

Программное обеспечение "Программный модуль игровой среды взаимодействия с сервером"

Руководство для установки и эксплуатации программного
обеспечения

Оглавление

1. Установка проекта на локальную машину	4
1.1 Установка ядра	4
1.2 Установка проекта хендлеров	5
1.3 Настройка базы данных	5
1.4 Установка Jetty	8
1.5 Установка шины	9
1.6 Установка ActiveMQ	9
1.7 Запуск проекта Servicebus	9
1.8 Тестовая задача	11
1.9 Установка плагина Lombok в IntelliJ IDEA	11

Введение

Данный документ описывает, как запустить проект servicebus на локальном компьютере

Для запуска проекта servicebus на локальной машине требуется выполнить следующие шаги:

1. Установить плагин Lombok для IntelliJ IDEA
2. Клонировать репозитории esb-core, esb-launcher, esb-handlers
3. Настроить локальные базы данных
4. Настроить файлы конфигурации
5. Настроить ActiveMQ
6. Запустить проект
7. Проверить работоспособность

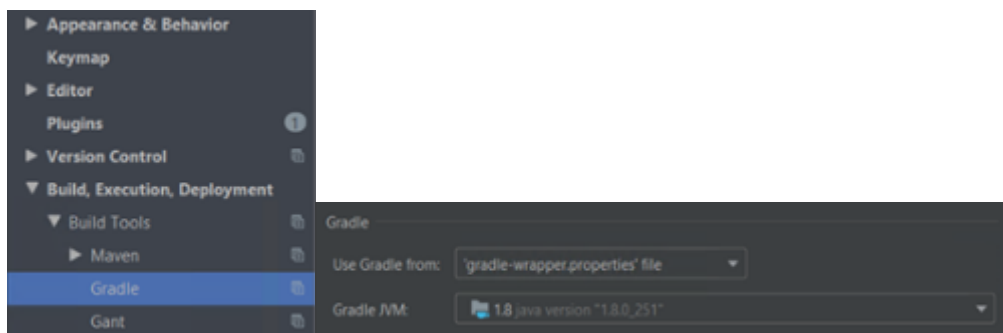
Данные шаги будут рассмотрены далее в руководстве

1. Установка проекта на локальную машину

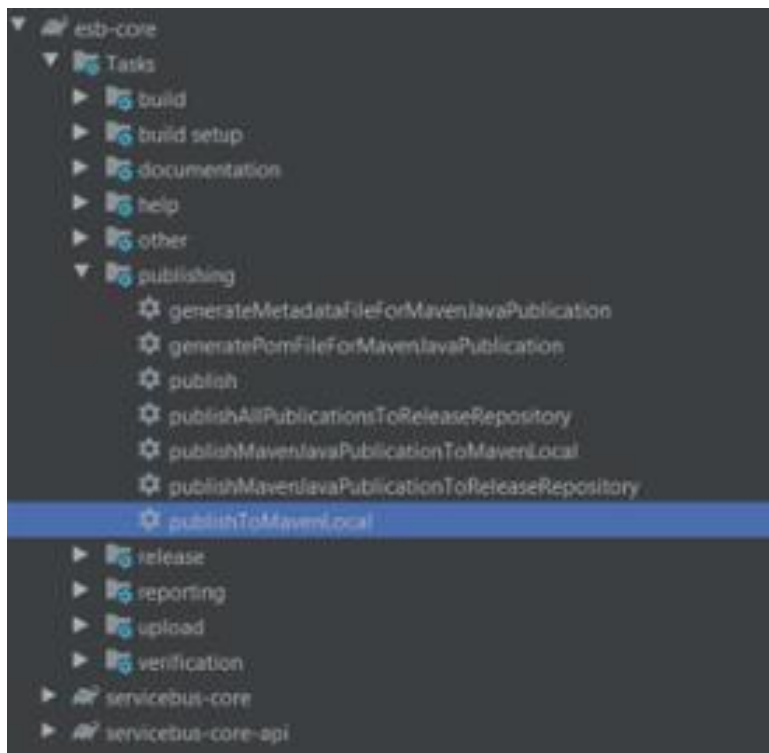
1.1 Установка ядра

1. Выкачать проект esb-core <git@git.tccenter.ru:tc-center/develop/integ-solutions/esb-core.git>
2. Настроить gradle

Для данного шага достаточно использовать wrapper, версия gradle подтягивается из файла gradle/wrapper/gradle-wrapper.properties



3. Вызвать gradle task 'publishMavenJavaPublicationToMavenLocal' (esb-core/Tasks/publishing/publishToMavenLocal) Проверить, что проект запушился в локальный репозиторий (к примеру в %USER_HOME%/.m2/repository/ru/gamble).



