

**Информационно-аналитическая система централизованного управления
имуществом, находящимся в собственности субъектов Российской
Федерации, муниципальных образований, а также имуществом
государственных компаний
(«БФТ.Управление активами»)**

**Инструкция по скачиванию, установке экземпляра программного
обеспечения и запуску системы «БФТ.Управление активами»**

Листов 23

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
1 Скачивание дистрибутива	3
2 Установка системы «БФТ.Управление активами».....	3
2.1 Установка программы для операционной системы Linux	3
2.1.1 Установка платформы OpenJRE	3
2.1.2 Установка СУБД PostgreSQL 12.....	3
2.1.3 Установка сервера приложений Tomcat	4
2.1.4 Установка Elasticsearch.....	9
2.1.5 Установка ZomboDB.....	11
2.1.6 Установка PostGIS	11
2.1.7 Создание базы приложения	12
2.1.8 Установка ApacheMQ.....	13
2.1.9 Установка сервера геоданных GeoServer	14
2.1.10 Настройка Tomcat	16
2.2 Настройка программы на условия конкретного применения	20
2.2.1 Настройка программы на условия конкретного применения	20
2.2.2 Настройка программы на условия конкретного применения для комплекса интеграции	21
3 Запуск Системы хранения и предоставления электронных документов («БФТ.Управление активами»).....	22

1 Скачивание дистрибутива

Скачивание дистрибутива ИС «БФТ.Управление активами» доступно по ссылке: <https://cloud.bftcom.com/owncloud/index.php/s/X12uYEDmsX07I8f>

Пароль для скачивания дистрибутива: Pp2182021

2 Установка системы «БФТ.Управление активами»

2.1 Установка программы для операционной системы Linux

Операции в ОС обязательно выполняем под пользователем суперюзер (напр. Administrator)

Последовательность установки программы для операционной системы Linux (с использованием интерфейса командной строки):

Порядок установки на примере системы **Linux CentOS 7**

2.1.1 Установка платформы OpenJRE

1. Перейти в директорию home:
`cd ~`
2. Скачать RPM-пакет Java OpenJDK 11:
`wget "http://link_copied_from_site"`
3. Установить RPM-пакет Java 11:
`sudo dnf install java-11-openjdk-devel`
4. Удалить RPM-пакет Java 11:
`rm ~/java-11-openjdk-devel`
5. Выбрать Java 11 по умолчанию:
`sudo alternatives --config java`

2.1.2 Установка СУБД PostgreSQL 12

1. Указать расположение репозитория:
`yum install`
`https://download.postgresql.org/pub/repos/yum/12/redhat/rhel-7-`

x86_64/filename.noarch.rpm

2. Инсталляция СУБД:

```
sudo yum install postgresql-server postgresql-contrib
```

3. Создание экземпляра базы:

```
sudo postgresql-setup initdb
```

4. Изменение аутентификации по паролю:

```
sudo vi /var/lib/pgsql/12/data/pg_hba.conf
```

В блоке:

```
host    all    all    127.0.0.1/32    ident
host    all    all    ::1/128         ident
```

Заменить данные на:

```
host    all    all    127.0.0.1/32    md5
host    all    all    ::1/128         md5
```

5. Установка пароля для пользователя postgres:

```
sudo -u postgres psql postgres
```

