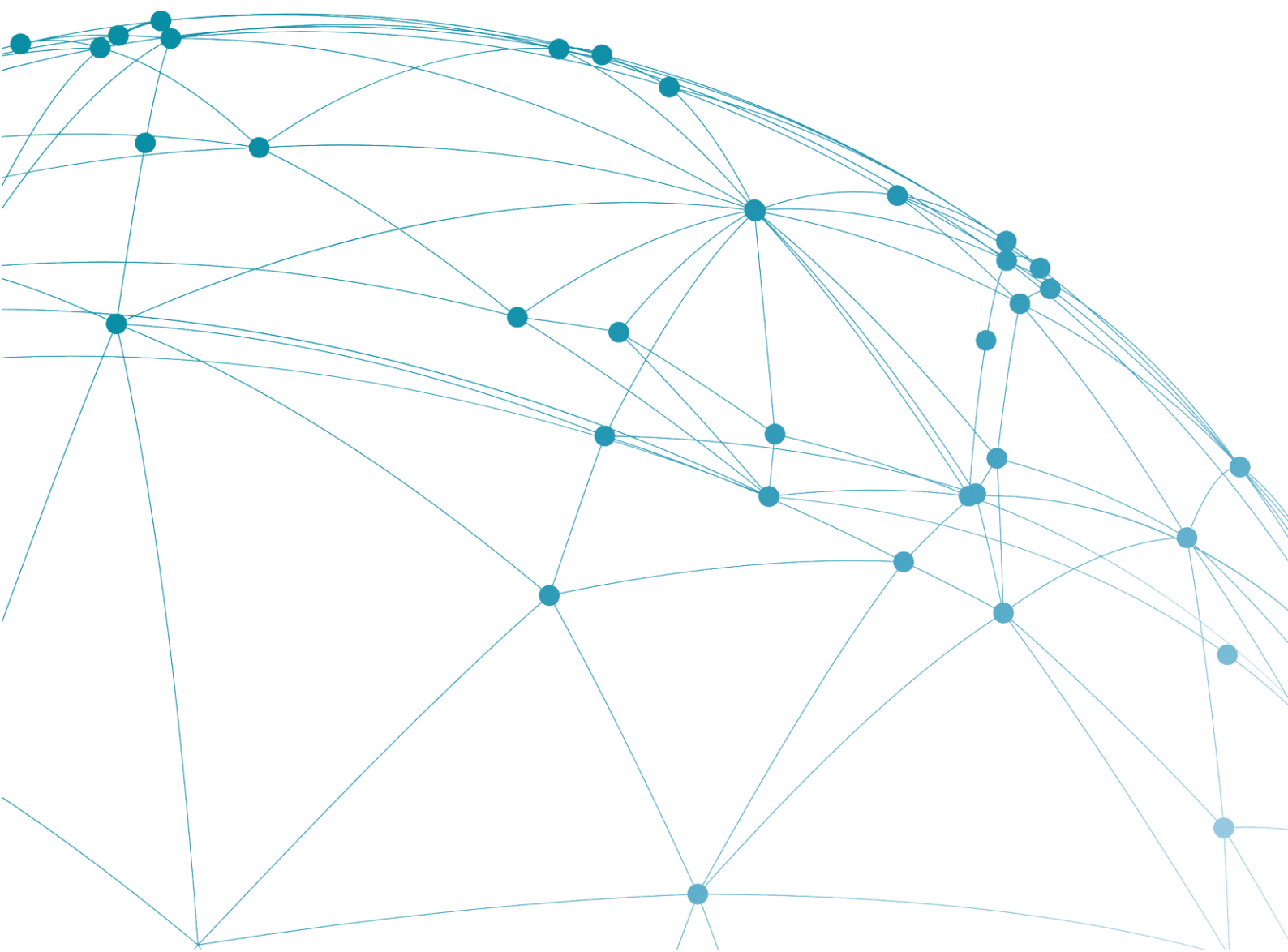

Цифровая платформа
«Управление в пространстве» (ЦП УВП)

версия 0.43

РУКОВОДСТВО АДМИНИСТРАТОРА



Содержание

1	Перечень терминов, сокращений и обозначений	5
2	Назначение и основные возможности	6
2.1	Ключевые особенности	6
2.2	Возможности платформы	6
2.3	Режимы работы	7
3	Архитектура платформы	8
3.1	Внешние зависимости и библиотеки	10
3.2	База данных приложения (репозиторий)	10
3.3	Лицензии и сертификаты	10
4	Установка платформы	11
4.1	Системные требования	11
4.2	SSL-сертификаты	12
4.3	Завершение установки	14
4.4	Установка с использованием docker compose	14
4.5	Переменные окружения	19
4.6	Песочница и удаленный запуск расчетов	19
4.7	Прокси-сервер для клиентского приложения	22
5	Аутентификация и авторизация	30
5.1	Способы аутентификации	30
5.2	Авторизация с помощью Kerberos	30
5.3	Настройка веб-сервера для Kerberos-аутентификации	31
5.4	Настройка клиентов для Kerberos-аутентификации	39
5.5	Авторизация с помощью LDAP	42
5.6	Анонимные (гостевые) пользователи	44
5.7	Работа с паролями пользователей	44
5.8	Токен	45
5.9	Внешняя авторизация (переход при выходе из приложения)	46
6	Разграничение прав доступа	47
6.1	Пользователи	47
6.2	Группы	49
6.3	Роли	51
6.4	Профиль пользователя	53
6.5	Просмотр выданного доступа к объектам	54
6.6	Выход из приложения	55
6.7	Активные пользователи	55
7	Настройка интерфейса	57
7.1	Стили и темы	57

7.2	Стили меню.....	57
7.3	Темы	63
7.4	Редактор кода	64
7.5	Скрипты	66
8	Взаимодействие с внешними системами	68
8.1	Работа с приложением через IFrame	68
9	Настройка расчетов	69
9.1	Получение пользовательского токена для расчетов.....	69
9.2	Удаление токена.....	69
9.3	Запуск расчета через API.....	69
9.4	Получение результатов расчета блока через API	71
9.5	Запуск расчетов на нескольких узлах	75
9.6	Режимы работы	75
9.7	Установка.....	75
9.8	Работа с RabbitMQ	76
9.9	Текущие расчеты.....	78
10	Лицензирование.....	80
10.1	Ввод лицензии через визуальный интерфейс.....	80
11	Администрирование.....	83
11.1	Оповещение администратора	83
11.2	Протокол доступа.....	83
11.3	Сборка мусора	84
11.4	Сообщение об ошибках	85
11.5	Настройки платформы.....	88
11.6	Управление доступом	88
11.7	Технические особенности платформы.....	94
11.8	Проекция графа.....	96
11.9	Управление пользователями	97
11.10	Управление ролями.....	106
11.11	Управление соединениями с БД.....	108
11.12	Мониторинг	119
11.13	Дополнительные сервисы	121
11.14	Время в приложении.....	128
11.15	Конфигурация приложения.....	128
11.16	Конфигурация клиентского приложения.....	130
11.17	Управление глобальными константами.....	131
11.18	Управление глобальными паролями	134
11.19	Управление файлами	137

11.20 Управление стилями карты	142
Приложение 1. Внешние зависимости и библиотеки	145
Приложение 2. Структура базы данных платформы	146
Скрипты создания репозитория	146
Структура таблиц репозитория	148
Схема базы данных (репозитория)	165
Функции и процедуры репозитория	166
Приложение 3. Метрики Prometheus по умолчанию	167

1 Перечень терминов, сокращений и обозначений

Сокращение	Определение
БД	База данных
НСИ	Нормативно-справочная информация
Платформа, ЦП УВП	Цифровая платформа «Управление в пространстве»
API	Application Programming Interface – программный интерфейс приложения
FIFO	англ. First In, First Out – принцип очередности «первый пришёл – первый ушёл»
LDAP	англ. Lightweight Directory Access Protocol – протокол доступа к службе каталогов
OLAP	англ. Online analytical processing (интерактивная аналитическая обработка) – технология обработки данных, заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу
SPN	англ. Service Principal Names – уникальный идентификатор экземпляра сервиса
SQL	англ. Structured Query Language (язык структурированных запросов) – декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных
SSL	англ. Secure Sockets Layer – протокол безопасности, создающий зашифрованное соединение между веб-сервером и веб-браузером
SSL-сертификат	Цифровой сертификат, удостоверяющий подлинность веб-сайта и позволяющий использовать зашифрованное соединение
SSO	англ. Single Sign-On – технология единого входа, при использовании которой пользователь переходит из одного раздела портала в другой, либо из одной системы в другую, не связанную с первой системой, без повторной аутентификации

2 Назначение и основные возможности

Цифровая платформа «Управление в пространстве» – платформа, автоматизирующая процесс загрузки, анализа и интерпретации данных на основе методов математического моделирования.

Российская аналитическая low-code BI-платформа «Управление в пространстве» со встроенным искусственным интеллектом и геоаналитикой, которые используются для поддержки принятия управленческих решений. Используется программное обеспечение с открытым исходным кодом (не используются права третьих лиц, одним из основных преимуществ является свобода использования, модифицирования программы), полностью собственная разработка компании [БИТ](#).

Включена в Единый реестр российских программ для ЭВМ и БД под рег. номером 7516. Может быть использована для импортозамещения таких продуктов как Power BI, Tableau, Qlik Sense, SPSS Statistics, ArcGIS Business Analyst.

BI-платформа «Управление в пространстве» в сравнении с другими BI-платформами на российском рынке имеет расширенные функциональные возможности в части решения задач 3D-картографического мониторинга, построения цифровых двойников, моделей и расчетов в пространстве, динамического моделирования, сценарного и целевого прогнозирования и формирования рекомендаций, low-code и визуального конструирования аналитических решений.

2.1 Ключевые особенности

- Платформа реализует комплексную аналитическую поддержку принятия решений с использованием технологий продвинутой аналитики на основе математического моделирования, геоаналитики, искусственного интеллекта и больших данных.
- Платформа обеспечивает визуальное конструирование аналитических приложений.

2.2 Возможности платформы

- Базовая аналитика:
 - сбор, загрузка и преобразование данных;
 - визуализация данных с использованием электронной картографии, таблиц и диаграмм, аналитических панелей.
- Продвинутая аналитика:
 - геоаналитика (продвинутая обработка пространственных данных, 3D-картография, геоинформационный мониторинг, пространственный анализ и пространственное моделирование);
 - математическое моделирование с использованием математической статистики и методов машинного обучения (ML);
 - сценарное многовариантное прогнозирование, оптимизационное и целевое управление;
 - предписывающая аналитика на основе генерации рекомендаций на естественном языке с использованием продукционного подхода.

2.3 Режимы работы

Цифровая платформа поддерживает два режима работы:

- конструкторский;
- пользовательский.

Конструкторский режим (Граф) – режим, который позволяет размещать в приложении блоки, связи, произвольные элементы (текст, изображение, видео), слайды и совершать операции над ними (копирование, удаление, перемещение).

Конструкторский режим предназначен для решения задач:

- конструирования бизнес-логики приложения для последующего использования в Пользовательском режиме предустановленных вариантов отображения информации;
- загрузки первичных данных средствами импорта.

Пользовательский режим (Пользовательское приложение) – режим при котором осуществляется визуализация данных в виде:

- картографической информации;
- дашбордов – Dashboard, аналитическая (контрольная, приборная) панель – общепринятое средство визуализации данных в современных BI-решениях.

Пользовательский режим предназначен для решения функциональных задач:

- мониторинг и анализ данных;
- прогнозирование и моделирование.

3 Архитектура платформы

Архитектура является микросервисной, с возможностью вынесения расчетчиков на дополнительный уровень с целью горизонтального масштабирования. При этом обработка всех запросов пользователей выполняется через серверное приложение **координатор** (Coordinator).

При необходимости можно нарастить количество координаторов для горизонтального масштабирования. Для этого необходимо расширить инфраструктуру с помощью дополнительных инструментов, позволяющих обеспечить автоматическую балансировку и отказоустойчивость.

Для работы нескольких экземпляров координатора или для общения между **расчетчиками** (Worker) необходима общая разделяемая область памяти. Так же Redis обеспечивает хранение in-memory объектов и их кэша для быстрого доступа при расчетах.

Внешние, по отношению к Координатору, расчетчики проводят расчеты задач для нескольких пользователей. Работа с ними организована через очередь сообщений RabbitMQ, которая обеспечивает отправку заявок на расчет из координатора, когда приложению необходимо его произвести, а также реализует сам механизм очереди в случае, если все расчетчики заняты.

Справочники обслуживаются в отдельном сервисе, который отвечает за полный цикл их существования.

Изолированное окружение (Docker) позволяет безопасно выполнять пользовательские скрипты на языке Python. При этом возможен запуск и на Воркерах, которые обслуживают расчеты

Архитектура платформы приведена на рисунке 1.

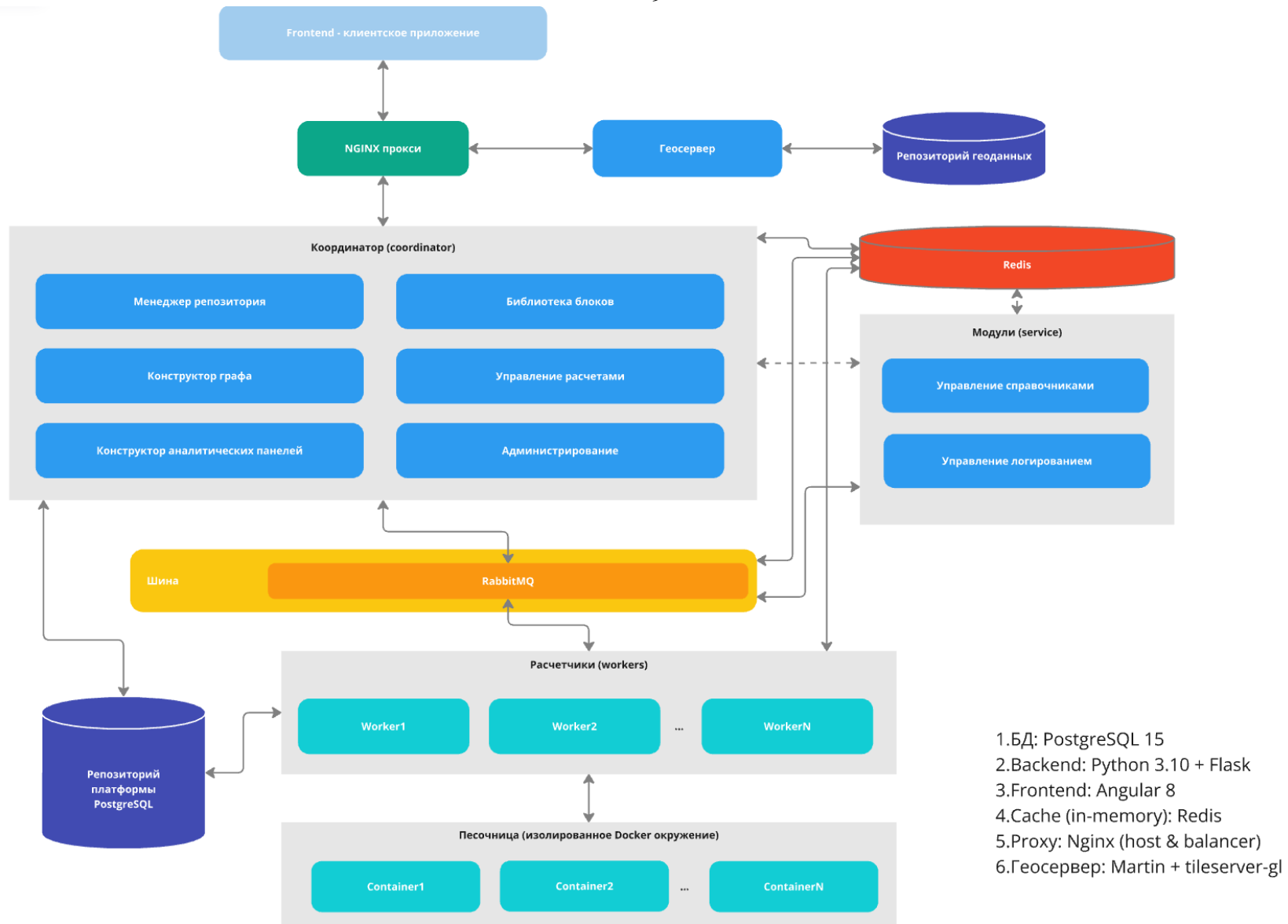


Рисунок 1 – Архитектура цифровой платформы «Управление в пространстве»

3.1 Внешние зависимости и библиотеки

Платформа разрабатывается с использованием программного обеспечения с открытым исходным кодом и с бесплатными лицензиями (FOSS – Free and Open Source Software).

Перечень и описание компонент веб-приложения и серверного приложения платформы, которые включаются в docker-образ, передаваемый для установки, приведены в Приложении 1.

3.2 База данных приложения (репозиторий)

Скрипты создания базы данных приложения (репозитория), описание структуры БД, функций и процедур БД приведены в Приложении 2.

3.3 Лицензии и сертификаты

- Цифровая платформа внесена в Единый реестр российских программ для ЭВМ и БД приказом Министерства цифрового развития, связи и массовых коммуникаций РФ от 30.11.2020 №634.
- Запись в Едином реестре российских программ для ЭВМ и БД №7516 от 30.11.2020: https://reestr.digital.gov.ru/reestr/308866/?sphrase_id=670579.
- Запись о регистрации платформы в Федеральной службе по интеллектуальной собственности (ФИПС) под номером №2020614741.

4 Установка платформы

4.1 Системные требования

4.1.1 Аппаратная часть

4.1.1.1 Сервер

Важно! Конфигурации могут меняться в зависимости от предполагаемой инфраструктуры.

- Процессор: 8 ядер, 2.30 ГГц и выше;
- Оперативная память: 8 ГБ и выше;
- Жесткий диск: от 20 Гб. *Размер определяется объемом базы данных;*
- Сетевой интерфейс: 1 Гбит/сек.

Для каждой установки необходимо определять объемы оперативной памяти / ядер / жесткого диска исходя из количества компонент, которые будут установлены на сервер, а также количества пользователей.

Указанные характеристики подходят для установки приложения и кэширующей БД для **25 пользователей**. Если необходимо дополнительно разместить на этом же сервере:

- Сервис расчетов R: ОЗУ + 2 Гб (определяется количеством расчетов), жесткий диск + 5 Гб;
- Сервис подготовки геоданных Martin: ОЗУ + 4 Гб, жесткий диск + 5 Гб (определяется объемами картографических данных);
- БД Postgres: ОЗУ +4 Гб, жесткий диск +10 Гб (определяется объемами данных).

4.1.1.2 Рабочая станция

- Процессор: 4 ядра, 2.40 ГГц и выше (Intel Core i3 6300 или аналогичный);
- Оперативная память: 4 ГБ и выше;
- Сетевой интерфейс: от 20 Мбит/сек.
- Рекомендуется дискретная видео карта с поддержкой технологии WebGL 2.0 для быстрой работы картографического визуализатора.

Дополнительно:

- Монитор 24" и более (разрешение 2560x1440) для оптимального отображения визуализаторов;
- Дискретная видеокарта для работы с большим количеством картографических визуализаторов.

4.1.2 Программная часть

4.1.2.1 Сервер

- Операционная система linux (Ubuntu, AstraLinux – debian 7+). Рекомендуется Ubuntu 20 Server LTS. В случае Astra-Linux рекомендуется версия 1.7+;
- Операционная система linux (CentOS, Oracle Linux, РЕД ОС – redhat 7.9+). Рекомендуется CentOS 7;
- БД приложения: PostgreSQL 15 + PostGIS 3.2;
- Кэширующая БД: Redis 6.0.7;

- docker + docker-compose – в случае работы в контейнерах;
- python 3.10, nginx 1.23.1 – в случае работы без контейнеров.

Опционально:

- Сервис подготовки геоданных Martin (<https://github.com/maplibre/martin>);
- Сервис расчетов R (<https://www.r-project.org/>, https://hub.docker.com/_/r-base).

4.1.2.2 Рабочая станция

- Веб-браузер (лучше использовать последнюю доступную версию):
 - Google Chrome 90+;
 - Mozilla Firefox 90+;
 - Edge 101+;
 - Яндекс.Браузер 21.6.3.

4.2 SSL-сертификаты

Для того чтобы платформа могла работать с использованием HTTPS-соединения, необходимо в прокси-сервере, на котором размещено клиентское приложение (frontend), добавить сертификаты. Для этого необходимо выписать SSL-сертификаты на сервер, на котором будет размещаться приложение.

4.2.1 Получение сертификатов

Получение сертификатов в центре сертификации остается вне этого руководства. Для получения самоподписанного сертификата для сервера можно использовать следующий скрипт команд (для сервера Ubuntu):

generate.sh

```
#!/bin/sh

if [ "$#" -ne 1 ]
then
    echo "Must supply a domain"
    return 1
fi

DOMAIN=$1

openssl genrsa -des3 -out ca.key 2048
openssl req -x509 -new -nodes -key ca.key -sha256 -days 1825 -out ca.pem

openssl genrsa -out $DOMAIN.key 2048
openssl req -new -key $DOMAIN.key -out $DOMAIN.csr

cat > $DOMAIN.ext << EOF
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names
[alt_names]
DNS.1 = $DOMAIN
EOF
```

```
openssl x509 -req -in $DOMAIN.csr -CA ca.pem -CAkey ca.key -CAcreateserial \
-out $DOMAIN.crt -days 825 -sha256 -extfile $DOMAIN.ext
```

Для вызова скрипта необходимо указать имя домена, для которого будет выпущен сертификат:

```
$ . generate.sh domain.name
```

Во время работы скрипта необходимо указать некоторые параметры для выпуска сертификатов.

После этого необходимо забрать полученные файлы:

```
domain.name.crt
domain.name.key
```

4.2.2 Запуск приложения в docker или docker-compose с использованием SSL-сертификатов

Для работы приложения необходимо удостовериться, что версия платформы была собрана с возможностью подключения SSL-сертификатов. Для этого необходимо проверить в контейнере с клиентским приложением (frontend) конфигурационный файл и убедиться, что в нем указаны сертификаты для работы по https.

Заходим в контейнер:

```
$ docker exec -it vismind-frontend /bin/sh
```

Проверяем содержимое файла конфигурации **nginx**:

```
$ cat /etc/nginx/conf.d/default.conf | grep "443 http2 ssl"
```

Если команда возвращает строку вида:

```
listen          443 http2 ssl;
```

значит, клиент настроен, и можно переходить к шагу замены сертификатов.

Если ответ пустой или строка начинается с символа # – необходимо обратиться к разработчику для получения дистрибутива с активированной функцией SSL-соединения.

4.2.2.1 Замена сертификатов

Если приложение настроено на работу с SSL, необходимо загрузить корректные сертификаты, выпущенные для сервера:

```
domain.name.crt
domain.name.key
```

Для этого необходимо в файле **docker-compose.yml** передать внутрь контейнера корректные файлы и пробросить наружу 443 порт. Проброс портов предназначен для предоставления внешним пользователям доступа к некоторым внутренним службам, используя единственный внешний IP-адрес.

Пример конфигурации:

```
vismind-frontend:
  container_name: vismind-frontend
  image: vismind-frontend:latest
  ports:
    - 8002:443
  volumes:
    - ./files/static:/usr/share/nginx/media/static
    - ./domain.name.crt:/etc/ssl/certs/domain.crt:ro;
    - ./domain.name.key:/etc/ssl/private/domain.key:ro;
```

```
depends_on:
  - vismind-backend
networks:
  - vismind-nt
restart: unless-stopped
```

После чего необходимо перезапустить приложение. После перезапуска приложение будет доступно на 443 порту по протоколу HTTPS.

4.3 Завершение установки

После завершения установки необходимо проверить, что приложение запустилось по выбранному адресу:

- <http://localhost:8001> – если не изменялись порт и не используется SSL;
- <https://localhost:8002> – если во время установки были добавлены SSL-сертификаты.

При использовании «чистой» установки по умолчанию настроена учетная запись:

```
login: admin
password: admin
```

После входа в приложение можно изменить пароль для Администратора и создать новых пользователей.

4.4 Установка с использованием docker compose

4.4.1 Поставка

Платформа поставляется в виде tar-архива, который содержит все необходимые компоненты для начала работы:

1. Docker-образы платформы.
2. Бэкап БД репозитория.
3. Файл лицензии.
4. Скрипты установки и запуска платформы.

Важно! В данной инструкции не рассматривается установка сервиса подготовки геоданных сервера **Martin** и его прокси-сервера **nginx proxy**, а также установка сервиса расчетов **R**.

4.4.2 Системные требования

Перед началом работы необходимо установить на сервер:

1. docker – [Инструкция по установке Docker](#).
2. docker compose – с июля 2023 считается устаревшим и является частью Docker. Если версия Docker установлена без него – можно установить по старому – [Инструкция по установке docker compose](#)

На сервере должен быть доступ в Интернет для загрузки образов PostgreSQL и Redis. Если доступа до docker hub на сервере нет – необходимо загрузить эти образы самостоятельно.

1. PostgreSQL – **postgis/postgis:15-3.3-alpine**
2. Redis – **redis:6.0.7-alpine**

4.4.3 Установка

Установка платформы использует Docker для запуска необходимых компонентов в контейнерах с помощью **docker compose**. Сам процесс установки автоматический и запускает платформу с использованием внутренней БД PostgreSQL 15.

Для установки необходимо выполнить команды:

```
$ tar -xvzf uvp_fortum.tar.gz
$ cd fortum-customer
$ sudo chmod +x install.sh
$ . install.sh
```

По окончании установки платформа будет доступна по адресу <http://localhost:8001>

4.4.4 Запуск и остановка

Для запуска приложения выполнить скрипт:

```
$ . start_application.sh
```

Для остановки приложения выполнить скрипт:

```
$ . stop_application.sh
```

4.4.5 Начало работы

Для начала работы необходимо открыть страницу авторизации по адресу <http://localhost:8001>.

Логин по умолчанию: admin

Пароль по умолчанию: admin

4.4.6 Конфигурация приложения

По умолчанию платформа запускается с использованием соединения с базой данных репозитория, которая создавалась во время установки. Она расположена в контейнере vismind-postgresql.

Вся внутренняя конфигурация серверной части платформы может быть изменена с помощью переменных окружения. Количество доступных переменных больше, чем представлено ниже. Ниже представлены переменные, нужные для настройки подключения к базе данных приложения и настройки аутентификации.

Переменные окружения задаются для контейнера **vismind-backend** в файле `./distr/docker-compose.yaml`:

```
...
vismind-backend:
  container_name: vismind-backend
  image: vismind-backend:0.36-fortum
  environment:
    # repository
    - VISMIND_REPOSITORY_HOST=vismind-postgresql
    - VISMIND_REPOSITORY_PORT=5432
    - VISMIND_REPOSITORY_NAME=repo_vismind
    - VISMIND_REPOSITORY_USER_LOGIN=
    - VISMIND_REPOSITORY_USER_PASSWORD=
    # redis
    - VISMIND_REDIS_HOST=vismind-redis
    - VISMIND_REDIS_PORT=6379
```

```

- VISMIND_REDIS_DB=1
- VISMIND_REDIS_PASSWORD=
# LDAP
- VISMIND_LDAP_ALLOW_LOGIN=1
- VISMIND_LDAP_HOST=
- VISMIND_LDAP_PORT=389
- VISMIND_LDAP_USE_SSL=0
- VISMIND_LDAP_GET_INFO=ALL
- VISMIND_LDAP_CLIENT_STRATEGY=SYNC
- VISMIND_LDAP_AUTHENTICATION=NTLM
- VISMIND_LDAP_USER_HOST=int.bittechno.ru
- VISMIND_LDAP_DOMAIN_COMPONENT=
- VISMIND_LDAP_USERS_PARTITION=
- VISMIND_LDAP_USERNAME=sAMAccountName
- VISMIND_LDAP_GROUPS=memberOf
- VISMIND_LDAP_FIRST_NAME=givenName
- VISMIND_LDAP_LAST_NAME=sn
-
- VISMIND_LDAP_EMAIL=mail
- VISMIND_LDAP_DEFAULT_GROUP=Администраторы
- VISMIND_AUTH_DEFAULT_GROUP=Администраторы
- VISMIND_LDAP_SYNC_GROUPS_WITH_AD=0
# kerberos
- VISMIND_KERBEROS_ALLOW_LOGIN=1
- VISMIND_KERBEROS_SYSTEM_USER_LOGIN=
- VISMIND_KERBEROS_SYSTEM_USER_PASSWORD=
volumes:
- ./files:/app/files
- ./kerberos_certs/krb5.conf:/etc/krb5.conf:ro
- ./kerberos_certs/krb5.keytab:/etc/krb5.keytab:ro
depends_on:
- vismind-redis
- vismind-postgresql
networks:
- vismind-nt
restart: unless-stopped
...

```

Если необходимо использовать внешнюю БД Redis или PostgreSQL – необходимо указать подключение к ним в переменных окружения.

Авторизация с помощью доменных учетных записей и сквозная авторизация с помощью доменных учетных записей по умолчанию включены на клиенте, но не будут работать, пока не будут включены в конфигурации серверной части. Для включения необходимо заменить конфигурацию в файле `./distr/docker-compose.yaml` и перезагрузить приложение:

```
$ docker compose -f ./distr/docker-compose.yaml down && docker compose -f
./distr/docker-compose.yaml up -d
```

4.4.6.1 Описание параметров конфигурации

Все переменные окружения, задающие конфигурацию платформы, начинаются с префикса `VISMIND_`. Затем следует имя переменной из конфигурационного файла приложения. Если значение конфигурации не задать переменной окружения – будут использоваться значения по умолчанию.

Значения по умолчанию не заданы для соединений с внешними службами (Postgres, Redis, Ldap).

4.4.6.1.1 PostgreSQL 15 repository

Определяет параметры подключения к БД репозитория приложения.

```
VISMIND_REPOSITORY_HOST      # имя сервера PostgreSQL
VISMIND_REPOSITORY_PORT      # порт, по умолчанию 5432
VISMIND_REPOSITORY_NAME      # имя БД репозитория платформы
VISMIND_REPOSITORY_USER_LOGIN # логин пользователя для доступа к репозиторию
VISMIND_REPOSITORY_USER_PASSWORD # пароль пользователя для доступа к репозиторию
```

База данных **не создается в процессе установки**. Ее развертывание происходит из файла **empty_repository.backup**. При необходимости переноса на другой сервер можно свернуть БД из контейнера **vismind-postgresql** и перенести ее на другой сервер. После этого можно будет изменять параметры соединения и перезапускать приложение.

4.4.6.1.2 Redis in-memory database

Определяет параметры подключения к Redis для хранения состояний и in-memory расчетов приложения.

```
VISMIND_REDIS_HOST      # имя сервера Redis
VISMIND_REDIS_PORT      # порт, по умолчанию 6379
VISMIND_REDIS_DB        # номер бд в которой будут храниться данные
VISMIND_REDIS_PASSWORD # пароль для доступа к Redis
```

БД Redis не хранит свое состояние, поэтому все временные данные, хранящиеся в ней, будут удалены при остановке контейнера.

4.4.6.1.3 LDAP Active Directory

Параметры подключения к серверам Active Directory для авторизации с использованием доменных учетных записей:

```
VISMIND_LDAP_ALLOW_LOGIN      # Использовать LDAP
VISMIND_LDAP_HOST              # Адрес сервера LDAP. Можно указывать
# множество хостов через разделитель <;> (точка с запятой)
VISMIND_LDAP_PORT              # Порт сервера LDAP
VISMIND_LDAP_USE_SSL           # Использовать ssl
VISMIND_LDAP_GET_INFO          # Чтение информации о сервере,
# возможные значения:
#   NONE
#   DSA
#   SCHEMA
#   ALL

VISMIND_LDAP_CLIENT_STRATEGY   # Стратегии соединения,
# возможные значения:
#   SYNC
#   ASYNC
#   LDIF
#   RESTARTABLE
#   REUSABLE
#   SAFE_SYNC
#   SAFE_RESTARTABLE

VISMIND_LDAP_AUTHENTICATION    # Аутентификация, возможные значения:
#   ANONYMOUS
#   SIMPLE
#   SASL
```

```

# NTLM

VISMIND_LDAP_USER_HOST # Наименование домена (обязательный)
VISMIND_LDAP_DOMAIN_COMPONENT # Домен, в котором записан пользователь.
Пример: dc=int,dc=bittechno,dc=ru
VISMIND_LDAP_USERS_PARTITION # Адрес до раздела с пользователями.
Пример: ou=Service Accounts,ou=Enterprise
# (если оставить пустой, то поиск будет
происходить по всему домену)

# Атрибуты LDAP для использования при
создании пользователя в платформе
VISMIND_LDAP_USERNAME # Логин пользователя
VISMIND_LDAP_GROUPS # Группы, которые надо проверить для
добавления в группы
VISMIND_LDAP_FIRST_NAME # Имя пользователя
VISMIND_LDAP_LAST_NAME # Фамилия пользователя
VISMIND_LDAP_EMAIL # Почта пользователя
VISMIND_LDAP_DEFAULT_GROUP=Администраторы # группа, в которую автоматически
попадет новый пользователь LDAP
VISMIND_AUTH_DEFAULT_GROUP=Администраторы # группа, в которую автоматически
попадает пользователь, созданный администратором
VISMIND_LDAP_SYNC_GROUPS_WITH_AD=0 # признак синхронизации групп
пользователя с теми группами, которые у него есть в LDAP

```

При использовании LDAP пользователь будет являться **внешним** по отношению к платформе. Внешнему пользователю нельзя изменить атрибуты, которые получаются при его подключении из LDAP. При первом подключении такого пользователя в платформе будет создана учетная запись, которая будет заполнена данными из LDAP. При настройках по умолчанию, пользователь будет добавлен в группу Администраторы.

Описание настройки сквозной авторизации с использованием SPNEGO Kerberos тикетов приведено в п. 5.2.1.

4.4.6.2 Некоторые комментарии по kerberos-аутентификации

При входе в платформу, если последней была использована kerberos-аутентификация, пользователю будет открыта страница kerberos-аутентификации. При этом сама аутентификация происходит через некоторое время по окончании загрузки всех ресурсов на странице. Если необходимо сменить тип авторизации, необходимо выйти из приложения. Для этого надо либо нажать «Выйти» в меню, либо перейти на страницу /logout.

В режиме инкогнито сквозная авторизация не работает, как и в случае если учетная запись пользователя, под которой осуществлен вход, не является доменной. В этом случае пользователю будет предложено ввести логин и пароль средствами браузера. В этом окне необходимо указывать логин и пароль с именем домена, например: MYDOMAIN\Username.

4.4.7 Установка клиента Oracle Database

Для обеспечения работы с базой данных Oracle необходимо на сервере, где будет развернуто приложение, провести установку клиента:

1. Скачать последнюю версию клиента Oracle с сайта: <https://www.oracle.com/ru/database/technologies/instant-client/linux-x86-64-downloads.html>

2. Распаковать zip-архив в папку /opt на сервере, где будет запущен контейнер **vismind-backend**.
3. Перейти в папку и добавить символическую ссылку: `ln -s . lib`
4. Смонтировать папку клиента при запуске контейнера, добавив опцию для **docker start**:
`--volume=/opt/instantclient_21_1:/opt/oracle`

4.5 Переменные окружения

Приложение при старте проверяет переменные окружения. Для того чтобы переменная из конфигурационного файла была считана из переменных окружения, она должна начинаться на префикс **VISMIND_** в переменных окружения.

Важно! Проверяются все переменные из конфигурационного файла в переменных окружения.

Например: считать название приложения.

Название приложения в конфигурационном файле находится в переменной **app_name**. Тогда к **app_name** добавим префикс **VISMIND** – получится параметр **VISMIND_APP_NAME**. Этот параметр будет искаться в переменных окружения.

И в переменную **app_name** присвоится значение:

- или из переменной окружения **VISMIND_APP_NAME**, если она будет найдена;
- или из **app_name**, если переменная окружения не найдена.

4.6 Песочница и удаленный запуск расчетов

Код из блока Python можно запустить на выполнение удаленно (Рисунок 2). При этом код будет выполняться внутри изолированного контейнера, что обеспечит дополнительный уровень безопасности.

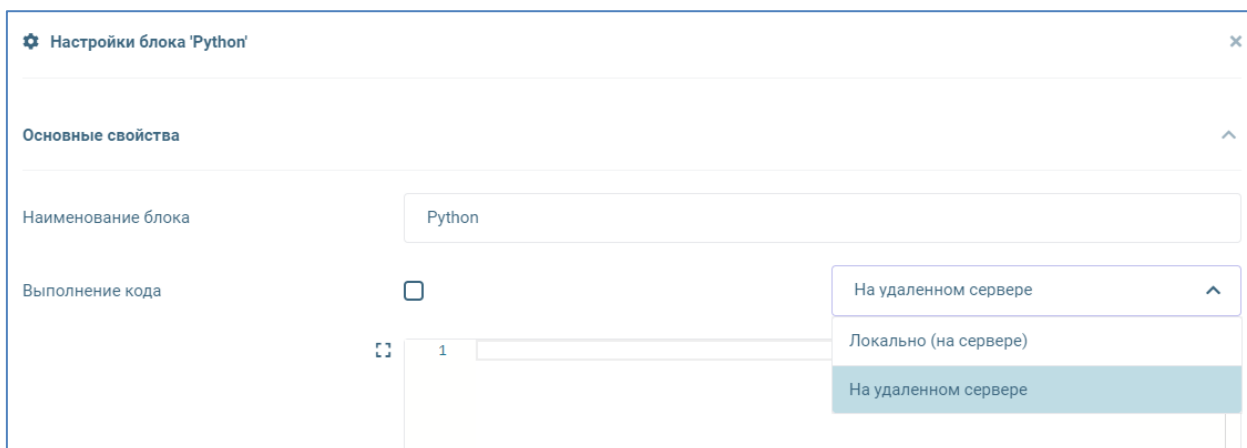


Рисунок 2 – Пример настройки блока Python

Это накладывает определенные ограничения на сам код – нельзя использовать методы библиотеки ядра, так как ядро системы отсутствует при выполнении в удаленном контейнере (методы `VmResource`).

4.6.1 Сборка docker-образа для запуска кода в удаленном окружении

Библиотеки, которые должны быть доступны внутри контейнера (подключаемые к коду), должны быть добавлены на этапе сборки образа.

1. Собрать образ приложения из репозитория **vismind-sandbox**.

```
docker build -f .\Dockerfile -t vmbackend:sandbox .
```

2. Загрузить в docker контейнер. Имя и тег по умолчанию: **vmbackend:sandbox**.

```
docker save -o D:\\vmbackend.sandbox.tar vmbackend:sandbox
```

Результатом будет файл **vmbackend.sandbox.tar**, который можно включить в инсталлятор приложения.

4.6.2 Загрузка образа в локальный docker репозиторий сервера

На сервер, на котором будет выполняться код в изолированном контейнере, необходимо загрузить образ:

```
docker load --input D:\\vmbackend.sandbox.tar
```

4.6.3 Настройка сервера для доступа к docker

После этого необходимо разрешить соединение и управление **docker** по сети. Приложение будет соединяться с сервером, запускать образ с кодом, который создал пользователь, и после получения результата, удалять за собой отработавший контейнер.

Необходимо создать файл:

```
/etc/systemd/system/docker.service.d/startup_options.conf
```

Без сертификатов безопасности (TCP-порт 2375):

```
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H fd:// -H tcp://0.0.0.0:2375 --
containerd=/run/containerd/containerd.sock
```

Перезапустить сервис:

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

Добавить разрешение в firewall:

```
sudo ufw allow 2375/tcp
```

После этого сервис доступен по сети на 2375 TCP-порту.

С использованием сертификатов безопасности (TCP-порт 2376):

Добавить файл расширения сервиса **systemd** (основной файл с описанием юнита `/etc/systemd/system/multi-user.target.wants/docker.service`) для изменения параметров старта демона:

```
/etc/systemd/system/docker.service.d/startup_options.conf
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H fd:// -H tcp://0.0.0.0:2376 --
containerd=/run/containerd/containerd.sock --tlsverify --
tlscacert=/etc/docker/ssl/ca.pem --tlscert=/etc/docker/ssl/cert.pem --
tlskey=/etc/docker/ssl/key.pem
```

Первой пустой строкой сбрасывается значение параметров старта для сервиса **docker**. Во второй необходимо указать:

- `-H fd://` — означает, что параметрами прослушивания сокета управляет **systemd** в файле `/etc/systemd/system/sockets.target.wants/docker.socket` (по умолчанию подключение к `dockerd` осуществляется через unix-сокет `/var/run/docker.sock`,

ИЗМЕНЯТЬ МОЖНО В ЮНИТЕ СОКЕТА ИЛИ ПАРАМЕТРОМ
`-H unix:///var/run/docker-bootstrap.sock);`

- `H tcp://0.0.0.0:2376` – указываем, что демон **dockerd** (бэкенд часть платформы контейнеризации, не путать с клиентской частью **docker**) запустит серверный TCP-порт 2376, слушающий входящие соединения на всех доступных IP-адресах сетевых интерфейсов (для запуска сокета только на конкретном IP-адресе укажите его вместо 0.0.0.0);
- `--tls*` – параметры, включающие защиту TLS-сертификатами сетевого подключения к сервису.

С TLS-сертификатами безопасности TCP-порт 2376. Разрешить доступ к порту в файрволе:

```
sudo ufw allow 2376/tcp
```

Обновите конфигурацию демонов **systemd** и перезагрузить сервис **docker**:

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

Посмотреть статус запущенного сервиса можно командой:

```
systemctl status docker
```

Основной процесс `/usr/bin/dockerd` должен быть запущен с новыми параметрами.

Проверим, что подключение без сертификата будет отклонено:

```
docker -H tcp://127.0.0.1:2376 info
```

Вывод ошибки:

```
Get http://127.0.0.1:2376/v1.21/containers/json: malformed HTTP response
"\x15\x03\x01\x00\x02\x02".
```

* Are you trying to connect to a TLS-enabled daemon without TLS?

4.6.4 Настройка образа в приложении

Для настройки образа в приложении необходимо:

1. Указать название образа контейнера:

```
docker_for_python_iso_name = "vmbackend:sandbox"
```

2. Указать URL docker-контейнера:

```
docker_for_python_base_url = "tcp://dev.int.bittechno.ru:2376"
```

3. Добавить сертификаты docker-контейнера:

- Путь к сертификату клиента:

```
docker_for_python_client_cert = "./conf/docker/cert.pem"
```

- Путь к ключу клиента:

```
docker_for_python_client_key = "./conf/docker/key.pem"
```

- Путь к центру сертификации:

```
docker_for_python_ca_cert = "./conf/docker/ca.pem"
```

- Установить флаг, указывающий необходимость удаления контейнера после выполнения:

```
docker_for_python_is_remove = 1
```

- Этот параметр нельзя отставлять выключенным, он нужен для отладки контейнера, в случае если код по какой-то причине не выполняется и приложению не отправляется корректная ошибка.
- Если его выключить, каждый запуск будет создавать контейнер, но **не будет его удалять** по завершению работы, что приведет к большому количеству контейнеров, которые останутся завершенными в docker сервера.
- Ограничение по используемым ресурсам:
 - по процессору `docker_for_python_cpus`. По умолчанию *1 процессор*
 - по памяти `docker_for_python_mem_limit`. По умолчанию *4g*

Если образ не установлен или настройки приложения заданы некорректно – при попытке запуска будет возвращена системная ошибка, которую вернет сам сервер. Это необходимо для корректной отладки соединения, в случае если настройки некорректные.

4.7 Прокси-сервер для клиентского приложения

Для корректной работы пользовательского приложения в docker-контейнере и без него нужен прокси-сервер, на котором будут выполняться роли:

1. Хостинг приложения по пути '/'.
 - /api
 - /socket.io
 - /metrics
2. Проксирование запросов к бэкенду по путям:
3. Подготовка и отправка статики по пути /static.
4. *Опционально*: предоставление доступа к эндпоинту /metrics для мониторинга приложения с использованием Prometheus.
5. *Опционально*: предоставление эндпоинта для проксирования запросов и организации кэширования ответов тайлового сервера Martin.

Таким образом, прокси-сервер выполняет роль диспетчера всех запросов приложения и является критически важным элементом при развертывании приложения.

В качестве прокси-сервера используется **nginx** (на 03 Feb 2023 используется версия 1.23.3).

4.7.1 Базовая конфигурация

nginx.conf

```
# Определяем пользователя
user  nginx;

# Определяем количество рабочих процессов автоматически
# Параметр auto поддерживается только начиная с версий 1.3.8 и 1.2.5.
worker_processes auto;

# Данная директива указывает сколько файловых дескрипторов будет использовать Nginx. На каждое соединение надо
# выделять по два дескриптора, даже для статических файлов (картинки/JS/CSS):
# один для соединения с клиентом,
```

```

# а второй – для открытия статического файла. Таким образом, значение
worker_rlimit_nofile должно быть не меньше
# удвоенного значению Max Clients.
# В системе это значение можно установить из командной строки ulimit -n 200000
или используя
# /etc/security/limits.conf.
# Проверить установленное значение в системе
# hard limit `ulimit -Hn` и soft limit `ulimit -Sn`
worker_rlimit_nofile 20000;

# !! для установки без контейнера надо вывод ошибок перенаправить в файлы, а не
STDOUT !!
# pid          /var/log/nginx/nginx.pid;
# задаем общие параметры логирования
# error_log    /var/log/nginx/error.log warn;
error_log      /dev/stdout info;
# логируем только критические ошибки
# error_log    /var/log/nginx/error.log crit
# Если нужно полностью отключить ошибки (использовать с пониманием зачем это
нужно)
# error_log    /dev/null crit;

events {
    # увеличение по сравнению со стандартным 1024, для продуктивного контура,
    можно увеличивать больше в зависимости от нагрузки и ресурсов
    # worker_processes * worker_connections = max_clients (кол-во одновременных
    соединений)
    worker_connections 2048;

    # предпочтительнее для Linux
    use epoll;

    # Для того, чтобы Nginx пытался принять максимальное количество подключений,
    необходимо включить
    # директиву multi_accept. Однако при слишком маленьком значении
    worker_connections, их лимит может быть
    # быстро исчерпан.
    multi_accept on;
}

http {

    include      /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';

    # Куда писать лог доступа и уровень логирования
    # добавляем buffer=16k для оптимизации логирования
    # access_log  /var/log/nginx/access.log  main buffer=16k;
    access_log  /dev/stdout main;
    # либо отключаем лог – так будет работать намного быстрее
    # access_log off;
}

```

```

# Для нормального ответа 304 Not Modified;
if_modified_since before;

# Включаем поддержку WebP
map $http_accept $webp_ext {
    default "";
    "~*webp" ".webp";
}

##
# Basic Settings
##

# Используется, если количество имен серверов большое
#server_names_hash_max_size 1200;
#server_names_hash_bucket_size 64;

### Обработка запросов ###

# Метод отправки данных sendfile более эффективен, чем стандартный метод
read+write
sendfile on;
# Будет отправлять заголовки и начало файла в одном пакете
tcp_nopush on;
tcp_nodelay on;

### Адрес резолвера DNS ###
# resolver 10.0.0.1 valid=30s;

### Информация о файлах ###

# Максимальное количество файлов, информация о которых будет содержаться в
кеше
open_file_cache max=200000 inactive=20s;
# Через какое время информация будет удалена из кеша
open_file_cache_valid 30s;
# Кеширование информации о тех файлах, которые были использованы хотя бы 2
раза
open_file_cache_min_uses 2;
# Кеширование информации об отсутствующих файлах
open_file_cache_errors on;

# Удаляем информацию об nginx в headers
server_tokens off;
# Будет ждать 30 секунд перед закрытием keepalive соединения
keepalive_timeout 30s;
## Максимальное количество keepalive запросов от одного клиента
keepalive_requests 100;

# Разрешает или запрещает сброс соединений по таймауту
reset_timedout_connection on;

```

```

# Будет ждать 30 секунд тело запроса от клиента, после чего сбросит
соединение
client_body_timeout 30s;
# В этом случае сервер не будет принимать запросы размером более 200Мб
client_max_body_size 200m;
# Если клиент прекратит чтение ответа, Nginx подождет 30 секунд и сбросит
соединение
send_timeout 30s;

# Proxy #
# Задаёт таймаут для установления соединения с проксированным сервером.
# Необходимо иметь в виду, что этот таймаут обычно не может превышать 75
секунд.
proxy_connect_timeout 30s;
# Задаёт таймаут при передаче запроса проксированному серверу.
# Таймаут устанавливается не на всю передачу запроса, а только между двумя
операциями записи.
# Если по истечении этого времени проксируемый сервер не примет новых
данных, соединение закрывается.
proxy_send_timeout 30s;
# Задаёт таймаут при чтении ответа проксированного сервера.
# Таймаут устанавливается не на всю передачу ответа, а только между двумя
операциями чтения.
# Если по истечении этого времени проксируемый сервер ничего не передаст,
соединение закрывается.
proxy_read_timeout 30s;

# Размер буфера используемого для проксированных запросов
# proxy_buffers 32 4k;

##
# Кэш для геосервера
##
# proxy_cache_path    /var/cache/nginx/
#                      levels=1:2
#                      max_size=10g
#                      inactive=60m
#                      use_temp_path=off
#                      keys_zone=tiles_cache:10m;

##
# Gzip Settings
##

# Включаем сжатие gzip
gzip on;
# Для IE6 отключить
gzip_disable "msie6";
# Добавляет Vary: Accept-Encoding в Headers
gzip_vary on;
# Сжатие для всех проксированных запросов (для работы NGINX+Apache)
gzip_proxied any;
# Устанавливает степень сжатия ответа методом gzip. Допустимые значения
находятся в диапазоне от 1 до 9
gzip_comp_level 6;

```

```

# Задаёт число и размер буферов, в которые будет сжиматься ответ
gzip_buffers 16 8k;
# Устанавливает минимальную HTTP-версию запроса, необходимую для сжатия
ответа. Значение по умолчанию
gzip_http_version 1.1;
# MIME-типы файлов в дополнение к text/html, которые нужно сжимать
gzip_types text/plain text/css application/json application/x-javascript
text/xml application/xml application/xml+rss text/javascript
application/javascript image/svg+xml;
# Минимальная длина файла, которую нужно сжимать
gzip_min_length 10;

# Подключаем конфиги конкретных сайтов
include /etc/nginx/conf.d/*.conf;

}

```

Для самого приложения в контейнере или в **nginx** настраивается конфигурация **default.conf**, которая располагается в каталоге `/etc/nginx/conf.d/`.

