

**КОМПЛЕКС СРЕДСТВ АВТОМАТИЗАЦИИ РЕГИОНАЛЬНОЙ ПЛАТФОРМЫ  
АППАРАТНО-ПРОГРАММНОГО КОМПЛЕКСА «БЕЗОПАСНЫЙ ГОРОД»**

**РУКОВОДСТВО ПО УСТАНОВКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ Т.НІР**

## 1. ПОСЛЕДОВАТЕЛЬНОСТЬ УСТАНОВКИ

Установка компонентов программного обеспечения проводится в следующей последовательности:

1. База данных;
2. Окружение Java;
3. Установка скриптов;
4. Установка сервера приложения Tomcat;
5. Установка Active MQ;
6. Установка ядра программного обеспечения;
7. Установка сервера географической информационной системы;
8. Установка сервера отчетов;
9. Установка сервера интернет-портала.

## 2. УСТАНОВКА И НАСТРОЙКА «БАЗЫ ДАННЫХ»

В качестве базы данных используется СУБД PostgreSQL. С целью инициализации и установки:

```
mkdir /opt/bg/data
mkdir /var/lib/pgsql
mkdir /var/lib/pgsql/10

ln -s /opt/bg/data /var/lib/pgsql/10/data
chown -R postgres:postgres /var/lib/pgsql/10
chown postgres:postgres /opt/bg/data
chown -R postgres:postgres /opt/bg/data/

yum install postgresql10-server postgresql10
/usr/pgsql-10/bin/postgresql-10-setup initdb

vi /var/lib/pgsql/10/data/pg_hba.conf
host    all             all             0.0.0.0/0

vi /var/lib/pgsql/10/data/postgresql.conf
listen_addresses = '*'

systemctl stop postgresql-10.service
systemctl start postgresql-10.service
systemctl enable postgresql-10.service
```

## 3. УСТАНОВКА ОКРУЖЕНИЯ JAVA

Для установки необходимы выполнить скрипт:

```
cd /opt
tar -xzf jdk-8u181-linux-x64.tar.gz
ln -s /opt/jdk1.8.0_181 /opt/java
```

## 4. УСТАНОВКА СКРИПТОВ

```
cp bg-app.sh /opt/bg/scripts
chmod a+rx /opt/bg/scripts/bg-app.sh
```

## 5. УСТАНОВКА СЕРВЕРА ПРИЛОЖЕНИЯ ТОМСАТ

Необходимо скачать дистрибутив Apache Tomcat, далее выполнить инструкции:

```
mkdir /root/downloads
cd /root/downloads

tar -xzf apache-tomcat-7.0.91.tar.gz
mv apache-tomcat-7.0.91 /opt/bg/
ln -s /opt/bg/apache-tomcat-7.0.91 /opt/bg/tomcat
useradd -r tomcat
vi /opt/bg/tomcat/bin/setenv.sh
JAVA_OPTS="$JAVA_OPTS -Djava.awt.headless=true -server -Xms512m -Xmx2G -
XX:MaxMetaspaceSize=1G -XX:+UseConcMarkSweepGC"

vi /opt/bg/tomcat/conf/tomcat-users.xml
<role rolename="tomcat"/>
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager"/>
<user username="admin" password="tomcat" roles="manager-gui,manager-
script"/>
<user username="tomcat" password="tomcat" roles="tomcat"/>
vi /opt/bg/tomcat/conf/server.xml
<Engine name="Catalina" defaultHost="localhost">
  <Realm className="org.apache.catalina.realm.MemoryRealm" />
<Connector port="8080" maxHttpHeaderSize="65536" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443"
  address="192.168.12.4" URIEncoding="UTF-8"/>
cp catalina.properties /opt/bg/tomcat/conf/
```

