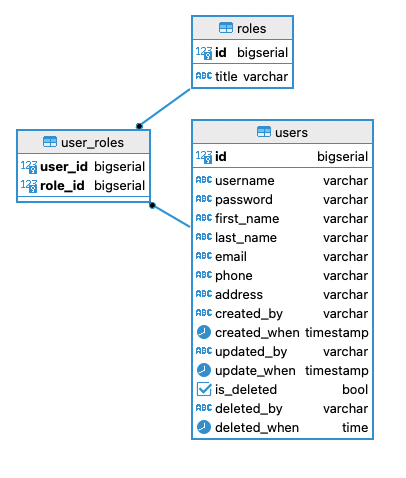
**Контракты**

**Список микросервисов:**

* gateway-service
  + Описание: гейтвей
  + Порт: 5555
  + Context-path: /gateway
  + Mapping:
    - /auth -> /auth-service
    - /cabinet -> /cabinet-order-service
    - /cart -> /cart-service
    - /catalog -> /catalog-service
* authorization-service
  + Описание: авторизация и работа с персональными данными
  + Порт: 8187
  + Context-path: /auth-service
  + Эндпойнты:
    - api/v1/auth - AuthController
      * /authenticate - авторизация по логину и паролю
        + @RequestBody JwtRequest authRequest
    - api/v1/reg - RegisterController
      * /register - регистрация пользователя
        + @RequestBody RegisterUserDto
        + если имя/фамилия/адрес/телефон пустой, нужно заполнить параметр на фронте как пустая строка “”
    - /api/v1/users - UserController
      * /delete - удаление пользователя
        + @RequestParam Long deleteUserId - кого удаляем
        + @RequestHeader String userId - кто удаляет
      * /all - вывод всех пользователей, включая удаленных
      * /all\_not\_deleted - вывод всех живых пользователей
      * /fullname\_by\_id - получение фамилии и имени пользователя по id
        + @RequestParam Long userId
      * /name\_and\_email\_by\_id - получение имени и емэйла пользователя по id
        + @RequestParam Long id
    - /api/v1/roles - RoleController
      * /update - изменение ролей
        + @RequestParam Long changeUserId - чью роль меняем
        + @RequestParam String role - новая роль
        + @RequestHeader String userId - кто меняет роль
  + В БД заведены для теста дефолтные пользователи:
    - Админ - username=”admin”, password=”admin”
    - Пользователь - username=”customer”, password=”customer”
    - Менеджер - username=”manager”, password=”manager”
  + Swagger localhost:8187/auth-service/swagger-ui/index.html
  + БД URL: jdbc:postgresql://194.87.101.43:5432/users\_roles\_db

user: security

password: SequrityVideoteka2023



* catalog-service
  + Описание: обслуживание каталога фильмов
  + Порт: 8189
  + Context-path: /catalog-service
  + Swagger http://localhost:8189/catalog-service/swagger-ui/index.html
  + Эндпойнты:

/api/v1/country/all - получение списка стран для фильтра

/api/v1/country/id - получение страны по id

/api/v1/country/new - добавление новой страны в каталог

/api/v1/director/all - получение списка режиссеров для фильтра

/api/v1/director/id - получение директора по id

/api/v1/director/new - добавление новой страны в каталог

/api/v1/all-with-filter - получение списка фильмов по набору фильтров

/api/v1/all-with-filter-dto - получение списка фильмов по набору фильтров в виде PageDto

/api/v1/film/title\_part - получение списка фильмов по части названия

/api/v1/film/id - получение фильма по id

/api/v1/film/min\_max\_year - получение min и max годов в списке фильмов

/api/v1/film/new-film - добавление фильма в каталог

/api/v1/film/all - полный список фильмов без фильтрации и пагинации

/api/v1/film/movie-change - изменение данных фильма

/api/v1/genre/all - получение списка жанров для фильтра

/api/v1/genre/id - получение жанра по id

/api/v1/price/prices\_filter - цены для фильтра

/api/v1/rating/all - вывод всех оценок и комментариев к фильмам

/api/v1/rating/new-comment - добавление новой оценки и комментария к фильму

/api/v1/rating/total\_film\_rating - Вывод средней оценки по всем пользователям по id фильма

/api/v1/rating/all\_grade\_and\_review\_by\_filmId - вывод отзывов по id фильма

/api/v1/rating/grade\_user\_by\_id\_film - вывод оценки и отзыва пользователя по id фильма

/api/v1/rating/all\_grade\_and\_review\_is\_not\_moderate - вывод всех отзывов требующих модерации

