**Задание-7**

**DB Migrations with your solution**

Предыдущее задание мы заканчивали очень пафосно! Давайте еще раз приведем эту фразу, и еще раз помечтаем, как успешно пройдет наше собеседование:

*…Однако, если вы найдете время на такое знакомство, и все-таки реализуете поддержку YAML-миграций, у вас будут просто железобетонные аргументы заявлять на собеседовании, что* ***«Тема инструментов для миграции баз данных мне извесна и понятна!»***

Но теперь все – можно заканчивать мечтать. Потому что эта фраза – **банальное вранье**! Ваше собеседование, если вы остановитесь на задании номер 6 (в одном шаге от финиша!), не будет успешным, сногсшибательным или восхитительным. Максимум, оно будет *«неплохим»*!

Потому что, чтобы собеседование прошло блестяще, и чтобы HR-ы выстраивались вдоль вашего пути, пока вы будете покидать компанию, просто чтобы взять у вас афтограф, вам не достаточно знать – *«Как можно применить тот или иной инструмент»*. Это банальщина, это просто показывает уровень ваших знаний… Так себе, прямо скажем, уровень, которого может достичь любой Junior после пяти минут, прведенных в Googl-е с фразой *«Liquibase quick start example»*.

Чтобы поражать слушателей, надо знать не ***«Как это применяется»***, а ***«Как это работает»***!

Друзья, ну что я вам рассказываю, ведь все мы и так помним вот это вот:

*… он просил не беспокоиться, уверял, что он видел не только голых женщин, но даже женщин с начисто содранной кожей…*

Дамы и господа, самое время заглянуть под ту самую содранную кожу! Время написать свой собственный DB Migration API! Вот почему-то прямо сейчас я назвал бы такой API – ***«Crazy Bases»***, но выбор названия, несомненно за вами ☺

Итак, задание. Реализуйте библиотеку, которая позаолит производить миграции баз данных по примеру того как это делают инструменты Liquibase или Flyway. Все ☺

Ну буквально еще несколько уточнений:

1. Ваш API обязаткльно должен быть реализован в виде отдельного проекта.
2. Из п.1 следует, что он должен собираться в JAR-файл.
3. Из п.2 следует, что вы должны также реализовать отдельный WEB-проект, где сможете применить этот JAR-файл.
4. Вы должны продумать структуру таблицы, которая будет хранить информацию об уже примененных ранее миграциях.
5. Вы должны продумать стратегию для именования файлов миграций.
6. Вы должны продумать расположение файлов миграций в проекте, который будет использовать ваш API.
7. Вы должны предусмотреть ситуацию, при которой одна миграция может быть некорректной – в этой ситуации обновление базы должно быть прекращено, а при следубющем запуске (при условии, что миграция исправлена), эта миграция должна примениться.
8. Так как все приведенные выше пункты – это банальное копирование стратегий Liquibase и Flyway (ну да, я не видел женщин с содраной кожей ☹), то, внимание: предложите хотя бы одно нововведение в вашем **API, которого нет у конкурентов**!

После реализации этой задачи, особенно – последнего требования – смело заводите инстаграмм – дня не пройдет, как к вам во френды толпами повалят не только Java-компании, но и спортсмены, политики, звезды шоубизнеса…

Ну подумайте, подумайте, каково это, отказывать HR-у из гугла фразой – «*Извините, в пятницу не могу собеседоваться. Мы с Анжелиной Джоли приглашены на ужин в белый дом. Старик Дональд хочет, таки посмотреть, чем мои миграции лучше евойных!*» – и вам наплевать, что вы неправильно говорите *«евойных»*!

Ну в общем, как то так. Приступайте незамедлительно! ☺