

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
ЦИФРОВАЯ ПЛАТФОРМА «АВАНГАРД»**

**РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА**

Листов 54

Москва  
2022

## АННОТАЦИЯ

В документе приведены общие сведения о назначении и архитектуре Цифровой платформы «Авангард» (далее – Система), требования к аппаратному и программному обеспечению Системы, к квалификации персонала, сопровождающего Систему. Подробно описаны процессы жизненного цикла программного обеспечения (далее также – ПО) на стадиях внедрения и сопровождения Системы:

- первоначальная установка и настройка программного обеспечения Системы;
- поддержание работоспособности программного обеспечения Системы, в том числе восстановление Системы после сбоев;
- обновление программного обеспечения Системы.

Документ не содержит описания процесса вывода программного обеспечения Системы из эксплуатации, так как регламентация процесса отличается в каждом конкретном случае.

## СОДЕРЖАНИЕ

1. Общие сведения о системе.....	7
1.1. Назначение Системы.....	7
1.2. Состав Системы .....	7
1.3. Требования к техническому и программному обеспечению .....	7
1.3.1. Требования к программному обеспечению .....	7
1.3.2. Требования к техническому обеспечению .....	8
1.4. Требования к квалификации персонала, обеспечивающего эксплуатацию Системы .....	9
2. Установка и настройка программного обеспечения .....	10
2.1. Установка и настройка сервиса авторизации .....	10
2.1.1. Подготовка СУБД .....	10
2.1.1.1. Развёртывание БД сервиса авторизации .....	10
2.1.1.2. Выполнение скрипта (для не CodeFirst миграций).....	11
2.1.2. Развёртывание .Net сервиса в linux .....	13
2.1.2.1. Установка .Net.....	13
2.1.2.2. Настройка бэкенд-сервиса .....	14
2.1.2.3. Настройка nginx .....	14
2.1.2.4. Перезапуск служб .....	17
2.1.2.5. Ошибки и решения .....	17
2.1.3. Настройка сервиса авторизации .....	17
2.1.3.1. Настройка конфигурации фронтенд .....	17
2.1.3.2. Настройка конфигурации .Net сервиса .....	18
2.2. Установка и настройка сервиса файлового хранилища .....	23
2.2.1. Подготовка СУБД .....	23
2.2.2. Развёртывание .Net сервисов в linux .....	24
2.2.2.1. Установка .Net.....	24
2.2.2.2. Настройка бэкенд-сервисов .....	24
2.2.2.3. Настройка nginx .....	24
2.2.2.4. Перезапуск служб .....	25
2.2.3. Настройка сервиса файлового хранилища .....	25
2.2.3.1. Настройка конфигурации .Net сервиса .....	25
2.3. Установка и настройка сервиса интерактивных рабочих столов и аналитики .....	25

2.3.1.	Подготовка СУБД .....	25
2.3.2.	Развёртывание .Net сервисов в linux .....	26
2.3.2.1.	Установка .Net.....	26
2.3.2.2.	Настройка бэкенд-сервисов .....	26
2.3.2.3.	Настройка nginx .....	27
2.3.2.4.	Перезапуск служб .....	27
2.3.3.	Настройка сервиса интерактивных рабочих столов и аналитики ..	28
2.3.3.1.	Настройка конфигурации .Net сервиса .....	28
2.4.	Установка и настройка сервиса оболочки .....	28
2.4.1.	Выполнить следующие действия: .....	28
2.4.2.	Настройка nginx.....	28
2.4.3.	Настройка сервиса оболочки.....	29
3.	Обновление программного обеспечения.....	30
3.1.	Обновление сервиса авторизации.....	30
3.2.	Обновление сервиса файлового хранилища .....	30
3.3.	Обновление сервиса интерактивных рабочих столов и аналитики ....	30
3.4.	Обновление сервиса оболочки .....	31
4.	Проверка, восстановление и поддержание работоспособности программного обеспечения.....	32
4.1.	Методы проверки работоспособности рабочих станций .....	32
4.2.	Методы проверки работоспособности сервера .....	32
4.3.	Методы проверки работоспособности базы данных .....	33
4.3.1.	Проверка физической целостности базы данных .....	33
4.3.2.	Проверка сохранения введенных данных.....	35
4.4.	Методы восстановления работоспособности сервера .....	35
4.5.	Методы восстановления работоспособности базы данных .....	36
4.5.1.	Методы восстановления работоспособности базы данных под управлением СУБД Postgres Pro 14.....	36
4.5.1.1.	Резервное копирование базы данных .....	36
4.5.1.2.	Восстановление базы данных .....	36
4.5.2.	Методы восстановления работоспособности базы данных под управлением СУБД MongoDB.....	36
4.5.2.1.	Резервное копирование базы данных .....	36
4.5.2.2.	Восстановление базы данных .....	37
4.6.	Методы поддержания целостности базы данных .....	37

4.7. Методы поддержания безопасности базы данных.....	37
4.8. Методы диагностирования проблем обновления баз данных .....	37
4.8.1. Диагностирование проблем в работе сервисов .....	37
5. Администрирование Системы.....	40
5.1. Доступ к сервису.....	40
5.2. Управление доступом.....	41
5.2.1. Клиенты.....	41
5.2.2. Ресурсы.....	44
5.2.2.1. Создание нового ресурса.....	45
5.2.2.2. Раздел API.....	45
5.2.2.3. Раздел Полномочия.....	46
5.2.3. Роли.....	47
5.2.4. Пользователи .....	49
5.2.5. Сотрудники .....	51
5.2.6. События .....	52
6. Аварийные ситуации .....	54

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

Сокращенное наименование	Полное наименование
БД	База данных
ЛКМ	Левая кнопка мыши
ОС	Операционная система
Система, программное обеспечение	Цифровая платформа «Авангард»
СУБД	Система управления базами данных

# **1. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ**

## **1.1. Назначение Системы**

Программное обеспечение Цифровая платформа «Авангард» предназначена для организации единого рабочего пространства организации с возможностью расширения набора автоматизируемых процессов благодаря использованию модульной архитектуры.

## **1.2. Состав Системы**

Система представляет собой web-приложение и состоит из следующих частей:

- серверная часть в составе:
  - сервер СУБД – управление данными;
  - сервер веб-приложения – бизнес-логика и внешние процедуры, реализованная посредством следующих компонент:
    - сервис авторизации (далее также – хаб);
    - сервис файлового хранилища;
    - сервис интерактивных рабочих столов и аналитики;
    - сервис оболочки.
- клиентская часть – работа с пользовательским графическим интерфейсом Системы посредством браузера.

## **1.3. Требования к техническому и программному обеспечению**

### **1.3.1. Требования к программному обеспечению**

Требуемый состав программного обеспечения сервера:

- серверная операционная система семейства Linux, включенная в единый реестр российских программ для электронных вычислительных машин и баз данных;
- СУБД PostgresPro либо PostgreSQL версии не ниже 14;
- СУБД MongoDB 5;
- прокси Nginx 1.14;
- программная платформа .NET sdk 6.0.

Требуемый состав программного обеспечения пользовательской рабочей станции:

- операционная система семейства Linux, включенная в единый реестр российских программ для электронных вычислительных машин и баз данных;
- браузер Google Chrome, Yandex Browser или Mozilla Firefox последней, или предпоследней версии.

### 1.3.2. Требования к техническому обеспечению

К аппаратной части серверной части предъявляются следующие требования:

- Процессоры:
  - количество не менее 2;
  - архитектура процессора x86-64;
  - ядер не менее 8;
  - потоков не менее 16;
  - тактовая частота в режиме повышенной нагрузки не менее 3,3 ГГц;
  - кэш не менее 20 Мб;
  - поддержка памяти ECC.
- Оперативная память:
  - объем не менее 64 Гб;
  - тип оперативной памяти DDR3/DDR4 с функцией коррекции ошибок.
- Сетевой интерфейс:
  - не менее 1 порта 100 Мб/с.
- Дисковая подсистема:
  - аппаратный RAID;
  - интерфейс SAS не менее 6 Гб/сек;
  - HDD с буфером обмена не менее 128 Мб либо SSD.

Коммуникационная среда должна обеспечивать информационное взаимодействие между компонентами Системы в соответствии с транспортным протоколом TCP/IP.

К аппаратной части рабочей станции пользователя предъявляются следующие требования:

- Процессоры:
    - количество не менее 1;
    - архитектура процессора x86-64;
    - ядер не менее 2;
- Допустимо использование следующих видов процессоров:
- настольные процессоры Intel и AMD, вышедшие на рынок не ранее 2013 года;
  - мобильные процессоры Intel и AMD, вышедшие на рынок не ранее 2015 года, кроме линейки процессоров Intel Atom;
  - процессоры Apple (M1, M1 PRO, M1 MAX).
- Оперативная память:
    - объем не менее 4Гб (рекомендуется 8Гб);



- тип оперативной памяти - DDR3/DDR4.
  - Сетевой интерфейс:
    - не менее 1 порта 100МБ/с
- (доступ к сервисам системы со скоростью не ниже 8 Мбит/с (для быстрой загрузки приложения рекомендуется 25 Мбит/с и выше)).
- Дисковая подсистема:
    - HDD с буфером обмена не менее 64 Мб либо SSD.
  - Графический режим монитора:
    - 1366x768 и выше (рекомендуется 1920x1080).
  - Клавиатура, мышь.

#### **1.4. Требования к квалификации персонала, обеспечивающего эксплуатацию Системы**

Персонал, выполняющий функции технического сопровождения Системы, должен обладать экспертными навыками:

- обеспечения функционирования серверов и рабочих станций в среде ОС семейства Linux;
- установки и настройки программного обеспечения, приведённого в п.1.3.1.

## 2. УСТАНОВКА И НАСТРОЙКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### 2.1. Установка и настройка сервиса авторизации

Процесс установки необходимого программного обеспечения описывается, исходя из предположения, что на сервере имеется доступ к репозиторию с необходимыми пакетами для установки. Для выполнения большинства операций потребуется вводить команды посредством интерфейса командной строки Linux (на примере ОС CentOS).

#### 2.1.1. Подготовка СУБД

##### 2.1.1.1. Развёртывание БД сервиса авторизации

Выполнить следующие действия:

- Установить на сервер СУБД PostgresPro либо PostgreSQL версии не ниже 14 в соответствии с руководством от производителя ПО;
- Подключиться к серверу, например, с помощью графического клиента pgAdmin;
- Создать БД hub и поднять бэкап (/Бэкапы баз данных/Бэкап базы хаба.bak.sql).

