

**ЕДИНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА
«АВАНГАРД. БУХГАЛТЕРИЯ»**

РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА

Листов 40

Москва
2022

АННОТАЦИЯ

В документе представлена информация по установке и настройке единой информационной системы «Авангард. Бухгалтерия» (далее – Система). Приведены требования аппаратному и программному обеспечению Системы.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ	6
1.1. Назначение Системы	6
1.2. Состав Системы	6
1.3. Требования к техническому и программному обеспечению	6
1.3.1. Требования к программному обеспечению	6
1.3.2. Требования к техническому обеспечению	7
2. УСТАНОВКА И НАСТРОЙКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	9
2.1. Установка и настройка сервиса авторизации	9
2.1.1. Подготовка СУБД	9
2.1.1.1. Развёртывание БД сервиса авторизации	9
2.1.1.2. Выполнение скрипта (для не CodeFirst миграций).....	10
2.1.2. Развёртывание .Net сервиса в linux	13
2.1.2.1. Установка .Net	13
2.1.2.2. Настройка бэкенд-сервиса	13
2.1.2.3. Настройка nginx	14
2.1.2.4. Перезапуск служб	16
2.1.2.5. Ошибки и решения	16
2.1.3. Настройка сервиса авторизации	16
2.1.3.1. Настройка конфигурации фронтенд	16
2.1.3.2. Настройка конфигурации .Net сервиса	17
2.2. Установка и настройка сервисов бухгалтерии и итоговой отчетности 21	
2.2.1. Подготовка СУБД	21
2.2.2. Развёртывание .Net сервисов в linux	21
2.2.2.1. Установка .Net	21
2.2.2.2. Настройка бэкенд-сервисов	22
2.2.2.3. Настройка nginx	23
2.2.2.4. Перезапуск служб	24
2.2.3. Настройка сервиса бухгалтерии	24
2.2.3.1. Настройка конфигурации фронтенд	24
2.2.3.2. Настройка конфигурации .Net сервиса	25

2.2.4. Настройка сервиса итоговой отчетности	26
2.2.4.1. Настройка конфигурации фронтенд	26
2.2.4.2. Настройка конфигурации .Net сервиса.....	27
3. ПРОВЕРКА РАБОТОСПОСОБНОСТИ СИСТЕМЫ	29
3.1. Проверка работоспособности приложения.....	29
3.2. Проверка работоспособности сервера СУБД	29
4. АДМИНИСТРИРОВАНИЕ СИСТЕМЫ	30
4.1. Доступ к административным функциям	30
4.2. Пользователи и роли	31
4.2.1. Добавление нового пользователя	32
4.2.2. Редактирование справочника пользователей	34
4.2.3. Закрепление пользователя за определенными ролями (рабочими местами)	35
4.2.4. Ввод новой роли	36
4.2.5. Редактирование справочника ролей	37
4.2.6. Добавление нового пользователя в сервисе авторизации.....	38
4.3. Работа с реестром удаленных документов	39

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

Сокращенное наименование	Полное наименование
БД	База данных
ОС	Операционная система
СУБД	Система управления базами данных

1. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ

1.1. Назначение Системы

Единая информационная система «Авангард. Бухгалтерия» предназначена для автоматизации бюджетного, бухгалтерского, налогового учетов и формирования регламентированной отчетности.

1.2. Состав Системы

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

- серверная часть в составе:
 - сервера СУБД – управление данными;
 - сервер веб-приложения – бизнес-логика и внешние процедуры, реализованная посредством следующих компонент:
 - сервис авторизации (далее также – хаб);
 - сервис бухгалтерии;
 - сервис итоговой отчетности.
- 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 9.6.11;
- прокси Nginx 1.14;
- программная платформа .NET sdk версий 2.0, 2.2 и 5.0.

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 или Yandex Browser последней версии или Mozilla Firefox ESR 52.6 или выше.

Для работы с шаблонами и печатными формами документов требуется офисный пакет приложений, поддерживающий форматы DOCX, XLSX серии форматов файлов Office Open XML.

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.

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 потребуются вводить команды посредством интерфейса командной строки Linux.

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 9.6.11;
- 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:

- Подключаем ISO образ дистрибутива:
`mount PostgresProEntCert-11.12.1.iso /mnt/cdrom/ -o loop`
- В файл репозитория добавляем 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`
- Далее обновляем информацию о пакетах и устанавливаем СУБД:
`yum update`
`yum install postgrespro-ent-11`
- Переключаемся на пользователя postgres:
`su postgres`
- Подключаемся к консоли postgresql:

```
psql
```

- Создаём базы:

```
create database hub;
```

```
create database logs;
```

- Подключаемся к базе hub:

```
\c hub;
```

- Выполнение скрипта (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 apt-get install -y dotnet-sdk-5.0
```

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\...

```
{
// API бэкенда:
"serverUrl": "https://localhost:1090",
// API приложения
"staffServerUrl": "http://X.X.X.X:1080",
"authentication": {
    "authority": "https://localhost:1090",
    "redirect_uri": "http://localhost:1090/login-callback",
    "post_logout_redirect_uri": "http://localhost:1090/login",
    "client_id": "admin",
    "response_type": "id_token token",
    "scope": "openid profile identityServer"
```



```
}  
}
```

2.1.3.2. Настройка конфигурации .Net сервиса

Конфигурационный файл appsettings.json размещается в папке \Clients\...

