



Получение доступа к событиям Traffic Monitor внешними системами (DataExport API SDK)

03/02/2025

Table of Contents

1.1	Функции DataExport API.....	2
1.2	Авторизация для доступа к DataExport API	4
1.2.1	Работа с плагинами	4
1.2.2	Работа с токенами.....	4
1.3	Регистрация стороннего компонента DataExport API.....	5
1.3.1	Формат файла регистрации стороннего компонента DataExport API.....	6
1.4	Функциональное описание программного интерфейса DataExport API	10
1.4.1	Общие заголовки GET/POST-запросов для DataExport API.....	10
1.4.2	Общие ошибки DataExport API	13
1.4.3	Получение списка фич по версии Traffic Monitor	18
1.4.4	Получение данных аудита	19
1.4.5	Получение организационной структуры	27
1.4.6	Получение событий.....	53
1.4.7	Получение списка пользователей консоли	86
1.4.8	Получение результатов выполнения запроса Traffic Monitor	87

1 Получение доступа к событиям Traffic Monitor внешними системами (DataExport API SDK)

Данный программный интерфейс предназначен для интеграции Traffic Monitor с внешними системами-приемниками данных. В рамках реализации DataExport API внешние системы получают из Traffic Monitor содержимое и метаданные событий, данные организационной структуры, списки пользователей Консоли управления и данные аудита их действий. DataExport API поддерживает двусторонний обмен данными между Traffic Monitor и внешними системами.

1.1 Функции DataExport API

DataExport API поддерживает следующий набор функций:

- - Возможность указать набор атрибутов, которые будут включены в результат:
 - ID события;
 - Отправители: список контактов и результат идентификации с учетом возможных конфликтов;
 - Получатели: список контактов и результат идентификации с учетом возможных конфликтов;
 - Компьютер отправителя с учетом возможных конфликтов;
 - Тип события;
 - Дата и время перехвата;
 - Протокол;
 - Уровень нарушения;
 - Вердикт;
 - Путь к файлу или адрес приемника;
 - Путь к файлу или адрес источника;
 - Решение пользователя;
 - Путь к файлу;
 - Адрес веб-ресурса;
 - Объекты защиты и каталоги;
 - Размер события;
 - Теги;
 - Набор пользовательских атрибутов, зарегистрированных через плагин;
 - Набор системных заголовков событий;
 - Дата и время вставки события в БД;
 - Наименование периметра, в которое вошло событие;
 - Наименование периметра, из которых ушло событие ;
 - Список наименований политик (с ID политик), сработавших на перехваченном объекте;
 - Группа правил;
 - Сервер перехвата;
 - Сервер перехвата IP;
 - Имя устройства приемника;

- Имя устройства источника;
- Тема письма;
- Вложение;
- Мандатный уровень;
- Мандатная категория.
- Возможность указать набор атрибутов, по которым будет осуществляться фильтрация:
 - Дата и время перехвата;
 - Тип события;
 - Протокол;
 - Каталог ОЗ;
 - Решение пользователя;
 - Размер события;
 - Теги;
 - Дата и время вставки события в БД;
 - Группа правил;
 - Уровень нарушения;
 - Вердикт;
 - Дата и время зарегистрированного действия пользователя;
 - Тип объекта, над которым осуществлялось действие.
- Возможность изменить в Traffic Monitor атрибуты события:
 - Вердикт пользователя по событию;
 - Вердикт пользователя по множеству событий.
- Получение метаданных отдельного события с возможностью указать набор атрибутов, которые будут включены в результат. При этом если в событии есть нескольких вложенных объектов, то весь извлеченный текст будет склеен в один файл;
- Получение всего извлеченного текста отдельного события;
- Получение полного списка системных заголовков и зарегистрированных с помощью плагина пользовательских атрибутов событий;
- Получение бинарных данных отдельного события;
- Получение списка пользователей Консоли управления;
- Запуск на выполнение и получение результатов выполнения запросов в Traffic Monitor:
 - список запросов;
 - статус выполнения запросов;
 - результаты запросов;
- Получение данных аудита пользователей:
 - ID записи в аудите сессий пользователей консоли;
 - Дата и время зарегистрированного действия пользователя в UTC;
 - ID пользователя, осуществившего действие;
 - Полное имя пользователя, осуществившего действие;
 - Тип действия, которое было произведено пользователем;
 - Тип объекта, над которым осуществлялось действие;

- ID объекта, над которым осуществлялось действие;
- Наименование объекта, над которым осуществлялось действие;
- Описание произошедших изменений с указанием новых и старых значений.

1.2 Авторизация для доступа к DataExport API

Авторизация для доступа к DataExport API предоставляется по механизму токенов, аналогичному другим API InfoWatch Traffic Monitor. Для получения токена и просмотра статуса лицензирования DataExport API в Системе необходимо добавить плагин нового типа - Плагин для DataExport API. Все действия по работе с плагинами выполняются в консоли управления Traffic Monitor.

1.2.1 Работа с плагинами

1.2.1.1 Добавление плагина

Чтобы добавить плагин:

1. Перейдите в раздел **Управление - Плагины**;
2. В области **Плагины** нажмите .
3. Нажмите **Добавить**.
4. Просмотрите описание плагина и нажмите **Установить**.

Примечание:

Чтобы обновить плагин, добавьте его обновленную версию.

1.2.1.2 Удаление плагина

Чтобы удалить плагин:

1. Перейдите в раздел **Управление - Плагины**;
2. В области **Плагины** нажмите .
3. Нажмите **Да**, чтобы подтвердить удаление.

1.2.2 Работа с токенами

Токен генерируется автоматически после добавления плагина. Все действия по работе с токенами выполняются в консоли управления InfoWatch Traffic Monitor.

1.2.2.1 Добавление токена

Чтобы добавить токен:

1. Перейдите в раздел **Управление - Плагины**;
2. Выберите плагин в области **Плагины**;
3. Перейдите в закладку **Токены**;
4. Нажмите .

Примечание:

Значение токена генерируется автоматически при создании токена и не может быть задано пользователем.

1.2.2.2 Редактирование данных токена

Чтобы изменить данные токена:

1. Перейдите в раздел **Управление - Плагины**;
2. Выберите плагин в области **Плагины**;
3. Перейдите в закладку **Токены**;
4. По двойному нажатию левой кнопкой мыши в графе *Имя* или *Описание* введите нужную информацию.

1.2.2.3 Обновление значения токена

Если токен скомпрометирован или появилась вероятность его утечки третьим лицам, нужно сгенерировать значение токена заново.

Чтобы обновить значение токена:

1. Перейдите в раздел **Управление - Плагины**;
2. Выберите плагин в области **Плагины**;
3. Перейдите в закладку **Токены**;

4. Нажмите  .

1.2.2.4 Копирование токена

Чтобы скопировать значение токена в буфер обмена:

1. Перейдите в раздел **Управление - Плагины**;
2. Выберите плагин в области **Плагины**;
3. Перейдите в закладку **Токены**;

4. Нажмите  .

1.2.2.5 Удаление токена

Чтобы удалить токен:

1. Перейдите в раздел **Управление - Плагины**;
2. Выберите плагин в области **Плагины**;
3. Перейдите в закладку **Токены**;

4. Нажмите  .

1.3 Регистрация стороннего компонента DataExport API

Для интеграции сторонней системы-приемника данных с системой Traffic Monitor нужно выполнить следующие шаги:

1. Зарегистрировать в компании InfoWatch строку-идентификатор компании-потребителя компонента. Это позволит связывать компанию-потребителя с лицензией, которая будет устанавливаться в систему Traffic Monitor;
2. Получить лицензию разработчика, связанную с идентификатором компании. Это позволит выгружать из Traffic Monitor данные о событиях;
3. Создать файл регистрации стороннего компонента в описанном ниже формате. В файл регистрации добавляется лицензия, полученная на шаге 2;
4. Загрузить файл регистрации стороннего компонента в систему Traffic Monitor через интерфейс консоли Traffic Monitor. Раздел «Управление» - «Плагины». (см. "Traffic Monitor. Руководство пользователя");

5. Получить токен доступа в свойствах загруженного стороннего компонента;
- Система готова отправлять данные о событиях заданной компании.

1.3.1 Формат файла регистрации стороннего компонента DataExport API

Плагин представляет собой архив в формате .zip. В состав архива входят:

- папка **licenses**, содержащая файлы лицензий;
- папка **icon**, содержащая файлы с используемыми пиктограммами для регистрируемых событий;
- файл **manifest.json**, содержащий информацию о плагине.

Примечание:

Папки **licenses** и **icon** не являются обязательными. Файлы лицензий и файлы с используемыми пиктограммами могут находиться в корне.

Для внешних систем-приемников данных из Traffic Monitor файл **manifest.json** должен содержать следующую информацию:

Содержимое	Тип данных	Является обязательным	Описание
{			
"PLUGIN_ID": "",	Строка	Да	Уникальный идентификатор (UUID) плагина, который создается его разработчиком. До 40 символов. Идентификатор используется для обновления плагина. Пример: "PLUGIN_ID": "189C38D390396EB6E0530100007F1CA200000001",
"DISPLAY_NAME": "",	Строка	Да	Отображаемое имя плагина. Пример: "DISPLAY_NAME": "Имя плагина",
"DESCRIPTION": "",	Строка	Нет	Отображаемое описание плагина. Пример: "DESCRIPTION": "Тестовый плагин для демонстрации",
"VERSION": "",	Строка	Да	Версия плагина. Пример: "VERSION": "1.0.0",
"VENDOR": "",	Строка	Да	Идентификатор компании-разработчика, соответствующий названию компании в лицензии. Пример: "VENDOR": "infowatch",
"LICENSE": [{ "PATH": "" }],	Массив Строка	Да	Файлы лицензии. Должны быть указаны пути к файлам лицензий относительно корня архива.

Содержимое	Тип данных	Является обязательным	Описание
			<pre>"LICENSE": [{ "PATH": "licenses/файл лицензии 1.license" }, { "PATH": "licenses/файл лицензии 2.license" },],</pre> Code Block 1 Пример
<code>"PATTERN_SEARCH_LICENSE": "",</code>	Строка	Нет	Шаблон для поиска загруженных ранее лицензий для привязки их к плагину. Имеет формат: <code>{operator:or and not,child[{name:value}]}</code>. <pre>"PATTERN_SEARCH_LICENSE": { "operator": "or", "conditions": [{ "origin": "dm" }, { "origin": "dmmobile" }], },</pre> Code Block 2 Пример operator:or

Содержимое	Тип данных	Является обязательным	Описание
			<pre>"PATTERN_SEARCH_LICENSE": { "operator": "or", "conditions": [{ "operator": "and", "conditions": [{"key1": "value1"}, {"key2": "value2"}, {"key3": "value3"}] }, { "operator": "and", "conditions": [{"key1": "value1"}, {"key2": "value2"}, {"key3": "value3"}] }] }</pre> Code Block 3 Пример operator:and <pre>"PATTERN_SEARCH_LICENSE": { "operator": "and", "conditions": [{"key1": "value1"}, {"key2": "value2"}, {"key3": "NONE"}, { "operator": "not", "conditions": [{"key4": null}] }] }</pre> Code Block 4 Пример operator:not
<pre>"DATA_RECEIVER": [""]</pre>	Массив Строка	Да	Список идентификаторов систем-приемников данных из Traffic Monitor. <pre>"DATA_RECEIVER": ["Events reciever1", "Events reciever2"]</pre> Code Block 5 Пример

Содержимое	Тип данных	Является обязательным	Описание
<pre>"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": "",}]</pre>	Массив Строка Объект СтрокаПеречисление ПеречислениеМассив строк Перечисление Перечисление Перечисление Перечисление Перечисление	Нет	Пользовательские атрибуты событий. Уникальный код атрибута, который будет использован в заголовках события и для идентификации в xAPI/PushAPI. Если идентификатор разработчика не "IW", код должен начинаться с префикса "<Идентификатор компании-разработчика>_". Пример: CISCO_CALL_DURATION Название атрибута, отображаемое в консоли Traffic Monitor. Тип значения. Формат записи. Возможны значения: число (целое, дробное), строка, дата и время в UTC + указание смещение часового пояса, длительность, гиперссылка, перечисление, логический тип. Сервисы, к событиям которых будет добавлен новый пользовательский атрибут. Использование атрибута в политиках. "1" - да, "0" - нет. Использование атрибута в почтовых уведомлениях. "1" - да, "0" - нет. Использование атрибута в запросах. "1" - да, "0" - нет. Отображение атрибута в табличном режиме просмотра событий. "1" - да, "0" - нет. Отображение атрибута в краткой форме просмотра события. "1" - да, "0" - нет. Отображение атрибута в детальной форме просмотра события. "1" - да, "0" - нет. Множественное значение. "1" - да, "0" - нет.

Содержимое	Тип данных	Является обязательным	Описание
			<pre> "OBJECT_HEADER": [{ "NAME": "header_multi_string", "NOTE": { "rus": "Заголовок строка1", "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" }] </pre>
}			

Code Block 6 Пример

1.4 Функциональное описание программного интерфейса DataExport API

1.4.1 Общие заголовки GET/POST-запросов для DataExport API

1.4.1.1 Авторизация

Токен, полученный для Public API, необходимо указывать в заголовке **X-API-Auth-Token** при каждом запросе к API. Токен может быть отозван/заблокирован/удалён через web-интерфейс IWTM. В случае, если это произойдет, токен считается недействительным. При указании недействительного токена возвращается сообщение об ошибке с кодом **401**.

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

Code Block 7 Пример ответа при недействительном токене

Если токен действующий, но у пользователя API недостаточно прав, то пользователю возвращается сообщение об ошибке с кодом **403**.

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

Code Block 8 Пример ответа при нехватке прав

1.4.1.2 Версионирование

Версия API передается с помощью заголовка **X-API-Version** и может иметь 3 состояния:

- **supported** — версия поддерживается и активно развивается;
- **deprecated** — версия доступна, но перестала поддерживаться. Рекомендуется перейти на новую версию, так как в следующих выпусках эта версия может быть удалена;
- **removed** — версия полностью удалена из продукта.