Ввести команду:

```
\password postgres
```

Ввести пароль

```
$_q
```

6. Создать базу приложения:

```
CREATE DATABASE saumi3;
```

Выйти из psql:

```
\q
```

2.1.3 Установка сервера приложений Tomcat

Для обозначения каталога установки любой копии Tomcat будет использоваться <ТМС>

1. Создать каталог
`sudo mkdir /opt/_Tomcat`
2. Создаём пользователя **tomcat**
`sudo useradd -M -s /bin/nologin -g tomcat -d /opt/_Tomcat tomcat`
3. Скачать Tomcat 9 с сайта проекта и распаковывать его в каталог **/opt/tomcat:**

```
## Скачивание
```

```
# с официального сайта
```

```
sudo wget http://apache.mirrors.ionfish.org/tomcat/tomcat-9/v9.0.36/bin/apache-tomcat-9.0.36.tar.gz
```

```
# в названиях каталогов приложений указываем номера сетевых портов, по которым будет работать соответствующее приложение
```

```
# это поможет при отладке и в дальнейшей работе с сервером
```

```
sudo mkdir /opt/_Tomcat/application1-8080
```

```
tar xvf apache-tomcat-9.0.*tar.gz -C /opt/_Tomcat/application1-8080 --strip-components=1
```

4. Установить права доступа:
`chown -R tomcat:tomcat /opt/_Tomcat`
`find /opt/_Tomcat/* -type d -exec chmod 770 {} \;`
`find /opt/_Tomcat/* -type f -name *.sh -exec chmod 770 {} \;`
5. Создать каталоги для журналов
`sudo mkdir -p /var/log/tomcat/application1-8080/archiv`
`sudo rmdir /opt/_Tomcat/application1-8080/logs`
`sudo ln -s /var/log/tomcat /logs`
`sudo ln -s /var/log/tomcat/application1-8080 /opt/_Tomcat/application1-8080/logs`
`sudo chmod -R 770 /var/log/tomcat`
`sudo chown -R tomcat:tomcat /var/log/tomcat`
`sudo restorecon -Rv /var/log/tomcat`

6. Создать скрипт для архивации журналов (архивирует все журналы за определённую дату и сохраняет архивы в каталоге <ТМС>/logs/archiv в течение недели):

```
sudo mkdir -p /root/_Scripts/  
sudo touch /root/_Scripts/arch_log_date-tomcat.sh
```

и привести его к виду:

```
#!/bin/bash  
  
mkdir -p $1/archiv  
  
find $1/*$(date --date '-1 day' +%Y-%m-%d)*.* -exec tar -r -f  
$1/archiv/$(date --date '-1 day' +%Y%m%d).tar --remove-files { } \  
&& find $1/archiv/*.tar -exec gzip { } \; && find $1/archiv/*.  
  
tar.gz -mtime +5 -delete  
  
chown $2:$3 $1/archiv/*.tar.gz  
chmod 640 $1/archiv/*.tar.gz
```

7. Для logrotate создать файл-конфигурацию ротации основного файла журнала Tomcat-a (при активной работе с приложениями в Tomcat он увеличивается очень быстро) /etc/logrotate.d/tomcat следующего содержания:

```
/var/log/tomcat/application1-8080/catalina.out  
{  
    rotate 10  
    size 200M  
    compress  
    notifempty  
    missingok  
    copytruncate  
    su tomcat tomcat  
}
```

8. В файл конфигурации "планировщика" /etc/crontab добавить строки:

```
0 */12 * * * root logrotate --force /etc/logrotate.d/tomcat > /dev/null
2>&1
```

```
0 1 * * * root /root/_Scripts/arch_log_date-tomcat.sh
/var/log/tomcat/application1-8080 tomcat tomcat > /dev/null 2>&1
```

По умолчанию в конфигурационных файлах **Tomcat** (Tomcat/opt/_Tomcat/application1-8080/conf/server.xml) в качестве основного порта доступа указан порт **8080**.

