

БФТ.Бюджетный контроль

Инструкция по скачиванию, установке и запуску экземпляра программного обеспечения

Листов 21

СОДЕРЖАНИЕ

1.	Скачивание дистрибутива.....	3
2.	Установка системы «БФТ.Бюджетный контроль».....	3
2.1	Установка OpenJDK 11	3
2.2	Установка PostgreSQL-12.....	4
2.3	Создание базы.....	4
2.4	Установка и настройка Tomcat 8	5
2.5	Установка приложения «БФТ.Бюджетный контроль»	9
2.6	Настройка портов брандмауэра	14
2.7	Настройка логирования	14
3.	Запуск системы «БФТ.Бюджетный контроль» и настройка конфигурации.....	16
3.1	Вход в систему.....	16
3.2	Установка конфигурации.....	18

1. Скачивание дистрибутива

Для получения дистрибутива необходимо пройти по ссылке, ввести пароль и скачать файл (azkbk_build.zip) из облачного хранилища.

Ссылка: <https://bftcloud.bftcom.com/index.php/s/KxZ8hzWWbxE9YM4>

Пароль: Ly6AIJ81

Архив содержит в себе файлы:

- **app.war** – файл сборки с приложением, устанавливается на сервер приложения;
- **azkbk_14_configuration.zip** – файл конфигурации, который разворачивается уже на запущенном приложении.

2. Установка системы «БФТ.Бюджетный контроль»

Порядок установки

- 1) Установка OpenJDK 11;
- 2) Установка PostgreSQL-12;
- 3) Создание базы приложения;
- 4) Установка и настройка Tomcat 8;
- 5) Установка приложения «БФТ.Бюджетный контроль»;
- 6) Настройка портов брандмауэра;
- 7) Настройка логирования.

2.1 Установка OpenJDK 11

Для установки OpenJRE 11 необходимо выполнить следующие действия:

- 1) Перейти в home директорию:

```
cd ~
```

- 2) Обновить кэш:

```
sudo yum makecache
```

- 3) Установить OpenJDK:

```
sudo yum install java-11-openjdk-devel
```

4) Выбрать java 11 по умолчанию:

```
sudo alternatives --config java
```

2.2 Установка PostgreSQL-12

Установить PostgreSQL-12

```
sudo yum install postgresql12 postgresql12-server postgresql12-contrib  
postgresql12-libs -y
```

Создать экземпляр базы:

```
sudo postgresql-setup initdb
```

Изменить аутентификации по паролю:

```
sudo vi /var/lib/pgsql/data/pg_hba.conf  
host all all 127.0.0.1/32 ident  
host all all ::1/128 ident
```

Изменить на:

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

Установить пароль для пользователя postgres:

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

Ввести команду:

```
\password postgres
```

Ввести пароль.

Выйти из psql:

```
\q
```

2.3 Создание базы

Создать базу azki:

```
CREATE DATABASE azki;
```

2.4 Установка и настройка Tomcat 8

Установка Tomcat 8

Для установки Tomcat 8 необходимо:

Установить Tomcat:

```
sudo yum install tomcat8-user tomcat8
```

Создать инстанс в необходимой директории (название модуля и порты указать свои):

```
tomcat8-instance-create -p 8083 -c 8015 azki
```

Настройка Tomcat 8

Положить файл `application.properties` в папку с томкатом.

```
#unix: -Djline.terminal=org.springframework.shell.core.IdeTerminal
```

```
#windows: -Djline.WindowsTerminal.directConsole=false -
```

```
Djline.terminal=jline.UnsupportedTerminal
```

```
dm.fieldSetPackages=com.bftcom.azki.common,com.bftcom.configurator.com  
mons,com.bftcom.reports.common,com.bftcom.ice.common,com.bftcom.icewi  
dgets.common,com.bftcom.widgets.store.common
```

```
ice.servicesPackages=com.bftcom.azki.common.service,com.bftcom.configura  
tor.common.services,com.bftcom.ice.common.service
```

```
ice.rls.enabled = true
```

```
ice.script.jvmPackages= com.bftcom.ice.common.maps.*,\
```

```
com.bftcom.ice.server.util.*,\
```

```
com.bftcom.ice.common.general.*,\
```

```
com.bftcom.ice.common.service.notification.*,\
```

```
com.bftcom.ice.common.service.notification.NotificationApi.*,\
```

```
com.bftcom.ice.common.utils.*
```