1.2 Установка проекта хендлеров

Выкачать проект esb-core [git@git.tccenter.ru:tc-center/develop/integ-solutions/esb-handlers.git](https://git.tccenter.ru:tc-center/develop/integ-solutions/esb-handlers.git)

По аналогии с проектом ядра (пункт 3.1) настроить gradle и вызвать задачу `publishToMavenLocal`

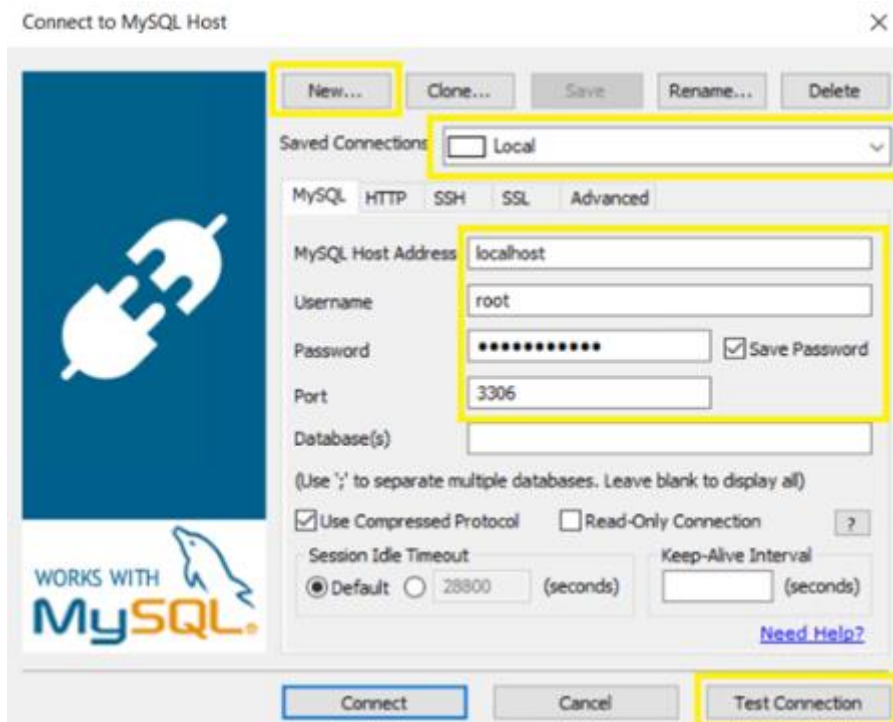
1.3 Настройки базы данных

1. Загрузить и установить MariaDB v.10.4.12. (Лучше уточнить актуальную версию СУБД, т.к. она периодически обновляется на всех стендах). Ввести root/root в качестве логина и пароля.

2. Во время установки оставьте опцию запуска базы данных как службы с установленной отметкой.

3. Загрузить и установить SQLyog (Ссылка на дистрибутив и патч: <https://build.wd.orglot.office//webdav/distr/SQLyog%20Ultimate%20v.11.3.3/>)

4. Создайте 3 новых соединения (File => New Connection [Ctrl+M]):