Если установка администратором произведена в другое место (при установке не в контейнере) – необходимо не забыть заменить в файле **nginx.conf** путь до конфигураций сайтов.

default.conf

```

upstream app_server {
    # указываем правильный адрес
    server vismind-backend:35115 max_fails=5 fail_timeout=5s;
    # ребуется для оптимизации keep alive соединений
    keepalive 50;
}

server { # подключение к бэкенду 80 и 443 порты

    listen      80;
    # listen    443 http2 ssl;
    # ssl_certificate      /etc/ssl/certs/ssl.crt;
    # ssl_certificate_key /etc/ssl/private/ssl.key;

    # устанавливать если меняется на доменное имя
    # server_name my_server_name;

    # Основное приложений
    location / {
        root    /usr/share/nginx/html;
        try_files $uri$args $uri$args/ $uri/ /index.html;
    }

    # Папка статики
    location /static/ {
        root /usr/share/nginx/media;
        add_header Access-Control-Allow-Origin *;
        # expires 1d; # VISMIND-6615
    }

    # Обратный прокси для бэкенда

```

```

location /api/ {
    proxy_http_version 1.1;
    proxy_set_header    "Connection" "";
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;

    proxy_connect_timeout    600;
    proxy_send_timeout       600;
    proxy_read_timeout       600;
    proxy_buffers            32 4k;

    proxy_redirect $scheme://$host/api/v1/ $scheme://$http_host/api/;

    # с бэкендом работа идет по http
    proxy_pass http://app_server/api/v1/;
}

# Проброс сокетов
location /socket.io {
    proxy_http_version 1.1;
    proxy_set_header    Upgrade $http_upgrade;
    proxy_set_header    Connection "Upgrade";
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_pass http://app_server;
    tcp_nodelay on;
    keepalive_timeout 600;
}

# Эндпоинт для получения метрик приложения
location /metrics {
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;
    proxy_pass http://app_server/metrics;
}

# Геосервер с кэшированием запросов
location ~ /tilecache/(?<fwd_path>.*) {
    proxy_pass
http://dev.int.bittechno.ru:3000/static/$fwd_path$sis_args$args;
}

# Геосервер без кэширования, по сути прокси
location ~ /tileserver/(?<fwd_path>.*) {
    proxy_pass
http://dev.int.bittechno.ru:3000/dynamic/$fwd_path$sis_args$args;
}

# 404 ошибка
#error_page 404 /404.html;

```

```

# перенаправление при ошибках 50x на /50x.html2
#error_page 500 502 503 504 /50x.html;
#location = /50x.html {
#   root /usr/share/nginx/html;
#}
}

```

4.7.2 Прокси-сервер для относительных адресов геосервера

Дополнительная роль, которую может выполнять **nginx** – перенаправление запросов на геосервер в случае использования относительных адресов. Это необходимо для того, чтобы не переписывать адреса при переносе задач между разными окружениями.

Для этого в конфигурации задается параметр **location** для кэшируемых и динамических тайлов. Геосервер должен иметь свой прокси-сервер с настроенным кэшированием по каталогам `/static` и `/dynamic`.

Пример location для reverse proxy

```

# Геосервер с кэшированием запросов
location ~ /tilecache/(?<fwd_path>.*) {
    proxy_pass
http://martin_address/static/$fwd_path$is_args$args;
}

# Геосервер без кэширования, по сути прокси
location ~ /tileserver/(?<fwd_path>.*) {
    proxy_pass
http://martin_address:3000/dynamic/$fwd_path$is_args$args;
}

```

При использовании проброса запросов на геосервер они направляются на прокси-сервер геосервера, который отвечает за кэширование и их перенаправления на сам геосервер (Рисунок 3).

Важно! При использовании этого варианта проксирования запросов нельзя изменять заголовки запросов, так как ответ геосервера содержит адрес, по которому надо получать тайлы, а этот адрес формируется из заголовков запросов пришедших на геосервер.

4.7.2.1 Схема работы прокси-сервера и геосервера

Схема работы прокси-сервера и геосервера приведена на рисунке (Рисунок 3).

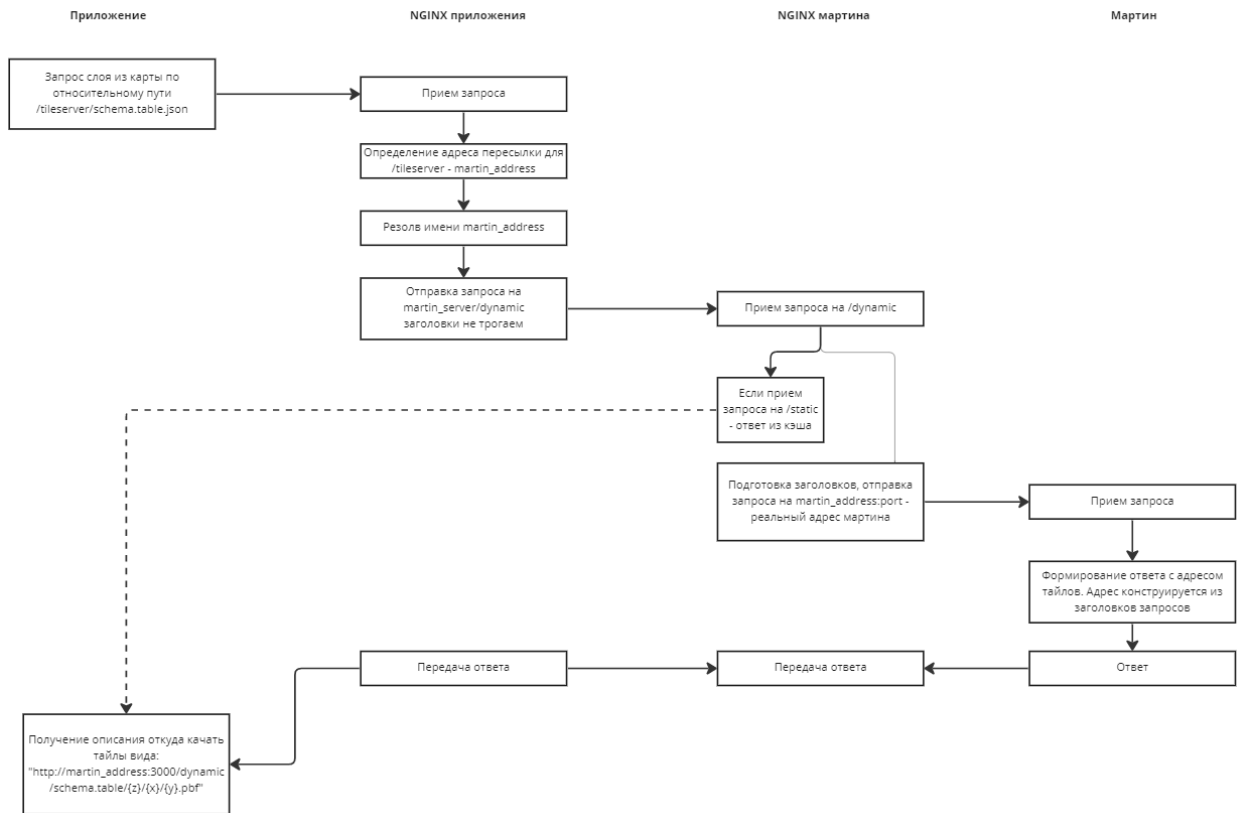


Рисунок 3 – Схема работы прокси-сервера и геосервера

5 Аутентификация и авторизация

5.1 Способы аутентификации

На странице аутентификации (Рисунок 4) в выпадающем списке можно выбрать следующие способы аутентификации:

- Вход по имени и паролю;
- Вход с использованием LDAP;
- Вход с использованием Kerberos.

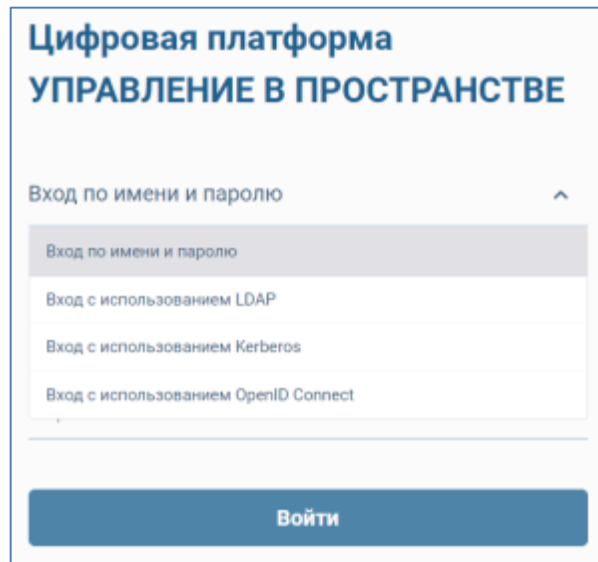


Рисунок 4 – Страница аутентификации

Элементы в списке отображаются согласно настройкам «AllowLogin» (true – отображается, false – не отображается) в файле конфигурации **app.json** в части frontend.

5.2 Авторизация с помощью Kerberos

Приложение (VISMIND) реализует сквозную авторизацию через Kerberos тикет на уровне бэкенда. Для авторизации пользователя используется API логина, которое позволяет получить информацию о пользователе и его группах. На стороне клиентского приложения дополнительных настроек, кроме включения авторизации, не требуется.

Приложение может работать с Kerberos-авторизацией. При этом попытка входа будет осуществляться автоматически при входе на главную страницу приложения.

Если адрес приложения не добавлен в доверенные, браузер предложит ввести логин и пароль для входа. Для подключения необходимо указать свой логин в формате DOMAIN\username и пароль во всплывающем окне.

5.2.1 Работа с SPNEGO и Kerberos-аутентификацией

Для настройки сквозной авторизации с использованием SPNEGO Kerberos тикетов, когда авторизация происходит с помощью доменной учетной записи, необходимо выполнить несколько шагов.

1. Включить доступ к механизмам авторизации в приложении, установив конфигурацию:

```
VISMIND_KERBEROS_ALLOW_LOGIN=1 # разрешить
использование механизма авторизации
VISMIND_KERBEROS_SYSTEM_USER_LOGIN=UserLogin # доменный
пользователь, под которым можно получить информацию в AD
# это тот же пользователь, которому выписывается keytab файл
VISMIND_KERBEROS_SYSTEM_USER_PASSWORD=Pa$$word # доменный пароль
пользователя
```

2. Для корректной работы необходимые сгенерированные на сервере файлы:

```
krb5.conf # файл конфигурации подключения к AD для библиотеки gssapi
krb5.keytab # файл, сгенерированный для приложения на сервере AD
```

Инструкция по настройке SPN записи и генерации keytab-файла приведена в п. 5.3.5.

1. Задать корректный hostname для серверной части приложения.

Важно! Он должен совпадать с тем FQDN, на который выписывается keytab-файл.

```
...
  container_name: vismind-backend
  image: vismind-backend:0.36-fortum
  hostname: some.fqdn.domain.ru # хостнейм для контейнера для
работы авторизации
  environment:
...

```

2. Перезапустить приложение.

5.3 Настройка веб-сервера для Kerberos-аутентификации

5.3.1 Введение

Настройка сквозной аутентификации для приложений в связке с доменом Active Directory достаточно нетривиальная задача. Схематично основные шаги по настройке приведены на рисунке (Рисунок 5).

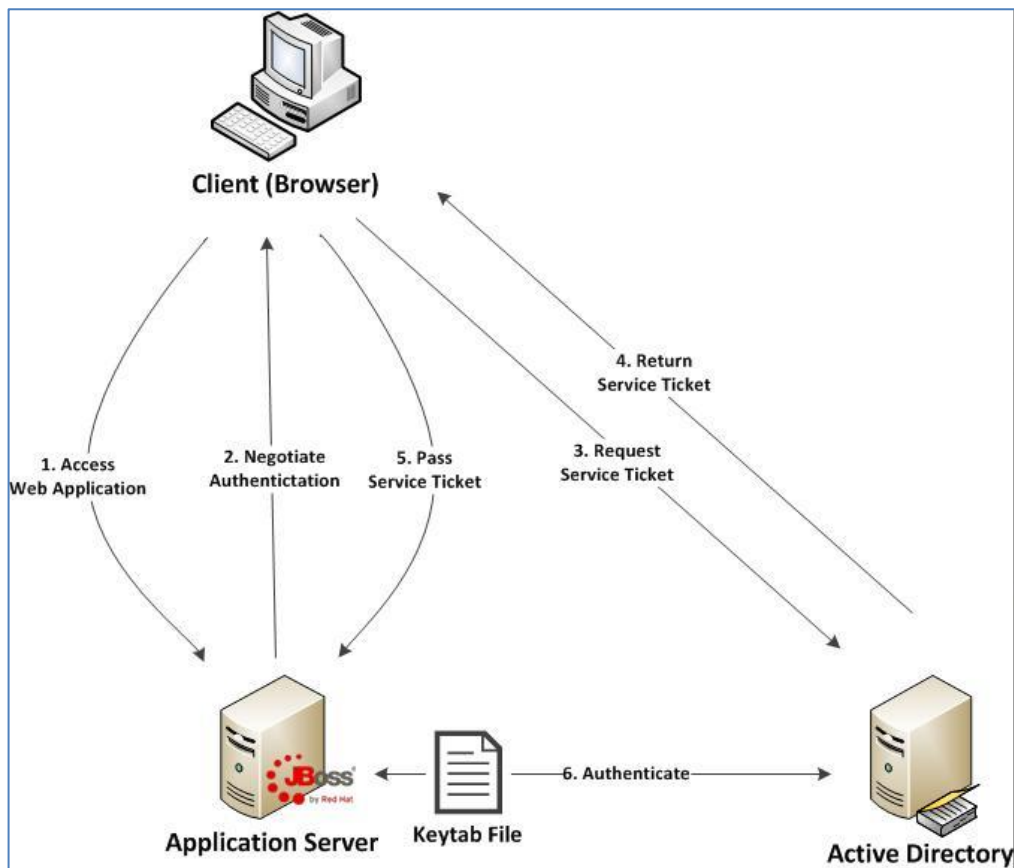


Рисунок 5 – Настройка сквозной аутентификации

5.3.2 Исходные данные

Служба каталогов **Microsoft Active Directory**

Имя домена Windows: int.bittechno.ru

Имя контроллера домена Windows 2012 Server: **win**

IP Address контроллера домена Windows Server: **192.168.88.3**

Сервер для работы приложения: **Ubuntu 20, Nginx, Python 3.7**

Hostname Linux Server: **srv-app**

IP Address Linux Server: **192.168.88.210**

URL Приложения: srv-app.int.bittechno.ru

Важно! Везде, где названия прописаны заглавными буквами – оставлять их заглавными. Регистр важен. Например, INT.BITTECHNO.RU

5.3.3 Решаемая задача

Организовать аутентификацию пользователей в приложении с использованием протокола kerberos, настроить автоматический вход пользователей Active Directory в приложение в браузере Google Chrome.

5.3.4 Алгоритм работы

1. Создаем и настраиваем пользователя для авторизации приложения в домене, генерируем ему **keytab-файл**.
2. Устанавливаем и настраиваем на каждом узле **kerberos**.

3. Для URL-адреса приложения создаём в Active Directory запись [Service Principal Names](#).
4. Настраиваем приложение и его сборку.

5.3.5 Подготовка контроллера домена

На работающем контроллере домена необходимо провести подготовительные работы, чтобы приложение имело доступ к Active Directory и могло аутентифицировать пользователей.

5.3.5.1 Создание пользователя

Для начала создадим пользователя, под которым приложение будет иметь доступ к домену. В домене создаём отдельного пользователя, который будет использоваться для аутентификации. Добавляем его в группу «Domain guests», назначаем ее основной и исключаем созданного пользователя из остальных групп. Также целесообразно запретить смену пароля пользователем и убрать ограничение на срок действия пароля.

Важно! Если ограничить пароль – **keytab-файлы** придется генерировать заново.

На скриншоте (Рисунок 6) пользователь **kerbuser** добавлен в несистемные группы для выдачи ему прав в приложении.

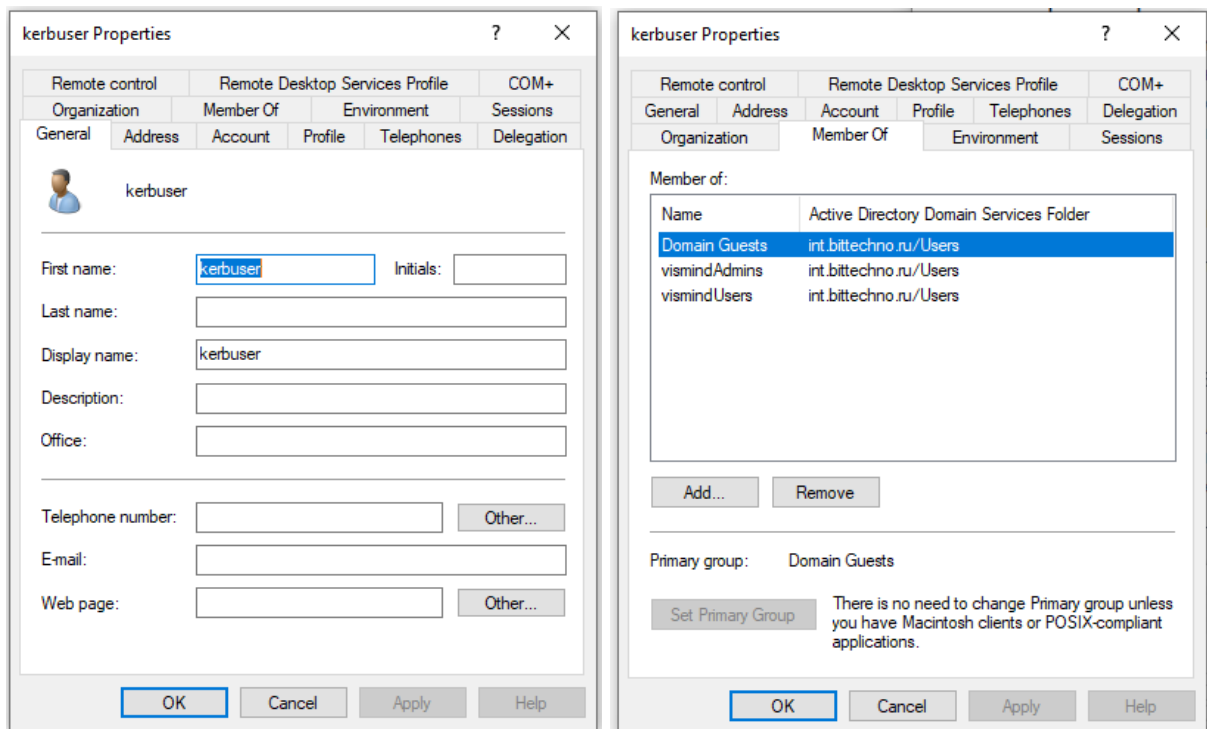


Рисунок 6 – Свойства пользователя **kerbuser**

Важно! В описании пользователя, пути его расположения не должно быть русских символов.

5.3.5.2 Создание DNS-записи

Необходимо настроить DNS-запись приложения. Kerberos работает только с записями типа **A**, то есть в DNS должна быть запись вида:

```
srv-app A 192.168.88.210
```

Запрос имени к DNS-серверу должен возвращать только IP-адрес, причём один, например:

```
nslookup srv-app.int.bittechno.ru
Server: WIN.int.bittechno.ru
```

Address: 192.168.88.3

Name: srv-app.int.bittechno.ru

Address: 192.168.88.210

В качестве имени DNS не должны выступать записи вида CNAME или ALIAS, также не должно быть нескольких записей с типом А. Пример настройки DNS-записи приведен на рисунке (Рисунок 7).

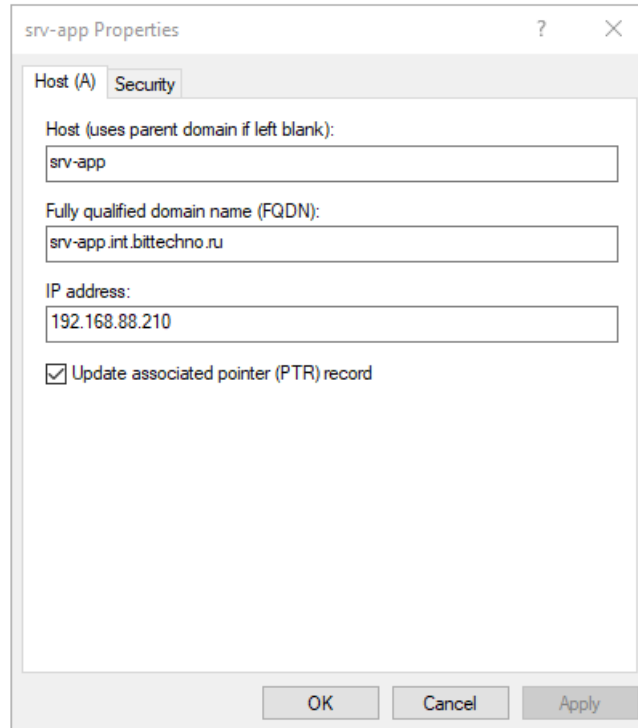


Рисунок 7 – Пример настройки DNS-записи

Если у вас несколько узлов, то необходимо настроить балансировщик, к примеру, [nginx](#), и затем настроить DNS-запись с типом А вашего приложения, указывающую на балансировщик.

5.3.5.3 Создание SPN-записи

Создадим SPN ([Service Principal Names](#)) для установленного приложения – HTTP/srv-app.int.bittechno.ru и привяжем его к пользователю **kerbuser**. Для этого в командной строке на контроллере домена выполним следующую команду:

```
setspn -A HTTP/srv-app.int.bittechno.ru kerbuser
Регистрация ServicePrincipalNames для CN=keruser,CN=Users,DC=domg,DC=testg
HTTP/srv-app.int.bittechno.ru
Обновленный объект
```

Проверить привязанные SPN у пользователя можно командой:

```
setspn -L kerbuser
Зарегистрирован ServicePrincipalNames для
CN=kerbuser,CN=Users,DC=domg,DC=testg:
HTTP/serv-app.int.bittechno.ru
```

5.3.5.4 Запрос данных пользователя

Можно запросить данные о пользователе из каталога:

```
dsquery * -filter "(objectClass=user) "
```

```
dsquery * -filter "(objectClass=organizationalPerson)"
dsquery users
```

5.3.5.5 Генерация keytab-файла

Внимание! На каждый SPN (адрес) необходимо создавать отдельного пользователя. Это связано со спецификой Active Directory, иначе в дальнейшем при привязке других SPN (адресов) к одному и тому же пользователю, остальные SPN (и keytab-файлы) перестанут работать.

Генерируем keytab-файл, он понадобится при настройке сервера с приложением.

```
ktpass /out C:/kerbuser.keytab /mapuser kerbuser@INT.BITTECHNO.RU /princ
HTTP/srv-app.int.bittechno.ru@INT.BITTECHNO.RU /pass Q!w2e3r4_ /ptype
KRB5_NT_PRINCIPAL /crypto All
```

Где:

- srv-app.int.bittechno.ru – DNS имя приложения, по которому будут обращаться пользователи;
- kerbuser – пользователь домена для аутентификации;
- Q!w2e3r4_ – пароль пользователя домена;
- INT.BITTECHNO.RU – realm домена Active Directory, в котором будет аутентифицировано приложение;
- krb5.keytab – имя keytab-файла.

Важно! Если при создании файла возникла ошибка Password set failed! 0x00000020 – это означает, что в указанном каталоге хранения пользователя или в его описании есть русские буквы. Необходимо использовать только латинские буквы.

5.3.6 Подготовка сервера

В случае если серверов несколько – каждый настраивается одинаково.

5.3.7 Синхронизации времени

Первым шагом необходимо обеспечить точное время на сервере, так как расхождение по времени с контроллером домена авторизация будет невозможной. Для обеспечения точного времени актуально использовать демон **ntp** или **chrony**, настроив его либо на синхронизацию времени с локальным сервером времени, либо с публичными серверами времени в Интернет.

5.3.7.1 Установка chrony

```
sudo apt install chrony
```

Проверить работу можно командой

```
chronyc activity
```

Список серверов, с которыми идет синхронизация времени, можно изменить командой:

```
sudo nano /etc/chrony/chrony.conf
```

```
sudo systemctl restart chrony
```

5.3.8 Установка kerberos библиотек

Перед установкой скопируем на сервер в /home/user/krb5.keytab, сгенерированный на контроллере домена. Скопировать файл можно в любой каталог, но придется указывать его при запросах. Если приложение одно (а на каждое приложение нужен свой keytab-файл), можно сохранить его в каталог /etc/krb5.keytab.

Устанавливаем необходимые библиотеки:

```
apt-get install krb5-doc krb5-user krb5-config libpam-krb5 libkrb5-dev
```

Редактируем файл конфигурации:

```
sudo nano -w /etc/krb5.conf
```

```
[libdefaults]
    default_realm = INT.BITTECHNO.RU
    dns_lookup_realm = true
    dns_lookup_kdc = true
    kdc_timesync = 1
    ccache_type = 4
    forwardable = true
    proxiable = true
    default_keytab_name = /etc/krb5.keytab
[realms]
    INT.BITTECHNO.RU = {
        kdc = win.int.bittechno.ru
        admin_server = win.int.bittechno.ru
        default_domain = INT.BITTECHNO.RU
    }
[domain_realm]
    .int.bittechno.ru = INT.BITTECHNO.RU
    int.bittechno.ru = INT.BITTECHNO.RU
[login]
    krb4_convert = true
    krb4_get_tickets = false
[logging]
    default = FILE:/var/log/krb5libs.log
    kdc = FILE:/var/log/krb5dc.log
    admin_server = FILE:/var/log/ksadmin.log
```

Обратите внимание, в строке вида:

```
kdc = win.int.bittechno.ru
```

должен быть либо IP-адрес контроллера домена, либо DNS-адрес, ссылающийся на один IP-адрес. С DNS-записью, ссылающейся на группу адресов, авторизация во время настройки не работает. Если контроллеров домена несколько, то их нужно указать в отдельных строках, например:

```
kdc = dc1
kdc = dc2
...
```

Через двоеточие можно указать порт:

```
kdc = dc1:88
kdc = dc2:88
...
```

Создаем файлы логов:

```
sudo touch /var/log/krb5libs.log && sudo touch /var/log/krb5dc.log && sudo touch /var/log/ksadmin.log
```

Указываем владельца файлов и устанавливаем права на keytab-файл.**Пример:**

```
chown gandrianov:gandrianov /home/gandrianov/krb5.keytab
```

```
chmod 0600 /home/gandrianov/krb5.keytab
```

Теперь проверим keytab-файл:

```
kinit -kt krb5.keytab HTTP/srv-app.int.bittechno.ru@INT.BITTECHNO.RU
```

Команда отработает без вывода. Затем можно просмотреть список полученных тикетов **kerberos** на сервере:

```
klist
```

```
Credentials cache: FILE:/tmp/krb5cc_1000
```

```
Principal: HTTP/srv-app.int.bittechno.ru@INT.BITTECHNO.RU
```

```

      Issued                Expires                Principal
Apr 28 18:24:07 2022      Apr 29 04:24:07 2022
krbtgt/INT.BITTECHNO.RU@INT.BITTECHNO.RU
```

Удаление тикета выполняется командой:

```
kdestroy
```

5.3.9 Настройка окружения

5.3.9.1 Настройка приложения

Приложение (VISMIND) реализует авторизацию через kerberos тикет на уровне серверного приложения (бэкенда). Для авторизации пользователя используется API логина, которое позволяет получить информацию о пользователе и его группах. На стороне клиентского приложения дополнительных настроек, кроме включения авторизации, не требуется.

Пример настройки адреса приложения на трёх узлах для **nginx**. Специфических настроек не требуется (*gss, pam*):

```

upstream spnego-kerb-cluster {
    ip_hash;
    server srv-app-1.int.bittechno.ru:5000;
    server srv-app-2.int.bittechno.ru:5000;
}

server {
    listen 80 default_server;

    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;
    server_name _;

    location / {
        root /var/www/html/angular-demo;
        try_files $uri$args $uri$args/ $uri/ /index.html;
        #index index.html index.htm;
    }

    location /api {
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}

```

```

        proxy_max_temp_file_size 0;
        client_max_body_size      10m;
        client_body_buffer_size   128k;
        proxy_connect_timeout     600;
        proxy_send_timeout        600;
        proxy_read_timeout        600;
        proxy_buffers              4 256k;
        proxy_buffer_size          128k;
        proxy_busy_buffers_size    256k;

    proxy_pass http://spnego-kerb-cluster;
    }
}

```

В настройках бэкенда необходимо задать значения для коннекта к AD и авторизации на нем (используется NTLM).

```

...
<ldap>
  <allow_login>1</allow_login>
  <host>192.168.88.3</host>
  <port>389</port>
  <domain_component>dc=int,dc=bittechno,dc=ru</domain_component>
  <users_common_name>users</users_common_name>
  <user>kerbuser</user>
  <password>Pa$$word</password>
</ldap>
...

```

5.3.9.2 Docker

Для запуска приложения в контейнере необходимо передать в него файлы **krb5.keytab** и **krb5.conf**. Пример **Dockerfile**:

```

FROM python:3.7-slim

COPY ./krb5.conf /etc/krb5.conf
COPY ./krb5.keytab /etc/krb5.keytab

# ENV KRB5_KTNAME=FILE:/etc/krb5.keytab
# ENV KRB5_CONFIG=

RUN apt-get update -y && \
    apt-get install -y curl python-dev gcc unixodbc unixodbc-dev libaiol && \
    apt-get install --reinstall build-essential -y && \
    apt-get install -y chrony && \
    apt-get install -y krb5-doc && \
    apt-get install -y krb5-user && \
    apt-get install -y krb5-config && \
    apt-get install -y libpam-krb5 && \
    apt-get install -y libkrb5-dev

RUN touch /var/log/krb5libs.log && touch /var/log/krb5dc.log && touch
/var/log/ksadmin.log

WORKDIR /app

```

```
COPY requirements.txt requirements.txt
```

```
RUN pip install -r requirements.txt -i
http://nexus.int.bittechno.ru/repository/python/simple --trusted-host
nexus.int.bittechno.ru
```

```
# файлы проекта для старта приложения через gunicorn
```

```
COPY ./app.py .
COPY ./gunicorn.conf.py .
COPY ./wsgi.py .
```

```
COPY ./docker-entrypoint.sh /
```

```
ENTRYPOINT [ "/docker-entrypoint.sh" ]
```

Для успешного запуска сервиса в контейнере необходимо, чтобы имя контейнера совпадало с SPN-записью на контроллере домена. Для этого, при старте контейнера, надо указать имя хоста через параметр **-h**:

```
docker run --name krb -p 5000:35115 -h srv-app.int.bittechno.ru -d krb
```

5.3.1 Дополнительные ссылки

<https://github.com/COUR4G3/flask-gssapi>

<https://habr.com/ru/post/305098/>

<https://winitpro.ru/index.php/2018/01/22/nastrojka-kerberos-autentifikacii-v-razlichnyx-brauzerax/>

<https://techexpert.tips/nginx/nginx-kerberos-authentication/>

<https://www.dmosk.ru/miniinstruktions.php?mini=nginx-spnego>

<https://www.altlinux.org/%D0%A1%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5%20SPN%20%D0%B8%20Keytab%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0>

<https://www.altlinux.org/Nginx/AD-auth>

<https://techexpert.tips/nginx/nginx-pam-authentication/>

<https://ubuntu.com/blog/ubuntu-bionic-using-chrony-to-configure-ntp>

5.4 Настройка клиентов для Kerberos-аутентификации

Для включения Kerberos-аутентификации в браузере Internet Explorer 11 выполните:

1. Откройте **Свойства браузера** -> **Безопасность** -> **Местная интрасеть** (Local intranet), нажмите на кнопку **Сайты** -> **Дополнительно** (Рисунок 8).

Добавьте в зону следующие записи:

- https://*.int.bittechno.ru
- http://*.int.bittechno.ru

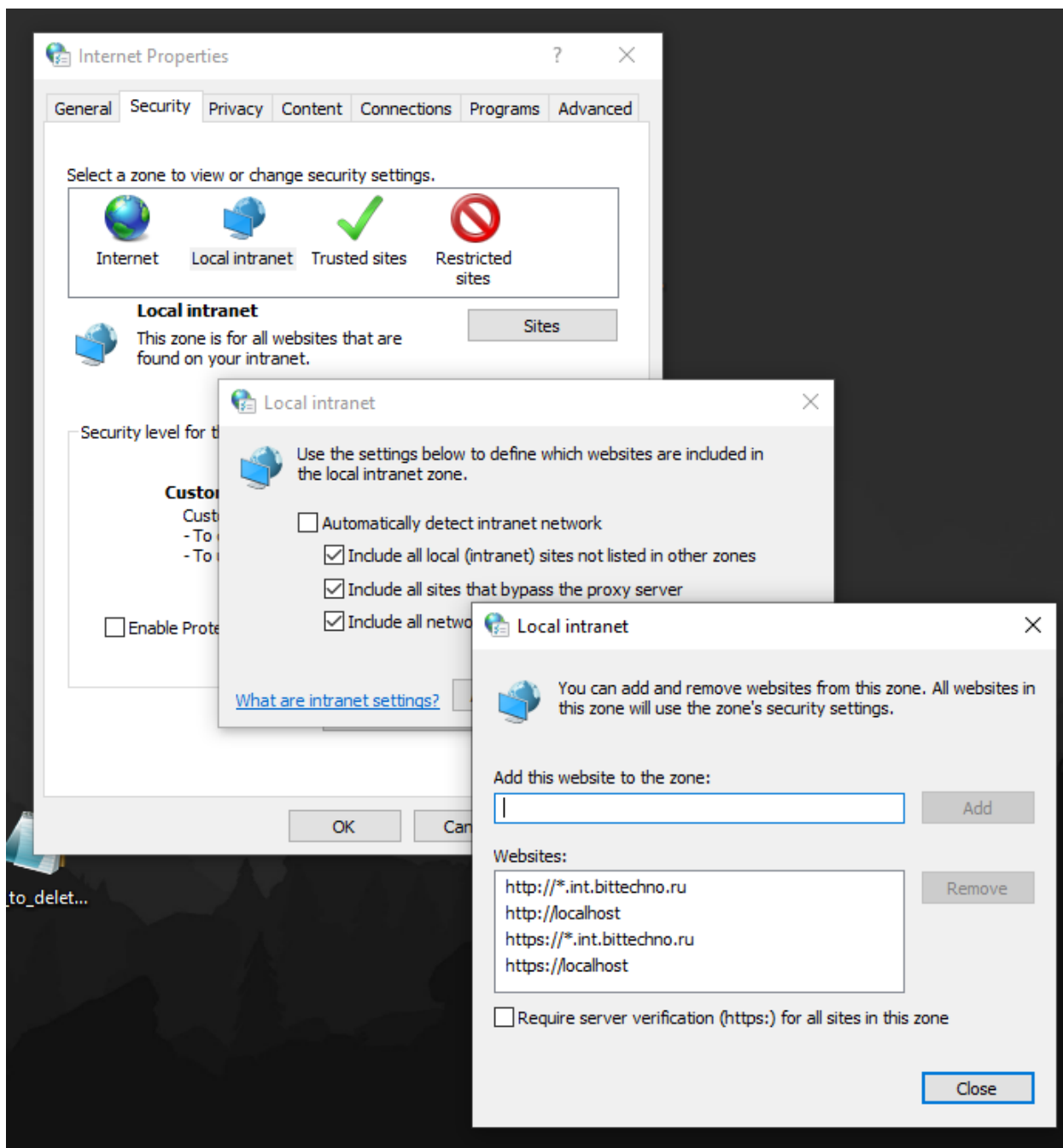


Рисунок 8 – Настройка зон безопасности в Internet Explorer

2. Далее перейдите на вкладку **Дополнительно** (Advanced) и в разделе **Безопасность** (Security) убедитесь, что включена опция **Разрешить встроенную проверку подлинности Windows** (Enable Integrated Windows Authentication).

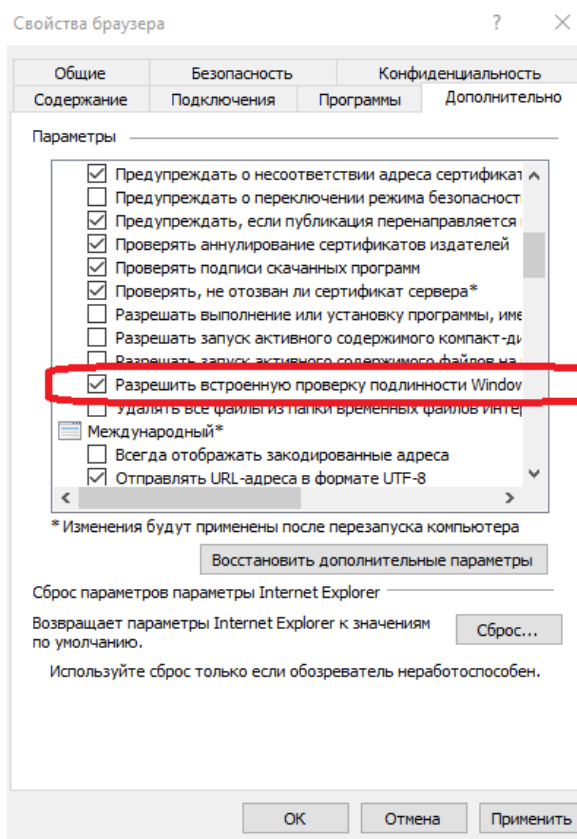


Рисунок 9 – Настройка зон безопасности в Internet Explorer

5.4.1 Настройка Kerberos-аутентификации в Google Chrome

Чтобы технология единого входа (SSO) работала в Google Chrome, нужно настроить Internet Explorer вышеописанным способом (Chrome испремяользует данные настройки IE). Кроме того, нужно отметить, что все новые версии Chrome автоматически определяют наличие поддержки Kerberos. В том случае если используется одна из старых версий Chrome (Chromium), для корректной авторизации на веб-серверах с помощью Kerberos, его нужно запустить с параметрами:

```
--auth-server-whitelist="*.int.bittechno.ru"
--auth-negotiate-delegate-whitelist="*.int.bittechno.ru"
```

Например:

```
"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --auth-server-
whitelist="*.int.bittechno.ru" --auth-negotiate-delegate-
whitelist="*.int.bittechno.ru"
```

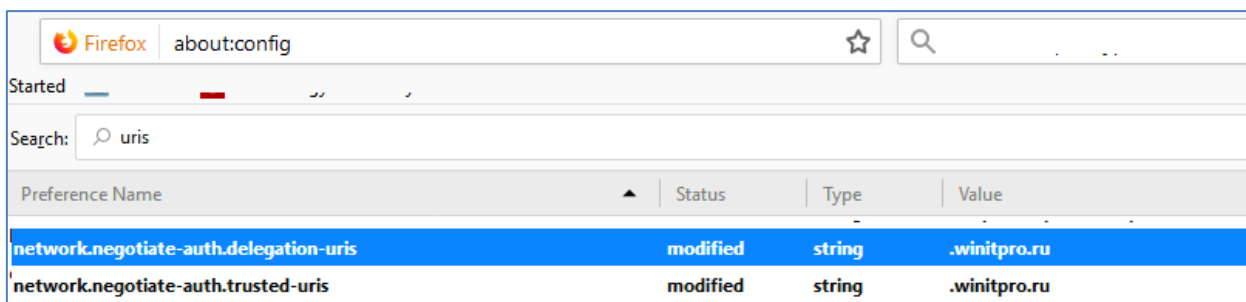
Либо эти параметры могут быть распространены через групповые политики для Chrome (политика AuthServerWhitelist) или строковый параметр реестра **AuthNegotiateDelegateWhitelist** (находится в каталоге HKLM\SOFTWARE\Policies\Google\Chrome).

Для вступления изменений в силу нужно перезагрузить браузер и сбросить тикеты Kerberos командой `klist purge` (см. [статью](#)).

5.4.2 Настройка Kerberos-аутентификации в Mozilla Firefox

По умолчанию поддержка Kerberos в Firefox отключена, для включения откройте окно конфигурации браузера (в адресной строке перейдите на адрес `about:config`). Затем в следующих параметрах укажите адреса веб-серверов, для которых должна использоваться Kerberos-аутентификация:

- `network.negotiate-auth.trusted-uris`
- `network.negotiate-auth.trusted-uris`



Preference Name	Status	Type	Value
<code>network.negotiate-auth.delegation-uris</code>	modified	string	<code>.winitpro.ru</code>
<code>network.negotiate-auth.trusted-uris</code>	modified	string	<code>.winitpro.ru</code>

Рисунок 10 – Настройка доступа в Mozilla Firefox

Для удобства, можно отключить обязательное указание FQDN адреса в адресной строке Mozilla Firefox, включив параметр **`network.negotiate-auth.allow-non-fqdn`**.

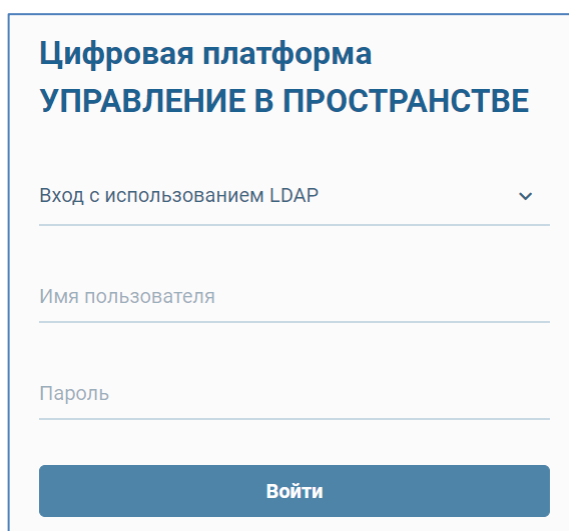
Проверить, что выполняется аутентификация на сервере с помощью Kerberos можно с помощью отладочного прокси-сервера **Fiddler** или команды: `klist tickets`.

5.5 Авторизация с помощью LDAP

В приложении доступна авторизация через протокол LDAP.

Для подключения необходимо:

1. В выпадающем списке выбрать «Вход с использованием LDAP» (Рисунок 11).
2. Ввести логин (без адреса сервера, он задается в конфигурационном файле системы) и пароль пользователя.
3. Нажать на кнопку «Войти».



Цифровая платформа
УПРАВЛЕНИЕ В ПРОСТРАНСТВЕ

Вход с использованием LDAP ▼

Имя пользователя

Пароль

Войти

Рисунок 11 – Вход с использованием LDAP

5.5.1 Настройки подключения

Для корректного подключения необходимо наличие группы в приложении, имеющей такое же название, как и группа в LDAP, к которой относится пользователь.

Группы пользователя, для которых нет аналогов в системе, будут пропущены.

Пользователь может входить в несколько групп одновременно.

Допустимо наличие пользователя без включения в группу.

В файле конфигурации, в разделе **Настройки авторизации и аутентификации** подраздела **Легковесный протокол доступа к каталогам Lightweight Directory Access Protocol (LDAP)** доступны следующие настройки:

- **ldap_allow_login** – флаг возможности авторизации через LDAP;
- **ldap_host** – адрес сервера LDAP. Можно указывать несколько серверов через разделитель ";" (точка с запятой);
- **ldap_port** – порт сервера LDAP, по умолчанию – 389;
- **ldap_use_ssl** – использование подключения к защищенному порту (подробная информация), по умолчанию – выключено;
- **ldap_timeout** – время ожидания для подключения. По умолчанию нет значения – не ограничено;
- **ldap_get_info** – получение информации о сервере (подробная информация);
- **ldap_client_strategy** – стратегия соединения с сервером (подробная информация);
- **ldap_authentication** – способ аутентификации (подробная информация);
- **ldap_user_host** – адрес сервера, на котором зарегистрирован пользователь;
- **ldap_domain_component** – адрес доменного имени (domain component, dc) для пользователя;
- **ldap_users_partition** – адрес до раздела с пользователями (если оставить пустой, то поиск будет происходить по всему домену);
- **ldap_username** – атрибут на сервере, отвечающий за логин пользователя. Поле обязательно к заполнению;
- **ldap_groups** – атрибут на сервере, отвечающий за группы пользователя. Поле обязательно к заполнению;
- **ldap_first_name** – атрибут на сервере, отвечающий за *имя* пользователя;
- **ldap_last_name** – атрибут на сервере, отвечающий за *фамилию* пользователя;
- **ldap_email** – атрибут на сервере, отвечающий за *электронную почту* пользователя;
- **ldap_sync_groups_with_ad** – флаг синхронизации групп при логине пользователя. По умолчанию – включено.

С включенным флагом при каждом подключении пользователя будут сопоставлены группы, которые есть у пользователя в Системе, и те, к которым он привязан в LDAP. Таким образом, если добавить пользователя в новую группу, не связанную с LDAP, то последующий перезаход в систему удалит эту группу у пользователя (так как в LDAP пользователь не состоит в этой группе).

- **ldap_default_group** – имя группы по умолчанию (группа к которой будет присоединен новый пользователь при выключенном флаге **ldap_sync_groups_with_ad**).

При отсутствии (или некорректной) группы в приложении вход через LDAP будет невозможен, и будет выдана ошибка с кодом ошибок 1522 и 1523.

5.5.2 Дополнительные ссылки

<https://ldap3.readthedocs.io/>

5.6 Анонимные (гостевые) пользователи

Анонимный (гостевой) пользователь – это пользователь, которому предоставлена временная возможность использовать систему с минимальными правами доступа.

Доступные действия:

- просмотр пресета;
- расчет события.

Все остальные действия в системе запрещены.

Получить права анонимного (гостевого) пользователя можно только при переходе по опубликованной ссылке приложения (публикация ссылки происходит на странице пресета, в пункте «Опубликовать пресет»).

Система проверит пользователя при переходе по короткой ссылке и, если пользователь не пройдет проверку, система создаст анонимного (гостевого) пользователя.

Пользователь может работать в системе до тех пор, пока активен токен (см. п. 5.8).

Информация об анонимном (гостевом) пользователе удалится при автоматической сборке мусора (см. п. 11.3).

5.7 Работа с паролями пользователей

При работе с паролями пользователей принимаются меры по предотвращению компрометации паролей и перехвата их злоумышленниками:

- введенный пользователем пароль не отображается в браузере (маскируется звездочками);
- при передаче пароля из веб-приложения на бэкенд для авторизации пользователя значение пароля шифруется, используется асимметричное шифрование ed25519 (схема цифровой подписи, использующая вариант схемы Шнора основанной на эллиптической кривой Эдвардса);
- при передаче пароля при изменении его пользователем или администратором значение пароля шифруется аналогично;
- пароли пользователей нигде не хранятся, вместо этого хранятся хэши паролей, вычисленные по безопасному алгоритму argon2.

Ключи шифрования хранятся в файле конфигурации, задаются в следующих параметрах:

- **back_private_key** – закрытый ключ бэкенда;
- **front_public_key** – открытый ключ фронтенда.

Аналогично в параметрах веб-приложения должны быть прописаны соответствующие части ключей шифрования.

Для генерации новых ключей шифрования можно использовать утилиту внутри проекта: **utils/ed25519_key_gen.py**, позволяющую вывести в консоль ключ шифрования на основе существующего файла ключа, или выполнить генерацию новой пары ключей в виде, подготовленном для добавления в файлы конфигурации.

Параметры команды (опциональные):

- `--private` – путь до файла, содержащего закрытый ключ;
- `--public` – путь до файла, содержащего открытый ключ;
- `--new` – флаг генерации новой пары ключей.

Файлы ключей могут быть сформированы, например, командой:

```
ssh-keygen -t ed25519
```

5.8 Токен

Токен – временный ключ, выдаваемый на время пользователю, для работы с системой.

5.8.1 Использование токенов доступа при работе с API

При авторизации пользователя с использованием пароля в ответе возвращается временный токен доступа, используемый в дальнейшем для доступа к API. Выданный токен привязан к пользователю, проверяя выданный ранее токен, приложение определяет пользователя, выполнившего запрос к API и его полномочия, при этом не требуется постоянно вводить и передавать логин и пароль. Время жизни токена доступа ограничено коротким интервалом времени, по окончании которого токен более не может быть использован, и потребуются повторная процедура авторизации. При активной работе пользователя токен автоматически продляется, что позволяет избежать лишних запросов на ввод пароля. При выходе пользователя из системы токен удаляется и не может быть использован для доступа к API.

При обращении к API токен доступа должен быть передан в заголовке запроса.

Authorization: Bearer

```
5dd3e16119a00e74ea838b692270ee32b68dd8292572c8df33137691a349f8cc
```

Время жизни токена определяется следующими параметрами конфигурации:

- `token_lifetime` – время жизни токена в сек.;
- `token_update_time` – интервал продления токена.

5.8.2 Передача токена в Cookies

В конфигурации приложения добавлен параметр, позволяющий передавать в запросах токен авторизации через Cookies. Имя параметра – **token_use_cookies**.

Если настройка для использования Cookies включена, выполняются следующие условия:

- Приложение будет использовать токен из поля **Authorization**, находящегося в Cookies.
- Эндпоинты аутентификации (`api/auth/sign-in` и `api/auth/login`) и эндпоинт создания пользовательской сессии (`api/auth/sessions`) не будут возвращать токен авторизации в теле сообщения.
- Эндпоинты аутентификации (`api/auth/sign-in` и `api/auth/login`) будут возвращать заголовок **Set-Cookie** с полем **Authorization**, хранящим токен авторизации, с флагами `secure=True` и `httponly=True`.
- Эндпоинт завершения сеанса (`api/auth/logout`) и эндпоинт открытия сессии (`api/auth/sessions`, если передать ему устаревший токен) будут удалять токен

авторизации в cookies клиента, возвращая заголовок **Set-Cookie** с пустым полем **Authorization** с датой экспирации 1970-01-01 и нулевым временем жизни.

Работа с сокетами при использовании cookies

При передаче сообщений в сокетах не требуется прикладывать токен авторизации в теле сообщения. Вместо этого будет использоваться токен, переданный при рукопожатии **handshake** с бэкендом (либо в заголовке **Authorization**, либо в cookies в поле **Authorization**, в зависимости от настройки).

Сокеты не будут работать до тех пор, пока у клиента не появится токен авторизации. Сам хендшейк **handshake** может быть выполнен, но любые попытки входа в неймспейсы/комнаты будут блокироваться бэкендом.

Работа с токенами при запуске приложения по HTTP/HTTPS

Если в приложении настроено использование cookies для передачи токена авторизации, приложение должно работать по протоколу HTTPS. Для токена в cookies задан флаг **Secure**, требующий передачи запросов только по HTTPS. Если приложение работает по HTTP, при включенной настройке cookies токен не будет отсылаться с клиентского приложения на бэкенд.

Таким образом:

- При работе по HTTPS приложение поддерживает передачу токена и через заголовок запроса, и через cookies;
- При использовании HTTP приложение работает только с токенами в заголовке запроса.

5.9 Внешняя авторизация (переход при выходе из приложения)


Для перехода на внешний сервис авторизации (редирект) при выходе из приложения, необходимо указать при запуске переменную окружения:

VISMIND_FRONTEND_AuthenticationExternal=http://servername

Если переменная не задана - выход будет происходить на страницу авторизации приложения.

6 Разграничение прав доступа

6.1 Пользователи

Для просмотра списка пользователей приложения необходимо открыть локальное меню в правом верхнем углу рабочей области, нажав на кнопку , расположенную рядом с именем пользователя, и выбрать пункт «Пользователи» (Рисунок 12).

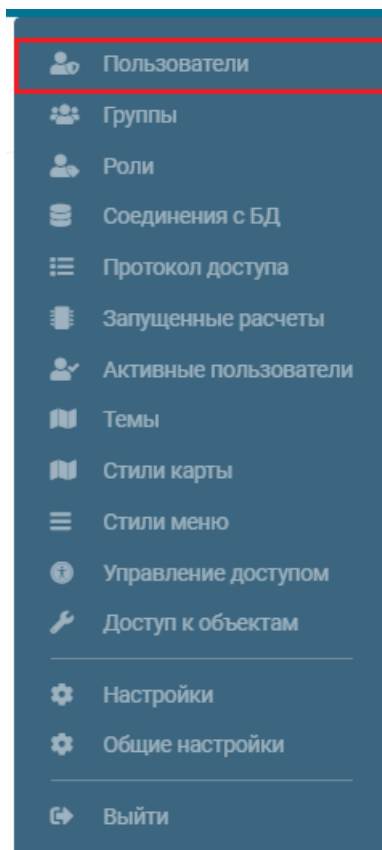


Рисунок 12 – Пункт меню «Пользователи»

Пункт меню будет доступен при наличии прав для просмотра списка пользователей.

