# Инструкция по устранению неисправностей в работе аналитической программы Биплан

(*версия от 21.05.2024*)

## Оглавление



## 1. Термины и определения

*Таблица 1 Термины и условные обозначения*

| Термин, обозначение | Описание |
|---|---|

| | |
|---|---|
| **БД** | База данных |
| **Браузер** | Веб-обозреватель, ПО на клиентском рабочем месте |
| **Группы доступа** | Группа пользователя с набором типовых прав |
| **Дашборд** | Аналитическая доска (закладка), в которой располагается отчет, набор отчетов, сформированный пользователем |
| **Заказчик** | Организация, заказавшая развертывание аналитической системы Биплан |
| **ПО** | Программное обеспечение |
| **Показатель** | Экономическое понятие той сущности, по которой строится диаграмма или один из видов отчётов |
| **Система** | Аналитическая система Биплан для анализа и визуализации данных |
| **Справочник** | Элементы, загруженные с названиями и кодом элемента справочников, а также с разными реквизитами |
| **СУБД** | Система управления базами данных |
| **ТЗ** | Техническое задание |
| **Фрейм** | Отдельный отчет на дашборде, выведенный в графической, текстовой, числовой форме или в виде таблицы |

# 2. Устранение неисправностей в работе серверной части аналитической Системы

## *2.1.Просмотр логов работы Системы*

Логи работы компонент Системы расположены по пути:

/home/[username]/biplane-api/storage/tenant[хэш тенанта]/logs/

Где [username] – имя пользователя, под которым работает biplane-api, а [хэш тенанта] – набор из 36 знаков (8 знаков – 4 знака – 4 знака – 4 знака – 12 знаков).

Они состоят из нескольких типов файлов:

| № п/п | Имя файла | Содержание лога |
|---|---|---|
| 1. | bitrix-{YYYY}-{MM}-{DD}.log | Протокол обмена данными с Битрикс24 |
| 2. | exportimport-{YYYY}-{MM}-{DD}.log | Протокол импорта/экспорта дампов баз данных |
| 3. | infos-{YYYY}-{MM}-{DD}.log | Создание новых пользователей |
| 4. | laravel-{YYYY}-{MM}-{DD}.log | Протокол работы Laravel |
| 5. | loader_data-{YYYY}-{MM}-{DD}.log | Протокол обмена по API с 1С |
| 6. | matview-{YYYY}-{MM}-{DD}.log | Создание и удаление матвью (представления) при обновлении параметров и аналитик |
| 7. | requests-{YYYY}-{MM}-{DD}.log | Содержит протокол запросов внутри системы (POST, GET) |

| | | |
|---|---|---|
| 8. | schedule-{YYYY}-{MM}-{DD}.log | Протокол работы планировщика заданий |
| 9. | startexchange-{YYYY}-{MM}-{DD}.log | Протокол диспетчера обмена (загрузки файлов) |

Имя файла лога формируются из его типа и текущей даты, каждый день при необходимости будет создан новый файл по данному шаблону.

## *2.2.Проверка наличия всех необходимых системных компонентов*

Обязательным этапом диагностики неисправностей является проверка наличия всех необходимых компонентов согласно системным требованиям по таблице ниже:

| № п/п | Параметр | Минимальное значение | Рекомендуемое значение |
|---|---|---|---|
| 1. | Операционная система | Windows 2008 Server<br>Linux Server с ядром 4.5+ | Windows 2012+ Server<br>Linux Server с ядром 5+ |
| 2. | Дополнительные компоненты | PostgreSQL 10<br>PHP 7.3<br>Node 16<br>Apache2.4 | PostgreSQL 11<br>PHP 7.3<br>Node 18<br>NGINX |

## *2.3. Проверка настройки учетной записи серверной части Системы*

Для корректной работы серверной части системы настройки прав доступа должны соответствовать инструкции по установке.

### *2.3.1. Проверка корректности настройки PHP*

Проверяем файл www.conf:

*sudo nano /etc/php/7.3/fpm/pool.d/www.conf*

Находим строчки ниже:

*user = [username] group =*

*[username] listen =*

*/run/php/php7.3-fpm.sock*

*listen.owner = [username]*

*listen.group = [username]*

*listen.mode = 0660*

Где [username] – должно соответствовать имени пользователя, под которым должен работать biplane-api.