1. Connection to your local MariaDB database.

MySQL host Address: localhost

Username: root

Password: root

Port: 3306

2. Подключитесь к

MySQL host Address: tfi-idf-shard1.tfi.idf.orglot.office

Username: servicebus_idf_oipo_adm

Password: <enUz<=IF(9D&0*T1m3n

Port: 3306

3. Подключитесь к

MySQL host Address: tfi-idf-shard2.tfi.idf.orglot.office

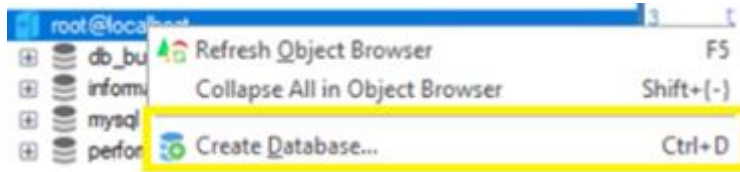
Username: servicebus_idf_oipo_adm

Password: <enUz<=IF(9D&0*T1m3n

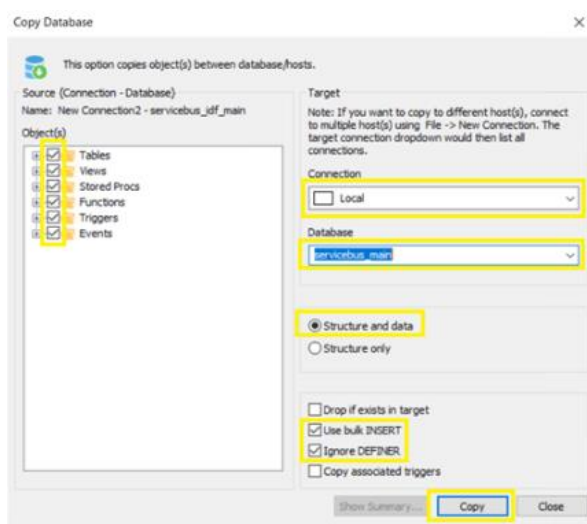
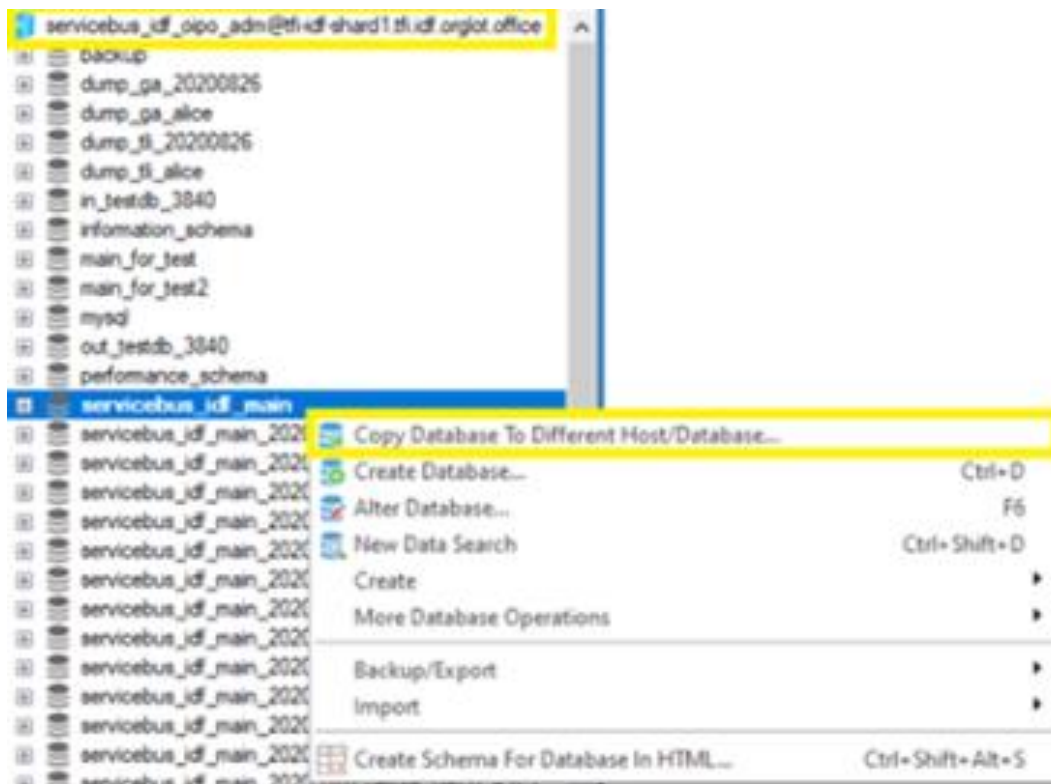
Port: 3306

4. Создайте 2 новых базы данных MariaDB в SQLyog:

- servicebus
- servicebus_main



5. Скопировать обе базы данных (structure and data) с удаленного хоста на локальный выбрав "Copy Database to Different Host/Database" из контекстного меню



- Скопировать "servicebus_idf_main" из "tfi-idf-shard1.tfi.idf.orglot.office" на localhost "servicebus_main".
- Скопировать "servicebus_idf" из "tfi-idf-shard2.tfi.idf.orglot.office" на localhost "servicebus_gamebet_main".