spring.profiles.active=postgresql

Настройка соединения

spring.datasource.url=jdbc:postgresql://localhost:5432/azki

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.compression.enabled=true

server.compression.min-response-size=2048

server.compression.mime-

types=application/pdf,application/json,application/xml,text/html,application/javascript,text/css,text/xml,text/plain

spring.servlet.multipart.max-file-size=32MB

spring.servlet.multipart.max-request-size=32MB

server.port=8080

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.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

timeMachine.enabled=true

spring.shell.interactive.enabled=false

#do disable RabbitMQ init connection on start and Quartz

spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.amqp.RabbitAutoConfiguration,org.springframework.boot.autoconfigure.quartz.QuartzAutoConfiguration,org.camunda.bpm.spring.boot.starter.CamundaBpmAutoConfiguration

disable BPM

camunda.bpm.enabled=false

log.access.enabled=false

log.access.httpRequests.maxPayloadLength=1000

log.access.httpRequests.includePayload=true

camunda.bpm.application.delete-upon-undeploy=true

Enable document conversions using locally installed

LibreOffice/OpenOffice.org

jodconverter.local.enabled=false

external.storage.enabled=false

external.storage.default=false

external.storage.archive.systemCode=REPORTS

external.storage.archive.scheme=http

external.storage.archive.host=doc-archive:8080/app

external.storage.archive.user=root

external.storage.archive.password=root

reports.storage=ARCHIVE

reports.planeta.url=http://planeta

replication

azk.replication.enabled=false

azk.replication.autostart=true

azk.replication.kafka.host=srv-test-esia

azk.replication.kafka.port=9093

azk.replication.kafka.group=11

azk.clients=[\

{\

"name": "TEST_ESIA_AZK", \

"displayName" : "АЦК Финансы", \

"description": "АЦК Финансы", \

"timeOffsetHours":0, \

"rmiHost": "srv-test", \


```
"rmiPort": 2095,\n"replicationType": "DEBEZIUM",\n"schema": "public"\n}\n]
```

```
azk.plan.rmi.extCode=TEST_ESIA_AZKSPB\nazk.plan.rmi.login=root\nazk.plan.rmi.password=toor\nazk.plan.rmi.extId=1
```

```
azk.replication.kafka.poll.intervalSeconds=500\nazk.replication.max.poll.records=10000\nazk.replication.max.partition.fetch.bytes=5000001\nazk.replication.fetch.max.bytes=10000001\nazk.replication.lockup.intervalSeconds=0\nazki.replication.tables=unit,budget\nazki.replication.classifiers=kes\nazk.replication.login=root
```

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

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

2.5 Установка приложения «БФТ.Бюджетный контроль»

Скопировать файл приложения app.war в директорию \\webapps сборки Tomcat;

Положить файл application.properties в папку с томкатом.

```
#unix: -Djline.terminal=org.springframework.shell.core.IdeTerminal
```

```
#windows: -Djline.WindowsTerminal.directConsole=false -
Djline.terminal=jline.UnsupportedTerminal
dm.fieldSetPackages=com.bftcom.azki.common,com.bftcom.configurator.com
mons,com.bftcom.reports.common,com.bftcom.ice.common,com.bftcom.icewi
dgets.common,com.bftcom.widgets.store.common

ice.servicesPackages=com.bftcom.azki.common.service,com.bftcom.configura
tor.common.services,com.bftcom.ice.common.service
ice.rls.enabled = true

ice.script.jvmPackages= com.bftcom.ice.common.maps.*,\
    com.bftcom.ice.server.util.*,\
    com.bftcom.ice.common.general.*,\
    com.bftcom.ice.common.service.notification.*,\
    com.bftcom.ice.common.service.notification.NotificationApi.*,\
    com.bftcom.ice.common.utils.*

spring.profiles.active=postgresql

# Настройка соединения
spring.datasource.url=jdbc:postgresql://localhost:5432/azki
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.compression.enabled=true
```

```
server.compression.min-response-size=2048
server.compression.mime-
types=application/pdf,application/json,application/xml,text/html,application/java
script,text/css,text/xml,text/plain
```

```
spring.servlet.multipart.max-file-size=32MB
spring.servlet.multipart.max-request-size=32MB
```

```
server.port=8080
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.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
```

```
timeMachine.enabled=true
```