9. При необходимости настройки работы по HTTPS необходимо добавить дополнительные настройки:

```
<Connector port="8443" SSLEnabled="true"
maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25"
    maxSpareThreads="200"
    enableLookups="false" disableUploadTimeout="true"
    acceptCount="100" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="/opt/_Tomcat/application1-
8080/conf/ssl/keystore.p12" keystorePass="*****"
    keystoreType="PKCS12"
    keyAlias="application1"/>
```

И создать сами ключи:

```
sudo mkdir -p /opt/_Tomcat/application1-8080/conf/ssl
sudo keytool -genkeypair -keyalg RSA -keysize 2048 -storetype
PKCS12 -keystore keystore.p12 -validity 3650 -alias application1 -file
/opt/_Tomcat/application1-8080/conf/ssl/keystore.p12
```

10. В интерактивном режиме указать требуемые параметры и получить по окончании файл-хранилище в ключами (закрытым (секретным) и публичным). В файл <TMC>/conf/context.xml перед закрывающим тегом </Context> добавить строку:

...

```
<Resources cachingAllowed="true" cacheMaxSize="100000"
cacheTtl="2000" />

</Context>
```

11. Создать файлы systemd-юнитов для запуска Tomcat-ов в качестве сервиса. Для Application1 /etc/systemd/system/tomcat-application1-8080.service:

```
# Systemd unit file for Tomcat - Application1

[Unit]

Description=Apache Tomcat Web Application Container

After=syslog.target network.target

[Service]

Type=forking

###--- Каталог установки JDK

### OpenJDK - JRE

# Environment='JAVA_HOME=/usr/lib/jvm/jre'

Environment='JAVA_HOME=/usr/lib/jvm/java-1.8.0'

### OpenJDK - JRE

# Environment='JAVA_HOME=/usr/java/latest/jre'

Environment='CATALINA_PID=/opt/_Tomcat/application1-
8080/temp/tomcat.pid'

Environment='CATALINA_HOME=/opt/_Tomcat/application1-8080'

Environment='CATALINA_BASE=/opt/_Tomcat/application1-8080'

Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -
XX:+UseParallelGC'

Environment='JAVA_OPTS=-Djava.awt.headless=true -
Djava.security.egd=file:/dev/./urandom'
```



```
WorkingDirectory=/opt/_Tomcat/application1-8080
ExecStart=/opt/_Tomcat/application1-8080/bin/startup.sh
ExecStop=/bin/kill -15 $MAINPID
```

```
User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always
```

```
[Install]
```

```
WantedBy=multi-user.target
```

12. Указать systemd, чтобы он прочитал новые юниты:

```
sudo systemctl daemon-reload
```

13. Запустить, проверить, поставить в автозапуск:

```
sudo systemctl enable tomcat-application1-8080.service
```

```
sudo systemctl start tomcat-application1-8080.service
```

```
sudo systemctl status tomcat-application1-8080.service
```

14. Если запуск Tomcat прошёл успешно, можно разворачивать приложение. При необходимости передавать некоторые параметры в приложение необходимо их прописывать в файл <ТМС>/conf/catalina.properties.

2.1.4 Установка Elasticsearch

1. Скачать elasticsearch 5.6.4:

```
Wget "http://link_copied_from_site"
```

2. Установить:

```
sudo rpm -ivh elasticsearch-1.7.3.noarch.rpm
```

3. Запустить как сервис:
`sudo systemctl enable elasticsearch.service`
4. Конфигурировать elasticsearch:
`sudo nano /etc/elasticsearch/elasticsearch.yml`
5. Прописать свои имена, ноды и кластера:
`node.name: "My First Node"`
`cluster.name: mycluster1`
6. В случае необходимости изменить дефолтную паку для данных:
`path.data: /media/different_media`
7. Установить дополнительные параметры:
`network.bind_host: 0`
`network.publish_host: 0.0.0.0`
`network.host: 0`
`http.compression: true`
`http.max_content_length: 1024mb`
8. Запустить сервис:
`sudo service elasticsearch restart`
9. Для проверки работы выполнить:
`curl -X GET 'http://localhost:9200'`
10. Ответ:

```
{  
  "status" : 200,  
  "name" : "CentOS Node",  
  "cluster_name" : "mysqlcluster",
```

```
"version" : {  
  "number" : "1.7.3",  
  "build_hash" : "05d4530971ef0ea46d0f4fa6ee64dbc8df659682",  
  "build_timestamp" : "2015-10-15T09:14:17Z",  
  "build_snapshot" : false,  
  "lucene_version" : "4.10.4"  
},  
"tagline" : "You Know, for Search"  
}
```

2.1.5 Установка ZomboDB

1. Скачать с сайта <https://www.zombodb.com/releases/> плагин и экстеншн для PostgreSQL.

