



FlowMon REST API

Developer Guide

June 22, 2015

Contents

1	Introduction	5
2	Authorization	6
2.1	Obtaining authentication token	6
2.2	Getting Rest API information	6
2.2.1	Getting list of available methods	7
2.2.2	Getting current version of FlowMon	7
3	FlowMon Configuration Center	8
3.1	Overview	9
3.1.1	Data Types	9
3.1.2	Methods	11
3.1.2.1	Full Device Info	11
3.1.2.2	Basic Device Info	13
3.1.2.3	Get CPU Usage	13
3.1.2.4	Get RAM Usage	14
3.1.2.5	Get Data Disk Usage	14
3.1.2.6	Get System Disk Usage	15
3.1.2.7	Get Boot Disk Usage	15
3.1.2.8	Get Interfaces	16
3.1.2.9	Get Exporters	17
4	FlowMon Monitoring Center	19
4.1	Profiles	20
4.1.1	Data Types	20
4.1.2	Get All Profiles	21
4.1.3	Get Profile detail	23
4.1.4	Creating New Profile	24
4.1.5	Modify Existing Profile	26
4.1.6	Delete Existing Profile	28
4.1.7	Get Profile Dependencies	29
4.2	Analysis	31
4.2.1	General Data Types	31
4.2.2	Chart Data Types	32
4.2.3	Analysis Data Types	32
4.2.4	General Analysis Methods	36
4.2.4.1	Get Profile Channels	36
4.2.4.2	Get Protocols	37

4.2.4.3	Get Columns Types	38
4.2.4.4	Get Columns	39
4.2.4.5	Get All Stored Filters	39
4.2.4.6	Get List Flows Outputs Definitions	40
4.2.5	Getting Chart Data Methods	41
4.2.5.1	Getting Chart Data Method	41
4.2.5.2	Getting Multiple Profiles Chart Data Method	43
4.2.6	Flows Search Related Methods	46
4.2.6.1	Get Flows Order Bys	46
4.2.6.2	Get Flows Listing	46
4.2.7	Statistics Search Related	48
4.2.7.1	Get Statistics Limit Types	48
4.2.7.2	Get Allowed Limit Operators	48
4.2.7.3	Get Statistics Methods	49
4.2.7.4	Get Statistics Order By Values	49
4.2.7.5	Get Aggregation Fileds	50
4.2.7.6	Get Statistics	51
4.2.8	Asynchronous Results	53
4.2.8.1	Get All Results	53
4.2.8.2	Get Result Data	54
4.3	Reports	55
4.3.1	Data Types	55
4.3.1.1	Chapter And Related Data Types	55
4.3.1.2	Getting Chapters Data Types	57
4.3.1.3	Report	58
4.3.2	Methods	60
4.3.2.1	Business Hours	60
4.3.2.1.1	Get List of All Business Hours	60
4.3.2.1.2	Get One Business Hours Detail	61
4.3.2.1.3	Create New Bussines Hours	62
4.3.2.1.4	Modify Existing Bussines Hours	63
4.3.2.1.5	Delete Existing Bussines Hours	64
4.3.2.2	Blacklists	65
4.3.2.2.1	Get Blacklists	65
4.3.2.2.2	Get One Blacklist Detail	66
4.3.2.2.3	Create New Blacklist	66
4.3.2.2.4	Modify Existing Blacklists	67
4.3.2.2.5	Delete Existing Blacklist	67
4.3.2.2.6	Get Dependencies	68

4.3.2.3	Chapters	69
4.3.2.3.1	Get Chapters	69
4.3.2.3.2	Get One Chapter Detail	70
4.3.2.3.3	Create New Chapters	71
4.3.2.3.4	Modify Existing Chapter	72
4.3.2.3.5	Delete Existing Chapter	74
4.3.2.3.6	Get Dependencies	75
4.3.2.4	Reports	76
4.3.2.4.1	Get Reports	76
4.3.2.4.2	Get One Report Detail	77
4.3.2.4.3	Create New Reports	79
4.3.2.4.4	Modify Existing Report	80
4.3.2.4.5	Delete Existing Report	83
4.3.2.4.6	Get Dependencies	83
4.3.2.5	Email Reports	84
4.3.2.5.1	Get All Email Reports	84
4.3.2.5.2	Get One Email Report Detail	85
4.3.2.5.3	Create New Email Report	86
4.3.2.5.4	Modify Existing Email Reports	87
4.3.2.5.5	Delete Existing Report	88
4.3.2.6	Get Chapter Data	89
5	Examples For Javascript In Web Page	92
6	Examples For CURL	94

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". 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 FlowMon Configuration Center

