**Тестовое задание**

Реализовать два приложения для отправки и получения сообщений пользователем. Первое приложение – отправляет сообщения пользователя в очередь. Второе – читает сообщение из очереди.

Обобщенный функционал приложений:

1. Отправляет сообщение в очередь
2. Принимает сообщения из очереди
3. Сохраняет в базу данных Дату с временем отправки сообщения и текст отправленного сообщения
4. Сохраняет в базу данных Дату с временем получения сообщения и текст полученного сообщения
5. База SqLite для приложения отправки и получения - разные

Использовать:

1. Среду разработки Delphi
2. RabbitMQ – брокер сообщений
3. SQLite – база данных

Условия приемки:

1. Рабочее приложение
2. Исходный код приложения
3. Инструкцию/скрипты настройки работы приложения.

**Выполнение.**

Инструкция по запуску приложения.

1. Установка RabitMQ

Вариант 1.

* 1. Установить Erlang <https://www.erlang.org/downloads>
  2. Установить RabiMQ <https://www.erlang.org/downloads>

Вариант 2

Установка chocolatey

[Chocolatey Software | Installing Chocolatey](https://chocolatey.org/install)

В PowerShell > *Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))*

После в cmd > *choco install rabbitmq*

Вариант 3

1. Установить Docker Desktop <https://www.docker.com/>
2. Запустить образ rabbitmq командой

“docker run --name rabbit1 -p 15672:15672 -p 5672:5672 rabbitmq:3.8.9-management”

, где

--name rabbit1 - присваивает запускаемому контейнеру имя rabbit1

-p 15672:15672 - пробрасывает порт с хоста в контейнер. 15672- порт на хосте, 15672 - порт в контейнере

rabbitmq:3.8.9-management - имя образа и его тег/версия, разделенные двоеточием

Вариант 3 не подошел, нужен образ с плагином stomp.

1. Установка модулей RabbitMQ

Из каталога с установленным rabbitmq (или в пуске запустить «RabbitMQ Command Prompt (sbin dir)») выполнить команды:

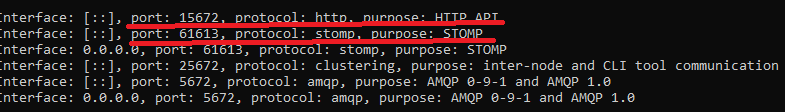
Установка STOMP > *rabbitmq-plugins enable rabbitmq\_stomp*

Установка Management> rabbitmq-plugins enable rabbitmq\_mangement

Панель management доступна по ссылке <http://localhost:15672/#/>

Проверка: выполняем команду> rabbitmqctl status

В выводе консоли должны увидеть протоколы http и stomp



1. Установка Delphi. Использована версия Delphi 10.4 Community Edition

Проблема в том, что для Delphi нет бесплатного клиента rabbitmq, либо я его не нашел, но есть решение с использование протокола [STOMP](https://stomp.github.io/) (The Simple Text Oriented Messaging Protocol)

Установить клиент STOMP для Delphi

[danieleteti/delphistompclient: STOMP client for Embarcadero Delphi and FreePascal. (github.com)](https://github.com/danieleteti/delphistompclient)

Если сокращенно, то нужно просто к приложению подключить файл 'StompClient.pas'.

uses

StompClient in 'StompClient.pas';

Дополнительно описание

[Add STOMP to your apps for lightweight real-time streaming (embarcadero.com)](https://blogs.embarcadero.com/add-stomp-to-your-apps-for-lightweight-real-time-streaming/)

Пришлось проверить другие варианты, которые не работают

[How and why I built rabbitmq client for delphi (delphi-developers-archive.blogspot.com)](https://delphi-developers-archive.blogspot.com/2018/07/how-and-why-i-built-rabbitmq-client-for.html)

<Https://github.com/lgadina/comotobo>  
<Https://github.com/lgadina/rabbitmq-library>

[HeZiHang/Delphi-RabbitMQ: RabbitMQ driver for Delphi (github.com)](https://github.com/HeZiHang/Delphi-RabbitMQ/)

1. SQLLite

SQLLite устанавливать не нужно, он поддерживается в Delphi

Для просмотра БД можно использовать SQLite Database Browser <http://sqlitebrowser.org/>

1. Запуск приложения.

В каталоге .\Win32\Debug\

Запускаем приложение Producer.exe (издатель), нажимаем кнопку Enter.

Приложение создает очередь.

Запускаем приложение Consumer.exe (подписчик), нажимаем кнопку Enter. Приложение подписывается на событие OnMessage очереди.

Теперь можно писать сообщения в приложение Producer, они будут поступать в приложение Consumer.

Вся история сообщений сохраняется в базах данных DBProducer.db и DBConsumer.db

После создания очереди, порядок запуска приложений неважен.