### *2.3.2. Проверка настройки web-сервера*

Проверяем конфигурацию site:

*sudo nano /etc/nginx/sites-enabled/biplane-api-nginx.conf* Находим

строчку "root /home/[username]/biplane-api/public"

Где [username] – должно соответствовать имени пользователя, под которым должен работать biplane-api.

Проверяем конфигурацию веб-сервера NGINX: *sudo*
*nano /etc/nginx/nginx.conf*

Находим строчку "user [username]"

Где [username] – должно соответствовать имени пользователя, под которым должен работать biplane-api.

#### *2.3.3. Проверка настройки Supervisor*

Нужно чтобы один процесс scheduler.sh и несколько процессов (5) queue.sh запускались при старте системы под учётной записью [username].

В директории /etc/supervisor/conf.d находим queue-supervisor.conf и schedulersupervisor.conf и проверяем, что там указано [username] - имя пользователя, под которым должен работать biplane-api.

### *2.4. Решение проблем при работе с базой данных Биплан*

#### *2.4.1. Пересоздание БД*

Создаём базу данных, проводим миграции, устанавливаем ключи доступа.

Добавляем пользователя "test" с паролем "test" в Систему с правами администратора.

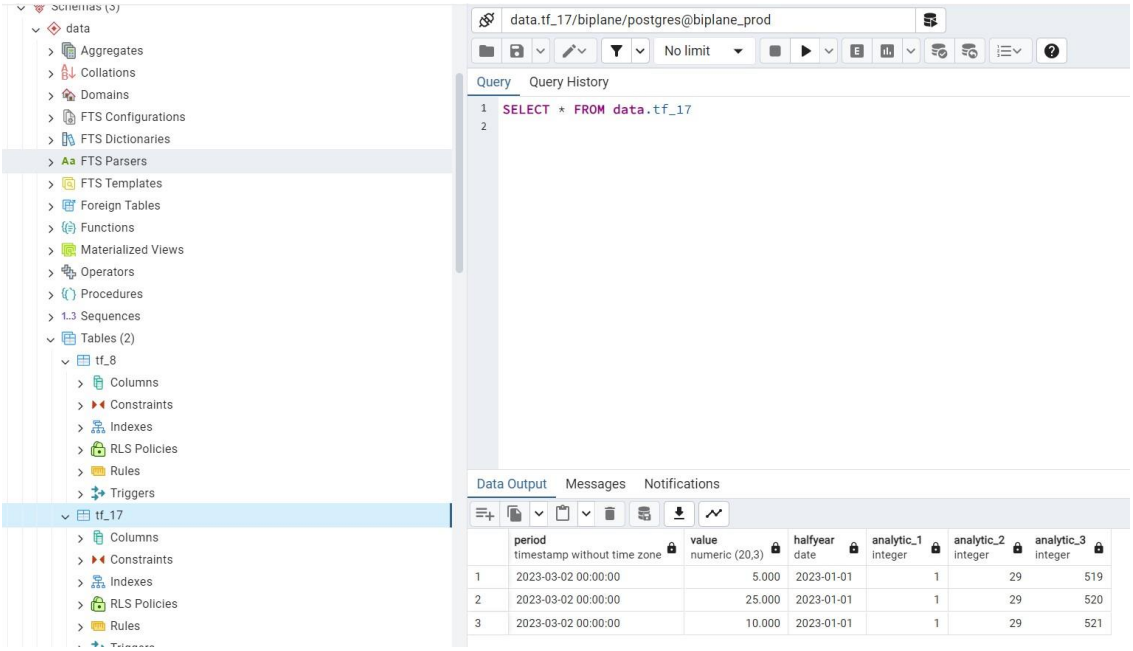
Для выполнения команд указанных ниже потребуются обычные права пользователя. Также нужно находится в директории /home/[username]/biplane-api:

Где [username] - имя пользователя системы.
*cd /home/[username]/biplane-api*
*./setbase.sh*
*./prerun.sh*

#### *2.4.2. Пересоздание связей загруженных данных со встроенными аналитиками*

При возникновении проблемы построения отчёта «что загружено в систему» по одному из показателей необходимо проверить значения по аналитикам, создаваемым стандартно самой программой, на присутствие в списке матвью базы. При таком расхождении решение проблемы следующее:

1. В PostgreSQL найти таблицу в schema data, содержащую данные показателей, вызывающих ошибку. Название таблицы формата tf_#. На примере – таблица tf_17.

