УТВЕРЖДАЮ

Директор ООО «БИТ»

_____Ю.В. Щербаков

«_3_» июня 2024 г.

ПРОГРАММА ДЛЯ ЭВМ
С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ПРОГНОЗИРОВАНИЯ ЧАСОВ ПИКОВОЙ НАГРУЗКИ И ОПТИМИЗАЦИИ ЭНЕРГОПОТРЕБЛЕНИЯ ОРГАНИЗАЦИИ («ЭНЕРГОБИТ»)

Инструкция для системного программиста

ЛИСТ УТВЕРЖДЕНИЯ

СОГЛАСОВАНО:

Главный консультант проекта

_____А.И. Лукашов

«_3_» июня 2024г.

Начальник отдела разработки

_____ У.Н. Асекритов

«_3_» июня 2024 г.

Руководитель IT-проектов

_____ С.С. Шахов

«_3_» июня 2024г.

2024

Инв. № подл. | Подпись и дата | Взам. инв. № | Инв. № дубл. | Подпись и дата

УТВЕРЖДЕН

ПРОГРАММА ДЛЯ ЭВМ
С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ПРОГНОЗИРОВАНИЯ ЧАСОВ ПИКОВОЙ НАГРУЗКИ И ОПТИМИЗАЦИИ ЭНЕРГОПОТРЕБЛЕНИЯ ОРГАНИЗАЦИИ («ЭНЕРГОБИТ»)

Инструкция для системного программиста

Листов 58

2024

Подпись и дата
Инв. № дубл.
Взам. инв. №
Подпись и дата
Инв. № подл.

АННОТАЦИЯ

Настоящий документ содержит описание назначения и функций, выполняемых ПО «ЭнергоБИТ», необходимые требования к техническим средствам для работы. Приведено описание основных характеристик, входных и выходных данных.

СОДЕРЖАНИЕ

1 Общие сведения 4
1.1 Обозначение и наименование программы 4
1.2 Назначение и функции ПО «ЭнергоБИТ» 4
1.3 Необходимые технические и программные средства 5
2 Структура ПО «ЭнергоБИТ» 7
2.1 Общая структура 7
3 Установка ПО «ЭнергоБИТ» 10
3.1. Необходимое общее программное обеспечение……10
3.2 Установка модулей подсистем, использующих методы искусственного интеллекта ..... **Ошибка! Закладка не определена.**
3.3 Установка java-приложения 16
3.4 Запуск web-фронтенд приложений……27
3.5 Установка сервиса Авторизации……32
3.6 Запуск сервиса подкачки данных из системы Элдис……..……37
4 Настройка ПО "ЭнергоБИТ"…….38
4.1 Создание пользователей и организаций в сервисе авторизации……38
4.2 Создание пользователей и организаций в сервисе администрирования……...……40
4.3 Сервис администрирования……42
Обозначения и сокращения 56
Термины и определения 57

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Обозначение и наименование программы

Программа для ЭВМ с использованием искусственного интеллекта для прогнозирования часов пиковой нагрузки и оптимизации энергопотребления организации.

Обозначение: ПО «ЭнергоБИТ».

1.2 Назначение и функции ПО «ЭнергоБИТ

1.2.1 ПО «ЭнергоБИТ» предназначено для автоматизации процесса принятия управленческих решений по снижению затрат на электроэнергию, получение пригодных для оперативного использования данных об энергопотреблении от интеллектуальных систем учёта электроэнергии, позволяющие оценить следующие параметры:

– текущие и перспективные затраты на оплату и физические объёмы потребления электроэнергии;

– удельные показатели затрат и объёмов потребления электроэнергии, с выявлением центров потерь;

– качество поставляемой электроэнергии;

– частота, продолжительность и последствия перерывов в электроснабжении;

– работоспособность системы интеллектуального учёта электроэнергии;

– уровень обеспечения информационной безопасности, количество и последствия попыток несанкционированного вмешательства в работу ИСУ;

– степень выполнения заданий по снижению электропотребления.

1.2.2 В ПО «ЭнергоБИТ» реализованы следующие функции:

– чтение объёмов энергопотребления, почасовых рыночных цен и часов пиковой нагрузки за определённый период для заданного объекта, категории, мощности и уровня напряжения;

– статистический анализ выбранного набора данных;

– построение графиков и диаграмм;

– отображение результатов прогнозирования в табличном и графическом виде с расчётом стандартных показателей точности прогноза;

– определение оптимального суточного профиля энергопотребления с учётом заданных ограничений.

1.3 Необходимые технические и программные средства

1.3.1 Серверная часть ПО «ЭнергоБИТ» разворачивается и функционирует на серверах (в том числе облачных) ООО «БИТ», а также по отдельной лицензии на серверах предприятия под управлением операционных систем Linux Ubuntu. Актуальные поддерживаемые версии Linux Ubuntu смотрите по ссылке: https://www.oracle.com/java/technologies/javase/products-doc-jdk21certconfig.html.

Ниже приведены минимальные и рекомендуемые параметры конфигурации серверной части. Для всех перечисленных аппаратных компонентов и их устройств должны быть установлены программные средства, систем хранения, сетевого и иного оборудования, драйверы производителя (или сертифицированные совместимые).

Минимальные требования к аппаратному обеспечению сервера:

– процессор Intel Core i5, количество ядер процессора не менее 2, частота ядер не менее 1,8 МГц,

– объем ОЗУ не менее 6 Гб,

– объем видеопамяти не менее 2 Гб,

– объем диска HDD не менее 250 Гб,

– скорость адаптера LAN не менее 1000 Мбит/с.

Рекомендуемые требования к аппаратному обеспечению:

– процессор Intel Core i9, количество ядер процессора 10, частота ядер 3,3 МГц,

– объем ОЗУ 32 Гб, 2

– графические карты NVIDIA RTX 2070 с объёмом видеопамяти 8 Гб,

– объем диска SSD 500 Мб,

– объем диска HDD 2 Тб,

– 2 адаптера LAN со скоростью 1000 Мбит/с.

На сервере должен быть установлен Docker.

1.3.2 Доступ пользователей к клиентской части ПО «ЭнергоБИТ» осуществляется через web-приложение с использованием одного из нижеперечисленных браузеров актуальной версии:

– Chrome;

– Chromium;

– Firefox;

– Edge.

.

2 СТРУКТУРА ПО «ЭНЕРГОБИТ»

2.1 Общая структура

В целом ПО «ЭнергоБИТ» состоит из следующих подсистем:

– подсистема получения и хранения информации об энергопотреблении, ценах на электроэнергию и часах пиковой нагрузки;

– подсистема выработки рекомендаций по оптимизации и управлению энергопотреблением в часы пиковой нагрузки;

– подсистема расчёта количественных и финансовых показателей по потреблённой электроэнергии и затратах на электроэнергию на заданный промежуток времени.

Указанные выше подсистемы включают в себя следующие компоненты ПО «ЭнергоБИТ»:

– сервисы с использованием методов искусственного интеллекта:
  – модуль анализа и прогнозирования энергопотребления;
  – модуль прогнозирования цен на электроэнергию;
  – модуль прогнозирования часов пиковой нагрузки.
– калькулятор ценовых категорий;
– базы данных;
– система управления правами пользователей;
– web-интерфейс.

Взаимодействие сервисов осуществляется с помощью системы gRPC, по протоколам HTTP/2.

Подробное описание компонентов ПО «ЭнергоБИТ» приведено в документе «Описание программы».

ПО «ЭнергоБИТ» имеет гибридную микросервисную архитектуру, то есть построено по принципам облачного сервиса, так и по принципам отдельно поставляемого продукта.

Схема архитектуры ПО «ЭнергоБИТ» представлена на Рисунке 1.

Основное приложение представляет собой монолит, содержащий веб-сервис с интерфейсной частью и логику отображения данных, оповещения о событиях, калькулятор ценовых категорий. Функции импорта данных по ценам, часам пиковой нагрузки, данных от приборов учёта, прочих данных выделены в отдельные микросервисы. Для удобства интеграции и обновления модули с использованием методов искусственного интеллекта тоже выделены в отдельные микросервисы (docker-контейнеры).

Вариант отдельной поставки приложения клиентам предполагает размещение на серверах Заказчика основного веб-сервиса и сервисов по сбору информации с приборов учёта. Данные хранятся в локальной клиентской базе данных. Все сведения, требуемые системой из внешних источников, синхронизируются с основного облачного сервиса ООО «БИТ». Основной сервер лицензий так же находится на серверах ООО «БИТ». На серверах Заказчиков может располагаться вспомогательный сервис лицензирования, позволяющий работать при отключении от интернета и/или в закрытом сетевом контуре. Из-за повышенных требований по безопасности при передаче данных между клиентским сервером и основным сервером приложения вся информация передаётся по шифрованным каналам связи.

Приоритетным является вариант, при котором все операции по выработке рекомендаций и расчёт прогнозов нейросетью осуществляются на облачных серверах ООО «БИТ».

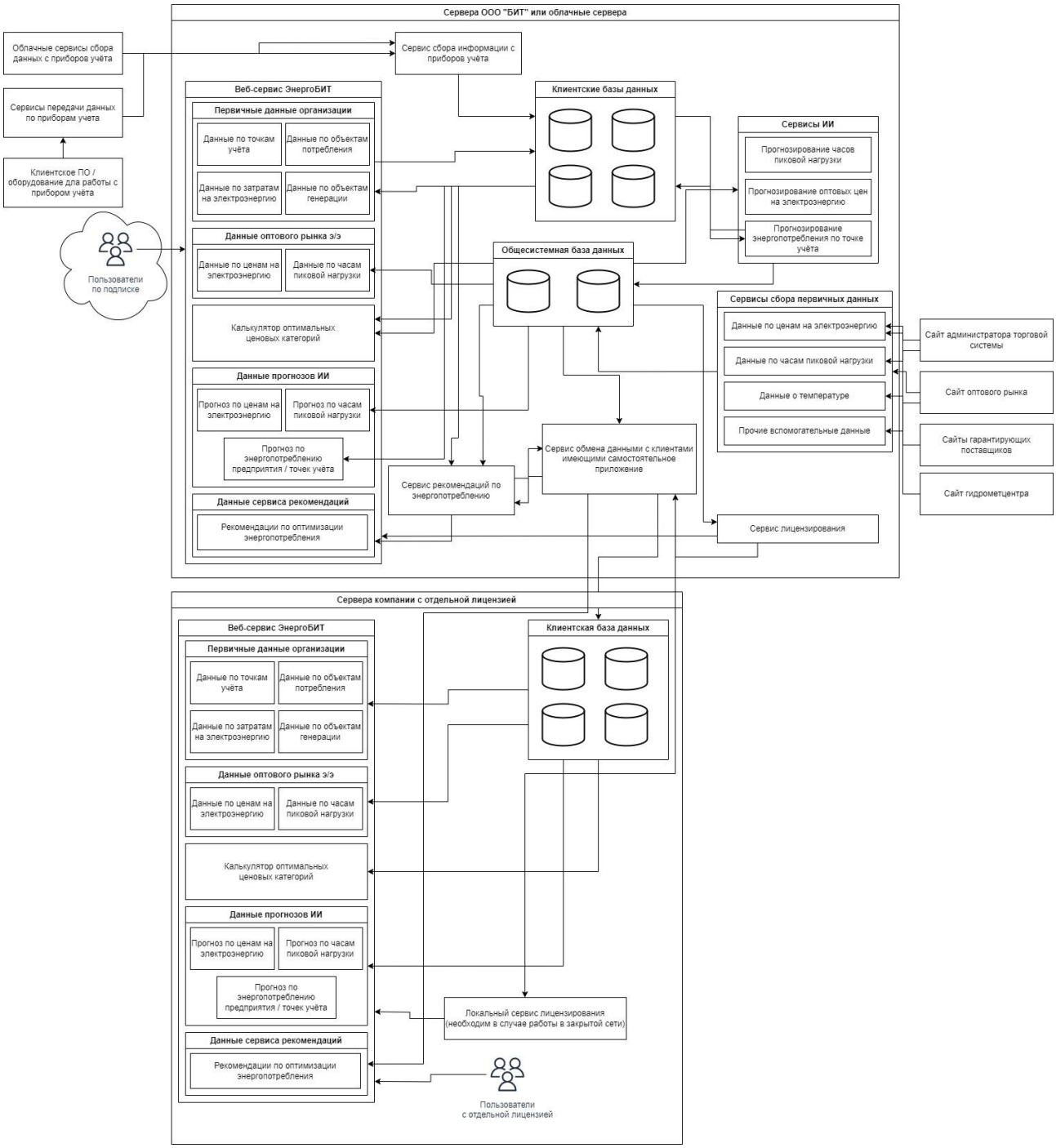


Рисунок 1. Схема архитектуры ПО «ЭнергоБИТ».

3 УСТАНОВКА ПО «ЭНЕРГОБИТ»

3.1 Необходимое общее программное обеспечение

- операционная система Ubuntu Linux 20.04 x64;
- Docker, docker compose;
- Dot.net;
- среда выполнения openjdk jre версии не ниже 21;
- СУБД PostgreSQL версии 15 и выше из репозиториев ОС;
- Nginx последней версии из репозиториев ОС (в качестве web сервера и прокси-сервера используется).

Все манипуляции за исключением копирования файлов и работы в PgAdmin4 выполняются через ssh.

Копирование файлов на сервере можно производить любым удобным способом.

3.2 Установка модулей подсистем, использующих методы искусственного интеллекта

3.2.1 Установка docker/docker-compose

Установить docker на компьютер под операционную систему Ubuntu, если вы еще это не сделали.

Настройка репозитория Docker. Обновите индекс пакета apt и установите пакеты, чтобы разрешить apt использовать репозиторий по протоколу HTTPS:

```sh
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
```

Добавьте официальный GPG-ключ Docker:

```sh
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

Используйте следующую команду для настройки репозитория:

```sh
echo \
      "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Установите Docker Engine.

Обновите индекс пакета apt:

```sh
sudo apt-get update
```

Установите Docker Engine, containerd и Docker Compose последней версии:

```sh
```

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

```

Убедитесь, что установка Docker Engine прошла успешно, запустив образ hello-world:

```sh
sudo docker run hello-world
```

Ссылка на инструкцию для установки docker - `https://docs.docker.com/engine/install/ubuntu/`.

Ссылка на основные команды docker - `https://docs.docker.com/engine/reference/commandline/compose/`

3.2.2 Установка сервиса price подсистемы прогнозирования цен на электроэнергию с использованием методов искусственного интеллекта

Распаковать архив с исходным кодом командой в корне проекта:

```sh
unzip price.zip
```

После распаковки архива появится папка с исходным кодом /price. Перейти в папку /price:

```sh

cd price/
```

Запустить контейнер с приложением:

```sh
docker-compose up -d --build
```

Проверить что контейнеры запустились без ошибок командой:

```sh
docker ps -a
```

после чего в терминале появится вывод:

```
```
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
1f368172655f   nginx:alpine   "/docker-entrypoint.…"   54 seconds ago   Up 53 seconds   0.0.0.0:80->80/tcp   proxy_nginx
d77a862717eb   price-app   "bash -c 'python mai…"   54 seconds ago   Up 53 seconds   0.0.0.0:8000->8000/tcp   price_app
```
```

Запрос POST `http://your_host_name:80/price`.

3.2.3 Установка сервиса hour подсистемы прогнозирования часов пиковой нагрузки с использованием методов искусственного интеллекта

Распаковать архив с исходным кодом командой в корне проекта:

```sh
```
```

unzip hour.zip
```

После распаковки архива появится папка с исходным кодом /hour. Перейти в папку /hour:

```sh
cd hour/
```

Запустить контейнер с приложением:

```sh
docker-compose up -d --build
```

Проверить что контейнеры запустились без ошибок командой:

```sh
docker ps -a
```

после чего в терминале появится вывод:

```
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                    NAMES
1f368172655f   nginx:alpine   "/docker-entrypoint.…"   54 seconds ago   Up 53 seconds   0.0.0.0:80->80/tcp       proxy_nginx
d77a862717eb   hour-app       "bash -c 'python mai…"   54 seconds ago   Up 53 seconds   0.0.0.0:8000->8000/tcp   hour_app
```
```

Запрос POST `http://your_host_name:80/hour`.

