

РУКОВОДСТВО АДМИНИСТРАТОРА

**Программное обеспечение
«Учет домашних животных»**



СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	3
1.1. Перечень инсталляционных файлов	3
1.2. Требования к аппаратному комплексу	3
1.3. Требования для установки	3
2. РАЗВЕРТЫВАНИЕ И КОНФИГУРАЦИЯ БАЗОВОГО И СПЕЦИАЛИЗИРОВАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	4
2.1. Развертывание и конфигурация PostgreSQL	4
2.2. Конфигурация платформы контейнеризации Docker CE.....	7
2.3. Конфигурация и запуск контейнера WSO2IS.....	8
2.4. Конфигурация и запуск контейнеров «Учет домашних животных» и их зависимостей.....	10
2.5. Конфигурация и запуск Nginx для web-сервера	10
3. АДМИНИСТРИРОВАНИЕ ПОЛЬЗОВАТЕЛЕЙ	14
3.1. Управление учетными записями пользователей	14
3.2. Присвоение ролей пользователям.....	15
3.3. Мониторинг действий пользователя	16



1. ВВЕДЕНИЕ

1.1. Перечень инсталляционных файлов

Пакет инсталляционных файлов содержит (Таблица 1):

- скомпилированный файл;
- файл для развертывания базы данных.

Таблица 1

№	Имя файла	Назначение
1.	animals_script.sql	SQL скрипт создания схемы базы данных
2.	animals.war pets-service.war	Скомпилированный веб-архив приложения

1.2. Требования к аппаратному комплексу

Для инсталляции программного обеспечения необходимо, чтобы на аппаратном комплексе было установлено следующее базовое лицензионное программное обеспечение третьих сторон:

- Astra linux 1.7.3;
- Nginx 1.22;
- СУБД PostgreSQL 11;
- WSO2 Identity Server 6.1;
- Redis 6.0.

1.3. Требования для установки

Программное обеспечение «Учет домашних животных» (далее – ПО) построено на основе спецификации Java Enterprise Edition. Использование данной технологии позволяет устанавливать ядро ПО «Учет домашних животных» практически на любую операционную систему (Linux), для которой существуют сервера контейнеризации, удовлетворяющие данной спецификации (docker).

В качестве СУБД для хранилища данных используются СУБД PostgreSQL.

Требования к аппаратному обеспечению зависят от количества одновременных поисковых запросов пользователей. В таблице 2 приведены минимальные требования к аппаратному обеспечению для сервера приложений с установленными СУБД Postgresql и docker на операционной системе Astra linux.

Таблица 2

Минимальные аппаратные требования к серверу приложений

Количество пользователей	Процессор	Оперативная память
1-20	Intel Pentium Xeon 2GHz	1 GB
20-50	Intel Pentium Core Duo 2GHz	2 GB
50-100	Intel Pentium Core Duo 2GHz	4 GB
>100	Intel Pentium Core Duo 2.5 GHz	8 GB



Для работы пользователей на клиентском месте необходим браузер, совместимый с Microsoft Edge. В браузере должно быть разрешено использование Cookie, выполнение Java Script.

2. РАЗВЕРТЫВАНИЕ И КОНФИГУРАЦИЯ БАЗОВОГО И СПЕЦИАЛИЗИРОВАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1. Развертывание и конфигурация и PostgreSQL

В ПО «Учет домашних животных» postgresql используется для двух контейнеров:

- контейнер аутентификации;
- контейнер «Учет домашних животных».

Для установки необходимых пакетов PostgreSQL необходимо выполнить команды:

- `sudo apt install postgresql-all`
- `sudo systemctl enable postgresql`
- `sudo systemctl start postgresql`

Основные настройки PostgreSQL находятся в двух файлах: `postgresql.conf` и `pg_hba.conf`. Эти файлы находятся в директории `/etc/postgresql/11/main/`.

В файле `postgresql.conf` содержатся параметры базы данных, а в `pg_hba.conf` – настройки доступа и идентификации пользователей.