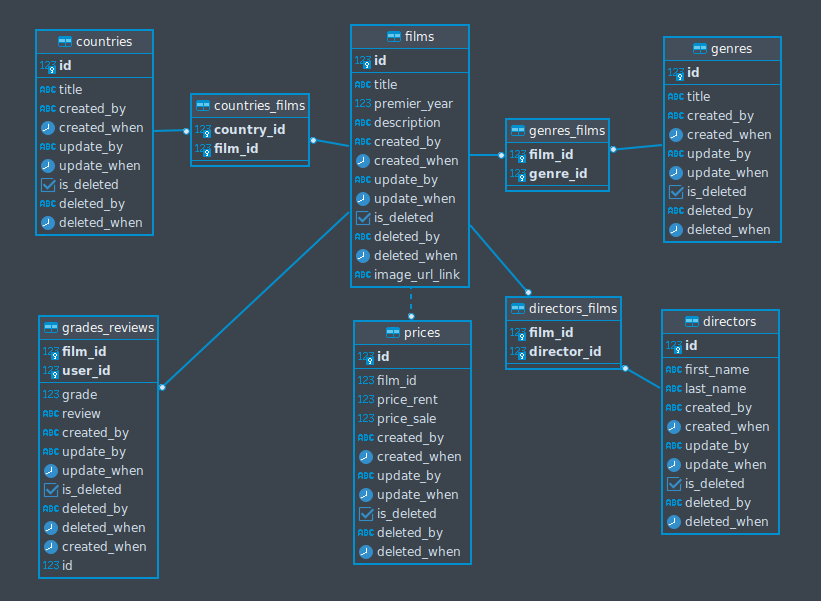
/api/v1/rating/moderate-success - простановка статуса успешной модерации отзыва

/api/v1/rating/moderate-rejected - простановка статуса удаления отзыва

* + БД URL: jdbc:postgresql://194.87.101.43:5432/catalog\_db

user: catalog

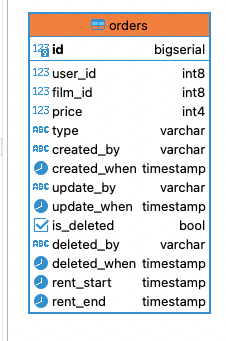
password: CatalogVideoteka2023



* cart-service
  + Описание: работа с корзиной
  + Порт: 8190
  + Context-path: /cart-service
  + Swagger <http://localhost:8190/cart-service/swagger-ui/index.html>
  + Эндпойнты:
  + /api/v1/cart/ - получение корзины пользователя
  + /api/v1/cart/generate - получение токена корзины
  + /api/v1/cart/add - добавление фильма в корзину
  + /api/v1/cart/remove - удаление фильма из корзины
  + /api/v1/cart/clear - очистка корзины после формирования заказа
  + /api/v1/cart/merge - объединение корзин
* cabinet-order-service
  + Описание: обслуживание заказов и личных кабинетов
  + Порт: 8192
  + Context-path: /cabinet-order-service
  + Swagger <http://localhost:8192/cabinet-order-service/swagger-ui/index.htm>
  + Эндпойнты:
  + /api/v1/orders - создание заказа
  + /api/v1/orders - получение заказа пользователя
  + /api/v1/orders/playFilm - имитация просмотра фильма
  + /api/v1/orders/userFilm - фильм пользователя
  + /api/v1/orders/delete - удаление фильма пользователя
  + /api/v1/orders/rent - фильмы пользователя в аренде
  + /api/v1/orders/sale - купленные фильмы пользователя
  + БД URL: jdbc:postgresql://194.87.101.43:5432/orders\_db

user: orders

password: OrdersVideoteka2023



* email-service
  + Описание: рассылка уведомлений пользователю на эл.почту
  + Порт: 8199
  + Context-path: /notification-service
  + Swagger <http://localhost:8199/email-service/swagger-ui/index.html>
  + Эндпойнты:
  + /api/v1/mail/send - отправка письма на почту
* front-service
  + Описание: фронт
  + Порт: 3000
  + Context-path: /front-end
  + dev-mode port: 8081
* Api
  + Описание: DTO-классы:

JWTRequest,

JWTResponse

* + Пакет ru.gb.api.dtos  
    FilmDto - выгрузка списка фильмов для фронта

CountryDto - выгрузка списка стран для фильтров

GenreDto - выгрузка списка жанров для фильтров

DirectorDto - выгрузка списка режиссеров для фильтров

PriceDto - минимальные и максимальные цены для фильтров

* common
  + - Пакет ru.gb.common.generic.entities

GenericEntity - абстрактный класс шаблона сущности

InfoMessage - перечисление, содержащее текстовки ошибок

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Могут понадобиться:

* test-front-service
  + Описание: тестовый фронт
  + Порт: 3001
  + Context-path: /test-front
* Social-service
  + Описание: сервис соцсети видеотеки
  + Порт:
  + Контроллеры:
* Grade-service
  + Описание: сервис рейтингов
  + Порт:
  + Контроллеры:
* Favorite-service
  + Описание: избранное
  + Порт:
  + Контроллеры:

Памятка, что нужно сделать чтобы запустить фронт через Spring

1. помним о том, что данный способ для статического приложения, не для разработки. Разработка ведется по-прежнему в папке /react\_front
2. сейчас в статику положена версия фронта на конец 2го спринта. Если необходимо будет собрать более свежую версию, выполняем последующие шаги:
3. забираем последнюю версию фронта с гитхаба
4. очищаем папку /front-service/src/main/resources/static
5. в терминале идеи переходим в /react\_front/src
6. делаем npm instal, а если инстал проходит с ошибками, то npm instal --force
7. собираем проект: npm run build
8. заходим в папку /build, копируем все ее содержимое и вставляем в /front-service/src/main/resources/static
9. фронт должен запуститься spring boot'ом на порту 3000 : localhost:3000/