @returns 204 No data "No blacklists defined."
- @returns 200 Blacklist (6.3.1.1)

Example Usage

```
GET /rest/fmc/blacklists/2
```

RETURNS:

```
{
  "id": 2,
  "name": "secret people",
  "records": [ "192.168.3.111" ]
}
```

6.3.2.2.3 Create New Blacklist

POST /rest/fmc/blacklists/?entity=Blacklists - creates new blacklist

- Creates new blacklist instance
- @params entity Blacklist (6.3.1.1)
- @returns 200 Blacklist (6.3.1.1)
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
POST /rest/fmc/blacklists/?entity={
  "name": "Department A",
  "records": ["192.168.3.110", "192.168.3.111"]
}
RETURNS:
{
  "id": 3,
  "name": "Department A",
  "records": [ "192.168.3.110", "192.168.3.111" ]
}
```

6.3.2.2.4 Modify Existing Blacklists

PUT /rest/fmc/blacklist?entity=Blacklist - modify new existing blacklist

- Modify existing blacklist instance
- @params entity Blacklist (6.3.1.1)
- @returns 200 Blacklist (6.3.1.1)
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
PUT /rest/fmc/blacklist/?entity={
  "id": 3,
  "name": "new name" ,
  "records": [ "192.168.3.110", "192.168.3.112" ]
}

RETURNS:
{
  "id": 3,
  "name": "new name",
  "records": [ "192.168.3.110", "192.168.3.112" ]
}
```

6.3.2.2.5 Delete Existing Blacklist

DELETE /rest/fmc/blacklists/<id>|<id>,<id>,... - single id, or list of ids separated by ,

- Removes Blacklist (6.3.1.1)
- @params id Blacklist.Id (6.3.1.1)
- @returns 204
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
DELETE /rest/fmc/blacklists/3
```

Returns 204

6.3.2.2.6 Get Dependencies

Returns Dependencies object

GET /rest/fmc/dependencies/blacklists/Blacklist.id

- @returns 200 Dependencies for delete action
- @throws 401 header("X-FM-Message": "No permission to manipulate blacklist")
- @throws 404 header("X-FM-Message": "Blacklist does not exist")

Example Usage

```
GET /rest/dependencies/blacklists/1
```

Returns 200

```
{
  "action": "delete",
  "items": [
    {
      "id": 1
    }
  ],
  "dependent": [
    {
      "action": "update",
      "type": "chapter",
      "items": [
        {
          "id": 3033
        }
      ]
    }
  ]
}
```

6.3.2.3 Chapters

6.3.2.3.1 Get Chapters

GET /rest/fmc/chapters

- Returns list of all defined chapters
- @params fields [String("Chapter key")] - if not mentioned, all keys will be delivered
- @returns 204 No data "No chapters defined."

- @returns 200 200 [Chapter] (6.3.1.1)

Example Usage

GET /rest/fmc/chapters

RETURNS:

```
[
  {
    "id": 1,
    "name": "Hosts with Top Data Transfers",
    "description": "This chapter summarizes ...",
    "localization": {
      "name": {
        "cz": "Stanice s nejvetsimi datovymi prenosy",
        "jp": "",
        "en": "Hosts with Top Data Transfers"
      },
      "description": {
        "cz": "Tato kapitola shrnuje ... ",
        "jp": "",
        "en": "This chapter summarizes ..."
      }
    },
    "type": "top",
    "profile": "live",
    "channels": [ "p3000-bb5174b", "p3000-d1c75f2", "cucori-e2e527", "zuzu-dba3c23c" ],
    "top": 10,
    "status": 1,
    "statsBy": "ip",
    "orderBy": "bytes",
    "middleOutputColumns": [ ],
    "outputColumns": [ "ip", "bytes" ]
  },
  ...
]
```

6.3.2.3.2 Get One Chapter Detail

GET /rest/fmc/chapters/id

- Returns list of all defined chapter definitions
- @params fields [String("Chapter key")] - if not mentioned, all keys will be delivered

- @returns 204 No data "No chapters defined."
- @returns 200 Chapter (6.3.1.1)

Example Usage

GET /rest/fmc/chapters/2

RETURNS:

```
{
  "id": 2,
  "name": "Hosts with Top Download Transfers in the Network",
  "description": "This chapter summarizes ...",
  "localization": {
    "name": {
      "cz": "Stanice s~nejvetsim objemem stazenych dat v~siti",
      "jp": "",
      "en": "Hosts with Top Download Transfers in the Network"
    },
    "description": {
      "cz": "Tato kapitola shrnuje ...",
      "jp": "",
      "en": "This chapter summarizes ..."
    }
  },
  "type": "top",
  "profile": "live",
  "channels": [ "p3000-bb5174b", "p3000-d1c75f2", "cucori-e2e527", "zuzu-dba3c23c" ],
  "top": 10,
  "status": 1,
  "statsBy": "dstip",
  "orderBy": "bytes",
  "middleOutputColumns": [ "bps" ],
  "outputColumns": [ "da", "bps", "bytes" ]
}
```

6.3.2.3.3 Create New Chapters

POST /rest/fmc/chapters/?entity=Chapter - creates new chapter

- Creates new chapters instance
- @params entity Chapter (6.3.1.1)

- @returns 200 Chapter (6.3.1.1)
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
POST /rest/fmc/chapters/?entity={
  "localization": {
    "name": {
      "cz": "---",
      "jp": "---",
      "en": "My New Chapter"
    },
    "description": {
      "cz": "---",
      "jp": "---",
      "en": "My description"
    }
  },
  "type": "top",
  "profile": "live",
  "channels": [ "p3000-bb5174b", "p3000-d1c75f2", "cucori-e2e527", "zuzu-dba3c23c" ],
  "top": 10,
  "status": 1,
  "statsBy": "dstip",
  "orderBy": "bytes",
  "middleOutputColumns": [ "bps" ],
  "outputColumns": [ "da", "bps", "bytes" ]
}
RETURNS:
{
  "id": 2627,
  "name": "My New Chapter",
  "description": "My description",
  "localization": {
    "name": {
      "cz": "---",
      "jp": "---",
      "en": "My New Chapter"
    },
    "description": {
      "cz": "---",
      "jp": "---",
      "en": "My description"
    }
  },
  "type": "top",
```

```
"profile": "live",
"channels": [ "p3000-bb5174b", "p3000-d1c75f2", "cucori-e2e527", "zuzu-dba3c23c" ],
"top": 10,
"status": 1,
"statsBy": "dstip",
"orderBy": "bytes",
"middleOutputColumns": [ "bps" ],
"outputColumns": [ "da", "bps", "bytes" ]
}
```

6.3.2.3.4 Modify Existing Chapter

PUT /rest/fmc/chapters?entity=Chapter - modify new existing chapter

- Modify existing Chapter instance
- @params entity Chapter (6.3.1.1)
- @returns 200 Chapter (6.3.1.1)
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
PUT /rest/fmc/chapters/?entity={
  "id":2627,
  "localization":{
    "name":{
      "cz":"---",
      "jp":"---",
      "en":"My New Chapter NAME"
    },
    "description":{
      "cz":"---",
      "jp":"---",
      "en":"My description DESCRIPTION"
    }
  },
  "type":"top",
  "profile":"live",
  "channels":[
    "p3000-bb5174b",
```

```
        "p3000-d1c75f2",
        "cucori-e2e527",
        "zuzu-dba3c23c"
    ],
    "top":10,
    "status":1,
    "statsBy":"dstip",
    "orderBy":"bytes",
    "middleOutputColumns":[
        "bps"
    ],
    "outputColumns":[
        "da",
        "bps",
        "bytes"
    ]
}

RETURNS:
{
    "id":2627,
    "name":"My New Chapter NAME",
    "description":"My description DESCRIPTION",
    "localization":{
        "name":{
            "cz":"---",
            "jp":"---",
            "en":"My New Chapter NAME"
        },
        "description":{
            "cz":"---",
            "jp":"---",
            "en":"My description DESCRIPTION"
        }
    },
    "type":"top",
    "profile":"live",
    "channels":[
        "p3000-bb5174b",
        "p3000-d1c75f2",
        "cucori-e2e527",
        "zuzu-dba3c23c"
    ],
    "top":10,
    "status":0,
    "statsBy":"dstip",
    "orderBy":"bytes",
    "middleOutputColumns":[]
}
```

```
    "bps"  
  ],  
  "outputColumns": [  
    "da",  
    "bps",  
    "bytes"  
  ]  
}
```

6.3.2.3.5 Delete Existing Chapter

DELETE /rest/fmc/chapters/<id>|<id>,<id>,... - single id, or list of ids separated by ,

- Removes Chapter (6.3.1.1)
- @params id Chapter.Id (6.3.1.1)
- @returns 204
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
DELETE /rest/fmc/chapters/2627
```

```
Returns 204
```

6.3.2.3.6 Get Dependencies

Returns Dependencies object

GET /rest/fmc/dependencies/chapters/Chapter.id

- @returns 200 Dependencies for delete action
- @throws 401 header("X-FM-Message": "No permission to manipulate chapter")
- @throws 404 header("X-FM-Message": "Chapter does not exist")

Example Usage

GET /rest/fmc/dependencies/chapters/2626

Returns 200

```
{
  "action": "delete",
  "items": [
    {
      "id": 2626
    }
  ],
  "dependent": [
    {
      "action": "update",
      "type": "report",
      "items": [
        {
          "id": 5
        },
        {
          "id": 9
        },
        {
          "id": 3
        },
        {
          "id": 10
        }
      ]
    }
  ]
}
```

6.3.2.4 Reports

6.3.2.4.1 Get Reports

GET /rest/fmc/reports

- Returns list of all defined reports
- @params fields [String("Report key")] - if not mentioned, all keys will be delivered

- @returns 204 No data "No reports defined."
- @returns 200 200 [Report] (6.3.1.3)

