

Инструкция по разворачиванию ПО ЭЛАР-Контекст на ОС ALT-Server 10.1

Оглавление

1	ПРЕДНАЗНАЧЕНИЕ ДОКУМЕНТА.....	2
2	КРАТКОЕ ОПИСАНИЕ ОКРУЖЕНИЯ И СХЕМЫ РАЗВЕРТЫВАНИЯ	3
3	Развертывание PostgreSQL 14.8	4
3.1	Установка и настройка PostgreSQL	4
3.2	Тестирование работоспособности PostgreSQL.....	6
4	Развертывание БД для приложения.....	7
4.1	Создание и настройка БД ElarContext:	7
4.2	Создание и настройка БД ELARContextAudit:	7
5	Установка Elasticsearch 7.10 (одна нода).....	8
6	Установка OpenJDK 17	11
7	Развертывание Tomcat 9	12
7.1	Установка пакета TomCat.....	12
7.2	Создание демона Tomcat в Systemd.....	12
8	Развертывание LibreOffice.....	16
8.1	Установка пакета из репозитория.....	16
8.2	Создание демона LibreOffice в Systemd	16
9	Установка приложения ElarContext в Tomcat.....	17
9.1	Конфигурирование приложения.....	17
9.2	Установка и запуск приложения.....	20

1 ПРЕДНАЗНАЧЕНИЕ ДОКУМЕНТА

Настоящий документ предназначен для разворачивания системы на тестовом стенде в целях удостоверения совместимости Программного обеспечения ЭЛАР Контекст версии 3.4 с операционной системой ALT Server 10.1

Полное руководство администратора предоставляется в комплекте с рабочей документацией по Системе при поставке Программного обеспечения Заказчику.

2 КРАТКОЕ ОПИСАНИЕ ОКРУЖЕНИЯ И СХЕМЫ РАЗВЕРТЫВАНИЯ

Сервер ЭК на ОС Alt Linux 10 (без графич. оболочки)

IP адрес: 10.4.12.126

Папка размещения БД: /opt/postgresDB/ElarContext_data

Папка установки ElasticSearch: /usr/share/elasticsearch

Папка размещения БД ElasticSearch: /opt/ElasticsearchDB

Папка установки OpenJDK: /usr/lib/jvm/java-17-openjdk-17.0.7.0.7-alt1.x86_64

Папка установки ElarContext: /opt/ElarContext

Папка установки LibreOffice: /usr/lib64/LibreOffice

Папка размещения хранилища данных ЭК: -

Папка размещения репозитория ЭК: /opt/ElarContextDB/ElarContext_repository

3 РАЗВЕРТЫВАНИЕ POSTGRESQL 14.8

3.1 Установка и настройка PostgreSQL

Обновляем списки программных пакетов в ОС:

```
apt-get update
```

Скачиваем и устанавливаем PostgreSQL:

```
apt-get install postgresql14-server
```

Перед запуском службы необходимо создать системные базы данных:

```
/etc/init.d/postgresql initdb --encoding UTF-8 --lc-collate='ru_RU.UTF-8' --lc-ctype='ru_RU.UTF-8'  
-D /var/lib/pgsql/data/
```

Запуск службы

```
systemctl start postgresql  
systemctl status postgresql
```

При необходимости, установить nano-редактор:

```
apt-get install nano
```

Включить авторизацию и создать пользователей, так как по умолчанию подключение локально разрешено для любого авторизованного в системе пользователя.

Для этого переключаемся в учётную запись postgres:

```
psql -U postgres
```

У пользователя *postgres* права *root* в СУБД. Необходимо создать пароль для пользователя:

```
ALTER ROLE postgres WITH PASSWORD 'your_password';  
(пример) ALTER ROLE postgres WITH PASSWORD 'Qwerty11!@';
```

Вернуться в консоль и в *root*, нажав сочетание клавиш «**Ctrl+d**»;

внести изменения в схему авторизации СУБД;

Для этого необходимо отредактировать файл, расположенный по адресу

```
/var/lib/pgsql/data/pg_hba.conf
```

```
nano /var/lib/pgsql/data/pg_hba.conf
```

В конце файла строки, отвечающие за разрешение заходить без авторизации с localhost:

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

заменить на строки:

```
local all all md5  
host all all 127.0.0.1/32 md5
```

При необходимости оставить подключение по IPv6.

