

INFOWATCH

InfoWatch Traffic Monitor SDK. Access to InfoWatch Traffic Monitor events for external systems (DataExport API SDK)

06/08/2021

© JSC “InfoWatch”

Tel/Fax +7 (495) 229-00-22

http://www.infowatch.com

Table of Contents

1 Purpose of DataExport API 4

2 Aauthorization to access DataExport API 5

2.1 Work with plugins 5

2.1.1 Adding plugin 5

2.1.2 Plugin removal 5

2.2 Work with tokens 5

2.2.1 Adding token 5

2.2.2 Token's data editing 6

2.2.3 Token's value update 6

2.2.4 Copying token 6

2.2.5 Deleting token 6

3 DataExport API functionality 7

4 DataExport API third-party component registration 9

4.1 DataExport API third-party component registry file format 9

5 Functional description of DataExport API application programming interface. 13

5.1 GET/POST requests common headers for DataExport API 13

5.1.1 Authorization 13

5.1.2 Versions 13

5.1.3 Licenses 14

5.2 Retrieving audit data 15

5.3 Getting organizational structure 21

5.3.1 Getting information on groups 21

5.3.1.1 Getting the list of groups 21

5.3.1.2 Getting statistics on groups 26

5.3.1.3 Getting the list with group contact types 28

5.3.2 Getting information on a person 28

5.3.2.1 Getting images of multiple persons 28

5.3.2.2 Getting person image 29

5.3.2.3 Getting the list of persons 29

5.3.2.4 Getting the list of person statuses 54

5.3.2.5 Getting the list of person contact types 56

5.3.2.6 Gettins statistics on persons 58

5.3.3 Getting information on workstations 60

5.3.3.1 Getting the list of workstations 60

5.3.3.2 Getting the list of workstation statuses 65
5.3.3.3 Getting the list of workstation contact types 65
5.3.3.4 Getting statistics on workstations 65
5.4 Getting events 67
5.4.1 Getting text extracted from multiple events 67
5.4.2 Getting extracted text of event 68
5.4.3 Getting event metadata 70
5.4.4 Getting multiple event contents 70
5.4.5 Getting intercepted file of event 72
5.4.6 Getting the list of titles 72
5.4.7 Getting list of security objects catalogues 81
5.4.8 Getting the list of protected objects 83
5.4.9 Getting the list of policies 86
5.4.10 Getting the List of Events 88
5.4.11 Getting tags list 90
5.4.12 Getting statistics on events 93
5.5 Retrieving the list of Console users 95

1 Purpose of DataExport API

This application programming interface is intended to integrate InfoWatch Traffic Monitor (Traffic Monitor) with external data receiving systems. As a part of DataExport API implementation, external systems get the content and metadata of events, organizational structure data, lists of the Management Console users, and audit data of their actions from Traffic Monitor. DataExport API supports only data reading operation from Traffic Monitor.

2 Aauthorization to access DataExport API

Authorization to access DataExport API is provided via tokens mechanism, similar to other Traffic Monitor API's. To get token and review DataExport API licensing status it is required adding new type of plugin into the System - Plugin for DataExport API. All the activities related to plugins are performed within the Traffic Monitor console.

2.1 Work with plugins

2.1.1 Adding plugin

To add plugin:

1. Go to **Administration - Plugins**;
2. Within the area **Plugins** click .
3. Click **Add**.
4. Review the plugin description and click **Install**.

> **Note:**
>
> To update a plugin, the updated version of this plugin should be added.

2.1.2 Plugin removal

To remove plugin:

1. Go to **Administration - Plugins**;
2. Within the area **Plugins** click 
3. Click **Yes** to confirm the deletion.

2.2 Work with tokens

Token is generated automatically after the plugin has been added. All the activities related to tokens are performed within the InfoWatch Traffic Monitor console.

2.2.1 Adding token

To add token:

1. Go to **Administration - Plugins**;
2. Choose the plugin within the **Plugins** area;
3. Go to **Tokens** bookmark;
4. Click .

Note:

Token's value is generated automatically when setting up a token and it cannot be set by user.

2.2.2 Token's data editing

To change token's data:

1. Go to **Administration - Plugins**;
2. Choose the plugin within the **Plugins** area;
3. Go to **Tokens** bookmark;
4. By double left button click, enter the required info within the fields *Name* and *Description*.

2.2.3 Token's value update

If the token was compromised or if there is a potential token leak to the third party, it is required generating a new token's value.

To update token's value:

1. Go to **Administration - Plugins**;
2. Choose the plugin within the **Plugins** area;
3. Go to **Tokens** bookmark;
4. Click  .

2.2.4 Copying token

To copy token's value into the clipboard:

1. Go to **Administration - Plugins**;
2. Choose the plugin within the **Plugins** area;
3. Go to **Tokens** bookmark;
4. Click  .

2.2.5 Deleting token

To delete token:

1. Go to **Administration - Plugins**;
2. Choose the plugin within the **Plugins** area;
3. Go to **Tokens** bookmark;
4. Click  .

3 DataExport API functionality

The functionality of DataExport API includes the following:

- Ability to specify the number of attributes to be included within the result:
    - Event ID;
    - Senders: list of contacts and authentication result, taking into account the potential conflicts;
    - Recipients: list of contacts, and authentication result, taking into account the potential conflicts;
    - Sender's computer, taking into account the potential conflict;
    - Type of event;
    - Date and time of interception;
    - Protocol;
    - Level of violation;
    - Verdict;
    - Path to file or receiver address;
    - Path to file or sender address;
    - User's decision;
    - Path to file;
    - Web resource address;
    - Security objects and catalogues;
    - Event's size;
    - Tags;
    - Set of user attributes logged via plugin;
    - Set of system titles for events;
    - Date and time of event pasting into DB;
    - Name of perimeter where the event has entered;
    - Name of perimeter left by event;
    - List of policies names (including policy ID), triggered by the intercepted object;
    - Group of rules;
    - Interception server;
    - IP interception server;
    - Receiver device name;
    - Sender device name;
    - E-mail subject;
    - Attachment;
    - Mandate level;
    - Mandate category.
- Ability to specify the number of attributes to filtrate:
    - Date and time of interception;
    - Type of event;
    - Protocol;
    - Security objects and catalogues;
    - User's decision;
    - Event's size;
    - Tags;
    - Date and time of event pasting into DB;
    - Group of rules;
    - Level of violation;

    - Verdict;
    - Data and time of the logged user action;
    - Type of the object the action applied to.
- Getting separate event metadata being able to specify the number of attributes to be included into the result. In this regard, if the event contains few enclosed objects, all the extracted text to be assembled into one file;
- Getting all the extracted text of an event;
- Getting a full list of system titles;
- Getting binary data of a separate event;
- Getting the list of the Management Console users;
- Retrieving user audit data:
    - Record ID in auditing Console user sessions;
    - Data and time of the logged user action in UTC;
    - ID of the user who performed the action;
    - Full name of the user who performed the action;
    - Type of the action the user has taken;
    - Type of the object the action applied to;
    - ID of the object the action applied to;
    - Name of the object the action applied to;
    - Description of changes with old and new values.

4 DataExport API third-party component registration

To integrate third-party receiving system with InfoWatch Traffic Monitor the following actions are to be taken:

1. Register the authentication type string of company - the component user in Infowatch JSC. Thus letting to connect a user company to the license, to be installed within Traffic Monitor System;
2. Obtain the developer's license related to company's ID. This would let unloading the events data from Traffic Monitor;
3. Create third-party component registry file in accordance with the format below. The license, obtained as per the point 2 is to be added to the file;
4. Download the third-party component registry file to InfoWatch Traffic Monitor System using the Traffic Monitor console interface. Section "Administration" - "Plugins". (see "Traffic Monitor. User Guide");
5. Obtain an access token within the downloaded third-party component's properties;

The System is ready to send the company's events data.

4.1 DataExport API third-party component registry file format

Plugin is a .zip format archive. The archive includes:

- **licenses** folder, containing license files;
- **icon** folder, containing the files with pictograms used for registered events;
- **manifest.json** file, containing information on a plugin.

> **Note:**
>
> Folders **licenses** and **icon** are optional. License files and files with pictograms can be located in the root folder.

For external systems, receiving data from Traffic Monitor **manifest.json** file should contain the following information:

Content	Type of data	Optional	Description
`{`			
`"PLUGIN_ID": "",`	String	Yes	Unique plugin identifier (UUID) created by developer. It may include up to 40 characters. The identificator is used for plugin update. **Example:** `"PLUGIN_ID": "189C38D390396EB6E0530100007F1CA200000001",`
`"DISPLAY_NAME": "",`	String	Yes	Plugin name displayed. **Example:** `"DISPLAY_NAME": "Plugin name",`
`"DESCRIPTION": "",`	String	No	Plugin description displayed. **Example:** `"DESCRIPTION": "Test plugin to demonstrate",`

Content	Type of data	Optional	Description
`"VERSION": "",`	String	Yes	Plugin version. **Example** `"VERSION": "1.0.0",`
`"VENDOR": "",`	String	Yes	Software company ID in accordance with the company name specified in the license. **Example** `"VENDOR": "infowatch",`

`"LICENSE": [ { "PATH":"" } ],`	Array
	String
		Yes	License files. The paths to license files should be specified in accordance with archive root folder.
			Example:
			`"LICENSE": [`
			`{`
			`"PATH":"licenses/license file 1.license"`
			`},`
			`{`
			`"PATH":"licenses/license file 2.license"`
			`},`
			`],`

`"PATTERN_SEARCH_LICENSE": "",`	String	No	Template to search previously downloaded licenses to be binded to plugin. It has the following format: `{operator:or|and,child[{name:value}]}`. **Example:** `"PATTERN_SEARCH_LICENSE": "{\"operator\":\"or\", \"conditions\":[{\"origin\":\"dm\"},{\"origin\": \"dmmobile\"}]}",`

`"DATA_RECEIVER": [ "" ]`	Array
	String
		Yes	List of identificators for systems receiving data from Traffic Monitor.
			Example: `"DATA_RECEIVER": [`
			`"Events reciever1",`
			`"Events reciever2"`
			`]`

Content	Type of data	Optional	Description

"OBJECT_HEADER": [
{
"NAME": "",
 	
"NOTE": {
"": "",
"": ""
},
"TYPE": "",
"FORMAT": "",
 	
"DATA_CLASS": [""],
"USE_IN_POLICY": "",
"USE_IN_NOTIFICATION": "",
"USE_IN_QUERY": "",
"USE_IN_LIST": "",
"USE_IN_SHOW": "",
"USE_IN_DETAIL": "",
"IS_MULTIPLE_VALUE": "",
}
]
	Object
	
	Array
	
	Object
	String
	
	Enum
	Enum
	
	Array
	Enum
	Enum
	
	Enum
	Enum
	Enum
	Enum
	Enum
		No	User events attributes.
			
			Unique attribute's code to be used in the events headers and to be identified in xAPI/PushAPI.
			
			If developer's ID is not "IW", the code should start with "`<Software company's ID>_`" prefix.
			
			For example: CISCO_CALL_DURATION
			
			Attribute's name, displayed on Traffic Monitor management console.
			
			Type of value.
			Recording format. Possible values: number (integer, fractional), string, date and time UTC + specifying the time zone shifting, duration, hyperlink, listing, logic type.
			Services, which events will be equipped with new user attribute.
			Using attribute in policies. `"1"` - yes, `"0"` - no.
			Using attribute in notifications. `"1"` - yes, `"0"` - no.
			
			Using attribute in queries. `"1"` - yes, `"0"` - no.
			Displaying attribute in tabular event viewing mode. `"1"` - yes, `"0"` - no.
			Displaying attribute in brief event viewing form. `"1"` - yes, `"0"` - no.
			Displaying attribute in detailed event viewing form. `"1"` - yes, `"0"` - no.
			Multiple value. `"1"` - yes, `"0"` - no.

Content	Type of data	Optional	Description

			"OBJECT_HEADER": [
			{
			"NAME": "header_multi_string",
			"NOTE": {
			"rus": "Заголовок строк1",
			"eng": "Header strings1"
			},
			"DATA_CLASS": ["kEmail"],
			"USE_IN_POLICY": "1",
			"USE_IN_QUERY": "1",
			"USE_IN_NOTIFICATION": "1",
			"USE_IN_LIST": "1",
			"USE_IN_SHOW": "1",
			"USE_IN_DETAIL": "1",
			"TYPE": "string",
			"FORMAT": "string",
			"IS_MULTIPLE_VALUE": "1"
			},
			{
			"NAME": "header_date",
			"NOTE": {
			"rus": "Заголовок дата1",
			"eng": "Header date1"
			},
			"DATA_CLASS": ["kEmail"],
			"USE_IN_POLICY": "1",
			"USE_IN_QUERY": "1",
			"USE_IN_NOTIFICATION": "1",
			"USE_IN_LIST": "0",
			"USE_IN_SHOW": "1",
			"USE_IN_DETAIL": "1",
			"TYPE": "string",
			"FORMAT": "date",
			"IS_MULTIPLE_VALUE": "0"
			}
			]

}			

5 Functional description of DataExport API application programming interface

This page has not yet been translated to английский.

5.1 GET/POST requests common headers for DataExport API

5.1.1 Authorization

Token received for Public API should be specified within **X-API-Auth-Token** header in each API request. The token can be canceled, blocked/deleted via IWTM interface. In this case, the token is considered to be invalid. When the token is invalid, **401** error message will be returned.

Error message example when token is invalid

```
{
    "error": {
        "code": "unauthorized"
    }
}
```

If the token is valid, but API user has no sufficient access rights, **403** error message will be returned.

Error message when user has no sufficient access rights

```
{
    "error": {
        "code": "forbidden"
    }
}
```

5.1.2 Versions

API version is sent via **X-API-Version** header and can be in one of the following states:

- supported — the version is supported and actively developing
- deprecated — the version is available but no longer supported. We recommend upgrading to a newer version as this version can be removed in future releases
- removed — the version is completely removed from the product

In case the server does not support the version, **400** error message is returned.

Response if the version is incorrect

```
{
    "error": {
        "code": "unsupported_version",
        "meta": {
```

```
            "versions": [
                {
                    "NAME":   "1",
                    "STATUS": "removed"
                },
                {
                    "NAME":   "1.1",
                    "STATUS": "deprecated"
                },
                {
                    "NAME":   "1.2",
                    "STATUS": "supported"
                },
                {
                    "NAME":   "1.3",
                    "STATUS": "supported"
                },
                {
                    "NAME":   "1.4",
                    "STATUS": "supported"
                }
            ]
        }
    }
}
```

5.1.3 Licenses

To operate properly, the module requires a license as follows:

```
{
    "common_name": "iw",
    "importer_name": "dome"
}
```

Where `common_name` — is a name of a licensed company and `importer_name` — is a product, receiving data from TM. **X-API-CompanyId** and **X-API-ProductName** headers should be sent in each request. In case the license is rejected **403** error message will be returned.

