ДОКУМЕНТАЦИЯ ПО УСТАНОВКЕ И ЭКСПЛУАТАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Программы для ЭВМ Информационно-аналитическая система «Энергобаланс»

ГУП «ТЭК СПб»
Санкт-Петербург, 2024 г.

ОГЛАВЛЕНИЕ

1 ВВЕДЕНИЕ

1.1 НАЗНАЧЕНИЕ ДОКУМЕНТА

Настоящий документ представляет собой руководство администратора для программы ЭВМ Информационно-аналитическая система «Энергобаланс» (далее по тексту - ИАС «Энергобаланс»).

Полное наименование системы – **Аналитическая система расчета технико-экономических показателей, оптимизации режимов работы системы теплоснабжения и построения Энергобаланса ГУП «ТЭК СПб».**

Краткое наименование системы – **ИАС «Энергобаланс»**. Также для обозначения системы в этом документе будут использованы наименования **система** или **программный комплекс**.

Настоящее руководство предназначено для администраторов системы, обладающих опытом работы с приложениями для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, а также навыками администрирования PostgresSQL (версии не ниже 12.4) для развёртывания баз данных.

1.2 ТЕРМИНЫ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

Термин/сокращение	Толкование сокращения/определение термина
Angular	JavaScript-фреймворк с открытым исходным кодом, предназначенный для разработки одностраничных приложений
CORS	от англ. Cross-origin resource sharing (совместное использование ресурсов между разными источниками) - технология современных браузеров, которая позволяет предоставить веб-странице доступ к ресурсам другого домена
CSS	от англ. Cascading Style Sheets (каскадные таблицы стилей) - формальный язык описания внешнего вида документа, написанного с использованием языка разметки
HTML	от англ. HyperText Markup Language (язык гипертекстовой разметки) - стандартизированный язык разметки документов в сети Интернет
JavaScript	Мультипарадигменный язык программирования
JSON	от англ. JavaScript Object Notation - текстовый формат обмена данными, основанный на JavaScript
.NET Core	Универсальная платформа разработки с открытым кодом, которая поддерживает работу во множестве ОС
Web API	Интерфейс программирования, представляющий собой набор готовых классов, процедур, функций, структур и констант для использования во внешних программных продуктах

АСУТП	Автоматизированная система управления технологическими процессами
БД	База данных
Модель ML	Модель машинного обучения. Алгоритм обучения, описывающий то, каким именно образом будет обучаться компьютер, какие данные и параметры потребуется задействовать в процессе, в какой очередности будут выполняться те или иные команды и др.
ML	Машинное обучение (англ. machine learning, ML) — класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение за счёт применения решений множества сходных задач.
Система	Информационно-аналитическая система «Энергобаланс»

2 ОБЩИЕ ПОЛОЖЕНИЯ

2.1 НАЗНАЧЕНИЕ СИСТЕМЫ

Программное обеспечение является универсальным модулем позволяющей создавать, обучать и использовать уже созданные модели машинного обучения для решения различного класса задач в части прогнозирования.

2.2 УСЛОВИЯ ИСПОЛЬЗОВАНИЯ

Для корректной работы клиентской части системы рекомендуется использовать браузер **Google Chrome**, версии не ниже 75.

Для корректной работы серверной части системы рекомендуется использовать свободно-распространяемую ОС на базе ядра Linux (например Debian), программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации – Docker, а также СУБД PostgreSQL.

3 АРХИТЕКТУРА СИСТЕМЫ

Система построена на базе трёхзвенной клиент-серверной архитектуры:

- Уровень **базы данных** – единая база данных под управлением СУБД PostgreSQL.
- Уровень **приложения** – сервисы, реализующие бизнес-логику. Приложение реализовано на платформе Microsoft .NET Core. Применяемый язык разработки – C#. Для связи с уровнем базы данных используется ORM Entity Framework. Для взаимодействия с сервисами используется подход REST API.
- Уровень **клиента** – web-клиент, в котором реализованы интерфейсы для выполнения основных бизнес-процессов пользователей. Web-клиент реализован на платформах Angular 14+ и PrimeNG. В качестве основных языков программирования применяются Typescript, JS, C#. Используемый протокол передачи данных между уровнями – HTTP.