3.1 Overview

3.1.1 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")  
}
```

3.1.2 Methods

3.1.2.1 Full Device Info

GET /rest/fcc/device

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

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
      }
    ]
  }
}
```

3.1.2.2 Basic Device Info

GET /rest/fcc/device/info

- Returns basic device info
- @returns 200 DeviceInfo

Example Usage

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

```
RETURNS:
```

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

3.1.2.3 Get CPU Usage

GET /rest/fcc/device/cpu

- Returns CPU load information.
- @returns 200 CpuLoad

Example Usage

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

```
RETURNS:
```

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

3.1.2.4 Get RAM Usage

GET /rest/fcc/device/ram

- Returns current RAM usage
- @returns 200 DiskUsage

Example Usage

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

```
RETURNS:
```

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

3.1.2.5 Get Data Disk Usage

GET /rest/fcc/device/datadisk

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

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  
    }  
  ]  
}
```

3.1.2.6 Get System Disk Usage

GET /rest/fcc/device/systemdisk

- Returns current system disk usage
- @returns 200 DiskUsage

Example Usage

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

RETURNS:

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

3.1.2.7 Get Boot Disk Usage

GET /rest/fcc/device/bootdisk

- Returns current boot disk usage
- @returns 200 DiskUsage

Example Usage

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

RETURNS:

```
{
```

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

3.1.2.8 Get Interfaces

GET /rest/fcc/interfaces

- Returns collection of all interfaces and its states
- @returns zoo [InterfaceInfo]

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
  }
]
```

3.1.2.9 Get Exporters

GET /rest/fcc/exporters

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

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
    }
]
```

4 FlowMon Monitoring Center

4.1 Profiles

4.1.1 Data Types

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

Report Representation

? - key is optional

```
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 identificator")  
  
    // 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]  
}  
  
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")

// is required only if parent profile is selected live
?sources: [Source.ip]

// array of parent profile channels
parent: [Channel.id]
filter: String("Filter")
}
```

4.1.2 Get All Profiles

GET /rest/fmc/profiles

- 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 (especially channels.filter and description) is time consuming operation. In case of large amount of profiles we recommend to specify only some fields (e.g. id and name) to get much faster response.
- @returns 204 No data "No profiles defined."
- @returns 200 [Profile]

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",
    "sources":[]
  },
  {
    "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",
    "sources":[]
  },
  {
    "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",
```

```
        "sources": [  
            ]  
        }  
    ]  
},  
...  
]
```

4.1.3 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]

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",
"sources": [ ]
},
{
  "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",
  "sources": [ ]
},
{
  "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",
  "sources": [ ]
}
]
}
```

4.1.4 Creating New Profile

POST /rest/fmc/profiles?entity=Profile

- Return newly created profile
- @params entity Profile
- @returns 201 Profile - 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-a4b2bd",
  "name": "my nice profile",
  "parent": "live",
  "description": "this is my profile",
  "start": "2015-05-23 12:10",
  "type": "real",
  "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",
    "sources": [ ]
  },
  {
    "id": "firstc-9dcd29",
    "name": "first channel",
    "sign": "+",
    "color": "#2466f4",
    "parent": [ "p3000-bb5174b" ],
    "filter": "any",
    "sources": [ ]
  }
]
```

4.1.5 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
- @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",
    "sources": [ ]
  },
  {
    "id": "firstc-9dcd29",
    "name": "first channel",
    "sign": "+",
    "color": "#2466f4",
    "parent": [ "p3000-bb5174b" ],
    "filter": "any",
    "sources": [ ]
  }
]
```

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",
```

```
    "sources": [ ]
  },
  {
    "id": "firstc-9dcd29",
    "name": "first channel",
    "sign": "+",
    "color": "#2466f4",
    "parent": [ "p3000-bb5174b" ],
    "filter": "any",
    "sources": [ ]
  }
]
```

4.1.6 Delete Existing Profile

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

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

Example of using

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

RETURNS:

Returns 204

Example of using

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

RETURNS:

Returns 204

4.1.7 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
- @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"
        }
      ]
    }
  ]
}
```

4.2 Analysis

Following data structures are used in query parameters or results. Asterisk sign [?] means that this field is optional.

4.2.1 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))  
}
```

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-o-0-1" this function returns "127-o-0-1".