Отредактируйте таблицу «share» в базе данных «servicebus_main», чтобы она соответствовала локальной конфигурации шарда и основной базы данных.

Изначально таблица «share» содержит все сегменты, которые были предварительно настроены в базе данных, из которой вы скопировали свои данные.

На локальной машине нам нужен только один шард и одна основная база данных. Поэтому удалите все записи с «псевдонимом» КРОМЕ («shard_1», «shard_main»).

Измените значения url записей shard_1 и shard_main, чтобы они указывали на ваши локальные базы данных maria, например:

Для shard_1

```
jdbc:mariadb://localhost:3306/servicebus?noAccessToProcedureBodies=true
```

Для shard_main

```
jdbc:mariadb://localhost:3306/servicebus_main?noAccessToProcedureBodies=true
```

Вставьте значение 'root' в колонку 'login'.

Вставьте «lDgJ1+GABVo=» как значение столбца «пароль» для обеих записей. Это значение представляет собой представление «root» в кодировке MD5. Если вы выбрали другой пароль для своей базы данных, сгенерируйте MD5 своего пароля, используя тестовый класс servicebus-core\src\test\java\ru\gamble\servicebus\core\service\CryptoServiceImplTest.java - часть модуля Servicebus Core . Поместите значение вашего пароля в метод cryptoTest и запустите тест.

Не используйте другие инструменты для шифрования пароля, так как вы можете получить другие результаты, и сервисная шина не сможет организовать соединение с базой данных.

1.4 Установка Jetty

1. Загрузить последнюю версию Jetty release и распаковать в произвольную директорию.

2. Установить переменную среды JETTY_HOME равную пути папки BIN, извлеченного Jetty.

1.5 Установка модуля

1. Выкачать `git@git.tccenter.ru:tc-center/develop/integ-solutions/esb-launcher.git` и переключиться на master ветку в IntelliJ IDEA
2. Изменить `esb-idf\profiles\local\servicebus.properties`:

Скопируйте `esb-idf\profiles\local\servicebus.properties.default` в ту же директорию с именем `servicebus.properties`.

В файле `properties`:

Измените URL-адрес для подключения к базе данных `servicebus` на `localhost` с соответствующими учетными данными.

```
jdbc.servicebus.url = jdbc:mariadb://localhost:3306/servicebus_main
```

```
jdbc.servicebus.user = root
```

```
jdbc.servicebus.pwd = root
```

Важно: Имя базы должно совпадать с именем локальной базы MariaDB.

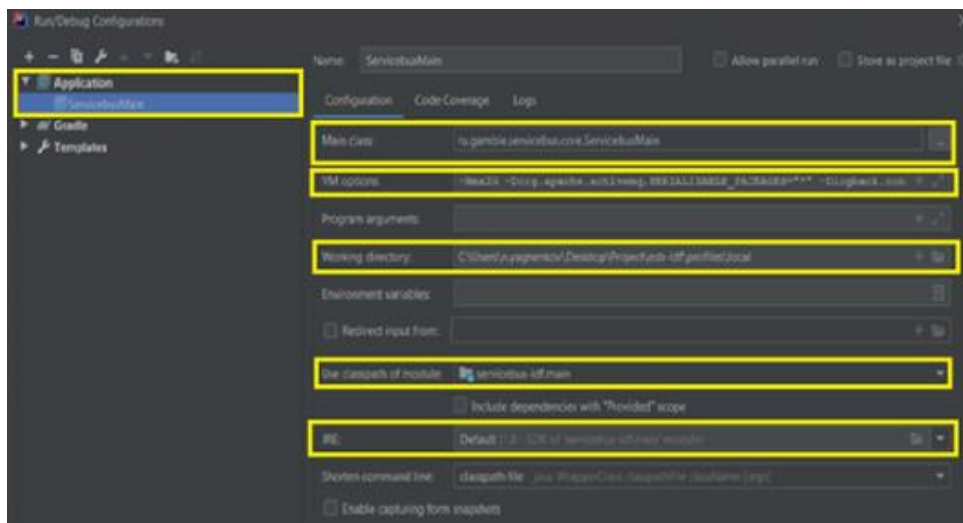
1. Измените свойство `'queue.url'`, чтобы использовать `localhost` вместо `0.0.0.0`
`queue.url= failover://(nio://localhost:61616)?randomize=false&maxReconnectDelay=1000&timeout=1000&jms.prefetchPolicy.queuePrefetch=1`
2. Выполните сборку `gradle`