Для ПО «Учет домашних животных» должно быть развернуто два экземпляра базы данных (**далее – БД**). Один используется для обеспечения работы контейнера аутентификации, другой для обеспечения работы остальных приложений.

Пример конфигурационного файла `postgresql.conf`:

```
ac_enable_maclabels_on_files = false
ac_ignore_maclabel          = false
ac_ignore_socket_maclabel   = true
ac_ignore_server_maclabel   = true
ac_enable_trusted_owner     = true
ac_enable_truncate          = true
ac_enable_sequence_ccr      = true
ac_enable_copy_to_file      = true
ac_auto_adjust_macs         = false
ac_enable_dblink_mac        = false
ac_enable_grant_options     = false
ac_enable_admin_options     = false
ac_caps_ttl                 = 10      # rbt capabilities Time to live, in seconds
ac_debug_print              = false
ac_audit_mode               = 'internal' # rbt audit mode. Possible values are
ac_audit_destination        = 'all'     # rbt Audit destination. Possible values are
ac_audit_log_only_failures  = false
ac_log_stmt_length          = 100     # rbt audit stmt length, 0 - unlimited
ac_wipe_buffers             = false
ac_wipe_pages               = false
ac_wipe_files               = false
```



```
data_directory = '/var/lib/postgresql/11/main'      # use data in another directory
hba_file = '/etc/postgresql/11/main/pg_hba.conf'   # host-based authentication file
ident_file = '/etc/postgresql/11/main/pg_ident.conf' # ident configuration file
external_pid_file = '/var/run/postgresql/11-main.pid' # write an extra PID file
listen_addresses = '*'          # what IP address(es) to listen on;
port = 5432                      # (change requires restart)
max_connections = 500            # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
ssl = on
ssl_cert_file = '/etc/ssl/certs/ssl-cert-snakeoil.pem'
ssl_key_file = '/etc/ssl/private/ssl-cert-snakeoil.key'
shared_buffers = 128MB          # min 128kB
temp_buffers = 256MB           # min 800kB
work_mem = 64MB                # min 64kB
dynamic_shared_memory_type = posix # the default is the first option
max_files_per_process = 8000    # min 25
max_wal_size = 1GB
min_wal_size = 80MB
from_collapse_limit = 20
join_collapse_limit = 20       # 1 disables collapsing of explicit
log_destination = 'stderr'     # Valid values are combinations of
logging_collector = on         # Enable capturing of stderr and csvlog
log_directory = 'pg_log'       # directory where log files are written,
log_filename = 'postgresql-%a.log' # log file name pattern,
log_truncate_on_rotation = on  # If on, an existing log file with the
log_rotation_age = 1d          # Automatic rotation of logfiles will
log_rotation_size = 0          # Automatic rotation of logfiles will
log_line_prefix = '%m [%p] %q%u@%d ' # special values:
log_timezone = 'Asia/Yekaterinburg'
cluster_name = '11/main'       # added to process titles if nonempty
stats_temp_directory = '/var/run/postgresql/11-main.pg_stat_tmp'
timezone = 'Asia/Yekaterinburg'
lc_messages = 'ru_RU.UTF-8'    # locale for system error message
lc_monetary = 'ru_RU.UTF-8'    # locale for monetary formatting
lc_numeric = 'ru_RU.UTF-8'     # locale for number formatting
lc_time = 'ru_RU.UTF-8'        # locale for time formatting
default_text_search_config = 'pg_catalog.russian'
shared_preload_libraries = 'online_analyze, plantuner' # (change requires restart)
max_locks_per_transaction = 150 # min 10
escape_string_warning = off
standard_conforming_strings = off
include_dir = 'conf.d'         # include files ending in '.conf' from
online_analyze.threshold = 50
online_analyze.scale_factor = 0.1
online_analyze.enable = off
online_analyze.verbose = off
online_analyze.local_tracking = on
online_analyze.min_interval = 10000
online_analyze.table_type = 'temporary'
plantuner.fix_empty_table = on
```



В конфигурационном файле `pg_hba.conf` необходимо разрешить доступ модулям «Учет домашних животных».