IP Protocol Data Type

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

Base Search Object

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

4.2.2 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)  
    }  
} extends BasicAnalysisSearch
```

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);
```

4.2.3 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
```

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.)")  
}
```

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": {
  ?"aggregateBy": Array(String(AnalysisAggregation.id)), "Aggregate by")
  ?"sourceIpMask": IpMask,
  ?"targetIpMask": IpMask,
  ?"sortBy": String(AnalysisFlowsOrderBy.id)
},

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

Sortable List Flows Fields

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

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"),
}
```

Sortable Statistics Fields

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

Field Which Can Be Used For Aggregation

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

Statistic Limitation Object

```
LimitedValueType = {  
    "id": String,  
    "name": String  
} << LimitedValueTypes;  
  
LimitOperator = String("Allowed limit operator") << ["<", ">"];
```

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"),  
        "orderBy": String(AnalysisStatisticsOrderBy.id)  
    },  
  
    ?"specific": {  
        ?"bidirectional": Boolean,  
        ?"aggregateBy": Array(String(AnalysisAggregation.id), "Aggregate results by")  
        ?"sourceIpMask": IpMask,  
        ?"targetIpMask": IpMask  
    },  
  
    ?limit: {  
        "onColumn": String(LimitedValueType.id),  
        "operator": String(LimitOperator),  
        "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.\"}"  
}
```

4.2.4 General Analysis Methods

4.2.4.1 Get Profile Channels

GET /rest/fmc/profiles/channels

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

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/imap3",
    "name": "imap3"
  },
  {
    "id": "mail/smtp",
    "name": "smtp"
  }
]
```

4.2.4.2 Get Protocols

GET /rest/fmc/protocols

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

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"
}
]
```

4.2.4.3 Get Columns Types

GET /rest/fmc/columnTypes

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

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"
  },
  ...
]
```

4.2.4.4 Get Columns

GET /rest/fmc/analysis/nfcolumns

- Returns list of all defined and used net flow columns.
- @returns 200 [ColumnType]

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"
  },
  ...
]
```

4.2.4.5 Get All Stored Filters

GET /rest/fmc/analysis/filters

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

Example Usage

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

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

4.2.4.6 Get List Flows Outputs Definitions

GET /rest/fmc/analysis/outputdefinitions

- Returns all saved outputs
- @returns 200 [SavedOutput]

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"  
    ]  
},  
    ...  
]
```

4.2.5 Getting Chart Data Methods

4.2.5.1 Getting Chart Data Method

GET | POST /rest/fmc/analysis/chart

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

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 ],
      ...
    ]
  }
]
```

```
}  
]
```

4.2.5.2 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]
- @returns 200 [AnalysisChartData]
- @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 ],
      ...
    ]
  }
]
```

4.2.6 Flows Search Related Methods

4.2.6.1 Get Flows Order Bys

GET /rest/fmc/analysis/flows/orderBys

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

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"
  },
  ...
]
```

4.2.6.2 Get Flows Listing

GET|POST /rest/fmc/analysis/flows

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

Example Usage With Aggregation By Source and Destination IPv4 Mask

```
GET /rest/fmc/analysis/flows?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

```
GET /rest/fmc/analysis/flows?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
}
```

4.2.7 Statistics Search Related

4.2.7.1 Get Statistics Limit Types

GET /rest/fmc/analysis/statistics/limitvaluetypes

- @returns 200 [LimitedValueType]

Example Usage

```
GET /rest/fmc/analysis/statistics/limitvaluetypes

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

4.2.7.2 Get Allowed Limit Operators

GET /rest/fmc/analysis/statistics/limitoperators

- @returns 200 [String]

Example Usage

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

RETURNS:

```
[
  "<",
  ">"
]
```

4.2.7.3 Get Statistics Methods

GET /rest/fmc/analysis/statistics/methods

- @returns 200 [AnalysisStatistic]

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"
  },
  ...
]
```

4.2.7.4 Get Statistics Order By Values

GET /rest/fmc/analysis/statistics/orderbys

- Returns list of all allowed order bys.

- @returns 200 [AnalysisStatisticsOrderBy]

Example Usage

```
GET /rest/fmc/analysis/statistics/orderbys
```

RETURNS:

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

4.2.7.5 Get Aggregation Fileds

GET /rest/fmc/analysis/aggregations

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

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"
  },
  ...
]
```

4.2.7.6 Get Statistics

GET /rest/fmc/analysis/statistics

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

Example Usage For Statistic by IP Addresses Ordered By Flows

```
GET /rest/fmc/analysis/statistics?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

```
GET /rest/fmc/analysis/statistics?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

GET /rest/fmc/analysis/statistics?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
}
```

4.2.8 Asynchronous Results

4.2.8.1 Get All Results

GET /rest/fmc/analysis/results

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

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"  
}  
},  
...  
]
```

4.2.8.2 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 - in case of still computing result
- @returns 200 [AnalysisTableData]

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"  
  },  
  ...  
]
```

4.3 Reports

4.3.1 Data Types

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

4.3.1.1 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  
  channels: [Channel.id]  
}
```

