

JDK 1.7

НАБОР ПРАКТИЧЕСКИХ ЗАДАНИЙ

JAVA SE & EE
SPRING FRAMEWORK

г. Москва

DENIS@VOLNENKO.RU

15 апреля 2019 г.

Содержание

I JAVA SE	7
1 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-01	8
2 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-02	9
3 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-03	16
4 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-04	17
5 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-05	18
6 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-06	19
7 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-07	20
8 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-08	21
9 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-09	25
10 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-10	26
11 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-11	29
12 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-12	43
13 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-13	49
14 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-14	52
15 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-15	55
16 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-16	57
17 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-17	59
18 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-18	61

19 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-19	67
II JAVA EE	69
20 ПРАКТИЧЕСКОЕ ЗАДАНИЕ ЕЕ-01	70
21 ПРАКТИЧЕСКОЕ ЗАДАНИЕ ЕЕ-02	77
22 ПРАКТИЧЕСКОЕ ЗАДАНИЕ ЕЕ-03	77
23 ПРАКТИЧЕСКОЕ ЗАДАНИЕ ЕЕ-04	78
24 ПРАКТИЧЕСКОЕ ЗАДАНИЕ ЕЕ-05	79
25 ПРАКТИЧЕСКОЕ ЗАДАНИЕ ЕЕ-06	82
III SPRING	83
26 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-01	84
27 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-02	84
28 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-03	85
29 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-04	85
30 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-05	85
31 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-06	85
32 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-07	85
33 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-08	86

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

НАЗВАНИЕ	ОПИСАНИЕ
CDI	Context and Dependency Injection
DDD	Domain Driven Development
CSS	Cascading Style Sheets
EJB	Enterprise Java Beans
GTM	Global Transaction Management
ESB	Enterprise Service Bus
HTML	HyperText Markup Language
HTTP	Hyper Text Transfer Protocol
IMDG	In-Memory Data Grid
JAX-WS	Java API for XML Web Services
JAX-RS	Java API for RESTful Web Services
JDBC	Java DataBase Connectivity
JNDI	Java Naming and Directory Interface
JMS	Java Message Service
JPA	Java Persistence API
JPQL	Java Persistence Query Language
JS	JavaScript
JSF	JavaServer Faces
JSP	JavaServer Pages
JTA	Java Transaction API
ORM	Object-Relational Mapping
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
RIA	Rich Internet Application
REST	Representational State Transfer
XML	eXtensible Markup Language
АОП	Аспектно-ориентированное программирование
БД	База данных
ИС	Информационная система

ООП	Объектно-ориентированное программирование
ОРСУБД	Объектно-реляционная СУБД
ПО	Программное обеспечение
СУБД	Система управления базами данных

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

НАЗВАНИЕ	ОПИСАНИЕ
База данных (Data Base)	Один или несколько файлов данных, предназначенных для хранения, изменения и обработки больших объемов взаимосвязанной информации.
Веб-интерфейс (Web Interface)	Организованная рабочая зона с подключением к удаленному компьютеру (серверу), которая предоставляет пользователю доступ к информации в структурированном виде.
Горизонтальное масштабирование	Разбиение системы на более мелкие структурные компоненты и их распределение по отдельным физическим машинам (или их группам) и (или) увеличение количества серверов, параллельно выполняющих одну и ту же функцию.
Двусторонний обмен данными	Данные пересыпаются в двух направлениях канала передачи.
Инцидент (Incident)	Отклонение от режима технологического процесса и нарушение внутренних нормативов.
Кластер (Cluster)	Объединение нескольких однородных элементов, которое может рассматриваться как самостоятельная единица, обладающая определенными свойствами.
Кластеризация (Clustering)	Задача разбиения множества объектов на группы, называемые кластерами; внутри каждой группы должны оказаться «похожие» объекты, а объекты разных групп должны быть как можно более отличны.
Конвертация данных	Преобразование данных из одного формата в другой, с сохранением основного логически-структурного содержания информации.
Масштабируемость (Scalability)	Способность системы, сети или процесса справляться с увеличением рабочей нагрузки (увеличивать свою производительность) при добавлении ресурсов (обычно аппаратных).
Односторонний обмен данными	Обмен данными происходит только в одном, заранее заданном, направлении канала передачи данных.
Очистка данных	Процесс анализа качества данных в источнике данных с выполняемым вручную утверждением или отклонением рекомендаций, даваемых системой, и внесением изменений в данные.

Ошибка (Defect)	Несоответствие требованию
Парсинг данных	Синтаксический анализ данных.
Потоковый интерфейс	Интерфейс взаимодействия, представляющий данные в виде потоков байтов, не интерпретируемых системой.
Потоковый режим обмена данными	Метод обмена данными, при котором информация передаётся пользователю от провайдера потокового вещания.
Продукт (Product)	Результат, произведенный в ходе выполнения проекта
Проект (Project)	Ограниченнная во времени деятельность, направленная на разработку уникального продукта
Производительность	Количественная характеристика скорости выполнения определённых операций.
Протокол HTTP/HTTPS	Протокол прикладного уровня передачи данных, поддерживающий шифрование.
Структурированные данные	Информация, оформленная в универсально понятной форме: соблюдение заданного набора правил, определяющих типы данных и отношения между ними.

Часть I

JAVA SE

1 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-01

1.1 РАЗРАБОТАТЬ КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ

1.1.1 РЕАЛИЗОВАТЬ ТОЧКУ ВХОДА PSVM

1.1.2 РЕАЛИЗОВАТЬ ОБРАБОТКУ ТЕРМИНАЛЬНЫХ КОМАНД

1.2 РЕАЛИЗОВАТЬ УПРАВЛЕНИЕ ПРОЕКТАМИ

1.2.1 СОЗДАНИЕ ПРОЕКТА

1.2.2 ПРОСМОТР ПРОЕКТА / ПРОЕКТОВ

1.2.3 РЕДАКТИРОВАНИЕ ПРОЕКТА

1.2.4 УДАЛЕНИЕ ПРОЕКТА

1.3 РЕАЛИЗОВАТЬ ВЫЗОВ СПРАВКИ

1.3.1 ПРИ НАБОРЕ "HELP" ВЫВОДИТСЯ СПИСОК КОМАНД

1.3.2 СОЗДАНИЕ ЗАДАЧИ

1.3.3 ПРОСМОТР ЗАДАЧИ / ЗАДАЧ

1.3.4 РЕДАКТИРОВАНИЕ ЗАДАЧИ

1.3.5 УДАЛЕНИЕ ЗАДАЧИ

1.4 РЕАЛИЗОВАТЬ СПРАВОЧНУЮ КОМАНДУ

1.4.1 ВЫВОДИТЬ СПИСОК ДОСТУПНЫХ КОМАНД

1.5 РЕКОМЕНДАЦИИ

1.5.1 ИСПОЛЬЗОВАТЬ ТОЛЬКО ВОЗМОЖНОСТИ JAVA SE

1.5.2 ИСПОЛЬЗОВАТЬ MAVEN-ПРОЕКТ ДЛЯ СБОРКИ

1.5.3 РАСПРЕДЕЛИТЬ КОД ПО НЕСКОЛЬКИМ КЛАССАМ

1.5.4 КЛАССЫ ПРИЛОЖЕНИЯ ХРАНИТЬ ВО ВЛОЖЕННОМ ПАКЕТЕ "ru./ВАША ФАМИЛИЯ/.tm"

1.5.5 СОБЛЮДАТЬ ТРЕБОВАНИЯ К ИМЕНОВАНИЮ В JAVA

2 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-02

2.1 СОЗДАТЬ GIT-РЕПОЗИТОРИЙ

2.2 СОЗДАТЬ АРАСНЕ MAVEN ПРОЕКТ

2.2.1 ДОБАВИТЬ ИНФОРМАЦИЮ О РАЗРАБОТЧИКЕ В СЕКЦИЮ "developers"

2.3 ДОБАВИТЬ GITIGNORE ДЛЯ НЕНУЖНЫХ ФАЙЛОВ В РЕПОЗИТОРИИ

2.3.1 ДОБАВИТЬ В ИГНОР ПАПКИ С БИНАРНЫМИ ФАЙЛАМИ СБОРКИ ПРОЕКТА ("out", "bin", "target")

2.3.2 ДОБАВИТЬ В ИГНОР ФАЙЛЫ И ПАПКИ ПРОЕКТА IDE (".idea", ".iml")