3.2.4 Установка сервиса consumption подсистемы анализа и прогнозирования энергопотребления с использованием методов искусственного интеллекта

Распаковать архив с исходным кодом командой в корне проекта:

```sh
unzip consumption.zip
```

После распаковки архива появится папка с исходным кодом /consumption. Перейти в папку /consumption:

```sh
cd consumption/
```

Запустить контейнер с приложением:

```sh
docker-compose up -d --build
```

Проверить что контейнеры запустились без ошибок командой:

```sh
docker ps -a
```

```

после чего в терминале появится вывод:

```
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
1f368172655f   nginx:alpine   "/docker-entrypoint.…"   54 seconds ago   Up 53 seconds   0.0.0.0:80->80/tcp   proxy_nginx
d77a862717eb   consumption-app   "bash -c 'python mai…"   54 seconds ago   Up 53 seconds   0.0.0.0:8000->8000/tcp   consumption_app
```

Запрос POST `http://your_host_name:80/consumption/<n_feature>`, где n_feature = 1 / 2 / 3

- `1` - потребление;
- `2` - потребление с учетом температуры;
- `3` - потребление с учетом высоты солнца.

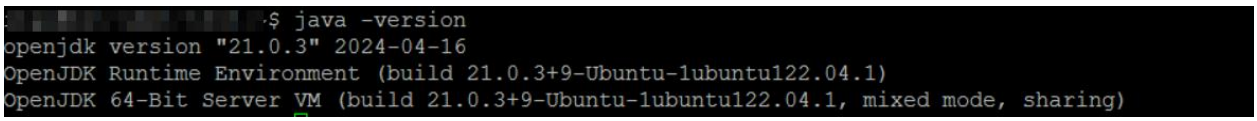
## 3.3 Установка Java-приложения

### 3.3.1 Установка JRE

Выполните установку командой sudo apt install openjdk-21-jre.

Проверьте, что версия установилась командой java–version.

Если установка произведена верно, вы увидите следующее:




### 3.3.2 Установка PostgreeSQL

sudo apt install postgresql postgresql-contrib

sudo systemctl enable postgresql.service
```

sudo systemctl start postgresql.service

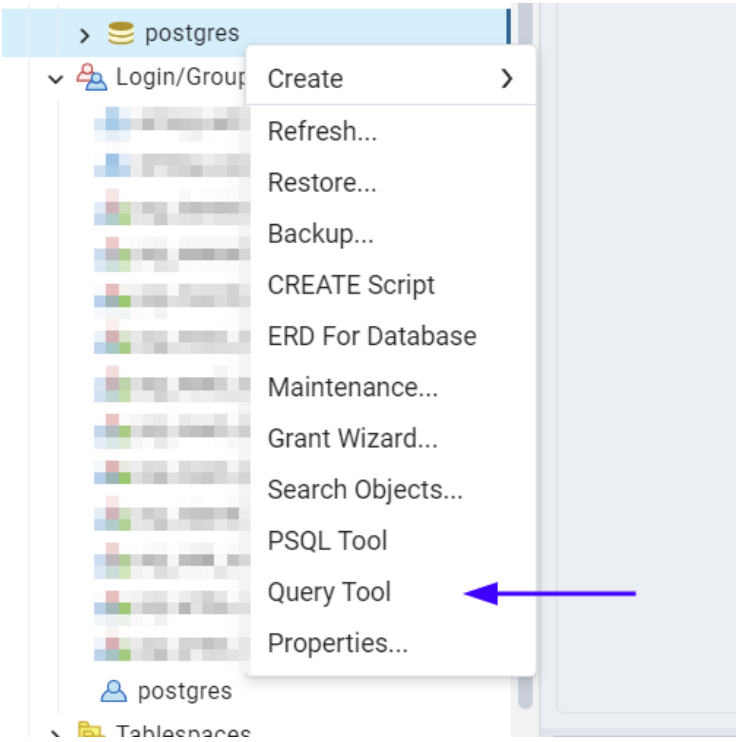
sudo systemctl status postgresql.service

3.3.3 Развёртывание баз данных PostgreeSQL

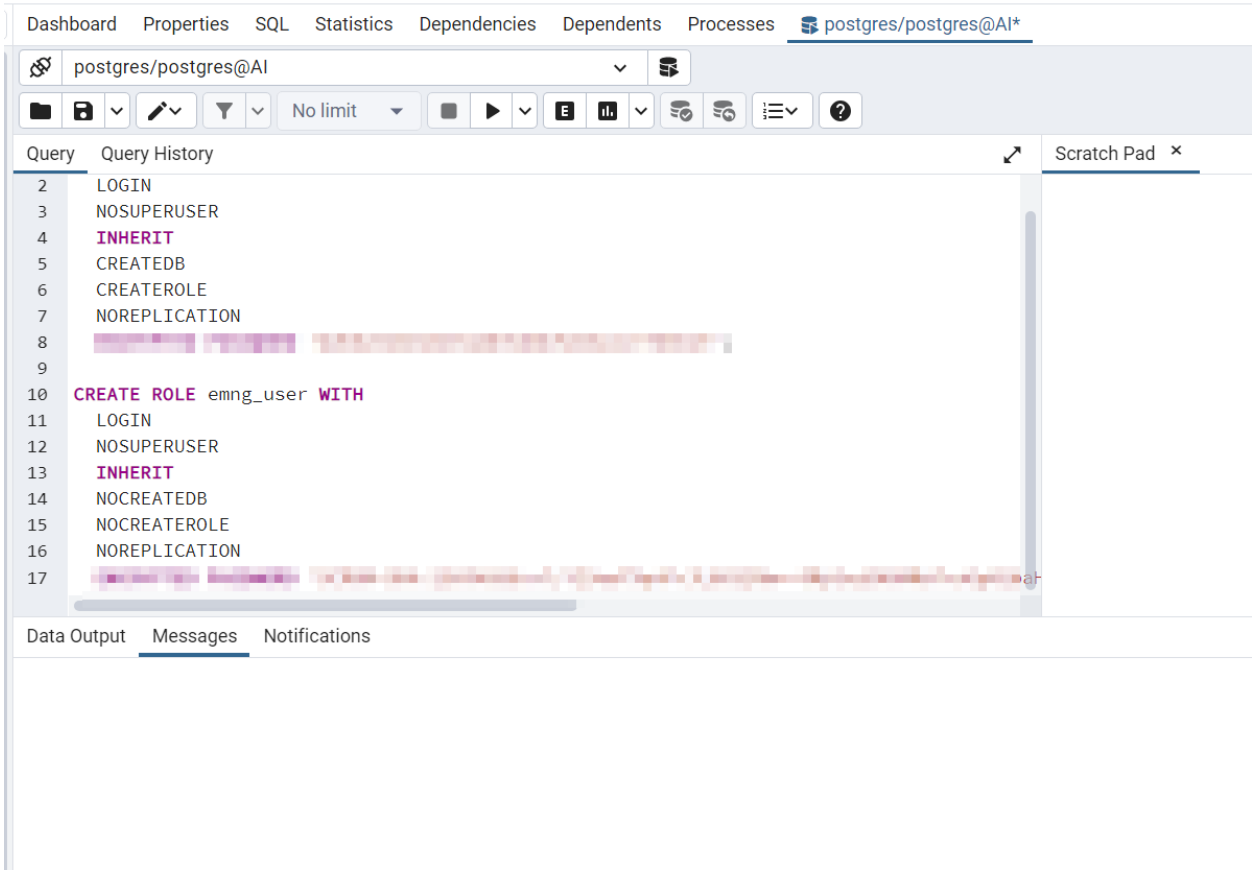
Базы данных разворачиваются из образов через pgAdmin4.

Предварительно необходимо выполнить user.sql в pgadmin4 (данный файл содержит запросы по созданию пользователей БД для работы приложения).

Для выполнения необходимо подключиться к установленной СУБД.

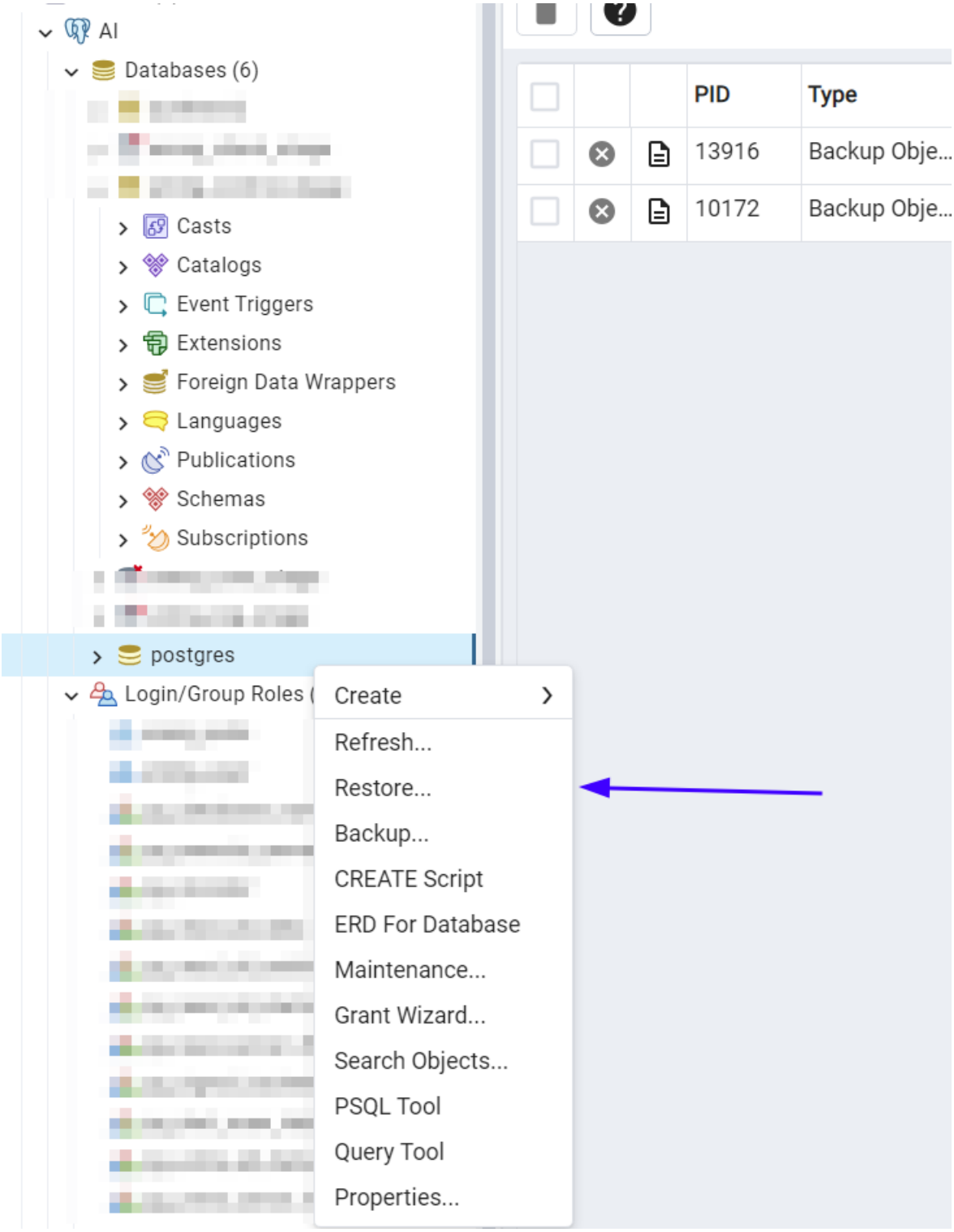


Пример вызова из среды PgAdmin

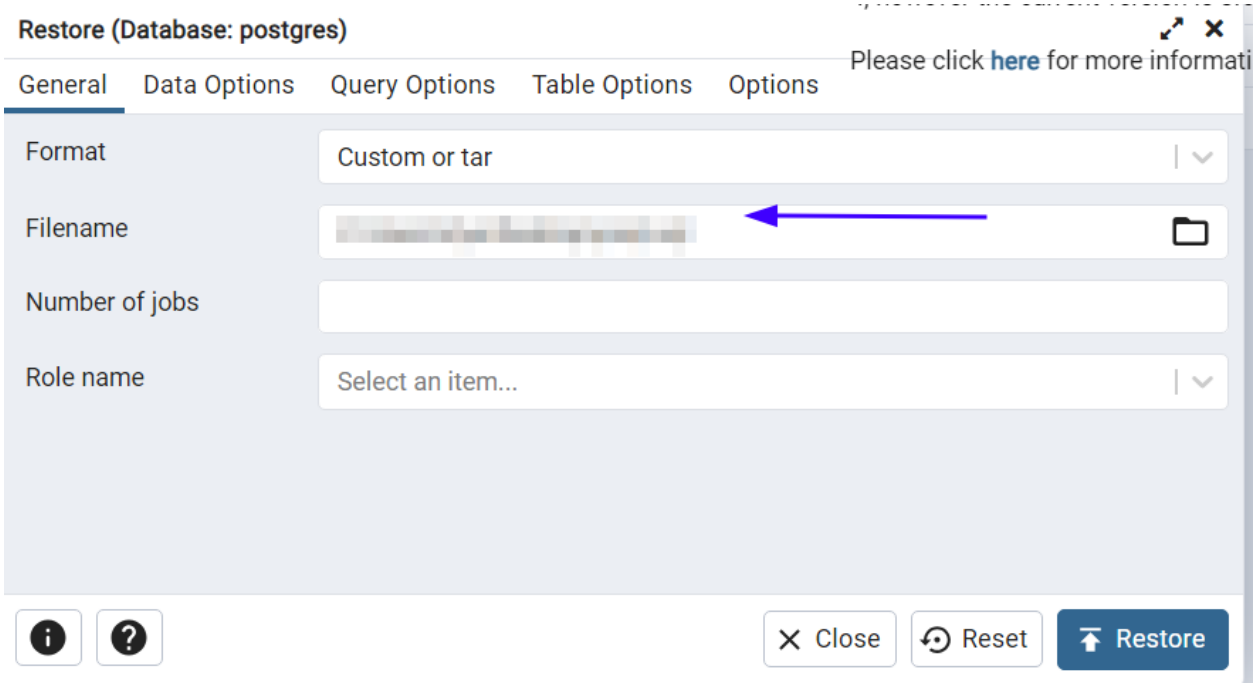


Выполнение запросов по созданию пользователей.