Установка СУБД на примере дистрибутива PostgresPro Enterprise 11 из ISO образа:

1. Подключить ISO образ дистрибутива:

```
mount PostgresProEntCert-11.12.1.iso /mnt/cdrom/ -o loop
```

2. В файл репозитория добавить iso образ  
/etc/yum.repos.d/RedOS-Sources.repo

```
[cdrom]
```

```
name= CDROM
```

```
baseurl=file:///mnt/cdrom/redos/7.2/os/x86_64/rpms
```

```
gpgkey=file:///mnt/cdrom/keys/GPG-KEY-POSTGRESPRO
```

```
enabled=1
```

```
gpgcheck=1
```

3. Далее обновить информацию о пакетах и устанавливаем СУБД:

```
yum update
```

```
yum install postgrespro-ent-11
```

4. Переключиться на пользователя postgres:

```
su postgres
```

5. Подключиться к консоли postgresql:

psql

6. Создать базы:

create database hub

7. create database logs

8. Подключиться к базе hub:

\c hub

9. Выполнить скрипт (см. п. 2.1.1.2)

**Примечание.** При высокой нагрузке логи можно выделить в отдельную БД. Для этого необходимо создать БД logs, прогнать на ней миграции и указать её в конфигурационном файле.

#### 2.1.1.2. Выполнение скрипта (для не CodeFirst миграций)

```
DO
$do$
begin
IF NOT EXISTS (
  SELECT FROM pg_catalog.pg_class c
  JOIN pg_catalog.pg_namespace n ON n.oid = c.relnamespace
  WHERE n.nspname = 'public'
  AND c.relname = 'migration_history'
  AND c.relkind = 'r'
) then
CREATE TABLE public.migration_history
(
  id          uuid    not null constraint pk_migration_history_id primary key
, file_name   text    not null
, date        date    not null
, number      integer not null default (0)
, author      text    not null
, name        text    not null
, request_number integer null
, begin_date  timestamp not null default (now()::timestamp)
, end_date    timestamp null
, error       text    null
, index       integer not null default (0)
, content     text    null
, type        integer not null
);
end if;
end
$do$;
```

```
create or replace function public.test_procedure(test_input integer)
    returns integer
    language plpgsql
AS
$function$
    begin
        return test_input;
    end;
$function$;
```

```
create or replace function public.run_migration_script (script text)
returns boolean
language plpgsql
AS
$function$
begin
    execute script;
    return true;
end;
$function$;
```

```
create or replace function public.add_migration(
    id uuid,
    file_name text,
    date timestamp,
    number integer,
    author text,
    name text,
    request_number integer = null,
    type integer = null,
    index integer = null,
    content text = null
)
    returns uuid
    language plpgsql
AS
$function$
Begin

    insert into public.migration_history (id, file_name, date, number, author, name,
        request_number, type, index, content)
    SELECT id, file_name, date::date, number, author, name, request_number, type,
        index, content;
```

```

    return id;
end;
$function$;

create or replace function public.complete_migration (
    migration_id uuid,
    migration_end_date timestamp = null,
    migration_error text = null
)
returns uuid
language plpgsql
AS
$function$
begin
    if migration_end_date is null then
        migration_end_date := now()::timestamp;
    end if;

    update public.migration_history mh
    set end_date = migration_end_date, error = migration_error
    where mh.id = migration_id;

    return migration_id;
end;
$function$;

create or replace function public.get_completed_migrations()
returns table(file_name text)
language plpgsql
AS
$function$
declare programmability_update_type integer;
begin
    programmability_update_type := 3;

    return query
    select (mh.file_name)
    from public.migration_history mh
    where mh.error is null and mh.type != programmability_update_type
    order by mh.begin_date desc;
end;
$function$;

```

## **2.1.2. Развёртывание .Net сервиса в linux**

### **2.1.2.1. Установка .Net**

Выполнить команды:

```
wget https://packages.microsoft.com/config/debian/10/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb
sudo yum install -y dotnet-sdk-6.0
```

### 2.1.2.2. Настройка бэкенд-сервиса

Выполнить следующие действия:

- Копировать дистрибутив сервиса (\Дистрибутивы сервисов\Сервис авторизации\) на сервер (/usr/share/hosting/auth).
- Создать пользователя, под которым будет запускаться приложение. Пользователь должен иметь полный доступ к директории с приложением и права открывать сокеты.
- Создать службу /etc/systemd/system/hub.service. Указать директории, пути к файлам и приложениям, описание и указать пользователя под кем будет запускать приложение:

```
[Unit]
Description=[Authentication] Портал Авторизации ХАБ

[Service]
WorkingDirectory=/usr/share/hosting/auth
ExecStart=/usr/bin/dotnet /usr/share/hosting/auth/Quarta.Auth.Web.dll
Restart=always
RestartSec=10
SyslogIdentifier=dotnet-auth
User=%ИМЯ СИСТЕМНОГО ПОЛЬЗОВАТЕЛЯ%
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false
Environment=DOTNET_CLI_TELEMETRY_OPTOUT=true
LimitNOFILE=49152

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

Перезагрузить информацию о сервисах, выполнив команду:  
systemctl daemon-reload

### 2.1.2.3. Настройка nginx

Установить Nginx (любая последняя версия), выполнив команды:

```
$ wget https://nginx.org/download/nginx-1.19.0.tar.gz
$ tar xzf nginx-1.19.0.tar.gz
$ cd nginx-1.19.0
```

Подготовить ключ (/home/hosting/crt/private.key) и сертификат (/home/hosting/crt/cert.crt) который будут использоваться для SSL шифрования (https).

**Ниже пример настройки nginx модуля http**

```
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;
    tcp_nopush     on;
    tcp_nodelay    on;

    keepalive_timeout 65;
    ssl_protocols  TLSv1.2 TLSv1.3;
    ssl_ciphers    HMAC-SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-
        AES256-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-
        SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-
        SHA384;

    #gzip on;
    client_max_body_size 1024M;
    large_client_header_buffers 4 16k;

    fastcgi_buffers 16 32k;
    fastcgi_buffer_size 64k;
    fastcgi_busy_buffers_size 64k;

    ##
    # Proxy
    ##
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection keep-alive;
    proxy_set_header Host $host:$server_port;
    proxy_cache_bypass $http_upgrade;
    proxy_read_timeout 1200;
    proxy_buffer_size 128k;
```

```
proxy_buffers 4 256k;
proxy_busy_buffers_size 256k;

##
# Compression settings
##

gzip on;
gzip_disable "msie6";
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 application/json
application/x-javascript application/javascript text/xml application/xml
application/xml+rss text/javascript;include /etc/nginx/conf.d/*.conf;
```



#Добавить в server прокси к приложению:

```
server {  
    listen    1090;  
    location / {  
        proxy_pass http://127.0.0.1:5090;  
    }  
}
```

#Пример при использовании SSL:

```
server {  
    listen    1090 ssl;  
    ssl_certificate /home/hosting/crt/cert.crt;  
    ssl_certificate_key /home/hosting/crt/private.key;  
    location / {  
        proxy_pass https://127.0.0.1:5090;  
    }  
}
```

#### 2.1.2.4. Перезапуск служб

Выполнить команды:

```
systemctl restart nginx
```

```
systemctl restart hub
```

#### 2.1.2.5. Ошибки и решения

##### Ошибка kestrel dotnet

Для устранения переустановить самоподписанный сертификат приложения dotnet.

Зайти под пользователем, под которым стартует служба, и выполнить:

```
dotnet dev-certs https --clean
```

```
dotnet dev-certs https -t
```

#### 2.1.3. Настройка сервиса авторизации

##### 2.1.3.1. Настройка конфигурации фронтенд

Конфигурационный файл configuration.json размещается в папке \quarta-authentication-web-app\assets\configurations\clients\...

```
{  
    //Хост API Хаба, куда будут идти запросы к данным Хаба.  
    "serverUrl": "https://localhost:5090",  
    //Хост API ресурса, который является источником кадровых сведений  
    "staffServerUrl": "http://localhost:5080",  
}
```

```
//Настройка аутентификации
"authentication": {
  //Identity Provider, должно совпадать с значением поля "serverUrl"
  "authority": "https://localhost:5090",
  //адрес куда осуществлять переадресацию, если пользователь пытается
  //попасть в режим, которые требует наличие аутентифицированного
  //пользователя
  "login_uri": "https://localhost:1090/login",
  //адрес куда осуществляется переадресация после успешного входа в
  //систему.
  "redirect_uri": "https://localhost:1090/login-callback",
  //адрес "тихой" аутентификации. Используется для рефреша сессии без
  //необходимости требовать от пользователя повторного входа
  "silent_redirect_uri": "https://localhost:1090/silent-callback",
  //адрес куда осуществляется переадресация, когда пользователь выходит из
  //системы.
  "post_logout_redirect_uri": "https://localhost:1090/login",
  //идентификатор SPA-приложения
  "client_id": "admin",
  //тип используемых токенов
  "response_type": "id_token token",
  //перечень запрашиваемых ресурсов (API) системы
  "scope": "openid profile identityServer"
},
//настройка навигации
"navigation": {
  //URL по которому запрашивается перечень пунктов навигации
  "api": "https://localhost:5090/api/navigation",
  //идентификатор модуля навигации (проверяется по в разделе
  //"Navigation/Origins" бэкенда.
  "domain": "Auth"
}
}
```

### 2.1.3.2. Настройка конфигурации .Net сервиса

Конфигурационный файл appsettings.json размещается в папке \Clients\...  
Все строковые настройки **РЕГИСТРОЗАВИСИМЫЕ**.