```
{  
  // Хост сервиса  
  "ApplicationUrls": [  
    "https://localhost:1090"  
  ],  
  // Строки подключения  
  "ConnectionStrings": {  
    // Основная база данных проекта  
    "DefaultConnection": "Server=X.X.X.X; Port=X; Database=X; User ID=X;  
Password=X",  
    "LogsConnection": "Server=X.X.X.X; Port=X; Database=X; User ID=X;  
Password=X",  
    // База данных  
    "StaffConnection": "Server=X.X.X.X; Port=X; Database=X; User ID=X;  
Password=X"  
  },  
  // Хосты, с которых можно осуществлять доступ к API  
  "CorsOrigins": [  
    "http://X.X.X.X:1080",  
    "http://X.X.X.X:X"  
  ],  
  // Периодически выполняемые задачи  
  "HostedServices": {  
    "Items": [  
      {  
        // Импорт перечня организаций  
        "Key": "OrganizationReplication",  
        "Enabled": true,  
        "Interval": "00:05:00"  
      },  
      {  
        // Кешировать ли настройку состояния грида  
        "key": "GridStateCaching",  
        "Enabled": true  
      }  
    ]  
  }  
}
```

```

    }
  ]
},
// Параметры логирования
"Logging": {
  "IncludeScopes": false,
  "LogLevel": {
    "Default": "Warning",
    "System": "Warning",
    "Microsoft": "Warning"
  }
},
// Хосты модулей для навигации (меню может перебрасывать на разные
сервисы) "Navigation": {
  "Origins": {
    "Auth": "https://X.X.X.X:1090", (опционально, если в меню будет ссылка
на хаб)
    "Sophie": "https://X.X.X.X:1080"
  },
  "GridState": {
    "FallbackOnNoCache": false
  }
},
// Возможность фильтрации данных по настраиваемым спискам
"RoleSettings": {
  "FilterByLists": false
},
"Workspace": {
  "OrgMode": "single",
  "RoleMode": "multi"
},
// Настройка аутентификации
"Authentication": {
  "Authority": "https://X.X.X.X:1090", // хост сервиса аутентификации
  "ApiName": "identityServer", // идентификатор api аутентификации
  "ApiSecret": "auth", // секретное слово для аутентификации
  "Password": {
    "RequireDigit": false,

```

```

    "RequiredLength": 1,
    "RequireNonAlphanumeric": false, // должны ли пароли содержать символ,
не являющийся буквенно-цифровым
    "RequireUppercase": false,
    "RequireLowercase": false,
    "RequiredUniqueChars": 1
  },
  // Настройка блокировки аккаунта при неверном вводе пароля
  "AccountBlocking": {
    "MaxLoginAttempts": 2
  },
  // Настройки доступа SPA приложения к API сервиса аутентификации
  "Identity": {
    "ClientId": "admin",
    "ClientName": "admin",
    "Host": "https://X.X.X.X:1090 http://localhost:1090",
    "CallbackUrl": "https://X.X.X.X:2090/login-callback
http://localhost:1090/login-callback",
    "PostLogoutUrl": "https://X.X.X.X:1090/login http://localhost:2090/login",
    "Scope": "openid profile identityServer"
  }
}
}
}

```

Все строковые настройки **РЕГИСТРОЗАВИСИМЫЕ**.

Дополнительное описание параметров:

1. ConnectionStrings – строки подключения к базам данных.
DefaultConnection – строка подключения к хабу.
StaffConnection – строка подключения к кадровой системе.
2. ApplicationUrls – порты размещения приложения
3. CorsOrigins – перечень URL, с которых можно принимать запросы
4. HostedServices – настройки работы системы
OrganizationReplication – настройка импорта перечня организаций и подразделений из кадров
GridStateCaching – инициализировать кэш при старте приложения
5. Logging – логирование

При установке значения в «Warning» аккумулируется информация об ошибках и потенциальных проблемах. При установке значения в «Information» аккумулируется вся информация

6. Navigation – справочник добавочных путей для навигации по меню
Запись вида «Sophie»: «http://XXX.XXX.XX.XXX:XXXX» указывает IP и порт для добавления префикса к URL путям пунктов меню, относящихся к «Sophie».
7. GridState – искать ли в БД данные по настройкам грида, если их не оказалось в кеше
8. Authentication – параметры аутентификации
Authority – путь к бэкенду
ApiName – наименование приложения
ApiSecret – секретный ключ для подключения. В БД хранится в виде хэша.
AccountBlocking – параметры блокировки аккаунта:
 - MaxLoginAttempts - тип integer
Максимальное количество попыток авторизации пользователя. При превышении происходит блокировка учетной записи.

Password – параметры настройки требований к паролю (политика безопасности):

- RequireDigit - тип bool.
Требуется хотя бы одна цифра.
- RequiredLength - тип integer.
Требование к длине пароля.
- RequireNonAlphanumeric - тип bool.
Требуется ли хотя бы один спецсимвол.
- RequireUppercase - тип bool.
Требуется хотя бы одна заглавная буква.
- RequireLowercase - тип bool.
Требуется хотя бы одна строчная буква.
- RequiredUniqueChars - тип integer.
Требование к количеству уникальных символов в пароле.

Identity – информация для идентификации клиентов (приложений) и предоставления им доступа к хабу:

- Host – перечень обслуживаемых ресурсов

- CallbackUrl – путь перенаправления после авторизации
- PostLogoutUrl – пути перенаправления пользователя после выхода из системы

Настройка разрешений для пользователей подсистемы аутентификации
Зайти в раздел «.../clients» и создать перечень приложений. Для развёртывания необходимо соответственно 2 приложения.

- Наименование приложения, например: Sophie_develop
(Должно совпадать с client_id в environment.prod.ts)
- Доступные ресурсы (API, OpenId, Profile) - перечислить разделы.
Например: openid profile identityServer
- Redirect Uri - то, куда направить после аутентификации.
Например: http://localhost:1080/
Post Logout Redirect Uri - то, куда направить после выхода.
Например: http://localhost:1080/Login
- CORS origins - перечислить пути, с которых следует разрешать запросы.
Например: http://localhost:1080
- Время жизни токена – время действия токена в секундах, по умолчанию 3600 секунд.

2.2. Установка и настройка сервисов бухгалтерии и итоговой отчетности

2.2.1. Подготовка СУБД

Необходимо создать БД **accounting** и поднять бэкап (*/Бэкапы баз данных/Бэкап базы бухгалтерии и итоговой отчётности.bak.sql*).

Если сервисы устанавливаются на выделенный сервер, то предварительно необходимо установить СУБД (см. п. 2.1.1.1.).

2.2.2. Развёртывание .Net сервисов в linux

2.2.2.1. 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 apt-get install -y dotnet-sdk- 2.0.9 (Для работы Бухгалтерии)
- sudo apt-get install -y dotnet-sdk- 2.2.8 (Для работы Итоговой отчетности)

2.2.2.2. Настройка бэкенд-сервисов

Выполнить следующие действия:

- Копировать дистрибутив **сервиса бухгалтерии** (*Дистрибутивы сервисов\Бухгалтерия*) на сервер (/home/midto/hosting/accounting/).
- Копировать дистрибутив **сервиса итоговой отчетности** (*Дистрибутивы сервисов\Итоговая отчетность*) на сервер (/home/midto/hosting/reporting/).
- При отсутствии создать пользователя, под которым будет запускать приложение. Пользователь должен иметь полный доступ к директории с приложением и права открывать сокеты.
- Для **сервиса бухгалтерии** создать службу /etc/systemd/system/accounting.service. Указать директории, пути к файлам и приложениям, описание и указать пользователя под кем будет запускать приложение:

```
[Unit]
Description=Accounting Development

[Service]
WorkingDirectory=/home/midto/hosting/accounting/
ExecStart=/usr/bin/dotnet
/home/midto/hosting/accounting/Quarta.Accounting.WebApi.dll
Restart=always
RestartSec=10
SyslogIdentifier=dotnet-accounting-development
User=%ИМЯ СИСТЕМНОГО ПОЛЬЗОВАТЕЛЯ%
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false
Environment=DOTNET_CLI_TELEMETRY_OPTOUT=true

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

- Для **сервиса итоговой отчетности** создать службу /etc/systemd/system/reporting.service. Указать директории, пути к файлам и приложениям, описание и указать пользователя под кем будет запускать приложение:

```
[Unit]
Description=Accounting Development

[Service]
```

```
WorkingDirectory=/home/user/hosting/reporting/  
ExecStart=/usr/bin/dotnet  
    /home/user/hosting/reporting/Quarta.Accounting.WebApi.dll  
Restart=always  
RestartSec=10  
SyslogIdentifier=dotnet-reporting-stable  
User=%ИМЯ СИСТЕМНОГО ПОЛЬЗОВАТЕЛЯ%  
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false  
Environment=DOTNET_CLI_TELEMETRY_OPTOUT=true  
  
[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.

#Для **сервиса бухгалтерии** добавить в server прокси к приложению:

```
server {  
    listen    1050;  
    location / {  
        proxy_pass http://127.0.0.1:5050;  
    }  
}
```

#Для **сервиса итоговой отчетности** добавить в server прокси к приложению:

```
server {  
    listen    1060;  
    location / {  
        proxy_pass http://127.0.0.1:5060;  
    }  
}
```

2.2.2.4. Перезапуск служб

Выполнить команды:

```
systemctl restart nginx
```

```
systemctl restart accounting
```

```
systemctl restart reporting
```

2.2.3. Настройка сервиса бухгалтерии

2.2.3.1. Настройка конфигурации фронтенд

Конфигурационный файл **configuration.json** размещается в директории
\\quarta-accounting-web-app\\assets\\configurations\\clients\\...

```
{  
    // API бэкенда  
    "serverUrl": "http://X.X.X.X:1050",  
    // Адрес для перехода на Итоговую отчетность  
    "consolidatedReportingUrl": "http://X.X.X.X:1060",  
    "authentication": {  
        // Хост аутентификации:  
        "authority": "https://X.X.X.X:1090",  
        // Url перенаправления после аутентификации:  
        "redirect_uri": "http://X.X.X.X:2080/#/account/login/esa/callback?",  
        //Url перенаправления после выхода из системы:  
        "post_logout_redirect_uri": "http://X.X.X.X:1050/ ",  
        // redirect_uri, post_logout_redirect_uri – должны вести на один и  
        тот же хост (данного приложения);  
        "silent_redirect_uri": "http://X.X.X.X:1050/#/account/login/esa/silent-callback?",  
        // Идентификатор данного приложения:  
        "client_id": "accounting_development",  
        // запрашиваемые данные о пользователе:  
        "response_type": "id_token token",  
        // Требуемые разрешения:  
        "scope": "openid accounting",  
        "useLocalStorage": true,
```



```

    "automaticSilentRenew": true
  },
}

```

2.2.3.2. Настройка конфигурации .Net сервиса

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

```

{
  // Хост сервиса
  "ApplicationUrls": [
    "http://localhost:X"
  ],
  // Параметры работы с хабом (сервисом авторизации)
  "AuthServer": {
    "Url": "https://X.X.X.X:X",
    "ApiName": "accounting",
    "ApiSecret": "accountingSecretKey",
    "ClaimType": "uri://schemas.quarta.su/permission-claim-type"
  },
  // Строки подключения
  "ConnectionStrings": {
    // Основная база данных проекта
    "DefaultConnection": "Server=X.X.X.X; Port=X; Database=X; User
ID=X; Password=X",
  },
  // Поставщик данных
  "DefaultDatabaseProvider": {
    "IsPostgreSqlProvider": "true",
  },
  // Хосты, с которых можно осуществлять доступ к API. Указать тут
  себя (для фронтенда) и хаб (сервис авторизации)
  "WithOrigins": [
    "http://X.X.X.X:X",
    "http://X.X.X.X:X"
  ],
  // Параметры логирования
  "Logging": {
    "LogLevel": {
      "Default": "Warning",

```

```

        "System": "Warning",
        "Microsoft": "Warning"
    }
},
}

```

2.2.4. Настройка сервиса итоговой отчетности

2.2.4.1. Настройка конфигурации фронтенд

Конфигурационный файл **configuration.json** размещается в директории \quarta-reporting-web-app\assets\configurations\clients\...