В случае, если сервер не поддерживает версию, ему возвращается сообщение об ошибке с кодом **400**.

```

{
  "error": {
    "code": "unsupported_version",
    "meta": {
      "versions": [
        {
          "NAME": "1",
          "STATUS": "removed"
        },
        {
          "NAME": "1.1",
          "STATUS": "deprecated"
        },
        {
          "NAME": "1.2",
          "STATUS": "deprecated"
        },
        {
          "NAME": "1.3",
          "STATUS": "deprecated"
        },
        {
          "NAME": "1.4",
          "STATUS": "deprecated"
        },
        {
          "NAME": "1.5",
          "STATUS": "deprecated"
        },
        {
          "NAME": "1.6",
          "STATUS": "deprecated"
        },
        {
          "NAME": "1.7",
          "STATUS": "supported"
        }
      ]
    }
  }
}

```

Code Block 9 Ответ в случае некорректной версии

1.4.1.3 Лицензирование

Для работы модуля требуется лицензия:

```

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

```

Где *common_name* — это название лицензируемой компании, а *importer_name* — продукт, который принимает данные от ТМ.В каждом запросе нужно передавать заголовки **X-API-CompanyId** и **X-API-ProductName**. В случае, если лицензия не проходит проверку, возвращается сообщение об ошибке с кодом **403**.

```

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

```

Code Block 10 Формат ответа при невалидной лицензии

Таким образом, каждый запрос к серверу должен содержать следующие обязательные заголовки:

- X-API-Version – указывается номер версии программного интерфейса. Текущая версия: **1.6**;
- X-API-CompanyId – указывается идентификатор компании-производителя, например: **iw**;
- X-API-ProductName – указывается продукт, принимающий данные, например: **dome**;
- X-API-Auth-Token – указывается токен авторизации.

```
X-API-Version: 1.6
X-API-CompanyId: iw
X-API-ProductName: dome
X-API-Auth-Token: gla20dos1pcdbr69caxz
```

Code Block 11 Пример запроса

1.4.2 Общие ошибки DataExport API

Описание возможных ошибок:

- not_unique_field – неуникальное значение;
- unique – уникальное значение;
- too_short – слишком короткое значение;
- too_long – слишком длинное значение;
- wrong_length – значение неправильной длины;
- repeat_exactly – значение не совпадает со значением в другом поле;
- required – у поля обязательно должно быть значение;
- not_exist_in_list – значение не входит в список;
- invalid – значение невалидно;
- contains_child – узел является не листовым;
- contains_items – узел категории содержит сущности;
- parent_own_child – родитель содержит детей;
- parent_self_set_current – попытка установить родителем самого себя;
- not_valid_json – JSON невалиден;
- invalid_cron_format – не правильный формат cron;
- read_only – объект доступен только для чтения;
- empty – значение не может быть пустым;
- datastorage_connection_error – нет соединения с Data Storage;
- datastorage_error – ошибка в работе с Data Storage.

Ошибка	Код возв рата	Описание и формат/тело ответа
Успешны й ответ	200	<pre>{ "data": "mixed", "meta": "mixed" }</pre> Code Block 12 Формат ответа:
Ошибка валидац ии данных/в ерсии	400	<pre>{ "properties": { "error": { "description": "Контейнер ошибки", "properties": { "code": { "description": "Общее описание ошибки", "type": "string", "value": "unsupported_version" }, "meta": { "description": "Дополнительные данные об ошибке", "type": "object", "properties": { "supported_versions": { "description": "Массив поддерживаемых версий", "type": "array", "items": { "description": "Поддерживаемая версия", "type": "string" } } } } } } } }</pre> Code Block 13 Формат ответа в случае неправильной версии API (JSON Scheme)

```

{
  "properties": {
    "error": {
      "description": "Контейнер ошибки",
      "properties": {
        "code": {
          "description": "Общее описание ошибки",
          "type": "string",
          "value": "validation"
        },
        "meta": {
          "description": "Дополнительные данные об
ошибке",
          "type": "object",
          "properties": {
            "validation": {
              "description": "Название полей модели",
              "type": "object",
              "properties": {
                "field": {
                  "description": "Поле модели",
                  "type": "array",
                  "items": {
                    "description": "Ошибка",
                    "type": "string",
                    "values": [
                      "not_unique_field",
                      "too_short",
                      "too_long",
                      "wrong_length",
                      "repeat_exactly",
                      "required",
                      "not_exist_in_list",
                      "invalid",
                      "contains_child",
                      "contains_items",
                      "parent_own_child",
                      "unique",
                      "not_valid_json",
                      "invalid_cron_format",
                      "read_only",
                      "parent_self_set_curren
t",
                      "empty"
                    ]
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}

```

Code Block 14 Формат ответа в случае ошибки валидации запроса (JSON Scheme)

Ошибка	Код возв рата	Описание и формат/тело ответа
		<pre> { "error": { "code": "unsupported_version", "meta": { "supported_versions": ["2", "3"] } } } </pre> Code Block 15 Пример ошибки при указании неправильной версии
Необходима авторизация	401	Возвращается в любом запросе, где требуется авторизация. Тело ответа: нет.
Недостаточно прав	403	Возвращается в любом запросе, если у текущего пользователя недостаточно прав для выполнения данной операции. Тело ответа: нет.
Сущность в системе не найдена	404	Возвращается, если определенная модель не найдена в системе. Например, не найдена эталонная выгрузка. Тело ответа: нет
Тело запроса слишком большое	413	Возвращается, если передан слишком большой запрос. Тело ответа: нет.
Превышено количество запросов к серверу	429	возвращается, если на сервер отправлено слишком много запросов за определенный промежуток времени. Может сопровождаться заголовком Retry-After, указывающим, через какое время можно повторить запрос.

Ошибка	Код возв рата	Описание и формат/тело ответа
		<pre>{ "properties": { "error": { "description": "Контейнер ошибки", "properties": { "code": { "description": "Общее описание ошибки", "type": "string", "value": "too_many_requests " }, "meta": { "description": "Дополнительные данные об ошибки", "type": "object", "properties": { "time": { "description": "HTTP-дата, либо целое число в секундах, через которые можно повторить запрос", "type": "string" } } } } } } }</pre> Code Block 16 Тело ответа (JSON Scheme):
Ошибка системы	500	Возвращается, если в системе произошла логическая ошибка, и далее невозможно выполнение запроса. <pre>{ "properties": { "error": { "description": "Контейнер ошибки", "properties": { "code": { "description": "Общее описание ошибки", "type": "string" }, "meta": { "description": "Дополнительные данные об ошибке", "type": "object" } } } } }</pre> Code Block 17 Тело ответа (JSON Scheme):
Сервер временн о не доступен	503	Информирует, что в данный момент невозможно выполнить запросы, и сервер находится на техническом обслуживании. Приблизительное время окончания технического обслуживания содержится в заголовке Retry-After (значением этого заголовка может быть либо HTTP-дата, либо целое число в секундах).

Ошибка	Код возв рата	Описание и формат/тело ответа
		<pre> { "properties": { "error": { "description": "Контейнер ошибки", "properties": { "code": { "description": "Общее описание ошибки", "type": "string", "value": "maintains" }, "meta": { "description": "Дополнительные данные об ошибке", "type": "object", "properties": { "time": { "description": "HTTP-дата, либо цело число в секундах, через которые можно повторить запрос", "type": "string" } } } } } } } </pre> Code Block 18 Тело ответа (JSON Scheme):

1.4.3 Получение списка фич по версии Traffic Monitor

Описание: Список фич для текущей версии Traffic Monitor.

Метод: *GET*

Ресурс: version/features

Параметры:

```

curl -k -i -X GET      -H "X-API-Auth-Token:1lap4tvpakbgsbj8tyeu"      -H "X-API-
Version:1.3"          -H "X-Enable-Openapi:1"          -H "X-API-CompanyId:iw"      -H "X-API-
ProductName:dome"     'https://server.company.ru/xapi/version/features'

```

Code Block 19 Пример запроса:

```

{
  "data": [
    {
      "VERSION_TM": "7.3.0.285",
      "VERSION_API": "1.6",
      "FEATURES": "sso - сквозная авторизация из vision в tm"
    }
  ]
}

```

Code Block 20 Пример ответа:

1.4.4 Получение данных аудита

Примечание:

В API получения данных аудита могут появляться новые действия. При этом версия API может не меняться.

Описание: Список данных аудита, отсортированных по дате и времени зарегистрированного действия пользователя.

Метод: `GET`

Ресурс: `audit`

Параметры:

- `filter` – список фильтрации:
 - `change_date[>]` – дата и время начала зарегистрированного действия пользователя в формате UNIX-timestamp и через точку microtime, например `filter[change_date][>]=1591368000.123456`
 - `change_date[<]` – дата и время окончания зарегистрированного действия пользователя в формате UNIX-timestamp и через точку microtime, например `filter[change_date][<]=1591368000.123456`
 - `change_date[>=]` – дата и время начала зарегистрированного действия пользователя в формате UNIX-timestamp и через точку microtime, например `filter[change_date][>=]=1591368000.123456`
 - `change_date[<=]` – дата и время окончания зарегистрированного действия пользователя в формате UNIX-timestamp и через точку microtime, например `filter[change_date][<=]=1591368000.123456`
 - `entity_type[]` – тип сущности, над которой было выполнено действие (возможно несколько значений);
 - `operation[]` – операция-действие пользователя (возможно несколько значений);
- `without[]` – получить данные без дополнительных атрибутов:
 - `property_changes` – данные расширенного аудита.
- `sort[]` – параметр для сортировки:
 - `change_date` — сортировать по дате изменения `asc`.
- `start` – с какой позиции начать отдавать данные аудита (по умолчанию 0)
- `limit` – сколько комплектов данных аудита отдавать (по умолчанию 100, не может быть больше 1000)

```

schemas:
  auditCommon:
    type: object
    required: [AUDIT_LOG_ID, CHANGE_DATE, OPERATION, ENTITY_TYPE]
    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
        - bruteforce
        - login
        - logout
        - change_password
        - commit
        - rollback
        - draft
        - add
        - import
        - export
        - export_object
        - create_widget
        - update_widget
        - delete_widget
        - 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
- upload_image
- delete_image
- create_contact
- update_contact
- delete_contact
- compile_create
- compile_update
- create_template
- update_template
- delete_template
- create_rule
- update_rule
- delete_rule
- 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
    nullable: true
    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
userBruteForce:
  type: object
  description: For ENTITY_TYPE User and OPERATION login with bruteforce
  properties:
    <<: *common
    OPERATION:
      type: string
      enum: [ bruteforce ]
    ENTITY_TYPE:
      type: string
      enum: [ User ]
    PROPERTY_CHANGES:
      type: object
      nullable: true
      required: [ login ]
      properties:
        request:
          required:
            - login
            - display_name
          properties:
            login:
              type: string
              title: User login
              description: Entered user login in input
            display_name:
              type: string
              title: User full name
              description: User full name
tagPropertyChanges:
  type: object
  properties:
    <<: *common
    ENTITY_TYPE:
      type: string
      enum: [ Object ]
    OPERATION:
      type: string

```

```

        enum: [ add_tag, remove_tag ]
    PROPERTY_CHANGES:
        type: object
        properties:
            new:
                $ref: '#/components/schemas/TagDiffProperties'
            old:
                $ref: '#/components/schemas/TagDiffProperties'
    TagDiffProperties:
        type: object
        properties:
            DISPLAY_NAME:
                type: string
            NOTE:
                type: string
    userDesicionPropertyChanges:
        type: object
        properties:
            <<: *common
            ENTITY_TYPE:
                type: string
                enum: [ Object ]
            OPERATION:
                type: string
                enum: [ decision_update, xapi_decision_update ]
            PROPERTY_CHANGES:
                type: object
                properties:
                    new:
                        type: object
                        properties:
                            USER_DECISION:
                                $ref: '#/components/schemas/UserDecisionState'
                    old:
                        type: object
                        properties:
                            USER_DECISION:
                                $ref: '#/components/schemas/UserDecisionState'
    UserDecisionState:
        type: string
        enum: [ NotProcessed, NoViolation, Violation, AdditionalProcessingNeeded ]
    objectExportPropertyChanges:
        type: object
        properties:
            <<: *common
            ENTITY_TYPE:
                type: string
                enum: [ Object ]
            OPERATION:
                type: string
                enum: [ export_object ]
            PROPERTY_CHANGES:
                type: object
                properties:
                    new:
                        type: object
                        properties:
                            PARAMS:
                                type: object
                                properties:
                                    QUERY_ID:
                                        type: integer
                                        nullable: true
                                    DISPLAY_NAME:
                                        type: string
                                    COMMENT:
                                        type: string
                                    FILTER:
                                        anyOf:

```

```

        - type: object
        - type: array
    SORT:
        type: object
    IS_ONE_REPORT:
        $ref: '#/components/schemas/BooleanAsInteger'
    ONE_LOAD_REPORT:
        $ref: '#/components/schemas/BooleanAsInteger'
    FORMAT:
        type: string
        enum: [ simple, full ]
    TYPE:
        type: object
        properties:
            one_report:
                type: string
                enum: [ docx, xls, xlsx, pdf ]
            several_report:
                type: string
                enum: [ docx, pdf ]
    IS_SEVERAL_REPORT:
        $ref: '#/components/schemas/BooleanAsInteger'
    SEVERAL_LOAD_REPORT:
        $ref: '#/components/schemas/BooleanAsInteger'
    LOAD_ATTACHMENT:
        $ref: '#/components/schemas/BooleanAsInteger'
    KEEP_HIERARCHY:
        $ref: '#/components/schemas/BooleanAsInteger'
    LOAD_SNIPPET:
        $ref: '#/components/schemas/BooleanAsInteger'
    LOAD_EML:
        $ref: '#/components/schemas/BooleanAsInteger'
    LOAD_DEBUG_OBJECT:
        $ref: '#/components/schemas/BooleanAsInteger'
    SCOPE:
        oneOf:
            - type: string
              enum: [ all ]
            - type: array
              items:
                type: integer
    GROUPING:
        type: array
        items:
            type: string
            enum: [CaptureDate, Computer, Object, ObjectType, Person]
    IS_REPORT_ARCHIVE:
        $ref: '#/components/schemas/BooleanAsInteger'
    BooleanAsStringOrInteger:
        $ref: 'common.yaml#/components/schemas/BooleanAsStringOrInteger'
    BooleanAsInteger:
        $ref: 'common.yaml#/components/schemas/BooleanAsInteger'
    BooleanAsString:
        $ref: 'common.yaml#/components/schemas/BooleanAsString'