Example Usage

```
GET /rest/fmc/reports
```

```
RETURNS:
```

```
[
  {
    "id":2,
    "name":"Default Report",
    "location":"localhost.localdomain",
    "device":"Invea FlowMon Collector VA",
    "localization":{
      "name":{
        "en":"Default Report",
        "cz":"Z\u00e1kladn\u00ed report",
        "jp":"\u30c7\u30d5\u30a9\u30eb\u30c8\u30ec\u30dd\u30fc\u30c8"
      },
      "location":{
        "en":"localhost.localdomain",
        "cz":"localhost.localdomain",
        "jp":"localhost.localdomain"
      },
      "device":{
        "en":"Invea FlowMon Collector VA",
        "cz":"Invea FlowMon Collector VA",
        "jp":"Invea FlowMon Collector VA"
      }
    },
    "owner":"admin",
    "isDefault":0,
    "chapters":[
      { "chapterId":1 },
      {
        "chapterId":271,
        "trafficChart":{
          "stacked":false,
          "logarithmic":false
        }
      },
      { "chapterId":2 },
      { "chapterId":18 },
      { "chapterId":6 },
      { "chapterId":5 },
      { "chapterId":8 }
    ]
  }
]
```



```
},...  
]
```

6.3.2.4.2 Get One Report Detail

GET /rest/fmc/reports/id

- Returns specified report definition
- @params fields [String("Report key")] - if not mentioned, all keys will be delivered
- @returns 204 No data "No reports defined."
- @returns 200 Report (6.3.1.3)

Example Usage

```
GET /rest/fmc/reports/2
```

RETURNS:

```
{  
  "id": "2",  
  "name": "Default Report",  
  "location": "localhost.localdomain",  
  "device": "Invea FlowMon Collector VA",  
  "localization": {  
    "name": {  
      "en": "Default Report",  
      "cz": "Z\u00e1kladn\u00ed report",  
      "jp": "\u30c7\u30d5\u30a9\u30eb\u30c8\u30ec\u30dd\u30fc\u30c8"  
    },  
    "location": {  
      "en": "localhost.localdomain",  
      "cz": "localhost.localdomain",  
      "jp": "localhost.localdomain"  
    },  
    "device": {  
      "en": "Invea FlowMon Collector VA",  
      "cz": "Invea FlowMon Collector VA",  
      "jp": "Invea FlowMon Collector VA"  
    }  
  },  
  "owner": "admin",  
  "isDefault": 0,  
}
```

```
"chapters": [
  { "chapterId":1 },
  {
    "chapterId":271,
    "trafficChart":{
      "stacked":false,
      "logarithmic":false
    }
  },
  { "chapterId":2 },
  { "chapterId":18 },
  { "chapterId":6 },
  { "chapterId":5 },
  { "chapterId":8 }
]
```

6.3.2.4.3 Create New Reports

POST /rest/fmc/reports/?entity=Report - creates new report

- Creates new Report instance
- @params entity Report (6.3.1.3)
- @returns 200 Report (6.3.1.3)
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
POST /rest/fmc/reports/?entity={
  "name": "aaa",
  "location": "localhost.localdomain - en",
  "device": "Invea FlowMon Collector VA - en",
  "localization": {
    "name": {
      "en": "aaa",
      "cz": "",
      "jp": ""
    },
    "location": {
```

```
    "en":"localhost.localdomain - en",
    "cz":"localhost.localdomain",
    "jp":"localhost.localdomain - jp"
  },
  "device":{
    "en":"Invea FlowMon Collector VA - en",
    "cz":"Invea FlowMon Collector VA",
    "jp":"Invea FlowMon Collector VA - jp"
  }
},
"owner":"admin",
"isDefault":0,
"chapters":[
  {
    "chapterId":1088
  }
]
}
```

RETURNS:

```
{
  "id":"11",
  "name":"aaa",
  "location":"localhost.localdomain - en",
  "device":"Invea FlowMon Collector VA - en",
  "localization":{
    "name":{
      "en":"aaa",
      "cz":"",
      "jp":""
    },
    "location":{
      "en":"localhost.localdomain - en",
      "cz":"localhost.localdomain",
      "jp":"localhost.localdomain - jp"
    },
    "device":{
      "en":"Invea FlowMon Collector VA - en",
      "cz":"Invea FlowMon Collector VA",
      "jp":"Invea FlowMon Collector VA - jp"
    }
  },
  "owner":"admin",
  "isDefault":0,
  "chapters":[
    {
      "chapterId":2626
    }
  ]
}
```

```
]
}
```

6.3.2.4.4 Modify Existing Report

PUT /rest/fmc/reports?entity=Chapter - modify new existing report

- Modify existing Report instance
- @params entity Report (6.3.1.3)
- @returns 200 Report (6.3.1.3)
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
PUT /rest/fmc/reports/?entity={
  "id": "2",
  "name": "Default Report",
  "location": "localhost.localdomain",
  "device": "Invea FlowMon Collector VA",
  "localization": {
    "name": {
      "en": "lol",
      "cz": "ooo",
      "jp": "ppp"
    },
    "location": {
      "en": "bbbb",
      "cz": "nnnnn",
      "jp": "mmmmmm"
    },
    "device": {
      "en": "xxxxx",
      "cz": "cccc",
      "jp": "vvvvv"
    }
  },
  "owner": "admin",
  "isDefault": 1,
  "chapters": [
```

```
{
  "chapterId":1
},
{
  "chapterId":271,
  "trafficChart":{
    "stacked":false,
    "logarithmic":false
  }
},
{
  "chapterId":2
},
{
  "chapterId":18
},
{
  "chapterId":6
},
{
  "chapterId":5
},
{
  "chapterId":8
}
]
}
```

