Руководство пользователя ПО «DataGo Streaming»

Аннотация

Настоящее руководство описывает подключение клиента к ПО «DataGo Streaming» и проверку поступления событий в базу данных клиента. Сервис работает на стороне DataGo. Отправка событий настраивается клиентом самостоятельно по инструкции либо специалистами DataGo через менеджер тегов клиента. Полученные данные клиент использует в своей базе.

Для описанных действий вход в личный кабинет DataGo Streaming не требуется. Настройки подключения предоставляет DataGo; данные клиент просматривает в своей базе.

Введение

ПО «DataGo Streaming» принимает события сайта и мобильного приложения клиента, обрабатывает их и доставляет в базу данных клиента. Результат работы — записи о событиях, доступные клиенту для SQL-запросов и использования в собственных аналитических инструментах.

Системные требования

Для работы Системы установка ПО клиентом не требуется, так как ПО предоставляется по модели SaaS (Интернет-сервис). Отдельного личного кабинета или приложения, требующего входа пользователя, у Системы нет.

Для подключения необходимы источник событий и база данных, доступная сервису DataGo для доставки событий. При самостоятельной настройке клиенту требуется доступ к настройкам источника. Если настройку выполняет DataGo через менеджер тегов, клиент предоставляет необходимый доступ.

Для дополнительной самостоятельной проверки можно использовать SQL-клиент с правом чтения целевой таблицы и терминал с curl. Эти средства не являются обязательными условиями использования сервиса.

Описание работы в Системе

Для подключения DataGo Streaming клиент предоставляет DataGo доступ к базе данных для записи событий. Отправка событий на адрес приёма DataGo настраивается одним из двух способов: клиентом самостоятельно по предоставленной инструкции либо специалистами DataGo через менеджер тегов клиента при предоставлении необходимого доступа.

DataGo настраивает приём, обработку и доставку событий в базу данных клиента. После подключения события передаются автоматически. Клиент может проверять поступление и содержимое данных в своей базе и обращаться в службу поддержки при возникновении вопросов.

1. Подготовьте подключение к базе данных. Передайте DataGo адрес и порт базы, имя базы данных, параметры учётной записи для доставки и требования к защищённому соединению. Согласуйте с DataGo сетевой доступ, необходимые права записи и целевую таблицу. Для просмотра результата используйте учётную запись клиента с правом чтения этой таблицы.
2. Согласуйте источники и состав событий. Сообщите DataGo домены сайта или идентификатор приложения, перечень событий и передаваемые параметры. Получите полный адрес приёма событий с путём, идентификатор подключения, необходимые реквизиты, формат сообщений и имя таблицы результата. Адрес и настройки относятся к вашему подключению; не заменяйте их значениями из других примеров.
3. Настройте отправку событий. При самостоятельном подключении выполните настройку источника по инструкции DataGo с использованием предоставленного адреса приёма и параметров подключения. Настройку отправки событий с сайта также могут выполнить специалисты DataGo через менеджер тегов клиента при предоставлении необходимого доступа. В мобильном приложении отправка событий настраивается по соответствующей инструкции. Секрет из серверного HTTP-примера ниже используется только в серверной интеграции или проверочном терминале.
4. Следующие действия и примеры HTTP-отправки и SQL-запросов предназначены для дополнительной самостоятельной проверки. Клиент может выполнить их при необходимости либо обратиться за проверкой в DataGo.
5. Проверьте доставку. Выполните тестовые действия в источнике и найдите соответствующие записи в согласованной таблице базы данных. Обработка выполняется асинхронно: ответ приёмника подтверждает приём запроса, а наличие записей в базе — обработку и доставку. При последующей работе события передаются из подключённого источника автоматически при выполнении настроенных действий.

Проверочная отправка по HTTP

Этот пример предназначен для серверной отправки из приложения или терминала. Он не требует установки серверных компонентов DataGo и не заменяет настройку счётчика сайта или трекера приложения. Подставьте полученные от DataGo значения: полный адрес приёма без параметров запроса,

идентификатор, секрет и согласованный домен источника. Команды предназначены для Bash или совместимой оболочки.

```
COLLECT_URL='<полный адрес приёма>'
TRACKING_ID='<идентификатор подключения>'
API_SECRET='<секрет>'
SOURCE_DOMAIN='<домен источника>'
CLIENT_ID='manual-check-001'
REQUEST_URL="${COLLECT_URL}?measurement_id=${TRACKING_ID}"
REQUEST_URL="${REQUEST_URL}&api_secret=${API_SECRET}"
```

Выполните следующий блок в том же терминале. Он отправляет три отдельных запроса с одним client_id и разными именами событий. Для отправки только одного события оставьте в первой строке только page_view.

```
for EVENT_NAME in page_view add_to_cart purchase; do
  curl --fail --silent --show-error --request POST \
    "${REQUEST_URL}" \
    --header 'Content-Type: application/json' \
    --header "X-Forwarded-Host: ${SOURCE_DOMAIN}" \
    --header 'User-Agent: Mozilla/5.0 Chrome/120.0.0.0' \
    --data-binary @- <<JSON
{
  "client_id": "${CLIENT_ID}",
  "user_id": "test-user",
  "events": [{
    "name": "${EVENT_NAME}",
    "params": {
      "page_location": "https://${SOURCE_DOMAIN}/catalog",
      "test_marker": "${CLIENT_ID}"
    }
  }]
}
JSON
  printf '\n'
done
```

Это синтетические события для проверки соединения и доставки. В рабочем источнике состав и значения параметров определяются разметкой клиента. Перед повторной проверкой задайте новый CLIENT_ID, например manual-check-002, и снова выполните блок отправки; в SQL используйте то же новое значение.