Respond when the license is invalid.

```
{
    "error": {
        "code": "licence_not_valid"
    }
}
```

Thus, each request to the server should contain the following obligatory headings:

- X-API-Version – a version of the programming interface is specified. Current version: **1.4**;
- X-API-CompanyId – an identifier of the vendor is specified, e.g. **iw**;
- X-API-ImporterName – data receiving product is specified, e.g. **dome**;
- X-API-Auth-Token – the authorization token is specified.

> **Warning!**
>
> In 6.10.10 version X-API-Version program interface number is **1.2**

Example of request

```
X-API-Version: 1.4
X-API-CompanyId: iw
X-API-ProductName: dome
X-API-Auth-Token: gla20dos1pcdbr69caxz
```

5.2 Retrieving audit data

Description: List of audit data sorted by date and time of logged user actions.

Method: `GET`

Resource: `audit`

Parameters:

- `filter` - filter list:
  - `CHANGE_DATE[>]` – start date and time of the logged user action. Are specified in UNIX timestamp format with microtime dot, for example, `filter[CHANGE_DATE][>]=1591368000.123456`
  - `CHANGE_DATE[<]` – end date and time of the logged user action. Are specified in UNIX timestamp format with microtime dot, for example, `filter[CHANGE_DATE][<]=1591368000.123456`
  - `CHANGE_DATE[>=]` – start date and time of the logged user action. Are specified in UNIX timestamp format with microtime dot, for example, `filter[CHANGE_DATE][>=]=1591368000.123456`
  - `CHANGE_DATE[<=]` – end date and time of the logged user action. Are specified in UNIX timestamp format with microtime dot, for example, `filter[CHANGE_DATE][<=]=1591368000.123456`
  - `ENTITY_TYPE[]` – the type of the entity to which the action applied (several values are possible)
- `without[]` – get data without additional attributes
  - PROPERTY_CHANGES – data of extended audit
- `sort[]` – parameter used for sorting
  - `change_date` – sort by date of `asc` change
- `start` – a position from which audit data should be delivered (the default value is 0)
- `limit` – how many audit data sets should be delivered (the default value is 100, the value cannot be larger than 1000)

OpenAPI scheme for delivering audit data

```
 schemas:
    auditCommon:
      type: object
      additionalProperties: false
      required: [AUDIT_LOG_ID, USER_ID, CHANGE_DATE, OPERATION, ENTITY_TYPE, ENTITY_ID,
ENTITY_DISPLAY_NAME]
      properties: &common
        AUDIT_LOG_ID:
          type: integer
          description: Audit log id
        USER_ID:
          type: integer
          nullable: true
          description: |
            User id who performed the action (See /xapi/user).
            USER_ID may be null on login or if changes make by plugin.
        CHANGER_NAME:
          type: string
          nullable: true
          description: |
            User name or Plugin name who performed the action.
            May be null on login.
        CHANGE_DATE:
          type: string
          description: Date and time of the registered user action with format "Y-m-d H:i:s.u" in UTC
          example: "2020-05-14 13:53:52.926560"
        OPERATION:
          type: string
          description: Operation type over which the action was performed. There may be values not from the
enum.
          enum:
            - run
            - canceled
            - start
            - stop
            - restart
            - view
            - create
            - update
            - change
            - delete
            - delete_ref
            - delete_hash
            - copy
            - move
            - login_failure
            - login
            - logout
            - change_password
            - commit
            - rollback
            - draft
            - add
            - import
            - export
```

```
            - export_object
            - download
            - sync
            - decision_update
            - add_tag
            - remove_tag
            - add_role
            - remove_role
            - add_visibility_area
            - remove_visibility_area
            - logging
            - apply_inbound
            - xapi_decision_update
            - start_scan_job
            - apply_result_scan_job
            - update_job_schedule
            - create_log
            - plugin_register
            - plugin_update
            - privileges_update
            - privileges_create
            - notification
            - notification_template
            - notification_test
        ENTITY_TYPE:
          type: string
          description: Entity type over which the action was performed. There may be values not from the
enum.
          enum: *entryTypes
        ENTITY_ID:
          type: string
          nullable: true
          description: |
            ID of the entity which the action was performed. May be GUID or integer as string
            May by null for:
            | OPERATION | ENTITY_TYPE |
            | :--- | :--- |
            | commit | Config |
            | run | QueryReportRun |
            | sync | Adlibitum |
            | login | User |
            | rollback | Config |
          example: "DBAF4F834F7647CBB297BC9DD17B77B100000000"
        ENTITY_DISPLAY_NAME:
          type: string
          nullable: true
          description: The name of the entity over which the action was carried out
          example: "Some name"
        PROPERTY_CHANGES:
          type: object
          nullable: true
          description: Null in default Audit log events
    userLoginLogout:
      type: object
      description: For ENTITY_TYPE User and OPERATION login or logout
      properties:
        <<: *common
```

```
        OPERATION:
          type: string
          enum: [login, logout]
        ENTITY_TYPE:
          type: string
          enum: [User]
        PROPERTY_CHANGES:
          type: object
          required: [request]
          properties:
            request:
              required:
                - hostname
                - ip
                - result
                - login
              properties:
                hostname:
                  type: string
                  title: Hostname
                  description: Hostname from which authorization passed
                ip:
                  type: string
                  title: Ip address
                  description: Ip address from which authorization passed
                result:
                  type: string
                  title: Result
                  description: Result
                  enum:
                    - login_failure
                    - login
                    - logout
                login:
                  type: string
                  title: User login
                  description: Entered user login in input
    policyPropertyChanges:
      type: object
      description: For ENTITY_TYPE Policy
      nullable: true
      properties:
        <<: *common
        ENTITY_TYPE:
          type: string
          enum: [Policy]
        PROPERTY_CHANGES:
          type: object
```

Example:

Query

```
curl -X GET "https://qa-2944.infowatch.ru/xapi/audit?sort[CHANGE_DATE]=asc&filter[ENTITY_TYPE]
[]=Adlibitum&filter[ENTITY_TYPE][]=Audit&filter[ENTITY_TYPE][]=Config&filter[ENTITY_TYPE]
```

```
[]=Perimeter&filter[ENTITY_TYPE][]=Policy&start=0&limit=5" -H "accept: application/json" -H "X-Enable-OpenAPI: 1"
```

Response


```
{
  "data": [
    {
      "AUDIT_LOG_ID": 0,
      "USER_ID": 0,
      "CHANGER_NAME": "string",
      "CHANGE_DATE": "2020-05-14 13:53:52.926560",
      "OPERATION": "run",
      "ENTITY_TYPE": "Adlibitum",
      "ENTITY_ID": "DBAF4F834F7647CBB297BC9DD17B77B100000000",
      "ENTITY_DISPLAY_NAME": "Some name",
      "PROPERTY_CHANGES": {}
    },
    {
      "AUDIT_LOG_ID": 0,
      "USER_ID": 0,
      "CHANGER_NAME": "string",
      "CHANGE_DATE": "2020-05-14 13:53:52.926560",
      "OPERATION": "login",
      "ENTITY_TYPE": "User",
      "ENTITY_ID": "DBAF4F834F7647CBB297BC9DD17B77B100000000",
      "ENTITY_DISPLAY_NAME": "Some name",
      "PROPERTY_CHANGES": {
        "request": {
          "hostname": "string",
          "ip": "string",
          "result": "login_failure",
          "login": "string"
        }
      }
    },
    {
      "AUDIT_LOG_ID": 0,
      "USER_ID": 0,
      "CHANGER_NAME": "string",
      "CHANGE_DATE": "2020-05-14 13:53:52.926560",
      "OPERATION": "run",
      "ENTITY_TYPE": "Policy",
      "ENTITY_ID": "DBAF4F834F7647CBB297BC9DD17B77B100000000",
      "ENTITY_DISPLAY_NAME": "Some name",
      "PROPERTY_CHANGES": {
        "old": {
          "DISPLAY_NAME": "string",
          "NOTE": "string",
          "STATUS": 0,
          "START_DATE": "string",
          "END_DATE": "string",
          "DATA": {
            "ITEMS": [
              {
                "TYPE": "catalog",
```

```
              "ID": "string",
              "NAME": "string"
            },
            {
              "TYPE": "fileformat",
              "ID": "string",
              "NAME": "string",
              "MIME_TYPE": "string",
              "ENCRYPTED": 0,
              "MERGED": 0,
              "MAX": 0,
              "MIN": 0,
              "MAX_TYPE": 0,
              "MIN_TYPE": 0
            }
          ]
        }
      },
      "new": {
        "DISPLAY_NAME": "string",
        "NOTE": "string",
        "STATUS": 0,
        "START_DATE": "string",
        "END_DATE": "string",
        "DATA": {
          "ITEMS": [
            {
              "TYPE": "catalog",
              "ID": "string",
              "NAME": "string"
            },
            {
              "TYPE": "fileformat",
              "ID": "string",
              "NAME": "string",
              "MIME_TYPE": "string",
              "ENCRYPTED": 0,
              "MERGED": 0,
              "MAX": 0,
              "MIN": 0,
              "MAX_TYPE": 0,
              "MIN_TYPE": 0
            }
          ]
        }
      },
      "rule": {
        "ENTITY_TYPE": "PolicyRule",
        "ENTITY_ID": "string",
        "ENTITY_DISPLAY_NAME": "string"
      }
    }
  }
],
"meta": {
  "totalCount": 0,
  "start": 0,
```

```
    "limit": 0
  }
}
```

5.3 Getting organizational structure

- Getting information on groups
- Getting information on a person
- Getting information on workstations

5.3.1 Getting information on groups

Getting the list of groups

Description: the list of groups

Method: `GET`

Resource: `group`

Parameters:

- `filter` — the filter list:
  - `change_date[from]` — since the date of the last change in UNIX-timestamp format (can be used separately from `change_date[to]`)
  - `change_date[to]` — up to the date of the last change in UNIX-timestamp format (can be used separately from `change_date[from]`)
  - `create_date[from]` — since the date of the last change in UNIX-timestamp format (can be used separately from `create_date[to]`)
  - `create_date[to]` — up to the date of the last change in UNIX-timestamp format (can be used separately from `create_date[from]`
  - is_deleted — flag 1 or 0. If the parameter is specified, then entities with the specified value will be returned
  - `disabled` — flag 1 or 0, indicates activity. If the flag is set, then the entity is inactive (the flag is obtained from LDAP)
  - `contact_type[]` — filtering by the contact type (several values can be passed, e.g., `/xapi/group?filter[contact_type][]=webaccount&filter[contact_type][]=icq`)
  - `contact_value[]` — filtering by the contact value (several values can be passed, e.g., `/xapi/group?filter[contact_value][]=user@ad&filter[contact_value][]=login@example.com`)
  - `parent_group_id[]` — filtering by groups (several values can be passed, e.g., `/xapi/group?filter[parent_group_id][]=FCE1409680E5D6469E1F7998266CE9E9&filter[parent_group_id][]=918462B6DC2C954BAAF8706F7DC53F14`)
- `with[]` — the list of additional entities to be added to the object:
  - `contacts` — group contacts
  - `children` — child groups

- `sort` — parameter for sorting
    - `change_date` — sort by the modification date `asc` or `desc` (e.g., `/xapi/group?sort[change_date]=asc`) ⚠️ may slow down returning the result

- `start` — a start position for returning items, the default value is 0
- `limit` — how many items to return (the default value is 100, cannot be more than 1000

Response: The list of persons described with the following JSON Schema:

JSON returning schema

```
{
    "$schema": "http://json-schema.org/draft-04/schema#" ,
    "type": "object",
    "required": [
        "data",
        "meta"
    ],
    "properties": {
        "data": {
            "description": "Array of groups",
            "type": "array",
            "items": {
                "description": "Group",
                "type": "object",
                "required": [
                    "GROUP_ID",
                    "CREATE_DATE",
                    "CHANGE_DATE",
                    "DISPLAY_NAME",
                    "DISTINGUISHEDNAME",
                    "GROUP_TYPE",
                    "IS_VISIBLE",
                    "IS_DELETED",
                    "childrenCount"
                ],
                "properties": {
                    "GROUP_ID": {
                        "description": "A unique identifier of the group",
                        "type": "string"
                    },
                    "CREATE_DATE": {
                        "description": "Entry creation date",
                        "type": "string"
                    },
                    "CHANGE_DATE": {
                        "description": "Entry modification date",
                        "type": "string"
                    },
                    "DISPLAY_NAME": {
                        "description": "Group name",
                        "type": "string"
                    },
                    "DISTINGUISHEDNAME": {
                        "description": "Full name",
                        "type": "string"
                    },
                    "GROUP_TYPE": {
                        "description": "Group type (tmRoot, tmGroup, adRoot, adGroup,
adContainer, adDomain, adOU from LDAP objectclass)",
                        "type": "string"
                    },
                    "IS_VISIBLE": {
```

```
            "description": "Group visibility",
            "type": "integer"
        },
        "IS_DELETED": {
            "description": "Whether the group is deleted from TM",
            "type": "integer"
        },
        "childrenCount": {
            "description": "The number of child elements in the group",
            "type": "integer"
        },
        "contacts": {
            "description": "Array of contacts",
            "type": "array",
            "items": {
                "description": "Contact",
                "type": "object",
                "required": [
                    "CONTACT_ID",
                    "CREATE_DATE",
                    "CHANGE_DATE",
                    "CONTACT_TYPE",
                    "VALUE",
                    "IS_PERSONAL",
                    "IS_PRIMARY",
                    "SOURCE"
                ],
                "properties": {
                    "CONTACT_ID": {
                        "description": "Object unique identifier",
                        "type": "string"
                    },
                    "CREATE_DATE": {
                        "description": "Creation date",
                        "type": "string"
                    },
                    "CHANGE_DATE": {
                        "description": "Modification date",
                        "type": "string"
                    },
                    "CONTACT_TYPE": {
                        "description": "Type",
                        "type": "string"
                    },
                    "VALUE": {
                        "description": "Value",
                        "type": "string"
                    },
                    "IS_PERSONAL": {
                        "description": "Belongs to (Work, Personal)",
                        "type": "integer"
                    },
                    "IS_PRIMARY": {
```

```
                                    "description": "Is primary",
                                    "type": "integer"
                                },
                                "SOURCE": {
                                    "description": "Type",
                                    "type": "string"
                                }
                            }
                        }
                    },
                    "children": {
                        "description": "An array of groups which belong to the
current group",
                        "type": "array",
                        "items": {
                            "description": "Group",
                            "type": "object",
                            "properties": {
                                "GROUP_ID": {
                                    "description": "A unique identifier of the group"
,
                                    "type": "string"
                                },
                                "PARENT_GROUP_ID": {
                                    "description": "A unique identifier of the parent
group",
                                    "type": "string"
                                },
                                "RELATION_TYPE": {
                                    "description": "Connection type: 0 - tm, 1 -
member, 2 - distinguishedname",
                                    "type": "integer"
                                }
                            }
                        }
                    }
                }
            },
            "meta": {
                "description": "Additional data",
                "type": "object",
                "required": [
                    "totalCount",
                    "start",
                    "limit"
                ],
                "properties": {
                    "totalCount": {
                        "type": "integer",
                        "description": "Total number"
                    },
                    "start": {
                        "type": "integer",
```

```
                        "description": "A position from which returning started"
                    },
                    "limit": {
                        "type": "integer",
                        "description": "Total number of returned items"
                    }
                }
            }
        }
    }
```