```
{
  //Перечень строк подключения
  "ConnectionStrings": {
    "DefaultConnection": "" //Основная строка подключения. В этой БД
    //располагаются таблицы Хаба.
    "LogsConnection": "" //Строка подключения к БД, куда будут помещаться
    //логи безопасности,
```

```

"StaffConnection": "" //Строка подключения к ресурсу с кадровыми
сведениями, откуда осуществляется репликация справочника организаций.
},
// Адрес по которому поднимается портал внутри веб-сервера Kestrel,
входящий в состав dotnet.
"ApplicationUrls": [
  "https://localhost:5091",
  "http://localhost:5090"
],
// Настройка CROSS-ORIGIN-RESOURCE-SHARING: Перечень адресов
которым разрешено обращаться к API приложения.
"CorsOrigins": [
  "^http[s]?://\\/.x\\.x\\.x(:\\d{1,6})?$", //
Поддерживаются регулярные выражения
  "^http[s]?://\\/.localhost(:\\d{1,6})?$"
],
// Перечень сервисов, которые работают в фоне приложения
"HostedServices": {
  "Items": [
    {
      //Репликация справочника организаций
      "Key": "OrganizationReplication",
      "Enabled": true, // ВКЛ\ВЫКЛ
      "Interval": "00:05:00" //интервал репликации (5 минут)
    },
    {
      //Кеширование состояний гридов. Поле Interval отсутствует, значит оно
      выполняется только один раз, при старте приложения.
      "key": "GridStateCaching",
      "Enabled": true
    }
  ]
},
//Настройка логирования, выводимого в STDOUT.
"Logging": {
  "IncludeScopes": false, //вывод вспомогательных параметров при
логировании (не используется).
  //Настройка минимальных уровней сообщений, которые должны войти в
лог.
  //Возможные значения:
  // Trace = 0, Debug = 1, Information = 2, Warning = 3, Error = 4, Critical = 5,
  and None = 6.
  // Логируются сообщения от меньшего (Trace) к высшему (Critical)
  // Работают по принципу топики. Т.е. namespace кода, вызвавшего лог
  пишет лог всех namespace входящий в указанные.

```

```

"LogLevel": {
  "Default": "Warning",
  "System": "Warning",
  "Microsoft": "Warning"
},
//Настройка кастомизации.
"Customization": {
  // Управление кешированием грида.
  "GridState": {
    //Нужно ли делать обращение в БД за проверкой состояния грида, если
    такой не найден в кэше.
    //Работает в связке с фоновой операцией
    HostedServices/Items['GridStateCaching']
    "FallbackOnNoCache": false
  }
},
//Настройки навигации, используемые при построении дашборда.
"Navigation": {
  // Хосты, на которых располагаются модули системы.
  // в appsettings.navigation.json в поле "domain" для пунктов и групп меню
  указывается ключ из этого списка.
  "Origins": {
    "Auth": "https://x.x.x.x:1090",
    "Sophie": "https://x.x.x.x:1080",
    "Wage": "https://x.x.x.x:1070",
    "Buch": "https://x.x.x.x:1050"
  }
},
// Режим работы "Рабочих мест". На текущий момент поддерживается
только указанный ниже вариант.
"Workspace": {
  "OrgMode": "multi",
  "RoleMode": "multi"
},
// Возможность фильтрации данных по настраиваемым спискам
"RoleSettings": {
  "FilterByLists": false
},
//Настройки SMTP-клиента
"Smtp": {
  //Хост
  "Host": "qexch03.office.quarta-vk.ru",
  //Порт
  "Port": 465,

```

```

//Использовать ли SSL при отправке писем
"IsSsl": false,
//От чьего имени отправляются письма
"Email": "developer@quarta.su",
//Логин
"UserName": "developer",
//Пароль
"Password": "",
//Проверять сертификаты на отзыв
"CheckCertificateRevocation": false
},
//Настройки аутентификации
"Authentication": {
  "Authority": "https://x.x.x.x:1090", //Хост Identity Provider, сам Хаб
  "ApiName": "identityServer", //идентификатор ресурса (для запросов на API
  используя JWT из SPA-приложения в составе модуля Хаба)
  "ApiSecret": "", //Секретное слово.
  "ClaimType": "uri://schemas.quarta.su/permission-claim-type", //Ключ для
  прав доступа (не менять).
  //Настройка политик безопасности (в отношении новых паролей), уже
  созданным учеткам будет разрешено войти с их действующими паролями.
  "Password": {
    //Требуется ли наличие хотя одной цифры
    "RequireDigit": false,
    //Минимальная длина пароля
    "RequiredLength": 1,
    //Требуется ли хотя бы один спецсимвол
    "RequireNonAlphanumeric": false,
    //Требуется хотя бы одна заглавная буква
    "RequireUppercase": false,
    //Требуется хотя бы одна строчная буква
    "RequireLowercase": false,
    //Требование к количеству уникальных символов в пароле.
    "RequiredUniqueChars": 1
  },
  //Настройка блокировки учетных записей
  "AccountBlocking": {
    // Максимальное количество неудачных попыток входа (правильный
    логин, неправильный пароль)
    "MaxLoginAttempts": 2
  },
  //Настройка SPA-приложения. Указанные настройки заливаются в БД при
  запуске приложения.
  "Identity": {
    //идентификатор приложения

```

```

"ClientId": "admin",
//имя приложения
"ClientName": "admin",
//Адреса с которых разрешены запросы к API Хаба.
"Host": "https://x.x.x.x:1090 https://localhost:1090",
//Адреса редиректов куда разрешено вернуться после удачной
аутентификации
//Следует менять только Хост.
"CallbackUrl": [
  "https://x.x.x.x:1090/login-callback",
  "https://x.x.x.x:1090/silent-callback",
  "https://localhost:1090/login-callback",
  "https://localhost:1090/silent-callback"
],
//Адреса на которые разрешено вернуться после разлогинивания
"PostLogoutUrl": "https://x.x.x.x:1090/login https://localhost:1090/login",
//перечень доступных ресурсов (API) системы.
"Scope": "openid profile identityServer"
}
}

```

#### #### Автоматически заливаемые настройки

1. ``ApiName``, ``ApiSecret`` заливаются в БД в режим "Ресурсы" для обеспечения учета Хаба как доступного ресурса.

2. ``Identity`` - весь раздел заливается в БД в режим "Клиенты" для обеспечения доступности SPA-приложения к данным Хаба. Настройки, указанные в данном разделе должны соответствовать настройкам ``configuration.json`` в SPA-приложении.

3. Учетная запись админа. В случае если в БД отсутствует учетная запись ``admin\_user/admin``, она будет создана. Данной учетной записи присваиваются все разрешения, указанные в файле ``appsettings.claims.json`` и убрать их можно только напрямую из БД.

4. ``openid``, ``profile`` - системные клеймы, заливаются в случае отсутствия.

#### #### Прочие настройки

1. ``Identity/Scope`` - перечень требуемых ресурсов для работы приложения.

Указываемые значения должны присутствовать в режиме "Ресурсы" приложения (за исключением ``openid``, ``profile``).

Данные настройки следует рассматривать как запрашиваемую аудиенцию приложения (те API-приложения в которые SPA-приложение требует доступ).

Эти настройки проверяются Хабом во время авторизации SPA-приложения.

Если приложение пытается запросить аудиенцию, которая ему не разрешена Хабом, то во время авторизации, пользователь увидит, что его приложение невалидно и не сможет зайти в систему.

## **2.2. Установка и настройка сервиса файлового хранилища**

### **2.2.1. Подготовка СУБД**

Необходимо выполнить следующие действия:

- Создать файл /etc/yum.repos.d/mongodb-org-5.0.repo со следующим содержимым:

```
[mongodb-org-5.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-
org/5.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc
```

- Обновить информацию о репозитории и установить MongoDB, выполнив команды:

```
sudo yum update
```

```
sudo yum install -y mongodb-org
```

- Запустить службу MongoDB, выполнив команды:

```
systemctl start mongod
```

```
systemctl enable mongod
```

- Убедиться, что MongoDB слушает свой порт, выполнив команду:

```
netstat -tunlp | grep -i mongo
```

- Создать базу и collection в ней (пока что пустой), выполнив команды:

```
use filestorage
```

```
db.files.insertOne( { x: 1 } )
```

В результате возможен запуск на localhost без пароля.

## 2.2.2. Развёртывание .Net сервисов в linux

### 2.2.2.1. Установка .Net

Если сервисы устанавливаются на выделенный сервер, то выполнить действия, описанные в п. 2.1.2.1.

### 2.2.2.2. Настройка бэкенд-сервисов

Выполнить следующие действия:

- Копировать дистрибутив сервиса (\Дистрибутивы сервисов\Файловое хранилище\) на сервер (/usr/share/hosting/fs).
- При отсутствии создать пользователя, под которым будет запускать приложение. Пользователь должен иметь полный доступ к директории с приложением и права открывать сокеты.
- Создать службу /etc/systemd/system/filestorage.service. Указать директории, пути к файлам и приложениям, описание и указать пользователя под кем будет запускать приложение:

```
[Unit]
Description=[File Storage] Файловый сервер

[Service]
WorkingDirectory=/usr/share/hosting/fs
ExecStart=/usr/bin/dotnet
    /usr/share/hosting/fs/Quarta.FileStorage.NetCore.Web.dll
Restart=always
RestartSec=10
SyslogIdentifier=dotnet-fs
User=%ИМЯ СИСТЕМНОГО ПОЛЬЗОВАТЕЛЯ%
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false
Environment=DOTNET_CLI_TELEMETRY_OPTOUT=false

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

- Перезагрузить информацию о сервисах, выполнив команду:  
systemctl daemon-reload

### 2.2.2.3. Настройка nginx

Если сервисы устанавливаются на выделенный сервер, то выполнить действия по установке Nginx и выпуску сертификата pfx, описанные в п. 2.1.2.3.

#Добавить в server прокси к приложению:

```
server {
    listen    1040;
```



```
location / {  
    proxy_pass http://127.0.0.1:5040;  
}  
}
```

#### **2.2.2.4. Перезапуск служб**

Выполнить команды:

```
systemctl restart nginx  
systemctl restart filestorage
```

#### **2.2.3. Настройка сервиса файлового хранилища**

##### **2.2.3.1. Настройка конфигурации .Net сервиса**

Конфигурационный файл **appsettings.json** размещается в директории  
\\Clients\\...

```
{  
  // Хост сервиса  
  "ApplicationUrls": [  
    "http://X.X.X.X:5048"  
  ],  
  "MongoDB": {  
    // Адрес сервера СУБД  
    "ConnectionString": "mongodb://192.168.50.69:27017",  
    // Наименование базы данных  
    "Catalog": "rc",  
    "DefaultCollection": "files"  
  },  
  // Параметры логирования  
  "Logging": {  
    "LogLevel": {  
      "Default": "Information",  
      "Microsoft": "Warning",  
      "Microsoft.Hosting.Lifetime": "Information"  
    }  
  },  
  // Перечень хостов, которым разрешено обращаться к сервису  
  "AllowedHosts": "*" }  
}
```

### **2.3. Установка и настройка сервиса интерактивных рабочих столов и аналитики**

#### **2.3.1. Подготовка СУБД**

Выполнить следующие действия:

- Установить на сервер СУБД PostgresPro либо PostgreSql версии не ниже 14;
- Подключиться к серверу, например, с помощью графического клиента pgAdmin;
- Создать БД BI.

Установка СУБД на примере дистрибутива PostgresPro Enterprise 11 из ISO образа:

1. Подключить ISO образ дистрибутива:

```
mount PostgresProEntCert-11.12.1.iso /mnt/cdrom/ -o loop
```

В файл репозитория добавить iso образ /etc/yum.repos.d/RedOS-Sources.repo

```
[cdrom]
```

```
name= CDROM
```

```
baseurl=file:///mnt/cdrom/redos/7.2/os/x86_64/rpms
```

```
gpgkey=file:///mnt/cdrom/keys/GPG-KEY-POSTGRESPRO
```

```
enabled=1
```

```
gpgcheck=1
```

2. Далее обновить информацию о пакетах и устанавливаем СУБД:

```
yum update
```

```
yum install postgrespro-ent-11
```

3. Переключиться на пользователя postgres:

```
su postgres
```

4. Подключиться к консоли postgresql:

```
psql
```

5. Создать базы:

```
create database bi.
```

## **2.3.2. Развёртывание .Net сервисов в linux**

### **2.3.2.1. Установка .Net**

Если сервисы устанавливаются на выделенный сервер, то выполнить действия, описанные в п. 2.1.2.1.