Поле сообщения	**Назначение / результат**
client_id	Псевдонимный идентификатор посетителя или клиентского экземпляра, задаваемый источником; в результате — user_pseudo_id. Для проверки выбирайте новое тестовое значение.
user_id	Идентификатор пользователя, заданный источником; в результате — user_id.
events[].name	Название события; в результате — event_name.

Поле сообщения	Назначение / результат
events[].params	Параметры события; в результате — event_params.
X-Forwarded-Host	Согласованный домен источника для приведённого HTTP-примера.

После выполнения команд проверьте ответы на каждый из трёх запросов. Успешный ответ содержит message: Ok и requestID. Сохраните идентификаторы запросов для возможного обращения в поддержку. Ответ подтверждает приём запроса; результат обработки и доставки проверяется отдельно по наличию событий в базе данных.

Если команда завершилась ошибкой или истекло время ожидания ответа, сначала проверьте базу по тому же client_id: запрос мог успеть поступить в сервис. Повторная отправка может создать дополнительные события.

Просмотр результата в базе данных

Выполните запрос в SQL-клиенте, подключённом к своей базе данных ClickHouse. Вместо <БД> и <таблица> укажите согласованные с DataGo имена. Для приведённой отправки используйте client_id manual-check-001.

```
SELECT event_name, toUInt32(count()) AS hits
FROM `<БД>`.`<таблица>`
WHERE user_pseudo_id = 'manual-check-001'
GROUP BY event_name
ORDER BY event_name;
```

После доставки ожидаются три строки — по одной на каждое событие:

```
event_name   | hits
add_to_cart  | 1
page_view    | 1
purchase     | 1
```

Если отправляли только page_view, ожидается одна строка с hits = 1. Если записи ещё не появились, повторите SQL-запрос без повторной отправки. Фильтр по новому client_id отделяет текущую проверку от прежних событий.

Для просмотра содержимого событий выполните:

```
SELECT event_name, user_pseudo_id, user_id, event_params
FROM `<БД>`.`<таблица>`
WHERE user_pseudo_id = 'manual-check-001'
ORDER BY event_name;
```

Убедитесь, что имена событий совпадают с отправленными, user_pseudo_id равен client_id проверки, user_id равен test-user, а в event_params присутствуют

page_location и test_marker с переданными значениями. В рабочем сценарии клиент использует эти данные в своих запросах и аналитических инструментах.

Возможные ошибки и рекомендации по устранению

Ситуация	Действия клиента
Нет соединения с адресом приёма	Сверьте полный адрес с настройками, полученными от DataGo. Проверьте доступ к нему из источника и сетевые ограничения клиента. Если адрес недоступен, обратитесь в DataGo.
Запрос принят, но записей в базе нет	Повторите SQL-запрос с тем же client_id. Проверьте идентификатор, секрет, домен, формат сообщения и имя целевой таблицы. Передайте DataGo время отправки и requestID.
Нет доступа к базе данных или целевой таблице	Проверьте адрес, порт, параметры соединения и права своей учётной записи на чтение. Для проверки доставки согласуйте с DataGo доступ сервиса к базе и права записи.
Не хватает событий или параметров	Проверьте, вызывается ли отправка при нужном действии и передаются ли согласованные поля. Для сайта проверьте срабатывание счётчика, для приложения — вызов отправки. Сравните исходное сообщение с записью в базе.
Проверка показывает старые или повторные события	Используйте новый client_id для новой проверки и тот же идентификатор в SQL. Для ожидания результата повторяйте только SQL: повторная отправка создаёт новые события.

Ответ Ok при ошибочных реквизитах не является признаком успешной доставки. Если самостоятельная проверка не помогла, DataGo проверяет обработку и доставку на своей стороне; доступ клиента к серверам DataGo для этого не требуется.

Обращение в службу поддержки

Электронная почта: contact@datago.ru.

Время работы службы поддержки: по рабочим дням с 08:00 до 17:00 по московскому времени.

Технический контакт: Ренат Тимербулатов, r.timerbulatov@datago.ru.

В обращении укажите идентификатор подключения, источник, время отправки с часовым поясом, requestID при наличии, тестовый client_id, если выполнялась проверка, ожидаемый результат и фактическую ошибку. При наличии приложите обезличенный пример события и результат SQL-запроса. Секрет приёма и пароль базы в описание ошибки не включайте.