

DRE Event Server

Руководство пользователя

Индекс	DREES-UG
Конфиденциальность	Публичный - L0
Ревизия	1.0
Статус	Согласован

Содержание

1. Аннотация	4
2. Начало работы	5
2.1. Отличия Windows версии	5
2.2. Доступ к API	5
3. Возможности утилиты	6
3.1. Общие сведения для выполнения выгрузок	7
3.1.1. Формирование запроса для выгрузки данных	7
3.1.2. Флаги	7
3.1.2.1. Фильтры запросов	9
3.2. Режимы работы	11
3.2.1. Dump	11
3.2.1.1. Партиционирование данных	11
3.2.1.2. Фильтрация данных	11
3.2.2. Stream	11
3.2.2.1. Сохранение состояния выгрузки	12
3.2.2.2. Фильтрация данных	12
3.3. YAML Шаблоны	13
3.3.1. Общие сведения	13
3.3.2. Структура шаблонов	13
3.3.2.1. Скрытые поля	15
3.3.2.2. Форматы даты\времени (date)	15
3.3.3. Доступные исходные поля для шаблонов	17
3.3.3.1. Информация о событии (events)	17
3.3.3.2. Информация о клиенте (clients)	17
3.3.4. Пример шаблона	18
3.3.5. Описание шаблонов	19
3.3.5.1. operationState.SmartHomeMonitoring	19
3.3.5.2. clientsHistoricity	22
3.3.5.3. vod.ContentWatched	23
3.3.5.4. operationState.OTTMonitoring	25
3.3.5.5. simple	28
3.3.5.6. simple.old.string.timestamp	30
4. Логирование	32
4.1. Форматы лог файлов	32
4.1.1. Consumer Client	32

4.1.2. Event Server	32
4.1.3. Api Gateway	33
4.1.4. Consumer Server	33

1. Аннотация

Документ предназначен для технических специалистов, занимающихся выгрузкой данных с использованием системы мониторинга DRE Event Server (далее - DREES, также в документе может употребляться условное наименование системы - Providence) и обладающих навыками работы с компьютером на профессиональном уровне.

2. Начало работы

Для получения выгрузок с DREES, необходимо иметь утилиту client, которая представляет из себя исполняемый файл для Windows, Linux или MacOS систем с архитектурой amd64/386.



Примечание

Для корректного получения выгрузок с DREES необходимо использовать последнюю версию утилиты client (версия утилиты client должна была равна версии DREES).

Использование предыдущих версий утилиты client не гарантирует получения корректных выгрузок.

2.1. Отличия Windows версии

При использовании утилиты на Windows, имя исполняемого файла оканчивается на ".exe" (в приведенных примерах client в начале строки означает client.exe, т.е. под client понимается условное название исполняемого файла). Запуск утилиты должен производиться с правами администратора. Во всех примерах далее будет использоваться Linux версия исполняемого файла (в тексте примеров обозначается client в начале строки, где под client понимается условное название исполняемого файла).

2.2. Доступ к API

Доступ к API предоставляется при наличии ключа авторизации в переменной окружения CONSUMER_SERVER_ACCESS_TOKEN. Так же с помощью параметров CONSUMER_SERVER_ADDRESS и CONSUMER_SERVER_TLS можно направить client на версию DREES, отличную от Production.

По умолчанию утилита будет обращаться в Production окружение по адресу [domain.name]:443 в режиме TLS соединения.

3. Возможности утилиты

Чтобы получить описание возможностей утилиты, можно, используя флаг `--help`, выполнить команду (в примере под `client` понимается условное название исполняемого файла):

```
consumer help
client --help
```

Данная команда выведет помощь по использованию на английском языке (список доступных к использованию [флагов](#) (Flags) и команд (Commands) для разных [режимов](#)). Пример того как это выглядит в консоли:

```
[natalia@MacBook-Pro-Natalia-859 client % ./consumer-cli-darwin-amd64 --help
Usage: consumer-cli-darwin-amd64 <command>

Supports three command: tpl, dump, stream. One deprecated: polling (use stream)

Flags:
  -h, --help          Show context-sensitive help.
  --debug             write debug log to file near executable
  --version

Commands:
  dump csv <template> <start-date> [<end-date>]
  dump single-csv <template> <start-date> [<end-date>]
  dump logstash <template> <start-date> [<end-date>]
  dump stdout <template> <start-date> [<end-date>]
  stream logstash <template>
  stream stdout <template>
  tpl list
    get list of templates
  tpl render <name>
  tpl render-all <directory>
    write yamls of template describes to directory

Run "consumer-cli-darwin-amd64 <command> --help" for more information on a command.
```

3.1. Общие сведения для выполнения выгрузок

3.1.1. Формирование запроса для выгрузки данных

Выполнение выгрузки вызывается командой со следующей структурой: `client consume_type output template <start-date> [<end-date>] params`, где:

- `client` - название исполняемого файла.
- `consume_type` - режим работы:
 - `dump`;
 - `stream`;
- `output` - способ вывода:
 - варианты для режима `dump`;
 - варианты для режима `stream`.
- `template` - шаблон выгрузки. Имя шаблона выгрузки (`template`) - это первый параметр любой конечной команды. Так же может являться путём до `yaml` файла, содержащего описание шаблона выгрузки.
Пример: `client dump csv vod.ContentWatched 2019-01 -f "group=vod"`
Список доступных шаблонов можно получить выполнив команду `client tpl list`
Подробнее о шаблонах [тут](#).
- `<start-date> [<end-date>]` - временной интервал для выполнения выгрузки данных. Данные параметры необходимы к заполнению только для режима `dump`.
- `params` - дополнительные флаги. Описание флагов приведено в разделе [Флаги](#).

3.1.2. Флаги

В данном разделе приведен перечень флагов, доступных для работы утилиты в режимах `dump`, `stream` (подробнее о режимах [тут](#)) и описание их функций:

Флаг	Описание	Доступность для способа вывода	
		Режим dump	Режим stream
<code>-h или --help</code>	позволяет вывести справку о любой команде.	csv logstash stdout single-csv	logstash stdout
<code>--debug</code>	позволяет выводить отладочную информацию во время работы утилиты в отдельный файл в той же директории, что исполняемый файл.	csv logstash stdout single-csv	logstash stdout
<code>--version</code>	позволяет просмотреть версию утилиты.	csv logstash stdout single-csv	logstash stdout

<code>-f, --filters=KEY=VALUE;...</code>	Использования фильтров при выгрузке данных через утилиту client. Подробнее об использовании данного флага - в разделе "Фильтры запросов"	csv logstash stdout single-csv	logstash stdout
<code>--timezone-location="Local"</code>	фильтрация выгрузки событий по часовой зоне. Значение по умолчанию: Local (текущий часовой пояс, установленный на машине, на которой запущен клиент).	csv single-csv logstash stdout	-
<code>--headers</code>	добавляет заголовки (названия столбцов) в начале файла CSV.	csv single-csv	-
<code>--file-path=""</code>	установка пути к директории, куда будет сохраняться файл выгрузки CSV. Значение по умолчанию - текущая директория. <i>Например, --file-path="/dir1/dir2/" - в данном случае файл выгрузки будет сохранен по указанному пути</i>	csv	-
<code>--file-prefix="events_"</code>	задание префикса файлу выгрузки. <i>Например: --file-prefix="events_" - в данном случае файл выгрузки будет сохранен с префиксом events_ в директории, указанной флагом --file-path</i>	csv	-
<code>--encoding=""</code>	задание кодировки для файла CSV. По умолчанию (если не указать ничего): UTF-8. Возможные варианты кодировок можно посмотреть в таблице по ссылке: https://encoding.spec.whatwg.org/#names-and-labels	csv single-csv	-
<code>--partition-size="d"</code>	размер одного сегмента данных (временного промежутка) для разбиения при выполнении выгрузки, возможные варианты - день "d", час "h", минута "m".	csv	-
<code>--threads=4</code>	количество одновременных потоков для выполнения выгрузки в режиме dump.	csv	-
<code>--address=""</code>	tcp-адрес сервера Logstash.	logstash	logstash
<code>--keepalive=30s</code>	интервал поддержания активности при выгрузке в Logstash TCP.	logstash	logstash
<code>--reconnect-timeout=1s</code>	интервал ожидания повторного подключения TCP при сетевой ошибке.	logstash	logstash
<code>--pretty-print</code>	структурировать текст под JSON формат.	stdout	stdout
<code>--file="consumer.csv"</code>	в зависимости от указанной в кавычках информации - установка одного из двух вариантов: <ol style="list-style-type: none">пути к директории, куда будет сохраняться файл выгрузки CSV. <i>Например, --file="/dir1/dir2/" - в данном случае файл выгрузки будет сохранен по указанному пути.</i>задание названия файла выгрузки. <i>Например: --file="NewFileName.csv" - в данном случае файл выгрузки будет сохранен в текущей директории, откуда запускается утилита, но название файла будет NewileName.csv</i>	single-csv	-
<code>--actual-clients</code>	флаг включает режим выгрузки событий с указанием состояния клиента на момент получения события из таблицы `clients_history` <i>Примечание: в данном режиме выгрузки не работают фильтры -f для следующих типов фильтра: device_type, dre_id, app_name, hw_id.</i>	csv single-csv logstash stdout	-

3.1.2.1. Фильтры запросов

Для получения выгрузок данных в режимах работы dump и stream с использованием одного/нескольких фильтров необходимо использовать флаг *-f или --filters*

Фильтр имеет следующий формат (обе записи равнозначны):

```
-f "тип_фильтра1=значение1,значение2,...;тип_фильтра2=значение1,...;..."
```

```
-f="тип_фильтра1=значение1,значение2,...;тип_фильтра2=значение1,...;..."
```

где запятые в запросе означают "ИЛИ", точки с запятой - "И".