### **2.3.2.2. Настройка бэкенд-сервисов**

Выполнить следующие действия:

- Копировать дистрибутив сервиса (\Дистрибутивы сервисов\Бизнес аналитика\) на сервер (/usr/share/hosting/bi).
- При отсутствии создать пользователя, под которым будет запускать приложение. Пользователь должен иметь полный доступ к директории с приложением и права открывать сокеты.
- Создать службу /etc/systemd/system/bi.service. Указать директории, пути к файлам и приложениям, описание и указать пользователя под кем будет запускать приложение:

```
[Unit]
Description=[File Storage] Бизнес-Аналитика

[Service]
WorkingDirectory=/usr/share/hosting/bi
ExecStart=/usr/bin/dotnet /usr/share/hosting/bi/Quarta.BI.WebApi.dll
Restart=always
RestartSec=10
SyslogIdentifier=dotnet-bi
User=%ИМЯ СИСТЕМНОГО ПОЛЬЗОВАТЕЛЯ%
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false
Environment=DOTNET_CLI_TELEMETRY_OPTOUT=false
Environment=ASPNETCORE_ENVIRONMENT=Production

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

- Перезагрузить информацию о сервисах, выполнив команду:  
systemctl daemon-reload

### 2.3.2.3. Настройка nginx

Если сервисы устанавливаются на выделенный сервер, то выполнить действия по установке Nginx и выпуску сертификата pfx, описанные в п. 2.1.2.3.

#Добавить в server прокси к приложению:

```
server {
    listen    1100;
    location / {
        proxy_pass http://127.0.0.1:5100;
    }
}
```

### 2.3.2.4. Перезапуск служб

Выполнить команды:

```
systemctl restart nginx
```

```
systemctl restart bi
```

### 2.3.3. Настройка сервиса интерактивных рабочих столов и аналитики

#### 2.3.3.1. Настройка конфигурации .Net сервиса

Конфигурационный файл **appsettings.json** размещается в директории \Clients\...

```
{
  "Logging": {
    "LogLevel": { //Уровни логирования
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "ConnectionStrings": { //Строка подключения к БД, созданной в п.2.4.1.1
    "DefaultConnection": "Server=localhost; Database=bi; User ID=postgres; Password=postgres"
  },
  "ApplicationUrls": [ //Хост занимаемый веб-сервером Kestrel
    "https://localhost:5100"
  ],
  "CorsOrigins": [ // Настройка CROSS-ORIGIN-RESOURCE-SHARING:
    Перечень адресов которым разрешено обращаться к API приложения.
    "^http[s]?://192\\.168\\.48\\.124(:\\d{1,6})?$",
    "^http[s]?://localhost(:\\d{1,6})?$"
  ]
}
```

## 2.4. Установка и настройка сервиса оболочки

### 2.4.1. Выполнить следующие действия:

– Копировать дистрибутив сервиса (Дистрибутивы сервисов\Оболочка\)) на сервер (/usr/share/hosting/shell).

### 2.4.2. Настройка nginx

Если сервисы устанавливаются на выделенный сервер, то выполнить действия по установке Nginx и выпуску сертификата pfx, описанные в п. 2.1.2.3.

#Добавить в server прокси к приложению:

```
server {
    listen    6200;
```

```
location / {
    root /usr/share/hosting/shell;
    try_files $uri /index.html;
}
}
```

### 2.4.3. Настройка сервиса оболочки

Конфигурационный файл **configuration.json** размещается в директории \config\...

```
{
  "modules": {
    "bi": {
      "serverUrl": "https://192.168.48.124:1100", //хост размещения сервиса BI
      (п. 2.4)
      "path": "bi", //адрес корня, в которое будет размещаться сервис BI.
      "remoteEntry": "https://192.168.48.124:1100/remoteEntry.js", //адрес
      загрузки файла, содержащего описание приложения BI
      "exposedModule": "./FeaturesModule", //экспортируемый корневой
      модуль
      "key": "FeaturesModule", //Имя модуля
      "name": "BI" // имя (описание) сервиса
    },
  },
  //секция настроек аутентификации и соединения с сервисом авторизации
  (п. 2.1)
  "authentication": {
    "authority": "https://192.168.48.124:1090",
    "redirect_uri": "http://localhost:6200/callback",
    "post_logout_redirect_uri": "http://localhost:6200/login",
    "silent_redirect_uri": "http://localhost:6200/silent-callback",
    "client_id": "shell",
    "response_type": "id_token token",
    "scope": "openid identityServer bi",
    "useLocalStorage": false,
    "automaticSilentRenew": true
  }
}
```

### **3. ОБНОВЛЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

#### **3.1. Обновление сервиса авторизации**

- 1) Распаковать дистрибутив с обновлением во временную папку.
- 2) Перенести существующие настройки (значения) из старого конфигурационного файла в новый.
  - Конфигурационный файл бэкэнда сервиса находится в директории `./Clients/Default/appsettings.json`
  - Конфигурационный файл фронтэнда сервиса находится в директории `./wwwroot/app/assets/configurations/clients/default/configuration.json`
- 3) Удалить директорию с сервисом, на место удаленной директории перенести директорию с настроенным обновлением.
- 4) Выдать разрешение на папку для пользователя, под которым настроена служба. Пользователь должен быть владельцем папки с правами на чтение / запись / выполнение.
- 5) Перезапустить службу.

#### **3.2. Обновление сервиса файлового хранилища**

- 1) Распаковать дистрибутив с обновлением во временную папку.
- 2) Перенести существующие настройки (значения) из старого конфигурационного файла в новый.
  - Конфигурационный файл бэкэнда сервиса находится в директории `./appsettings.json`
- 3) Удалить директорию с сервисом, на место удаленной директории перенести директорию с настроенным обновлением.
- 4) Выдать разрешение на папку для пользователя, под которым настроена служба. Пользователь должен быть владельцем папки с правами на чтение / запись / выполнение.
- 5) Перезапустить службу.

#### **3.3. Обновление сервиса интерактивных рабочих столов и аналитики**

- 1) Распаковать дистрибутив с обновлением во временную папку.
- 2) Перенести существующие настройки (значения) из старого конфигурационного файла в новый.

- Конфигурационный файл бэкэнда сервиса находится в директории `./Clients/Default/appsettings.json`
  - Конфигурационный файл фронтэнда сервиса находится в директории `./quarta-bi-web-app/config/clients/default/configuration.json`
- 3) Удалить директорию с сервисом, на место удаленной директории перенести директорию с настроенным обновлением.
  - 4) Выдать разрешение на папку для пользователя, под которым настроена служба. Пользователь должен быть владельцем папки с правами на чтение / запись / выполнение.
  - 5) Перезапустить службу.

### **3.4. Обновление сервиса оболочки**

- 1) Распаковать дистрибутив с обновлением во временную папку.
- 2) Перенести существующие настройки (значения) из старого конфигурационного файла в новый.
  - Конфигурационный файл фронтэнда сервиса находится в директории `./config/configuration.json`
- 3) Удалить директорию с сервисом, на место удаленной директории перенести директорию с настроенным обновлением.
- 4) Выдать разрешение на папку для пользователя, под которым запущен публикующий этот сервис HTTP-сервер. Пользователь должен быть владельцем папки с правами на чтение / запись / выполнение.

## 4. ПРОВЕРКА, ВОССТАНОВЛЕНИЕ И ПОДДЕРЖАНИЕ РАБОТОСПОСОБНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### 4.1. Методы проверки работоспособности рабочих станций

Чтобы проверить работоспособность рабочей станции (клиентской части Системы), требуется выполнить следующие действия:

- запустить рабочую станцию;
- запустить web-браузер;
- в адресной строке ввести адрес сервера Системы;
- в окне авторизации ввести логин и пароль (login: admin\_user, пароль: admin), нажать на кнопку «Войти».

Система работоспособна, если в результате выполнения действий отображается главная страница.

### 4.2. Методы проверки работоспособности сервера

Чтобы проверить работоспособность сервера, требуется выполнить следующие действия:

- убедиться в том, что службы Postgres Pro, nginx, MongoDB и сервисов Системы работают.

- Для этого убеждаемся, что их статус **active** командами:

`systemctl status nginx`

```
● nginx.service - The nginx
   Loaded: loaded (/usr/lib/
   Active: active (running)
```

`systemctl status postgrespro-ent-14`

```
● postgrespro-ent-14.service
   Loaded: loaded (/usr/lib/s
   Active: active (running) s
   Process: 100577 ExecStartP
```

`systemctl status auth` (в зависимости от того, какое имя службе модуля хаба выдали);

```
● auth.service - [Authentication]
   Loaded: loaded (/etc/systemd/s
   Active: active (running) since
   Main PID: 122667 (dotnet)
```

`systemctl status fs` (в зависимости от того, какое имя службе модуля файлового хранилища выдали);

```
● fs.service - [File Storage]
   Loaded: loaded (/etc/syst
   Active: active (running)
   Main PID: 1044 (dotnet)
```



`systemctl status mongod` (в зависимости от того, какое имя службе модуля файлового хранилища выдали).