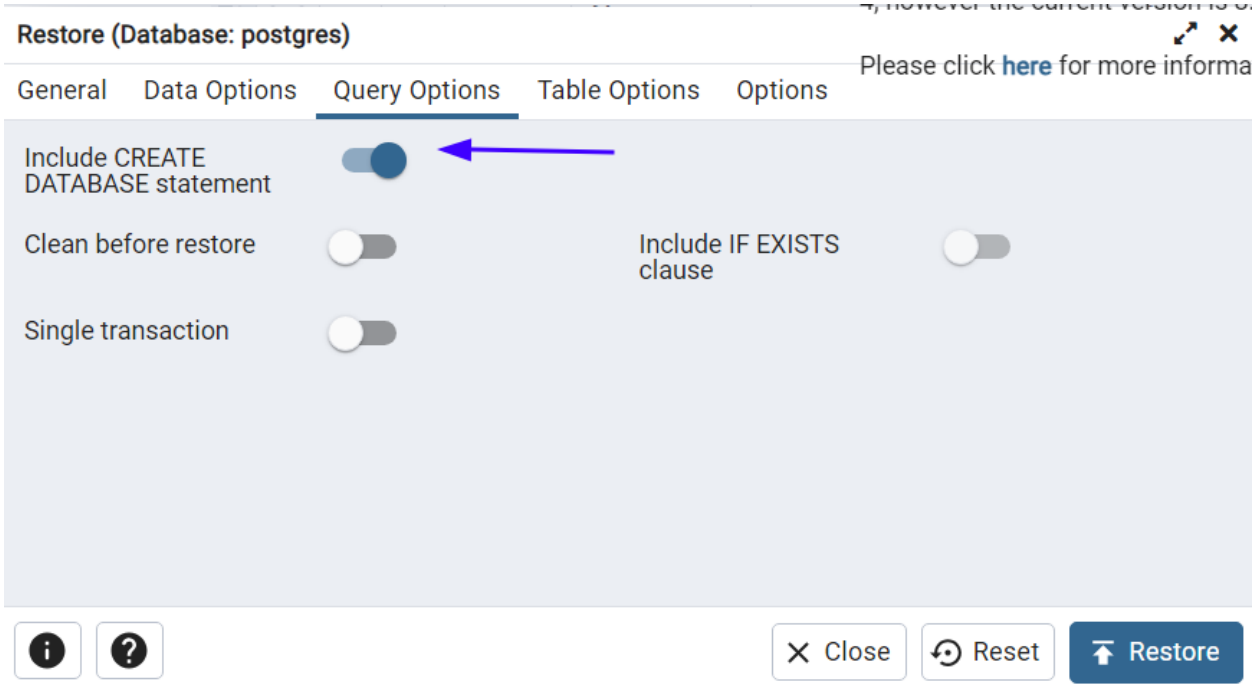
Выполняем развертывание БД из архивных копий



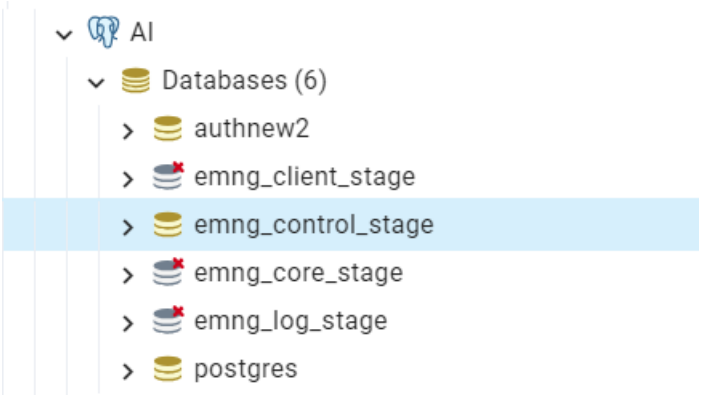
Выбираем путь до файлов БД в каталогах для установки



Включаем развертывание БД



Данную операцию необходимо провести для каждой из приложенных в дистрибутиве БД и вы должны увидеть следующий результат:



3.3.4 Копирование дистрибутивов на сервер

Необходимо скопировать дистрибутивы ai-connector-0.1.0.jar, api-gateway-0.1.0.jar, core-0.1.0.jar, client-0.1.0.jar, control-0.3.0.jar, data-import-0.0.1.jar, log-0.3.0.jar и каталог neural_networks

Размещение дистрибутивов показано на скриншоте

```
/ai.connector
/api.gateway
/control
/core
/data_import
/emng_client
/emng_log
/neural_networks
```

Необходимо добавить локального пользователя от чьего имени будут запускаться экземпляры программы

useradd -g ai -d /opt/aiee -s /usr/sbin/nologin ai

выдаем права

chown -R ai:ai /opt/aiee

и разместить jar файлы в соответствующие каталоги

Для работы программы должны быть развёрнуты следующие сервисы:

1. Api-gateway. К нему поступают запросы из вне (api-gateway-0.1.0.jar);
2. Core. Он используется для работы с core базой (core-0.1.0.jar);
3. Ai-connector. Он используется для прогнозирования данных (ai-connector-0.1.0.jar);

4. Client. Он используется для работы с client базой. Для каждого клиента свой сервис и своя база данных (client-0.1.0.jar);
5. Control. Контрольный сервис, он управляет всеми выполняемыми задачами (control-0.3.0.jar);
6. Data import. Сервис импорта данных из внешних источников (data-import-0.0.1.jar);
7. Log. Сервис логирования, собирает данные о действиях и активности пользователей (log-0.3.0.jar).

Для разворачивания вышеуказанных сервисов рядом с .jar файлом сервиса разместить файл application.properties, в котором прописать:

1. Api-gateway:

server.port= //порт, на котором работает сервис (прим. 9002)
grpc.server.port= //порт, по которому происходит взаимодейсвие по grpc между сервисами (прим. 9012)
aiConnector.host= //хост ai-connector сервиса (прим. localhost)
aiConnector.port= //grpc порт (grpc.server.port) ai-connector сервиса
coreDb.host= //хост core сервиса (прим. localhost)
coreDb.port= //grpc порт (grpc.server.port) core сервиса
log-service.host=//Адрес Сервиса логирования (Log) (прим. localhost)
log-service.port=// Порт Сервиса логирования (Log) (прим. 7910)

2. Core:

spring.datasource.url=//url core базы (прим. jdbc:postgresql://127.0.0.1:5432/emng_core_stage)
spring.datasource.hikari.schema=//схему базы (прим. emng)
spring.datasource.username=//username пльзователя (прим. emng_user)
spring.datasource.password=//password пользователя

grpc.server.port= //порт, по которому происходит взаимодейсвие по grpc между сервисами (прим. 9010)
server.port=//порт, на котором работает сервис (прим. 9000)
crm.auth=//адрес сервиса авторизации (прим. http://127.0.0.1:44345)
control-service.host=//Адрес Контрольного сервиса (Control) (прим. localhost)
control-service.port=// Порт Контрольного сервиса (Control) (прим. 9041)
log-service.host=//Адрес Сервиса логирования (Log) (прим. localhost)

log-service.port=// Порт Сервиса логирования (Log) (прим. 7910)

3. Ai-connector:

server.port= //порт, на котором работает сервис (прим. 9001)

grpc.server.port= //порт, по которому происходит взаимодейсвие по grpc между сервисами (прим. 9011)

coreDb.host= //хост core сервиса (прим. localhost)

coreDb.port= //порт (grpc.server.port) core сервиса

model.path= //путь к моделям по часам пик (прим. /opt/aiee/neural_networks/), модели взять из каталога /aiee/neural_networks/ дистрибутива

consumption.url= //адрес модели с прогнозированием потребления (прим. http://127.0.0.1:8003/consumption/)

price.url= //адрес модели с прогнозированием цены (прим. http://127.0.0.1:8002/price/)

log-service.host=//Адрес Сервиса логирования (Log) (прим. localhost)

log-service.port=// Порт Сервиса логирования (Log) (прим. 7910)

4. Client:

server.port= //порт, на котором работает сервис (прим. 9003)

grpc.server.port= //порт, по которому происходит взаимодейсвие по grpc между сервисами (прим. 9013)

spring.datasource.url==//url базы данных клиентского сервиса (прим. jdbc:postgresql://127.0.0.1:5432/emng_client_stage)

spring.datasource.hikari.schema= //схему базы (прим. emng)

spring.datasource.username= //username пользователя (прим. emng_user)

spring.datasource.password= //password пользователя

coreDb.host= //хост core сервиса (прим. localhost)

coreDb.port= //порт (grpc.server.port) core сервиса

eldis.login=//логин для доступа к АИИС Элдис для данного клиента

eldis.pass=//пароль для доступа к АИИС Элдис для данного клиента

eldis.key=//ключ API для доступа к АИИС Элдис для данного клиента

eldis.host=//Адрес сервиса для доступа к АИИС Элдис (прим. localhost)

eldis.port=//Порт сервиса для доступа к АИИС Элдис (прим. 5000)

log-service.host=//Адрес Сервиса логирования (Log) (прим. localhost)

log-service.port=// Порт Сервиса логирования (Log) (прим. 7910)

5. Control:

server.port= //порт, на котором работает сервис (прим. 9031)

grpc.server.port= //порт, по которому происходит взаимодейсвие по grpc между сервисами (прим. 9041)

spring.datasource.url= //url базы данных Контрольного сервиса (прим. jdbc:postgresql://127.0.0.1:5432/emng_control_stage)

spring.datasource.hikari.schema= //схему базы (прим. emng)

spring.datasource.username= //username пользователя (прим. emng_user)

spring.datasource.password= //password пользователя

coreDb.host= //хост core сервиса (прим. localhost)

coreDb.port= //порт (grpc.server.port) core сервиса

log-service.host=//Адрес Сервиса логирования (Log) (прим. localhost)

log-service.port=// Порт Сервиса логирования (Log) (прим. 7910)

6. Log:

server.port= //порт, на котором работает сервис (прим. 7900)

grpc.server.port= //порт, по которому происходит взаимодейсвие по grpc между сервисами (прим. 7910)

spring.datasource.url= //url базы данных сервиса логирования (прим. jdbc:postgresql://127.0.0.1:5432/emng_log_stage)

spring.datasource.hikari.schema= //схему базы (прим. emng)

spring.datasource.username= //username пользователя (прим. emng_user)

spring.datasource.password= //password пользователя

coreDb.host= //хост core сервиса (прим. localhost)

coreDb.port= //порт (grpc.server.port) core сервиса

7. Data import:

server.port= //порт, на котором работает сервис (прим. 9030)

spring.datasource.url= //url базы данных сервиса логирования (прим. jdbc:postgresql://127.0.0.1:5432/emng_core_stage)

spring.datasource.hikari.schema= //схему базы (прим. emng)

spring.datasource.username= //username пользователя (прим. emng_user)

spring.datasource.password= //password пользователя

server.servlet.context-path=/data-import

3.3.5 Настройка запуска сервисов в режиме системных служб

Для настройки запуска сервисов в режиме служб необходимо перейти в каталог со службами

cd /etc/systemd/system/

и разместить туда файлы конфигурации, имеющие следующий вид:

```
[Unit]
Description=имя службы
After=syslog.target network.target

[Service]
Type=simple
User=ai
Group=ai

ExecStart=/usr/bin/java -Xms512m -Xmx2g -файл для запуска с расширением.jar
Restart=on-failure
RestartSec=100
WorkingDirectory=/opt/aiee/путь до фала запуска
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target
```

НАПРИМЕР

```
[Unit]
Description=ai.client-service
After=syslog.target network.target

[Service]
Type=simple
User=ai
Group=ai

ExecStart=/usr/bin/java -Xms512m -Xmx2g -jar client-0.3.0.jar
Restart=on-failure
RestartSec=100
WorkingDirectory=/opt/aiee/emng_client
```

SuccessExitStatus=143

[Install]
WantedBy=multi-user.target

Добавление служб в автозапуск:

systemctl daemon-reload
systemctl enable ai_core.service
systemctl enable ai.client.service
systemctl enable ai.connector.service
systemctl enable ai.gateway.service
systemctl enable ai.logger.service
systemctl enable dataimport.service

Запуск служб:

systemctl start ai_core.service
systemctl start ai.gateway.service
systemctl start ai.connector.service
systemctl start dataimport.service

systemctl start ai.client.service
systemctl start ai.logger.service

Проверка статуса запущенных сервисов:

Напимер

systemctl status ai.logger.service

проверяет статус сервиса логирования
Вызывая аналогичную команду для других сервисов, можно проверить все остальные.

3.4 Запуск web-фронтенд приложений

3.4.1 Установка web-сервера nginx

```
apt install nginx
systemctl enable nginx
cd /etc/nginx/sites-available/
```

3.4.2 Конфигурация клиентского web-сервиса

```
sudo nano enmgm.bit76.ru
server {



listen 80;
server_name emng-stage.bit76.ru;
return 301 https://emng-stage.bit76.ru$request_uri;



}
server {

        listen 443 ssl;
        server_name emng-stage.bit76.ru;

        underscores_in_headers on;
        location / {
        root   /var/www/enmgm;
        index try_files  /index.html;
        try_files $uri /index.html;

        }
         location ^~ /api {
            proxy_pass http://127.0.0.1:9002/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-Scheme $scheme;
        proxy_set_header X-NginX-Proxy true;
        add_header Cache-Control 'no-store, no-cache, must-revalidate,
proxy-revalidate, max-age=0';

    }
    location ^~ /core {
        proxy_pass http://127.0.0.1:9000/core;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header                    X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header X-Scheme $scheme;
        proxy_set_header Host $http_host;
        proxy_set_header X-NginX-Proxy true;
        add_header Cache-Control 'no-store, no-cache, must-revalidate,
proxy-revalidate, max-age=0';

    }

    location ^~ /connect {
        proxy_pass http://127.0.0.1:44345/connect;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header                    X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header X-Scheme $scheme;
        proxy_set_header Host $http_host;
        proxy_set_header X-NginX-Proxy true;
        add_header Cache-Control 'no-store, no-cache, must-revalidate,
proxy-revalidate, max-age=0';
        client_max_body_size 128m;

    }
    location ^~ /data-import {
        proxy_pass http://127.0.0.1:9030/data-import;
        proxy_set_header X-Real-IP $remote_addr;
```

```
        proxy_set_header                                X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header X-Scheme $scheme;
        proxy_set_header Host $http_host;
        proxy_set_header X-NginX-Proxy true;
        add_header Cache-Control 'no-store, no-cache, must-revalidate,
proxy-revalidate, max-age=0';
        client_max_body_size 128m;

    }
}
```

3.4.3 Конфигурация web-сервиса администрирования

```
sudo nano admemng-stage.bit76.ru
server {
listen 80;
server_name admemng-stage.bit76.ru;
return 301 https://admemng-stage.bit76.ru$request_uri;


}

server {

    listen 443 ssl;


    server_name admemng-stage;

    underscores_in_headers on;
    location / {

    root   /var/www/admemgm;
    index try_files  /index.html;
    try_files $uri /index.html;
```

```
        }
         location ^~ /api {
            proxy_pass http://127.0.0.1:9002/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-Scheme $scheme;
            proxy_set_header X-NginX-Proxy true;
            add_header Cache-Control 'no-store, no-cache, must-revalidate,
proxy-revalidate, max-age=0';

        }
        location ^~ /core {
            proxy_pass http://127.0.0.0:9000/core;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header                                X-Forwarded-For
$proxy_add_x_forwarded_for;
            proxy_set_header X-Scheme $scheme;
            proxy_set_header Host $http_host;
            proxy_set_header X-NginX-Proxy true;
            add_header Cache-Control 'no-store, no-cache, must-revalidate,
proxy-revalidate, max-age=0';

        }

        location ^~ /connect {
            proxy_pass http://127.0.0.1:44345/connect;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header                                X-Forwarded-For
$proxy_add_x_forwarded_for;
            proxy_set_header X-Scheme $scheme;
            proxy_set_header Host $http_host;
            proxy_set_header X-NginX-Proxy true;
            add_header Cache-Control 'no-store, no-cache, must-revalidate,
proxy-revalidate, max-age=0';
            client_max_body_size 128m;
```

```
    }
    location ^~ /data-import {
        proxy_pass http://127.0.0.1:9030/data-import;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Scheme $scheme;
        proxy_set_header Host $http_host;
        proxy_set_header X-NginX-Proxy true;
        add_header Cache-Control 'no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0';
        client_max_body_size 128m;

    }

}
```

3.4.4 Активация установленных конфигураций

Для добавления ссылок на созданные файлы для включения в активные необходимо выполнить следующие команды:

```
sudo ln -s /etc/nginx/sites-available/admemng-stage.bit76.ru /etc/nginx/sites-enabled/
sudo ln -s /etc/nginx/sites-available/emng-stage.bit76.ru /etc/nginx/sites-enabled/
```

Размещаем файлы из каталога web в директорию /var/www/
После чего перезапускаем nginx

```
sudo systemctl restart nginx
```

3.5 Установка сервиса Авторизации

3.5.1 Установка dot.net runtime

wget https://dot.net/v1/dotnet-install.sh -O dotnet-install.sh

bash dotnet-install.sh --channel 6.0 --runtime aspnetcore

(без sudo - установится в текущего пользователя, с sudo - в рут, во время выполнения скрипта видно куда был распакован рантайм)

Если после установки скриптом, при выполнении dotnet пишет что не найдено, то можно, либо использовать абсолютный путь, либо прописать линк как то так (путь куда был распакован рантайм скорректировать), :

sudo ln -s /root/.dotnet/dotnet /usr/bin/dotnet

3.5.2 Развёртывание сервиса авторизации

Копируем дистрибутив сервиса авторизации например в каталог /home/forecast

Конфигурационный файл appsettings.json в каталоге например /home/forecast :

Чтоб указать с каким портом запустить, скорректируем "Urls": "http://localhost:44345").

