

Тестовое задание Java Mastery 2021. Могилев.

Требования

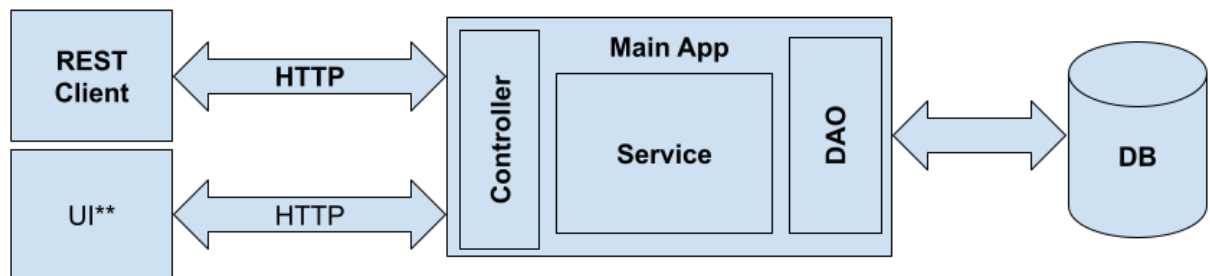
Разработать веб-приложение, представляющее из себя REST сервис, для выполнения CRUD операций над сущностями.

Доменная область приложения: он-лайн каталог книг, где можно добавлять, получать, редактировать и удалять информацию о книгах и их авторах:

1. Две сущности: книга и автор.
2. Книга должна содержать следующую информацию: название, год публикации, авторов, издательство.
3. Книга может иметь одного или более авторов.
4. Книга обязательно должна иметь хотя бы одного автора.
5. Автор должен содержать следующую информацию: имя, фамилия, дата рождения, пол (male, female, undefined).
6. Автор может иметь любое количество книг в под своим авторством, в том числе и 0.
7. Возможность добавлять/удалять/получать/редактировать как Книгу, так и Автора.
8. Приложение должно предоставлять возможность следующих критериев поиска книг (в том числе их комбинаций):
 - По названию
 - По году издания
 - По названию издательства
 - По имени/фамилии автора (в том числе и по первым буквам)
 - По полу
 - По дате рождения авторов

Архитектура

Приветствуется следующая архитектура приложения:



Main App - самостоятельное приложение предоставляющее REST API (это обязательно должен быть REST сервис) и позволяющее выполнять CRUD операции над книгами и авторами. Приложение состоит из трех основных элементов (слоев):

- Controller – это слой REST контроллеров, позволяющий отобразить HTTP запросы на методы обрабатывающие их;
- Service - это сервис выполняющий бизнес-логику, в данном задании бизнес-логики может практически не быть, и сервис просто может делать вызов DAO;
- DAO - выполняет запрос в базу
- Rest Client – можно использовать Postman, curl или любой другой REST клиент, из которого вы будете делать HTTP запросы к своему приложению.
- **UI – не является обязательным к реализации**

Стек технологий

Главные требования – это должно быть веб приложение на основе Spring framework в стиле REST. Минимальная версия Java для написания приложения – 8. Обязательно использовать build tools: Maven или Gradle (Ant очень нежелательно). Всё остальное из списка носит исключительно рекомендательный характер и/или варианты реализации:

- Java 8 или выше (Java 11 приветствуется).
- Spring Boot или Spring MVC без автоконфигурации
- Build tools: Maven или Gradle
- Data layer: Pure JDBC *или* Spring JdbcTemplate *или* Spring + JPA + Hibernate *или* Spring Data JPA
- Database: можно использовать H2 in-memory
- Можно использовать Lombok

Прочие требования:

- Приложение обязательно должно содержать SQL-script для инициализации данных при запуске приложения. То есть изначально должно существовать пару книг и авторов.
- Приложение обязательно должно содержать README файл и с инструкциями по запуску.
- Свое приложение следует выложить на <https://github.com/>

Бальная система оценок

За выполнение описанных требований кандидат получает **40 баллов**. На выполнение задания дается **7 дней с момента получения**. Есть возможность набрать больше баллов, реализовав любые из следующих фич:

- Unit Testing. Покрытие хотя бы половины кода юнит тестами будет оцениваться в +5 за Junit 4 и +10 баллов за Junit 5. Главное условие: покрыть каждый слой (контроллер, сервис, дата)

- Функциональность User (username, password). Сделать пользователя, который может добавлять книги к себе, а также просматривать этот список и удалять книги из него. Пользователи должны быть заранее определены через SQL скрипт. +10 баллов
- UI. Добавить UI с использованием удобных кандидату фреймворков и технологий +10 баллов. Если UI реализован вместе с функциональностью User, то +20 баллов.
- Добавление Spring Security для защиты API. Достаточно будет реализации Basic Authentication +5 баллов. При реализации с UI, функциональностью юзера и страничкой логина +10 баллов
- Использовать в качестве базы данных контейнер Docker. Использовать любую БД (PostgreSQL приветствуется). Команду для старта контейнера поместить в README файл. +10 баллов.

Штрафы

Балы могут отниматься за следующие недочеты.

- -2 балла за каждый день просрочки дедлайна.
- -5 за несоблюдение пункта основных требований
- -5 за отсутствия валидации API (например, при редактировании или добавлении нового автора я могу указать в поле пол значение esfheshfhkes и сервер запишет эту информацию в БД).
- -3 за нарушение архитектуры REST
- -2 за любой мажорный code smell.

ПОМНИТЕ, основная цель задания - разобраться, как работает ваше приложение!!!