spring.shell.interactive.enabled=false

#do disable RabbitMQ init connection on start and Quartz

spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.amqp.RabbitAutoConfiguration,org.springframework.boot.autoconfigure.quartz.QuartzAutoConfiguration,org.camunda.bpm.spring.boot.starter.CamundaBpmAutoConfiguration

disable BPM

camunda.bpm.enabled=false

log.access.enabled=false

log.access.httpRequests.maxPayloadLength=1000

log.access.httpRequests.includePayload=true

camunda.bpm.application.delete-upon-undeploy=true

Enable document conversions using locally installed LibreOffice/OpenOffice.org

jodconverter.local.enabled=false

external.storage.enabled=false

external.storage.default=false

external.storage.archive.systemCode=REPORTS

external.storage.archive.scheme=http

external.storage.archive.host=doc-archive:8080/app

external.storage.archive.user=root

external.storage.archive.password=root

reports.storage=ARCHIVE

reports.planeta.url=http://planeta

replication

azk.replication.enabled=false

azk.replication.autostart=true

azk.replication.kafka.host=svr-test-esia

azk.replication.kafka.port=9093

azk.replication.kafka.group=11

azk.clients=[\

{\

"name": "TEST_ESIA_AZK", \

"displayName" : "АЦК Финансы", \

"description": "АЦК Финансы", \

"timeOffsetHours":0, \

"rmiHost": "svr-test", \

"rmiPort": 2095,\

"replicationType": "DEBEZIUM",\

"schema": "public"\

}\

]

azk.plan.rmi.extCode=TEST_ESIA_AZKSPB

azk.plan.rmi.login=root

azk.plan.rmi.password=toor

azk.plan.rmi.extId=1

azk.replication.kafka.poll.intervalSeconds=500

azk.replication.max.poll.records=10000

azk.replication.max.partition.fetch.bytes=5000001

```
azk.replication.fetch.max.bytes=10000001
```

```
azk.replication.lockup.intervalSeconds=0
```

```
azki.replication.tables=unit,budget
```

```
azki.replication.classifiers=kes
```

```
azk.replication.login=root
```

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

```
export CATALINA_OPTS="-
```

```
Dspring.config.location=file:полный_путь_до_application.properties"
```

2.6 Настройка портов брандмауэра

Для возможности подключения к серверу с других машин в сети на данной машине необходимо разрешить порту приложения (порт приложения указан в файле server.xml в поддиректории Tomcat'a conf – в параметре port тега Connector) принимать запросы извне.

```
sudo firewall-cmd --zone=public --permanent --add-port=8080/tcp
```

Для применения правил необходимо их перезагрузить:

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

2.7 Настройка логирования

Для настройки логирования необходимо:

Создать файл logger.xml в директории Tomcat'a с содержимым:

```
<configuration>
```

```
  <appender name="STDOUT"
```

```
    class="ch.qos.logback.core.ConsoleAppender">
```

```
      <encoder>
```

```
        <pattern>%d{YYYY-MM-dd HH:mm:ss.SSS} [%thread] %-5level
```

```
%logger{36} - %msg%n</pattern>
```

```
      </encoder>
```

```

    </appender>
<appender name="FILE"
class="ch.qos.logback.core.rolling.RollingFileAppender"> <!--
логгер
работает с файлом -->
<File>logs/access.log</File> <!--расположение файла -->
<rollingPolicy <rollingPolicy
class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy"><!--
политика
для
разделения файлов логов по времени -->
<FileNamePattern>logs/access.%d{yyyy-MM-
dd}.%i.log.zip</FileNamePattern> <!--
шаблон для
имени файла, в который будут архивироваться логи -->
<timeBasedFileNamingAndTriggeringPolicy
class="ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP">
    <maxFileSize>10MB</maxFileSize> <!--максимальный
размер файлов логов-->
</timeBasedFileNamingAndTriggeringPolicy>
    </rollingPolicy>
    <append>true</append>
    <immediateFlush>true</immediateFlush>
    <encoder>
        <pattern>%d{YYYY-MM-dd HH:mm:ss.SSS}
%msg%n</pattern>
    </encoder>
</appender>
<logger name="com.bftcom.ice.server.logging.AccessLoggingFilter">

```

```

    <appender-ref ref="FILE"/>
    <level value="DEBUG"/>
</logger>
<root level="INFO">
    <appender-ref ref="STDOUT"/>
</root>
<logger
name="com.bftcom.docarch.server.rest.DocArchiveDocumentController
" level="DEBUG"/>
    <logger name="com.bftcom.ice.server.services.DataServiceImpl"
level="OFF"/>
    <logger name="com.bftcom.ice.server.services.DeltaMachine"
level="OFF"/>
    <logger
name="com.bftcom.ice.server.security.SecuredDataServiceImpl"
level="OFF"/>
</configuration>

```

Прописать в catalina.sh:

```

export CATALINA_OPTS="-
Dlogback.configurationFile=$CATALINA_HOME/logback.xml
-Dlogging.config=file:$CATALINA_HOME/logback.xml
-Dspring.config.location=file:$CATALINA_HOME/application.properties"

```

3. Запуск системы «БФТ.Бюджетный контроль» и настройка конфигурации

3.1 Вход в систему

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:

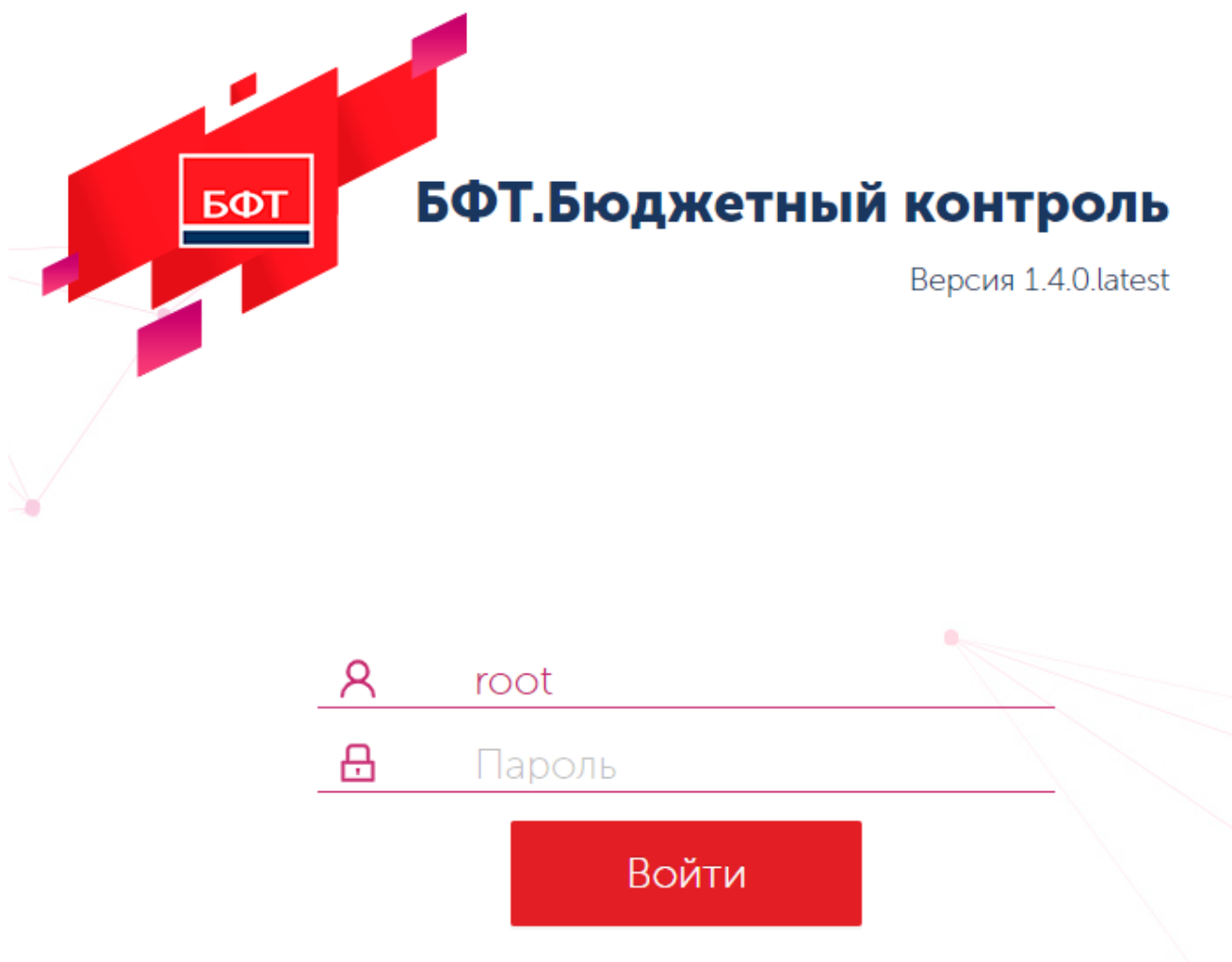


Рисунок 1 – Окно входа в систему

Для входа в систему необходимо указать следующую информацию:

- **Логин:** «root» - имя пользователя.
- **Пароль:** «root» - пароль пользователя.

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.

3.2 Установка конфигурации

Последним этапом настройки приложения является установка конфигурации. После того, как администратор зашел в систему под системной учетной записью следует перейти в пункт меню «Конфигуратор-импорт конфигурации»

