Добавляем БД PostgreSQL к RESTful сервису на Spring Boot.

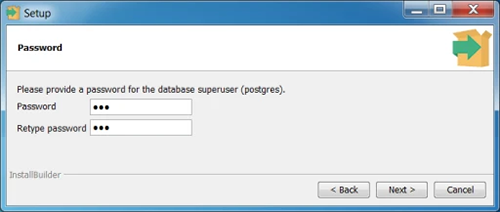
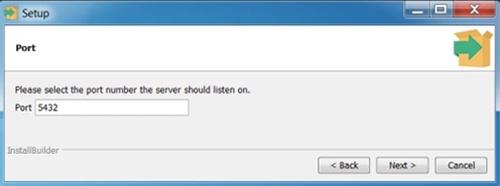
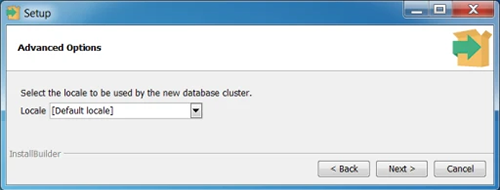
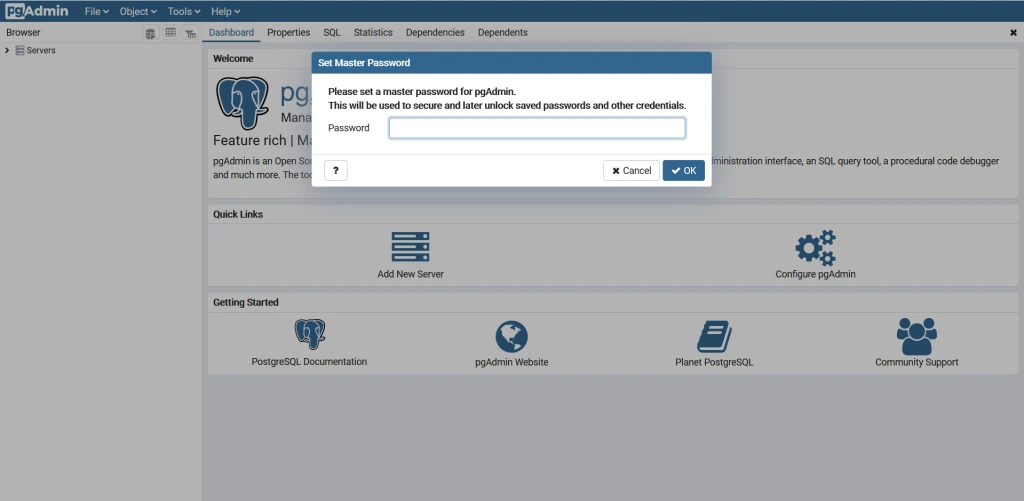