Таким образом - у всех событий должен быть тип_фильтра1 равен ИЛИ значению1 ИЛИ значению2 И тип_фильтра2 равен значению1 и тд.

Возможные типы фильтров:

- client_id - по идентификатору клиента;
- name - по имени события и имени группы. Данный тип фильтра нельзя использовать в режиме работы dump. При использовании типа фильтра "name" в запросе должен так же обязательно использоваться тип фильтра "group".
- group - по имени группы;
- app_name - по названию приложения;
- dre_id - по DRE ID;
- device_type - по типу устройства.
- hw_id - по HW ID устройства;
- attribute - по атрибуту события. Данный тип фильтра доступен для использования только в режиме работы dump.

Фильтр attribute фильтрует атрибуты из json поля attributes базы данных, поэтому формат его ввода в консоль немного отличается от других фильтров:

```
-f "attribute.имя_поля_в_json.можно_еще_одно_вложенность_не_ограничена=значение1,..."
```



Если при выполнении выгрузки флаг *-f или --filters* не указан, то в режимах работы dump и stream будет следующее поведение:

- dump - будет получена ошибка. Выполнять выгрузку в режиме работы dump без указания фильтра (-f, --filters) не рекомендуется.
- stream - выгрузятся требуемые события.

Примеры:

-f "dre_id=1,2;device_type=DT_ANDROID;app_name=app1,app2" # Получить все события, у которых dre_id = 1 или 2 И тип девайса android И app_name = app1 или app2.

-f "hw_id=1;attribute.contentId=123,7;attribute.app.contentsec=3" # Получить все события, у которых hw_id = 1 И events.attributes.contentId = 123 ИЛИ 7 И events.attributes.app.contentsec = 3.

-f "group=iptv" # Получить все события, у которых группа iptv

-f "name=FirstScreenStart" # Ошибка, так как нет фильтра по группе.

3.2. Режимы работы

Утилита предоставляет 2 основных способа работы: дамп (dump) и стриминг (stream).

3.2.1. Dump

Данный режим работы позволяет скачать данные за определённый период времени. Для него поддерживаются следующие способы вывода :

- stdout - вывод в стандартный вывод терминала
- csv - вывод в партиционированные CSV файлы
- single-csv - вывод в единый CSV файл
- logstash - вывод в TCP сервер Logstash

Примеры вызова команды:

```
client dump stdout vod.ContentWatched 2020-01 #Выгрузка за период с [2020-01-01 00:00:00 по 2020-02-01 00:00:00) с партиционированием по суткам (d)
```

```
client dump csv vod.ContentWatched 2020-01-02 --partition-size=h #Выгрузка в CSV за период с [2020-01-02 00:00:00 по 2020-01-03 00:00:00) с партиционированием по часам (h)
```

```
client dump csv vod.ContentWatched 2020-01-02 --headers --partition-size=m #Выгрузка за период с [2020-01-02 00:00:00 по 2020-01-03 00:00:00) с партиционированием по минутам (m) и с добавлением названий столбцов в CSV файлах
```

3.2.1.1. Партиционирование данных

При дампе данных в CSV имеется специальная возможность партиционирования файлов по периоду времени - день (d), час (h) или минута (m). По умолчанию используется партиционирование по дням. Для избежания партиционирования используйте команду `client dump csv-single`.

3.2.1.2. Фильтрация данных

В утилите имеются возможности для фильтрации данных, подробнее - в разделе ["Фильтры запросов"](#).

3.2.2. Stream

Данный режим работы позволяет передавать данные в около-реальном времени начиная с определённой даты. Для него поддерживаются следующие способы вывода:

- stdout - вывод в стандартный вывод терминала
- logstash - вывод в TCP сервер Logstash

client stream stdout simple --filters="group=vod" #команда начнёт выводить в stdout терминала все поступающие события для группы событий vod

client stream logstash operationState.OTTMonitoring --filters="group=operationState" --address=<logstash-host:port> # команда начнёт стриминг в logstash по указанному адресу

3.2.2.1. Сохранение состояния выгрузки

Режим предусматривает сохранение состояния выгрузки на стороне сервера с возможностью последующей догрузки данных. При выполнении запроса с одинаковыми критериями выгрузки состояние будет сохраняться.

3.2.2.2. Фильтрация данных

В утилите имеются возможности для фильтрации данных, подробнее - в разделе "[Фильтры запросов](#)".

3.3. YAML Шаблоны

3.3.1. Общие сведения

С помощью шаблонов можно задавать какие именно данные необходимо выгружать и какие трансформации значений производить.