Example:

Query

```
/xapi/group?filter[change_date][from]=1&filter[change_date][to]=2&with[]=contacts&
with[]=children&limit=10&sort[change_date]=asc
```

Getting statistics on groups

Description: Statistics on groups

Method: `GET`

Resource: group `/stat`

Parameters:

- `filter` — the filter list:
  - `change_date[from]` — since the date of the last change in UNIX-timestamp format (can be used separately from `change_date[to]`)
  - `change_date[to]` — up to the date of the last change in UNIX-timestamp format (can be used separately from `change_date[from]`)

Response:

JSON returning schema

```
{
    "$schema": "http://json-schema.org/draft-04/schema#" ,
    "description": "Statistics",
    "type": "object",
    "properties": {
        "data": {
            "description": "Not used",
            "type": "array",
            "items": {}
        },
        "meta": {
            "description": "Statistics",
            "type": "object",
            "properties": {
                "CHANGE_DATE": {
                    "description": "Modification date",
                    "type": "object",
                    "properties": {
                        "from": {
                            "description": "The first modification date",
                            "type": "string"
                        },
                        "to": {
                            "description": "The last modification date",
                            "type": "string"
                        }
                    }
                },
                "totalCount": {
                    "description": "Total number of items",
                    "type": "integer"
                }
            }
        }
    }
}
```


Example:

```
GET /xapi/group/stat?filter[CAPTURE_DATE][to]=1494951762&filter[CAPTURE_DATE][from]=1494951000
```


```
{
  "data": [],
  "meta": {
    "CHANGE_DATE": {
      "from": "2017-03-24 10:40:03",
      "to": "2017-05-24 12:10:22"
    },
    "totalCount": 539693
  }
}
```

Getting the list with group contact types

Description: The list of all possible types of the group contacts

Method: `GET`

Resource: `group/contact_type`

- `start` — a start position for returning items, the default value is 0
- `limit` — how many items to return (the default value is 100, cannot be more than 1000

Example:

Query

```
GET /xapi/group/contact_type?start=0&limit=10
```

5.3.2 Getting information on a person

Getting images of multiple persons

Method: POST

Resource: `person/images`

Parameters (are passed in the form of JSON):

- `response_type` — the response type:
  - chunked — `Transfer-Encoding: chunked` is used, where each image is passed in its own block and in the order which is specified in the request body; if there is no image, then null character "\0" is passed instead
  - `multipart/mixed` — `multipart/mixed` [rfc tech-ms], format is used. This format is not part of the HTTP standard, however, it is used in emails, so it can be easily implemented. Besides, it is very similar to `multipart/form-data`. Each image is passed in its block and is separately marked with X-Person-ID title indicating the event the image belongs to. If there is no image, then it is either not passed or the image body is missing
- `person_id` — an array of person IDs

Examples:

chunked :

Query

```
POST /xapi/person/images
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(пустая строка)
{"response_type": "chunked", "person_id": ["DEF1CC1FA015CA428D5DE3CD5B804B69",
"E277B624BC988C4C96590A8435F634ED"]}
```

Response

```
HTTP/1.1 200 OK
Content-Type: image/png
Transfer-Encoding: chunked
(empty line)
1
\0 (no-image character for person DEF1CC1FA015CA428D5DE3CD5B804B69)
48AC
(image content for person E277B624BC988C4C96590A8435F634ED)
0
```

multipart/mixed:

```
POST /xapi/person/images
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(empty line)
{"response_type": "multipart/mixed", "person_id": [
"DEF1CC1FA015CA428D5DE3CD5B804B69", "E277B624BC988C4C96590A8435F634ED"]}
```

Response

```
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=Asrf456BGe4h
Content-Length: (total volume including titles, can be missing!)
(empty line)
(missing preamble)
--Asrf456BGe4h
Content-Type: application/x-empty
Content-Disposition: attachment; filename="DEF1CC1FA015CA428D5DE3CD5B804B69"
X-Person-ID: DEF1CC1FA015CA428D5DE3CD5B804B69
(empty line and zero content after)
--Asrf456BGe4h
Content-Disposition: attachment; filename="E277B624BC988C4C96590A8435F634ED.jpg"
Content-Type: image/jpeg
X-Person-ID: E277B624BC988C4C96590A8435F634ED
(empty line)
(binary contents of the second person E277B624BC988C4C96590A8435F634ED)
--Asrf456BGe4h--
(missing epilogue - empty line)
```

Getting person image

Method: `GET`

Resource: `person/image/{person_id}`

Response: The person image

Getting the list of persons

Description: The list of persons

Method: `GET`

Resource: `person`

Parameters:

- `filter` — the filter list:
  - `change_date[from]` — since the date of the last change in UNIX-timestamp format (can be used separately from `change_date[to]`)
  - `change_date[to]` — up to the date of the last change in UNIX-timestamp format (can be used separately from `change_date[from]`)
  - `create_date[from]` — since the date of the last change in UNIX-timestamp format (can be used separately from `create_date[to]`)
  - `create_date[to]` — up to the date of the last change in UNIX-timestamp format (can be used separately from `create_date[from]`
  - is_deleted — flag 1 or 0. If the parameter is specified, then entities with the specified value will be returned
  - `disabled` — flag 1 or 0, indicates activity. If the flag is set, then the entity is inactive (the flag is obtained from LDAP)
  - `contact_type[]` — filtering by the contact type (several values can be passed, e.g., `/xapi/person?filter[contact_type][]=webaccount&filter[contact_type][]=icq`)
  - `contact_value[]` — filtering by the contact value (several values can be passed, e.g., `/xapi/person?filter[contact_value ][]=user@ad&filter[contact_value ][]=login@example.com )`
  - `group_id[]` — filtering by groups (several values can be passed, e.g.,: `/xapi/person?filter[group_id][]=webaccount&filter[group_id][]=icq`)
- `with[]` — the list of additional entities to be added to the object:
  - `contacts` — person contact
  - `p2g` — connection of the person with groups
  - `status` — person statuses
- `sort` — parameter for sorting
  - `change_date` — sort by the modification date `asc` or `desc` (e.g. `/xapi/person?sort[change_date]=asc`) ⚠️ may slow down returning the result
- `start` — a start position for returning items, the default value is 0
- `limit` — how many items to return (the default value is 100, cannot be more than 1000

Response: The list of persons described with the following JSON Schema:

JSON schema of returning the list of persons

```
{
    "$schema": "http://json-schema.org/draft-04/schema#" ,
    "type": "object",
    "required": [
        "data",
        "meta"
    ],
    "properties": {
        "data": {
            "description": "Array of perons",
            "type": "array",
            "items": {
                "description": "Person",
                "type": "object",
                "required": [
                    "PERSON_ID",
                    "CREATE_DATE",
                    "CHANGE_DATE",
                    "DISTINGUISHEDNAME",
                    "DISPLAY_NAME",
                    "GIVENNAME",
                    "SN",
                    "DISABLED",
                    "IS_EMPLOYEE",
                    "IS_DELETED"
                ],
                "properties": {
                    "PERSON_ID": {
                        "description": "Object unique identifier",
                        "type": "string"
                    },
                    "CREATE_DATE": {
                        "description": "Creation date",
                        "type": "string"
                    },
                    "CHANGE_DATE": {
                        "description": "Last modification date",
                        "type": "string"
                    },
                    "DISTINGUISHEDNAME": {
                        "description": "Full name",
                        "type": "string"
                    },
                    "DISPLAY_NAME": {
                        "description": "First and last names",
                        "type": "string"
                    },
                    "GIVENNAME": {
                        "description": "First name",
                        "type": "string"
                    },
                    "SN": {
                        "description": "Last name",
```

```
            "type": "string"
        },
        "DISABLED": {
            "description": "Person state",
            "type": "integer"
        },
        "IS_EMPLOYEE": {
            "description": "If the person is the company employee",
            "type": "integer"
        },
        "IS_DELETED": {
            "description": "If the person is deleted from TM",
            "type": "integer"
        },
        "contacts": {
            "description": "Array of contacts",
            "type": "array",
            "items": {
                "description": "Contact",
                "type": "object",
                "required": [
                    "CONTACT_ID",
                    "CREATE_DATE",
                    "CHANGE_DATE",
                    "CONTACT_TYPE",
                    "VALUE",
                    "IS_PERSONAL",
                    "IS_PRIMARY",
                    "SOURCE"
                ],
                "properties": {
                    "CONTACT_ID": {
                        "description": "Object unique identifier",
                        "type": "string"
                    },
                    "CREATE_DATE": {
                        "description": "Creation date",
                        "type": "string"
                    },
                    "CHANGE_DATE": {
                        "description": "Modification date",
                        "type": "string"
                    },
                    "CONTACT_TYPE": {
                        "description": "Type",
                        "type": "string"
                    },
                    "VALUE": {
                        "description": "Value",
                        "type": "string"
                    },
                    "IS_PERSONAL": {
                        "description": "Belong to (Work, Personal)",
```

```
                                "type": "integer"
                            },
                            "IS_PRIMARY": {
                                "description": "Is primary",
                                "type": "integer"
                            },
                            "SOURCE": {
                                "description": "Type",
                                "type": "string"
                            }
                        }
                    }
                },
                "p2g": {
                    "description": "Array of connection with groups",
                    "type": "array",
                    "items": {
                        "description": "Connection with group",
                        "type": "object",
                        "required": [
                            "PERSON_ID",
                            "PARENT_GROUP_ID",
                            "RELATION_TYPE"
                        ],
                        "properties": {
                            "PERSON_ID": {
                                "description": "Person unique identifier",
                                "type": "string"
                            },
                            "PARENT_GROUP_ID": {
                                "description": "Group unique identifier",
                                "type": "string"
                            },
                            "RELATION_TYPE": {
                                "description": "Connection type: 0 - tm, 1 -
member, 2 - distinguishedname, 3 - primarygroupsid",
                                "type": "integer"
                            }
                        }
                    }
                },
                "status": {
                    "description": "Array of statuses",
                    "type": "array",
                    "items": {
                        "description": "Person status",
                        "type": "object",
                        "required": [
                            "IDENTITY_STATUS_ID",
                            "CREATE_DATE",
                            "CHANGE_DATE",
                            "ASSIGNMENT_DATE",
                            "DISPLAY_NAME",
```

```
                                "COLOR",
                                "EDITABLE"
                            ],
                            "properties": {
                                "IDENTITY_STATUS_ID": {
                                    "description": "Status unique identifier",
                                    "type": "string"
                                },
                                "CREATE_DATE": {
                                    "description": "Status creation date",
                                    "type": "string"
                                },
                                "CHANGE_DATE": {
                                    "description": "Status modification date",
                                    "type": "string"
                                },
                                "ASSIGNMENT_DATE": {
                                    "description": "The date when the status was
assigned to the person",
                                    "type": "string"
                                },
                                "DISPLAY_NAME": {
                                    "description": "Status name",
                                    "type": "string"
                                },
                                "COLOR": {
                                    "description": "Status color (e.g. #ff0000)",
                                    "type": "string"
                                },
                                "EDITABLE": {
                                    "description": "Indicator that the status is not
system",
                                    "type": "integer"
                                }
                            }
                        }
                    }
                }
            },
            "meta": {
                "description": "Additional data",
                "type": "object",
                "required": [
                    "totalCount",
                    "start",
                    "limit"
                ],
                "properties": {
                    "totalCount": {
                        "type": "integer",
                        "description": "Total number"
                    },
```

```
                "start" : {
                    "type" :  "integer" ,
                    "description" :  "A position from which returning started"
                },
                "limit" : {
                    "type" :  "integer" ,
                    "description" :  "Total number of returned items"
                }
            }
        }
    }
}
```

Example:

Query

```
GET /xapi/person?start= 0 &limit= 10 &filter[change_date][from]= 0 &filter[change_date]
[to]= 1 &with[]=contacts
```