## 6. УСТАНОВКА ACTIVE MQ

Необходимо скачать Apache Active MQ, далее выполнить инструкции:

```
cd /root/downloads
tar -xzf apache-activemq-5.15.6-bin.tar.gz -C /opt/bg
mv apache-activemq-5.15.6 /opt/bg/
ln -s /opt/bg/apache-activemq-5.15.6 /opt/bg/activemq
useradd -r activemq
chown activemq:activemq /opt/bg/activemq
chown -R activemq:activemq /opt/bg/activemq/
vi /etc/init.d/activemq

#!/bin/bash
#
# activemq          Starts ActiveMQ.
#
```

```

# chkconfig: 345 88 12
# description: ActiveMQ is a JMS Messaging Queue Server.
### BEGIN INIT INFO
# Provides: $activemq
### END INIT INFO
. /opt/bg/scripts/env.conf

RETVAL=0

umask 077

start() {
    echo "Starting ActiveMQ: "
    su -c "/opt/bg/activemq/bin/activemq start" activemq
    return 0
}
stop() {
    echo "Shutting down ActiveMQ: "
    su -c "/opt/bg/activemq/bin/activemq stop" activemq
    return 0
}
restart() {
    stop
    start
    return 0
}
case "$1" in
start)
    start
    exit $?
;;
stop)
    stop
    exit $?
;;
restart|reload)
    restart
    exit $?
;;
*)
    echo $"Usage: $0 {start|stop|restart}"
    exit 1
esac
chmod +x /etc/init.d/activemq
chkconfig --add activemq
chkconfig activemq on

```

## Выполнить настройку Active MQ:

```

vi /opt/bg/activemq/conf/activemq.xml
    <destinationPolicy>
      <policyMap>
        <policyEntries>
          <policyEntry topic="" >
            <pendingMessageLimitStrategy>
              <constantPendingMessageLimitStrategy limit="1000"/>
            </pendingMessageLimitStrategy>
          </policyEntry>
          <policyEntry queue="">
            <deadLetterStrategy>
              <individualDeadLetterStrategy queuePrefix="DLQ."/>
            </deadLetterStrategy>
          </policyEntry>

```

```

        </policyEntries>
      </policyMap>
    </destinationPolicy>
  <transportConnector name="stomp"
uri="stomp://0.0.0.0:61613?maximumConnections=1000&wireFormat.maxFrameSize=209715200"/>

```

## 7. УСТАНОВКА ЯДРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Необходимо загрузить дистрибутив (предоставляется по запросу) и выполнить команды:

```

mkdir /tmp/bg/lucene
chown -R bg-core:bg-core /tmp/bg/lucene

```

Сообщения, обрабатываемые ядром системы бывают трех типов: "Received", "Delivered", "Not\_delivered".

### *Сообщений типа Received*

Сообщение от адаптера в ядро с карточкой происшествия:

Поле	Обязательность	Пример значения/Комментарий
header.JMSType	обязательно	"HELIOS_CARD"
header.Status	обязательно	"Received"
CardData.externalCardId	обязательно	
CardData.incidentType	обязательно	
CardData.ext_system_code	обязательно	из конфига (smp03 или s112)
CardData.dateTime	обязательно	

### *Сообщений типа Delivered*

Сообщение от адаптера об успешной доставке карточки происшествия во внешнюю систему:

Поле	Обязательность	Пример значения/Комментарий
header.JMSType	обязательно	"HELIOS_CARD"
header.Status	обязательно	"Delivered"
CardData.externalCardId	обязательно	
CardData.incidentID	обязательно	из header.IncidentUUID
CardData.ext_system_code	обязательно	из конфига (smp03 или s112)
CardData.transferCall	при наличии в карточке CardData sent	

## ***Сообщений типа Not\_delivered***

Сообщение от адаптера о неуспешной доставке карточки происшествия во внешнюю систему

<b>Поле</b>	<b>Обязательность</b>	<b>Пример значения/Комментарий</b>
header.JMSType	обязательно	"HELIOS_CARD"
header.Status	обязательно	"Not_delivered"
CardData.incidentID	обязательно	из header.IncidentUUID
CardData.ext_system_code	обязательно	из конфига (smp03 или s112)

## ***Исходящие сообщения***

### ***Сообщений типа Sent***

Сообщение из ядра в адаптер с карточкой происшествия:

<b>Поле</b>	<b>Обязательность</b>	<b>Пример значения/Комментарий</b>
header.JMSType	обязательно	"HELIOS_CARD"
header.externalSystemType	обязательно	"SMP03" или "S112"
header.IncidentUUID	обязательно	из CardData.incidentId
header.moID	обязательно	из CardData.mo
CardData.incidentID	обязательно	
CardData.externalCardId	при наличии значения в ядре	
CardData.incidentType	обязательно	
CardData.dateTime	обязательно	

