



Руководство администратора

Установка приложения GetRail

Контроль над документом

Запись изменений

Дата	Автор	Версия	Состав изменений

Рецензенты

Дата	ФИО	Версия

Согласовано

Должность	ФИО	Подпись	Дата

Список рассылки

Копия	Имя	Местонахождение

Связанные документы

№ копии	Название	Файл	Описание

Содержание

Контроль над документом.....	2
Запись изменений.....	2
Рецензенты	2
Согласовано	2
Список рассылки.....	2
Связанные документы	2
Установка схем базы данных.....	4
Настройка сервера приложений.....	8
Установка программного обеспечения	8
Список ПО, требуемого для работы приложения:.....	8
Описание и настройка приложений:	8
Tomcat	8
MongoDB.....	9
Redis.....	10
RabbitMQ.....	10
Установка GetRail на сервере приложений:	10

Установка схем базы данных

Postgres должен быть установлен в папку c:\PostgreSQL\13.

Установка расширений:

Поскольку для создания БД по умолчанию используется шаблон template1, то требуется установить extensions в него.

Для этого выполнить команду:

```
psql -U postgres -d template1
```

Вы подключены к базе данных "template1" как пользователь "postgres".

```
template1=# CREATE EXTENSION dblink;
template1=# CREATE EXTENSION plpgsql;
template1=# CREATE EXTENSION "uuid-oss";
template1=# CREATE EXTENSION adminpack;
```

-- Установить расширение plpgsql_check

1. copy `plpgsql_check.dll` to `PostgreSQL\9.3\lib`
2. copy `plpgsql_check.control` and `plpgsql_check--0.8.sql` to `PostgreSQL\9.3\share\extension`

```
psql -U postgres -d template1
```

```
template1=# CREATE EXTENSION plpgsql_check;
```

-- Установить расширение pljava

```
c:\OpenJDK\jdk-16.0.1\bin\java -jar c:\INSTALL\pljava-pg13.jar
```

```
psql -U postgres -d template1
```

```
SET pljava.libjvm_location TO 'c:\OpenJDK\jdk-16.0.1\bin\server\jvm.dll';
ALTER DATABASE template1 SET pljava.libjvm_location FROM CURRENT;
show pljava.libjvm_location;
```

```
template1=# CREATE EXTENSION pljava;
```

Установка приложения GetRail

-- Установить расширение pgagent

Распаковать pgagent.zip в c:\PostgreSQL\13\share\extension\

Распаковать pgagent_exe.zip в c:\PostgreSQL\13\bin\

```
psql -d template01
```

Проверка установленных расширений:

```
psql -d template01
```

```
\dx
```

Список установленных расширений

Имя	Версия	Схема	Описание
adminpack	2.1	pg_catalog	administrative functions for PostgreSQL
dblink	1.2	public	connect to other PostgreSQL databases from within a database
pgagent	4.2	pgagent	A PostgreSQL job scheduler
pljava		2-SNAPSHOT	sqlj PL/Java procedural language (https://tada.github.io/pljava/)
plpgsql	1.0	pg_catalog	PL/pgSQL procedural language
plpgsql_check	1.15	public	extended check for plpgsql functions
uuid-oss	1.1	public	generate universally unique identifiers (UUIDs)

(7 строк)

Получен дамп АСУТЛ на postgres (test_30.03.2022.dump), разворачиваем его в новую БД test:

```
psql
```

```
CREATE DATABASE test;
```

```
CREATE ROLE "NB" WITH
```

```
LOGIN
```

```
NOSUPERUSER
```

```
NOINHERIT
```

```
NOCREATEDB
```

```
NOCREATEROLE
```

```
NOREPLICATION
```

```
ENCRYPTED PASSWORD 'md527a6c059c1c304d15057ed2f12bafbe4';
```