Response

```
{
    "data": [{
        "PERSON_ID": "532ECB81F75DDC46BE023478C2E86BD9",
        "WHENCHANGED": "20151229230303.0Z",
        "WHENCREATED": "20151229230303.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Annett Rist,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Annett Rist",
        "GIVENNAME": "Annett",
        "SN": "Rist",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156702",
        "SAMACCOUNTNAME": "AC428097",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC428097@acme.com",
        "MAIL": "Annett.Rist@acme.com",
        "TELEPHONENUMBER": "+49-30-626763",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.401138",
        "CHANGE_DATE": "2017-03-10 08:40:35.741001",
        "SOURCE": "ad",
        "HAS_TM_THUMBNAILPHOTO": false,
        "HAS_THUMBNAILPHOTO": false,
        "TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
        "contacts": [{
            "CONTACT_ID": "4A211BA3896A7620E0530100007F388100000000",
            "IDENTITY_ID": "532ECB81F75DDC46BE023478C2E86BD9",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac428097@acme.com",
            "IS_PRIMARY": 0,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.402060",
            "CHANGE_DATE": "2017-03-07 08:54:56.402060",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389687620E0530100007F388100000000",
            "IDENTITY_ID": "532ECB81F75DDC46BE023478C2E86BD9",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac428097@trafmon",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.401926",
            "CHANGE_DATE": "2017-03-07 08:54:56.401926",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389697620E0530100007F388100000000",
            "IDENTITY_ID": "532ECB81F75DDC46BE023478C2E86BD9",
            "IDENTITY_TYPE": "person",
```

```
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac428097@trafmon.iw",
            "IS_PRIMARY": 0,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.401998",
            "CHANGE_DATE": "2017-03-07 08:54:56.401998",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389667620E0530100007F388100000000",
            "IDENTITY_ID": "532ECB81F75DDC46BE023478C2E86BD9",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "email",
            "VALUE": "annett.rist@acme.com",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.401457",
            "CHANGE_DATE": "2017-03-07 08:54:56.401457",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389677620E0530100007F388100000000",
            "IDENTITY_ID": "532ECB81F75DDC46BE023478C2E86BD9",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "phone",
            "VALUE": "+49-30-626763",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.401670",
            "CHANGE_DATE": "2017-03-07 08:54:56.401670",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3896B7620E0530100007F388100000000",
            "IDENTITY_ID": "532ECB81F75DDC46BE023478C2E86BD9",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156702",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.402130",
            "CHANGE_DATE": "2017-03-07 08:54:56.402130",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "92FB7D1EC79A9947938BE582D355B109",
        "WHENCHANGED": "20151229230304.0Z",
        "WHENCREATED": "20151229230303.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Lieselene SchÃ¶nemann,OU=Testou,DC=trafmon,DC=iw",
```

```
"DISPLAY_NAME": "Lieselene SchÃ¶nemann",
"GIVENNAME": "Lieselene",
"SN": "SchÃ¶nemann",
"OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156706",
"SAMACCOUNTNAME": "AC727058",
"NETBIOSNAME": "TRAFMON",
"DNSROOT": "trafmon.iw",
"USERPRINCIPALNAME": "AC727058@acme.com",
"MAIL": "Lieselene.SchÃ¶nemann@acme.com",
"TELEPHONENUMBER": "+49-2183-398305",
"SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
"CREATE_DATE": "2017-03-07 08:54:56.402195",
"CHANGE_DATE": "2017-03-10 08:40:35.741198",
"SOURCE": "ad",
"HAS_TM_THUMBNAILPHOTO": false,
"HAS_THUMBNAILPHOTO": false,
"TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
"contacts": [{
    "CONTACT_ID": "4A211BA389707620E0530100007F388100000000",
    "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac727058@acme.com",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.403048",
    "CHANGE_DATE": "2017-03-07 08:54:56.403048",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA3896E7620E0530100007F388100000000",
    "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac727058@trafmon",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.402920",
    "CHANGE_DATE": "2017-03-07 08:54:56.402920",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA3896F7620E0530100007F388100000000",
    "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac727058@trafmon.iw",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.402984",
    "CHANGE_DATE": "2017-03-07 08:54:56.402984",
    "SOURCE": "ad"
```

```
        }, {
            "CONTACT_ID": "4A211BA3896C7620E0530100007F388100000000",
            "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "email",
            "VALUE": "lieselene.schÃ¶nemann@acme.com",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.402517",
            "CHANGE_DATE": "2017-03-07 08:54:56.402517",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3896D7620E0530100007F388100000000",
            "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "phone",
            "VALUE": "+49-2183-398305",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.402731",
            "CHANGE_DATE": "2017-03-07 08:54:56.402731",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389717620E0530100007F388100000000",
            "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156706",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.403117",
            "CHANGE_DATE": "2017-03-07 08:54:56.403117",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "F0A3298727D7E0498106AE678CC1F5F8",
        "WHENCHANGED": "20151229230304.0Z",
        "WHENCREATED": "20151229230304.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Kasimir Eggers,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Kasimir Eggers",
        "GIVENNAME": "Kasimir",
        "SN": "Eggers",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156710",
        "SAMACCOUNTNAME": "AC217045",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC217045@acme.com",
```

```
"MAIL": "Kasimir.Eggers@acme.com",
"TELEPHONENUMBER": "+49-7191-389863",
"SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
"CREATE_DATE": "2017-03-07 08:54:56.403185",
"CHANGE_DATE": "2017-03-10 08:40:35.741388",
"SOURCE": "ad",
"HAS_TM_THUMBNAILPHOTO": false,
"HAS_THUMBNAILPHOTO": false,
"TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
"contacts": [{
    "CONTACT_ID": "4A211BA389767620E0530100007F388100000000",
    "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac217045@acme.com",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.404114",
    "CHANGE_DATE": "2017-03-07 08:54:56.404114",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389747620E0530100007F388100000000",
    "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac217045@trafmon",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.403984",
    "CHANGE_DATE": "2017-03-07 08:54:56.403984",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389757620E0530100007F388100000000",
    "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac217045@trafmon.iw",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.404055",
    "CHANGE_DATE": "2017-03-07 08:54:56.404055",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389727620E0530100007F388100000000",
    "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "email",
    "VALUE": "kasimir.eggers@acme.com",
    "IS_PRIMARY": 1,
```

```
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.403507",
            "CHANGE_DATE": "2017-03-07 08:54:56.403507",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389737620E0530100007F388100000000",
            "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "phone",
            "VALUE": "+49-7191-389863",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.403722",
            "CHANGE_DATE": "2017-03-07 08:54:56.403722",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389777620E0530100007F388100000000",
            "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156710",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.404184",
            "CHANGE_DATE": "2017-03-07 08:54:56.404184",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "52BDB45E49090F4D986899E29FEB35CB",
        "WHENCHANGED": "20151229230304.0Z",
        "WHENCREATED": "20151229230304.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Leoni Huck,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Leoni Huck",
        "GIVENNAME": "Leoni",
        "SN": "Huck",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156709",
        "SAMACCOUNTNAME": "AC716847",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC716847@acme.com",
        "MAIL": "Leoni.Huck@acme.com",
        "TELEPHONENUMBER": "+49-9184-802313",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.404249",
        "CHANGE_DATE": "2017-03-10 08:40:35.741577",
        "SOURCE": "ad",
        "HAS_TM_THUMBNAILPHOTO": false,
        "HAS_THUMBNAILPHOTO": false,
```

```
"TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
"contacts": [{
    "CONTACT_ID": "4A211BA3897C7620E0530100007F388100000000",
    "IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac716847@acme.com",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.405104",
    "CHANGE_DATE": "2017-03-07 08:54:56.405104",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA3897A7620E0530100007F388100000000",
    "IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac716847@trafmon",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.404978",
    "CHANGE_DATE": "2017-03-07 08:54:56.404978",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA3897B7620E0530100007F388100000000",
    "IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac716847@trafmon.iw",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.405040",
    "CHANGE_DATE": "2017-03-07 08:54:56.405040",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389787620E0530100007F388100000000",
    "IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "email",
    "VALUE": "leoni.huck@acme.com",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.404575",
    "CHANGE_DATE": "2017-03-07 08:54:56.404575",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389797620E0530100007F388100000000",
    "IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB",
    "IDENTITY_TYPE": "person",
```

```
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "phone",
            "VALUE": "+49-9184-802313",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.404789",
            "CHANGE_DATE": "2017-03-07 08:54:56.404789",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3897D7620E0530100007F388100000000",
            "IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156709",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.405174",
            "CHANGE_DATE": "2017-03-07 08:54:56.405174",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "BE20920FFD837142AAC09C87A874CCD4",
        "WHENCHANGED": "20151229230304.0Z",
        "WHENCREATED": "20151229230304.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Adelgund Krebs,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Adelgund Krebs",
        "GIVENNAME": "Adelgund",
        "SN": "Krebs",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156708",
        "SAMACCOUNTNAME": "AC944381",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC944381@acme.com",
        "MAIL": "Adelgund.Krebs@acme.com",
        "TELEPHONENUMBER": "+49-6392-145757",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.405238",
        "CHANGE_DATE": "2017-03-10 08:40:35.741765",
        "SOURCE": "ad",
        "HAS_TM_THUMBNAILPHOTO": false,
        "HAS_THUMBNAILPHOTO": false,
        "TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
        "contacts": [{
            "CONTACT_ID": "4A211BA389827620E0530100007F388100000000",
            "IDENTITY_ID": "BE20920FFD837142AAC09C87A874CCD4",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac944381@acme.com",
```

```
    "IS_PRIMARY" : 0,
    "IS_PERSONAL" : 0,
    "CREATE_DATE" : "2017-03-07 08:54:56.406146",
    "CHANGE_DATE" : "2017-03-07 08:54:56.406146",
    "SOURCE" : "ad"
}, {
    "CONTACT_ID" : "4A211BA389807620E0530100007F388100000000",
    "IDENTITY_ID" : "BE20920FFD837142AAC09C87A874CCD4",
    "IDENTITY_TYPE" : "person",
    "IDENTITY_SOURCE" : "ad",
    "CONTACT_TYPE" : "auth",
    "VALUE" : "ac944381@trafmon",
    "IS_PRIMARY" : 1,
    "IS_PERSONAL" : 0,
    "CREATE_DATE" : "2017-03-07 08:54:56.406026",
    "CHANGE_DATE" : "2017-03-07 08:54:56.406026",
    "SOURCE" : "ad"
}, {
    "CONTACT_ID" : "4A211BA389817620E0530100007F388100000000",
    "IDENTITY_ID" : "BE20920FFD837142AAC09C87A874CCD4",
    "IDENTITY_TYPE" : "person",
    "IDENTITY_SOURCE" : "ad",
    "CONTACT_TYPE" : "auth",
    "VALUE" : "ac944381@trafmon.iw",
    "IS_PRIMARY" : 0,
    "IS_PERSONAL" : 0,
    "CREATE_DATE" : "2017-03-07 08:54:56.406089",
    "CHANGE_DATE" : "2017-03-07 08:54:56.406089",
    "SOURCE" : "ad"
}, {
    "CONTACT_ID" : "4A211BA3897E7620E0530100007F388100000000",
    "IDENTITY_ID" : "BE20920FFD837142AAC09C87A874CCD4",
    "IDENTITY_TYPE" : "person",
    "IDENTITY_SOURCE" : "ad",
    "CONTACT_TYPE" : "email",
    "VALUE" : "adelgund.krebs@acme.com",
    "IS_PRIMARY" : 1,
    "IS_PERSONAL" : 0,
    "CREATE_DATE" : "2017-03-07 08:54:56.405584",
    "CHANGE_DATE" : "2017-03-07 08:54:56.405584",
    "SOURCE" : "ad"
}, {
    "CONTACT_ID" : "4A211BA3897F7620E0530100007F388100000000",
    "IDENTITY_ID" : "BE20920FFD837142AAC09C87A874CCD4",
    "IDENTITY_TYPE" : "person",
    "IDENTITY_SOURCE" : "ad",
    "CONTACT_TYPE" : "phone",
    "VALUE" : "+49-6392-145757",
    "IS_PRIMARY" : 1,
    "IS_PERSONAL" : 0,
    "CREATE_DATE" : "2017-03-07 08:54:56.405836",
    "CHANGE_DATE" : "2017-03-07 08:54:56.405836",
    "SOURCE" : "ad"
```

```
        }, {
            "CONTACT_ID": "4A211BA389837620E0530100007F388100000000",
            "IDENTITY_ID": "BE20920FFD837142AAC09C87A874CCD4",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156708",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.406214",
            "CHANGE_DATE": "2017-03-07 08:54:56.406214",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "54C067BB3B8518489D098154447242DB",
        "WHENCHANGED": "20151229230304.0Z",
        "WHENCREATED": "20151229230304.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Kilian Adolph,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Kilian Adolph",
        "GIVENNAME": "Kilian",
        "SN": "Adolph",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156707",
        "SAMACCOUNTNAME": "AC598197",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC598197@acme.com",
        "MAIL": "Kilian.Adolph@acme.com",
        "TELEPHONENUMBER": "+49-30-399513",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.406279",
        "CHANGE_DATE": "2017-03-10 08:40:35.741957",
        "SOURCE": "ad",
        "HAS_TM_THUMBNAILPHOTO": false,
        "HAS_THUMBNAILPHOTO": false,
        "TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
        "contacts": [{
            "CONTACT_ID": "4A211BA389887620E0530100007F388100000000",
            "IDENTITY_ID": "54C067BB3B8518489D098154447242DB",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac598197@acme.com",
            "IS_PRIMARY": 0,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.407140",
            "CHANGE_DATE": "2017-03-07 08:54:56.407140",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389867620E0530100007F388100000000",
            "IDENTITY_ID": "54C067BB3B8518489D098154447242DB",
```

```
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac598197@trafmon",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.407000",
    "CHANGE_DATE": "2017-03-07 08:54:56.407000",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389877620E0530100007F388100000000",
    "IDENTITY_ID": "54C067BB3B8518489D098154447242DB",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac598197@trafmon.iw",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.407081",
    "CHANGE_DATE": "2017-03-07 08:54:56.407081",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389847620E0530100007F388100000000",
    "IDENTITY_ID": "54C067BB3B8518489D098154447242DB",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "email",
    "VALUE": "kilian.adolph@acme.com",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.406603",
    "CHANGE_DATE": "2017-03-07 08:54:56.406603",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389857620E0530100007F388100000000",
    "IDENTITY_ID": "54C067BB3B8518489D098154447242DB",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "phone",
    "VALUE": "+49-30-399513",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.406812",
    "CHANGE_DATE": "2017-03-07 08:54:56.406812",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389897620E0530100007F388100000000",
    "IDENTITY_ID": "54C067BB3B8518489D098154447242DB",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "sid",
    "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156707",
    "IS_PRIMARY": 1,
```

```
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.407217",
            "CHANGE_DATE": "2017-03-07 08:54:56.407217",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
        "WHENCHANGED": "20151229230305.0Z",
        "WHENCREATED": "20151229230305.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Lene Escher,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Lene Escher",
        "GIVENNAME": "Lene",
        "SN": "Escher",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156712",
        "SAMACCOUNTNAME": "AC673590",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC673590@acme.com",
        "MAIL": "Lene.Escher@acme.com",
        "TELEPHONENUMBER": "+49-6131-340459",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
        "CREATE_DATE": "2017-03-07 08:54:56.407282",
        "CHANGE_DATE": "2017-03-10 08:40:35.742722",
        "SOURCE": "ad",
        "HAS_TM_THUMBNAILPHOTO": false,
        "HAS_THUMBNAILPHOTO": false,
        "TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
        "contacts": [{
            "CONTACT_ID": "4A211BA3898E7620E0530100007F388100000000",
            "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac673590@acme.com",
            "IS_PRIMARY": 0,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.408129",
            "CHANGE_DATE": "2017-03-07 08:54:56.408129",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3898C7620E0530100007F388100000000",
            "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac673590@trafmon",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.408006",
            "CHANGE_DATE": "2017-03-07 08:54:56.408006",
```

```
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3898D7620E0530100007F388100000000",
            "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "auth",
            "VALUE": "ac673590@trafmon.iw",
            "IS_PRIMARY": 0,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.408071",
            "CHANGE_DATE": "2017-03-07 08:54:56.408071",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3898A7620E0530100007F388100000000",
            "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "email",
            "VALUE": "lene.escher@acme.com",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.407607",
            "CHANGE_DATE": "2017-03-07 08:54:56.407607",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3898B7620E0530100007F388100000000",
            "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "phone",
            "VALUE": "+49-6131-340459",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.407817",
            "CHANGE_DATE": "2017-03-07 08:54:56.407817",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3898F7620E0530100007F388100000000",
            "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156712",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.408198",
            "CHANGE_DATE": "2017-03-07 08:54:56.408198",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "7B238377335B7B48AFDC99851D8F4793",
```

```
"WHENCHANGED": "20151229230305.0Z",
"WHENCREATED": "20151229230305.0Z",
"DISABLED": 0,
"IS_EMPLOYEE": 1,
"DISTINGUISHEDNAME": "CN=Annett Schwanke,OU=Testou,DC=trafmon,DC=iw",
"DISPLAY_NAME": "Annett Schwanke",
"GIVENNAME": "Annett",
"SN": "Schwanke",
"OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156711",
"SAMACCOUNTNAME": "AC275412",
"NETBIOSNAME": "TRAFMON",
"DNSROOT": "trafmon.iw",
"USERPRINCIPALNAME": "AC275412@acme.com",
"MAIL": "Annett.Schwanke@acme.com",
"TELEPHONENUMBER": "+49-33633-744751",
"SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
"CREATE_DATE": "2017-03-07 08:54:56.408263",
"CHANGE_DATE": "2017-03-10 08:40:35.742148",
"SOURCE": "ad",
"HAS_TM_THUMBNAILPHOTO": false,
"HAS_THUMBNAILPHOTO": false,
"TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
"contacts": [{
    "CONTACT_ID": "4A211BA389947620E0530100007F388100000000",
    "IDENTITY_ID": "7B238377335B7B48AFDC99851D8F4793",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac275412@acme.com",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.409108",
    "CHANGE_DATE": "2017-03-07 08:54:56.409108",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389927620E0530100007F388100000000",
    "IDENTITY_ID": "7B238377335B7B48AFDC99851D8F4793",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac275412@trafmon",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.408985",
    "CHANGE_DATE": "2017-03-07 08:54:56.408985",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389937620E0530100007F388100000000",
    "IDENTITY_ID": "7B238377335B7B48AFDC99851D8F4793",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac275412@trafmon.iw",
```

```
            "IS_PRIMARY" : 0,
            "IS_PERSONAL" : 0,
            "CREATE_DATE" : "2017-03-07 08:54:56.409051",
            "CHANGE_DATE" : "2017-03-07 08:54:56.409051",
            "SOURCE" : "ad"
        }, {
            "CONTACT_ID" : "4A211BA389907620E0530100007F388100000000",
            "IDENTITY_ID" : "7B238377335B7B48AFDC99851D8F4793",
            "IDENTITY_TYPE" : "person",
            "IDENTITY_SOURCE" : "ad",
            "CONTACT_TYPE" : "email",
            "VALUE" : "annett.schwanke@acme.com",
            "IS_PRIMARY" : 1,
            "IS_PERSONAL" : 0,
            "CREATE_DATE" : "2017-03-07 08:54:56.408589",
            "CHANGE_DATE" : "2017-03-07 08:54:56.408589",
            "SOURCE" : "ad"
        }, {
            "CONTACT_ID" : "4A211BA389917620E0530100007F388100000000",
            "IDENTITY_ID" : "7B238377335B7B48AFDC99851D8F4793",
            "IDENTITY_TYPE" : "person",
            "IDENTITY_SOURCE" : "ad",
            "CONTACT_TYPE" : "phone",
            "VALUE" : "+49-33633-744751",
            "IS_PRIMARY" : 1,
            "IS_PERSONAL" : 0,
            "CREATE_DATE" : "2017-03-07 08:54:56.408799",
            "CHANGE_DATE" : "2017-03-07 08:54:56.408799",
            "SOURCE" : "ad"
        }, {
            "CONTACT_ID" : "4A211BA389957620E0530100007F388100000000",
            "IDENTITY_ID" : "7B238377335B7B48AFDC99851D8F4793",
            "IDENTITY_TYPE" : "person",
            "IDENTITY_SOURCE" : "ad",
            "CONTACT_TYPE" : "sid",
            "VALUE" : "s-1-5-21-4236050433-3472386798-155597913-156711",
            "IS_PRIMARY" : 1,
            "IS_PERSONAL" : 0,
            "CREATE_DATE" : "2017-03-07 08:54:56.409176",
            "CHANGE_DATE" : "2017-03-07 08:54:56.409176",
            "SOURCE" : "ad"
        }],
        "status" : []
    }, {
        "PERSON_ID" : "E277B624BC988C4C96590A8435F634ED",
        "WHENCHANGED" : "20151229230305.0Z",
        "WHENCREATED" : "20151229230305.0Z",
        "DISABLED" : 0,
        "IS_EMPLOYEE" : 1,
        "DISTINGUISHEDNAME" : "CN=Edgar Genz,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME" : "Edgar Genz",
        "GIVENNAME" : "Edgar",
        "SN" : "Genz",
```

```
"OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156714",
"SAMACCOUNTNAME": "AC938446",
"NETBIOSNAME": "TRAFMON",
"DNSROOT": "trafmon.iw",
"USERPRINCIPALNAME": "AC938446@acme.com",
"MAIL": "Edgar.Genz@acme.com",
"TELEPHONENUMBER": "+49-7191-731883",
"SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
"CREATE_DATE": "2017-03-07 08:54:56.409247",
"CHANGE_DATE": "2017-03-10 08:40:35.742335",
"SOURCE": "ad",
"HAS_TM_THUMBNAILPHOTO": false,
"HAS_THUMBNAILPHOTO": false,
"TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
"contacts": [{
    "CONTACT_ID": "4A211BA3899A7620E0530100007F388100000000",
    "IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac938446@acme.com",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.410087",
    "CHANGE_DATE": "2017-03-07 08:54:56.410087",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389987620E0530100007F388100000000",
    "IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac938446@trafmon",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.409967",
    "CHANGE_DATE": "2017-03-07 08:54:56.409967",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389997620E0530100007F388100000000",
    "IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac938446@trafmon.iw",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.410030",
    "CHANGE_DATE": "2017-03-07 08:54:56.410030",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA389967620E0530100007F388100000000",
    "IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED",
```

```
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "email",
            "VALUE": "edgar.genz@acme.com",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.409572",
            "CHANGE_DATE": "2017-03-07 08:54:56.409572",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA389977620E0530100007F388100000000",
            "IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "phone",
            "VALUE": "+49-7191-731883",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.409781",
            "CHANGE_DATE": "2017-03-07 08:54:56.409781",
            "SOURCE": "ad"
        }, {
            "CONTACT_ID": "4A211BA3899B7620E0530100007F388100000000",
            "IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED",
            "IDENTITY_TYPE": "person",
            "IDENTITY_SOURCE": "ad",
            "CONTACT_TYPE": "sid",
            "VALUE": "s-1-5-21-4236050433-3472386798-155597913-156714",
            "IS_PRIMARY": 1,
            "IS_PERSONAL": 0,
            "CREATE_DATE": "2017-03-07 08:54:56.410155",
            "CHANGE_DATE": "2017-03-07 08:54:56.410155",
            "SOURCE": "ad"
        }],
        "status": []
    }, {
        "PERSON_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",
        "WHENCHANGED": "20151229230305.0Z",
        "WHENCREATED": "20151229230305.0Z",
        "DISABLED": 0,
        "IS_EMPLOYEE": 1,
        "DISTINGUISHEDNAME": "CN=Velten SchÃ¶ning,OU=Testou,DC=trafmon,DC=iw",
        "DISPLAY_NAME": "Velten SchÃ¶ning",
        "GIVENNAME": "Velten",
        "SN": "SchÃ¶ning",
        "OBJECTSID": "S-1-5-21-4236050433-3472386798-155597913-156713",
        "SAMACCOUNTNAME": "AC617874",
        "NETBIOSNAME": "TRAFMON",
        "DNSROOT": "trafmon.iw",
        "USERPRINCIPALNAME": "AC617874@acme.com",
        "MAIL": "Velten.SchÃ¶ning@acme.com",
        "TELEPHONENUMBER": "+49-9184-565860",
        "SERVER_ID": "65E2D37A643561E5E43C7E757BE121096253B4D6",
```

```
"CREATE_DATE": "2017-03-07 08:54:56.410219",
"CHANGE_DATE": "2017-03-10 08:40:35.742522",
"SOURCE": "ad",
"HAS_TM_THUMBNAILPHOTO": false,
"HAS_THUMBNAILPHOTO": false,
"TM_PHOTO": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
"contacts": [{
    "CONTACT_ID": "4A211BA389A07620E0530100007F388100000000",
    "IDENTITY_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac617874@acme.com",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.411056",
    "CHANGE_DATE": "2017-03-07 08:54:56.411056",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA3899E7620E0530100007F388100000000",
    "IDENTITY_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac617874@trafmon",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.410937",
    "CHANGE_DATE": "2017-03-07 08:54:56.410937",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA3899F7620E0530100007F388100000000",
    "IDENTITY_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "auth",
    "VALUE": "ac617874@trafmon.iw",
    "IS_PRIMARY": 0,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.411000",
    "CHANGE_DATE": "2017-03-07 08:54:56.411000",
    "SOURCE": "ad"
}, {
    "CONTACT_ID": "4A211BA3899C7620E0530100007F388100000000",
    "IDENTITY_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",
    "IDENTITY_TYPE": "person",
    "IDENTITY_SOURCE": "ad",
    "CONTACT_TYPE": "email",
    "VALUE": "velten.schÃ¶ning@acme.com",
    "IS_PRIMARY": 1,
    "IS_PERSONAL": 0,
    "CREATE_DATE": "2017-03-07 08:54:56.410541",
    "CHANGE_DATE": "2017-03-07 08:54:56.410541",
```

```
            "SOURCE" : "ad"
        }, {
            "CONTACT_ID" : "4A211BA3899D7620E0530100007F388100000000",
            "IDENTITY_ID" : "DEF1CC1FA015CA428D5DE3CD5B804B69",
            "IDENTITY_TYPE" : "person",
            "IDENTITY_SOURCE" : "ad",
            "CONTACT_TYPE" : "phone",
            "VALUE" : "+49-9184-565860",
            "IS_PRIMARY" : 1,
            "IS_PERSONAL" : 0,
            "CREATE_DATE" : "2017-03-07 08:54:56.410750",
            "CHANGE_DATE" : "2017-03-07 08:54:56.410750",
            "SOURCE" : "ad"
        }, {
            "CONTACT_ID" : "4A211BA389A17620E0530100007F388100000000",
            "IDENTITY_ID" : "DEF1CC1FA015CA428D5DE3CD5B804B69",
            "IDENTITY_TYPE" : "person",
            "IDENTITY_SOURCE" : "ad",
            "CONTACT_TYPE" : "sid",
            "VALUE" : "s-1-5-21-4236050433-3472386798-155597913-156713",
            "IS_PRIMARY" : 1,
            "IS_PERSONAL" : 0,
            "CREATE_DATE" : "2017-03-07 08:54:56.411129",
            "CHANGE_DATE" : "2017-03-07 08:54:56.411129",
            "SOURCE" : "ad"
        }],
        "status" : []
    }],
    "meta" : {
        "totalCount" : 232525,
        "start" : 0,
        "limit" : 10
    }
}
```