### ***Пример CardData для общего случая***

```
<?xml version="1.0" encoding="UTF-8"?>
<cardData>
  <incidentID>cda31013-cbf8-4982-931a-f98976142123</incidentID>
  <mo>84701000</mo>
  <externalCardId>123qweqwe</externalCardId>
  <ext_system_code>SMP03</ext_system_code>
  <status>REGISTERED</status>
  <dateTime>2019-06-05T12:21:27.148+03:00</dateTime>
  <description>руддщ</description>
  <injuredPersons>0</injuredPersons>
  <deadPersons>0</deadPersons>
  <address>
    <address>р-н Братский, д Куватка, ул Ангарская</address>
    <placeName>р-н Братский, д Куватка, ул Ангарская</placeName>
  </address>
  <coordinateData />
  <callerInfo>
    <phoneNumber />
    <surname />
  </callerInfo>
</cardData>
```

```

        <name />
        <patronymic />
    </callerInfo>
    <autodetectedCallerInfo>
        <phoneNumber />
        <surname />
        <name />
        <patronymic />
    </autodetectedCallerInfo>
    <transferCall>false</transferCall>
    <peopleInDanger>false</peopleInDanger>
    <emergencySituation>false</emergencySituation>
    <additionalPropertyList>
        <key>STR_STREET_KLADR</key>
    </additionalPropertyList>
    <additionalPropertyList>
        <key>STR_DECLARANT_STREET_KLADR</key>
    </additionalPropertyList>
</cardData>

```

## 8. УСТАНОВКА СЕРВЕРА ГЕОГРАФИЧЕСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Необходимо загрузить дистрибутив программного обеспечения (предоставляется по запросу) и выполнить инструкции:

1. Создать пользователя bg-gis с правами LOGIN SUPERUSER INHERIT CREATEDB CREATEROLE NOREPLICATION
2. Создать базу bggis с владельцем bg-gis
3. Применить к базе bggis последовательно скрипты **db\_structure.sql** и **bg\_db\_data.sql**
4. Создать на сервере папку **/opt/bg/tomcat/bggis-conf**
5. Разместить в папке **/opt/bg/tomcat/bggis-conf** файлы **bggis.properties**, **gs-adapter.properties**, **helios.properties**
6. Актуализировать эти конфигурационные файлы
7. Опционально обновить координаты центров обстановок и координаты по умолчанию для создания объектов
8. Опционально включить использование локального геокодера для всех обстановок
9. Разместить .war файлы компонентов bggis
10. Создать базу данных gis с владельцем bg-gis
11. Включить в базе данных gis расширения postgis и uuid-ossr
12. Наполнить базу данных gis данными OSM

Пример конфигурационных файлов:

```

export PATH=$PATH:/usr/pgsql-10/bin
psql -U postgres -c "CREATE ROLE \"bg-gis\" LOGIN SUPERUSER INHERIT
CREATEDB CREATEROLE NOREPLICATION;"
psql -U postgres -c "ALTER USER \"bg-gis\" WITH PASSWORD 'bg-gis';"
psql -U postgres -c "CREATE DATABASE \"bggis\" WITH OWNER = \"bg-gis\"
ENCODING = 'UTF8' LC_COLLATE = 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8'
CONNECTION LIMIT = -1 TEMPLATE = \"template0\";"

```

```

export BGGIS_DB_STRUCTURE_SQL="/opt/gis-install/db_structure.sql"
export BGGIS_DB_DATA_SQL="/opt/gis-install/bg_db_data.sql"

psql -U bg-gis -d bggis -a -f $BGGIS_DB_STRUCTURE_SQL
psql -U bg-gis -d bggis -a -f $BGGIS_DB_DATA_SQL

mkdir /opt/bg/tomcat/bggis-conf
cp /opt/gis-install/gs-adapter.properties /opt/bg/tomcat/bggis-conf/gs-
adapter.properties
cp /opt/gis-install/helios.properties /opt/bg/tomcat/bggis-
conf/helios.properties
cp /opt/gis-install/bggis.properties /opt/bg/tomcat/bggis-
conf/bggis.properties

export DEFAULT_LAT="55.696950"
export DEFAULT_LON="37.360645"

OLD_STRING="map.center.lat=56.151362"
NEW_STRING="map.center.lat=$DEFAULT_LAT"
sed -i "s~$OLD_STRING~$NEW_STRING~g" /opt/bg/tomcat/bggis-
conf/bggis.properties

OLD_STRING="map.center.lon=101.63408"
NEW_STRING="map.center.lon=$DEFAULT_LON"
sed -i "s~$OLD_STRING~$NEW_STRING~g" /opt/bg/tomcat/bggis-
conf/bggis.properties

psql -U bg-gis -d bggis -c "UPDATE situation SET map_center_lat =
$DEFAULT_LAT"
psql -U bg-gis -d bggis -c "UPDATE situation SET map_center_lon =
$DEFAULT_LON"

psql -U bg-gis -d bggis -c "UPDATE situation SET use_local_geocoder =
TRUE"

cp /opt/gis-install/geoserver-adapter.war
/opt/bg/tomcat/webapps/geoserver-adapter.war
cp /opt/gis-install/geoserver.war /opt/bg/tomcat/webapps/geoserver.war
cp /opt/gis-install/bggis.war /opt/bg/tomcat/webapps/bggis.war
service tomcat start