2.3.3 УБЕДИТЬСЯ В ОТСУТСТВИИ НЕНУЖНЫХ ФАЙЛОВ В РЕПОЗИТОРИИ ИСХОДНОГО КОДА

2.3.4 УБЕДИТЬСЯ В ПРИСУТСТВИИ ВСЕХ КЛАССОВ ПРИЛОЖЕНИЯ В РЕПОЗИТОРИИ

2.4 ДОБАВИТЬ ПРОЕКТНУЮ ДОКУМЕНТАЦИЮ В ФАЙЛ README.MD

2.4.1 ДОБАВИТЬ ТРЕБОВАНИЯ К SOFTWARE

2.4.2 ДОБАВИТЬ ОПИСАНИЕ СТЕКА ТЕХНОЛОГИЙ

2.4.3 ДОБАВИТЬ ИМЯ РАЗРАБОТЧИКА И КОНТАКТЫ

2.4.4 ДОБАВИТЬ КОМАНДЫ ДЛЯ СБОРКИ ПРИЛОЖЕНИЯ

2.4.5 ДОБАВИТЬ КОМАНДЫ ДЛЯ ЗАПУСКА ПРИЛОЖЕНИЯ

2.5 РЕАЛИЗОВАТЬ ВОЗМОЖНОСТЬ ЗАПУСКАТЬ ПРИЛОЖЕНИЕ ИЗ КОНСОЛИ

2.5.1 ДОБАВИТЬ MAVEN-ПЛАГИН ДЛЯ ГЕНЕРАЦИИ МАНИФЕСТА ПРИЛОЖЕНИЯ С УКАЗАНИЕМ ТОЧКИ ВХОДА

2.5.2 ПРОВЕРИТЬ ЗАПУСК ПРИЛОЖЕНИЯ ИЗ КОНСОЛЬНОГО ОКРУЖЕНИЯ В ОПЕРАЦИОННОЙ СИСТЕМЕ

2.5.3 РЕАЛИЗОВАТЬ ЛОГИКУ CRUD-ОПЕРАЦИЙ В ОТДЕЛЬНЫХ МЕТОДАХ ДЛЯ УПРАВЛЕНИЯ ПРЕДМЕТНОЙ ОБЛАСТЬЮ

2.5.4 ИСПОЛЬЗОВАТЬ ЭКЗЕМПЛЯРЫ КЛАССОВ ВМЕТО СТАТИЧЕСКИХ МЕТОДОВ ДЛЯ ВЫЗОВА ФУНКЦИОНАЛА

2.5.5 ИСПОЛЬЗОВАТЬ DATE FORMATTER ДЛЯ РАБОТЫ С ДАТОЙ В ФОРМАТЕ DD.MM.YYYY

2.6 СКРИНШОТЫ

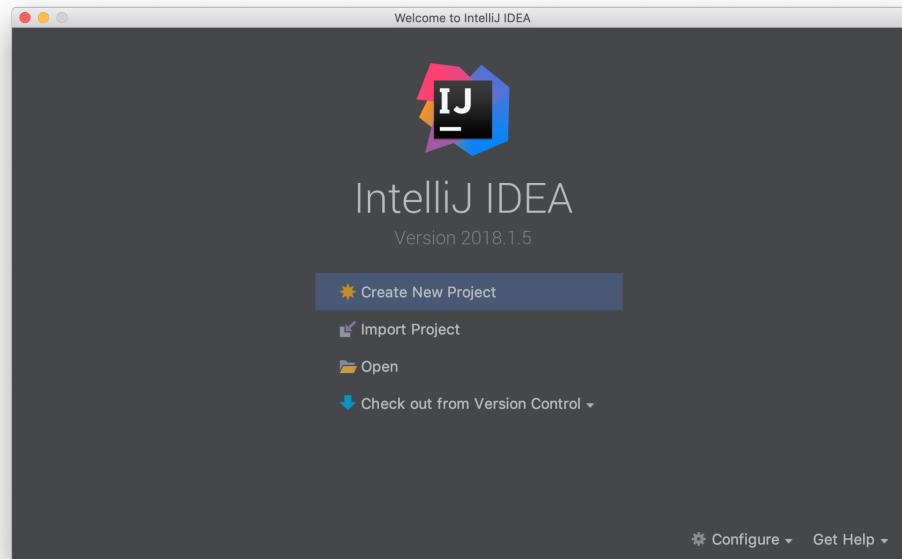


Рис. 1: ПРИВЕТСТВИЕ IDE INTELLIJ IDEA

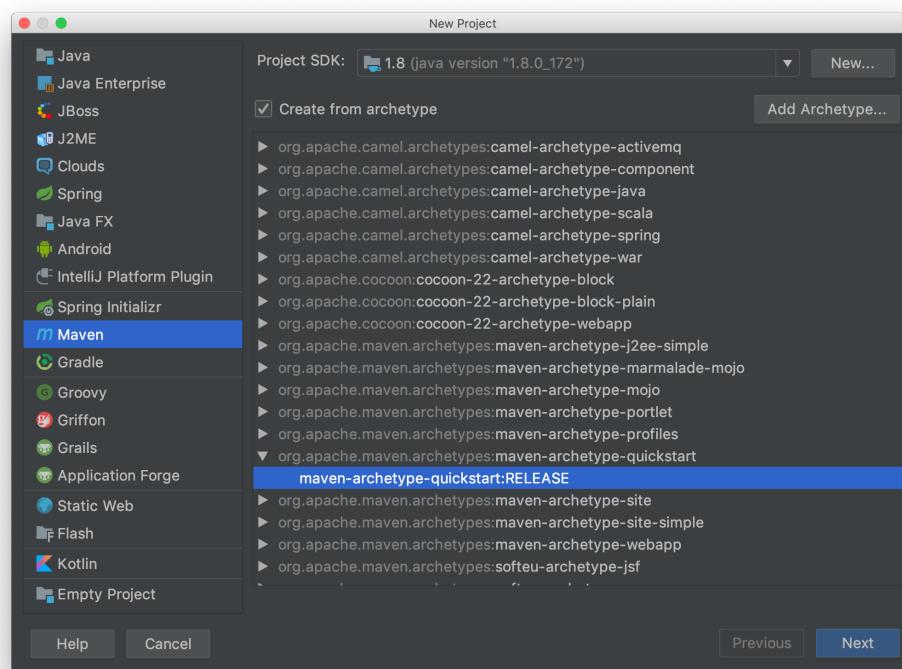


Рис. 2: СОЗДАНИЕ ПРОЕКТА ИЗ MAVEN ARCHETYPE

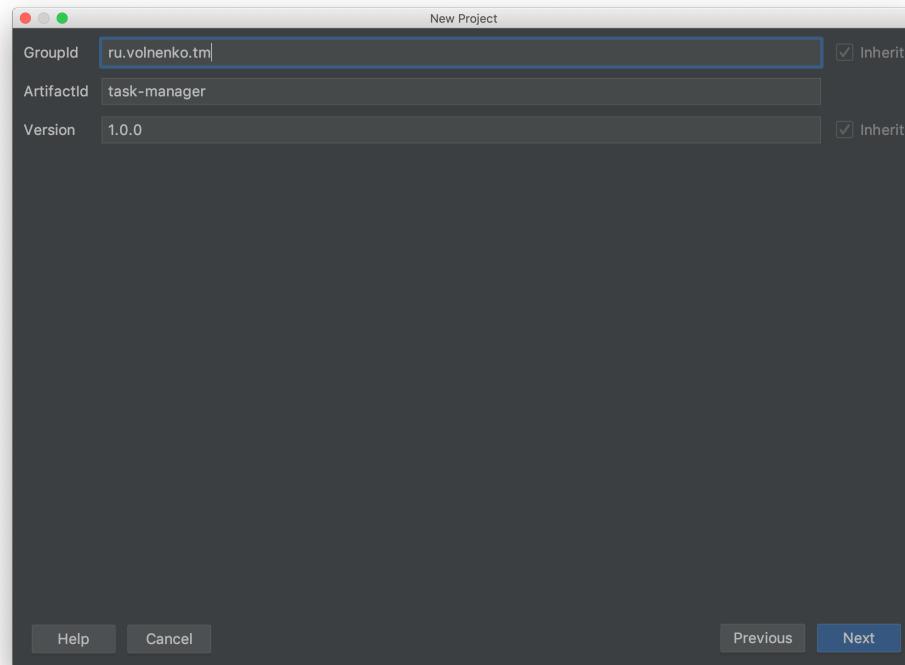


Рис. 3: КОНФИГУРИРОВАНИЕ MAVEN ARCHETYPE

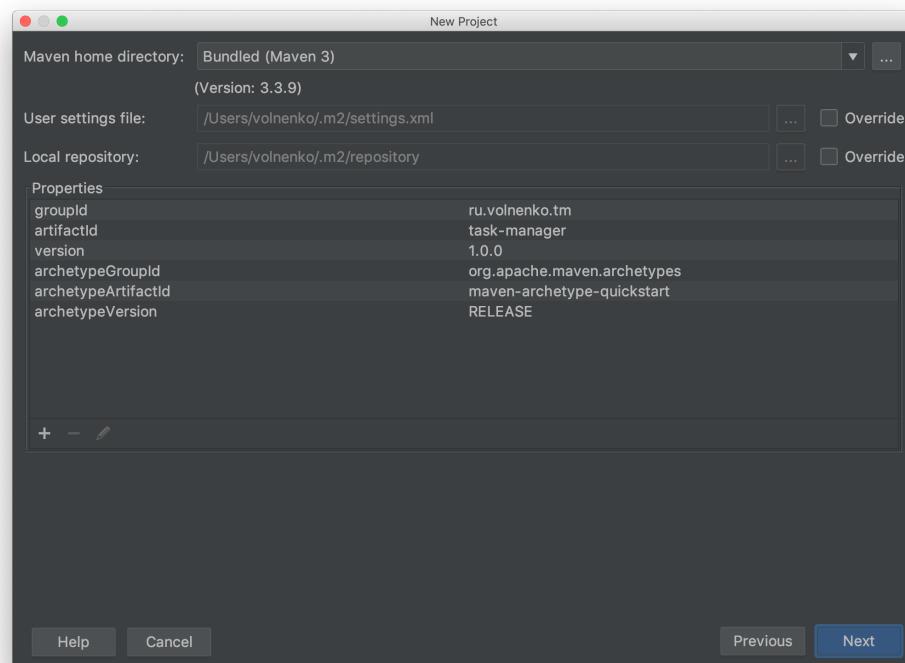


Рис. 4: ПОДТВЕРЖДЕНИЕ КОНФИГУРАЦИИ MAVEN ARCHETYPE

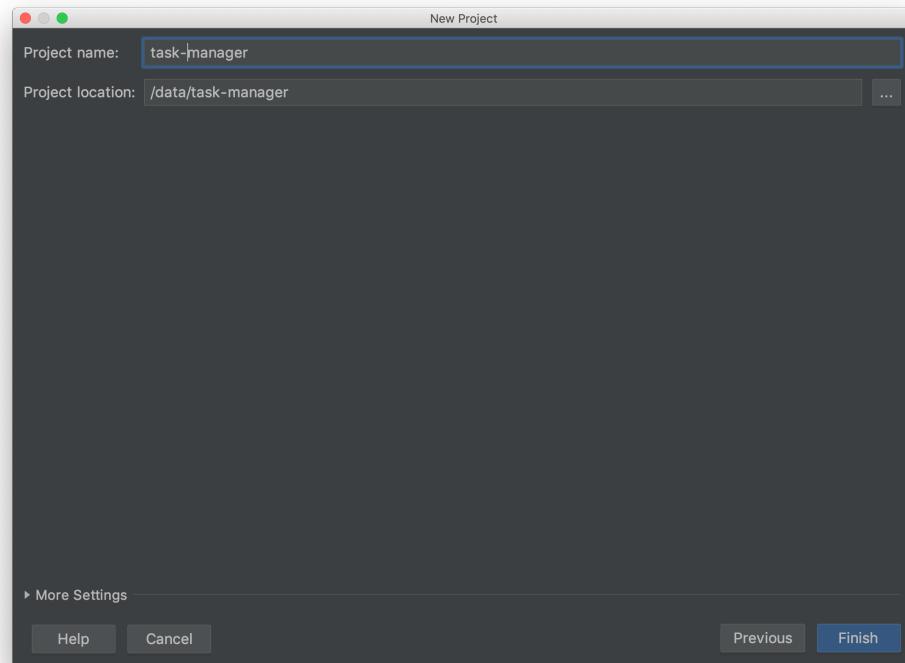


Рис. 5: КОНФИГУРИРОВАНИЕ PROJECT WORKSPACE

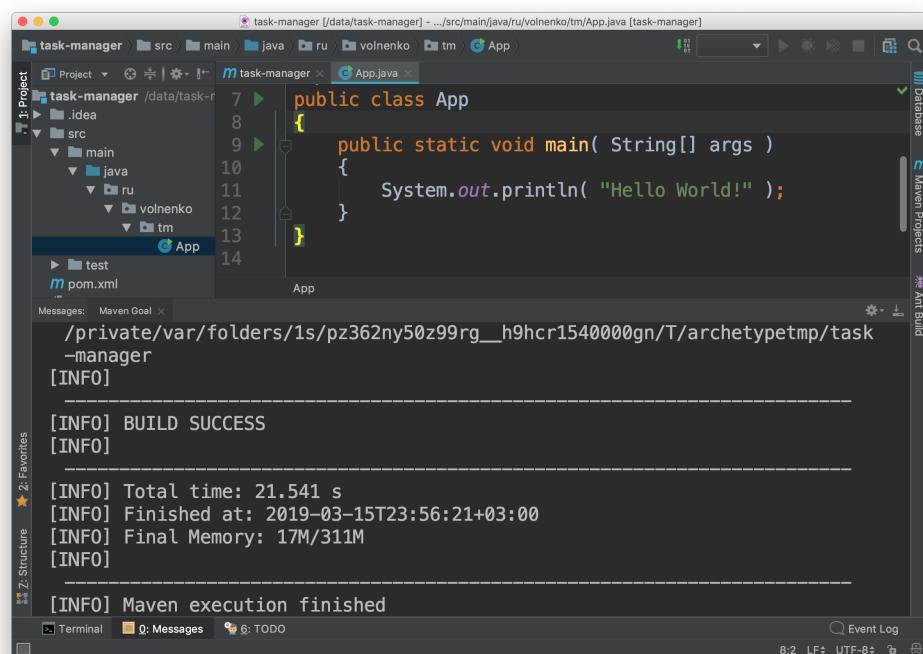


Рис. 6: СОЗДАННАЯ ТОЧКА ВХОДА В ПРИЛОЖЕНИЕ

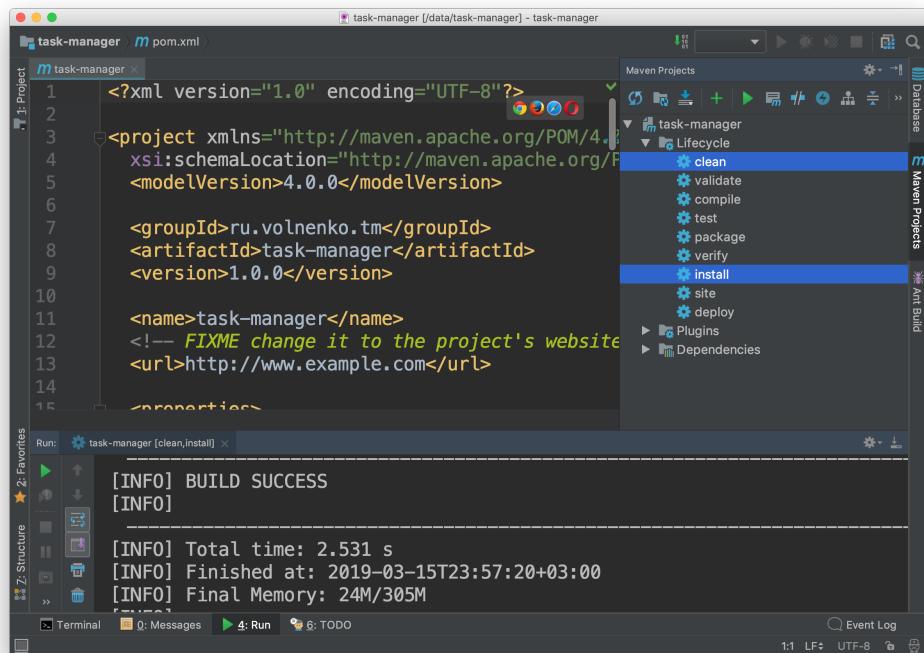


Рис. 7: СОЗДАННЫЙ ИЗ ШАБЛОНА MAVEN-ПРОЕКТ

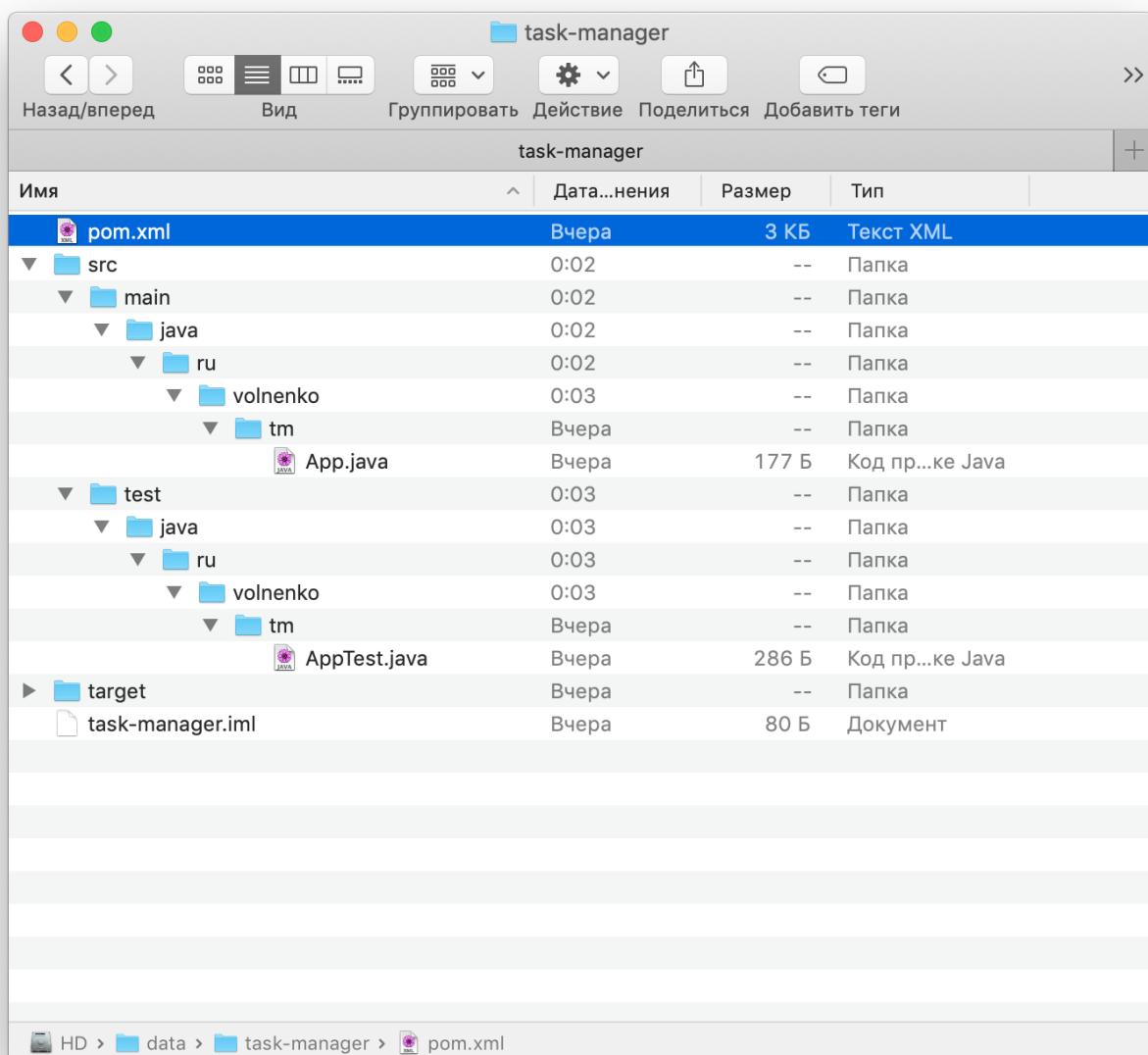
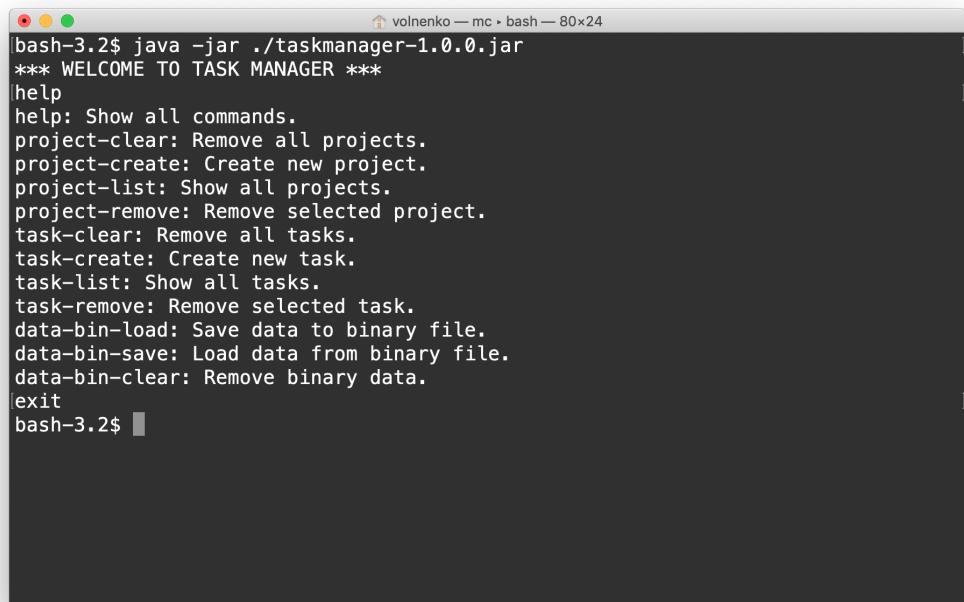