Первая строка разрешает подключение к СУБД для всех локальных пользователей, но при этом требует с них ввод пароля. Вторая – делает практически то же самое, но эта строка оставляется для

того, чтобы подключаться к СУБД могли даже те пользователи, у кого нет системной учетной записи, а есть только роль в СУБД.

Ещё один момент. Если нужно удаленное подключение, то необходимо действовать по аналогии и добавить строку:

```
host all all 123.456.678.910/24 md5
```

Конечный вариант последних строк документа:

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 md5 <-- вместо ident для доступа из localhost
host all all 0.0.0.0/0 md5 <-- адрес для удаленного доступа из сети с любой
машины
# IPv6 local connections:
host all all ::1/128 ident
# Allow replication connections from localhost, by a user with the
# replication privilege.
local all all md5
host all all 127.0.0.1/32 md5
```

указать СУБД, какие интерфейсы прослушивать;

```
nano /var/lib/pgsql/data/postgresql.conf
```

В файле *postgresql.conf* все изменяемые далее строки должны быть раскомментированы, никаких «#»! В т.ч. нужно снять комментарий «#» с порта:

```
port = 5432
вместо: #port = 5432
```

Для этого в файле конфигурации */var/lib/pgsql/data/postgresql.conf* найти строку:

```
#listen_addresses = 'localhost'
```

и исправить ее на:

```
listen_addresses = '*'
```

Это заставит СУБД слушать все сетевые интерфейсы в поисках входящих подключений:

```
listen_addresses = '*'
...
log_timezone = 'Europe/Moscow'
...
datestyle = 'iso, dmy'
...
timezone = 'Europe/Moscow'
...
lc_messages = 'ru_RU.UTF-8'
lc_monetary = 'ru_RU.UTF-8'
lc_numeric = 'ru_RU.UTF-8'
lc_time = 'ru_RU.UTF-8'
```

```
...  
default_text_search_config = 'pg_catalog.russian'
```

Также в файле конфигурации `/var/lib/pgsql/data/postgresql.conf` изменить настройки:

```
shared_buffers = 512MB  
temp_buffers = 16MB  
work_mem = 128MB  
maintenance_work_mem = 192MB  
effective_cache_size = 8GB
```

Примечание – Подбор данных настроек производится индивидуально и зависит как от характеристик сервера, на котором будет работать сервер PostgreSQL, так и от характеристик информационных баз, которые будут обрабатываться на этом сервере. Перезапустить службу:

```
systemctl restart postgresql
```

Примечание – Если необходимо зайти под другим пользователем, следует также указывать базу данных, к которой надо подключиться, иначе PostgreSQL попытается подключиться к БД с тем же названием, что и имя пользователя, не найдет ее и выдаст сообщение об ошибке.

3.2 Тестирование работоспособности PostgreSQL

Для тестирования PostgreSQL необходимо:

Посмотреть наличие ошибок в логах `/var/lib/pgsql/data/log/`:

```
cd /var/lib/pgsql/data/log  
ls -l
```

(пример): `cat postgresql-2023-07-13_123906.log`

(в зависимости от времени название файлов изменяется)

проверить статус сервера

```
/etc/init.d/postgresql status
```

```
«postgres is running»
```

проверить использование порта 5432

проверить статус службы:

```
systemctl status postgresql
```

4 РАЗВЕРТЫВАНИЕ БД ДЛЯ ПРИЛОЖЕНИЯ

4.1 Создание и настройка БД ElarContext:

Создаем папку /opt/postgresDB/ElarContext_data и изменяем владельца папки на postgres:postgres и права для отработки последующего скрипта:

```
mkdir /opt/postgresDB
mkdir /opt/postgresDB/ElarContext_data
chown postgres:postgres /opt/postgresDB/ElarContext_data
chmod 775 /opt/postgresDB/ElarContext_data
```

Заходим в pgAdmin на рабочем удаленном компьютере и пытаемся приконnectиться по ip-адресу сервера, где установлена БД.

Создание табличного пространства для размещения БД на другом диске:

```
CREATE TABLESPACE ElarContext_data LOCATION '/opt/postgresDB/ElarContext_data';
```

Создание БД ELARContext:

Если будут проблемы с локалями то добавить к скрипту создания БД в конце «TEMPLATE template0;»

```
-- DROP DATABASE "ELARContext";
CREATE DATABASE "ELARContext"
WITH
OWNER = postgres
ENCODING = 'UTF8'
LC_COLLATE = 'ru_RU.UTF-8'
LC_CTYPE = 'ru_RU.UTF-8'
TABLESPACE = ElarContext_data
CONNECTION LIMIT = -1;
```

Установить все привилегии для пользователя postgres по отношению к БД ELARContext:

```
GRANT ALL ON DATABASE "ELARContext" TO postgres;
GRANT TEMPORARY, CONNECT ON DATABASE "ELARContext" TO PUBLIC;
```

4.2 Создание и настройка БД ELARContextAudit:

```
-- DROP DATABASE "ELARContextAudit";
CREATE DATABASE "ELARContextAudit"
WITH OWNER = postgres
ENCODING = 'UTF8'
LC_COLLATE = 'ru_RU.UTF-8'
LC_CTYPE = 'ru_RU.UTF-8'
TABLESPACE = ElarContext_data
CONNECTION LIMIT = -1;
GRANT ALL ON DATABASE "ELARContextAudit" TO postgres;
GRANT TEMPORARY, CONNECT ON DATABASE "ELARContextAudit" TO PUBLIC;
```

Сброс сессий БД:

```
SELECT pg_terminate_backend(pid) FROM pg_stat_activity WHERE pid <> pg_backend_pid()
AND datname = 'ELARContext';
```

5 УСТАНОВКА ELASTICSEARCH 7.10 (ОДНА НОДА)

Помещаем в папку «/opt» (cd /opt) rpm-файл elasticsearch-7.10.0-x86_64.rpm и устанавливаем:

```
rpm -ivh elasticsearch-7.10.0-x86_64.rpm
```

ElasticSearch установлен в папке /usr/share/elasticsearch,

конфигурация в /etc/elasticsearch

Установка переменных среды

```
echo 'ES_HOME="/usr/share/elasticsearch"' >> /etc/environment  
echo 'ES_PATH_CONF="/etc/elasticsearch"' >> /etc/environment
```

Установка плагина analysis-phonetic (если его положить по пути «/opt/analysis-phonetic-7.10.0.zip»):

```
/usr/share/elasticsearch/bin/elasticsearch-plugin install file:///opt/analysis-phonetic-7.10.0.zip
```

Бэкапим старый конфиг

```
cp /etc/elasticsearch/elasticsearch.yml /etc/elasticsearch/elasticsearch.yml.bak
```

Копируем новый

```
nano /etc/elasticsearch/elasticsearch.yml
```

В начале файла закомментировать параметры:

Имеется:

```
path.data: /var/lib/elasticsearch  
path.logs: /var/log/elasticsearch
```

Должно быть:

```
#path.data: /var/lib/elasticsearch  
#path.logs: /var/log/elasticsearch
```

Добавить в конец файла (здесь также изменяется путь к БД и логам в конфиге)

```
cluster.name: elar_elasticsearch  
node.name: node1  
node.roles: [data, master]  
path.data: /opt/ElasticsearchDB/data  
path.logs: /opt/ElasticsearchDB/logs  
bootstrap.memory_lock: true  
http.port: 9200  
discovery.seed_hosts: ["127.0.0.1:9300"]  
#discovery.seed_hosts: ["127.0.0.1:9300", "127.0.0.1:9301"]  
cluster.initial_master_nodes: ["node1"]  
transport.port: 9300  
indices.query.bool.max_clause_count: 10240  
#network.host: [localhost, _site_]  
network.host: 0.0.0.0
```

Создание папок для БД и логов Elasticsearch