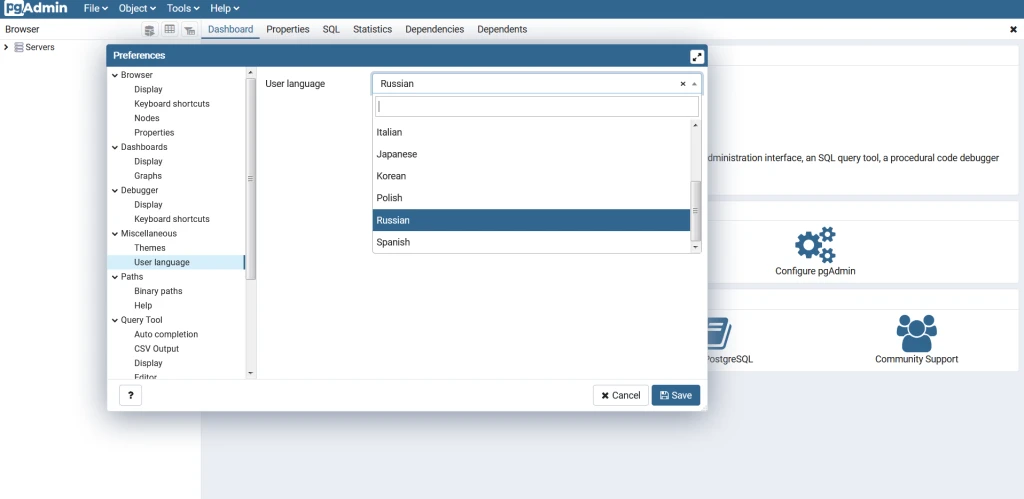
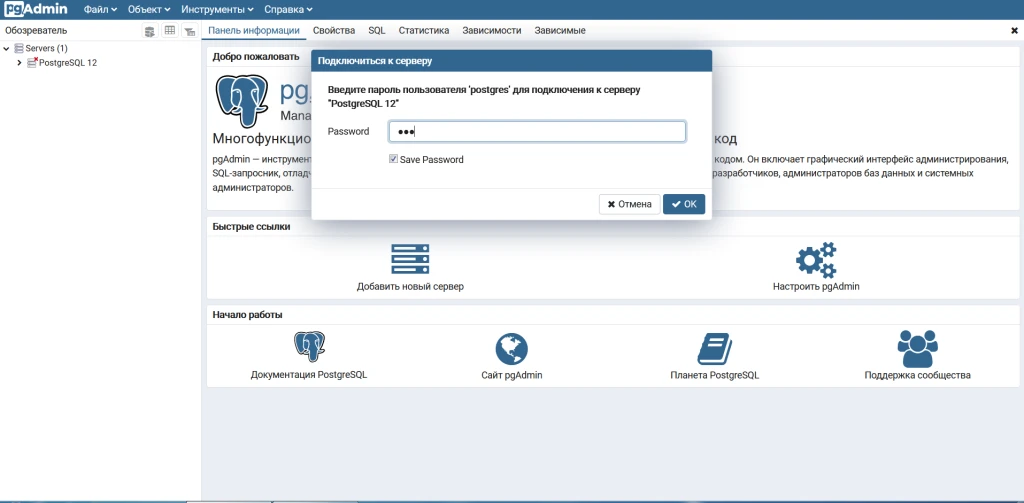
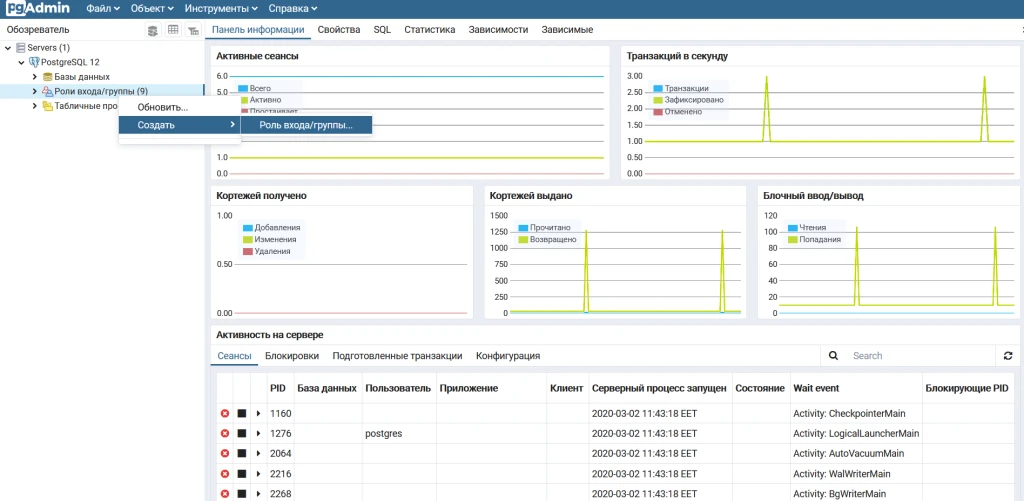
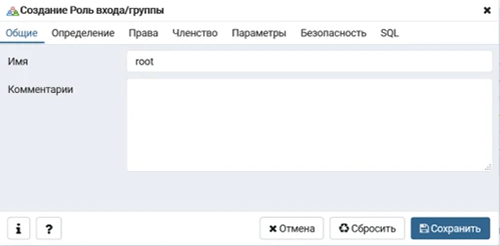
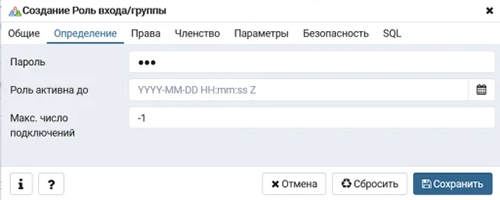