6.1.1 Окно пользователей

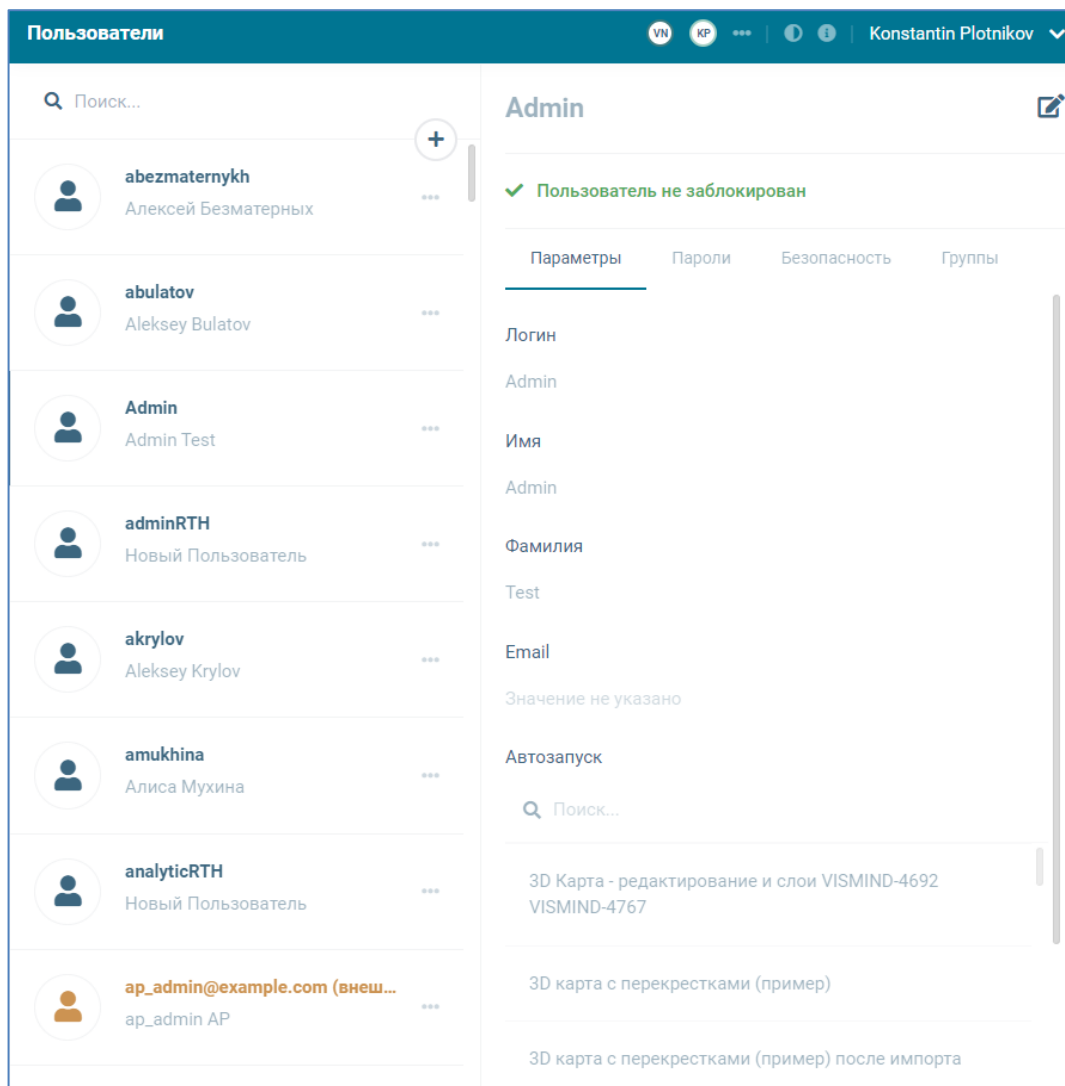




Рисунок 13 – Окно «Пользователи»

Окно «Пользователи» содержит:

- С правой стороны расположен список пользователей.
- С левой стороны – описание выбранного пользователя.
- Кнопка  позволяет добавить нового пользователя.
- Кнопка  позволяет отредактировать выбранного пользователя.

Цветовое разграничение пользователей:

- Желтый цвет – обозначает, что пользователь является внешним (см. п. 11.9.2):

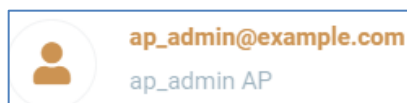


Рисунок 14 – Индикация внешнего пользователя

- Красный цвет – обозначает, что пользователь заблокирован.




Рисунок 15 – Индикация заблокированного пользователя


6.1.2 Окно настройки пользователя

Для просмотра информации о пользователях доступны 5 вкладок:

- **Параметры** (подробности в разделе 6.4 «Профиль пользователя»);
- **Пароли** (подробности в разделе 6.4 «Профиль пользователя»);
- **Безопасность** - настройки ограничения учетной записи:
 - Блокировать, если учетная запись не активна указанное число дней - количество дней, через которые пользователь будет заблокирован при отсутствии активности (отсутствие каких-либо действий в приложении). 0 - настройка не активна.
 - Число параллельных сеансов - раздел 11.6.5 «Ограничение числа одновременных сеансов пользователя».
 - Временная учетная запись - флаг активации временной учетной записи, которая доступна ограниченный период времени, а в остальное время является заблокированной.
 - Временной диапазон - диапазон активности временной учетной записи.
 - Минимальная длина пароля (в символах): число, от 2 до 20 включительно. По умолчанию минимальная длина не задана. Если настройка задана, она имеет более высокий приоритет над аналогичной глобальной настройкой.
- **Группы** - список групп, в которые входит пользователь. Не редактируется.
- **Автозапуск** - список задач, которые будут открыты при входе пользователя в систему.

Для редактирования свойств пользователя необходимо выбрать пользователя и нажать на кнопку редактирования .

6.2 Группы

Для просмотра списка групп приложения необходимо открыть локальное меню в правом верхнем углу рабочей области, нажав на кнопку , расположенную рядом с именем пользователя, и выбрать пункт «Группы» (Рисунок 16).

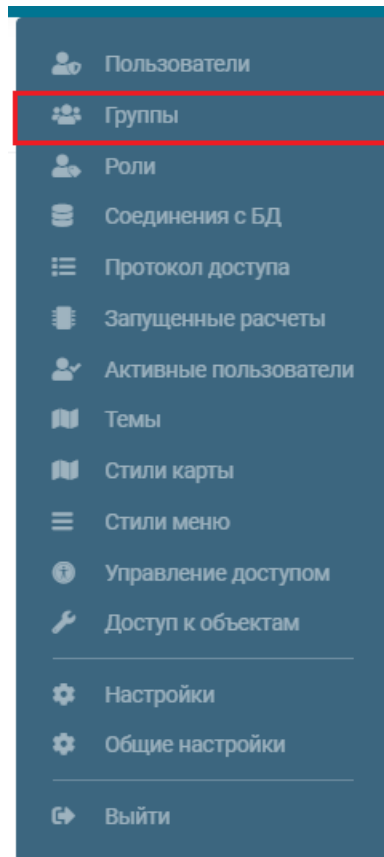


Рисунок 16 – Пункт меню «Группы»

Пункт меню будет доступен при наличии прав для просмотра списка групп.

6.2.1 Окно групп

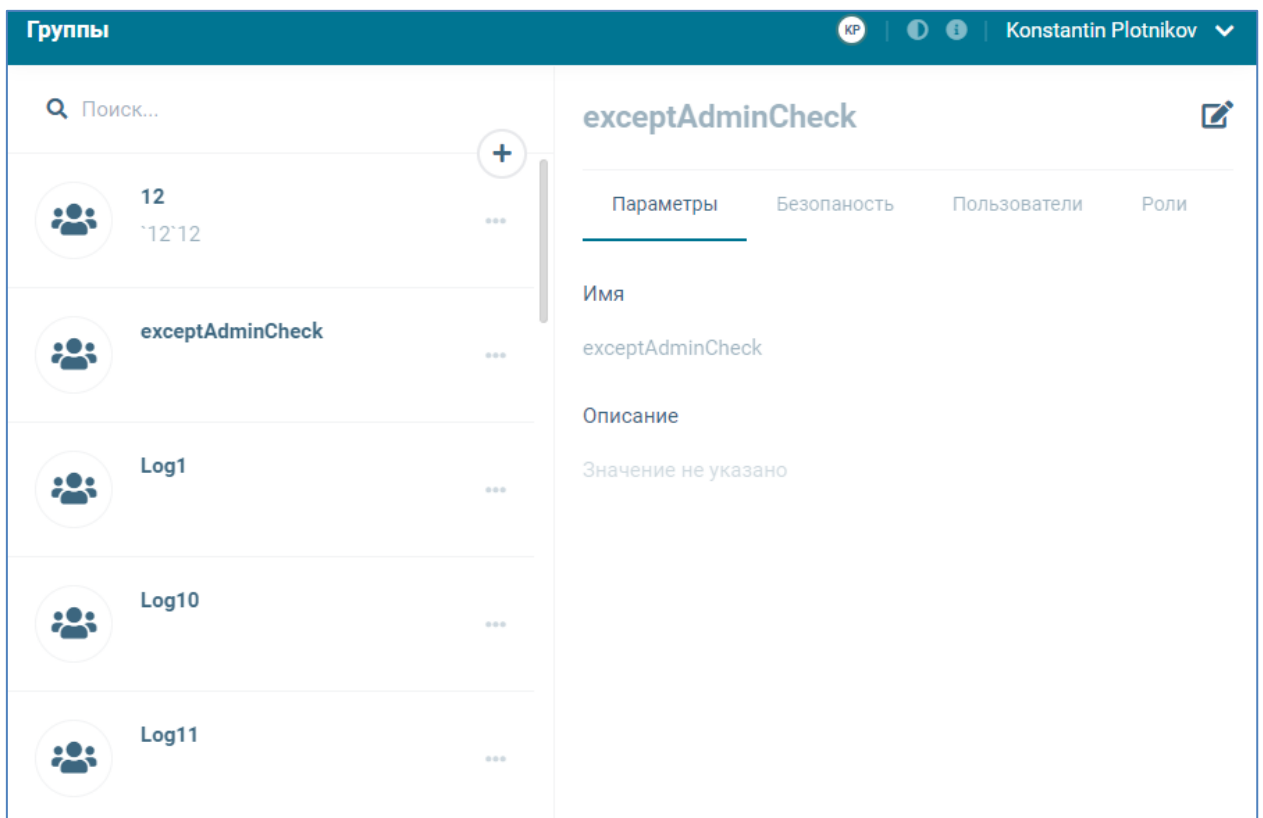








Рисунок 17 – Окно «Группы»


Окно «Группы» содержит:

- С правой стороны расположен список групп.
- С левой стороны – описание выбранной группы.
- Кнопка  позволяет добавить новую группу.
- Кнопка  позволяет отредактировать выбранную группу.


6.2.2 Окно настройки группы

Для просмотра информации о группе доступны четыре вкладки:

- **Параметры** – общая информация о группе:
 - **Имя** – наименование группы.
 - **Описание** – описание группы.
- **Безопасность** – настройки ограничения группы:
 - **Число параллельных сеансов** (см. п. 11.6.5).
- **Пользователи** – список пользователей, входящих в группу (см. п. 6.1).
 -  **Добавить пользователя** – открывает выпадающий список с пользователями для добавления в группу;
 -  **Удалить пользователей** – удаляет выбранных пользователей.
- **Роли** – список ролей, доступной данной группе (см. п. 6.3).
 -  **Назначить роль** – открывает выпадающий список с ролями для добавления в группу;
 -  **Удалить роли** – удаляет выбранные роли из группы.

Для редактирования свойств группы необходимо выбрать нужную вкладку и нажать на кнопку редактирования .

6.3 Роли

Для просмотра списка ролей приложения необходимо открыть локальное меню в правом верхнем углу рабочей области, нажав на кнопку , расположенную рядом с именем пользователя, и выбрать пункт «Роли» (Рисунок 18).

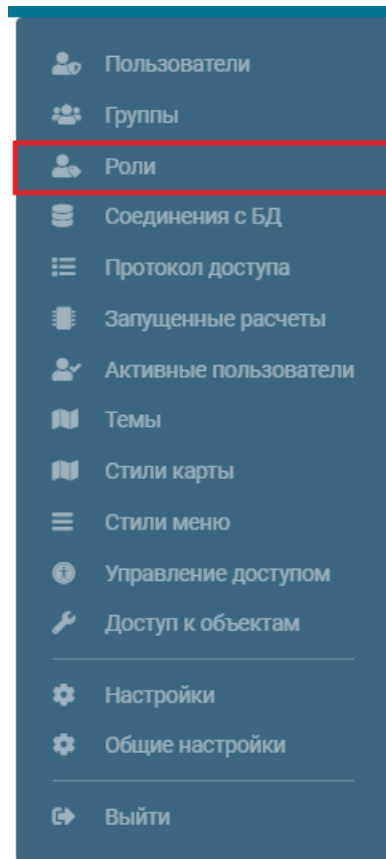


Рисунок 18 – Пункт меню «Роли»

Пункт меню будет доступен при наличии прав для просмотра ролей.

6.3.1 Окно ролей

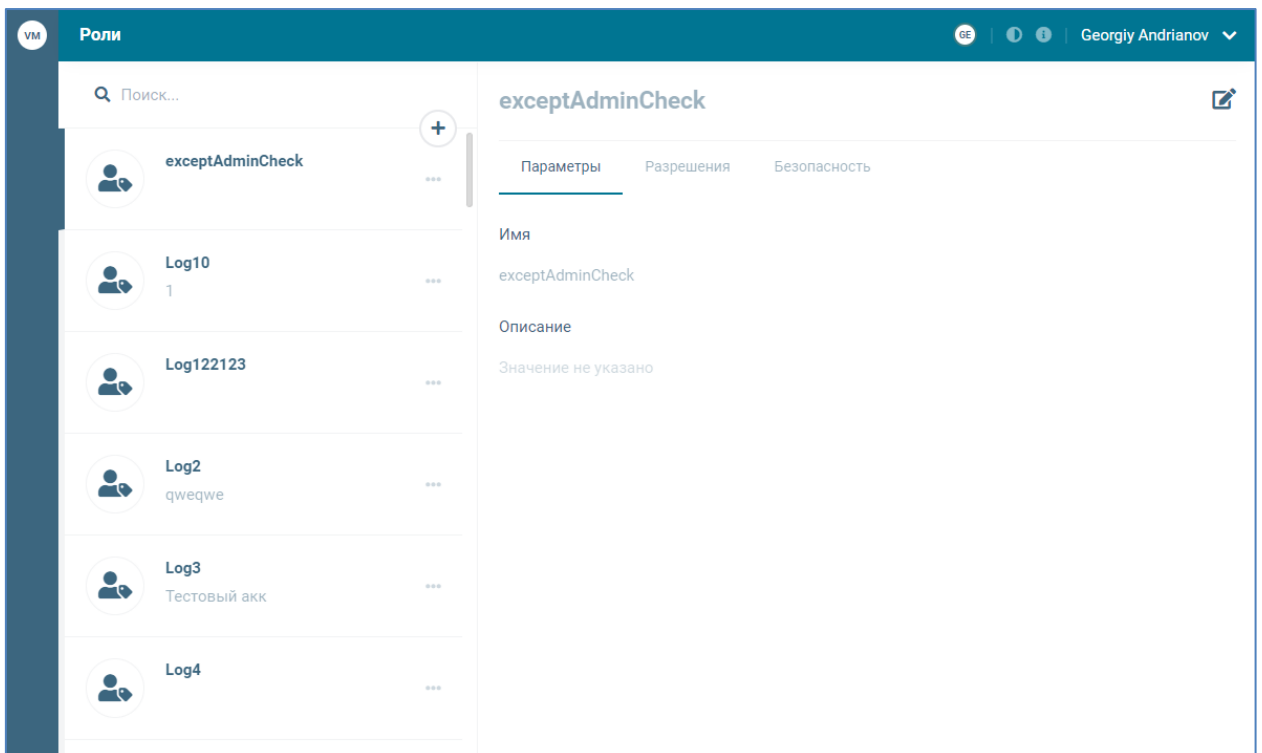




Рисунок 19 – Окно «Роли»


Окно «Роли» содержит:

- С правой стороны расположен список ролей.
- С левой стороны – описание выбранной роли.
- Кнопка  позволяет добавить новую роль.
- Кнопка  позволяет отредактировать выбранную роль.

6.3.2 Окно настройки роли

Для просмотра информации о группе доступны три вкладки:

- **Параметры** – общая информация о роли:
 - Имя – наименование роли.
 - Описание – описание роли.
- **Разрешения** – список разрешений для данной роли. Разрешения делятся на подгруппы:
 - Группа – разрешения для доступа к настройкам (см. п. 6.2).
 - Пользователь – разрешения для доступа к настройкам (см. п. 6.1).
 - Протокол доступа.
 - Расчеты задач – разрешения на просмотр списка расчетов задач.
 - Меню.
 - Приложение.
 - Темы.
 - Граф – разрешения на доступ к графу.
 - Пресет – разрешения на доступ к пресету.
 - Соединение с БД.
 - Карты.
 - Настройки.
 - Администрирование – настройки администрирования приложения.
- **Безопасность** – настройки ограничения роли:
 - Число параллельных сеансов (см. п. 11.6.5).

Для редактирования свойств роли необходимо выбрать нужную вкладку и нажать на кнопку редактирования .

6.4 Профиль пользователя

Для просмотра информации о пользователе доступны 3 вкладки:

- **Параметры** - общая информация о пользователе:
 - Логин - логин пользователя в системе. Не редактируется.
 - Имя - имя пользователя.
 - Фамилия - фамилия пользователя.

- Email - электронная почта пользователя.
- Дата рождения - дата рождения пользователя. Необязательный параметр.
- Использовать экспериментальные функции - включить некоторые экспериментальные функции по работе с пользователями.
- **Пароли** - при редактировании вкладки позволяет сменить пароль пользователя
- **Токен** - работа с API-токеном пользователя

6.5 Просмотр выданного доступа к объектам

В приложении реализована возможность просмотра всех эффективных разрешений, выданных пользователю на объекты.

Для просмотра выданного доступа к объектам необходимо открыть раздел меню «Доступ к объектам». Пункт меню доступен при наличии у пользователя роли «Редактирование доступа к объектам».

На странице «Доступ к объектам» (Рисунок 20) представлен список всех задач и пресетов репозитория, в том числе НСИ и Источники данных, доступных пользователю.

Для каждой колонки в таблице возможно использование сортировки и фильтрации.

№	Наименование	Тип	Мест
1	Маршруты (3d карта, темная)	Пресет задачи	копия Республика Татарстан - развитие инфраструк
2	Корреспонденции (3d карта, темная)	Задача	копия Республика Татарстан - развитие инфраструк
3	Новый пресет	Пресет НСИ	копия Республика Татарстан - развитие инфраструк
4	память	НСИ	копия Республика Татарстан - развитие инфраструк
5	Население (3d карта, светлая)	Пресет источника данных	копия Республика Татарстан - развитие инфраструк
6	Копия Республика Татарстан - развитие инфраструктуры	Источник данных	
7	Новый пресет	Пресет задачи	Задачи / Примеры реализации фич / импорт задач / Москва
8	Москва (Инциденты)	Задача	Задачи / Примеры реализации фич / импорт задач
9	Новый пресет	Пресет задачи	Задачи / Булатов / Требования v2
10	Требования v2	Задача	Задачи / Булатов
11	Абоненты В2С	Пресет задачи	Задачи / Примеры реализации фич / импорт задач / Копия К
12	Сборка блоков на сетке	Пресет задачи	Задачи / Примеры реализации фич / импорт задач / Копия К
13	Тесты блоков	Пресет задачи	Задачи / Примеры реализации фич / импорт задач / Копия К
14	Макет панели ЭРТХ	Пресет задачи	Задачи / Примеры реализации фич / импорт задач / Копия К
15	(Пилот) Макет панели ЭРТХ	Пресет задачи	Задачи / Примеры реализации фич / импорт задач / Копия К
16	Копия Копия Макет технической панели ЭР-Телеком (удаляются входы)	Задача	Задачи / Примеры реализации фич / импорт задач
17	Задача по сцеплению рядов	Пресет задачи	Задачи / (задание по практике) Прогнозы в р. Татарстан
18	Все значения	Пресет задачи	Задачи / (задание по практике) Прогнозы в р. Татарстан

Рисунок 20 – Окно «Доступ к объектам»

6.6 Выход из приложения

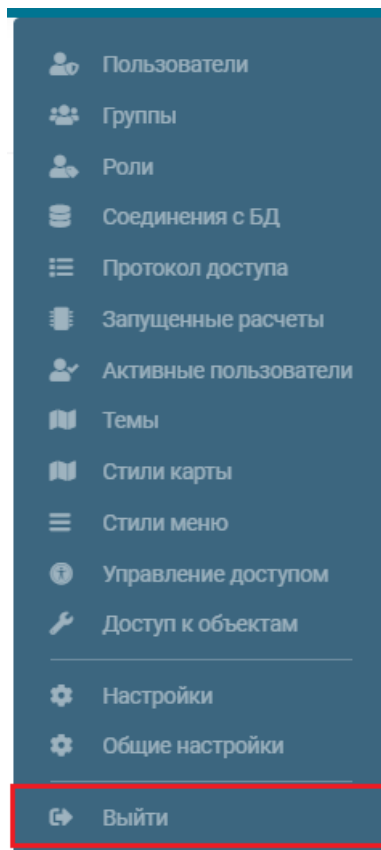


Рисунок 21 – Пункт меню «Выйти»

Для выхода из приложения необходимо воспользоваться пунктом меню «Выйти», расположенным в локальном меню приложения.

После выхода пользователь будет переадресован на страницу авторизации.

Аналогичным функционалом обладает адрес https://адрес_приложения/logout.

6.7 Активные пользователи

Активные пользователи – это пользователи, которые в данный момент работают в приложении.

Страница, отображающая список активных пользователей в приложении (Рисунок 22), содержит информацию:

- Активность – тип работы пользователя.
- Имя задачи – наименование задачи, с которой работает пользователь.
- ID задачи – идентификатор задачи, с которой работает пользователь.
- Имя блока – наименование блока, с которым работает пользователь.
- ID блока – идентификатор блока, с которым работает пользователь.
- Имя пресета – наименование пресета, с которым работает пользователь.
- ID пресета – идентификатор пресета, с которым работает пользователь.
- Пользователь – имя пользователя.
- IP пользователя – ip-адрес пользователя.

- Функция – последнее вызванное действие пользователя.
- Период неактивности – сколько времени прошло с действия пользователя.
- Процент расчета – процент расчета (если активность – это расчет задачи).

Активные пользователи

Отобразить 50 элементов

Поиск:

	Активность	Имя задачи	ID задачи	Имя блока
1	Preset	VISMIND-1978 питон	4bf34f8a-d553-4875-91aa-de2b41e44655	
2	Admin	Администрирование пользователей		
3	Preset	Фактические данные среднесрочного прогнозирования	55661808-564b-4a17-ab99-5d403c362c81	
4	Graph	Список задач		
5	Graph	Список задач		
6	Admin	Администрирование групп		
7	Graph	Sandbox	98092109-08c1-4556-a2d4-740661624cfe	
8	Graph	Sandbox	98092109-08c1-4556-a2d4-740661624cfe	e4621a72-2a0f-
9	Graph	VISMIND-1978 питон	4bf34f8a-d553-4875-91aa-de2b41e44655	

Страница 1 из 1

Рисунок 22 – Страница «Активные пользователи»

7 Настройка интерфейса

7.1 Стили и темы

В приложении реализована возможность выбора темы (цветового оформления интерфейса):

- Темная (по умолчанию);
- Светлая.

При открытии приложения по умолчанию используется «Темная тема», в дальнейшем пользователь может выбрать другую, используя кнопку «Сменить тему» на верхней панели (Рисунок 23).



Рисунок 23 – Кнопка «Сменить тему»

Выбранная тема (Рисунок 24, Рисунок 25) сохраняется в настройках приложения и будет использована при следующем открытии приложения.

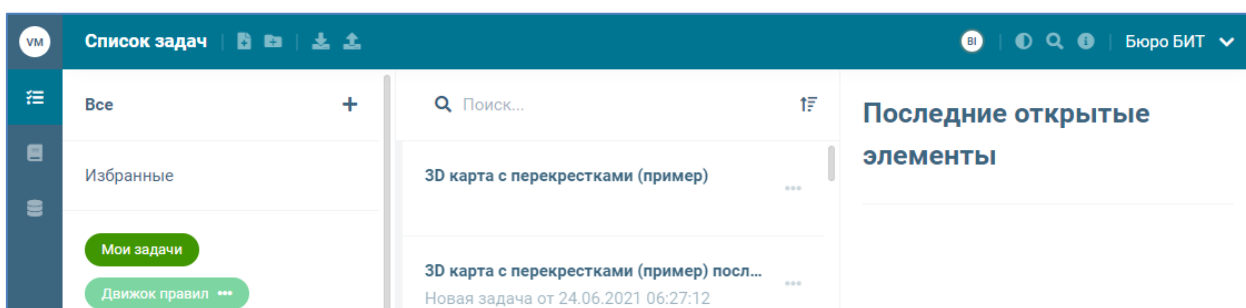


Рисунок 24 – Темная тема (по умолчанию)

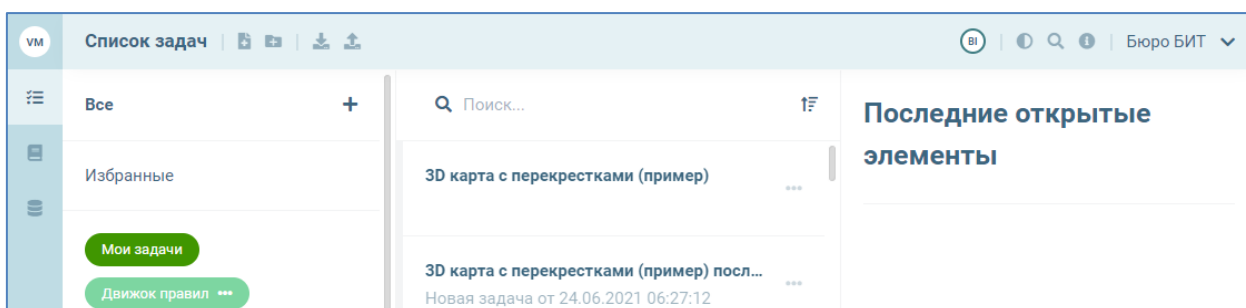


Рисунок 25 – Светлая тема

7.2 Стили меню

Стили меню – настройки оформления меню, с помощью которого осуществляется переключение между пресетами, при использовании блока «Меню» (Рисунок 26).

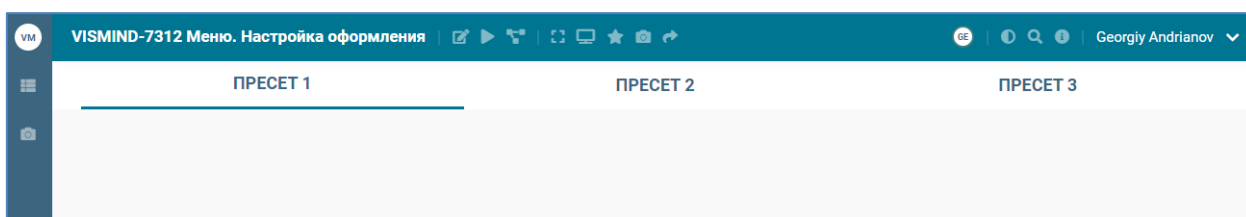


Рисунок 26 – Стили блока «Меню»

7.2.1 Настройка стилей

Для осуществления настроек необходимо открыть пункт меню «Стили меню» в локальном меню пользователя (Рисунок 27).

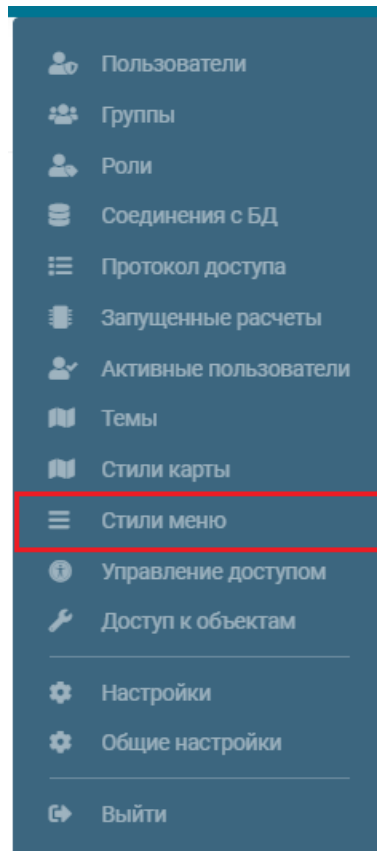


Рисунок 27 – Пункт меню «Стили меню»

Для редактирования стилей требуется отдельное разрешение. На странице «Стили меню» слева расположен список стилей. Если выделить стиль, то справа открываются его настройки (Рисунок 28).

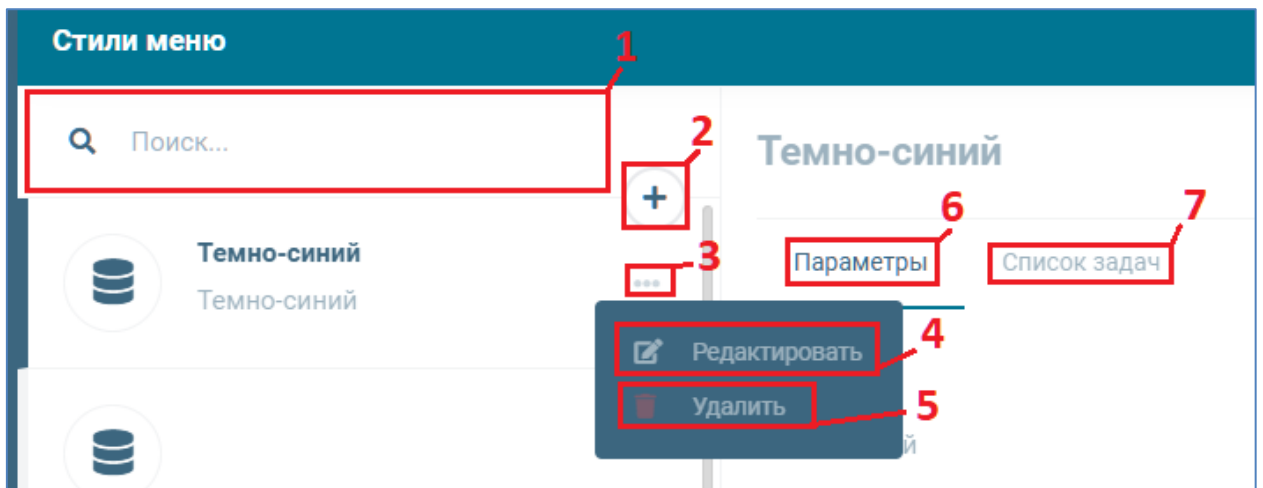



Рисунок 28 – Страница «Стили меню»

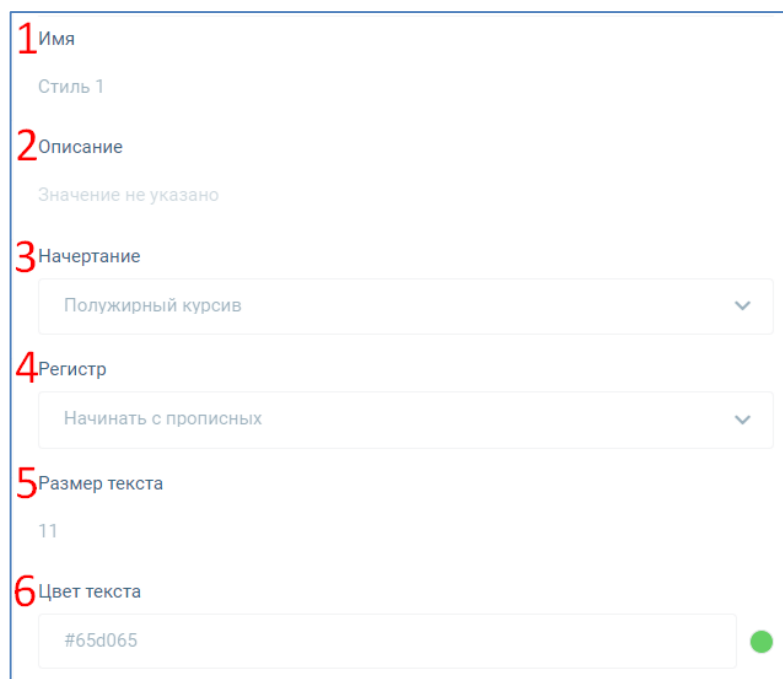
Страница «Стили меню» содержит:

1. Поиск стиля.
2. Создание нового стиля.

3. Открыть контекстное меню.
4. Редактировать стиль.
5. Удалить стиль.
6. Настройки выбранного стиля.
7. Список задач, в которых используется стиль.

Перечень доступных настроек стиля (Рисунок 29):

1. Имя.
2. Описание.
3. Начертание:
 - Обычный.
 - Курсив.
 - Полужирный.
 - Полужирный курсив.
4. Регистр:
 - Как в предложениях.
 - Все строчные.
 - Все прописные.
 - Начинать с прописных.
5. Размер текста (указать целое число).
6. Цвет текста (колорпикер , форматы HEX, RGBA).



1 Имя
Стиль 1

2 Описание
Значение не указано

3 Начертание
Полужирный курсив

4 Регистр
Начинать с прописных

5 Размер текста
11

6 Цвет текста
#65d065

Рисунок 29 – Окно настройки стиля

7.2.2 Применение настроенных стилей

Выбор настроенных стилей осуществляется в настройках блока «Меню» графа (Рисунок 30).

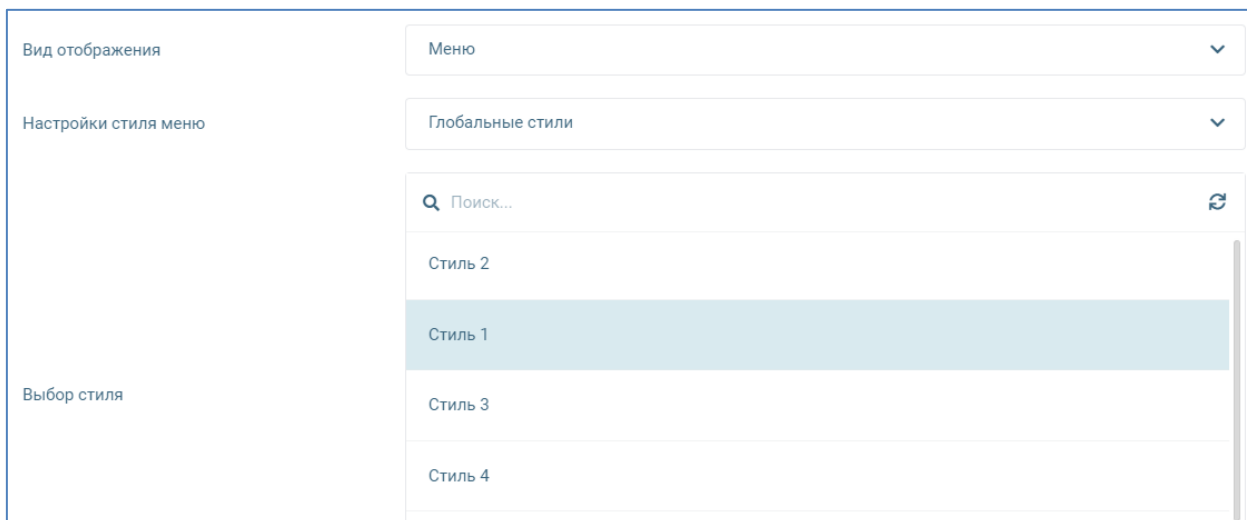


Рисунок 30 – Пример блока «Меню» графа

Для выбора настроенных стилей необходимо:

1. Открыть граф, в котором нужно настроить меню.
2. Открыть настройки блока «Меню». Блок нужно добавить из библиотеки блоков, если его еще нет в графе.
3. В пункте «Вид отображения» выбрать значение выпадающего списка «Меню».
4. В пункте «Настройки стиля меню» выбрать значение выпадающего списка «Глобальные стили».
5. В пункте «Выбор стиля» выбрать желаемый стиль.
6. Сохранить настройки блока.
7. Рассчитать блок.


Для просмотра результата открыть любой пресет данной задачи.

7.2.3 Глобальные стили карты

В приложении реализована возможность создать «Глобальный стиль» и использовать его для настройки отображения карт в любых задачах.

Доступ к разделу «Стили карты» должен быть разрешен для роли пользователя в разделе управления ролями (см. п. 6.3).

Для добавления нового стиля карты необходимо:

1. Открыть локальное меню в правом верхнем углу рабочей области, нажав на кнопку , расположенную рядом с именем пользователя (Рисунок 31).
2. Перейти в раздел «Стили карты», нажав на соответствующую кнопку.

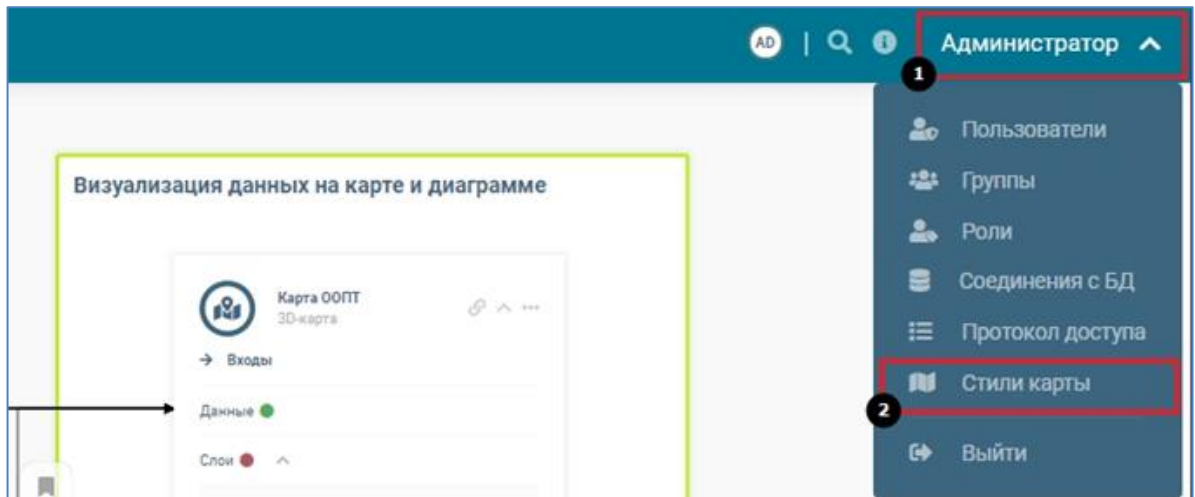



Рисунок 31 – Раздел меню «Стили карты»

3. На странице «Стили карты» создать новый стиль карты, нажав на кнопку .

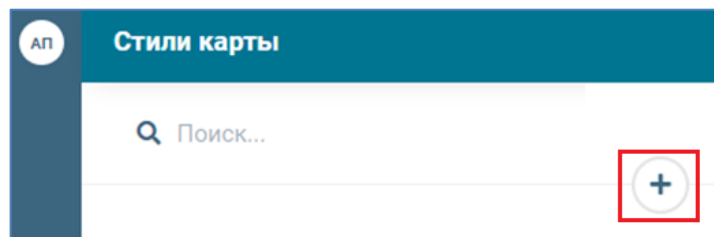


Рисунок 32 – Кнопка для создания стиля карты

4. Добавить имя и описание нового стиля карты, изменить параметры настройки стиля карты в формате json и сохранить изменения (Рисунок 33).

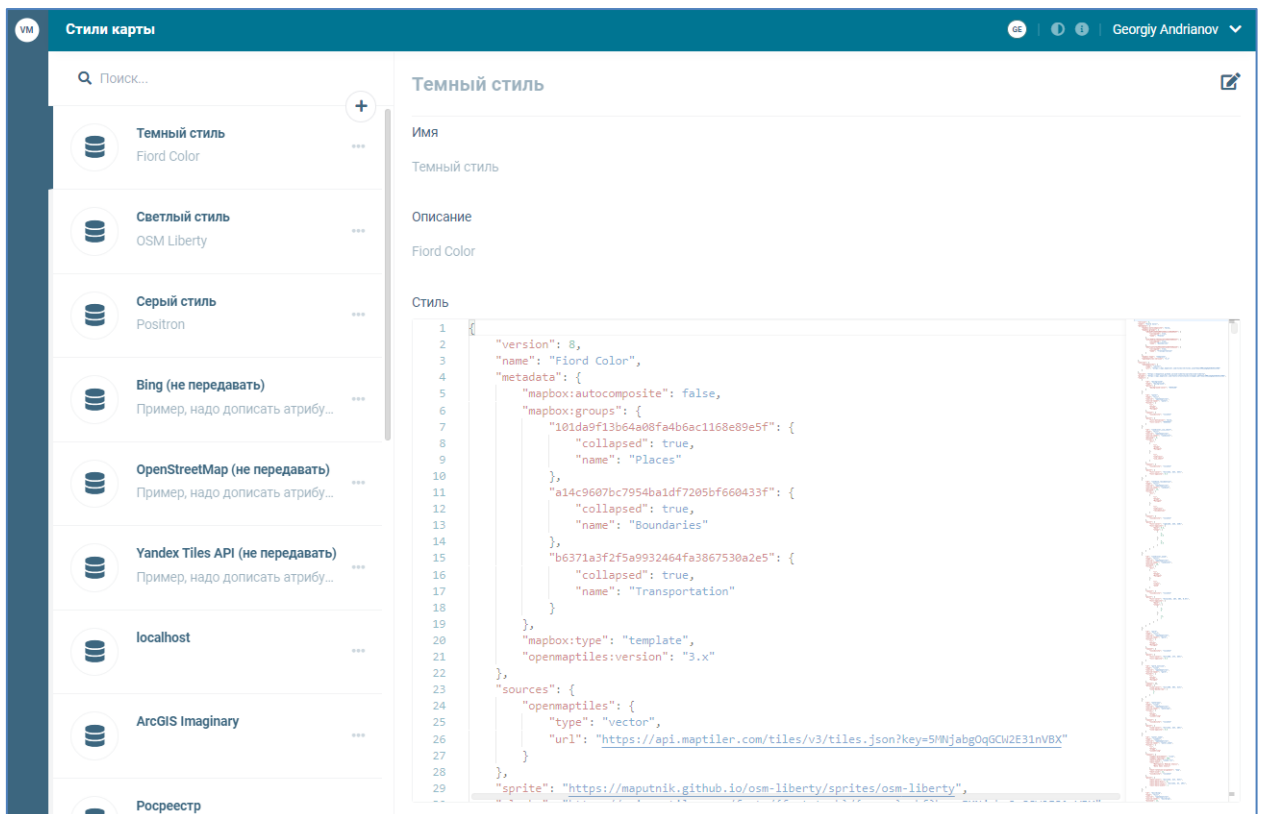


Рисунок 33 – Настройка нового стиля карты

5. После добавления и сохранения всех настроек новый стиль карты появится в списке стилей карты, и отобразится информационное сообщение об успешном создании нового элемента в правом верхнем углу рабочей области (Рисунок 34).

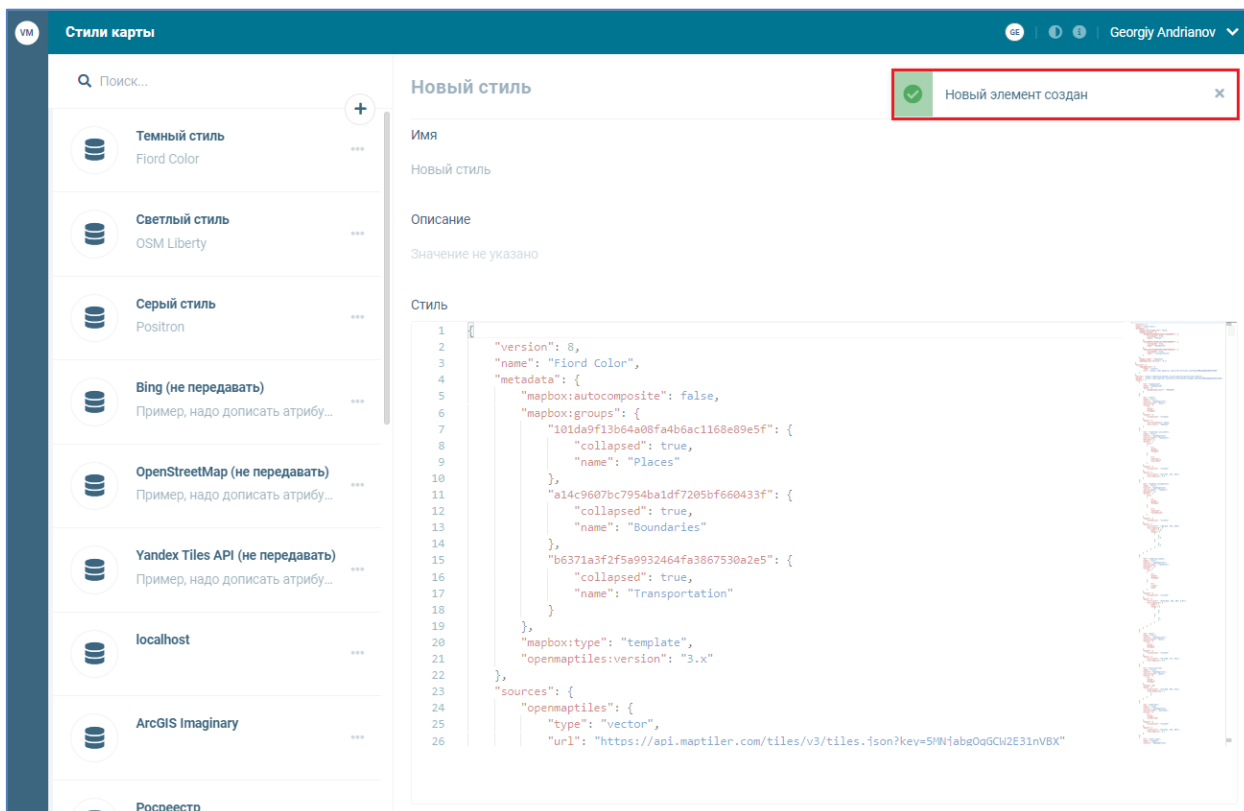


Рисунок 34 – Информационное сообщение «Новый элемент создан»

6. После успешного добавления нового стиля карты в настройках блока «3D-карта» необходимо:
- изменить настройки стиля карты на «Глобальные стили».
 - Имя - наименование стиля
 - Описание - описание стиля.
 - Стиль - настройки стиля карты в формате JSON.

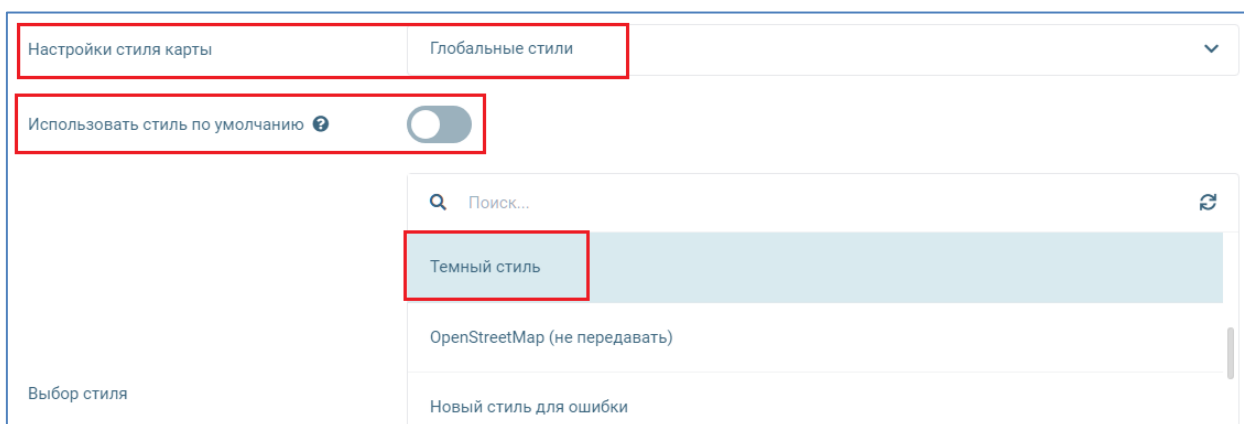



Рисунок 35 – Настройка блока «3D-карта»

7.3 Темы

Темы – цветовое оформление приложения.

7.3.1 Настройка тем

Для настройки темы необходимо перейти в меню настроек в правом верхнем углу  и в выпадающем меню выбрать пункт «Темы».

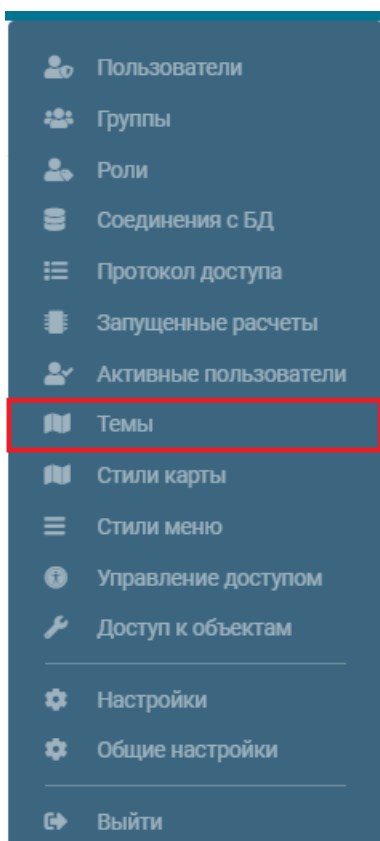



Рисунок 36 – Пункт меню «Темы»

Для редактирования тем требуется отдельное разрешение.

Страница «Темы» содержит (Рисунок 36):

1. Системные темы:
 - Темы «Темная» (по умолчанию) и «Светлая» являются системными. Их нельзя удалить, и они всегда есть в приложении.
2. Кнопка «Сменить тему»  выполняет переключение между системными темами.