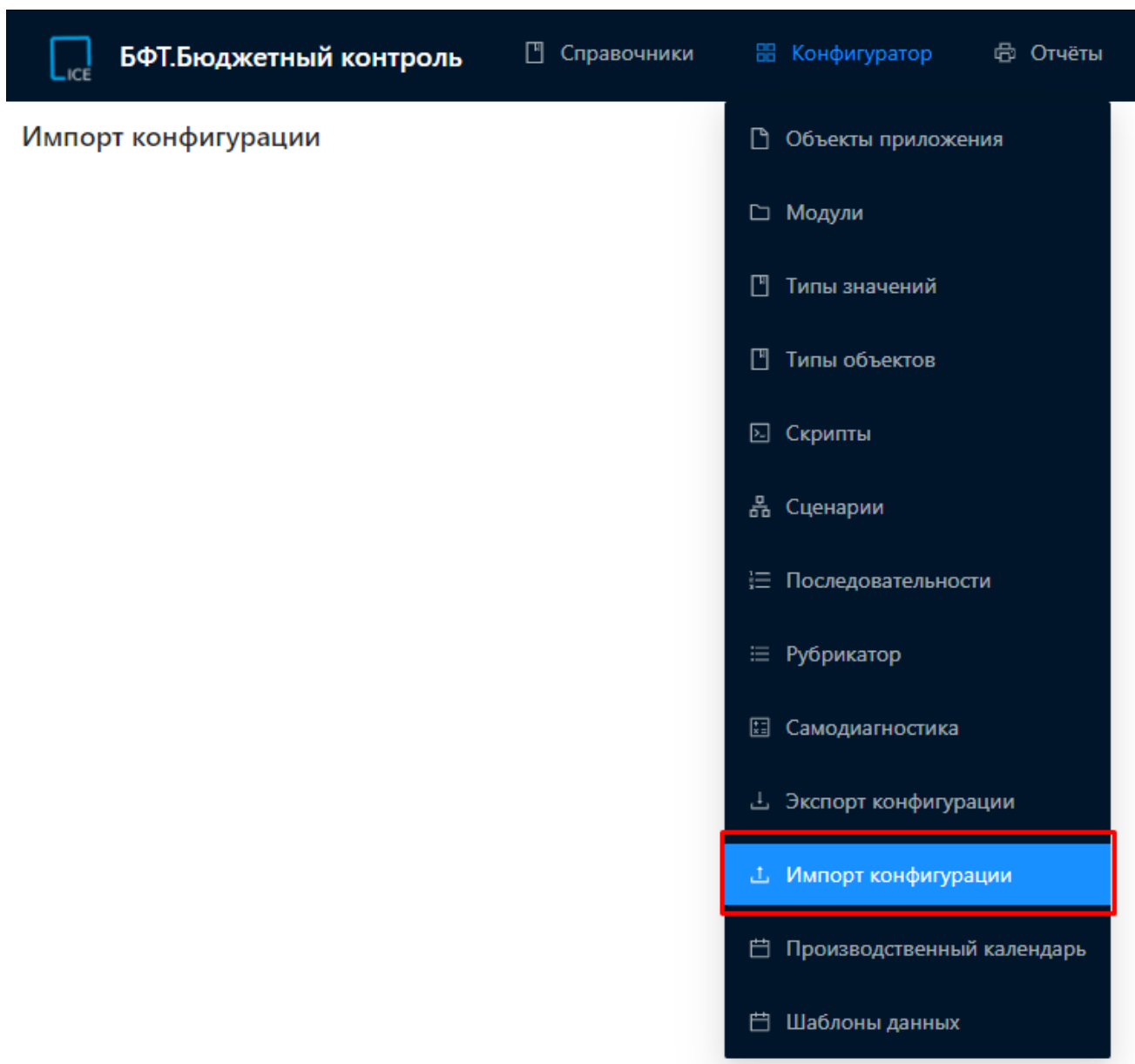


Рисунок 2 – Пункт меню импорт конфигурации

Далее следует выбрать файл конфигурации «azkbk_14_configuration.zip». В открывшейся форме следует выбрать все пункты на вкладке объекты