```

Code Block 21 OpenAPI-схема отдачи данных аудита

Пример:

```

curl -X GET
"https://server.company.ru/xapi/audit?sort[change_date]=asc&filter[entity_type][]=
Adlibitum&filter[entity_type][]=Audit&filter[entity_type][]=Config&filter[entity_t
ype][]=Perimeter&filter[entity_type][]=Policy&start=0&limit=5" -H "accept:
application/json" -H "X-Enable-OpenAPI: 1"

```

Code Block 22 Запрос


```

{
  "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
}
}

```

Code Block 23 Ответ

1.4.5 Получение организационной структуры

1.4.5.1 Получение информации по группам

1.4.5.1.1 Получение списка групп

Описание: Список групп

Метод: `GET`

Ресурс: `group`

Параметры:

- `filter` — список фильтрации:
 - `change_date[from]` — от даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[to]`)
 - `change_date[to]` — до даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[from]`)
 - `create_date[from]` — от даты создания в формате UNIX-timestamp (можно использовать отдельно от `create_date[to]`)

- `create_date[to]` — до даты создания в формате UNIX-timestamp (можно использовать отдельно от `create_date[from]`)
 - `is_deleted` — флаг 1 или 0. Если параметр указан, то будут выдавать сущности с указанным значением
 - `disabled` — флаг 1 или 0, признак активности. Если выставлен, то сущность неактивна (признак берется из LDAP)
 - `contact_type[]` — фильтрация по типу контакта (можно передать несколько, например: `/xapi/group?filter[contact_type][]=webaccount&filter[contact_type][]=icq`)
 - `contact_value[]` — фильтрация по значению контакта (можно передавать несколько, например: `/xapi/group?filter[contact_value][]=user@ad&filter[contact_value][]=login@example.com`)
 - `parent_group_id[]` — фильтрация по группам (можно передать несколько, например: `/xapi/group?filter[parent_group_id][]=FCE1409680E5D6469E1F7998266CE9E9&filter[parent_group_id][]=918462B6DC2C954BAAF8706F7DC53F14`)
- `with[]` — список дополнительных сущностей, которые должны быть добавлены к объекту:
 - `contacts` — контакты группы
 - `children` — подчиненные группы
- `sort` — параметр для сортировки
 - `change_date` — сортировать по дате изменения `asc` или `desc` (например: `/xapi/group?sort[change_date]=asc`) ⚠ может замедлить выдачу
- `start` — с какой позиции начать отдавать элементы, по умолчанию 0
- `limit` — сколько элементов отдавать, по умолчанию 100, не может быть больше 1000

Ответ: Список персон, описанный следующей [JSON-Schema](#):

JSON схема отдачи

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "required": [
    "data",
    "meta"
  ],
  "properties": {
    "data": {
      "description": "Массив групп",
      "type": "array",
      "items": {
        "description": "Группа",
        "type": "object",
        "required": [
          "GROUP_ID",
          "CREATE_DATE",
          "CHANGE_DATE",
          "DISPLAY_NAME",
          "DISTINGUISHEDNAME",
          "GROUP_TYPE",
          "IS_VISIBLE",
          "IS_DELETED",
          "childrenCount"
        ],
        "properties": {
          "GROUP_ID": {
            "description": "Уникальный идентификатор группы",
            "type": "string"
          },
          "CREATE_DATE": {
            "description": "Дата создания записи",
            "type": "string"
          },
          "CHANGE_DATE": {
            "description": "Дата изменения"
```

```

записи",
    "type": "string"
},
    "DISPLAY_NAME": {
        "description": "Название
группы",
        "type": "string"
},
    "DISTINGUISHEDNAME": {
        "description": "Полное
имя",
        "type": "string"
},
    "GROUP_TYPE": {
        "description": "Тип группы
(tmRoot, tmGroup, adRoot, adGroup, adContainer, adDomain, adOU from LDAP
objectclass)",
        "type": "string"
},
    "IS_VISIBLE": {
        "description": "Видимость
группы",
        "type": "integer"
},
    "IS_DELETED": {
        "description": "Удалена
ли группа из
ТМ",
        "type": "integer"
},
    "childrenCount": {
        "description": "Количество детей в
группе",
        "type": "integer"
},
    "contacts": {
        "description": "Массив
контактов",
        "type": "array",
        "items": {
            "description": "Контакт",
            "type": "object",
            "required": [
                "CONTACT_ID",
                "CHANGE_DATE",
                "VALUE",
                "IS_PRIMARY",
                "IS_PERSONAL",
                "SOURCE"
            ],
            "properties": {
                "CONTACT_ID": {
                    "description": "Уникальный идентификатор
объекта",
                    "type": "string"
},
                "CREATE_DATE": {
                    "description": "Дата
создания",
                    "type": "string"
},
                "CHANGE_DATE": {
                    "description": "Дата
изменения",
                    "type": "string"
},
                "CONTACT_TYPE": {
                    "description": "Тип",
                    "type": "string"
},
                "VALUE": {
                    "description": "Значение",
                    "type": "string"
},
                "IS_PERSONAL": {
                    "description": "Принадлежность (Рабочий,
Личный)",
                    "type": "integer"
},
                "IS_PRIMARY": {
                    "description": "Признак
основной",
                    "type": "integer"
},
                "SOURCE": {
                    "description": "Тип",
                    "type": "string"
}
}
},
    "children": {
        "description": "Массив групп
принадлежащих текущей
группе",
        "type": "array",
        "items": {
            "description": "Группа",
            "type": "object",
            "properties": {
                "GROUP_ID": {
                    "description": "Уникальный идентификатор
группы",
                    "type": "string"
}
}
}
}

```

```

    },
    "PARENT_GROUP_ID":
{
    "description": "Уникальный идентификатор
группы
родителя",
    "type": "string"
},
    "RELATION_TYPE":
{
    "description": "Тип связи: 0 - tm, 1 -
member, 2 -
distinguishedname",
    "type": "integer"
}
},
    "meta":
{
    "description": "Дополнительные
данные",
    "type": "object",
    "required":
[
    "totalCount",
    "start",
    "limit"
],
    "properties":
{
    "totalCount":
{
    "type": "integer",
    "description": "Общее количество",
    "start":
{
    "type": "integer",
    "description": "С какой позиции началось
отдаваться"
    },
    "limit":
{
    "type": "integer",
    "description": "Сколько всего
отдалось"
    }
}
}
}
}
}

```

Пример:

Запрос

```

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

```

1.4.5.1.2 Получение списка с типами контактов группы

Описание : Список всех возможных типов контактов группы

Метод: `GET`

Ресурс: `group/contact_type`

- `start` — с какой позиции начать отдавать элементы, по умолчанию 0
- `limit` — сколько элементов отдавать, по умолчанию 100, не может быть больше 1000

Пример:

Запрос

```

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

```

1.4.5.1.3 Получение статистики по группам

Описание: Статистика по группам

Метод: `GET`

Ресурс: `group/stat`

Параметры:

- `filter` — список фильтрации:
 - `change_date[from]` — от даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[to]`)

- `change_date[to]` — до даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[from]`)

Ответ:

JSON схема отдачи

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Статистика",
  "type": "object",
  "properties": {
    "data": {
      "description": "Не используется",
      "type": "array",
      "items": {}
    },
    "meta": {
      "description": "Статистика",
      "type": "object",
      "properties": {
        "CHANGE_DATE": {
          "description": "Дата изменения",
          "type": "object",
          "properties": {
            "from": {
              "description": "Самая ранняя дата изменения",
              "type": "string"
            },
            "to": {
              "description": "Самая поздняя дата изменения",
              "type": "string"
            }
          },
          "totalCount": {
            "description": "Общее количество элементов",
            "type": "integer"
          }
        }
      }
    }
  }
}
```

Пример:

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

Ответ

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

1.4.5.2 Получение информации по персоне

1.4.5.2.1 Получение изображений множества персон

Метод: POST

Ресурс: `person/images`

Параметры (передаются в виде JSON):

- `response_type` — тип ответа:
 - `chunked` — используется `Transfer-Encoding: chunked`, где каждое изображение отдаётся в своём блоке и в порядке, переданном в теле запроса, если изображения нет, то вместо изображения передаётся null символ "\0"
 - `multipart/mixed` — используется формат `multipart/mixed` [rfc tech-ms](http://rfc.tech-ms), он не входит в стандарт HTTP, но используется в электронных письмах. Поэтому реализовать его не составляет труда. Так же он очень похож на `multipart/form-data`. Каждое изображение отдаётся в своём блоке и помечено отдельно заголовком X-Person-ID, к какой персоне оно принадлежит. Если изображения нет, то оно либо не передаётся, либо тело этого изображения отсутствует

- `person_id` — массив ID персон

Примеры:

`chunked`:

Запрос

```
POST /xapi/person/imagesX-API-Version: 1X-API-CompanyId: iwX-API-ImporterName: domeX-API-Auth-Token: brse99dugplcdbr69axz (пустая строка){ "response_type": "chunked", "person_id": ["DEF1CC1FA015CA428D5DE3CD5B804B69", "E277B624BC988C4C96590A8435F634ED"] }
```

Ответ

```
HTTP/1.1 200 OKContent-Type: image/pngTransfer-Encoding: chunked(пустая строка)1\0 (символ отсутствия изображения для персоны DEF1CC1FA015CA428D5DE3CD5B804B69) 48AC (содержимое изображения для персоны E277B624BC988C4C96590A8435F634ED) 0
```

`multipart/mixed`:

Запрос

```
POST /xapi/person/imagesX-API-Version: 1X-API-CompanyId: iwX-API-ImporterName: domeX-API-Auth-Token: brse99dugplcdbr69axz (пустая строка){ "response_type": "multipart/mixed", "person_id": ["DEF1CC1FA015CA428D5DE3CD5B804B69", "E277B624BC988C4C96590A8435F634ED"] }
```

Ответ

```
HTTP/1.1 200 OKContent-Type: multipart/mixed; boundary=Asrf456BGe4hContent-Length: (суммарный объем, включая дочерние заголовки, может отсутствовать!) (пустая строка) (отсутствующая преамбула) --Asrf456BGe4hContent-Type: application/x-emptyContent-Disposition: attachment; filename="DEF1CC1FA015CA428D5DE3CD5B804B69"X-Person-ID: DEF1CC1FA015CA428D5DE3CD5B804B69(пустая строка и после нулевое содержимое) --Asrf456BGe4hContent-Disposition: attachment; filename="E277B624BC988C4C96590A8435F634ED.jpg"Content-Type: image/jpegX-Person-ID: E277B624BC988C4C96590A8435F634ED(пустая строка) (двоичное содержимое второй персоны E277B624BC988C4C96590A8435F634ED) --Asrf456BGe4h-- (отсутствующий эпилог - пустая строка)
```

1.4.5.2.2 Получение изображения персоны

Метод: `GET`

Ресурс: `person/image/{person_id}`

Ответ: Изображение пользователя

1.4.5.2.3 Получение списка персон

Описание: Список персон

Метод: `GET`

Ресурс: `person`

Параметры:

- `filter` — список фильтрации:
 - `change_date[from]` — от даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[to]`)
 - `change_date[to]` — до даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[from]`)
 - `create_date[from]` — от даты создания в формате UNIX-timestamp (можно использовать отдельно от `create_date[to]`)
 - `create_date[to]` — до даты создания в формате UNIX-timestamp (можно использовать отдельно от `create_date[from]`)
 - `is_deleted` — флаг 1 или 0. Если параметр указан, то будут выдавать сущности с указанным значением
 - `disabled` — флаг 1 или 0, признак активности. Если выставлен, то сущность неактивна (признак берется из LDAP)
 - `contact_type[]` — фильтрация по типу контакта (можно передать несколько, например: `/xapi/person?filter[contact_type][]=webaccount&filter[contact_type][]=icq`)
 - `contact_value[]` — фильтрация по значению контакта (можно передавать несколько, например: `/xapi/person?filter[contact_value][]=user@ad&filter[contact_value][]=login@example.com`)
 - `group_id[]` — фильтрация по группам (можно передать несколько, например: `/xapi/person?filter[group_id][]=webaccount&filter[group_id][]=icq`)
- `with[]` — список дополнительных сущностей, которые должны быть добавлены к объекту:
 - `contacts` — контакты персоны
 - `p2g` — связь персоны с группами
 - `status` — статусы персоны
- `sort` — параметр для сортировки
 - `change_date` — сортировать по дате изменения `asc` или `desc` (например: `/xapi/person?sort[change_date]=asc`)
⚠ может замедлить выдачу
- `start` — с какой позиции начать отдавать элементы, по умолчанию 0
- `limit` — сколько элементов отдавать, по умолчанию 100, не может быть больше 1000

Ответ: Список персон, описанный следующей [JSON-Schema](#):

JSON схема отдачи списка персон

```
{  "$schema": "http://json-schema.org/draft-04/schema#",  "type": "object",  "required": [    "data",    "meta"  ],  "properties": {    "data": {      "description": "Массив"
```

```

персоны",
{
    "type": "array",
    "description": "Персона",
    "items": {
        "type": "object",
        "required": [
            "PERSON_ID",
            "CREATE_DATE",
            "CHANGE_DATE",
            "DISTINGUISHEDNAME",
            "DISPLAY_NAME",
            "GIVENNAME",
            "SN",
            "DISABLED",
            "IS_EMPLOYEE",
            "IS_DELETED"
        ],
        "properties": {
            "PERSON_ID": {
                "description": "Уникальный идентификатор объекта",
                "type": "string"
            },
            "CREATE_DATE": {
                "description": "Дата создания",
                "type": "string"
            },
            "CHANGE_DATE": {
                "description": "Дата последнего изменения",
                "type": "string"
            },
            "DISTINGUISHEDNAME": {
                "description": "Полное имя",
                "type": "string"
            },
            "DISPLAY_NAME": {
                "description": "Имя и фамилия",
                "type": "string"
            },
            "GIVENNAME": {
                "description": "Имя",
                "type": "string"
            },
            "SN": {
                "description": "Фамилия",
                "type": "string"
            },
            "DISABLED": {
                "description": "Состояние персоны",
                "type": "integer"
            },
            "IS_EMPLOYEE": {
                "description": "Является ли сотрудником компании",
                "type": "integer"
            },
            "IS_DELETED": {
                "description": "Удалена ли персона из ТМ",
                "type": "integer"
            },
            "contacts": {
                "description": "Массив контактов",
                "type": "array",
                "items": {
                    "description": "Контакт",
                    "type": "object",
                    "required": [
                        "CONTACT_ID",
                        "CREATE_DATE",
                        "CHANGE_DATE",
                        "CONTACT_TYPE",
                        "VALUE",
                        "IS_PERSONAL",
                        "IS_PRIMARY",
                        "SOURCE"
                    ],
                    "properties": {
                        "CONTACT_ID": {
                            "description": "Уникальный идентификатор объекта",
                            "type": "string"
                        },
                        "CREATE_DATE": {
                            "description": "Дата создания",
                            "type": "string"
                        },
                        "CHANGE_DATE": {
                            "description": "Дата изменения",
                            "type": "string"
                        },
                        "CONTACT_TYPE": {
                            "description": "Тип",
                            "type": "string"
                        },
                        "VALUE": {
                            "description": "Значение",
                            "type": "string"
                        },
                        "IS_PERSONAL": {
                            "description": "Принадлежность (Рабочий, Личный)",
                            "type": "integer"
                        },
                        "IS_PRIMARY": {