```

{
    // API бэкенда:
    "serverUrl": "",
    "inputDebounceMs": 300,
    "accountingUrl": "http://X.X.X.X:1050",
    "authentication": {

        // Хост аутентификации:
        "authority": "https://X.X.X.X:1090",

        // Url перенаправления после аутентификации:
        "redirect_uri": "http://X.X.X.X:1060/callback",
        //Url перенаправления после выхода из системы:
        "post_logout_redirect_uri": "http://X.X.X.X:1060/ ",
        // redirect_uri, post_logout_redirect_uri – должны вести на один и
        тот же хост (данного приложения);

        "silent_redirect_uri": "http://X.X.X.X:1060/silent-callback?",

        // Идентификатор данного приложения:
        "client_id": "reporting_development",

        // запрашиваемые данные о пользователе:
        "response_type": "id_token token",

        // Требуемые разрешения:
        "scope": "identityServer profile openid reporting",

        "useLocalStorage": true,
    }
}

```

```

    "automaticSilentRenew": true
  },
  // Источник данных о пользовательских настройках грида (обычно
это хаб):
  "gridState": {
    "api": "https://X.X.X.X:1090/api/grid-state"
  },
}

```

2.2.4.2. Настройка конфигурации .Net сервиса

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

```

{
  // Хост сервиса
  "ApplicationUrls": [
    "http://localhost:X"
  ],
  // Параметры работы с хабом (сервисом авторизации)
  "AuthServer": {
    "Url": "https://X.X.X.X:X",
    "ApiName": "sophie",
    "ApiSecret": "sophieSecretKey",
  },
  // Параметры взаимодействия с базой данных ФИАС
  "Fias": {
    "ConnectionStringName": "FiasConnection",
    "Enabled": true
  },
  // Строки подключения
  "ConnectionStrings": {
    // Основная база данных проекта
    "DefaultConnection": "Server=X.X.X.X; Port=X; Database=X; User
ID=X; Password=X",
  },
  // Хосты, с которых можно осуществлять доступ к API. Указать тут
себя (для фронтенда) и хаб (сервис авторизации)
  "WithOrigins": [
    "http://X.X.X.X:X",
    "http://X.X.X.X:X"
  ]
}

```

```
],  
// Параметры логирования  
"Logging": {  
  "LogLevel": {  
    "Default": "Warning",  
    "System": "Warning",  
    "Microsoft": "Warning"  
  }  
},  
}
```

3. ПРОВЕРКА РАБОТОСПОСОБНОСТИ СИСТЕМЫ

3.1. Проверка работоспособности приложения

Для проверки работоспособности Системы необходимо выполнить следующие действия:

- запустить Интернет-браузер;
- авторизоваться в Системе, выполнив действия, описанные в подразделе 4.1.

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

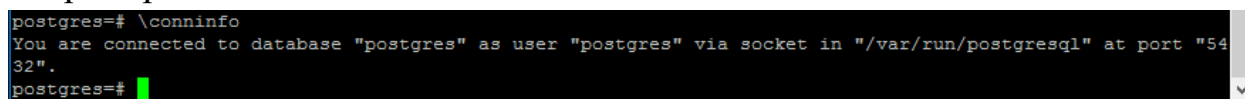
3.2. Проверка работоспособности сервера СУБД

Для проверки работоспособности сервера СУБД требуется выполнить следующие действия:

- Убедиться в том, что сервисы сервера СУБД находятся в состоянии «Работает».
- Подключиться к БД.

Для этого достаточно выполнить команду `psql` и в запустившемся 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=#
```

Рис. 3.1.

Успешное соединение с БД свидетельствует о работоспособности сервера.

4. АДМИНИСТРИРОВАНИЕ СИСТЕМЫ

4.1. Доступ к административным функциям

Для получения доступа к полному перечню административных функций Системы необходимо:

запустить Интернет-браузер;

- в **адресной строке** ввести адрес сервера Системы. Адрес сервера имеет следующий вид - http://адрес_веб-сервера:1050 (пример: <http://localhost:1050>);
- в окне авторизации ввести учетные данные пользователя с правами администратора (*по умолчанию доступен пользователь «test» с паролем «1»*), нажать на кнопку **«Войти»**:

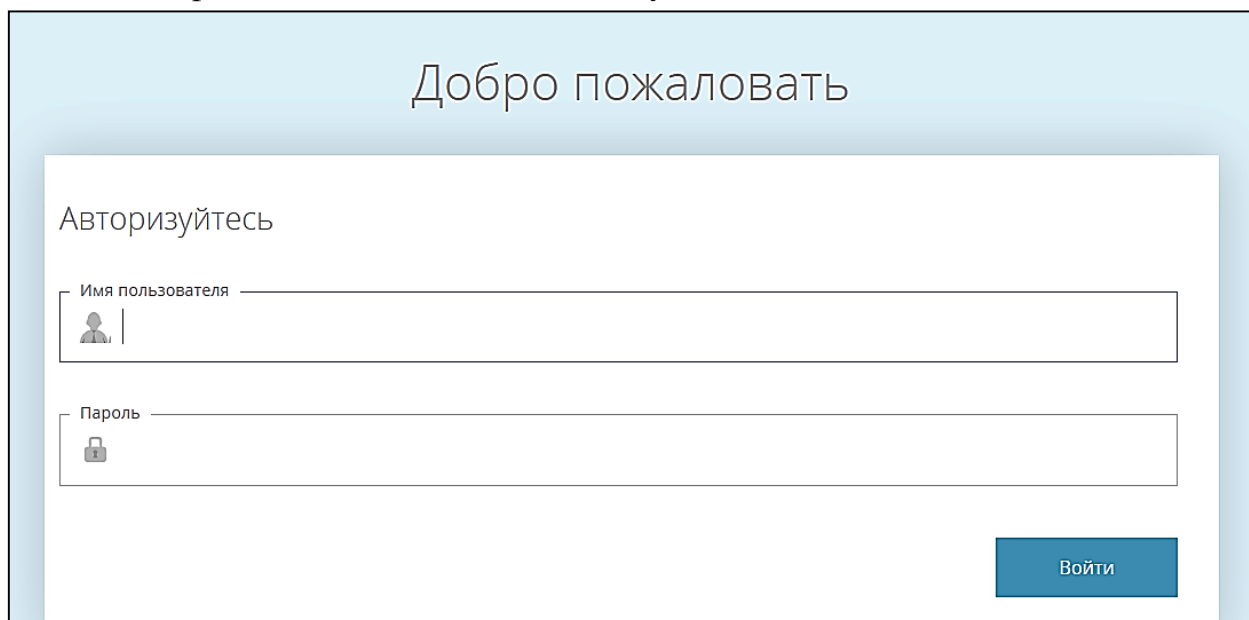



Рис. 4.1.

- открыть панель главного меню по кнопке  и выбрать пункт меню «Сервис», откроется страница с разделом «Администрирование».

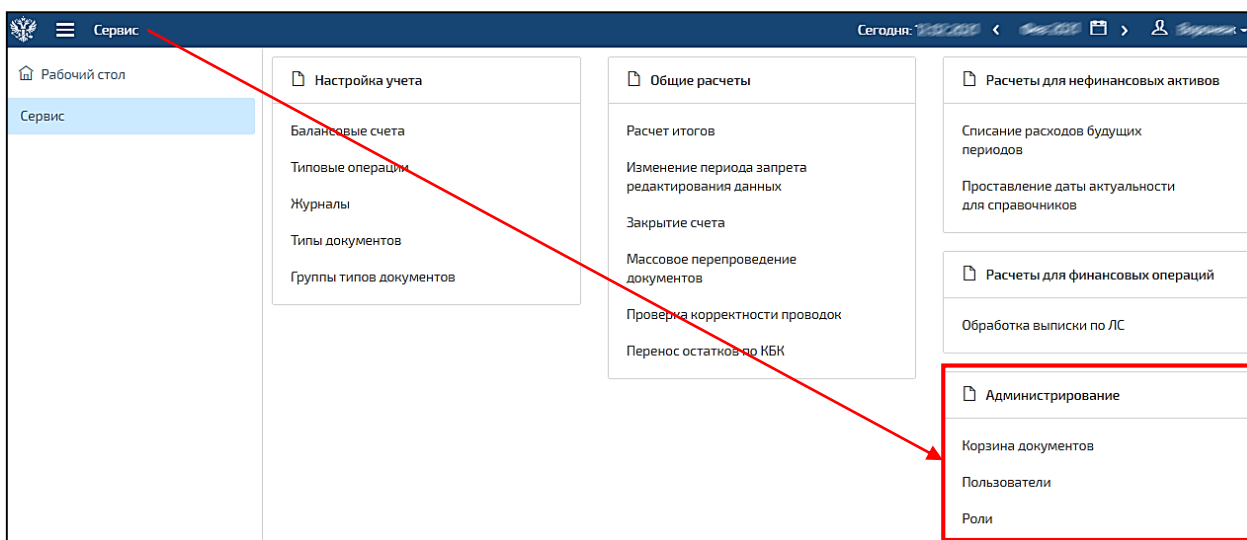


Рис. 4.2.

Раздел «Администрирование» состоит из следующих пунктов:

- **Корзина документов** – список всех удаленных документов пользователями Системы с возможностью восстановления или безвозвратного их удаления.
- **Пользователи** – ссылка на справочник «Пользователи», в котором реализована возможность управления учетными записями пользователей.

Примечание. Нового пользователя необходимо добавить как в сервис бухгалтерии (см. п. **Ошибка! Источник ссылки не найден.**), так и в сервис авторизации (см. п. 4.2.6) с идентичными логинами.

- **Роли** – перечень ролей в Системе.

4.2. Пользователи и роли

В Системе понятие «роль» определяет типы документов и способы обработки информации в них.

После установки в Системе зарегистрированы следующие роли:

- Администрирование;
- Расчеты – доступен только просмотр списка расчетов.

Сразу после установки в Системе есть один зарегистрированный пользователь Администратор с правами администрирования.

Каждому пользователю может быть присвоено несколько ролей. В системе реализованы два справочника, обеспечивающие работу пользователей в Системе:

- справочник пользователей, в котором хранится имя пользователя, пароль для входа в систему и другие атрибуты учетных записей

пользователей;

- справочник ролей (для каждой роли настраиваются требуемые права доступа).

На базе этих двух справочников администратор системы может сформировать связь «пользователь – роль – права доступа», т.е. будет создана информация о том, какой пользователь может осуществлять работу с теми или иными объектами Системы.