Пример конфигурационного файла `pg_hba.conf`:

```
local all postgres peer
local all all peer
host all all 0.0.0.0/0 md5
host all all ::1/128 md5
local replication all peer
host replication all 127.0.0.1/32 md5
host replication all ::1/128 md5
```

После запуска сервиса БД необходимо выполнить процедуру создания базы данных, и создать пользователей (роли) для подключения к базе данных.

Создание баз данных осуществляется следующей SQL командой:

```
create database wso2is;
create database animals
```

Создание пользователя (роли) для ПО «Учет домашних животных» и назначение прав осуществляется следующей SQL командой:

```
CREATE ROLE animals WITH
NOSUPERUSER
NOCREATEDB
NOCREATEROLE
INHERIT
LOGIN
NOREPLICATION
NOBYPASSRLS
CONNECTION LIMIT -1
PASSWORD 'пароль';
CREATE SCHEMA animals AUTHORIZATION animals;
GRANT ALL ON SCHEMA animals TO animals;
```

Создание пользователя (роли) для сервиса аутентификации осуществляется следующей SQL командой:

```
CREATE ROLE wso2is WITH
NOSUPERUSER
NOCREATEDB
NOCREATEROLE
NOINHERIT
```



```
LOGIN
NOREPLICATION
NOBYPASSRLS
CONNECTION LIMIT -1
PASSWORD 'пароль';
```

После выполнения создания роли wso2is, необходимо выполнить скрипты первоначального наполнения БД. Скрипт wso2is.sql передается в составе инсталляционных файлов.

Для выполнения скрипта необходимо выполнить команду:

```
psql -U wso2is -f wso2is.sql -d wso2is -h localhost
```

2.2. Конфигурация платформы контейнеризации Docker CE

Платформа контейнеризации приложений Docker CE используется для упаковки сервисов ПО «Учет домашних животных» и всех его зависимостей вместе, и запуска их в изолированных контейнерах.

Контейнер (Docker Container) представляет собой окружение для выполнения какого-либо одного процесса. Контейнер создается на основе образа (Docker Image). Образ Docker – это неизменяемый пакет файлов, содержащий библиотеки, зависимости, инструменты и другие файлы, необходимые для запуска приложения. Сервисы «Учет домашних животных» поставляются в виде готовых образов, упакованных в tar архив.

Для базовой установки необходимых пакетов Docker нужно выполнить следующие команды:

```
sudo apt install docker docker-compose
sudo systemctl enable docker
sudo systemctl start docker
```

Перед созданием контейнеров, в локальное хранилище docker необходимо загрузить все необходимые файлы образов. Для этого файлы архивов копируются на сервер в каталог /opt/images/. Краткое описание образов приведено ниже в таблице.

Имя файла образа	Описание
wso2is.tar	docker образ сервера аутентификации
redis.tar	docker образ сервиса redis
pets-client.tar	docker образ модуля клиентской части приложения
pets-api.tar	docker образ модуля серверной части приложения
openresty.tar	docker образ веб сервера nginx с модулем Lua

Делается это с помощью команд:

```
docker load -i /opt/images/wso2is.tar
docker load -i /opt/images/redis.tar
```



```
docker load -i /opt/images/pets-api.tar
docker load -i /opt/images/pets-client.tar
docker load -i /opt/images/openresty.tar
```

2.3. Конфигурация и запуск контейнера WSO2IS

Шаг 1. Скопировать и распаковать архив wso2is.zip в /opt/ и перейти в директорию /opt/wso2is

Шаг 2. Скорректировать параметры в файле conf/deployment.toml

hostname = "" – имя, по которому будет доступен сервер из вне

node_ip = - IP адрес сервера

assertion_consumer_service_url = "https://hostname/acs"

identity_provider_sso_service_url = "https://hostname/samlssso"

Так же скорректировать параметры в файле docker-compose.yml:

ADMIN_PASS= - пароль администратора

DB_PORT= - порт службы postgresql

DB_HOST= имя или ip адрес службы postgresql

DB_USER= Название роли созданной на шаге 1.1.1

DB_PASS= Пароль роли созданной на шаге 1.1.1

DB_NAME= Название бд созданной на шаге 1.1.1

Создать сеть docker для сервиса wso2is:

```
docker network create wso2-network
```

Шаг 3. Запустить контейнер:

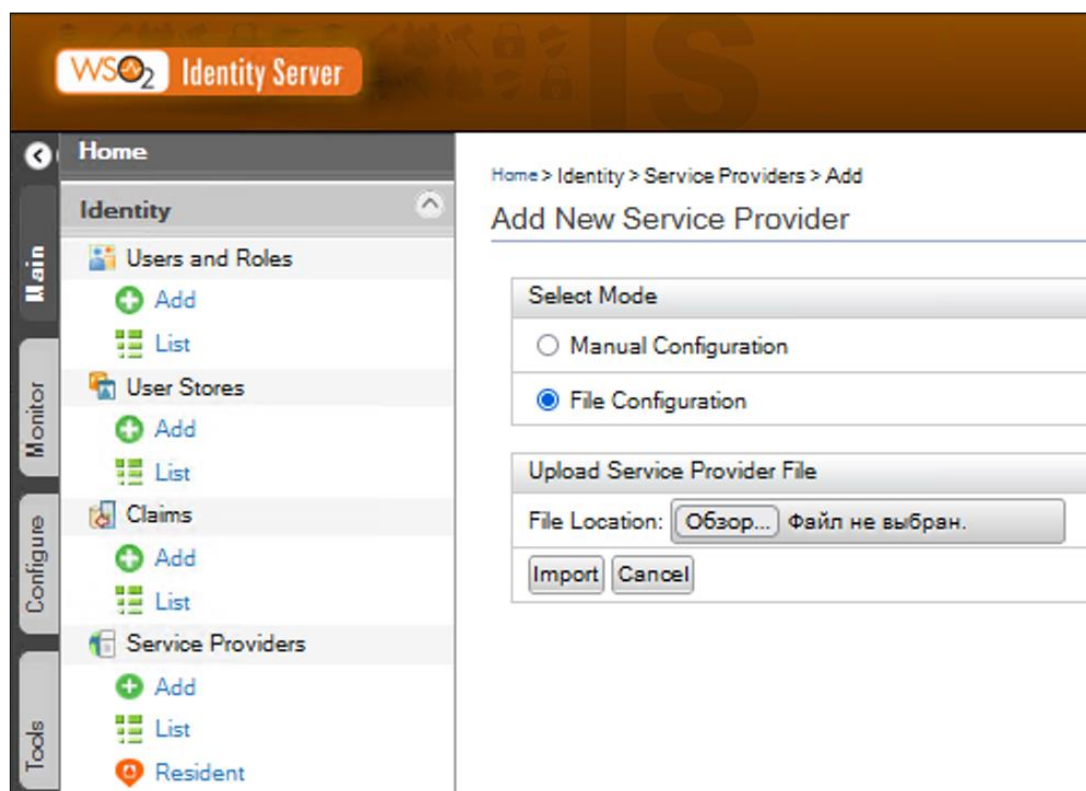
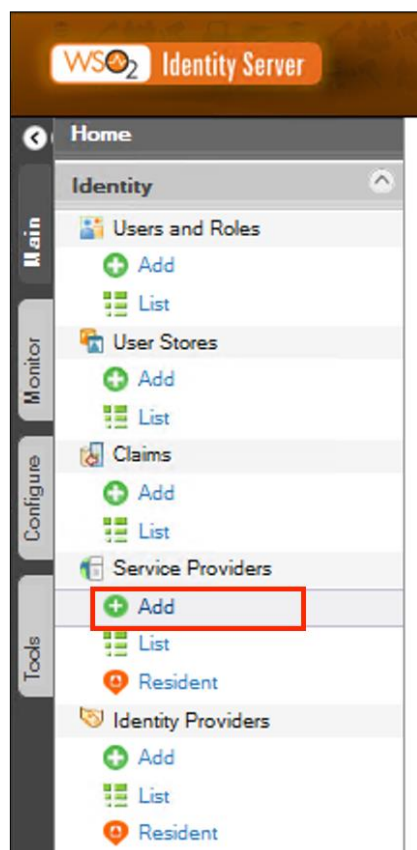
```
Docker-compose up -d
```

