# ЕДИНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА
# «АВАНГАРД. КАДРЫ И ЗАРПЛАТА»

# РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА

Листов 43

Москва

2022

# АННОТАЦИЯ

В документе представлена информация по установке и настройке единой информационной системы «Авангард. Кадры и зарплата» (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 сервиса файлового хранилища ..... 21
2.2.1. Подготовка СУБД ..... 21
2.2.2. Развёртывание .Net сервисов в linux ..... 22
2.2.2.1. Установка .Net ..... 22
2.2.2.2. Настройка бэкенд-сервисов ..... 22
2.2.2.3. Настройка nginx ..... 23
2.2.2.4. Перезапуск служб ..... 23
2.2.3. Настройка сервиса файлового хранилища ..... 24
2.2.3.1. Настройка конфигурации .Net сервиса ..... 24
2.3. Установка и настройка сервиса кадров ..... 24
2.3.1. Подготовка СУБД ..... 24
2.3.2. Развёртывание .Net сервисов в linux ..... 25

2.3.2.1. Установка .Net ... 25
2.3.2.2. Настройка бэкенд-сервисов ... 25
2.3.2.3. Настройка nginx ... 26
2.3.2.4. Перезапуск служб ... 26
2.3.3. Настройка сервиса кадров ... 26
2.3.3.1. Настройка конфигурации фронтенд ... 26
2.3.3.2. Настройка конфигурации .Net сервиса ... 27
2.4. Установка и настройка сервиса зарплаты ... 29
2.4.1. Подготовка СУБД ... 29
2.4.2. Развёртывание .Net сервисов в linux ... 29
2.4.2.1. Установка .Net ... 29
2.4.2.2. Настройка бэкенд-сервисов ... 29
2.4.2.3. Настройка nginx ... 30
2.4.2.4. Перезапуск служб ... 30
2.4.3. Настройка сервиса зарплаты ... 31
2.4.3.1. Настройка конфигурации фронтенд ... 31
2.4.3.2. Настройка конфигурации .Net сервиса ... 32
3. ПРОВЕРКА РАБОТОСПОСОБНОСТИ СИСТЕМЫ ... 34
3.1. Проверка работоспособности приложения ... 34
3.2. Проверка работоспособности сервера СУБД ... 34
4. АДМИНИСТРИРОВАНИЕ СИСТЕМЫ ... 35
4.1. Доступ к сервису ... 35
4.2. Управление доступом ... 36
4.2.1.1. Приложения и сервисы ... 36
4.2.1.2. Роли ... 38
4.2.1.3. Пользователи ... 40
4.2.1.4. События ... 41
4.3. Настройка параметров функционирования системы ... 42

## 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. Кадры и зарплата» предназначена для автоматизации процессов управления персоналом и расчета заработной платы в государственных учреждениях и коммерческих организациях.

## 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 кадров;
    - сервис зарплаты.
- 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;
- СУБД MongoDB 5;
- прокси Nginx 1.14;
- программная платформа .NET sdk 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 потребуется 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:

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 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), вполнив команды:

```
$ wget https://nginx.org/download/nginx-1.19.0.tar.gz
$ tar zxf 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 кадров
    "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 из кадров
    "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 **<u>РЕГИСТРОЗАВИСИМЫЕ</u>**.

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) - перичислить 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 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 кадров

### 2.3.1. Подготовка СУБД

Необходимо создать БД **sophie** и поднять бэкап *(/Бэкапы баз данных/Бэкап базы кадров и зарплаты.bak.sql)*.

Если сервисы устанавливаются на выделенный сервер, то предварительно необходимо установить СУБД (см. п. 2.1.1.1.) и выполнить скрипт (см. п. 2.1.1.2).

### 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\Кадры\)* на сервер *(/home/midto/hosting/sophie/development)*.
- 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/sophie.service. Указать директории, пути к файлам и приложениям, описание и указать пользователя под кем будет запускать приложение:

```
[Unit]
Description=Sophie

[Service]
WorkingDirectory=/home/midto/hosting/sophie/development
ExecStart=/usr/bin/dotnet
   /home/midto/hosting/sophie/development/Quarta.StaffPortal.WebApi.dll
Restart=always
RestartSec=10
SyslogIdentifier=dotnet-sophie-development
User=%ИМЯ СИСТЕМНОГО ПОЛЬЗОВАТЕЛЯ%
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false
Environment=DOTNET_CLI_TELEMETRY_OPTOUT=true
StandardOutput=/var/log/sophie-develop.all.log
StandardError=/var/log/sophie-develop.err.log

[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       1080;
                location / {
                   proxy_pass http://127.0.0.1:5080;
                }
 }
```

#### 2.3.2.4. Перезапуск служб

Выполнить команды:

systemctl restart nginx

systemctl restart sophie

### 2.3.3. Настройка сервиса кадров

#### 2.3.3.1. Настройка конфигурации фронтенд

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

```
{
   // API бэкенда:
   "serverUrl": "http://X.X.X.X:1080",
   "authentication": {

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

         // Url перенаправления после аутентификации:
      "redirect_uri": "http://X.X.X.X:1080/callback",
         //Url перенаправления после выхода из системы:
         "post_logout_redirect_uri": "http://X.X.X.X:1080/login",
         // redirect_uri, post_logout_redirect_uri – должны вести на один и тот же хост (данного приложения);