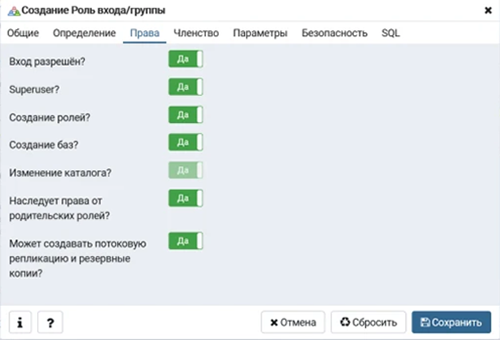
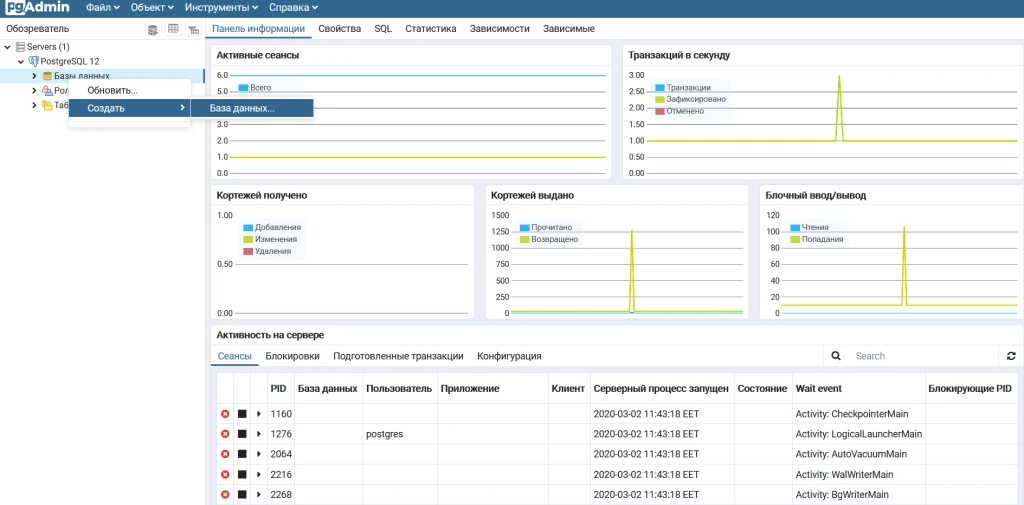
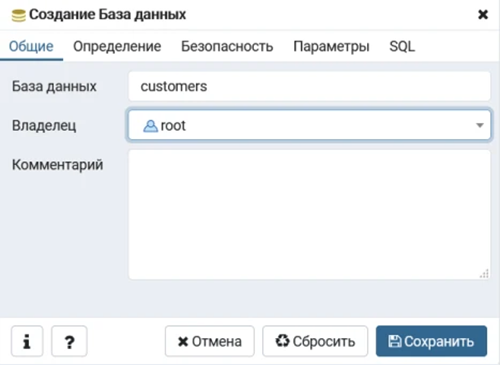
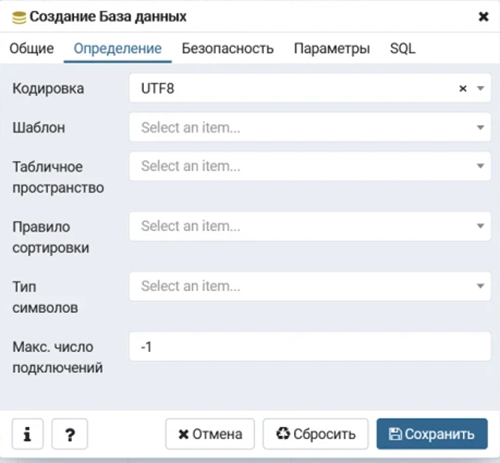
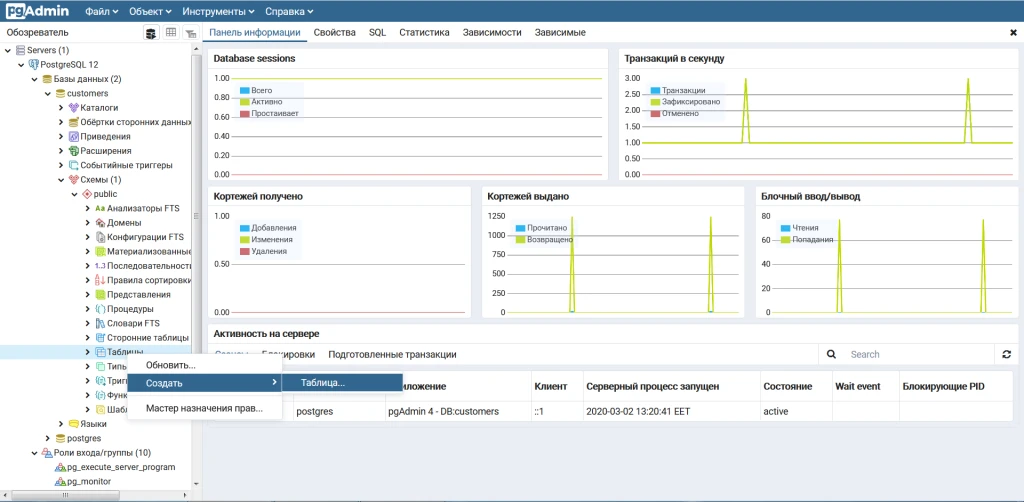
