



# **ADS REST API**

## Developer Guide

June 18, 2015

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Authorization</b>	<b>7</b>
2.1	Obtaining authentication token . . . . .	7
<b>3</b>	<b>Data Types</b>	<b>8</b>
3.1	Entity validation result . . . . .	8
3.2	Configuration resources . . . . .	8
3.3	Events . . . . .	9
3.4	Charts and statistics . . . . .	11
3.5	Asynchronous results . . . . .	12
3.6	High level events . . . . .	12
3.7	Filters . . . . .	13
3.8	Perspectives . . . . .	14
3.9	Methods . . . . .	14
3.10	Event reports . . . . .	15
3.11	NetFlow sources . . . . .	16
3.12	PDF report chapters . . . . .	17
3.13	PDF reports . . . . .	18
3.14	PDF report schedules . . . . .	19
3.15	False positives . . . . .	20
<b>4</b>	<b>Methods</b>	<b>21</b>
4.1	Asynchronous processing . . . . .	21
4.1.1	Async results . . . . .	21
4.2	Configuration resources . . . . .	23
4.2.1	Priorities . . . . .	23
4.3	Events . . . . .	23
4.3.1	List of events . . . . .	23
4.3.2	Event detail . . . . .	26
4.3.3	Aggregated events by type . . . . .	27
4.3.4	Aggregated events by source . . . . .	28
4.3.5	Aggregated events by type and priority . . . . .	29
4.4	Charts and statistics . . . . .	30
4.4.1	Chart data for given search . . . . .	30
4.4.2	Overview statistics for given search . . . . .	31
4.5	Threats . . . . .	33
4.5.1	List of threats . . . . .	33

4.5.2	Events for given threat . . . . .	35
4.6	Event reports . . . . .	36
4.6.1	Empty template . . . . .	36
4.6.2	Retrieve all . . . . .	37
4.6.3	Retrieve one . . . . .	38
4.6.4	Create . . . . .	39
4.6.5	Update . . . . .	40
4.6.6	Delete . . . . .	41
4.6.7	Activate . . . . .	42
4.6.8	Deactivate . . . . .	43
4.7	False positives . . . . .	44
4.7.1	Empty template . . . . .	44
4.7.2	Retrieve all . . . . .	45
4.7.3	Retrieve one . . . . .	46
4.7.4	Create . . . . .	47
4.7.5	Delete . . . . .	49
4.8	Filters . . . . .	49
4.8.1	Empty template . . . . .	49
4.8.2	Retrieve all . . . . .	50
4.8.3	Retrieve one . . . . .	51
4.8.4	Create . . . . .	51
4.8.5	Update . . . . .	52
4.8.6	Dependencies . . . . .	53
4.8.7	Delete . . . . .	54
4.9	Methods . . . . .	55
4.9.1	Retrieve all . . . . .	55
4.9.2	Retrieve one . . . . .	56
4.9.3	Dependencies . . . . .	56
4.9.4	Activate . . . . .	57
4.9.5	Deactivate . . . . .	58
4.10	Method instances . . . . .	59
4.10.1	Create empty template . . . . .	59
4.10.2	Retrieve one . . . . .	60
4.10.3	Create . . . . .	62
4.10.4	Update . . . . .	64
4.10.5	Dependencies . . . . .	66
4.10.6	Delete . . . . .	67
4.10.7	Activate . . . . .	68
4.10.8	Deactivate . . . . .	68

4.11	NetFlow sources . . . . .	69
4.11.1	Empty template . . . . .	69
4.11.2	Retrieve all . . . . .	70
4.11.3	Retrieve one . . . . .	71
4.11.4	Create . . . . .	73
4.11.5	Update . . . . .	75
4.11.6	Dependencies . . . . .	76
4.11.7	Delete . . . . .	77
4.11.8	Activate . . . . .	78
4.11.9	Deactivate . . . . .	79
4.12	PDF reports . . . . .	80
4.12.1	Empty template . . . . .	80
4.12.2	Retrieve all . . . . .	80
4.12.3	Retrieve one . . . . .	82
4.12.4	Create . . . . .	84
4.12.5	Update . . . . .	85
4.12.6	Dependencies . . . . .	86
4.12.7	Delete . . . . .	87
4.13	PDF report chapters . . . . .	88
4.13.1	Empty template . . . . .	88
4.13.2	Retrieve all . . . . .	89
4.13.3	Retrieve one . . . . .	89
4.13.4	Create . . . . .	90
4.13.5	Update . . . . .	91
4.13.6	Dependencies . . . . .	91
4.13.7	Delete . . . . .	92
4.14	PDF report schedules . . . . .	93
4.14.1	Empty template . . . . .	93
4.14.2	Retrieve all . . . . .	94
4.14.3	Retrieve one . . . . .	95
4.14.4	Create . . . . .	96
4.14.5	Update . . . . .	97
4.14.6	Delete . . . . .	98
4.14.7	Activate . . . . .	99
4.14.8	Deactivate . . . . .	100
4.15	Perspectives . . . . .	100
4.15.1	Empty template . . . . .	100
4.15.2	Retrieve all . . . . .	101
4.15.3	Retrieve one . . . . .	102

4.15.4 Create . . . . .	103
4.15.5 Update . . . . .	104
4.15.6 Dependencies . . . . .	105
4.15.7 Delete . . . . .	106

# 1 Introduction

This document describes usage of REST API for Flowmon ADS plugin in version 7.02.00 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": "AnISCIWYbchsCc5sdc5ac4caca8a2",
  "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
```

## 3 Data Types

Following data structures are used in query parameters or results. Qustion mark [?] represents read only attribute. Id is usually read only attribute thus is written as ?id. Read only attributes should be omitted when CREATING a resource.

### 3.1 Entity validation result

ValidationError

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

ValidationResult

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

### 3.2 Configuration resources

Priority

```
Priority = {  
  id: Integer,  
  label: String  
} << Priorities;
```

PerspectiveAndPriority

```
PerspectiveAndPriority = {  
  id: Integer,  
  name: String,  
  priority: Integer(Priority.id)  
}
```



## 3.3 Events

### Event

```
Event = {  
  id: Integer("Event id"),  
  time: String(Timestamp, "Event time"),  
  source: IpAddressFull("Ip address of event source"),  
  type: String(Method.code,  
    "Code of method that detected the event."),  
  nfSource: NetFlowSource("Net Flow source the event was detected upon."),  
  batch: Integer(UNIX_TIMESTAMP, "Id of a~batch event belongs to."),  
  flowStamp: String(Timestamp, "Flow time"),  
  detail: String("Event detail"),  
  targets: Array(IpAddressFull, "Event targets, only first 11 targets.  
    Full targets list can be obtained with event detail."),  
  perspectives: Array(PerspectiveAndPriority,  
    "Perspectives event belongs to."),  
  priority: Integer(Priorities.id,  
    "For perspective search shows the priority  
    in first matched perspective")  
}
```

### EventsSearch

```
EventsSearch = {  
  from: String(Timestamp),  
  to: String(Timestamp),  
  ?source: String(IP) or Array,  
  ?target: String(IP) or Array,  
  ?type: String(Method.code) or Array,  
  ?filter: Integer(Filter.id) or Array,  
  ?nfSource: Integer(NfSource.id) or Array,  
  ?perspective: Integer(Perspective.id),  
  ?priority: Array(Priority.id, "Priorities to be included,  
    only if any perspective is given")  
}
```

### EventsByType

```
EventsByType = {  
  type: String(Method.code),  
  count: Integer  
}
```

### EventsBySource

```
EventsBySource = {  
  source: IPAddressFull,  
  count: Integer  
}
```

### EventsByTypeAndPriority

```
EventsByTypeAndPriority = {  
  type: String(Method.type),  
  priority: Integer(Priority.id),  
  count: Integer  
}
```

### EventsBySource

```
OrderByColumns = [  
  "type",  
  "source",  
  "detail",  
  "time",  
  "nfSource",  
  "target"  
]  
OrderBy = String(String("Event property name")<<  
  OrderByColumns+" "+String("Direction")<<  
  ["asc", "desc"])
```

### Comment

```
Comment = {  
  id: Integer,  
  user: Integer,  
  time: Timestamp,  
  text: String  
}
```

### EventDetailed

```
EventDetailed = {  
  ?targets: Array(IPAddressFull, "Full list of event targets"),  
  falsePositive: Integer(Boolean,  
    "1 if event is classified as false positive"),  
  comments: Array(Comment, "Comments to the event"),  
  flowsSearch: FlowsListingAnalysisSearch("Flowmon analysis  
    search object to retrieve flows evidence for given event.")  
} extends Event
```

## 3.4 Charts and statistics

### ChartDataSearch

```
ChartDataSearch = {  
  from: String(Timestamp),  
  to: String(Timestamp),  
  ?nfSource: Integer(NfSource.id),  
  perspective: Integer(Perspective.id)  
}
```

### ChartData

```
ChartData = [  
  Integer(UnixTimestamp, "Batch time"),  
  Number("Unclassified flows"),  
  Number(Bytes, "Unclassified bytes"),  
  Number("Unclassified packets"),  
  
  Number("Low priority flows"),  
  Number(Bytes, "Low priority bytes"),  
  Number("Low priority packets"),  
  
  Number("Meduim priority flows"),  
  Number(Bytes, "Meduim priority bytes"),  
  Number("Meduim priority packets"),  
  
  Number("High priority flows"),  
  Number(Bytes, "High priority bytes"),  
  Number("High priority packets")  
]
```

### ChartStatistics

```
ChartStatistics = {  
  flows: Number("Absolute flows"),  
  bytes: Number("Absolute bytes"),  
  packets: Number("Absolute packets"),  
  
  lFlows: Number("Absolute low priority Flows"),  
  lBytes: Number("Absolute low priority Bytes"),  
  lPackets: Number("Absolute low priority Packets"),  
  
  mFlows: Number("Absolute medium priority Flows"),  
  mBytes: Number("Absolute medium priority Bytes"),  
  mPackets: Number("Absolute medium priority Packets"),  
}
```

```
hFlows: Number("Absolute high priority Flows"),
hBytes: Number("Absolute high priority Bytes"),
hPackets: Number("Absolute high priority Packets"),

flowsAvg: Number("Average flows"),
bytesAvg: Number("Average bytes"),
packetsAvg: Number("Average packets"),

lFlowsAvg: Number("Average low priority Flows"),
lBytesAvg: Number("Average low priority Bytes"),
lPacketsAvg: Number("Average low priority Packets"),

mFlowsAvg: Number("Average medium priority Flows"),
mBytesAvg: Number("Average medium priority Bytes"),
mPacketsAvg: Number("Average medium priority Packets"),

hFlowsAvg: Number("Average high priority Flows"),
hBytesAvg: Number("Average high priority Bytes"),
hPacketsAvg: Number("Average high priority Packets")
}
```

## 3.5 Asynchronous results

AsyncResult

```
AsyncResult = {
  id: Integer,
  ?apiMethod: String("Name of api method that has been queried.")
}
```

## 3.6 High level events

ThreatMethod

```
ThreatMethod = {
  code: String,
  name: String
}
```

Threat

```
Threat = {
```

```
id: Integer("Threat id"),
source: IpAddressFull,
targets: Array(IpAddressFull, "First 11 targets"),
type: String(ThreatMethod.code),
detail: String,
closed: Integer(Boolean),
start: Timestamp,
end: Timestamp,
perspectives: Array(PerspectiveAndPriority),
priority: Integer(Priority.id)
}
```

#### ThreatsSearch

```
ThreatsSearch = {
  ?type: String(ThreatMethod.code) or Array,
} extends EventsSearch
```

#### ThreatDetailed

```
ThreatDetailed = {
  parts: Array(Event)
} extends Threat
```

## 3.7 Filters

#### Filter

```
Filter = {
  ?id: Integer,
  name: String,
  patterns: [IpPattern]
}
```

#### IpPattern

```
IpPattern = {
  pattern: String("Ip, range, ip with mask, wildcard ip, negation of previous"),
  note: String("Comment")
}
```

#### Action

```
Action = String("deactivate","delete");
```

#### Dependencies

```
Dependencies = {
  action: Action("Action the items are checked against."),
  items: [AdsObjects("Checked objects, may be in short form.")],

  dependent: [
    {
      action: Action("Action that would be performed in case checked action
                     was performed."),
      type: String("Data type of AdsObjects"),
      items: [AdsObject]
    },
    ...
  ]
}
```

## 3.8 Perspectives

#### PrioritySetting

```
PrioritySetting = {
  methodCode: String(Method.code),
  value: Priority.id,
  nfSource: NetFlowSource,
  filterIn: Filter("Source IP filter"),
  filterOut: Filter("Target IP filter")
}
```

#### Perspective