```

```

{
    "description": "Признак
ОСНОВНОЙ",
    "type": "integer"
},
    "SOURCE":
{
    "description": "Тип",
    "type": "string"
}
},
    "p2g": {
        "description": "Массив связи с
группами",
        "type": "array",
        "items": {
            "description": "Связь с
группой",
            "type": "object",
            "required":
            [
                "PERSON_ID",
                "PARENT_GROUP_ID",
                "RELATION_TYPE"
            ],
            "properties":
            {
                "PERSON_ID":
                {
                    "description": "Уникальный идентификатор
персоны",
                    "type": "string"
                },
                "PARENT_GROUP_ID":
                {
                    "description": "Уникальный идентификатор
группы",
                    "type": "string"
                },
                "RELATION_TYPE":
                {
                    "description": "Тип связи: 0 - tm, 1 -
member, 2 - distinguishedname, 3 -
primarygroupsid",
                    "type": "integer"
                }
            },
            "status":
            {
                "description": "Массив
статусов",
                "type": "array",
                "items": {
                    "description": "Статус
персоны",
                    "type": "object",
                    "required":
                    [
                        "IDENTITY_STATUS_ID",
                        "CREATE_DATE",
                        "ASSIGNMENT_DATE",
                        "CHANGE_DATE",
                        "DISPLAY_NAME",
                        "COLOR",
                        "EDITABLE"
                    ],
                    "properties": {
                        "IDENTITY_STATUS_ID":
                        {
                            "description": "Уникальный идентификатор
статуса",
                            "type": "string"
                        },
                        "CREATE_DATE":
                        {
                            "description": "Дата создания
статуса",
                            "type": "string"
                        },
                        "CHANGE_DATE":
                        {
                            "description": "Дата изменения
статуса",
                            "type": "string"
                        },
                        "ASSIGNMENT_DATE":
                        {
                            "description": "Дата проставления
статуса
персоне",
                            "type": "string"
                        },
                        "DISPLAY_NAME":
                        {
                            "description": "Название
статуса",
                            "type": "string"
                        },
                        "COLOR":
                        {
                            "description": "Цвет статуса (например
#ff0000)",
                            "type": "string"
                        },
                        "EDITABLE":
                        {
                            "description": "Признак, что статус не
является
системным",
                            "type": "integer"
                        }
                    }
                },
                "meta":
                {
                    "description": "Дополнительные
данные",
                    "type": "object",
                    "required":
                    [
                        "totalCount",
                        "start",
                        "limit"
                    ]
                }
            }
        }
    }
}

```

```

    ],
    "properties": {
      "type": "integer",
      "description": "Общее количество"
    },
    "start": {
      "type": "integer",
      "description": "С какой позиции началось отдаваться"
    },
    "limit": {
      "type": "integer",
      "description": "Сколько всего отдавалось"
    }
  }
}

```

Пример:

Запрос

```

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

```

Ответ

```

{
  "data": [
    {
      "PERSON_ID": "532ECB81F75DDC46BE023478C2E86BD9",
      "WHENCHANGE": "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": "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"
        }
      ]
    }
  ]
}

```

```

": "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" } ] }, { "PERSON_ID": "92FB7D1EC79A9947938BE582D355B109", "WHENCHANGED": "20151229230304.OZ", "WHENCREATED": "20151229230303.OZ", "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": "da39a3ee5e6b4b0d3255bfe95601890afd80709", "contacts": [ { "CONTACT_ID": "4A211BA389707620E0530100007F388100000000", "IDENTITY_ID": "92FB7D1EC79A9947938BE582D355B109", "IDENTITY_TYPE": "person", "IDENTITY_SOURCE": "ad", "CONTACT_TYPE": "E": "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", "DI
SABLED": 0, "IS_EMPLOYEE": 1, "DISTINGUISHEDNAME": "CN=Kasimir
Eggers,OU=Testou,DC=trafmon,DC=iw", "DISPLAY_NAME": "Kasimir
Eggers", "GIVENNAME": "Kasimir", "SN": "Eggers", "OBJECT
SID": "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": "+4
9-7191-
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": fals
e, "HAS_THUMBNAILPHOTO": false, "TM_PHOTO": "da39a3ee5e6b4b0d32
55bfef95601890afd80709", "contacts":
[ {
  "CONTACT_ID": "4A211BA389767620E0530100007F388100000000",
  "IDENTITY_ID": "F0A3298727D7E0498106AE678CC1F5F8", "IDENTITY
TYPE": "person", "IDENTITY_SOURCE": "ad", "CONTACT_TYP
E": "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_PRI
MARY": 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": "4A211BA38977620E0530100007F388100000000",
"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", "DI
SABLED": 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": fals
e, "HAS_THUMBNAILPHOTO": false, "TM_PHOTO": "da39a3ee5e6b4b0d32
55bfef95601890afd80709", "contacts":
[ { "CONTACT_ID": "4A211BA3897C7620E0530100007F388100000000",
"IDENTITY_ID": "52BDB45E49090F4D986899E29FEB35CB", "IDENTITY
_TYPE": "person", "IDENTITY_SOURCE": "ad", "CONTACT_TYP
E": "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",          "DI
SABLED": 0,          "IS_EMPLOYEE": 1,          "DISTINGUISHEDNAME": "CN=Adelgund
Krebs,OU=Testou,DC=trafmon,DC=iw",          "DISPLAY_NAME": "Adelgund
Krebs",          "GIVENNAME": "Adelgund",          "SN": "Krebs",          "OBJECTS
ID": "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": "+4
9-6392-
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_THUMBNAIAPHOTO": fals
e,          "HAS_THUMBNAIAPHOTO": false,          "TM_PHOTO": "da39a3ee5e6b4b0d32
55bfef95601890afd80709",          "contacts":
[{}
  "CONTACT_ID": "4A211BA389827620E0530100007F388100000000",
  "IDENTITY_ID": "BE20920FFD837142AAC09C87A874CCD4",          "IDENTITY
_TYPE": "person",          "IDENTITY_SOURCE": "ad",          "CONTACT_TYP
E": "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_PRI
MARY": 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", "OBJECTS_ID": "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": "da39a3ee5e6b4b0d3255bfe95601890afd80709", "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",          "DI
SABLED": 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": fals
e,          "HAS_THUMBNAILPHOTO": false,          "TM_PHOTO": "da39a3ee5e6b4b0d32
55bfef95601890afd80709",          "contacts":
[ {
    "CONTACT_ID": "4A211BA3898E7620E0530100007F388100000000",
    "IDENTITY_ID": "D4B1F21E90098D4880ABAB5C81F53F7C",          "IDENTITY
_TYPE": "person",          "IDENTITY_SOURCE": "ad",          "CONTACT_TYP
E": "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 PRIMAR
Y": 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",          "DI

```

```

SABLED": 0, "IS_EMPLOYEE": 1, "DISTINGUISHEDNAME": "CN=Annett
Schwanke,OU=Testou,DC=trafmon,DC=iw", "DISPLAY_NAME": "Annett
Schwanke", "GIVENNAME": "Annett", "SN": "Schwanke", "OBJ
ECTSID": "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": fals
e, "HAS_THUMBNAILPHOTO": false, "TM_PHOTO": "da39a3ee5e6b4b0d32
55bfef95601890afd80709", "contacts":
[{"CONTACT_ID": "4A211BA389947620E0530100007F388100000000",
"IDENTITY_ID": "7B238377335B7B48AFDC99851D8F4793", "IDENTITY
_TYPE": "person", "IDENTITY_SOURCE": "ad", "CONTACT_TYP
E": "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_PR
IMARY": 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", "DI
SABLED": 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": fals
e, "HAS_THUMBNAILPHOTO": false, "TM_PHOTO": "da39a3ee5e6b4b0d32
55bfef95601890afd80709", "contacts":
[{
"CONTACT_ID": "4A211BA3899A7620E0530100007F388100000000",
"IDENTITY_ID": "E277B624BC988C4C96590A8435F634ED", "IDENTITY
_TYPE": "person", "IDENTITY_SOURCE": "ad", "CONTACT_TYP
E": "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", "DI
SABLED": 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", "O
BJECTSID": "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_THUMBNAIAPHOTO": fals
e,          "HAS_THUMBNAIAPHOTO": false,          "TM_PHOTO": "da39a3ee5e6b4b0d32
55bfef95601890afd80709",          "contacts":
[
  {
    "CONTACT_ID": "4A211BA389A07620E0530100007F388100000000",
    "IDENTITY_ID": "DEF1CC1FA015CA428D5DE3CD5B804B69",          "IDENTITY
_TYPE": "person",          "IDENTITY_SOURCE": "ad",          "CONTACT_TYP
E": "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.schÄ¶ning@acme.com",          "IS_P
RIMARY": 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          }}

```

1.4.5.2.4 Получение списка статусов персоны

Описание : Список всех возможных статусов персон

Метод: `GET`

Ресурс: `person/status`

Параметры:

- `start` — с какой позиции начать отдавать элементы, по умолчанию 0
- `limit` — сколько элементов отдавать, по умолчанию 100, не может быть больше 1000

Ответ: Список статусов, описанный следующей [JSON-Schema](#):

JSON схема отдачи списка статусов

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "description": "Массив статусов",
      "type": "array",
      "items": {
        "description": "Статус персоны",
        "type": "object",
        "required": [
          "IDENTITY_STATUS_ID",
          "CHANGE_DATE",
          "ASSIGNMENT_DATE",
          "DISPLAY_NAME",
          "COLOR",
          "EDITABLE"
        ],
        "properties": {
          "IDENTITY_STATUS_ID": {
            "description": "Уникальный идентификатор статуса",
            "type": "string"
          },
          "CREATE_DATE": {
            "description": "Дата создания статуса",
            "type": "string"
          },
          "CHANGE_DATE": {
            "description": "Дата изменения статуса",
            "type": "string"
          },
          "ASSIGNMENT_DATE": {
            "description": "Дата проставления статуса персоне",
            "type": "string"
          },
          "DISPLAY_NAME": {
            "description": "Название статуса",
            "type": "string"
          },
          "COLOR": {
            "description": "Цвет статуса (например #ff0000)",
            "type": "string"
          },
          "EDITABLE": {
            "description": "Признак, что статус не является системным",
            "type": "integer"
          }
        },
        "meta": {
          "description": "Дополнительные данные",
          "type": "object",
          "required": [
            "totalCount",
            "start",
            "limit"
          ],
          "properties": {
            "totalCount": {
              "description": "Общее количество",
              "type": "integer"
            },
            "start": {
              "description": "С какой позиции началось отдаваться",
              "type": "integer"
            },
            "limit": {
              "description": "Сколько всего отдалось",
              "type": "integer"
            }
          }
        }
      }
    }
  }
}
```

Пример:

Запрос

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

1.4.5.2.5 Получение списка с типами контактов персоны

Описание : Список всех возможных типов контактов персоны

Метод: GET

Ресурс: `person/contact_type`

Параметры:

- `start` — с какой позиции начать отдавать элементы, по умолчанию 0
- `limit` — сколько элементов отдавать, по умолчанию 100, не может быть больше 1000

Ответ: Список типов, описанный следующей [JSON-Schema](#):

JSON схема отдачи списка контактов

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "description": "Массив типов контактов",
      "type": "array",
      "items": {
        "description": "Тип контакта",
        "type": "object",
        "properties": {
          "CONTACT_TYPE_ID": {
            "description": "Уникальный идентификатор типа контактов",
            "type": "string"
          },
          "DISPLAY_NAME": {
            "description": "Название типа контакта",
            "type": "string"
          },
          "LANGUAGE": {
            "description": "Язык перевода",
            "type": "string"
          },
          "MNEMO": {
            "description": "Мнемонический идентификатор (используется в сущности контактов)",
            "type": "string"
          }
        },
        "meta": {
          "description": "Дополнительные данные",
          "type": "object",
          "required": [
            "totalCount",
            "start",
            "limit"
          ],
          "properties": {
            "totalCount": {
              "description": "Общее количество",
              "type": "integer"
            },
            "start": {
              "description": "С какой позиции началось отдаваться",
              "type": "integer"
            },
            "limit": {
              "description": "Сколько всего отдалось",
              "type": "integer"
            }
          }
        }
      }
    }
  }
}
```

Пример:

Запрос

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

1.4.5.2.6 Получение статистики по персонам

Описание: Статистика по персонам

Метод: `GET`

Ресурс: `person/stat`

Параметры:

- `filter` — список фильтрации:

- `change_date[from]` — от даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[to]`)
- `change_date[to]` — до даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[from]`)

Ответ:

JSON схема отдачи

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Статистика",
  "type": "object",
  "properties": {
    "data": {
      "description": "Не используется",
      "type": "array",
      "items": {}
    },
    "meta": {
      "description": "Статистика",
      "type": "object",
      "properties": {
        "CHANGE_DATE": {
          "description": "Дата изменения",
          "type": "object",
          "properties": {
            "from": {
              "description": "Самая ранняя дата изменения",
              "type": "string"
            },
            "to": {
              "description": "Самая поздняя дата изменения",
              "type": "string"
            }
          }
        },
        "totalCount": {
          "description": "Общее количество элементов",
          "type": "integer"
        }
      }
    }
  }
}
```

Пример:

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

Ответ

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

1.4.5.3 Получение информации по рабочим станциям

1.4.5.3.1 Получение списка рабочих станций

Описание : Список рабочих станций

Метод: `GET`

Ресурс: `workstation`

Параметры:

- `filter` — список фильтрации:
 - `change_date[from]` — от даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[to]`)
 - `change_date[to]` — до даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[from]`)

- `create_date[from]` — от даты создания в формате UNIX-timestamp (можно использовать отдельно от `create_date[to]`)
- `create_date[to]` — до даты создания в формате UNIX-timestamp (можно использовать отдельно от `create_date[from]`)
- `is_deleted` — флаг 1 или 0. Если параметр указан, то будут выдавать сущности с указанным значением
- `disabled` — флаг 1 или 0, признак активности. Если выставлен, то сущность неактивна (признак берется из LDAP)
- `contact_type[]` — фильтрация по типу контакта (можно передать несколько, например: `/xapi/workstation?filter[contact_type][]=webaccount&filter[contact_type][]=icq`)
- `contact_value[]` — фильтрация по значению контакта (можно передавать несколько, например: `/xapi/workstation?filter[contact_value][]=user@ad&filter[contact_value][]=login@example.com`)
- `group_id[]` — фильтрация по группам (можно передать несколько, например: `/xapi/workstation?filter[group_id][]=FCE1409680E5D6469E1F7998266CE9E9&filter[group_id][]=918462B6DC2C954BAAF8706F7DC53F14`)
- `with[]` — список дополнительных сущностей, которые должны быть добавлены к объекту:
 - `contacts` — контакты рабочей станции
 - `w2g` — связь рабочей станции с группами
 - `status` — статусы рабочей станции
- `sort` — параметр для сортировки
 - `change_date` — сортировать по дате изменения `asc` или `desc` (например: `/xapi/workstation?sort[change_date]=asc`) может замедлить выдачу
- `start` — с какой позиции начать отдавать элементы, по умолчанию 0
- `limit` — сколько элементов отдавать, по умолчанию 100, не может быть больше 1000

Ответ : Список рабочих станций, описанный в следующей [JSON-Schema](#) :

JSON схема отдачи

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "required": [
    "data",
    "meta"
  ],
  "properties": {
    "data": {
      "description": "Массив Рабочих станций",
      "type": "array",
      "items": {
        "description": "Рабочая станция",
        "type": "object",
        "required": [
          "WORKSTATION_ID",
          "CREATE_DATE",
          "CHANGE_DATE",
          "DISPLAY_NAME",
          "DISTINGUISHEDNAME",
          "DNSHOSTNAME",
          "DISABLED",
          "TYPE",
          "IS_DELETED"
        ],
        "properties": {
          "WORKSTATION_ID": {
            "description": "Уникальный идентификатор рабочей станции",
            "type": "string"
          },
          "CREATE_DATE": {
            "description": "Дата создания"
          }
        }
      }
    }
  }
}
```

```

записи",
    "CHANGE_DATE": {
        "type": "string",
        "description": "Дата
изменения
записи",
        "type": "string",
        "description": "Название рабочей
станции",
        "type": "string",
        "description": "Отличительное
имя",
        "type": "string",
        "description": "Полное
имя",
        "type": "string",
        "description": "Состояние
рабочей
станции",
        "type": "integer",
        "description": "Тип: computer
- Рабочая станция, mobile - Мобильное устройство, terminal - Терминальный
сервер",
        "type": "string",
        "description": "Удалена
ли рабочая станция из
ТМ",
        "type": "integer",
        "description": "Массив
контактов",
        "type": "array",
        "description": "Контакт",
        "required": [
            "CONTACT_ID",
            "CHANGE_DATE",
            "CONTACT_TYPE",
            "VALUE",
            "IS_PERSONAL",
            "IS_PRIMARY",
            "SOURCE"
        ],
        "properties": {
            "CONTACT_ID": {
                "description": "Уникальный идентификатор
объекта",
                "type": "string",
                "description": "Дата
создания",
                "type": "string",
                "description": "Дата
изменения",
                "type": "string",
                "description": "Тип",
                "type": "string",
                "description": "Значение",
                "type": "string",
                "description": "Принадлежность (Рабочий,
Личный)",
                "type": "integer",
                "description": "Признак
основной",
                "type": "integer",
                "description": "Тип",
                "type": "string",
                "description": "Массив связи с
группами",
                "type": "array",
                "description": "Связь с
группой",
                "type": "object",
                "required": [
                    "WORKSTATION_ID",
                    "PARENT_GROUP_ID",
                    "RELATION_TYPE"
                ]
            }
        }
    }
}

```

```

    ],
    "properties": {
      "WORKSTATION_ID": {
        "description": "Уникальный идентификатор рабочей станции",
        "type": "string"
      },
      "PARENT_GROUP_ID": {
        "description": "Уникальный идентификатор группы",
        "type": "string"
      },
      "RELATION_TYPE": {
        "description": "Тип связи: 0 - tm, 1 - member, 2 - distinguishedname, 3 - primarygroupid",
        "type": "integer"
      }
    },
    "status": {
      "description": "Массив статусов",
      "type": "array",
      "items": {
        "description": "Статус персоны",
        "type": "object",
        "required": [
          "IDENTITY_STATUS_ID",
          "CREATE_DATE",
          "ASSIGNMENT_DATE",
          "CHANGE_DATE",
          "DISPLAY_NAME",
          "COLOR",
          "EDITABLE"
        ],
        "properties": {
          "IDENTITY_STATUS_ID": {
            "description": "Уникальный идентификатор статуса",
            "type": "string"
          },
          "CREATE_DATE": {
            "description": "Дата создания статуса",
            "type": "string"
          },
          "CHANGE_DATE": {
            "description": "Дата изменения статуса",
            "type": "string"
          },
          "ASSIGNMENT_DATE": {
            "description": "Дата проставления статуса персоне",
            "type": "string"
          },
          "DISPLAY_NAME": {
            "description": "Название статуса",
            "type": "string"
          },
          "COLOR": {
            "description": "Цвет статуса (например #ff0000)",
            "type": "string"
          },
          "EDITABLE": {
            "description": "Признак, что статус не является системным",
            "type": "integer"
          }
        }
      }
    },
    "meta": {
      "description": "Дополнительные данные",
      "type": "object",
      "required": [
        "totalCount",
        "start",
        "limit"
      ],
      "properties": {
        "totalCount": {
          "description": "Общее количество",
          "type": "integer",
          "start": 0
        },
        "start": {
          "description": "С какой позиции началось отдаваться",
          "type": "integer",
          "limit": 1000
        },
        "limit": {
          "description": "Сколько всего отдалось",
          "type": "integer"
        }
      }
    }
  }
}

```

Пример:Запрос

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

1.4.5.3.2 Получение списка со статусами рабочих станций

Описание : Список всех возможных статусов рабочих станций

Метод: `GET`

Ресурс: `workstation/status`

- `start` — с какой позиции начать отдавать элементы, по умолчанию 0
- `limit` — сколько элементов отдавать, по умолчанию 100, не может быть больше 1000

Пример:

Запрос

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

1.4.5.3.3 Получение списка с типами контактов рабочих станций

Описание : Список всех возможных типов контактов рабочих станций

Метод: `GET`

Ресурс: `workstation/contact_type`

- `start` — с какой позиции начать отдавать элементы, по умолчанию 0
- `limit` — сколько элементов отдавать, по умолчанию 100, не может быть больше 1000

Пример:

Запрос

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

1.4.5.3.4 Получение статистики по рабочим станциям

Описание: Статистика по рабочим станциям

Метод: `GET`

Ресурс: `workstation /stat`

Параметры:

- `filter` — список фильтрации:
 - `change_date[from]` — от даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[to]`)
 - `change_date[to]` — до даты последнего изменения в формате UNIX-timestamp (можно использовать отдельно от `change_date[from]`)

Ответ:

JSON схема отдачи

```
{  "$schema": "http://json-schema.org/draft-04/schema#",  "description": "Статистика",  "type": "object",  "properties": {    "data": {      "description": "Не используется",      "type": "array",      "items": {}    },    "meta": {      "description": "Статистика",      "type": "object",      "properties": {        "CHANGE_DATE": {}      }    }  }
```

```
{
  "description": "Дата
изменения",
  "type": "object",
  "properties": {
    "from": {
      "description": "Самая ранняя дата
изменения",
      "type": "string"
    },
    "to": {
      "description": "Самая поздняя дата
изменения",
      "type": "string"
    }
  },
  "totalCount": {
    "description": "Общее количество
элементов",
    "type": "integer"
  }
}
```

Пример:

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

Ответ

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

1.4.6 Получение событий

1.4.6.1 Получение извлеченного текста множества событий

Метод: `POST`

Ресурс: `event/texts`

Параметры (передаются в виде JSON):

- `response_type` — тип ответа:
 - `chunked` — используется `Transfer-Encoding: chunked`, где каждый текст отдаётся в своём блоке и в порядке, переданном в теле запроса, если текста нет, то вместо текста передаётся null символ "\0"
 - `multipart/mixed` — используется формат `multipart/mixed` [rfc tech-ms](http://rfc.tech-ms), он не входит в стандарт HTTP, но используется в электронных письмах. Поэтому реализовать его не составляет труда. Также он очень похож на `multipart/form-data`. Каждый текст отдаётся в своём блоке и помечен отдельно заголовком X-Event-ID, к какому событию он принадлежит. Если текста нет, то он либо не передаётся, либо тело этого текста отсутствует
- `event` — массив объектов с полями:
 - `id` — ID события
 - `tbsId` — Табличное пространство, в котором событие

Примеры:

`chunked`:

```
POST /xapi/event/texts
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugplcdbr69axz
(пустая строка)
{"response_type": "chunked", "events": [{"id": 2, "tbsId": 4}, {"id": 4, "tbsId": 4}]}
```

Code Block 24 Запрос

```
HTTP/1.1 200 OK
Content-Type: text/plain
Transfer-Encoding: chunked
(пустая строка)
1
\0 (символ отсутствия текста для события 2)
48AC
(содержимое текста для события 4)
0
```

Code Block 25 Ответ

multipart/mixed:

```
POST /xapi/event/texts
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugplcdbr69axz
(пустая строка)
{"response_type": "multipart/mixed", "events": [{"id": 2, "tbsId": 4}, {"id": 4, "tbsId": 4}]}
```

Code Block 26 Запрос

```
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=Asrf456BGe4h
Content-Length: (суммарный объём, включая дочерние заголовки, может отсутствовать!)
(пустая строка)
(отсутствующая преамбула)
--Asrf456BGe4h
Content-Type: application/x-empty
Content-Disposition: attachment; filename="text-2.txt"
X-Event-ID: 2
(пустая строка и после нулевое содержимое)
--Asrf456BGe4h
Content-Disposition: attachment; filename="text-4.txt"
Content-Type: text/plain
X-Event-ID: 4
(пустая строка)
(содержимое текста для события 4)
--Asrf456BGe4h--
(отсутствующий эпилог - пустая строка)
```

Code Block 27 Ответ

1.4.6.2 Получение извлеченного текста события

Метод: *POST*

Ресурс: *event/texts*

Параметры (передаются в виде JSON):

- `response_type` — тип ответа:
 - `chunked` — используется `Transfer-Encoding: chunked`, где каждый текст отдаётся в своём блоке и в порядке, переданном в теле запроса, если текста нет, то вместо текста передаётся null символ "\0"
 - `multipart/mixed` — используется формат `multipart/mixed` [rfc tech-ms](#), он не входит в стандарт HTTP, но используется в электронных письмах. Поэтому реализовать его не составляет труда. Так же он очень похож на `multipart/form-data`. Каждый текст отдаётся в своём блоке и помечен отдельно заголовком X-Event-ID, к какому событию он принадлежит. Если текста нет, то оно либо не передаётся, либо тело этого текста отсутствует
- `events` — массив объектов с полями:
 - `id` — ID события
 - `tbsId` — Табличное пространство события

Примеры:

`chunked`:

```
POST /xapi/event/texts
X-API-Version: 1.1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugplcdbl69axz
(пустая строка)
{"response_type": "chunked", "events": [{"id": 2, "tbsId": 4}, {"id": 4, "tbsId": 4}]}
```

Code Block 28 Запрос

```
HTTP/1.1 200 OK
Content-Type: text/plain
Transfer-Encoding: chunked
(пустая строка)
1
\0 (символ отсутствия текста для события 2)
48AC
(содержимое текста для события 4)
0
```

Code Block 29 Ответ

`multipart/mixed`:

```
POST /xapi/event/texts
X-API-Version: 1.1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugplcdbl69axz
(пустая строка)
{"response_type": "multipart/mixed", "events": [{"id": 2, "tbsId": 4}, {"id": 4, "tbsId": 4}]}
```

Code Block 30 Запрос

```

HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=Asrf456BGe4h
Content-Length: (суммарный объем, включая дочерние заголовки, может отсутствовать!)
(пустая строка)
(отсутствующая преамбула)
--Asrf456BGe4h
Content-Type: application/x-empty
Content-Disposition: attachment; filename="text-2.txt"
X-Event-ID: 2
(пустая строка и после нулевое содержимое)
--Asrf456BGe4h
Content-Disposition: attachment; filename="text-4.txt"
Content-Type: text/plain
X-Event-ID: 4
(пустая строка)
(содержимое текста для события 4)
--Asrf456BGe4h--
(отсутствующий эпилог - пустая строка)

```

Code Block 31 Ответ

1.4.6.3 Получение метаданных события

Метод: `GET`

Ресурс: `event/{id}`

Параметры:

- `{id}` - `OBJECT_ID` события
- `with[]` - список дополнительных сущностей, которые должны быть добавлены к объекту (например: отправители, получатели и т.д.)