Установка приложения GetRail

```
CREATE ROLE "ISUGDP_GROUP" WITH  
NOLOGIN  
NOSUPERUSER  
INHERIT  
NOCREATEDB  
NOCREATEROLE  
NOREPLICATION  
VALID UNTIL 'infinity';
```

```
COMMENT ON ROLE "ISUGDP_GROUP" IS 'Группа ИС УЖДП';
```

```
CREATE ROLE adapter WITH  
LOGIN  
SUPERUSER  
INHERIT  
NOCREATEDB  
NOCREATEROLE  
NOREPLICATION  
VALID UNTIL 'infinity';
```

```
GRANT "ISUGDP_GROUP" TO adapter;
```

```
CREATE ROLE asutl WITH  
LOGIN  
SUPERUSER  
INHERIT  
NOCREATEDB  
NOCREATEROLE  
NOREPLICATION  
ENCRYPTED PASSWORD 'md5333a9abcb8dd84ee3271dc0398ee6e48';
```

```
CREATE ROLE isugdp WITH  
LOGIN  
NOSUPERUSER  
INHERIT  
NOCREATEDB  
NOCREATEROLE
```

Установка приложения GetRail

```
NOREPLICATION
```

```
ENCRYPTED PASSWORD 'md561b2c9754a326e0cb463aed350a60345'
```

```
VALID UNTIL 'infinity';
```

```
GRANT "ISUGDP_GROUP" TO isugdp;
```

Далее необходимо распаковать архив PG_CATALOG.zip и прогнать в базе test:

```
psql -d test -f CREATE_DOMAINS.sql -f DBTIMEZONE.sql -f INSTR.sql -f NVL.sql -f REPLACE.sql  
-f SESSIONTIMEZONE.sql -f SYSDATE.sql -f SYSTIMESTAMP.sql -f TABLE.sql -f TO_CLOB.sql -f  
TO_TIMESTAMP_TZ.sql -f VALUE.sql -f ADD_MONTHS.sql
```

Запустить импорта дампа:

```
psql -d test -f test_30.03.2022.dump
```

Настройка сервера приложений

Установка программного обеспечения

Список ПО, требуемого для работы приложения:

1. Java JDK версии 8
2. Tomcat версии 8/9 - сервер приложений
3. MongoDB версии 4 или выше - база данных
4. Redis версии 5.0.5 или выше - база данных типа key: value
5. RabbitMQ версии 3.8.2 или выше - менеджер очередей
6. Nginx - Web сервер

Tomcat + JDK:

<https://routerus.com/how-to-install-tomcat-9-on-ubuntu-20-04/>

MongoDB

<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/>

Redis

<https://redis.io/docs/getting-started/installation/install-redis-on-linux/>

RabbitMQ

<https://losst.ru/ustanovka-rabbitmq-v-ubuntu-20-04>

Описание и настройка приложений:

Tomcat

Приложение для экземпляра prod устанавливается в папку /opt/tomcat8

Порт для экземпляра prod: 8080

Веб-интерфейс доступен по данным портам.

Логин/пароль: admin / acubmb

Команды:

Prod servicename: tomcat8

service "servicename" start - запуск

service "servicename" restart - перезапуск

service "servicename" stop - остановка

Логи prod: /opt/tomcat8/log/catalina.out

Установка приложения GetRail

Конфигурационные файлы:

Настройка контекста (подключение к внешним сервисам) для экземпляра prod:
/opt/tomcat8/conf/context.xml

Добавлены блоки:

Подключение к PostgreSQL:

```
<Resource name="jdbc/asutlfx-postgress" auth="Container"
  type="javax.sql.DataSource" maxActive="100"
  maxIdle="30" maxWait="10000"
  username="asutl"
  password="asutl"
  driverClassName="org.postgresql.Driver"
  url="jdbc:postgresql://app_server.com:5432/ugdtdb"/>
```

Подключение к LDAP:

```
<Resource name="asutl/ldap" auth="Container"
  type="ru.itlc.asutl.ldap.LdapConnectionFactory"
  factory="ru.itlc.asutl.ldap.LdapConnectionFactory"
  INITIAL_CONTEXT_FACTORY="com.sun.jndi.ldap.LdapCtxFactory"
  SECURITY_AUTHENTICATION="simple"
  SECURITY_PRINCIPAL="itluser1"
  SECURITY_CREDENTIALS="***"
  PROVIDER_URL="ldap://ldap.com"
  REFERRAL="follow"
  DC="DC=DC=DC=com"/>
```

MongoDB

Приложение для установлено в папку /opt/mongodb

Порт: 27017

В базе данных развернуть схемы:

asutl - для экземпляра prod

Установка приложения GetRail

Команды:

```
service mongodb start - запуск  
service mongodb restart - перезапуск  
service mongodb stop - остановка  
mongodb - консольный клиент
```

Redis

Приложение для установлено из исходников с помощью `make install` (файлы приложения в размещены в стандартных для Linux папках)

Порт: 6379

Команды:

```
service redis start - запуск  
service redis restart - перезапуск  
service redis stop - остановка  
redis-cli - консольный клиент
```

RabbitMQ

Приложение установлено в среде Docker ввиду невозможности его установки при отсутствии интернета.