         "silent_redirect_uri": "http://X.X.X.X1080/silent-callback",
```

```
        // Идентификатор данного приложения:
        "client_id": "sophie_development",

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

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

      "useLocalStorage": true,
      "automaticSilentRenew": true
   },
   // Источник навигационных данных (обычно это хаб):
   "navigation": {
      "api": "https://X.X.X.X:1090/api/navigation",
      "domain": "Sophie"
   },
   // Источник данных о пользовательских настройках грида (обычно это
хаб):
   "gridState": {
      "api": "https://X.X.X.X:1090/api/grid-state"
   },
   // Сервис файлового хранилища:
   "filestorage": {
      "fileStorageUrl": "http://X.X.X.X:1080/api/file-storage-proxy/",
      "maxAllowedFileSize": 1073741824
   }
}
```

**Конфигурирование гридов**

В папке с дистрибутивом приложения должна располагаться папка view-configs, которая содержит список файлов в формате JSON.

#### 2.3.3.2. 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:1090",
    "ApiName": "sophie",
    "ApiSecret": "sophieSecretKey",
    "ClaimType": "uri://schemas.quarta.su/permission-claim-type"
  },
  // Параметры взаимодейтсвия 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",
    // База данных ФИАС
    "FiasConnection": "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)
  "CorsOrigins": [
    "http://X.X.X.X:1090",
    "http://X.X.X.X:1090"
  ],
  // Параметры логирования
  "Logging": {
    "LogLevel": {
      "Default": "Warning",
      "System": "Warning",
      "Microsoft": "Warning"
    }
  },
  // Доступные в интерфейсе варианты степеней родства
  "KinshipCodes": {
    "Values": "101;102;105;106"
  },
  // Периодически выполняемые задачи
  "HostedServices": {
    "Items": [
      {
        // Пересчёт трудовой деятельности
        "Key": "CurrentWorkActivityCalculation",
        "Enabled": true,
```

```
      "Interval": "1.00:00:00"
    },
    {
      // Пересчёт статусов в истории назначений кандидатов на должность
      "Key": "PersonalFileState",
      "Enabled": true,
      "Interval": "1.00:00:00",
      "PreferredTime": "23:00:00"
    }
  ]
 },
 // Параметры сервера эл. почты. Необходимо для рассылки писем
 "EmailConfig": {
  "Host": X.X.X.ru",
  "Port": 465,
  "IsSsl": false,
  "Email": "X@X.X",
  "UserName": "X",
  "Password": "X",
  "CheckCertificateRevocation": false
 }
}
```

## 2.4. Установка и настройка сервиса зарплаты

### 2.4.1. Подготовка СУБД

Дополнительных настроек производить не требуется. Достаточно выполненных действий в соответствии с п. 2.3.1.

### 2.4.2. Развёртывание .Net сервисов в linux

#### 2.4.2.1. Установка .Net

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

#### 2.4.2.2. Настройка бэкенд-сервисов

Выполнить следующие действия:

- Копировать дистрибутив сервиса *(\Дистрибутивы сервисов\Зарплата\)* на сервер *(/home/user/hosting/wage/)*.
- 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/wage.service. Указать директории, пути к файлам и приложениям, описание и указать пользователя под кем будет запускать приложение:

```
[Unit]
Description=Wage

[Service]
WorkingDirectory=/home/user/hosting/wage/
ExecStart=/usr/bin/dotnet /home/user/hosting/wage/Quarta.Wage.WebApi.dll
Restart=always
RestartSec=10
SyslogIdentifier=dotnet-wage-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

#### 2.4.2.3. 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       1070;
                location / {
                    proxy_pass http://127.0.0.1:5070;
                }
 }
```

#### 2.4.2.4. Перезапуск служб

Выполнить команды:
  systemctl restart nginx

systemctl restart wage

### 2.4.3. Настройка сервиса зарплаты

#### 2.4.3.1. Настройка конфигурации фронтенд

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

```
{
  // API бэкенда:
  "serverUrl": "http://X.X.X.X:1070",
  "authentication": {

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

      // Url перенаправления после аутентификации:
    "redirect_uri": "http://X.X.X.X:1070/callback",
      //Url перенаправления после выхода из системы:
      "post_logout_redirect_uri": "http://X.X.X.X: 1070/login",
      // redirect_uri, post_logout_redirect_uri – должны вести на один и тот
же хост (данного приложения);

      "silent_redirect_uri": "http://X.X.X.X: 1070/silent-callback",

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

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

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

    "useLocalStorage": true,
    "automaticSilentRenew": true
  },
   // Источник навигационных данных (обычно это хаб):
  "navigation": {
    "api": "https://X.X.X.X:1090/api/navigation",
    "domain": "Wage"
  },
  // Источник данных о пользовательских настройках грида (обычно это
хаб):
```

```
  "gridState": {
    "api": "https://X.X.X.X:1090/api/grid-state"
  }
}
```

**Конфигурирование гридов**

В папке с дистрибутивом приложения должна располагаться папка view-configs, которая содержит список файлов в формате JSON.