Getting the list of person statuses

Description: The list of all possible person statuses

Method: `GET`

Resource: `person/status`

Parameters:

- `start` — a start position for returning items, the default value is 0
- `limit` — how many items to return (the default value is 100, cannot be more than 1000

Response: The list of statuses described with the following JSON Schema:

JSON schema for returning the list of statuses

```
{
    "$schema": "http://json-schema.org/draft-04/schema#" ,
    "type": "object",
    "properties": {
        "data": {
            "description": "Array of statuses",
            "type": "array",
            "items": {
                "description": "Person status",
                "type": "object",
                "required": [
                    "IDENTITY_STATUS_ID",
                    "CREATE_DATE",
                    "CHANGE_DATE",
                    "ASSIGNMENT_DATE",
                    "DISPLAY_NAME",
                    "COLOR",
                    "EDITABLE"
                ],
                "properties": {
                    "IDENTITY_STATUS_ID": {
                        "description": "Status unique identifier",
                        "type": "string"
                    },
                    "CREATE_DATE": {
                        "description": "Status creation date",
                        "type": "string"
                    },
                    "CHANGE_DATE": {
                        "description": "Status modification date",
                        "type": "string"
                    },
                    "ASSIGNMENT_DATE": {
                        "description": "The date when the status was assigned to the
person",
                        "type": "string"
                    },
                    "DISPLAY_NAME": {
                        "description": "Status name",
                        "type": "string"
                    },
                    "COLOR": {
                        "description": "Status color (e.g. #ff0000)",
                        "type": "string"
                    },
                    "EDITABLE": {
                        "description": "Indicator that the status is not system",
                        "type": "integer"
                    }
                }
            }
        },
        "meta": {
```

```
            "description" :  "Additional data" ,
            "type" :  "object" ,
            "required" : [
                "totalCount" ,
                "start" ,
                "limit"
            ],
            "properties" : {
                "totalCount" : {
                    "description" :  "Total number" ,
                    "type" :  "integer"
                },
                "start" : {
                    "description" :  "A position from which returning started" ,
                    "type" :  "integer"
                },
                "limit" : {
                    "description" :  "Total number of returned items" ,
                    "type" :  "integer"
                }
            }
        }
    }
}
```

Example:

Query

```
GET /xapi/person/status?start=0&limit=10
```

Getting the list of person contact types

Description: The list of all possible contact types

Method: `GET`

Resource: `person/contact_type`

Parameters:

- `start` — a start position for returning items, the default value is 0
- `limit` — how many items to return (the default value is 100, cannot be more than 1000

Response: The list of types described with the following JSON Schema:

JSON schema of returning the list of contacts

```
{
    "$schema": "http://json-schema.org/draft-04/schema#" ,
    "type": "object",
    "properties": {
        "data": {
            "description": "Array of contact types",
            "type": "array",
            "items": {
                "description": "Contact type",
                "type": "object",
                "properties": {
                    "CONTACT_TYPE_ID": {
                        "description": "A unique identifier of the contact type",
                        "type": "string"
                    },
                    "DISPLAY_NAME": {
                        "description": "Contact type name",
                        "type": "string"
                    },
                    "LANGUAGE": {
                        "description": "Translation language",
                        "type": "string"
                    },
                    "MNEMO": {
                        "description": "Mnemonic identifier (is used in contact
entity)",
                        "type": "string"
                    }
                }
            }
        },
        "meta": {
            "description": "Additional data",
            "type": "object",
            "required": [
                "totalCount",
                "start",
                "limit"
            ],
            "properties": {
                "totalCount": {
                    "description": "Total number",
                    "type": "integer"
                },
                "start": {
                    "description": "A position where to start returning items",
                    "type": "integer"
                },
                "limit": {
                    "description": "Total number of returned items",
                    "type": "integer"
                }
            }
```

```
        }
    }
}
```

Example:

Query

```
GET /xapi/person/contact_type?start=0&limit=10
```

Gettins statistics on persons

Description: Statistics on persons

Method: `GET`

Resource: `person/stat`

Parameters:

- `filter` — the filter list:
  - `change_date[from]` — since the date of the last change in UNIX-timestamp format (can be used separately from `change_date[to]`)
  - `change_date[to]` — up to the date of the last change in UNIX-timestamp format (can be used separately from `change_date[from]`)

Response:

JSON returning schema

```
{
    "$schema": "http://json-schema.org/draft-04/schema#" ,
    "description": "Statistics",
    "type": "object",
    "properties": {
        "data": {
            "description": "Not used",
            "type": "array",
            "items": {}
        },
        "meta": {
            "description": "Statistics",
            "type": "object",
            "properties": {
                "CHANGE_DATE": {
                    "description": "Modification date",
                    "type": "object",
                    "properties": {
                        "from": {
                            "description": "The first modification date",
                            "type": "string"
                        },
                        "to": {
                            "description": "The last modification date",
                            "type": "string"
                        }
                    }
                },
                "totalCount": {
                    "description": "Total number of items",
                    "type": "integer"
                }
            }
        }
    }
}
```


Example:

```
GET /xapi/person/stat?filter[change_date][to]=1494951762&filter[change_date][from]=1494951000
```

Response


```
{
  "data": [],
  "meta": {
    "CHANGE_DATE": {
      "from": "2017-03-24 10:40:03",
      "to": "2017-05-24 12:10:22"
    },
    "totalCount": 539693
  }
}
```

5.3.3 Getting information on workstations

Getting the list of workstations

Description: The list of workstations

Method: `GET`

Resource: `workstation`

Параметры:

- `filter` — the filter list:
  - `change_date[from]` — since the date of the last change in UNIX-timestamp format (can be used separately from `change_date[to]`)
  - `change_date[to]` — up to the date of the last change in UNIX-timestamp format (can be used separately from `change_date[from]`)
  - `create_date[from]` — since the date of the last change in UNIX-timestamp format (can be used separately from `create_date[to]`)
  - `create_date[to]` — up to the date of the last change in UNIX-timestamp format (can be used separately from `create_date[from]`
  - is_deleted — flag 1 or 0. If the parameter is specified, then entities with the specified value will be returned
  - `disabled` — flag 1 or 0, indicates activity. If the flag is set, then the entity is inactive (the flag is obtained from LDAP)

  - `contact_type[]` — filtering by the contact type (several values can be passed, e.g., `/xapi/workstation?filter[contact_type][]=webaccount&filter[contact_type][]=icq`)
  - `contact_value[]` — filtering by the contact value (several values can be passed, e.g., `/xapi/workstation?filter[contact_value][]=user@ad&filter[contact_value][]=login@example.com`)
  - `group_id[]` — filtering by groups (several values can be passed, e.g., `/xapi/workstation?filter[group_id][]=FCE1409680E5D6469E1F7998266CE9E9&filter[group_id][]=918462B6DC2C954BAAF8706F7DC53F14`)
- `with[]` — the list of additional entities to be added to the object:
  - `contacts` — workstation contacts
  - `w2g` — workstation connections with groups
  - `status` — workstation statuses
- `sort` — parameter for sorting
  - `change_date` — sort by the modification date `asc` or `desc` (e.g., `/xapi/workstation?sort[change_date]=asc`) может замедлить выдачу
- `start` — a start position for returning items, the default value is 0
- `limit` — how many items to return (the default value is 100, cannot be more than 1000

Response: The list of workstations described with the following JSON Schema :

JSON returning schema

```
{
    "$schema": "http://json-schema.org/draft-04/schema#" ,
    "type": "object",
    "required": [
        "data",
        "meta"
    ],
    "properties": {
        "data": {
            "description": "Array of workstations",
            "type": "array",
            "items": {
                "description": "Workstation",
                "type": "object",
                "required": [
                    "WORKSTATION_ID",
                    "CREATE_DATE",
                    "CHANGE_DATE",
                    "DISPLAY_NAME",
                    "DISTINGUISHEDNAME",
                    "DNSHOSTNAME",
                    "DISABLED",
                    "TYPE",
                    "IS_DELETED"
                ],
                "properties": {
                    "WORKSTATION_ID": {
                        "description": "Workstation unique identifier",
                        "type": "string"
                    },
                    "CREATE_DATE": {
                        "description": "Entry creation date",
                        "type": "string"
                    },
                    "CHANGE_DATE": {
                        "description": "Entry modification date",
                        "type": "string"
                    },
                    "DISPLAY_NAME": {
                        "description": "Workstation name",
                        "type": "string"
                    },
                    "DISTINGUISHEDNAME": {
                        "description": "Distinguished name",
                        "type": "string"
                    },
                    "DNSHOSTNAME": {
                        "description": "Full name",
                        "type": "string"
                    },
                    "DISABLED": {
                        "description": "Workstation state",
                        "type": "integer"
```

```
                    },
                    "TYPE": {
                        "description": "Type: computer – Workstation, mobile – Mobile
device, terminal – Terminal server",
                        "type": "string"
                    },
                    "IS_DELETED": {
                        "description": "If the workstation is deleted from TM",
                        "type": "integer"
                    },
                    "contacts": {
                        "description": "Array of contacts",
                        "type": "array",
                        "items": {
                            "description": "Contact",
                            "type": "object",
                            "required": [
                                "CONTACT_ID",
                                "CREATE_DATE",
                                "CHANGE_DATE",
                                "CONTACT_TYPE",
                                "VALUE",
                                "IS_PERSONAL",
                                "IS_PRIMARY",
                                "SOURCE"
                            ],
                            "properties": {
                                "CONTACT_ID": {
                                    "description": "Object unique identifier",
                                    "type": "string"
                                },
                                "CREATE_DATE": {
                                    "description": "Creation date",
                                    "type": "string"
                                },
                                "CHANGE_DATE": {
                                    "description": "Modification date",
                                    "type": "string"
                                },
                                "CONTACT_TYPE": {
                                    "description": "Type",
                                    "type": "string"
                                },
                                "VALUE": {
                                    "description": "Value",
                                    "type": "string"
                                },
                                "IS_PERSONAL": {
                                    "description": "Belongs to (Work, Personal)",
                                    "type": "integer"
                                },
                                "IS_PRIMARY": {
                                    "description": "Is primary",
```

```
                                "type": "integer"
                            },
                            "SOURCE": {
                                "description": "Type",
                                "type": "string"
                            }
                        }
                    }
                },
                "w2g": {
                    "description": "Array of connections with groups",
                    "type": "array",
                    "items": {
                        "description": "Connection with group",
                        "type": "object",
                        "required": [
                            "WORKSTATION_ID",
                            "PARENT_GROUP_ID",
                            "RELATION_TYPE"
                        ],
                        "properties": {
                            "WORKSTATION_ID": {
                                "description": "Workstation unique identifier",
                                "type": "string"
                            },
                            "PARENT_GROUP_ID": {
                                "description": "Group unique identifier",
                                "type": "string"
                            },
                            "RELATION_TYPE": {
                                "description": "Connection type: 0 - tm, 1 -
member, 2 - distinguishedname, 3 - primarygroupsid",
                                "type": "integer"
                            }
                        }
                    }
                },
                "status": {
                    "description": "Array of statuses",
                    "type": "array",
                    "items": {
                        "description": "Person status",
                        "type": "object",
                        "required": [
                            "IDENTITY_STATUS_ID",
                            "CREATE_DATE",
                            "CHANGE_DATE",
                            "ASSIGNMENT_DATE",
                            "DISPLAY_NAME",
                            "COLOR",
                            "EDITABLE"
                        ],
                        "properties": {
```

```
                                "IDENTITY_STATUS_ID": {
                                    "description": "Status unique identifier",
                                    "type": "string"
                                },
                                "CREATE_DATE": {
                                    "description": "Status creation date",
                                    "type": "string"
                                },
                                "CHANGE_DATE": {
                                    "description": "Status modification date",
                                    "type": "string"
                                },
                                "ASSIGNMENT_DATE": {
                                    "description": "The date when the status was
assigned to the person",
                                    "type": "string"
                                },
                                "DISPLAY_NAME": {
                                    "description": "Status name",
                                    "type": "string"
                                },
                                "COLOR": {
                                    "description": "Status color (e.g. #ff0000)",
                                    "type": "string"
                                },
                                "EDITABLE": {
                                    "description": "Indicator that the status is not
system",
                                    "type": "integer"
                                }
                            }
                        }
                    }
                }
            },
            "meta": {
                "description": "Additional data",
                "type": "object",
                "required": [
                    "totalCount",
                    "start",
                    "limit"
                ],
                "properties": {
                    "totalCount": {
                        "type": "integer",
                        "description": "Total number"
                    },
                    "start": {
                        "type": "integer",
                        "description": "A position from which returning started"
                    },
```

```
                "limit" : {
                    "type" :  "integer" ,
                    "description" :  "Total number of returned items"
                }
            }
        }
    }
}
```

Example:

```
/xapi/workstation?filter[change_date][from]=1&filter[change_date][to]=2& with
[]=contacts& with []=w2g&limit=10&sort[change_date]=asc
```

Getting the list of workstation statuses

Description: The list of all possible workstation statuses

Method: `GET`

Resource: `workstation/status`

- `start` — a start position for returning items, the default value is 0
- `limit` — how many items to return (the default value is 100, cannot be more than 1000

Example:

Query

```
GET /xapi/workstation/status?start= 0 &limit= 10
```

Getting the list of workstation contact types

Description: The list of all possible types of workstation contacts

Method: `GET`

Resource: `workstation/contact_type`

- `start` — a start position for returning items, the default value is 0
- `limit` — how many items to return (the default value is 100, cannot be more than 1000

Example:

Query

```
GET /xapi/workstation/contact_type?start= 0 &limit= 10
```

Getting statistics on workstations

Description: Statistics on workstations

Method: `GET`

Resource: workstation `/stat`

Parameters:

- `filter` — the filter list:
  - `change_date[from]` — since the date of the last change in UNIX-timestamp format (can be used separately from `change_date[to]`)
  - `change_date[to]` — up to the date of the last change in UNIX-timestamp format (can be used separately from `change_date[from]`)

Response:

JSON returning schema

```
{
    "$schema": "http://json-schema.org/draft-04/schema#" ,
    "description": "Statistics",
    "type": "object",
    "properties": {
        "data": {
            "description": "Not used",
            "type": "array",
            "items": {}
        },
        "meta": {
            "description": "Statistics",
            "type": "object",
            "properties": {
                "CHANGE_DATE": {
                    "description": "Modification date",
                    "type": "object",
                    "properties": {
                        "from": {
                            "description": "The first modification date",
                            "type": "string"
                        },
                        "to": {
                            "description": "The last modification date",
                            "type": "string"
                        }
                    }
                },
                "totalCount": {
                    "description": "Total number of items",
                    "type": "integer"
                }
            }
        }
    }
}
```

Example:

```
GET /xapi/workstation/stat?filter[CAPTURE_DATE][to]=1494951762&filter[CAPTURE_DATE]
[from]=1494951000
```

```
{
  "data" : [],
  "meta" : {
    "CHANGE_DATE" : {
      "from" : "2017-03-24 10:40:03",
      "to" : "2017-05-24 12:10:22"
    },
    "totalCount" : 539693
  }
}
```

5.4 Getting events

5.4.1 Getting text extracted from multiple events

Method: POST

Resource: `event/texts`

Parameters (passed in the form of JSON):

- `response_type` — the response type:
  - chunked — `Transfer-Encoding: chunked` is used, where each text is passed in its own block and in the order which is specified in the request body; if there is no text, then null character "\0" is passed instead.
  - `multipart/mixed` — `multipart/mixed` rfc tech-ms format is used. This format is not part of the HTTP standard, however, it is used in emails, so it can be easily implemented. Besides, it is very similar to `multipart/form-data`. Each text is passed in its block and is separately marked with X-Event-ID title indicating the event the text belongs to. If there is no text, then it is either not passed or the text body is missing.
- `event` — an array of objects with the following fields:
  - `id` — event ID
  - `tbsId` — tablespace with the event

Examples:

chunked :

Query

```
POST /xapi/event/texts
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(empty line)
{"response_type": "chunked", "events": [{"id": 2, "tbsId": 4}, {"id": 4, "tbsId": 4}]}
```

Response

```
HTTP/1.1 200 OK
```

```
Content-Type: text/plain
Transfer-Encoding: chunked
(empty line)
1
\0 (no-text character for event 2)
48AC
(text content for event 4)
0
```

`multipart/mixed`:

Query

```
POST /xapi/event/texts
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(empty line)
{"response_type": "multipart/mixed", "events": [{"id": 2, "tbsId": 4}, {"id": 4, "tbsId": 4}]}
```

Response

```
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=Asrf456BGe4h
Content-Length: (total volume including titles, can be missing!)
(empty line)
(missing preamble)
--Asrf456BGe4h
Content-Type: application/x-empty
Content-Disposition: attachment; filename="text-2.txt"
X-Event-ID: 2
(empty line and zero content then)
--Asrf456BGe4h
Content-Disposition: attachment; filename="text-4.txt"
Content-Type: text/plain
X-Event-ID: 4
(empty line)
(event content for event 4)
--Asrf456BGe4h--
(missing epilogue - empty line)
```

5.4.2 Getting extracted text of event

Method: `POST`

Resource: `event/texts`

Parameters (transferred as JSON):

- `response_type` — type of response:
- chunked — using `Transfer-Encoding: chunked`, where each text is left within its block and order, transferred within the query body. In case there is no text, null, symbol "\0" is trasferred instead of the text

- `multipart/mixed` — `multipart/mixed`[rfc tech-ms] format is used, it is not a HTTP format, but it is used in e-mails. That's why this can be realised. It is also similar to `multipart/form-data` . Each text is left within its block and is specified by X-Event-ID title in accordance with event it belongs to. In case there is no text, it cannot be transferred or the text body is absent
- `events` — objects massive with fields:
- `id` — event ID
- `tbsId` — event tablespace

Examples:

chunked :

Query

```
 POST /xapi/event/texts
X-API-Version: 1.1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(empty string)
{"response_type": "chunked", "events": [{"id": 2, "tbsId": 4}, {"id": 4, "tbsId": 4}]}
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/plain
Transfer-Encoding: chunked
(empty string)
1
\0 (text absence symbol for event 2)
48AC
(text content for event 4)
0
```

multipart/mixed :

Query

```
POST /xapi/event/texts
X-API-Version: 1.1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(empty string)
{"response_type": "multipart/mixed", "events": [{"id": 2, "tbsId": 4}, {"id": 4, "tbsId": 4}]}
```

Response

```
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=Asrf456BGe4h
Content-Length: (summary amount, including child titles can be absent!)
(empty string)
(absent preamble)
```

```
--Asrf456BGe4h
Content-Type: application/x-empty
Content-Disposition: attachment; filename="text-2.txt"
X-Event-ID: 2
(empty string and null content after)
--Asrf456BGe4h
Content-Disposition: attachment; filename="text-4.txt"
Content-Type: text/plain
X-Event-ID: 4
(empty string)
(text content for event 4)
--Asrf456BGe4h--
(absent epilogue - empty string)
```

5.4.3 Getting event metadata

Method: `GET`

Resource: `event/{id}`

Parameters:

- `{id}` - `OBJECT_ID` of event
- `with[]` - list of additional entities, to be added to the object (for example, senders, recipients, etc.).