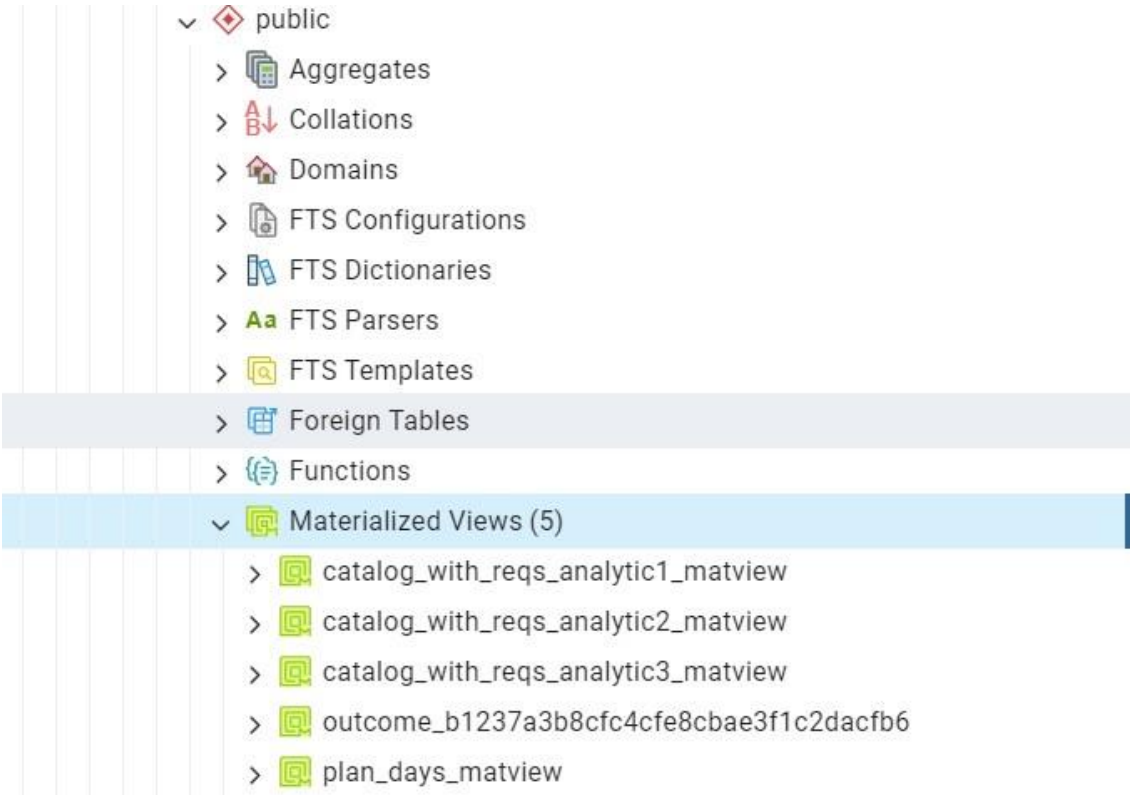



2. Удалить колонки по аналитикам, которых не существует. В данном случае колонки analytic_1, analytic_2, так как они указывали на матвью по дням недели и часам, которых в базе не было. Удаление произвести командой:

*ALTER TABLE <название таблицы> DROP COLUMN <название столбца>;*

Если удаление колонок не исправило ошибку, сделать следующие шаги:

3. Открыть структуру матвью в schema public. В матвью отсутствовали catalog_with_reqs_analytic1_matview и catalog_with_reqs_analytic2_matview (скриншот сделан после восстановления, номера матвью с днями недели и часами могут отличаться, см. какие столбцы были удалены на предыдущем шаге).



Обычно матвью по дням недели и часам создаются последними при первой загрузке данных в программу и имеют номера #количество_аналитик_в_файле_загрузки +1 и +2 соответственно.