Функциональная архитектура системы представлена ниже (Рис. 1).

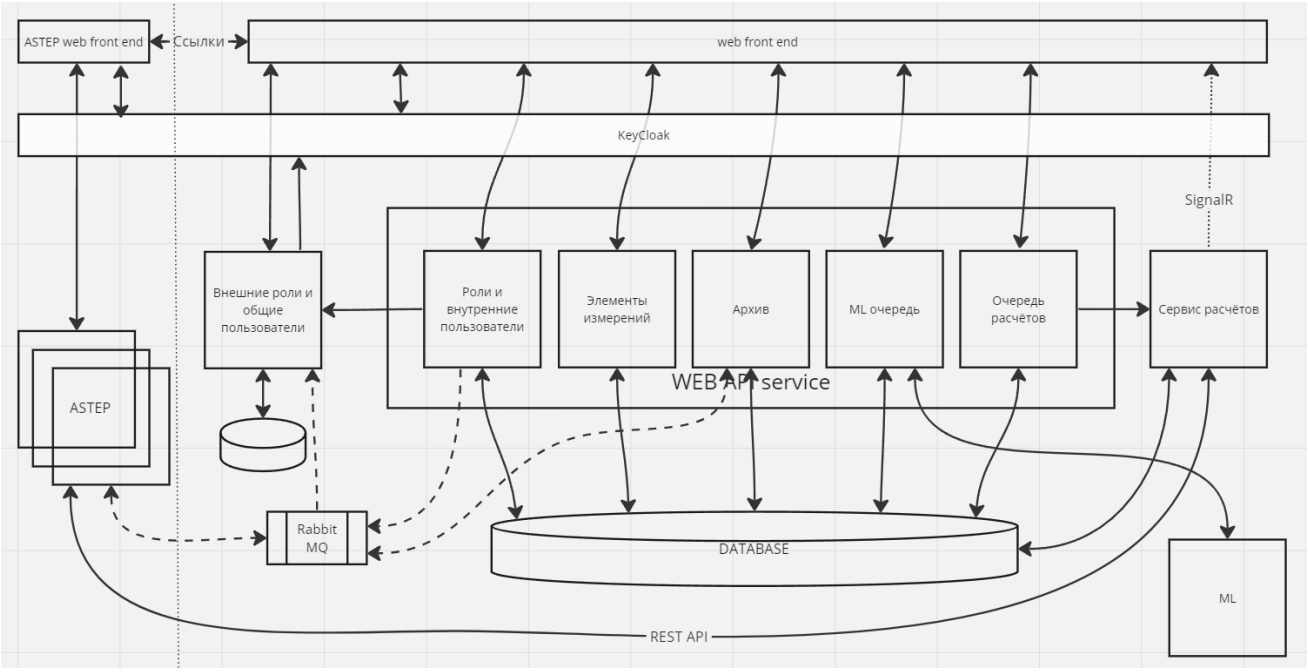

Рис. 1 – Общий вид архитектуры системы

4 ПОДКЛЧЕНИЕ К ВНУТРЕННИЙ СЕТИ ГУП «ТЭК СПБ»

Для получения доступа к внутренней корпоративной сети ГУП «ТЭК СПб», в которой размещен сервер с ПО ИАС «Энергобаланс», необходимо

5 РАЗВЕРТЫВАНИЕ DOCKER НА СЕРВЕРЕ ПРИЛОЖЕНИЙ

1) Выполнить обновление путем выполнения команды:

$ sudo apt update

2) Установить пакеты, которые требуются для добавления нового репозитория через HTTPS:

$ sudo apt install apt-transport-https ca-certificates curl gnupg lsb-release -y

3) Добавить ключ GPG репозитория Docker командой:

$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

4) Добавить репозиторий командой:

$ sudo echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian \

$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

5) Установить Docker и Docker-compose выполнив последовательно команды:

$ sudo apt update

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

6) Добавить Docker в автозагрузку и запустите его командой:

$ sudo systemctl enable docker

$ sudo systemctl start docker

$sudo groupadd docker

$sudo usermod -aG docker $USER

$sudo systemctl enable docker.service

$sudo systemctl enable containerd.service

7) Проверьте версию docker и состояние его служб последовательно командами:

$ sudo docker version

$ sudo systemctl status docker

6 РАЗВЕРТЫВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ

1) На сервере приложений перейти в директорию для ПО, путем выполнения команды:

cd /var/opt/

2) Перенести в директорию /var/opt/ архив front_ml.tar.gz содержащий контейнеры и базу приложения. Для загрузки архива можно воспользоваться ПО «Winscp» выполнив команду:

scp front_ml.tar.gz root@123.123.123.123:/directory

,где 123.123.123.123 адрес целевого хоста.

3) Распаковать архив в /var/opt/, выполнив команду:

tar -zxvf front_ml.tar.gz

4) Перейти в директорию var/opt/front_ml выполнив команду:

cd /var/opt/front_ml

5) В директории /var/opt/front_ml выполнить команду для загрузки образов докер в хранилище:

docker load --input api.tar && docker load --input fond_ml_api.tar && docker load --input front.tar && docker load --input postgres.tar

6) Находясь в директории /var/opt/front_ml необходимо указать IP адрес хоста на котором будут запущены docker контейнеры, для этого выполняем команду:

sed -i "s^192.168.100.2^xxx.xxx.xxx.xxx^g" docker-compose.yaml

, где xxx.xxx.xxx.xxx целевой IP адрес хоста.

7) Находясь в директории /var/opt/front_ml запустить контейнеры, путем выполнения команды:

docker compose up -d

Будут запущены 4 контейнера:

- Контейнер СУБД;
- Контейнер содержащий бэкенд;
- Контейнер содержащий фронтенд;
- Контейнер содержащий сервис ML.

8) Проверить, что веб-приложение доступно для работы, путем запуска браузера и ввода в строке адреса, IP адреса приложения следующего вида:

xxx.xxx.xxx.xxx:4200

, где "xxx.xxx.xxx.xxx" адрес сервера на котором развернуты контейнеры.

9) Выполнить авторизацию в приложении путем ввода логина и пароля.
Логин: root
Пароль: infoproXZQ45

7 РАЗВЕРТЫВАНИЕ ПРИЛОЖЕНИЕЯ С ПОМОЩЬЮ DOCKER-COMPOSE

Приложение состоит из основных и контейнеров сопутствующих сервисов.

В Docker-compose Энергобаланса необходимо добавить ip адрес сервера на котором развернуто приложение, а также адрес сервера баз данных.

Перед запуском Docker-compose Энергобаланса небходимо выполнить команду docker login dr-1.info-pro , используя логин robot$infbalans+install и пароль YPA4IbGWJs4vzC6JXz1ztAuIjDniBxc5 автоизоваться в репозитории для скачивания и запуска контейнеров. После выполнить команду dcoker-compose up -d . Если все выполнено верно, контейнеры будут скачены из dr-1.info-pro и запушены.