В общем случае для организации работы с точки зрения распределения прав пользователей по ролям необходимо:

- ввести пользователей в справочник (справочник «Пользователи»);
- ввести необходимые роли (рабочие места) (справочник «Роли»), которые имеют необходимые настройки прав доступа;
- организовать в нужных сочетаниях пары «пользователь - роль» (кнопка «Роли» в справочнике «Пользователи»).

4.2.1. Добавление нового пользователя

Ниже приведена последовательность действий для добавления нового пользователя.

1. Вход в режим работы администратора (см. п.4.1).
2. Вызов справочника пользователей. Выбрать пункт меню «**Пользователи**». Будет открыт перечень пользователей, созданных в Системе на данный момент:

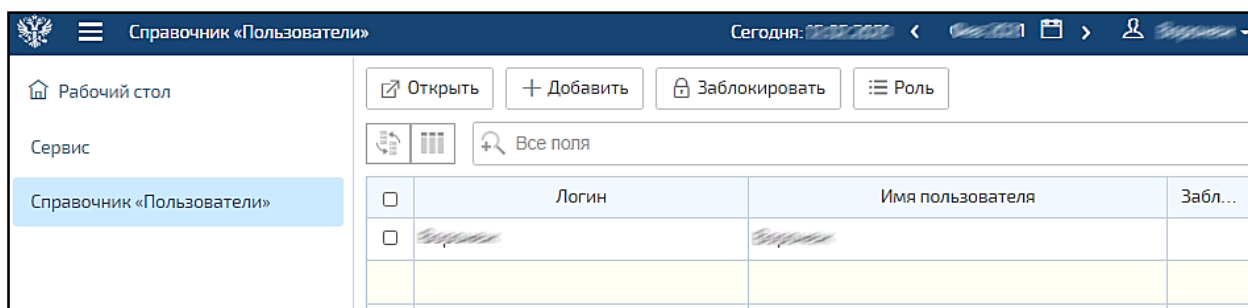


Рис. 4.3.

3. Заполнить данные нового пользователя. Нажать кнопку «**Добавить**». Появится окно для ввода данных по новому пользователю:

Добавление нового значения

Имя для меню *

Имя для печати *

Должность пользователя

Телефон (рабочий)

Дата блокировки

Дата последнего входа

Сотрудник

Задать пароль

Организации

Открыть + Добавить Удалить

Все поля

Организация	Тип доступа

Сохранить Заккрыть

Рис. 4.4.

4. Заполнить все необходимые поля:

- **Имя для меню** – краткое имя пользователя (например, фамилия и инициалы);
- **Имя для печати** – полное имя пользователя, например, ФИО без сокращений (будет отображаться в справочнике пользователей);
- **Должность пользователя** – заполняется вручную;
- **Телефон (рабочий)** – контактный телефон пользователя;
- **Дата блокировки** – срок действия пользователя в Системе (по истечению срока действия пользователь удаляется из Системы);
- **Дата последнего входа** – заполняется автоматически датой последнего входа в Систему. Так как пользователь новый, поле не заполняется;
- **Сотрудник** - выбор сотрудника из одноимённого справочника;
- **Задать пароль** – при нажатии на кнопку открывается поле «Пароль» для заполнения. С этим паролем пользователь будет входить в

Систему. При запуске Системы пароль отображается знаками «*», поэтому при вводе необходимо учитывать языковую раскладку клавиатуры и регистр ввода;

5. Заполнить табличную часть «Организации»:

- нажать кнопку «Добавить», откроется окно выбора организации:

Рис. 4.5.

- выбрать организацию из раскрывающегося списка;
 - выбрать вид доступа и нажать «Добавить»:
 - Можно просматривать данные только указанной организации;
 - Можно просматривать данные указанной организации и ее подчиненных организаций;
 - Можно просматривать данные только организаций, подчиненных данной.
6. Сохранение данных. Для сохранения введенных данных нажать кнопку «Сохранить». Данные будут сохранены, форма ввода закрыта. В справочнике пользователей появится новая запись.

4.2.2. Редактирование справочника пользователей

Данные о любом пользователе Системы могут быть отредактированы. Редактирование данных пользователя выполняется в следующем порядке:

1. Вход в систему в режиме администратора.
2. Открытие справочника пользователей. Выбрать пункт меню «Сервис» → «Пользователи».
3. Установка курсора на нужную строку справочника пользователей. Выделить запись пользователя, данные которого требуется изменить.
4. Открытие формы редактирования. Нажать кнопку «Открыть». На экране будет представлена форма с данными выбранного пользователя, аналогичная форме для ввода нового пользователя.

5. Внесение изменений. Отредактировать нужные данные (имя пользователя, пароль и т.д.).
6. Сохранение изменений. Нажать кнопку «Сохранить».

Для блокировки пользователя необходимо выполнить шаги с 1 по 3, описанные для редактирования данных пользователя. Далее нажать кнопку «**Заблокировать**». После подтверждения процедуры пользователь будет заблокирован (в колонке «Заблокирован» списка пользователей появится «флаг»):

<input type="checkbox"/>	Логин	Имя пользователя	Заблокирован
<input type="checkbox"/>	11111	1111	✓