```
● mongod.service - MongoDB Daemon
   Loaded: loaded (/usr/lib/systemd/systemd; vendor preset: enabled)
   Active: active (running) since Wed 2018-08-08 12:00:00 MSK; 1min 1s ago
     Docs: https://docs.mongodb.org/manual/
```

– убедиться в том, что Postgres Pro доступен. Для этого достаточно выполнить команду `psql` и в запущившемся интерактивном терминале Postgres Pro выполнить команду: `\conninfo`. При успешном соединении отобразится соответствующее сообщение, например:

```
postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in "/var/run/postgresql" at port "5432".
postgres=#
```

– убедиться в том, что Postgres Pro, nginx, MongoDB и Система открыли все настроенные порты:

`netstat -tnulp`

```
[root@DTS-KORUPCIA-APP ~]# netstat -tnulp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:5000          0.0.0.0:*                LISTEN      51763/dotnet
tcp        0      0 127.0.0.1:5001          0.0.0.0:*                LISTEN      51763/dotnet
tcp        0      0 0.0.0.0:1040            0.0.0.0:*                LISTEN      15021/nginx: master
tcp        0      0 127.0.0.1:5040          0.0.0.0:*                LISTEN      1044/dotnet
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN      1147/sshd
tcp        0      0 0.0.0.0:5432            0.0.0.0:*                LISTEN      100581/postgres
tcp        0      0 0.0.0.0:1080            0.0.0.0:*                LISTEN      15021/nginx: master
tcp        0      0 127.0.0.1:5080          0.0.0.0:*                LISTEN      1043/dotnet
tcp        0      0 127.0.0.1:17017         0.0.0.0:*                LISTEN      1347/mongod
tcp        0      0 127.0.0.1:25            0.0.0.0:*                LISTEN      1336/master
tcp        0      0 0.0.0.0:1050            0.0.0.0:*                LISTEN      15021/nginx: master
tcp        0      0 0.0.0.0:1090            0.0.0.0:*                LISTEN      15021/nginx: master
tcp        0      0 127.0.0.1:5090          0.0.0.0:*                LISTEN      1041/dotnet
```

### 4.3. Методы проверки работоспособности базы данных

В проверку работоспособности базы данных входит:

- 1) проверка физической целостности базы данных;
- 2) проверка сохранения введенных данных.

#### 4.3.1. Проверка физической целостности базы данных

1) Проверка физической целостности базы данных проводится подсчётом контрольных сумм на файлах баз данных кластера.

Для этого следует предварительно выключить службу СУБД:

`systemctl stop postgrespro-ent-14`

Запустить проверку:

```
pg_verify_checksums -D /var/lib/pgpro/ent-14/data/
```

```
Проверка контрольных сумм завершена
Версия контрольных сумм данных: 1
Просканировано файлов: 6147
Просканировано блоков: 13774
Неверные контрольные суммы: 0
```

2) Проверить целостность базы данных mongodb

Для этого подключиться к mongo и выполнить:

```
use filestorage
```

где filestorage наше имя базы данных.

Открыть список коллекций:

```
db.getCollectionInfos();
```

Далее для каждой коллекции выполнить проверку

```
db.files.validate({full:true})
```

В результате valid должен иметь значение true

```
db.files.validate({full:true})
{
  "ns" : "filestorage.files",
  "nInvalidDocuments" : 0,
  "nrecords" : 90,
  "nIndexes" : 1,
  "keysPerIndex" : {
    "_id_" : 90
  },
  "indexDetails" : {
    "_id_" : {
      "valid" : true
    }
  },
  "valid" : true,
  "repaired" : false,
  "warnings" : [ ],
  "errors" : [ ],
  "extraIndexEntries" : [ ],
  "missingIndexEntries" : [ ],
  "corruptRecords" : [ ],
  "ok" : 1
}
```

#### **4.3.2. Проверка сохранения введенных данных**

Для проверки сохранения данных, вносимых в базу данных Системы через браузер, нужно выполнить следующие действия:

- 1) запустить рабочую станцию;
- 2) открыть браузер;
- 3) в адресной строке ввести адрес сервера Системы;
- 4) в окне авторизации ввести логин и пароль, нажать на кнопку «Войти»;
- 5) на главной странице выбрать режим **Противодействие коррупции – Учет сообщений о гражданах, замещавших в ОГВ должности ГГС**;
- 6) нажать на кнопку добавления записи, в форме записи заполнить необходимые поля, прикрепить файл с помощью кнопки «Выбрать файл» и выполнить сохранение записи;
- 7) закрыть браузер;
- 8) повторно выполнить пп. 2-5;
- 9) в списке сообщений найти добавленную запись, открыть для просмотра, открыть прикрепленный файл кликом по гиперссылке;
- 10) закрыть форму просмотра записи, удалить запись;

Сохранение добавленной записи, в том числе прикрепленного файла, свидетельствует о работоспособности БД Системы.

#### **4.4. Методы восстановления работоспособности сервера**

Работоспособность сервера, в случае его отказа, восстанавливается специалистами из подразделения технической поддержки. Если технических неисправностей в оборудовании сервера не обнаружено и операционная система сервера работает без сбоев, то для восстановления его работоспособности рекомендуется переустановить серверную часть Системы.

## 4.5. Методы восстановления работоспособности базы данных

### 4.5.1. Методы восстановления работоспособности базы данных под управлением СУБД Postgres Pro 14

Для обеспечения возможности восстановления базы данных (например, поврежденной) администратор должен периодически выполнять её резервное копирование.

#### 4.5.1.1. Резервное копирование базы данных

1). Подключиться под пользователем **postgres**:

```
su postgres
```

2). Создать резервную копию базы данных «*database*» (здесь подставить имя исходной базы данных) в файл по пути **/tmp/database.bak**:

```
pg_dump database > /tmp/database.bak
```

В результате процедуры по указанному пути будет создан файл резервной копии.

**Рекомендация.** Для гарантирования сохранности данных файл рекомендуется скопировать (средствами операционной системы) на внешнее устройство (CD-диск, флэш-карту или др.).

#### 4.5.1.2. Восстановление базы данных

1) Подключиться под пользователем **postgres**:

```
su postgres
```

2) Восстановить базу «*database*» из файла бэкапа **/tmp/database.bak**:

```
psql database < /tmp/database.bak
```

### 4.5.2. Методы восстановления работоспособности базы данных под управлением СУБД MongoDB

#### 4.5.2.1. Резервное копирование базы данных

Для резервного копирования на сервере необходимо запустить процедуру **mongodump**. В качестве параметров следует указать порт (**port**), на котором запущена служба, базу данных (**db**) и путь к директории (**out**):

```
mongodump --port=17017 --db filestorage --out=/tmp/mongo
```

В результате процедуры по указанному пути будет создана директория с резервной копией базы данных.

#### **4.5.2.2. Восстановление базы данных**

Для резервного копирования на сервере необходимо запустить процедуру mongorestore. В качестве параметров указываем порт (port) и директорию с бэкапом (dir).

```
mongorestore --port=17017 --dir=/tmp/mongo/
```

#### **4.6. Методы поддержания целостности базы данных**

Целостность базы данных обеспечивается встроенными средствами СУБД Postgres Pro 14 и СУБД MongoDB 5.

Кроме того, целостность баз данных Системы обеспечивается на этапе ввода данных посредством заданных разработчиком правил.

#### **4.7. Методы поддержания безопасности базы данных**

Безопасность базы данных обеспечивается встроенными средствами СУБД Postgres Pro 14, СУБД MongoDB 5 и операционными системами серверов.

#### **4.8. Методы диагностирования проблем обновления баз данных**

Обновление баз данных осуществляются через утилиту Quarta.Migrator.exe

В процессе прогона миграционных скриптов и обновления функций, триггеров и др. в случае возникновения ошибок утилита логирует проблемы.

##### **4.8.1. Диагностирование проблем в работе сервисов**

В процессе работы сервисы записывают ошибки в лог.

Для диагностирования проблем необходим SSH Клиент (например, Putty или аналог).

- 1) Подключиться к серверу, на котором осуществляется хостинг проблемного сервиса.

Для входа требуется:

- 2) Получить идентификатор интересующего сервиса.

```
docker ps -a
```

Пример:

```
CONTAINER ID    IMAGE
5fa324f5c2f7   harbor.quarta.su/app-documentstorage/frontend:latest
a5998baf3ecd   harbor.quarta.su/app-documentstorage/app:latest
40ad2cab9706   harbor.quarta.su/app-integration-gosduma/app:latest
20305c5b0349   harbor.quarta.su/app-integration-gosduma/frontend:latest
ea3db8e85d49   harbor.quarta.su/app-notification/frontend:latest
32f44cc86e5d   harbor.quarta.su/app-notification/app:latest
da9c4c91f3f2   harbor.quarta.su/app-staffstructure/app:latest
4f5c1b57cc5f   harbor.quarta.su/app-staffstructure/frontend:latest
```

Например, необходимы логи сервиса «Хранение и оборот документов». Его имя: App-documentstorage, значит, идентификатор: a5998baf3ecd.

Если docker не используется, то логи доступны через команды вида

```
journalctl -u %наименование сервиса% -n 100 -f
```

- 3) Получить логи сервиса с помощью команды команда:

## **docker logs a5998baf3ecd**

```

warn: Microsoft.EntityFrameworkCore.Model.Validation[10022]
    Entity 'Person' has a global query filter defined and is the required end of a relationship with the entity 'PersonStructuralUnit'. This may lead to unexpected results when the required entity is filtered out. Either configure query filters for both entities in the navigation. See https://go.microsoft.com/fwlink/?linkid=2131316 for more information.
Quarta.Migrator запускается...
[OK] Проверка соединения с базой... успешно.
Quarta.Migrator готов к работе.
Запуск миграции SQL.
Миграция запущена.
Получение заархивных миграций... готово!
Получение миграционных скриптов...
Основной комплект: ./Migrations/Migrations
Миграция 0 MigrationInfo 20221228_001_kuznetsov_vlter_table_route_history.sql... добавлена... Выполнение операции прервано вследствие ошибки. 42703: столбец "document_executor_id" в таблице "document_route_history" не существует
crit: Microsoft.AspNetCore.Hosting.Diagnostics[6]
    Application startup exception
    System.Exception: Мигратор завершил работу с ошибкой.
    at Quarta.DS.Migrator.Builder.RunMigrations(IApplicationBuilder builder) in /app/Quarta.DS.Migrator/Builder.cs:line 67
    at Quarta.DS.WebApi.Startup.Configure(IApplicationBuilder builder, IWebHostEnvironment env) in /app/Quarta.DS.WebApi/Startup.cs:line 110
    at System.RuntimeMethodHandle.InvokeMethod(Object target, Span`1 arguments, Signature, Boolean constructor, Boolean wrapExceptions)
    at System.Reflection.RuntimeMethodInfo.Invoke(Object obj, BindingFlags invokeAttr, Binder binder, Object[] parameters, CultureInfo culture)
    at Microsoft.AspNetCore.Hosting.ConfigureBuilder.Invoke(Object instance, IApplicationBuilder builder)
    at Microsoft.AspNetCore.Hosting.ConfigureBuilder.<>c__DisplayClass0_0.b__0(IApplicationBuilder builder)
    at Microsoft.AspNetCore.Hosting.GenericWebHostBuilder.<>c__DisplayClass15_0.b__1(IApplicationBuilder app)
    at Microsoft.AspNetCore.Mvc.Filters.MiddlewareFilterBuilderStartupFilter.<>c__DisplayClass0_0.b__0(MiddlewareFilterBuilder(IApplicationBuilder builder))
    at Microsoft.AspNetCore.Hosting.FilteringStartupFilter.<>c__DisplayClass0_0.b__0(IApplicationBuilder builder)
    at Microsoft.AspNetCore.Hosting.GenericWebHostService.StartAsync(CancellationToken cancellationToken)
Unhandled exception. Microsoft.AspNetCore.Hosting.WebHostException: Мигратор завершил работу с ошибкой.

```

В логах можно отследить предупреждения и ошибки работы .Net-сервисов в зависимости от настройки секции Logging конфигурационного файла appsettings.json.

Настройка фиксации только ошибок:

```
"Logging": {  
  "LogLevel": {  
    "Default": "Error",  
    "System": " Error" } },
```

Настройка логирования предупреждений:

```
"Logging": {  
  "LogLevel": {  
    "Default": "Warning",  
    "System": "Warning" } },
```

Настройка логирования максимальной информации (не рекомендуется,  
т.к. лог будет перегружен):

```
"Logging": {  
  "LogLevel": {  
    "Default": "Info",  
    "System": " Info " } },
```

## 5. АДМИНИСТРИРОВАНИЕ СИСТЕМЫ

Задачи администрирования Системы выполняются в сервисе авторизации (Hub).

Сервис авторизации предназначен для управления доступом к системе, а также настройки некоторых параметров функционирования системы.

Сервис включает следующие режимы:

- Пользователи;
- Роли;
- Клиенты;
- Ресурсы;
- События аутентификации;
- События безопасности.

### 5.1. Доступ к сервису

Для получения доступа к сервису авторизации необходимо запустить Интернет-браузер.

1) в адресной строке ввести адрес сервера. Адрес сервера имеет следующий вид - http://адрес\_веб-сервера: порт (пример: http://localhost:8080);

2) в окне авторизации ввести логин и пароль пользователя с правами администратора (*по умолчанию доступен пользователь «test» с паролем «1»*), нажать на кнопку **«Войти»**:

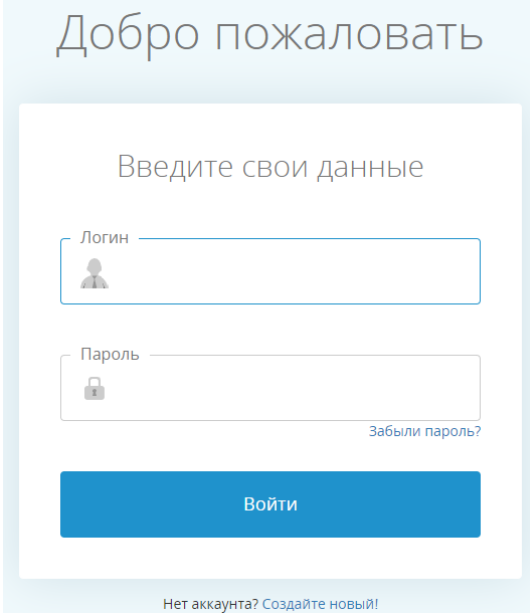


Рис. 5.1. Окно авторизации

Для перехода в раздел администрирование нужно нажать кнопку профиля и в открывшемся меню выбрать **Настройки**.



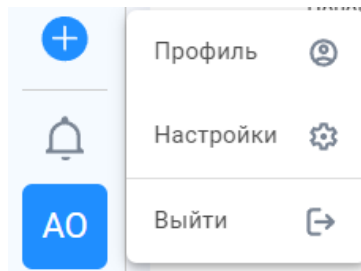


Рис. 5.2

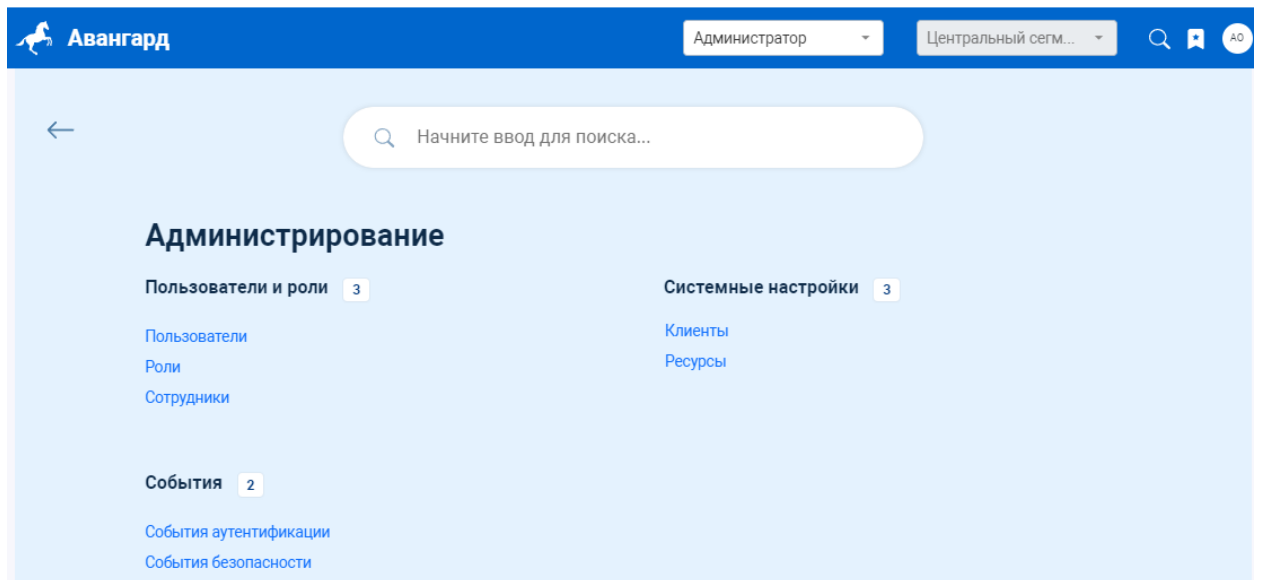


Рис. 5.3. Главная страница сервиса администрирования

## 5.2. Управление доступом

К задачам управления доступом относятся:

- Настройка фронтенд-приложений и бэкенд-сервисов системы (клиентов, ресурсов);
- Настройка прав доступа к режимам Системы (формирование ролей, списка пользователей и сотрудников).

При запуске Системы выполняется сначала верификация клиента, потом авторизация пользователя. В ходе работы пользователя клиент (приложение) взаимодействует с ресурсами и (сервисами).

### 5.2.1. Клиенты

Нажать ЛКМ на подраздел «Системные настройки» Клиенты. Откроется списочная форма Клиент-Приложений.

Идентификатор	Наименование	Scopes	Redirect	Post logout redirect	CORS
accounting_rc	accounting-rc	identityServer profile openid accounting	http://localhost:1050/#/... callback? http://localhost:1050/#/... http://192.168.48.124:1... callback? http://192.168.48.124:1...	http://192.168.48.124:1... http://localhost:1050	http://192.168.48.124:1... http://localhost:1050
admin	admin	identityServer profile openid	http://192.168.48.124:1... callback https://192.168.48.124:1... callback	http://192.168.48.124:1... https://192.168.48.124:1...	http://192.168.48.124:1... https://192.168.48.124:1...

Рис. 5.4. Клиенты

Списочная форма Клиент-Приложений содержит следующий набор сведений о настройках приложений:

- 1) **Идентификатор** – уникальный строковый идентификатор приложения;
- 2) **Наименование** – Читабельное представление (описание) приложения;
- 3) **Scopes** – Перечень разрешений которые может запрашивать приложение;
- 4) **Redirect** – Перечень разрешенных переадресаций пользователя после прохода аутентификации;
- 5) **Post Logout redirect** – Перечень разрешенных переадресаций пользователя после того, как пользователь выходит из системы;
- 6) **CORS** – Перечень хостов с которых приложению разрешено осуществлять запросы на аутентификацию/авторизацию пользователя.

Для добавления нового клиента необходимо нажать ЛКМ на «+» над списочной формой клиентов. Будет открыта форма внесения данных о клиенте с параметрами настройки его подключения:

**Управление клиентом admin**

ID приложения \*

admin

Наименование приложения \*

admin

Доступные ресурсы (API, OpenId, Profile)

identityServer profile openid

Redirect Uri \*

http://192.168.48.124:1098/login-callback  
https://192.168.48.124:1098/login-callback

Post Logout Redirect Uri \*

http://192.168.48.124:1098/login  
https://192.168.48.124:1098/login

CORS origins

http://192.168.48.124:1098  
https://192.168.48.124:1098

Время жизни токена (сек) \*

86400

Тип токена

Reference

Сохранить Заккрыть

Рис. 5.5. Описание клиента

Поля повторяют ранее описанные поля для списочной формы. Для всех полей, где возможен ввод нескольких значений, их порядок не имеет значения. Из особенностей отдельных полей:

**Доступные ресурсы** – каждый клиент должен содержать обязательные параметры: openid, profile. Остальные значения должны браться из списка идентификаторов Ресурсов, заполненных в режиме «Ресурсы». Разделителем всех значений является символ «пробел».

**Время жизни токена** – заполняется исходя из требований к безопасности. Чем меньше интервал жизни токена, тем чаще осуществляется его автоматическое обновление, однако не рекомендуется делать его короче среднего времени сессии пользователя.

**Тип токена** – влияет на то как будет передаваться токен авторизации из SPA-Приложения в Ресурс:

- 1) Reference: передаваться будет только обезличенный хэш-ключ, который требует подтверждения от Хаба;
- 2) JWT – самодостаточный токен.

Коды аутентификации — ключи для автоматизации отдельных аспектов процесса разработки, не предусмотрены для использования в реальных сценариях.