Создадим дерикторию для разворачивания приложение и перейдем в нее:

mkdir -p /opt/infbalans && cd /opt/infbalans

6.1 Запуск Docker-compose Энергобаланса создадим файл docker-compose.yml :

Командой создадим nano docker-compose.yml файл и и перенесем в него ниже описанную конфигурацию.

```
version: '3.0'

services:

  front:

    image: dr-1.info-pro.ru/infbalans/front:latest

    container_name: infbalans_front

    environment:

      NODE_ENV: Production

      ENVIRONMENT: prod

    ports:

      - 4200:80

    restart: unless-stopped

  api:

    hostname: INF.BALANS.WEB.API

    image: dr-1.info-pro.ru/infbalans/api:latest

    container_name: infbalans_api

    environment:

     ASPNETCORE_ENVIRONMENT: Production

     MlOptions__MLApiOptions__BaseAddress: http://api_ml:8015
```

```
    CalculationApiOptions__BaseAddress: http://api_calc:9995
    SecurityApiOptions__BaseAddress: http://api_security:9990
    MLOptions__StoragePath: /app/wwwroot/mldata
    CorsOptions__AllowedOrigins__0: http://"ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ":4200
    Serilog__MinimumLevel__Override_Quartz: Information
    Serilog__WriteTo__2__Args__nodeUris: http://"ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ":9200
    Serilog__Properties__ApplicationName: INF.BALANS.WEB.API
    Serilog__WriteTo__3__Args__enabled: "true"
    ConnectionStrings__DbContext:
"ВНЕСТИ_IP_АДРЕС_БД;Port=5432;Database=infbalans;Username=postgres;Password=CegthGfhjkm
;Include Error Detail=true;Command Timeout=18000"
    AuthOptions__LdapOptions__Enabled: "false"
    AuthOptions__JwtBearerOptions__Authority:
http://"ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ":8181/realms/sso
    AuthOptions__JwtBearerOptions__TokenValidationParameters__ValidIssuer:
http://"ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ":8181/realms/sso
    ServiceBusOptions__RabbitMqOptions__Host: "ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ"
    ServiceBusOptions__RabbitMqOptions__VHost: /infbalans
    Quartz__ArchiveValuesMetricsJob: null
    Quartz__RemoveDeprecatedArchiveValuesJob: "0 0 21 ? * * *"
    Quartz__RemoveDeprecatedFactArchiveValuesJob: "0 0 0-20,22,23 ? * * *"
    RemoveDeprecatedArchiveValuesJobOptions__ClearFact: "false"
  restart: unless-stopped
  ports:
    - 9999:9999
  volumes:
    - mldata_api:/app/wwwroot/mldata
api_calc:
  image: dr-1.info-pro.ru/infbalans/api_calc:latest
```

```
  container_name: infbalans_api_calc
  environment:
   - ApplicationProperties__ApplicationId=INF.BALANS.WEB
   - ASPNETCORE_ENVIRONMENT=Production
   - SecurityApiOptions__BaseAddress=http://api_security:9990
   -
ConnectionStrings__DbContext=ВНЕСТИ_IP_АДРЕС_БД;Port=5432;Database=infbalans;Username=p
ostgres;Password=CegthGfhjkm;Include Error Detail=true;Command Timeout=9000