```

## ***Установка и настройка OSRM***

```

yum install -y yum-utils centos-release-scl
yum-config-manager --enable rhel-server-rhscl-7-rpms
yum install -y devtoolset-6
scl enable devtoolset-6 bash

export OSM_PBF_FILE_PATH=/opt/gis-install/OSM-extract.osm.pbf
export OSM_PBF_FILE_NAME=$(basename $OSM_PBF_FILE_PATH)
export OSRM_PBF_FILE_NAME=$(basename $OSM_PBF_FILE_PATH .osm.pbf)

yum -y install git zlib-devel nano
yum remove cmake -y
cd /opt
wget https://cmake.org/files/v3.12/cmake-3.12.3.tar.gz
tar xfv cmake-3.12.3.tar.gz
cd cmake-3.12.3/
./bootstrap --prefix=/usr/local
make
make install

```

```

mkdir /opt/osrm
cd /opt/osrm
git clone https://github.com/Project-OSRM/osrm-backend.git
cd /opt/osrm/osrm-backend
mkdir /opt/osrm/osrm-backend/build
cd /opt/osrm/osrm-backend/build
cmake .. -DENABLE_MASON=ON
make
make install
mkdir /opt/osrm/data
cp -rf $OSM_PBF_FILE_PATH /opt/osrm/data/$OSM_PBF_FILE_NAME
/usr/local/bin/osrm-extract /opt/osrm/data/$OSM_PBF_FILE_NAME -p
/opt/osrm/osrm-backend/profiles/car.lua
/usr/local/bin/osrm-contract /opt/osrm/data/$OSRM_PBF_FILE_NAME
tee /etc/systemd/system/osrm.service << EOF
[Unit]
Description=OSRM service
After=network.target
[Service]
ExecStart = /usr/local/bin/osrm-routed /opt/osrm/data/$OSRM_PBF_FILE_NAME
[Install]
WantedBy=multi-user.target
EOF
systemctl daemon-reload
systemctl enable osrm
systemctl start osrm

```

## ***Установка и настройка Nominatim***

```

export THIS_IP=$(hostname -I | xargs)
export PATH=$PATH:/usr/pgsql-10/bin

yum install -y epel-release
yum install -y      cmake make gcc gcc-c++ libtool policycoreutils-python
\
                        php-pgsql php php-pear php-pear-DB php-intl libpqxx-
devel \
                        proj-epsg bzip2-devel proj-devel libxml2-devel boost-
devel \
                        expat-devel zlib-devel \
                        llvm5.0 devtoolset-7 llvm-toolset-7
yum install bzip2

echo "date.timezone = 'Europe/Moscow'" > /etc/php.d/timezone.ini

useradd -s /bin/bash -m nominatim
echo "nominatim" | passwd "nominatim" --stdin
usermod -aG wheel nominatim

export APACHE_USER="apache"

psql -U postgres -c "CREATE ROLE \"nominatim\" LOGIN SUPERUSER INHERIT
CREATEDB CREATEROLE NOREPLICATION;"
psql -U postgres -c "ALTER USER \"nominatim\" WITH PASSWORD 'nominatim';"
psql -U postgres -c "CREATE ROLE \"$APACHE_USER\" NOSUPERUSER NOCREATEDB
NOCREATEROLE INHERIT LOGIN;"

tee /etc/httpd/conf.d/nominatim.conf << EOFAPACHECONF
<Directory "/opt/nominatim/Nominatim/build/website">
    Options FollowSymLinks MultiViews
    AddType text/html .php
    DirectoryIndex search.php