Для указания шаблона при выгрузке необходимо передать `id`, название или путь к YAML шаблону при запуске `client` после указания плагина: `client [режим работы] [название плагина] [шаблон] [ключи запуска]`

Список доступных шаблонов можно получить выполнив команду: `client tpl list`

Конфигурацию конкретного шаблона можно получить, выполнив команду: `tpl render <template_name>`

Для выгрузки всех шаблонов в `yaml` формате в указанную папку используется команда: `tpl render-all <directory>`

Доступные шаблоны:

- `operationState.OTTMonitoring` - выгрузка событий `operationState.OTTMonitoring`;
- `operationState.SmartHomeMonitoring` - выгрузка событий `operationState.SmartHomeMonitoring`;
- `simple` - простой шаблон для поддержки любого рода событий. Выгружает все `timestamp` в формате `RFC3339nano`;
- `simple.old.string.timestamp` - старый шаблон `simple`, отдающий `timestamp` в виде строки. Выгружает все `timestamp` в формате `RFC3339nano` за исключением способа вывода в `csv` - в этом случае будет использоваться формат `RFC3339`;
- `vod.ContentWatched` - выгрузка событий `vod.ContentWatched`.
- `clientsHistoricity` - выгрузка историчности изменений для клиента.

Описание шаблонов (исходные и получаемые поля, возможные варианты преобразований) приведено в разделе "[Описание шаблонов](#)".

3.3.2. Структура шаблонов

Шаблон состоит из нескольких разделов, описывающих необходимые данные:

- `columns` - список необходимых полей. Для каждого поля необходимо описывать:
 - `dbName` - название исходного поля.
 - `name` - название поля в итоговых данных.
- `description` - текстовое описание шаблона.
- `name` - название шаблона.
- `encoding` - кодировка при выводе в текстовый файл. Применимо для `CSV`.

- pipeline - описание механизмов преобразования исходных данных (к пр. конвертация числовых типов устройств в строковые значения). Допустимые типы данных и их параметры:
 1. intEnum - маппинг числовых значений к строковым.
 - a. column - название поля.
 - b. map - описание маппинга исходного цифрового значения к строковому.
 - c. unknownValue - значение по умолчанию для не описанных значений.
 2. stringEnum - маппинг строковых значений к строковым
 - a. column - название поля.
 - b. map - описание маппинга исходного строкового значения к строковому.
 - c. unknownValue - значение по умолчанию для не описанных значений.
 3. concat - объединение значений нескольких полей в одну.
 - a. columns - список исходных полей.
 - b. separator - разделитель.
 - c. columnName - название итогового поля.
 4. date - конвертация формата даты\времени.
 - a. column - название поля.
 - b. originalFormat - исходный формат даты\времени.
 - c. format - итоговый формат даты\времени.
 5. rawJson - конвертация исходных данных к JSON формату.
 - a. column - название поля.
 6. string - конвертация исходных данных к строковому типу.
 - a. column - название поля.
 7. coalesce - выбор первого встретившегося значения из нескольких полей.
 - a. columns - поля в заданном порядке.
 - b. toColumn - имя поля, в которое поместить результат.

3.3.2.1. Скрытые поля

В некоторых ситуациях может возникнуть необходимость обработать результат нескольких выбираемых полей и сложить его в результирующее. Для того чтобы избежать добавление исходных полей в результат выгрузки, можно скрывать такие поля двумя способами:

1. Если указать пустой 'dbName', то в результат добавится новое пустое поле.
2. Если указать имя поля 'name', начинающееся с точки (например, '.dreld'), то оно не будет добавлено в итоговую выгрузку.

Это особенно полезно при работе с преобразованиями данных "coalesce" и "concat".

coalesce example:

name: coalesce-example

description: Шаблон для демонстрации coalesce

columns:

...

- dbName: clients.dre_id # clients.dre_id - это последнее доступное значение dre_id

name: .clientsDreld # поле будет скрыто из выгрузки, т.к. имя начинается на "."

targetType: string

- dbName: events.dre_id # events.dre_id - это dre_id, который был прислан в момент возникновения события

name: .eventsDreld # поле будет скрыто из выгрузки, т.к. имя начинается на "."

targetType: string

- name: dreld # dbName не указан - пустое поле

targetType: string

...

pipeline:

- coalesce:

columns:

- .eventsDreld

- .clientsDreld

toColumn: dreld # заполняем пустое поле результатом

3.3.2.2. Форматы даты\времени (date)

Название	Пример
DF_UNIX_SECONDS	1591881177
DF_UNIX_MILLI	1591881177261

DF_UNIX_NANO	1591881177261967893
DF_RFC3339	2020-06-11T16:14:54+03:00
DF_RFC3339Nano	2020-06-11T16:14:54.808525904+03:00

3.3.3. Доступные исходные поля для шаблонов

3.3.3.1. Информация о событии (events)