Рис. 8: СТРУКТУРА MAVEN-ПРОЕКТА



```
bash-3.2$ java -jar ./taskmanager-1.0.0.jar
*** WELCOME TO TASK MANAGER ***
/help
help: Show all commands.
/project-clear: Remove all projects.
/project-create: Create new project.
/project-list: Show all projects.
/project-remove: Remove selected project.
/task-clear: Remove all tasks.
/task-create: Create new task.
/task-list: Show all tasks.
/task-remove: Remove selected task.
/data-bin-load: Save data to binary file.
/data-bin-save: Load data from binary file.
/data-bin-clear: Remove binary data.
/exit
bash-3.2$
```

Рис. 9: ЗАПУСК ПРИЛОЖЕНИЯ ИЗ ТЕРМИНАЛА

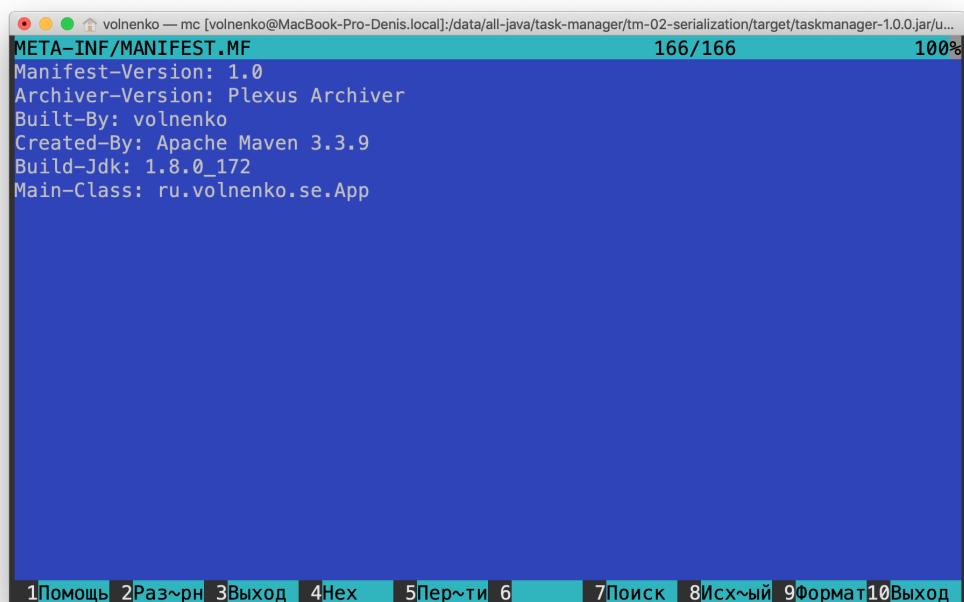


Рис. 10: ЗАПУСКАЕМЫЙ КЛАСС В МАНИФЕСТЕ ПРИЛОЖЕНИЯ

3 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-03

3.1 ПРОВЕСТИ РЕФАКТОРИНГ ПРИЛОЖЕНИЯ

3.1.1 ВЫНЕСТИ ЛОГИКУ ТЕРМИНАЛЬНЫХ КОМАНД В ОТДЕЛЬНЫЕ МЕТОДЫ

3.1.2 ЗАМЕНИТЬ СТРОЧНЫЕ ЛИТЕРАЛЫ НА КОНСТАНТЫ ДЛЯ ТЕРМИНАЛЬНЫХ КОМАНД

3.1.3 ЗАМЕНИТЬ УСЛОВНЫЕ БЛОКИ С ОТРИЦАНИЕМ НА SWITCH ДЛЯ ВЫБОРА ТЕРМИНАЛЬНОЙ КОМАНДЫ

3.2 РАЗРАБОТАТЬ ПРЕДМЕТНУЮ ОБЛАСТЬ

3.3 ВЫДЕЛИТЬ ОТДЕЛЬНЫЙ КЛАСС ДЛЯ ПРОЕКТА

3.3.1 СОЗДАТЬ ПОЛЯ "ID", "НАЗВАНИЕ", "ОПИСАНИЕ", "ДАТА НАЧАЛА", "ДАТА ОКОНЧАНИЯ"

3.4 ВЫДЕЛИТЬ ОТДЕЛЬНЫЙ КЛАСС ДЛЯ ЗАДАЧИ

3.4.1 СОЗДАТЬ ПОЛЯ "ID", "НАЗВАНИЕ", "ОПИСАНИЕ", "ДАТА НАЧАЛА", "ДАТА ОКОНЧАНИЯ", "ID ПРОЕКТА"

3.5 РЕАЛИЗОВАТЬ ЗАВЕРШЕНИЕ ПРИЛОЖЕНИЯ ПО ТЕРМИНАЛЬНОЙ КОМАНДЕ "EXIT"

3.5.1 ПЕРЕЙТИ НА ИСПОЛЬЗОВАНИЕ ТОЛЬКО ОДНОГО БЕСКОНЕЧНОГО ЦИКЛА В ГЛАВНОМ ПОТОКЕ ПРИЛОЖЕНИЯ

3.6 ДОБАВИТЬ ТЕКСТОВОЕ ОПИСАНИЕ ДЛЯ ТЕРМИНАЛЬНЫХ КОМАНД ПРИ ВЫЗОВЕ СПРАВКИ

3.6.1 ПРИ ВЫЗОВЕ СПРАВКИ ДОЛЖНО ОТОБРАЖАТЬСЯ ОПИСАНИЕ ДЛЯ КАЖДОЙ ТЕРМИНАЛЬНОЙ КОМАНДЫ

3.7 РЕАЛИЗОВАТЬ СВЯЗЬ ЗАДАЧИ С ПРОЕКТОМ

3.7.1 РЕАЛИЗОВАТЬ ВОЗМОЖНОСТЬ ПРИКРЕПЛЕНИЯ ЗАДАЧИ К ВЫБРАННому ПРОЕКТУ

3.7.2 РЕАЛИЗОВАТЬ ПРОСМОТР ВСЕХ ЗАДАЧ ПО ВЫБРАННому ПРОЕКТУ

3.7.3 РЕАЛИЗОВАТЬ АВТОМАТИЧЕСКОЕ УДАЛЕНИЕ ЗАДАЧ ПРИ УДАЛЕНИИ ВЫБРАННОГО ПРОЕКТА

3.8 РЕКОМЕНДАЦИИ

3.8.1 КЛАССЫ ПРЕДМЕТНОЙ ОБЛАСТИ ХРАНИТЬ В ПАКЕТЕ "entity"

3.8.2 ИСПОЛЬЗОВАТЬ ДЛЯ СОЕДИНЕНИЯ СУЩНОСТЕЙ СТРОЧНЫЕ ЗНАЧЕНИЯ С ГЕНЕРАЦИЕЙ ПО СТРАТЕГИИ UUID

4 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-04

4.1 СОЗДАТЬ СЛОЙ РЕПОЗИТОРИЕВ В ПАКЕТЕ "repository"

- 4.1.1 КЛАССЫ РЕПОЗИТОРИЕВ ДОЛЖНЫ ОТВЕЧАТЬ ЗА ОПЕРАЦИИ ЧТЕНИЯ И ЗАПИСИ ДАННЫХ В ПРИЛОЖЕНИИ
- 4.1.2 ДОСТУП К КЛАССАМ РЕПОЗИТОРИЕВ МОЖЕТ БЫТЬ ТОЛЬКО ЧЕРЕЗ СООТВЕТСТВУЮЩИЕ СЕРВИСЫ
- 4.1.3 РЕПОЗИТОРИИ ДОЛЖНЫ БЫТЬ РАЗРАБОТАНЫ С ИСПОЛЬЗОВАНИЕМ СТРУКТУРЫ МАР
- 4.1.4 РЕПОЗИТОРИИ ДОЛЖНЫ СОДЕРЖАТЬ СЛЕДУЮЩИЕ МЕТОДЫ "findAll", "findOne", "persist", "merge", "remove", "removeAll"

4.2 СОЗДАТЬ СЛОЙ СЕРВИСОВ В ПАКЕТЕ "service"

- 4.2.1 СЕРВИСЫ ДОЛЖНЫ ДЕЛЕГИРОВАТЬ ВЫЗОВЫ МЕТОДОВ ИЗ РЕПОЗИТОРИЕВ
- 4.2.2 СЕРВИСЫ ДОЛЖНЫ ПРОВОДИТЬ ПЕРВИЧНУЮ ПРОВЕРКУ ДАННЫХ НА ПУСТЫЕ ЗНАЧЕНИЯ СТРОК И NULL
- 4.2.3 СЛОЙ СЕРВИСОВ ДОЛЖЕН СОДЕРЖАТЬ БИЗНЕС ЛОГИКУ ПРИЛОЖЕНИЯ

4.3 СОЗДАТЬ ЗАГРУЗЧИК ПРИЛОЖЕНИЯ НА БАЗЕ ОТДЕЛЬНОГО КЛАССА

- 4.3.1 КЛАСС ЗАГРУЗЧИКА ПРИЛОЖЕНИЯ ДОЛЖЕН НАЗЫВАТЬСЯ "Bootstrap"
- 4.3.2 ЗАГРУЗЧИК ПРИЛОЖЕНИЯ ДОЛЖЕН ПРОИЗВОДИТЬ СОЗДАНИЕ ЭКЗЕМПЛЯРОВ РЕПОЗИТОРИЕВ И СЕРВИСОВ
- 4.3.3 ЗАГРУЗЧИК ПРИЛОЖЕНИЯ ДОЛЖЕН ОТВЕЧАТЬ ЗА ОБРАБОТКУ ТЕРМИНАЛЬНЫХ КОМАНД

4.4 РЕКОМЕНДАЦИИ

- 4.4.1 РЕПОЗИТОРИИ И СЕРВИСЫ НЕ ДОЛЖНЫ СОДЕРЖАТЬ ИНСТРУКЦИИ РАБОТЫ С КОНСОЛЬЮ НА ВВОД / ВЫВОД
- 4.4.2 КАЖДАЯ СУЩНОСТЬ ПРИЛОЖЕНИЯ ДОЛЖНА ИМЕТЬ ПО ОДНОМУ РЕПОЗИТОРИЮ И СЕРВИСУ
- 4.4.3 ЭКЗЕМПЛЯР РЕПОЗИТОРИЯ ДОЛЖЕН ПЕРЕДАВАТЬСЯ В КОНСТРУКТОР КЛАССА СЕРВИСА
- 4.4.4 КЛАСС ТОЧКИ ВХОДА ПРИЛОЖЕНИЯ ДОЛЖЕН НАЗЫВАТЬСЯ "Application" И ВЫЗЫВАТЬ МЕТОД "init" ИЗ "Bootstrap"

5 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-05

5.1 СОЗДАТЬ СЛОЙ КОМАНД

- 5.1.1 СОЗДАТЬ ОТДЕЛЬНЫЙ КЛАСС ДЛЯ КАЖДОЙ ТЕРМИНАЛЬНОЙ КОМАНДЫ
- 5.1.2 КЛАСС КОМАНДЫ ДОЛЖЕН ИМЕТЬ СУФФИКС "Command"
- 5.1.3 КЛАСС КОМАНДЫ ДЛЯ CRUD-ОПЕРАЦИИ ДОЛЖЕН НАЧИНАТЬСЯ С НАЗВАНИЯ СУЩНОСТИ
- 5.1.4 КЛАСС КОМАНДЫ ДЛЯ CRUD-ОПЕРАЦИИ ДОЛЖЕН В НАЗВАНИИ СОДЕРЖАТЬ ИМЯ ОПЕРАЦИИ
- 5.1.5 КЛАСС КОМАНДЫ ДОЛЖЕН БЫТЬ УНАСЛЕДОВАН ОТ КЛАССА АБСТРАКТНОЙ ТЕРМИНАЛЬНОЙ КОМАНДЫ
- 5.1.6 КЛАСС АБСТРАКТНОЙ КОМАНДЫ ДОЛЖЕН ИМЕТЬ МЕТОД "getName" ДЛЯ ВЫДАЧИ НАЗВАНИЯ КОМАНДЫ
- 5.1.7 КЛАСС АБСТРАКТНОЙ КОМАНДЫ ДОЛЖЕН ИМЕТЬ МЕТОД "getDescription" ДЛЯ ВЫДАЧИ ОПИСАНИЯ КОМАНДЫ
- 5.1.8 КЛАСС АБСТРАКТНОЙ КОМАНДЫ ДОЛЖЕН ИМЕТЬ МЕТОД "execute" ДЛЯ БЛОКА ИСПОЛНЯЕМЫХ ИНСТРУКЦИЙ