Urls – список адресов, которые слушает сервер. Если всё на одном сервере и переадресовывается при обращении в рамках него же - можно оставить как в примере. Если отдельный сервер и надо слушать открытый порт для входящий , то "http://*:44345".

UserAdmin - логин пароль административного пользователя, который будет создан в базе.

PostgresConnect там 2 строки подключения, одна к DefaultDataBase - доступ к базе postgres возможно нужен для миграций,

DataBase и DataBaseName относятся к базе, в которой будут храниться данные относящиеся к серверу авторизации.

3.5.3 Настройка запуска сервиса авторизации в режиме системной службы

Необходимо перейти в каталог системных служб:

cd /etc/systemd/system/

Создаем файл конфигурация для службы авторизации

nano auth.service

Пример конфигурации:

```
[Unit]
Description=auth service .NET Web API Application running on Ubuntu

[Service]
WorkingDirectory=/home/forecast/auth
ExecStart=/usr/bin/dotnet /home/forecast/auth/WebServiceAuthorization.dll
Restart=always
RestartSec=30
SyslogIdentifier=dotnet-auth
User=root
Environment=ASPNETCORE_ENVIRONMENT=Production

[Install]
WantedBy=multi-user.target
```

```
systemctl daemon-reload
sudo systemctl enable auth.service
```

```
sudo systemctl start auth.service
sudo systemctl status auth.service
```

Проверка доступности порта сервиса авторизации

```
wget localhost:44345
```

3.5.4 Добавление сервиса авторизации в web сервер nginx

```
cd /etc/nginx/sites-available/
nano auth-stage.bit76.ru
```

Пример конфигурации:

```
server {

listen 80;

server_name auth-stage.bit76.ru;

return 301 https://auth-stage.bit76.ru$request_uri;

        location / {
                proxy_pass http://127.0.0.1:44345/;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                proxy_set_header Host auth-stage.bit76.ru;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header                                X-Forwarded-For
$proxy_add_x_forwarded_for;
                proxy_connect_timeout 600;
                proxy_send_timeout 600;
                proxy_read_timeout 600;
```

```
            send_timeout 600;
            client_max_body_size 4M;
        }

        location ^~ /connect {
            proxy_pass http://127.0.0.1:44345/connect;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Scheme $scheme;
            proxy_set_header Host $http_host;
            proxy_set_header X-NginX-Proxy true;
            add_header Cache-Control 'no-store, no-cache, must-revalidate,
proxy-revalidate, max-age=0';
            client_max_body_size 128m;
        }


}

server {

        listen 443 ssl;
        server_name auth-stage.bit76.ru;


        location / {
            proxy_pass http://127.0.0.1:44345/;
            proxy_ssl_server_name on;
            proxy_ssl_name auth-stage.bit76.ru;
            proxy_http_version 1.1;
            proxy_set_header Host $host:$server_port;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header                                X-Forwarded-For
$proxy_add_x_forwarded_for;
```

```
            proxy_connect_timeout       600;
            proxy_send_timeout          600;
            proxy_read_timeout          600;
            send_timeout                600;
            client_max_body_size        4M;
        }

        location ^~ /connect {
            proxy_pass http://127.0.0.1:44345/connect;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Scheme $scheme;
            proxy_set_header Host $http_host;
            proxy_set_header X-NginX-Proxy true;
            add_header Cache-Control 'no-store, no-cache, must-revalidate,
proxy-revalidate, max-age=0';
            client_max_body_size 128m;
        }
}
```

После установки конфигурации неообходимо перезапустить nginx

sudo systemctl restart nginx

3.5.5 Установка certbot для выпуска сертификата ssl

apt install certbot-python-nginx

Запуск процедуры выпуска сертификата
sudo certbot –nginx

Необходимо указать все адреса для которых будет выпущен сертификат, конфигурационные файлы certbot отредактирует автоматически.

После выпуска ssl-сертификатов необходим перезапуск nginx

sudo systemctl restart nginx

3.6 Запуск сервиса подкачки данных из системы Элдис

Необходимо скопировать файлы в новый каталог /home/forecast/GrpcEldisService

Запуск контейнера производится командой:

docker run -it -p 5000:80 -e TZ=Europe/Moscow -d --restart unless-stopped --name testgrpc testgrpc .

4 НАСТРОЙКА ПО «ЭНЕРГОБИТ»

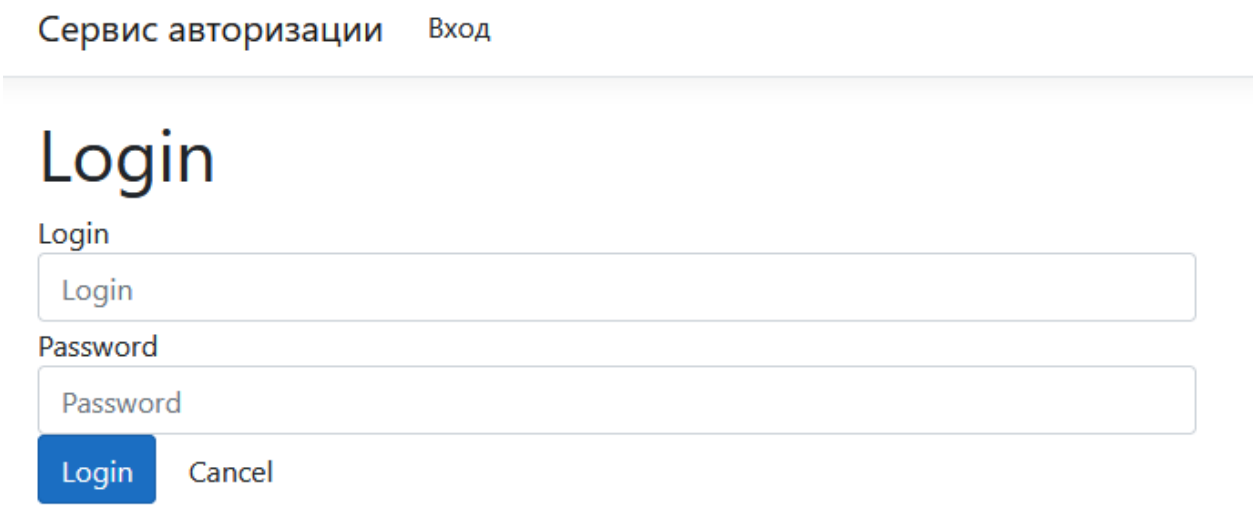
4.1 Создание пользователей и организаций в сервисе авторизации

Для входа в систему, пользователю требуется запросить у администратора системы имя и пароль пользователя, а также адрес приложения в сети интернет.

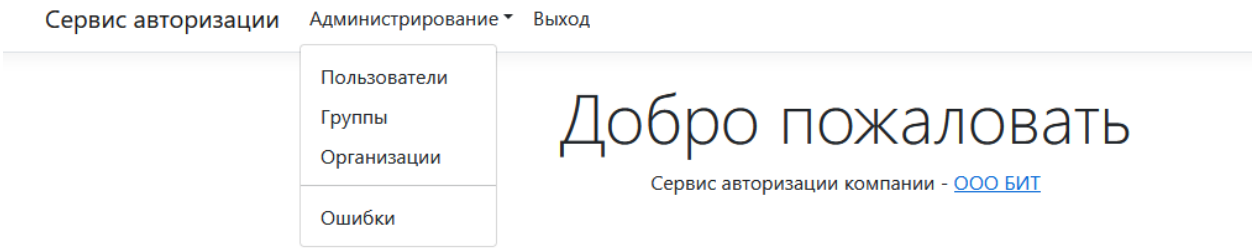
Сервис авторизации, в котором администратор системы может создать нового пользователя и выполнить другие первоначальные подготовки выглядит таким образом:

Для вызова модального окна авторизации требуется нажать на кнопку **Вход** в верхней части экрана.

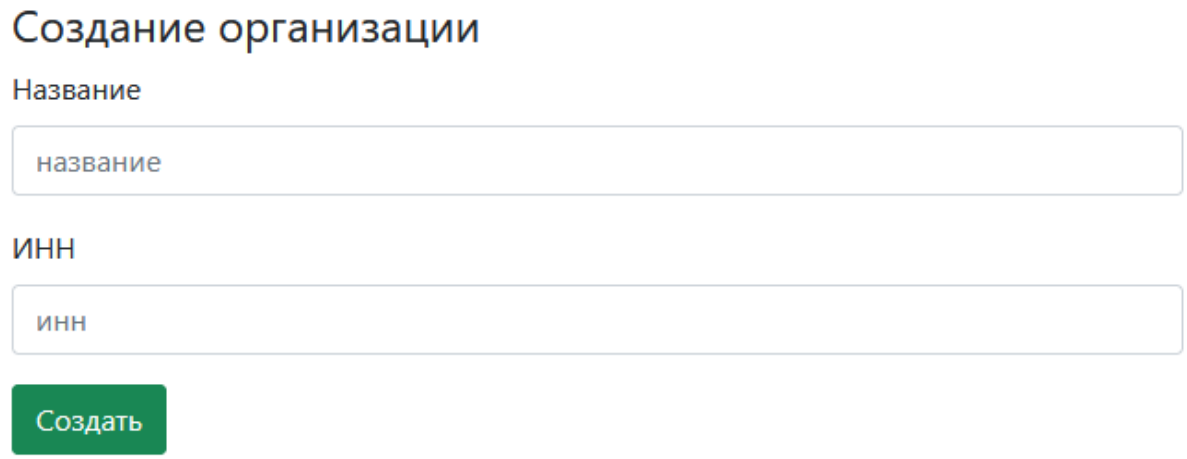
Окно авторизации в сервисе для администратора системы выглядит таким образом:



После авторизации, у администратора появляется возможность создания или изменения данных в разделах **Пользователи**, **Группы** и **Организации**:



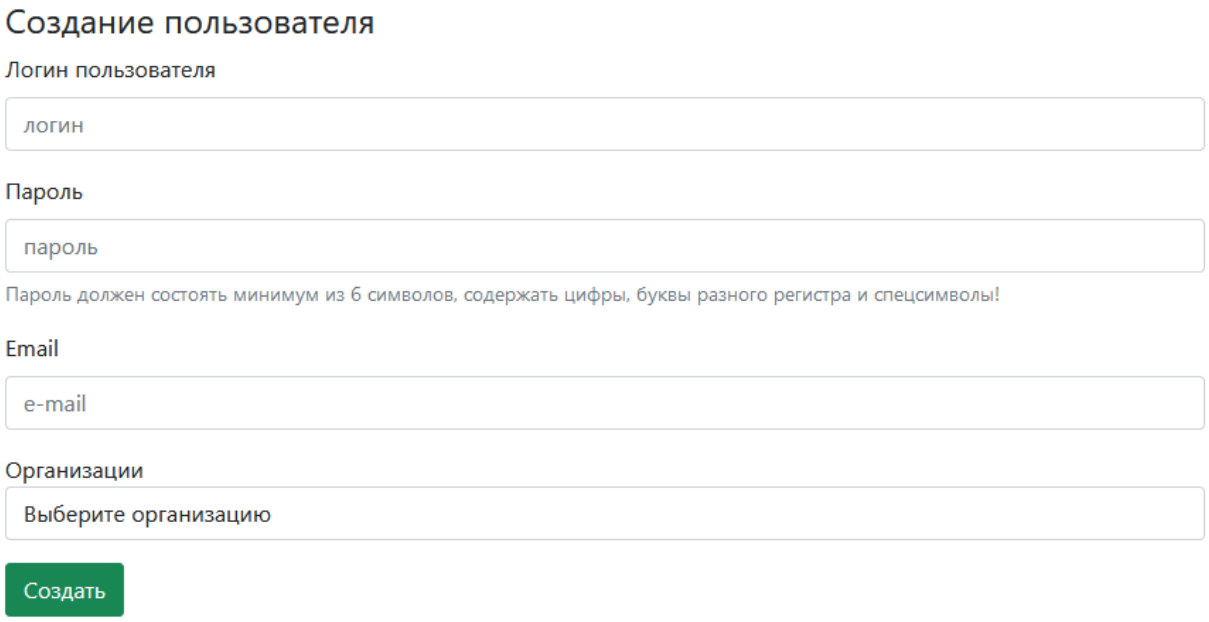
При первоначальной настройке требуется создать организацию, для этого в соответствующем разделе требуется нажать на кнопку **Добавить**. После чего, ввести название новой организации, её ИНН и нажать кнопку **Создать**.



После этого в разделе **Организации** станет доступна запись созданной организации.

Затем, для создания нового пользователя, требуется в разделе **Пользователи** нажать кнопку **Добавить**.

В окне создания записи о новом пользователе требуется ввести его данные - связку из логина и пароля (с обязательным соблюдением указанных условий создания пароля), электронную почту пользователя. Далее требуется добавить связанную с пользователем организацию и нажать кнопку **Создать**.



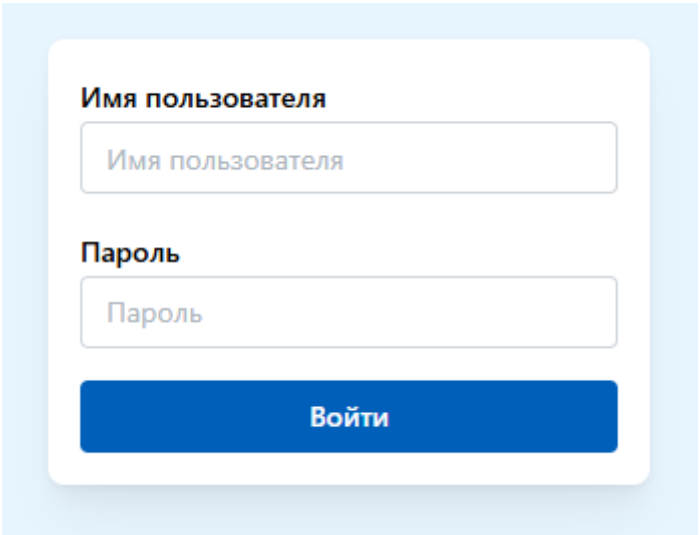
После этого в разделе **Пользователи** станет доступна запись созданного пользователя.

В случае необходимости, можно создать группу для объединения нескольких пользователей.