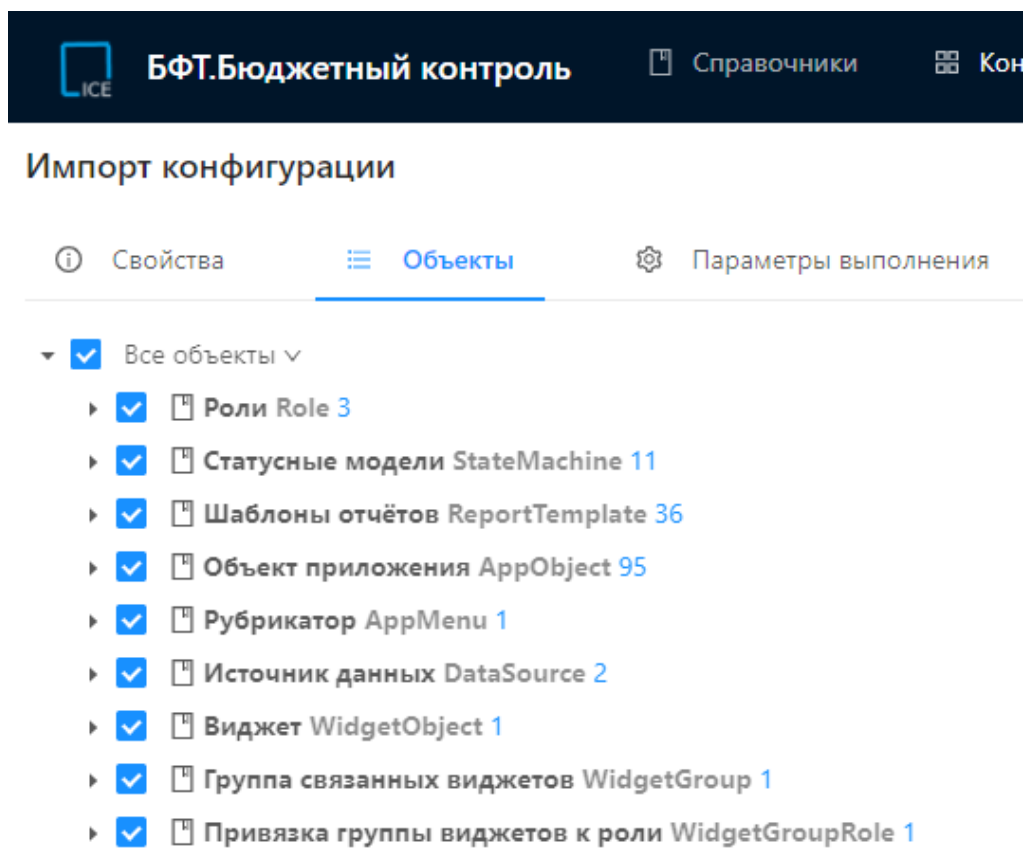


Рисунок 3 – Выбор объектов для импорта конфигурации

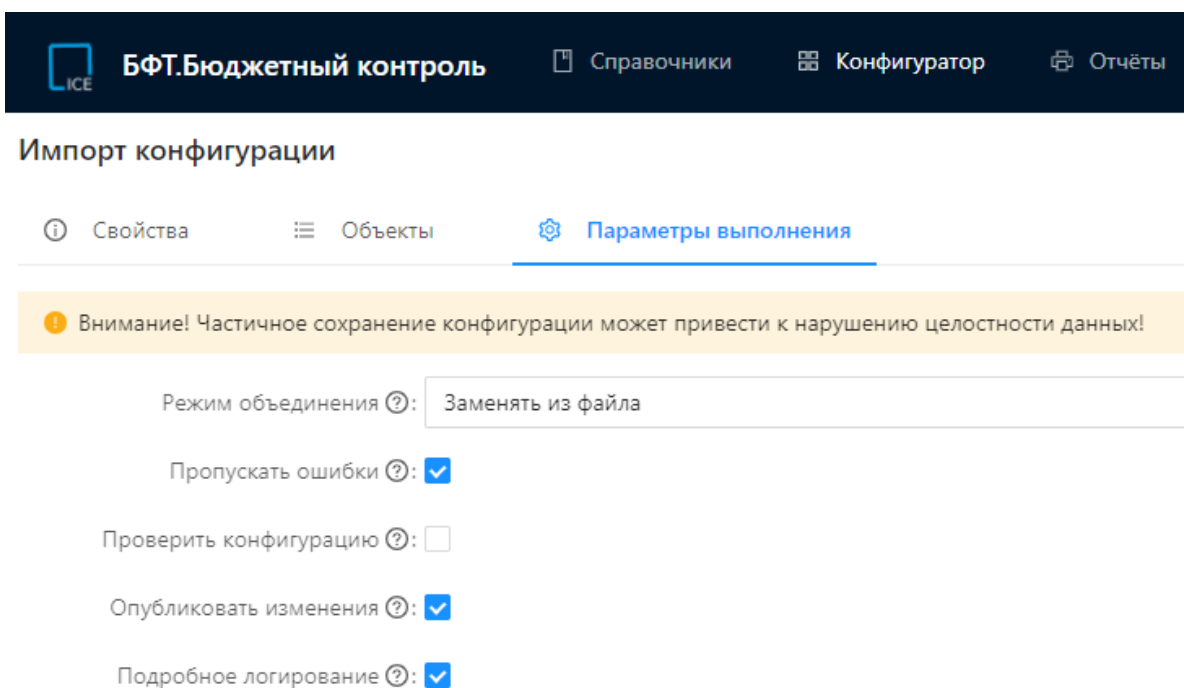


Рисунок 4 – Настройка параметров выполнения импорта конфигурации

На вкладке параметры выполнения следует выполнить настройку согласно рисунку 4.