Развернуты два контейнера:

```
rabbitmq - для экземпляра prod
```

Порт для экземпляра prod: 15672

Веб-интерфейс доступен по данным портам.

Логин/пароль: `guest / guest`

Установка GetRail на сервере приложений:

Система АСУТЛ состоит из нескольких сервисов:

```
Asutl-rest - серверная часть приложения АСУТЛ  
AsutlFX пользовательский интерфейс  
Asutl-producer - сервис для доставки различных сообщений  
Asutl-gir - график исполненной работы
```

Установка приложения GetRail

Для работы приложения необходимо произвести первичную установку компонента Asutl-producer. Данный компонент представляет из себя Java-приложение, которое может быть запущено как сервис с использованием встроенных средств Linux:

Необходимо создать файл asutl-producer.service в папке /etc/systemd/system/.

Пример содержания файла:

```
[Unit]
Description="Asutl2-producer service"
After=network.target

[Service]
ExecStart=/usr/bin/java -jar /home/itladmin/asutl/producer/asutl-producer-prod.jar
User=root

WorkingDirectory=/home/itladmin/asutl/producer/
SuccessExitStatus=143
TimeoutStopSec=10
Restart=on-failure
RestartSec=5

[Install]
WantedBy=multi-user.target
```

После создания файлов необходимо выполнить команду:

```
systemctl daemon-reload
```

Управление сервисами осуществляется стандартными командами Linux.

В дополнение к jar-файлам идут файлы config.xml в которых необходимо указать Настройки подключений к различным источникам и потребителям данных.

Пример конфигурации для Asutl-producer:

```
<?xml version="1.0" encoding="UTF-8"?>
<app-configuration>
  <!--URL Данные подключения к RabbitMQ-->
  <rabbitUrl>localhost</rabbitUrl>
  <rabbitPort>5672</rabbitPort>
  <rabbitLogin>guest</rabbitLogin>
  <rabbitPassword>guest</rabbitPassword>
  <!--RabbitMQ exchange name-->
  <rabbitExchangeName>debug</rabbitExchangeName>
  <!-- Данные коннекта к БД -->
  <sqlDriver>postgres.jdbc.postgresDriver</sqlDriver>
  <sqlUrl>jdbc:postgres:thin:@CONNECTION_URL:PORT:SCHEMA</sqlUrl>
  <sqlUser>USER</sqlUser>
  <sqlPassword>PASSWORD</sqlPassword>
  <!--ID предприятия-->
  <cntId>1</cntId>
  <!-- URL к базе REDIS -->
  <redisUrl>localhost</redisUrl>
  <!-- Пароль к БД -->
  <redisPassword>REDIS_PASSWORD</redisPassword>
  <!-- Номер БД REDIS -->
  <redisDB>0</redisDB>
</app-configuration>
```

Установка приложения GetRail

Для установки приложения на сервере должен присутствовать установочный скрипт `install.sh`:

```
#!/bin/bash
DATE=$(date +%Y_%m_%d_%H-%M)
REST_URL=$REST_URL
CLIENT_VERSION=$CLIENT_VERSION
TOMCAT_PATH="/ПУТЬ_СТАНОВКИ_ТОМКАТ/webapps/ROOT"
TOMCAT_CREDS="ПАРОЛЬ:ЛОГИН"
VERSION=$VERSION
MONGO_DB=$MONGO_DB
DIR=DEVELOP_DIR
PWD="ПУТЬ_К_ПАПКЕ/asutl.tar.gz"
LL="`$PWD | grep asutl.tar.gz`"

echo "Проверка наличия новой версии $DATE"
echo $LL

if [[ "${LL}" == 'asutl.tar.gz' ]]; then
    tar -C $PWD -xvf $PWD/asutl.tar.gz
    . $PWD/env.sh
    sleep 30
    mongo <<EOF
use asutl
db.setup.update({id: 1}, {\$set: {version: "${CLIENT_VERSION}", dateUpdate:
"${DATE}", fileUrl: "http://${REST_URL}/asutlFX-Main-${CLIENT_VERSION}.jar"}})
quit()
EOF
    echo "Восстановление БД. Код завершения: $?"
    echo "Установка приложения ASUTL2"
    cp $PWD/asutlFX-Main-"${CLIENT_VERSION}".jar ${TOMCAT_PATH}
    curl -T "$PWD/asutlFX-REST-$VERSION.war"
"http://${TOMCAT_CREDS}@${REST_URL}/manager/text/deploy?path=/asutlFX-REST&update=true"
    sleep 10
    curl -X GET "http://${REST_URL}/asutlFX-
REST/rest/api/asutl/v2.0/liquibase/update"
    rm -rf $PWD/*
    echo "Приложение ASUTL2 успешно установлено"
```