#### 2.4.3.2. Настройка конфигурации .Net сервиса

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

```
{
 // Хост сервиса
 "ApplicationUrls": [
  "http://localhost:5070"
 ],
 // Параметры работы с хабом (сервисом авторизации)
 "AuthServer": {
  "Url": "https://X.X.X.X:1090",
  "ApiName": "sophie",
  "ApiSecret": "sophieSecretKey",
  "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",
  // База данных Бухгалтерии
  "AccountingConnection": "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)
 "CorsOrigins": [
  "http://X.X.X.X:X",
  "http://X.X.X.X:X"
 ],
 // Параметры логирования
 "Logging": {
  "LogLevel": {
```

```
      "Default": "Warning",
      "System": "Warning",
      "Microsoft": "Warning"
    }
  },
  // Параметры сервиса файлового хранилища
  "FileStorage": {
    "Url": "http://X.X.X.X:1040/"
  }
}
```

# 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. Для проверки работоспособности Системы авторизоваться можно под тестовым пользователем (login: test, пароль: 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:

**1)** Убедиться в том, что сервисы сервера СУБД находятся в состоянии «Работает».

**2)** 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 "54
32".
postgres=#
```

Рис. 3.1.

Успешное соединение с БД свидетельствует о работоспособности сервера.

# 4. 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;
– Константы;
– Клиенты;
– Ресурсы;
– События аутентификации;
– События безопасности.

## 4.1. 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»**:

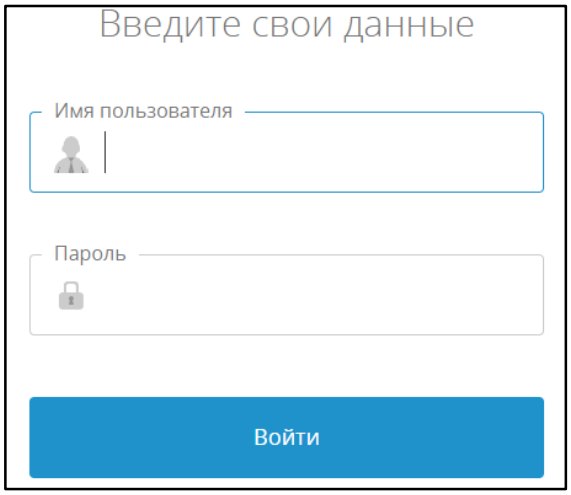


Рис. 4.1. Окно авторизации

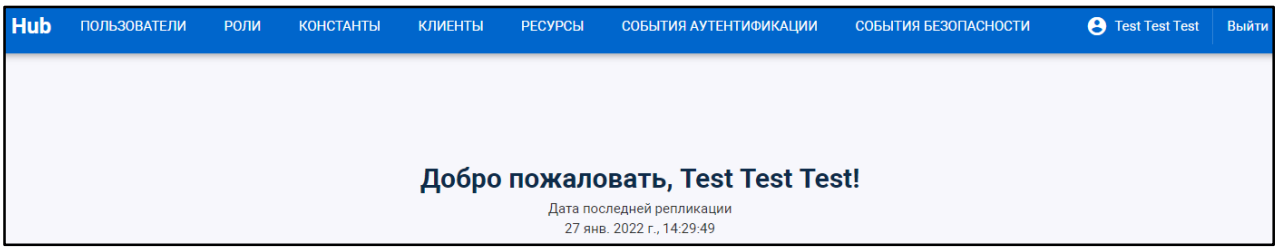


Рис. 4.2. Главная страница сервиса авторизации

## 4.2. Управление доступом

К задачам управления доступом относятся:

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