5.2 РЕАЛИЗОВАТЬ РЕГИСТРАЦИЮ ТЕРМИНАЛЬНЫХ КОМАНД В ЗАГРУЗЧИКЕ ПРИЛОЖЕНИЯ

- 5.2.1 ОРГАНИЗОВАТЬ ХРАНЕНИЕ ЭКЗЕМПЛЯРОВ КОМНАД НА БАЗЕ СТРУКТУРЫ МАР В ЗАГРУЗЧИКЕ ПРИЛОЖЕНИЯ
- 5.2.2 ДЛЯ КЛЮЧА КОМАНДЫ В "МАР" ИСПОЛЬЗОВАТЬ НАЗВАНИЕ ТЕРМИНАЛЬНОЙ КОМАНДЫ
- 5.2.3 ДЛЯ РЕГИСТРАЦИИ ЭКЗЕМПЛЯРОВ КОМАНД ИСПОЛЬЗОВАТЬ МЕТОД "init"
- 5.2.4 ПЕРЕЙТИ НА ИСПОЛЬЗОВАНИЕ СТРУКТУРЫ "МАР" ВМЕСТО SWITCH-БЛОКА

5.3 РЕКОМЕНДАЦИИ

- 5.3.1 КЛАССЫ ТЕРМИНАЛЬНЫХ КОМАНД МОГУТ ОБРАЩАТЬСЯ ТОЛЬКО К СЕРВИСНОМУ УРОВНЮ
- 5.3.2 КЛАССЫ ТЕРМИНАЛЬНЫХ КОМАНД НЕ МОГУТ ОСУЩЕСТВЛЯТЬ ВЫЗОВЫ РЕПОЗИТОРИЕВ
- 5.3.3 КЛАССЫ КОМАНД ДОЛЖНЫ СОДЕРЖАТЬ ЛОГИКУ РАБОТЫ С ВВОДОМ / ВЫВОДОМ КОНСОЛИ ПРИЛОЖЕНИЯ
- 5.3.4 ПЕРЕДАВАТЬ ЗАГРУЗЧИК ПРИЛОЖЕНИЯ В КОНСТРУКТОР АБСТРАКТНОЙ КОМАНДЫ ДЛЯ ДОСТУПА К СЕРВИСАМ

6 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-06

6.1 ДОБАВИТЬ ПЕРЕЧИСЛЕНИЕ "ТИП РОЛИ"

6.1.1 РЕАЛИЗОВАТЬ ОТОБРАЖЕНИЕ ТИПА РОЛИ В ВИДЕ ЛОГИЧЕСКОГО НАЗВАНИЯ "displayName"

6.1.2 ПРИЛОЖЕНИЕ ДОЛЖНО ПОДДЕРЖИВАТЬ ДВА ТИПА РОЛИ "АДМИНИСТРАТОР" И "ОБЫЧНЫЙ ПОЛЬЗОВАТЕЛЬ"

6.2 ДОБАВИТЬ СУЩНОСТЬ "ПОЛЬЗОВАТЕЛЬ"

6.2.1 ПОЛЬЗОВАТЕЛЬ ДОЛЖЕН ИМЕТЬ ПОЛЕ ЛОГИН И ХЕШ ПАРОЛЯ ДЛЯ РЕАЛИЗАЦИИ АВТОРИЗАЦИИ

6.2.2 ПОЛЬЗОВАТЕЛЬ ДОЛЖЕН ИМЕТЬ ПОЛЕ С ТИПОМ РОЛИ

6.2.3 СОЗДАТЬ РЕПОЗИТОРИЙ И СЕРВИС ДЛЯ СУЩНОСТИ "ПОЛЬЗОВАТЕЛЬ"

6.2.4 РЕАЛИЗОВАТЬ КОМАНДУ АВТОРИЗАЦИИ ПОЛЬЗОВАТЕЛЯ

6.2.5 РЕАЛИЗОВАТЬ КОМАНДУ ЗАВЕРШЕНИЯ СЕАНСА ПОЛЬЗОВАТЕЛЯ

6.2.6 РЕАЛИЗОВАТЬ КОМАНДУ РЕГИСТРАЦИИ ПОЛЬЗОВАТЕЛЯ

6.2.7 РЕАЛИЗОВАТЬ КОМАНДУ ОБНОВЛЕНИЯ ПАРОЛЯ ПОЛЬЗОВАТЕЛЯ

6.2.8 РЕАЛИЗОВАТЬ КОМАНДЫ ПРОСМОТРА И РЕДАКТИРОВАНИЯ ТЕКУЩЕГО ПРОФИЛЯ ПОЛЬЗОВАТЕЛЯ

6.2.9 РЕАЛИЗОВАТЬ СОЗДАНИЕ ДВУХ ПРЕДОПРЕДЕЛЕННЫХ УЧЕТНЫХ ЗАПИСЕЙ ПОЛЬЗОВАТЕЛЕЙ С РАЗНЫМИ РОЛЯМИ

6.2.10 РЕАЛИЗОВАТЬ ПРИВЯЗКУ ПРОЕКТОВ И ЗАДАЧ К УЧЕТНОЙ ЗАПИСИ ПОЛЬЗОВАТЕЛЯ ЧЕРЕЗ ПОЛЕ "userId"

6.2.11 РАЗДЕЛИТЬ КОМАНДЫ НА БЕЗОПАСНЫЕ И ТРЕБУЮЩИЕ АВТОРИЗАЦИИ

6.2.12 ВСЕ CRUD-ОПЕРАЦИИ ДОЛЖНЫ ВЫПОЛНЯТЬСЯ ТОЛЬКО ПОСЛЕ АВТОРИЗАЦИИ ПОЛЬЗОВАТЕЛЯ В ПРИЛОЖЕНИИ

6.3 РЕКОМЕНДАЦИИ

6.3.1 ДОБАВИТЬ ВЫЗОВ СОЗДАНИЯ ПОЛЬЗОВАТЕЛЕЙ В МЕТОД ИНИЦИАЛИЗАЦИИ ЗАГРУЗЧИКА ПРИЛОЖЕНИЯ

6.3.2 ДЛЯ ХЕШИРОВАНИЯ ПАРОЛЯ НЕОБХОДИМО ИСПОЛЬЗОВАТЬ MD5-АЛГОРИТМ

6.3.3 РЕАЛИЗОВАТЬ ХРАНЕНИЕ ТЕКУЩЕГО ПОЛЬЗОВАТЕЛЯ В ЗАГРУЗЧИКЕ ПРИЛОЖЕНИЯ

6.3.4 ПРОВЕРИТЬ ИЗОЛЯЦИЮ ДАННЫХ ПО ПРОЕКТАМ И ЗАДАЧАМ МЕЖДУ РАЗНЫМИ УЧЕТНЫМИ ЗАПИСЯМИ

7 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-07

7.1 РАЗРАБОТАТЬ ИНТЕРФЕЙСЫ ДЛЯ РЕПОЗИТОРИЕВ

7.1.1 ВЫНЕСТИ ВСЕ CRUD-ОПЕРАЦИИ РЕПОЗИТОРИЕВ В ИНТЕРФЕЙСЫ

7.2 РАЗРАБОТАТЬ ИНТЕРФЕЙСЫ ДЛЯ СЕРВИСОВ

7.2.1 ИСПОЛЬЗОВАТЬ ИНТЕРФЕЙСЫ РЕПОЗИТОРИЕВ В РЕАЛИЗАЦИИ СЕРВИСОВ

7.3 РАЗРАБОТАТЬ ИНТЕРФЕЙС ДЛЯ ЗАГРУЗЧИКА

7.3.1 СОЗДАТЬ ИНТЕРФЕЙС "ServiceLocator" ДЛЯ ЗАГРУЗЧИКА ПРИЛОЖЕНИЯ

7.3.2 ДОБАВИТЬ ДЕКЛАРАЦИЮ GET-МЕТОДОВ ДЛЯ ПОЛУЧЕНИЯ СЕРВИСОВ В ИНТЕРФЕЙС СЕРВИС ЛОКАТОРА

7.3.3 РЕАЛИЗОВАТЬ ДОСТУП К СЕРВИСАМ ИЗ КОМАНД ТОЛЬКО ЧЕРЕЗ ИНТЕРФЕЙС СЕРВИС ЛОКАТОРА

7.4 ПРОВЕСТИ РЕФАКТОРИНГ ПРИЛОЖЕНИЯ

7.4.1 ПРИМЕНИТЬ К КЛАССАМ, ПЕРЕМЕННЫМ И АРГУМЕНТАМ КЛЮЧЕВОЕ СЛОВО "final"

7.4.2 ВЫНЕСТИ ОБЩУЮ ЛОГИКУ СЕРВИСОВ И РЕПОЗИТОРИЕВ В АБСТРАКТНЫЕ КЛАССЫ

7.4.3 ПРИМЕНИТЬ ОБОБЩЕНИЯ ДЛЯ СЕРВИСОВ И РЕПОЗИТОРИЕВ С ЧАСТИ ИНТЕРФЕЙСОВ И РЕАЛИЗАЦИЙ

7.5 ДОБАВИТЬ ИНФОРМАЦИЮ О РАЗРАБОТЧИКЕ И ВЕРСИИ ПРИЛОЖЕНИЯ В МАНИФЕСТ

7.5.1 ПОДКЛЮЧИТЬ MAVEN-ПЛАГИН ДЛЯ ПОДСЧЕТА НОМЕРА ТЕКУЩЕЙ СБОРКИ И ПЕРЕДАТЬ НОМЕР В МАНИФЕСТ

7.5.2 ПОДКЛЮЧИТЬ MAVEN-ЗАВИСИМОСТЬ ДЛЯ ЧТЕНИЯ ИНФОРМАЦИИ ИЗ ФАЙЛА "MANIFEST.MF"

7.5.3 РЕАЛИЗОВАТЬ ТЕРМИНАЛЬНУЮ КОМАНДУ "about" ДЛЯ ОТОБРАЖЕНИЯ ИНФОРМАЦИИ О СБОРКЕ ПРИЛОЖЕНИЯ

7.6 РЕКОМЕНДАЦИИ

7.6.1 ПРИ ДЕКЛАРАЦИИ ПЕРЕМЕННЫХ И КОНСТАНТ ВСЕГДА ИСПОЛЬЗОВАТЬ ИНТЕРФЕЙСЫ ПРИ ИХ НАЛИЧИИ

8 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-08

8.1 ПРОВЕСТИ РЕФАКТОРИНГ ПРИЛОЖЕНИЯ

- 8.1.1 ДОБАВИТЬ АННОТАЦИИ "@NotNull" И "@Nullable" ИЗ ПАКЕТА "JetBrains" ДЛЯ ВСЕХ ПАРАМЕТРОВ И ПЕРЕМЕННЫХ
- 8.1.2 ДОБАВИТЬ LOMBOK АННОТАЦИИ ДЛЯ ГЕНЕРАЦИИ КОНСТРУКТОРОВ И GET / SET МЕТОДОВ

8.2 НАСТРОИТЬ СБОРКУ РЕЛИЗА ПРИЛОЖЕНИЯ

- 8.2.1 ПРИЛОЖЕНИЕ ДОЛЖНО ЗАПУСКАТЬСЯ ИЗ КОНСОЛИ ВМЕСТЕ С ИСПОЛЬЗУЕМЫМИ ЗАВИСИМОСТЯМИ БЕЗ ОШИБОК
- 8.2.2 ПОДКЛЮЧИТЬ MAVEN-ПЛАГИН ДЛЯ КОПИРОВАНИЯ ВСЕХ ЗАВИСИМОСТЕЙ В СБОРОЧНУЮ ДИРЕКТОРИЮ "release/lib"
- 8.2.3 ПОДКЛЮЧИТЬ MAVEN-ПЛАГИН ДЛЯ КОПИРОВАНИЯ СБОРКИ ПРИЛОЖЕНИЯ ДИРЕКТОРИЮ "release/bin"

8.3 ДОРАБОТАТЬ ТЕРМИНАЛЬНЫЕ КОМАНДЫ

- 8.3.1 РЕАЛИЗОВАТЬ РЕГИСТРАЦИЮ КОМАНД С ИСПОЛЬЗОВАНИЕМ ПЕРЕЧИСЛЕНИЯ КЛАССОВ В МАССИВЕ
- 8.3.2 РАЗМЕСТИТЬ МАССИВ С КЛАССАМИ ТЕРМИНАЛЬНЫХ КОМАНД В ТОЧКЕ ВХОДА В КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ
- 8.3.3 ПЕРЕДАВАТЬ МАССИВ С КЛАССАМИ ТЕРМИНАЛЬНЫХ КОМАНД В МЕТОД ИНИЦИАЛИЗАЦИИ ЗАГРУЗЧИКА ПРИЛОЖЕНИЯ
- 8.3.4 РАЗРАБОТАТЬ МЕТОД РЕГИСТРАЦИИ ТЕРМИНАЛЬНОЙ КОМАНДЫ В ЗАГРУЗЧИКИ ПРИЛОЖЕНИЯ ДЛЯ РАЗМЕЩЕНИЯ В СТРУКТУРЕ МАР
- 8.3.5 РЕАЛИЗОВАТЬ ПРОВЕРКУ РОДИТЕЛЬСКОГО КЛАССА ТЕРМИНАЛЬНОЙ КОМАНДЫ ПРИ РЕГИСТРАЦИИ В ЗАГРУЗЧИКЕ ПРИЛОЖЕНИЯ

