

Вопрос 1



Необходимо разработать бэкенд приложения “Фильмотека”, который предоставляет REST API для управления базой данных фильмов.

Приложение должно поддерживать следующие функции:

- добавление информации об актёре (имя, пол, дата рождения),
- изменение информации об актёре.

Возможно изменить любую информацию об актёре, как частично, так и полностью:

- удаление информации об актёре,
- добавление информации о фильме.

При добавлении фильма указываются его название (не менее 1 и не более 150 символов), описание (не более 1000 символов), дата выпуска, рейтинг (от 0 до 10) и список актёров:

- изменение информации о фильме.

Возможно изменить любую информацию о фильме, как частично, так и полностью:

- удаление информации о фильме,
- получение списка фильмов с возможностью сортировки по названию, по рейтингу, по дате выпуска. По умолчанию используется сортировка по рейтингу (по убыванию),
- поиск фильма по фрагменту названия, по фрагменту имени актёра,
- получение списка актёров, для каждого актёра выдаётся также список фильмов с его участием,

- получение списка актёров, для каждого актёра выдаётся также список фильмов с его участием,
- API должен быть закрыт авторизацией,
- поддерживаются две роли пользователей - обычный пользователь и администратор. Обычный пользователь имеет доступ только на получение данных и поиск, администратор - на все действия. Для упрощения можно считать, что соответствие пользователей и ролей задаётся вручную (например, напрямую через БД).

Требования к реализации:

- язык реализации - go,
- для хранения данных используется реляционная СУБД (предпочтительно - PostgreSQL),
- предоставлена спецификация на API (в формате Swagger 2.0 или OpenAPI 3.0).

Бонус: используется подход api-first (генерация кода из спецификации) или code-first (генерация спецификации из кода).

- Для реализации http сервера разрешается использовать только стандартную библиотеку http (без фреймворков),
- логирование - в лог должна попадать базовая информация об обрабатываемых запросах, ошибки,
- код приложения покрыт юнит-тестами не менее чем на 70%,
- Dockerfile для сборки образа,
- docker-compose файл для запуска окружения с работающим приложением и СУБД.