4. Если матвью не существуют, то создать их следующими скриптами:

**Матвью для аналитики по часам:**

```
-- View: public.catalog_with_reqs_analytic1_matview
-- DROP MATERIALIZED VIEW IF EXISTS public.catalog_with_reqs_analytic1_matview;
CREATE MATERIALIZED VIEW IF NOT EXISTS public.catalog_with_reqs_analytic1_matview
TABLESPACE pg_default
AS
 WITH sorted_catalogs AS (
SELECT catalogs.id,
catalogs.parent_id,
catalogs.name,
catalogs.analytic_id,
        row_number() OVER (ORDER BY (catalogs.name COLLATE "ru-RU-x-icu")) AS idx_name,
row_number() OVER (ORDER BY (catalogs.name COLLATE "ru-RU-x-icu")) AS idx_name_desc
FROM catalogs
       WHERE catalogs.analytic_id = 1
      ), t AS (
       WITH RECURSIVE catalog_tree_ AS (
            SELECT  v1.id,
v1.parent_id,
v1.name,                2 AS
level,
v1.idx_name,
              ARRAY[v1.idx_name] AS idx_p_name,
              ARRAY[v1.idx_name_desc] AS idx_p_name_desc,
              ARRAY[v1.id::bigint] AS path_ids
             FROM sorted_catalogs v1
            WHERE v1.parent_id IS NULL OR (EXISTS ( SELECT
                    FROM catalogs c
                   WHERE c.id = v1.parent_id AND c.analytic_id <> 1))
UNION               SELECT v2.id,                v2.parent_id,
v2.name,
              ct.level + 1 AS level,
v2.idx_name,
              ct.idx_p_name || v2.idx_name AS idx_p_name,
              ct.idx_p_name_desc || v2.idx_name_desc AS idx_p_name_desc,
              ct.path_ids || v2.id::bigint AS path_ids
             FROM sorted_catalogs v2
               JOIN catalog_tree_ ct ON ct.id = v2.parent_id
            WHERE NOT (v2.id = ANY (ct.path_ids))
           )
      SELECT catalog_tree.id,
catalog_tree.name,          catalog_tree.level,
catalog_tree.idx_name,
catalog_tree.idx_p_name,
catalog_tree.idx_p_name_desc,
        catalog_tree.path_ids,
           CASE
              WHEN (EXISTS ( SELECT
                FROM catalogs t2
               WHERE catalog_tree.id = t2.parent_id AND t2.analytic_id = 1)) THEN true
              ELSE false
           END AS folder
       FROM catalog_tree_ catalog_tree
     )
```

```
 SELECT t.level,
    t.folder,
    t.id,
    t.name,
    count(*) OVER () AS total_rows,
t.idx_name,
    row_number() OVER (ORDER BY (
        CASE
            WHEN t.folder OR t.level > 2 THEN 1
            ELSE 2
        END), (
        CASE
            WHEN t.folder THEN t.idx_p_name
            ELSE t.idx_p_name[:t.level - 2] || (t.idx_p_name[t.level - 1] + '100000000000'::numeric::bigint)
        END)) AS idx_p_name,
row_number() OVER (ORDER BY (
        CASE
            WHEN t.folder OR t.level > 2 THEN 1
            ELSE 2
        END), (
        CASE
            WHEN t.folder THEN t.idx_p_name_desc
            ELSE t.idx_p_name_desc[:t.level - 2] || (t.idx_p_name_desc[t.level - 1] + '100000000000'::numeric::bigint)
END)) AS idx_p_name_desc,     t.path_ids
   FROM t
  ORDER BY (row_number() OVER (ORDER BY (
        CASE
            WHEN t.folder OR t.level > 2 THEN 1
            ELSE 2
        END), (
        CASE
            WHEN t.folder THEN t.idx_p_name
            ELSE t.idx_p_name[:t.level - 2] || (t.idx_p_name[t.level - 1] + '100000000000'::numeric::bigint)
END)))
WITH DATA;
ALTER TABLE IF EXISTS public.catalog_with_reqs_analytic1_matview
    OWNER TO postgres;
CREATE INDEX catalog_with_reqs_analytic1_matview_folder_idx
    ON public.catalog_with_reqs_analytic1_matview USING btree
    (folder)
    TABLESPACE pg_default;
CREATE INDEX catalog_with_reqs_analytic1_matview_idx_name_idx
    ON public.catalog_with_reqs_analytic1_matview USING btree
    (idx_name)
    TABLESPACE pg_default;
CREATE INDEX catalog_with_reqs_analytic1_matview_idx_p_name_desc_idx
    ON public.catalog_with_reqs_analytic1_matview USING btree
    (idx_p_name_desc)
    TABLESPACE pg_default;
CREATE INDEX catalog_with_reqs_analytic1_matview_idx_p_name_idx
    ON public.catalog_with_reqs_analytic1_matview USING btree
    (idx_p_name)
    TABLESPACE pg_default;
CREATE INDEX catalog_with_reqs_analytic1_matview_level_idx
    ON public.catalog_with_reqs_analytic1_matview USING btree
    (level)
    TABLESPACE pg_default;
CREATE INDEX catalog_with_reqs_analytic1_matview_path_ids_idx
    ON public.catalog_with_reqs_analytic1_matview USING btree
(path_ids)
    TABLESPACE pg_default;
```