```
Perspective = {
  id: Integer,
  name: String,
  priorities: [PrioritySetting]
}
```

## 3.9 Methods

#### Method

```
Method = {
  id: Integer,
  code: String("Method code"),
}
```

```
name: String("Method name"),
active: Integer(Boolean, "1 if method is active"),
description: String("Method description"),
instancesLeft: Integer("Count of instances left to be created"),
defaultFilters: [Filter],
instances: [MethodInstance]
}
```

#### MethodParameter

```
MethodParameter = {
  id: Integer,
  method: String("Method code"),
  subId: Integer,
  name: String("Parameter name"),
  description: String("Paramater description"),
  value: String("Parameter value"),
  isFilter: Integer(Boolean, "1 if parameter values are treated as Filter ids"),
  group: String("Logical grouping of parameters by its meaning")
}
```

#### MethodInstance

```
MethodInstance = {
  id: Integer,
  methodCode: String("Code of parent method"),
  name: String,
  active: Integer(Boolean, "1 if method instance is active"),
  nfSources: [NetFlowSource("NetFlow source assigned to the instance")],
  filters: [Filter("Filter assigned to the instance")],
  parameters: [MethodParameter]
}
```

## 3.10 Event reports

#### EventReportFormat

```
EventReportFormat = {
  id: Integer,
  name: String
} << [
  {id: 0, name: "Full"},
  {id: 1, name: "Compact"},
  {id: 2, name: "Extra compact"},
  {id: 3, name: "RTTicket"},
]
```

```
{id: 4, name: "DantePlainText"}
]
```

#### EventReport

```
EventReport = {
  id: Integer,
  name: String,
  perspective: Perspective,
  priority: Integer(Priority.id),
  active: Integer(Boolean),
  fromEmail: String("From email"),
  emails: [String("Recipient email")],
  preventEmpty: Integer(Boolean),
  format: Integer(EventReportFormat.id),
  restHours: Integer("Count of hours to prevent same event reporting"),
  activeLinks: Integer(Boolean, "Relevant for full html report")
}
```

## 3.11 NetFlow sources

#### ActiveStatus

```
ActiveStatus = Integer(
  0 – inactive,
  1 – active,
  2 – starting
)
```

#### NetFlowSource

```
NetFlowSource = {
  ?id: Integer,
  name: String,
  active: Integer(ActiveStatus, "Computed each time NetFlow source is queried"),
  deduplicate: Integer(Boolean, "1 to deduplicate"),
  checkTimestamps: Integer(Boolean),
  isSip: Integer(Boolean),
  samplingRate: Integer,

  profile: String("Id of profile, see GET /rest/fmc/profiles/"),
  channels: [String("Id of profile channel, see GET /rest/fmc/profiles/")],

  //proxy settings
  proxyActive: Integer(Boolean),
```



```
virtualEnabled: Integer(Boolean, "1 to treat each channel as virtual source"),
?virtual: Integer(Boolean, "1 if source is virtual source"),
?parentSource: NetFlowSource(virtualEnables==1 && virtual==1, "Master source
                           or virtual source.")

?lastStart: String(Timestamp),
?lastStop: String(Timestamp),
?lastProcessed: Integer("Count of last processed flows"),

proxyActive: Integer(Boolean),
tolerance: Integer(1-1000),
reqOverload: Integer(1-1000),
resOverload: Integer(1-1000),
proxies: [Proxy]
}
```

#### Proxy

```
Proxy = {
  in: String("IP"),
  inPort: Integer("Port number"),
  out: String("IP"),
  filter: Filtre
}
```

## 3.12 PDF report chapters

#### PdfReportChapter

```
PdfReportChapter = {
  id: Integer,
  type: String(PdfReportChapterType.id),
  name: String,
  description: String,
  parameters: String(JSON_ENCODED(PdfReportChapterType.parameters))
}
```

PdfReportChapterType is one of OverallStatusChapterType, EventMatrixChapterType or EventsListChapterType.

#### OverallStatusChapterType

```
OverallStatusChapterType = {
  id: "OverallStatusChapterType",
}
```

```
parameters: {
  nfSource: Integer(NetFlowSource.id),
  perspective: Integer(Perspective.id),
  aspect: String("flows, bytes or packets"),
  logAxes: Integer(Boolean, "1 to use logarithmic axis in chart")
},
connectionParameters: {
  nfSource: Integer(NetFlowSource.id),
  aspect: String("flows, bytes or packets"),
  logAxes: Integer(Boolean, "1 to use logarithmic axis in chart")
}
}
```

#### EventMatrixChapterType

```
EventMatrixChapterType = {
  id: "EventMatrixChapterType",
  parameters: {
    perspective: Integer(Perspective.id),
    priority: Integer(1-5, "Minimal priority to report"),
    source: [String("IP address to limit search to")],
    filter: [Integer(Filter.id, "Filters to limit search to")],
    nfSource: [Integer(NetFlowSource.id, "NetFlow source to limit search to")],
    type: [String(Method.code, "Methods to limit search to")]
  },
  connectionParameters: {
    nfSource: [Integer(NetFlowSource.id, "NetFlow source to limit search to")]
  }
}
```

#### EventsListChapterType

```
EventsListChapterType = {
  id: "EventsListChapterType",
  parameters: {
    perspective: Integer(Perspective.id),
    priority: [Integer(1-5, "Priority to report")],
    limit: Integer("Limit of records")
  },
  connectionParameters: {}
}
```

## 3.13 PDF reports

#### PdfReport

```
PdfReport = {  
  id: Integer,  
  name: String,  
  description: String,  
  public: Integer(Boolean),  
  ?user: String("User that has created the report – name"),  
  connectedChapters: [ChapterConnection("Represents connection  
                                of chapter into the report")]  
}
```

#### ChapterConnection

```
ChapterConnection = {  
  chapter: PdfReportChapter,  
  parameters: String(JSON_ENCODE(PdfReportChapterType.connectionParameters))  
}
```

## 3.14 PDF report schedules

#### Day

```
Day = {  
  id: Integer,  
  name: String  
}
```

#### Days

```
Days = [Day] <<  
[  
  {id: 0, name: "Monday"},  
  {id: 1, name: "Tuesday"},  
  {id: 2, name: "Wednesday"},  
  {id: 3, name: "Thursday"},  
  {id: 4, name: "Friday"},  
  {id: 5, name: "Saturday"},  
  {id: 6, name: "Sunday"}  
]
```

#### Interval

```
Interval = [String] << ["day", "week", "month", "custom"];
```

#### PdfReportSchedule

```
PdfReportSchedule = {  
  id: Integer,  
  report: PdfReport,  
  interval: String(Interval),  
  from: String(Timestamp),  
  to: String(Timestamp),  
  sendOn: [Integer(Day.id)],  
  sender: String("Sender email"),  
  recipients: [String("Email")],  
  ?createdBy: String("User who has created the report – name"),  
  active: Integer(Boolean),  
  ?scheduled: String(Timestamp, "Last time scheduling was performed")  
}
```

## 3.15 False positives

#### FalsePositive

```
FalsePositive = {  
  ?id: Integer,  
  
  code: [String(Method.code, "Method of the rule")],  
  source: [String(IP or IPWildcard)],  
  target: [String(IP or IPWildcard)],  
  netFlowSources: [NetFlowSource],  
  sourceFilters: [Filter],  
  targetFilters: [Filter],  
  days: [Integer(Day.id)],  
  timeFrom: String("Time of the day the rule starts to be valid"),  
  timeTo: String("Time of the day the rule ends to be valid"),  
  validTo: String(Timestamp),  
  
  ?counter: Integer("Counts how many times the rule was used"),  
  ?sourceFilterCounter: Integer("Counts how many times the rule was used  
                                because of source filter"),  
  ?targetFilterCounter: Integer("Counts how many times the rule was used  
                                because of target filter"),  
  ?created: String(Timestamp),  
  ?lastIgnored: String(Timestamp)  
}
```

## 4 Methods

Data retrieving methods (GET) accept "fields" parameter to determine, which fields are to be contained in resulting data. In case fields parameter is not set all fields are involved if not said otherwise.

### 4.1 Asynchronous processing

Some results can be processed asynchronously. If async parameter is set to 1, server does not wait for result to be computed. It returns AsyncResult object with identification of a result that will contain computed data as soon as computation is finished. Finished results can be retrieved at GET /rest/ads/results/<id> API endpoint. Custom headers that would be produced by queried endpoint ("X-FM-TotalCount") are sent with GET /rest/ads/results/<id> response.

#### 4.1.1 Async results

Returns result of async endpoint query.

**GET /rest/ads/results/<id>**

- **<id>** Integer(AsyncResult.id) Id of requested result
- @throws 405 Validation error In case result id is invalid
- @returns 202 Not finished In case result with given id is being computed but has not finished yet.
- @returns 204 No data In case result with given id does not exist
- @returns 200 Object - computed async result - type is given by async endpoint that created the result.

Example Usage

```
GET /rest/ads/results/4589
```

```
Returns 200
Header "X-FM-TotalCount": 45687
[
  {
    "id": 3054669,
    "time": "2014-03-25 11:00:00",
    "source": {
      "ip": "192.168.9.3",
      "resolved": "aaaa",
      "country": "US",
      "blacklisted": 0
    },
    "type": "DNSANOMALY",
    "nfSource": {
      "id": 2,
      "name": "Some name"
    },
    "batch": 1395741600,
    "flowStamp": "2014-03-25 11:06:06",
    "detail": "Use of unexpected DNS server
(with response, connections: 2,
commonly used DNS servers: 8.8.8.8).",
    "targets": [
      {
        "ip": "192.168.209.10",
        "resolved": "aaaa",
        "country": "US",
        "blacklisted": 0
      }
    ],
    "priority": 0
  },
  ...
]
```

## 4.2 Configuration resources

### 4.2.1 Priorities

Returns list of priorities.

**GET /rest/ads/priorities**

- @returns 200 [Priority]

Example Usage

**GET /rest/ads/priorities**

```
Returns 200
[
  {id: 1, label: "Information"},
  {id: 2, label: "Low"},
  {id: 3, label: "Medium"},
  {id: 4, label: "High"},
  {id: 5, label: "Critical"}
]
```

## 4.3 Events

### 4.3.1 List of events

Returns list of events for given search object.

**GET /rest/ads/events**

- @?param **async** Integer(Boolean, "1 for asynchronous processing.")  
*Default 0*
- @?param **count** Integer(Boolean, "1 to include total count of events") Total count of events is

provided using "X-FM-TotalCount" response header.

*Default 1*

- @?param **limit** Integer("Count of rows to return")  
*Default 500*
- @?param **offset** Integer("Search offset")  
*Default 0*
- @?param **orderBy** Array(OrderBy, "Array of order by commands")  
*Default "time desc"*
- @param **search** EventsSearch
- @throws 405 Validation error In case search object is invalid
- @returns 204 No data
- @returns 200 [Event]

#### Example Usage

```
GET /rest/ads/events?search=JSON_ENCODE({
    from: "2014-01-01 12:00",
    to: "2014-01-01 12:00",
    source: ["12.25.54.2", "12.45.45.45"],
    target: "45.12.23.34",
    type: "DDOS",
    filter: 1,
    perspective: 2
})
```

Returns 200

Header "X-FM-TotalCount": 45687

```
[
  {
    "id": 3054669,
    "time": "2014-03-25 11:00:00",
    "source": {
      "ip": "192.168.9.3",
      "resolved": "aaaa",
      "country": "US",
      "blacklisted": 0
    },
    "type": "DNSANOMALY",
    "nfSource": {
```



```
        "id": 2,
        "name": "Some name"
    },
    "batch": 1395741600,
    "flowStamp": "2014-03-25 11:06:06",
    "detail": "Use of unexpected DNS server
(with response, connections: 2,
commonly used DNS servers: 8.8.8.8).",
    "targets": [
        {
            "ip": "192.168.209.10",
            "resolved": "aaaa",
            "country": "US",
            "blacklisted": 0
        }
    ],
    "priority": 0
},
...
]
```

#### Example of async processing

```
GET /rest/ads/events?async=1&count=0&limit=1000&search=JSON_ENCODE({
    from: "2014-01-01 12:00",
    to: "2014-01-01 12:00",
    source: ["12.25.54.2", "12.45.45.45"],
    target: "45.12.23.34",
    type: "DDOS",
    filter: 1,
    perspective: 2
})
```

Returns 200