Данная таблица содержит информацию о событии и информацию о клиенте, которую он прислал в момент возникновения события:

Название поля	Тип данных	Описание
events.id	числовой	Идентификатор события
events.name	строковый	Название события
events.group	строковый	Группа
events.timestamp	временная метка с часовым поясом	Временная метка
events.attributes	JSON-строка	Дополнительные атрибуты
events.client_id	строковый	Идентификатор клиента
events.ip	строковый	IP-адрес
events.local_timestamp	временная метка с часовым поясом	Локальный часовой пояс

3.3.3.2. Информация о клиенте (clients)

Информация о клиенте - это последняя доступная информация о клиенте. То есть, в случаях когда параметры клиента изменяются, они обновляются в данном словаре. Для историчных выгрузок нужно использовать соответствующие поля в таблице events.

Название поля	Тип данных	Описание
clients.id	числовой	Идентификатор клиента
clients.device_brand	строковый	Фирма изготовитель устройства
clients.device_model	строковый	Модель устройства
clients.device_type	числовой	Тип устройства
clients.os_name	строковый	Название ОС
clients.os_version	строковый	Версия ОС
clients.app_name	строковый	Название приложения

clients.app_version	строковый	Версия приложения
clients.dre_id	строковый	DRE ID клиента
clients.hw_id	строковый	Идентификатор устройства
clients.user_agent	строковый	Название приложения на сетевом уровне
clients.serial_number	строковый	Серийный номер
clients.appsflyer_id	строковый	идентификатор в аналитической системе Appsflyer
clients.module_version	строковый	Номер версии ПО крипточипа

3.3.4. Пример шаблона

columns:

- dbName: events.timestamp
name: timestamp
targetType: datetime
- dbName: events.attributes
name: attributes
targetType: json
- dbName: clients.device_type
name: deviceType
targetType: string

description: Primer shablona

id: "7"

name: example

encoding: windows-1251

pipeline:

- date:
 - column: timestamp
 - format: DF_RFC3339
- intEnum:
 - column: deviceType
 - map:
 - "1": STB-GW
 - "2": STB
 - unknownValue: unknown
- rawJson:
 - column: attributes

3.3.5. Описание шаблонов

3.3.5.1. operationState.SmartHomeMonitoring

Поле шаблона	Исходное значение	Модификатор
timestamp	events.timestamp	Поле отдается в не измененном виде
hwId	clients.hw_id	Поле отдается в не измененном виде
clientId	events.client_id	Поле отдается в не измененном виде
deviceBrand	clients.device_brand	Поле отдается в не измененном виде
deviceModel	clients.device_model	Поле отдается в не измененном виде
serialNumber	clients.serial_number	Поле отдается в не измененном виде
moduleVersion	clients.module_version	Поле отдается в не измененном виде
eventId	events.id	Поле отдается в не измененном виде
eventName	events.name	Поле отдается в не измененном виде
eventGroup	events.group	Поле отдается в не измененном виде
daemon	events.attributes.daemon	0: "Not active", 1: "Active, cloud not connected", 2: "Active, cloud connected"
dongle	events.attributes.dongle	"0": "Not active"
configuration	events.attributes.configuration	0: "Not available", 1: "Ok"
startSmh	events.attributes.startSmh	0: "Not active", 1: "Ok"
cloudPing	events.attributes.cloudPing	0: "Not available", 1: "Ok"
drmToken	events.attributes.drmToken	0: "Not available", 1: "Ok"
smsBill	events.attributes.smsBill	0: "Not active",
rcBill	events.attributes.rcBill	0: "Not active",

		1: "Ok"
deviceCount	events.attributes.deviceCount	0: "No devices"
drmMetrics	events.attributes.drmMetrics	Поле отдается в не измененном виде
smhStatistics	events.attributes.smhStatistics	Поле отдается в не измененном виде
smhWsDiagnostic	events.attributes.smhWsDiagnostic	Поле отдается в не измененном виде
ip	events.ip	Поле отдается в не измененном виде
domainId	clients.dre_id	Поле отдается в не измененном виде
deviceType	clients.device_type	0: "NONE" 1: "STB" 2: "STB-GW" 3: "STB-CLIENT" 4: "STB-IP" 5: "IOS" 6: "ANDROID" 7: "WEBOS" 8: "TIZEN" 9: "NETCAST" 10: "ANDROID-TV" 11: "MOZILLA" 12: "SAFARI" 13: "TVOS" 14: "HUB-HEIMAN" 15: "SMHDaemon" 16: "HUB-c211" 17: "STB-IPS" 18: "DT_VIDAA" 19: "DT_COOLITA" 20: "DT_TRICOLOROS"
osName	clients.os_name	Поле отдается в не измененном виде
osVersion	clients.os_version	Поле отдается в не измененном виде
appName	clients.app_name	Поле отдается в не измененном виде
appVersion	clients.app_version	Поле отдается в не измененном виде