Response: Event

Example:

Get event OBJECT_ID "1" with senders and recipients:

```
GET /xapi/event/1?with[]=senders&with[]=recipients
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: aplab
X-API-Auth-Token: gla20dos1pcdbr69caxz
```

5.4.4 Getting multiple event contents

Method: `POST`

Resource: `event/raws`

Parameters (passed in the form of JSON):

- `response_type` — the response type:
  - chunked — `Transfer-Encoding: chunked` is used, where each content is passed in its own block and in the order which is specified in the request body; if there is no content, then null character "\0" is passed instead
  - `multipart/mixed` — `multipart/mixed` [rfc tech-ms] format is used. This format is not part of the HTTP standard, however, it is used in emails, so it can be easily implemented. Besides, it is very similar to `multipart/form-data`. Each content is passed in its block and is separately marked with X-Event-ID and X-Content-ID titles which indicate the event and the

content the block belongs to.f there is no content, then it is either not passed or the text body is missing.

- `event` — an array of objects with the following fields:
  - `id` — event ID;
  - `tbsId` — tablespace with the event;
  - `contentId` — content ID. If not specified, the main node is passed.

Examples:

chunked :

Query

```
POST /xapi/event/raws
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(empty line)
{"response_type": "chunked", "events": [{"id": 2, "tbsId": 4}, {"id": 3, "tbsId": 4}, {"id": 4, "tbsId": 4,
"contentId": 8}]}
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/plain
Transfer-Encoding: chunked
(empty line)
1
\0 (no-content character for event 2)
3E4
(root content data for event 3)
48AC
(data of content 8 for event 4)
0
```

multipart/mixed :

Query

```
POST /xapi/event/raws
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
(empty line)
{"response_type": "multipart/mixed", "events": [{"id": 2, "tbsId": 4}, {"id": 3, "tbsId": 4}, {"id": 4,
"tbsId": 4, "contentId": 8}]}
```

Response

```
HTTP/1.1 200 OK
```

```
Content-Type: multipart/mixed; boundary=Asrf456BGe4h
Content-Length: (total volume including titles, can be missing!)
(empty line)
(missing preamble)
--Asrf456BGe4h
Content-Type: application/x-empty
Content-Disposition: attachment; filename="content-2"
X-Event-ID: 2
(empty line and then zero content)
--Asrf456BGe4h
Content-Disposition: attachment; filename="content-3-5.eml"
Content-Type: email
X-Event-ID: 3
X-Content-ID: 5
(empty line)
(root content data for event 3)
--Asrf456BGe4h
Content-Disposition: attachment; filename="my-project.jpg"
Content-Type: image/jpeg
X-Event-ID: 4
X-Content-ID: 8
(empty line)
(data of content 8 for event 4)
--Asrf456BGe4h--

(missing epilogue - empty line)
```

5.4.5 Getting intercepted file of event

Method: `GET`

Resource: `event/{id}/raw?download=0`

Parameters:

- `{id}` - OBJECT_ID of event, mandatory
- `download` - adding HTTP-heading to download a file (`Content-Disposition`, `Content-Transfer-Encoding` and others), mandatory, the value is 1 or 0 (default value is 0)

Supporting HTTP-heading `Range`

Response: File of intercepted event.For example, eml (for emails) or the file itself for copying file event.

5.4.6 Getting the list of titles

Description: Getting the list of additional fields of the event.

Method: `GET`

Resource: `event/header`

Parameters:

- `start` - the start position for returning titles; the default value is 0
- `limit` - the number of titles to be returned; the default value is 100; cannot be more than 1000

Response: The list of additional fields described with the following **JSON return schema:**

```
{
  "$id": "http://infowatch.com/iwtm/xapi/event/header.json",
  "type": "object",
  "$schema": "http://json-schema.org/draft-07/schema#",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "NAME": {
            "type": "string",
            "title": "Attribute code that will be used in the event titles and for identification in xAPI/PushAPI",
            "examples": [
              "mac_level"
            ]
          },
          "NOTE": {
            "type": "string",
            "title": "Attribute name which is displayed in TM Console",
            "examples": [
              "Mandatory access level"
            ]
          },
          "TYPE": {
            "type": "string",
            "title": "One of the following values: number (integer, fractional, string), date and time in UTC format + specifying time zone offset, duration, hyperlink",
            "enum": [
              "integer",
              "float",
              "string",
              "date",
              "UTC",
              "duration",
              "link"
            ]
```

```
},
"IS_DELETED": {
  "type": "integer",
  "title": "Whether the title is removed from the System (for obsolete objects, it remains)",
  "default": 0,
  "enum": [
    0,
    1
  ]
},
"FORMAT": {
  "type": "string",
  "title": "Value type (only for version 1.2)",
  "enum": [
    "string",
    "date",
    "number"
  ]
},
"IS_SYSTEM": {
  "type": "integer",
  "title": "Whether the title is system (only for version 1.1)",
  "default": 1,
  "enum": [
    0,
    1
  ]
},
"IS_MULTIPLE_VALUE": {
  "type": "integer",
  "title": "Can the title contain several values (for verrion 1.1)",
  "default": 0,
  "enum": [
    0,
    1
  ]
},
```

```
            "locale": {
              "type": "array",
              "title": "Titles localization (for version 1.2)",
              "items": {
                "type": "object",
                "title": "Title localization",
                "properties": {
                  "NAME": {
                    "type": "string",
                    "title": "Attribute code that will be used in the event titles and for identification in xAPI/
PushAPI"
                  },
                  "LANGUAGE": {
                    "type": "string",
                    "title": "Language name",
                    "enum": [
                      "rus",
                      "eng"
                    ]
                  },
                  "DISPLAY_NAME": {
                    "type": "string",
                    "title": "Attribute name which is displayed in TM Console"
                  }
                }
              }
            }
          }
        }
      },
      "meta": {
        "description": "Additional data",
        "type": "object",
        "properties": {
          "totalCount": {
            "type": "integer",
            "description": "Total number of protected objects"
```

```
        },
        "start": {
          "type": "integer",
          "description": "From which position has it started to return"
        },
        "limit": {
          "type": "integer",
          "description": "Total number of returned"
        }
      },
      "required": [
        "totalCount",
        "start",
        "limit"
      ]
    }
  },
  "required": [
    "data",
    "meta"
  ]
}
```


Example:

Query

```
GET /xapi/event/header
X-API-Version: 1.1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
```

```
{
  "data": [
    {
      "NAME": "device_type",
      "NOTE": "Device type",
      "TYPE": "enum",
      "IS_DELETED": 0,
      "IS_SYSTEM": 1,
      "IS_MULTIPLE_VALUE": 0
    },
    {
      "NAME": "destination_type",
      "NOTE": "Type of the scanning source",
      "TYPE": "enum",
      "IS_DELETED": 0,
      "IS_SYSTEM": 1,
      "IS_MULTIPLE_VALUE": 0
    },
    {
      "NAME": "destination_url",
      "NOTE": "URL",
      "TYPE": "string",
      "IS_DELETED": 0,
      "IS_SYSTEM": 1,
      "IS_MULTIPLE_VALUE": 0
    },
    {
      "NAME": "print_server",
      "NOTE": "Print server name",
      "TYPE": "string",
      "IS_DELETED": 0,
      "IS_SYSTEM": 1,
      "IS_MULTIPLE_VALUE": 0
    },
    {
      "NAME": "location",
```

```
    "NOTE": "Printer location",
    "TYPE": "string",
    "IS_DELETED": 0,
    "IS_SYSTEM": 1,
    "IS_MULTIPLE_VALUE": 0
  },
  {
    "NAME": "comment",
    "NOTE": "Comment to the printer",
    "TYPE": "string",
    "IS_DELETED": 0,
    "IS_SYSTEM": 1,
    "IS_MULTIPLE_VALUE": 0
  },
  {
    "NAME": "port_name",
    "NOTE": "Name of the printer port",
    "TYPE": "string",
    "IS_DELETED": 0,
    "IS_SYSTEM": 1,
    "IS_MULTIPLE_VALUE": 0
  },
  {
    "NAME": "copies",
    "NOTE": "The number of copies",
    "TYPE": "number",
    "IS_DELETED": 0,
    "IS_SYSTEM": 1,
    "IS_MULTIPLE_VALUE": 0
  },
  {
    "NAME": "task_name",
    "NOTE": "Task name",
    "TYPE": "string",
    "IS_DELETED": 0,
    "IS_SYSTEM": 1,
    "IS_MULTIPLE_VALUE": 0
```

```
},
{
  "NAME": "print_task_name",
  "NOTE": "Print task name",
  "TYPE": "string",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
  "IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "task_run_date",
  "NOTE": "Task start date",
  "TYPE": "date",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
  "IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "create_date",
  "NOTE": "File creation date",
  "TYPE": "date",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
  "IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "modify_date",
  "NOTE": "File modification date",
  "TYPE": "date",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
  "IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "source_ip",
  "NOTE": "IP address of the source",
  "TYPE": "string",
```

```
    "IS_DELETED": 0,
    "IS_SYSTEM": 1,
    "IS_MULTIPLE_VALUE": 0
  },
  {
    "NAME": "workstation_type",
    "NOTE": "Workstation type",
    "TYPE": "enum",
    "IS_DELETED": 0,
    "IS_SYSTEM": 1,
    "IS_MULTIPLE_VALUE": 0
  },
  {
    "NAME": "application_path_from",
    "NOTE": "Path to the application from which the event has been copied",
    "TYPE": "string",
    "IS_DELETED": 0,
    "IS_SYSTEM": 1,
    "IS_MULTIPLE_VALUE": 0
  },
  {
    "NAME": "application_path_to",
    "NOTE": "Path to the application, to which the event has been copied",
    "TYPE": "string",
    "IS_DELETED": 0,
    "IS_SYSTEM": 1,
    "IS_MULTIPLE_VALUE": 0
  },
  {
    "NAME": "terminal_session",
    "NOTE": "Terminal session indicator",
    "TYPE": "number",
    "IS_DELETED": 0,
    "IS_SYSTEM": 1,
    "IS_MULTIPLE_VALUE": 0
  },
  {
```

```
      "NAME": "emitter",
      "NOTE": "Event source",
      "TYPE": "enum",
      "IS_DELETED": 0,
      "IS_SYSTEM": 1,
      "IS_MULTIPLE_VALUE": 0
    },
    {
      "NAME": "charset",
      "NOTE": "Encoding of the attachment text",
      "TYPE": "string",
      "IS_DELETED": 0,
      "IS_SYSTEM": 1,
      "IS_MULTIPLE_VALUE": 0
    }
  ],
  "meta": {
    "totalCount": 20,
    "start": 0,
    "limit": 100
  }
}
```


5.4.7 Getting list of security objects catalogues

Description: List of security objects catalogues, sorted out by name.

Method: `GET`

Resource: `event/protectedCatalog`

Parameters:

- `start` - from what position the catalogue should be started to be given (default value is 0)
- `limit` - how many catalogues are to be given (default value is 100, but it can't be more than 1000)

Response: List of security objects catalogues, described in the JSON-Schema below:

JSON scheme of catalogues list output


```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "data": {
```

```
            "type": "array",
            "description": "Массив каталогов ОЗ",
            "items": {
                "type": "object",
                "description": "Каталог ОЗ",
                "properties": {
                    "CATALOG_ID": {
                        "type": "string",
                        "description": "Идентификатор"
                    },
                    "DISPLAY_NAME": {
                        "type": "string",
                        "description": "Имя"
                    },
                    "CREATE_DATE": {
                        "type": "string",
                        "description": "Дата создания"
                    },
                    "IS_DELETED": {
                        "type": "integer",
                        "description": "Удален ли из активной конфигурации",
                        "enum": [
                            0,
                            1
                        ]
                    }
                },
                "required": [
                    "CATALOG_ID",
                    "DISPLAY_NAME",
                    "CREATE_DATE",
                    "IS_DELETED"
                ]
            }
        },
        "meta": {
            "description": "Дополнительные данные",
            "type": "object",
            "properties": {
                "totalCount": {
                    "type": "integer",
                    "description": "Общее количество ОЗ"
                },
                "start": {
                    "type": "integer",
                    "description": "С какой позиции началось отдаваться"
                },
                "limit": {
                    "type": "integer",
                    "description": "Сколько всего отдалось"
                }
            },
            "required": [
                "totalCount",
                "start",
                "limit"
            ]
```

```
        }
    },
    "required": [
        "data",
        "meta"
    ]
}
```

Example:

Request

```
GET /xapi/event/protectedCatalog?start=0&limit=10
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: aplab
X-API-Auth-Token: gla20dos1pcdbr69caxz
```

Response

```
{
    "data": [{
        "CATALOG_ID": "3D0F8CDAF9E74F82AC85EB32D12727A700000000",
        "DISPLAY_NAME": "Грифованная информация",
        "CREATE_DATE": "2016-05-12 10:24:41.342554",
        "IS_DELETED": 0
    }],
    "meta": {
        "totalCount": 1,
        "start": 0,
        "limit": 10
    }
}
```

5.4.8 Getting the list of protected objects

Description: The list of protected objects sorted by name.