1.6 Запуск ActiveMQ

В файле `esb-idf/src/main/resources/META-INF/spring/camelContext.xml` раскомментировать строку: `<import resources="file:activemq.xml">` (Важно: Не коммитить в репозиторий!!!)

1.7 Запуск проекта Servicebus

Создать новую Run/Debug конфигурацию в IntelliJ IDEA для проекта `servicebus` проекта, окно конфигурации должно выглядеть следующим образом:



MainClass: gamble.servicebus.core.ServicebusMain

VM options: -Xmx2G -

Dorg.apache.activemq.SERIALIZABLE_PACKAGES="*" -

Dlogback.configurationFile=./logback.xml -

Djavax.xml.parsers.DocumentBuilderFactory=com.sun.org.apache.xerces.internal.

jaxp.DocumentBuilderFactoryImpl -

Dorg.eclipse.jetty.util.log.class=org.eclipse.jetty.util.log.StdErrLog

Working Directory: [YOUR_PATH_TO_THE_PROJECT]\esb-idf\profiles\local

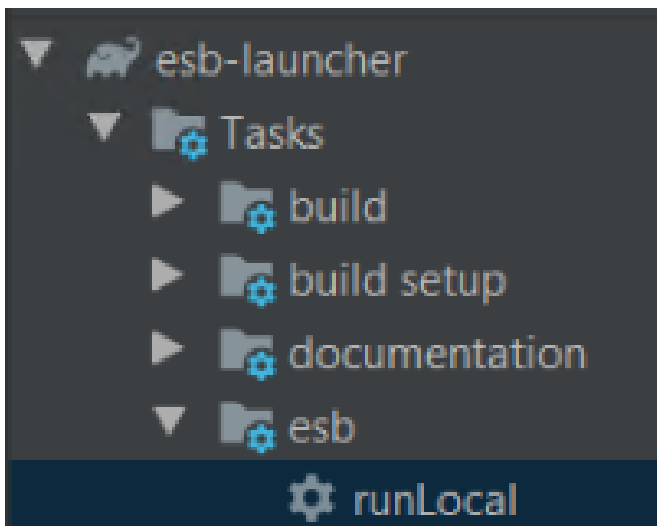
Use classpath of module: servicebus-idf.main

JRE: 1.8

Before launch: Build

Примените конфигурацию и запустите проект. Подождите около минуты, пока запустится шина.

Также запуск модуля можно осуществить через gradle task: runLocal



1.8 Тестовая задача

Теперь проект должен быть запущен на порту `http://localhost:8663`. Для тестирования можно запустить любой REST клиент и отправить запрос. Ниже представлен пример запроса из Postman:

HTTP Method: POST

Host/port: `http://localhost:8663`

Path: `/system/create_task`

Content-Type: `application/json`

Request Body => Text: `{"tasktype":"FAIL_ENDPOINT"}`

Пример ответа:

```
{"taskId":"3068202501"}
```

1.9 Установка плагина Lombok в IntelliJ IDEA

Для удобства рекомендуется установить плагин IntelliJ Lombok.

1. Открыть окно настроек IntelliJ IDEA.
2. Выбрать пункт Plugins.
3. Найти Lombok plugin, в окно поиска ввести "Lombok".
4. Кликнуть на кнопку
5. Нажать кнопку