RETURNS:

```
{
  "id":"2",
  "name":"lol",
  "location":"bbbb",
  "device":"xxxxx",
  "localization":{
    "name":{
      "en":"lol",
      "cz":"ooo",
      "jp":"ppp"
    },
    "location":{
      "en":"bbbb",
      "cz":"nnnnn",
      "jp":"mmmmmm"
    },
    "device":{
      "en":"xxxxx",
      "cz":"cccc",

```

```
        "jp": "vvvvv"
    },
    "owner": "admin",
    "isDefault": 1,
    "chapters": [
        {
            "chapterId": 1
        },
        {
            "chapterId": 271,
            "trafficChart": {
                "stacked": false,
                "logarithmic": false
            }
        },
        {
            "chapterId": 2
        },
        {
            "chapterId": 18
        },
        {
            "chapterId": 6
        },
        {
            "chapterId": 5
        },
        {
            "chapterId": 8
        }
    ]
}
```

6.3.2.4.5 Delete Existing Report

DELETE /rest/fmc/reports/<id>|<id>,<id>,... - single id, or list of ids separated by ,

- Removes Report (6.3.1.3)
- @params id Report.Id (6.3.1.3)
- @returns 204

- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
DELETE /rest/fmc/reports/2627
```

```
Returns 204
```

6.3.2.4.6 Get Dependencies

Returns Dependencies object

GET /rest/fmc/dependencies/reports/Report.id

- @returns 200 Dependencies for delete action
- @throws 401 header("X-FM-Message": "No permission to manipulate report")
- @throws 404 header("X-FM-Message": "Report does not exist")

Example Usage

```
GET /rest/dependencies/reports/2626
```

```
Returns 200
```

```
{
  "action": "delete",
  "items": [
    {
      "id": 3
    }
  ],
  "dependent": [
    {
      "action": "update",
      "type": "emailReports",
      "items": [
        3
      ]
    }
  ]
}
```

6.3.2.5 Email Reports

6.3.2.5.1 Get All Email Reports

GET /rest/fmc/emailreports

- Returns list of all defined email reports
- @params fields [String("EmailReport key")] - if not mentioned, all keys will be delivered
- @returns 204 No data "No EmailReport defined."
- @returns 200 200 [EmailReport] (6.3.1.3)

Example Usage

```
GET /rest/fmc/emailreports
```

```
RETURNS:
```

```
[
  {
    "id":2,
    "report":7,
    "lang":"cz",
    "period":{
      "id":"day"
    },
    "days":{
      "monday":true,
      "tuesday":true,
      "wednesday":true,
      "thursday":true,
      "friday":true,
      "saturday":true,
      "sunday":true
    },
    "active":true
  },...
]
```


6.3.2.5.2 Get One Email Report Detail

GET /rest/fmc/emailreports/id

- Returns specified email report definition
- @params fields [String("EmailReport key")] - if not mentioned, all keys will be delivered
- @returns 204 No data "No email reports defined."
- @returns 200 EmailReport (6.3.1.3)

Example Usage

```
GET /rest/fmc/emailreports/7
```

```
RETURNS:
```

```
{
  "id":7,
  "report":7,
  "lang":"cz",
  "period":{
    "id":"day"
  },
  "days":{
    "monday":true,
    "tuesday":true,
    "wednesday":true,
    "thursday":true,
    "friday":true,
    "saturday":true,
    "sunday":true
  },
  "active":false
}
```

6.3.2.5.3 Create New Email Report

POST /rest/fmc/emailreports/?entity=EmailReport - creates new report

- Creates new chapters instance

- @params entity EmailReport (6.3.1.3)
- @returns 200 EmailReport (6.3.1.3)
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

POST /rest/fmc/emailreports/?entity={

```
"report":7,
"lang":"cz",
"period":{"
  "id":"day"
},
"days":{"
  "monday":true,
  "tuesday":true,
  "wednesday":true,
  "thursday":true,
  "friday":true,
  "saturday":true,
  "sunday":true
},
"active":true
```

}

RETURNS:

```
{
  "id":10,
  "report":7,
  "lang":"cz",
  "period":{"
    "id":"day"
  },
  "days":{"
    "monday":true,
    "tuesday":true,
    "wednesday":true,
    "thursday":true,
    "friday":true,
    "saturday":true,
    "sunday":true
  },
  "active":true
```

}

6.3.2.5.4 Modify Existing Email Reports

PUT /rest/fmc/emailreports?entity=Chapter - modify new existing report

- Modify existing EmailReport instance (6.3.1.3)
- @params entity EmailReport (6.3.1.3)
- @returns 200 EmailReport (6.3.1.3)
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
PUT /rest/fmc/emailreports/?entity={
  "id":2,
  "report":7,
  "lang":"cz",
  "period":{
    "id":"day"
  },
  "days":{
    "monday":true,
    "tuesday":true,
    "wednesday":true,
    "thursday":true,
    "friday":true,
    "saturday":true,
    "sunday":true
  },
  "active":true
}
```