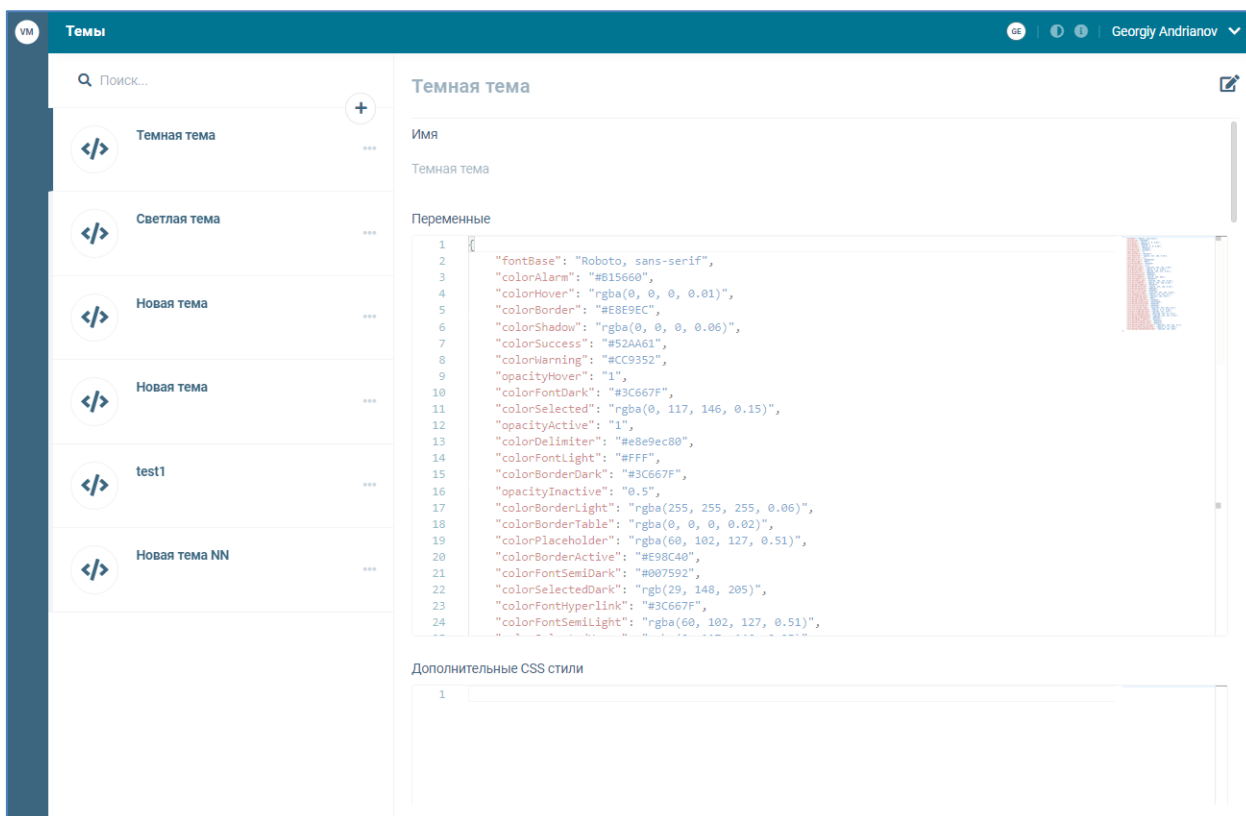


Рисунок 37 – Страница «Темы»

7.3.2 Параметры темы

При создании новой темы доступны следующие параметры:

- **Имя** – наименование будущей темы.
- **Алиас** – уникальный идентификатор темы.
- **Дополнительные CSS стили** – дополнительные настройки тем.
- **Стили визуализаторов** – стили для визуализаторов по умолчанию.

7.4 Редактор кода

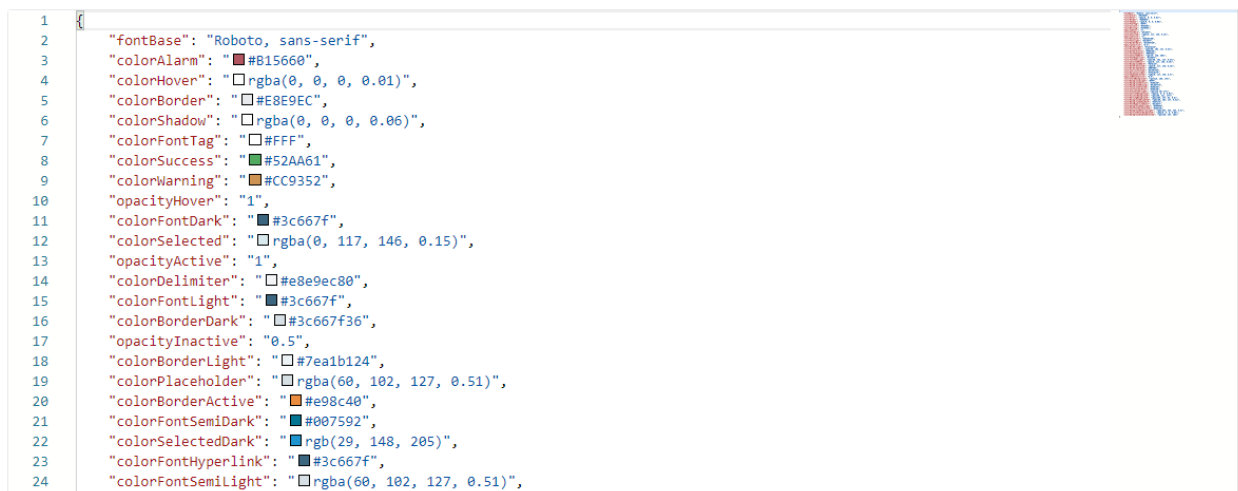



Рисунок 38 – Пример окна с редактированием кода

7.4.1 Настройки цвета

Редактор кода позволяет отображать и выбирать цвет


Если значение соответствует одному из поддерживаемых форматов цветов и указано внутри двойных кавычек, тогда слева от значения отобразится небольшой квадрат с данным цветом " #52AA61".

Поддерживаемые форматы:

- RGB/RGBA
- HEX
- HSL
- Name (Название цвета)

Перечень поддерживаемых названий цветов:

Название цвета	HEX код
black	#000000
silver	#C0C0C0
gray	#808080
white	#FFFFFF
maroon	#800000
red	#FF0000
purple	#800080
fuchsia	#FF00FF
green	#008000
lime	#00FF00
olive	#808000
yellow	#FFFF00
navy	#000080
blue	#0000FF
teal	#008080
aqua	#00FFFF

Чтобы изменить указанный цвет, необходимо навести курсор на значение  #52AA61", после чего появится окно для выбора цвета:

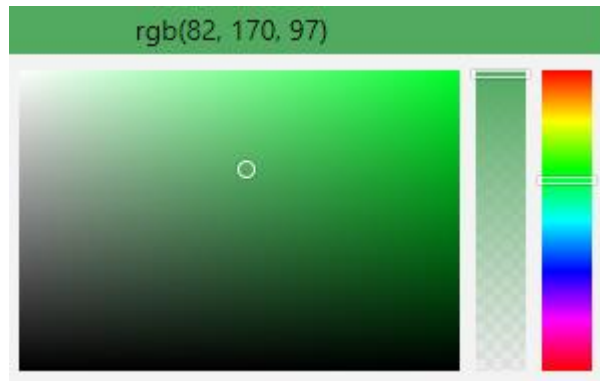


Рисунок 39 – Пример окна с редактированием цвета

Выбранный цвет будет в формате RGB или RGBA в зависимости от заданных настроек прозрачности.

Максимальная длина строки

Максимальная длина строки в редакторах кода установлена в 80 символов - отображается в редакторе как вертикальная черта.

7.5 Скрипты

В разделе хранятся все пользовательские скрипты, доступные для импорта в блоках Python через выражение `from vmscripts import module`. Все скрипты группируются по задачам, к которым они привязаны.

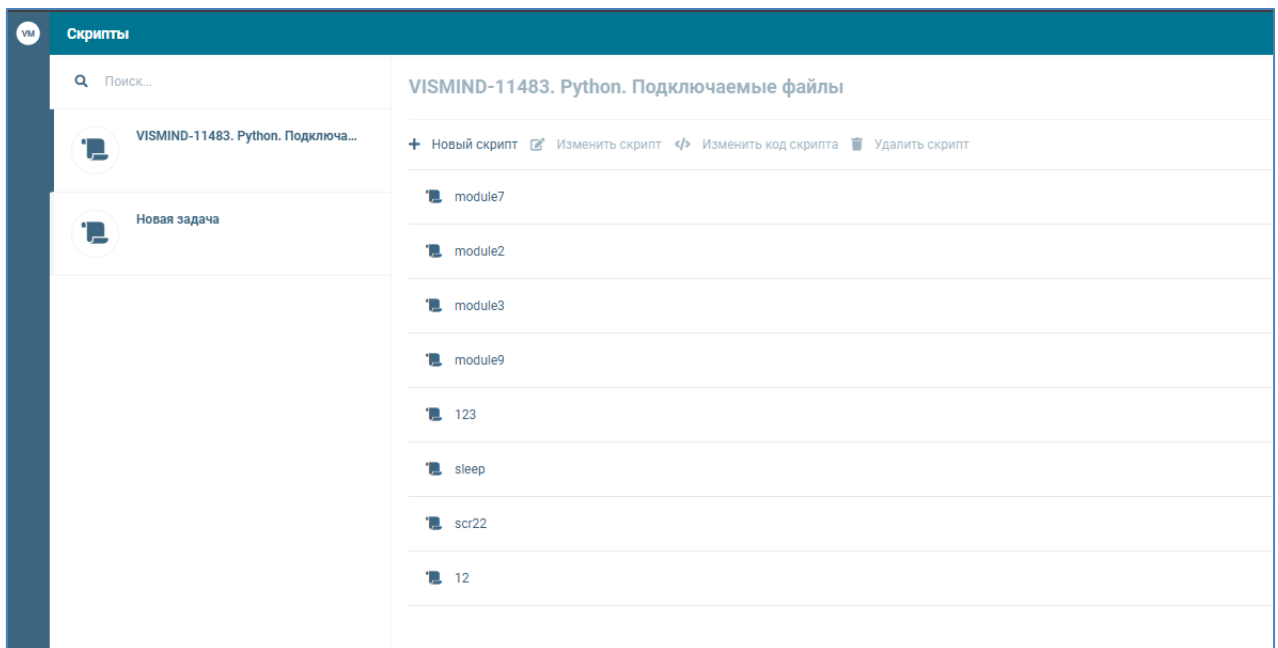


Рисунок 40 – Пример окна со скриптами

7.5.1 Создание скрипта

Чтобы создать скрипт для существующей задачи, необходимо выбрать нажать кнопку "Новый скрипт". Откроется модальное окно с двумя полями - названием скрипта и флагом глобального доступа. Название скрипта может состоять из латинских букв, символа нижнего подчеркивания и цифр. Название должно начинаться с буквы или символа

нижнего подчеркивания. Флаг глобального доступа позволит импортировать скрипт за пределами задачи, к которой он привязан.

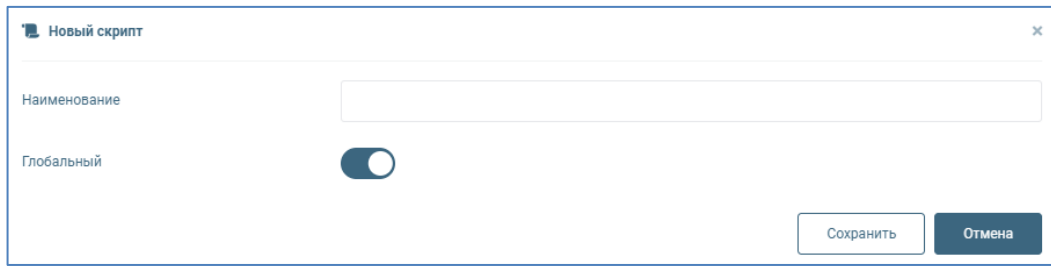


Рисунок 41 – Создание нового Скрипта

После нажатия кнопки "Сохранить" скрипт появится в списке скриптов задачи.

7.5.2 Редактирование скрипта

Чтобы отредактировать код, необходимо выбрать скрипт для редактирования и нажать кнопку "Изменить код скрипта". Откроется редактор кода, в котором можно вносить изменения в код.

Редактирование скрипта также возможно из всех графов приложения, где этот скрипт доступен.

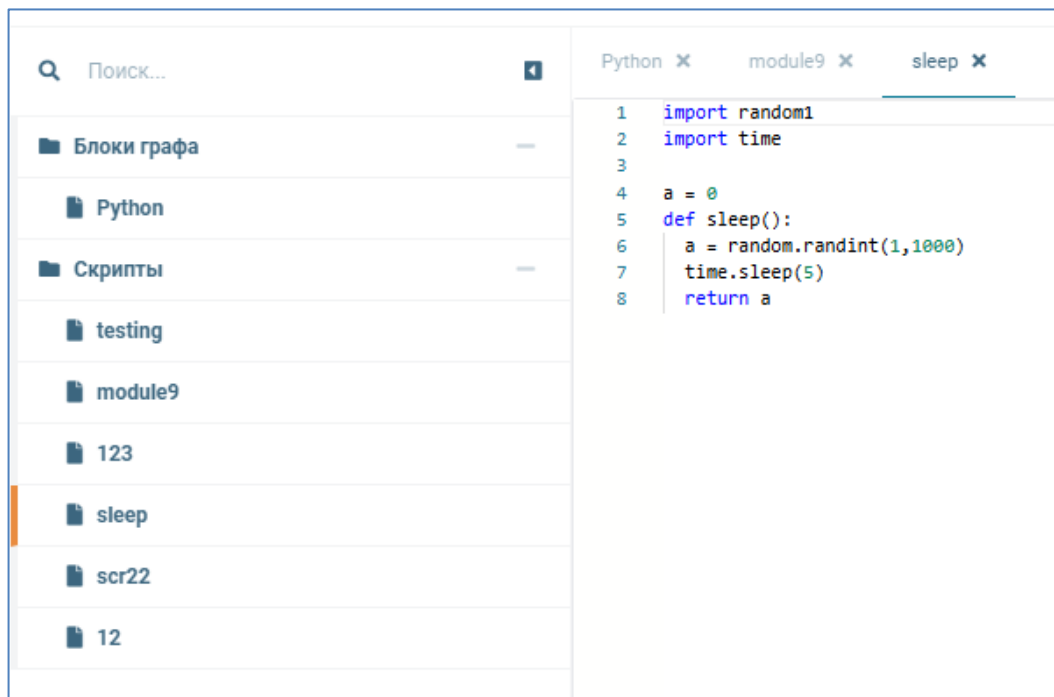


Рисунок 42 – Редактирование скрипта

7.5.3 Удаление скрипта

Чтобы удалить скрипт, нужно выбрать скрипт для удаления и нажать кнопку "Удалить скрипт".

8 Взаимодействие с внешними системами

8.1 Работа с приложением через IFrame

Аналитические панели ЦП УВП могут быть встроены как часть другого веб-приложения, располагаясь внутри фрейма (iframe). В этом случае в параметрах URL можно задать особое поведение при открытии ссылки:

- `embedded` – задает режим использования встраивания во фрейм, в этом режиме скрываются системные панели приложения и разрешено использование параметра `token`, по умолчанию «0» – выключено, для включения устанавливаем «1»;
- `token` – токен авторизации, работает только при задании `embedded =1`, позволяет передать в приложение полученный ранее токен авторизации, при его наличии и валидности не будет выводиться окно авторизации пользователя.

Пример встраивания приложения в коде страницы:

```
<iframe src="https://stage.int.bittechno.ru/analytics?task=8a531a45-bab4-4747-9251-044db3a882df&preset=0af128eb-7dbb-44f4-814a-7d2f5c9d11cb&embedded=1&token=5d7437a3716d348b7b494694e474cffff02cba3c2feeb02a07bc67c41df1671e" id="presetFrame"></iframe>
```

Для получения токена авторизации внешнее веб-приложение может использовать метод `/users/login` API бэкенда:

```
curl 'https://stage.int.bittechno.ru/api/users/login' \
  -H 'Accept: application/json, text/plain, */*' \
  -H 'Connection: keep-alive' \
  -H 'Content-Type: application/json' \
  --data-raw '{"user":"user","password":"password","ldap":0}'
```

Замечания по безопасности

Токен авторизации передается в адресной строке в открытом виде.

При авторизации с использованием метода `/users/api` пароль пользователя передается в открытом виде.

9 Настройка расчетов

9.1 Получение пользовательского токена для расчетов

Приложение допускает взаимодействие с собой через API. При этом для работы необходимо в разделе настроек пользователя получить специальный ключ (токен), который позволит работать с приложением (Рисунок 43).

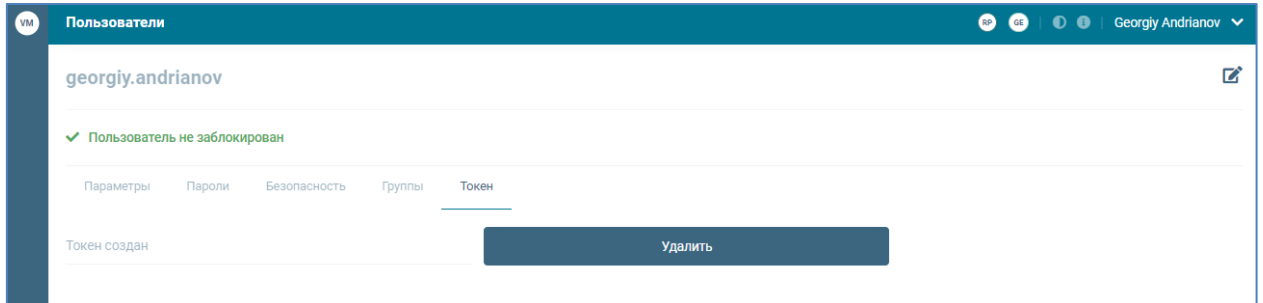


Рисунок 43 – Страница «Пользователи»

Токен можно получить любому пользователю, если у него есть соответствующие разрешения. При этом запуск на расчет самой задачи будет доступен, только если у пользователя есть на это разрешения (доступ к расчету и доступ к самой задаче).

Пользователю доступен только один токен одновременно.

Токен можно просмотреть и скопировать только в момент получения. После этого его можно только удалить.

9.2 Удаление токена

Для удаления токена необходимо нажать кнопку «Удалить». После этого выданный ранее токен перестанет работать.

9.3 Запуск расчета через API

Расчет графа / блока / цепочки блоков доступен через API с использованием пользовательского токена для расчета. Токен для расчета задач – это бессрочный токен, который выдается пользователю только для расчета задач (в основном, чтобы использовать его для расчета задач по расписанию).

Для запуска расчета необходимо отправить GET запрос на API `http://application/api/calculate?token={token}`. В теле запроса BODY нужно передать следующие параметры:

- token – токен для расчета задач;
- task – id задачи;
- block – id блока;
- branch – рассчитать ветку, начиная с блока block. Данный параметр должен быть в паре с параметром block (branch=1);
- async – синхронный/асинхронный режим (по умолчанию используется асинхронный режим).

Ответы с параметром **async** отличаются в том, что при асинхронном режиме возвращается ссылка для опроса расчета, а при синхронном сразу вернется результат расчета.

Примеры работы с API:

1. Запуск расчета **задачи**:

- обязательные параметры – token и task;
- опциональный параметр – async.

```
api/calculate?token=2186aeef3cd340e9ebca69d358acc2e336b5ee781ff2439f636154acc2c9a67d&task=86dfbce7-6c83-42b3-b19c-725c8f29a273
```

Ответ:

```
{"Code":0,"Info":"","Body":{"location":"/api/v1/tasks/86dfbce7-6c83-42b3-b19c-725c8f29a273/calc/c11b1532-020f-4265-8413-088c461a917d"},"Path":""}
```

```
api/calculate?token=2186aeef3cd340e9ebca69d358acc2e336b5ee781ff2439f636154acc2c9a67d&task=86dfbce7-6c83-42b3-b19c-725c8f29a273&async=0
```

Ответ:

```
{"Code":0,"Info":"","Body":{"log":[]},"Path":""}
```

2. Запуск расчета **блока**:

- обязательные параметры – token, task и block;
- опциональный параметр – async.

```
api/calculate?token=2186aeef3cd340e9ebca69d358acc2e336b5ee781ff2439f636154acc2c9a67d&task=86dfbce7-6c83-42b3-b19c-725c8f29a273&block=b861fc1e-fb4a-43b9-8d46-3fc871115405
```

Ответ:

```
{"Code":0,"Info":"","Body":{"location":"/api/v1/tasks/86dfbce7-6c83-42b3-b19c-725c8f29a273/calc/c11b1532-020f-4265-8413-088c461a917d"},"Path":""}
```

```
api/calculate?token=2186aeef3cd340e9ebca69d358acc2e336b5ee781ff2439f636154acc2c9a67d&task=86dfbce7-6c83-42b3-b19c-725c8f29a273&block=b861fc1e-fb4a-43b9-8d46-3fc871115405&async=0
```

Ответ:

```
{"Code":0,"Info":"","Body":{"log":[]},"Path":""}
```

3. Запуск расчета **ветки** графа:

- обязательные параметры – token, task, block и branch;
- опциональный параметр – async.

```
api/calculate?token=2186aeef3cd340e9ebca69d358acc2e336b5ee781ff2439f636154acc2c9a67d&task=86dfbce7-6c83-42b3-b19c-725c8f29a273&block=b861fc1e-fb4a-43b9-8d46-3fc871115405&branch=1
```

Ответ:

```
{"Code":0,"Info":"","Body":{"location":"/api/v1/tasks/86dfbce7-6c83-42b3-b19c-725c8f29a273/calc/c11b1532-020f-4265-8413-088c461a917d"},"Path":""}
```

```
api/calculate?token=2186aeef3cd340e9ebca69d358acc2e336b5ee781ff2439f636154acc2c9a67d&task=86dfbce7-6c83-42b3-b19c-725c8f29a273&block=b861fc1e-fb4a-43b9-8d46-3fc871115405&branch=1&async=0
```

Ответ:

```
{"Code":0,"Info":"","Body":[],"Path":""}
```

9.4 Получение результатов расчета блока через API

Получение данных с входов, выходов, а также лога расчета, по одному блоку с помощью токена для расчетов.

Обращение должно выполняться с помощью метода POST на специальное API: `http://application/api/calculate/result`.

Параметры BODY:

- token – пользовательский токен для расчетов;
- task_id – идентификатор задачи;
- block_id – идентификатор блока;
- filter – фильтр получения результатов (необязательный параметр);
 - input – получить только входы блока;
 - output – получить только выходы блока;
 - log – получить только лог расчета блока.

Получение результатов для блоков, которые рассчитываются после события, возможно только для событий, вызванных пользователем с токеном. Результаты расчетов других пользователей через API получить нельзя.

Пример BODY:

Настройки body:

```
{ "task_id": "c553e35e-8776-4ca2-a0d2-8c27c5afaea9", "block_id": "74521857-5795-4d0d-92d2-339cdbb817e0", "token": "2186aeeef3cd340e9ebca69d358acc2e336b5ee781ff2439f636154acc2c9a67d", "filter": "output" }
```

Пример ответа (данные возвращаются во внутренних структурах приложения):

Ответ:

```
{
  "Code": 0,
  "Info": "",
  "Body": {
    "input": [],
    "output": [
      {
        "id": "ts",
        "desc": "",
        "name": "Ряды данных",
        "type": "series",
        "array": 1,
        "mandatory": 1,
        "val": [
          {
            "rkey": {
              "ind": "Индекс физического объема валового регионального продукта, % к предыдущему году",
              "region_id": 94,
              "datasource": "Все данные",
              "date_level": 1
            },

```

```

"keys": [
  "01.01.2004",
  "01.01.2005",
  "01.01.2006",
  "01.01.2007",
  "01.01.2008",
  "01.01.2009",
  "01.01.2010",
  "01.01.2011",
  "01.01.2012",
  "01.01.2013",
  "01.01.2014",
  "01.01.2015",
  "01.01.2016",
  "01.01.2017",
  "01.01.2018",
  "01.01.2019"
],
"v1": [
  113.2,
  106.9,
  112.2,
  112.4,
  110.2,
  89.2,
  106.4,
  106.8,
  102.5,
  102.9,
  103.2,
  104.0,
  103.2,
  102.4,
  102.0,
  103.1
],
"attrs": {
  "model_name": "Факт",
  "model_ts_name": "Индекс физического объема валового регионального продукта, % к предыдущему году - Факт",
  "order_index": 1
},
"_source": "bdb98be7-6690-449f-8963-31f4dbdd34f8",
"_params": {
  "ts": [
    {
      "key": "data",
      "mask": "%d.%m.%Y",
      "frequency": "Y"
    }
  ],
  "v1": "value",
  "v1_type": 4,
  "lazy": [],
  "fkey": [

```

```

        "data"
    ]
}
},
{
    "pkey": {
        "ind": "Индекс физического объема валового
регионального продукта, % к предыдущему году",
        "region_id": 94,
        "datasource": "Все данные",
        "date_level": 1
    },
    "keys": [
        "01.01.2005",
        "01.01.2006",
        "01.01.2007",
        "01.01.2008",
        "01.01.2009",
        "01.01.2010",
        "01.01.2011",
        "01.01.2012",
        "01.01.2013",
        "01.01.2014",
        "01.01.2015",
        "01.01.2016",
        "01.01.2017",
        "01.01.2018",
        "01.01.2019"
    ],
    "v1": [
        108.698902356683,
        112.502579558463,
        112.352433137157,
        109.115299282301,
        90.4375245862415,
        106.789523575674,
        105.213190587568,
        103.710788767414,
        104.208332138152,
        103.476860697072,
        104.506739500737,
        101.258359959271,
        101.829134303034,
        102.341639491086,
        100.958692059145
    ],
    "attrs": {
        "model_name": "Модель",
        "model_ts_name": "Индекс физического объема
валового регионального продукта, % к предыдущему году - Модель",
        "order_index": 2
    },
    "_source": "ba9a3ae4-9cf5-4f55-9087-37551aa16b07",
    "_params": {
        "ts": [

```

```

    {
      "key": "data",
      "mask": "%d.%m.%Y",
      "frequency": "Y"
    }
  ],
  "v1": "value",
  "v1_type": 4,
  "lazy": [],
  "fkey": [
    "data"
  ]
}
},
{
  "pkey": {
    "ind": "Индекс физического объема валового
регионального продукта, % к предыдущему году",
    "region_id": 94,
    "datasource": "Все данные",
    "date_level": 1
  },
  "keys": [
    "01.01.2019",
    "01.01.2020",
    "01.01.2021",
    "01.01.2022",
    "01.01.2023",
    "01.01.2024",
    "01.01.2025"
  ],
  "v1": [
    100.958692059145,
    94.5755800494571,
    102.191627916015,
    102.738841560607,
    101.530720505869,
    102.304618016907,
    null
  ],
  "attrs": {
    "model_name": "Прогноз",
    "model_ts_name": "Индекс физического объема
валового регионального продукта, % к предыдущему году - Прогноз",
    "order_index": 3
  },
  "_source": "b9c60cc8-caeb-42cf-84d5-0c7fa26f8900",
  "_params": {
    "ts": [
      {
        "key": "data",
        "mask": "%d.%m.%Y",
        "frequency": "Y"
      }
    ]
  },
  ],

```

```

        "v1": "value",
        "v1_type": 4,
        "lazy": [],
        "fkey": [
            "data"
        ]
    }
}
]
}
],
"log": [],
"iterations": []
},
"Path": "",
"DeveloperInfo": ""
}

```

9.5 Запуск расчетов на нескольких узлах

При необходимости горизонтального масштабирования приложения можно перевести в режим работы с внешними расчетчиками (воркерами), которые будут заниматься расчетами графов. При этом их количество теоретически не ограничено.

Приложение в части бэкенда может работать в нескольких режимах:

1. Режим в виде координатора **Coordinator**.
2. Режим в виде расчетчика **Worker**.
3. Полный режим **ALL** (включает в себя пункты 2 и 3).

9.6 Режимы работы

9.6.1 Режим координатора

Приложение обрабатывает все запросы от клиентской части. Запросы, которые относятся к расчетам, складываются в очередь сообщений. Приложение, работающее в данном режиме, называется **координатор**.

9.6.2 Режим расчетчика

Рассчитывает задачи из очереди, в которую складывает задачи координатор. Не взаимодействует с пользовательским приложением. Приложение, работающее в данном режиме, называется **воркер (worker или расчетчик)**.

У координатора может быть несколько воркеров.

9.7 Установка

Для перевода приложения из одного режима в другой необходимо в настройках приложения указать тип запуска в конфигурации приложения (или через переменные окружения):

```

type_run = "ALL"
# Тип запуска, возможные значения:
#   Coordinator
#   Worker
#   ALL

```

Важно! Если у координатора не будет ни одного **воркера** – он перейдет в режим расчетов, аналогичный режиму полный **ALL**.

При этом для режима **координатора** и **воркера** необходимо указывать адрес очереди сообщения RabbitMQ.

9.7.1 Ресурсы

При запуске приложения в режиме координатора с использованием воркеров можно ограничить использование ресурсов средствами **docker**, ведь каждый экземпляр приложения запускается в виде отдельного контейнера. При этом воркеры могут располагаться на любой машине, не только на той, на которой расположено основное приложение.

Если ресурса **воркера** не хватает на выполнение задачи, он перезагружается и передает расчет следующему **воркеру**.

Если по какой-то причине, воркеры **три раза** не смогли рассчитать задачу – она переводится в режим «Запрещена для расчета» (Рисунок 44) и может быть возвращена обратно только ручным переключением этой опции в настройке графа.

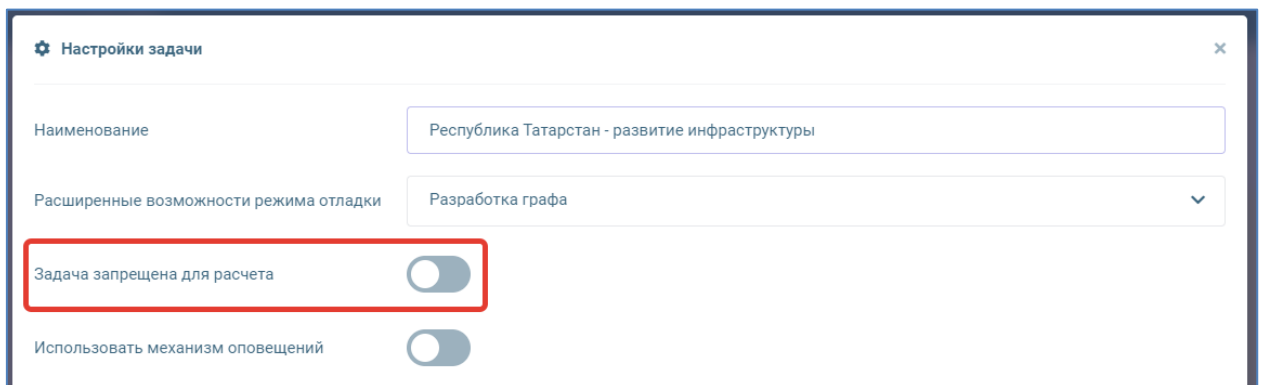


Рисунок 44 – Настройка задачи

Важно! Принцип передачи задачи на вычисление с использованием воркера – **FIFO**. Первый воркер, который будет доступен, возьмет задачу на расчет. Воркеры не ранжируются по возможностям и ресурсам.

9.8 Работа с RabbitMQ

Очередь сообщений между координатором и воркерами реализована через RabbitMQ.

9.8.1 Работа воркера (для сервера RabbitMQ является консьюмером – consumer)

1. При запуске воркера происходит подключение к серверу RabbitMQ.
2. Воркер подключается к очереди (если очереди нет, она создается).
3. Начинает прослушивать очередь (ожидает, когда задача придет на расчет).
4. Получает задачу на расчет.
5. Рассчитывает задачу.
6. Отправляет подтверждение серверу, что задача рассчитана.
7. Все повторяется с пункта 3.

9.8.2 Работа координатора (для сервера RabbitMQ является продюсером – producer)

1. Ожидает запросы клиентского приложения.
2. Получает запрос.
3. Если это запрос на расчет (расчет графа, блока, ветки, события, открытия пресета) переходим к пункту 7, иначе к пункту 4.
4. Выполняется запрос.
5. Выдает ответ клиенту.
6. Все повторяется с пункта 1.
7. Формируется запрос для расчета.
8. Отправляется rest запросом на сервер RabbitMQ.
9. Выдает ответ клиентскому приложению.
10. Все повторяется с пункта 1.

9.8.3 Работа сервера RabbitMQ

Получает сообщения от продюсера **Producer**, маршрутизирует их в очередь от данного продюсера **Producer**, затем отправляет свободному консьюмеру **Consumer**. Схематично работа сервера RabbitMQ представлена на рисунке (Рисунок 45).

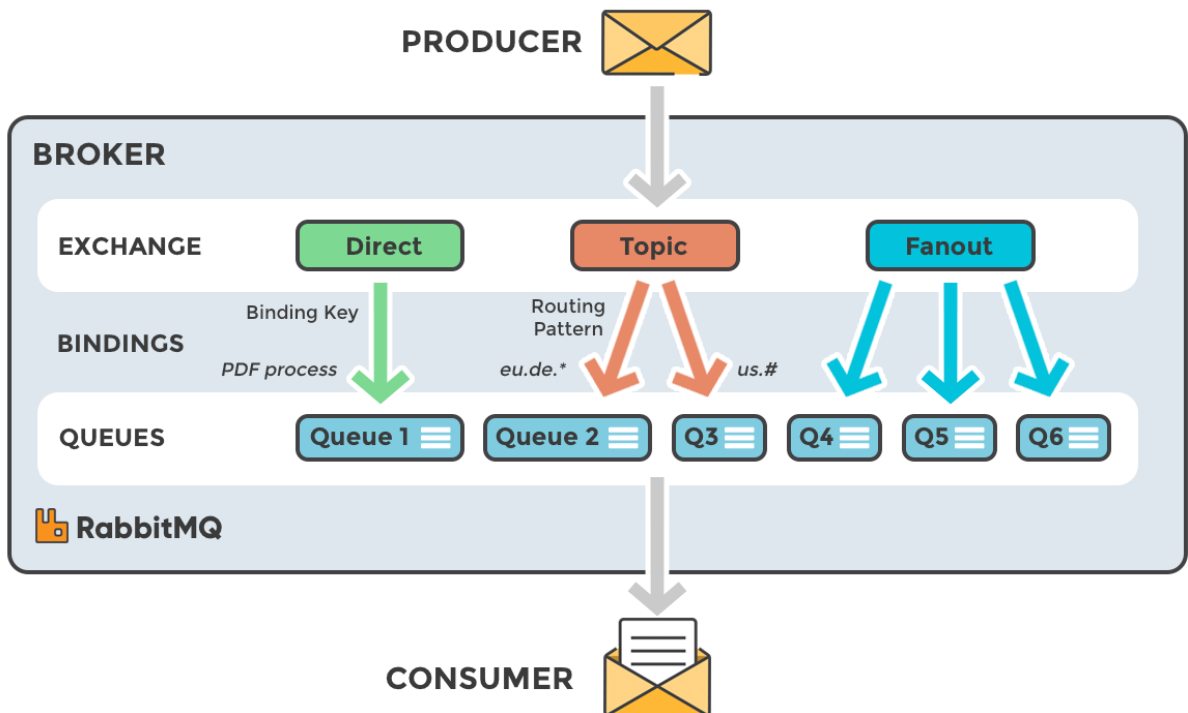


Рисунок 45 – Работа сервера RabbitMQ

Для настройки необходимо указать конфигурацию для подключения к очереди:

```
# --- Настройка RabbitMQ ---
```

```
rabbit_host = "hostname"
```

```
# Адрес сервера RabbitMQ
```

```

rabbit_port = 5672 # Порт RabbitMQ для amqp
rabbit_port_api = 15672 # Порт RabbitMQ для api
rabbit_virtual_host = "/" # Суффикс к серверу RabbitMQ
rabbit_user_login = "username" # Логин к серверу RabbitMQ
rabbit_user_password = "userpassword" # Пароль к серверу RabbitMQ

rabbit_heartbeat = 600 # Время опроса сервера RabbitMQ,
# что он доступен (в секундах)

rabbit_timeout = 1 # Время ожидания ответа
# (в секундах)

```

В конфиге RabbitMQ параметр `consumer_timeout` должен быть больше, чем параметр `calc_max_calculation_time` в конфиге приложения `backend`

9.8.4 Дополнительные ссылки

Официальный сайт [RabbitMQ](https://www.rabbitmq.com/).

9.9 Текущие расчеты

На странице «Текущие расчеты» (Рисунок 46) отображается список всех расчетов, которые запускались в приложении с возможностью просмотра:

1. Список расчетов с указанием типа: «Ручной» или «По расписанию».
2. Пользователь, который запустил расчет.
3. Тип расчета: «Расчет события», «Расчет пресета», «Расчет графа».
4. Дата начала, дата окончания и общее время выполнения.
5. Если расчет еще не завершен – отображается процент выполнения, а также имеется возможность получить его лог с помощью панели управления в шапке таблицы.

№	Задача	Пользователь	Тип	Тип запуска	Дата начала	Дата окончания	Время выполнения	Процент выполнения
1	VISMIND-2418 унификация сохранения данных и атрибутов	Krylov Aleksey	Расчет пресета	Ручной	07.08.2023 10:54:47	07.08.2023 10:54:47	0:00:00.372133	100%
2	Республика Татарстан - развитие инфраструктуры	Andrianov Georgiy	Расчет пресета	Ручной	07.08.2023 10:53:53	07.08.2023 10:53:53	0:00:00.2189718	100%
3	Республика Татарстан - развитие инфраструктуры	Andrianov Georgiy	Расчет пресета	Ручной	07.08.2023 10:53:48	07.08.2023 10:53:48	0:00:00.215854	100%
4	Республика Татарстан - развитие инфраструктуры	Andrianov Georgiy	Расчет пресета	Ручной	07.08.2023 10:53:44	07.08.2023 10:53:44	0:00:00.193643	100%
5	VISMIND-2418 унификация сохранения данных и атрибутов	Krylov Aleksey	Расчет пресета	Ручной	07.08.2023 10:50:32	07.08.2023 10:50:32	0:00:00.340734	100%
6	VISMIND-2418 унификация сохранения данных и атрибутов	Krylov Aleksey	Расчет события	Ручной	07.08.2023 10:50:21	07.08.2023 10:50:22	0:00:00.419504	100%
7	VISMIND-2418 унификация сохранения данных и атрибутов	Krylov Aleksey	Расчет пресета	Ручной	07.08.2023 10:50:18	07.08.2023 10:50:18	0:00:00.414286	100%
8	VISMIND-2418 унификация сохранения данных и атрибутов	Krylov Aleksey	Расчет события	Ручной	07.08.2023 10:50:01	07.08.2023 10:50:02	0:00:00.752681	100%
9	VISMIND-2418 унификация сохранения данных и атрибутов	Krylov Aleksey	Расчет пресета	Ручной	07.08.2023 10:49:55	07.08.2023 10:49:56	0:00:00.694171	100%
10	Отладка показа	Krylov Aleksey	Расчет события	Ручной	07.08.2023 10:34:29	07.08.2023 10:34:30	0:00:00.917121	100%
11	Отладка показа	Krylov Aleksey	Расчет события	Ручной	07.08.2023 10:34:25	07.08.2023 10:34:25	0:00:01.055347	100%
12	Отладка показа	Krylov Aleksey	Расчет пресета	Ручной	07.08.2023 10:34:23	07.08.2023 10:34:24	0:00:00.847652	100%
13	Отладка показа	Krylov Aleksey	Расчет пресета	Ручной	07.08.2023 10:33:25	07.08.2023 10:33:25	0:00:00.817643	100%
14	Отладка показа	Krylov Aleksey	Расчет события	Ручной	07.08.2023 10:12:21	07.08.2023 10:12:22	0:00:00.721317	100%
15	Отладка показа	Krylov Aleksey	Расчет пресета	Ручной	07.08.2023 10:12:14	07.08.2023 10:12:14	0:00:00.407034	100%

Рисунок 46 – Пример страницы «Текущие расчеты»

Таблица выводится в постраничном режиме, по умолчанию отображаются первые 50 элементов.

Доступны функции поиска и фильтрации по столбцам.

Индикация в первом столбце имеет 4 цвета:

- серый – расчет завершен;
- красный – в процессе расчета получены ошибки;
- желтый – в процессе расчета получены предупреждения;
- зеленый – расчет еще в процессе.

Записи в таблице хранятся 24 часа.

10 Лицензирование

Какие лицензии бывают

- Полная – нет ограничений на использование приложения
- Ограниченная:
 - датой окончания
 - количеством пользователи
 - количеством активных подключений
 - списком разрешенных блоков

Как загрузить лицензию в приложение

- С помощью графического интерфейса пользователя в приложении (см. п. 10.1).

Нужна ли перезагрузка для применения лицензии

При применении через API или интерфейс приложения перезагрузка приложения не нужна, но можно перезагрузить, если надо закрыть все активные соединения.

Если установка лицензии выполняется сразу в БД, перезагрузка нужна (не рекомендуемый вариант).

Как приложение реагирует на изменение лицензии

При применении лицензии пользователей, которые в данный момент находятся в приложении, не выкидывает из приложения, они продолжают работать. Таким образом, ограничение на количество пользователей и количество токенов на них не отразится (до момента, когда они заново не авторизуются). Однако ограничение для блоков отобразится сразу, не будут возвращаться данные в пресет по неразрешенным блокам, а в графе не станут рассчитываться блоки. При обновлении страницы с графом, если в графе использовались блоки, которые в лицензии запрещены, они подсвелятся красным цветом с описанием того, что на их использовании нет лицензии. Также до обновления страницы в графе, будут отображаться блоки в списке, но если их попробуют создать, то выйдет ошибка, что они не поддерживаются лицензией.

10.1 Ввод лицензии через визуальный интерфейс

Перед установкой лицензии работы в платформе должны быть завершены, т.к. автоматически будет произведен логат всех пользователей.

10.1.1 Если есть действующая лицензия

1. Открыть меню приложения.

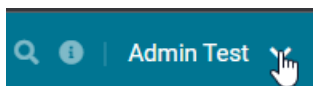


Рисунок 47 – Меню приложения

2. Открыть раздел «Общие настройки».

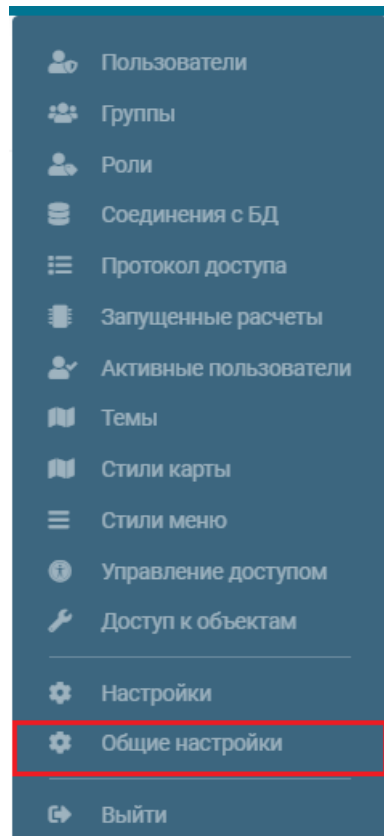


Рисунок 48 – Раздел «Общие настройки»

3. В поле «Текст лицензии» вставить лицензию.

Лицензия	
Дата окончания лицензии	Неограниченная лицензия
Количество пользователей	Не ограничено
Количество активных пользователей	Не ограничено
Текст лицензии	<div style="border: 2px solid red; height: 40px; width: 100%;"></div>

Рисунок 49 – Настройка лицензий

4. Нажать кнопку «Сохранить».

10.1.2 Если нет лицензии или истек срок действия лицензии

1. Открыть страницу <Адрес приложения>/license (или любую другую страницу приложения, после чего произойдет переадресация на страницу /license) (Рисунок 50).



The screenshot shows a web interface for a digital platform. At the top, the title reads 'Цифровая платформа УПРАВЛЕНИЕ В ПРОСТРАНСТВЕ'. Below the title is a text input field with the placeholder text 'Введите текст лицензии'. Underneath the input field is a dark blue button with the text 'Применить лицензию'. At the bottom of the page, there is a small copyright notice: 'Copyright © 2022 "Бюро Информационных Технологий" Версия приложения - 0.29.0-06651c45-review-vismind-8152-front-license-with-dev-backend'.

Рисунок 50 – Переадресация на страницу ввода лицензии

2. В поле «Введите текст лицензии» вставить лицензию.
3. Нажать кнопку «Применить лицензию».

11 Администрирование

11.1 Оповещение администратора

Реализована возможность оповещения в ситуациях, требующих вмешательства администратора, например, при обнаружении попыток подбора пароля. Оповещение производится посредством отправки письма E-mail или сообщения в чат Telegram, настройки задается в разделе **alert_messaging** файла конфигурации:

- **telegram** – флаг отправки сообщения в чат Telegram;
- **telegram_chat_id** – идентификатор чата Telegram;
- **telegram_api_token** – токен безопасности Telegram;
- **telegram_disable_notification** – флаг управления оповещением в чате Telegram;
- **email** – флаг отправки сообщения по E-mail;
- **smtp_host** – адрес почтового сервера для отправки сообщения;
- **smtp_port** – порт почтового сервера для отправки сообщения;
- **smtp_login** – учетная запись для отправки сообщения;
- **smtp_password** – пароль учетной записи;
- **email_from** – отправитель письма;
- **email_to** – получатель письма.

11.2 Протокол доступа

Страница, отображающая список действий пользователей в приложении, приведена ниже на рисунке (Рисунок 51).

	Дата	ID пользователя	Пользователь	Действие	ID объекта	
1	27.07.2023 15:51:51	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Администрирование		
2	27.07.2023 15:51:51	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Администрирование		
3	27.07.2023 15:51:51	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Авторизация пользователя		
4	27.07.2023 15:51:51			Без категории		
5	27.07.2023 15:51:51			Общая информация		
6	27.07.2023 15:49:43			Администрирование	5fe58d53-f012-4be5-a3a6-389f9a631dcb	Новый
7	27.07.2023 15:49:30	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Администрирование		
8	27.07.2023 15:49:26			Администрирование	e4d82020-5b86-4fc0-adad-a98f77c4807a	
9	27.07.2023 15:49:21	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Администрирование		
10	27.07.2023 15:49:16	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Администрирование		
11	27.07.2023 15:49:14	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Администрирование		
12	27.07.2023 15:49:14	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Администрирование		
13	27.07.2023 15:49:12	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Администрирование		
14	27.07.2023 15:49:12	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Авторизация пользователя		
15	27.07.2023 15:49:11			Без категории		
16	27.07.2023 15:49:11			Общая информация		
17	27.07.2023 15:48:53	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Администрирование		
18	27.07.2023 15:48:53	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Администрирование		
19	27.07.2023 15:48:53	e4e04f47-acc3-427c-8e02-f8b53e96b67a	Роман Писцов	Авторизация пользователя		

Рисунок 51 – Протокол действий пользователей

Протокол действий пользователей содержит информацию:

- Дата – когда было действие;
- ID пользователя – идентификатор пользователя, совершившего действие;
- Пользователь – имя пользователя, совершившего действие;
- Действие – свершенное действие;
- ID объекта – идентификатор объекта;
- Имя объекта – наименование объекта;
- Тип объекта – тип объекта;
- Метод запроса – метод вызываемой API;
- Запрос – вызываемая API;
- URI – адрес ссылки;
- Remote address – с какого ip-адреса;
- Remote port – с какого порта;
- Server port – на какой порт сервера;
- Server protocol – по какому протоколу;
- Server host – на какой ip-сервер;
- X real ip – с какого ip-адреса;
- X forwarded for – доп. информация;
- Браузер – какой браузер использовался;
- Referer – доп информация.

Ограничение: отображает последние 7000 действий.

Примечание

- Протокол доступа отображает последние 7000 запросов;
- На текущий момент логируются лишь следующие объекты приложения: пользователи, группы, роли, а также политика доступа;
- Изменение объектов при следующих операциях:
 - Создание/изменение/удаление пользователя
 - Блокировка/разблокировка пользователя
 - Создание/изменение/удаление группы
 - Создание/изменение/удаление роли
 - Изменение политики доступа

11.3 Сборка мусора

Bash скрипт, запускаемый в автоматическом режиме, предназначенный для удаления из системы неактуальных данных:

- биндингов выходов и выходов;

- просроченных токенов доступа;
- анонимных (гостевых) пользователей;
- файлов и других папок, если их давно не изменяли.

11.3.1 Скрипт

Запуск команды **cron** на сервере для выполнения очистки внутри контейнера, через вызов shell скрипта:

Важно! Не забыть раздать права на выполнение **+x** и изменять имя контейнера на нужное.

Исходный код скрипта. При необходимости можно изменить имя **контейнера приложения** и файлов, в которые выводится лог.

garbage_collector.sh

```
docker exec vismind-backend-container python utils/c_garbage_collector.py >>
./garb_coll_log
```

11.3.2 Расписание

Стандартное расписание для запуска скрипта сборки мусора:

```
0 2 * * * /usr/bin/sh ~/cronscripts/garbage_collector.sh
```

11.3.3 Установка

Скрипт устанавливается автоматически вместе с платформой, по умолчанию выполняется каждый день, в 2:00 по серверному времени.

После установки можно вручную скорректировать как сам скрипт, так и его расписание.

Установка происходит от имени пользователя, под которым был запущен основной процесс установки.

11.4 Сообщение об ошибках

11.4.1 Глобальные ошибки приложения

11.4.1.1 Доступ запрещен (403)

Код ответа на статус ошибки **HTTP 403 Forbidden** указывает, что сервер понял запрос, но отказывается его авторизовать.

В случае с приложением, данная ошибка может появиться при выполнении операции, к которой у пользователя нет доступа (к примеру, попытка зайти в задачу).

11.4.1.2 Ресурс не найден (404)

Код ответа на ошибку **HTTP 404 Not Found** указывает, что сервер не может найти запрошенный ресурс.

В случае с приложением, данная ошибка может появиться при попытке зайти в несуществующую задачу.

11.4.1.3 Ошибка сервера (500)

Код ответа сервера **HTTP 500 Internal Server Error** указывает на то, что сервер столкнулся с неожиданной ошибкой, которая помешала ему выполнить запрос.

Возникает в случае, когда во время выполнения запроса на сервере произошло критическое исключение, и оно не было корректным образом обработано.

11.4.1.4 Сервис недоступен (503)

HTTP 503 Service Unavailable код состояния сервера, указывающий на то, что сервер недоступен и отклонил запрос.

Причиной этого может быть отключение сервера для технических работ или отсутствие ресурсов для обработки запроса, в случае если сервер перегружен.

11.4.1.5 Разрыв соединения

В случае если соединение с сервером потеряно или его невозможно установить, пользователь получит уведомление о переподключении.

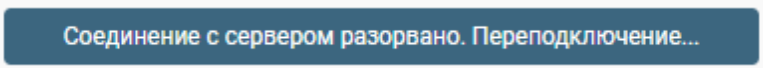


Рисунок 52 – Уведомление о переподключении

Попытка повторного подключения к серверу происходит раз в несколько секунд.

11.4.1.6 Ошибки при аутентификации

При вводе неверного логина или пароля отобразится сообщение в правом верхнем углу экрана.

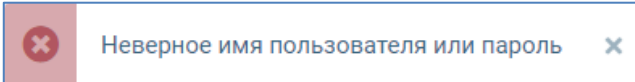


Рисунок 53 – Ошибки при аутентификации

11.4.1.7 Ошибки при расчете графа

Если блоки графа некорректно настроены или связаны, то в логе расчета графа отобразятся ошибки, они отмечены красной иконкой с восклицательным знаком (Рисунок 54).