Установка приложения GetRail

```
else
    echo "Нет файлов для установки"
    exit 0
fi
```

В данном скрипте необходимо заполнить переменные:

```
ТОМКАТ_PATH="/ПУТЬ_СТАНОВКИ_ТОМКАТ/webapps/ROOT"
ТОМКАТ_CREDS="ПАРОЛЬ:ЛОГИН"
PWD="ПУТЬ_К_ПАПКЕ/asutl.tar.gz"
```

Приложение АСУТЛ2 поставляется в виде архива asutl.tar.gz. Путь к данному архиву должен, соответствующей переменной:

```
PWD="ПУТЬ_К_ПАПКЕ/asutl.tar.gz"
```

Запуск установки обновления производится путем запуска скрипта, командой:

```
/bin/bash /ПУТЬ_К_СКРИПТУ/install.sh
```

Для вывода процесса установки в файл лога возможно запустить скрипт с выводом в файл:

```
/bin/bash /ПУТЬ_К_СКРИПТУ/install.sh >> /ПУТЬ_К_ФАЙЛУ_ВЫВОДА/
```

В случае успешной установки в консоль (или в файл лога) будет выведено сообщение: "Приложение ASUTL2 успешно установлено"

В случае неуспешной установки в консоль (или файл лога) должно быть выведено сообщение об ошибке.

Установка ГИРа

Компонент Asutl-gir состоит из двух сервисов: gir-aggregator и gir-producer, а так же из набора статичных файлов, для отображения информации в браузере. Для работы компонента Asutl-gir необходим установленный и настроенный веб-сервер nginx.

Пример настройки nginx:

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    root /var/lib/tomcat8/webapps/ROOT;
    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html index.php;
    server_name _;
    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }
}
```

Установка приложения GetRail

```
location /gir{
    alias /home/gir/html;
}
location /gir/rest/{
    proxy_pass http://127.0.0.1:8085;
    proxy_redirect off;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
}
}
```

При данной настройке статичный файлы компонента приложения должны находиться в папке /home/gir/html.

Для запуска компонентов gir-aggregator и gir-producer в качестве сервиса необходимо создать файлы сервисов как описано для Asutl-producer.

В дополнение к jar-файлам идут файлы config.xml в которых необходимо указать Настройки подключений к различным источникам и потребителям данных.

Пример конфигурации для gir-aggregator:

```
<?xml version="1.0"?>
<configurations>
  <configuration type="mongodb">
    <host>localhost</host>
    <port>27017</port>
    <db>gir</db>
  </configuration>
  <configuration type="rabbitmq">
    <host>localhost</host>
    <port>5673</port>
    <login>guest</login>
    <password>guest</password>
    <exchange>asutl_test.gir</exchange>
  </configuration>
</configurations>
```

Пример конфигурации для gir-producer:

```
<?xml version="1.0"?>
<configurations>
  <configuration type="db">
    <type>postgres</type>
    <host>postgres_URL</host>
    <port>postgres_PORT</port>
    <db>SCHEMA</db>
    <login>postgres_USER</login>
    <password>postgres_PASSWORD</password>
  </configuration>
  <configuration type="rabbitmq">
    <host>localhost</host>
    <port>5673</port>
    <login>guest</login>
    <password>guest</password>
    <exchange>asutl_test.gir</exchange>
  </configuration>
</configurations>
```

Установка приложения GetRail

```
</configuration>  
</configurations>
```