```
{
    id: 4589,
    apiMethod: "/ads/events"
}
```

To get results client must query **GET** /rest/ads/results/4589 endpoint.

### 4.3.2 Event detail

Returns detail of given event.

**GET /rest/ads/event/<id>**

- **<id>** Integer(Event.id)
- @param **targets** Integer(Boolean, "1 to include all targets, may contain many records.")  
*Default 1*
- @throws 405 Validation error In case search object is invalid
- @returns 204 No data
- @returns 200 EventDetailed

#### Example Usage

**GET /rest/ads/event/3054669**

Returns 200

```
{
  "id": 3054669,
  "time": "2014-03-25 11:00:00",
  "source": {
    "ip": "192.168.9.3",
    "resolved": "aaaa",
    "country": "US",
    "blacklisted": 0
  },
  "type": "DNSANOMALY",
  "nfSource": {
    "id": 2,
    "name": "Some name"
  },
  "batch": 1395741600,
  "flowStamp": "2014-03-25 11:06:06",
  "detail": "Use of unexpected DNS server
(with response, connections: 2,
commonly used DNS servers: 8.8.8.8).",
  "targets": [
    {
      "ip": "192.168.209.10",
```

```
        "resolved": "aaaa",
        "country": "US",
        "blacklisted": 0
    },
    ...
    ...
],
perspectives: [
    {
        id: 1,
        name: "Security issues",
        priority: 4
    },
    ...
],
"priority": 0,

falsePositive: 0,
comments: [
    {
        id: 5,
        user: 1,
        time: "2014-07-31 12:29:32",
        text: "Zajimave"
    }
],

flowsSearch: {
    from: "2014-01-01 12:00",
    to: "2014-01-02 12:00",
    profile: "Sources/live",
    filter: "source port 80"
}
}
```

### 4.3.3 Aggregated events by type

Returns events aggregated by event type and ordered by count desc.

**GET /rest/ads/events/byType**

- @?param **async** Integer(Boolean, "1 for asynchronous processing.")  
*Default 0*
- @?param **limit** Integer("Top n limit")  
*Default 10*
- @param **search** EventsSearch
- @throws 405 Validation error In case search object is invalid
- @returns 204 No data
- @returns 200 [EventsByType]

#### Example Usage

```
GET /rest/ads/events/byType?search=JSON_ENCODE({
    from: "2014-01-01 12:00",
    to: "2014-01-01 12:00",
    source: ["12.25.54.2", "12.45.45.45"],
    target: "45.12.23.34",
    type: "DDOS",
    filter: 1,
    perspective: 2
})
```

Returns 200

```
[
  {
    type: "DDOS",
    count: 23
  },
  ...
]
```

### 4.3.4 Aggregated events by source

Returns events aggregated by event source ip address and ordered by count desc.

**GET /rest/ads/events/bySource**

- @?param **async** Integer(Boolean, "1 for asynchronous processing.")  
*Default 0*
- @?param **limit** Integer("Top n limit")  
*Default 10*
- @param **search** EventsSearch
- @throws 405 Validation error In case search object is invalid
- @returns 204 No data
- @returns 200 [EventsBySource]

#### Example Usage

```
GET /rest/ads/events/bySource?search=JSON_ENCODE({
    from: "2014-01-01 12:00",
    to: "2014-01-01 12:00",
    source: ["12.25.54.2", "12.45.45.45"],
    target: "45.12.23.34",
    type: "DDOS",
    filter: 1,
    perspective: 2
})
```

Returns 200

```
[
  {
    source: {
      ip: "12.12.15.2",
      resolved: "google",
      country: "CZ",
      blacklisted: 0
    },
    count: 25
  },
  ...
]
```

### 4.3.5 Aggregated events by type and priority

Returns events aggregated by event source ip address and ordered by count desc.

## GET /rest/ads/events/byTypeAndPriority

- @?param **async** Integer(Boolean, "1 for asynchronous processing.")  
*Default 0*
- @?param **limit** Integer  
*Default 10*
- @param **search** EventsByTypeAndPrioritySearch
- @throws 405 Validation error In case search object is invalid
- @returns 204 No data
- @returns 200 [EventsByTypeAndPriority]

### Example Usage

```
GET /rest/ads/events/byTypeAndPriority?search=JSON_ENCODE({
    from: "2014-01-01 12:00",
    to: "2014-01-01 12:00",
    perspective: 13,
    priority: [1,2]
})

Returns 200
[
  {
    type: "LATENCY",
    priority: 2,
    count: 45
  },
  ...
]
```

## 4.4 Charts and statistics

### 4.4.1 Chart data for given search

Returns events aggregated by event source ip address and ordered by count desc.

## GET /rest/ads/overview/chart

- @?param **async** Integer(Boolean, "1 for asynchronous processing.")  
*Default 0*
- @param **limit** Integer("Maximal count of returned values.") Values are returned for the whole time range but may be aggregated not to exceed limit parameter value. Limitation is useful not to return more values than the client is able to visualize. 1 value per pixel should be the highest possible resolution.  
*Default 800*
- @param **search** ChartDataSearch
- @throws 405 Validation error In case search object is invalid
- @returns 204 No data
- @returns 200 [ChartData]

### Example Usage

```
GET /rest/ads/overview/chart?limit=400&search=JSON_ENCODE({
    from: "2014-01-01 12:00",
    to: "2014-01-01 12:00",
    nfSource: 1,
    perspective: 2
})

Returns 200
[
    [1393515300,161.763,6728323.333,8894.937,
    0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000],
    [1393515600,161.763,6728323.333,8894.937
    ,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000],
    ...
]
```

## 4.4.2 Overview statistics for given search

Returns overview statistics for given search.

## GET /rest/ads/overview/statistics

- @?param **async** Integer(Boolean, "1 for asynchronous processing.")  
*Default 0*
- @param **search** ChartDataSearch
- @throws 405 Validation error In case search object is invalid
- @returns 204 No data
- @returns 200 ChartStatistics

### Example Usage

```
GET /rest/ads/overview/statistics?search=JSON_ENCODE({
    from: "2014-01-01 12:00",
    to: "2014-01-01 12:00",
    nfSource: 1,
    perspective: 2
})
```

Returns 200

```
{
  "flows": 6124121,
  "bytes": 206126132019,
  "packets": 289247487.5,
  "lFlows": 23825,
  "lBytes": 17969160840,
  "lPackets": 27659721,
  "mFlows": 0,
  "mBytes": 0,
  "mPackets": 0,
  "hFlows": 5081.5,
  "hBytes": 81840661.5,
  "hPackets": 166581.5,
  "flowsAvg": 68.502,
  "bytesAvg": 2305661.432,
  "packetsAvg": 3235.431,
  "lFlowsAvg": 0.266,
  "lBytesAvg": 200997.325,
  "lPacketsAvg": 309.393,
  "mFlowsAvg": 0,
  "mBytesAvg": 0,
  "mPacketsAvg": 0,
```



```
"hFlowsAvg": 0.057,  
"hBytesAvg": 915.444,  
"hPacketsAvg": 1.863  
}
```

## 4.5 Threats

### 4.5.1 List of threats

Returns list of threats for given search object.

#### **GET /rest/ads/threats**

- @?param **async** Integer(Boolean, "1 for asynchronous processing.")  
*Default 0*
- @?param **count** Integer(Boolean, "1 to include total count of events") Total count of events is provided using "X-FM-TotalCount" response header.  
*Default 1*
- @?param **limit** Integer("Count of rows to return")  
*Default 500*
- @?param **offset** Integer("Search offset")  
*Default 0*
- @?param **orderBy** Array(ThreatsOrderBy, "Array of order by commands")  
*Default "start desc"*
- @param **search** ThreatsSearch
- @throws 405 Validation error In case search object is invalid
- @returns 204 No data
- @returns 200 [Threat]

#### Example Usage

```
GET /rest/ads/threats?search=JSON_ENCODE({
```

```
    from: "2014-01-01 12:00",
    to: "2014-01-01 12:00",
    source: ["12.25.54.2", "12.45.45.45"],
    type: "NETDISCOVERY",
    filter: 1,
    perspective: 2
  })
```

Returns 200

Header "X-FM-TotalCount": 45687

```
[
  {
    id: 26250,
    source: {
      ip: "12.12.12.1",
      resolved: "seznam.cz",
      country: "CZ",
      blacklisted: 0
    },
    targets: [{
      ip: "12.12.12.1",
      resolved: "seznam.cz",
      country: "CZ",
      blacklisted: 0
    }],
    type: "NETDISCOVERY",
    detail: "Total 297 attempts to discovery the network.",
    closed: 0,
    start: "2014-07-23 17:35:00",
    end: "2014-07-31 13:35:00",
    perspectives: [
      {
        id: 1,
        name: "Security issues",
        priority: 4
      },
      ...
    ]
    priority: 0
  }
]
```

## 4.5.2 Events for given threat

Returns list of events creating given threat.

### GET /rest/ads/threats/<id>/events

- **<id>** Integer(Threat.id)
- @?param **async** Integer(Boolean, "1 for asynchronous processing.")  
*Default 0*
- @?param **count** Integer(Boolean, "1 to include total count of events") Total count of events is provided using "X-FM-TotalCount" response header.  
*Default 1*
- @?param **limit** Integer("Count of rows to return")  
*Default 500*
- @?param **offset** Integer("Search offset")  
*Default 0*
- @?param **orderBy** Array(OrderBy, "Array of order by commands")  
*Default "start desc"*
- @returns 204 No data
- @returns 200 [Event]

### Example Usage

**GET** /rest/ads/threats/2154/events

```
Returns 200
Header "X-FM-TotalCount": 4007
[
  {
    "id": 3054669,
    "time": "2014-03-25 11:00:00",
    "source": {
      "ip": "192.168.9.3",
      "resolved": "aaaa",
      "country": "US",
      "blacklisted": 0
    },
  },
]
```

```

    "type": "DNSANOMALY",
    "nfSource": {
      "id": 2,
      "name": "Some name"
    },
    "batch": 1395741600,
    "flowStamp": "2014-03-25 11:06:06",
    "detail": "Use of unexpected DNS server
(with response, connections: 2,
commonly used DNS servers: 8.8.8.8).",
    "targets": [
      {
        "ip": "192.168.209.10",
        "resolved": "aaaa",
        "country": "US",
        "blacklisted": 0
      }
    ],
    "priority": 0
  },
  ...
]

```

## 4.6 Event reports

### 4.6.1 Empty template

Returns template of EventReport object. Template can be used as a base to fill in data and create new instance of EventReport.

#### **GET /rest/ads/templates/event-report**

- @returns 200 EventReport("Empty event-report")
- @throws 401 header("X-FM-Message": "No permission to manipulate event-reports")

Example Usage

```
GET /rest/ads/templates/event-report
```

```
Returns 200
{
  "id": null,
  "name": null,
  "perspective": null,
  "priority": 4,
  "active": null,
  "fromEmail": "flowmon@invea.com",
  "emails": [],
  "preventEmpty": 1,
  "format": null,
  "script": null,
  "restHours": null,
  "activeLinks": 1
}
```

## 4.6.2 Retrieve all

Returns list of all defined event-reports.

### **GET /rest/ads/event-reports**

- @param **fields**
- @returns 204 header("X-FM-Message": "No event reports defined")
- @returns 200 [EventReport]

#### Example Usage

**GET /rest/ads/event-reports**

```
Returns 200
[
  {
    "id": 1,
    "name": "Test",
    "perspective": {
      "id": 44,
```

```
        "name": "Operational issues"
      },
      "priority": 4,
      "active": 1,
      "fromEmail": "flowmon@invea.com",
      "emails": [
        "test@company.cz"
      ],
      "preventEmpty": 1,
      "format": "o",
      "script": null,
      "restHours": null,
      "activeLinks": 1
    },
    ...
  ]
```

### 4.6.3 Retrieve one

Returns requested EventReport. 404 is returned if event-report does not exists or user has no permission to see it.

#### **GET /rest/ads/event-reports/<id>**

- @param **fields**
- @returns 200 EventReport
- @throws 404 header("X-FM-Message": "EventReport does not exist")

#### Example Usage

**GET /rest/ads/event-reports/1**

```
Returns 200
{
  "id": 1,
  "name": "Test",
  "perspective": {
    "id": 44,
```

```
        "name": "Operational issues"
    },
    "priority": 4,
    "active": 1,
    "fromEmail": "flowmon@invea.com",
    "emails": [
        "test@company.cz"
    ],
    "preventEmpty": 1,
    "format": "o",
    "script": null,
    "restHours": null,
    "activeLinks": 1
}
```

## 4.6.4 Create

### **POST /rest/ads/event-reports?entity=JSON\_ENCODE(EventReport)**

- @returns 201 EventReport - created event-report
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

#### Example Usage

```
POST /rest/ads/event-reports?entity=JSON_ENCODE({
    "name": "Test",
    "perspective": 44,
    "priority": 4,
    "active": 1,
    "fromEmail": "flowmon@invea.com",
    "emails": [
        "test@company.cz"
    ],
    "preventEmpty": 1,
    "format": "o",
    "restHours": null,
    "activeLinks": 1
})
```

Returns 201