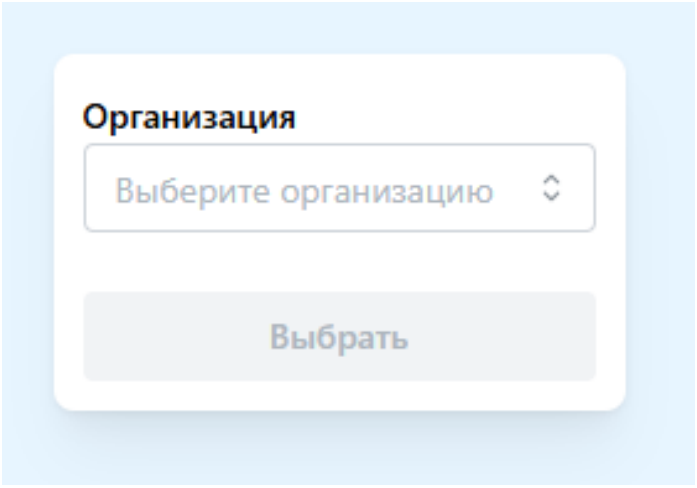


4.2 Создание пользователей и организаций в сервисе администрирования

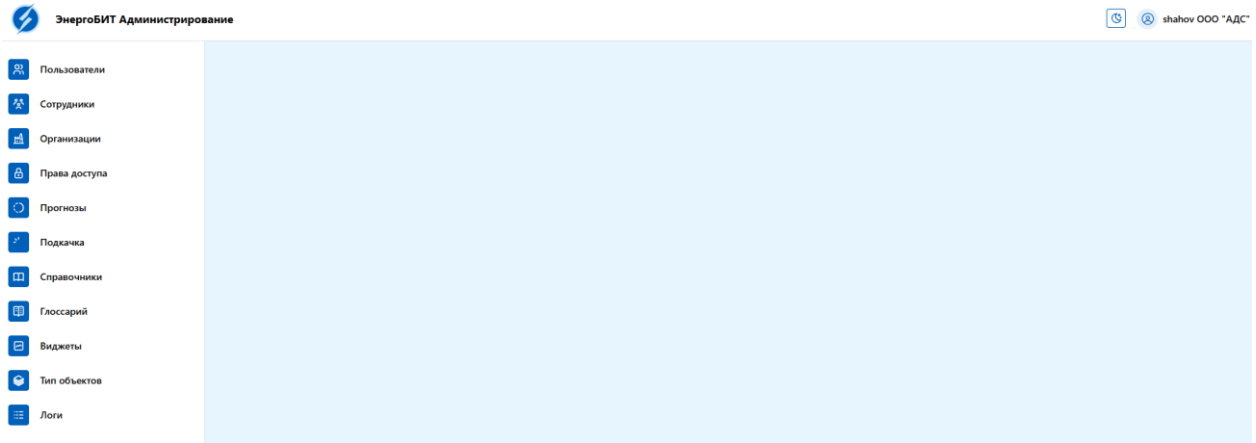
Для авторизации в сервисе администрирования, администратору системы требуется перейти по адресу веб-сервиса. Первоначальный экран содержит форму ввода связки из логина и пароля:



После первичной авторизации требуется выбрать организацию из списка доступных организаций:



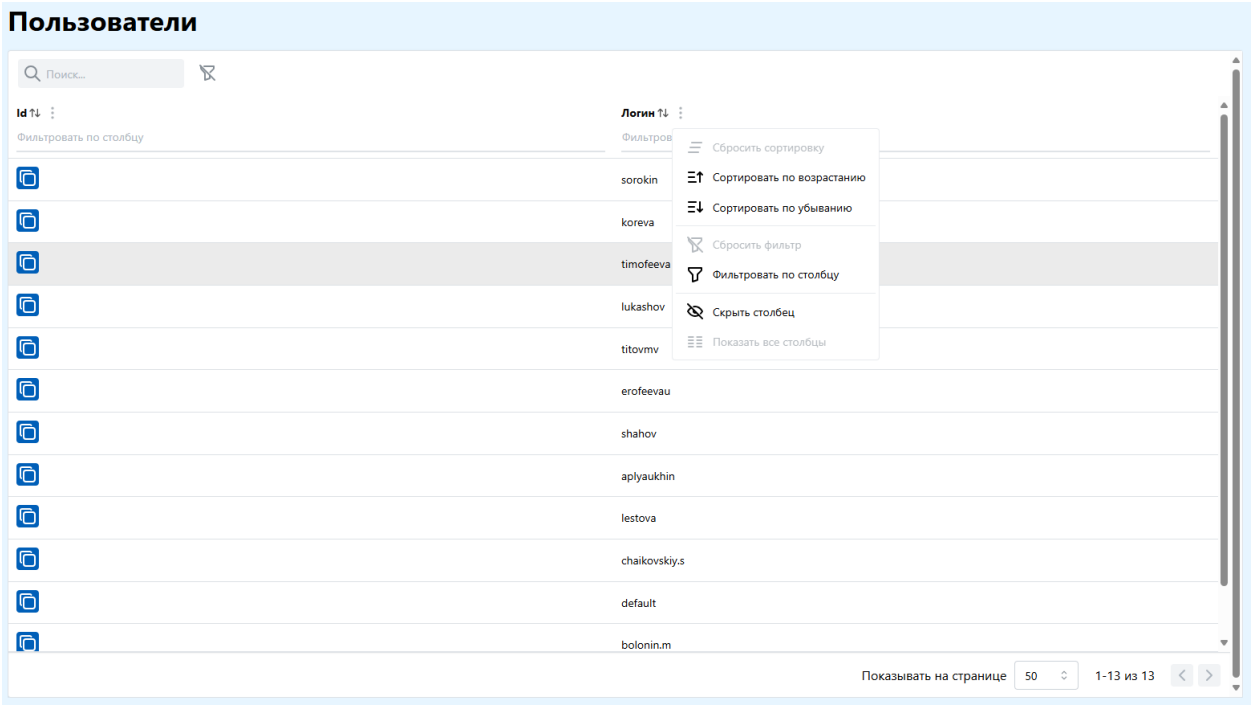
После этого становится доступной основная часть веб-сервиса администрирования системы ЭнергоБИТ:



4.3 Сервис администрирования

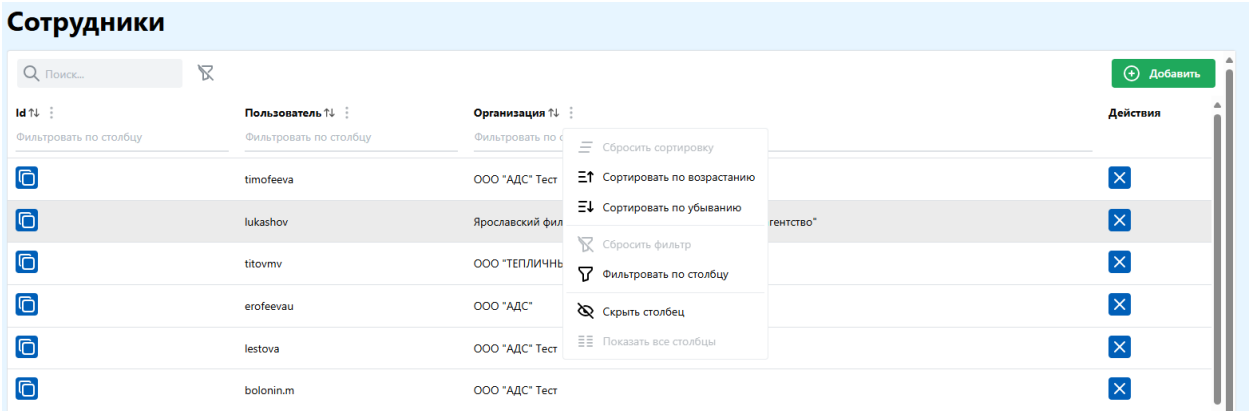
Сервис администрирования включает в себя следующие разделы:

4.3.1 Пользователи



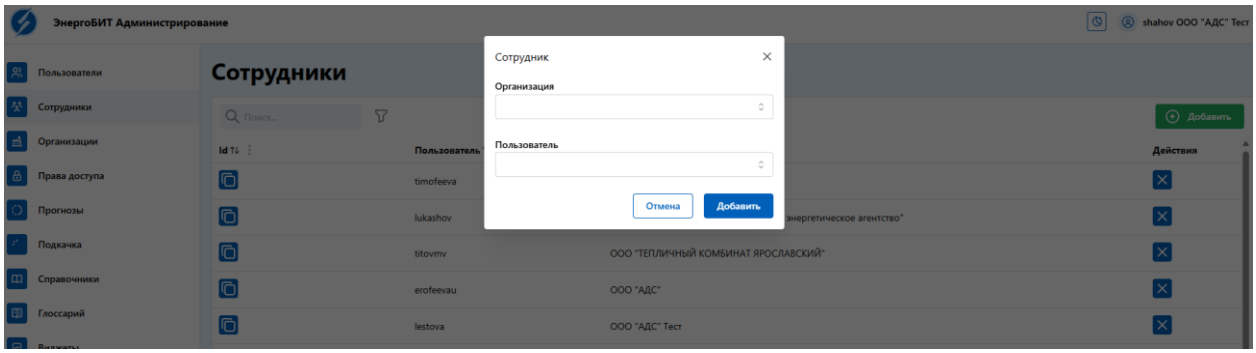
В данном разделе отображается список созданных в сервисе авторизации пользователей с возможностью сортировки их отображения и поиска конкретного пользователя.

4.3.2 Сотрудники

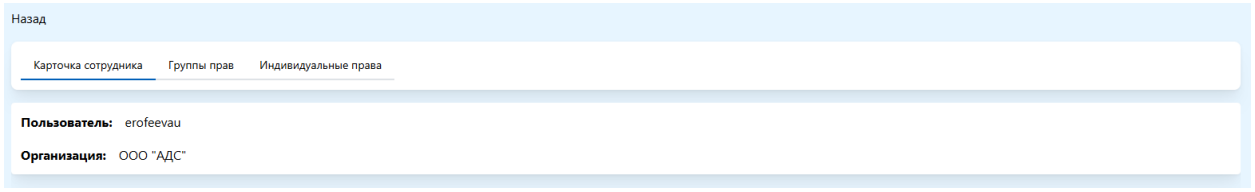


В разделе отображаются связанные с пользователями сотрудники. В разделе также есть возможность сортировки их отображения и поиска конкретного сотрудника. Также можно создать нового сотрудника или удалить имеющегося.

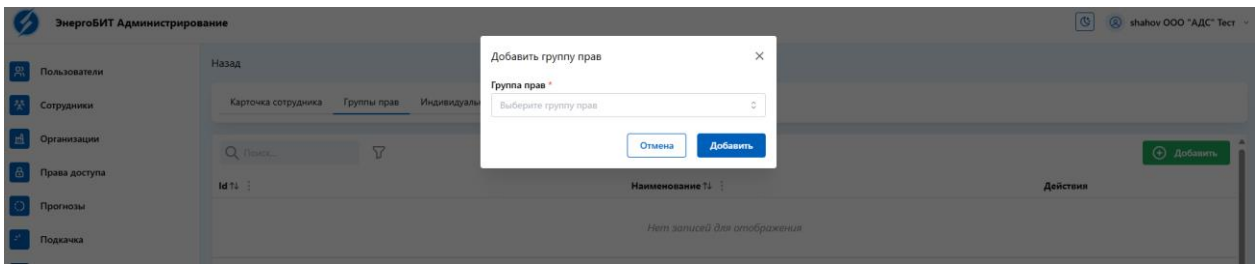
При нажатии кнопки **Добавить** открывается модальное окно для добавления нового сотрудника:



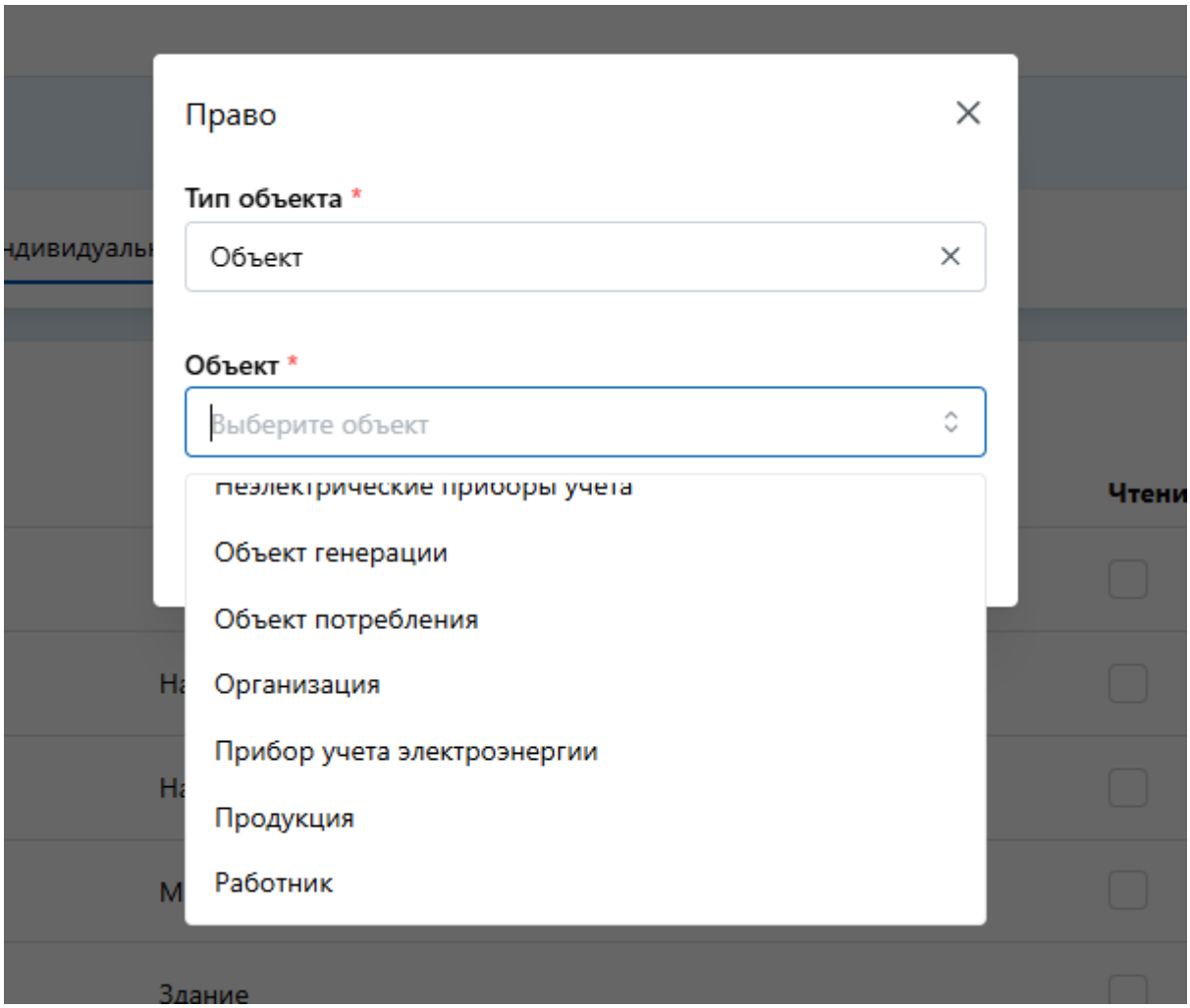
В карточке сотрудника доступны для просмотра данные на дополнительных вкладках.