Ответ: Событие

Пример:

```

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

```

Code Block 32 Получить событие с OBJECT_ID "1" с отправителями, получателями:

1.4.6.4 Получение множества контентов событий

Метод: `POST`

Ресурс: `event/raws`

Параметры (передаются в виде JSON):

- `response_type` – тип ответа:
 - `chunked` – используется `Transfer-Encoding: chunked`, где каждый контент отдается в своем блоке и в порядке, переданном в теле запроса, если контента нет, то вместо контента передается null-символ `"\0"`
 - `multipart/mixed` – используется формат `multipart/mixed` [rfc tech-ms](#), он не входит в стандарт HTTP, но используется в электронных письмах. Поэтому реализовать его не составляет труда. Также он очень похож на `multipart/form-data`. Каждый контент отдается в своем блоке и помечен отдельно заголовками `X-Event-ID`, `X-Content-ID` и `X-Content-Deleted`, к какому событию и контенту он принадлежит. Если контента нет, то оно либо не передается, либо тело этого текста отсутствует

- `event` – массив объектов с полями:
 - `id` – ID события
 - `tbsId` – Табличное пространство, в котором событие
 - `contentId` – ID контента. Если не указан, то отдается главный узел

Примеры:

`chunked`:

```
POST /xapi/event/raws
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugplcdbr69axz
(пустая строка)
{"response_type": "chunked", "events": [{"id": 2, "tbsId": 4}, {"id": 3, "tbsId": 4}, {"id": 4, "tbsId": 4, "contentId": 8}]}
```

Code Block 33 Запрос

```
HTTP/1.1 200 OK
Content-Type: text/plain
Transfer-Encoding: chunked
(пустая строка)
1
\0 (символ отсутствия контента для события 2)
3E4
(содержимое корневого контента для события 3)
48AC
(содержимое контента 8 для события 4)
0
```

Code Block 34 Ответ

`multipart/mixed`:

```
POST /xapi/event/raws
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: dome
X-API-Auth-Token: brse99dugplcdbr69axz
(пустая строка)
{"response_type": "multipart/mixed", "events": [{"id": 2, "tbsId": 4}, {"id": 3, "tbsId": 4}, {"id": 4, "tbsId": 4, "contentId": 8}]}
```

Code Block 35 Запрос

```

HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=Asrf456BGe4h
Content-Length: (суммарный объем, включая дочерние заголовки, может
отсутствовать!)
(пустая строка)
(отсутствующая преамбула)
--Asrf456BGe4h
Content-Type: application/x-empty
Content-Disposition: attachment; filename="content-2"
X-Event-ID: 2
(пустая строка и после нулевое содержимое)
--Asrf456BGe4h
Content-Disposition: attachment; filename="content-3-5.eml"
Content-Type: email
X-Event-ID: 3
X-Content-ID: 5
(пустая строка)
(содержимое корневого контента для события 3)
--Asrf456BGe4h
Content-Disposition: attachment; filename="my-project.jpg"
Content-Type: image/jpeg
X-Event-ID: 4
X-Content-ID: 8
(пустая строка)
(содержимое контента 8 для события 4)
--Asrf456BGe4h--

(отсутствующий эпилог - пустая строка)

```

Code Block 36 Ответ

1.4.6.5 Получение информации о контенте события

Метод: *POST*

Ресурс: *event/contents*

Параметры (передаются в виде JSON):

- *response_type* – тип ответа:
 - *chunked* – используется *Transfer-Encoding: chunked*, где каждый контент отдаётся в своем блоке и в порядке, переданном в теле запроса, если контента нет, то вместо контента передаётся null-символ `"\0"`;
 - *multipart/mixed* – используется формат *multipart/mixed* [rfc tech-ms](#), он не входит в стандарт HTTP, но используется в электронных письмах. Поэтому реализовать его не составляет труда. Также он очень похож на *multipart/form-data*. Каждый контент отдается в своем блоке и помечен отдельно заголовками *X-Event-ID* и *X-Content-ID*, к какому событию и контенту он принадлежит. Если контента нет, то оно либо не передается, либо тело этого текста отсутствует.
- *events* – массив объектов с полями:
 - *id* – ID одного или нескольких событий, по которым нужно получить контенты;
 - *tbsId* – ID табличных пространств для каждого события;
- *filter[mime]* – типы контентов с возможностью указать несколько значений (по точному совпадению или по маске) или сразу все;
- *start* – с какой позиции начать отдавать контенты, по умолчанию 0;
- *limit* – сколько контентов отдавать. По умолчанию 100, не может быть больше 1000 в одном запросе.

Пример:

```
curl -X POST
"https://server.company.ru/xapi/event/contents?filter[mime][]=image%2Fjpeg&filter[mime][]=text%2FA&start=1&limit=10" -H "accept: application/json" -H "Content-Type: application/json" -H "X-Enable-OpenAPI: 1" -d
"{\"events\": [{\"id\": 2, \"tbsId\": 4}]}"
```

Code Block 37 Запрос

По запросу будут предоставлены ID контентов, соответствующих условиям запроса, с указанием следующих атрибутов:

```
{
  "data": [
    {
      "OBJECT_CONTENT_ID": 15,
      "MIME": "string",
      "OBJECT_ID": 2,
      "TBS_ID": 4,
      "CONTENT_SIZE": 1500,
      "is_filename": 0           // признак, что контент является или не является
именем файла
    }
  ],
  "meta": {
    "totalCount": 1,
    "start": 1,
    "limit": 10
  }
}
```

Code Block 38 Ответ

1.4.6.6 Получение перехваченного файла события

Метод: `GET`

Ресурс: `event/{id}/raw?download=0`

Параметры:

- `{id}` - OBJECT_ID события, обязательный
- `download` - добавляет HTTP-заголовки для скачивания файла (*Content-Disposition*, *Content-Transfer-Encoding* и прочие), обязательный, значения 1 или 0 (по умолчанию 0)

Поддерживается HTTP-заголовок `Range`

Ответ: Файл перехваченного события. Например: eml (для писем) или сам файл для событий копирования файла.

1.4.6.7 Получение списка ОЗ

Описание : Список ОЗ, отсортированный по имени.

Метод: `GET`

Ресурс: `event/protectedDocument`

Параметры:

- `start` - с какой позиции начать отдавать ОЗ, по умолчанию 0
- `limit` - сколько ОЗ отдавать, по умолчанию 100, не может быть больше 1000
- `with[]` — список дополнительных сущностей, которые должны быть добавлены к ОЗ:
 - `pd2pc` — связь объекта защиты к каталогу объекта защиты

Ответ: Список ОЗ, описанный следующей **JSON схемой отдачи списка ОЗ:**

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "array",
      "description": "Массив 03",
      "items": {
        "type": "object",
        "description": "03",
        "properties": {
          "DOCUMENT_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]
          },
          "pd2pc": {
            "description": "Массив связи с каталогами 03",
            "type": "array",
            "items": {
              "description": "Связь с каталогом 03",
              "type": "object",
              "required": [
                "CATALOG_ID",
                "DOCUMENT_ID"
              ],
              "properties": {
                "CATALOG_ID": {
                  "description": "Уникальный идентификатор каталога 03",
                  "type": "string"
                },
                "DOCUMENT_ID": {
                  "description": "Уникальный идентификатор 03",
                  "type": "string"
                }
              },
              "required": [
                "DOCUMENT_ID",
                "DISPLAY_NAME",
                "CREATE_DATE",
                "IS_DELETED"
              ],
              "meta": {
                "description": "Дополнительные данные",
                "type": "object",
                "properties": {
                  "totalCount": {
                    "type": "integer",
                    "description": "Общее количество 03",
                    "start": {
                      "type": "integer",
                      "description": "С какой позиции началось отдаваться"
                    },
                    "limit": {
                      "type": "integer",
                      "description": "Сколько всего отдалось"
                    },
                    "required": [
                      "totalCount",
                      "start",
                      "limit"
                    ]
                  }
                },
                "required": [
                  "data",
                  "meta"
                ]
              }
            }
          }
        }
      }
    }
  }
}

```

Пример:

Запрос

```

GET /xapi/event/protectedDocument?start=0&limit=10X-API-Version: 1X-API-CompanyId: iwX-API-ImporterName: domeX-API-Auth-Token: brse99dugplcdbr69axz

```

Ответ

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

```

1.4.6.8 Получение списка заголовков

Описание: Получение списка дополнительных полей события.

Метод: `GET`

Ресурс: `event/header`

Параметры:

- `start` - с какой позиции начать отдавать заголовки, по умолчанию 0
- `limit` - сколько заголовков отдавать, по умолчанию 100, не может быть больше 1000

Ответ: Список дополнительных полей, описанных следующей **JSON схемой отдачи:**

```
{
  "$id": "http://infowatch.com/iwtm/xapi/event/header.json",
  "type": "object",
  "$schema": "http://json-schema.org/draft-07/schema#",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "NAME": {
            "type": "string",
            "title": "Код атрибута, который будет использоваться в заголовках события и для идентификации в xAPI/PushAPI",
            "examples": [
              "mac_level"
            ]
          },
          "NOTE": {
            "type": "string",
            "title": "Отображаемое в консоли ТМ название атрибута",
            "examples": [
              "Уровень мандатного доступа"
            ]
          },
          "TYPE": {
            "type": "string",

```

```
        "title": "Одно из следующих значений: число (целое,
дробное, строка), дата и время в UTC + указание смещение часового пояса,
длительность, гиперссылка",
        "enum": [
            "integer",
            "float",
            "string",
            "date",
            "UTC",
            "duration",
            "link"
        ]
    },
    "IS_DELETED": {
        "type": "integer",
        "title": "Является ли заголовок удаленным из системы
(у старых объектов он остаётся)",
        "default": 0,
        "enum": [
            0,
            1
        ]
    },
    "FORMAT": {
        "type": "string",
        "title": "Тип значения (с версии 1.2)",
        "enum": [
            "string",
            "date",
            "number"
        ]
    },
    "IS_SYSTEM": {
        "type": "integer",
        "title": "Является ли заголовок системным (с версии
1.1)",
        "default": 1,
        "enum": [
            0,
            1
        ]
    },
    "IS_MULTIPLE_VALUE": {
```

```

        "type": "integer",
        "title": "Может ли заголовок содержать несколько значений (с версии 1.1)",
        "default": 0,
        "enum": [
            0,
            1
        ]
    },
    "locale": {
        "type": "array",
        "title": "Локализация для заголовков (с версии 1.2)",
        "items": {
            "type": "object",
            "title": "Локализация для заголовка",
            "properties": {
                "NAME": {
                    "type": "string",
                    "title": "Код атрибута, который будет использоваться в заголовках события и для идентификации в xAPI/PushAPI"
                },
                "LANGUAGE": {
                    "type": "string",
                    "title": "Название языка",
                    "enum": [
                        "rus",
                        "eng"
                    ]
                },
                "DISPLAY_NAME": {
                    "type": "string",
                    "title": "Отображаемое в консоли ТМ название атрибута"
                }
            }
        }
    },
    "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"
    ]
}

```

Пример:

Запрос

```

GET /xapi/event/headerX-API-Version: 1.1X-API-CompanyId: iwX-API-
ImporterName: domeX-API-Auth-Token: brse99dugplcdbr69axz

```

Ответ

```

{
    "data": [
        {
            "NAME": "device_type",
            "NOTE": "Тип устройства",

```

```
        "TYPE": "enum",
        "IS_DELETED": 0,
        "IS_SYSTEM": 1,
        "IS_MULTIPLE_VALUE": 0
    },
    {
        "NAME": "destination_type",
        "NOTE": "Тип источника сканирования",
        "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": "Имя сервера печати",
        "TYPE": "string",
        "IS_DELETED": 0,
        "IS_SYSTEM": 1,
        "IS_MULTIPLE_VALUE": 0
    },
    {
        "NAME": "location",
        "NOTE": "Расположение принтера",
        "TYPE": "string",
        "IS_DELETED": 0,
        "IS_SYSTEM": 1,
        "IS_MULTIPLE_VALUE": 0
    },
    {
        "NAME": "comment",
        "NOTE": "Комментарий к принтеру",
        "TYPE": "string",
```

```
"IS_DELETED": 0,
"IS_SYSTEM": 1,
"IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "port_name",
  "NOTE": "Имя порта принтера",
  "TYPE": "string",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
  "IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "copies",
  "NOTE": "Количество копий",
  "TYPE": "number",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
  "IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "task_name",
  "NOTE": "Имя задания",
  "TYPE": "string",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
  "IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "print_task_name",
  "NOTE": "Имя задания на печать",
  "TYPE": "string",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
  "IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "task_run_date",
  "NOTE": "Дата запуска задания",
  "TYPE": "date",
  "IS_DELETED": 0,
```

```
"IS_SYSTEM": 1,
"IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "create_date",
  "NOTE": "Дата создания файла",
  "TYPE": "date",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
  "IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "modify_date",
  "NOTE": "Дата модификации файла",
  "TYPE": "date",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
  "IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "source_ip",
  "NOTE": "IP адрес источника",
  "TYPE": "string",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
  "IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "workstation_type",
  "NOTE": "Тип рабочей станции",
  "TYPE": "enum",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
  "IS_MULTIPLE_VALUE": 0
},
{
  "NAME": "application_path_from",
  "NOTE": "Путь к приложению из которого скопировано событие",
  "TYPE": "string",
  "IS_DELETED": 0,
  "IS_SYSTEM": 1,
```

```
        "IS_MULTIPLE_VALUE": 0
    },
    {
        "NAME": "application_path_to",
        "NOTE": "Путь к приложению в которое скопировано событие",
        "TYPE": "string",
        "IS_DELETED": 0,
        "IS_SYSTEM": 1,
        "IS_MULTIPLE_VALUE": 0
    },
    {
        "NAME": "terminal_session",
        "NOTE": "Признак терминальной сессии",
        "TYPE": "number",
        "IS_DELETED": 0,
        "IS_SYSTEM": 1,
        "IS_MULTIPLE_VALUE": 0
    },
    {
        "NAME": "emitter",
        "NOTE": "Источник события",
        "TYPE": "enum",
        "IS_DELETED": 0,
        "IS_SYSTEM": 1,
        "IS_MULTIPLE_VALUE": 0
    },
    {
        "NAME": "charset",
        "NOTE": "Кодировка текста вложения",
        "TYPE": "string",
        "IS_DELETED": 0,
        "IS_SYSTEM": 1,
        "IS_MULTIPLE_VALUE": 0
    }
],
"meta": {
    "totalCount": 20,
    "start": 0,
    "limit": 100
}
}
```

1.4.6.9 Получение списка каталогов ОЗ

Описание : Список каталогов ОЗ, отсортированных по имени.