2. Установить экстеншн:

```
sudo rpm -ivh extension.rpm
```

3. Прописать в "main/postgresql.conf" в конец:

```
local_preload_libraries = 'zombodb.so'  
zombodb.default_elasticsearch_url = 'http://localhost:9200/'
```

4. Установить плагин:

```
sudo /usr/share/elasticsearch/bin/elasticsearch-plugin install  
file:///path/to/plugin.zip
```

5. Перезапустить оба сервиса.

2.1.6 Установка PostGIS

1. Скачать с сайта <https://postgis.net/> расширение PostGIS для СУБД PostgreSQL

2. Установить расширение:

```
sudo apt install postgis
```

2.1.7 Создание базы приложения

1. Создать базу ice_mdm и выполнить на ней:

```
CREATE EXTENSION zombodb;
```

2. Создать базу geo_saumi3 и выполнить на ней:

```
CREATE EXTENSION postgis;
```

```
CREATE EXTENSION postgis_raster;
```

```
CREATE EXTENSION postgis_sfcgal;
```

```
CREATE EXTENSION address_standardizer;
```

```
CREATE EXTENSION fuzzystrmatch;
```

```
CREATE EXTENSION postgis_topology;
```

```
CREATE EXTENSION postgis_tiger_geocoder;
```

В созданной базе данных создать таблицы для слоев с геоданными, выполняя скрипт:

```
CREATE TABLE [имя таблицы: layer1] (
```

```
    id    BIGINT PRIMARY KEY,
```

```
    cad_num VARCHAR,
```

```
    geom  geometry
```

```
);
```

3. Создать базу ice_esb

2.1.8 Установка ApacheMQ

1. Скачать arachemq:

```
wget https://archive.apache.org/dist/activemq/
```

```
5.15.12/apache-activemq-5.15.12-bin.tar.gz
```

2. Распаковать:

```
sudo mkdir -p /opt/activemq
```

```
sudo tar -zxvf apache-activemq-*-bin.tar.gz -C /opt/activemq --strip-components=1
```

3. Создать пользователя и установить права на каталог программы

```
sudo groupadd activemq
```

```
sudo useradd -M -s /bin/nologin -g activemq -d /opt/activemq/ activemq
```

```
sudo passwd activemq -l
```

```
sudo cd /opt/activemq/
```

```
sudo chown -R activemq:activemq /opt/activemq/
```

4. Создать линк:

```
sudo ln -s /opt/apache-activemq-5.15.12 /opt/activemq
```

5. Создать файл для запуска ApacheMQ как сервиса:

```
sudo vi /usr/lib/systemd/system/activemq.service
```

```
[Unit]
```

```
Description=activemq message queue
After=network.target

[Service]

User=activemq
Group=activemq
WorkingDirectory=/opt/activemq/bin/
PIDFile=/opt/activemq/data/activemq.pid
ExecStart=/opt/activemq/bin/activemq start
ExecStop=/opt/activemq/bin/activemq stop

[Install]

WantedBy=multi-user.target
```

6. Сохранить. Перезагрузить настройки systemd:
`sudo systemctl daemon-reload`
7. Перевести в автозапуск и запустить как сервис:
`sudo systemctl enable activemq.service --now`

2.1.9 Установка сервера геоданных GeoServer

1. Скачать с сайта <http://geoserver.org/release/stable> GeoServer версии 2.16.0 (не ниже)
2. Распаковать архив с GeoServer в папку на сервере приложений
3. Настроить GeoServer:
В файле %GEOSERVER_HOME\start.ini установить значение параметра
`jetty.port=8585`