```
{
  id: 2,
  "name": "Test",
  "perspective": {
    "id": 44,
    "name": "Operational issues"
  },
  "priority": 4,
  "active": 1,
  "fromEmail": "flowmon@invea.com",
  "emails": [
    "test@company.cz"
  ],
  "preventEmpty": 1,
  "format": "o",
  "restHours": null,
  "activeLinks": 1
}
```

## 4.6.5 Update

**PUT /rest/ads/event-reports?entity=JSON\_ENCODE(EventReport)**

- @returns 200 EventReport - updated event-report
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

Example Usage

```
PUT /rest/ads/event-reports?entity=JSON_ENCODE({
  "id": 1,
  "name": "Test renamed",
  "perspective": 44,
  "priority": 4,
  "active": 1,
  "fromEmail": "flowmon@invea.com",
```



```
"emails": [  
    "test@company.cz"  
],  
"preventEmpty": 1,  
"format": "o",  
"restHours": null,  
"activeLinks": 1  
})
```

Returns 200

```
{  
    id: 1,  
    "name": "Test renamed",  
    "perspective": {  
        "id": 44,  
        "name": "Operational issues"  
    },  
    "priority": 4,  
    "active": 1,  
    "fromEmail": "flowmon@invea.com",  
    "emails": [  
        "test@company.cz"  
    ],  
    "preventEmpty": 1,  
    "format": "o",  
    "restHours": null,  
    "activeLinks": 1  
}
```

## 4.6.6 Delete

Deletes event report. 404 is returned in case event report does not exists or user has no permission to see the event report at all. (EventReport does not exist for user.) 401 is returned if user sees the event-report but has no permission to delete it. 403 is returned if it is not possible to delete event report for some other reason. (Would result in inconsistency.)

**DELETE /rest/ads/event-reports/<id>**

- @param **<id>** Integer(EventReport.id)
- @returns 204 header("X-FM-Message": "Deleted")
- @throws 401 header("X-FM-Message": "No permission to delete event report")
- @throws 403 header("X-FM-Message": "Unable to delete event report")
- @throws 404 header("X-FM-Message": "Event report does not exist")

#### Example Usage

```
DELETE /rest/ads/event-reports/1
```

```
Returns 204
```

## 4.6.7 Activate

### PUT /rest/ads/event-reports/<id>/activate

- @param **<id>** Integer(EventReport.id)
- @returns 200 EventReport
- @throws 401 header("X-FM-Message": "No permission to activate event report")
- @throws 403 header("X-FM-Message": "Unable to activate event report")
- @throws 404 header("X-FM-Message": "Event report does not exist")

#### Example Usage

```
PUT /rest/ads/event-reports/1/activate
```

```
Returns 200
```

```
{
  "id": 1,
  "name": "Test",
  "perspective": {
    "id": 44,
    "name": "Operational issues"
  },
}
```

```
{
  "priority": 4,
  "active": 1,
  "fromEmail": "flowmon@invea.com",
  "emails": [
    "test@company.cz"
  ],
  "preventEmpty": 1,
  "format": "o",
  "restHours": null,
  "activeLinks": 1
}
```

## 4.6.8 Deactivate

### PUT /rest/ads/event-reports/<id>/deactivate

- @param <id> Integer(EventReport.id)
- @returns 200 EventReport
- @throws 401 header("X-FM-Message": "No permission to deactivate event report")
- @throws 403 header("X-FM-Message": "Unable to deactivate event report")
- @throws 404 header("X-FM-Message": "Event report does not exist")

#### Example Usage

```
PUT /rest/ads/event-reports/1/deactivate
```

Returns 200

```
{
  "id": 1,
  "name": "Test",
  "perspective": {
    "id": 44,
    "name": "Operational issues"
  },
  "priority": 4,
  "active": 0,
```

```
"fromEmail": "flowmon@invea.com",  
"emails": [  
  "test@company.cz"  
],  
"preventEmpty": 1,  
"format": "o",  
"restHours": null,  
"activeLinks": 1  
}
```

## 4.7 False positives

### 4.7.1 Empty template

Returns template of FalsePositive object. Template can be used as a base to fill in data and create new instance of FalsePositive.

#### **GET /rest/ads/templates/false-positive**

- @returns 200 FalsePositive("Empty false-positive")
- @throws 401 header("X-FM-Message": "No permission to manipulate false-positive")

#### Example Usage

**GET /rest/ads/templates/false-positive**

Returns 200

```
{  
  "id": null,  
  "code": [],  
  "source": [],  
  "target": [],  
  
  "netFlowSources": [],  
  "sourceFilters": [],  
  "targetFilters": [],  
}
```

```
"days": [0,1,2,3,4,5,6],  
"timeFrom": "00:00:00",  
"timeTo": "23:59:59",  
"validTo": "2015-03-31"  
}
```

## 4.7.2 Retrieve all

Returns list of all defined false-positives.

### **GET /rest/ads/false-positives**

- @param **fields**
- @returns 204 header("X-FM-Message": "No false-positives defined")
- @returns 200 [FalsePositive]

#### Example Usage

**GET /rest/ads/false-positives**

Returns 200

```
[  
  {  
    "id": 2,  
    "code": ["ANOMALY"],  
    "source": ["12.2.1.*"],  
    "target": ["1.2.3.4"],  
  
    "netFlowSources": [  
      {  
        "id": 2,  
        "name": "localhost",  
        "virtual": 0  
      }  
    ],  
    "sourceFilters": [  
      {"id": 1, "name": "LAN"}  
    ]  
  }  
]
```

```
    ],
    "targetFilters": [
      { "id": 1, "name": "LAN" }
    ],

    "days": [0,1,2,3,4,5,6],
    "timeFrom": "00:00:00",
    "timeTo": "23:59:59",
    "validTo": "2015-03-31",

    "created": "2015-03-30 12:28:17",
    "lastIgnored": null,
    "counter": 0,
    "sourceFilterCounter": 0,
    "targetFilterCounter": 0
  },
  ...
]
```

### 4.7.3 Retrieve one

Returns requested FalsePositive. 404 is returned if false-positive does not exist or user has no permission to see it.

**GET /rest/ads/false-positives/<id>**

- @param **fields**
- @returns 200 FalsePositive
- @throws 404 header("X-FM-Message": "FalsePositive does not exist")

Example Usage

**GET /rest/ads/false-positives/2**

Returns 200

```
{
```

```
"id": 2,
"code": ["ANOMALY"],
"source": ["12.2.1.*"],
"target": ["1.2.3.4"],

"netFlowSources": [
  {
    "id": 2,
    "name": "localhost",
    "virtual": 0
  }
],
"sourceFilters": [
  {"id": 1, "name": "LAN"}
],
"targetFilters": [
  {"id": 1, "name": "LAN"}
],

"days": [0,1,2,3,4,5,6],
"timeFrom": "00:00:00",
"timeTo": "23:59:59",
"validTo": "2015-03-31",

"created": "2015-03-30 12:28:17",
"lastIgnored": null,
"counter": 0,
"sourceFilterCounter": 0,
"targetFilterCounter": 0
}
```

## 4.7.4 Create

**POST /rest/ads/false-positives?entity=JSON\_ENCODE(FalsePositive)**

- @returns 201 FalsePositive - created false-positive
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

### Example Usage

```
POST /rest/ads/false-positives?entity=JSON_ENCODE({
  "code": ["ANOMALY"],
  "source": ["12.2.1.*"],
  "target": ["1.2.3.4"],

  "netFlowSources": [2],
  "sourceFilters": [1],
  "targetFilters": [1],

  "days": [0,1,2,3,4,5,6],
  "timeFrom": "00:00:00",
  "timeTo": "23:59:59",
  "validTo": "2015-03-31"
})
```

Returns 201

```
{
  "id": 3,
  "code": ["ANOMALY"],
  "source": ["12.2.1.*"],
  "target": ["1.2.3.4"],

  "netFlowSources": [
    {
      "id": 2,
      "name": "localhost",
      "virtual": 0
    }
  ],
  "sourceFilters": [
    {"id": 1, "name": "LAN"}
  ],
  "targetFilters": [
    {"id": 1, "name": "LAN"}
  ],

  "days": [0,1,2,3,4,5,6],
  "timeFrom": "00:00:00",
  "timeTo": "23:59:59",
  "validTo": "2015-03-31"
}
```



## 4.7.5 Delete

Deletes false-positive. 404 is returned in case false-positive does not exist or user has no permission to see the false-positive at all. (FalsePositive does not exist for user.) 401 is returned if user sees the false-positive but has no permission to delete it. 403 is returned if it is not possible to delete false-positive for some other reason. (Would result in inconsistency.)

### **DELETE /rest/ads/false-positives/<id>**

- @param <id> Integer(EventReport.id)
- @returns 204 header("X-FM-Message": "Deleted")
- @throws 401 header("X-FM-Message": "No permission to delete false positive")
- @throws 403 header("X-FM-Message": "Unable to delete false positive")
- @throws 404 header("X-FM-Message": "false positive does not exist")

#### Example Usage

```
DELETE /rest/ads/false-positives/1
```

```
Returns 204
```

## 4.8 Filters

### 4.8.1 Empty template

Returns template of Filter object. Template can be used as a base to fill in data and create new instance of Filter.

### **GET /rest/ads/templates/filter**

- @returns 200 Filter("Empty filter")
- @throws 401 header("X-FM-Message": "No permission to manipulate filters")

## Example Usage

**GET** /rest/ads/templates/filter

Returns 200

```
{
  id: null,
  name: null,
  patterns: []
}
```

## 4.8.2 Retrieve all

Returns list of all defined filters.

**GET** /rest/ads/filters

- @param **fields**
- @returns 204 header("X-FM-Message": "No filters defined")
- @returns 200 [Filter]

## Example Usage

**GET** /rest/ads/filters

Returns 200

```
[
  {
    id: 1,
    name: "LAN",
    patterns: [
      {
        pattern: "12.2.*.2",
        note: null
      },
      ...
    ]
  },
  ...
]
```

### 4.8.3 Retrieve one

Returns requested Filter. 404 is returned if filter does not exist or user has no permission to see it.

#### **GET /rest/ads/filters/<id>**

- @param **fields**
- @returns 200 Filter
- @throws 404 header("X-FM-Message": "Filter does not exist")

#### Example Usage

```
GET /rest/ads/filters/1
```

Returns 200

```
{
  id: 1,
  name: "LAN",
  patterns: [
    {
      pattern: "12.2.*.2",
      note: null
    },
    ...
  ]
}
```

### 4.8.4 Create

#### **POST /rest/ads/filters?entity=JSON\_ENCODE(Filter)**

- @returns 201 Filter - created filter
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

#### Example Usage

```
POST /rest/ads/filters?entity=JSON_ENCODE({
  name: "LAN",
  patterns: [
    {
      pattern: "12.2.*.2",
      note: null
    },
    ...
  ]
})
```

Returns 201

```
{
  id: 12,
  name: "LAN",
  patterns: [
    {
      pattern: "12.2.*.2",
      note: null
    },
    ...
  ]
}
```

## 4.8.5 Update

**PUT /rest/ads/filters?entity=JSON\_ENCODE(Filter)**

- @returns 200 Filter - updated filter
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

#### Example Usage

```
PUT /rest/ads/filters?entity=JSON_ENCODE({
  id: 2
  name: "Renamed",
```

```
    patterns: [  
      {  
        pattern: "12.2.*.2",  
        note: null  
      },  
      ...  
    ]  
  })
```

Returns 200

```
{  
  id: 2,  
  name: "Renamed",  
  patterns: [  
    {  
      pattern: "12.2.*.2",  
      note: null  
    },  
    ...  
  ]  
}
```

## 4.8.6 Dependencies

Returns dependencies object - what would happen in case given filter was deleted.

**GET /rest/ads/dependencies/filters/<id>**

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

Example Usage

**GET /rest/ads/dependencies/filters/1**

Returns 200

```
{
  action: "delete",
  items: [
    {id: 1, name: "LAN"}
  ],
  dependent: [
    {
      action: "delete",
      type: "MethodClone",
      items: [
        {id: 12, code: "ANOMALY", name: "Default"},
        {id: 24, code: "DOS", name: "Default"}
      ]
    },
    {
      action: "deactivate",
      type: "HLMethod",
      items: [
        {code: "DOSATTACK", name: "Different kinds of DOS attacks"}
      ]
    }
  ]
}
```

## 4.8.7 Delete

Deletes filter. 404 is returned in case filter does not exist or user has no permission to see the filter at all. (Filter does not exist for user.) 401 is returned if user sees the filter but has no permission to delete it. 403 is returned if it is not possible to delete event report for some other reason. (Would result in inconsistency.)

### **DELETE /rest/ads/filters/<id>**

- @param **<id>** Integer(Filter.id)
- @returns 204 header("X-FM-Message": "Deleted")
- @throws 401 header("X-FM-Message": "No permission to delete filter")

- @throws 403 header("X-FM-Message": "Unable to delete filter")
- @throws 404 header("X-FM-Message": "Filter does not exist")

#### Example Usage