   - CorsOptions__AllowedOrigins__0=http://"ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ":4200
   - Serilog__MinimumLevel_Override_Quartz=Information
   - Serilog__WriteTo__2__Args__nodeUris=http://"ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ":9200
   - Serilog__Properties__ApplicationName=INF.BALANS.WEB.CALC.API
   - Serilog__WriteTo__3__Args__enabled=true
   -
AuthOptions__JwtBearerOptions__Authority=http://"ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ":8181/realm
s/sso
   -
AuthOptions__JwtBearerOptions__TokenValidationParameters__ValidIssuer=http://"ВНЕСТИ_IP_АДРЕ
С_ПРИЛОЖЕНИЯ":8181/realms/sso
   - ServiceBusOptions__RabbitMqOptions__Host="ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ"
   - ServiceBusOptions__RabbitMqOptions__VHost=/infbalans
  restart: unless-stopped
  ports:
    - 9995:9995
  volumes:
    - logs:/app/logs
    - mldata:/app/wwwroot/mldata
  logging:
    driver: "json-file"
    options:
```

```
        max-size: "500m"
        max-file: "5"
  api_security:
    image: dr-1.info-pro.ru/infbalans/api_security:latest
    container_name: infbalans_api_security
    environment:
     - CorsOptions__AllowedOrigins__0=http://"ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ":4200
     - ASPNETCORE_ENVIRONMENT=Production
     -
ConnectionStrings__DbContext=ВНЕСТИ_IP_АДРЕС_БД;Port=5432;Database=infbalans;Username=p
ostgres;Password=CegthGfhjkm;Include Error Detail=true;Command Timeout=9000
     - Serilog__WriteTo__2__Args__nodeUris=http://"ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ":9200
     - Serilog__Properties__ApplicationName=INF.BALANS.WEB.SECURITY.API
     - Serilog__WriteTo__3__Args__enabled=true
     -
AuthOptions__JwtBearerOptions__Authority=http://"ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ":8181/realm
s/sso
     -
AuthOptions__JwtBearerOptions__TokenValidationParameters__ValidIssuer=http://"ВНЕСТИ_IP_АДРЕ
С_ПРИЛОЖЕНИЯ":8181/realms/sso
     - ServiceBusOptions__RabbitMqOptions__Host="ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ"
     - ServiceBusOptions__RabbitMqOptions__VHost=/infbalans
     - KeyCloakServiceOptions__BaseAddress=http://"ВНЕСТИ_IP_АДРЕС_ПРИЛОЖЕНИЯ":8181
    restart: unless-stopped
    ports:
      - 9990:9990
    volumes:
      - logs:/app/logs
    logging:
      driver: "json-file"
```

```
    options:
      max-size: "500m"
      max-file: "5"
  api_ml:
    image: dr-1.info-pro.ru/infbalans/api_ml:latest
    container_name: api_ml
    volumes:
      - logs:/usr/local/app_ml/logs
      - models:/usr/local/app_ml/models
      - datasets:/usr/local/app_ml/datasets
      - scaler:/usr/local/app_ml/scaler
    restart: unless-stopped
    ports:
      - 8015:8015
volumes:
    logs: {}
    models: {}
    mldata: {}
    datasets: {}
    scaler: {}
    mldata_api: {}
```