```

```

    Require all granted
</Directory>

Alias /nominatim /opt/nominatim/Nominatim/build/website
EOFAPACHECONF
systemctl enable httpd
systemctl restart httpd

mkdir /opt/nominatim
chmod a+rxw /opt/nominatim

su nominatim

export NOMINATIM_URL=https://www.nominatim.org/release/Nominatim-
3.1.0.tar.bz2
export NOMINATIM_ARCHIVE_NAME="${NOMINATIM_URL##*/}"
export NOMINATIM_FOLDER_NAME=$(basename $NOMINATIM_ARCHIVE_NAME .tar.bz2)

export APACHE_USER="apache"
export DEFAULT_LAT="55.696950"
export DEFAULT_LON="37.360645"
export DEFAULT_ZOOM="9"
export OSM_PBF_FILE_PATH=/opt/gis-install/OSM-extract.osm.pbf
export THIS_IP=$(hostname -I | xargs)
export RAM_MEGABYTES=$(free -m | awk '/Mem\:/ { print $2 }')
export NOMINATIM_IMPORT_CACHE_SIZE=$(awk "BEGIN {printf \"%0.0f\\",
$RAM_MEGABYTES / 3.0 * 2.0}")

export PATH=$PATH:/usr/pgsql-10/bin

cd /opt/nominatim
wget $NOMINATIM_URL
tar xjf $NOMINATIM_ARCHIVE_NAME
mv $NOMINATIM_FOLDER_NAME Nominatim
cp -rf /opt/gis-install/utfasciitable.h
/opt/nominatim/Nominatim/module/utfasciitable.h

cd /opt/nominatim/Nominatim
mkdir build
cd build
cmake /opt/nominatim/Nominatim -DCMAKE_PREFIX_PATH=/usr/pgsql-10
make

export USERNAME=nominatim
export USERHOME=/opt/nominatim

sudo semanage fcontext -a -t httpd_sys_content_t
"$USERHOME/Nominatim/(website|lib|settings) (/*)?"
sudo semanage fcontext -a -t httpd_sys_content_t
"$USERHOME/Nominatim/build/(website|lib|settings) (/*)?"
sudo semanage fcontext -a -t lib_t
"$USERHOME/Nominatim/build/module/nominatim.so"
restorecon -R -v $USERHOME/Nominatim
restorecon -R -v $USERHOME/Nominatim/build

tee /opt/nominatim/Nominatim/build/settings/local.php << EOF
<?php
@define('CONST_Database_Web_User', '$APACHE_USER');
@define('CONST_Database_DSN',
'pgsql://nominatim:nominatim@$THIS_IP:5432/nominatim');
@define('CONST_Website_BaseURL', 'http://$THIS_IP/nominatim/');
@define('CONST_Use_Extra_US_Postcodes', false);
@define('CONST_Use_US_Tiger_Data', false);
@define('CONST_Use_Aux_Location_data', false);

```

```

@define('CONST_Languages', 'ru');
@define('CONST_Default_Language', 'ru');
@define('CONST_Default_Lat', $DEFAULT_LAT);
@define('CONST_Default_Lon', $DEFAULT_LON);
@define('CONST_Default_Zoom', $DEFAULT_ZOOM);
@define('CONST_PolygonOutput_MaximumTypes', 1);
@define('CONST_Log_DB', false);
@define('CONST_Postgresql_Version', '10');
@define('CONST_Postgis_Version', '2.5');
EOF

#cd /opt/nominatim/Nominatim/data
#wget https://www.nominatim.org/data/wikipedia_article.sql.bin
#wget https://www.nominatim.org/data/wikipedia_redirect.sql.bin

/opt/nominatim/Nominatim/build/utils/setup.php --osm-file
$OSM_PBF_FILE_PATH --all --osm2pgsql-cache $NOMINATIM_IMPORT_CACHE_SIZE
2>&1 | tee /opt/nominatim/setup.log
( /opt/nominatim/Nominatim/build/utils/setup.php --osm-file /u01/central-
fed-district-latest.osm.pbf --all 2>&1 | tee setup.log)

/opt/nominatim/Nominatim/build/utils/update.php --recompute-word-counts
/opt/nominatim/Nominatim/build/utils/specialphrases.php --wiki-import >
/opt/nominatim/Nominatim/data/specialphrases.sql
psql -U nominatim -d nominatim -f
/opt/nominatim/Nominatim/data/specialphrases.sql

