

Тестовое задание

На выбор дается 2 задания (плюс выбор front end framework за вами, из списка предлагаемых).

Результат выкладываете на GitHub или bitbucket – заодно познакомитесь с ними.

После выполнения задания просьба зайти в свой профиль на JavaRush и заполнить ссылку на решенное задание. Файлы проекта **.idea** и скомпилированные классы в **output** или **target** выкладывать не надо.

Ранее задание было строго на GWT, и только одно (CRUD).

Требуемые технологии:

- Maven (для сборки проекта)
- Spring
- Hibernate (для маппинга сущностей приложения на БД, и работы с БД)
- Tomcat 7,8 (для тестирования своего приложения)
- MySQL (база данных). Для упрощения тестирования называйте все свою базу **test**, с логином и паролем **root** (нам не нужно будет для тестирования создавать кучу лишних и ненужных баз)
- Frontend: **angular.js** or **Vaadin** (можно и **GWT**, но будут сложности) or **ZK framework** or **Spring MVC**

Версии можно смело брать самые последние. Конфликтов быть не должно.

Если копируете решение из интернета, то постарайтесь хотя бы сами разобраться что и как работает – очень поможет в дальнейшем в реальном проекте.

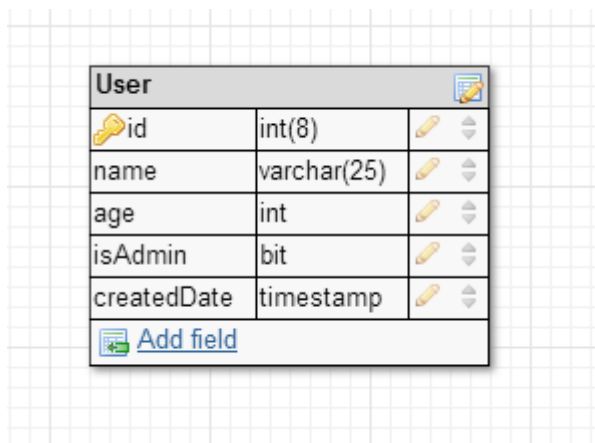
Приложение должно быть в виде проекта, который собирается с помощью Maven. Обязательно должен присутствовать скрипт для создания и наполнения тестовыми данными вашей базы данных. Если для старта приложения после успешного деплоя нужны какие-то дополнительные махинации или танцы с бубном – просьба все это описать в каком-нибудь readme в корне проекта.

Визуальную часть проектов не рисую, т.к. хочу увидеть, как каждый сам реализует и посчитает нормальной свою реализацию задачи.

Огромная просьба: тестируйте свое приложение. Бывало много случаев, когда присылали полностью нерабочее приложение, или с огромным количеством багов.

1. CRUD (create, read, update, delete).

У вас есть всего 1 табличка User:



User		
id	int(8)	
name	varchar(25)	
age	int	
isAdmin	bit	
createdAt	timestamp	
Add field		

isAdmin – в приложении это Boolean переменная.

Необходимо реализовать стандартное CRUD приложение, которое отображает список всех юзеров в базе (с пейджингом). С возможностью их удаления, редактирования, добавления новых, и поиска по уже существующим.

По какому полю искать – каждый решает для себя сам. Можно ограничиться полем name, можно реализовать фильтр для любого поля.

2. TODO list

Реализовать простенькое приложение todolist, для отображения списка дел.

Нужно показывать список уже созданных дел (с пейджингом). Каждое из них можно редактировать, добавлять новые, отмечать как «Выполнено», удалять. Список можно фильтровать как «Все дела», «Только невыполненные», «Выполненные».

Дела хранить в базе. Схему таблички для хранения нужно придумать самому (можно ограничиться одной таблицей, можно разбить на несколько).

На info.javarush.ru вы можете задавать вопросы по тестовому заданию, если, например, непонятно что такое пейджинг и т.п. На задания вам дается 2 недели (вполне достаточно если учесть, что они делаются и за день). Если у вас уходит много времени и что-то не получается, то не расстраивайтесь – возможно вы сильно забурились в материал, что совсем необязательно.

Ниже я привожу ссылку, откуда можно скачать дополнительную литературу, которая может помочь вам в решении тестового задания.

После успешного завершения проекта можно зайти в свой профайл и заполнить ссылку на свой проект.

Ниже приводится ссылка на архив с полезными книгами, которые помогут вам в решении тестового задания.

[ТЫЦ](#)