Method: `GET`

Resource: `event/protectedDocument`

Parameters:

- `start` - a position from which to start returning protected objects, the default value is 0
- `limit` - the number of protected objects to be returned; the default value is 100; cannot be more than 1000
- `with[]` — the list of additional entities to be added to the protected object:
    - `pd2pc` — a link of the protected object with the catalog of protected objects

Response: The list of protected objects described with the following JSON Schema:

JSON schema of returning the list of protected objects:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#" ,
    "type": "object",
    "properties": {
        "data": {
            "type": "array",
            "description": "Array of protected objects",
            "items": {
                "type": "object",
                "description": "Protected object",
                "properties": {
                    "DOCUMENT_ID": {
                        "type": "string",
                        "description": "Identifier"
                    },
                    "DISPLAY_NAME": {
                        "type": "string",
                        "description": "Name"
                    },
                    "CREATE_DATE": {
                        "type": "string",
                        "description": "Creation date"
                    },
                    "IS_DELETED": {
                        "type": "integer",
                        "description": "Whether it is removed from the active
configuration",
                        "enum": [
                            0,
                            1
                        ]
                    },
                    "pd2pc": {
                        "description": "Array of links to catalogs of protected
objects",
                        "type": "array",
                        "items": {
                            "description": "Link to the catalog of protected objects"
,
                            "type": "object",
                            "required": [
                                "CATALOG_ID",
                                "DOCUMENT_ID"
                            ],
                            "properties": {
                                "CATALOG_ID": {
                                    "description": "A unique identifier of the
catalog of protected objects",
                                    "type": "string"
                                },
                                "DOCUMENT_ID": {
                                    "description": "A unique identifier of the
protected object",
```

```
                                    "type": "string"
                                }
                            }
                        }
                    }
                },
                "required": [
                    "DOCUMENT_ID",
                    "DISPLAY_NAME",
                    "CREATE_DATE",
                    "IS_DELETED"
                ]
            }
        },
        "meta": {
            "description": "Additional data",
            "type": "object",
            "properties": {
                "totalCount": {
                    "type": "integer",
                    "description": "The total number of protected objects"
                },
                "start": {
                    "type": "integer",
                    "description": "The position from which returning has started"
                },
                "limit": {
                    "type": "integer",
                    "description": "The number of returned objects"
                }
            },
            "required": [
                "totalCount",
                "start",
                "limit"
            ]
        }
    },
    "required": [
        "data",
        "meta"
    ]
}
```

Example:

Query

```
GET /xapi/event/protectedDocument?start=0&limit=10
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
```

Response

```
{
    "data" : [{
        "DOCUMENT_ID" :  "C78E821A67C7F34A1B1D63A4151CEF6300000000" ,
        "DISPLAY_NAME" :  "Security labels" ,
        "CREATE_DATE" :  "2016-05-12 10:24:41.342554" ,
        "IS_DELETED" : 0
    }],
    "meta" : {
        "totalCount" : 1,
        "start" : 0,
        "limit" : 10
    }
}
```


5.4.9 Getting the list of policies

Description: The list of policies sorted by name.

Method: `GET`

Resource: `event/policy`

Parameters:

- `start` — a position from which to start returning policies, the default value is 0
- `limit` — the number of policies to be returned; the default value is 100; cannot be more than 1000

Response: The list of policies described with the following JSON Schema:

JSON schema of returing the list of policies:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#" ,
    "type": "object",
    "properties": {
        "data": {
            "type": "array",
            "description": "Array of policies",
            "items": {
                "type": "object",
                "description": "Policy",
                "properties": {
                    "POLICY_ID": {
                        "type": "string",
                        "description": "Identifier"
                    },
                    "DISPLAY_NAME": {
                        "type": "string",
                        "description": "Name"
                    },
                    "CREATE_DATE": {
                        "type": "string",
                        "description": "Creation date"
                    },
                    "IS_DELETED": {
                        "type": "integer",
                        "description": "Whether it is removed from the active
configuration",
                        "enum": [
                            0,
                            1
                        ]
                    }
                },
                "required": [
                    "POLICY_ID",
                    "DISPLAY_NAME",
                    "CREATE_DATE",
                    "IS_DELETED"
                ]
            }
        },
        "meta": {
            "description": "Additional data",
            "type": "object",
            "properties": {
                "totalCount": {
                    "type": "integer",
                    "description": "Total number of policies"
                },
                "start": {
                    "type": "integer",
                    "description": "A position where to start returning policies"
                },
```

```
                "limit": {
                    "type": "integer",
                    "description": "Total number of returned policies"
                }
            },
            "required": [
                "totalCount",
                "start",
                "limit"
            ]
        }
    },
    "required": [
        "data",
        "meta"
    ]
}
```

Example:

Query

```
GET /xapi/event/policy?start=0&limit=10
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
```

Response

```
{
    "data": [{
        "POLICY_ID": "3D0F8CDAF9E74F82AC85EB32D12727A700000000",
        "DISPLAY_NAME": "Security labels",
        "CREATE_DATE": "2016-05-12 10:24:41.342554",
        "IS_DELETED": 0
    }],
    "meta": {
        "totalCount": 1,
        "start": 0,
        "limit": 10
    }
}
```

5.4.10 Getting the List of Events

Description: The list of events sorted out by the time of interception (from new to old ones).

Method: `GET`

Resource: `event`

Parameters:

- `filter` - the filter list:

- `date[from]` — start date in UNIX-timestamp format (can be used separately from `date[to]`)
- `date[to]` — end date in UNIX-timestamp format (can be used separately from `date[from]`)
- `insert_date[>]` — start insert date in UNIX-timestamp format separated by microtime dot, for example, `filter[insert_date][>]=1234.123312`
- `insert_date[<]` – end insert date in UNIX-timestamp separated by microtime dot, for example, `filter[insert_date][<]=1234.123312`
- `insert_date[>=]` — start insert date in UNIX-timestamp format separated by microtime dot, for example `filter[insert_date][>=]=1234.123312` (incl.) (API 1.1 and later versions)
- `insert_date[<=]` — end insert date in UNIX-timestamp separated by microtime dot, for example `filter[insert_date][<=]=1234.123312` (incl.) (API 1.1 and later versions)
- `rule_group_type[]` — rule group (API 1.1 and later versions)
- `violation_level[]` — violation level (API 1.1 and later versions)
- `verdict[]` — verdict (API 1.1 and later versions)
- `object_type_code[]` — type of event (can be several values)
- `protocol[]` — protocol (can be several values)
- `user_decision[]` — user decision (can be several values)
- `object_size[from]` — the minimum number of bytes for event search (can be used separately from `object_size[to]`)
- `object_size[to]` — the maximum number of bytes for event search (can be used separately from `object_size[from]`)
- `protected_catalog_id[]` — an ID of the catalog with protected objects (can be several values)
- `tag_id[]` — tag ID (can be several values)

- `with[]` - the list of additional entities to be added to the object (for example, senders, recipients, etc.):
    - `protected_catalogs`
    - `protected_documents`
    - `policies`
    - `senders_keys`
    - `senders`
    - `recipients_keys`
    - `recipients`
    - `workstations_keys`
    - `workstations`
    - `tags`
    - `perimeters` (API 1.1 and later versions)
    - `attachments` (API 1.1 and later versions)
    - `files` (API 1.1 and later versions) — an undocumented parameter which returns all contents with unempty FILE_NAME field
  - `without[]` — additional flags
    - `count` — omit calculating `totalCount` (calculating `totalCount` may take more than halftime of the query, so we recommend using `event/stat` method instead)
  - `headers[]` — the list of additional fields (titles) of the event (for all fields, you * character). You can get the list with `event/header` method (API 1.1 and later versions)

- `sort[]` — an undocumented field for sorting the result (significantly slows down getting events; for getting the first/last event, we recommend using `event/stat` method. Possible keys:
  - `capture_date` — `asc` (API 1.1 and later vesions) `/desc`, for example, `?sort[capture_date]=desc`
  - `insert_date` — `asc/desc`, for example, `?sort[insert_date]=asc` (API 1.1 and later versions)

- `start` - a start position for returning events (the default value is 0)
- `limit` - how many events to return (the default value is 100, cannot be more than 1000)

Response: The list of events, sorted out by the time of interception (from new to old ones).

Examples:

Get the first 100 events of January 2015

```
GET /xapi/event?filter[date][from]=1420059600&filter[date][to]=1422737999&start=0&limit=100
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: aplab
X-API-Auth-Token: gla20dos1pcdbr69caxz
```

Steps to take when receiving events for the first time:

1. Define the date starting from which it is required to receive events. For example, to receive an interval INSERT_DATE from the events statistics (GET /xapi/event/stat).
2. Divide the interval by the smalest value: number of streams (not higher than 32) **or** number of cores on server.
3. In each stream, create and run the first request to receive a list of event which will look as follows:
   GET /xapi/event?without[]=count&filter[insert_date][>=]={interval_minimum}
   &filter[insert_date][<=]={interval_maximum}&sort[insert_date]=asc&limit=1000
4. Remember all OBJECT_ID in the received list.
5. In the received list, find the last INSERT_DATE and use it as `interval_minimum` in the request described at step 3.
6. Exclude the events which were remembered in the last iteration, from the new list.
7. If no `interval_maximum`, start again from step 4.
8. Save to persistent storage:
   a. INSERT_DATE maximum value, received at step 1.
   b. OBJECT_ID list from the interval, which includes INSERT_DATE maximum value.

5.4.11 Getting tags list

Description: Tags list, sorted out by name.

Method: `GET`

Resource: `event/tag`

Parameters:

- `start` - from what position the catalogue should be started to be given (default value is 0)
- `limit` - how many catalogues are to be given (default value is 100, but it can't be more than 1000)

Response: List of tags, described in the JSON-Schema below:

JSON scheme of tags list output


```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "data": {
            "type": "array",
            "description": "Массив тегов",
            "items": {
                "type": "object",
                "description": "Тег",
                "properties": {
                    "TAG_ID": {
                        "type": "string",
                        "description": "Идентификатор"
                    },
                    "DISPLAY_NAME": {
                        "type": "string",
                        "description": "Имя"
                    },
                    "COLOR": {
                        "type": "string",
                        "description": "Цвет тега в HEX RGP, например #ffffff"
                    },
                    "CREATE_DATE": {
                        "type": "string",
                        "description": "Дата создания"
                    },
                    "IS_DELETED": {
                        "type": "integer",
                        "description": "Удален ли из активной конфигурации"
                    },
                    "IS_SYSTEM": {
                        "type": "integer",
                        "description": "Является ли тег системным"
                    }
                },
                "required": [
                    "TAG_ID",
                    "DISPLAY_NAME",
                    "COLOR",
                    "CREATE_DATE",
                    "IS_DELETED",
                    "IS_SYSTEM"
                ]
            }
        },
        "meta": {
            "description": "Дополнительные данные",
            "type": "object",
            "properties": {
```

```
                "totalCount": {
                    "type": "integer",
                    "description": "Общее количество ОЗ"
                },
                "start": {
                    "type": "integer",
                    "description": "С какой позиции началось отдаваться"
                },
                "limit": {
                    "type": "integer",
                    "description": "Сколько всего отдалось"
                }
            },
            "required": [
                "totalCount",
                "start",
                "limit"
            ]
        }
    },
    "required": [
        "data",
        "meta"
    ]
}
```

Example:

Request

```
GET /xapi/event/tag?start=0&limit=10
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: aplab
X-API-Auth-Token: gla20dos1pcdbr69caxz
```

Response

```
{
    "data": [{
        "TAG_ID": "0AF69DD5742E6977E0530100007F2BB700000000",
        "DISPLAY_NAME": "VIP",
        "COLOR": "#4e954e",
        "CREATE_DATE": "2016-05-12 10:24:40.002673",
        "IS_DELETED": 0,
        "IS_SYSTEM": 1
    }],
    "meta": {
        "totalCount": 1,
        "start": 0,
        "limit": 10
    }
}
```

5.4.12 Getting statistics on events

Description: An undocumented method of getting statistics on events, for now, it only provides statistics by the interception date.

Method: `GET`

Resource: `event/stat`

Parameters:

- fields — a list of fields to be selected: CAPTURE_DATE , INSERT_DATE , totalCount (API 1.2 version)
- `filter` — a filtering list:
  - `date[from]` — start date in the UNIX-timestamp format (can be used separately from `date[to]`)
  - `date[to]` — end date in the UNIX-timestamp format (can be used separately from `date[from]`)
  - `insert_date[>]` — date of start pasting in the UNIX-timestamp format and via microtime point, for example `filter[insert_date][>]=1234.123312` (starting from API 1.1 version)
  - `insert_date[<]` — date of end pasting in the UNIX-timestamp format and via microtime point, for example `filter[insert_date][>]=1234.123312` (starting from API 1.1 version)
  - `insert_date[>=]` — date of start pasting in the UNIX-timestamp format and via microtime point, for example `filter[insert_date][>]=1234.123312`(including) (starting from API 1.1 version)
  - `insert_date[<=]` — date of end pasting in the UNIX-timestamp format and via microtime point, for example `filter[insert_date][>]=1234.123312`(including) (starting from API 1.1 version)
  - `object_type_code[]` — type of event (could be few values) (starting from API 1.1 version)
  - `violation_level[]` — level of violation (starting from API 1.1 version)

Ответ: Statistics on events described with the following JSON Schema:

JSON return schema:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "data": {
            "description": "Empty section",
            "items": {},
            "type": "array"
        },
        "meta": {
            "description": "Additional data",
            "type": "object",
            "properties": {
                "CAPTURE_DATE": {
                    "description": "Statistics by field CAPTURE_DATE",
                    "type": "object",
                    "properties": {
                        "from": {
                            "type": "string",
                            "description": "The date of capture of the first event in
selection"
                        },
                        "to": {
                            "type": "string",
                            "description": "The date of capture of the last event in
selection"
                        }
                    }
                },
                "INSERT_DATE": {
                    "description": "Statistics by field INSERT_DATE (API 1.1
version)",
                    "type": "object",
                    "properties": {
                        "from": {
                            "type": "string",
                            "description": "The date of insertion of the first event
in selection"
                        },
                        "to": {
                            "type": "string",
                            "description": "The date of insertion of the last event
in selection"
                        }
                    }
                },
                "totalCount": {
                    "type": "integer",
                    "description": "Total number of events"
                }
            }
        }
    }
```

```
}
```

Example:

Query

```
GET /xapi/event/stat?filter[data][from]=0
X-API-Version: 1.3
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugp1cdbr69axz
```

Response

```
{
    "data": [],
    "meta": {
        "CAPTURE_DATE": {
            "from": "2017-04-16 00:00:04",
            "to": "2017-05-16 16:20:36"
        },
        "INSERT_DATE": {
            "from": "2017-04-16 00:01:07",
            "to": "2017-05-16 16:20:40"
        },
        "totalCount": 225075
    }
}
```

5.5 Retrieving the list of Console users

Description: The list of Traffic Monitor Console users (including former employees)

Method: `GET`

Resource: `user`

Parameters (are passed in JSON format):

- `start` - a position from which user data should be delivered (the default value is 0)
- `limit` - how many user data sets should be delivered (the default value is 100, the value cannot be larger than 1000)

JSON scheme for delivering user data

```
schemas:
    user:
      type: object
      additionalProperties: false
      properties:
        USER_ID:
          type: integer
          description: User ID
        USERNAME:
          type: string
          description: Login
```

```
        DISPLAY_NAME:
          type: string
          description: Full name
        STATUS:
          type: integer
          description: Status (0 - active, 1 - disabled, 2 - deleted)
          enum: [0, 1, 2]
        CREATE_DATE:
          type: string
          description: Create Date with format "Y-m-d H:i:s.u" in UTC
        EMAIL:
          type: string
          description: E-mail
        NOTE:
          type: string
          description: Note
```