RETURNS:

```
{
  "id":2,
  "report":7,
  "lang":"cz",
  "period":{
    "id":"day"
  },
  "days":{
    "monday":true,
    "tuesday":true,
```

```
"wednesday":true,  
"thursday":true,  
"friday":true,  
"saturday":true,  
"sunday":true  
},  
"active":true  
}
```

6.3.2.5.5 Delete Existing Report

DELETE /rest/fmc/emailreports/<id> | <id>,<id>,... - single id, or list of ids separated by ,

- Removes Email Report
- @params id EmailReport.Id (6.3.1.3)
- @returns 204
- @returns 401 header("X-FM-Message": "Not allowed to perform this action")
- @returns 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
DELETE /rest/fmc/emailreports/4
```

Returns 204

6.3.2.6 Get Chapter Data

GET /rest/fmc/chapterData/id

- Returns chapter data for given chapter
- @param id Integer(Chapter.id) Id of a chapter to get data for.
- @param search ChapterDataSearch (6.3.1.2) - Search object specifying time range and business hours.

- @returns 204 No data "No data satisfying search conditions found."
- @returns 200 TopChapterData or TrafficChapterData (6.3.1.2 or 6.3.1.2)

Example of Getting Top Chapter Data

```
GET /rest/fmc/chapterData/20?search={
  "from": "2014-11-28 00:00",
  "to": "2014-11-28 12:00"
}

RETURNS:
{
  "data": [
    { "portp": "22", "proto": 0, "ibyt": 197552038 },
    { "portp": "33862", "proto": 0, "ibyt": 126833838 },
    { "portp": "44988", "proto": 0, "ibyt": 53064906 },
    { "portp": "443", "proto": 0, "ibyt": 40614793 },
    { "portp": "27530", "proto": 0, "ibyt": 33242330 },
    { "portp": "8182", "proto": 0, "ibyt": 33242330 },
    { "portp": "33860", "proto": 0, "ibyt": 6287814 },
    { "portp": "25", "proto": 0, "ibyt": 2424864 },
    { "portp": "49991", "proto": 0, "ibyt": 2424864 },
    { "portp": "38192", "proto": 0, "ibyt": 2378240 }
  ],
  "summary": {
    "type": "total",
    "values": { "portp": "", "proto": 0, "ibyt": 101644308943 }
  },
  "blacklisted": {
    "values": { "portp": "", "proto": 0, "ibyt": 101370468636 }
  }
}
```

Example of Getting Traffic Chapter Data

```
GET /rest/fmc/chapterData/15?search={
  "from": "2014-11-28 00:00",
  "to": "2014-11-28 12:00"
}

RETURNS:
{
  "table": {
    "data": [
      {
        "Service": "smb",
        "ibyt": 48399940,
        "bps": 8901.1383908046,
```

```
        "mbps": 35232.746666667
      },
      {
        "Service": "dhcp",
        "ibyt": 47970357,
        "bps": 8822.1346206897,
        "mbps": 12046.026666667
      },
      ...
    ],
    "summary": {
      "type": "total",
      "values": {
        "Service": "total",
        "ibyt": 120894926,
        "bps": 22233.552,
        "mbps": 84828.533333333
      }
    }
  },
  "chart": [
    {
      "channel": "smb",
      "values": [
        [ 1417133100, 3599.14664, 4.28, 0.01333 ],
        [ 1417133400, 4167.33336, 4.59667, 0.23667 ],
        [ 1417133700, 3602.48, 4.27, 0.01667 ],
        ...
      ]
    },
    {
      "channel": "dhcp",
      "values": [
        [ 1417133100, 47.84, 0.02667, 0.02667 ],
        [ 1417133400, 47.84, 0.02667, 0.02667 ],
        [ 1417133700, 47.84, 0.02667, 0.02667 ],
      ]
    },
    ...
  ]
}
```

6.4 Sources

6.4.1 Data Types

Following data structures are used in query parameters or results.

Asterisk sign [?] means that this field is optional.

Source Object

```
Source = {  
  id: Integer,  
  ip: IPv4, // netflow source IP  
  name: String("Source"), // custom user name  
  hostname: String("Source"), // source hostname readed via SNMP  
  ?profile: Profile.id, // if source is profiled, contains profile id.  
}
```

6.4.2 Methods

6.4.2.1 Get Sources List

GET /rest/fmc/sources

- Returns array of sources objects visible to the user
- @params fields [String("Source or channel key")] - if not mentioned, all key will be delivered
- @returns 200 [Source]

Example Usage

```
GET /rest/fmc/sources
```

RETURNS:

```
[  
  {  
    "id": 490,  
    "ip": "127.0.0.1",  
    "name": "localhost (127.0.0.1)",
```