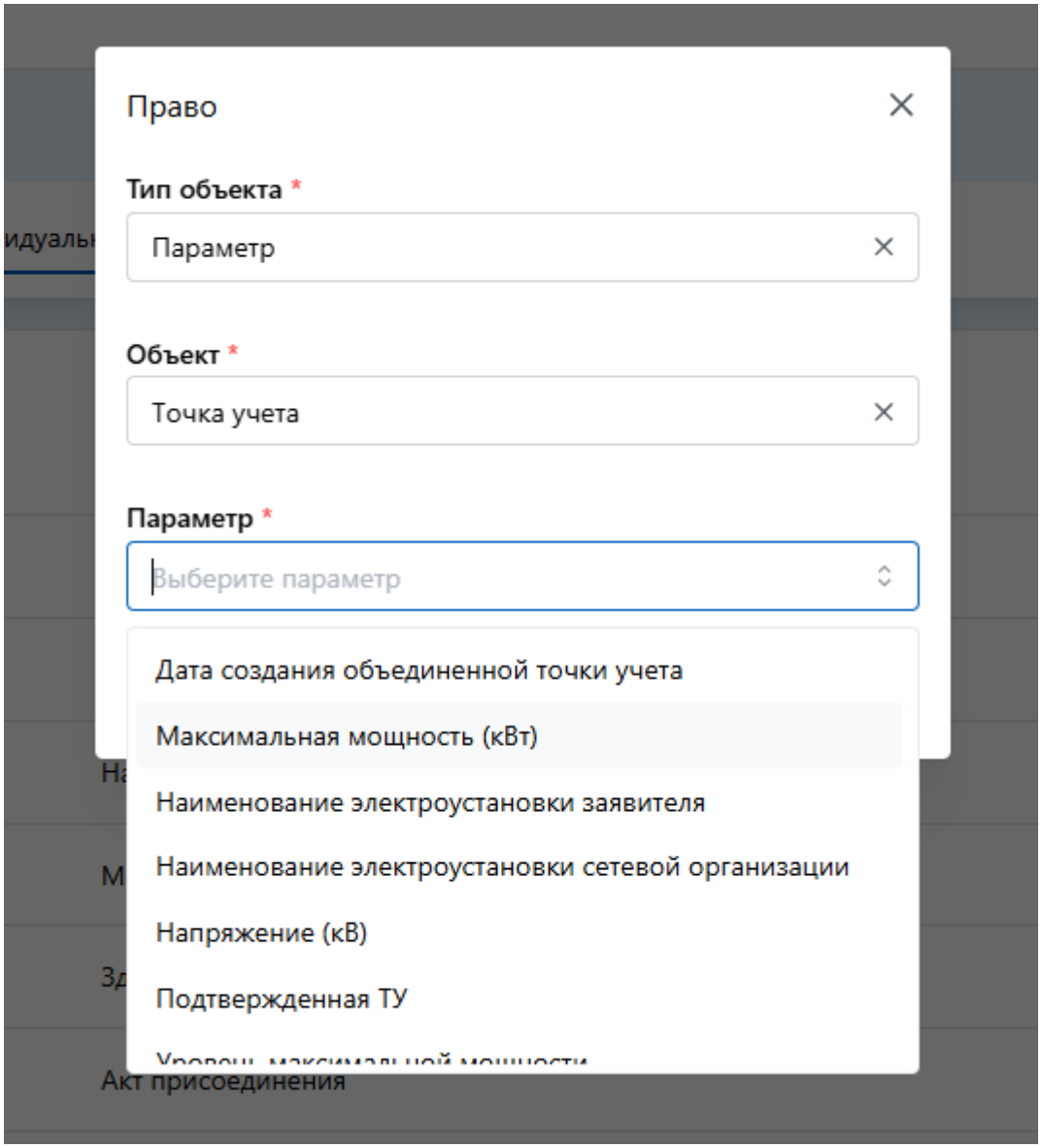
Сотрудник может быть добавлен в группу прав:



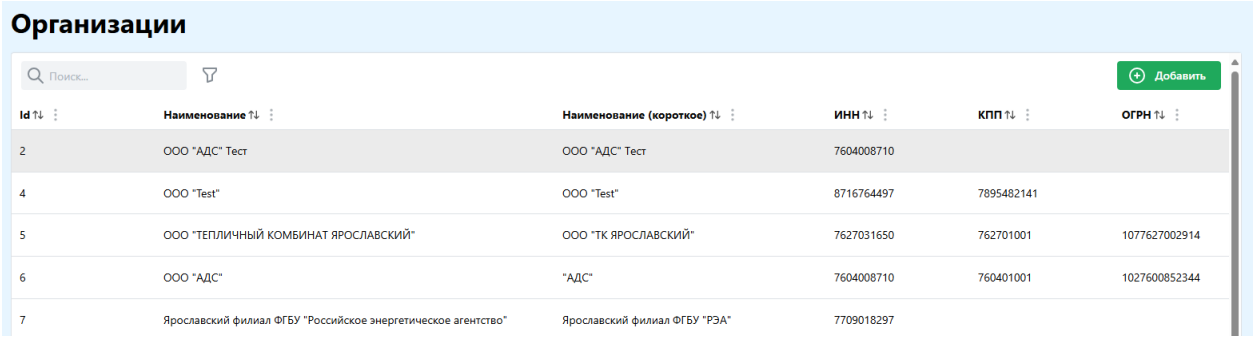
На вкладке индивидуальных прав сотруднику могут быть выданы права на объекты:



Или отдельные параметры:

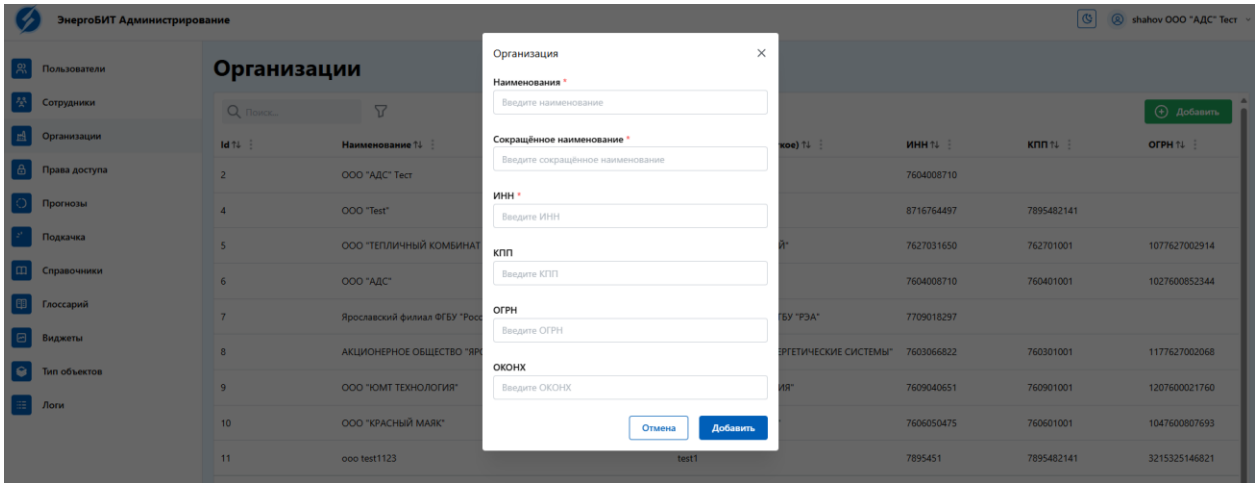


4.3.3 **Организации**



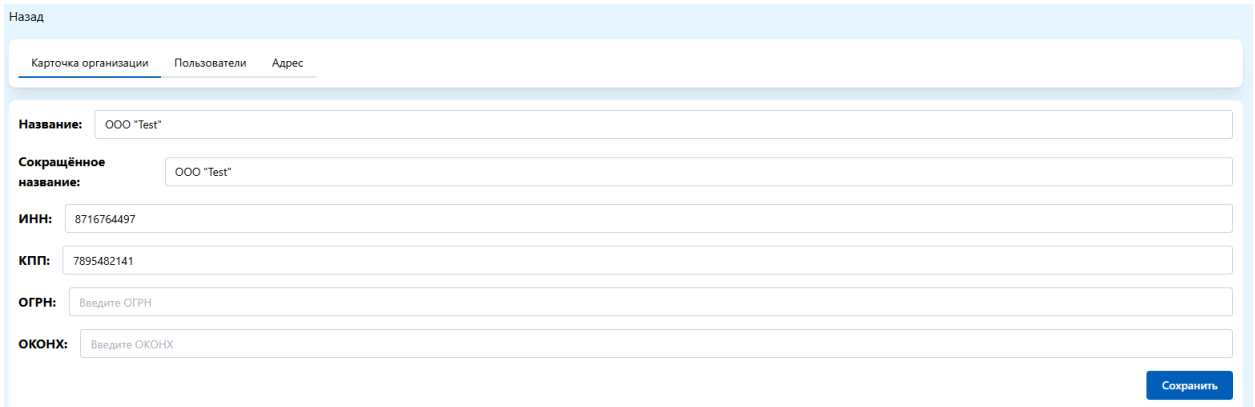
В разделе отображаются созданные организации. В разделе есть возможность сортировки их отображения и поиска конкретной организации. Также можно создать новую организацию или удалить имеющуюся.

При нажатии кнопки **Добавить** открывается модальное окно для добавления новой организации:

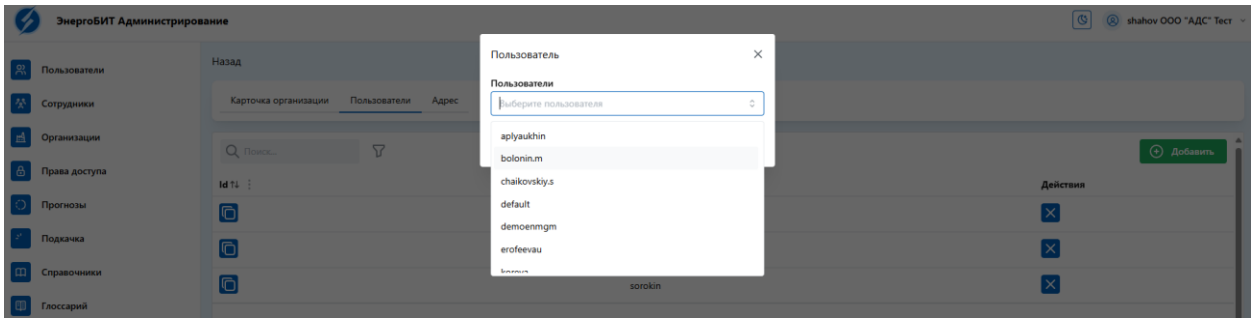


В нём необходимо внести данные организации. Есть и обязательные данные, без которых создание новой организации невозможно. Они отмечены звёздочкой - Наименование, Сокращённое наименование и ИНН.

В карточке организации есть возможность как скорректировать добавленные ранее данные по организации:

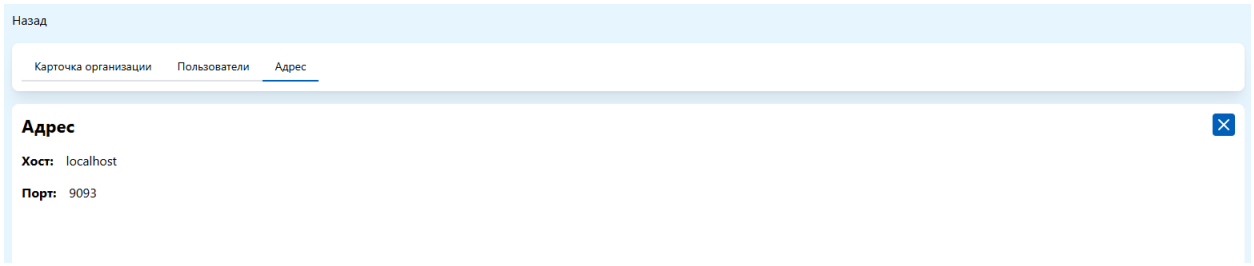


Так и добавить пользователей, связанных с организацией:

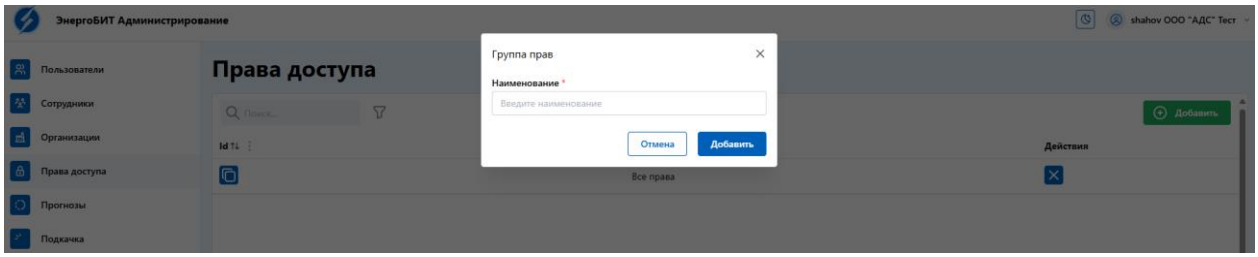


В разделе есть возможность сортировки отображения добавленных пользователей и поиска по их списку. Также можно удалить связь организации с пользователем.

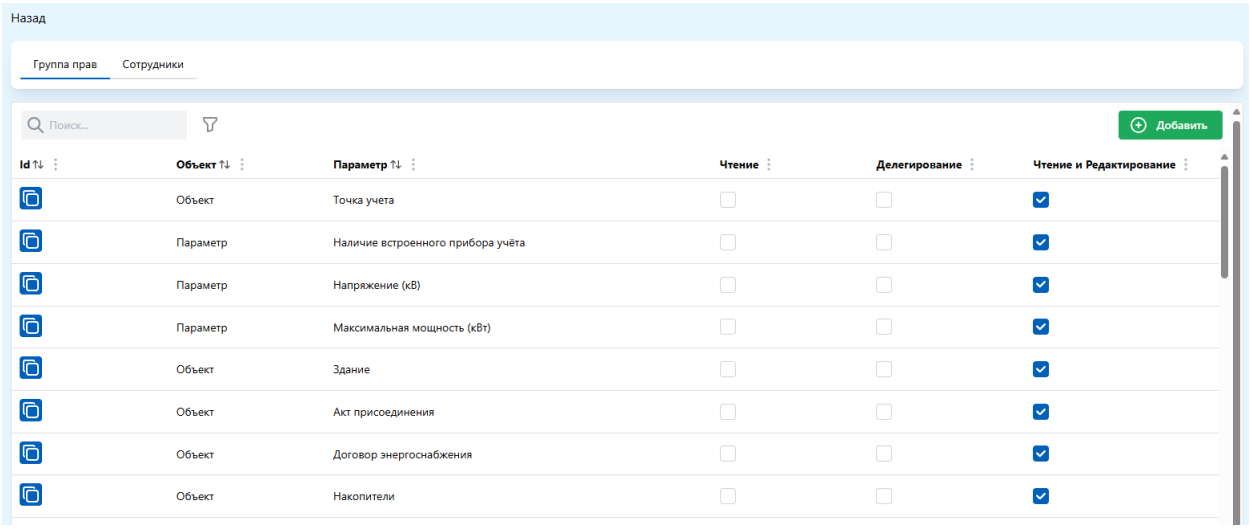
На странице **Адреса** указана информация о хосте и соответствующем порте:



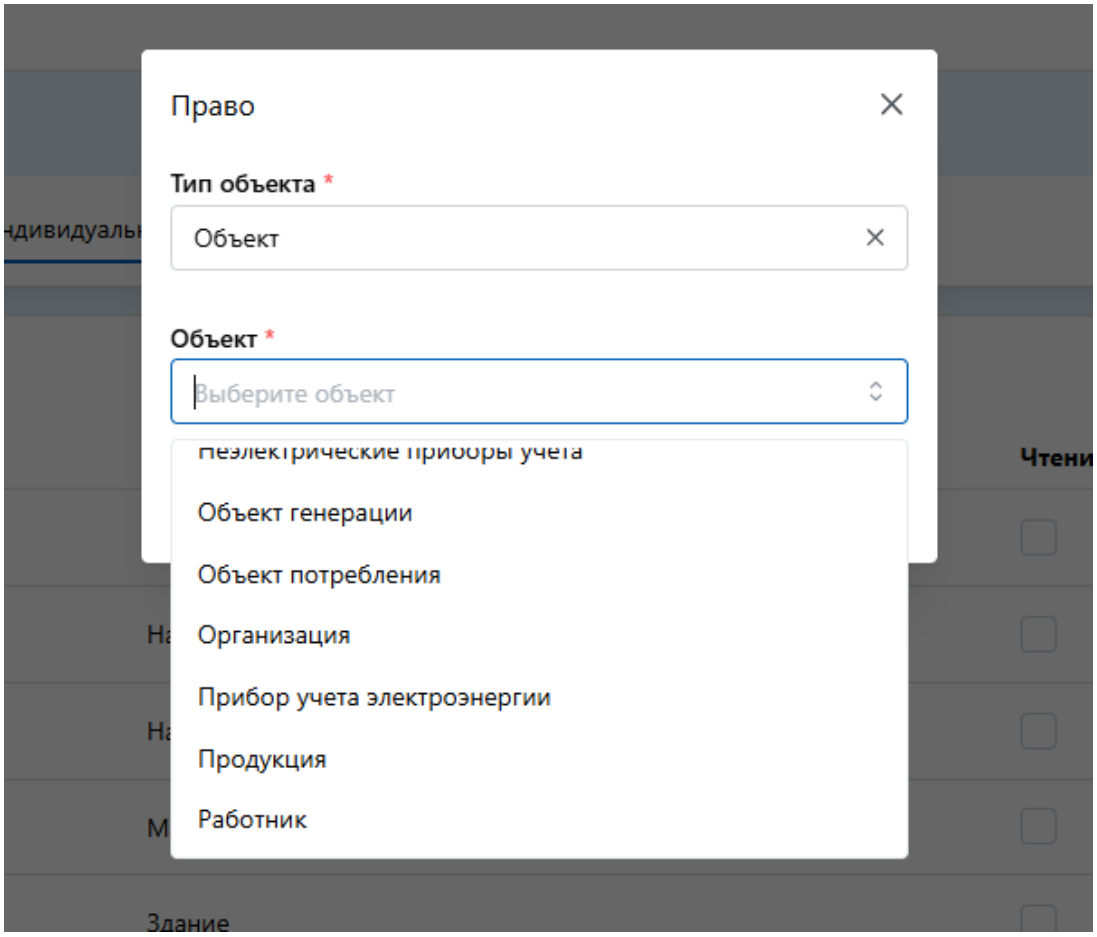
4.3.4 **Права доступа**



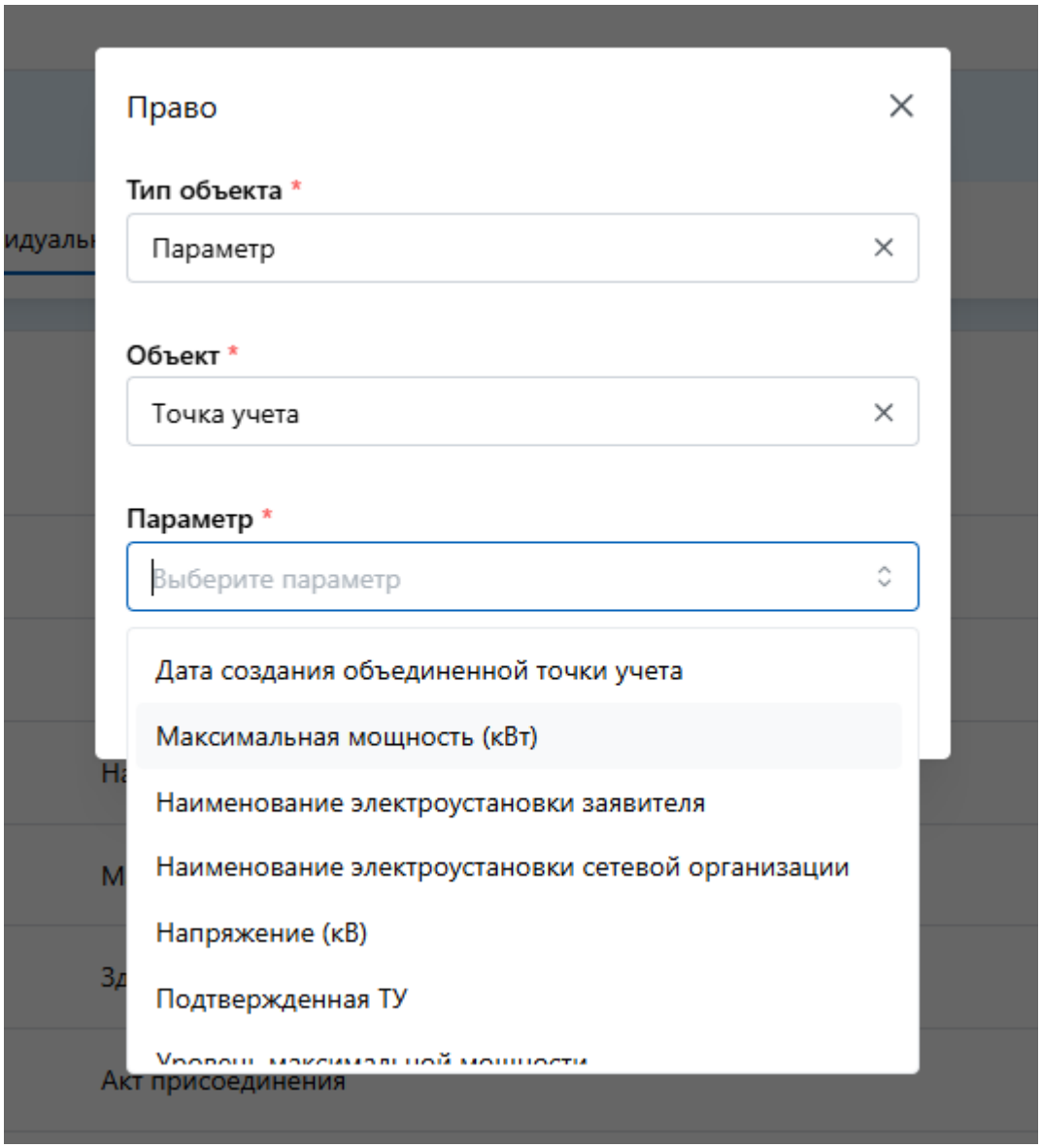
В данном разделе сначала создаются новые группы прав, и затем наполняются отдельными правами доступа:



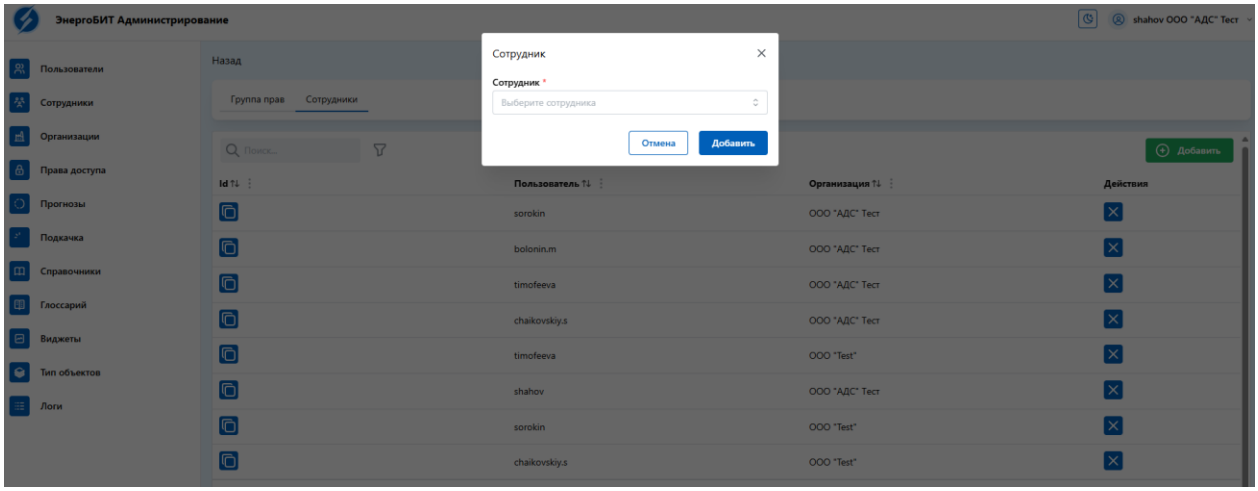
При нажатии на кнопку **Добавить**, могут быть добавлены в группу права на объекты:



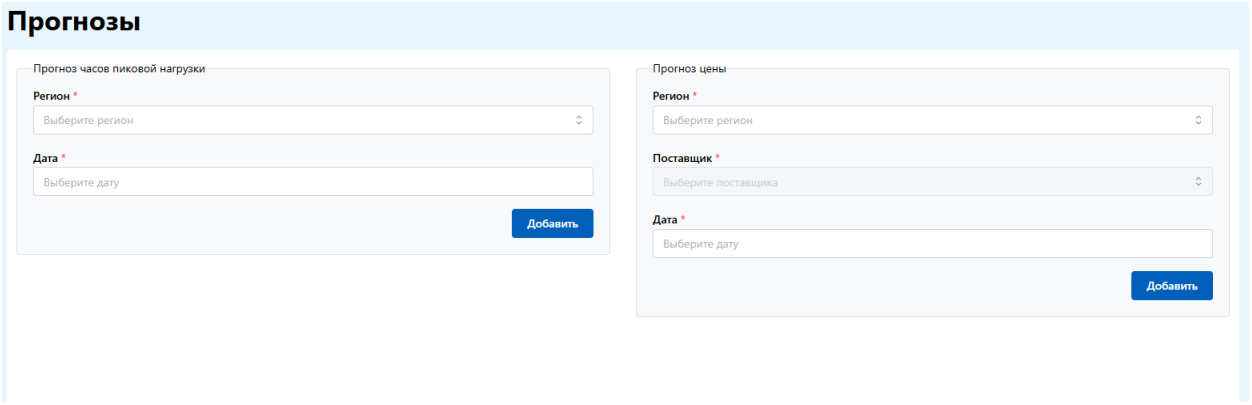
Или отдельные параметры:



На вкладке **Сотрудники** можно добавить конкретных сотрудников, на которых будут распространяться выданные в группе права:

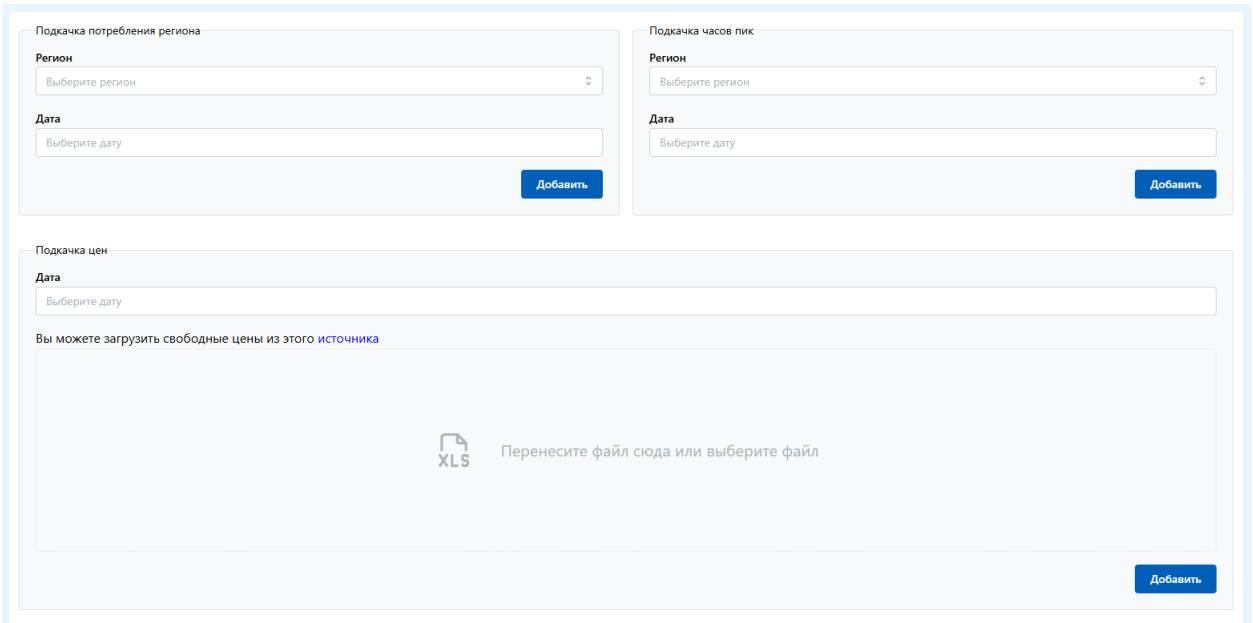


4.3.5 **Прогнозы**



В разделе **Прогнозы** могут быть сформированы прогнозы часов пиковой нагрузки, а также прогнозы цены. Для этого требуется заполнить обязательные поля (осуществляется выбором соответствующих значений из списка) и нажать на кнопку **Добавить**.

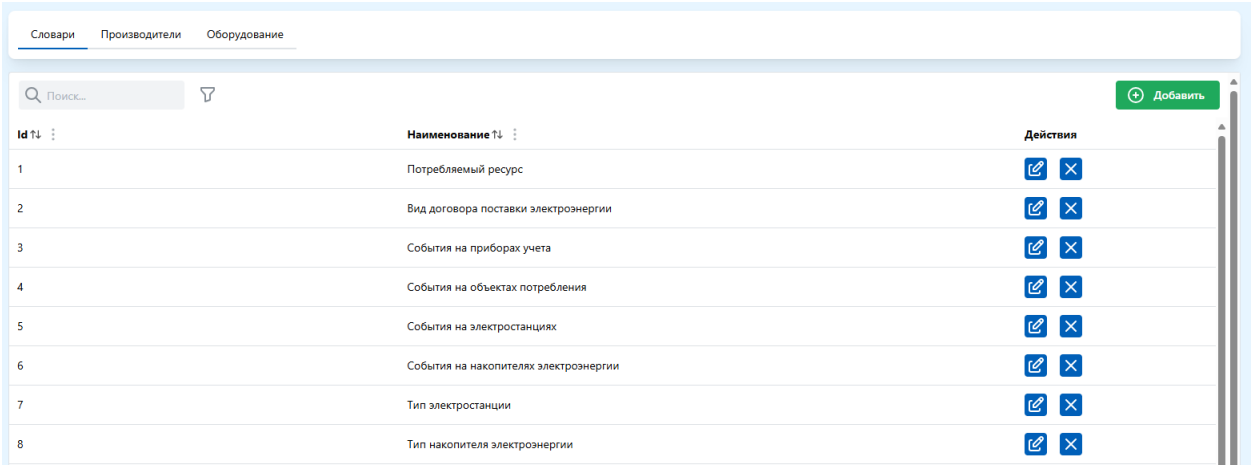
4.3.6 **Подкачка**



На странице раздела **Подкачка** возможна подкачка данных о потреблении региона, подкачка данных о часах пик и ценах на электроэнергию.

