

Управление полетами

0. Внимательно прочтите данную задачу. Только после того, как вы прочли всю задачу, собрали все вопросы, вы имеете право контактировать с Александром по телеграмму - **@Cawakaga**

1. У вас есть до 4 часов на выполнение данной задачи.
2. Когда закончите, то вышлите линк на ваш git repository на мейл **alexander@quantum-projects.com** и свяжитесь с **@Cawakaga** по телеграмму.
3. Укажите точные инструкции как запускать ваш код в файле README.md.

+++++

В данной задачи, вам нужно будет **имитировать** базу данных (как часть вашего кода), с встроенными данными (hardcoded data), которая и будет функционировать как ваша база данных. Поэтому не нужно использовать внешние базы данных такие как SQL, H2, redis итд. или использовать Hibernate, Spring Data.

Создайте application по управлению полетами, в которой будет:

- а. Проверка свободен ли билет - 10 баллов:
 - а1. Пользователь указывает ticketId (число), а сервер возвращает ответ (boolean) если билет свободен или нет (ответ будет базироваться на данных, которые вы создали в вашей базе данных).
- б. Создание чек ин (check-in) багажу- 10 баллов:
 - б1. Пользователь предоставляет индекс места назначения (destinationId) и индекс багажа (baggageId) - оба индекса являются числами.
 - б2. Сервер вернет boolean ответ если чекин был успешен.
- в. Добавление возможности купона для скидок (coupon code) - 10 баллов:
 - в1. Пользователь предоставляет индекс купона (coupon id) и цену билета (double).
 - в2. Сервер вернет валидный ответ, если купон существует - создайте заранее таблицу с валидными купонами. Если купон существует, то вернется новая цена в зависимости от скидки, которая выбирается рандомально из списка - 10%, 50%, 60%. Если купон не существует, то сервер вернет ошибку.
- г. Локальный кеш (cache) внутри вашей памяти - 15 баллов:
 - г1. Кеш должен быть создан вами и нельзя использовать внешние библиотеки как spring / guava cache. Например, если идут одни и те же запросы подряд, то ответы будут возвращаться с кеша, а не с базы данных. Вы должны понять, что кеш намного меньше базы данных.

Дизайн - 20 баллов:

1. Дизайн всех слоев архитектуры
2. Дизайн всех интерфейсов
3. Дизайн имплементации интерфейсов
4. Базовая валидация данных и инпутов
5. Минимальные unit tests
6. Логи
7. Чистый и понятный код

Технология - 20 баллов:

1. Используйте maven / gradle для билда
2. Используйте spring boot
3. Используйте github или ему подобные с открытым доступом (мы должны его увидеть и клонировать)

Рабочая задача - 15 баллов:

Лучше сдать неполную рабочую задачу (compile + run), чем задачу не рабочую но с большим количеством фичеров.

+++++

Спасибо и удачи!