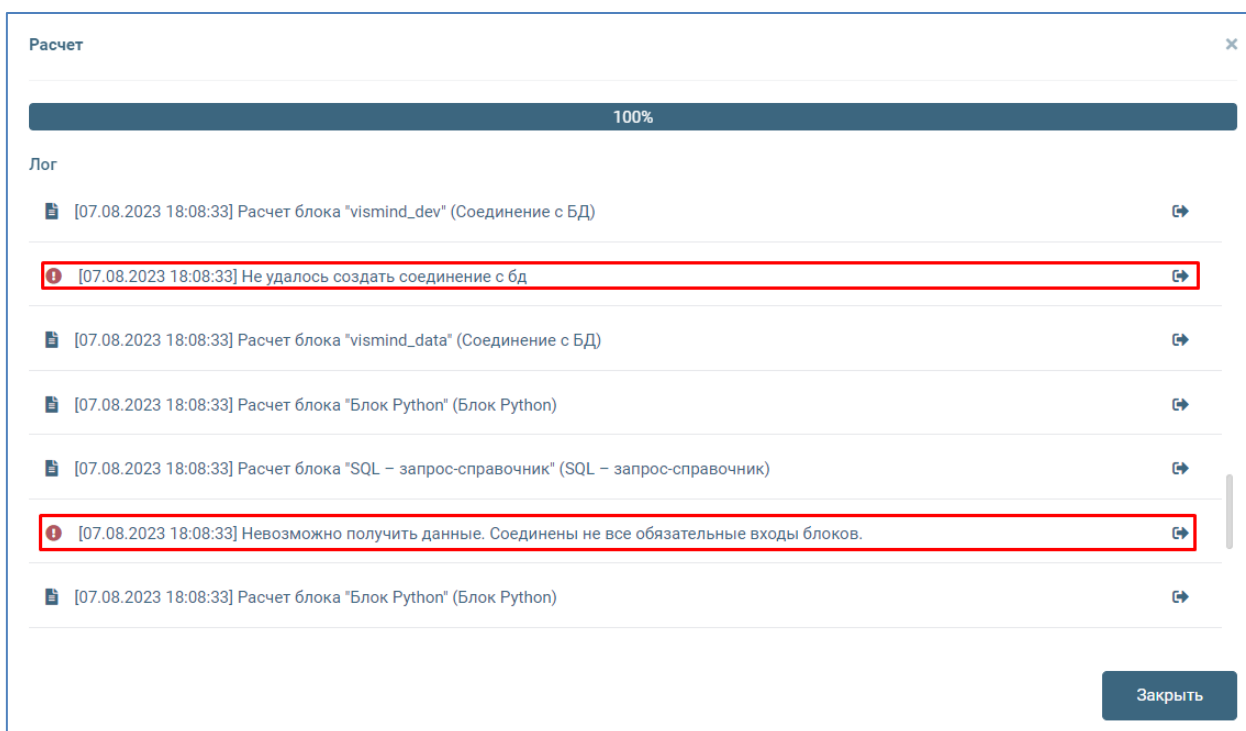


Рисунок 54 – Лог расчета графа с ошибками

На блок, в котором произошла ошибка, можно перейти, нажав на значок .

11.4.1.8 Ошибки в пресете

При работе с пресетом пользователь может увидеть сообщение об ошибке в правом верхнем углу экрана (Рисунок 55).

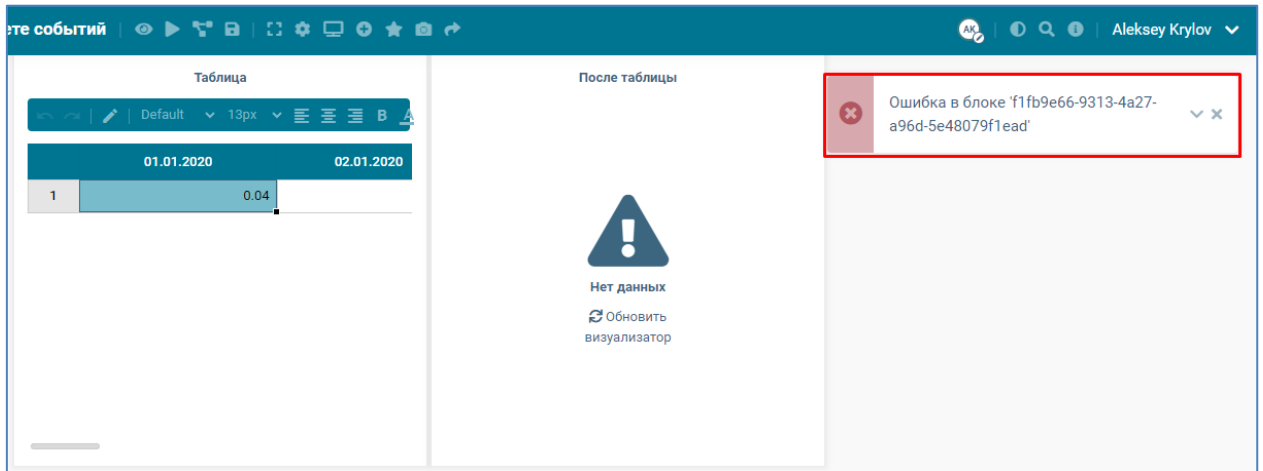



Рисунок 55 – Сообщение об ошибке в пресете

Данное сообщение возникает при ошибке в расчете цепочки графа, который был вызван событием в пресете.

11.4.2 Взаимодействие пользователя с сообщением

Пользователь может:

1. Раскрыть подробную информацию об ошибке, нажав на значок  .

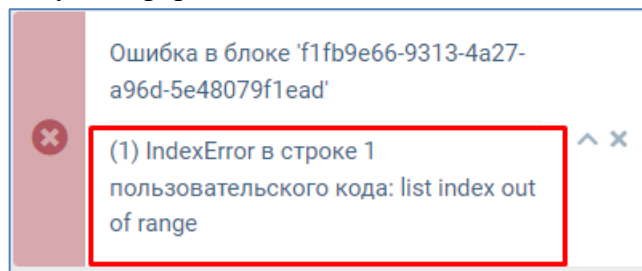


Рисунок 56 – Отображение информации об ошибке

2. Перейти в граф на блок, который был рассчитан с ошибкой, щелкнув левой клавишей мыши (ЛКМ) на идентификатор блока.

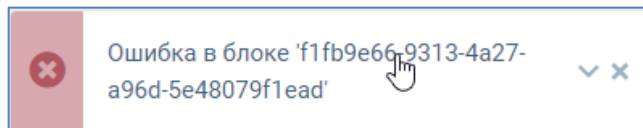



Рисунок 57 – Настройка

3. Закрыть сообщение, нажав на значок  (закрывать сообщение необязательно, через несколько секунд оно закроется автоматически).

11.4.3 Прочие ошибки

Ошибки, которые не обрабатываются специальным образом, могут быть выведены на экран так же в виде всплывающего окна справа вверху или сообщения в логе (но уже без ссылок на источник ошибки).

11.5 Настройки платформы

11.5.1 Вывод подробной информации об ошибках

В приложении предусмотрена возможность скрытия от пользователя подробного описания ошибок.

Если настройка включена:

- Приложение сообщает об обрабатываемых исключениях приложения. В подробном описании исключения передаётся, в какой части приложения оно было вызвано;
- Приложение сообщает о системных ошибках, которые приложение не обрабатывает. Аналогично, в подробном описании передаётся, где и почему возникла ошибка.

Если настройка выключена:

- Приложение сообщает об обрабатываемых исключениях приложения. Подробности ошибки не передаются;
- Приложение сообщает о системных ошибках, помечая их как "Неопознанная ошибка". Подробности ошибки не передаются пользователю.

Настройка регулируется параметром **send_system_errors**.

Независимо от настройки, все типы исключений логируются внутри приложения.

11.5.2 Запрет на загрузку файлов определенного типа

В конфигурационном файле приложения можно задать список расширений файлов, загрузка которых в приложение будет невозможна.

В разделе "Настройка безопасности" параметр **blocked_file_types** - строка, в которой, через запятую без пробелов перечисляются расширения, запрещенные для загрузки.

Пример: `blocked_file_types = "exe,txt"`

11.5.3 Принудительное прекращение сессии пользователя в случае его бездействия

Для настройки принудительного прекращения сессии пользователя используются два параметра, задающиеся в конфигурационном файле приложения **conf/config.toml**:

- **token_idle_time_start** – время (в минутах) до начала отсчета времени бездействия;
- **token_idle_time** – время допустимого бездействия пользователя (в минутах). При установке значения «0», время отсчитываться не будет.

11.6 Управление доступом

Задание ограничений на пароли, время сессии и другие настройки выполняется в разделе «Управление доступом». Для перехода следует выбрать соответствующий пункт в выпадающем меню (Рисунок 58). Для доступа к разделу пользователь должен обладать необходимыми полномочиями.

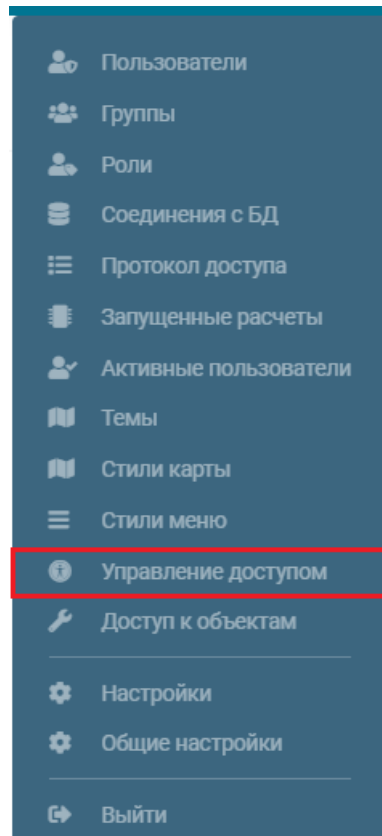


Рисунок 58 – Раздел «Управление доступом»

При переходе в раздел отображается мастер, содержащий следующие группы настроек: «Пароли», «Доступ».

11.6.1 Пароли

Максимальный срок действия пароля (в днях)	<input type="text" value="0"/>
Минимальная длина пароля (в символах)	<input type="text" value="2"/>
Запретить совпадение имени пользователя и пароля	<input type="text" value="Выключено"/>
Новый пароль не совпадает с предыдущими (количество)	<input type="text" value="1"/>
Запретить использование пароля из одинаковых символов	<input type="text" value="Выключено"/>
Запретить использование пароля из букв одного регистра	<input type="text" value="Выключено"/>
Пароль должен содержать цифры и буквы	<input type="text" value="Выключено"/>
Пароль должен содержать спецсимволы (_@, #,\$%&*,^)	<input type="text" value="Выключено"/>
Максимальное количество повторяющихся символов	<input type="text" value="2"/>
Проверять пароль на вхождение запрещенных строк	<input type="text" value="Выключено"/>

Рисунок 59 – Группа настроек «Пароли»

Группа настроек «Пароли» содержит следующие параметры, влияющие на криптостойкость допустимых в приложении паролей:

- Минимальная длина пароля (в символах): число, от 2 до 20 включительно. По умолчанию минимальная длина – 6 символов.
- Запретить совпадение имени пользователя и пароля: переключатель.
- Новый пароль не совпадает с предыдущими (количество). При значении 0 сравнение со старыми паролями не выполняется. По умолчанию настройка отключена.
- Запретить использование пароля из одинаковых символов: переключатель.
- Запретить использование пароля из букв одного регистра: переключатель.
- Пароль должен содержать цифры и буквы: переключатель.
- Пароль должен содержать спецсимволы (_,@, #,\$,%,&,* ,^): переключатель.
- Максимальное количество повторяющихся символов: число, от нуля (не контролируется) до 128.
- Проверять пароль на вхождение запрещенных строк: переключатель.
 - Если включено, то становится доступным текстовое поле, в которое можно ввести через пробел варианты запрещенных строк.
- Максимальный срок действия пароля (в днях). По умолчанию значение «0» (срок действия пароля неограничен).

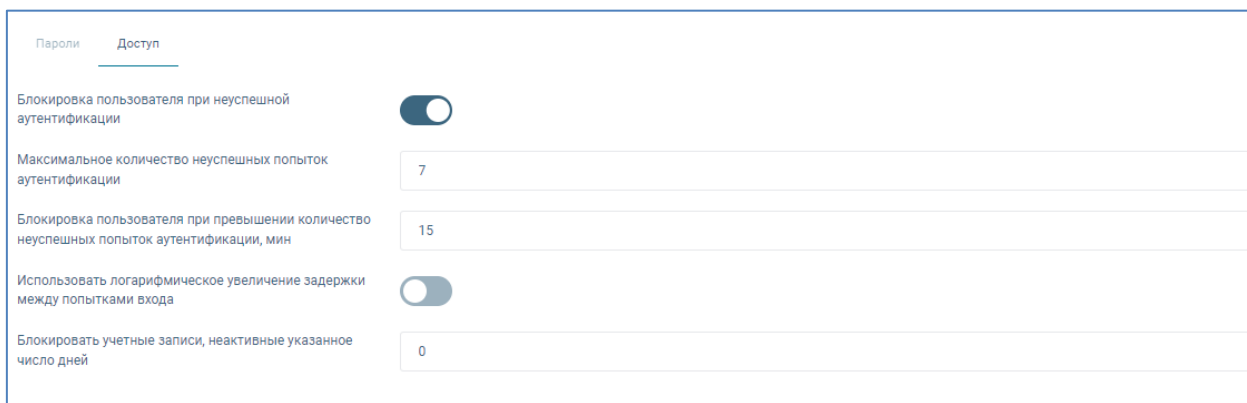
11.6.1.1 Срок действия пароля

Группа настроек «Пароли» содержит параметр «Максимальный срок действия пароля (в днях)», влияющий на необходимость регулярной смены паролей. Настройка применима лишь к учетным записям, созданным внутри приложения. Для внешних учетных записей настройка неприменима.

Настройка выключена, если в поле задан параметр «0». Если настройка включена, то пользователи с истекшим паролем теряют возможность входа в приложение до тех пор, пока не изменят пароль. Активные сессии пользователя при этом сохраняются: ограничение действует только на новые попытки входа.

Смена истекшего пароля выполняется через страницу смены пароля. Для смены пароля требуются логин пользователя, старый пароль и новый пароль. Пароли не должны совпадать между собой.

11.6.2 Доступ



Настройка	Значение
Блокировка пользователя при неуспешной аутентификации	<input checked="" type="checkbox"/>
Максимальное количество неуспешных попыток аутентификации	7
Блокировка пользователя при превышении количество неуспешных попыток аутентификации, мин	15
Использовать логарифмическое увеличение задержки между попытками входа	<input checked="" type="checkbox"/>
Блокировать учетные записи, неактивные указанное число дней	0

Рисунок 60 – Группа настроек «Доступ»

Группа настроек «Доступ» содержит следующие параметры, препятствующие взлому пароля:

- Блокировка пользователя при неуспешной аутентификации: флаг, позволяющий включить настройку блокировки. При включении настройки становятся доступны следующие параметры:
 - Блокировать пользователя после указанного числа неудачных попыток входа: число, выполняется временная блокировка пользователя, для предотвращения попыток подбора пароля (brute-force).
 - Сбрасывать счетчик неудачных попыток входа (через указанное число минут): число.
 - Минимальное время блокировки (указанное число секунд): число, фактическое время блокировки будет зависеть также от числа неудачных попыток аутентификации, зафиксированных за время до сброса счетчика попыток.
 - Увеличение времени блокировки при росте неудачных попыток: флаг, позволяющий увеличивать срок блокировки при каждой неудачной попытке входа пользователя. Срок блокировки при этом будет расти логарифмически от исходного значения, заданного в параметре «Минимальное время блокировки».
- Блокировать учетные записи, неактивные указанное число дней: число дней, через которые пользователь будет заблокирован при отсутствии активности (отсутствие каких-либо действий в приложении). Значение «0» – настройка не активна.

Подробное описание механизма блокировки при неуспешной аутентификации приведено далее в п. 11.6.3.

11.6.3 Ограничение числа попыток аутентификации

Настройка предназначена для защиты от подбора паролей к учетным записям пользователей.

11.6.3.1 Описание настройки

При включении параметра для всех пользователей задаётся ограничение на количество неудачных попыток входа в приложение. Предусмотрен следующий сценарий:

1. Пользователь выполняет попытку входа с неверными данными аутентификации.
2. Система фиксирует неудачную попытку входа и учитывает её при следующих попытках войти. Каждая неудачная попытка имеет время жизни в системе, по истечению которого она перестаёт учитываться. Данный параметр может быть настроен в разделе «Управление доступом» / «Доступ».
3. При достижении лимита попыток пользователь получает временную блокировку на вход в систему.
4. Если во время блокировки пользователь попытается повторно войти в систему, его блокировка будет продлена.

Под блокировкой понимается блокировка входа в систему, а не полная блокировка учетной записи пользователя. Это значит, что при блокировке входа пользователи, уже вошедшие в учетную запись, сохраняют возможность работы с платформой.

Дополнительным элементом защиты является возможность оповещения администратора посредством E-Mail или в чате Telegram.

11.6.3.2 Задание настроек

Механизм настраивается в разделе "Управление доступом", в подразделе "Доступ".
 Подробнее описание настроек находится в п. 11.6.

протоко

11.6.3.3 Поведение механизма при смене настроек

Если в ходе эксплуатации системы были изменены параметры блокировки (количество максимальных попыток/время блокировки/время мониторинга), либо настройка была полностью отключена, изменения не отразятся на ранее заблокированных пользователях. Если пользователь повторно попытается войти в систему, его время блокировки будет продлено на основе времени задержки, которое использовалось в момент блокировки пользователя, а не на основе новых настроек.

Примеры:

1. Пользователь заблокирован, срок блокировки 15 минут, время фиксированное. Если администратор выключит механизм, пользователь останется заблокированным на 15 минут, и при повторных попытках время блокировки будет продлеваться на 15 минут.
2. Пользователь заблокирован, срок блокировки 10 минут, время нарастающее. Если администратор сменит общее время блокировки на 5 минут и сделает время фиксированным, то пользователь останется заблокированным на 10 минут, а блокировки будут продлеваться на фиксированные 10 минут.

11.6.3.4 Применение настройки по типам аутентификации

Механизм ограничивает попытки входа в систему внутренних пользователей приложения, а также пользователей, использующих LDAP и OAuth в качестве метода аутентификации.

11.6.3.4.1 LDAP (Active Directory)

В окружениях, использующих аутентификацию LDAP, необходимо учесть следующие условия:

- Если уже настроена защита от подбора паролей в Active Directory, то можно отключить механизм защиты в приложении;
- Если не настроена защита в Active Directory, но необходимо использовать механизм защиты в приложении, тогда его нужно включить;
- Заказчик не планирует пользоваться защитой. Тогда защита должна быть отключена и в приложении, и в Active Directory;
- Администратор желает иметь защиту и в Active Directory, и в приложении. В этом случае он должен настроить параметры идентично и в приложении, и в AD, в противном случае корректная работа не гарантируется

Если защита от подбора пароля включена и в Active Directory, и в приложении, возможна блокировка входа для пользователя AD, чья учетная запись ещё не зарегистрирована в системе. Такого пользователя невозможно разблокировать вручную через раздел администрирования в приложении, поскольку его УЗ отсутствует в приложении.

Для восстановления возможности входа потребуется обнулить счётчик неудачных попыток пользователя в Redis. Счётчик доступен только при включённой защите от подбора в платформе.

11.6.3.4.2 Kerberos

Для пользователей, проходящих аутентификацию с помощью Kerberos, неудачные попытки входа не регистрируются

11.6.3.5 Защита от угадывания логинов пользователей

В текущей реализации механизм защиты от угадывания не предусмотрен, потому что факт блокировки фиксируется по идентификатору пользователя. Для несуществующих пользователей не существует идентификатора.

11.6.4 Блокировка пользователей при попытках подбора пароля

В целях защиты от подбора пароля методом перебора (brute force attack) реализован специальный алгоритм обнаружения такой атаки и противодействия ей. При обнаружении в течение определенного времени значительного количества попыток авторизации происходит задержка при следующей авторизации атакуемого пользователя. Пауза при авторизации вместо применения реальной блокировки позволяет зайти легальному пользователю, но существенно ограничивает возможности злоумышленника, не позволяя провести количество попыток, достаточное для взлома пароля. При увеличении количества попыток авторизации увеличивается также время задержки (логарифмическая зависимость). Через определенное время при отсутствии попыток авторизации счетчик для пользователя сбрасывается, задержки в авторизации отключаются.

Дополнительным элементом защиты является возможность оповещения администратора посредством E-mail или в чате Telegram (см. п. 11.1).

В файле конфигурации в разделе **preventing_brute_force** задаются следующие параметры, управляющие поведением временной блокировки:

- `monitoring_period` – период отслеживания в секундах, за который считаются попытки авторизации, если по истечении указанного времени авторизация пользователя не выполняется, счетчик сбрасывается;
- `max_attempts` – количество попыток авторизации, достижение которого за период отслеживания считается подозрительным, при его превышении включается временная блокировка пользователя и происходят задержки в авторизации пользователя;
- `delay` – базовая задержка авторизации в секундах.

11.6.5 Ограничение числа одновременных сеансов пользователя

Настройка предназначена для ограничения одновременной работы нескольких лиц под одной учетной записью.

11.6.5.1 Описание настройки

Механизм работает следующим образом: если количество активных сеансов пользователя превышает допустимое значение, то более ранние сеансы помечаются как неактивные. При этом лица, чей сеанс стал неактивен, получают сообщение о завершении сессии из-за превышения лимита одновременных сеансов и будут переведены на страницу аутентификации.

Настройка может быть задана любым целым неотрицательным числом N , где N – максимальное количество одновременных сеансов. Параметр является опциональным. Значение по умолчанию – «0». В этом случае число сеансов неограниченно.

Настройка доступна только администратору. Она размещена в разделе администрирования, в разделах «Пользователи», «Группы» и «Роли».

11.6.5.2 Наследование настройки и приоритеты

Параметр может быть задан на трёх уровнях: на уровне пользователя, на уровне группы и на уровне роли:

- на уровне роли: все пользователи, входящие в группы с данной ролью, наследуют параметр;
- на уровне группы: все пользователи группы наследуют параметр;
- на уровне пользователя: параметр задаётся индивидуально для отдельного пользователя.

Каждый из уровней имеет свой приоритет: у параметра в роли самый низкий приоритет, у параметра пользователя – самый высокий. Таким образом, параметр роли может быть переназначен для отдельной группы, а параметр группы может быть переназначен для отдельного пользователя.

Если на более высоком уровне применена настройка, а на более низком установлено значение по умолчанию, то будет учитываться настройка более высокого уровня. Например, если для отдельного пользователя задано неограниченное число сеансов, но он входит в группу с лимитом в 10 сеансов, то итоговый лимит пользователя – 10 сеансов.

Если пользователь входит в несколько групп, то среди всех доступных параметров будет выбрано минимальное значение, не считая значения по умолчанию. Аналогично для ролей: если группе назначено несколько ролей, то из всех параметров будет выбран наименьший.

11.6.5.3 Актуализация сеансов при изменении настроек администрирования

В случае изменения настроек администрирования будут актуализированы сеансы пользователей: с учетом изменений будет обновлено количество активных сессий для всех пользователей, которых коснулась настройка.

Актуализация происходит в следующих случаях:

1. Если изменена настройка пользователя.
2. Пользователь добавлен в группу или удалён из неё.
3. Группа удалена.
4. Изменена настройка группы.
5. Группе назначена роль или у группы удалена роль.
6. Роль удалена.
7. Изменена настройка роли.

Изменение (1) и (2) применяется к отдельному пользователю. Изменения (3), (4), (5) применяются ко всем пользователям, входящим в изменённую группу. Изменения (6), (7) применяются к пользователям групп, входящим в изменённую роль. При этом если параметр был увеличен, то ранее неактивные сеансы, станут снова активными.

11.7 Технические особенности платформы

Платформа реализована с использованием трехзвенной архитектуры. При этом дополнительно используется сервер Redis как часть ядра системы для общей памяти и кэширования. При запуске и работе платформы есть некоторые особенности, которые необходимо учитывать при организации технической инфраструктуры. Все разделы актуальны и для одного сервера платформы и для воркеров.

11.7.1 База данных и организация хранения

Платформа по своей сути является stateless приложением, все данные хранятся в БД репозитория платформы. Для этого используется СУБД PostgreSQL. Платформа к БД репозитория подключается с помощью технической учетной записи, указанной в конфигурации серверной части приложения.

Прямое соединение с БД репозитория должно быть доступно со стороны серверной части приложения (сервера). В репозитории располагаются все данные обо всех объектах системы, их структура и т.д. Размер этой БД напрямую зависит от объема данных, которые рассчитываются на графах.

Во время расчетов результаты последнего расчета каждого блока сохраняются в репозиторий.

11.7.1.1 Соединение с внешними источниками данных

Для соединения с внешними источниками данных с использованием коннекторов они должны быть доступны с сервера платформы. Клиентское приложение к источникам доступа не имеет.

При работе в режиме воркеров платформы доступ к источникам данных должен иметь каждый воркер. Это необходимо принимать во внимание в случае, если воркеры располагаются на разных машинах.

11.7.2 Использование Redis

Во время работы платформы Redis решает две основные задачи: кэширование и разделяемая память.

11.7.2.1 Кэширование

Для обеспечения кэширования результатов расчетов, прав доступа и других внутренних структур платформы во время работы в Redis создается отдельная БД (коллекция). При этом Redis не хранит свои состояния на диске, то есть работает только в режиме нахождения в оперативной памяти.

Если сервер Redis будет выключен, то все данные из него будут удалены. При этом система будет недоступна для работы. По умолчанию, сервер Redis запускается рядом с системой в виде отдельного контейнера docker.

При обработке большого количества данных, особенно для картографических визуализаторов, необходимо подбирать объемы памяти в соответствии с объемами.

11.7.2.2 Разделяемая память

На серверной части платформы используется Python. Его особенность состоит в том, что он является однопоточным, сервера приложения (wsgi) и **gunicorn**, которые используются для обслуживания большого количества запросов, используют воркеры (workers), которые запускаются в виде отдельных процессов на сервере. Каждый воркер, внутри gunicorn, представляет собой отдельную копию приложения, запущенную в виде отдельного процесса. При этом процессы не имеют общей памяти, каждый из них изолирован от другого и не знает о том, что рядом запущены еще воркеры.

Gunicorn сервер принимает запросы с клиентов и распределяет их по воркерам по принципу **roundrobin**, то есть заранее нельзя сказать, какой воркер будет обрабатывать следующий запрос. При этом внутри приложения все равно необходима организация хранилища для взаимодействия между воркерами. Для этих целей используется Redis – воркеры обмениваются информацией друг с другом с помощью хранения состояний в Redis.

11.7.3 Ограничения по потребляемой памяти

Для работы самой платформы (не Redis) так же требуется достаточно большое количество оперативной памяти. Есть прямая зависимость между объемами оперативной памяти и объемами обрабатываемых данных. Все расчеты происходят в памяти сервера, поэтому в случае, если требуется обработка больших объемов данных одновременно, и нет возможности разбить их на части, необходимо подбирать объем оперативной памяти сервера в соответствии с планируемыми объемами данных.

11.7.4 Количество нод расчетчиков (workers) и workers сервера gunicorn

У платформы есть режим работы с собственными воркерами (workers), которые могут осуществлять расчеты, находясь на удаленных серверах, а не на основном. При этом воркер – это полноценный сервер платформы, запущенный в другом режиме. Для его работы так же нужно соединение с БД репозитория.

Воркеры платформы используют очередь сообщения RabbitMQ для общения с координатором (основным приложением).

11.7.4.1 Количество воркеров платформы

Количество воркеров для расчетов можно рассчитать исходя из предполагаемой нагрузки именно по расчетам. Один воркер можно обслуживать расчеты от 10 до 50 пользователей. Количество пользователей напрямую зависит от того, насколько «тяжелые» расчеты предполагаются к выполнению.

11.7.5 Gunicorn workers

Внутри воркеров платформы и координатора запущен веб-сервер **gunicorn** со своими воркерами, которые обслуживают запросы пользователей.

Для режима только координатора используется 6 воркеров gunicorn.

Для режима координатора с воркерами в координаторе используется 3 воркера gunicorn. Внутри каждого воркера платформы используется по 1 воркеру gunicorn.

11.7.6 Особенности расчетов с высокой нагрузкой на CPU

Если расчет графа представляет собой «тяжелое» математическое вычисление, особенно с использованием внешних библиотек типа statmodels – это может привести к тому, что воркер gunicorn, на котором происходит выполнение, перестает отвечать и обрабатывать запросы (так как Python является однопоточным по своей природе). Это приводит к тому, что если таких расчетов запустить несколько – бэкенд сервера может перестать отвечать. В этом случае может потребоваться перезагрузка или ожидание окончания расчета. В случае если такие ситуации происходят – желательно рассмотреть вариант работы с воркерами платформы.

11.8 Проекция графа

В меню при работе с графом есть пункт, который отвечает за то, какой граф рассматривается:

- основной;
- проекция – отображает блоки, которые находятся на пресете.

Данный пункт выбирается в меню графа.



Рисунок 61 – Меню графа

- Основной – имеет такое же название, как и у самого графа.
- Проекция – имеет название, как у пресета, по которому надо будет отобразить блоки, которые нужны для его расчета.

В проекции доступны только следующие действия:

- передвигать отображаемые блоки;
- менять стили отображаемых блоков.

11.9 Управление пользователями

11.9.1 Общее описание

Функционал управления пользователями осуществляется через страницы администрирования «Пользователи» и «Настройки» (Рисунок 62).

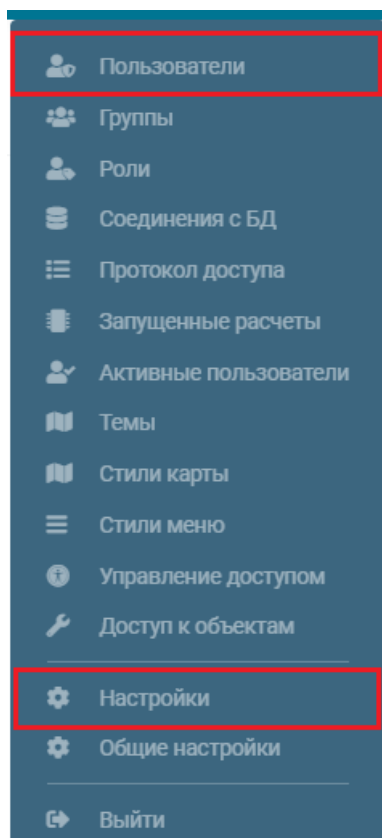


Рисунок 62 – Разделы меню «Пользователи» и «Настройки»

Доступность разделов меню зависит от разрешений, заданных в ролях пользователя.

На странице «Пользователи» слева расположен список пользователей. Если выделить пользователя, то справа открываются его настройки (Рисунок 63).

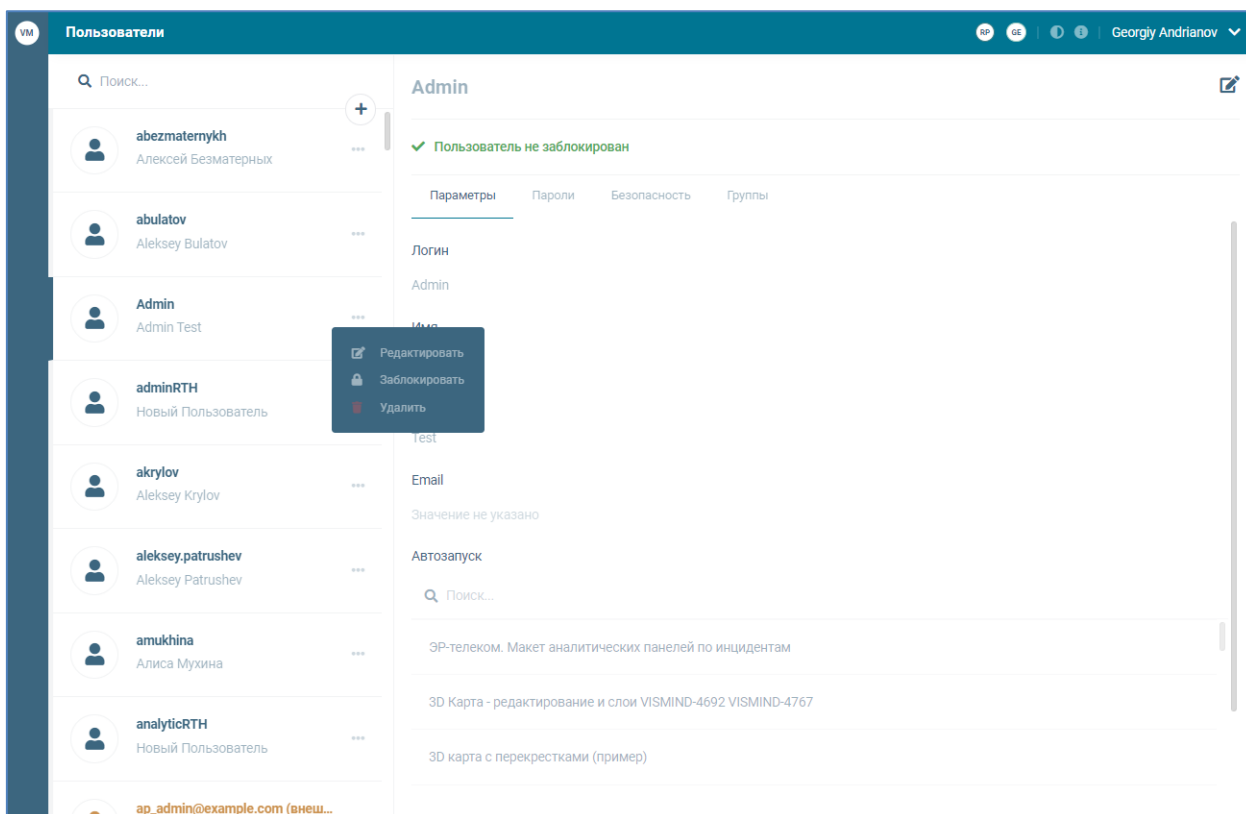








Рисунок 63 – Группа настроек «Пользователи»

На странице «Пользователи» доступен следующий функционал:

1. Поиск пользователя (кнопка ).
2. Создание пользователя (кнопка ).
3. Открыть контекстное меню (кнопка ).
4. Редактировать пользователя (кнопка ).
5. Заблокировать пользователя (пункт контекстного меню ).
6. Удалить пользователя (пункт контекстного меню ).
7. Индикация заблокированного пользователя.
8. Индикация внешнего пользователя.

На странице «Пользователи» (Рисунок 64) представлены настройки текущего пользователя.

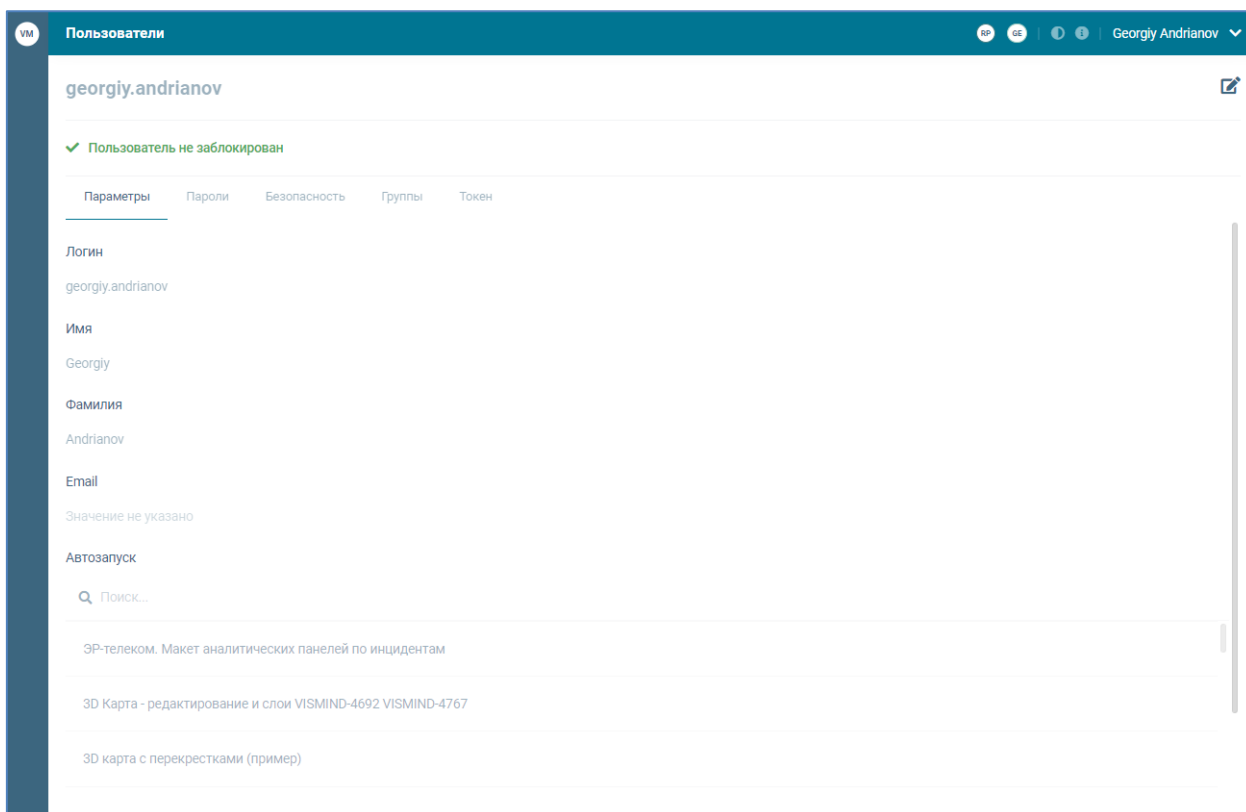


Рисунок 64 – Страница «Пользователи»

Список настроек аналогичен настройкам на странице «Пользователи», дополнительно имеется возможность создать токен для расчетов (Рисунок 65).

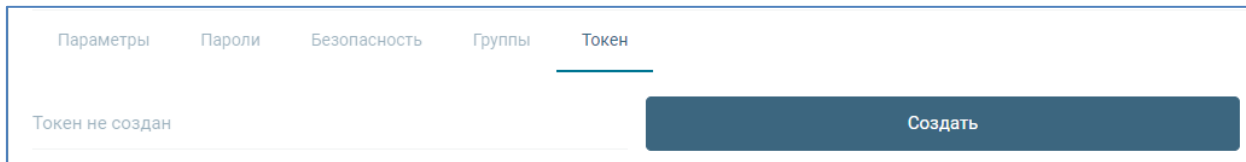


Рисунок 65 – Создание токена

11.9.2 Внешние пользователи

Существует возможность автоматического создания пользователей, если пользователи уже существуют в рамках внешних сервисов авторизации (например, Active Directory).

- Настройка автоматического создания внешних пользователей осуществляется в файле конфигурации backend-приложения.
- Создание пользователя происходит при первом входе в приложение.
- Атрибуты пользователя в приложении (логин, имя, фамилия, email, группы) будут соответствовать атрибутам во внешних системах при соответствующих настройках в файле конфигурации

Важно! В конфигурационном файле приложения можно указать группу по умолчанию (созданную в рамках приложения), в которую будут добавляться внешние пользователи.

- У внешних пользователей можно редактировать задачу для автозапуска и группы (остальные атрибуты редактировать запрещено).

11.9.3 Индикация пользователей

В случае если пользователь заблокирован, в списке он будет выделен красным цветом и в скобках будет указано, что пользователь является заблокированным.

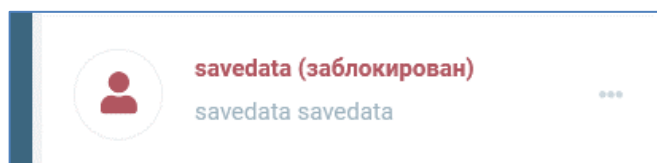


Рисунок 66 – Индикация заблокированного пользователя

Если пользователь внешний по отношению к системе (например, создан через Active Directory), он будет отображаться оранжевым цветом, и будет указано, что пользователь является внешним.

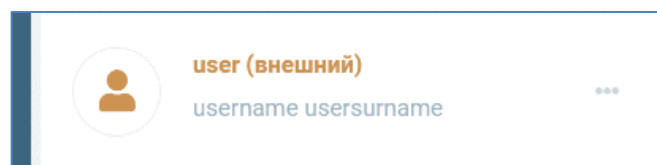


Рисунок 67 – Индикация внешнего пользователя

Важно!

- Для «внешних» пользователей нет возможности редактирования части данных: имя, фамилия, email.
- Для «внешних» пользователей нет вкладки с изменением пароля.

11.9.4 Задача для автостарта пользователя

Пользователю возможно установить задачу для автостарта, чтобы после аутентификации в приложении пользователь перенаправляется на пресет установленной задачи.

Для этого необходимо:

1. Перейти на страницу администрирования «Пользователи» (Рисунок 68).

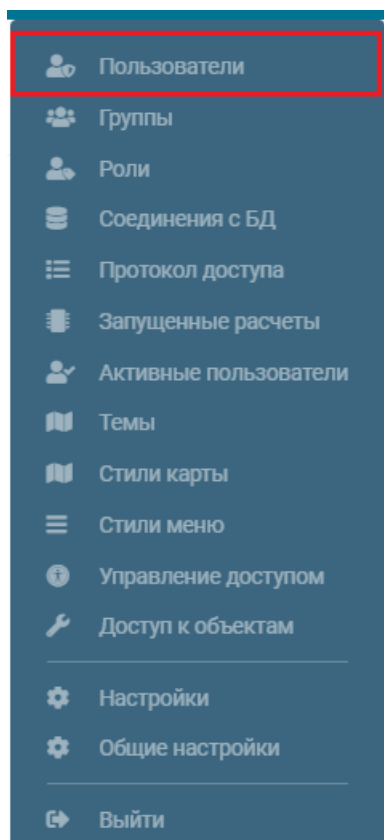


Рисунок 68 – Раздел меню «Пользователи»

2. В списке пользователей выбрать нужного пользователя (Рисунок 69).

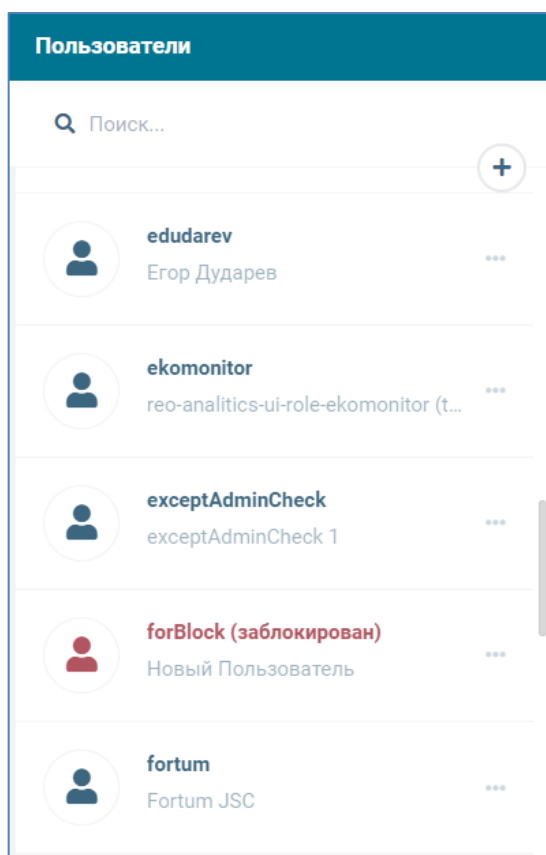



Рисунок 69 – Список пользователей

3. Включить редактирование, нажав на кнопку .
4. В разделе «Автозапуск» выбрать задачу, на которую будет перенаправлен пользователь после аутентификации (Рисунок 70).

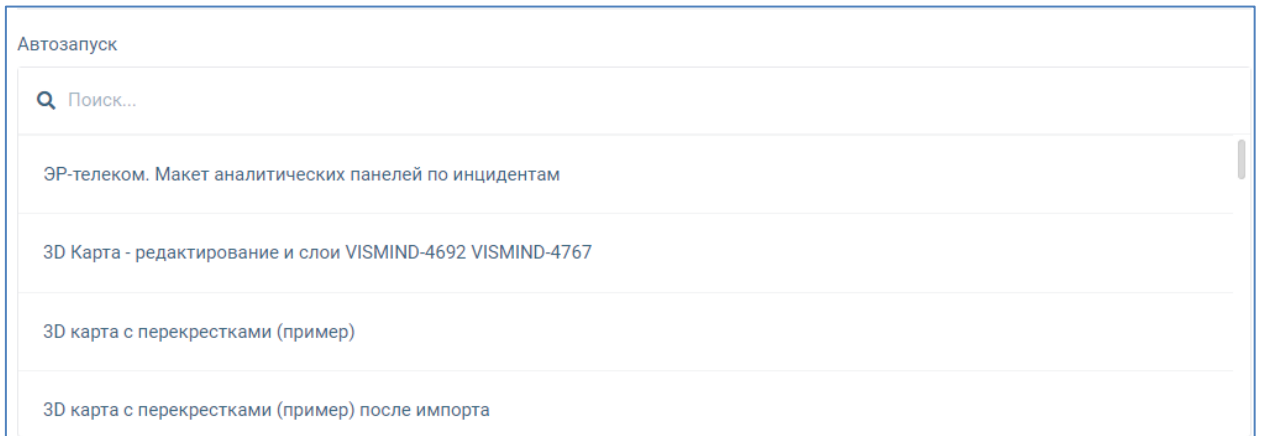


Рисунок 70 – Настройка «Автозапуск»

5. После нажать «Сохранить».



Важно! Пользователь перейдёт на первый пресет выбранной задачи.

11.9.5 Создание и редактирование пользователя

11.9.5.1 Создание пользователя

1. Создание пользователей происходит в меню «Пользователи» (Рисунок 71).

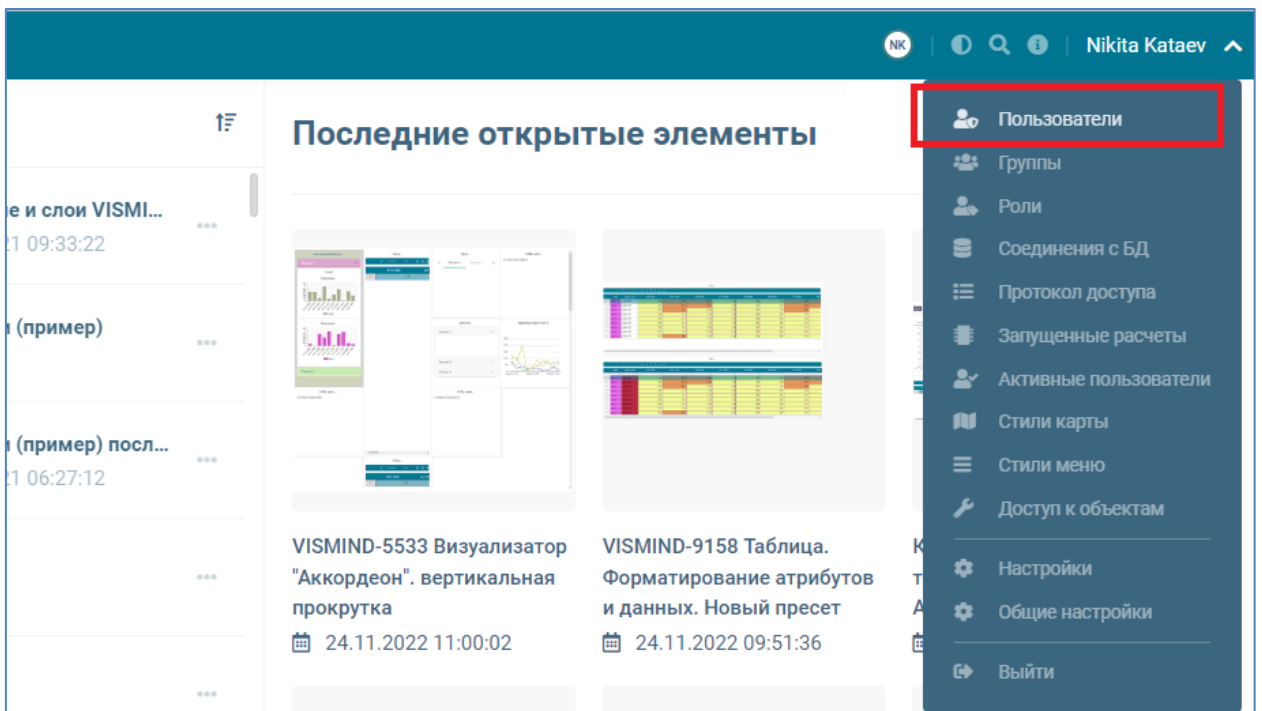



Рисунок 71 – Раздел меню «Пользователи»

2. Для того чтобы создать пользователя, надо нажать на кнопку  (Рисунок 72).

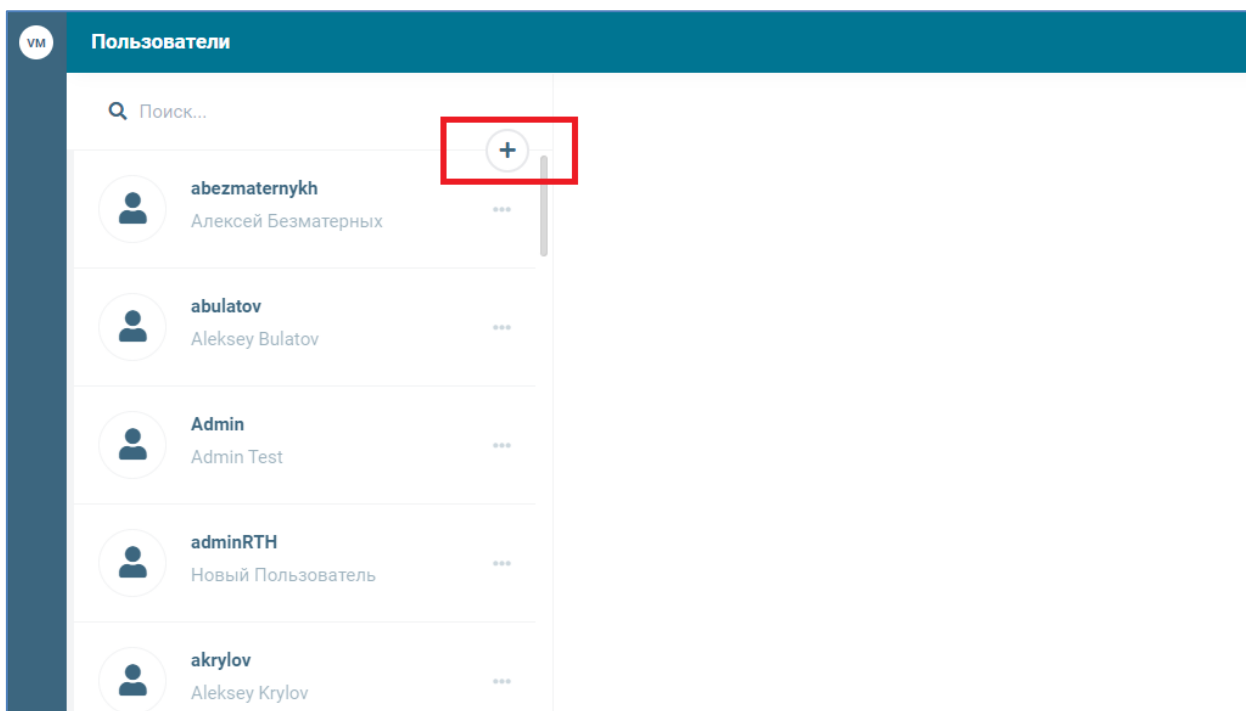


Рисунок 72 – Создание пользователя

3. После чего откроется форма для создания нового пользователя (Рисунок 73).