Требования к полям **Redirect Uri**, **Post Logout Redirect Uri** разрешения адреса указываются в формате URL без указания возможных параметров или якорей: <схема>: [//<хост>[:<порт>]][/URL-путь>]

Пример корректно написанных URL:

https://auth.gd-workspace.ru/callback

http://192.168.48.124:1090/login

https://gd-workspace.ru

Требования к полю CORS: разрешенный адрес должен содержать только элементы: :<схема>: [//<хост>[:<порт>]] (т.е. без путей, параметров и якорей).

Пример:

https://gd-workspace.ru

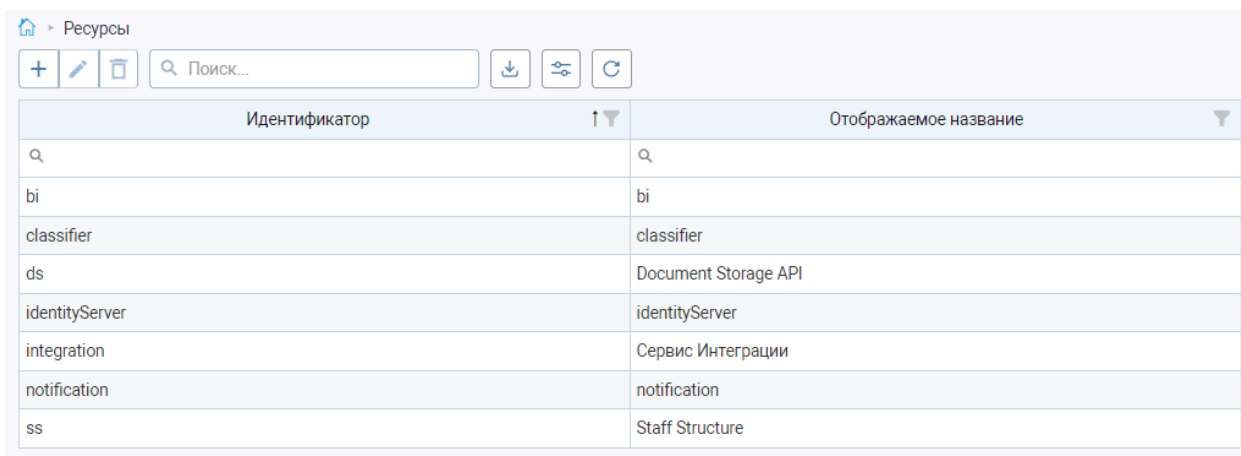
http://auth-test.ru

Для сохранения нового клиента, или редактирования имеющегося, нужно нажать кнопку «Сохранить». В случае успешного сохранения будет выполнена переадресация на списочную форму.

### 5.2.2. Ресурсы

Нажать ЛКМ на подраздел «Системные настройки» - «Ресурсы».

Откроется списочная форма Ресурсов:



Идентификатор	Отображаемое название
bi	bi
classifier	classifier
ds	Document Storage API
identityServer	identityServer
integration	Сервис Интеграции
notification	notification
ss	Staff Structure

Рис. 5.6. Ресурсы

Списочная форма Ресурсов содержит следующий набор сведений:

**Идентификатор** — уникальный строковый идентификатор ресурса;

**Отображаемое название** — читабельное представление (описание) приложения.

### 5.2.2.1. Создание нового ресурса

Необходимо нажать ЛКМ на «+» над списочной формой клиентов. Будет открыта форма заполнения данных:

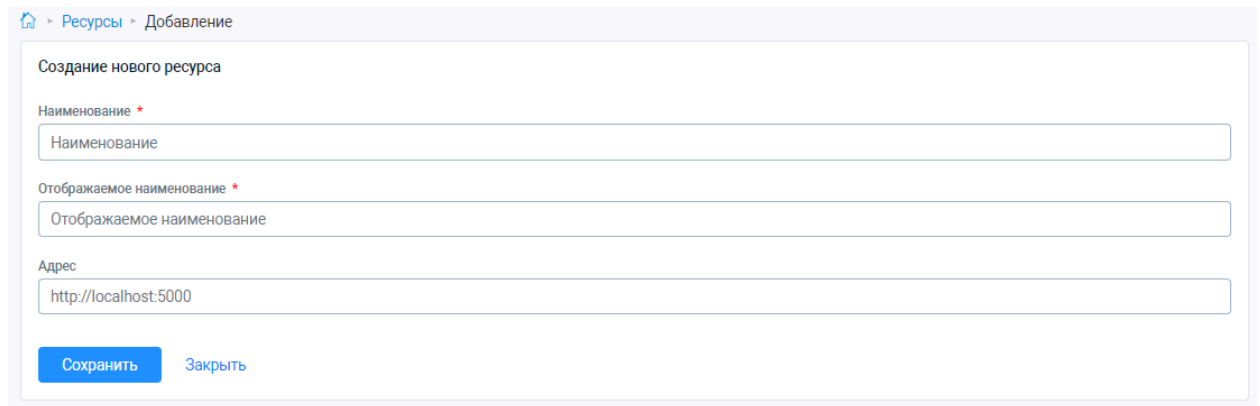


Рис. 5.7. Создание нового ресурса

Из особенностей полей:

Наименование не должно содержать пробелов в названии.

Адрес должен соответствовать формату <схема>: [//<хост>[:<порт>]]

Например: <https://gd-workspace.ru>

В случае успешного сохранения будет выполнена переадресация на карточку Ресурса.

Пример:



Рис. 5.8. Описание ресурса

### 5.2.2.2. Раздел API

Если Поле «Адрес» было заполнено корректно и Ресурс активен (запущен на сервере и имеет действующий ключ аутентификации), в данном режиме будет отражаться перечень его URL, которые могут быть использованы приложениями для обращения к данным в соответствии с их бизнес-логикой.

Кнопка «Перезагрузить» позволяет вручную запросить данный список у Ресурса. Полезен для случаев, когда спецификации Ресурса изменяются и эти изменения не были занесены в Хаб во время запуска приложения.

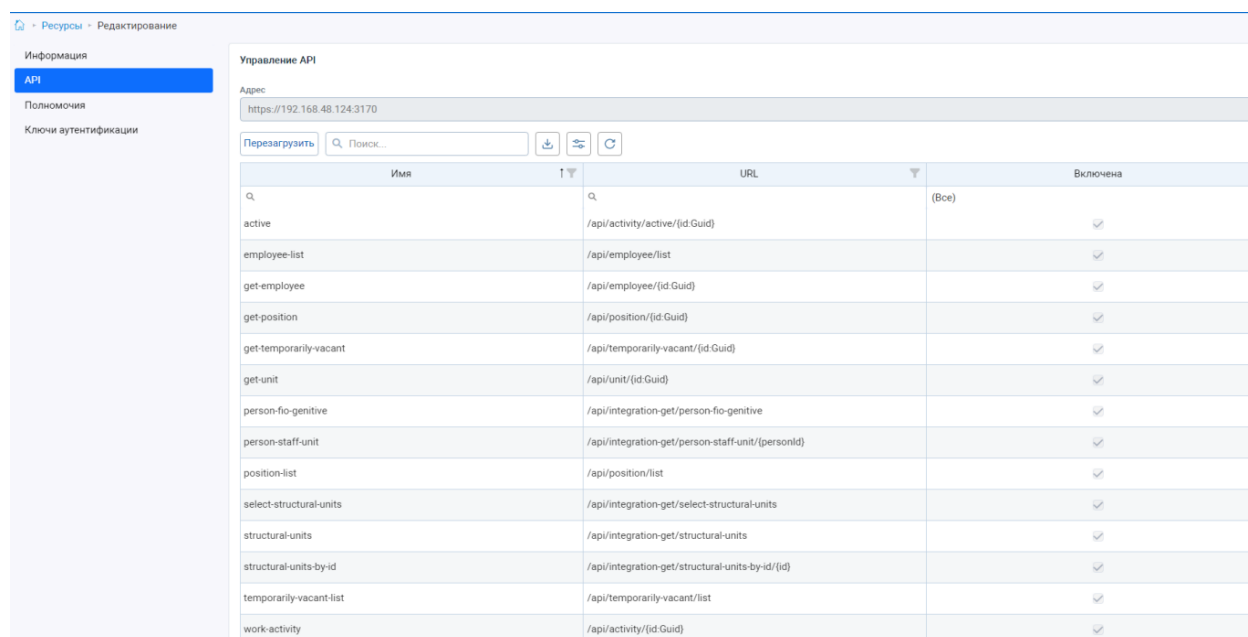


Рис. 5.9. Управление API

### 5.2.2.3. Раздел Полномочия

В данном разделе описывается справочник прав доступа, которые использует Ресурс при обращении к своему API. Эти данные обновляются при каждом запуске Ресурса на сервере.

Носит уведомительный характер и используется как источник данных при назначении Полномочий ролям в режиме «Роли».

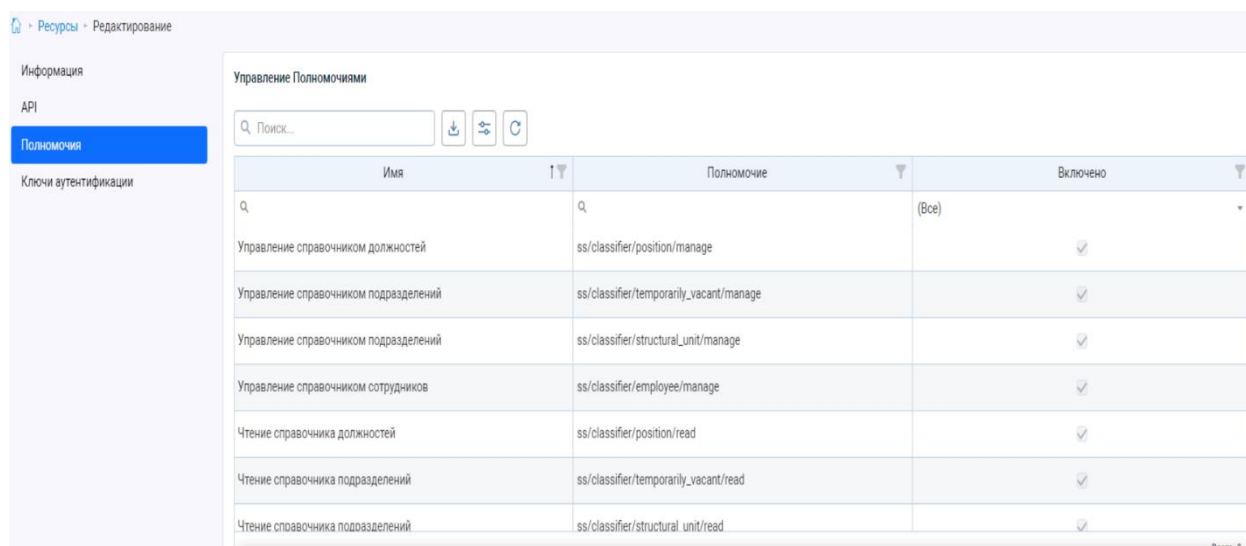


Рис. 5.10. Управление Полномочиями

### Раздел Ключи аутентификации:

В данном разделе вносятся ключи, которые могут использоваться Ресурсом при следующих обращениях к Хабу:

- 1) Верификация токена, полученного от Клиента;
- 2) Межсерверная передача данных (Полномочия);
- 3) Логирование и пр.

Ресурс может иметь больше одного действующего ключа, это позволяет выполнять постепенный rollout-ключей в случае кластерного развертывания Ресурса (полезно для сервисов с высокой нагрузкой).

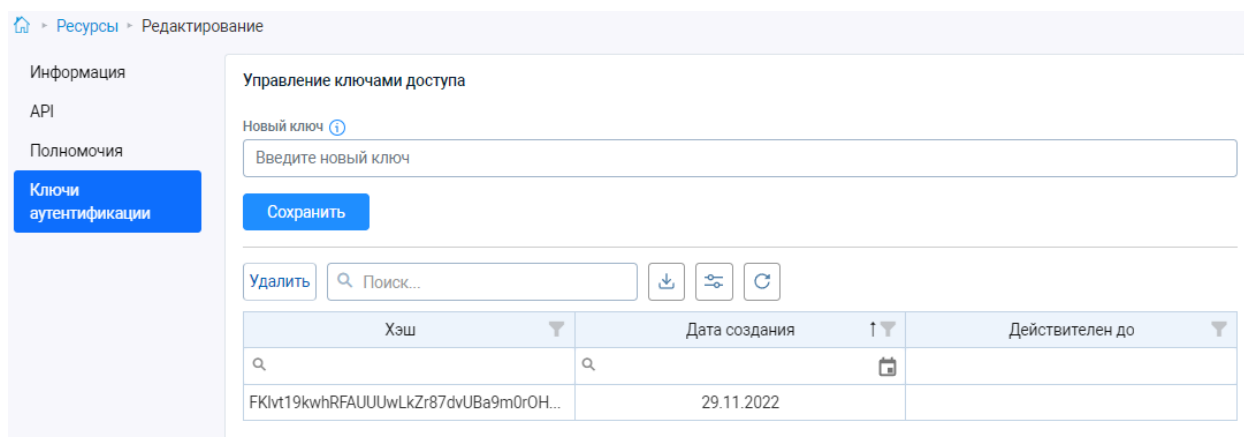


Рис. 5.11. Управление ключами доступа

Ключи хранятся в захешированном виде что исключает возможность их чтения после записи.

В случае отсутствия хоть одного действующего ключа, Ресурс не сможет аутентифицироваться в Хабе. Это приведет к тому, что Ресурс не сможет проверить входящие токены и будет обязан отвергнуть запросы пользователей как недоверенные.

После настройки Администратором Системы комбинации Клиента + Ресурса, пользователь сможет зайти в Приложение (ака Клиент) по адресу на котором он развернут:

Например: <https://192.168.48.124:6200>

В случае успешной настройки, пользователь будет переадресован на страницу ввода Логина/Пароля, а после — обратно в приложение.

В случае если настройка была выполнена с ошибкой, то возможны следующие сценарии:

1) Ошибка настройки Клиента — будет выведена ошибка при переадресации на страницу ввода логина при входе в систему;

2) Ошибка настройки Ресурса — будет выведена **Ошибка Аутентификации** при попытке пользователя запросить данные из Ресурса, или ошибка **Отказано в Доступе** в случае, если у пользователя недостаточно прав для выбранного действия.

### 5.2.3. Роли

При описании роли:

- определяется список доступных объектов и, для некоторых

объектов, права доступа к объекту (управление или только чтение);

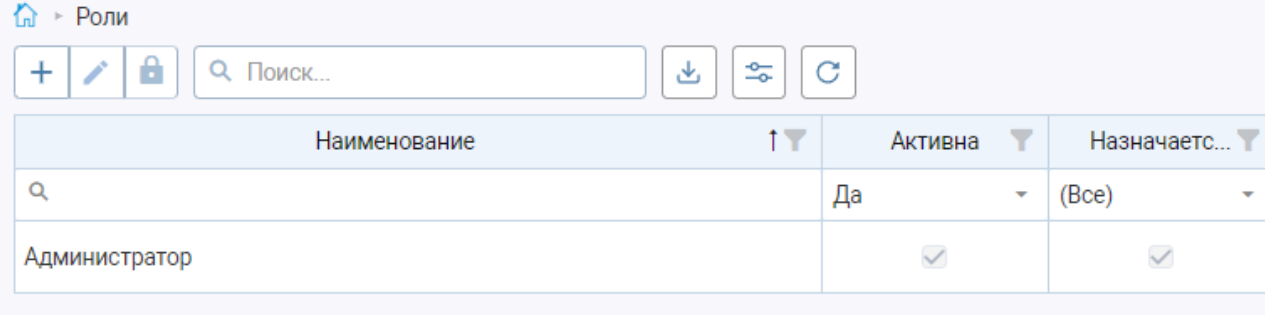
– могут быть ограничены права на уровне записи (разделение права для пользователей в разрезе динамически меняющихся данных).

В дереве присутствуют различные объекты: объекты структуры системы (подсистемы и модули, режимы), объекты интерфейса (разделы, пункты меню), объекты с общим назначением (приказы) и другие.

Чекбокс объекта имеет вид ☒, если роли предоставлен доступ к объекту и ко всем подчиненным объектам текущего объекта, вид ☐, если предоставлен доступ к некоторым подчиненным объектам.