```
DELETE /rest/ads/filters/1
```

```
Returns 204
```

## 4.9 Methods

### 4.9.1 Retrieve all

Returns list of all defined methods.

#### **GET /rest/ads/methods**

- @param **fields**
- @returns 204 header("X-FM-Message": "No methods defined")
- @returns 200 [Method]

#### Example Usage

```
GET /rest/ads/methods
```

```
Returns 200
```

```
[  
  {  
    "id": 79,  
    "code": "BLACKLIST",  
    "name": "Communication with blacklisted hosts",  
    "active": 0,  
    "description": "Detection of communication with blacklisted  
                  IP addresses",  
    "instancesLeft": 0,  
    "defaultFilters": []  
  },  
]
```

```
    ...  
  ]
```

## 4.9.2 Retrieve one

Returns requested Method. 404 is returned if method does not exists or user has no permission to see it.

**GET /rest/ads/methods/<id>**

- @param **fields**
- @returns 200 Method
- @throws 404 header("X-FM-Message": "Method does not exist")

### Example Usage

**GET /rest/ads/methods/79**

Returns 200

```
{  
  "id": 79,  
  "code": "BLACKLIST",  
  "name": "Communication with blacklisted hosts",  
  "active": 0,  
  "description": "Detection of communication with blacklisted IP addresses",  
  "instancesLeft": 0,  
  "defaultFilters": [],  
  "instances": [MethodInstance]  
}
```

## 4.9.3 Dependencies

Returns dependencies object - what would happen in case given method was deleted.



### GET /rest/ads/dependencies/methods/<id>

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

#### Example Usage

**GET** /rest/ads/dependencies/methods/1

Returns 200

```
{
  action: "deactivate",
  items: [
    {id: 1, code: "DOS"}
  ],
  dependent: [
    {
      action: "deactivate",
      type: "HLMethod",
      items: [
        {code: "DOSATTACK", name: "Different kinds of DOS attacks"}
      ]
    }
  ]
}
```

## 4.9.4 Activate

### PUT /rest/ads/methods/<id>/activate

- @param <id> Integer(Method.id)
- @returns 200 Method
- @throws 401 header("X-FM-Message": "No permission to activate method")
- @throws 403 header("X-FM-Message": "Unable to activate method")

- @throws 404 header("X-FM-Message": "Method does not exist")

#### Example Usage

```
PUT /rest/ads/methods/79/activate
```

Returns 200

```
{
  "id": 79,
  "code": "BLACKLIST",
  "name": "Communication with blacklisted hosts",
  "active": 1,
  "description": "Detection of communication with blacklisted IP addresses",
  "instancesLeft": 0,
  "defaultFilters": []
}
```

## 4.9.5 Deactivate

**PUT /rest/ads/methods/<id>/deactivate**

- @param <id> Integer(Method.id)
- @returns 200 Method
- @throws 401 header("X-FM-Message": "No permission to deactivate method")
- @throws 403 header("X-FM-Message": "Unable to deactivate method")
- @throws 404 header("X-FM-Message": "Method does not exist")

#### Example Usage

```
GET /rest/ads/methods/79/deactivate
```

Returns 200

```
{
  "id": 79,
  "code": "BLACKLIST",
  "name": "Communication with blacklisted hosts",
}
```

```
"active": 0,  
"description": "Detection of communication with blacklisted IP addresses",  
"instancesLeft": 0,  
"defaultFilters": []  
}
```

## 4.10 Method instances

### 4.10.1 Create empty template

Creates empty method instance for given method type. Empty instance has preset parameters and filters.

**GET /rest/ads/templates/method-instances/<id>**

- @param **<id>** Integer(Method.id)
- @returns 200 MethodInstance("Empty method instance")
- @throws 401 header("X-FM-Message": "No permission to manipulate methods")

Example Usage

**GET /rest/ads/templates/method-instances/79**

Returns 200

```
{  
  "id": null,  
  "subId": null,  
  "methodCode": "BLACKLIST",  
  "name": null, //UPDATE  
  "active": 0,  
  "nfSources": null, //UPDATE  
  "filters": [], //UPDATE  
  "parameters": [  
    {  
      "id": 1890,  
      "method": "BLACKLIST",
```

```
        "subId": 0,
        "name": "DeleteEventsAfter",
        "description": "Delete events generated by this method after
                        desired days (0–don't delete events).",
        "value": "183", //UPDATE
        "isFilter": 0,
        "group": "Common",
        "groupOrder": 1
    },
    {
        "id": 1886,
        "method": "BLACKLIST",
        "subId": 0,
        "name": "BotnetActivities",
        "description": "Detect botnet activities (0–no, 1–yes).",
        "value": "1", //UPDATE
        "isFilter": 0,
        "group": "General",
        "groupOrder": 2
    },
    ...
]
}
```

## 4.10.2 Retrieve one

Returns requested Method instance. 404 is returned if method-instance does not exists or user has no permission to see it.

**GET /rest/ads/method-instances/<id>**

- @param **fields**
- @returns 200 MethodInstance
- @throws 404 header("X-FM-Message": "Method instance does not exist")

Example Usage

```
GET /rest/ads/method-instances/3254
```

Returns 200

```
{
  "id": 3254,
  "subId": 1,
  "methodCode": "DNSREVERSE",
  "name": "Default",
  "active": 1,
  "deactivate": 1,
  "nfSources": [
    {
      "id": 8,
      "name": "osmicka?",
      "virtual": 0,
      "active": 0
    },
    {
      "id": 55,
      "name": "eerew",
      "virtual": 0,
      "active": 0
    },
    {
      "id": 47,
      "name": "bbb",
      "virtual": 0,
      "active": 0
    },
    {
      "id": 4,
      "name": "Test",
      "virtual": 0,
      "active": 0
    }
  ],
  "filters": [],
  "parameters": [
    {
      "id": 12947,
      "method": "DNSREVERSE",
      "subId": 1,
      "name": "DeleteEventsAfter",
      "description": "Delete events generated by this method"
    }
  ]
}
```

```

        after desired days (0–don't delete events).",
        "value": "183",
        "isFilter": 0,
        "group": "Common",
        "groupOrder": 1
    },
    {
        "id": 12946,
        "method": "DNSREVERSE",
        "subId": 1,
        "name": "MinimalTransfer",
        "description": "Include only devices which transferred more
                        than desired amount of data.",
        "value": "3000",
        "isFilter": 0,
        "group": "General",
        "groupOrder": 2
    }
]
}

```

### 4.10.3 Create

**POST /rest/ads/method-instances?entity=JSON\_ENCODE(MethodInstance)**

- @returns 201 MethodInstance - created method-instance
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

#### Example Usage

```

POST /rest/ads/method-instances?entity=JSON_ENCODE({
    "id": null,
    "subId": null,
    "methodCode": "BLACKLIST",
    "name": "My new instance", //UPDATE
    "active": 0,
    "nfSources": [1], //UPDATE
    "filters": [], //UPDATE

```

```
"parameters": [  
  {  
    "id": 1890,  
    "method": "BLACKLIST",  
    "subId": 0,  
    "name": "DeleteEventsAfter",  
    "description": "Delete events generated by this method after  
                  desired days (0–don't delete events).",  
    "value": "183", //UPDATE  
    "isFilter": 0,  
    "group": "Common",  
    "groupOrder": 1  
  },  
  {  
    "id": 1886,  
    "method": "BLACKLIST",  
    "subId": 0,  
    "name": "BotnetActivities",  
    "description": "Detect botnet activities (0–no, 1–yes).",  
    "value": "1", //UPDATE  
    "isFilter": 0,  
    "group": "General",  
    "groupOrder": 2  
  },  
  ...  
]  
}))
```

Returns 201

```
{  
  "id": 20132,  
  "subId": 2,  
  "methodCode": "BLACKLIST",  
  "name": "My new instance",  
  "active": 0,  
  "nfSources": [NetFlowSource(with id=1)],  
  "filters": [],  
  "parameters": [  
    {  
      "id": 123456,  
      "method": "BLACKLIST",  
      "subId": 2,  
      "name": "DeleteEventsAfter",  
      "description": "Delete events generated by this method after
```

```

        desired days (0–don't delete events).",
        "value": "183",
        "isFilter": 0,
        "group": "Common",
        "groupOrder": 1
    },
    {
        "id": 123457,
        "method": "BLACKLIST",
        "subId": 2,
        "name": "BotnetActivities",
        "description": "Detect botnet activities (0–no, 1–yes).",
        "value": "1",
        "isFilter": 0,
        "group": "General",
        "groupOrder": 2
    },
    ...
]
}

```

## 4.10.4 Update

**PUT /rest/ads/method-instances?entity=JSON\_ENCODE(MethodInstance)**

- @returns 200 MethodInstance - updated method-instance
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

### Example Usage

```

PUT /rest/ads/method-instances?entity=JSON_ENCODE({
    "id": 123,
    "subId": 2,
    "methodCode": "BLACKLIST",
    "name": "My new instance renamed", //UPDATE
    "active": 0,
    "nfSources": [1], //UPDATE
    "filters": [], //UPDATE

```



```
"parameters": [  
  {  
    "id": 1890,  
    "method": "BLACKLIST",  
    "subId": 0,  
    "name": "DeleteEventsAfter",  
    "description": "Delete events generated by this method after  
                  desired days (0–don't delete events).",  
    "value": "10", //UPDATE  
    "isFilter": 0,  
    "group": "Common",  
    "groupOrder": 1  
  },  
  {  
    "id": 1886,  
    "method": "BLACKLIST",  
    "subId": 0,  
    "name": "BotnetActivities",  
    "description": "Detect botnet activities (0–no, 1–yes).",  
    "value": "1", //UPDATE  
    "isFilter": 0,  
    "group": "General",  
    "groupOrder": 2  
  },  
  ...  
]  
}))
```

Returns 200

```
{  
  "id": 123,  
  "subId": 2,  
  "methodCode": "BLACKLIST",  
  "name": "My new instance renamed",  
  "active": 0,  
  "nfSources": [NetFlowSource(with id=1)],  
  "filters": [],  
  "parameters": [  
    {  
      "id": 123456,  
      "method": "BLACKLIST",  
      "subId": 2,  
      "name": "DeleteEventsAfter",  
      "description": "Delete events generated by this method after
```

```

        desired days (0–don't delete events).",
        "value": "10",
        "isFilter": 0,
        "group": "Common",
        "groupOrder": 1
      },
      {
        "id": 123457,
        "method": "BLACKLIST",
        "subId": 2,
        "name": "BotnetActivities",
        "description": "Detect botnet activities (0–no, 1–yes).",
        "value": "1",
        "isFilter": 0,
        "group": "General",
        "groupOrder": 2
      },
      ...
    ]
  }

```

## 4.10.5 Dependencies

Returns dependencies object - what would happen in case given method instance was deactivated or deleted.

**GET /rest/ads/dependencies/method-instances/<id>**

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

Example Usage

**GET /rest/ads/dependencies/method-instances/1203?action="delete"**

```

Returns 200
{

```

```
    action: "delete",
    items: [
      {id: 1203, name: "Default DOS instance"}
    ],
    dependent: [
      {
        action: "deactivate",
        type: "Method",
        items: [
          {code: "DOS", name: "Denial of service"}
        ]
      }
    ]
  }
}
```

## 4.10.6 Delete

Deletes method instance. 404 is returned in case method instance does not exist or user has no permission to see the method-instance at all. (Method instance does not exist for user.) 401 is returned if user sees the method-instance but has no permission to delete it. 403 is returned if it is not possible to delete event report for some other reason. (Would result in inconsistency.)

### **DELETE /rest/ads/method-instances/<id>**

- @param <id> Integer(MethodInstance.id)
- @returns 204 header("X-FM-Message": "Deleted")
- @throws 401 header("X-FM-Message": "No permission to delete method-instance")
- @throws 403 header("X-FM-Message": "Unable to delete method-instance")
- @throws 404 header("X-FM-Message": "Method instance does not exist")

#### Example Usage