```
"hostname": "localhost",  
  "profile": "Sources/localh-62c967"  
}, ...  
]
```


6.5 Active Devices

6.5.1 Data Types

Following data structures are used in query parameters or results.

Asterisk sign [?] means that this field is optional.

Active Devices Search Object Definition

```
MacAddressFull = {
  mac: String(Mac),
  vendor: String("Vendor name")
}

ActiveDeviceSearch = {
  from: String(Timestamp),
  to: String(Timestamp),
  ?device: String("Device identified by IPv4, IPv6 or MAC address"),
  ?mergeGap: Integer("Maximal gap (minutes) between flows to merge them."),
  ?listeningPorts: Array(String(Channel.id), "Ids of chanel of live profile."),
  ?vlan: Integer("VLAN number"),
  ?showRouters: Integer(Boolean),
  ?aggregateByIps: Integer(Boolean),
  ?aggregateByMac: Integer(Boolean)
}
```

Active Device Object

```
ActiveDeviceRecord = {
  first: String(Timestamp, "Time when device was seen for the first time."),
  last: String(Timestamp, "Time when device was seen for the last time."),
  ?ip: IpAddressFull,
  ?mac: macAddressFull,
  vlan: Integer("VLAN number"),
  listeningPort: Channel
}
```

Active Devices Response Data

```
ActiveDevicesData = {
  data: Array(ActiveDeviceRecord),
  summary: [
    {
```

```
    type: "total",
    values: {
      uniqueIp4: Integer("Total count of all unique IPv4 addresses in the network."),
      uniqueIp6: Integer("Total count of all unique IPv6 addresses in the network."),
      uniqueMac: Integer("Count of all unique mac addresses in the network.")
    }
  }
}
```

6.5.2 Methods

6.5.2.1 Get Active Devices Data

GET /fmc/activeDevices

- Returns active devices according to given search object
- @param search ActiveDeviceSearch (6.5.1)
- @param limit Integer Default 100
- @throws 405 ValidationResult (??) - in case search parameters are invalid.
- @returns 204 No data "No data satisfiing search conditions found."
- @returns 200 ActiveDevicesData (6.5.1)

Example Usage

```
GET /fmc/activeDevices?limit=500&search={
  "from": "2015-06-01 00:00",
  "to": "2015-07-01 14:00",
  "device": "0/0",
  "mergeGap": 15,
  "listeningPorts": ["p3000"],
  "showRouters": 1,
  "aggregateByIps": 0,
  "aggregateByMac": 1
}
```

RETURNS:

```
[
  {
    "first": "2015-06-30 09:39",
    "last": "2015-06-30 12:59",
    "ip": "10.0.0.0",
    "vlan": "0",
    "listeningPort": "p3000"
  },
  {
    "first": "2015-06-30 08:53",
    "last": "2015-07-01 08:38",
    "ip": "10.0.0.2",
    "vlan": "0",
    "listeningPort": "p3000"
  }, ...
]
```

7 FlowMon Dashboard

7.1 Flowmon Dashboard

Asterisk sign [?] means that this field is optional.

7.1.1 Dashboard Data Types

Dashboard Object

```
Dashboard = {  
  id: Integer,  
  name: String,  
  description: String,  
  order: Integer,  
  isDefault: Bool,  
  user: DashboardUser("User that has created the dashboard.")  
}
```

DashboardUser Object

```
DashboardUser = {  
  id: Integer,  
  login: String,  
  email: String  
}
```

Widget Search Object

```
WidgetSearch = {  
  dashboard: Dashboard.id  
}
```

Widget Object

```
Widget = {  
  id: Integer,  
  row: Integer, // widget location row  
  col: Integer, // widget location column  
  options: JsonObject, // you can ignore this so far  
  dashboard: Dashboard.id // in which dashboard is this widget located  
}
```

WidgetData Object

```
// object containing data to show to the user  
WidgetData = {  
  name: String("Widget name"), // widget name in users language
```

```
?table:{
  header:[
    String("Header column title"),
    ...
  ],
  rows:[
    [
      String|Integer
    ],
    ...
  ],
  ?summaries:[
    [
      String("Summary name"),
      Float|Integer|String(""),
      ...
    ],
    // "table.summaries" columns count must match "table.rows" column count
    ...
  ],
  ?rowsColors: [
    String("Hex encoded color ... e.g. #45aa54 "),
    ...
  ],
  ?summariesColors: [
    String("Hex encoded color ... e.g. #45aa54 "),
    ...
  ]
},
?piechart:{
  values:[
    {
      name:String("For example IP address,
      port number,
      ADS Event etc ..."),
      value:Integer|float,
      color:String("Hex encoded color ... e.g. #45aa54 ")
    },
    ...
  ]
},
?summaryPiechart:{
  values:[
    {
      name:String("Summary value name"),
      value:Integer|float,
      color:String("Hex encoded color ... e.g. #45aa54 ")
    },
    ...
  ]
}
```