Чтобы предоставить доступ к объекту, следует выполнить клик на чекбоксе. При предоставлении доступа к объекту автоматически предоставляется доступ к подчиненным объектам.

Роль может быть заблокирована / разблокирована (в форме списка ролей).



Роли		
Наименование	Активна	Назначается...
Администратор	Да	(Все)

Рис. 5.12. Роли



Рис. 5.13. Роль. Основная информация

Рис. 5.14. Роль. Полномочия

Рис. 5.15. Роль. Фильтры

#### 5.2.4. Пользователи

При описании пользователя определяются параметры авторизации и аутентификации (логин, пароль) и список доступных организаций/подразделений с указанием роли.

Доступные организации и указанные роли формируют списки выбора пользователя (на панели управления) при работе с системой. Текущая организация ограничивает отображение и выбор данной организацией и подчиненными подразделениями.

Пользователь может быть заблокирован / разблокирован (в форме списка пользователей).

ФИО	Логин	СНИЛС	Блокировка
admin admin admin	admin		<input type="checkbox"/>
Admin A D	admin_user		<input type="checkbox"/>

Рис. 5.16. Пользователи

Основная информация
Роли
Смена пароля

Основная информация

Логин \*

Фамилия \*

Имя \*

Отчество

Сотрудник

Электронная почта \*

СНИЛС

☐ Системная учетная запись? ⓘ

Сохранить    Закрыть

Рис. 5.17. Пользователь. Основные сведения

🏠 > Пользователи > Управление пользователем "admin"

Основная информация

**Роли**

Смена пароля

Рабочие места

🔍 Начать поиск...

Роли	Назначенные роли
🔍	(Все) ▼
▼ Центральный сегмент	Администратор ✕
▼ Получатель бюджетных средств (ПБС)	Выбрать...
▼ Администрация	Выбрать...
Администрация	Выбрать...
Кварта	Выбрать...
Помощники, Советники	Выбрать...
Центр разработки	Выбрать...
Экспертное управление	Выбрать...
▼ Аппарат	Выбрать...

Сохранить    Закрыть

Рис. 5.18. Пользователь. Роли

🏠 > Пользователи > Управление пользователем "admin"

Основная информация

Роли

**Смена пароля**

Изменение пароля пользователя admin admin admin

Введите новый пароль

Сохранить    Закрыть

Рис. 5.19. Смена пароля

### 5.2.5. Сотрудники

Режим сотрудники используется для просмотра информации личных данных сотрудников.

🏠 > Сотрудники

+
✎
🗑
🔍 Поиск...
📄
🔗
🔄

ФИО ▼	Должность ▼	Пол ▼	СНИЛС ▼	Дата рожд... ▼	Учетные записи
🔍					
admin admin admin					admin; admin_user

Рис. 5.20 Сотрудники

🏠 > Сотрудники > Редактирование

### Основная информация

Фамилия \*

Имя \*

Отчество

Дата рождения  
 📅

Электронная почта \*

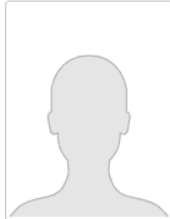
СНИЛС

Пол

Должность

Подразделение

Учетная запись

Фотография  
 +  
 .jpg, .png, .jpeg  
 Максимальный объем: 5 MB

[Сохранить](#) [Заккрыть](#)

Рис. 5.21 Сотрудники. Основная информация

### 5.2.6. События

Режимы просмотра событий используются для контроля подключения пользователей к системе и проверки прав доступа к объектам системы.

🏠 > События подсистемы безопасности

🔍 Поиск... 📄 🔄

Логин	Приложение	Успешен ли вход	Время входа
🔍	🔍	(Все)	🔍 📅
admin_user	gosduma	☑	среда, 7 декабря 2022 г., 08:26:07
test	gosduma	☐	среда, 7 декабря 2022 г., 08:25:49
admin_user	gosduma	☐	среда, 7 декабря 2022 г., 08:25:41

Рис. 5.22. События аутентификации

События подсистемы безопасности								
Поиск...								
Ло...	Пр...	Ре...	Код права	Иденти...	Иденти...	Успе...	Описание	В...
Q	Q	Q	Q	Q	Q	(Все)	Q	Q
admin...	ds	ds	ds/file_library/read	eae5f50f-2239-40c6-ba23-64ee9f729c37	6b600191-8099-4370-b7bf-7508956ed375	✓	Вызвана проверка права (ds/file_library/read) * (Имя не найдено)* для пользователя admin_user. Пользователь работает под ролью "Администратор"(eae5f50f-2239-40c6-ba23-64ee9f729c37) в организации "Центральный сегмент"(6b600191-8099-4370-b7bf-7508956ed375). В доступе разрешено.	среда, 7 декабря 2022 г., 11:26:09
test	sophie	sophie	uri://schemas.quarta.su...registration/read	6480aca8-4e2a-4834-80f3-f9beafa70481	2873b295-a717-4b8e-bc05-6c23f52a627d	✓	Вызвана проверка права (uri://schemas.quarta.su/staff/anticorruption/exter...registration/read) *Чтение сведений об адресах сайтов и (или) страниц сайтов" для пользователя test. Пользователь работает под ролью "Администратор ЯНАО"(6480aca8-4e2a-4834-80f3-f9beafa70481) в организации "Правительство Ямало-Ненецкого автономного округа"(2873b295-a717-4b8e-bc05-6c23f52a627d). В доступе разрешено.	среда, 7 декабря 2022 г., 11:26:07

Рис. 5.23. События безопасности

## **6. АВАРИЙНЫЕ СИТУАЦИИ**

Отсутствие доступа к Системе, нарушение функционирования Системы, нештатное поведение Системы может возникать по причине сбоев в функционировании аппаратного обеспечения и каналов передачи данных, прикладного и специального программного обеспечения, ошибок ввода данных авторизации и ошибок администрирования Системы, а также по причине несанкционированного вмешательства в данные Системы.

В зависимости от предполагаемой причины сбоя следует проверить:

- подключение аппаратных средств и работоспособность каналов передачи данных;
- работоспособность серверного программного обеспечения и БД Системы (пп. 4.2, 4.3);
- журнал событий и корректность распределения прав доступа пользователей (п. 5), отсутствие вредоносного ПО на сервере и рабочих станциях пользователей.

Восстановление работоспособности программного обеспечения и БД Системы описано в пп. 4.4, 4.5.