```
DELETE /rest/ads/method-instances/1
```

```
Returns 204
```

## 4.10.7 Activate

### PUT /rest/ads/method-instances/<id>/activate

- @param <id> Integer(MethodInstance.id)
- @returns 200 MethodInstance
- @throws 401 header("X-FM-Message": "No permission to activate method instance")
- @throws 403 header("X-FM-Message": "Unable to activate method instance")
- @throws 404 header("X-FM-Message": "Method instance does not exist")

#### Example Usage

```
PUT /rest/ads/method-instances/1023/activate
```

Returns 200

```
{
  "id": 1023,
  "subId": 1,
  "methodCode": "DNSREVERSE",
  "name": "Default",
  "active": 1,
  ...
}
```

## 4.10.8 Deactivate

### PUT /rest/ads/method-instances/<id>/deactivate

- @param <id> Integer(MethodInstance.id)
- @returns 200 MethodInstance
- @throws 401 header("X-FM-Message": "No permission to deactivate method instance")
- @throws 403 header("X-FM-Message": "Unable to deactivate method instance")

- @throws 404 header("X-FM-Message": "Method instance does not exist")

#### Example Usage

```
PUT /rest/ads/method-instances/1023/activate
```

Returns 200

```
{
  "id": 1023,
  "subId": 1,
  "methodCode": "DNSREVERSE",
  "name": "Default",
  "active": 0,
  ...
}
```

## 4.11 NetFlow sources

### 4.11.1 Empty template

Returns template of NetFlowSource object. Template can be used as a base to fill in data and create new instance of NetFlowSource.

#### **GET /rest/ads/templates/net-flow-source**

- @returns 200 NetFlowSource("Empty net-flow-source")
- @throws 401 header("X-FM-Message": "No permission to manipulate net-flow-source")

#### Example Usage

```
GET /rest/ads/templates/net-flow-source
```

Returns 200

```
{
  "id": null,
  "name": null,
  "active": null,
}
```

```
"deduplicate": null ,
"checkTimestamps": 1,
"isSip": 0,
"lastStart": null ,
"lastStop": null ,
"lastProcessed": null ,
"profile": null ,
"channels": null ,
"samplingRate": 1,
"proxyActive": 0,
"tolerance": 100,
"reqOverload": 20,
"resOverload": 20,
"proxies": [],
"virtualEnabled": 0,
"virtual": 0,
"parentSource": null
}
```

## 4.11.2 Retrieve all

Returns list of all defined net-flow-sources.

### **GET /rest/ads/net-flow-sources**

- @param **fields**
- @returns 204 header("X-FM-Message": "No net-flow-sources defined")
- @returns 200 [NetFlowSource]

#### Example Usage

**GET /rest/ads/net-flow-sources**

Returns 200

```
[
  {
    "id": 2,
    "active": 1,
```

```
    "deduplicate": 0,
    "lastStart": "2015-03-30 09:50:16",
    "lastStop": "2015-03-30 09:51:07",
    "lastProcessed": 4284,
    "name": "demo.invea.cz",
    "profile": "live",
    "channels": [
        "mirror-HP2650"
    ],
    "samplingRate": 1,
    "checkTimestamps": 1,
    "isSip": 0,
    "proxyActive": 0,
    "tolerance": 5,
    "reqOverload": 5,
    "resOverload": 5,
    "proxies": [
        {
            "in": "12.2.2.2",
            "inPort": 555,
            "out": "12.3.3.3",
            "filter": {
                "id": 6,
                "name": "ADS"
            }
        }
    ],
    "virtualEnabled": 0,
    "virtual": 0,
    "parentSource": null
}
,
...
]
```

### 4.11.3 Retrieve one

Returns requested NetFlowSource. 404 is returned if net-flow-source does not exists or user has no permission to see it.

**GET /rest/ads/net-flow-sources/<id>**

- @param **fields**
- @returns 200 NetFlowSource
- @throws 404 header("X-FM-Message": "NetFlowSource does not exist")

## Example Usage

**GET /rest/ads/net-flow-sources/2**

Returns 200

```
{
  "id": 2,
  "active": 1,
  "deduplicate": 0,
  "lastStart": "2015-03-30 09:50:16",
  "lastStop": "2015-03-30 09:51:07",
  "lastProcessed": 4284,
  "name": "demo.invea.cz",
  "profile": "live",
  "channels": [
    "mirror-HP2650"
  ],
  "samplingRate": 1,
  "checkTimestamps": 1,
  "isSip": 0,
  "proxyActive": 0,
  "tolerance": 5,
  "reqOverload": 5,
  "resOverload": 5,
  "proxies": [
    {
      "in": "12.2.2.2",
      "inPort": 555,
      "out": "12.3.3.3",
      "filter": {
        "id": 6,
        "name": "ADS"
      }
    }
  ],
  "virtualEnabled": 0,
```



```
"virtual": 0,  
"parentSource": null  
}
```

## 4.11.4 Create

### **POST /rest/ads/net-flow-sources?entity=JSON\_ENCODE(NetFlowSource)**

- @returns 201 NetFlowSource - created net-flow-source
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

#### Example Usage

```
POST /rest/ads/net-flow-sources?entity=JSON_ENCODE({  
  "active": 1,  
  "deduplicate": 0,  
  "name": "demo.invea.cz",  
  "profile": "live",  
  "channels": [  
    "mirror-HP2650"  
  ],  
  "samplingRate": 1,  
  "checkTimestamps": 1,  
  "isSip": 0,  
  "proxyActive": 1,  
  "tolerance": 5,  
  "reqOverload": 5,  
  "resOverload": 5,  
  "proxies": [  
    {  
      "in": "12.2.2.2",  
      "inPort": 555,  
      "out": "12.3.3.3",  
      "filter": {  
        "id": 6,  
        "name": "ADS"  
      }  
    }  
  ]  
})
```

```
    ],  
    "virtualEnabled": 0  
  })  
  
Returns 201  
{  
  "id": 2,  
  "active": 1,  
  "deduplicate": 0,  
  "lastStart": null,  
  "lastStop": null,  
  "lastProcessed": 0,  
  "name": "demo.invea.cz",  
  "profile": "live",  
  "channels": [  
    "mirror-HP2650"  
  ],  
  "samplingRate": 1,  
  "checkTimestamps": 1,  
  "isSip": 0,  
  "proxyActive": 0,  
  "tolerance": 5,  
  "reqOverload": 5,  
  "resOverload": 5,  
  "proxies": [  
    {  
      "in": "12.2.2.2",  
      "inPort": 555,  
      "out": "12.3.3.3",  
      "filter": {  
        "id": 6,  
        "name": "ADS"  
      }  
    }  
  ],  
  "virtualEnabled": 0,  
  "virtual": 0,  
  "parentSource": null  
}
```

## 4.11.5 Update

**PUT /rest/ads/net-flow-sources?entity=JSON\_ENCODE(NetFlowSource)**

- @returns 200 NetFlowSource - updated net-flow-source
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

### Example Usage

```
POST /rest/ads/net-flow-sources?entity=JSON_ENCODE({
  "id": 2,
  "active": 1,
  "deduplicate": 1,
  "name": "demo.invea.cz-2",
  "profile": "live",
  "channels": [
    "mirror-HP2650"
  ],
  "samplingRate": 1,
  "checkTimestamps": 1,
  "isSip": 0,
  "proxyActive": 1,
  "tolerance": 5,
  "reqOverload": 5,
  "resOverload": 5,
  "proxies": [
    {
      "in": "12.2.2.2",
      "inPort": 555,
      "out": "12.3.3.3",
      "filter": {
        "id": 6,
        "name": "ADS"
      }
    }
  ],
  "virtualEnabled": 0
})
```

Returns 200

```
{
  "id": 2,
  "active": 1,
  "deduplicate": 1,
  "lastStart": "2015-03-30 09:50:16",
  "lastStop": "2015-03-30 09:51:07",
  "lastProcessed": 4284,
  "name": "demo.invea.cz-2",
  "profile": "live",
  "channels": [
    "mirror-HP2650"
  ],
  "samplingRate": 1,
  "checkTimestamps": 1,
  "isSip": 0,
  "proxyActive": 1,
  "tolerance": 5,
  "reqOverload": 5,
  "resOverload": 5,
  "proxies": [
    {
      "in": "12.2.2.2",
      "inPort": 555,
      "out": "12.3.3.3",
      "filter": {
        "id": 6,
        "name": "ADS"
      }
    }
  ],
  "virtualEnabled": 0,
  "virtual": 0,
  "parentSource": null
}
```

## 4.11.6 Dependencies

Returns dependencies object - what would happen in case given NetFlow was deleted.

**GET /rest/ads/dependencies/net-flow-sources/<id>**

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

#### Example Usage

```
PUT /rest/ads/dependencies/net-flow-sources/1
```

```
Returns 200
```

```
{
  action: "delete",
  items: [
    {id: 1, name: "Default"}
  ],
  dependent: [
    {
      action: "delete",
      type: "MethodClone",
      items: [
        {id: 12, code: "ANOMALY", name: "Default"},
        {id: 24, code: "DOS", name: "Default"}
      ]
    },
    {
      action: "deactivate",
      type: "HLMethod",
      items: [
        {code: "DOSATTACK", name: "Different kinds of DOS attacks"}
      ]
    }
  ]
}
```

### 4.11.7 Delete

Deletes NetFlow. 404 is returned in case NetFlow does not exists or user has no permission to see the NetFlow at all. (NetFlow does not exist for user.) 401 is returned if user sees the NetFlow

but has no permission to delete it. 403 is returned if it is not possible to delete event report for some other reason. (Would result in inconsistency.)

#### **DELETE /rest/ads/net-flow-sources/<id>**

- @param **<id>** Integer(NetFlow.id)
- @returns 204 header("X-FM-Message": "Deleted")
- @throws 401 header("X-FM-Message": "No permission to delete NetFlow")
- @throws 403 header("X-FM-Message": "Unable to delete NetFlow")
- @throws 404 header("X-FM-Message": "NetFlow does not exist")

##### Example Usage

```
DELETE /rest/ads/net-flow-sources/1
```

```
Returns 204
```

## 4.11.8 Activate

#### **PUT /rest/ads/net-flow-sources/<id>/activate**

- @param **<id>** Integer(NetFlow.id)
- @returns 200 NetFlow
- @throws 401 header("X-FM-Message": "No permission to activate NetFlow")
- @throws 403 header("X-FM-Message": "Unable to activate NetFlow")
- @throws 404 header("X-FM-Message": "NetFlow does not exist")

##### Example Usage

```
PUT /rest/ads/net-flow-sources/1/activate
```

```
Returns 200
```

```
{
  "id": 1,
  "name": "Default",
  "active": 1,
  ...
}
```

## 4.11.9 Deactivate

**PUT /rest/ads/net-flow-sources/<id>/deactivate**

- @param <id> Integer(NetFlow.id)
- @returns 200 NetFlow
- @throws 401 header("X-FM-Message": "No permission to deactivate NetFlow")
- @throws 403 header("X-FM-Message": "Unable to deactivate NetFlow")
- @throws 404 header("X-FM-Message": "NetFlow does not exist")

### Example Usage

```
PUT /rest/ads/net-flow-sources/1/deactivate
```

Returns 200

```
{
  "id": 1,
  "name": "Default",
  "Active": 0,
  ...
}
```

## 4.12 PDF reports

### 4.12.1 Empty template

Returns template of PdfReport object. Template can be used as a base to fill in data and create new instance of PdfReport.

#### **GET /rest/ads/templates/pdf-report**

- @returns 200 PdfReport("Empty pdf-report")
- @throws 401 header("X-FM-Message": "No permission to manipulate pdf-reports")

#### Example Usage

**GET** /rest/ads/templates/pdf-report

Returns 200

```
{
  "id": null,
  "name": null,
  "description": null,
  "public": 1,
  "user": "John Smith",
  "connectedChapters": []
}
```

### 4.12.2 Retrieve all

Returns list of all defined pdf-reports.

#### **GET /rest/ads/pdf-reports**

- @param **fields**
- @returns 204 header("X-FM-Message": "No pdf-reports defined")
- @returns 200 [PdfReport]



## Example Usage

**GET** /rest/ads/pdf-reports

Returns 200