Шаг 4. Настроить поставщика услуг аутентификации

Раздел Service Providers содержит основные настройки о способе аутентификации пользователей в приложениях.

Настройка сервис провайдера осуществляется в веб-консоли сервера идентификации <https://carbon>. Веб-консоль будет доступна после настройки и запуска веб-сервера nginx (пункт 1.52.5).

В меню Service Providers выберете Add и загрузите конфигурацию из файла pets.xml.





2.4. Конфигурация и запуск контейнеров «Учет домашних животных» и их зависимостей

Шаг 1. Скопировать и распаковать архив animals.zip в /opt/ и перейти в директорию /opt/animals

Шаг 2. Скорректировать параметры в файле .env и .proхu.env

DATASOURCE_HOST – имя сервера БД;

DATASOURCE_PORT – порт БД;

DATASOURCE_DB – имя БД;

DATASOURCE_SCHEMA – схема «Учет домашних животных» в БД;

DATASOURCE_USER – имя пользователя БД;

DATASOURCE_PASSWORD – пароль пользователя БД;

MAIL_HOST – имя хоста исходящего почтового сервера;

MAIL_PORT – используемый порт исходящих подключений;

MAIL_PASS – пароль от учётной записи;

MAIL_USER – учетная запись исходящих подключений;

MAIL_SMTP_FROM – почта для исходящих подключений;

REDIS_HOST – имя сервера Redis;

REDIS_PORT – используемый Redis порт;

REDIS_PASSWORD – пароль для доступа к Redis;

WSO2_URL – URL сервера идентификации WSO2;

WSO2_CLIENT_ID – публичный идентификатор OAuth клиента;

WSO2_CLIENT_SECRET – приватный токен OAuth клиента;

BASED_DOMAIN – внешнее имя «Учет домашних животных»

SESSION_SECRET – ключ шифрования сессии

OAUTH_CLIENT_ID= публичный идентификатор OAuth клиента;

OAUTH_CLIENT_SECRET - приватный токен OAuth клиента;

Шаг 3. Создать сеть docker для сервиса animals:

```
docker network create pets-network
```

Шаг 4. Запустить контейнеры:

```
Docker-compose up -d
```

2.5. Конфигурация и запуск Nginx для web-сервера

ПО «Учет домашних животных» использует два экземпляра Nginx. Один используется как прокси сервер для доступа ко всем сервисам. Второй используется для аутентификации пользователей при обращении к сервисам специализированного программного обеспечения.

Для установки nginx выполните следующие команды:

```
sudo apt install nginx
```



```
sudo systemctl enable nginx
```

Настойка nginx осуществляется с помощью двух конфигурационных файлов:

/etc/nginx/nginx.conf – основной файл конфигурации;

/etc/nginx/conf.d/pets.conf – дополнительный файл конфигурации.

Содержимое файла nginx.conf:

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;
load_module modules/nginx_http_vhost_traffic_status_module.so;
load_module modules/nginx_http_headers_more_filter_module.so;
load_module modules/nginx_http_upstream_ntlm_module.so;

events {
    worker_connections 1024;
}
http {
    vhost_traffic_status_zone;

    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_nodelay on;
    keepalive_timeout 60;
    types_hash_max_size 2048;
    server_names_hash_max_size 1024;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=backcache:8m
max_size=50m;
    proxy_cache_key "$scheme$request_method$host$request_uri$is_args$args";
    proxy_cache_valid 200 302 10m;
    proxy_cache_valid 404 1m;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/nginx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;

    index index.html;
}
```



Содержимое файла pets.conf:

```
upstream openresty-pets-bk {
    server 192.168.55.139:80;
    keepalive 100;
}
upstream wso-pets-bk {
    server 192.168.55.139:9443;
    keepalive 100;
}
server {
    listen 80;
    server_name pets.technocom.tech;

    if ($host = pets.technocom.tech) {
        return 301 https://$host$request_uri;
    }
}
server {
    listen 443 ssl http2;
    server_name pets.technocom.tech;

    proxy_read_timeout 15m;
    proxy_send_timeout 15m;

    client_max_body_size 2G;
    large_client_header_buffers 4 32k;

    ssl_certificate /etc/nginx/ssl/certificate.crt;
    ssl_certificate_key /etc/nginx/ssl/private.key;

    access_log /var/log/nginx/pets.technocom.tech-ssl-access.log;
    error_log /var/log/nginx/pets.technocom.tech-ssl-error.log;

    proxy_redirect ~^http:\V(\w+\.technocom\.tech)(:\d+)?(V.*)$ https://$1$3;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_set_header X-Forwarded-Proto $scheme;

    location / {
        rewrite ^/animals/ permanent;
    }

    location ~* ^(login|logout)($|V) {
        proxy_pass http://openresty-pets-bk;
    }
    location ~* ^(animals|pets-api)($|V) {
        proxy_pass http://openresty-pets-bk;
    }
}
```