Когда все настройки произведены следует начать импорт через соответствующую кнопку «Импорт» и дождаться результата исполнения. В случае возникновения ошибок во время выполнения импорта они будут зафиксированы в журнале выполнения. В таком случае следует скачать файл журнала и передать его системному администратору или иному лицу сопровождающему систему

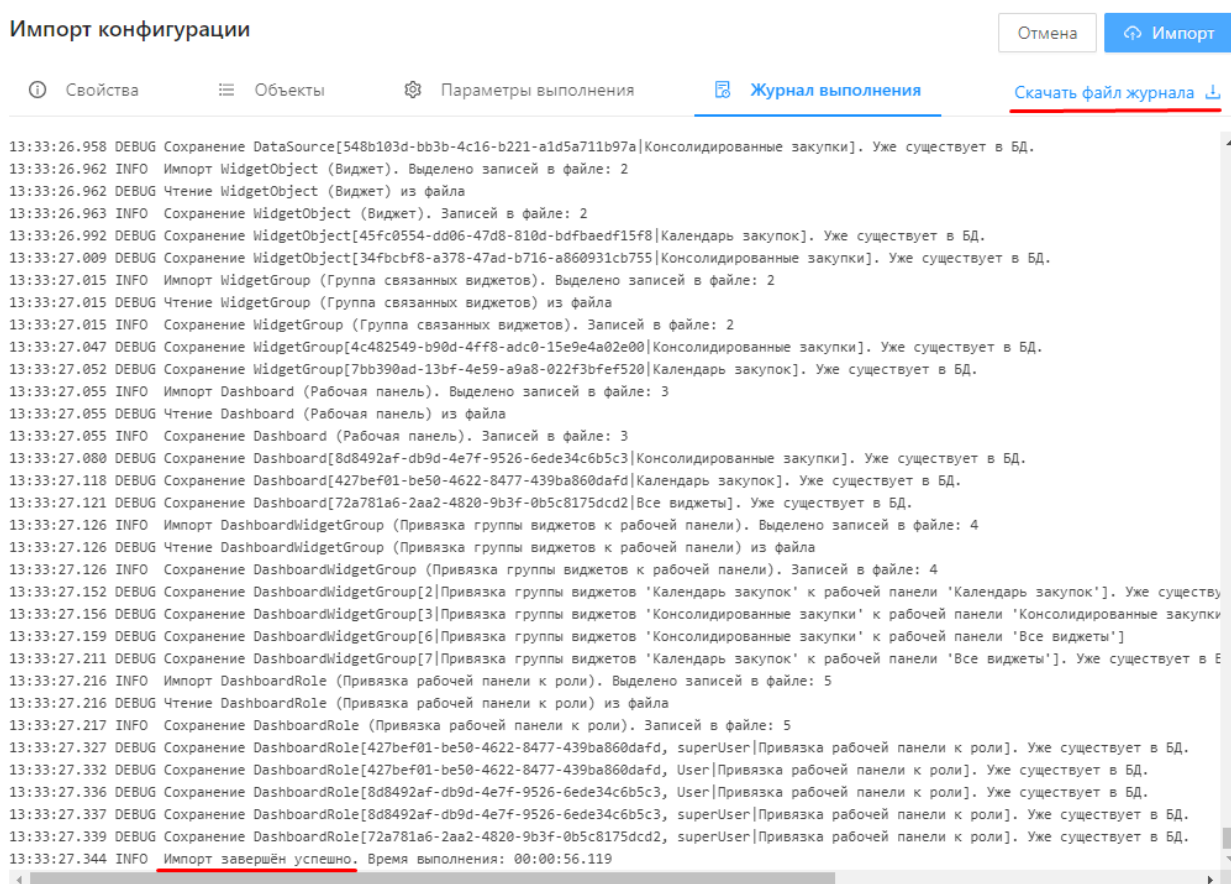


Рисунок 5 – Журнал выполнения импорта конфигурации