

INFOWATCH

InfoWatch Traffic Monitor SDK. Получение доступа к событиям InfoWatch Traffic Monitor внешними системами

06/08/2018

© АО “ИнфоВотч”

Тел./Факс +7 (495) 229-00-22

http://www.infowatch.ru

СОДЕРЖАНИЕ

1 Назначение программного интерфейса DataExport API 3

2 Авторизация для доступа к DataExport API 4

2.1 Работа с плагинами 4

2.1.1 Добавление плагина 4

2.1.2 Удаление плагина 4

2.2 Работа с токенами 4

2.2.1 Добавление токена 4

2.2.2 Редактирование данных токена 5

2.2.3 Обновление значения токена 5

2.2.4 Копирование токена 5

2.2.5 Удаление токена 5

3 Функции DataExport API 6

4 Регистрация стороннего компонента DataExport API 7

4.1 Формат файла регистрации стороннего компонента DataExport API 7

5 Функциональное описание программного интерфейса DataExport API 9

5.1 Общие заголовки GET/POST-запросов для DataExport API 9

5.2 Получение извлеченного текста события 9

5.3 Получение метаданных события 9

5.4 Получение перехваченного файла события 10

5.5 Получение списка каталогов ОЗ 10

5.6 Получение списка событий 12

5.7 Получение списка тегов 13

1 Назначение программного интерфейса DataExport API

Данный программный интерфейс предназначен для интеграции InfoWatch Traffic Monitor (далее Traffic Monitor) с внешними системами-приемниками данных. В рамках реализации DataExport API внешние системы получают из Traffic Monitor содержимое и метаданные событий. DataExport API поддерживает только операцию чтения данных из Traffic Monitor.

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.

2.1 Работа с плагинами

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.

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.

2.2 Работа с токенами

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

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.

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.

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  .

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  .

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  .

3 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;
  - Решение пользователя;
  - Путь к файлу;
  - Адрес веб-ресурса;
  - Объекты защиты и каталоги;
  - Решение пользователя;
  - Размер события;
  - Теги;
- Возможность указать набор атрибутов, по которым будет осуществляться фильтрация:
  - Дата и время перехвата;
  - Тип события;
  - Протокол;
  - Каталог ОЗ;
  - Решение пользователя;
  - Размер события;
  - Теги;
- 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 бинарных данных отдельного события.

4 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 в пункте 2;
4. Загрузить файл регистрации стороннего компонента в систему Traffic Monitor через интерфейс консоли Traffic Monitor. Раздел «Управление» - «Плагины». (см. "Traffic Monitor. Руководство администратора");
5. Получить токен доступа в свойствах загруженного стороннего компонента;

Система готова отправлять данные о событиях заданной компании.

4.1 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. До 40 символов. **Пример:** `"PLUGIN_ID": "189C38D390396EB6E0530100007F1CA200000001",`
`"DISPLAY_NAME": "",`	Строка	Да	Отображаемое имя плагина. **Пример:** `"DISPLAY_NAME": "Имя плагина",`

Содержимое	Тип данных	Является обязательным	Описание
`"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. **Пример:** `"LICENSE": [ { "PATH":"licenses/файл лицензии 1.license" }, { "PATH":"licenses/файл лицензии 2.license" }, ],`
`"PATTERN_SEARCH_LICENSE": "",`	Строка	Нет	Шаблон для поиска загруженных ранее лицензий для привязки их к плагину. Имеет формат: `{operator:or|and,child[{name:value}]}`". **Пример:** `"PATTERN_SEARCH_LICENSE": "{\"operator\":\"or\",\"conditions\":[{\"origin\":\"dm\"},{\"origin\":\"dmmobile\"}]}",`
`"DATA_RECEIVER": [ "" ]`	Массив Строка	Да	Список идентификаторов систем-приемников данных из Traffic Monitor. **Пример:** `"DATA_RECEIVER": [ "Events reciever1", "Events reciever2" ]`
`}`			

5 Функциональное описание программного интерфейса DataExport API

5.1 Общие заголовки GET/POST-запросов для DataExport API

Каждый запрос к серверу должен содержать следующие обязательные заголовки:

- X-API-Version – указывается номер версии программного интерфейса. **Текущая версия: 1**;
- X-API-CompanyId – указывается идентификатор компании-производителя;
- X-API-ImporterName – указывается продукт, принимающий данные;
- X-API-Auth-Token – указывается токен авторизации.

Если сервер не поддерживает версию, код ответа - **400**.
Если лицензия не проходит проверку, код ответа - **403**.
Если указан невалидный токен, код ответа - **401**.
Если указан валидный токен, но у пользователя API недостаточно прав, код ответа - **403**.

Пример:

Заголовки запроса

```
X-API-Version: 1
X-API-CompanyId: iw
X-API-ImporterName: aplab
X-API-Auth-Token: gla20dos1pcdbr69caxz
```

5.2 Получение извлеченного текста события

Метод: `GET`

Ресурс: `event/{id}/text`

Параметры:

- `{id}` - OBJECT_ID события

Поддерживается HTTP-заголовок `Range`

Ответ: Весь извлеченный текст перехваченного события (если есть нескольких вложенных документов, то текст извлекается из всех документов и склеивается в один блок).

```
Content-type: text/plain.
```

5.3 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:

Получить событие с OBJECT_ID "1" с отправителями, получателями:

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

5.4 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.

5.5 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:

JSON схема отдачи списка каталогов.

```
{
    "$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"
    ]
}
```

Пример:

Запрос

```
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
```

Ответ

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

5.6 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]`)
  - `object_type_code[]` — тип события (возможно несколько значений)
  - `protocol[]` — протокол (возможно несколько значений)
  - `user_decision[]` — решение пользователя (возможно несколько значений)

    - `object_size[from]` — от скольки байт искать событие (можно использовать отдельно от `object_size[to]`)
    - `object_size[to]` — до скольки 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.)
- `start` - с какой позиции начать отдавать события (по умолчанию 0)
- `limit` - сколько событий отдавать (по умолчанию 100, не может быть больше 1000)

Ответ: Список событий, отсортированный по времени перехвата (от новых к старым).

Пример:

Получить первые 100 событий с за январь 2015 года:

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

5.7 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:

JSON схема отдачи списка тэгов

```
{
   "$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"
    ]
}
```

Пример:

Запрос

```
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
```

Ответ

```
{
    "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
    }
}
```