```
}  
location ~* ^(carbon|console|myaccount) {  
    proxy_pass https://wso-pets-bk;  
}  
location /t/ {  
    proxy_pass https://wso-pets-bk;  
}  
location ~*  
^(authenticationendpoint|logincontext|commonauth|accountrecoveryendpoint|oauth2|  
api|oidc|scim|iwa-kerberos|saml|sso|acs|samlartresolve) {  
    proxy_pass https://wso-pets-bk;  
}  
}
```



3. АДМИНИСТРИРОВАНИЕ ПОЛЬЗОВАТЕЛЕЙ

3.1. Управление учетными записями пользователей

Для того чтобы приступить к администрированию пользователей, необходимо выбрать «Реестр пользователей» (Рисунок 1).

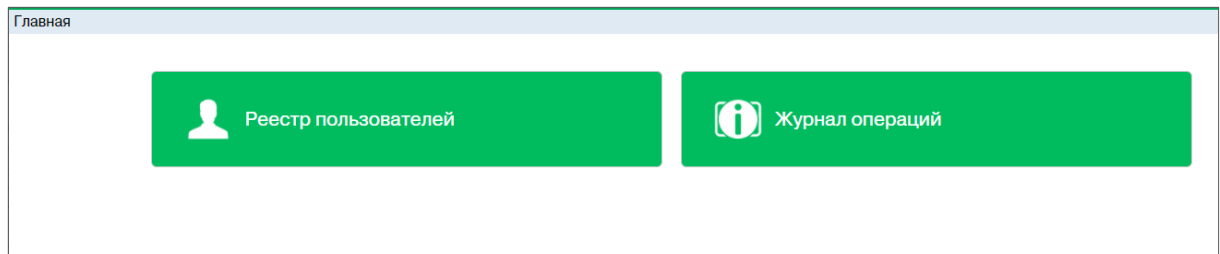


Рисунок 1. Реестры

ПО «Учет домашних животных» отобразит форму со списком зарегистрированных пользователей (Рисунок 2). Учетные записи пользователей создаются в репозитории WSO2 Identity Server, после чего синхронизируются с базой данных ПО «Учет домашних животных». Помимо логина и пароля информация о пользователе содержит его e-mail адрес для осуществления рассылки системных сообщений в случае, если это необходимо.

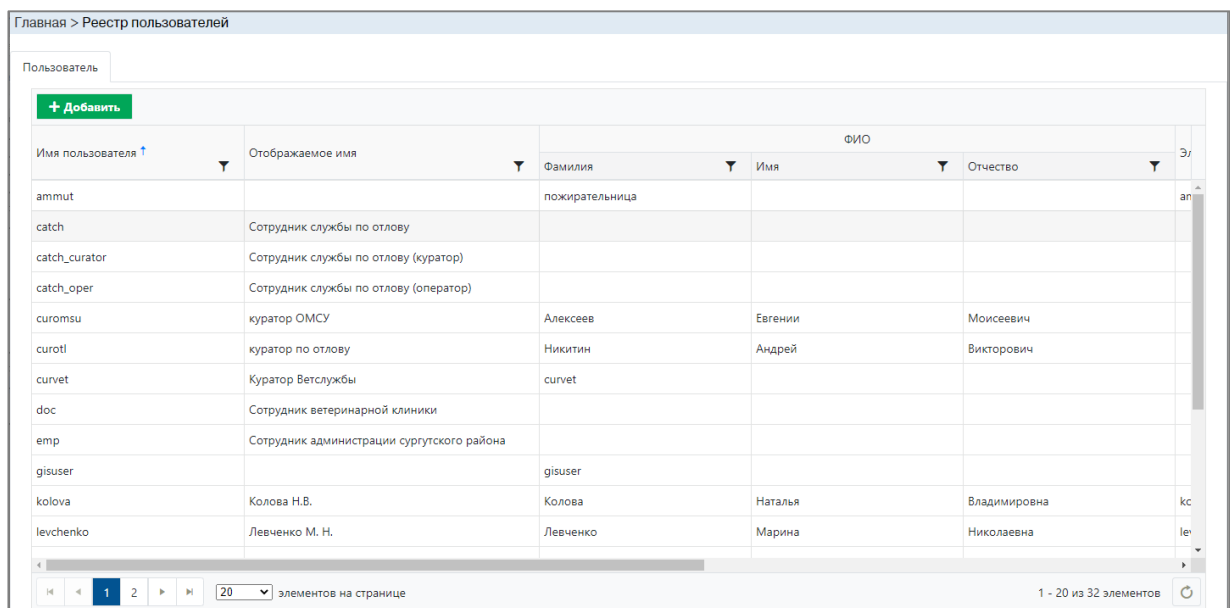


Рисунок 2. Страница со списком пользователей

Для создания новой учетной записи пользователя на странице реестра пользователей необходимо выбрать действие «Добавить».

В форме для ввода информации о пользователе на вкладке «Информация» в разделе «Информация о пользователе» (Рисунок 3) заполнить все поля, а также задать пароль для входа (пользователь сможет сменить пароль через форму аутентификации, ссылка для смены пароля будет направлена ему на указанный адрес электронной почты).



Пользователь

Информация Роли

Информация о пользователе

Логин *

Пароль

Отображаемое имя

Телефон *

+7() - - - -

Адрес электронной почты

Фамилия *

Имя *

Сохранить Закрыть

Рисунок 3. Страница создания учетной записи пользователя

В разделе «Организация» выбрать действие «Выбрать...» и в открывшемся окне со списком организаций выбрать организацию, с которой будет связана учетная запись пользователя (Рисунок 4).

Главная > Реестр пользователей

Пользователь

+ Добавить

Имя пользователя ↑

ammut

catch

catch_curator

catch_oper

curomsu

currotl

curvet

doc

emp

gisuser

kolova

levchenko

Пользователь

Информация Роли

Имя *

Отчество *

Должность *

Отдел *

Активная запись

Организация

Выбрать... Удалить

Наименование организац

Организации

ИНН	КПП	Муниципальное образование	Тип организации	ИП/юрлицо
11078344	860101001	город Сургут	Орган местного самоуправления	Юридическое лицо
8608040971	860801001	город Когалым	Орган местного самоуправления	Юридическое лицо
860801889350		г Лянтор	Организация по отлову	Индивидуальный предпр

1 2 20 элементов на странице 1 - 20 из 22 элементов

Выбрать Закрыть

Сохранить Закрыть

Рисунок 4. Назначение организации для учетной записи пользователя

3.2. Присвоение ролей пользователям

Состав информации и функциональных возможностей ПО «Учет домашних животных»



животных», отображаемых в окне интерфейса, зависит от роли пользователя, которая определяет доступность информации и функциональных возможностей ПО «Учет домашних животных».