userAgent

clients.user_agent

Поле отдается в не измененном виде

3.3.5.2. clientsHistoricity

Поле шаблона	Исходное значение	Модификатор
clientId	clients_history.client_id	Поле отдается в не измененном виде
deviceBrand	clients_history.device_brand	Поле отдается в не измененном виде
deviceModel	clients_history.device_model	Поле отдается в не измененном виде
deviceType	clients_history.device_type	Поле отдается в не измененном виде
osName	clients_history.os_name	Поле отдается в не измененном виде
osVersion	clients_history.os_version	Поле отдается в не измененном виде
appName	clients_history.app_name	Поле отдается в не измененном виде
appVersion	clients_history.app_version	Поле отдается в не измененном виде
dreId	clients_history.dre_id	Поле отдается в не измененном виде
hwId	clients_history.hw_id	Поле отдается в не измененном виде
updatedAt	clients_history.updated_at	Поле отдается в не измененном виде
userAgent	clients_history.user_agent	Поле отдается в не измененном виде
serialNumber	clients_history.serial_number	Поле отдается в не измененном виде
appsflyerId	clients_history.appsflyer_id	Поле отдается в не измененном виде
moduleVersion	clients_history.module_version	Поле отдается в не измененном виде

3.3.5.3. vod.ContentWatched

Поле шаблона	Исходное значение	Модификатор
timestamp	events.timestamp	Поле отдается в не измененном виде
hwId	clients.hw_id	Поле отдается в не измененном виде
serialNumber	clients.serial_number	Поле отдается в не измененном виде
clientId	events.client_id	Поле отдается в не измененном виде
deviceBrand	clients.device_brand	Поле отдается в не измененном виде
deviceModel	clients.device_model	Поле отдается в не измененном виде
eventId	events.id	Поле отдается в не измененном виде
eventName	events.name	Поле отдается в не измененном виде
eventGroup	events.group	Поле отдается в не измененном виде
duration	events.attributes.duration	Поле отдается в не измененном виде
startWatchTime	events.attributes.startWatchTime	Преобразование из unix timestamp формата (appsflyerId) в формат RFC3339 (2006-01-02T15:04:05Z07:00)
contentId	events.attributes.contentId	Поле отдается в не измененном виде
ip	events.ip	Поле отдается в не измененном виде
domainId	clients.dre_id	Поле отдается в не измененном виде
deviceType	clients.device_type	0: "NONE" 1: "STB" 2: "STB-GW" 3: "STB-CLIENT" 4: "STB-IP" 5: "IOS" 6: "ANDROID" 7: "WEBOS" 8: "TIZEN" 9: "NETCAST" 10: "ANDROID-TV"

		11: "MOZILLA" 12: "SAFARI" 13: "TVOS" 14: "HUB-HEIMAN" 15: "SMHDaemon" 16: "HUB-c211" 17: "STB-IPS" 18: "DT_VIDAA" 19: "DT_COOLITA" 20: "DT_TRICOLOROS"
osName	clients.os_name	Поле отдается в не измененном виде
osVersion	clients.os_version	Поле отдается в не измененном виде
appName	clients.app_name	Поле отдается в не измененном виде
appVersion	clients.app_version	Поле отдается в не измененном виде
userAgent	clients.user_agent	Поле отдается в не измененном виде

3.3.5.4. operationState.OTTMonitoring

Поле шаблона	Исходное значение	Модификатор
timestamp	events.timestamp	Поле отдается в не измененном виде
hwId	clients.hw_id	Поле отдается в не измененном виде
clientId	events.client_id	Поле отдается в не измененном виде
deviceBrand	clients.device_brand	Поле отдается в не измененном виде
deviceModel	clients.device_model	Поле отдается в не измененном виде
serialNumber	clients.serial_number	Поле отдается в не измененном виде
moduleVersion	clients.module_version	Поле отдается в не измененном виде
eventId	events.id	Поле отдается в не измененном виде
eventName	events.name	Поле отдается в не измененном виде
eventGroup	events.group	Поле отдается в не измененном виде
source	events.attributes.source	0: "Standby" 1: "DTH" 2: "PushVOD" 3: "Timeshift" 4: "OTT Live" 5: "CatchUp" 6: "Startover" 7: "Application" 8: "Media" 9: "VOD"
sourceId	events.attributes.sourceId	Поле отдается в не измененном виде
playlist	events.attributes.playlist	0: "Ok" 1: "Not parsed" 2: "Not downloaded" 3: "Not downloaded MPL"
segment	events.attributes.segment	0: "Ok" 1: "Not decrypted" 2: "Not downloaded"