Рисунок 73 – Форма для создания нового пользователя

В форме для создания нового пользователя необходимо заполнить поля:

1. На вкладке «Параметры»:
 - Логин (обязательно);
 - Имя (обязательно);
 - Фамилия (обязательно);
 - Email (не обязательно);
 - Автозапуск (не обязательно) – если выбрать задачу, то при открытии приложения будет открываться данная задача.
2. На вкладке «Безопасность»:
 - Пароль (обязательно).
3. На вкладке «Группы» добавить группы, в которые входит пользователь (Рисунок 74).

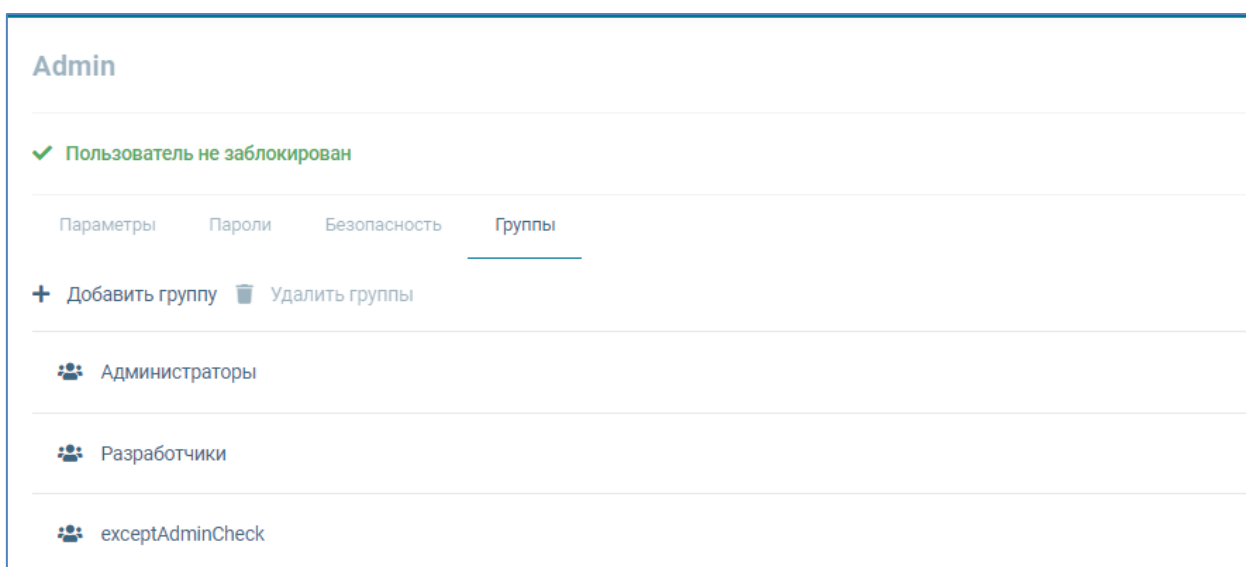


Рисунок 74 – Вкладка «Группы»

4. После того как будут заполнены обязательные поля, нажать кнопку «Сохранить» (Рисунок 75).

Admin

✓ Пользователь не заблокирован

Параметры Пароли Безопасность Группы

Логин
Admin

Имя
Admin

Фамилия
Test

Email
Значение не указано

Автозапуск

Поиск...
agrometeoservice

Сохранить Отмена

Рисунок 75 – Форма для создания нового пользователя

11.9.5.2 Токены пользователя для расчетов

При редактировании своей собственной информации, если у пользователя есть роль «Обновление себя», в интерфейсе становится доступна генерация токена для запуска расчетов через API (см. п. 9.3).

На вкладке «Токен» (Рисунок 76) пользователь может сгенерировать персональный ключ, с использованием которого можно запускать расчеты задач через вызов по API.

Если ключ еще не создавался – будет доступно создание нового. Если ключ уже был создан – будет доступно его удаление.

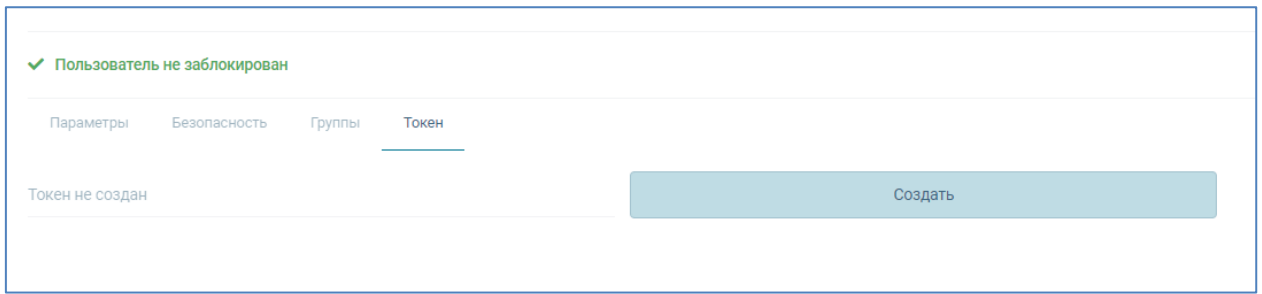


Рисунок 76 – Вкладка «Токен»

11.9.5.2.1 Создание токена

Для создания токена необходимо нажать на кнопку «Создать». При нажатии будет создан токен для запуска задачи на расчет через API (Рисунок 77). **Его необходимо скопировать и сохранить**, так как получить его еще раз нельзя.

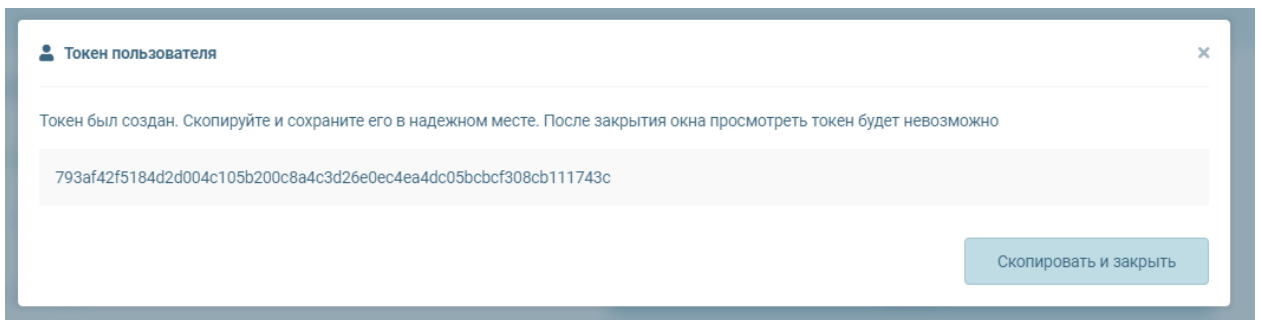


Рисунок 77 – Создание токена пользователя

После закрытия всплывающего окна повторно получить этот же токен невозможно. Доступно только удаление.

11.9.5.2.2 Удаление токена

Для удаления токена необходимо нажать на кнопку «Удалить».

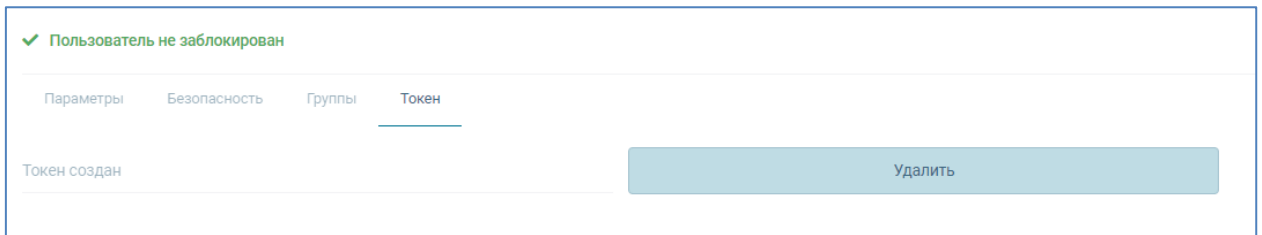


Рисунок 78 – Удаление токена пользователя

Удаление доступно только если токен был ранее создан.

После нажатия на кнопку «Удалить» необходимо подтвердить действие. Токен будет удален и появится возможность создать новый.

11.10 Управление ролями

Наименование	Разрешение
Группа	
groupCreate	Разрешение на создание групп пользователей
groupEdit	Разрешение на изменение групп пользователей

Наименование	Разрешение
groupDelete	Разрешение на удаление групп пользователей
groupRead	Разрешение на чтение групп пользователей
Пользователь	
userCreate	Разрешение на создание пользователей
userEdit	Разрешение на изменение пользователей
userEditSelf	Разрешение на изменение настроек своего профиля
userDelete	Разрешение на удаление пользователей
userRead	Разрешение на чтение информации пользователей
userSessions	Разрешение на просмотр списка активных пользователей
Граф	
graphCreate	Разрешение на создание графов\задач
graphEdit	Разрешение на изменение графов\задач
graphDelete	Разрешение на удаление графов\задач
graphRead	Разрешение на просмотр графов\задач
graphCalc	Разрешение на расчет графов\задач
graphCopy	Разрешение на копирование графов\задач
graphPythonEdit	Разрешение на редактирование блоков Python в графах\задачах
graphJSEdit	Разрешение на редактирование блоков JavaScript в графах\задачах
Пресет	
presetCreate	Разрешение на создание пресетов
presetEdit	Разрешение на изменение пресетов
presetDelete	Разрешение на удаление пресетов
presetRead	Разрешение на просмотр пресетов
presetCopy	Разрешение на копирование пресетов
Данные	
dataEdit	Разрешение на редактирование данных (writeback)
dataExportTable	Разрешение на экспорт данных из таблиц
Глобальные параметры соединений с БД	
connectionEdit	Разрешение на управление соединениями с БД
Глобальные параметры стилей 3D-карты	
map3dStyles	Разрешение на управление подложками 3D-карты
Глобальные параметры стилей меню	
menuStyles	Разрешение на управление стилями меню
Глобальные параметры тем	

Наименование	Разрешение
themesEdit	Разрешение на редактирование темами приложения
Глобальные константы	
constEdit	Разрешение на редактирование глобальных констант
Глобальные пароли	
globalPasswordEdit	Разрешение на редактирование глобальных паролей
Расчеты задач	
logCalcRead	Разрешение на просмотр списка расчетов задач
logCalcEdit	Разрешение на редактирование списка расчетов задач
Протокол доступа	
logAccessRead	Разрешение на просмотр протокола доступа
logAccessEdit	Разрешение на редактирование протокола доступа
Приложение	
allowSystemPanels	Разрешение на отображение системных панелей
Общие настройки	
optionsEdit	Разрешение на редактирование общих настроек
adminAccessEdit	Разрешение на редактирование доступа к объектам
adminExperimental	Разрешение на отображение экспериментальных функции
accessOptionsEdit	Разрешение на управление доступом
adminAccess	Роль Администратора
adminBlockingUsers	Разрешение на блокирование пользователей

11.10.1 Примечание

Права на роли собираются из прав на группы и пользователей:

Наименование	Описание
groupCreate и userCreate	Разрешение на создание ролей
groupEdit и userEdit	Разрешение на изменение ролей
groupDelete и userDelete	Разрешение на удаление ролей
groupRead и userRead	Разрешение на чтение ролей

11.11 Управление соединениями с БД

В приложении есть глобальные соединения с БД, которые можно использовать внутри блока «Соединение с БД».

11.11.1 Настройка соединений с БД

Для осуществления настроек необходимо открыть раздел «Соединение с БД» в локальном меню пользователя (Рисунок 79).

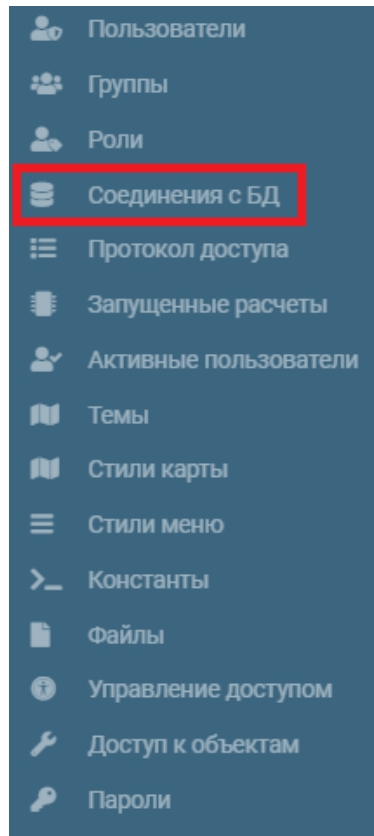


Рисунок 79 – Раздел меню «Соединение с БД»

На странице «Соединение с БД» слева расположен список соединений с БД. Если выделить соединение с БД, то справа открываются его настройки (Рисунок 80).

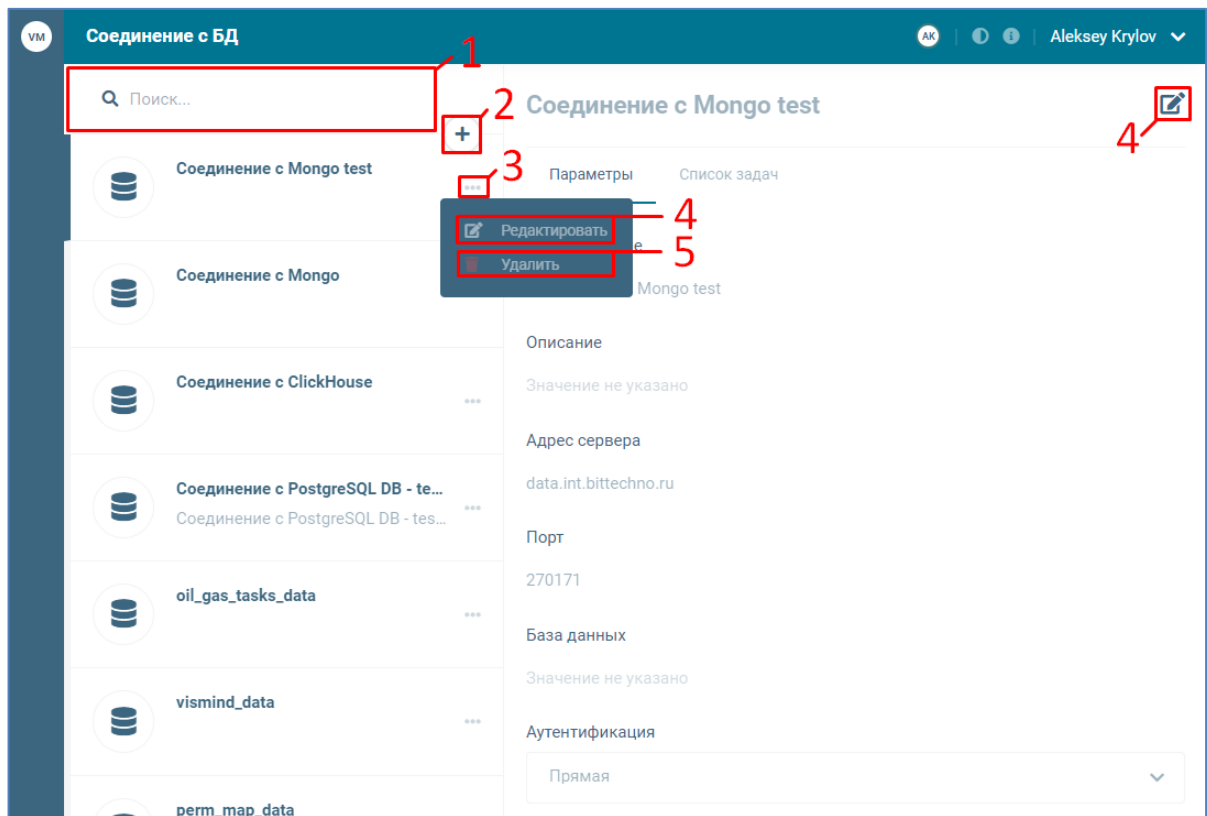


Рисунок 80 – Настройка соединения с БД

На странице «Соединение с БД» доступен следующий функционал:

1. Поиск соединения с БД.
2. Создание соединения с БД.
3. Открыть контекстное меню.
4. Редактировать соединение с БД.
5. Удалить соединение с БД.

На вкладке «Параметры» расположен перечень доступных настроек.

1. Наименование.
2. Описание.
3. Адрес сервера (Host).
4. Порт.
5. База данных.
6. Аутентификация:
 - Логин / Пароль;
 - Прямая.

Параметры Список задач

1 Наименование
Соединение с Mongo test

2 Описание
Значение не указано

3 Адрес сервера
data.int.bittechno.ru

4 Порт
270171

5 База данных
Значение не указано

6 Аутентификация
Прямая

Рисунок 81 – Настройка параметров соединения с БД

После того как все необходимые настройки указаны, перед сохранением можно нажать кнопку «Проверка соединения», чтобы проверить корректность работы соединения (Рисунок 82).

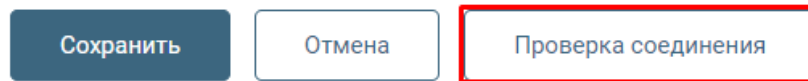


Рисунок 82 – Кнопка «Проверка соединения»

На вкладке «Список задач» (Рисунок 83) отображается список задач, в которых используется данное соединение.

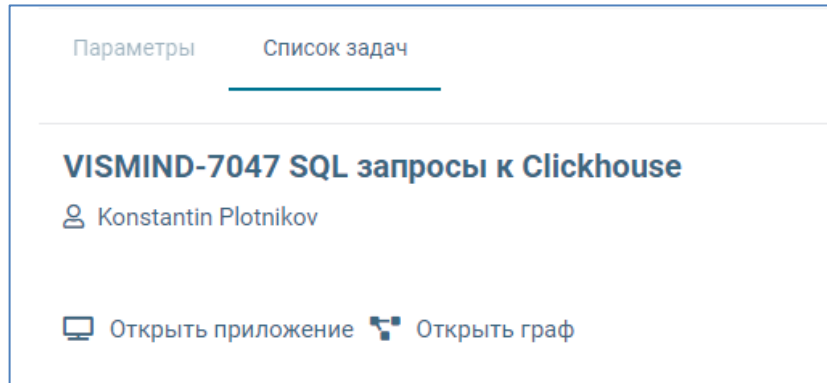


Рисунок 83 – Вкладка «Список задач»

11.11.2 Использование соединений с БД

Выбор настроенных соединений с БД осуществляется в настройках блоков, в которых осуществляется подключение к базе данных (такие как «Соединение с PostgreSQL DB», «Соединение с Microsoft SQL DB» и т.д.) (Рисунок 84).

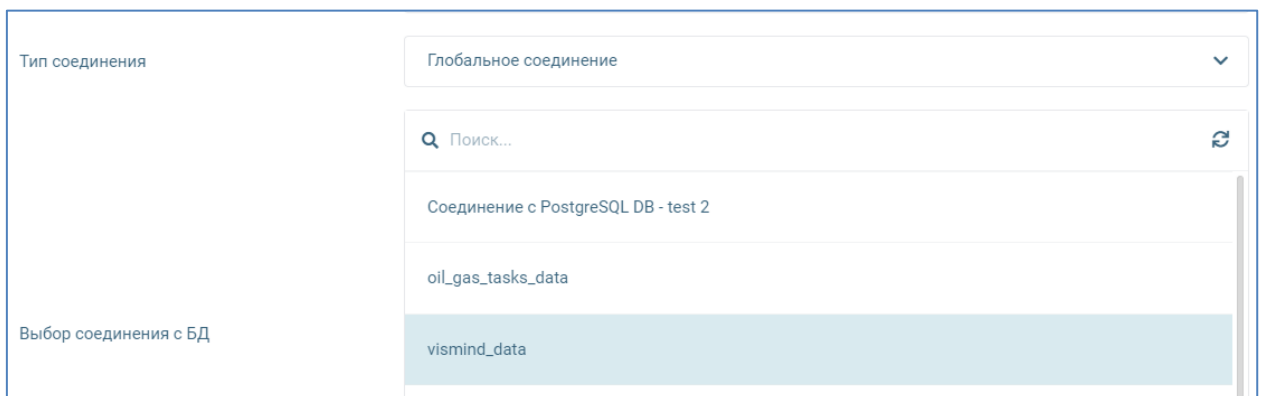


Рисунок 84 – Настройка соединений с БД

Чтобы использовать настроенное соединение с БД необходимо:

1. Открыть граф, в котором нужно настроить соединение.
2. Добавить блок из библиотеки, например: «Соединение с PostgreSQL DB».
3. В пункте «Тип соединения» выбрать значение из выпадающего списка «Глобальное соединение».
4. В пункте «Выбор соединения с БД» выбрать настроенное соединение.
5. Сохранить настройки блока.
6. Рассчитать блок.

11.11.2.1 Глобальное соединение

Для добавления нового источника данных (глобальное соединение с БД) необходимо:

1. Открыть меню в правом верхнем углу рабочей области.
2. Выбрать раздел «Соединения с БД» (Рисунок 85).

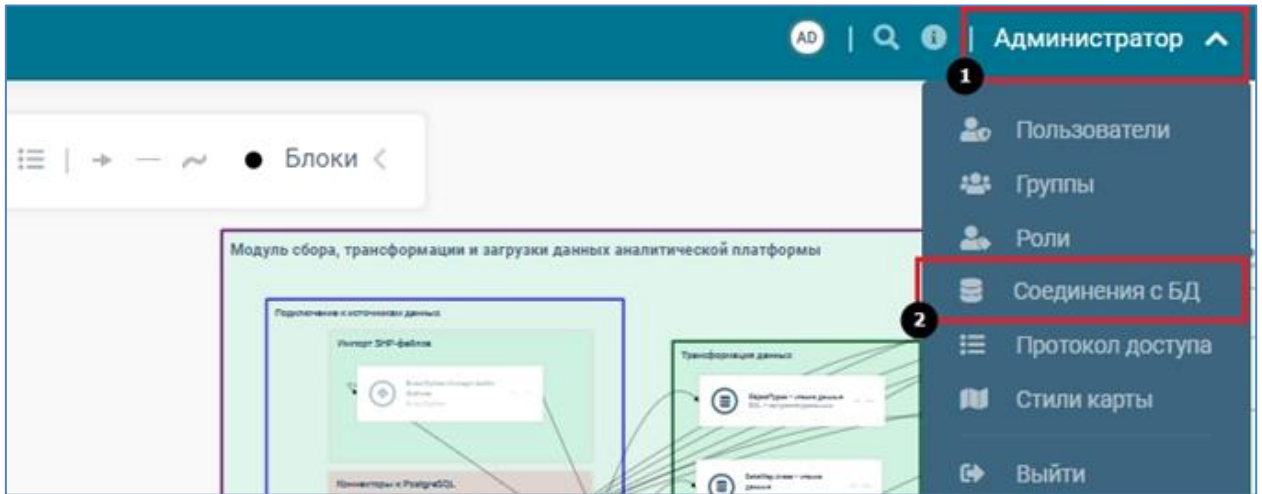



Рисунок 85 – Настройка нового источника данных (глобальное соединение с БД)

3. Добавить новое соединение, нажав на кнопку  (Рисунок 86).
4. Выбрать вид соединения из списка: Соединение с PostgreSQL DB, Соединение с Microsoft SQL DB, Соединение с Oracle DB (чаще всего используется соединение с БД PostgreSQL DB).

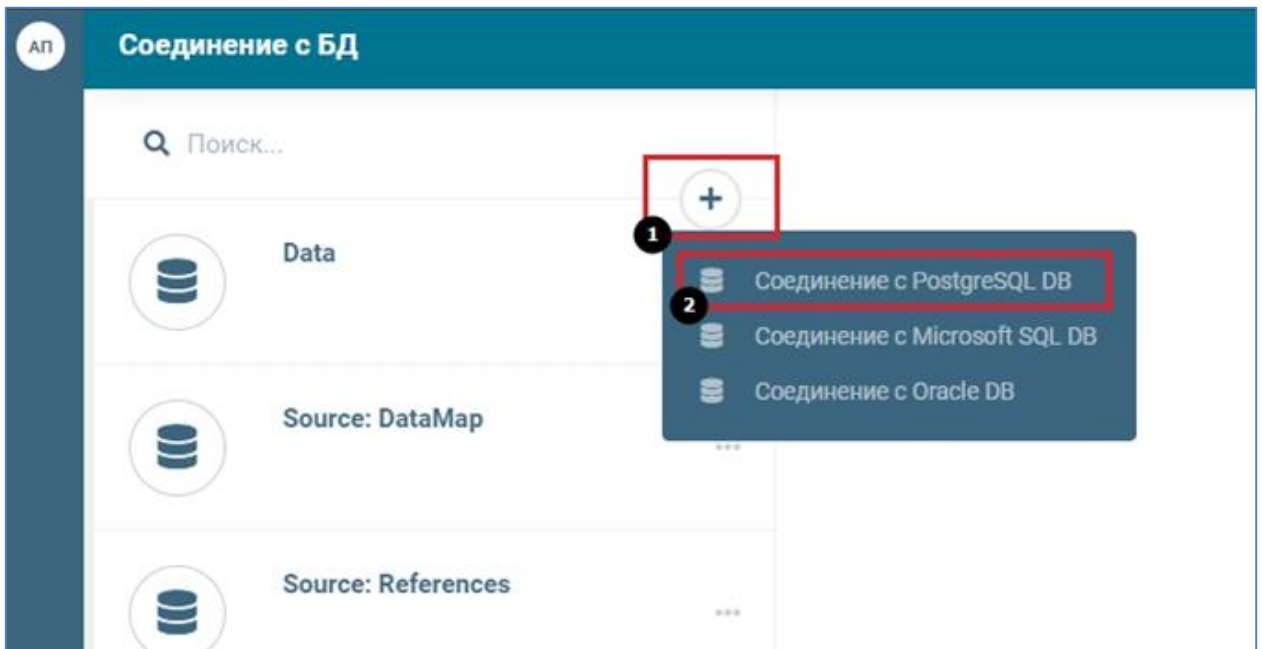


Рисунок 86 – Выбор вида соединения «Соединение с PostgreSQL DB»

5. Прописать настройки соединения с БД (наименование соединения, адрес сервера, порт, наименование БД, имя пользователя и пароль) и сохранить изменения (Рисунок 87).

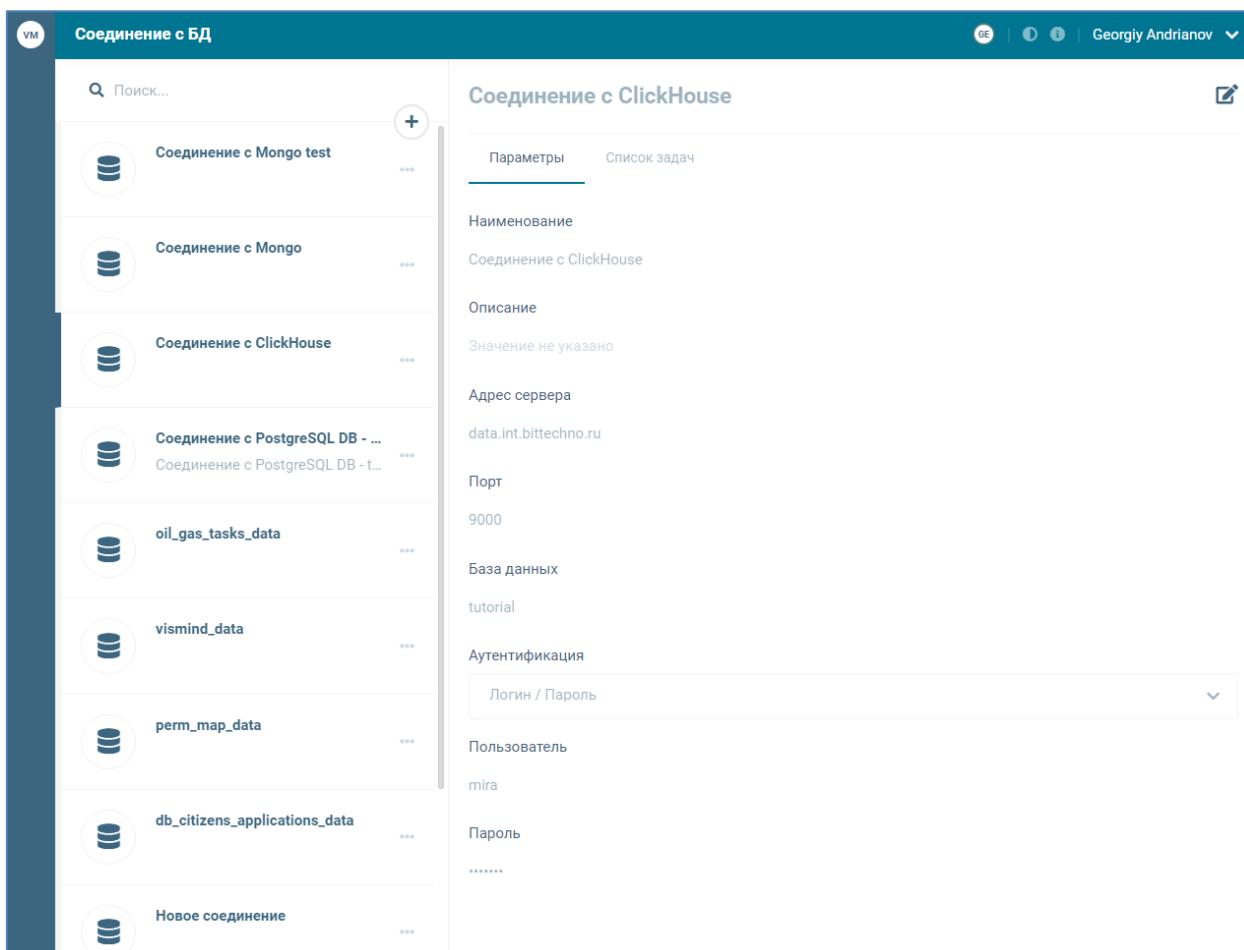


Рисунок 87 – Страница «Соединение с БД»

6. Вернуться в рабочую область графа, где необходимо добавить соединение с БД.
7. В рабочей области графа добавить блок «Соединение с PostgreSQL DB» из библиотеки блоков.
8. Открыть настройки блока «Соединение с PostgreSQL DB», выбрать тип соединения – глобальное соединение.
9. Выбрать соединение с БД, настройки которой были прописаны ранее в разделе «Соединение с БД».
10. Проверить соединение, нажав на соответствующую кнопку внутри настроек блока.
11. Сохранить изменения настроек блока, нажав на соответствующую кнопку (Рисунок 88).

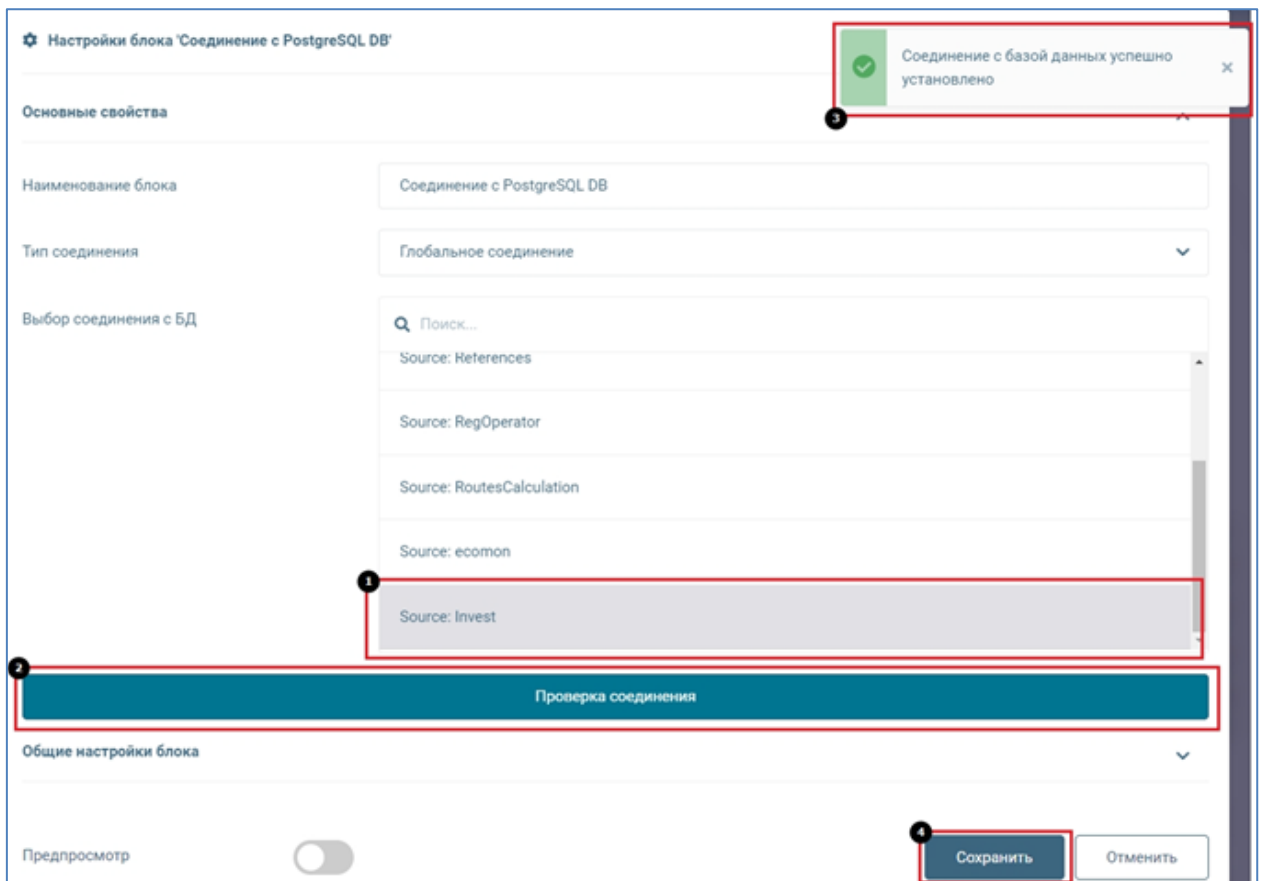


Рисунок 88 – Настройка блока «Соединение с PostgreSQL DB»

12. Открыть меню блока, нажав на кнопку  в правой части блока, затем нажать на соответствующую кнопку для расчета блока (Рисунок 89).

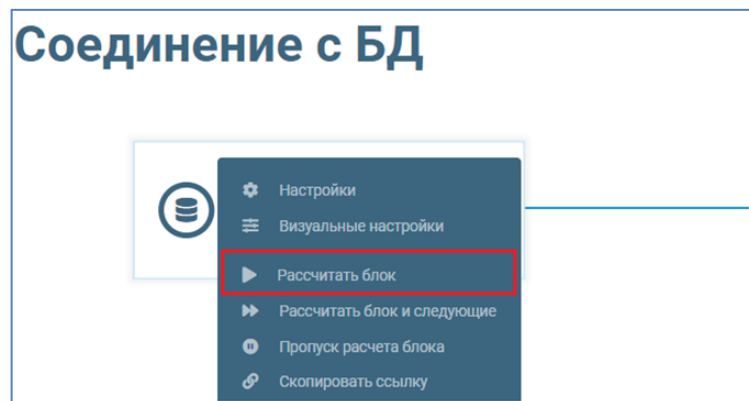


Рисунок 89 – Кнопка «Рассчитать блок»

13. Добавить блок «SQL – запрос-справочник» из библиотеки блоков на рабочую область графа.
14. Связать выход «БД» блока «Соединение с PostgreSQL DB» с входом «БД» блока «SQL – запрос-справочник» (Рисунок 90).

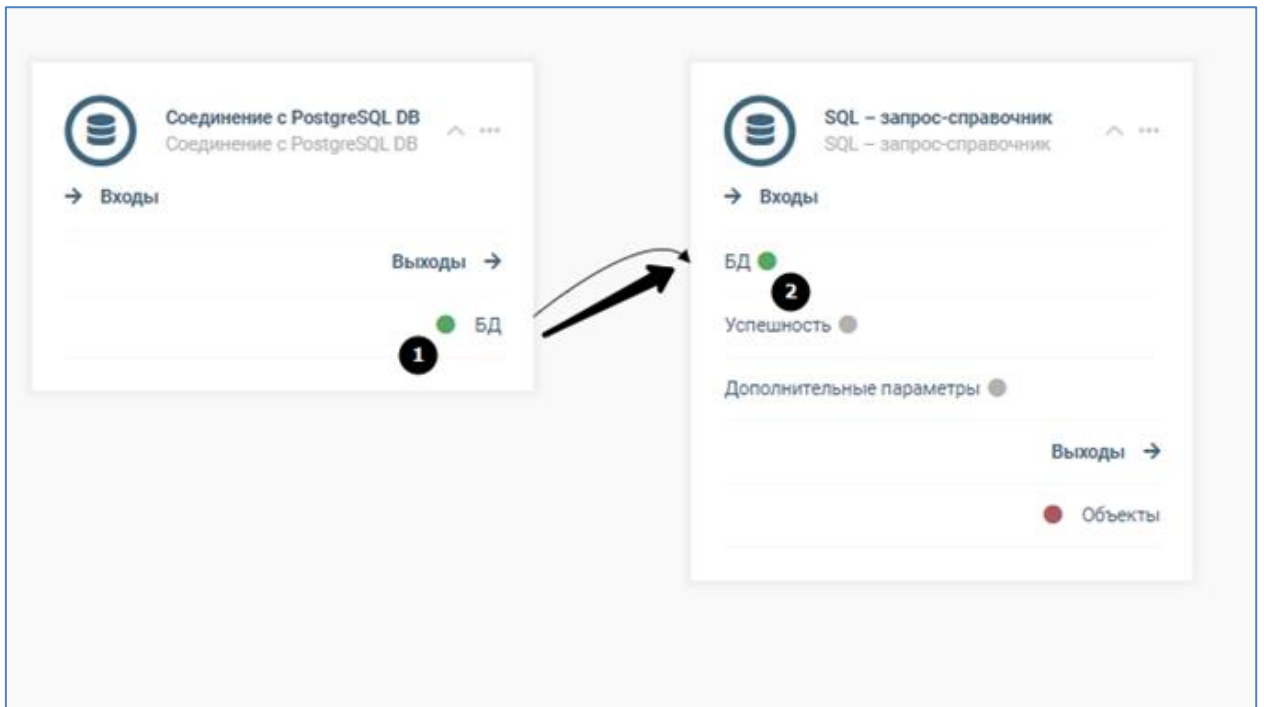


Рисунок 90 – Пример настройки связи блока

15. Открыть настройки блока «SQL – запрос-справочник».
16. Внутри настроек блока написать запрос для получения необходимых данных из источника данных.
17. Сохранить изменения настроек блока, нажав на соответствующую кнопку, и рассчитать блок.

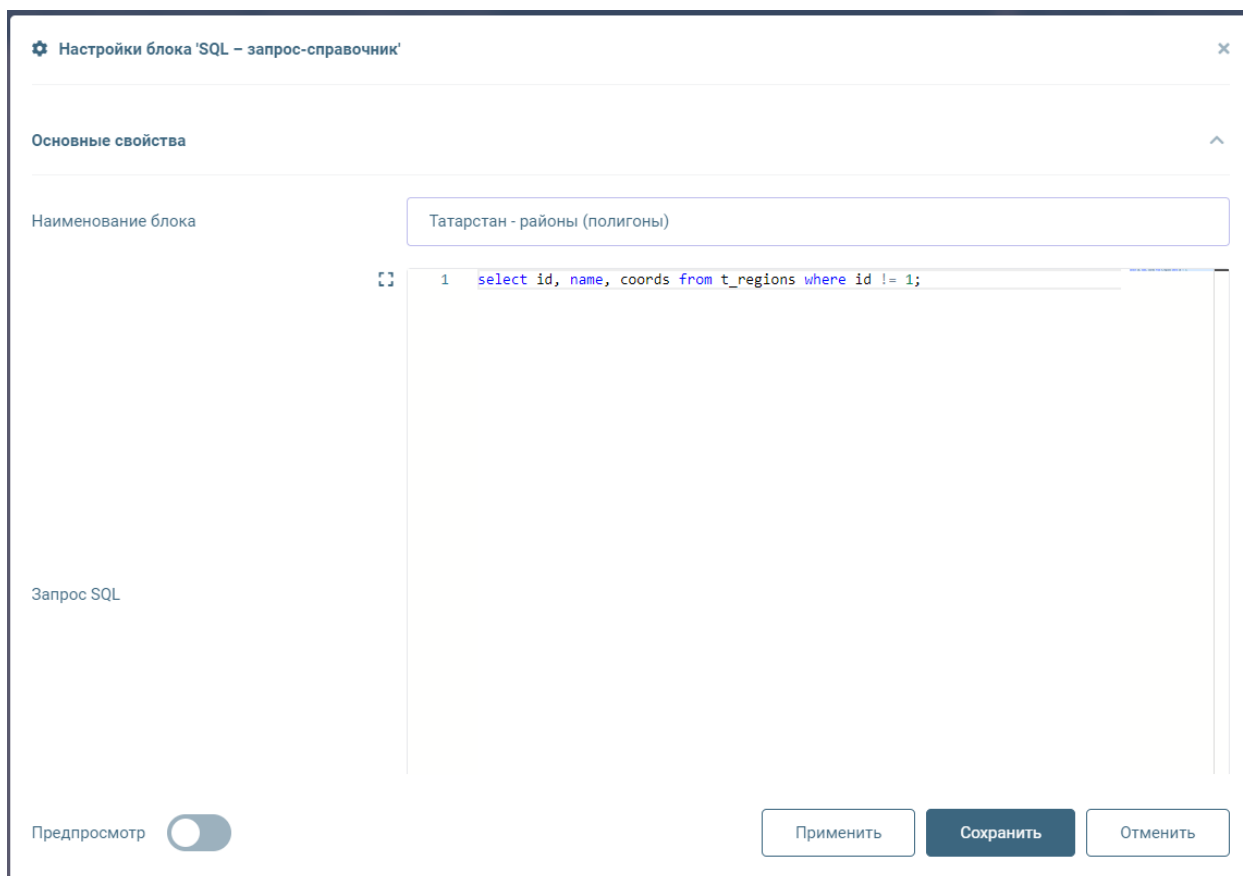


Рисунок 91 – Настройка блока «SQL – запрос-справочник»

11.11.2.2 Задание параметров вручную

Для добавления нового источника данных (задание параметров вручную) необходимо (Рисунок 92):

1. В рабочей области графа добавить блок «Соединение с PostgreSQL DB» из библиотеки блоков.
2. Открыть настройки блока «Соединение с PostgreSQL DB», выбрать тип соединения – задать параметры соединения вручную.
3. Задать вручную тип соединения, адрес сервера, порт, наименование БД, имя пользователя и пароль.
4. Проверить соединение, нажав на соответствующую кнопку внутри настроек блока.
5. Сохранить изменения настроек блока, нажав на соответствующую кнопку.

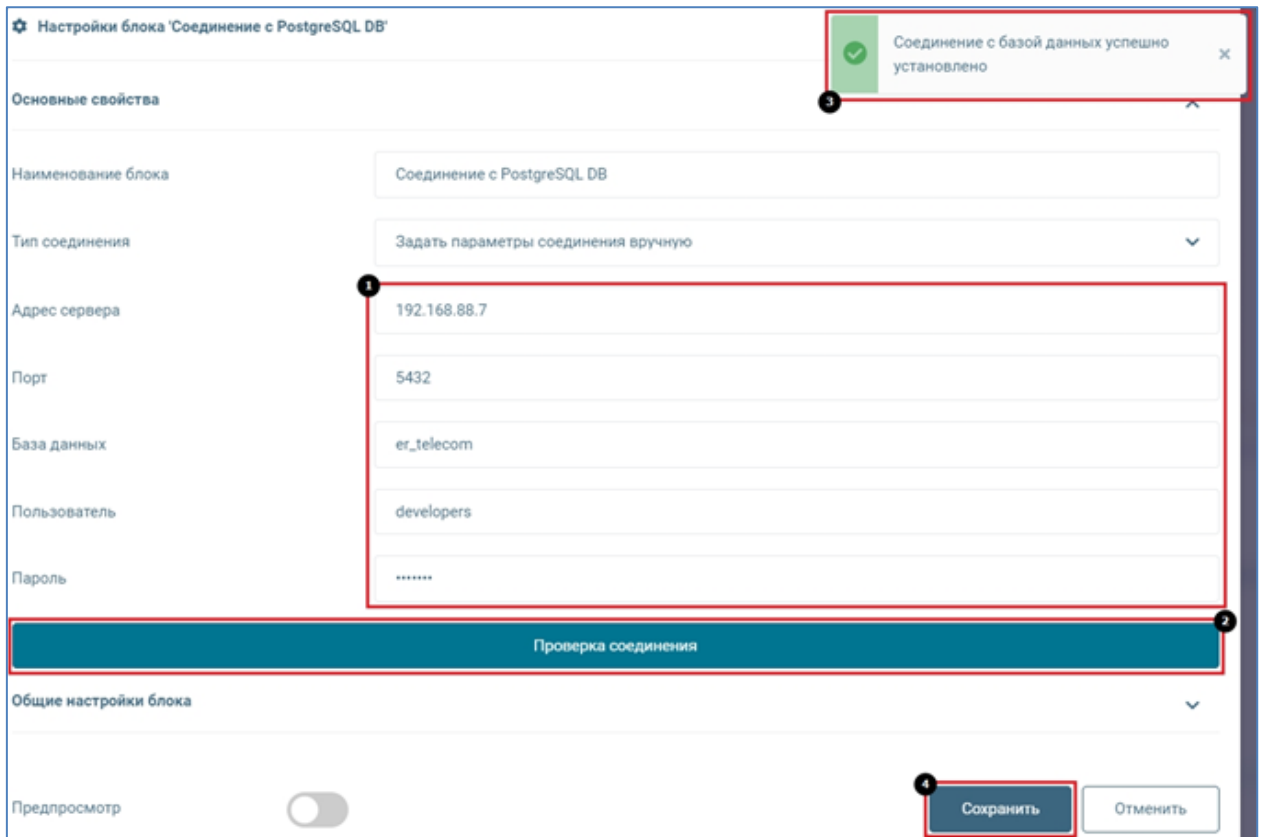



Рисунок 92 – Настройка блока «Соединение с PostgreSQL DB»

- б. Открыть меню блока, нажав на кнопку  в правой части блока, и нажать на соответствующую кнопку для расчета блока (Рисунок 93).

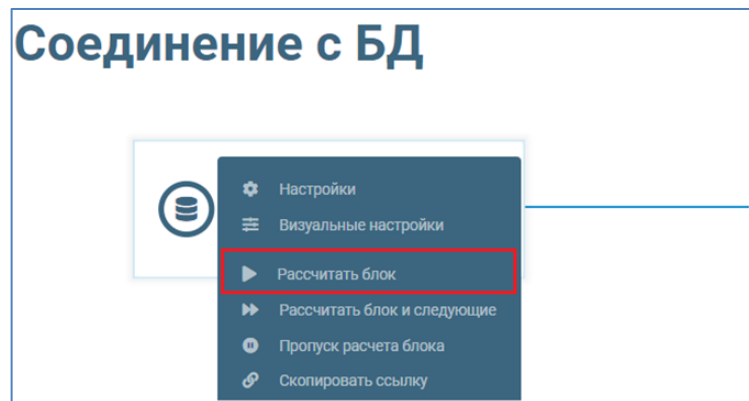


Рисунок 93 – Кнопка «Расчитать блок»

7. Добавить блок «SQL – запрос-справочник» из библиотеки блоков на рабочую область графа.
8. Связать выход «БД» блока «Соединение с PostgreSQL DB» с входом «БД» блока «SQL – запрос-справочник» (Рисунок 94).

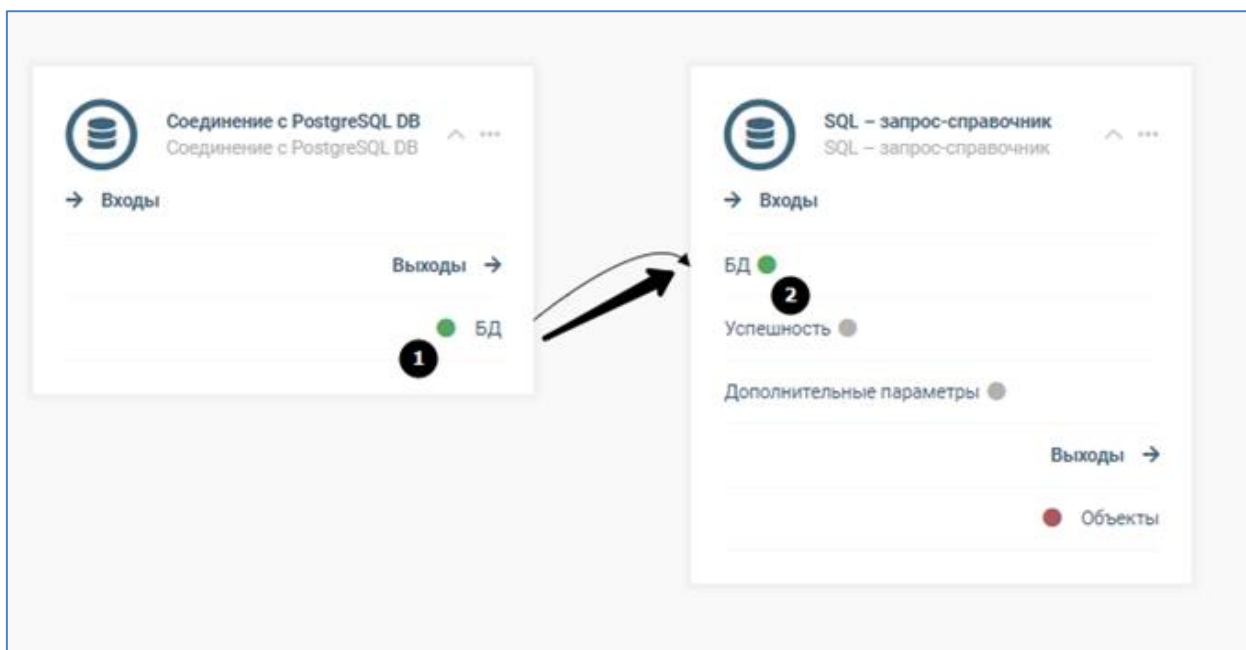


Рисунок 94 – Настройка нового источника данных

9. Открыть настройки блока «SQL – запрос-справочник» (Рисунок 95).
10. Внутри настроек блока написать запрос для получения необходимых данных из источника данных.
11. Сохранить изменения настроек блока, нажав на соответствующую кнопку, и рассчитать блок.