```

## 9. УСТАНОВКА СЕРВЕРА ОТЧЕТОВ

Необходимо скачать и установить Jasper Reports, далее выполнить инструкции:

Измените порты на отличные от портов других серверов tomcat, например:

```

<Server port="8006" shutdown="SHUTDOWN">

<Connector port="8083" URIEncoding="UTF-8" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="9443" />

<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8109" protocol="AJP/1.3" redirectPort="9443" />

```

Скопировать JAR библиотеки в папку <js-install>/apache-tomcat/webapps/jasperserver/WEB-INF/lib: - bg-reporting-authentication.jar, - bg-reporting-utils.jar, - bg-reporting-images.jar, - bg-reporting-fonts.jar, - bg-reporting-charts.jar, - JDBC драйвер.

```

cd /root/downloads
mv bg-reporting-authentication.jar /opt/bg/jasperreports/apache-
tomcat/webapps/jasperserver/WEB-INF/lib/
mv bg-reporting-utils.jar /opt/bg/jasperreports/apache-
tomcat/webapps/jasperserver/WEB-INF/lib/
mv bg-reporting-images.jar /opt/bg/jasperreports/apache-
tomcat/webapps/jasperserver/WEB-INF/lib/
mv bg-reporting-fonts.jar /opt/bg/jasperreports/apache-
tomcat/webapps/jasperserver/WEB-INF/lib/
mv bg-reporting-charts.jar /opt/bg/jasperreports/apache-
tomcat/webapps/jasperserver/WEB-INF/lib/

```

```
mv postgresql-42.2.5.jre6.jar /opt/bg/jasperreports/apache-  
tomcat/webapps/jasperserver/WEB-INF/lib/
```

Скопировать содержимое "bg-distribution-bin/jasperserver/apache-tomcat/webapps/jasperserver/scripts" в папку <js-install>/apache-tomcat/webapps/jasperserver/scripts:

```
cp -rf jasperserver/scripts/* /opt/bg/jasperreports/apache-  
tomcat/webapps/jasperserver/scripts/
```

Скопировать файл bg.properties "bg-distribution-bin/jasperserver/apache-tomcat/webapps/jasperserver/WEB-INF/classes/bg.properties" в папку <js-install>/apache-tomcat/webapps/jasperserver/WEB-INF/classes

```
cp /root/downloads/bg.properties /opt/bg/jasperreports/apache-  
tomcat/webapps/jasperserver/WEB-INF/classes/
```

Скопировать содержимое "bg-distribution-bin/jasperserver/apache-tomcat/webapps/jasperserver/WEB-INF/jsp/modules/reportScheduling" в папку <js-install>/apache-tomcat/webapps/jasperserver/WEB-INF/jsp/modules/reportScheduling:

```
cp -rf jasperserver/WEB-INF/jsp/modules/reportScheduling/*  
/opt/bg/jasperreports/apache-tomcat/webapps/jasperserver/WEB-  
INF/jsp/modules/reportScheduling/
```

Скопировать файл applicationContext-externalAuth-db.xml в папку <js-install>/apache-tomcat/webapps/jasperserver/WEB-INF:

```
cp jasperserver/WEB-INF/applicationContext-externalAuth-db.xml  
/opt/bg/jasperreports/apache-tomcat/webapps/jasperserver/WEB-INF/
```

В файле applicationContext-externalAuth-db.xml заменить параметр serviceUrl на актуальный URL сервисов модуля core.

```
vi /opt/bg/jasperreports/apache-tomcat/webapps/jasperserver/WEB-  
INF/applicationContext-externalAuth-db.xml  
<bean name="userService" parent="hessianProxyFactoryBean">  
    <property name="serviceUrl" value="http://192.168.12.4:8081/user" />  
    <property name="serviceInterface"  
value="ru.reksoft.s112.core.service.UserService" />  
</bean>
```

Добавить описание соединения к БД системы "bg" в контексте Jasper Server-a context.xml)

```
vi /opt/bg/jasperreports/apache-tomcat/webapps/jasperserver/META-  
INF/context.xml  
<Resource name="jdbc/postgresql/bg"  
    auth="Container"  
    driverClassName="org.postgresql.Driver"  
    maxActive="50"  
    maxIdle="30"  
    maxWait="-1"  
    type="javax.sql.DataSource"  
    url="jdbc:postgresql://192.168.12.2:5432/bg-core"  
    username="bg-core"  
    password="bg-core"/>
```