		3: "No cek_key" 4: "No permissions"
epg	events.attributes.epg	0: "Ok" 1: "Current/Following" 2: "Schedule" 3: "No EPG" 4: "Partially EPG"
status	events.attributes.status	0: "Release" 1: "Developer"
lk	events.attributes.lk	0: "Ok" 1: "Not available"
connection	events.attributes.connection	0: "Ethernet" 1: "Wifi" 2: "USB" 4: "GSM - 2G" 5: "GSM - 3G" 6: "GSM - 4G" 7: "GSM - 5G" 8: "GSM - Unknown"
selectedBitrate	events.attributes.selectedBitrate	0: "Auto"
instantBitrate	events.attributes.instantBitrate	Поле отдается в не измененном виде
realBitrate	events.attributes.realBitrate	Поле отдается в не измененном виде
displayError	events.attributes.displayError	Поле отдается в не измененном виде
DRMAuthStatus	events.attributes.drmMetrics.DRM-auth-status	Поле отдается в не измененном виде
receiverMode	events.attributes.receiverMode	0: "Satellite" 1: "Internet" 2: "Hybrid"
ip	events.ip	Поле отдается в не измененном виде
domainId	clients.dre_id	Поле отдается в не измененном виде
deviceType	clients.device_type	0: "NONE" 1: "STB"

		2: "STB-GW" 3: "STB-CLIENT" 4: "STB-IP" 5: "IOS" 6: "ANDROID" 7: "WEBOS" 8: "TIZEN" 9: "NETCAST" 10: "ANDROID-TV" 11: "MOZILLA" 12: "SAFARI" 13: "TVOS" 14: "HUB-HEIMAN" 15: "SMHDaemon" 16: "HUB-c211" 17: "STB-IPS" 18: "DT_VIDAA" 19: "DT_COOLITA" 20: "DT_TRICOLOROS"
osName	clients.os_name	Поле отдается в не измененном виде
osVersion	clients.os_version	Поле отдается в не измененном виде
appName	clients.app_name	Поле отдается в не измененном виде
appVersion	clients.app_version	Поле отдается в не измененном виде
userAgent	clients.user_agent	Поле отдается в не измененном виде

3.3.5.5. simple

Поле шаблона	Исходное значение	Модификатор
eventId	events.id	Поле отдается в не измененном виде
timestamp	events.timestamp	Поле отдается в не измененном виде
local_timestamp	events.local_timestamp	Поле отдается в не измененном виде
eventName	events.name	Поле отдается в не измененном виде
eventGroup	events.group	Поле отдается в не измененном виде
attributes	events.attributes	Поле отдается в не измененном виде
ip	events.ip	Поле отдается в не измененном виде
clientId	events.client_id	Поле отдается в не измененном виде
hwId	clients.hw_id	Поле отдается в не измененном виде
deviceBrand	clients.device_brand	Поле отдается в не измененном виде
deviceModel	clients.device_model	Поле отдается в не измененном виде
serialNumber	clients.serial_number	Поле отдается в не измененном виде
moduleVersion	clients.module_version	Поле отдается в не измененном виде
dreId	clients.dre_id	Поле отдается в не измененном виде
deviceType	clients.device_type	0: "NONE" 1: "STB" 2: "STB-GW" 3: "STB-CLIENT" 4: "STB-IP" 5: "IOS" 6: "ANDROID" 7: "WEBOS" 8: "TIZEN" 9: "NETCAST" 10: "ANDROID-TV" 11: "MOZILLA" 12: "SAFARI"

		13: "TVOS" 14: "HUB-HEIMAN" 15: "SMHDaemon" 16: "HUB-c211" 17: "STB-IPS" 18: "DT_VIDAA" 19: "DT_COOLITA" 20: "DT_TRICOLOROS"
osName	clients.os_name	Поле отдается в не измененном виде
osVersion	clients.os_version	Поле отдается в не измененном виде
appName	clients.app_name	Поле отдается в не измененном виде
appVersion	clients.app_version	Поле отдается в не измененном виде
userAgent	clients.user_agent	Поле отдается в не измененном виде
appsflyerId	clients.appsflyer_id	Поле отдается в не измененном виде
clientAttributes	clients.attributes	Поле отдается в не измененном виде

3.3.5.6. simple.old.string.timestamp

Поле шаблона	Исходное значение	Модификатор
timestamp	events. timestamp	Поле отдается в не измененном виде
eventName	events.name	Поле отдается в не измененном виде
eventGroup	events.group	Поле отдается в не измененном виде
attributes	events. attributes	Поле отдается в не измененном виде
ip	events.ip	Поле отдается в не измененном виде
clientId	events.client_id	Поле отдается в не измененном виде
hwId	clients.hw_id	Поле отдается в не измененном виде
deviceBrand	clients. device_brand	Поле отдается в не измененном виде
deviceModel	clients. device_model	Поле отдается в не измененном виде
serialNumber	clients. serial_number	Поле отдается в не измененном виде
moduleVersion	clients. module_version	Поле отдается в не измененном виде
dreId	clients.dre_id	Поле отдается в не измененном виде
deviceType	clients. device_type	0: "NONE" 1: "STB" 2: "STB-GW" 3: "STB-CLIENT" 4: "STB-IP" 5: "IOS" 6: "ANDROID" 7: "WEBOS" 8: "TIZEN"