Top Chapter Definition

```
TopChapter extends ChapterBase = {  
  top: Integer("Top-n limit"),  
  statsBy: AnalysisStatistic.id,  
  orderBy: AnalysisStatisticsOrderBy.id,  
  ?specific: {  
    bidirectional: 0, // 0 as false ... 1 as true  
    ?aggregateBy: [AnalysisAggregation.Id],  
    ?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  
  middleOutputColumns: Array(AnalysisStatisticsOrderBy.id),  
  ?filter: String("BPF format filter"),  
  ?blacklist: Int(Blacklist.id),  
  // 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  
  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

4.3.1.2 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 = {Object
  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(CharDataRecordsFull)
}
```

Traffic Chapter Search Results

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

4.3.1.3 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)
}

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
}
```

4.3.2 Methods

4.3.2.1 Business Hours

4.3.2.1.1 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]

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
    }
  }
]
```

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

4.3.2.1.2 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

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  
  }  
}
```

4.3.2.1.3 Create New Bussines Hours

POST /rest/fmc/businessHours?entity=BusinessHours

- Creates new bussines hours instance
- @params entity BusinessHours
- @returns 200 BusinessHours
- @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
}
```

}

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
  }
}
```

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

4.3.2.1.4 Modify Existing Bussines Hours

PUT /rest/fmc/businessHours?entity=BusinessHours

- Modify existing bussines hours instance
- @params entity BusinessHours
- @returns 200 BusinessHours
- @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
  }
}
```

4.3.2.1.5 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
```

4.3.2.2 Blacklists

4.3.2.2.1 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]

Example Usage

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

```
RETURNS:
```

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

4.3.2.2.2 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

Example Usage

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

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

4.3.2.2.3 Create New Blacklist

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

- Creates new blacklist instance
- @params entity Blacklist
- @returns 200 Blacklist
- @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" ]  
}
```

4.3.2.2.4 Modify Existing Blacklists

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

- Modify existing blacklist instance
- @params entity Blacklist
- @returns 200 Blacklist
- @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" ]  
}
```

4.3.2.2.5 Delete Existing Blacklist

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

- Removes Blacklist
- @params id Blacklist.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/blacklists/3
```

```
Returns 204
```

4.3.2.2.6 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
      }
    ]
  }
}
```

4.3.2.3 Chapters

4.3.2.3.1 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]

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" ]  
},  
...  
]
```

4.3.2.3.2 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

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 ..."  
    }  
  }  
}
```

```
    "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" ]
}
```

4.3.2.3.3 Create New Chapters

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

- Creates new chapters instance
- @params entity Chapter
- @returns 200 Chapter
- @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" ]
}
```

4.3.2.3.4 Modify Existing Chapter

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

- Modify existing Chapter instance

- @params entity Chapter
- @returns 200 Chapter
- @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"
]
}
```

4.3.2.3.5 Delete Existing Chapter

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

- Removes Chapter
- @params id Chapter.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/chapters/2627
```

```
Returns 204
```

4.3.2.3.6 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/dependencies/chapters/2626
```

```
Returns 200
```

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

```
{
  {
    "id":9
  },
  {
    "id":3
  },
  {
    "id":10
  }
]
}
```

4.3.2.4 Reports

4.3.2.4.1 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]

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 }
]
}, ...
]
```

4.3.2.4.2 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

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 }
  ]
}
```

4.3.2.4.3 Create New Reports

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

- Creates new Report instance
- @params entity Report
- @returns 200 Report
- @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
    }
  ]
}
```

4.3.2.4.4 Modify Existing Report

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

- Modify existing Report instance
- @params entity Report
- @returns 200 Report
- @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
}
]
```

4.3.2.4.5 Delete Existing Report

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

- Removes Report
- @params id Report.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/reports/2627
```

Returns 204

4.3.2.4.6 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
      ]
    }
  ]
}
```

4.3.2.5 Email Reports

4.3.2.5.1 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]

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
  },...
]
```

4.3.2.5.2 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

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
}
```

4.3.2.5.3 Create New Email Report

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

- Creates new chapters instance
- @params entity EmailReport
- @returns 200 EmailReport
- @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
}
```

4.3.2.5.4 Modify Existing Email Reports

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

- Modify existing EmailReport instance
- @params entity EmailReport
- @returns 200 EmailReport
- @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
}
```

4.3.2.5.5 Delete Existing Report

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

- Removes Report
- @params id EmailReport.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/emailreports/4
```

```
Returns 204
```

4.3.2.6 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 Search object specifying time range and business hours.
- @returns 204 No data "No data satisfying search conditions found."
- @returns 200 TopChapterData or TrafficChapterData

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

5 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>
```

6 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],  
...]
```

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.