8.4 РЕКОМЕНДАЦИИ

- 8.4.1 РЕАЛИЗОВАТЬ СОЗДАНИЕ ТЕРМИНАЛЬНЫХ КОМАНД ПРИ РЕГИСТРАЦИИ ЧЕРЕЗ ВЫЗОВ МЕТОДА "newInstance" ИЗ КЛАССА КОМАНДЫ
- 8.4.2 РЕАЛИЗОВАТЬ ПЕРЕДАЧУ ССЫЛКИ НА СЕРВИС ЛОКАТОР В ТЕРМИНАЛЬНУЮ КОМАНДУ ЧЕРЕЗ ВЫЗОВ SET-МЕТОДА
- 8.4.3 РЕАЛИЗОВАТЬ ТЕРМИНАЛЬНЫЙ СЕРВИС ДЛЯ ЗАХВАТА КОМАНД С КОНСОЛИ

8.5 СКРИНШОТЫ

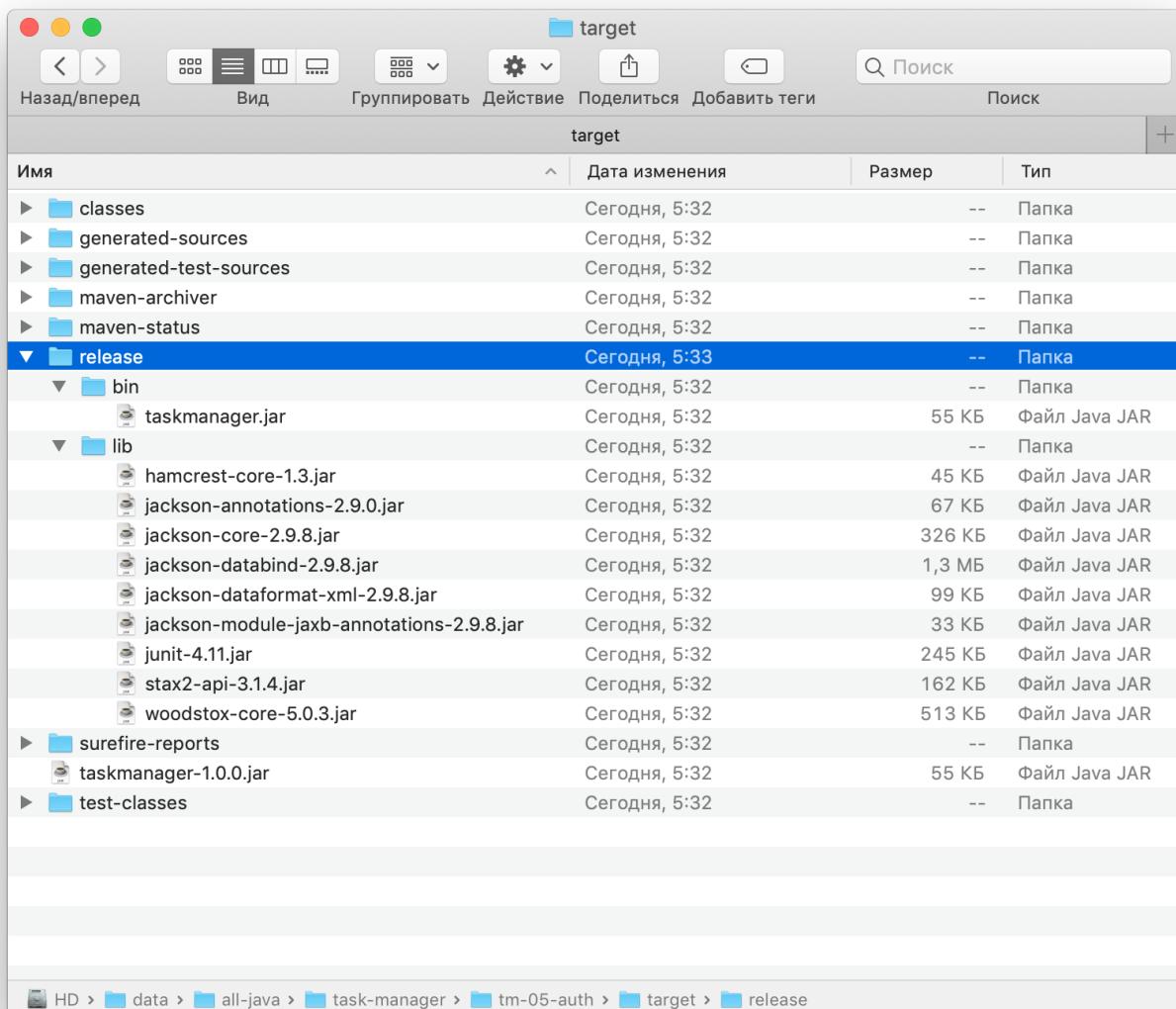
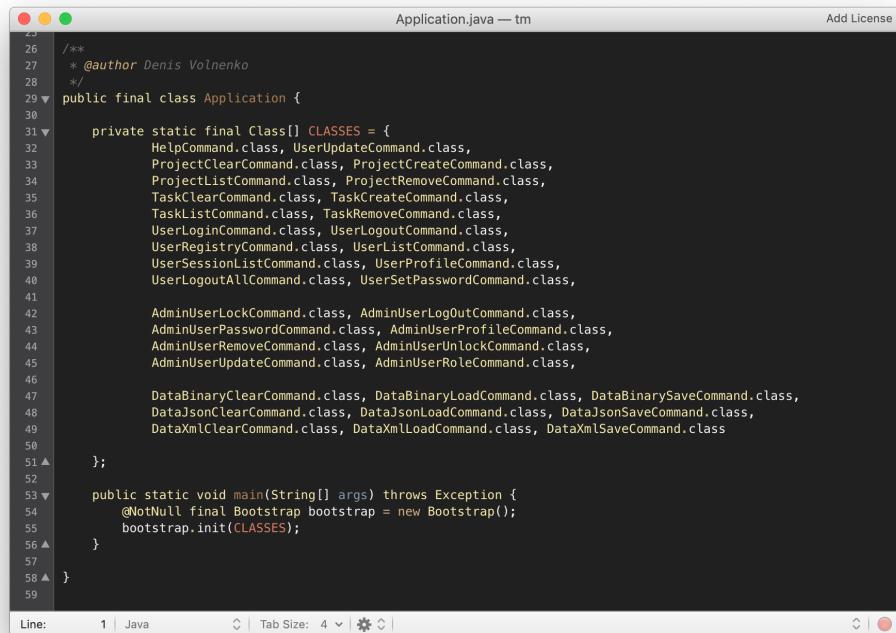


Рис. 11: РЕЛИЗ ПРИЛОЖЕНИЯ С ЗАВИСИМОСТЯМИ

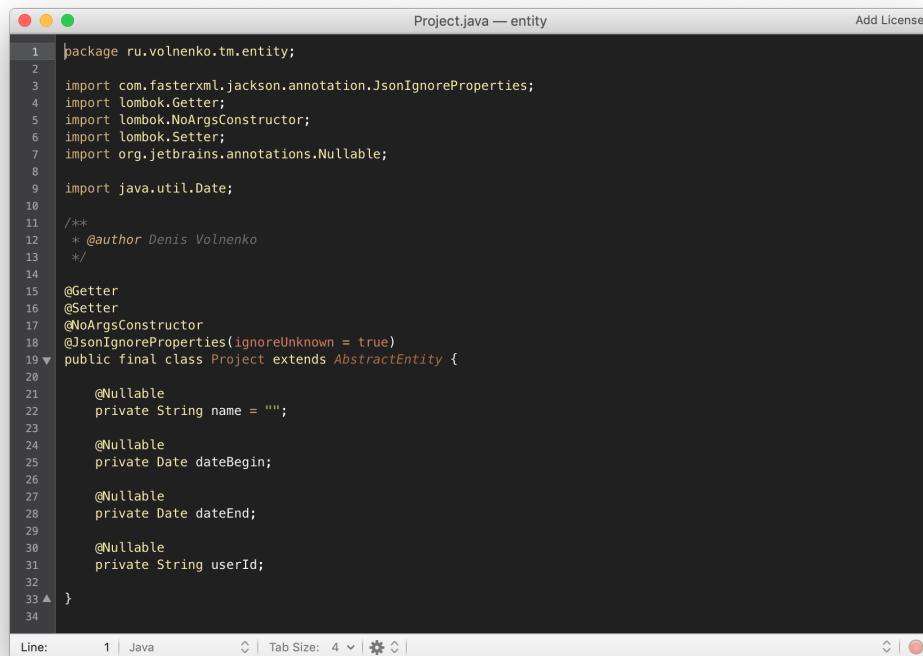


```

23
24 /**
25 * @author Denis Volnenko
26 */
27
28 public final class Application {
29
30     private static final Class[] CLASSES = {
31         HelpCommand.class, UserUpdateCommand.class,
32         ProjectClearCommand.class, ProjectCreateCommand.class,
33         ProjectListCommand.class, ProjectRemoveCommand.class,
34         TaskClearCommand.class, TaskCreateCommand.class,
35         TaskListCommand.class, TaskRemoveCommand.class,
36         UserLoginCommand.class, UserLogoutCommand.class,
37         UserRegistryCommand.class, UserListCommand.class,
38         UserSessionListCommand.class, UserProfileCommand.class,
39         UserLogoutAllCommand.class, UserSetPasswordCommand.class,
40
41         AdminUserLockCommand.class, AdminUserLogOutCommand.class,
42         AdminUserPasswordCommand.class, AdminUserProfileCommand.class,
43         AdminUserRemoveCommand.class, AdminUserUnlockCommand.class,
44         AdminUserUpdateCommand.class, AdminUserRoleCommand.class,
45
46         DataBinaryClearCommand.class, DataBinaryLoadCommand.class, DataBinarySaveCommand.class,
47         DataJsonClearCommand.class, DataJsonLoadCommand.class, DataJsonSaveCommand.class,
48         DataXmlClearCommand.class, DataXmlLoadCommand.class, DataXmlSaveCommand.class
49
50     };
51 }
52
53 public static void main(String[] args) throws Exception {
54     @NotNull final Bootstrap bootstrap = new Bootstrap();
55     bootstrap.init(CLASSES);
56 }
57
58 }
59

```

Рис. 12: РЕГИСТРАЦИЯ КЛАССОВ КОМАНД



```

1 package ru.volnenko.tm.entity;
2
3 import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
4 import lombok.Getter;
5 import lombok.NoArgsConstructor;
6 import lombok.Setter;
7 import org.jetbrains.annotations.Nullable;
8
9 import java.util.Date;
10
11 /**
12 * @author Denis Volnenko
13 */
14
15 @Getter
16 @Setter
17 @NoArgsConstructor
18 @JsonIgnoreProperties(ignoreUnknown = true)
19 public final class Project extends AbstractEntity {
20
21     @Nullable
22     private String name = "";
23
24     @Nullable
25     private Date dateBegin;
26
27     @Nullable
28     private Date dateEnd;
29
30     @Nullable
31     private String userId;
32
33 }
34

```

Рис. 13: GET И SET МЕТОДЫ ОТ LOMBOK

```

116
117
118    @Override
119    public Session open(@Nullable final String login, @Nullable final String password) {
120        final boolean check = checkDataAccess(login, password);
121        if (!check) return null;
122        @Nullable final User user = serviceLocator.getUserService().findByLogin(login);
123        if (user == null) return null;
124        @Nullable final Session session = new Session();
125        session.setUserId(user.getId());
126        session.setTimestamp(System.currentTimeMillis());
127        repository.merge(session);
128        return sign(session);
129    }
130
131    @Override
132    public boolean checkDataAccess(@Nullable final String login, @Nullable final String password) {
133        if (login == null || login.isEmpty()) return false;
134        if (password == null || password.isEmpty()) return false;
135        @Nullable final User user = serviceLocator.getUserService().findByLogin(login);
136        if (user == null) return false;
137        @Nullable final String passwordHash = PasswordHashUtil.md5(password);
138        if (passwordHash == null || passwordHash.isEmpty()) return false;
139        return passwordHash.equals(user.getPasswordHash());
140    }
141
142    @Override
143    public void signOutByLogin(@Nullable final String login) {
144        if (login == null || login.isEmpty()) return;
145        @Nullable final User user = serviceLocator.getUserService().findByLogin(login);
146        if (user == null) return;
147        @Nullable final String userId = user.getId();
148        repository.removeById(userId);
149    }
150

```

Рис. 14: АННОТАЦИЯ NULLABLE ОТ JETBRAINS

```

13    public final class PropertyService implements IPropertyService {
14
15        @NotNull
16        private final String NAME = "/application.properties";
17
18        @NotNull
19        private final Properties properties = new Properties();
20
21        public void init() throws Exception {
22            @NotNull final InputStream inputStream = PropertyService.class.getResourceAsStream(NAME);
23            properties.load(inputStream);
24        }
25
26        @NotNull
27        @Override
28        public String getServerHost() {
29            @NotNull final String propertyHost = properties.getProperty("server.host");
30            @NotNull final String envHost = System.getProperty("server.host");
31            if (envHost != null) return envHost;
32            return propertyHost;
33        }
34
35        @NotNull
36        @Override
37        public Integer getServerPort() {
38            @NotNull final String propertyPort = properties.getProperty("server.port");
39            @NotNull final String envPort = System.getProperty("server.port");
40            @NotNull String value = propertyPort;
41            if (envPort != null) value = envPort;
42            if (value == null) return null;
43            return Integer.parseInt(value);
44        }
45
46        @NotNull
47        @Override

```

Рис. 15: АННОТАЦИЯ NOTNULL ОТ JETBRAINS

9 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-09

9.1 РЕАЛИЗОВАТЬ СТАТУС ЗАДАЧ И ПРОЕКТОВ

- 9.1.1 РАЗРАБОТАТЬ ПЕРЕЧИСЛЕНИЕ СТАТУСОВ ГОТОВНОСТИ СО ЗНАЧЕНИЯМИ (ЗАПЛАНИРОВАНО, В ПРОЦЕССЕ И ГОТОВО)
- 9.1.2 РЕАЛИЗОВАТЬ ПОЛЕ "displayName" ДЛЯ ПЕРЕЧИСЛЕНИЯ СТАТУСОВ ГОТОВНОСТИ
- 9.1.3 ИСПОЛЬЗОВАТЬ СТАТУС "ЗАПЛАНИРОВАНО" ДЛЯ НАЧАЛЬНОГО ЗНАЧЕНИЯ В СУЩНОСТЯХ ЗАДАЧИ И ПРОЕКТА