Для того чтобы назначить те или иные роли зарегистрированному в ПО «Учет домашних животных» пользователю, необходимо в Реестре пользователей выбрать интересующего пользователя, открыть его карточку двойным нажатием левой клавиши мыши и на вкладке «Роли» отметить необходимые роли (Рисунок 5). Общие права доступа пользователя будут рассматриваться как объединение множеств, представленных отдельными ролями.

Название роли
<input type="checkbox"/> Администратор данных
<input type="checkbox"/> Администратор пользователей
<input type="checkbox"/> Ветврач
<input type="checkbox"/> Ветврач приюта
<input type="checkbox"/> Куратор ВетСлужбы
<input type="checkbox"/> Куратор по отлову
<input type="checkbox"/> Оператор
<input type="checkbox"/> Оператор ветклиники
<input type="checkbox"/> Оператор ВетСлужбы
<input type="checkbox"/> Оператор по отлову
<input type="checkbox"/> Оператор приюта

Рисунок 5. Форма установки ролей пользователю

Для сохранения изменений необходимо нажать кнопку «Сохранить».

3.3. Мониторинг действий пользователя

Данный раздел предоставляет возможность просмотра действий пользователей, а также ведения статистики о работе пользователей в ПО «Учет домашних животных», объемах и динамике загрузки информации.

Для перехода в раздел мониторинга действий пользователей необходимо на главной странице выбрать «Журнал операций». ПО «Учет домашних животных» отобразит форму со списком операций пользователей (Рисунок 6). Имеющиеся на странице фильтры позволяют вывести список операций по определенному пользователю, уровню или методу.



Главная > Журнал операций

Удалить Экспорт в Excel

Дата	Имя пользователя	Уровень	Метод	Путь	Запрос	Тело запроса	Ответ
8 июн. 2020 г., 11:54:17	curvet	ERROR	POST	/auth-controller/role		Role(id=null, name=test, users=null)	Тест
20 окт. 2020 г., 10:08:18	ammut	ERROR	POST	/auth-controller/roles	page=0&size=20		could not execute statement; SQL [n/a]; nested exception is org.hibernate.exception.DataException: could not execute statement
20 окт. 2020 г., 10:16:44	ammut	ERROR	POST	/auth-controller/roles	page=0&size=20		could not execute statement; SQL [n/a]; nested exception is org.hibernate.exception.DataException: could not execute statement
20 окт. 2020 г., 10:20:18	kolova	INFO	POST	/auth-controller/roles	page=0&size=20		[{"content":[{"id":1,"name":"Администратор данных"}, {"id":2,"name":"Администратор пользователей"}, {"id":3,"name":"Ветврач"}, {"id":4,"name":"Ветврач приюта"}, {"id":5,"name":"Куратор ВетСлужбы"}, {"id":7,"name":"Куратор по отлову"}, {"id":8,"name":"Оператор"}, {"id":9,"name":"Оператор ветклиники"}, {"id":10,"name":"Оператор ВетСлужбы"}, {"id":12,"name":"Оператор по отлову"}, {"id":13,"name":"Оператор приюта"}, {"id":14,"name":"Подписант ВетСлужбы"}, {"id":16,"name":"Подписант по отлову"}, {"id":17,"name":"Распределитель субвенции"}, {"id":18,"name":"Куратор приюта"}, {"id":19,"name":"Подписант приюта"}, {"id":20,"name":"Гость или любая роль"}, {"id":21,"name":"Супер админ с полным доступом"}, {"id":6,"name":"Куратор ОМСУ МО"}]

1 20 элементов на странице 1 - 12 из 12 элементов

Рисунок 6. Форма мониторинга действий пользователей