```
mkdir /opt/ElasticsearchDB  
mkdir /opt/ElasticsearchDB/data
```



```
mkdir /opt/ElasticsearchDB/logs
chown -R elasticsearch:elasticsearch /opt/ElasticsearchDB/
```

Редактируем ограничения по работе с памятью

```
nano /etc/security/limits.conf
```

Установить следующие значения в конец файла

```
elasticsearch soft memlock unlimited
elasticsearch hard memlock unlimited
nano /etc/default/elasticsearch
```

Установить следующие значения

```
ES_JAVA_OPTS="-Xms4g -Xmx4g"
MAX_LOCKED_MEMORY=unlimited
nano /usr/lib/systemd/system/elasticsearch.service
```

Установить следующие значения в конец файла

```
[Service]
LimitMEMLOCK=infinity

nano /etc/elasticsearch/jvm.options
```

Установить следующие значения

```
-Xms4g
-Xmx4g
```

В папку конфигурации /etc/elasticsearch копируем:

папку analysis (предварительно распаковав архив «rus_synonyms.7z»)

папку hunspell

Если есть проблемы с правами доступа:

```
chmod 775 /etc/elasticsearch
```

Меняем права для скопированных папок:

```
chown -R root:root /etc/elasticsearch/analysis
chown -R root:root /etc/elasticsearch/hunspell
chmod -R 775 /etc/elasticsearch/hunspell
chmod -R 775 /etc/elasticsearch/analysis
```

Запускаем Elasticsearch

```
systemctl daemon-reload
systemctl enable elasticsearch.service
systemctl start elasticsearch.service
systemctl status elasticsearch.service
```

Если не установлен curl:

```
apt-get install curl
```

Настройка Elasticsearch

```
curl -X DELETE "localhost:9200/_all"  
curl -XPUT -u elastic "localhost:9200/_template/my_global_template" -H "Content-Type:  
application/json" -d " { \"template\": \"*\", \"settings\": { \"max_result_window\": 1000000 } }"
```

Проверка в браузере с рабочего ПК работы elastic:

```
http://10.4.12.126:9200/_cluster/health
```

Предварительно можно с помощью telnet пропинговать доступность портов со сторонних компьютеров (пример для linux, при включенном telnet на win10 также подойдет):

```
telnet 10.4.12.126 9200
```

6 УСТАНОВКА OPENJDK 17

Производим установку пакета из репозитория

```
apt-get install java-17-openjdk
```

Установка переменных окружения

```
echo 'PATH="/usr/lib/jvm/java-17-openjdk-17.0.7.0.7-alt1.x86_64/bin"' >> /etc/profile  
echo 'JAVA_HOME="/usr/lib/jvm/java-17-openjdk-17.0.7.0.7-alt1.x86_64"' >> /etc/environment
```

Подтягивание ОС новых конфиг-файлов

```
source /etc/profile  
source /etc/environment
```

Проверка установки ссылок

```
java -version
```

7 РАЗВЕРТЫВАНИЕ ТОМСАТ 9

7.1 Установка пакета TomCat

Перед установкой внимательно перепроверяем все пути и версии в конфигурационных файлах и командах!!!

Помещаем архив с TomCat в любую папку (например, в /opt) и переходим туда:

```
cd /opt
tar -xvf ./apache-tomcat-9.0.76.tar.gz
mv /opt/apache-tomcat-9.0.76 /opt/ElarContext/
cd /opt/ElarContext
groupadd tomcat
useradd -s /bin/false -g tomcat -d /opt/ElarContext tomcat
chown -R tomcat:tomcat /opt/ElarContext
chmod -R 775 /opt/ElarContext
echo 'CATALINA_HOME="/opt/ElarContext/"' >> /etc/environment
```

Создаем папку и даём права для tomcat для папки хранения репозитория и сканов ЭК