```
[
  {
    "id": 1,
    "name": "Report",
    "description": "Report description",
    "public": 1,
    "user": "John Smith",
    "connectedChapters": [
      {
        "chapter": {
          "id": 1,
          "name": "Chapter 1",
          "type": "EventsListChapterType",
          "description": null,
          "parameters": "{\"perspective\":127,\"priority\":[1,2,3,5],\"limit\":500}"
        },
        "parameters": "{}"
      },
      {
        "chapter": {
          "id": 2,
          "name": "Chapter 2",
          "type": "EventMatrixChapterType",
          "description": null,
          "parameters": "{\"perspective\":127,\"priority\":4,\"source\":\"\", \"filter\":1076,\"nfSource\":4,\"type\":[\"ALIENDEV\", \"ANOMALY\", \"BITTORRENT\", \"BLACKLIST\", \"BROKENSEN\", \"DHCPANOM\", \"DIRINET\", \"DIVCOM\", \"DNSANOMALY\", \"DNSQUERY\", \"DNSREVERSE\", \"DOS\", \"HIGHTRANSF\", \"HONEYPOT\", \"HTTPDICT\", \"ICGUARD\", \"ICMPANOM\", \"INSTMSG\", \"L3ANOMALY\", \"LATENCY\", \"MULTICAST\", \"OUTSPAM\", \"RDPDICT\", \"SCANS\", \"SRVNA\", \"SSHDICT\", \"SYSCHECK\", \"TEAMVIEWER\", \"TELNET\", \"TOR\", \"UPLOAD\", \"VOIP\", \"VPN\", \"WEBSHARE\"]}"
        }
      }
    ]
  }
]
```

```

    },
    "parameters": "{\\\"nfSource\\\":[4,47]}"
  },
  {
    "chapter": {
      "id": 3,
      "name": "Chapter 3",
      "type": "OverallStatusChapterType",
      "description": null,
      "parameters": "{\\\"nfSource\\\":4,\\\"perspective\\\":127,\\\"aspect\\\":\\\"flows\\\",\\\"logAxes\\\":1}"
    },
    "parameters": "{\\\"nfSource\\\":4,\\\"aspect\\\":\\\"flows\\\",\\\"logAxes\\\":1}"
  }
]
},
...
]

```

### 4.12.3 Retrieve one

Returns requested PdfReport. 404 is returned if pdf-report does not exists or user has no permission to see it.

**GET /rest/ads/pdf-reports/<id>**

- @param **fields**
- @returns 200 PdfReport
- @throws 404 header("X-FM-Message": "PdfReport does not exist")

#### Example Usage

**GET /rest/ads/pdf-reports/1**

Returns 200

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

```

"name": "Report",
"description": "Report description",
"public": 1,
"user": "John Smith",
"connectedChapters": [
  {
    "chapter": {
      "id": 1,
      "name": "Chapter 1",
      "type": "EventsListChapterType",
      "description": null,
      "parameters": "{\"perspective\":127,\"priority\":[1,2,3,5],\"limit\":500}"
    },
    "parameters": "{}"
  },
  {
    "chapter": {
      "id": 2,
      "name": "Chapter 2",
      "type": "EventMatrixChapterType",
      "description": null,
      "parameters": "{\"perspective\":127,\"priority\":4,\"source\":\"\", \"filter\":1076,\"nfSource\":4,\"type\":[\"ALIENDEV\", \"ANOMALY\", \"BITTORRENT\", \"BLACKLIST\", \"BROKENSEN\", \"DHCPANOM\", \"DIRINET\", \"DIVCOM\", \"DNSANOMALY\", \"DNSQUERY\", \"DNSREVERSE\", \"DOS\", \"HIGHTRANSF\", \"HONEYPOT\", \"HTTPDICT\", \"ICGUARD\", \"ICMPANOM\", \"INSTMSG\", \"L3ANOMALY\", \"LATENCY\", \"MULTICAST\", \"OUTSPAM\", \"RDPDICT\", \"SCANS\", \"SRVNA\", \"SSHDICT\", \"SYSCHECK\", \"TEAMVIEWER\", \"TELNET\", \"TOR\", \"UPLOAD\", \"VOIP\", \"VPN\", \"WEBSHARE\"]}"
    },
    "parameters": "{\"nfSource\":[4,47]}"
  },
  {
    "chapter": {
      "id": 3,
      "name": "Chapter 3",
      "type": "OverallStatusChapterType",

```

```

        "description": null ,
        "parameters": "{ \"nfSource\":4, \"perspective\":127 ,
                          \"aspect\": \"flows\" , \"logAxes\":1} "
      },
      "parameters": "{ \"nfSource\":4, \"aspect\": \"flows\" ,
                      \"logAxes\":1} "
    }
  ]
}

```

## 4.12.4 Create

### POST /rest/ads/pdf-reports?entity=JSON\_ENCODE(PdfReport)

- @returns 201 PdfReport - created pdf-report
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

#### Example Usage

```

POST /rest/ads/pdf-reports?entity=JSON_ENCODE({
  "name": "Report",
  "description": "Report description",
  "public": 1,
  "user": "John Smith",
  "connectedChapters": [
    {
      "chapter": {
        "id": 1,
        "name": "Chapter 1",
        "type": "EventsListChapterType",
        "description": null,
        "parameters": "{ \"perspective\":127, \"priority\":[1,2,3,5],
                          \"limit\":500} "
      },
      "parameters": "{}"
    }
  ]
})

```

```
Returns 201
{
  "id": 3,
  "name": "Report",
  "description": "Report description",
  "public": 1,
  "user": "John Smith",
  "connectedChapters": [
    {
      "chapter": {
        "id": 1,
        "name": "Chapter 1",
        "type": "EventsListChapterType",
        "description": null,
        "parameters": "{ \"perspective\":127, \"priority\":[1,2,3,5], \"limit\":500}"
      },
      "parameters": "{}"
    }
  ]
}
```

## 4.12.5 Update

**PUT /rest/ads/pdf-reports?entity=JSON\_ENCODE(PdfReport)**

- @returns 200 PdfReport - updated pdf-report
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

### Example Usage

```
PUT /rest/ads/pdf-reports?entity=JSON_ENCODE({
  "id": 2,
  "name": "Report renamed",
  "description": "Report description",
  "public": 1,
  "user": "John Smith",
  "connectedChapters": [
    {
```

```
        "chapter": {
            "id": 1,
            "name": "Chapter 1",
            "type": "EventsListChapterType",
            "description": null,
            "parameters": "{\"perspective\":127,\"priority\":[1,2,3,5],\"limit\":500}"
        },
        "parameters": "{}"
    ]
})

Returns 200
{
    "id": 2,
    "name": "Report renamed",
    "description": "Report description",
    "public": 1,
    "user": "John Smith",
    "connectedChapters": [
        {
            "chapter": {
                "id": 1,
                "name": "Chapter 1",
                "type": "EventsListChapterType",
                "description": null,
                "parameters": "{\"perspective\":127,\"priority\":[1,2,3,5],\"limit\":500}"
            },
            "parameters": "{}"
        }
    ]
}
```

## 4.12.6 Dependencies

Returns dependencies object - what would happen in case given pdf-report was deleted.

**GET /rest/ads/dependencies/pdf-reports/<id>**

- @returns 200 Dependencies for delete action

- @throws 401 header("X-FM-Message": "No permission to manipulate pdf-reports")
- @throws 404 header("X-FM-Message": "PdfReport does not exist")

#### Example Usage

**GET** /rest/ads/dependencies/pdf-reports/1

Returns 200

```
{
  action: "delete",
  items: [
    {id: 1, name: "Report"}
  ],
  dependent: [
    {
      action: "delete",
      type: "ScheduledReport",
      items: [
        PdfReportSchedule
      ]
    }
  ]
}
```

## 4.12.7 Delete

Deletes pdf-report. 404 is returned in case pdf-report does not exists or user has no permission to see the pdf-report at all. (PdfReport does not exist for user.) 401 is returned if user sees the pdf-report but has no permission to delete it. 403 is returned if it is not possible to delete event report for some other reason. (Would result in inconsistency.)

**DELETE** /rest/ads/pdf-reports/<id>

- @param <id> Integer(PdfReport.id)
- @returns 204 header("X-FM-Message": "Deleted")
- @throws 401 header("X-FM-Message": "No permission to delete pdf-report")

- @throws 403 header("X-FM-Message": "Unable to delete pdf-report")
- @throws 404 header("X-FM-Message": "PdfReport does not exist")

#### Example Usage

```
DELETE /rest/ads/pdf-reports/1
```

```
Returns 204
```

## 4.13 PDF report chapters

### 4.13.1 Empty template

Returns template of PdfReportChapter object. Template can be used as a base to fill in data and create new instance of PdfReportChapter.

#### **GET /rest/ads/templates/pdf-report-chapter**

- @returns 200 PdfReportChapter("Empty pdf-report-chapter")
- @throws 401 header("X-FM-Message": "No permission to manipulate pdf-report-chapters")

#### Example Usage

```
GET /rest/ads/templates/pdf-report-chapter
```

```
Returns 200
```

```
{
  "id": null ,
  "type": null ,
  "name": null ,
  "description": null ,
  "parameters": null
}
```



## 4.13.2 Retrieve all

Returns list of all defined pdf-report-chapters.

### **GET /rest/ads/pdf-report-chapters**

- @param **fields**
- @returns 204 header("X-FM-Message": "No pdf-report-chapters defined")
- @returns 200 [PdfReportChapter]

#### Example Usage

**GET /rest/ads/pdf-report-chapters**

Returns 200

```
[
  {
    "id": 1,
    "type": "EventsListChapterType",
    "name": "Chapter 1",
    "description": "This is first chapter showing list of events.",
    "parameters": "{\"perspective\":127,\"priority\":[1,2,3,5],\"limit\":500}"
  },
  ...
]
```

## 4.13.3 Retrieve one

Returns requested PdfReportChapter. 404 is returned if pdf-report-chapter does not exists or user has no permission to see it.

### **GET /rest/ads/pdf-report-chapters/<id>**

- @param **fields**
- @returns 200 PdfReportChapter

- @throws 404 header("X-FM-Message": "PdfReportChapter does not exist")

#### Example Usage

**GET** /rest/ads/pdf-report-chapters/1

Returns 200

```
{
  "id": 1,
  "type": "EventsListChapterType",
  "name": "Chapter 1",
  "description": "This is first chapter showing list of events.",
  "parameters": "{\"perspective\":127,\"priority\":[1,2,3,5],\"limit\":500}"
}
```

## 4.13.4 Create

**POST** /rest/ads/pdf-report-chapters?entity=JSON\_ENCODE(PdfReportChapter)

- @returns 201 PdfReportChapter - created pdf-report-chapter
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

#### Example Usage

**POST** /rest/ads/pdf-report-chapters?entity=JSON\_ENCODE({  
 "type": "EventsListChapterType",  
 "name": "Chapter 2",  
 "description": "This is first chapter showing list of events.",  
 "parameters": "{\"perspective\":127,\"priority\":[1,2,3,5],\"limit\":500}"  
})

Returns 201

```
{
  "id": 2,
  "type": "EventsListChapterType",
  "name": "Chapter 2",
  "description": "This is first chapter showing list of events.",
  "parameters": "{\"perspective\":127,\"priority\":[1,2,3,5],\"limit\":500}"
}
```

## 4.13.5 Update

**PUT /rest/ads/pdf-report-chapters?entity=JSON\_ENCODE(PdfReportChapter)**

- @returns 200 PdfReportChapter - updated pdf-report-chapter
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

### Example Usage

```
PUT /rest/ads/pdf-report-chapters?entity=JSON_ENCODE({
  "id": 2,
  "type": "EventsListChapterType",
  "name": "Chapter 2 – renamed",
  "description": "This is first chapter showing list of events.",
  "parameters": "{\"perspective\":127,\"priority\":[1,2,3,5],\"limit\":500}"
})

Returns 200
{
  "id": 2,
  "type": "EventsListChapterType",
  "name": "Chapter 2 – renamed",
  "description": "This is first chapter showing list of events.",
  "parameters": "{\"perspective\":127,\"priority\":[1,2,3,5],\"limit\":500}"
}
```

## 4.13.6 Dependencies

Returns dependencies object - what would happen in case given pdf-report-chapter was deleted.

**GET /rest/ads/dependencies/pdf-report-chapters/<id>**

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

### Example Usage

**GET** /rest/ads/dependencies/pdf-report-chapters/1

Returns 200

```
{
  action: "delete",
  items: [
    {id: 1, name: "Chapter 1"}
  ],
  dependent: [
    {
      action: "removeFromReports",
      type: "PdfReport",
      items: [
        {id: 1, name: "Default report"}
      ]
    },
    {
      action: "delete",
      type: "PdfReport",
      items: [
        {id: 2, name: "Report with only chapter 1"}
      ]
    }
  ]
}
```

## 4.13.7 Delete

Deletes pdf-report-chapter. 404 is returned in case pdf-report-chapter does not exist or user has no permission to see the pdf-report-chapter at all. (PdfReportChapter does not exist for user.) 401 is returned if user sees the pdf-report-chapter but has no permission to delete it. 403 is returned if it is not possible to delete event report for some other reason. (Would result in inconsistency.)

**DELETE** /rest/ads/pdf-report-chapters/<id>

- @param <id> Integer(PdfReportChapter.id)
- @returns 204 header("X-FM-Message": "Deleted")
- @throws 401 header("X-FM-Message": "No permission to delete pdf-report-chapter")
- @throws 403 header("X-FM-Message": "Unable to delete pdf-report-chapter")
- @throws 404 header("X-FM-Message": "PdfReportChapter does not exist")

#### Example Usage