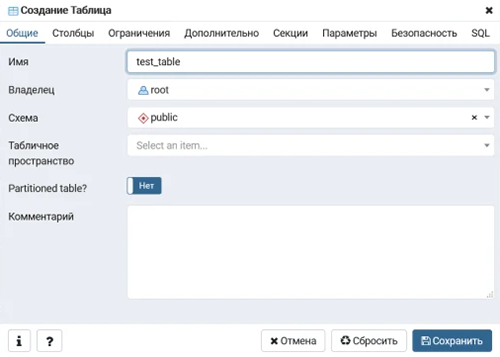
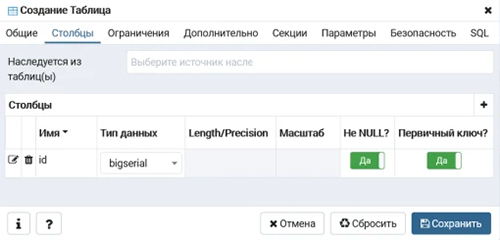
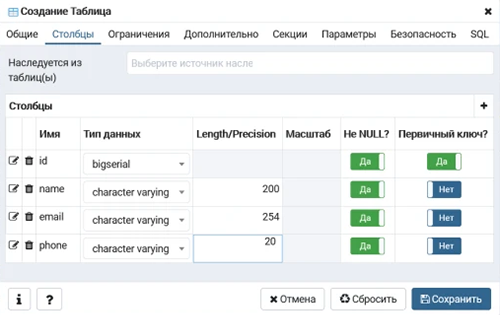
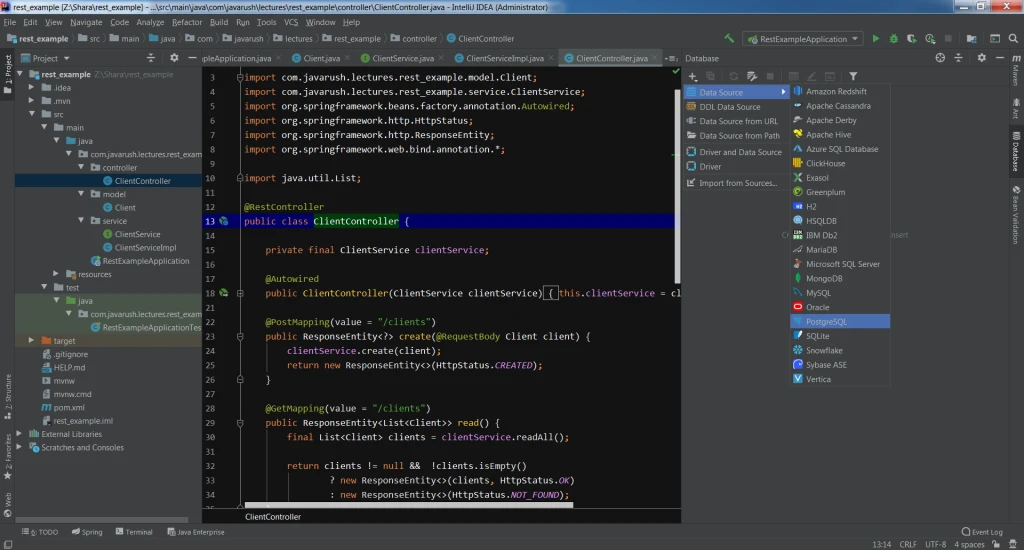
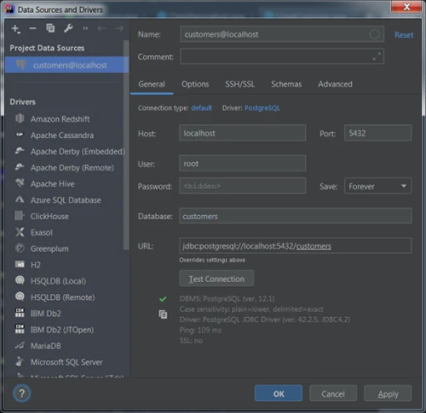
Создание базы данных в PostgreSQL отличается от MySQL, тем, что в MySQL мы создаем новую схему *(и ее название прописываем в подключение -* *spring.datasource.url=jdbc:mysql://localhost:3306/****forflyway****?serverTimezone=UTC*).