**Матвью для аналитики по дням недели:**

```
-- View: public.catalog_with_reqs_analytic2_matview
-- DROP MATERIALIZED VIEW IF EXISTS public.catalog_with_reqs_analytic2_matview;
CREATE MATERIALIZED VIEW IF NOT EXISTS public.catalog_with_reqs_analytic2_matview TABLESPACE pg_default
AS
 WITH sorted_catalogs AS (
SELECT catalogs.id,
catalogs.parent_id,
catalogs.name,
catalogs.analytic_id,
        row_number() OVER (ORDER BY (catalogs.name COLLATE "ru-RU-x-icu")) AS idx_name,
row_number() OVER (ORDER BY (catalogs.name COLLATE "ru-RU-x-icu")) AS idx_name_desc
FROM catalogs
       WHERE catalogs.analytic_id = 2
      ), t AS (
       WITH RECURSIVE catalog_tree_ AS (
            SELECT  v1.id,
v1.parent_id,
v1.name,                2 AS
level,
               v1.idx_name,
               ARRAY[v1.idx_name] AS idx_p_name,
               ARRAY[v1.idx_name_desc] AS idx_p_name_desc,
               ARRAY[v1.id::bigint] AS path_ids
              FROM sorted_catalogs v1
             WHERE v1.parent_id IS NULL OR (EXISTS ( SELECT
                    FROM catalogs c
                   WHERE c.id = v1.parent_id AND c.analytic_id <> 2))
           UNION
SELECT v2.id,
v2.parent_id,
v2.name,
               ct.level + 1 AS level,
v2.idx_name,
               ct.idx_p_name || v2.idx_name AS idx_p_name,
               ct.idx_p_name_desc || v2.idx_name_desc AS idx_p_name_desc,
ct.path_ids || v2.id::bigint AS path_ids
              FROM sorted_catalogs v2
                JOIN catalog_tree_ ct ON ct.id = v2.parent_id
             WHERE NOT (v2.id = ANY (ct.path_ids))
           )
      SELECT catalog_tree.id,
catalog_tree.name,          catalog_tree.level,
catalog_tree.idx_name,
catalog_tree.idx_p_name,
catalog_tree.idx_p_name_desc,
catalog_tree.path_ids,
           CASE
              WHEN (EXISTS ( SELECT
                FROM catalogs t2
               WHERE catalog_tree.id = t2.parent_id AND t2.analytic_id = 2)) THEN true
              ELSE false
           END AS folder
        FROM catalog_tree_ catalog_tree
      )
 SELECT t.level,
   t.folder,
   t.id,
   t.name,    count(*) OVER () AS
total_rows,    t.idx_name,
row_number() OVER (ORDER BY (
```

```
        CASE
            WHEN t.folder OR t.level > 2 THEN 1
            ELSE 2
        END), (
        CASE
            WHEN t.folder THEN t.idx_p_name
            ELSE t.idx_p_name[:t.level - 2] || (t.idx_p_name[t.level - 1] + '100000000000'::numeric::bigint)
        END)) AS idx_p_name,
row_number() OVER (ORDER BY (
        CASE
            WHEN t.folder OR t.level > 2 THEN 1
            ELSE 2
        END), (
        CASE
            WHEN t.folder THEN t.idx_p_name_desc
            ELSE t.idx_p_name_desc[:t.level - 2] || (t.idx_p_name_desc[t.level - 1] + '100000000000'::numeric::bigint)
END)) AS idx_p_name_desc,     t.path_ids
   FROM t
  ORDER BY (row_number() OVER (ORDER BY (
        CASE
            WHEN t.folder OR t.level > 2 THEN 1
            ELSE 2
        END), (
        CASE
            WHEN t.folder THEN t.idx_p_name
            ELSE t.idx_p_name[:t.level - 2] || (t.idx_p_name[t.level - 1] + '100000000000'::numeric::bigint)
END)))
WITH DATA;
ALTER TABLE IF EXISTS public.catalog_with_reqs_analytic2_matview
    OWNER TO postgres;
CREATE INDEX catalog_with_reqs_analytic2_matview_folder_idx
    ON public.catalog_with_reqs_analytic2_matview USING btree
    (folder)
    TABLESPACE pg_default;
CREATE INDEX catalog_with_reqs_analytic2_matview_idx_name_idx
    ON public.catalog_with_reqs_analytic2_matview USING btree
    (idx_name)
    TABLESPACE pg_default;
CREATE INDEX catalog_with_reqs_analytic2_matview_idx_p_name_desc_idx
    ON public.catalog_with_reqs_analytic2_matview USING btree
    (idx_p_name_desc)
    TABLESPACE pg_default;
CREATE INDEX catalog_with_reqs_analytic2_matview_idx_p_name_idx
    ON public.catalog_with_reqs_analytic2_matview USING btree
    (idx_p_name)
    TABLESPACE pg_default;
CREATE INDEX catalog_with_reqs_analytic2_matview_level_idx
    ON public.catalog_with_reqs_analytic2_matview USING btree
    (level)
    TABLESPACE pg_default;
CREATE INDEX catalog_with_reqs_analytic2_matview_path_ids_idx
    ON public.catalog_with_reqs_analytic2_matview USING btree
(path_ids)
    TABLESPACE pg_default;
```