```
DELETE /rest/ads/pdf-report-chapters/1
```

```
Returns 204
```

## 4.14 PDF report schedules

### 4.14.1 Empty template

Returns template of PdfReportSchedule object. Template can be used as a base to fill in data and create new instance of PdfReportSchedule.

#### **GET /rest/ads/templates/pdf-report-schedule**

- @returns 200 PdfReportSchedule("Empty pdf-report-schedule")
- @throws 401 header("X-FM-Message": "No permission to manipulate pdf-report-schedules")

#### Example Usage

```
GET /rest/ads/templates/pdf-report-schedule
```

```
Returns 200
```

```
{
  "id": null,
  "report": null,
  "interval": null,
  "from": null,
  "to": null,
```

```
"sendOn": [],  
"sender": null,  
"recipients": [],  
"createdBy": "John Smith",  
"active": 1,  
"scheduled": null  
}
```

## 4.14.2 Retrieve all

Returns list of all defined pdf-report-schedules.

### GET /rest/ads/pdf-report-schedules

- @param **fields**
- @returns 204 header("X-FM-Message": "No pdf-report-schedules defined")
- @returns 200 [PdfReportSchedule]

#### Example Usage

#### GET /rest/ads/pdf-report-schedules

Returns 200

```
[  
  {  
    "id": 7,  
    "report": {  
      "id": 1,  
      "name": "Report"  
    },  
    "interval": "day",  
    "from": null,  
    "to": null,  
    "sendOn": [  
      0,  
      1,  
      2,  
    ],  
  },  
]
```

```
        3,  
        4,  
        5,  
        6  
    ],  
    "sender": "flowmon@invea.cz",  
    "recipients": [  
        "recipient@company.com"  
    ],  
    "createdBy": "John Smith",  
    "active": 1,  
    "scheduled": null  
},  
...  
]
```

### 4.14.3 Retrieve one

Returns requested PdfReportSchedule. 404 is returned if pdf-report-schedule does not exists or user has no permission to see it.

**GET /rest/ads/pdf-report-schedules/<id>**

- @param **fields**
- @returns 200 PdfReportSchedule
- @throws 404 header("X-FM-Message": "PdfReportSchedule does not exist")

Example Usage

**GET /rest/ads/pdf-report-schedules/7**

Returns 200

```
{  
    "id": 7,  
    "report": {  
        "id": 1,  
        "name": "Report"  
    },  
}
```

```
    "interval": "day",
    "from": null,
    "to": null,
    "sendOn": [
        0,
        1,
        2,
        3,
        4,
        5,
        6
    ],
    "sender": "flowmon@invea.cz",
    "recipients": [
        "recipient@company.com"
    ],
    "createdBy": "John Smith",
    "active": 1,
    "scheduled": null
}
```

## 4.14.4 Create

**POST /rest/ads/pdf-report-schedules?entity=JSON\_ENCODE(PdfReportSchedule)**

- @returns 201 PdfReportSchedule - created pdf-report-schedule
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

### Example Usage

```
POST /rest/ads/pdf-report-schedules?entity=JSON_ENCODE({
    "report": 1,
    "interval": "day",
    "from": null,
    "to": null,
    "sendOn": [0,1,2,3,4,5,6],
    "sender": "flowmon@invea.cz",
```



```
"recipients": [
    "recipient@company.com"
],
"createdBy": "John Smith",
"active": 1
})

Returns 201
{
    "id": 2,
    "report": {
        "id": 1,
        "name": "Report"
    },

    "interval": "day",
    "from": null,
    "to": null,
    "sendOn": [0,1,2,3,4,5,6],
    "sender": "flowmon@invea.cz",
    "recipients": [
        "recipient@company.com"
    ],
    "active": 1
}
```

## 4.14.5 Update

**PUT /rest/ads/pdf-report-schedules?entity=JSON\_ENCODE(PdfReportSchedule)**

- @returns 200 PdfReportSchedule - updated pdf-report-schedule
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

### Example Usage

```
PUT /rest/ads/pdf-report-schedules?entity=JSON_ENCODE({
    "id": 2,
    "report": 1,
```

```
"interval": "day",
"from": null,
"to": null,
"sendOn": [0],
"sender": "flowmon@invea.cz",
"recipients": [
    "recipient@company.com"
],
"active": 1
})
```

Returns 200

```
{
  "id": 2,
  "report": {
    "id": 1,
    "name": "Report"
  },

  "interval": "day",
  "from": null,
  "to": null,
  "sendOn": [0],
  "sender": "flowmon@invea.cz",
  "recipients": [
    "recipient@company.com"
  ],
  "active": 1
}
```

## 4.14.6 Delete

Deletes pdf-report-schedule. 404 is returned in case pdf-report-schedule does not exist or user has no permission to see the pdf-report-schedule at all. (PdfReportSchedule does not exist for user.) 401 is returned if user sees the pdf-report-schedule but has no permission to delete it. 403 is returned if it is not possible to delete event report for some other reason. (Would result in inconsistency.)

**DELETE /rest/ads/pdf-report-schedules/<id>**

- @param **<id>** Integer(PdfReportSchedule.id)
- @returns 204 header("X-FM-Message": "Deleted")
- @throws 401 header("X-FM-Message": "No permission to delete pdf-report-schedule")
- @throws 403 header("X-FM-Message": "Unable to delete pdf-report-schedule")
- @throws 404 header("X-FM-Message": "PdfReportSchedule does not exist")

#### Example Usage

```
DELETE /rest/ads/pdf-report-schedules/1
```

```
Returns 204
```

## 4.14.7 Activate

### **PUT /rest/ads/pdf-report-schedules/<id>/activate**

- @param **<id>** Integer(PdfReportSchedule.id)
- @returns 200 PdfReportSchedule
- @throws 401 header("X-FM-Message": "No permission to activate pdf report schedule")
- @throws 403 header("X-FM-Message": "Unable to activate pdf report schedule")
- @throws 404 header("X-FM-Message": "PDF report schedule does not exist")

#### Example Usage

```
PUT /rest/ads/pdf-report-schedules/1/activate
```

```
Returns 200
```

```
{  
  "id": 1,  
  "active": 1,  
  ...  
}
```

## 4.14.8 Deactivate

**PUT /rest/ads/pdf-report-schedules/<id>/deactivate**

- @param <id> Integer(PdfReportSchedule.id)
- @returns 200 PdfReportSchedule
- @throws 401 header("X-FM-Message": "No permission to deactivate pdf report schedule")
- @throws 403 header("X-FM-Message": "Unable to deactivate pdf report schedule")
- @throws 404 header("X-FM-Message": "PDF report schedule does not exist")

### Example Usage

```
PUT /rest/ads/pdf-report-schedules/1/deactivate
```

Returns 200

```
{
  "id": 1,
  "Active": 0,
  ...
}
```

## 4.15 Perspectives

### 4.15.1 Empty template

Returns template of Perspective object. Template can be used as a base to fill in data and create new instance of Perspective.

**GET /rest/ads/templates/perspective**

- @returns 200 Perspective("Empty perspective")
- @throws 401 header("X-FM-Message": "No permission to manipulate perspectives")

#### Example Usage

**GET** /rest/ads/templates/perspective

Returns 200

```
{
  id: null,
  name: null,
  priorities: []
}
```

## 4.15.2 Retrieve all

Returns list of all defined perspectives.

**GET** /rest/ads/perspectives

- @param **fields**
- @returns 204 header("X-FM-Message": "No perspectives defined")
- @returns 200 [Perspective]

#### Example Usage

**GET** /rest/ads/perspectives

Returns 200

```
[
  {
    "id": 128,
    "name": "Test",
    "priorities": [
      {
        "methodCode": "ANOMALY",
        "value": 5,
        "nfSource": {
          "id": 4,
          "name": "Test",
          "virtual": 0,
          "active": 0
        }
      }
    ]
  }
]
```

```

    },
    "filterIn": {
      "id": 1075,
      "name": "Full -12"
    },
    "filterOut": null
  }
]
},
...
]

```

### 4.15.3 Retrieve one

Returns requested Perspective. 404 is returned if perspective does not exists or user has no permission to see it.

**GET /rest/ads/perspectives/<id>**

- @param **fields**
- @returns 200 Perspective
- @throws 404 header("X-FM-Message": "Perspective does not exist")

#### Example Usage

**GET /rest/ads/perspectives/128**

Returns 200

```

{
  "id": 128,
  "name": "Test",
  "priorities": [
    {
      "methodCode": "ANOMALY",
      "value": 5,
      "nfSource": {
        "id": 4,
        "name": "Test",

```

```
        "virtual": 0,  
        "active": 0  
    },  
    "filterIn": {  
        "id": 1075,  
        "name": "Full -12"  
    },  
    "filterOut": null  
}  
]  
}
```

## 4.15.4 Create

### **POST /rest/ads/perspectives?entity=JSON\_ENCODE(Perspective)**

- @returns 201 Perspective - created perspective
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

#### Example Usage

```
POST /rest/ads/perspectives?entity=JSON_ENCODE({  
    "name": "Test",  
    "priorities": [  
        {  
            "methodCode": "ANOMALY",  
            "value": 5,  
            "nfSource": 4,  
            "filterIn": 1075,  
            "filterOut": null  
        }  
    ]  
})
```

Returns 201

```
{  
    "id": 128,  
    "name": "Test",
```

```

    "priorities": [
      {
        "methodCode": "ANOMALY",
        "value": 5,
        "nfSource": {
          "id": 4,
          "name": "Test",
          "virtual": 0,
          "active": 0
        },
        "filterIn": {
          "id": 1075,
          "name": "Full -12"
        },
        "filterOut": null
      }
    ]
  }

```

## 4.15.5 Update

### PUT /rest/ads/perspectives?entity=JSON\_ENCODE(Perspective)

- @returns 200 Perspective - updated perspective
- @throws 401 header("X-FM-Message": "Not allowed to perform this action")
- @throws 405 ValidationResultSummary header("X-FM-Message": "Validation failed")

#### Example Usage

```

PUT /rest/ads/perspectives?entity=JSON_ENCODE(
  {
    "id": 128,
    "name": "Renamed",
    "priorities": [
      {
        "methodCode": "ANOMALY",
        "value": 5,
        "nfSource": 4,
        "filterIn": {

```



```

        "id": 1075,
        "name": "Full -12"
      },
      "filterOut": null
    }
  ]
})

```

Returns 200

```

{
  "id": 128,
  "name": "Renamed",
  "priorities": [
    {
      "methodCode": "ANOMALY",
      "value": 5,
      "nfSource": {
        "id": 4,
        "name": "Test",
        "virtual": 0,
        "active": 0
      },
      "filterIn": {
        "id": 1075,
        "name": "Full -12"
      },
      "filterOut": null
    }
  ]
}

```

## 4.15.6 Dependencies

Returns dependencies object - what would happen in case given perspective was deleted.

**GET /rest/ads/dependencies/perspectives/<id>**

- @returns 200 Dependencies for delete action

- @throws 401 header("X-FM-Message": "No permission to manipulate perspectives")
- @throws 404 header("X-FM-Message": "Perspective does not exist")

#### Example Usage

**GET** /rest/ads/dependencies/perspectives/1

Returns 200

```
{
  action: "delete",
  items: [
    {id: 128, name: "Test"}
  ],
  dependent: [
    {
      action: "delete",
      type: "Report",
      items: [
        {id: 1, name: "Pdf report for Test perspective"}
      ]
    }
  ]
}
```

## 4.15.7 Delete

Deletes perspective. 404 is returned in case perspective does not exists or user has no permission to see the perspective at all. (Perspective does not exist for user.) 401 is returned if user sees the perspective but has no permission to delete it. 403 is returned if it is not possible to delete event report for some other reason. (Would result in inconsistency.)

**DELETE** /rest/ads/perspectives/<id>

- @param <id> Integer(Perspective.id)
- @returns 204 header("X-FM-Message": "Deleted")
- @throws 401 header("X-FM-Message": "No permission to delete perspective")

- @throws 403 header("X-FM-Message": "Unable to delete perspective")
- @throws 404 header("X-FM-Message": "Perspective does not exist")

#### Example Usage

```
DELETE /rest/ads/perspectives/128
```

```
Returns 204
```

## Contacts

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

Web: [www.invea.com](http://www.invea.com)  
Email: [info@invea.com](mailto: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](mailto: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.