### 4.2.1.1. Приложения и сервисы

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

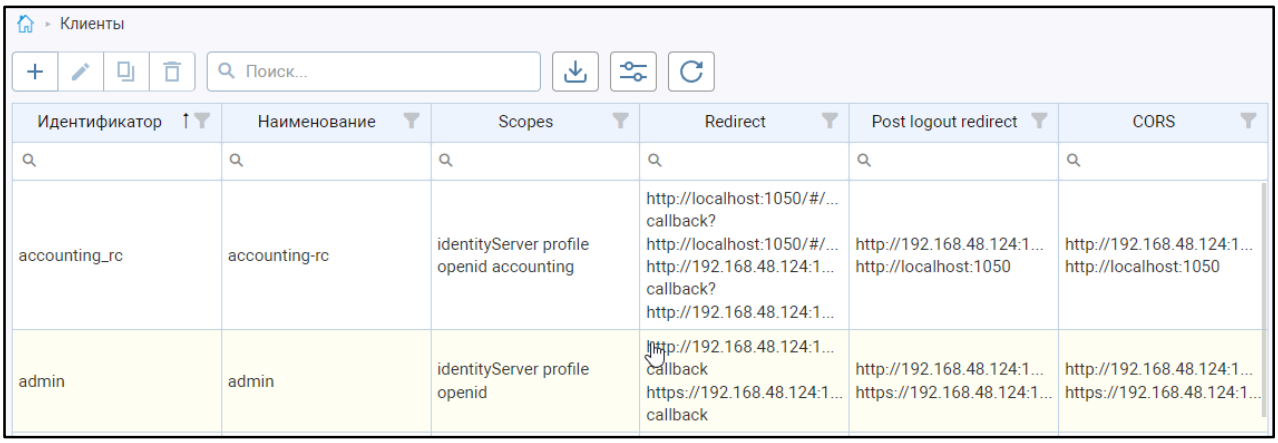


Рис. 4.3. Клиенты

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 Закрыть

Рис. 4.4. Описание клиента

Ресурсы

Поиск...

| Идентификатор | Отображаемое название |
|---|---|
| accounting | accounting |
| identityServer | identityServer |
| reporting | reporting |
| sophie | sophie |
| wage | wage |

Рис. 4.5. Ресурсы

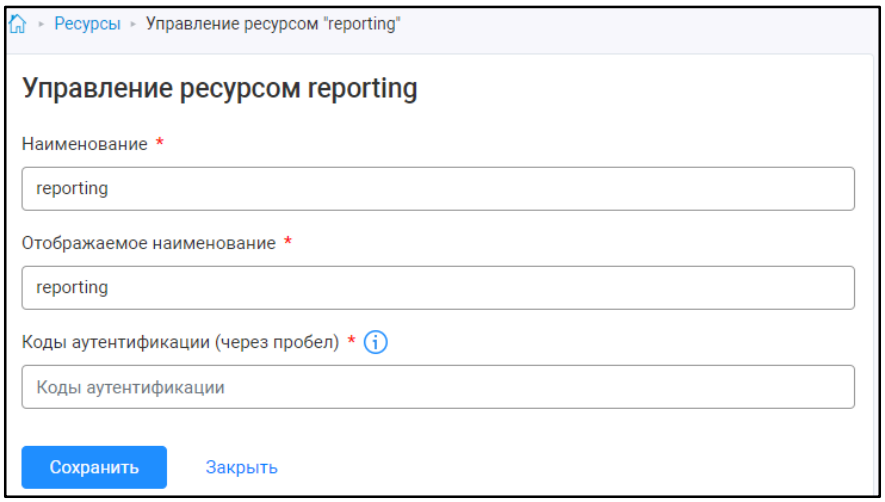


Рис. 4.6. Описание ресурса

#### 4.2.1.2. 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).

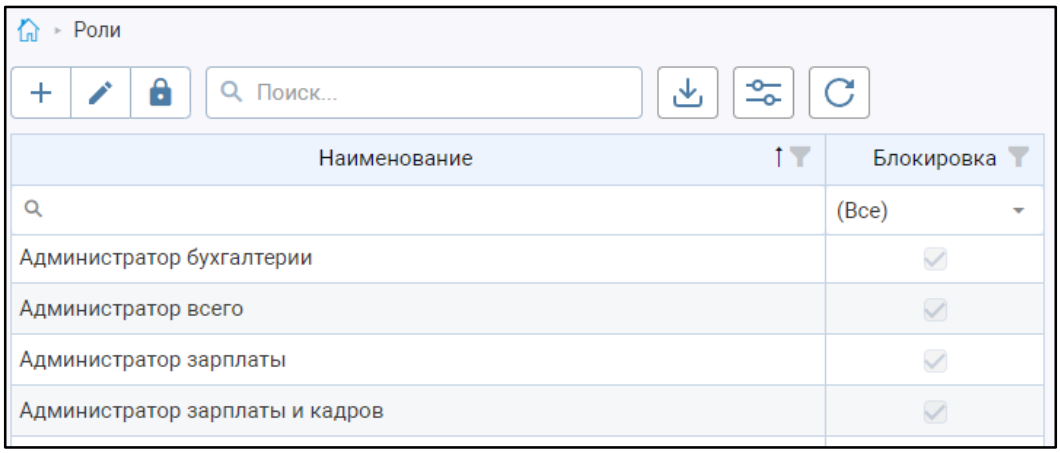


Рис. 4.7. Роли

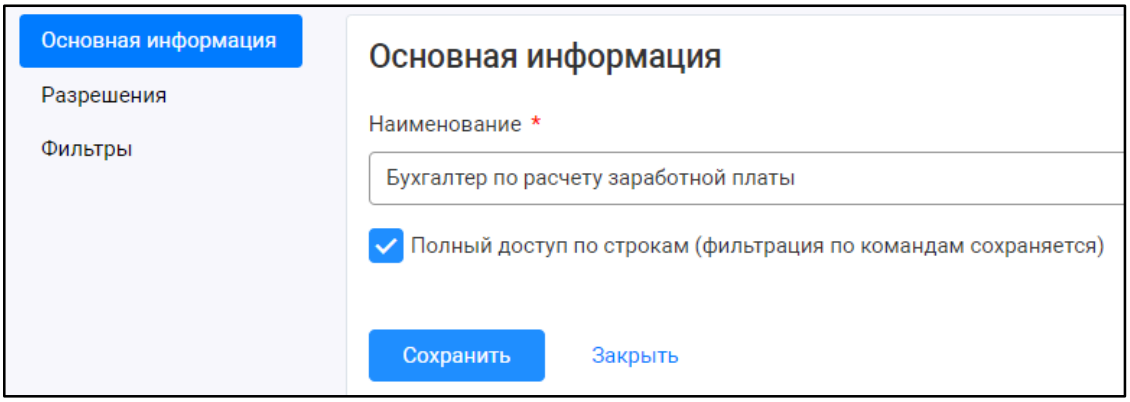


Рис. 4.8. Роль. Основная информация

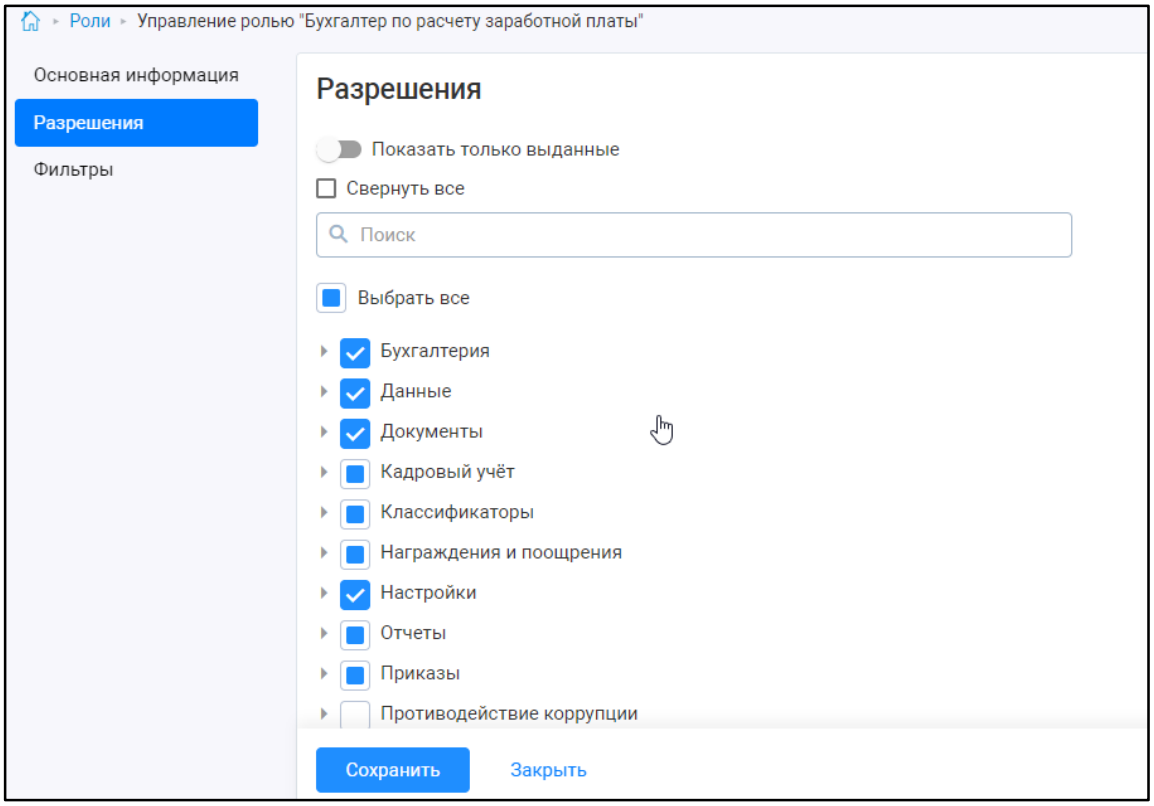


Рис. 4.9. Роль. Разрешения

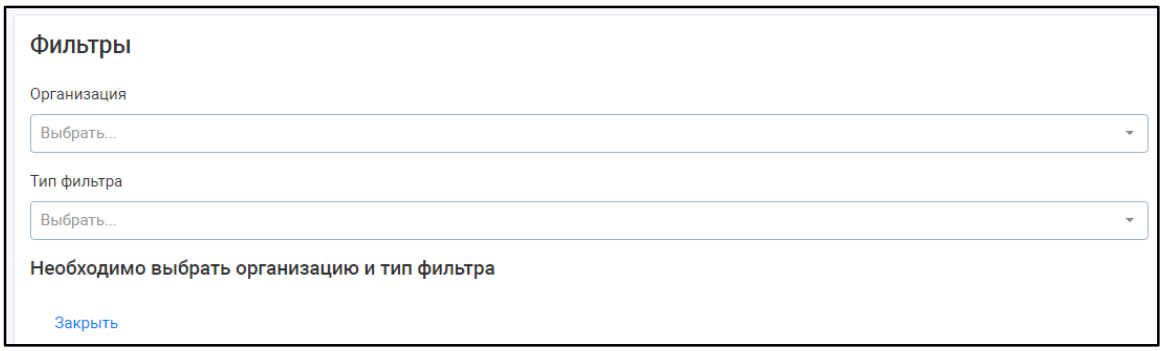


Рис. 4.10. Роль. Фильтры