При необходимости заменить номера таблиц для матвью (свериться с id аналитики по таблице public.analytics)

5. Если создание матвью не исправило ошибку – открыть данные из таблицы public.indicators. Удалить данные по показателю, который вызывает ошибку, либо полностью очистить таблицу (если проблема с несколькими показателями)

# 3. Решение проблем с работой клиентской части Системы

## *3.1. Ошибка «Ошибка загрузки данных: возможно у Вас где-то еще открыта сессия!»*

Данная ошибка связана с попыткой параллельной работы с двух рабочих мест одним пользователем или незавершенной сессией работы пользователя. Возникает обычно, если пользователь не завершает сессию при работе с Системой, а просто закрывает браузер и позже открывает его с ранее открытыми страницами.

Обходным решением является принудительное завершение сессии пользователя. Данный способ доступен самому пользователю системы. Для этого сделайте несколько шагов:

1. Закройте окно с ошибкой и очистите кэш браузера.
2. Откройте окно браузера в режиме «Инкогнито».
3. Зайдите на страницу авторизации в Системе.
4. Авторизуйтесь в Системе и корректно завершите сессию с помощью пункта меню «Покинуть аккаунт».
5. Закройте окно в режиме «Инкогнито».
6. Снова зайдите в Систему в обычном режиме.

## *3.2. Ошибка «Ошибка получения данных: Время жизни токена истекло. Необходимо перезайти в программу!»*

Данная ошибка связана с истечением сессии работы пользователя. Возникает обычно, если пользователь не завершает сессию при работе с Системой, а позже продолжает работу с ранее открытыми страницами.

Обходным решением является принудительный перезапуск сессии пользователя. Данный способ доступен самому пользователю системы. Для этого сделайте несколько шагов:

7. Закройте окно с ошибкой и очистите кэш браузера.
8. Откройте окно браузера в режиме «Инкогнито».
9. Зайдите на страницу авторизации в Системе.

10. Авторизуйтесь в Системе и корректно завершите сессию с помощью пункта меню «Покинуть аккаунт».
11. Закройте окно в режиме «Инкогнито».
12. Снова зайдите в Систему в обычном режиме.