В файле %GEOSEVER_HOME\webapps\geoserver\WEB-INF\web.xml раскомментировать следующие фильтры:

```
<filter>
    <filter-name>cross-origin</filter-name>
    <filter-class>org.eclipse.jetty.servlets.CrossOriginFilter</filter-
class>
    ...
</filter>
```

```
<filter-mapping>
    <filter-name>cross-origin</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

4. Запустить GeoServer из файла:

```
%GEOSEVER_HOME\bin\startup.sh
```

5. Авторизоваться

Пройти в браузере по ссылке:

```
http://{имя_хоста}:8585/geoserver
```

Логин: admin

Пароль: geoserver

6. Создать рабочую область:

Данные: Рабочие области -> Добавить новую рабочую область

Name: drr

URI: com.bftcom.geoserver.drr

По умолчанию: true

7. Настроить хранилище:

Создать новое хранилище (Данные: Хранилища -> Добавить новое хранилище -> PostGIS)

Указать рабочую область и параметры базы данных, ввести название хранилища, нажать "Сохранить"

8. Настроить слои:

Создать слой (Данные: Слои -> Добавить новый слой). Указать хранилище в качестве источника

Напротив нужной таблицы нажать кнопку опубликовать. Данная таблица будет источником данных для слоя

В разделе "Охваты" нажать кнопку "Вычислить по данным" и "вычислить из родного охвата"

Нажать "Сохранить"

2.1.10 Настройка Tomcat

1. Для всех компонентов в указанный файл **<ТМС>/conf/catalina.properties** надо добавить параметр:

```
ice.projectRoot=<ТМС>
```

где **<ТМС>** - домашний каталог конкретного экземпляра установленной копии Tomcat.

2. И создать соответствующий каталог:

```
sudo mkdir -p <ТМС>/ice
```

```
sudo chown tomcat:tomcat <ТМС>/ice
```

```
sudo chmod 750 <ТМС>/ice
```

3. Так же создать файл **/spring-shell.log** (его необходимость зависит от параметров, указанных в **<ТМС>/conf/catalina.properties**):

```
sudo touch /spring-shell.log
```

```
sudo chown tomcat:tomcat /spring-shell.log
```

```
sudo chmod 750 /spring-shell.log
```

4. Открыть доступ к порту:

i. на межсетевом экране firewalld:

```
# открываем доступ к контейнерам и интерфейсу управления для
указанной сети
```

```
# добавляем требуемые порты 8080
```

```
sudo firewall-cmd --permanent --new-service=tomcat-http
```

```
sudo firewall-cmd --permanent --service=tomcat-http --add-
port=8080/tcp
```

```
sudo firewall-cmd --permanent --add-service=tomcat-http
```

```
sudo firewall-cmd --reload
```

```
### Далее - примеры (!) для настройки более "тонкой" настройки
доступа:
```

```
sudo firewall-cmd --permanent --add-rich-rule 'rule family="ipv4"
source address="192.168.1.0/24" service name="http" accept'
```

```
# открываем полный доступ с сервера мониторинга Zabbix
```

```
sudo firewall-cmd --permanent --add-rich-rule 'rule family="ipv4"
source address="192.168.19.34/32" accept'
```

ii. в **SELinux**:

5. Проверить, что SELinux активен:

```
sudo sestatus
```

6. Если в ответе на эту команду в строке "SELinux status:" будет enabled - проверяем целевые порты и добавляем по необходимости:

```
# Посмотреть, открыты ли порты наших серверов Tomcat
```

```
sudo semanage port -l | grep -E "8080"
```

7. Так же проверяем, если ли блокировки со стороны SELinux:

```
sudo cat /var/log/audit/audit.log | grep denied | grep catalina.sh
```

8. Если в выводе будут строки, то создаём и применяем на основе их разрешающее правило:

```
sudo cat /var/log/audit/audit.log | grep denied | grep catalina.sh |
audit2allow -M my_tomcat
```

```
sudo semodule -i my_tomcat.pp
```

9. При случае, когда утилита `semanage` не установлена на сервере - установить её (и повторить затем предыдущие команды):

```
sudo yum install polycoreutils-python
```

8. Положить файл `application.properties` в папку с Tomcat.