```
mkdir /opt/ElarContextDB/
chown -R tomcat:tomcat /opt/ElarContextDB/
```

7.2 Создание демона Tomcat в Systemd

Создание конфига демона

```
nano /etc/systemd/system/tomcat.service
```

Добавить в конфиг:

```
[Unit]
Description=Apache Tomcat Web Application Container
After=syslog.target network.target
[Service]
Type=forking
Environment=JAVA_HOME=/usr/lib/jvm/java-17-openjdk-17.0.7.0.7-alt1.x86_64
Environment=CATALINA_HOME=/opt/ElarContext
Environment=CATALINA_BASE=/opt/ElarContext
Environment=CATALINA_TMPDIR=/opt/ElarContext/temp
Environment=CATALINA_PID=/opt/ElarContext/temp/tomcat.pid
Environment='CATALINA_OPTS='
Environment='JAVA_OPTS= -Xms4G -Xmx8G -Xss1m -Xmn256m -Djava.awt.headless=true -
server -Djava.security.egd=file:/dev/./urandom'
ExecStart=/opt/ElarContext/start.sh
ExecStop=/opt/ElarContext/bin/shutdown.sh
StandardOutput=inherit
StandardError=inherit
User=root
Group=root
UMask=0007
RestartSec=10
Restart=always
[Install]
```

WantedBy=multi-user.target

Создание файла запуска демона

```
nano /opt/ElarContext/start.sh
```

Добавить в файл:

```
#!/bin/bash
```

```
WORKDIR=/opt/ElarContext
```

```
JAVA_HOME= /usr/lib/jvm/java-17-openjdk-17.0.7.0.7-alt1.x86_64
```

Для версии ЭК 3.3 и выше

```
JAVA_OPTS="$JAVA_OPTS --add-opens=java.base/java.lang=ALL-UNNAMED";
```

```
JAVA_OPTS="$JAVA_OPTS --add-opens=java.base/java.io=ALL-UNNAMED";
```

```
JAVA_OPTS="$JAVA_OPTS --add-opens=java.base/java.util=ALL-UNNAMED";
```

```
JAVA_OPTS="$JAVA_OPTS --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED";
```

```
cd $WORKDIR
```

```
${WORKDIR}/bin/startup.sh
```

Для корректной работы Apache Tomcat необходимо дополнить папку lib библиотеками, не входящими в пакет библиотек по умолчанию, а именно:

log4j-1.2.17.jar

log4j-initializer.jar

mail-1.4.7.jar

jstl-1.2.jar

postgresql-42.3.6.jar (библиотека для работы с PostgreSQL)

jaxb-api-2.4.0-b180830.0359.jar

javax.activation-api-1.2.0.jar

jasperreports-javaflow.jar

Свериться с библиотеками из папки lib для установки. Если что-то не хватает – докинуть. По аналогии с эластиком проверить доступность порта 8080. При недоступности – открыть. При необходимости можно изменить настройки логирования библиотеки log4j в файле log4j.properties (например, для уменьшения объема логов при массовой генерации данных).

Для этого при установке Apache Tomcat необходимо:

- задать путь к файлу конфигурации логов log4j.properties (рекомендуется хранить его в папке conf домашней директории Apache Tomcat);
- добавить log4j.properties в указанную директорию;
- внести необходимые настройки в файл log4j.properties (за тип выводимых сообщений в лог отвечают строки log4j.logger.*).

Пример конфигурационного файла:

```
#test
```

```
log4j.rootLogger=INFO, DEBUG, WARNING, ERROR, CONSOLE, CATALINA
```

```
# Define all the appenders
```

```
log4j.appender.CATALINA=org.apache.log4j.DailyRollingFileAppender
```

```
log4j.appender.CATALINA.File=${catalina.base}/logs/catalina.log
```

```
log4j.appender.CATALINA.Append=true
```

```
log4j.appender.CATALINA.Encoding=UTF-8
```

```

# Roll-over the log once per day
log4j.appender.CATALINA.DatePattern='yyyy-MM-dd'.log'
log4j.appender.CATALINA.layout = org.apache.log4j.PatternLayout
log4j.appender.CATALINA.layout.ConversionPattern = %d [%t] %-5p %-c- %m%n

log4j.appender.LOCALHOST=org.apache.log4j.DailyRollingFileAppender
log4j.appender.LOCALHOST.File=${catalina.base}/logs/localhost.log
log4j.appender.LOCALHOST.Append=true
log4j.appender.LOCALHOST.Encoding=UTF-8
log4j.appender.LOCALHOST.DatePattern='yyyy-MM-dd'.log'
log4j.appender.LOCALHOST.layout = org.apache.log4j.PatternLayout
log4j.appender.LOCALHOST.layout.ConversionPattern = %d [%t] %-5p %-c- %m%n

log4j.appender.MANAGER=org.apache.log4j.DailyRollingFileAppender
log4j.appender.MANAGER.File=${catalina.base}/logs/manager.log
log4j.appender.MANAGER.Append=true
log4j.appender.MANAGER.Encoding=UTF-8
log4j.appender.MANAGER.DatePattern='yyyy-MM-dd'.log'
log4j.appender.MANAGER.layout = org.apache.log4j.PatternLayout
log4j.appender.MANAGER.layout.ConversionPattern = %d [%t] %-5p %-c- %m%n

log4j.appender.HOST-MANAGER=org.apache.log4j.DailyRollingFileAppender
log4j.appender.HOST-MANAGER.File=${catalina.base}/logs/host-manager.log
log4j.appender.HOST-MANAGER.Append=true
log4j.appender.HOST-MANAGER.Encoding=UTF-8
log4j.appender.HOST-MANAGER.DatePattern='yyyy-MM-dd'.log'
log4j.appender.HOST-MANAGER.layout = org.apache.log4j.PatternLayout
log4j.appender.HOST-MANAGER.layout.ConversionPattern = %d [%t] %-5p %-c- %m%n

log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.CONSOLE.Encoding=UTF-8
log4j.appender.CONSOLE.layout = org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern = %d [%t] %-5p %-c- %m%n

# Configure which loggers log to which appenders
log4j.logger.org.apache.catalina.core.ContainerBase.[Catalina].[localhost]=ERROR
log4j.logger.org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager]=ERROR
log4j.logger.org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-
manager]=ERROR

# OpenKM
log4j.logger.org.hibernate=ERROR
log4j.logger.org.jbpm=ERROR
log4j.logger.com.openkm=ERROR, FATAL, stdout
log4j.logger.ru.elar=ERROR
log4j.logger.waffle=ERROR
#log4j.logger.org.hibernate.type=trace

```

Установить права на добавленные библиотеки и конфиг-файлы:

```
chown -R tomcat:tomcat /opt/ElarContext  
chmod -R 775 /opt/ElarContext
```

Подгружаем созданную конфигурацию демона
Старт и проверка работы Tomcat

```
systemctl enable tomcat  
systemctl start tomcat  
systemctl status tomcat
```

8 РАЗВЕРТЫВАНИЕ LIBREOFFICE

8.1 Установка пакета из репозитория

```
apt-get install LibreOffice  
apt-get install LibreOffice-langpack-ru
```

Настройка прав доступа

```
chown -R tomcat:tomcat /usr/lib64/LibreOffice  
chmod -R 775 /usr/lib64/LibreOffice
```

8.2 Создание демона LibreOffice в Systemd

Создание конфига демона

```
nano /etc/systemd/system/libreoffice.service
```

Добавить в конфиг:

```
[Unit]  
Description=Lunch soffice instance  
After=syslog.target  
  
[Service]  
Type=simple  
ExecStart=/usr/lib64/LibreOffice/program/soffice.bin --headless \  
--accept="socket,host=127.0.0.1,port=8100;urp;StarOffice.ServiceManager" \  
--pidfile=/var/run/soffice.pid --nologo --nolockcheck --backtrace  
  