Метод: `GET`

Ресурс: `event/protectedCatalog`

Параметры:

- `start` - с какой позиции начать отдавать каталоги (по умолчанию 0)
- `limit` - сколько каталогов отдавать (по умолчанию 100, не может быть больше 1000)

Ответ: Список каталогов ОЗ, описанный следующей JSON-Schema:

```

{
  "$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"
  ]
}

```

Code Block 39 JSON схема отдачи списка каталогов.

Пример:

```
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
```

Code Block 40 Запрос

```
{
  "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
  }
}
```

Code Block 41 Ответ

1.4.6.10 Получение списка политик

Описание: Список политик отсортированный по имени.

Метод: `GET`

Ресурс: `event/policy`

Параметры:

- `start` — с какой позиции начать отдавать политики, по умолчанию 0
- `limit` — сколько политик отдавать, по умолчанию 100, не может быть больше 1000

Ответ: Список политик, описанный следующей **JSON схемой отдачи списка каталогов:**

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "type": "array",
      "description": "Массив политик",
      "items": {
        "type": "object",
        "description": "Политика",
        "properties": {
          "POLICY_ID": {
            "type": "string",
            "description": "Идентификатор"
          },
          "DISPLAY_NAME": {
            "type": "string",
            "description": "Имя"
          },
          "CREATE_DATE": {
            "type": "string",
            "description": "Дата создания"
          },
          "IS_DELETED": {
            "type": "integer",
            "description": "Удалена ли из активной конфигурации",
            "enum": [0, 1]
          },
          "required": [
            "POLICY_ID",
            "DISPLAY_NAME",
            "CREATE_DATE",
            "IS_DELETED"
          ],
          "meta": {
            "description": "Дополнительные данные",
            "type": "object",
            "properties": {

```

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

Пример:

Запрос

```
GET /xapi/event/policy?start=0&limit=10X-API-Version: 1X-API-CompanyId: iwX-API-ImporterName: domeX-API-Auth-Token: brse99dugplcdbr69axz
```

Ответ

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

1.4.6.11 Получение списка событий

Описание: Список событий, отсортированных по времени перехвата (от новых к старым).

Метод: `GET`

Ресурс: `event`

Параметры:

- `filter` - список фильтрации:
 - `date[from]` — дата начала в формате UNIX-timestamp (можно использовать отдельно от `date[to]`)
 - `date[to]` — дата окончания в формате UNIX-timestamp (можно использовать отдельно от `date[from]`)
 - `insert_date[>]` — дата вставки начала в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][>]=1234.123312`
 - `insert_date[<]` — дата вставки окончания в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][<]=1234.123312`
 - `insert_date[>=]` — дата вставки начала в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][>]=1234.123312` (включительно) (с версии API 1.1)
 - `insert_date[<=]` — дата вставки окончания в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][<=]=1234.123312` (включительно) (с версии API 1.1)
 - `rule_group_type[]` — группа правил (с версии API 1.1)
 - `violation_level[]` — уровень нарушения (с версии API 1.1)
 - `verdict[]` — вердикт (с версии API 1.1)

- `object_type_code[]` — тип события (возможно несколько значений)
 - `protocol[]` — протокол (возможно несколько значений)
 - `user_decision[]` — решение пользователя (возможно несколько значений)
 - `document_id[]` — идентификатор документа идентификатор
 - `object_size[from]` — от сколько байт искать событие (можно использовать отдельно от `object_size[to]`)
 - `object_size[to]` — до сколько байт искать событие (можно использовать отдельно от `object_size[from]`)
 - `protected_catalog_id[]` — идентификатор каталога ОЗ (возможно несколько значений)
 - `tag_id[]` — идентификатор тега (возможно несколько значений)
- `with[]` - список дополнительных сущностей, которые должны быть добавлены к объекту (например: отправители, получатели и т.д.):
 - `protected_catalogs`
 - `protected_documents`
 - `policies`
 - `senders_keys`
 - `senders`
 - `recipients_keys`
 - `recipients`
 - `workstations_keys`
 - `workstations`
 - `tags`
 - `perimeters` (с версии API 1.1)
 - `attachments` (с версии API 1.1)
 - `files` (с версии API 1.1) — недокументированный параметр, отдающий все контенты, в которых непустое поле FILE_NAME
 - `contents_renderer` (с версии API 1.8) - прикрепить контенты рендерера различного типа
- `without[]` — дополнительные флаги
 - `count` — не считать `totalCount` (расчет `totalCount` может занимать больше половины времени запроса, поэтому лучше воспользоваться методом `event/stat`)
- `headers[]` — список дополнительных полей (заголовков) события (можно написать *, если нужны все). Список можно получить с помощью метода `event/header` (с версии API 1.1)
- `sort[]` — недокументированное поле, позволяющее сортировать результат (получение событий замедляется существенно, для получения последнего/первого события лучше использовать метод `event/stat`. Возможные ключи:
 - `capture_date` — `asc` (с версии API 1.1) /`desc`, например: `?sort[capture_date]=desc`
 - `insert_date` — `asc`/`desc`, например: `?sort[insert_date]=asc` (с версии API 1.1)
- `start` - с какой позиции начать отдавать события (по умолчанию 0)
- `limit` - сколько событий отдавать (по умолчанию 100, не может быть больше 1000)

Алгоритм при первом получении событий:

1. Определить, с какого времени нужно получать события. Например, получить из статистики событий (GET /xapi/event/stat) интервал INSERT_DATE.
2. Разделить интервал на меньшее из значений: количество потоков (но не более 32) **или** количество ядер на сервере.
3. В каждом потоке составить и запустить первый запрос на получение списка события вида:
GET
/xapi/event?without[]=count&filter[insert_date][>]={минимум_интервала}&filter[insert_date][<=]={максимум_интервала}&sort[insert_date]=asc&limit=1000
4. В полученном списке запомнить все OBJECT_ID.
5. В полученном списке найти последнюю INSERT_DATE и использовать её в запросе из пункта 3, в качестве *минимум_интервала*.
6. В полученном новом списке исключить события, которые запомнили в предыдущей итерации.
7. Если не дошли до *максимум_интервала*, снова начать с пункта 4.
8. Сохранить в персистентное хранилище:
 - a. максимальное значение INSERT_DATE, полученное из пункта 1.
 - b. список OBJECT_ID из интервала, в который входит максимальное значение INSERT_DATE.

Примеры:

```
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: gla20doslpcdbr69caxz
```

Code Block 42 Получить первые 100 событий с за январь 2015 года:

```
GET /xapi/event?with=contents_renderer&limit=1000
X-API-Version: 1.8
X-API-CompanyId: IW
X-API-productName: Vision
X-API-Auth-Token: 1eja07v23vs9kls5ij5k
```

Code Block 43 Получить 1000 событий с письмами, содержащими контент:

Ответ содержит список событий, отсортированный по времени перехвата от новых к старым. Для разных типов событий структура ответа будет отличаться.

```
[
  OBJECT_ID: 1
  VIOLATION_LEVEL: 'High'
  CAPTURE_DATE: '12-03-2022 15:02:25'
  contents_renderer:
  [
    SUBJECT: [
      CONTENT_ID: 1
      OBJECT_ID: 1
      CONTENT_SIZE: 26
    ]
    BODY: [
      CONTENT_ID: 2
      OBJECT_ID: 1
      CONTENT_SIZE: 5125
    ]
  ]
]
```

Code Block 44 Пример ответа

1.4.6.12 Получение списка тегов

Описание: Список тегов, отсортированных по имени.

Метод: `GET`

Ресурс: `event/tag`

Параметры:

- `start` - с какой позиции начать отдавать теги (по умолчанию 0)
- `limit` - сколько тегов отдавать (по умолчанию 100, не может быть больше 1000)

Ответ: Список тегов, описанный следующей JSON-Schema:

```

{
  "$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"
    ]
}

```

Code Block 45 JSON схема отдачи списка тэгов

Пример:

```

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

```

Code Block 46 Запрос

```

{
  "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
  }
}

```

Code Block 47 Ответ

1.4.6.13 Получение статистики по событиям

Описание: Недокументированный метод получения статистики по событиям, отдает только статистику по дате перехвата.

Метод: *GET*

Ресурс: *event/stat*

Параметры:

- *fields* — список полей, которые нужно выбрать: *CAPTURE_DATE*, *INSERT_DATE*, *totalCount* (с версии API 1.2)
- *filter* — список фильтрации:
 - *date[from]* — дата начала в формате UNIX-timestamp (можно использовать отдельно от *date[to]*)
 - *date[to]* — дата окончания в формате UNIX-timestamp (можно использовать отдельно от *date[from]*)
 - *insert_date[>]* — дата вставки начала в формате UNIX-timestamp и через точку microtime, например *filter[insert_date][>]=1234.123312* (с версии API 1.1)
 - *insert_date[<]* — дата вставки окончания в формате UNIX-timestamp и через точку microtime, например *filter[insert_date][<]=1234.123312* (с версии API 1.1)

- `insert_date[>=]` — дата вставки начала в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][>]=1234.123312`(включительно) (с версии API 1.1)
- `insert_date[<=]` — дата вставки окончания в формате UNIX-timestamp и через точку microtime, например `filter[insert_date][<]=1234.123312`(включительно) (с версии API 1.1)
- `object_type_code[]` — тип события (возможно несколько значений) (с версии API 1.1)
- `violation_level[]` — уровень нарушения (с версии API 1.1)

Ответ: Статистика по событиям, описанная следующей [JSON-Schema](#):

JSON схема отдачи:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "data": {
      "description": "Пустая секция",
      "items": {
        "type": "array"
      },
      "meta": {
        "description": "Дополнительные данные",
        "type": "object",
        "properties": {
          "CAPTURE_DATE": {
            "description": "Статистика по полю CAPTURE_DATE",
            "type": "object",
            "properties": {
              "from": {
                "type": "string",
                "description": "Дата перехвата первого события в выборке"
              },
              "to": {
                "type": "string",
                "description": "Дата перехвата последнего события в выборке"
              }
            }
          },
          "INSERT_DATE": {
            "description": "Статистика по полю INSERT_DATE (с версии API 1.1)",
            "type": "object",
            "properties": {
              "from": {
                "type": "string",
                "description": "Дата вставки первого события в выборке"
              },
              "to": {
                "type": "string",
                "description": "Дата вставки последнего события в выборке"
              }
            }
          },
          "totalCount": {
            "type": "integer",
            "description": "Общее количество событий"
          }
        }
      }
    }
  }
}
```

Пример:

Запрос

```
GET /xapi/event/stat?filter[data][from]=0X-API-Version: 1.3X-API-CompanyId: iwX-API-ImporterName: domeX-API-Auth-Token: brse99dugplcdbr69axz
```

Ответ

```
{
  "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      }}
```

1.4.6.14 Получение вердикта

Версия API: 1.8 и выше

Метод: *GET*

Ресурс: */event/verdict*

Описание:

Метод возвращает вердикт события по ID документа:

- *Разрешено (Allowed)*;
- *Помещено на карантин (Quarantined)*;
- *Заблокировано (Forbidden)*.

Кроме того, вывод метода содержит поля:

- *OBJECT_ID, integer* – уникальный идентификатор объекта;
- *TBS_ID, number* – идентификатор табличного пространства для секционирования.

Параметры:

- *documentId* – идентификатор перехваченного документа;
- *date* – дата перехвата в формате *ГГГГ-ММ-ДД*.

Пример:

```
curl -X 'GET' \
  'https://server.company.ru/xapi/event/verdict?documentId=b042cae7-cd7f-4ff3-
9af7-5ed41ec0dc23&date=2024-01-15' \
  -H 'X-API-Version: 1.8'
```

Code Block 48 Запрос

```
{
  "data": [
    {
      "VERDICT": "Allowed",
      "DOCUMENT_ID": "string",
      "OBJECT_ID": 0,
      "TBS_ID": 0
    }
  ]
}
```

Code Block 49 Ответ

Для быстрого получения вердиктов по событиям рекомендуется делать запрос сразу по нескольким *documentID*:

```
https://server.infowatch.ru/xapi/event/verdict?documentId=ID1,ID2ID3,ID4,ID5,ID6,I
Dnnn&date=DATE2'
```

Code Block 50 Запрос

1.4.6.14.1 Отслеживание вердикта по документу

Если требуется проверить набор файлов из внешней системы, и при условии срабатывания политик и на основании решения пользователя по событию, предпринять какие-либо действия, то необходимо:

1. Отправить файлы для проверки в ТМ.
2. Получить и сохранить информацию о событии ТМ.
3. Отслеживать статус события.

1.4.6.14.1.1 Отправка файлов для проверки в ТМ

В системе сохраняется информация о ходе проверки файлов. Она имеет следующую структуру:

- `file_path` – путь или полное имя файла, отправляемого на анализ;
- `document_id` – GUID события, сгенерированный при отправке файла на анализ;
- `object_id` – идентификатор события, созданного ТМ в результате анализа;
- `tbs_id` – идентификатор табличного пространства;
- `verdict` – вердикт, предоставленный в результате анализа;
- `user_decision` – решение пользователя после помещения события на карантин

Отправка файла осуществляется путем выполнения запроса PushAPI:

```
pushapi-util --host 127.0.0.1 --token 8iwivt0cmirlcceuul21 --id IW --class
fileexch --service im_vk --evtattr event_name:"vkontakte Event" --sender
"vkontakte:sender1111:'{"display_name\":\"Врбт
Тестер\"','url\":\"https://www.facebook.com/100006160150433\"','login\":\"new_logi
n\"}'}" --receiver vkontakte:vkontakteressivier1 --receiver
"vkontakte:11111123:'{"url\":\"https://vk.com/vrbttester\"','login\":\"some_login
\"}'}" --data file:file.txt,filename:file.txt
```

Code Block 51 Пример запроса для отправки файла на анализ в ТМ

```
Object database GUID: df5f046e-0555-46e1-be48-8da33f41b9bb
```

Code Block 52 Пример ответа с получением `document_id` события

Данную операцию необходимо повторить для каждого файла и сохранить все полученные GUID. Они будут использоваться далее для отслеживания состояния анализа и изменения решения пользователя.