```
ice.servicesPackages=com.bftcom.mdm.common,  
com.bftcom.ice.common.service
```

```
spring.profiles.active=postgresql,postgresql95
```

```
spring.datasource.url=jdbc:postgresql://localhost:5432/ice_mdm
```

```
spring.datasource.username=postgres
```

```
spring.datasource.password=postgres
```

```
#spring.devtools.restart.trigger-file = application.properties
```

```
spring.devtools.restart.enabled = false
```

```
spring.devtools.livereload.enabled = false
```

```
server.servlet.contextPath=/app
```

```
# Charset of HTTP requests and responses. Added to the "Content-  
Type" header if not set explicitly.
```

```
spring.http.encoding.charset=UTF-8
```

```
# Enable http encoding support.
```

```
spring.http.encoding.enabled=true
```

```
# Force the encoding to the configured charset on HTTP requests and  
responses.
```

```
spring.http.encoding.force=true
```

spring.jackson.serialization.write-dates-as-timestamps=true

spring.jackson.default-property-inclusion = non_null

spring.servlet.multipart.maxFileSize=100MB

spring.servlet.multipart.maxRequestSize=100MB

spring.cache.jcache.provider=org.ehcache.jsr107.EhcacheCachingProvider

spring.cache.jcache.config=classpath:ehcache.xml

security.jwt.secret=random_secret_key

security.jwt.expirationTime=604800000

spring.quartz.job-store-type=jdbc

spring.quartz.jdbc.initialize-schema=never

spring.quartz.properties.org.quartz.jobStore.tablePrefix=quartz.qrtz_

spring.quartz.properties.org.quartz.jobStore.driverDelegateClass =
org.quartz.impl.jdbcjobstore.PostgreSQLDelegate

mdm.jms.url = tcp://localhost:61616

mdm.jms.username = admin

mdm.jms.password = admin

mdm.jms.outbound.queue.name = mdmOutgoingQueue

mdm.jms.inbound.queue.name = mdmIncomingQueue

mdm.elasticsearch.url = http://localhost:9200

10. Прописать полный путь до этого файла в bin/startup.sh:

```
export CATALINA_OPTS="-  
Dspring.config.location=file:полный_путь_до_application.properties"
```

2.2 Настройка программы на условия конкретного применения

2.2.1 Настройка программы на условия конкретного применения

Настройка программы на условиях конкретного применения для комплексов:

- Комплекс ведения основных данных (КВОД);
- Комплекс технологических функций (КТФ);
- Комплекс управления имуществом (КУИ);
- Комплекс управления эксплуатацией (КУЭ);
- Комплекс планирования и контроля мероприятий, направленных на повышение энергоэффективности и экологической безопасности (КЭЭЭБ);
- Комплекс анализа данных (КАД).

Требуется выполнить следующие настройки для начала работы с компонентом:

- установить war-архив приложения на сервер приложения Apache Tomcat.
- настроить файл application.properties:
 - настройки подключения к базе;
 - настройки подключения к очереди jsm.
- установить linux-сервис elasticsearch для реализации полнотекстового поиска.
- установить расширение zombodb для базы данных PostgreSQL.

Инициализация базы приложения происходит автоматически при первом старте.

2.2.2 Настройка программы на условия конкретного применения для комплекса интеграции

Требуется выполнить следующие настройки для начала работы с компонентом:

- установить war-архив приложения на сервер приложения Apache Tomcat.
- настроить файл *catalina.properties*:
 - Настройки подключения к базе
- настроить файл *application.properties*:
 - настройки подключения к базе;
 - настройки подключения к очереди jsm.
- установить linux-сервис elasticsearch для реализации полнотекстового поиска.
- установить расширение zombodb для базы данных PostgreSQL.
- установить расширение PostGis для базы данных PostgreSQL
- установить сервер геоданных GeoServer
- настроить хранилище и слои для GeoServer
- установить брокер ActiveMQ
- установить Адаптер СМЭВ версии 3.1.x

Инициализация базы приложения происходит автоматически при первом старте.

Для настройки программы на условия конкретного применения выполняются следующие действия:

- Настроить файл *catalina.properties*:

Параметры подключения Shine

spring.profiles.active=postgresql

spring.datasource.url=jdbc:postgresql://ServerName:5432/db_name