```
...
]
},
?trafficChart:{
  lines:[
    {
      name:String("Line name"),
      values:[
        [
          Integer("Timestamp in seconds"),
          Integer|Float
        ],
        ...
      ],
      color:String("Hex encoded color ... e.g. #45aa54 ")
    },
    ...
  ],
  metricLabel: String("e.g. bit/s") // label for Y axis
}
}
```

7.1.2 Dashboard Methods

7.1.2.1 Get Dashboards Method

GET /rest/fmd/overview/dashboards

- Returns list of current user dashboards
- @returns 204 No data
- @returns 200 [Dashboard] (7.1.1)

Example Usage

GET /rest/fmd/overview/dashboards/

RETURNS:

```
[
  {
    "id": 1,
    "name": "Default",
    "description": "Default dashboard",
```

```
[
  {
    "order": 1,
    "user": {
      "id": 1,
      "login": "admin",
      "email": "admin@example.com"
    }
  }
]
```

7.1.3 Dashboard Widget Methods

7.1.3.1 Get Dashboard Widgets Method

GET /rest/fmd/overview/widgets/find

- Returns list of widgets for given dashboard. If no parameter **search** given all widgets will be returned. If admin role is logged, widgets of all users will be returned.
- @param search WidgetSearch (7.1.1)
- @returns 204 No data
- @returns 200 [Widget] (7.1.1)

Example Usage

```
GET /rest/fmd/overview/widgets/find?search={"dashboard": 1}
```

RETURNS:

```
[
  {
    "id": 2,
    "row": 0,
    "col": 0,
    "dashboard": {
      "id": 1,
      "name": "Default",
      "user": {
        "id": 1,
        "login": "admin",
        "email": "admin@example.com"
      }
    }
  }
]
```



```
    },  
    "options": {...}  
  }  
},  
...  
]
```

7.1.4 Get Widget Data Method

7.1.4.1 Get Widget Data Method

GET /rest/fmd/overview/widget/data/<Widget.id>

- Returns data to display for given widget
- @returns 204 No data
- @returns 200 [WidgetData] (7.1.1)

Example Usage

```
GET /rest/fmd/overview/widget/data/5
```

RETURNS:

```
{  
  "name": "Top Network Services over UDP",  
  "piechart": {  
    "values": [  
      {  
        "color": "#2465f4",  
        "name": "9996",  
        "value": "25068815468"  
      },  
      ...  
    ]  
  },  
  "summaryPiechart": {  
    "values": []  
  },  
  "table": {  
    "header": [  
      "Any Port",
```

```
    "IP Protocol",
    "Bytes"
  ],
  "rows": [
    [
      "9996",
      "UDP",
      "23.35 GiB"
    ],
    ...
  ],
  "rowsColors": [
    "#2465f4",
    "#f44200",
    "#ffa216",
    "#1cb500",
    "#b400bd",
    "#00ace5",
    "#fb397c",
    "#79c800",
    "#d82b21",
    "#306fb6"
  ],
  "summaries": [
    [
      "Blacklisted",
      "",
      0
    ],
    [
      "Total",
      "",
      "33.98 GiB"
    ]
  ],
  "summariesColors": [
    "#000000",
    "#000000"
  ]
}
```

8 System

8.1 System Messages

8.1.1 System Messages Types

Severity Object

```
MessageSeverity = {  
  id: Integer,  
  name: String("Severity name"),  
}
```

Message Object

```
Message = {  
  id: Integer,  
  time: String("YYYY-MM-DD HH:MM:SS"), "2016-02-29 14:53:09",  
  severity: Severity.id,  
  plugin: String("Plugin short"),  
  text: String("Message text"),  
  hasBeenRead: Boolean  
}
```

8.1.2 System Messages Methods

8.1.2.1 Get Message Severity Method

GET /rest/messages/severities

- Returns list of message severities
- @returns 204 No data
- @returns 200 [MessageSeverity] (8.1.1)

Example Usage

```
GET /rest/messages/severities
```

```
RETURNS:
```

```
[  
  {  
    "id": 7,  
    "name": "DEBUG"
```

```
},
{
  "id": 6,
  "name": "INFO"
},
{
  "id": 5,
  "name": "NOTICE"
},
{
  "id": 4,
  "name": "WARNING"
},
{
  "id": 3,
  "name": "ERROR"
},
{
  "id": 2,
  "name": "CRITICAL"
},
{
  "id": 1,
  "name": "ALERT"
},
{
  "id": 0,
  "name": "EMERGENCY"
}
]
```

8.1.2.2 Get Message Method

GET /rest/messages/find

- Returns list of messages
- @returns 204 No data
- @returns 200 [Message] (8.1.1)

Example Usage

```
GET /rest/messages/find
```

RETURNS:

```
[
  {
    "id": 2,
    "time": "2016-06-03 12:00:05",
    "severity": 4,
    "plugin": null,
    "text": "Your hard disk needs a~consistency check, please go to Configuration ...",
    "hasBeenRead": false
  },
  ...
]
```

9 Examples For Javascript In Web Page

This chapter show how to use REST API with javascript with jQuery in browser.

Javascript Code For Getting Authentication Token