1.4.6.14.1.2 Получение событий из ТМ

Это действие необходимо для того, чтобы получить информацию о сработавших политиках и вынесенном вердикте по событию в ТМ по результатам анализа содержимого файлов. Помимо этого на данном этапе требуется получить идентификаторы событий OBJECT_ID и идентификаторы табличных пространств TBS_ID на основе их GUID (`document_id`).

Для этого необходимо выполнить запрос Public API вида:

```
https://server.infowatch.ru/xapi/event?filter[document_id]={GUID_1},{GUID_2},...,{
GUID_N}&filter[date][from]={START_TIME}&filter[date][to]={END_TIME}
```

Code Block 53 Пример запроса

Где `{GUID_i}` - GUID(`document_id`) файла, полученный при отправке на анализ.

Результатом данного запроса будет массив, элементами которого являются события ТМ. Сопоставляя `document_id`, сохраненные в результате отправки файлов на анализ, с `document_id` события, полученного на этом шаге, сохраняем OBJECT_ID и TBS_ID, а также параметры, необходимые для принятия дальнейшего решения по проверяемым файлам (например, вердикт).

На обработку событий в ТМ может потребоваться некоторое время, поэтому возможно, что в результат выполнения API-запроса попадет информация не обо всех событиях, `document_id` которых был передан. Следовательно, для тех `document_id`, события по которым не были

получены, необходимо повторное выполнение запроса к public API до тех пор, пока они все не будут обработаны.

1.4.6.14.1.3 Отслеживание статуса события

Для отслеживания изменения пользовательского решения рекомендуется периодически выполнять запросы на получение аудита по событиям, где было изменено решение пользователя. Запрос может иметь следующий вид:

```
https://server.infowatch.ru/xapi/audit?filter[change_date][>]={START_TIME}&filter[entity_type][]=Object
```

Code Block 54 Пример запроса

Где **{START_TIME}** - unix timestamp момента, после последнего запроса на получение аудита (для первой итерации – момента после отправки первого файла на анализ).

После выполнения запроса все события аудита перебираются по очереди, в них поле ENTITY_ID сопоставляется с сохраненными ранее OBJECT_ID. Решение можно принимать на основе полей аудита OPERATION и PROPERTY_CHANGES.

1.4.6.15 Изменение атрибутов событий

1.4.6.15.1 Назначение вердикта пользователя

Версия API: ≥ 1.2

Метод: PATCH

Ресурс: event/{id}/userDecision?tbsId={tbsId}

Описание:

Метод назначает решение пользователя. Поля:

- **{id}** — ID события (**OBJECT_ID**)
- **{tbsId}** — Табличное пространство события (**TBS_ID**)
- **USER_DECISION** — Решение пользователя. Возможные значения:
 - NotProcessed - Решение не принято
 - Violation - Нарушение
 - NoViolation - Нет нарушения
 - AdditionalProcessingNeeded - Требуется дополнительной обработки
- **NOTE** — комментарий, который сохраняется в разделе Аудит. Может быть заполнено, например, именем пользователя сторонней системы, который изменил значение атрибута. Длина поля не должна превышать 1000 символов.

Вызов данного API записывается в раздел Аудит. При назначении решения пользователя в событии могут автоматически поменяться следующие поля:

- **TBS_ID**
- **VERDICT**
- **FORWARD_STATE_CODE**

Поэтому в ответе API отдаются значения, которые можно будет обновить у вашей копии события.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "title": "Set user decision schema",
  "required": [
    "USER_DECISION"
  ],
  "properties": {
    "USER_DECISION": {
      "$id": "#/properties/USER_DECISION",
      "type": "string",
      "title": "User decision",
      "enum": [
        "NotProcessed",
        "Violation",
        "NoViolation",
        "AdditionalProcessingNeeded"
      ]
    },
    "NOTE": {
      "$id": "#/properties/NOTE",
      "type": "string",
      "title": "Note",
      "maxLength": 1000
    }
  }
}
```

Code Block 55 Тело запроса:JSON схема назначения вердикта пользователя

```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "title": "Result setting user decision schema",
  "required": [
    "OBJECT_ID",
    "TBS_ID",
    "VERDICT",
    "USER_DECISION",
    "FORWARD_STATE_CODE"
  ],
  "properties": {
    "OBJECT_ID": {
      "$id": "#/properties/OBJECT_ID",
      "type": "integer",
      "title": "ID События"
    },
    "TBS_ID": {
      "$id": "#/properties/TBS_ID",
      "type": "integer",
      "title": "ID табличного пространства"
    },
    "VERDICT": {
      "$id": "#/properties/VERDICT",
      "type": "string",
      "title": "Вердикт",
      "enum": [
        "Allowed",
        "Quarantined",
        "Forbidden"
      ]
    },
    "USER_DECISION": {
      "$id": "#/properties/USER_DECISION",
      "type": "string",
      "title": "Проставленное решение пользователя",
      "enum": [
        "NotProcessed",
        "NoViolation",
        "Violation",
        "AdditionalProcessingNeeded"
      ]
    },
    "FORWARD_STATE_CODE": {
      "$id": "#/properties/FORWARD_STATE_CODE",
      "type": "string",
      "title": "Статус отправки",
      "examples": [
        "Sent",
        "NotSent",
        "Pending"
      ]
    }
  }
}

```

Code Block 56 Тело ответа: JSON схема ответа на назначение вердикта пользователя

Пример: Назначение вердикта "Нарушение" с комментарием событию ID 194 в TBS 1 с помощью

```

curl 'https://example.com/xapi/event/195/userDecision?tbsId=1' \
  -X PATCH \
  -H 'X-API-Auth-Token: 1792jmclf7ferlikuhby' \
  -H 'X-API-Version: 1' \
  -H 'X-API-CompanyId: IW' \
  -H 'X-API-ImporterName: dome' \
  -H 'Content-Type: application/json' \
  --data-binary '{"USER_DECISION": "Violation", "NOTE": "Назначение вердикта через Dome. Сотрудник Иванов Виктор Павлович."}'

```

1.4.6.15.2 Назначение вердикта пользователю множеству событий

Версия API: ≥ 1.2

Метод: PATCH

Ресурс: event/userDecision

Описание:

Метод позволяет массово назначать решения пользователя (принимает не более **1000** решений пользователя, иначе возвращается сообщение об ошибке с кодом **413**). ID события и ID табличного пространства передаются в теле запроса с решением пользователя и комментарием.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "array",
  "title": "Set batch user decision schema",
  "items": {
    "$id": "#/items",
    "type": "object",
    "required": [
      "OBJECT_ID",
      "TBS_ID",
      "USER_DECISION"
    ],
    "properties": {
      "OBJECT_ID": {
        "$id": "#/items/properties/OBJECT_ID",
        "type": "integer",
        "title": "ID События"
      },
      "TBS_ID": {
        "$id": "#/items/properties/TBS_ID",
        "type": "integer",
        "title": "ID табличного пространства"
      },
      "USER_DECISION": {
        "$id": "#/items/properties/USER_DECISION",
        "type": "string",
        "title": "User decision",
        "enum": [
          "NotProcessed",
          "Violation",
          "NoViolation",
          "AdditionalProcessingNeeded"
        ]
      },
      "NOTE": {
        "$id": "#/items/properties/NOTE",
        "type": "string",
        "title": "Note",
        "maxLength": 1000
      }
    }
  }
}
```

Code Block 57 Тело запроса: JSON схема для множественного назначения вердиктов

```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "array",
  "title": "Array result",
  "items": {
    "$id": "#/items",
    "type": "object",
    "title": "Result setting user decision schema",
    "required": [
      "OBJECT_ID",
      "TBS_ID",
      "VERDICT",
      "USER_DECISION",
      "FORWARD_STATE_CODE"
    ],
    "properties": {
      "OBJECT_ID": {
        "$id": "#/items/properties/OBJECT_ID",
        "type": "integer",
        "title": "ID События"
      },
      "TBS_ID": {
        "$id": "#/items/properties/TBS_ID",
        "type": "integer",
        "title": "ID табличного пространства"
      },
      "VERDICT": {
        "$id": "#/items/properties/VERDICT",
        "type": "string",
        "title": "Вердикт",
        "enum": [
          "Allowed",
          "Quarantined",
          "Forbidden"
        ]
      },
      "USER_DECISION": {
        "$id": "#/items/properties/USER_DECISION",
        "type": "string",
        "title": "Проставленное решение пользователя",
        "enum": [
          "NotProcessed",
          "NoViolation",
          "Violation",
          "AdditionalProcessingNeeded"
        ]
      },
      "FORWARD_STATE_CODE": {
        "$id": "#/items/properties/FORWARD_STATE_CODE",
        "type": "string",
        "title": "Статус отправки",
        "examples": [
          "Sent",
          "NotSent",
          "Pending"
        ]
      }
    }
  }
}

```

Code Block 58 Тело ответа: JSON схема для множественного назначения вердиктов

Пример:

```
curl 'https://example.com/xapi/event/userDecision' \ -X PATCH \ -H 'X-API-Auth-Token: 1792jmclf7ferlikuhby' \ -H 'X-API-Version: 1' \ -H 'X-API-CompanyId: IW' \ -H 'X-API-ImporterName: dome' \ -H 'Content-Type: application/json' \ --data-binary '[{"OBJECT_ID": 1, "TBS_ID": 1, "USER_DECISION": "Violation", "NOTE": "Проставление вердикта через Dome. Сотрудник Иванов Виктор Павлович."}, {"OBJECT_ID": 24, "TBS_ID": 2, "USER_DECISION": "AdditionalProcessingNeeded"}]'
```

1.4.7 Получение списка пользователей консоли

Описание: Список пользователей консоли Traffic Monitor (включая бывших сотрудников)

Метод: `GET`

Ресурс: `user`

Параметры (передаются в виде JSON):

- `start` – с какой позиции начать отдавать данные пользователей (по умолчанию 0)
- `limit` – сколько комплектов данных пользователей отдавать (по умолчанию 100, не может быть больше 1000)

```
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
```

Code Block 59 JSON-схема отдачи списка пользователей

Пример:

```
curl -X GET "https://server.company.ru/xapi/user?start=1&limit=100" -H "accept: application/json" -H "X-Enable-OpenAPI: 1"
```

Code Block 60 Запрос

```
{
  "data": [
    {
      "USER_ID": 0,
      "USERNAME": "string",
      "DISPLAY_NAME": "string",
      "STATUS": 0,
      "CREATE_DATE": "string",
      "EMAIL": "string",
      "NOTE": "string"
    }
  ],
  "meta": {
    "totalCount": 0,
    "start": 0,
    "limit": 0
  }
}
```

Code Block 61 Ответ

1.4.8 Получение результатов выполнения запроса Traffic Monitor

1.4.8.1 Получение списка запросов

Описание: Список запросов Traffic Monitor

Метод: *GET*

Ресурс: *query*

Параметры (передаются в виде JSON):

- *userId* - id пользователя, список запросов которого требуется получить
- *start* - с какой позиции начать отдавать запросы (по умолчанию 0)
- *limit* - сколько запросов отдавать (по умолчанию 100, не может быть больше 1000)

По запросу будет предоставлен список уже существующих в ТМ запросов с указанием ID запроса и его текстовым именем.

JSON-схема отдачи списка запросов:

```
{
  "data": [
    {
      "QUERY_ID": 0,
      "DISPLAY_NAME": "string"
    }
  ],
  "meta": {
    "totalCount": 0,
    "start": 0,
    "limit": 0
  }
}
```

Пример запроса:

```
curl -X GET "https://server.company.ru/xapi/query?userId=5&start=0&limit=10" -H
"accept: application/json" -H "X-Enable-OpenAPI: 1"
```

1.4.8.2 Запуск запроса на выполнение

Описание: Запуск запроса на выполнение в Traffic Monitor

Метод: `POST`

Ресурс: `query/{queryId}/exec`

Параметры:

- `queryId` – ID запроса, который требуется запустить

JSON-схема ответа на запуск запроса:

```
{
  "data": {
    "QUERY_ID": 0,
    "QUERY_TYPE": "query",
    "DISPLAY_NAME": "string",
    "CREATE_DATE": "string",
    "CHANGE_DATE": "string"
  }
}
```

Пример запроса:

```
curl -X POST "https://server.company.ru/xapi/query/0/exec" -H "accept:
application/json" -H "X-Enable-OpenAPI: 1"
```

1.4.8.3 Получение статуса запроса

Описание: Получение статуса запроса

Метод: `GET`

Ресурс: `query/{queryId}/status`

Параметры:

- `queryId` - id запроса, статус которого требуется получить

JSON-схема ответа:

```
{
  "data": {
    "QUERY_ID": 0,
    "USER_ID": 0,
    "STATUS": 0,
    "RUN_DATE": "string",
    "COMPLETE_DATE": "string",
    "PERCENT_COMPLETE": 0,
    "OBJECT_COUNT": 0
  }
}
```

Пример запроса:

```
curl -X GET "https://server.company.ru/xapi/query/0/status" -H "accept:
application/json" -H "X-Enable-OpenAPI: 1"
```

1.4.8.4 Получение результатов запроса

Описание: Список с результатами запросов Traffic Monitor

Метод: `GET`

Ресурс: `query/{queryId}/result`

Параметры (передаются в виде JSON):

- `queryId` - id запроса;
- `start` - с какой позиции начать отдавать запросы (по умолчанию 0)
- `limit` - сколько запросов отдавать (по умолчанию 100, не может быть больше 1000)
- `fields`:
 - `OBJECT_ID` - идентификатор события в ТМ (положительное целое число);
 - `TBS_ID` - идентификатор табличного пространства (положительное целое число).

В ответ на ранее инициированный запрос (см. "[Запуск запроса на выполнение](#)") будет предоставлен список событий с указанием их ID.

JSON-схема ответа:

```
{
  "data": [
    {
      "OBJECT_ID": 0,
      "TBS_ID": 0
    }
  ],
  "meta": {
    "totalCount": 0,
    "start": 0,
    "limit": 0
  }
}
```

Пример запроса:

```
curl -X GET "https://server.company.ru/xapi/query/5/result?start=0&limit=10" -H
"accept: application/json" -H "X-Enable-OpenAPI: 1"
```