```
spring.datasource.username=*****  
spring.datasource.password=*****  
spring.shell.interactive.enabled=false  
ice.projectRoot=/opt/_Tomcat/shine-8080  
backlog.jms.broker-url = tcp://127.0.0.1:61616  
shine.gateway.addr=http://127.0.0.1:8081/gate
```

- настроить файл *logback.xml*:
 - настройка необходимости журналирования обмена:
logger name="Camellog" level="OFF" или "INFO"
 - настройки электронной подписи
 - настройка через пользовательский интерфейс используемых слоев сервера геоданных и привязка слоев к объектам приложения
 - настроить Адаптер СМЭВ версии 3.1.x
 - настроить разделе Информационные системы подключение к модулю Подсистема информационного взаимодействия. Указать способ использования адаптера - JMS
 - настроить подпись (ЭП-ОВ)
- настроить модуль интеграции через JMS: указать адрес брокера ActiveMQ.

3 Запуск Системы хранения и предоставления электронных документов («БФТ.Управление активами»)

Работа в платформе доступна только для зарегистрированных пользователей.

Для перехода к окну авторизации в строке адреса браузера вводится адрес сервера системы. Откроется окно авторизации пользователя:

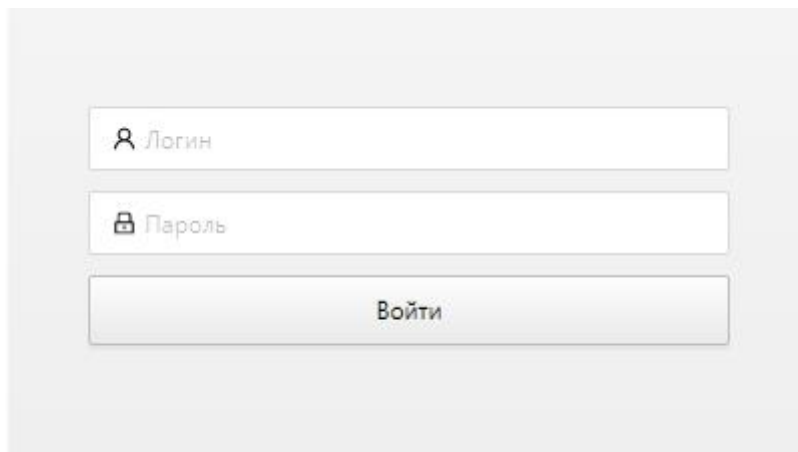


Рисунок 1 – Окно входа в систему

Для входа в систему необходимо указать следующую информацию (Рисунок 1):

- **Логин** - имя пользователя (Administrator).
- **Пароль** - пароль пользователя (Administrator).

Далее нажимается кнопка **Войти**.

В случае ввода верной информации (указаны зарегистрированный в системе пользователь и правильный пароль) осуществится вход в главное окно системы. При вводе неверного **Пользователя** или **Пароля** на экране появится сообщение об ошибке «**Неправильное имя пользователя или пароль**».