А в PostgreSQL мы создаем новую базу данных (*и ее название прописываем в подключение - spring.datasource.url=jdbc:postgresql://localhost:5432/****forflyway****?serverTimezone=UTC*).

Кроме того при подключении отличаются номера портов и названий БД.

Также нужно изменить зависимость в файле “pom.xml”.

И нужно дописать/изменить еще часть строк в файле “application.properties” (см. в самом конце).

Я решил использовать PostgreSQL, так как мне понравился дружественный интерфейс ее админки. Для начала нам нужно скачать установщик PostgreSQL, например отсюда: [PostgreSQL Database Download](https://www.enterprisedb.com/downloads/postgres-postgresql-downloads) Полный процесс установки я описывать не буду, т.к. он может отличаться в зависимости от вашей ОС. Однако отмечу, что в процессе вам потребуется придумать пароль администратора для дефолтной роли (учетной записи) postgres и ввести его 2 раза. Для простоты я установил пароль 123. Конечно, в реальных проектах необходимо придумать что-то посложнее. Далее программа установки предлагает выбрать порт, я оставил его значение по умолчанию. Locale тоже оставил по умолчанию. Установили, дальше открываем интерфейс администратора pgAdmin. Oткроется браузер со всплывающим окошком, где нас попросят ввести созданный ранее пароль. Теперь нам нужно установить язык интерфейса. Для этого нажимаем File --> Preferences --> Miscellaneous --> User Language, выбираем нужный язык --> Save, и перезагружаем страницу браузера. В левом верхнем углу страницы нажимаем Server, появляется окошко «Подключиться к серверу». Вводим наш пароль еще раз и ставим галку Save Password, чтобы не вводить его каждый раз. Можно было бы и далее использовать роль администратора для всех баз данных, но лучше создадим новую роль, ведь у нас может быть много баз данных и много программ, которые их используют. Для этого в левом верхнем углу жмем на PostgreSQL 12 --> ПКМ на Роли входа/группы --> Создать --> Роль входа/группы Во всплывающем окне на вкладке «Общие» вписываем имя роли. Я назвал роль root. На вкладке «Определение» создаем новый пароль, я оставил 123, просто чтобы не запутаться. Переходим на вкладку «Права» и отмечаем все необходимые пункты. Я установил все пункты «ДА». Нажимаем «Сохранить» Переходим к созданию базы данных. Жмем ПКМ на «Базы данных» --> Создать --> База данных На вкладке «Общие» создаем название базы данных. Пусть это будет, например, customers. Владельцем назначаем root, которого мы создали на предыдущем этапе. На вкладке «Определение» проверяем, что у нас установлена кодировка UTF8. Нажимаем «Сохранить». Всё, наша база данных создана (пока что пустая). На этом можно было бы закончить с pgAdmin, т.к. таблицы мы будем создавать программно, но, на всякий случай, покажу как создать таблицу вручную. Разворачиваем дерево customers --> Схемы --> public. Жмем ПКМ Таблицы --> Создать --> Таблица. Откроется всплывающее окно. На вкладке «Общие» назначаем имя нашей таблице, например test\_table, владельцем назначаем root. Переходим на вкладку «Столбцы», нажимаем на "+" для создания нового столбца. Вводим имя “id” и тип данных bigserial, что эквивалентно типу Long в Java, но с авто-инкрементом (при добавлении новой записи, id будет автоматически увеличиваться на единицу). Не NULL отмечаем как «Да», первичный ключ тоже «Да». Создаем таким же образом столбцы ”name”, “email” и ”phone”. Тип данных выбираем character varying, это соответствует типу String, но позволяет задать максимальную длину. Задаем максимальную длину имени в 200 символов для заполнения Ф.И.О. в одном столбце. Задаем максимальную длину email в 254 символа. Почему у email такая максимальная длина можно узнать [здесь](https://stackoverflow.com/questions/386294/what-is-the-maximum-length-of-a-valid-email-address). Для телефонного номера выделяем 20 символов, этого должно хватить. Немного о телефонных номерах: [Заблуждения программистов о телефонных номерах](https://habr.com/ru/post/279751/) (Хабр) Не NULL во всех столбцах отмечаем как «Да», если хотим, чтобы эти данные были обязательными. Нажимаем «Сохранить». Всё, таблица создана. Для удаления нашей тестовой таблицы нажмите ПКМ по ней (в дереве) и «удалить», т.к. она нам больше не нужна, ведь мы будем создавать таблицы из нашей программы. Закрываем pgAdmin, и переходим к нашей программе. Открываем IDEA с нашим проектом. Нажимаем Database в правом столбце интерфейса, нажимаем на "+" для добавления нашей БД. Далее Data Source --> PostgreSQL. Во всплывающем окне вписываем в поле User роль root, которую мы создали ранее, и наш пароль 123 в поле password. В поле Database пишем название нашей БД customers. Нажимаем кнопку Test Connection, и если видим зеленую галку под ней, то все в порядке, и нажимаем кнопку OK. 

Всё, к БД подключились, теперь переходим к файлу pom.xml и добавим зависимости. Для работы с БД ORM:

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

Для создания REST контроллера:

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-rest</artifactId>

</dependency>

Для сервера Tomcat:

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-tomcat</artifactId>

</dependency>

Для PostgreSQL:

<dependency>

<groupId>org.postgresql</groupId>

<artifactId>postgresql</artifactId>

<version>42.2.10</version>

</dependency>

Разобрались с pom.xml, перейдем в папку resources, и заполним файл application.properties следующим образом:

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

spring.datasource.username=root

spring.datasource.password=123

spring.datasource.driver-**class**-name=org.postgresql.Driver

spring.jpa.database=postgresql

spring.jpa.database-platform=org.hibernate.dialect.PostgreSQL10Dialect