```
var host = '192.168.51.145';
$.ajax({
  type: 'POST',
  url: "https://" + host + "/resources/oAuth/token",
  data: {
    "username" : 'admin',
    "password" : 'admin',
    "client_id" : 'invea-tech',
    "grant_type" : 'password'
  },
  success: function(msg) {
    alert(msg.access_token);
  },
  error: function(a, b, err) {
    console.log(err );
  },
});
```

Complete Example Of Getting Chart Data

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js" />
</script>
<title>REST API</title>

<script>
  $(document).ready(function() {
    var host = '192.168.51.145';

    $.ajax({
      type: 'POST',
      url: "https://" + host + "/resources/oAuth/token",
      data: {
        "username" : 'admin',
        "password" : 'admin',
        "client_id" : 'invea-tech',
        "grant_type" : 'password'
      },
      success: function(msg) {
        token = msg.access_token;
        getResults(msg.access_token);
      }
    });
  });
</script>
```

```
    },
    error: function(a, b, err) {
        console.log(err );
    },
});

function getResults(t) {
    $.ajax({
        type: 'GET',
        url: "https://" + host + "/rest/fmc/analysis/chart",
        headers: {
            "Authorization" : 'bearer_' + t
        },
        data: {
            search: JSON.stringify({
                "from": "2014-11-04_00:05",
                "to": "2014-11-04_14:20",
                "profile": "live",
                "chart": {
                    "measure": "traffic",
                    "protocol": 1
                }
            })
        },
        success: function(msg) {
            $('body').html(JSON.stringify(msg));
        },
        error: function(a, b, err) {
            console.log(err );
        },
    });
}

});
</script>
</head>
<body></body>
</html>
```


10 Examples For CURL

This chapter show how to use REST API with CURL command.

Command For Getting Authentication Token

```
$ curl https://<ip>/resources/oauth/token -k  
-d 'grant_type=password' -d 'client_id=invea-tech'  
-d 'username=admin' -d 'password=admin'
```

RETURNS:

```
{"access_token": "3a58010ee8a41acad77682d47365675112ceaff4a71a3d1ea58c05165948ec13",  
"expires_in": 604800, "token_type": "bearer", "refresh_token":  
"f8b97dc8f7517165002e1357c91bd5f171a5209f0fc28aefbef1c02a6f6cf5cb"}
```

Command For Getting Authentication Token

```
$ curl https://<ip>/rest/fmc/analysis/chart  
?search=%7B%20%20%20%20%20%22from%22%3A%20%222014-09-07%2023%3A05%22%2C%20  
%20%20%20%20%22to%22%3A%20%222014-09-08%2014%3A20%22%2C%20%20%20%20%20%22  
profile%22%3A%20%22live%22%2C%20%20%20%20%20%22chart%22%3A%20%7B%20%20%20  
%20%20%20%20%20%22measure%22%3A%20%22traffic%22%2C%20%20%20%20%20%20%  
20%20%20%22protocol%22%3A%201%20%20%20%20%7D%20%7D -k -H  
'Authorization: bearer 3a58010ee8a41acad77682d47365675112ceaff4a71a3d1ea58c05165948ec13'
```

RETURNS:

```
[{"channel": {"id": "live\\p9996", "name": "NetFlow-port9996"},  
"values": [[1410127800000, 0], [1410128100000, 0]  
, [1410128400000, 0], [1410128700000, 0], [1410129000000, 0],  
[1410129300000, 0], [1410129600000, 0]  
, [1410129900000, 0], [1410130200000, 0], [1410130500000, 0],  
[1410130800000, 0], [1410131100000, 0],  
[1410131400000, 0], [1410131700000, 0], [1410132000000, 0],  
...]
```

11 REST API changes

Version 8.1.1

- added Flowmon Dashboard methods
- added system messages reading methods
- modified ChartDataRecord array, where first item of couple is now epoch time in milliseconds at UTC

Contacts

INVEA-TECH a.s.
U Vodarny 2965/2
Brno 61600

Web: www.invea.com
Email: info@invea.com
Tel.: +420 511 205 251

Feedback

We would be pleased if you tell us your comments to this text (typing errors, incomplete or unclear information). Please, contact us via email support@invea.com.

Copyright

Except as stated here, none of the document may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of INVEA-TECH. Any unauthorized use of this specification may violate copyright laws, trademark laws, the laws of privacy and publicity, and communications regulations and statutes.

INVEA-TECH, the company logo, and other designated brands included herein are trademarks of INVEA-TECH a.s. All other trademarks are the property of their respective owners.

This product uses NfSen and NFDUMP software Copyright (c) 2004, SWITCH - Teleinformatikdienste fuer Lehre und Forschung.

This product uses compress library zlib Copyright (C) 1995-1998 Jean-loup Gailly and Mark Adler.

This product uses library libpcap Copyright (c) 1993, 1994, 1995, 1996, 1997, 1998 The Regents of the University of California and Copyright (c) 1999 - 2003 NetGroup, Politecnico di Torino (Italy).

This product includes sFlow(TM), freely available from <http://www.inmon.com/>.

Copyright (c) 2007 - 2014 INVEA-TECH a.s. All rights reserved.