RestartSec=5  
KillSignal=SIGQUIT  
NotifyAccess=all  
  
User=root  
#Group=tomcat  
  
[Install]  
WantedBy=multi-user.target
```

Включение и запуск демона:

```
systemctl daemon-reload  
systemctl enable libreoffice.service  
systemctl start libreoffice
```


9 УСТАНОВКА ПРИЛОЖЕНИЯ ELARCONTEXT В ТОМСАТ

9.1 Конфигурирование приложения

Конфиг /opt/ElarContext/elarctx.cfg

```
#####  
# release-3.4 #  
#####  
  
# ElasticSearch  
elar.search.server.elasticsearch.index_schema_management_strategy=create  
#elar.search.server.elasticsearch.index_schema_management_strategy=drop-and-create  
elar.search.server.elasticsearch.host=http://localhost:9200  
elar.search.server.elasticsearch.username=elastic  
elar.search.server.elasticsearch.password=elastic  
elar.search.server.elasticsearch.mode=real_time  
#elar.search.server.elasticsearch.data=not_full_text  
  
# Hibernate  
#hibernate.dialect=ru.elar.hibernate.MSSQLDialect  
hibernate.dialect=ru.elar.hibernate.PostgreSQLDialect  
#hibernate.dialect=ru.elar.hibernate.MySQLDialect  
#hibernate.dialect=ru.elar.hibernate.Oracle10gServerDialect  
  
#!!!DATABASE!!!  
# Установить значение create при первоначальной установке или update - при обновлении  
hibernate.hbm2ddl=create  
#hibernate.hbm2ddl=none  
hibernate.show_sql=false  
  
# Net health monitoring  
  
# ELAR Context  
use.storage.pools=true  
  
# SeaDragon (big_image)  
#seadragon.width=1000  
#seadragon.height=1000  
#seadragon.mimes=image/jpeg,image/gif,image/png,image/tiff,image/tif  
  
# Preview  
#preview.mimetype.pdf=embedded  
#preview.mimetype.jpg=thumb  
#preview.mimetype.png=zoomable  
  
# Openoffice Используется для конвертации файлов  
  
system.openoffice.server=127.0.0.1  
system.openoffice.port=8100
```

```

system.openoffice.path=
#system.openoffice.path=/opt/libreoffice7.2/
#system.openoffice.path=/usr/lib/libreoffice/program/soffice

# Imagemagick
#system.imagemagick.convert.max.tiff.pages=40
system.imagemagick.convert=/usr/bin/convert

ru.elar.check_folder_access_recursive=FALSE

# Initial configuration - Linux
#system.swftools.pdf2swf=/usr/bin/pdf2swf -f -T 9 -t -s storeallcharacters ${fileIn} -o ${fileOut}

# Initial configuration - Windows
#pdfconverter.application/vnd.ms-outlook=http://10.4.12.22:8118/msg2pdf${msg}
#system.swftools.pdf2swf=C:/MosComArhitecture/bin/pdf2swf.exe -f -T 9 -t -s storeallcharacters
${fileIn} -o ${fileOut}
#system.utils.path=C:\\ELARContext\\bin

#onlyoffice.callback.host=http://10.4.8.28:8282

# Настройка репозитория
# Windows
#repository.home=C:\\ELARCONTEXT_Repository
# Linux
repository.home=/opt/ElarContextDB/ElarContext_repository
repository.depth=2
repository.depth.prev=4

# Batch file operations
#system.batch.files.sourcePath=C:/ELARContext/temp
#system.batch.files.incomeFolder=/okm:root/Income
#system.batch.files.systemUserName=ak

thumb.width=800
thumb.height=800

# URL ElarContext
application.url=http://10.4.12.126:8080/index.jsp

# Brend view elements
brend.favicon=/ELARContext/img/favicon/favicon.ico
brend.title=ЭЛАР Контекст
brend.logo=img/logo_login.png

