

качайте учебный проект по ссылке

<https://assets.codeforces.com/files/6e78d88da1176e86/4d/fw1.7z>

Откройте его в IDEA как maven-проект, запустите в локальном Tomcat, используя корневой контекст /.

Убедитесь в работоспособности. Должны отображаться страницы, похожие на Codeforces. Форма регистрации должна функционировать, по ссылке /users должен быть список зарегистрированных пользователей.

Прочитайте исходный код проекта.

Задания

Задание 1

На форму регистрации добавьте еще одно поле passwordConfirmation и дополнительно валидируйте, что они совпали. То есть после выполнения этого задания на форме регистрации будет три поля: login, password, passwordConfirmation. И появится доп. сообщение об ошибке.

Задание 2

Добавьте пользователю long-поле с названием id. При регистрации пользователя устанавливайте ему id по порядку от 1 (потом это будет делать база, думать о race condition пока не надо).

Аналогично, добавьте пользователю поле email. Он указывает email при регистрации. Проверьте уникальность и примерную корректность (мягко, например, содержит ровно один символ @). Добавьте такое поле в регистрацию.

Возможно, вам понадобится удалить файл UserRepositoryImpl из %TEMP%, чтобы почистить данные (ведь у старых пользователей не будет стоять id и email).

Добавьте колонки id и email на страницу с пользователями.

Задание 3

Обратите внимание, что в настоящий момент для всех action-методов в страницах требуются два обязательных параметра HttpServletRequest request, Map<String, Object> view. В ряде случаев (часто) достаточно одного из них или они вообще не нужны.

Измените исходный код FrontServlet так, чтобы в качестве action-метода подыскивался метод просто по названию (нет требования к точной сигнатуре как сейчас). При запуске надо смотреть на типа параметров метода (method.getParameterTypes()) и

если параметр имеет тип `Map.class`, то ожидать что это `view` (передать), тоже самое сделать для `HttpServletRequest`.

После этого следует удалить из всех страниц из всех `action`-методов лишние параметры и убедиться в работоспособности.

Задание 4

Добавьте в футер информацию об общем кол-ве зарегистрированных пользователей. Для этого сделайте методы `findCount` в `UserRepository/UserService`. Добавьте в `macro` для страницы (`common.ftl`) использование переменной шаблона `${userCount}`.

После этого вам надо как-то во `view` во всех страницах всегда класть по ключу `"userCount"` значение `userService.findCount()`.

Для этого сделайте общий базовый класс для всех страниц `Page` и пару методов у него:

- `void before(HttpServletRequest request, Map<String, Object> view)`
- `void after(HttpServletRequest request, Map<String, Object> view)`

В потомках (всех страницах) можно будет переопределять этот метод (не забывайте вызывать `super.before()/super.after()`).

Добавьте в код `FrontServlet` поиск и запуск `before/after`-методов до и после запуска `action` соответственно.

После этого у базового класса `Page` в `before` (или `after`) следует добавить помещение во `view` нужного значения и теперь всюду в футере будет нужная информация.

Задание 5

Сделайте форму авторизации `EnterPage`. Она будет похожа на форму регистрации. После успешной авторизации текущего пользователя (прям объект `User`) можно сложить в сессию. Сделайте так, что если пользователь авторизован, то вместо `Enter | Register` показывается `<UserLogin> | Logout`. Сделайте простую `logout`-страницу, посещение которой трёт из сессии атрибут `"user"` и перенаправляет на главную с соответствующим сообщением. (Вообще, так делать плохо из соображений безопасности, но об этом в нашем курсе позже.)

Задание 6

Добавьте класс предметной области `News` (новость). У него должны быть поля `userId` и `text` (кто создал её и сам текст). Для авторизованного пользователя сделайте еще одну ссылку в основном меню `"Add News"` с простой формой добавления (просто одна `textarea` + кнопка). Сами новости выводите в сайдбаре: добавьте блок `News` и там выводите все новости - её текст и от кого она (логин). Возможно вам понадобится

метод `find(long id)` в `UserRepository/UserService` для поиска нужного пользователя по `userId` из новости. Не переживайте за `performance` (что на каждую новость надо искать пользователя), это учебное задание, а не реальный проект.

