

# Task330. Модуляризация приложения с базой Cassandra

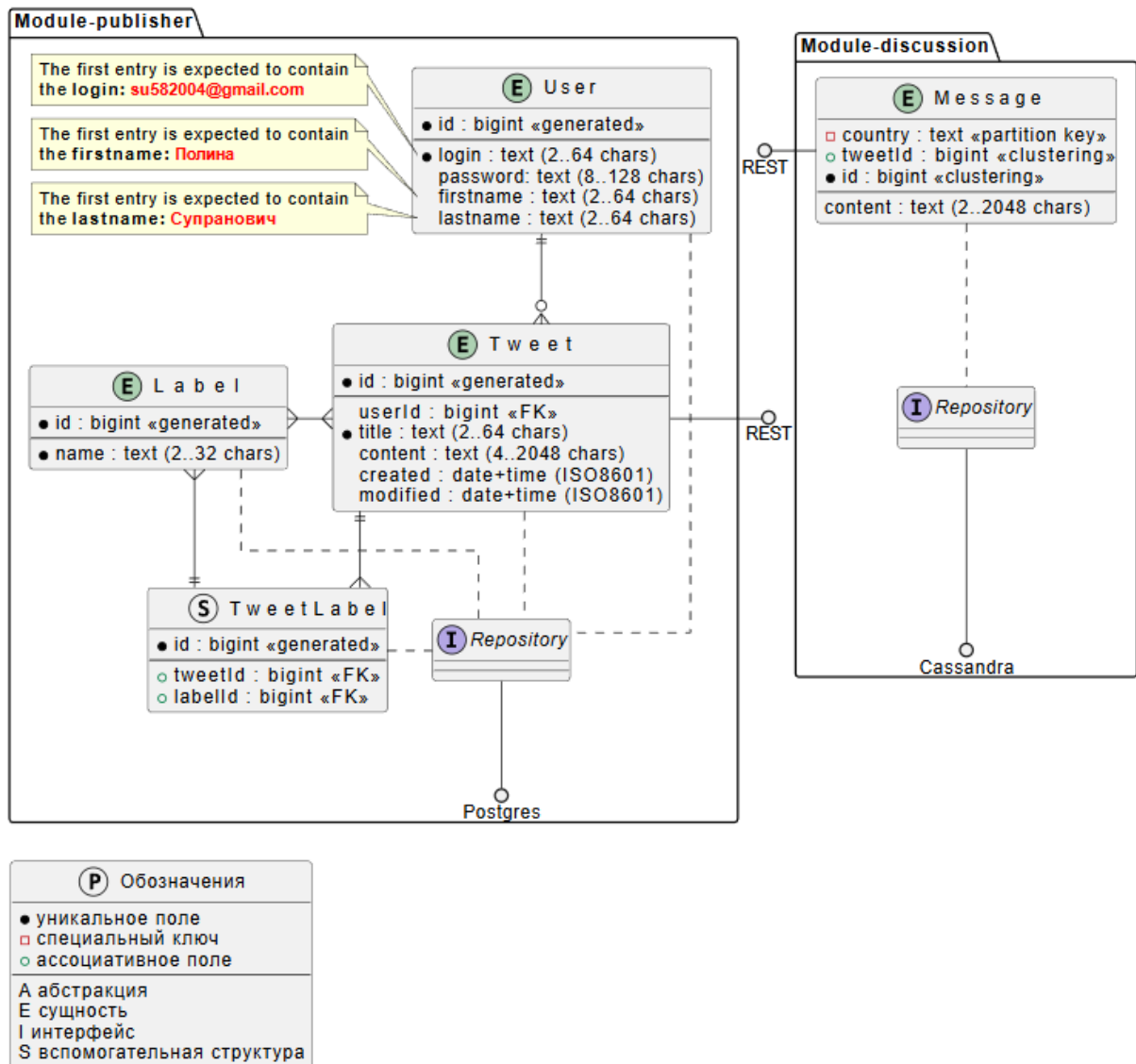
## Дано:

- Разрабатываемая система обрабатывает сущности **User**, **Tweet**, **Label** и **Message**, которые логически связаны отношениями
  - один-ко-многим (**User** и **Tweet**, **Tweet** и **Message**)
  - многие-ко-многим (**Tweet**, **Label**).
- Необходимо выполнить перенос хранения сущности **Message** из реляционной базы данных **Postgres** в новый модуль/микросервис с другой базой данных, а именно **Cassandra** или **MongoDB**

## Задание

В ранее разработанном приложении необходимо выделить обсуждения на основе сущности **Message** в отдельный модуль **discussion**. Для корректного распределения информации в нодах базы данных следует изменить структуру сущности **Message** согласно схеме. Вместо модуля допускается разработка отдельного микросервиса **discussion**.

Основой для выполнения задания является следующая схема приложения



## Технические требования

- Используйте префикс **/api/v1.0/** для контроллеров REST и их методов,
- Используйте адрес и порт **localhost:24110** для самого приложения (согласно схеме это модуль **publisher**).
- Используйте адрес и порт **localhost:24130** для реализуемого модуля **discussion**.
- Используйте следующие данные при подключении к **Cassandra**
  - Префикс драйвера **jdbc:cassandra:**
  - Адрес базы данных **localhost**
  - Порт базы данных **9042**
  - user не нужен
  - password не нужен

- Схема **distcomp**
- Используйте обязательный префикс **tbl\_** для таблиц(ы) в базе данных

## Рекомендации

### Настройка проекта (рекомендуется Spring)

1. Если это еще не сделано преобразуйте проект в многомодульный. Структура произвольная, но минимально должно быть два модуля: **publisher** и **discussion**
2. Далее нужно добавить зависимости в **discussion** для работы с Cassandra (для maven это файл pom.xml)
3. Создать конфигурацию **discussion** для подключения к базе Cassandra

### Разработка модуля **discussion**

1. Создать класс(ы) модели данных
2. Создать репозитории для взаимодействия с базой Cassandra
3. Настроить маппинг сущностей на таблицы в базе данных
4. Реализовать бизнес-логику сервисов **discussion**, включая DTO-преобразования

### Создание REST контроллеров

1. Реализовать поддержку REST для модуля **discussion**
  - Создать контроллеры для обработки HTTP запросов через вспомогательный REST-адрес **localhost:24130**
  - Настроить маршрутизацию запросов к соответствующим методам контроллеров
2. Использовать данное подключение REST в модуле **publisher** (для Spring рекомендуется WebClient или RestClient)

### Запуск и тестирование модуля **discussion**

1. С помощью **liquibase** выполнить инициализацию базы данных Cassandra, схема - **distcomp**. Это может быть не просто (гугл в помощь).
2. Выполнить тестовое покрытие модуля/микросервиса **discussion** (рекомендуется **jacoco** для измерения уровня покрытия, желательно получить 80% и более)
3. Из среды разработки или системы сборки проверить работу модульных тестов.
4. Проверить интеграционными тестами работоспособность модуля через вспомогательный REST-адрес **localhost:24130** модуля **discussion**