# Privacy policy
pp.save.username=false
pp.save.password=false

```

Выключение/включение аудита
events.audit.enable=true

Открытие файла в новом окне браузера
open.file.in.new_window=false

Индикатор новой задачи
bpm.notifications.enabled=true

Доработка CNTX-4193
bpm.files.separate=true

Доработка CNTX-4216
bpm.notifications.enabled=true
bpm.notifications.rate=120

TaskFilter AutoSearch counts
ru.elar.task_filter_auto_search_counts_enabled=true
ru.elar.task_filter_auto_search_counts_auto_update_enabled=true
ru.elar.task_filter_auto_search_counts_auto_update_rate=10000

Установка часового пояса
ru.elar.gwt_time_zone=+0300
ru.elar.date_time_control_format=yyyy-MM-dd'T'HH:mm:ss

Конфиг /opt/ElarContext/conf/context.xml

```
<Context>
  <WatchedResource>WEB-INF/web.xml</WatchedResource>
  <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
  <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
  <ResourceLink global="mail/ELARContext" name="mail/ELARContext"
type="javax.mail.Session" />
  <ResourceLink global="jdbc/ELARContextDS" name="jdbc/ELARContextDS"
type="javax.sql.DataSource" />
  <ResourceLink global="jdbc/ELARContextAuditDS" name="jdbc/ELARContextAuditDS"
type="javax.sql.DataSource" />
</Context>
```

Конфиг /opt/ElarContext/conf/server.xml

```
<GlobalNamingResources>
<!-- Editable user database that can also be used by
UserDatabaseRealm to authenticate users
-->
  <Resource name="UserDatabase" auth="Container"
type="org.apache.catalina.UserDatabase"
description="User database that can be updated and saved"
factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
```

```

    pathname="conf/tomcat-users.xml" />
<!--Resource auth="Container"
    mail.from="testing@elar.ru"
    mail.smtp.host="localhost"
    name="mail/OpenKM"
    type="javax.mail.Session"/-->
<Resource auth="Container"
    driverClassName="org.postgresql.Driver"
    type="javax.sql.DataSource"
    maxTotal="100"
    maxIdle="30"
    maxWaitMillis="10000"
    name="jdbc/ELARContextAuditDS"
    username="postgres"
    password="Qwerty11!@"
    url="jdbc:postgresql://localhost/ELARContextAudit"
    validationQuery="select 1"/>
<Resource auth="Container"
    driverClassName="org.postgresql.Driver"
    type="javax.sql.DataSource"
    maxTotal="100"
    maxIdle="30"
    maxWaitMillis="10000"
    name="jdbc/ELARContextDS"
    username="postgres"
    password="Qwerty11!@"
    url="jdbc:postgresql://localhost/ELARContext"
    validationQuery="select 1"/>
</GlobalNamingResources>

```

9.2 Установка и запуск приложения

Бэкап конфигурации

```

cp /opt/ElarContext/conf/context.xml /opt/ElarContext/conf/context.xml.bak
cp /opt/ElarContext/conf/server.xml /opt/ElarContext/conf/server.xml.bak

```

Копируем заранее подготовленные три конфиг-файла с вышеописанными параметрами по следующему путем:

```

- elarcntx.cfg → /opt/ElarContext/elarcntx.cfg
- server.xml → /opt/ElarContext/conf/server.xml
- context.xml → /opt/ElarContext/conf/context.xml

```

Останавливаем томкат:

```
systemctl stop tomcat
```

Удаляем папку ROOT

```
rm -rf /opt/ElarContext/webapps/ROOT/
```

Копируем ЭК в папку Tomcat

```
cp /opt/ELARContext-release-3.4-57.war /opt/ElarContext/webapps/ROOT.war
```

Установить права на добавленные конфиг-файлы:

```
chown -R tomcat:tomcat /opt/ElarContext  
chmod -R 775 /opt/ElarContext  
systemctl start tomcat
```

В папке webapps Tomcat должна распаковаться папка ROOT

```
ls -l /opt/ElarContext/webapps/
```

Если все хорошо, то ЭК возможно открыть через браузер по адресу:

<http://10.4.12.126:8080/>