4.3.7 **Справочники**

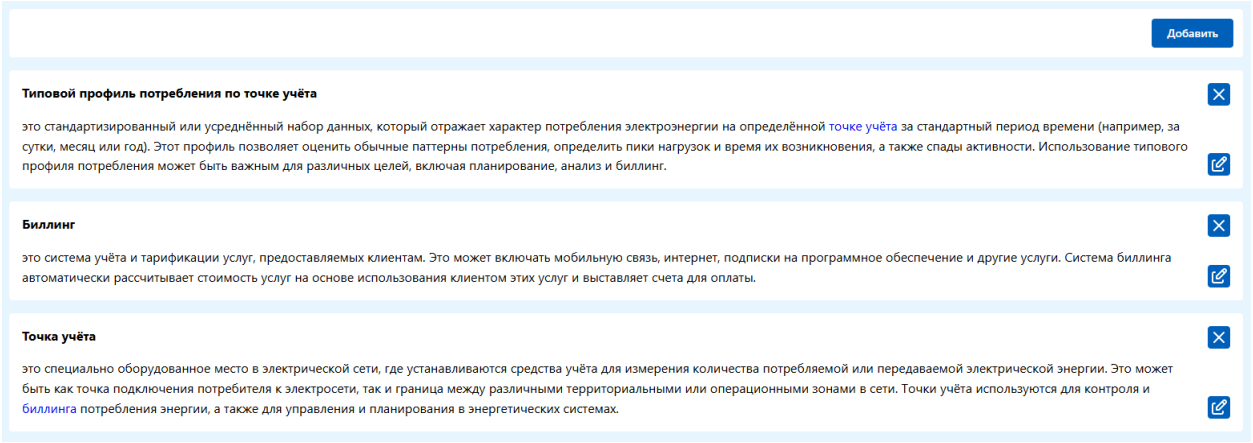
В данном разделе собраны все справочники системы:



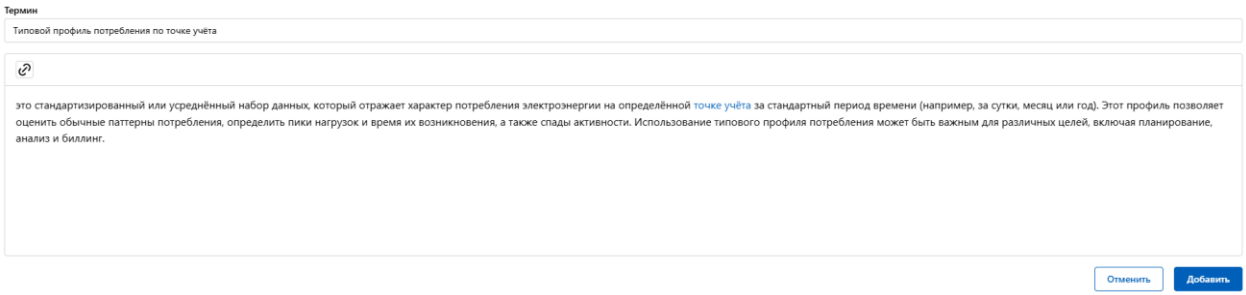
Справочники могут быть как добавлены (нажатие на кнопку **Добавить**), так и отредактированы (кнопка с иконкой карандаша в строке со справочником).

4.3.8 **Глоссарий**

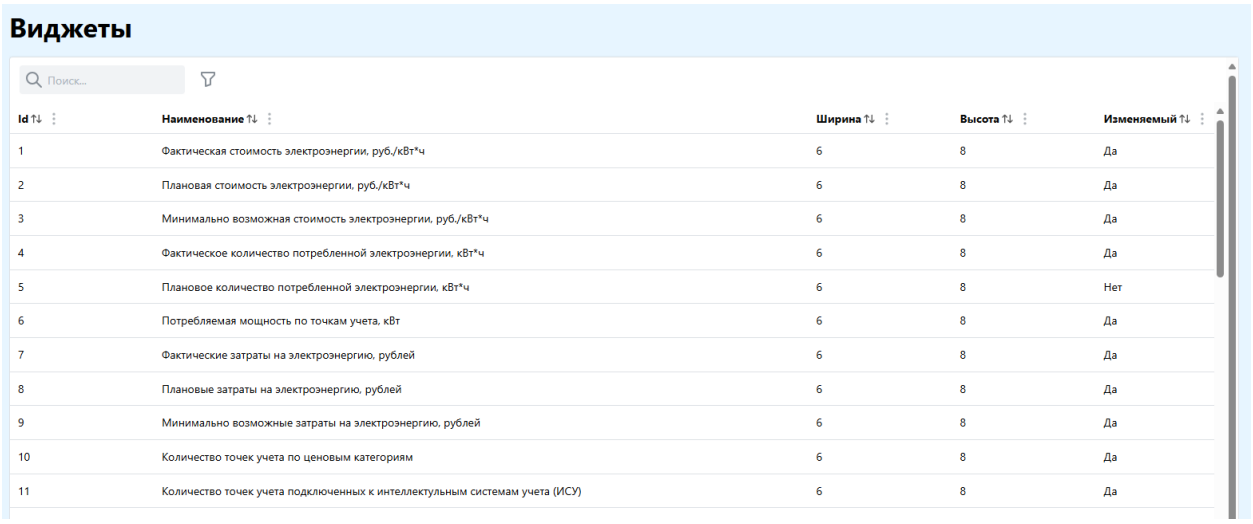
В разделе Глоссарий собираются специальные термины и определения, относящиеся к использованию программы.



Могут быть как добавлены новые, так и отредактированы добавленные:



4.3.9 **Виджеты**



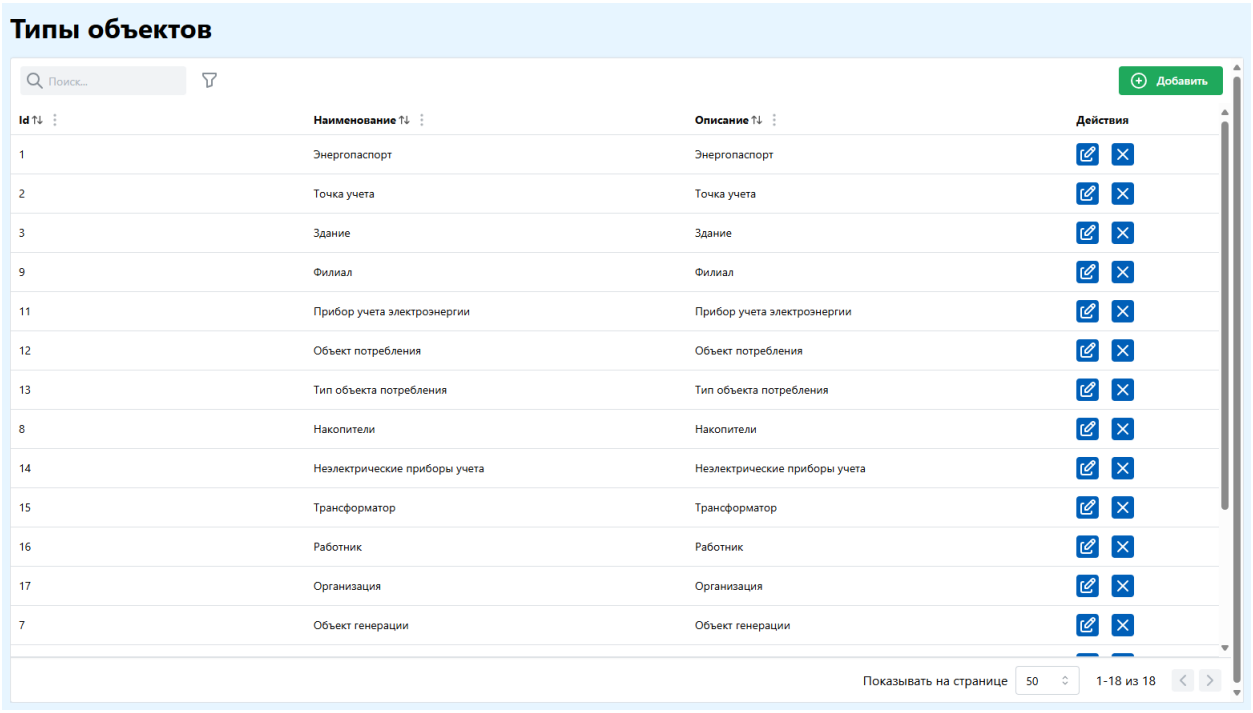
В разделе отображается перечень добавленных виджетов в виде списка.

При переходе к содержимому карточки отдельного виджета, можно отредактировать его характеристики:



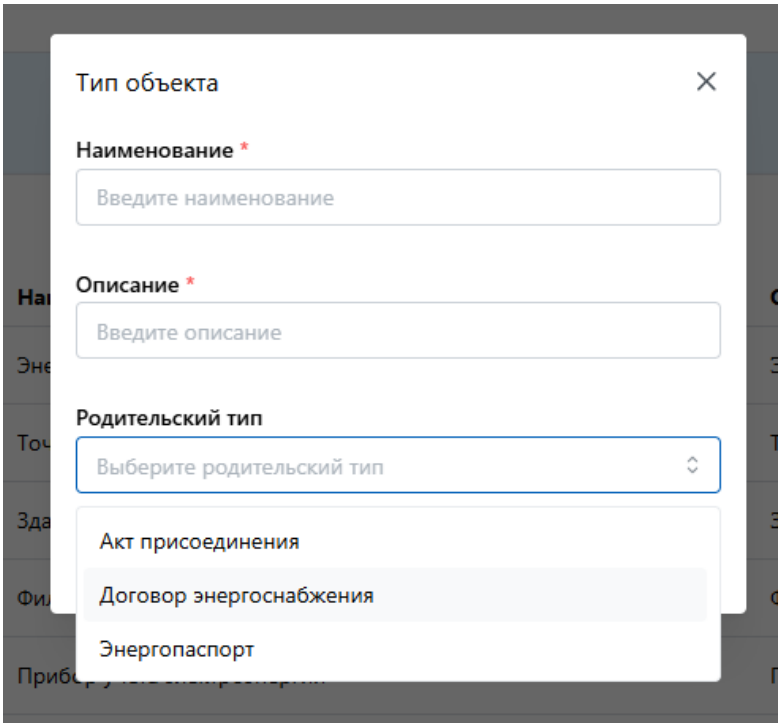
Здесь меняются высота и ширина виджета, указывается возможность его масштабирования в интерфейсе программы, а также выбирается связанный с ним термин.

4.3.10 **Тип объектов**



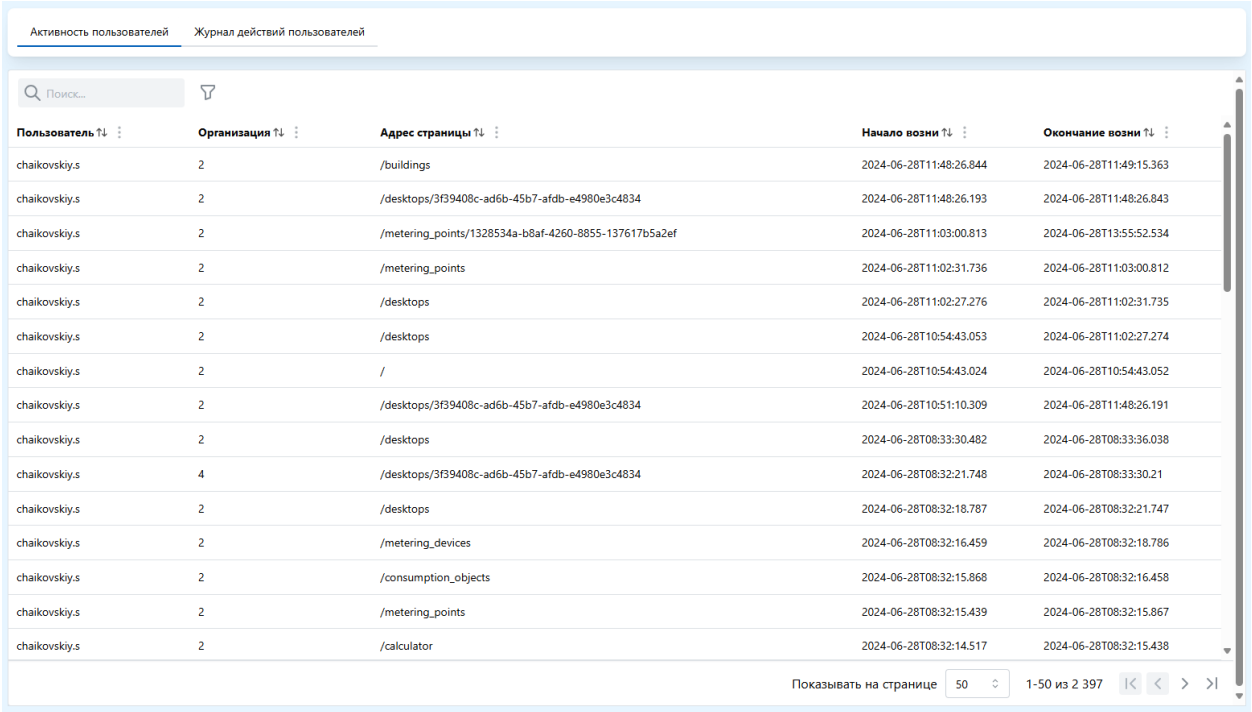
В данном перечне могут быть добавлены (нажатие на кнопку **Добавить**) или отредактированы (кнопка с иконкой карандаша в строке со справочником) различные типы объектов. Нажатие на кнопку с крестиком в строке типа объекта приведёт к его удалению.

При добавлении или редактировании типа объекта доступны для заполнения следующие параметры:

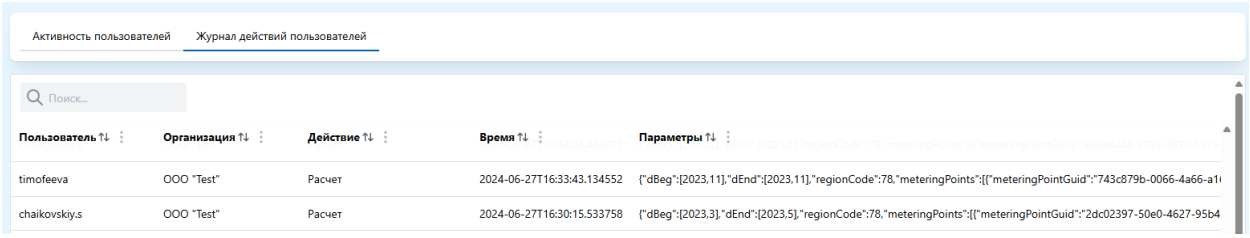


4.3.11 **Логи**

На странице данного раздела доступны все логи программы.



Можно оценить как активность отдельных пользователей, так и ознакомиться с журналом их действий:


Активность пользователей
Журнал действий пользователей
Поиск...
Пользователь
Организация
Действие
Время
Параметры
timofeeva
ООО "Test"
Расчет
2024-06-27T16:33:43.134552
chaikovskiy.s
ООО "Test"
Расчет
2024-06-27T16:30:15.533758

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ПК	-	персональный компьютер
ПО	-	программное обеспечение
ЭВМ	-	электронная вычислительная машина

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

ПО «ЭнергоБИТ»	-	программа для ЭВМ с использованием искусственного интеллекта для прогнозирования часов пиковой нагрузки и оптимизации энергопотребления организаций.
Web-интерфейс	-	дизайн приложения ПО «ЭнергоБИТ», доступ к которому осуществляется через web-браузера.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Наименование организации	Должность исполнителя	Ф.И.О.	Подпись	Дата