### 4.2.1.3. 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).

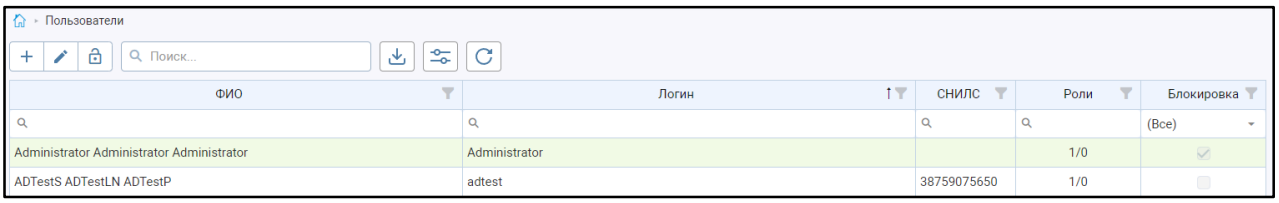


Рис. 4.11. Пользователи

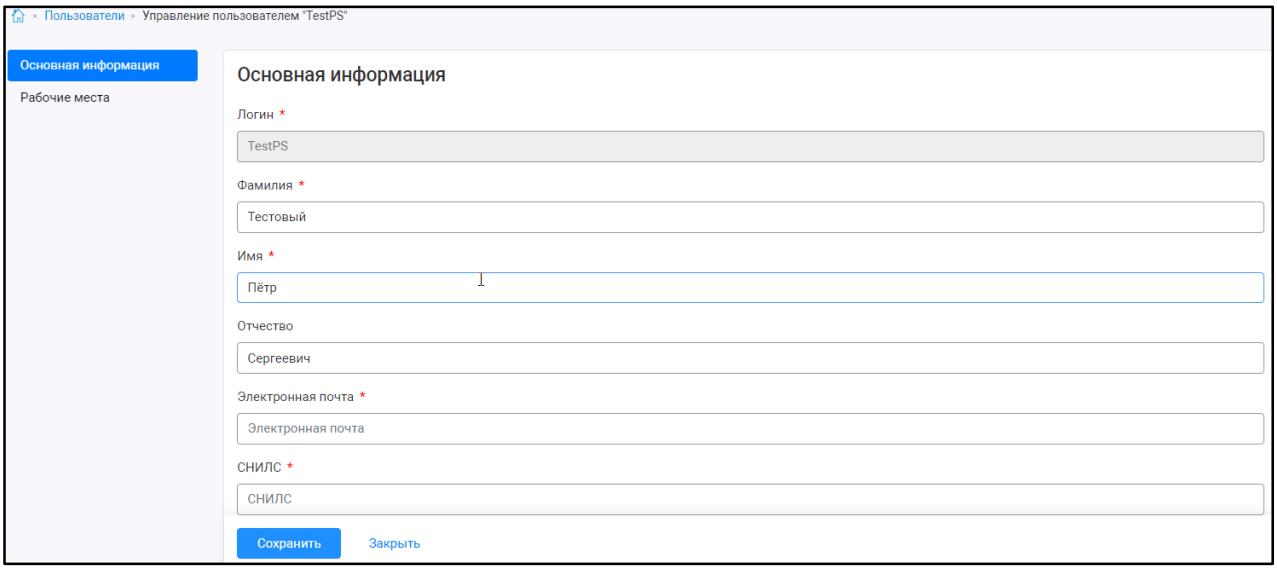


Рис. 4.12. Пользователь. Основные сведения

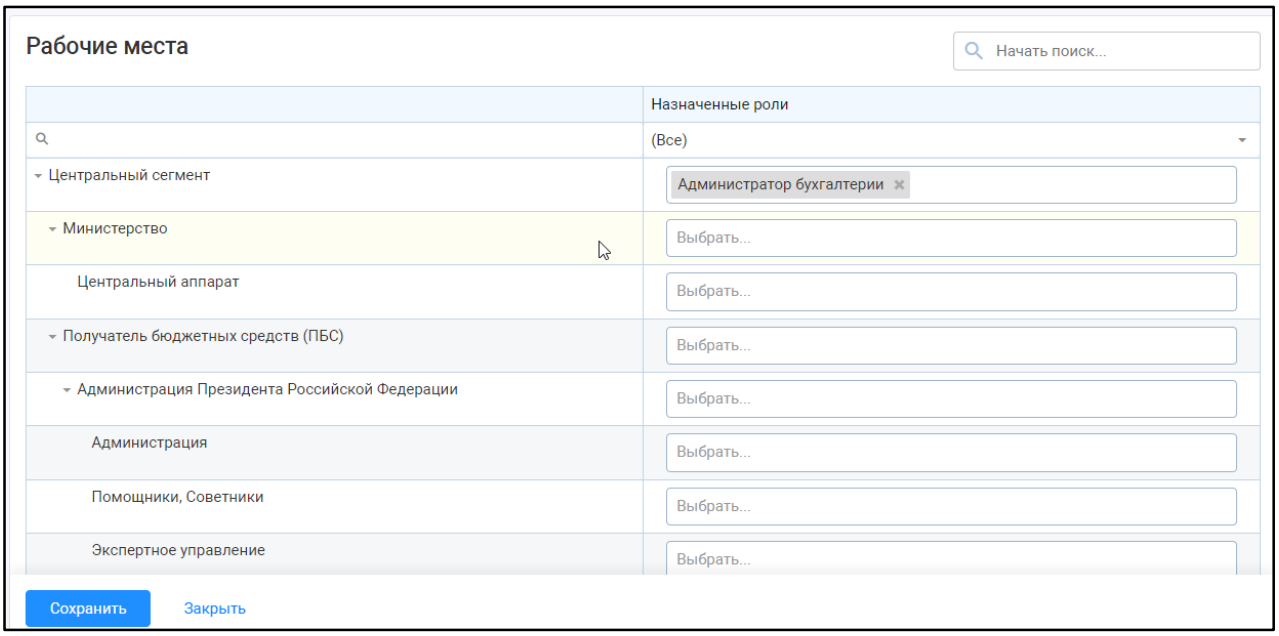


Рис. 4.13. Пользователь. Рабочие места

#### 4.2.1.4. События

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

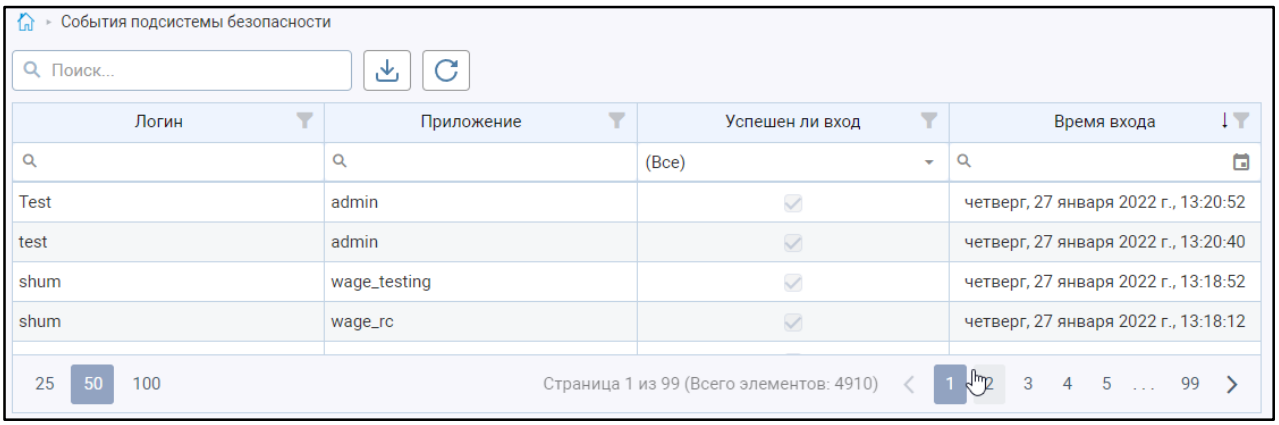


Рис. 4.14. События аутентификации

События подсистемы безопасности

Поиск...

| Логин | Прил... | Ресурс | Код права | Идентифик... | Идентифик... | Успешна ... | Описание | Врем... |
|---|---|---|---|---|---|---|---|---|
| | | | | | | (Все) | | |
| Test | sophie | sophie | uri://schemas.quarta.su/staff... | 4fdf14c6-2a79-49e2-b66d-5037659790a1 | 1e7c1215-b702-4bed-b0f7-ccb8a1e64d6a | ☑ | Вызвана проверка права (uri://schemas.quarta.su/staff/classifier/signer/read) "Чтение списка подписей" для пользователя Test. Пользователь работает