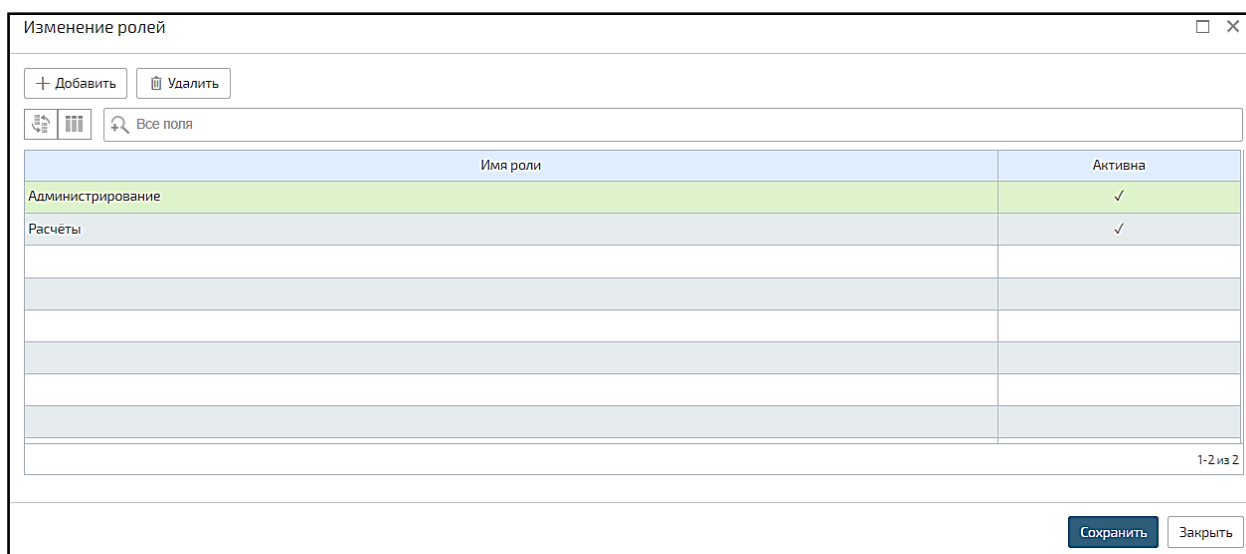
Рис. 4.6.

Чтобы разблокировать пользователя необходимо выполнить шаги с 1 по 3, описанные для редактирования данных пользователя. Далее выбрать пользователя и нажать кнопку «**Разблокировать**».

4.2.3. Закрепление пользователя за определенными ролями (рабочими местами)

Кнопка «**Роли**» в справочнике пользователей дает возможность просмотреть информацию о том, какие роли (рабочие места) «закреплены» за тем или иным пользователем. Для добавления роли пользователю необходимо:

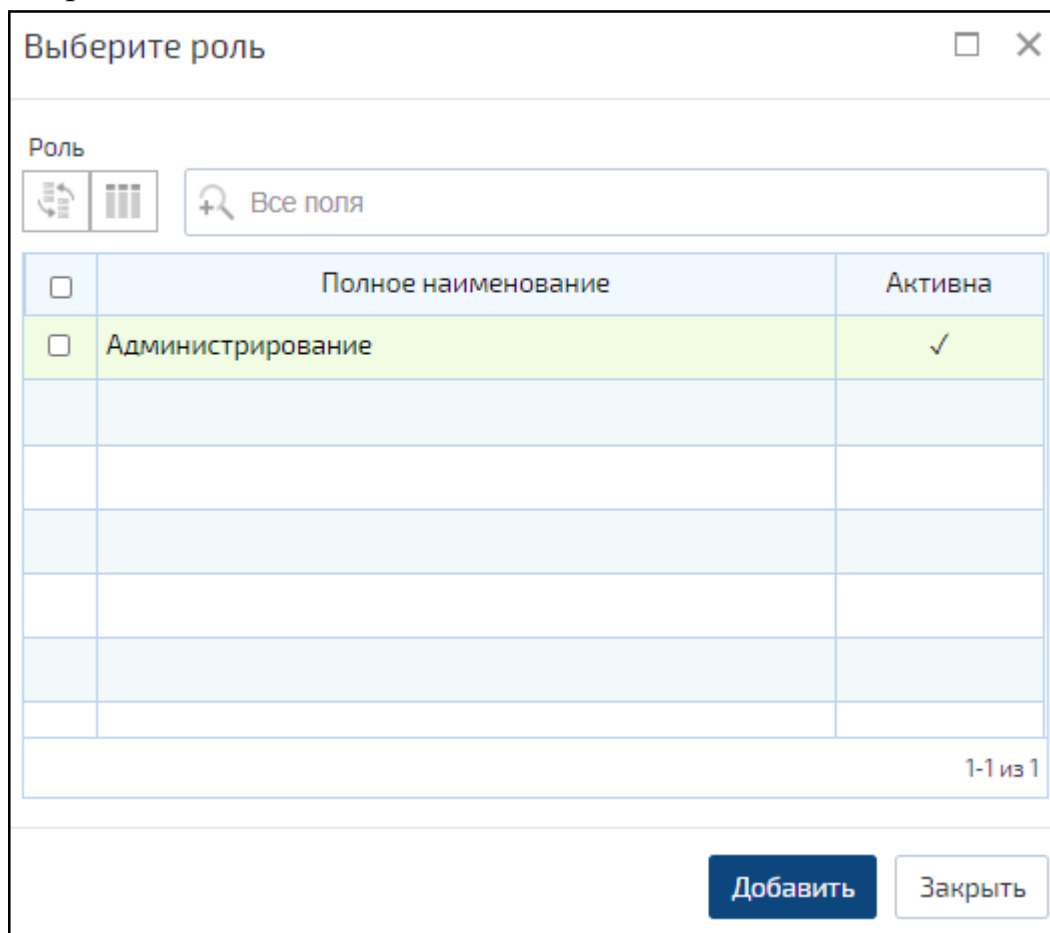
1. Выполнить шаги с 1 по 3, описанные для редактирования данных пользователя.
2. Нажать кнопку «**Роли**». Появится окно с перечнем ролей пользователя:



Имя роли	Активна
Администрирование	✓
Расчёты	✓

Рис. 4.7.