8 ЗАПУСК DOCKER-COMPOSE СОПУТСТВУЮЩИХ СЕРВИСОВ:

Создадим директорию для разворачивания приложение и перейдем в нее:

mkdir -p /opt/service && cd /opt/service

скачаем в нее файлы https://cloud.mail.ru/public/K6Ky/sj1SZzbVN

Разархивировать файл infbalansDATA.tar

Далее архивы themes.tar и date.tar распаковать в /opt/services/volumes/keycloak.

Архив rabbit.tar загрузить командой docker load < /home/user/rabbit.tar

Командой создадим nano docker-compose.yml файл и и перенесем в него нижеописанную конфигурацию.

Застим приложение командой docker-compose up -d

```
version: '3.0'

        services:
          grafana:
            image: grafana/grafana:latest
            container_name: grafana
            volumes:
              - grafana_data:/var/lib/grafana
              - ./volumes/grafana/provisioning:/etc/grafana/provisioning
            environment:
              - GF_SECURITY_ADMIN_USER=admin
              - GF_SECURITY_ADMIN_PASSWORD=admin
              - GF_USERS_ALLOW_SIGN_UP=false
            restart: unless-stopped
            ports:
              - 3000:3000
          kibana:
            container_name: kibana
            image: kibana:7.17.3
            ports:
              - 5601:5601
            depends_on:
              - elasticsearch
            environment:
              ELASTICSEARCH_HOSTS: http://elasticsearch:9200
              ELASTICSEARCH_URL: http://elasticsearch:9200
            restart: unless-stopped
          keycloak:
            image: quay.io/keycloak/keycloak:20.0.3
            user: keycloak
            restart: always
            container_name: keycloak
            ports:
              - 8181:8080
            environment:
              - KEYCLOAK_USER=admin
```

```
      - KEYCLOAK_PASSWORD=admin
      - KEYCLOAK_ADMIN=admin
      - KEYCLOAK_ADMIN_PASSWORD=admin
      - KEYCLOAK_LOGLEVEL=TRACE
    volumes:
      - ./volumes/keycloak/data:/opt/keycloak/data
      - ./volumes/keycloak/themes:/opt/keycloak/themes
    command: start --hostname-strict=false --http-enabled=true
  rabbitmq:
    container_name: rabbitmq
    image: bitnami/rabbitmq:latest
    hostname: rabbitmq
    restart: always
    environment:
      - RABBITMQ_DEFAULT_USER=rmquser
      - RABBITMQ_DEFAULT_PASS=rmqpass
      - RABBITMQ_VHOSTS=/infbalans
      - RABBITMQ_SERVER_ADDITIONAL_ERL_ARGS=-rabbit log_levels [{connection,error},{default,error}]
      - RABBITMQ_NODE_NAME=rabbit@rabbitmq
    ports:
      - 15672:15672
      - 5672:5672
  redis:
    container_name: redis
    image: redis:6.2-alpine
    restart: always
    ports:
      - 6379:6379
    command: redis-server --loglevel warning --proto-max-bulk-len 4294967296 --client-query-buffer-limit 4294967296
    volumes:
      - ./volumes/redis:/data
volumes:
    grafana_data: {}
    prometheus_data: {}
```

9 УСТАНОВКА СУБД POSTGRESQL С РЕПОЗИТОРИЯ ASTRA LINUX

Добавьте официальный репозиторий Astra Linux для доступа к PostgreSQL 14. Для этого выполните следующие шаги:

Шаг 1. Откройте терминал комбинацией **Alt + T**;

Шаг 2. Войдите под root пользователем;

sudo su

Перед подключением репозиториев необходимо установить дополнительные пакеты. Если у вас имеется образ с репозиторием Astra Linux, обновите пакеты, используя команду **apt update**, а затем проведите установку:

apt install ca-certificates apt-transport-https

Шаг3. Добавьте адрес дополнительного репозитория в файл /etc/apt/source.list:

deb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/ 1.7_x86-64 astra-ce

Для добавления репозиториев в файл рекомендуем использовать текстовый редактор **nano**.

Шаг 1. Откройте файл с помощью команды (sudo) **nano /etc/apt/sources.list**

Шаг 2. Скопируйте репозитории и вставьте их с помощью комбинации **Ctrl + U (Shift + Insert** или **Правка → Вставить)**;

Шаг 3. Сохраните файл с помощью комбинации **Ctrl + O**;

Для выхода используйте **Ctrl + X**;

Шаг 4. Обновите пакеты;
apt update

Шаг 5. Установите PostgreSQL;
apt install -y postgresql

Настройка PostgreSQL

Шаг 1. Добавьте службу **postgresql** в автозапуск;

systemctl enable postgresql

```
root@astra-db:/etc/apt# systemctl enable postgresql
Synchronizing state of postgresql.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable postgresql
```

Шаг 2.Задайте пользователю пароль для подключения к СУБД;

PostgreSQL по умолчанию создает супер-пользователя **postgres**.

sudo su postgres
psql -U postgres
ALTER USER **postgres** WITH PASSWORD '<новый пароль>';
ШАГ 3. Создание базы:
CREATE DATABASE "infbalans" WITH OWNER **postgres** ENCODING 'UTF8' LC_COLLATE = 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8';
Шаг 4. Завершите сессию командой **exit**

psql -h localhost -U postgres -d INF_BALANS_TEST_Q01 -f infbalans21052024.sql

Шаг 5. Разворачивание базы Энергобаланса и архива

Для разворачиване базы нвыполнить команду psql -h localhost -U postgres -d infbalans -f infbalans.sql

10 АВТОРИЗАЦИЯ В ПРИЛОЖЕНИИ

Для авторизации в приложении необходимо перейти по адресу http://10.1.2.187:4200/archive/registry и в окне авторизации ввести логин: operator, пароль: operator.