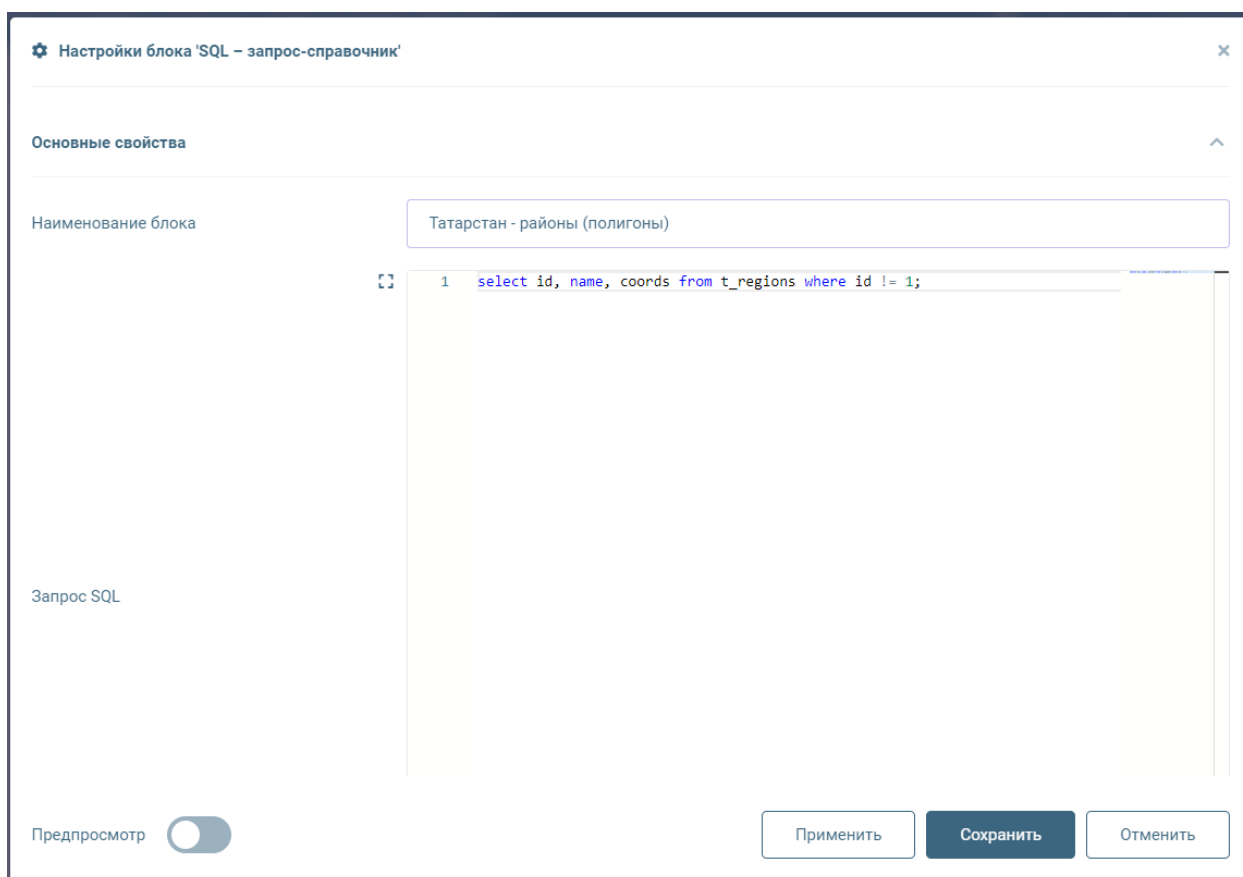


Рисунок 95 – Настройка блока «SQL – запрос-справочник»

11.12 Мониторинг

Для мониторинга состояния приложения и его внутренних активностей доступен эндпоинт (endpoint) с метриками Prometheus. Для работы с Prometheus используются библиотеки: [prometheus client](#), [prometheus flask exporter](#).

Описание Prometheus: <https://prometheus.io/docs/introduction/overview/>.

Из приложения метрики доступны по API: <http://application/metrics>.

Метрики могут быть закрыты паролем с использованием basic HTTP auth.

Список доступных метрик по умолчанию (prometheus_flask_exporter) приведен в Приложении 3.

Специфические метрики приложения, добавленные помимо основных метрик:

- статические (собираются при запуске приложения)
 - vismind_build{version}- информация о версии приложения
 - vismind_metrics_calculations- количество активных расчетов с разбивкой по типу расчета (type_calculation)
 - Обновляется когда забирается метрика.
 - Аналогично количеству активных расчетов на странице "Текущие расчеты"
 - Возможные варианты:
 - Graph - расчет графа
 - Block - расчет блока
 - Blocks - расчет блоков
 - Event - расчет события
 - Branch - расчет ветки
 - Branches - расчет веток
 - Preset - расчет пресета
 - Preview - расчет предпросмотра
 - BlockAllPath - расчет блока и перерасчет всех блоков перед ним
 - MasterAutoFactor - расчет мастера автоподбора в линейной регрессии
 - MasterArimaAnalysis - анализ временных рядов для аримы
 - MasterArimaAuto - расчет мастера аналитики аримы
 - PresetOptimizationAnalysis - анализ оптимизации пресета
 - BlockSave - сохранения
 - BlockSaveAttribute - сохранение только атрибутов
 - vismind_metrics{name="tokens"} - количество активных токенов
 - Обновляется когда забирается метрика.

- динамические (собираются при запуске приложения и обновляются во время использования приложения)
 - vismind_metrics{name="open_presets"} - количество открытых пресетов.
 - Обновляется когда какой-либо пользователь заходит на пресет или когда из пресета выходит.
 - vismind_metrics{name="auth_correct"} - количество корректных авторизаций
 - Обновляется когда какой-либо пользователь корректно авторизуется в системе
 - При получении метрики показатель обнуляется
 - vismind_metrics{name="auth_not_correct"} - количество ошибочных авторизаций
 - Обновляется когда какой-либо пользователь не проходит авторизацию в системе
 - При получении метрики показатель обнуляется
 - vismind_metrics_users - количество пользователя в комнате
 - login - логин пользователя
 - room - в которой находится пользователь
 - Варианты:
 - если начитается с Р - это комната пресета и его id
 - если начитается с Т - это комната графа и его id
 - если начитается с А - это комната администрирования
 - Обновляется когда какой-либо пользователь заходит в комнату или когда выходит из нее
 - vismind_metrics_users_operation - количество операций по пользователю
 - login - логин пользователя
 - Обновляется когда какой-либо пользователь вызывает любую апи
 - При получении метрики показатель обнуляется

Метрики по умолчанию (prometheus_flask_exporter)

```
# HELP build_version_info Multiprocess metric
# TYPE build_version_info gauge
build_version_info{version="0.37.0.18276-241f3ba6-develop-09.08.2023 11:01:30-coordinator"} 1.0
# HELP count_for_sum Multiprocess metric
# TYPE count_for_sum gauge
count_for_sum{count="calculations"} 0.0
count_for_sum{count="open_presets"} 1.0
```

Из-за того, что приложение работает на нескольких gunicorn workers, динамические метрики могут обновляться с запозданием или быть неточными.

11.13 Дополнительные сервисы

11.13.1 Геосервер Martin

Для работы с картой иногда необходимо использовать геосервер, который позволяет получать данные напрямую из БД на слои в виде векторных MVT тайлов.

Геосервер Martin представляет собой быстрый сервер для получения из таблиц БД с помощью PostGIS функций данных в нужном формате. Слой, в этом случае, описывается стилем как обычный слой, но вместо данных ему указывается ссылка на этот слой в Martin.

Для добавления необходимо в БД **gisdb** в любой таблице добавить поле с **SRID** и геометрией. После этого вызвать для геосервера обновление источников данных (если он запущен с переменной `watch_mode = true`), обратившись к `/index.json`

Важно! Для ускорения выполнения запросов, особенно на таблицах, содержащих много данных, необходимо создать **gist** индекс для поля с геометрией.

11.13.1.1 Кэширование данных с помощью Nginx прокси-сервера

Для обеспечения кэширования слоев и снижения нагрузки на БД необходимо настроить кэширующий прокси-сервер, например, Nginx.

Для запуска можно использовать следующий конфигурационный файл:

nginx.conf

```
user nginx;
worker_processes auto;

error_log /var/log/nginx/error.log debug;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    tcp_nopush on;

    keepalive_timeout 65;

    gzip on;
    gzip_vary on;
    gzip_proxied any;
    gzip_comp_level 6;
    gzip_buffers 16 8k;
```

```

gzip_http_version 1.1;
gzip_min_length 256;
gzip_types          text/plain text/css text/xml text/javascript
application/x-javascript
                    application/xml application/javascript application/json
application/x-protobuf;

proxy_cache_path    /var/cache/nginx/
                    levels=1:2
                    max_size=10g
                    inactive=60m
                    use_temp_path=off
                    keys_zone=tiles_cache:10m;

upstream martin_upstream {
    server martin:3001;
}

server {
    listen 3001 ssl;
    listen 3000;
    server_name localhost:3000;
    ssl_certificate /etc/ssl/certs/int.bittechno.ru.crt;
    ssl_certificate_key /etc/ssl/private/int.bittechno.ru.key;

    location ~ /static/(?<fwd_path>.*) {
        proxy_set_header    X-Forwarded-Host $host:$server_port;
        proxy_set_header    X-Forwarded-Proto $scheme;
        proxy_set_header    Host $host;
        proxy_set_header    X-Real-IP $remote_addr;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header    X-Rewrite-URL $uri;
        proxy_redirect       off;

        proxy_connect_timeout 15m;
        proxy_send_timeout    15m;
        proxy_read_timeout    15m;
        send_timeout           15m;

        proxy_cache           tiles_cache;
        proxy_cache_lock      on;
        proxy_cache_revalidate on;

        # Set caching time for responses
        proxy_cache_valid     200 204 302 24h;
        proxy_cache_valid     404 1m;

        proxy_cache_use_stale error timeout http_500 http_502 http_503
http_504;
        add_header            X-Cache-Status $upstream_cache_status;

        proxy_pass
http://martin_upstream/$fwd_path$is_args$args;
    }
}

```

```

location ~ /dynamic/(?<fwd_path>.*) {
    proxy_set_header    X-Forwarded-Host $host:$server_port;
    proxy_set_header    X-Forwarded-Proto $scheme;
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Rewrite-URL $uri;
    proxy_redirect      off;

    proxy_connect_timeout 15m;
    proxy_send_timeout    15m;
    proxy_read_timeout    15m;
    send_timeout          15m;

    proxy_pass
http://martin_upstream/$fwd_path$is_args$args;

}

location ~ /(?!<fwd_path>.*) {
    proxy_set_header    X-Forwarded-Host $host:$server_port;
    proxy_set_header    X-Forwarded-Proto $scheme;
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Rewrite-URL $uri;
    proxy_redirect      off;

    proxy_connect_timeout 15m;
    proxy_send_timeout    15m;
    proxy_read_timeout    15m;
    send_timeout          15m;

    proxy_pass
http://martin_upstream/$fwd_path$is_args$args;

}
}
}

```

Запуск всего в docker-compose

docker-compose.yml

```

version: '3'

services:
  nginx:
    image: nginx:alpine
    restart: unless-stopped
    ports:
      - 3000:3000
      - 3001:3001
    volumes:
      - ./cache:/var/cache/nginx
      - ./nginx.conf:/etc/nginx/nginx.conf:ro
      - ./int.bittechno.ru.crt:/etc/ssl/certs/int.bittechno.ru.crt:ro

```

```

- ./int.bittechno.ru.key:/etc/ssl/private/int.bittechno.ru.key:ro
depends_on:
- martin

martin:
  image: urbica/martin
  volumes:
  - ./martin.yml:/etc/martin.yml:ro
  command: martin --config=/etc/martin.yml
  expose:
  - 3001
  restart: unless-stopped

```

11.13.1.1.1 Относительная адресация при запросах к геосерверу

Если в прокси-сервере для клиентского приложения настроено проксирование относительных адресов геосервера (см. п. 4.7.2) – на работу самого геосервера это не повлияет. Самое главное, чтобы прокси-сервер корректно пробрасывал (перенаправлял) все заголовки оригинального запроса.

11.13.1.1.2 Реализация в инфраструктуре

Предусматривается два вида слоев:

1. Если данные меняются – слой считается динамическим и его нельзя кэшировать.
2. Если данные статичны – слой статичен и не меняется – в этом случае кэширование нужно.

Основная цель кэширования – снизить нагрузку на БД и ускорить получение данных. На тестах прирост загрузки статичных кэшированных слоев составил +300%.

Для подключения слоя с кэшем надо использовать адрес: /static, например: /static/test.mp_buildings_test.json.

Для работы со слоем без кэша: /dynamic, например: /dynamic/test.mp_buildings_test.json.

Важно! В БД по умолчанию вход осуществляется под пользователем **martin** с ролью **readonly** и доступом только к **gisdb**. При добавлении новых схем необходимо выдавать пользователю права на них.

Кэш валиден 24 часа.

11.13.1.1.3 Дополнительные ссылки

Геосервер Martin – <https://github.com/maplibre/martin>

Стиль maplibre – <https://maplibre.org/maplibre-gl-js-docs/style-spec/>

PostGIS – <https://postgis.net/>

Служебные функции Postgres – <https://github.com/mapbox/postgis-vt-util>

11.13.2 Пулер соединений БД pgBouncer

В PostgreSQL для обработки каждого соединения клиента создается отдельный процесс. Чем больше число соединений, тем больше процессов, которые используют оперативную память.

Для управления соединениями клиентов можно использовать пулер соединений. Клиенты подключаются не напрямую к PostgreSQL, а к пулеру соединений. При этом поддерживается небольшое количество подключений между пулером и сервером PostgreSQL – пулер создает новое соединение или повторно использует одно из существующих.

11.13.2.1 Установка и конфигурация

11.13.2.1.1 Установка

Установка выполняется командой:

```
sudo apt-get install pgbouncer
```

11.13.2.1.2 Конфигурация

Пример файла /etc/pgbouncer/userlist.txt с перечнем разрешенных пользователей в формате пользователь и отпечаток пароля в формате md5 или SCRAM-SHA-256:

```
"djangoapp" "md512345659d2636855c09af03933123456"
"nodeapp" "md512345659d2636123456af03933123456"
```

Выгрузка списка пользователей:

```
psql -Atq -U postgres -d postgres -c "SELECT concat('\\"', username, '\\\" \\'',
passwd, '\\\"') FROM pg_shadow"
```

/etc/pgbouncer/pgbouncer.ini – основной файл конфигурации работы пуллера соединений. Задаёт режимы работы пула, размеры пулов и [прочие параметры](#). Можно использовать настройки по умолчанию **default** для всего **pgbouncer** или задавать разные параметры для работы конкретных баз и пользователей.

11.13.2.2 Реализация в инфраструктуре

Pgbouncer запускается рядом с Postgres на виртуальной машине.

Конфигурационный файл /etc/pgbouncer/pgbouncer.ini

```
[databases]
```

```
; содержит список баз и параметры для них. Параметры будут переопределять
настройки из блока [pgbouncer]
```

```
; для двух баз ниже задаем определенные размеры пулов соединений
```

```
repo_vismind_test    = host=localhost    port=5432    dbname=repo_vismind_test
pool_mode=transaction pool_size=100
repo_vismind_stage   = host=localhost    port=5432    dbname=repo_vismind_stage
pool_mode=transaction pool_size=70
```

```
; создаем псевдонимы для баз Pgbouncer для всех остальных баз Postgres с
сохранение исходных имен, для них размеры пулов и т.д. будут устанавливаться по
умолчанию
```

```
* = host=localhost port=5432
```

```
[users]
```

```
; содержит список пользователей и параметры для них. Параметры будут
переопределять настройки из блока [pgbouncer]
```

```
; для пользователя martin задаем режим session - обусловлено запросами какие
посылает Martin
```

```
martin = pool_mode=session
```

```
[pgbouncer]
```

```
; основные настройки pgbouncer
```

```
logfile = /var/log/postgresql/pgbouncer.log
```

```
pidfile = /var/run/postgresql/pgbouncer.pid
```

```

;;;
;;; Where to wait for clients
;;;

;; IP address or * which means all IPs
;; Список IP и порт по умолчанию 6432
listen_addr = *
listen_port = 6432

unix_socket_dir = /var/run/postgresql

;; тип авторизации клиентов
auth_type = scram-sha-256
auth_file = /etc/pgbouncer/userlist.txt

;; comma-separated list of users who are allowed to change settings
;; список пользователей кто может управлять pgbouncer
admin_users = postgres

;; When server connection is released back to pool:
;; session - after client disconnects (default)
;; transaction - after transaction finishes
;; statement - after statement finishes
pool_mode = transaction

;; Comma-separated list of parameters to ignore when given in startup
;; packet. Newer JDBC versions require the extra_float_digits here.
ignore_startup_parameters = extra_float_digits

;; Total number of clients that can connect
max_client_conn = 10000

;; Default pool size. 20 is good number when transaction pooling
;; is in use, in session pooling it needs to be the number of
;; max clients you want to handle at any moment
default_pool_size = 50

;; Minimum number of server connections to keep in pool.
min_pool_size = 10

; how many additional connection to allow in case of trouble
reserve_pool_size = 20

;; If a clients needs to wait more than this many seconds, use reserve
;; pool.
reserve_pool_timeout = 2

```

Для пользователя **martin** выбран режим **session**, потому что Martin посылает операторы PREPARE.

11.13.2.3 Ограничения режима transaction

Режим **transaction** нарушает работу некоторых механизмов PostgreSQL. Выберите другой режим, если клиенты используют эти опции. Некоторые флаги подключения могут распределяться между разными клиентами – это может привести к непредсказуемому поведению и некорректным результатам.

В режиме **transaction** не работают:

- команды SET/RESET и LISTEN/NOTIFY;
- WITH HOLD CURSOR;
- PRESERVE/DELETE ROWS во временных таблицах;
- подготовленные операторы (prepared statements): protocol-level prepared plans, PREPARE, DEALLOCATE;
- оператор LOAD;
- рекомендательные блокировки Session-level advisory locks.

Подробнее приведено в [списке несовместимых опций](#).

11.13.2.4 Управление и взаимодействие с PgBouncer

Подключится к виртуальной базе pgbouncer под пользователем из списка admin_users, перечисленных в **pgbouncer.ini**:

```
sudo su postgres
psql -p 6432 pgbouncer
```

11.13.2.4.1 Основные команды

Основные команды:

- show pool – просмотр текущих подключений и параметров (Рисунок 96);
- show databases – просмотр размеров пулов соединений для баз данных (Рисунок 97);
- show config – позволяет сравнить текущие настройки с настройками по умолчанию для pgbouncer;
- PAUSE – pgbouncer пытается отключиться от всех серверов, сначала ожидая завершения всех запросов. Выполнение этой команды заканчивается, только когда завершаются все запросы. Эта команда должна применяться во время перезапуска базы данных.

Если указано имя базы данных, будет приостановлена работа только с ней. Новые подключения клиентов к базе, поставленной на паузу, будут находиться в состоянии ожидания, пока не будет выполнена команда RESUME.

- RELOAD – указывает процессу pgbouncer перезагрузить файл конфигурации и обновить значения изменяемых параметров. Позволяет менять режим работы pgbouncer налету после изменения в конфигурационном файле без рестарта службы pgbouncer.

```
pgbouncer=# show pools;
```

database	user	cl_active	cl_waiting	cl_active_cancel_req	cl_waiting_cancel_req	sv_active	sv_active_cancel
db_ertelecom_demo	martin	20	0	0	0	0	0
db_small_business	martin	23	0	0	0	0	0
pgbouncer	pgbouncer	1	0	0	0	0	0
postgres	postgres	0	0	0	0	0	0
repo_vismind_stage	vmbackend	11	0	0	0	0	0
repo_vismind_test	ashilkov	0	0	0	0	0	0
repo_vismind_test	vmbackend	22	0	0	0	0	0

(7 rows)

Рисунок 96 – Просмотр текущих подключений и параметров

name	host	port	database	force_user	pool_size	min_pool_size	reserve_pool	pool_mode
db_ertelecom_demo	localhost	5432	db_ertelecom_demo		50	10	20	
db_small_business	localhost	5432	db_small_business		50	10	20	
pgbouncer		6432	pgbouncer	pgbouncer	2	0	0	statement
repo_vismind_stage	localhost	5432	repo_vismind_stage		70	10	20	transaction
repo_vismind_test	localhost	5432	repo_vismind_test		100	10	20	transaction

Рисунок 97 – Просмотр размеров пулов соединений для баз данных

11.13.2.5 Дополнительные ссылки

Документация на русском PostgresPro – <https://postgrespro.ru/docs/postgrespro/15/pgbouncer>
 Официальная документация – <https://www.pgbouncer.org/config.html>

11.14 Время в приложении

В логах платформы и в системных разделах время с сервера передается в формате UTC0 "YYYY-MM-DDTHH:mi:SSZ", после чего клиент преобразует его в формат времени клиента с учетом временного пояса. Время при работе прикладного функционала (визуализаторы, фильтры данных) может иметь произвольный формат и не изменяется в зависимости от часового пояса.

11.15 Конфигурация приложения

11.15.1 Расположение

Конфигурационный файл находится в каталоге ./conf/config.toml

11.15.2 Описание формата конфигурационного файла

Конфигурационный файл разбит на разделы и подразделы.

Обозначение начала раздела конфигурации:

```
#-----  
#----- Настройки авторизации и аутентификации -----  
#-----
```

Обозначение начала подраздела конфигурации:

```
# --- Настройка токена ---
```

Название переменных в конфигурационном файле должны быть уникальными.

Важно! При наличии двух одинаковых записей будет применена последняя считанная из файла. Чтение переменных идет сверху вниз.

Переменные состоят из символов:

- малых латинских символов (a-z);
- заглавных латинских символов (A-Z);
- цифр (0-9);
- подчерка (_);
- тире (-).

Комментарий к переменной начинается с 35 символа. Если переменная со значением длиннее 35 символов, то комментарий переносится на следующую строку.

11.15.3 Локализация

Язык задается по стандарту ISO 639-1 (2-буквенные сокращения)

11.15.3.1 Язык по-умолчанию

Язык по-умолчанию задается в конфигурации приложения.

Поддерживаемые языки

- 1) ru – Русский
- 2) en – Английский
- 3) kz - Казахский

11.15.3.2 Язык локализации для пользователя

Если язык локализации не задан для пользователя, приложение будет отображаться пользователю в локализации по умолчанию.

nkataev

✓ User is not blocked

Parameters

Passwords

Security

Autostart

Groups

Login

nkataev

First Name

Nikita

Last Name

Kataev

Email

Value is empty

Date of Birth

Value is empty

Use experimental functions



Language

Русский

Рисунок 98 – Просмотр текущих подключений и параметров

Задать язык отображения можно в приложении в администрировании пользователя в поле "Язык\Language"

11.15.4 Переменные окружения

Приложение при старте проверяет переменные окружения.

Для того что бы переменная из конфигурационного файла была считана из переменных окружения, она должна начинаться на `VISMIND_` в переменных окружения.

Переменная окружения должна начинаться на **vismind**.

Важно! Проверяются **все переменные** из конфигурационного файла в переменных окружения.

Например: считать название приложения

Название приложения в конфигурационном файле находится в переменной `app_name`. Тогда к `app_name` добавим префикс `VISMIND_` – получится параметр `VISMIND_APP_NAME`. Этот параметр будет искаться в переменных окружения.

И в переменную `app_name` присвоится значение:

- или из переменной окружения **VISMIND_APP_NAME**, если она будет найдена
- или из `app_name`, если переменная окружения не найдена

Так же через переменные окружения можно задавать настройки Gunicorn.

Для того что бы переменная из конфигурационного файла была считана из переменных окружения, она должна начинаться на `vismind_gunicorn_` в переменных окружения.

Переменная окружения должна начинаться на **vismind_gunicorn** для настройки Gunicorn.

Например: `vismind_gunicorn_workers=2` будут заданы 2 воркера gunicorn

В связи с нестабильной работой gunicorn при динамическом расчете параметров, количество воркеров можно задать вручную, рассчитав по формуле:

(количество ядер * 2) + 1 (формула взята из документации [Gunicorn](#)).

11.15.5 Использование в приложении

Чтобы использовать в приложении настройки конфигурации, надо импортировать переменную `SETTINGS`.

Пример импорта

```
from core.config import SETTINGS
```

Можно обращаться к разделу и переменной

Пример получения конфига из раздела:

```
SETTINGS.AUTH.DEFAULT_GROUP
```

Можно обращаться к разделу, к подразделу и переменной

Пример получения конфига из подраздела

```
SETTINGS.AUTH.OIDC.ALLOW_LOGIN
```

11.16 Конфигурация клиентского приложения

Конфигурация клиентского приложения содержится в бэкенде приложения и выдаётся клиенту через системный эндпоинт.

11.16.1 Расположение

Файл конфигурации расположен по пути `./conf/frontend_config.json`.

11.16.2 Описание формата конфигурационного файла

Файл конфигурации описан в JSON-формате и имеет вложенную структуру. Ниже представлен фрагмент файла конфигурации.

```
{
  "ApplicationLanguage" : "ru",
  "ApplicationVersion" : "0.40.0",
  "ApplicationName" : "Цифровая платформа УПРАВЛЕНИЕ В ПРОСТРАНСТВЕ",
  "ApplicationShortName" : "ЦП УВП",
  "ApplicationDevelopers" : "\"Бюро Информационных Технологий\"",
  "Routing" : {
    "License" : "license",
    "PasswordChange": "password",
    "Graph" : "graph",
    "UserApplication" : "analytics",
    "UserApplicationPrint" : "print",
```

Файл конфигурации копируется из клиентского репозитория в момент сборки Docker образа серверной части (в случае использования Docker).

11.16.3 Переменные окружения

Приложение при старте проверяет переменные окружения.

Для изменения настроек используются переменные окружения. Приложение разово собирает переменные при первом обращении за настройками.

Переменные окружения для клиентского приложения назначаются с префиксом VISMIND_FRONTEND_.

Например, чтобы изменить параметр ApplicationLanguage, необходимо создать переменную VISMIND_FRONTEND_ApplicationLanguage=en.

Для вложенных настроек указывается полный путь через точку. Например, для следующей настройки:

```
"LdapLoginConfiguration" : {
  "IsDefault" : true,
```

необходимо создать переменную VISMIND_FRONTEND_LdapLoginConfiguration.IsDefault=false.

Обратите внимание, что префикс VISMIND_FRONTEND_ отличается от префикса, используемого в настройках серверного приложения - VISMIND_.

11.17 Управление глобальными константами

В приложении есть возможность создавать и использовать глобальные константы - переменные, задающиеся в настройках и используемые в любых блоках приложения.

11.17.1 Настройка глобальных констант

Для настройки глобальных констант необходимо перейти в меню настроек в правом верхнем углу и в выпадающем меню выбрать пункт "Константы".

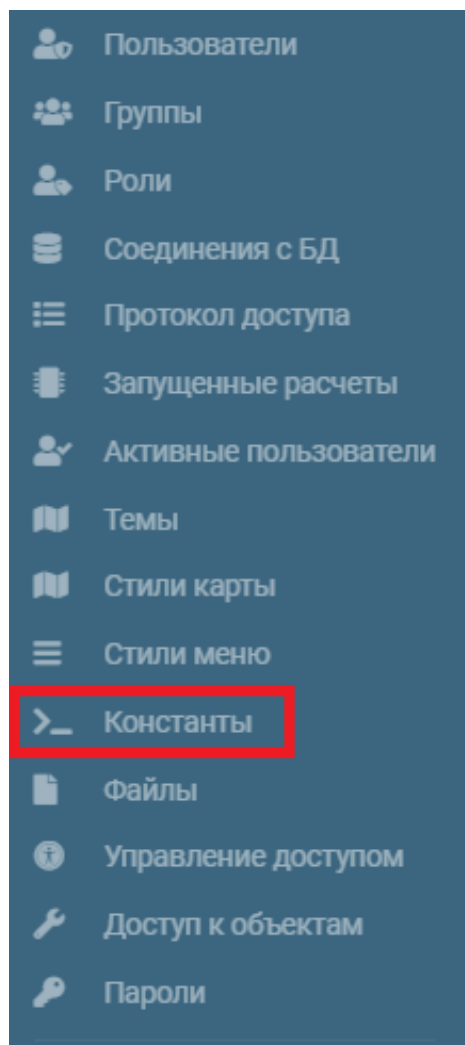


Рисунок 99 – Открытие раздела управления константами в меню администрирования

Важно! Для редактирования констант требуется отдельная роль.

После чего откроется раздел настройки констант:

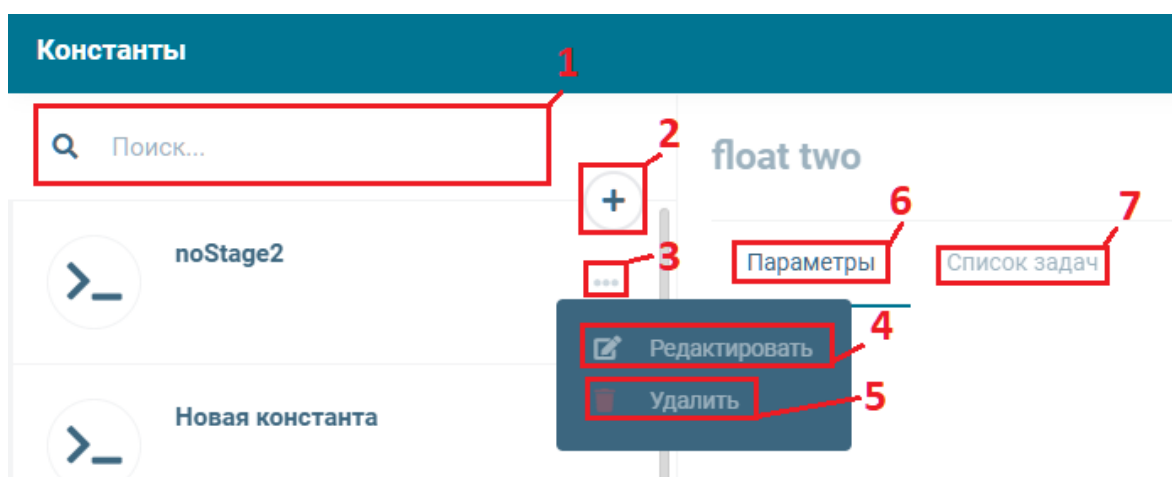


Рисунок 100 – Раздел управления константами

На рис. 94 представлен следующий функционал:

1. Поиск константы
2. Создание константы

3. Открыть контекстное меню
4. Редактирование константы
5. Удалить константу
6. Настройки выбранной константы
7. Список задач, в которых используется константа

При добавлении новой или редактировании существующей константы появляется окно добавления константы, рис. 95.

Новая константа

Имя

Новая константа

Ключ

Значение не указано

Описание

Значение не указано

Тип

Строковый

Значение

Значение не указано

Рисунок 101 – Доавление константы

- **Имя** - наименование константы.
- **Ключ** - ключ константы для ее использования.

- Ключ константы должен быть уникален в рамках приложения.
- **Описание** - описание константы.
- **Тип** - тип константы.
- **Значение** - значение константы. В зависимости от выбранного типа, поле ввода значения меняется.

11.17.2 Использование констант

В зависимости от блоков, формат написания констант может меняться.


Не рекомендуется совмещать несколько форматов написания констант в одном поле.

- Общий формат написания (доступен во всех блоках, допускающих ввод текста, в т.ч. "Визуализатор JavaScript"):
 - `<const: ключКонстанты>`
- JINJA формат:
 - `{{ const("ключКонстанты") }}`
- JSON формат:
 - `{"$const": "ключКонстанты"}`
- Формат блока Python:
 - `execution_context.const("ключКонстанты")`
- Формат блоков "Правила"/"Выражение":
 - `константа("ключКонстанты")`
 - `const("ключКонстанты")`

11.18 Управление глобальными паролями

В приложении есть возможность создавать и использовать глобальные пароли. Функционал аналогичен глобальным константам.

11.18.1 Настройка паролей

Для настройки глобальных паролей необходимо перейти в меню настроек в правом верхнем углу  и в выпадающем меню выбрать пункт "Пароли":

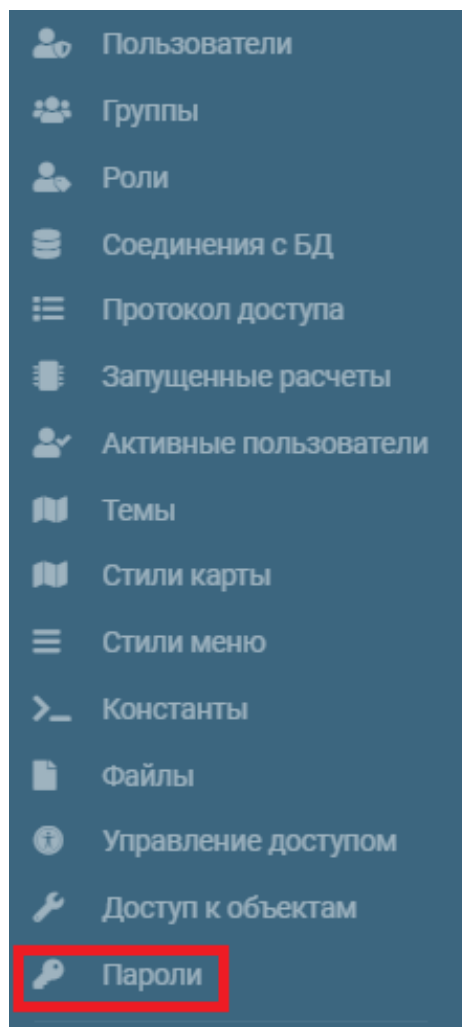


Рисунок 102 – Открытие раздела управления паролями в меню администрирования
Важно! Для редактирования паролей требуется отдельное разрешение.
 После чего откроется раздел настройки паролей:

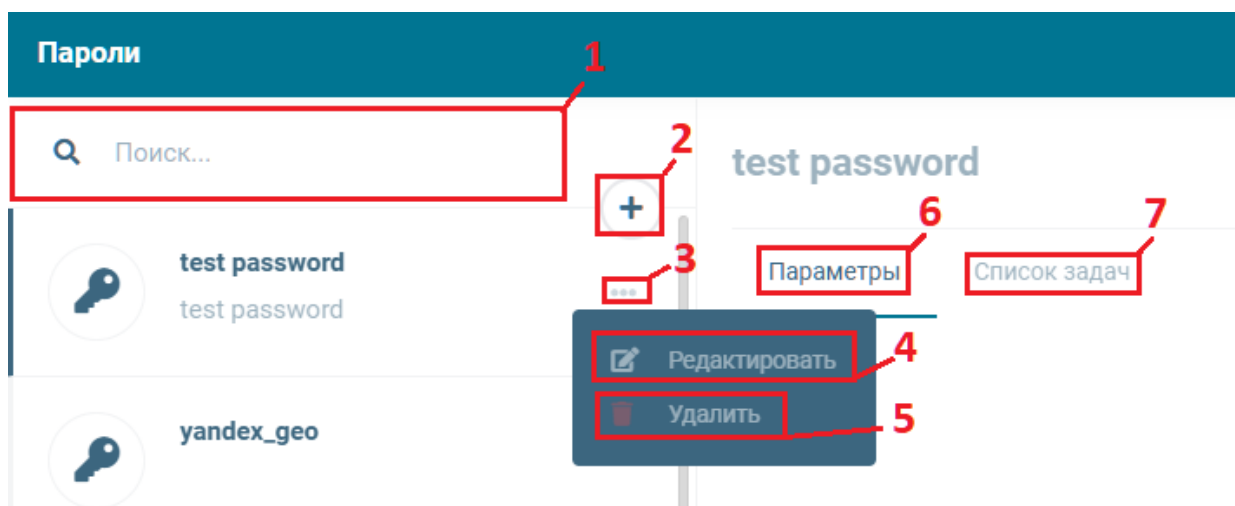


Рисунок 103 – Раздел управления паролями

На рис. 99 представлен следующий функционал:

- Поиск пароля

- Создание нового пароля
- Открыть контекстное меню
- Редактирование пароля
- Удаление пароля

При добавлении или редактировании пароля появляется следующее окно, рис. 100.

Новый пароль

Ключ

Значение не указано

Имя

Новый пароль

Пароль

Значение не указано

Описание

Рисунок 104 – Доавление пароля

- **Ключ** - ключ пароля для его использования в блоках.
 - Ключ должен быть уникален в рамках приложения.
- **Имя** - наименование пароля.
- **Пароль** - значение пароля.
- **Описание** - описание пароля.

11.18.2 Использование глобальных паролей

Глобальные пароли используются аналогично глобальным константам, но только в полях предназначенных для ввода паролей. Формат написания:

- <pass: ключПароля>

Для блока "3D-карта" доступен вариант ввода пароля из выпадающего списка (Геокодирование → Ввод пароля).

В блоке Python доступен вызов в формате:

- `execution_context.password("ключПароля")`

Результатом выполнения функции будет являться дешифрованный пароль.

Формат для подстановки в поля JSON (пароль будет в строковом варианте):

- `{"$pass": "ключПароля"}`

Результатом выполнения функции будет являться дешифрованный пароль.

Пример использования с JSON-настройками:

- Завели глобальный пароль с ключом "foo" и значением "bar" в админке
- В JSON-настройках прописали следующие поля:

```
{  
  "secret-password": {"$pass": "foo"}  
}
```

При обработке JSON-настроек поле `{"$pass": "foo"}` будет заменено на строку "bar". Фигурные скобки вокруг `"$pass": "foo"` обязательны, без них глобальный пароль не распознается.

11.19 Управление файлами

Служит для просмотра и удаления всех загруженных файлов в приложении.

11.19.1 Общее описание

Функционал управления файлами осуществляется через страницу "Файлы" в локальном меню пользователя

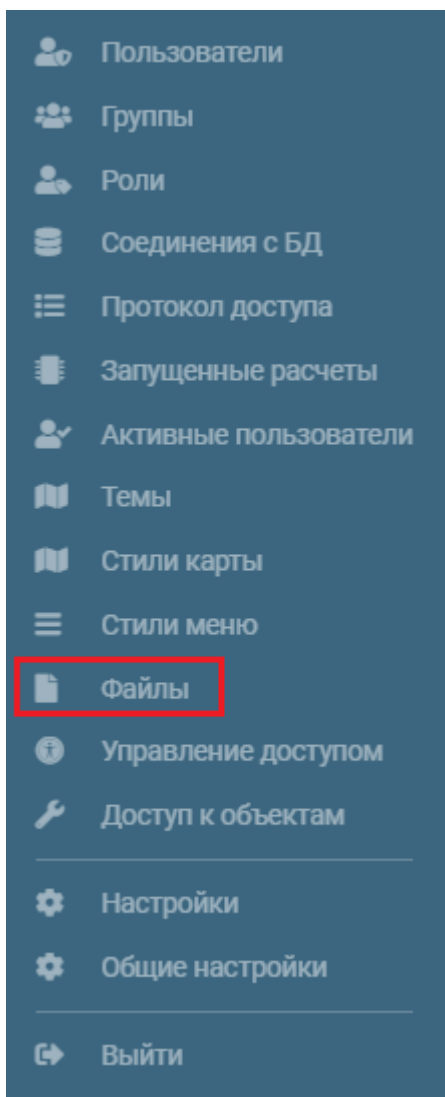


Рисунок 105 – Открытие раздела управления файлами в меню администрирования
Доступность страницы зависит от разрешений предоставленных в ролях пользователя.

11.19.2 Страница "Файлы"

На странице файлов показан список всех задач, содержащих какие-либо файлы, а так же функционал по взаимодействию с ними.

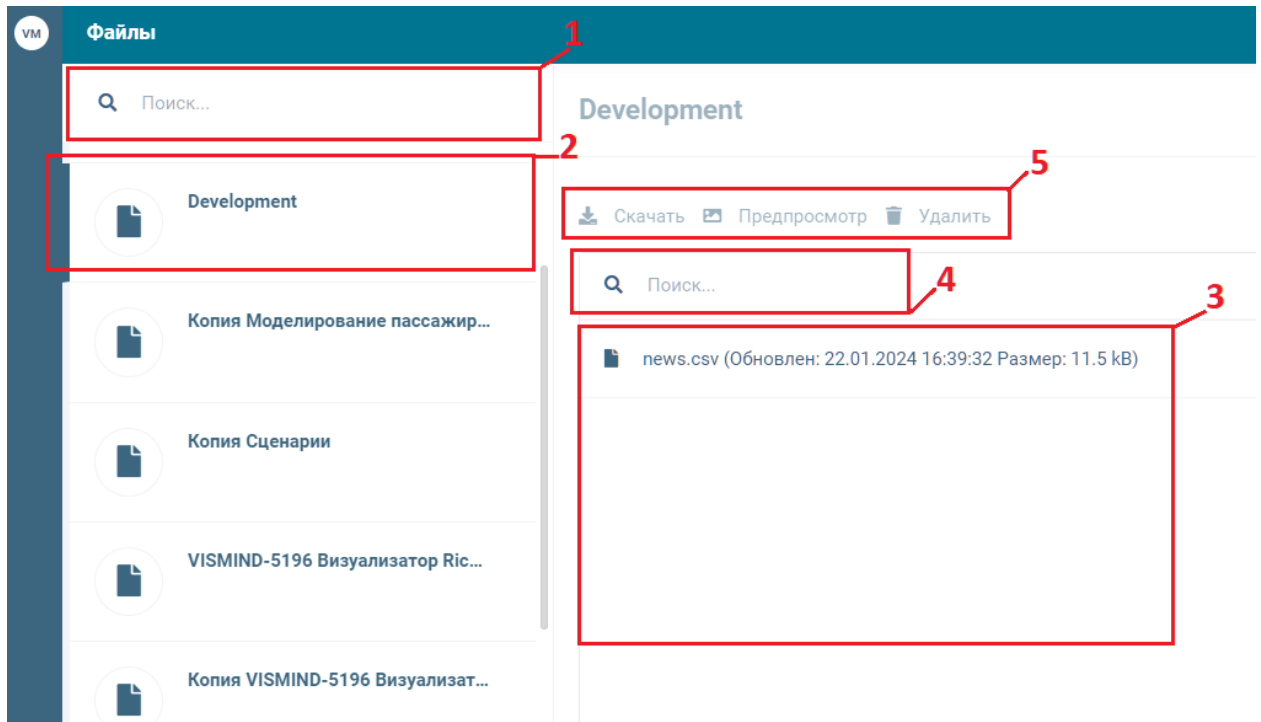


Рисунок 106 – Страница «Файлы»

На рис. выше представлен следующий функционал:

1. Поиск задач по названию
2. Выбранная задача
3. Список файлов, содержащихся в задаче
 - Если внутри задачи есть папки, то будет выведена древовидная структура:

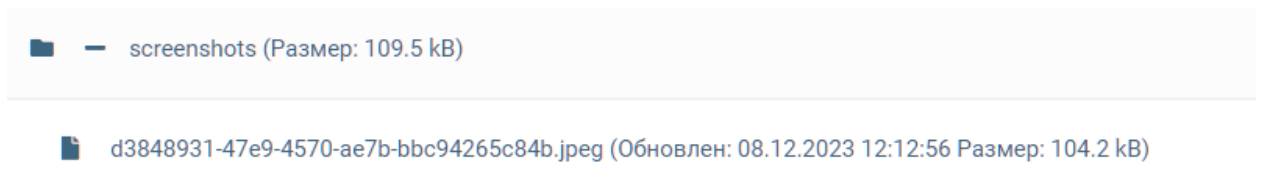


Рисунок 107 – Пример древовидной структуры

- Все снимки пресетов, созданные в задаче, будут сохранены в папку screenshots (если ее нет, она будет создана).
4. Поиск файлов по наименованию
 5. Кнопки для управления выбранным файлом:
 - **Скачать** - скачивает файл
 - **Предпросмотр** - предварительный просмотр файла
 - Предпросмотр работает только для изображений
 - **Удалить** - удаление файла

11.19.3 Работа с файлами

Импорт/экспорт задачи не сохраняют импортируемые/экспортируемые задачи на файловый сервер.

Снимки пресетов сохраняются в папку **screenshots**.

При экспорте пресетов, результат будет сохранен во временную папку **temp**. Результат экспорта будет удален через 24 часа.

11.19.3.1 Блоки, взаимодействующие с файловой системой

Область видимости всех блоков ограничена той задачей, в которой они находятся. Т.е. нельзя, к примеру, загрузить файл, созданный в другой задаче.

Блок "Загрузить файл" позволяет загрузить файл на сервер. Файл будет сохранен в корне задачи.

Блок "Скачать файл" позволяет скачивать файл с сервера.

Блок "Импорт файла" сохранит файл импорта на сервер в корень задачи.

Блок "Python" (метод `save_bytes`) позволяет сохранить файл на сервер. По умолчанию сохранение происходит в папку **doc**. Метод `get_bytes` позволяет получить файл в бинарном виде.

Блок "Таблица" (при экспорте таблицы) сохранит таблицу во временную папку **tables/temp**. Результат экспорта будет удален через 24 часа.

11.19.4 Работа с файловым сервером

Общие настройки файловой системы в файле конфигурации:

- **path_storage** - Конкретная папка для хранения файлов
- **file_lifetime** - Время жизни файлов (в секундах)
- **temp_path_storage** - Путь для сохранения временных файлов. Чаще всего используется при импорте файлов. После загрузки файлы удаляются из папки
 - Путь должен начинаться с папки, указанной в **path_storage**
- **file_storage** - Путь до файлового хранилища
 - Путь должен начинаться с папки, указанной в **path_storage**
- **static_path** - Путь к статическому содержимому. Файлы из этой папки доступны фронт-у
 - Путь должен начинаться с папки, указанной в **path_storage**
- **export_path** - Папка внутри статики (**static_path**), в которой будут храниться экспортируемые файлы

Существует два режима работы с файловым сервером:

- Работа в файловом хранилище;
- Работа с сервером MinIO.

Вне зависимости от выбранного режима работы, структура папок будет идентичная:

- Задачи (в MinIO каждая задача создает отдельный контейнер (bucket))
 - Папка со скриншотами
 - Папка с таблицами
 - Папка с документами (Блок "Загрузить файл")

Название папок можно задать в конфигурационном файле:

- **global_file_folder** - Наименование контейнера/папки для хранения всех файлов, не привязанных к конкретной задаче
 - Папка и все ее содержимое не отображается в менеджере файлов.
- **screenshot_folder** - Папка для хранения скриншотов
- **temporary_folder** - Папка для хранения временных файлов (все данные в этой папке будут периодически удаляться, при достижении времени жизни, указанного в **file_lifetime**)
- **table_folder** - Папка для хранения таблиц
- **doc_folder** - Папка для хранения документов

11.19.5 Файловое хранилище

Приложение по умолчанию работает в режиме хранилища.

В этом случае все файлы будут храниться по пути, указанному в **file_storage**.

11.19.6 Сервер MinIO

Файловый сервер будет развернут на основе MinIO.

Для включения необходимо включить флаг **minio_enabled** в файле конфигурации.

Для настройки файлового сервера доступны следующие настройки:

- **minio_enabled** - Флаг включения MinIO
- **minio_host** - Адрес сервера MinIO
- **minio_port** - Порт сервера MinIO
- **minio_access_key** - Access Key к серверу MinIO
- **minio_secret_key** - Secret Key к серверу MinIO
- **minio_secure_tls** - Флаг включения шифрования TLS
- **minio_url_lifetime** - Время жизни временных URL (секунд)
- **minio_file_lifetime** - Время жизни временных файлов (дней)

11.19.6.1 Особенности взаимодействия с MinIO

Рекомендуется создавать отдельного пользователя для работы с файловой системой.

Временные файлы помечены тегом **temporary: 1**

11.19.6.2 Перенос файлов при смене типа файлового сервера

При смене типа файлового сервера (включение/выключение MinIO) для переноса файлов из выключенного хранилища в текущее можно выполнить API

POST|GET /system/files/transfer

параметр:


- **debug: bool** - проверка корректности переноса файлов. По умолчанию флаг включен.

При переносе файлы с одинаковыми именами будут перезаписаны.

11.20 Управление стилями карты

В приложении доступна возможность создавать глобальные стили карты и, в дальнейшем, использовать их в любом блоке "3D-карта".

11.20.1 Настройка стилей карты

Для настройки глобальных констант необходимо перейти в меню настроек в правом верхнем углу  и в выпадающем меню выбрать пункт "Стили карты":

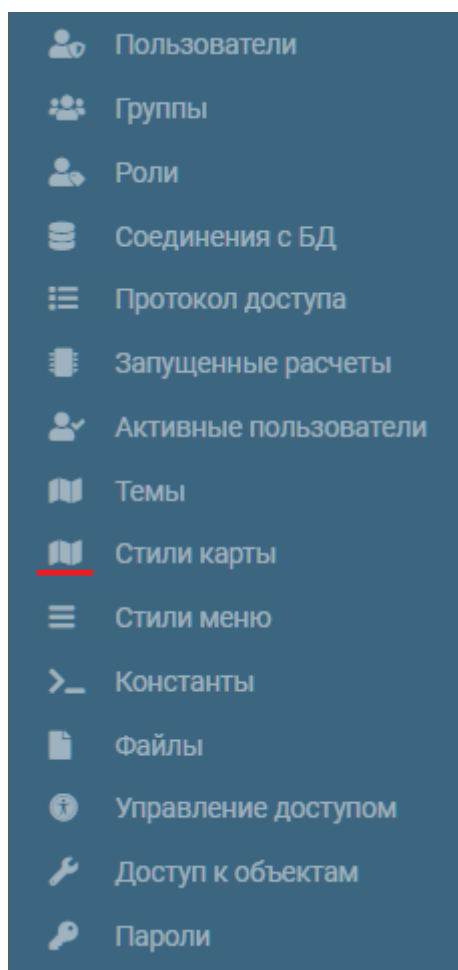


Рисунок 108 – Открытие раздела управления стилями карты в меню администрирования

Важно! Для редактирования стилей карты требуется отдельное разрешение.

После чего откроется раздел настройки стилей карты:

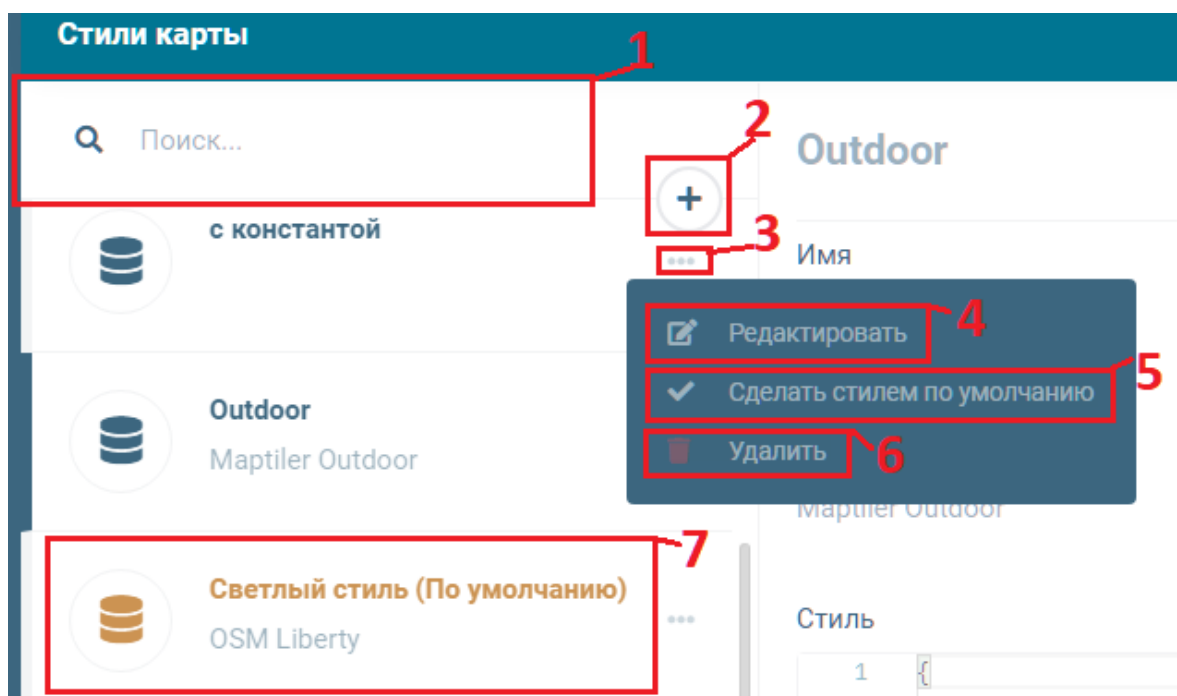


Рисунок 109 – Раздел управления стилями карты

На рис. 105 представлен следующий функционал:

1. Поиск стиля
2. Создание нового стиля
3. Открыть контекстное меню
4. Редактирование стиля
5. Отметка стиля в качестве стиля по умолчанию
6. Удаление стиля
7. Стиль, отмеченный по умолчанию

При добавлении или редактировании пароля появляется следующее окно, рис. 106.