9.2 РЕАЛИЗОВАТЬ СОРТИРОВКУ ПРОЕКТОВ И ЗАДАЧ

- 9.2.1 РЕАЛИЗОВАТЬ ВОЗМОЖНОСТЬ ВЫВОДИТЬ ПРОЕКТЫ И ЗАДАЧИ В ПОРЯДКЕ ИХ СОЗДАНИЯ В СИСТЕМЕ
- 9.2.2 РЕАЛИЗОВАТЬ ВОЗМОЖНОСТЬ ВЫВОДИТЬ ПРОЕКТЫ И ЗАДАЧИ ОТСОРТИРОВАННЫЕ ПО ДАТЕ НАЧАЛА
- 9.2.3 РЕАЛИЗОВАТЬ ВОЗМОЖНОСТЬ ВЫВОДИТЬ ПРОЕКТЫ И ЗАДАЧИ ОТСОРТИРОВАННЫЕ ПО ДАТЕ ОКОНЧАНИЯ
- 9.2.4 РЕАЛИЗОВАТЬ ВОЗМОЖНОСТЬ ВЫВОДИТЬ ПРОЕКТЫ И ЗАДАЧИ ОТСОРТИРОВАННЫЕ ПО СТАТУСУ ГОТОВНОСТИ

9.3 РЕАЛИЗОВАТЬ ПОИСК ПРОЕКТОВ И ЗАДАЧ ПО ЧАСТИ НАЗВАНИЯ ИЛИ ОПИСАНИЯ

9.4 РЕАЛИЗОВАТЬ ОБРАБОТКУ ИСКЛЮЧЕНИЙ

- 9.4.1 ДОБАВИТЬ ВЫБРОС ИСКЛЮЧЕНИЙ ИЗ МЕТОДА ВЫПОЛНЕНИЯ АБСТРАКТНОЙ ТЕРМИНАЛЬНОЙ КОМАНДЫ
- 9.4.2 РЕАЛИЗОВАТЬ ЕДИНЫЙ БЛОК ОБРАБОТКИ ИСКЛЮЧЕНИЙ В ГЛАВНОЙ ПЕТЛЕ ПРИЛОЖЕНИЯ

9.5 РЕКОМЕНДАЦИИ

- 9.5.1 РЕАЛИЗОВАТЬ ОБРАБОТКУ ИСКЛЮЧЕНИЙ БЕЗ ПРЕРЫВАНИЯ ВЫПОЛНЕНИЯ ОСНОВНОГО ПОТОКА

10 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-10

10.1 РЕАЛИЗОВАТЬ СОХРАНЕНИЕ ДАННЫХ

- 10.1.1 РЕАЛИЗОВАТЬ КОМАНДУ СОХРАНЕНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ С ИСПОЛЬЗОВАНИЕМ СЕРИАЛИЗАЦИИ
- 10.1.2 РЕАЛИЗОВАТЬ КОМАНДУ СОХРАНЕНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ С ИСПОЛЬЗОВАНИЕМ ЭКСТЕРНАЛТИЗАЦИИ И ТЕХНОЛОГИИ JAX-B В ТРАНСПОРТНОМ ФОРМАТЕ XML
- 10.1.3 РЕАЛИЗОВАТЬ КОМАНДУ СОХРАНЕНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ С ИСПОЛЬЗОВАНИЕМ ЭКСТЕРНАЛТИЗАЦИИ И ТЕХНОЛОГИИ JAX-B В ТРАНСПОРТНОМ ФОРМАТЕ JSON
- 10.1.4 РЕАЛИЗОВАТЬ КОМАНДУ СОХРАНЕНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ С ИСПОЛЬЗОВАНИЕМ ЭКСТЕРНАЛТИЗАЦИИ И БИБЛИОТЕКИ FASTERXML В ТРАНСПОРТНОМ ФОРМАТЕ XML
- 10.1.5 РЕАЛИЗОВАТЬ КОМАНДУ СОХРАНЕНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ С ИСПОЛЬЗОВАНИЕМ ЭКСТЕРНАЛТИЗАЦИИ И БИБЛИОТЕКИ FASTERXML В ТРАНСПОРТНОМ ФОРМАТЕ JSON

10.2 РЕАЛИЗОВАТЬ ЗАГРУЗКУ ДАННЫХ

- 10.2.1 РЕАЛИЗОВАТЬ КОМАНДУ ЗАГРУЗКИ ПРЕДМЕТНОЙ ОБЛАСТИ С ИСПОЛЬЗОВАНИЕМ СЕРИАЛИЗАЦИИ
- 10.2.2 РЕАЛИЗОВАТЬ КОМАНДУ ЗАГРУЗКИ ПРЕДМЕТНОЙ ОБЛАСТИ С ИСПОЛЬЗОВАНИЕМ ЭКСТЕРНАЛТИЗАЦИИ И ТЕХНОЛОГИИ JAX-B В ТРАНСПОРТНОМ ФОРМАТЕ XML
- 10.2.3 РЕАЛИЗОВАТЬ КОМАНДУ ЗАГРУЗКИ ПРЕДМЕТНОЙ ОБЛАСТИ С ИСПОЛЬЗОВАНИЕМ ЭКСТЕРНАЛТИЗАЦИИ И ТЕХНОЛОГИИ JAX-B В ТРАНСПОРТНОМ ФОРМАТЕ JSON
- 10.2.4 РЕАЛИЗОВАТЬ КОМАНДУ ЗАГРУЗКИ ПРЕДМЕТНОЙ ОБЛАСТИ С ИСПОЛЬЗОВАНИЕМ ЭКСТЕРНАЛТИЗАЦИИ И БИБЛИОТЕКИ FASTERXML В ТРАНСПОРТНОМ ФОРМАТЕ XML
- 10.2.5 РЕАЛИЗОВАТЬ КОМАНДУ ЗАГРУЗКИ ПРЕДМЕТНОЙ ОБЛАСТИ С ИСПОЛЬЗОВАНИЕМ ЭКСТЕРНАЛТИЗАЦИИ И БИБЛИОТЕКИ FASTERXML В ТРАНСПОРТНОМ ФОРМАТЕ JSON

10.3 РЕКОМЕНДАЦИИ

- 10.3.1 ПРИ ИСПОЛЬЗОВАНИИ ТРАНСПОРТНЫХ ФОРМАТОВ ДАННЫХ ПРИМЕНЯТЬ PRETTY ФОРМАТИРОВАНИЕ ДАННЫХ
- 10.3.2 ПРИ ИСПОЛЬЗОВАНИИ ВНЕШНИХ ЭКСТЕРНАЛТИЗАТОРОВ ПРИМЕНЯТЬ РЕЖИМ ИГНОРИРОВАНИЯ НЕИЗВЕСТНЫХ АТРИБУТОВ БЕЗ ЖЕСТКОЙ ВАЛИДАЦИИ ДОКУМЕНТА

10.4 ПРОГРАММНЫЙ КОД

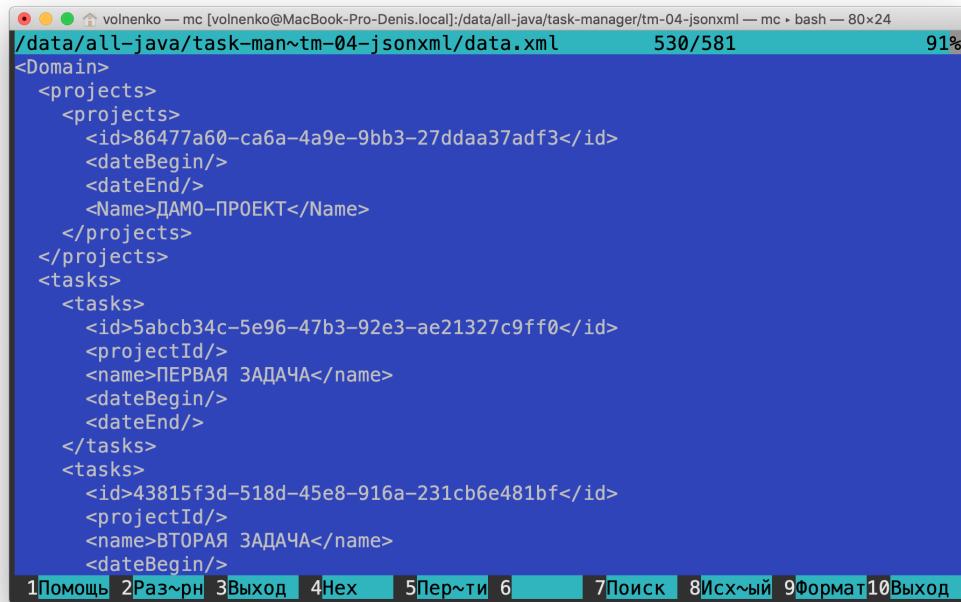
```
<dependencies>
    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-databind</artifactId>
        <version>2.9.8</version>
    </dependency>
    <dependency>
        <groupId>com.fasterxml.jackson.dataformat</groupId>
        <artifactId>jackson-dataformat-xml</artifactId>
        <version>2.9.8</version>
    </dependency>
</dependencies>
```

Рис. 16: MAVEN-ЗАВИСИМОСТИ ДЛЯ FASTERXML

```
<dependency>
    <groupId>org.eclipse.persistence</groupId>
    <artifactId>org.eclipse.persistence.moxy</artifactId>
    <version>2.6.0</version>
</dependency>
```

Рис. 17: MAVEN-ЗАВИСИМОСТИ ДЛЯ JAX-B JSON

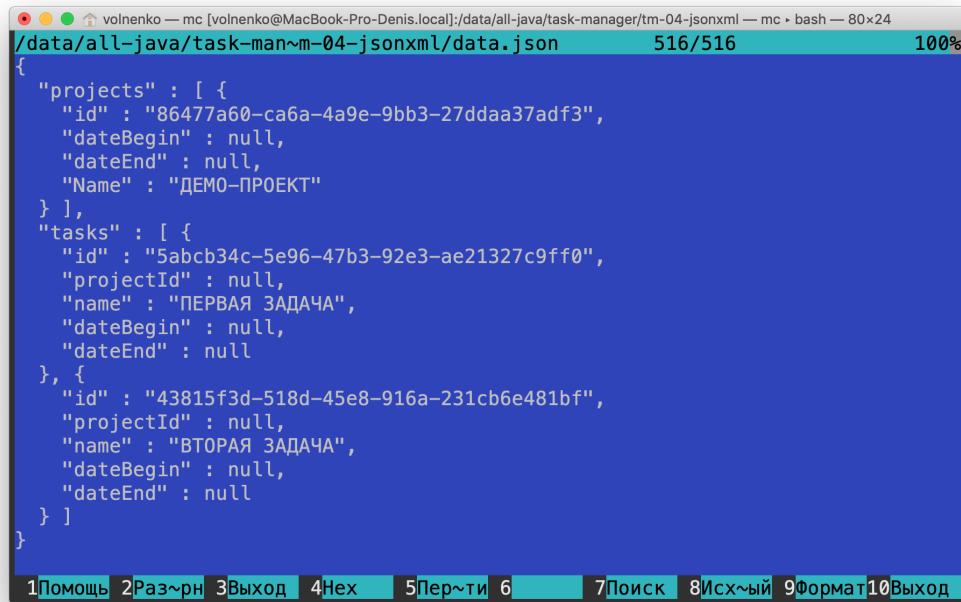
10.5 СКРИНШОТЫ



The screenshot shows a terminal window with the following details:

- Terminal title: volnenko — mc [volnenko@MacBook-Pro-Denis.local]:/data/all-java/task-manager/tm-04-jsonxml — mc > bash — 80x24
- Path: /data/all-java/task-man~tm-04-jsonxml/data.xml
- File size: 530/581
- Progress: 91%
- Content: XML dump of task manager data. It includes a single project named "ДАМО-ПРОЕКТ" with two tasks: "ПЕРВАЯ ЗАДАЧА" and "ВТОРАЯ ЗАДАЧА".
- Bottom menu: 1Помощь, 2Раз~рн, 3Выход, 4Hex, 5Пер~ти, 6, 7Поиск, 8Исх~ый, 9Формат, 10Выход

Рис. 18: ДАМП ДАННЫХ В ТРАНСПОРТНОМ ФОРМАТЕ ДАННЫХ XML



The screenshot shows a terminal window with the following details:

- Terminal title: volnenko — mc [volnenko@MacBook-Pro-Denis.local]:/data/all-java/task-manager/tm-04-jsonxml — mc > bash — 80x24
- Path: /data/all-java/task-man~tm-04-jsonxml/data.json
- File size: 516/516
- Progress: 100%
- Content: JSON dump of task manager data. It includes a single project named "ДЕМО-ПРОЕКТ" with two tasks: "ПЕРВАЯ ЗАДАЧА" and "ВТОРАЯ ЗАДАЧА".
- Bottom menu: 1Помощь, 2Раз~рн, 3Выход, 4Hex, 5Пер~ти, 6, 7Поиск, 8Исх~ый, 9Формат, 10Выход

```
{
  "projects" : [ {
    "id" : "86477a60-сa6a-4a9e-9bb3-27ddaa37adf3",
    "dateBegin" : null,
    "dateEnd" : null,
    "Name" : "ДЕМО-ПРОЕКТ"
  }],
  "tasks" : [ {
    "id" : "5abcb34c-5e96-47b3-92e3-ae21327c9ff0",
    "projectId" : null,
    "name" : "ПЕРВАЯ ЗАДАЧА",
    "dateBegin" : null,
    "dateEnd" : null
  },
  {
    "id" : "43815f3d-518d-45e8-916a-231cb6e481bf",
    "projectId" : null,
    "name" : "ВТОРАЯ ЗАДАЧА",
    "dateBegin" : null,
    "dateEnd" : null
  }
}
```