		<p>9: "NETCAST"</p> <p>10: "ANDROID-TV"</p> <p>11: "MOZILLA"</p> <p>12: "SAFARI"</p> <p>13: "TVOS"</p> <p>14: "HUB-HEIMAN"</p> <p>15: "SMHDaemon"</p> <p>16: "HUB-c211"</p> <p>17: "STB-IPS"</p> <p>18: "DT_VIDAA"</p> <p>19: "DT_COOLITA"</p> <p>20: "DT_TRICOLOROS"</p>
osName	clients.os_name	Поле отдается в не измененном виде
osVersion	clients. os_version	Поле отдается в не измененном виде
appName	clients. app_name	Поле отдается в не измененном виде
appVersion	clients. app_version	Поле отдается в не измененном виде
userAgent	clients. user_agent	Поле отдается в не измененном виде
. clientsAppsflyerId	clients. appsflyer_id	Значение appsflyer_id берется из клиента и атрибутов события, для вывода в appsflyerId используется первое непустое значение
. eventsAppsflyerId	events. attributes. appsflyerId	

4. Логирование

Лог-файлы компонентов системы могут быть получены через стандартный механизм доступа к логам в среде kubernetes при помощи команды `kubectl logs <pod name>`.

4.1. Форматы лог файлов

4.1.1. Consumer Client

Лог-файл компонента Consumer Client имеет следующий формат:

[Time] [Level] Info

Time - Дата и время формирования записи с точностью до секунд и с дробными секундами.

Level - Уровень логирования. Один из уровней (по возрастанию): *trace, debug, info, warn, error, fatal*.

Info - Текстовая информация - Дополнительная информация, которую можно использовать для поиска и трассировки ошибок.

Пример записи нескольких строчек лог-файла Consumer Client:

```
[2021-04-23T14:20:04.74306876+02:00][debug] Sent keepalive message
[2021-04-23T14:20:04.7514284+02:00][trace] recv: finish streamLoaderId=c0b5e3d5-282d-4697-8c28-2279dae075f7
[2021-04-23T14:20:04.751475174+02:00][trace] msgpack: start      streamLoaderId=c0b5e3d5-282d-4697-8c28-2279dae075f7
[2021-04-23T14:20:04.751734195+02:00][trace] msgpack: finish  streamLoaderId=c0b5e3d5-282d-4697-8c28-2279dae075f7
```

Пример записи нескольких строчек лог-файла при наличии ошибок:

```
2023-12-26T16:52:58+03:00 INF such info message
2023-12-26T16:52:58+03:00 ERR such error message
```

4.1.2. Event Server

Лог-файл компонента Event Server имеет следующий формат:

[level][component][time][message]

level - уровень логирования. Один из уровней (по возрастанию): *trace, debug, info, warn, error, fatal*.

component - часть системы, вызвавшая сообщение.

time - дата и время формирования записи.

message - логируемое сообщение.

Помимо [Level][Component][Time][msg] в записи могут опционально присутствовать дополнительные поля с информацией.

Пример записи нескольких строчек лог-файла компонента Event Server:

```
{ "level": "info", "component": "grpc", "method": "/gs_labs.lambda.providence.event_server.EventServer/SendEvents", "duration": "0.0002", "time": "2022-09-29T10:17:17+03:00", "message": "request time" }
{ "level": "debug", "component": "pool", "time": "2022-09-29T10:17:17+03:00", "message": "add job with 1 events" }
{ "level": "debug", "component": "pool", "time": "2022-09-29T10:17:18+03:00", "message": "flush events" }
{ "level": "debug", "component": "flusher", "time": "2022-09-29T10:17:18+03:00", "message": "events saved" }
{ "level": "debug", "component": "flusher", "clientCount": 1, "clientIds": [ "32cb0c69-7565-4f73-b44b-c7937f9ebe45" ], "eventIds": [ 44070 ], "time": "2022-09-29T10:17:18+03:00", "message": "Database flusher debug" }
```

4.1.3. Api Gateway

Формат лог-файла аналогичен формату описанному в разделе 4.1.2

4.1.4. Consumer Server

Формат лог-файла аналогичен формату описанному в разделе 4.1.2

© ООО "Цифра", 2019-2025

Документация "DRE Event Server. Руководство пользователя" является объектом авторского права. Воспроизведение всего произведения или любой его части воспрещается без письменного разрешения правообладателя.