Новый стиль

Имя

Новый стиль

Описание

Значение не указано

Стиль

1

Рисунок 110 – Доавление пароля

- **Имя** - наименование стиля.
- **Описание** - описание стиля.
- **Стиль** - настройки стиля карты в формате JSON.

Стиль, отмеченный по умолчанию

Любой стиль можно сделать стилем по умолчанию. Единоmomentно может быть только один стиль по умолчанию (отметка другого стиля при этом слетает).

Стиль по умолчанию выделяется цветом, а также припиской «По умолчанию» в наименовании стиля.

11.20.2 Использование стилей карты

Глобальный стиль карты можно выбрать в блоке "3D-карта", если в параметре "Настройка стиля карты" выбран элемент "Глобальные стили".

В случае, если флаг "Использовать стиль по умолчанию" включен, будет использован стиль карты по умолчанию. При отсутствующем стиле карты по умолчанию будет использоваться пустой стиль карты.

Приложение 1. Внешние зависимости и библиотеки

Платформа разрабатывается с использованием программного обеспечения с открытым исходным кодом и с бесплатными лицензиями (FOSS – Free and Open Source Software).

С текстом лицензий можно ознакомиться по ссылкам в столбце «Лицензия» напротив каждой из компонент.

Полный актуальный перечень используемых библиотек доступен в документе «Технологии и библиотеки_ЦП_УвП_версия_X» соответствующей версии.

Приложение 2. Структура базы данных платформы

Скрипты создания репозитория

№	Скрипт	Назначение
1.	create_repository.sql	Создание пустой БД репозитория
2.	extensions.sql	Установка необходимых расширений PostgreSQL
3.	tables.sql	Создание таблиц и индексов
4.	create_search_dict_multi.sql	Создание справочников для мультязычного поиска по тексту
5.	tr_project_check_parent.sql	Триггер для проверки корректности иерархии при создании или обновлении задач
6.	tr_project_updated.sql	Триггер на актуализацию времени изменения задачи при изменении блоков, связей и пресетов
7.	tr_block_update_vector.sql	Триггер для пересчета индекса полнотекстового поиска по блокам
8.	tr_preset_update_vector.sql	Триггер для пересчета индекса полнотекстового поиска по пресетам
9.	tr_project_update_vector.sql	Триггер для пересчета индекса полнотекстового поиска по задачам
10.	tr_vis_deleted.sql	Триггер для пересчета индекса полнотекстового поиска при удалении визуализатора
11.	vm_delete_old_calculations.sql	Функция для удаления старых расчетов во время сборки мусора
12.	vm_get_presets.sql	Функция получения списка пресетов для «Меню»
13.	vm_get_related_tasks.sql	Функция получения списка связанных задач
14.	vm_get_block_outputs.sql	Функция получения списка выходов блока с учетом иерархии
15.	vm_get_block_inputs.sql	Функция получения списка входов блока с учетом иерархии
16.	vm_project_copy.sql	Функция копирования задачи
17.	vm_get_block_settings.sql	Функция получения списка настроек блока с учетом иерархии
18.	vm_get_block_setting_value.sql	Функция получения значения настройки блока с учетом иерархии
19.	vm_get_blocks_using_table.sql	Функция получения списка блоков (Справочник или Показатели), использующих таблицу БД
20.	library_hierarchy.sql	Создание иерархии и «заглушек» блоков библиотеки
21.	library_common_modules.sql	Создание общих настроек блоков библиотеки
22.	library_transforms.sql	Создание схем преобразования блоков (статистические блоки и SQL)
23.	data_t_app_styles.sql	Создание настроек стандартных тем приложения
24.	data_t_config_license.sql	Создание записи с лицензией в конфигурации приложения
25.	data_t_config_version.sql	Создание записи с номером версии в конфигурации приложения

№	Скрипт	Назначение
26.	data_t_library_docs.sql	Создание ссылок на документацию блоков
27.	data_t_vis_styles_update.sql	Создание дефолтных стилей визуализаторов
28.	data_access_policy.sql	Создание настроек политик доступа по умолчанию
29.	data_completions.sql	Создание автоподстановок в коде Python
30.	data_user_roles.sql	Создание первоначальных пользователей, групп и ролей
31.	data_delete_setup_token.sql	Удаление временного токена, использующегося для актуализации блоков библиотеки

Структура таблиц репозитория

d_lib_item_types

Типы записей в библиотеке блоков

Поле	Тип	Описание
id	integer	Идентификатор
name	character varying	Название

d_proj_item_types

Типы записей задач

Поле	Тип	Описание
id	integer	Идентификатор
name	character varying	Название

t_access_actions

Логируемые типы действий пользователей

Поле	Тип	Описание
id	integer	Идентификатор
name	character varying	Название

t_access_log

Журнал доступа к API для новой структуры

Поле	Тип	Описание
action_id	smallint	Категория действия
caller_id	uuid	Пользователь, вызвавший API, если известен
caller_name	character varying	Пользователь, вызвавший API, если известен
dt	timestamp with time zone	Время действия
http_host	character varying	Заголовки запроса
http_referer	character varying	Заголовки запроса
http_user_agent	character varying	Заголовки запроса
http_x_forwarded_for	character varying	Заголовки запроса
http_x_real_ip	character varying	Заголовки запроса
id	uuid	Идентификатор
path_info	character varying	Заголовки запроса
remote_addr	character varying	Заголовки запроса
remote_port	character varying	Заголовки запроса
request_method_id	smallint	Тип запроса:
request_uri	character varying	Заголовки запроса
server_port	character varying	Заголовки запроса
server_protocol	character varying	Заголовки запроса

t_added_rows

Добавленные ряды таблиц

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
dat	json	Данные ряда
id	uuid	Идентификатор
user_id	uuid	Идентификатор пользователя

t_app_styles

Темы приложения

Поле	Тип	Описание
dat	json	Настройки
id	uuid	Идентификатор
ord	integer	Порядок

t_block_filters

Фильтры блоков (отметки, заданные пользователем)

Поле	Тип	Описание
block_id	uuid	Идентификатор
filter	json	Фильтр (отметка)
session_id	character varying	Идентификатор сессии (не используется в текущей реализации)
user_id	uuid	Идентификатор пользователя

t_block_item_bindings

Привязки данных блоков к выходам блоков (динамические выходы)

Поле	Тип	Описание
addr	character varying	Адрес с учетом иерархии
binding	json	Данные привязки
block_id	uuid	Идентификатор блока

t_block_secrets

Сохраненные зашифрованные пароли от соединений с источниками данных

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
setting_id	character varying	Идентификатор настройки блока с учетом иерархии
val	bytea	Зашифрованное значение

t_block_series

Справочники OLAP

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
dict	character varying	Идентификатор справочника
id	character varying	Ключ
ind	integer	Индекс
name	character varying	Название (значение) элемента
pid	character varying	Родительский ключ (для иерархического справочника)

t_block_states

Сохраненные состояния визуализаторов

Поле	Тип	Описание
block_id	uuid	Идентификатор блока-визуализатора
created	timestamp with time zone	Дата и время создания
dat	json	Данные
id	uuid	Идентификатор
local	boolean	Локальный для задачи или доступен из любой задачи
preset_id	uuid	Идентификатор пресета
private	boolean	Флаг доступности только автору (не используется в текущей реализации)
proj_id	uuid	Идентификатор задачи
saves	json	Сохраненные данные визуализатора (для статичного состояния)
static	boolean	Статичный или динамический
updated	timestamp with time zone	Дата и время последнего изменения
user_id	uuid	Идентификатор пользователя
user_updated_id	uuid	Идентификатор пользователя, изменявшего блок последним

t_block_states_styles

Стили статических блоков

Поле	Тип	Описание
dat	jsonb	Описание стиля
preset_id	uuid	Идентификатор пресета
proj_id	uuid	Идентификатор задачи
state_id	uuid	Идентификатор статического блока

t_block_styles

Стили блоков

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
dat	jsonb	Описание стиля
preset_id	uuid	Идентификатор пресета
proj_id	uuid	Идентификатор задачи

t_blocks

Блоки

Поле	Тип	Описание
author_id	uuid	Идентификатор пользователя, создавшего блок
created	timestamp with time zone	Дата и время создания
dat	jsonb	Данные блока: описание настроек, входов и выходов блока, значения настроек блока
id	uuid	Идентификатор

pid	uuid	Родительский элемент (используется для блоков Фрейм)
proj_id	uuid	Идентификатор задачи
updated	timestamp with time zone	Дата и время последнего существенного (по крайней мере настройки) изменения
updated_easy	timestamp with time zone	Дата и время последнего несущественного (перемещение, раскраска) изменения
user_updated_easy_id	uuid	Идентификатор пользователя, несущественно (перемещение, раскраска) изменявшего блок последним
user_updated_id	uuid	Идентификатор пользователя, существенно (по крайней мере настройки) изменявшего блок последним

t_blocks_aux

Общие модули у блоков

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
dat	jsonb	Данные модуля: описание настроек
execution	integer	Порядок выполнения модуля: 0 – нет выполняемого модуля 1 – до выполнения блока 2 – после выполнения блока 3 – до и после выполнения блока
id	uuid	Идентификатор
ord	integer	Порядок

t_blocks_cache

Кэшированные данные для слоев 3D-карты

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
dat	jsonb	Данные
params	character varying	Дополнительные параметры
user_id	uuid	Идентификатор пользователя

t_blocks_calc_status

Статусы расчета блоков

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
status	integer	Статус

t_blocks_calc_status_users

Статусы расчета блоков по пользователям

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
status	integer	Статус

user_id	uuid	Идентификатор пользователя
---------	------	----------------------------

t_blocks_calc_visual_list

Визуализаторы, обновляемые после событий

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
dat	jsonb	Данные
event_name	character varying	Идентификатор события

t_blocks_const_settings_name_tsv

Хранение дефолтных настроек (не меняются от блока к блоку)

Поле	Тип	Описание
name	character varying	Наименование
set_id	character varying	Идентификатор настройки
tsv	tsvector	Описание настройки
type	character varying	Тип настройки

t_blocks_projection

Настройки блоков по проекциям графа

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
block_state	jsonb	Статусы блока на проекции
block_style	jsonb	Стили блока на проекции
preset_id	uuid	Идентификатор пресета (проекции)
updated	timestamp with time zone	Дата и время изменения
user_updated_id	uuid	Идентификатор пользователя который изменил проекцию

t_blocks_settings_name_tsv

Хранение списка настроек

Поле	Тип	Описание
block_lib_id	uuid	Идентификатор блока библиотеки
name	character varying	Наименование
set_id	character varying	
tsv	tsvector	
type	character varying	Тип настройки

t_blocks_settings_tsv

Хранение значений настроек блока

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
set_id	character varying	
tsv	tsvector	

t_blocks_tsv

Блоки: полнотекстовый поиск

Поле	Тип	Описание
block_id	uuid	Идентификатор блока

tsv	tsvector	Информация для полнотекстового поиска
t_blocks_user_filters		
Пользовательские фильтры для блоков-визуализаторов		
Поле	Тип	Описание
block_id	uuid	Идентификатор блока
user_id	uuid	Идентификатор пользователя
filter_id	uuid	Идентификатор фильтра
name	character varying	Наименование фильтра
dat	json	Данные по отметке фильтра

t_calc_data

Данные с расчетами задач

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
log	json	Лог расчета
logb	bytea	Лог расчета
outb	bytea	Данные на выходе блока
outputs	json	Данные на выходе блока
saves	json	Данные визуализатора (для блоков, являющихся визуализаторами)
vis	bytea	Данные визуализатора (для блоков, являющихся визуализаторами)

t_calc_data_event

Данные с расчетами событий

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
event_id	character varying	Идентификатор пользователя, запустившего расчет события
log	json	Лог расчета
logb	bytea	Лог расчета
outb	bytea	Данные на выходе блока
outputs	json	Данные на выходе блока
saves	json	Данные визуализатора (для блоков, являющихся визуализаторами)
user_id	uuid	Идентификатор пользователя, запустившего расчет события
vis	bytea	Данные визуализатора (для блоков, являющихся визуализаторами)

t_calc_data_event_iterator

Данные с расчетами итераторов событий

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
event_id	character varying	Идентификатор пользователя, запустившего расчет события
iteration	character varying	Наименование итерации
iterator_id	uuid	Идентификатор итерации

log	json	Лог расчета
logb	bytea	Лог расчета
outb	bytea	Данные на выходе блока
outputs	json	Данные на выходе блока
saves	json	Данные визуализатора (для блоков, являющихся визуализаторами)
user_id	uuid	Идентификатор пользователя, запустившего расчет события
vis	bytea	Данные визуализатора (для блоков, являющихся визуализаторами)

t_calc_data_iterator

Данные с расчетами итераторов

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
iteration	character varying	Наименование итерации
iterator_id	uuid	Идентификатор итерации
log	json	Лог расчета
logb	bytea	Лог расчета
outb	bytea	Данные на выходе блока
outputs	json	Данные на выходе блока
saves	json	Данные визуализатора (для блоков, являющихся визуализаторами)
vis	bytea	Данные визуализатора (для блоков, являющихся визуализаторами)

t_calc_data_preview_iterator

Данные с расчетами итераторов для предпросмотра

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
iteration	character varying	Наименование итерации
iterator_id	uuid	Идентификатор итерации
log	json	Лог расчета
logb	bytea	Лог расчета
outb	bytea	Данные на выходе блока
outputs	json	Данные на выходе блока
saves	json	Данные визуализатора (для блоков, являющихся визуализаторами)
user_id	uuid	Идентификатор пользователя, запустившего расчет события
vis	bytea	Данные визуализатора (для блоков, являющихся визуализаторами)

t_calc_token

Токены для доступа к API (выполнение расчетов по расписанию)

Поле	Тип	Описание
created	timestamp without time zone	Дата и время создания
id	uuid	Идентификатор

tk	character varying	Идентификатор (токен)
user_id	uuid	Идентификатор пользователя

t_calculations

Расчеты задач (устарела, в настоящее время не используется)

Поле	Тип	Описание
actual	boolean	Флаг актуальности
date_end	timestamp without time zone	Дата и время завершения расчета
date_start	timestamp without time zone	Дата и время начала расчета
id	uuid	Идентификатор
proj_id	uuid	Идентификатор задачи
successful	boolean	Флаг успешности

t_calculations_log

Логи расчетов (устарела, в настоящее время не используется)

Поле	Тип	Описание
dt	timestamp without time zone	Дата и время расчета
log	text	Лог расчета
proj_id	uuid	Идентификатор задачи

t_common_modules

Общие настройки блоков библиотеки (общие модули)

Поле	Тип	Описание
dat	json	Данные модуля: описание настроек
execution	integer	Порядок выполнения модуля: 0 – нет выполняемого модуля 1 – до выполнения блока 2 – после выполнения блока 3 – до и после выполнения блока
id	uuid	Идентификатор
ord	integer	Порядок
rules	json	Правила поведения настроек в интерфейсе для добавления в общие правила блоков
sett_position	integer	Вставка настроек из общего модуля: 0 – до настроек блока 1 – после настроек блока

t_common_vis_styles

Общие настройки для стилей визуализаторов

Поле	Тип	Описание
dat	json	Настройки
id	uuid	Идентификатор
ord	integer	Порядок
sett_position	integer	Номер позиции настройки

t_config

Настройки приложения

Поле	Тип	Описание
name	character varying	Наименование
val_bytes	bytea	Значение
val_json	jsonb	Значение
val_string	character varying	Значение

t_files

Загруженные файлы

Поле	Тип	Описание
author_id	uuid	Идентификатор пользователя, загрузившего файл
created	timestamp with time zone	Дата и время создания
descr	text	Краткое описание
id	uuid	Идентификатор
info	jsonb	Информация о статусе загрузки
name	character varying	Название
proj_id	uuid	Идентификатор задачи
upd_user_id	uuid	Идентификатор пользователя, перезагрузившего файл
updated	timestamp with time zone	Дата и время последнего изменения

t_frames

Пресеты во фреймах

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
dat	json	Данные пресета
views	json	Список визуализаторов

t_global_const

Глобальные константы

Поле	Тип	Описание
id	uuid	Идентификатор константы
name	varchar	Наименование константы
desc	text	Описание константы
type	varchar	Тип константы
key	varchar	Уникальный ключ константы (для получения константы в приложении)
value	text	Значение константы
system	bool	Является ли константа системной
locale	varchar	Локаль, для которой используется константа

t_global_const_links

Связь констант с блоками, в которых она используется

Поле	Тип	Описание
const_id	uuid	Идентификатор константы
block_id	uuid	Идентификатор блока

t_graphs

Подготовленный для расчета граф, состоящий из блоков и связей задачи

Поле	Тип	Описание
actual	boolean	Флаг актуальности
dat	json	Данные
project_id	uuid	Идентификатор задачи
updated	timestamp with time zone	Дата и время последнего изменения

t_groups

Группы пользователей

Поле	Тип	Описание
descr	text	Краткое описание
id	uuid	Идентификатор
name	character varying	Название
sessions_number	integer	Максимальное количество сессий

t_groups_roles

Назначенные группам роли

Поле	Тип	Описание
group_id	uuid	Идентификатор группы
role_id	uuid	Идентификатор роли

t_json_dict_items

Данные JSON-Справочник

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
dat	jsonb	Данные
item_id	integer	Ключ элемента

t_json_indicator_items

Данные JSON-Показатели

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
dat	jsonb	Данные
dt	timestamp without time zone	Временная отметка
dt_level	integer	Уровень календаря
row_id	integer	Ключ ряда

t_lazy_data

«Ленивые» данные расчета для использования в визуализаторах с большим объемом прикладных данных либо при передаче между модулями большого объема данных

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
created	timestamp with time zone	Дата и время создания
dat	jsonb	Данные
id	uuid	Идентификатор

t_library

Библиотека блоков

Поле	Тип	Описание
dat	json	Данные блока: описание настроек, входов и выходов блока, правил поведения настроек в интерфейсе
descr	text	Текстовое описание блока
hidden	boolean	Скрытый элемент
id	uuid	Идентификатор
name	character varying	Название
ord	integer	Порядок внутри папки
pid	uuid	Родительский элемент
type_id	integer	Тип элемента (папка или блок)

t_library_completions

Автоподстановки в коде

dat	json	Данные для автоподстановок
data_type	character varying	Код языка программирования
vl_type	character varying	Тип данных

t_library_custom

Библиотека, избранное – пользовательские блоки

Поле	Тип	Описание
dat	json	Данные блока: описание настроек, входов и выходов блока, правил поведения настроек в интерфейсе
descr	text	Текстовое описание блока
hidden	boolean	Скрытый элемент
id	uuid	Идентификатор
name	character varying	Название
ord	integer	Порядок внутри папки
pid	uuid	Родительский элемент
type_id	integer	Тип элемента (папка или блок)

t_library_custom_links

Библиотека, избранное – связи для фреймов

Поле	Тип	Описание
custom_id	uuid	Идентификатор блока из избранного
dat	jsonb	Данные связи
id	uuid	Идентификатор

t_library_docs

Библиотека, ссылки на документацию

Поле	Тип	Описание
env	character varying	Окружение
href	character varying	Ссылка на документацию
lib_id	uuid	Идентификатор блока библиотеки

t_library_ref

Привязки общих настроек блоков к блокам библиотеки

Поле	Тип	Описание
lib_id	uuid	Идентификатор блока библиотеки
mod_id	uuid	Идентификатор
ord	integer	Порядок

t_library_transforms

Библиотека, преобразования блоков

Поле	Тип	Описание
id	uuid	Идентификатор
name	character varying	Название

t_library_transforms_blocks

Список блоков, которые можно трансформировать в другой блок

Поле	Тип	Описание
lib_id	uuid	Идентификатор блока библиотеки
name	character varying	Название
transform_id	uuid	Идентификатор преобразования

t_links

Связи между блоками

Поле	Тип	Описание
dat	jsonb	Данные о связи
id	uuid	Идентификатор связи
proj_id	uuid	Идентификатор задачи

t_map_3d_popup_data

Данные по попапам

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
dat	jsonb	Данные
feature_id	uuid	Идентификатор объекта
user_id	uuid	Идентификатор пользователя

t_map_3d_popup_data_default

Дефолтные данные попапа

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
dat	jsonb	Данные
feature_id	uuid	Идентификатор объекта

t_olap_slices

Фильтры блоков OLAP

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
filter	json	Фильтр (отметка)

t_presets

Пресеты приложения

Поле	Тип	Описание
author_id	uuid	Идентификатор пользователя, создавшего пресет
checked	boolean	Отметка (используется при формировании Меню)
created	timestamp with time zone	Дата и время создания
dat	json	Данные пресета
descr	text	Краткое описание
home	boolean	Флаг «Домашняя страница»
id	uuid	Идентификатор пресета
img	bytea	Превью
opened	timestamp with time zone	Дата и время последнего открытия
ord	integer	Порядок (используется при формировании Меню)
ord2	integer	Порядок (используется в мастере)
proj_id	uuid	Идентификатор задачи
updated	timestamp with time zone	Дата и время последнего изменения
user_opened_id	uuid	Идентификатор пользователя, открывавшего пресет последним
user_updated_id	uuid	Идентификатор пользователя, изменявшего пресет последним
views	json	Список визуализаторов пресета

t_presets_short_link

Короткие ссылки на пресеты

Поле	Тип	Описание
id	character varying	Короткая ссылка
preset_id	uuid	Идентификатор пресета
public	boolean	Флаг публичности ссылки

t_presets_tsv

Пресеты: полнотекстовый поиск

Поле	Тип	Описание
preset_id	uuid	Идентификатор пресета
tsv	tsvector	Информация для полнотекстового поиска

t_projects

Задачи

Поле	Тип	Описание
author_id	uuid	Идентификатор пользователя, создавшего задачу
created	timestamp with time zone	Дата и время создания
dac	boolean	Разрешение общего доступа к задаче
descr	text	Краткое описание
fav	boolean	Избранная задача (используется при формировании Меню)
hidden	boolean	Скрытый элемент
id	uuid	Идентификатор

lock	boolean	Защита от удаления
name	character varying	Название
opened	timestamp with time zone	Дата и время последнего открытия
ord	integer	Порядок (используется при формировании Меню)
pid	uuid	Родительский элемент
settings	jsonb	Настройки задачи
state	jsonb	Данные о состоянии задачи
style	jsonb	Настройки отображения задачи
type_id	integer	Тип элемента
updated	timestamp with time zone	Дата и время последнего изменения
user_opened_id	uuid	Идентификатор пользователя, открывавшего задачу последним
user_updated_id	uuid	Идентификатор пользователя, изменявшего задачу последним

t_projects_dac

Детальные права доступа к задачам

Поле	Тип	Описание
group_id	uuid	Идентификатор группы
preset_id	uuid	Идентификатор роли
project_id	uuid	Идентификатор задачи
user_id	uuid	Идентификатор пользователя

t_projects_screenshots

Снимки экрана

Поле	Тип	Описание
created	timestamp with time zone	Дата и время создания
id	uuid	Идентификатор
img	bytea	Изображение
proj_id	uuid	Идентификатор задачи
thumbnail	bytea	Изображение для предпросмотра

t_projects_tags

Теги, назначенные задачам

Поле	Тип	Описание
ord	bigint	Порядок
proj_id	uuid	Идентификатор задачи
tag_id	uuid	Идентификатор тега

t_projects_tsv

Задачи: полнотекстовый поиск

Поле	Тип	Описание
proj_id	uuid	Идентификатор задачи
tsv	tsvector	Информация для полнотекстового поиска

t_roles

Роли

Поле	Тип	Описание
descr	text	Краткое описание
id	uuid	Идентификатор
name	character varying	Название
permissions	jsonb	Назначенные для роли права
permissions_bitwise	bit varying	Назначенные для роли права в виде битовой маски
sessions_number	integer	Максимальное количество сессий

(Устарело) t_scenarios

Сценарии

Поле	Тип	Описание
author_id	uuid	Идентификатор пользователя, создавшего сценарий
color	character varying	Цвет
created	timestamp with time zone	Дата и время создания
descr	text	Краткое описание
id	uuid	Идентификатор
name	character varying	Название
ord	integer	Порядок
proj_id	uuid	Идентификатор задачи
show_in_app	boolean	Флаг доступности в приложении для использования
updated	timestamp with time zone	Дата и время последнего изменения

t_sessions

Сессии пользователей

Поле	Тип	Описание
created	timestamp without time zone	Дата и время авторизации
expired	timestamp without time zone	Дата и время окончания срока сессии
sessions_limit_exceeded	boolean	Флаг, если true то сессия закрыта из-за превышения лимита параллельных сеансов
tk	character varying	Идентификатор (токен)
updated	timestamp without time zone	Дата и время последнего продления сессии
user_id	uuid	Идентификатор пользователя

t_storage_map_graph

Сохраненные данные графа задач

Поле	Тип	Описание
dat	json	Данные
project_id	uuid	Идентификатор задачи

t_table_owners

Блоки – "владельцы" таблиц БД (Справочник и Показатели)

Поле	Тип	Описание
------	-----	----------

block_id	uuid	Идентификатор блока
connection_data	character varying	Данные Соединения с БД
schema_name	character varying	Название схемы БД
table_name	character varying	Название таблицы БД

t_tags

Теги

Поле	Тип	Описание
color	character varying	Цвет
id	uuid	Идентификатор
name	character varying	Название
used	timestamp with time zone	Дата и время последнего использования

t_user_access_actions

Детальная настройка действий, которые необходимо логировать, в разрезе пользователей и категорий (устарела, в настоящее время не используется)

Поле	Тип	Описание
action_id	integer	Идентификатор категории
user_id	uuid	Идентификатор пользователя

t_user_presets

Использование пресетов пользователями

Поле	Тип	Описание
opened	timestamp with time zone	Дата и время последнего открытия пресета пользователем
preset_id	uuid	Идентификатор пресета
user_id	uuid	Идентификатор пользователя

(Устарело) t_user_scenarios

Выбранные пользователями сценарии

Поле	Тип	Описание
active	boolean	Флаг активности
proj_id	uuid	Идентификатор задачи
scenario_id	uuid	Идентификатор сценария
user_id	uuid	Идентификатор пользователя

t_users

Пользователи

Поле	Тип	Описание
auth_type	integer	Тип авторизации
days_before_blocking	integer	Количество дней до блокировки
email	character varying	E-mail пользователя
fname	character varying	Имя
hash	character varying	Хэш пароля пользователя
id	uuid	Идентификатор
last_active	timestamp with time zone	Дата и время изменения
lname	character varying	Фамилия
lock_	boolean	Блокировка пользователя

login	character varying	Уникальный логин пользователя
pass_min_length	integer	Минимальная длина пароля
password_changed	timestamp with time zone	Дата и время изменения пароля
patronym	character varying	Отчество
projects_order	integer	Сортировка задач по умолчанию
sessions_number	integer	Максимальное количество сессий
startup_task_id	uuid	Идентификатор задачи, открываемой после авторизации пользователя
temp_user	boolean	Флаг, временный пользователь
temp_user_finish	timestamp with time zone	Дата и время окончания активации пользователя
temp_user_start	timestamp with time zone	Дата и время начала активации пользователя
use_exp_func	boolean	Флаг, разрешающий использовать экспериментальный функционал

t_users_groups

Вхождение пользователей в группы

Поле	Тип	Описание
group_id	uuid	Идентификатор группы
user_id	uuid	Идентификатор пользователя

t_users_locks

Заблокированные пользователи

Поле	Тип	Описание
lock_date	timestamp without time zone	Дата и время блокировки пользователя
lock_duration	integer	Длительность блокировки, секунды
lock_type	integer	Тип блокировки
unlock_date	timestamp without time zone	Дата и время разблокировки пользователя
user_id	uuid	Идентификатор пользователя

t_vis_styles

Стили визуализаторов

Поле	Тип	Описание
block_id	uuid	Идентификатор блока
dat	json	Данные
descr	text	Описание стиля
hidden	boolean	Флаг, скрыт или нет
id	uuid	Идентификатор
name	character varying	Наименование
ord	integer	Порядок

t_vis_styles_ref

Ссылки на общие настройки стилей визуализаторов

Поле	Тип	Описание
lib_id	uuid	Идентификатор стиля
mod_id	uuid	Идентификатор общего стиля

ord	integer	Порядок
-----	---------	---------

t_vis_tsv

Визуализаторы: полнотекстовый поиск

Поле	Тип	Описание
name	character varying	Название
preset_id	uuid	Идентификатор пресета
tsv	tsvector	Информация для полнотекстового поиска
vis_id	uuid	Идентификатор визуализатора

t_workers

Зарегистрированные узлы расчета

Поле	Тип	Описание
address	character varying	Адрес сервиса
api_prefix	character varying	Префикс доступа к API сервиса
environment	character varying	Окружение с которым работает
id	uuid	Идентификатор
name	character varying	Название

Схема базы данных (репозитория)

Схема базы данных (репозитория) доступна по запросу к разработчику платформы.

Функции и процедуры репозитория

№	Функция	Описание	Параметры
1.	vm_get_presets	Получение списка пресетов для отображения в «Меню»	задача uuid
2.	vm_get_related_tasks	Получение списка связанных задач, при использовании блока «Задача»	задача uuid
3.	vm_get_block_outputs	Получение списка выходов блока с учетом иерархии	<ul style="list-style-type: none"> • задача uuid • блок uuid
4.	vm_get_block_inputs	Получение списка входов блока с учетом иерархии	<ul style="list-style-type: none"> • задача uuid • блок uuid
5.	vm_get_block_settings	Получение списка значений настроек блока с учетом иерархии	<ul style="list-style-type: none"> • задача uuid • блок uuid
6.	vm_get_block_setting_value	Получение одного значения из настроек блока с учетом иерархии	<ul style="list-style-type: none"> • задача uuid • блок uuid • имя настройки с учетом иерархии varchar
7.	vm_project_copy	Создание полной копии задачи, включая: <ul style="list-style-type: none"> • блоки • связи • пресеты • сценарии • привязки блоков 	<ul style="list-style-type: none"> • задача uuid • новое название varchar • автор uuid
8.	vm_get_blocks_using_table	Получение списка блоков (справочник или показатели), использующих указанную таблицу БД, поиск выполняется по всем задачам	<ul style="list-style-type: none"> • блок-владелец таблицы uuid • имя таблицы varchar • пользователь uuid • полномочия на чтение графа bool
9.	vm_delete_old_calculations	Функция для удаления старых расчетов во время сборки мусора	-

Приложение 3. Метрики Prometheus по умолчанию

```
# HELP flask_http_request_duration_seconds Multiprocess metric
# TYPE flask_http_request_duration_seconds histogram
flask_http_request_duration_seconds_sum{endpoint="versions.version",method="GET",status="200"} 0.8981670029461384
flask_http_request_duration_seconds_sum{endpoint="application.get_styles",method="GET",status="200"} 1.4094492476433516
flask_http_request_duration_seconds_sum{endpoint="auth.status_post",method="POST",status="200"} 1.340100983157754
flask_http_request_duration_seconds_sum{endpoint="admin.get_license",method="GET",status="200"} 0.7679055598564446
flask_http_request_duration_seconds_sum{endpoint="tasks.get_tasks",method="GET",status="200"} 1.2158617144450545
flask_http_request_duration_seconds_sum{endpoint="tags.get_tags",method="GET",status="200"} 0.14136453811079264
flask_http_request_duration_seconds_sum{endpoint="tasks.get_tasks_last_opened",method="GET",status="200"} 0.1545978100039065
flask_http_request_duration_seconds_sum{endpoint="scenarios.get_all_tasks_with_scenario",method="GET",status="200"} 0.9224562519229949
flask_http_request_duration_seconds_sum{endpoint="progressbar.calc_graph_get_progress",method="POST",status="200"} 258.88298390991986
flask_http_request_duration_seconds_sum{endpoint="graph.get_blocks_state",method="POST",status="200"} 0.1778613799251616
flask_http_request_duration_seconds_sum{endpoint="debug.get_block_debug_input_and_output",method="GET",status="200"} 61.21176861273125
flask_http_request_duration_seconds_sum{endpoint="library.get_libraries",method="GET",status="200"} 0.5758497272618115
flask_http_request_duration_seconds_sum{endpoint="tasks.get_task",method="GET",status="200"} 0.995703399181366
flask_http_request_duration_seconds_sum{endpoint="graph.get_tasks",method="GET",status="200"} 1.8132902421057224
flask_http_request_duration_seconds_sum{endpoint="presets.get_presets",method="GET",status="200"} 16.25772683136165
flask_http_request_duration_seconds_sum{endpoint="scenarios.get_scenario_by_task",method="GET",status="200"} 0.4887871718965471
flask_http_request_duration_seconds_sum{endpoint="debug.get_debug_vis",method="GET",status="200"} 44.124159295111895
flask_http_request_duration_seconds_sum{endpoint="presets.update_preset",method="PUT",status="200"} 0.44251092430204153
flask_http_request_duration_seconds_sum{endpoint="debug.get_block_debug_info",method="GET",status="200"} 0.7266020360402763
flask_http_request_duration_seconds_sum{endpoint="system.get_menu_styles_list",method="GET",status="200"} 0.23752294667065144
flask_http_request_duration_seconds_sum{endpoint="admin.get_users",method="GET",status="200"} 3.246767120435834
flask_http_request_duration_seconds_sum{endpoint="admin.get_roles",method="GET",status="200"} 0.09917925577610731
flask_http_request_duration_seconds_sum{endpoint="admin.get_groups",method="GET",status="200"} 3.4793411530554295
```

```
flask_http_request_duration_seconds_sum(endpoint="admin.get_users_tasks",meth
od="GET",status="200") 0.540447766892612
flask_http_request_duration_seconds_sum(endpoint="states.find_states_by_block
",method="POST",status="200") 0.49723414750769734
flask_http_request_duration_seconds_sum(endpoint="states.get_states",method="
GET",status="200") 0.26628985488787293
flask_http_request_duration_seconds_sum(endpoint="screenshots.get_screenshots
",method="GET",status="200") 0.26606357656419277
flask_http_request_duration_seconds_sum(endpoint="presets.get_preset",method=
"GET",status="200") 1.7903634295798838
flask_http_request_duration_seconds_sum(endpoint="library.get_blocks_preset",
method="GET",status="200") 0.2519649784080684
flask_http_request_duration_seconds_sum(endpoint="progressbar.calc_event",met
hod="POST",status="200") 1.165907091461122
flask_http_request_duration_seconds_sum(endpoint="system.get_connections_list
",method="GET",status="200") 0.07805519551038742
flask_http_request_duration_seconds_sum(endpoint="system.check_connection",me
thod="POST",status="200") 4.445009535178542
flask_http_request_duration_seconds_sum(endpoint="versions.version",method="H
EAD",status="200") 0.018980565946549177
flask_http_request_duration_seconds_sum(endpoint="globals.get_menu_styles",me
thod="GET",status="200") 0.11861382890492678
flask_http_request_duration_seconds_sum(endpoint="globals.get_connections",me
thod="GET",status="200") 0.6563644618727267
flask_http_request_duration_seconds_sum(endpoint="globals.get_connection_type
s",method="GET",status="200") 0.030361573211848736
flask_http_request_duration_seconds_bucket(endpoint="versions.version",le="0.
005",method="GET",status="200") 0.0
flask_http_request_duration_seconds_bucket(endpoint="versions.version",le="0.
01",method="GET",status="200") 0.0
flask_http_request_duration_seconds_bucket(endpoint="versions.version",le="0.
025",method="GET",status="200") 9.0
flask_http_request_duration_seconds_bucket(endpoint="versions.version",le="0.
05",method="GET",status="200") 9.0
flask_http_request_duration_seconds_bucket(endpoint="versions.version",le="0.
075",method="GET",status="200") 9.0
flask_http_request_duration_seconds_bucket(endpoint="versions.version",le="0.
1",method="GET",status="200") 10.0
flask_http_request_duration_seconds_bucket(endpoint="versions.version",le="0.
25",method="GET",status="200") 10.0
flask_http_request_duration_seconds_bucket(endpoint="versions.version",le="0.
5",method="GET",status="200") 10.0
flask_http_request_duration_seconds_bucket(endpoint="versions.version",le="0.
75",method="GET",status="200") 11.0
flask_http_request_duration_seconds_bucket(endpoint="versions.version",le="1.
0",method="GET",status="200") 11.0
flask_http_request_duration_seconds_bucket(endpoint="versions.version",le="2.
5",method="GET",status="200") 11.0
flask_http_request_duration_seconds_bucket(endpoint="versions.version",le="5.
0",method="GET",status="200") 11.0
```

```
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="7.5",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="10.0",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="+Inf",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_count{endpoint="versions.version",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="0.05",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="0.075",method="GET",status="200"} 9.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="0.1",method="GET",status="200"} 9.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="0.25",method="GET",status="200"} 10.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="0.5",method="GET",status="200"} 10.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="0.75",method="GET",status="200"} 10.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="1.0",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="2.5",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="5.0",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="7.5",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="10.0",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="application.get_styles",le="+Inf",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_count{endpoint="application.get_styles",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="0.005",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="0.01",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="0.025",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="0.05",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="0.075",method="POST",status="200"} 5.0
```

```
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="0.1",method="POST",status="200"} 7.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="0.25",method="POST",status="200"} 10.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="0.5",method="POST",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="0.75",method="POST",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="1.0",method="POST",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="2.5",method="POST",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="5.0",method="POST",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="7.5",method="POST",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="10.0",method="POST",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="auth.status_post",le="+Inf",method="POST",status="200"} 11.0
flask_http_request_duration_seconds_count{endpoint="auth.status_post",method="POST",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="0.025",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="0.05",method="GET",status="200"} 7.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="0.075",method="GET",status="200"} 7.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="0.1",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="0.25",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="0.5",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="0.75",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="1.0",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="2.5",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="5.0",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="7.5",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="10.0",method="GET",status="200"} 11.0
```

```
flask_http_request_duration_seconds_bucket{endpoint="admin.get_license",le="+Inf",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_count{endpoint="admin.get_license",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="0.05",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="0.075",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="0.1",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="0.25",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="0.5",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="0.75",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="1.0",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="2.5",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="5.0",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="7.5",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="10.0",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks",le="+Inf",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_count{endpoint="tasks.get_tasks",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="0.05",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="0.075",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="0.1",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="0.25",method="GET",status="200"} 2.0
```

```
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="0.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="0.75",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="1.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="2.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="5.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="7.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="10.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tags.get_tags",le="+Inf",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_count{endpoint="tags.get_tags",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="0.075",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="0.1",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="0.25",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="0.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="0.75",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="1.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="2.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="5.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="7.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="10.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_tasks_last_opened",le="+Inf",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_count{endpoint="tasks.get_tasks_last_opened",method="GET",status="200"} 2.0
```

```

flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="0.05",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="0.075",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="0.1",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="0.25",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="0.5",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="0.75",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="1.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="2.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="5.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="7.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="10.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_all_tasks_
with_scenario",le="+Inf",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_count{endpoint="scenarios.get_all_tasks_w
ith_scenario",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_g
et_progress",le="0.005",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_g
et_progress",le="0.01",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_g
et_progress",le="0.025",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_g
et_progress",le="0.05",method="POST",status="200"} 299.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_g
et_progress",le="0.075",method="POST",status="200"} 2455.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_g
et_progress",le="0.1",method="POST",status="200"} 2473.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_g
et_progress",le="0.25",method="POST",status="200"} 2516.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_g
et_progress",le="0.5",method="POST",status="200"} 2550.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_g
et_progress",le="0.75",method="POST",status="200"} 2551.0

```

```
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_get_progress",le="1.0",method="POST",status="200"} 2551.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_get_progress",le="2.5",method="POST",status="200"} 2553.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_get_progress",le="5.0",method="POST",status="200"} 2555.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_get_progress",le="7.5",method="POST",status="200"} 2555.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_get_progress",le="10.0",method="POST",status="200"} 2555.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_graph_get_progress",le="+Inf",method="POST",status="200"} 2559.0
flask_http_request_duration_seconds_count{endpoint="progressbar.calc_graph_get_progress",method="POST",status="200"} 2559.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="0.005",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="0.01",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="0.025",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="0.05",method="POST",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="0.075",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="0.1",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="0.25",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="0.5",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="0.75",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="1.0",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="2.5",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="5.0",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="7.5",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="10.0",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_blocks_state",le="+Inf",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_count{endpoint="graph.get_blocks_state",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_input_and_output",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_input_and_output",le="0.01",method="GET",status="200"} 0.0
```

```

flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="0.075",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="0.1",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="0.25",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="0.5",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="0.75",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="1.0",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="2.5",method="GET",status="200"} 7.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="5.0",method="GET",status="200"} 7.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="7.5",method="GET",status="200"} 7.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="10.0",method="GET",status="200"} 7.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_in
put_and_output",le="+Inf",method="GET",status="200"} 9.0
flask_http_request_duration_seconds_count{endpoint="debug.get_block_debug_inp
ut_and_output",method="GET",status="200"} 9.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",l
e="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",l
e="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",l
e="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",l
e="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",l
e="0.075",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",l
e="0.1",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",l
e="0.25",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",l
e="0.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",l
e="0.75",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",l
e="1.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",l
e="2.5",method="GET",status="200"} 4.0

```

```

flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",le="5.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",le="7.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",le="10.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_libraries",le="+Inf",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_count{endpoint="library.get_libraries",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="0.075",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="0.1",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="0.25",method="GET",status="200"} 7.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="0.5",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="0.75",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="1.0",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="2.5",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="5.0",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="7.5",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="10.0",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="tasks.get_task",le="+Inf",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_count{endpoint="tasks.get_task",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="0.05",method="GET",status="200"} 0.0

```

```
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="0.075",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="0.1",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="0.25",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="0.5",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="0.75",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="1.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="2.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="5.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="7.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="10.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="graph.get_tasks",le="+Inf",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_count{endpoint="graph.get_tasks",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="0.075",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="0.1",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="0.25",method="GET",status="200"} 9.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="0.5",method="GET",status="200"} 9.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="0.75",method="GET",status="200"} 9.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="1.0",method="GET",status="200"} 10.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="2.5",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="5.0",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="7.5",method="GET",status="200"} 11.0
```

```
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="10.0",method="GET",status="200"} 11.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_presets",le="+Inf",method="GET",status="200"} 12.0
flask_http_request_duration_seconds_count{endpoint="presets.get_presets",method="GET",status="200"} 12.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="0.05",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="0.075",method="GET",status="200"} 7.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="0.1",method="GET",status="200"} 7.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="0.25",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="0.5",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="0.75",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="1.0",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="2.5",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="5.0",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="7.5",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="10.0",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="scenarios.get_scenario_by_task",le="+Inf",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_count{endpoint="scenarios.get_scenario_by_task",method="GET",status="200"} 8.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="0.075",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="0.1",method="GET",status="200"} 0.0
```

```

flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="0.25",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="0.5",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="0.75",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="1.0",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="2.5",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="5.0",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="7.5",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="10.0",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_debug_vis",le="+Inf",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_count{endpoint="debug.get_debug_vis",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="0.005",method="PUT",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="0.01",method="PUT",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="0.025",method="PUT",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="0.05",method="PUT",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="0.075",method="PUT",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="0.1",method="PUT",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="0.25",method="PUT",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="0.5",method="PUT",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="0.75",method="PUT",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="1.0",method="PUT",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="2.5",method="PUT",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="5.0",method="PUT",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="7.5",method="PUT",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="10.0",method="PUT",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="presets.update_preset",le="+Inf",method="PUT",status="200"} 3.0

```

```

flask_http_request_duration_seconds_count{endpoint="presets.update_preset",method="PUT",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="0.05",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="0.075",method="GET",status="200"} 5.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="0.1",method="GET",status="200"} 5.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="0.25",method="GET",status="200"} 5.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="0.5",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="0.75",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="1.0",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="2.5",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="5.0",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="7.5",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="10.0",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="debug.get_block_debug_info",le="+Inf",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_count{endpoint="debug.get_block_debug_info",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="0.05",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="0.075",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="0.1",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="0.25",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="0.5",method="GET",status="200"} 6.0

```

```

flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="0.75",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="1.0",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="2.5",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="5.0",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="7.5",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="10.0",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_menu_styles_list",le="+Inf",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_count{endpoint="system.get_menu_styles_list",method="GET",status="200"} 6.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="0.075",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="0.1",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="0.25",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="0.5",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="0.75",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="1.0",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="2.5",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="5.0",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="7.5",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="10.0",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users",le="+Inf",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_count{endpoint="admin.get_users",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="0.05",method="GET",status="200"} 0.0

```

```
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="0.05",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="0.075",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="0.1",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="0.25",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="0.5",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="0.75",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="1.0",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="2.5",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="5.0",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="7.5",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="10.0",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_roles",le="+Inf",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_count{endpoint="admin.get_roles",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="0.075",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="0.1",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="0.25",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="0.5",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="0.75",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="1.0",method="GET",status="200"} 1.0
```

```

flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="2.5",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="5.0",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="7.5",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="10.0",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_groups",le="+Inf",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_count{endpoint="admin.get_groups",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="0.075",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="0.1",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="0.25",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="0.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="0.75",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="1.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="2.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="5.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="7.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="10.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="admin.get_users_tasks",le="+Inf",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_count{endpoint="admin.get_users_tasks",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="0.005",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="0.01",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="0.025",method="POST",status="200"} 0.0

```

```

flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="0.05",method="POST",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="0.075",method="POST",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="0.1",method="POST",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="0.25",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="0.5",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="0.75",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="1.0",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="2.5",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="5.0",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="7.5",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="10.0",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.find_states_by_block",le="+Inf",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_count{endpoint="states.find_states_by_block",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="0.075",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="0.1",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="0.25",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="0.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="0.75",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="1.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="2.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="5.0",method="GET",status="200"} 4.0

```

```

flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="7
.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="1
0.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="states.get_states",le="+
Inf",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_count{endpoint="states.get_states",method
="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="0.05",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="0.075",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="0.1",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="0.25",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="0.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="0.75",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="1.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="2.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="5.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="7.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="10.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="screenshots.get_screensh
ots",le="+Inf",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_count{endpoint="screenshots.get_screensho
ts",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
0.075",method="GET",status="200"} 0.0

```

```

flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
0.1",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
0.25",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
0.5",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
0.75",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
1.0",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
2.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
5.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
7.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
10.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="presets.get_preset",le="
+Inf",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_count{endpoint="presets.get_preset",metho
d="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="0.05",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="0.075",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="0.1",method="GET",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="0.25",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="0.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="0.75",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="1.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="2.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="5.0",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="7.5",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_prese
t",le="10.0",method="GET",status="200"} 4.0

```

```
flask_http_request_duration_seconds_bucket{endpoint="library.get_blocks_preset",le="+Inf",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_count{endpoint="library.get_blocks_preset",method="GET",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="0.005",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="0.01",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="0.025",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="0.05",method="POST",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="0.075",method="POST",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="0.1",method="POST",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="0.25",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="0.5",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="0.75",method="POST",status="200"} 3.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="1.0",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="2.5",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="5.0",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="7.5",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="10.0",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="progressbar.calc_event",le="+Inf",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_count{endpoint="progressbar.calc_event",method="POST",status="200"} 4.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="0.05",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="0.075",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="0.1",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="0.25",method="GET",status="200"} 2.0
```

```

flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="0.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="0.75",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="1.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="2.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="5.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="7.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="10.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="system.get_connections_list",le="+Inf",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_count{endpoint="system.get_connections_list",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="0.005",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="0.01",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="0.025",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="0.05",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="0.075",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="0.1",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="0.25",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="0.5",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="0.75",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="1.0",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="2.5",method="POST",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="5.0",method="POST",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="7.5",method="POST",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="10.0",method="POST",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="system.check_connection",le="+Inf",method="POST",status="200"} 1.0
flask_http_request_duration_seconds_count{endpoint="system.check_connection",method="POST",status="200"} 1.0

```

```

flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="0.005",method="HEAD",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="0.01",method="HEAD",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="0.025",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="0.05",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="0.075",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="0.1",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="0.25",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="0.5",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="0.75",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="1.0",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="2.5",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="5.0",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="7.5",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="10.0",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="versions.version",le="+Inf",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_count{endpoint="versions.version",method="HEAD",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles",le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles",le="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles",le="0.075",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles",le="0.1",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles",le="0.25",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles",le="0.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles",le="0.75",method="GET",status="200"} 2.0

```

```

flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles"
,le="1.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles"
,le="2.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles"
,le="5.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles"
,le="7.5",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles"
,le="10.0",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_menu_styles"
,le="+Inf",method="GET",status="200"} 2.0
flask_http_request_duration_seconds_count{endpoint="globals.get_menu_styles",
method="GET",status="200"} 2.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="0.01",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="0.05",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="0.075",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="0.1",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="0.25",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="0.5",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="0.75",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="1.0",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="2.5",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="5.0",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="7.5",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="10.0",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connections"
,le="+Inf",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_count{endpoint="globals.get_connections",
method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="0.005",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="0.01",method="GET",status="200"} 0.0

```

```

flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="0.025",method="GET",status="200"} 0.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="0.05",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="0.075",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="0.1",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="0.25",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="0.5",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="0.75",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="1.0",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="2.5",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="5.0",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="7.5",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="10.0",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_bucket{endpoint="globals.get_connection_t
ypes",le="+Inf",method="GET",status="200"} 1.0
flask_http_request_duration_seconds_count{endpoint="globals.get_connection_ty
pes",method="GET",status="200"} 1.0
# HELP build_version_info Multiprocess metric
# TYPE build_version_info gauge
build_version_info{version="0.37.0.18276-241f3ba6-develop-09.08.2023 11:01:30-
coordinator"} 1.0
# HELP flask_http_request_total Multiprocess metric
# TYPE flask_http_request_total counter
flask_http_request_total{method="GET",status="200"} 139.0
flask_http_request_total{method="POST",status="200"} 2582.0
flask_http_request_total{method="PUT",status="200"} 3.0
flask_http_request_total{method="HEAD",status="200"} 1.0
# HELP flask_exporter_info Multiprocess metric
# TYPE flask_exporter_info gauge
flask_exporter_info{version="0.20.2"} 1.0
# HELP count_for_sum Multiprocess metric
# TYPE count_for_sum gauge
count_for_sum{count="calculations"} 0.0
count_for_sum{count="open_presets"} 1.0

```

История изменений

№ п/п	Версия	Дата	Внесенные изменения
1.	1.0	31.01.2025	Руководство администратора ЦП УВП (версия платформы 0.43)