под ролью "Администратор зарплаты и кадров" (4fdf14c6-2a79-49e2-b66d-5037659790a1) в организации "Управление делами Президента Российской Федерации" (1e7c1215-b702-4bed-b0f7-ccb8a1e64d6a). В доступе разрешено. | четверг, 27 января 2022 г., 17:18:09 |
| Test | sophie | sophie | uri://schemas.quarta.su/staff... | 4fdf14c6-2a79-49e2-b66d-5037659790a1 | 1e7c1215-b702-4bed-b0f7-ccb8a1e64d6a | ☑ | Вызвана проверка права (uri://schemas.quarta.su/staff/classifier/signer/read) "Чтение списка подписей" для пользователя Test. Пользователь работает под ролью "Администратор зарплаты и кадров" (4fdf14c6-2a79-49e2-b66d-5037659790a1) в организации "Управление делами Президента Российской Федерации" (1e7c1215-b702-4bed-b0f7-ccb8a1e64d6a). В доступе разрешено. | четверг, 27 января 2022 г., 17:18:07 |

25 50 100 Страница 1 из 2412 (Всего элементов: 120591) 1 2 3 4 5 ... 2412

Рис. 4.15. События безопасности

## 4.3. Настройка параметров функционирования системы

Константы – параметры функционирования системы, настраиваемые для организации / подразделения.

Перечень констант формируется компанией-разработчиком. Администратор системы со стороны Заказчика может пополнять список возможных значений константы, настраивать значения по умолчанию и разрешенные для организации/подразделения значения.

При настройке значений допускается множественный выбор.

Ресурсы › Управление ресурсом "reporting"

Управление ресурсом reporting

Наименование *

reporting

Отображаемое наименование *

reporting

Коды аутентификации (через пробел) *

Коды аутентификации

Сохранить Закрыть

Рис. 4.16. Константы

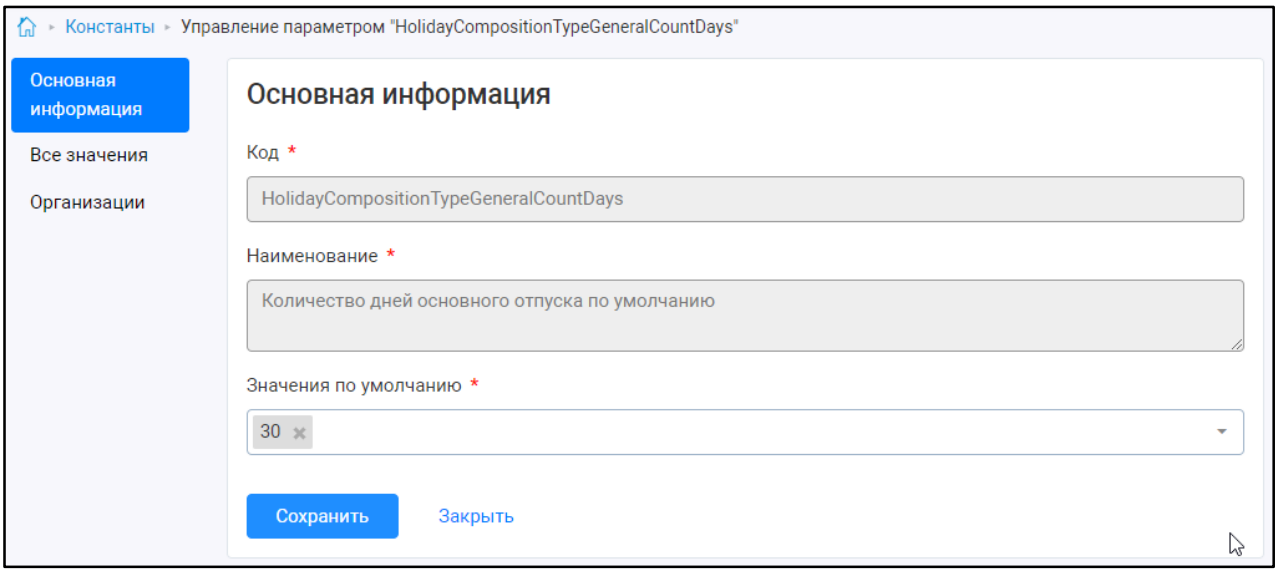


Рис. 4.17. Константа. Основная информация

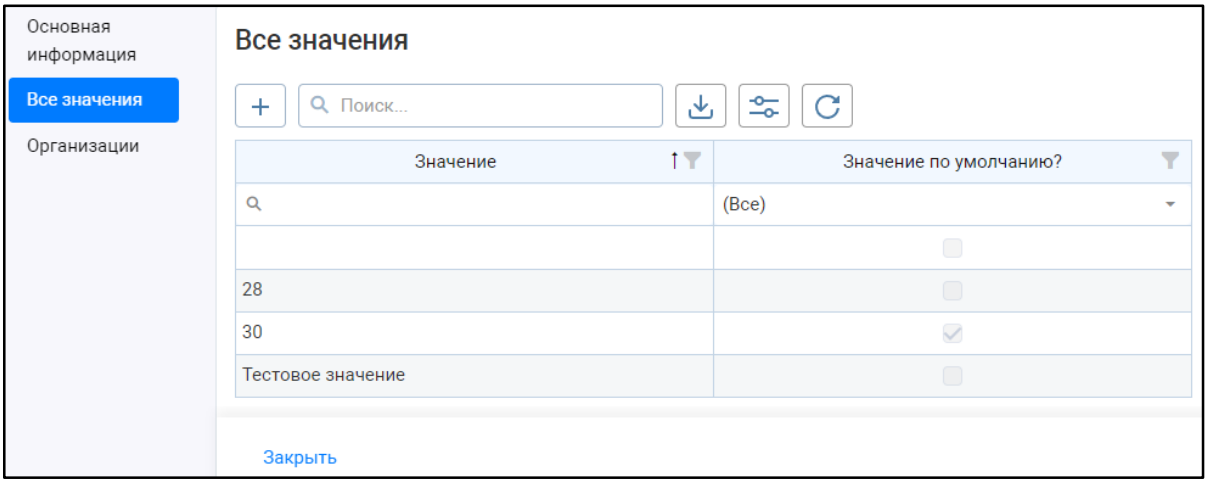


Рис. 4.18. Константа. Все значения

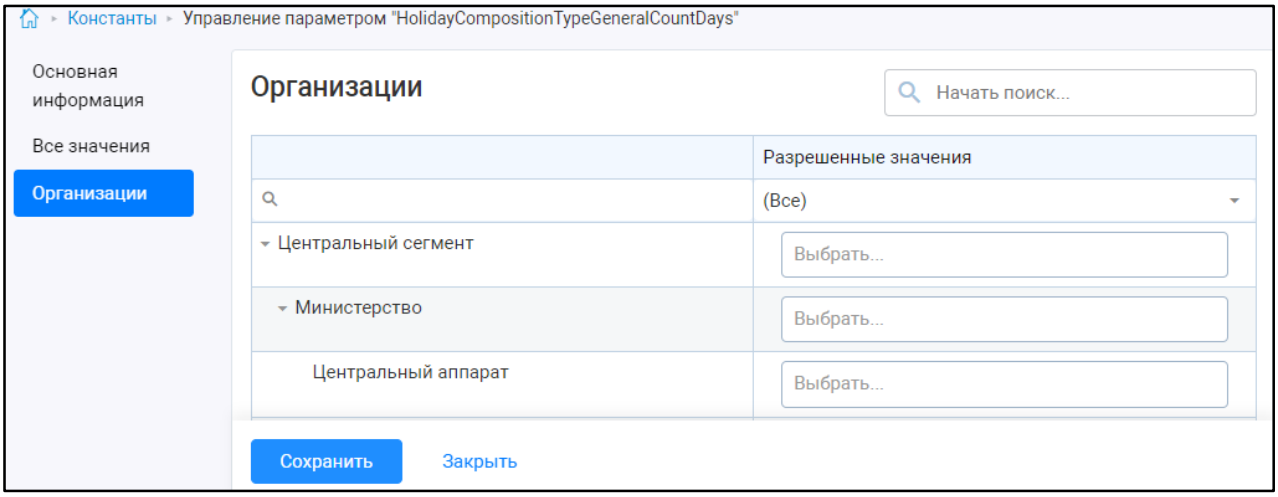


Рис. 4.19. Константа. Организации