Добавить поддержку русской Locale (ru\_RU), для этого в файл applicationContext-security.xml добавить в bean "userLocalesList" следующее значение "<value type="java.util.Locale">ru</value>".

```
vi /opt/bg/jasperreports/apache-tomcat/webapps/jasperserver/WEB-INF/applicationContext-security.xml
```

В applicationContext.xml выполнить следующие действия:

```
vi /opt/bg/jasperreports/apache-tomcat/webapps/jasperserver/WEB-INF/applicationContext.xml
```

1. Добавить поддержку кодировки "UTF-8" для всех файлов локализации, для этого в файл applicationContext.xml добавить в bean "messageSource" следующее значение "<property name="defaultEncoding" value="UTF-8"/>".
2. Добавить поддержку "иконок" в случае экспорта отчёта в файл формата MS Excel, для этого в файле applicationContext.xml изменить значение свойства "ignoreGraphics" в bean-е "xlsExportParameters" на "false", т.е. "<property name="ignoreGraphics" value="false"/>".
3. Изменить разделитель CSV (по умолчанию ",",) на разделитель соответствующий русской локале, т.е. на ";", для этого в файле applicationContext.xml изменить значение свойства fieldDelimiter, т.е. <property name="fieldDelimiter" value=";" />

Установить видимость колонтитулов для формата MS Excel, для этого в файле jasperreports.properties удалить или сделать комментарием строки:

```
vi /opt/bg/jasperreports/apache-tomcat/webapps/jasperserver/WEB-INF/classes/jasperreports.properties
#net.sf.jasperreports.export.xls.exclude.origin.band.2=pageFooter
#net.sf.jasperreports.export.xls.exclude.origin.band.4=columnFooter
```

для корректного отображения шрифтов добавить настройку в jasperreports.properties:

```
# ignore missing fonts in reports
net.sf.jasperreports.awt.ignore.missing.font=true
net.sf.jasperreports.default.font.name=Arial
net.sf.jasperreports.default.pdf.font.name=Arial
net.sf.jasperreports.default.pdf.encoding=Cp1251
```

Для настройки почты следует: - в файле applicationContext-report-scheduling.xml в bean-е "reportSchedulerMailSender" установить свойство mail.smtp.auth в true (в случае, если на почтовом сервере аутентификация по протоколу smpt установлена обязательной);

```
vi /opt/bg/jasperreports/apache-tomcat/webapps/jasperserver/WEB-INF/applicationContext-report-scheduling.xml
```

- в файле js.quartz.properties установить основные настройки почтового сервера, т.е.:

```
vi /opt/bg/jasperreports/apache-tomcat/webapps/jasperserver/WEB-INF/js.quartz.properties
report.scheduler.mail.sender.host=s112-app.reksoft.ru
```

```
report.scheduler.mail.sender.username=test@s112.ru
report.scheduler.mail.sender.password=test
report.scheduler.mail.sender.from=test@s112.ru
report.scheduler.mail.sender.protocol=smtp
report.scheduler.mail.sender.port=25
```

**Изменить regex в validation.properties, это необходимо для работы отчетов, запросы в которых не начинаются сразу со слова select.**

```
vi /opt/bg/jasperreports/apache-tomcat/webapps/jasperserver/WEB-INF/classes/esapi/validation.properties
"Validator.ValidSQL=(?is)^\s*(select)\s+[\^;]+;\?\s*$" на
"Validator.ValidSQL=(?is)^\s*(select|DECLARE|SET)\s+.*$"
```

**Измените значение 'security.validation.sql.on' на false в файле:**

```
vi /opt/bg/jasperreports/apache-tomcat/webapps/jasperserver/WEB-INF/classes/esapi/security-config.properties
```

- **в файле js.quartz.properties измените report.scheduler.web.deployment.uri jasperserver-pro на jasperserver и укажите правильный порт:**

```
vi /opt/bg/jasperreports/apache-tomcat/webapps/jasperserver/WEB-INF/js.quartz.properties
```

```
report.scheduler.web.deployment.uri=http://192.168.12.4:8083/jasperserver
```

```
useradd -r jasperreports
chown jasperreports:jasperreports /opt/bg/jasperreports
chown -R jasperreports:jasperreports /opt/bg/jasperreports/
chown -R postgres:postgres /opt/bg/jasperreports/postgresql/
```