3. Добавление новой связи «пользователь–роль». Для того чтобы сформировать новую связь «пользователь–роль» необходимо:
 - нажать кнопку «**Добавить**». Появится форма выбора доступных ролей:



Выберите роль

Роль

Все поля

<input type="checkbox"/>	Полное наименование	Активна
<input type="checkbox"/>	Администрирование	✓

1-1 из 1

Добавить Закрыть

Рис. 4.8.

- выбрать роль (поставив первую колонку «флагом») и нажать кнопку «**Добавить**»;
4. Сохранение данных. Для сохранения введенных данных нажать кнопку «**Сохранить**». Данные будут сохранены,
- Для удаления роли у пользователя необходимо повторить шаги 1 -2, нажать кнопку «**Удалить**». После подтверждения процедуры пара «пользователь - роль» будет удалена.

4.2.4. Ввод новой роли

Ниже приведена последовательность действий для добавления новой роли.

1. Вход в режим работы администратора (см. п.4.1).
2. Вызов справочника ролей. Выбрать пункт меню «**Роли**». Будет открыт перечень ролей, созданных в Системе на данный момент:

3. Установка курсора на нужную строку справочника ролей. Выделить запись пользователя, данные которого требуется изменить.
4. Открытие формы редактирования. Нажать кнопку **«Открыть»**. На экране будет представлена форма с данными выбранной роли, аналогичная форме для ввода нового значения.
5. Внесение изменений. Отредактировать нужные данные (название, набор действий с объектами Системы).
6. Сохранение изменений. Нажать кнопку **«Сохранить»**.

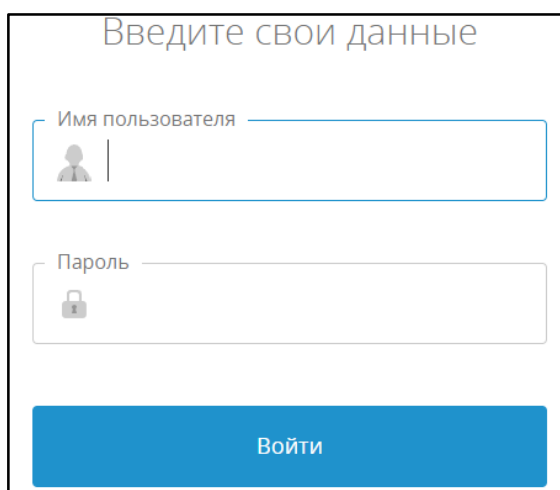
Для удаления роли необходимо выполнить шаги с 2-3, описанные для редактирования данных. Далее нажать кнопку **«Удалить»**. После подтверждения процедуры роль будет удалена.

4.2.6. Добавление нового пользователя в сервис авторизации

Сервис авторизации (хаб) разрабатывается как единый модуль управления доступом и настройки параметров функционирования для всех продуктов линейки «Авангард».

В настоящее время ввод сведений о пользователе, а также настройка ролей и прав выполняется средствами сервиса бухгалтерии. Но для успешной авторизации пользователя он также должен быть добавлен в сервис авторизации.

Сервис авторизации является отдельным приложением с собственным адресом.



Введите свои данные

Имя пользователя

Пароль

Войти

Рис. 4.11. Окно авторизации

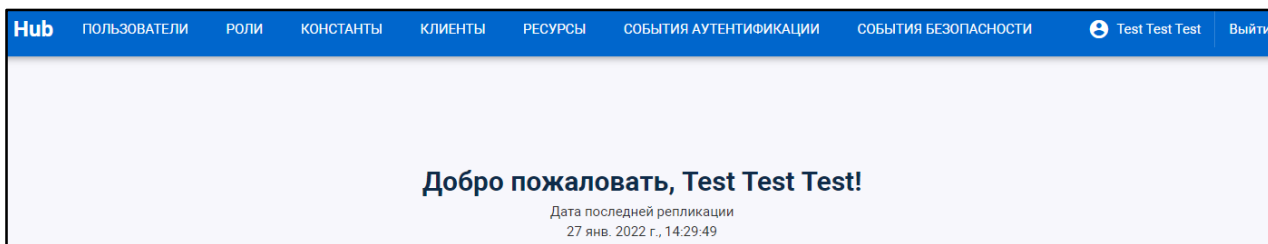


Рис. 4.12. Главная страница сервиса

ФИО	Login	СНИЛС	Роли	Блокировка
Administrator Administrator Administrator	Administrator		1/0	<input checked="" type="checkbox"/>
ADTestS ADTestLN ADTestP	adtest	38759075650	1/0	<input type="checkbox"/>

Рис. 4.13. Пользователи

Основная информация

Рабочие места

Основная информация

Логин *

TestPS

Фамилия *

Тестовый

Имя *

Пётр

Отчество

Сергеевич

Электронная почта *

Электронная почта

СНИЛС *

СНИЛС

Сохранить Закрыть

Рис. 4.14. Пользователь. Основные сведения

4.3. Работа с реестром удаленных документов

Ниже приведена последовательность действий для работы с реестром удаленных документов.

1. Вход в режим работы администратора (см. п.4.1).
2. Вход в реестр удаленных документов. Выбрать пункт меню «Корзина документов». Будет открыт перечень удаленных документов:

<input type="checkbox"/> Восстановить		<input type="checkbox"/> Удалить безвозвратно						
<input type="checkbox"/> <input type="checkbox"/>		<input type="text"/> Все поля						
<input type="checkbox"/>	Тип документа	Номер	Дата документа	Дата проводки	Сумма	Примечание	Кто удалил	Когда удален
<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>

Рис. 4.15.

- Пометить «флагом» в первой колонке документы и выбрать действие **«Восстановить»** или **«Удалить безвозвратно»**. После подтверждения процедуры документ будет удален или восстановлен.