Рис. 19: ДАМП ДАННЫХ В ТРАНСПОРТНОМ ФОРМАТЕ ДАННЫХ JSON

11 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-11

- 11.1 СОЗДАТЬ АГРЕГАТОР МОДУЛЕЙ ПРИЛОЖЕНИЯ
 - 11.1.1 НАСТРОИТЬ СВЯЗИ МЕЖДУ АГРЕГАТОРОМ И ДОЧЕРНИМИ МОДУЛЯМИ
 - 11.1.2 ВЫНЕСТИ ОБЩИЕ НАСТРОЙКИ СБОРКИ ПРОЕКТОВ В МОДУЛЬ АГРЕГАТОРА
- 11.2 СОЗДАТЬ МОДУЛЬ СЕРВЕРНОГО ПРИЛОЖЕНИЯ
 - 11.2.1 РЕАЛИЗОВАТЬ ЗАГРУЗЧИК И СЕРВИС ЛОКАТОР ДЛЯ СЕРВЕРНОГО ПРИЛОЖЕНИЯ
 - 11.2.2 РЕАЛИЗОВАТЬ ДЛЯ КАЖДОЙ СУЩНОСТИ ОТДЕЛЬНУЮ ТОЧКУ ДОСТУПА
 - 11.2.3 ДЛЯ РЕАЛИЗАЦИИ ТОЧЕК ДОСТУПА ИСПОЛЬЗОВАТЬ ТЕХНОЛОГИЮ JAX-WS
 - 11.2.4 РЕАЛИЗОВАТЬ ДЕЛЕГИРОВАНИЕ ВЫЗОВОВ МЕТОДОВ ИЗ СЕРВИСОВ ЧЕРЕЗ СООТВЕТСТВУЮЩИЕ ТОЧКИ ДОСТУПА
- 11.3 СОЗДАТЬ МОДУЛЬ КЛИЕНТСКОГО ПРИЛОЖЕНИЯ
 - 11.3.1 РЕАЛИЗОВАТЬ ЗАГРУЗЧИК И СЕРВИС ЛОКАТОР ДЛЯ КЛИЕНТСКОГО ПРИЛОЖЕНИЯ
 - 11.3.2 ПЕРЕНЕСТИ ТЕРМИНАЛЬНЫЕ КОМАНДЫ В КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ
 - 11.3.3 РЕАЛИЗОВАТЬ В КОНСОЛЬНЫХ КОМАНДАХ ВЫЗОВ ВЕБ-СЕРВИСОВ ВМЕСТО СЕРВИСОВ
 - 11.3.4 ПОДКЛЮЧИТЬ MAVEN-ПЛАГИН ENUNCIATE ДЛЯ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ДОКУМЕНТАЦИИ ПО WSDL
- 11.4 РЕАЛИЗОВАТЬ СЕТЕВУЮ АВТОРИЗАЦИЮ
 - 11.4.1 РЕАЛИЗОВАТЬ ХРАНЕНИЕ СЕССИИ В ВИДЕ СУЩНОСТИ
 - 11.4.2 РЕАЛИЗОВАТЬ ПОДПИСЬ СЕССИИ НА СЕРВЕРЕ
 - 11.4.3 РЕАЛИЗОВАТЬ ПРОВЕРКУ СЕССИИ И РОЛЕЙ НЕ СЕРВЕРЕ
- 11.5 РЕКОМЕНДАЦИИ
 - 11.5.1 НЕОБХОДИМО ИСПОЛЬЗОВАТЬ УНИКАЛЬНЫЕ НАЗВАНИЯ МЕТОДОВ В ТОЧКАХ ДОСТУПА В РАМКАХ ПРИЛОЖЕНИЯ
 - 11.5.2 КОМАНДЫ СОХРАНЕНИЯ И ЗАГРУЗКИ ДАННЫХ ДОЛЖНЫ ВЫПОЛНЯТЬСЯ ОТ РОЛИ АДМИНИСТРАТОРА
 - 11.5.3 РЕАЛИЗОВАТЬ ПЕРЕДАЧУ ОБЪЕКТА СЕССИИ В ВЕБ-МЕТОДЕ ПЕРВЫМ ПАРАМЕТРОМ

11.6 ПРОГРАММНЫЙ КОД

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
          http://maven.apache.org/maven-v4_0_0.xsd">

    <modelVersion>4.0.0</modelVersion>
    <groupId>ru.volnenko.tm</groupId>
    <version>1.0.0</version>
    <artifactId>task-manager-application</artifactId>
    <packaging>pom</packaging>

    <modules>
        <module>tm-client</module>
        <module>tm-server</module>
    </modules>

</project>
```

Рис. 20: КОНФИГУРАЦИЯ MAVEN SUPER POM

```
<plugin>
    <groupId>com.webcohesion.enunciate</groupId>
    <artifactId>enunciate-maven-plugin</artifactId>
    <version>2.11.1</version>
    <executions>
        <execution>
            <goals>
                <goal>docs</goal>
            </goals>
            <configuration>
                <docsDir>${project.build.directory}/docs</docsDir>
            </configuration>
        </execution>
    </executions>
</plugin>
```

Рис. 21: КОНФИГУРАЦИЯ ENUNCATE MAVEN-PLUGIN

```
<plugin>
    <groupId>org.apache.cxf</groupId>
    <artifactId>cxf-codegen-plugin</artifactId>
    <version>3.2.7</version>
    <executions>
        <execution>
            <id>generate-sources</id>
            <phase>generate-sources</phase>
            <configuration>
                <sourceRoot>
                    ${project.basedir}/src/main/java
                </sourceRoot>
                <wsdlOptions>
                    <wsdlOption>
                        <wsdl>
                            http://localhost:8080/ProjectEndpoint?wsdl
                        </wsdl>
                    </wsdlOption>
                    <wsdlOption>
                        <wsdl>
                            http://localhost:8080/SessionEndpoint?wsdl
                        </wsdl>
                    </wsdlOption>
                    <wsdlOption>
                        <wsdl>
                            http://localhost:8080/TaskEndpoint?wsdl
                        </wsdl>
                    </wsdlOption>
                    <wsdlOption>
                        <wsdl>
                            http://localhost:8080/UserEndpoint?wsdl
                        </wsdl>
                    </wsdlOption>
                </wsdlOptions>
            </configuration>
            <goals><goal>wsdl2java</goal></goals>
        </execution>
    </executions>
</plugin>
```

Рис. 22: КОНФИГУРАЦИЯ ГЕНЕРАЦИИ КЛИЕНТА ПО WSDL

```
public final class PasswordHashUtil {  
  
    @Nullable  
    public static String md5(@Nullable final String md5) {  
        if (md5 == null) return null;  
        try {  
            @NotNull final java.security.MessageDigest md =  
                java.security.MessageDigest.getInstance("MD5");  
            @NotNull final byte[] array = md.digest(md5.getBytes());  
            @NotNull final StringBuilder sb = new StringBuilder();  
            for (int i = 0; i < array.length; ++i) {  
                sb.append(Integer.toHexString((array[i] & 0xFF) | 0x100)  
                    .substring(1, 3));  
            }  
            return sb.toString();  
        } catch (java.security.NoSuchAlgorithmException e) {  
            e.printStackTrace();  
        }  
        return null;  
    }  
}
```

Рис. 23: ФУНКЦИЯ MD5-ХЕШИРОВАНИЯ

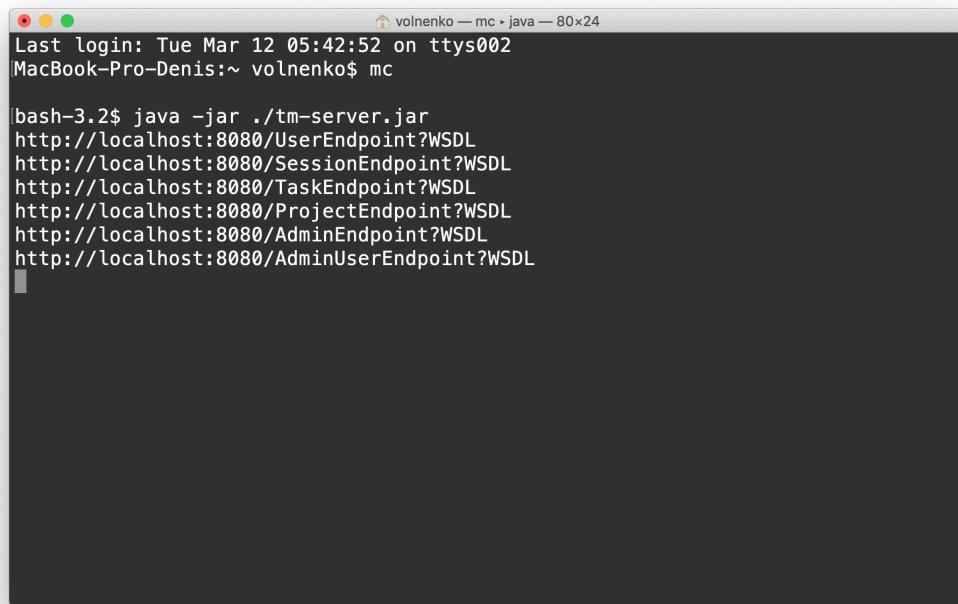
```
public final class SignatureUtil {

    @Nullable
    public static String sign(
        @Nullable final Object value,
        @Nullable final String salt,
        @Nullable final Integer cycle
    ) {
        try {
            @NotNull final ObjectMapper objectMapper =
                new ObjectMapper();
            @NotNull final String json =
                objectMapper.writeValueAsString(value);
            return sign(json, salt, cycle);
        } catch (final JsonProcessingException e) {
            return null;
        }
    }

    @Nullable
    public static String sign(
        @Nullable final String value,
        @Nullable final String salt,
        @Nullable final Integer cycle
    ) {
        if (value == null || salt == null || cycle == null) return null;
        @Nullable String result = value;
        for (int i = 0; i < cycle; i++) {
            result = PasswordHashUtil.md5(salt + result + salt);
        }
        return result;
    }
}
```

Рис. 24: ФУНКЦИЯ ЦИФРОВОЙ ПОДПИСИ ДАННЫХ

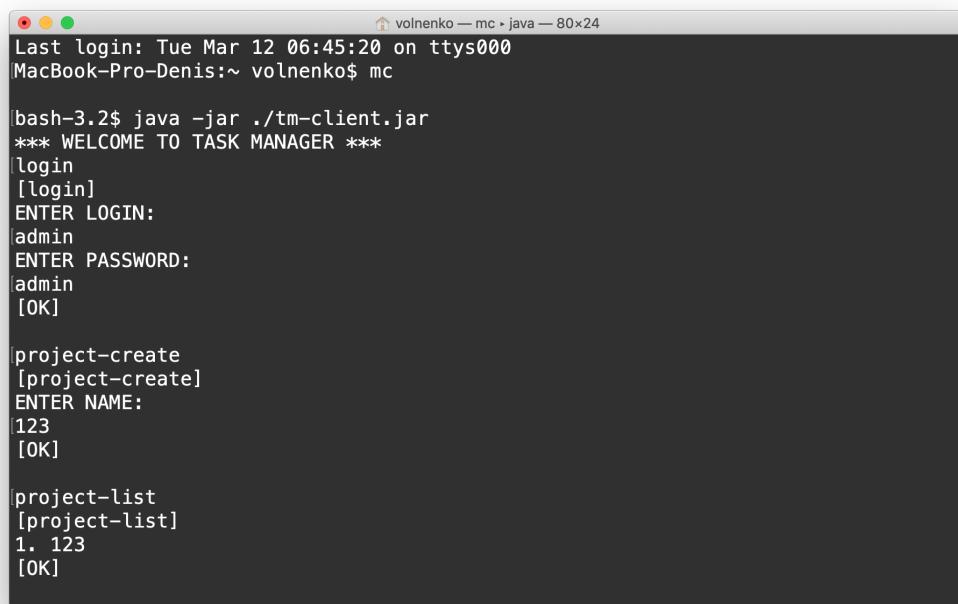
11.7 СКРИНШОТЫ



```
Last login: Tue Mar 12 05:42:52 on ttys002
[MacBook-Pro-Denis:~ volnenko$ mc

bash-3.2$ java -jar ./tm-server.jar
http://localhost:8080/UserEndpoint?WSDL
http://localhost:8080/SessionEndpoint?WSDL
http://localhost:8080/TaskEndpoint?WSDL
http://localhost:8080/ProjectEndpoint?WSDL
http://localhost:8080/AdminEndpoint?WSDL
http://localhost:8080/AdminUserEndpoint?WSDL
```

Рис. 25: СЕРВЕРНОЕ КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ



```
Last login: Tue Mar 12 06:45:20 on ttys000
[MacBook-Pro-Denis:~ volnenko$ mc

bash-3.2$ java -jar ./tm-client.jar
*** WELCOME TO TASK MANAGER ***
[login]
[login]
ENTER LOGIN:
[admin]
ENTER PASSWORD:
[admin]
[OK]

project-create
[project-create]
ENTER NAME:
[123]
[OK]

project-list
[project-list]
1. 123
[OK]
```

Рис. 26: КЛИЕНТСКОЕ КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ

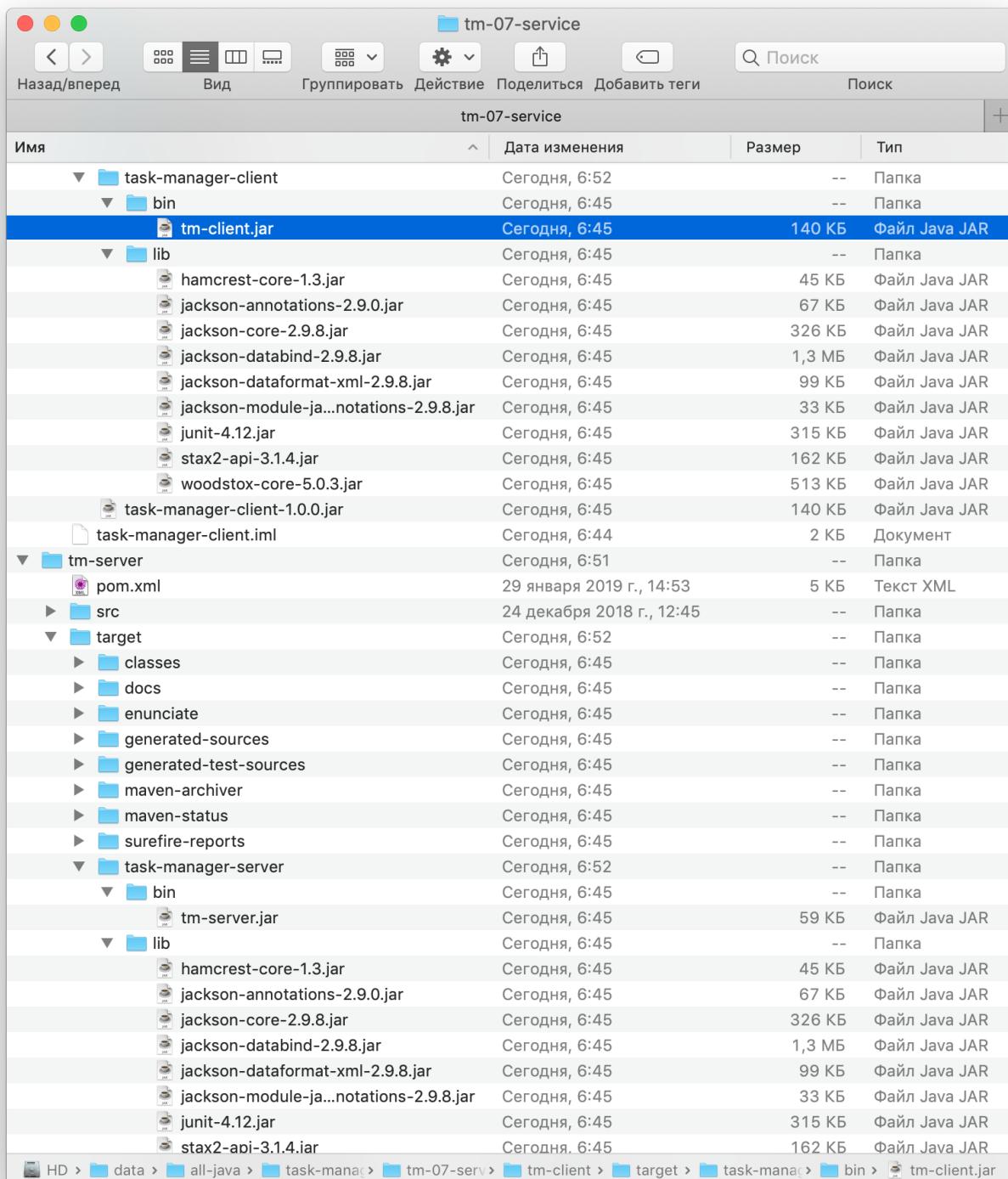
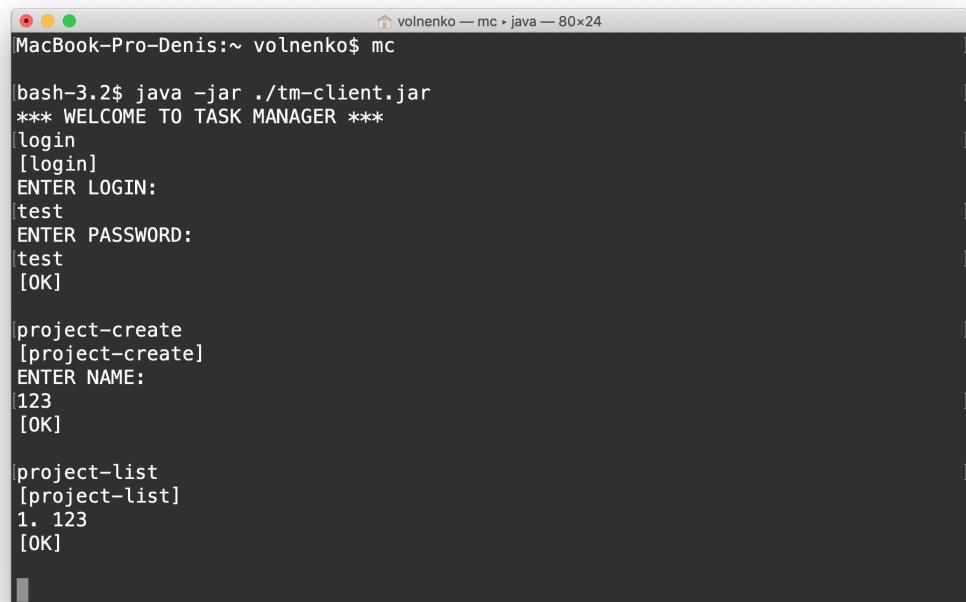


Рис. 27: РЕЛИЗНЫЕ ПАПКИ КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ

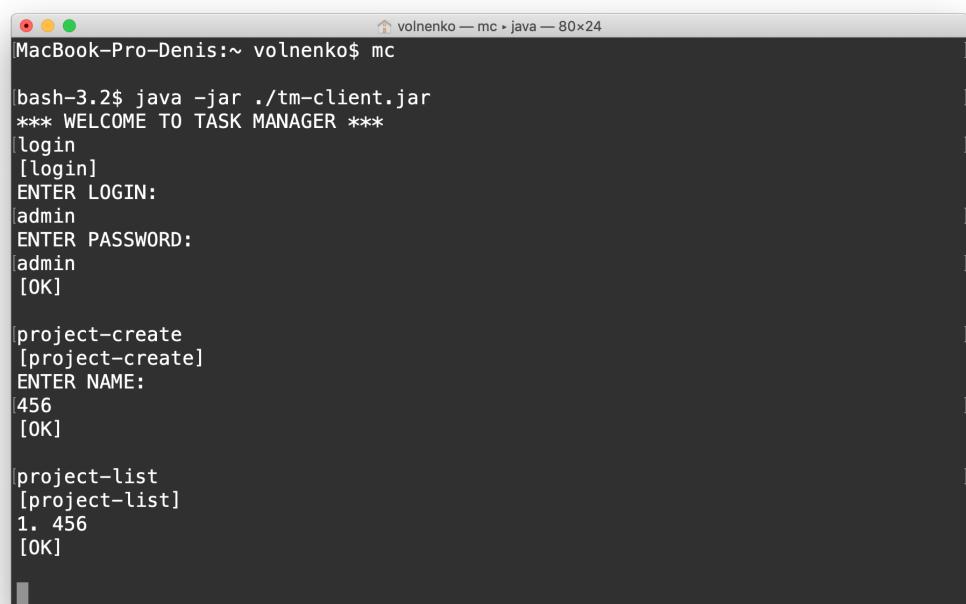


```
MacBook-Pro-Denis:~ volnenko$ mc
[bash-3.2$ java -jar ./tm-client.jar
*** WELCOME TO TASK MANAGER ***
[login
[login]
ENTER LOGIN:
[test
ENTER PASSWORD:
[test
[OK]

[project-create
[project-create]
ENTER NAME:
[123
[OK]

[project-list
[project-list]
1. 123
[OK]
```

Рис. 28: СЕАНС ТЕСТОВОГО ПОЛЬЗОВАТЕЛЯ



```
MacBook-Pro-Denis:~ volnenko$ mc
[bash-3.2$ java -jar ./tm-client.jar
*** WELCOME TO TASK MANAGER ***
[login
[login]
ENTER LOGIN:
[admin
ENTER PASSWORD:
[admin
[OK]

[project-create
[project-create]
ENTER NAME:
[456
[OK]

[project-list
[project-list]
1. 456
[OK]
```

Рис. 29: СЕАНС АДМИНИСТРАТОРА ПРИЛОЖЕНИЯ

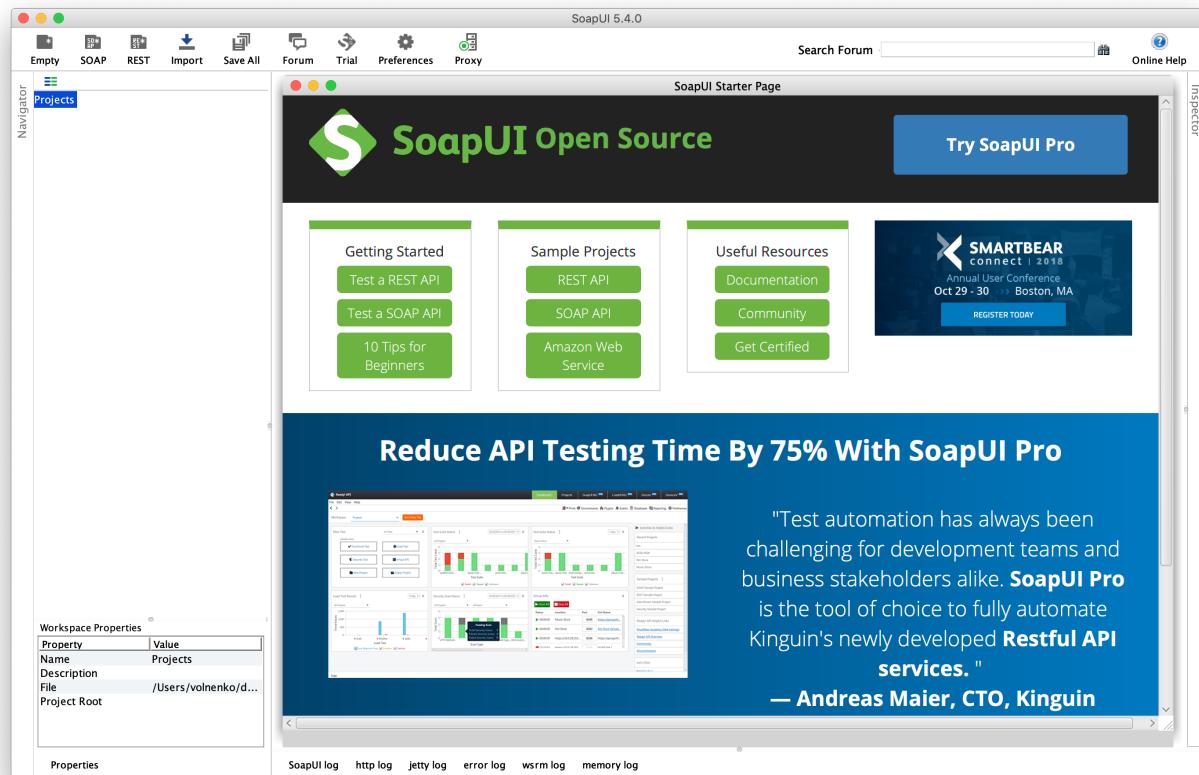


Рис. 30: ВНЕШНИЙ КЛИЕНТ SOAPUI

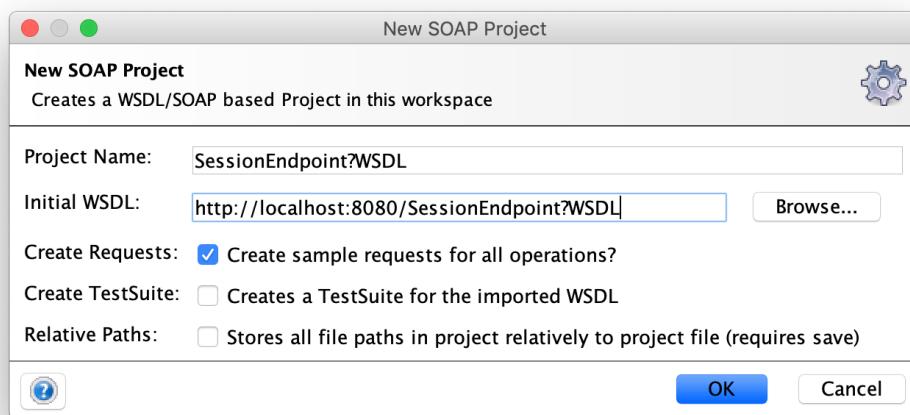


Рис. 31: ИМПОРТ WSDL-КОНТРАКТА ДЛЯ ВЕБ-СЕРВИСА СЕССИЙ

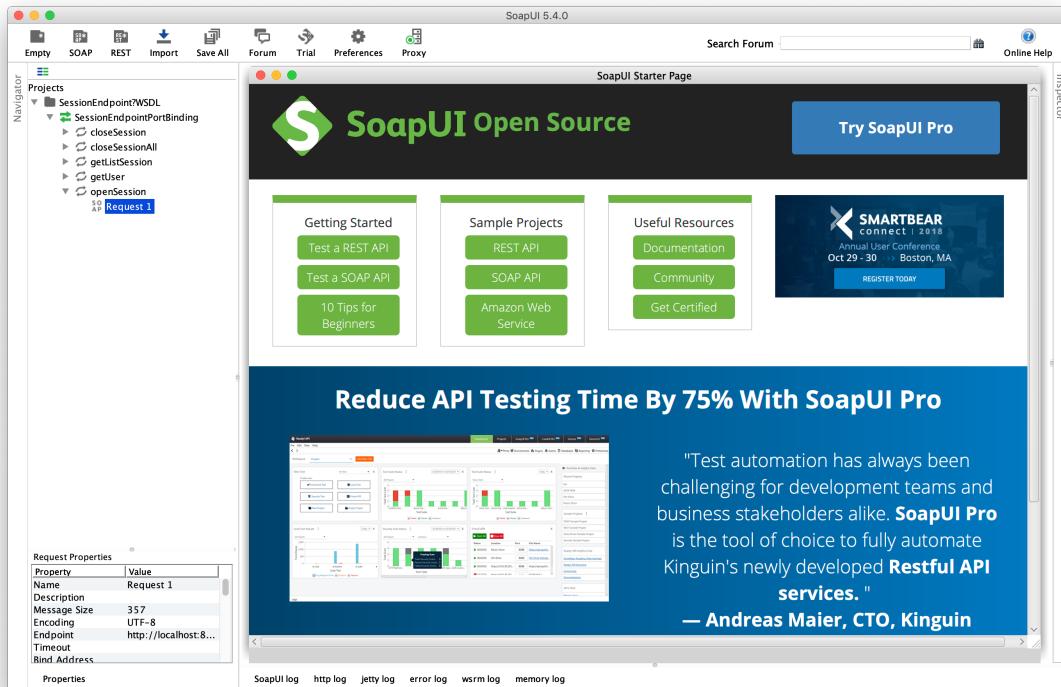


Рис. 32: ВЫЗОВ ВЕБ-ОПЕРАЦИИ ОТКРЫТИЯ СЕССИИ

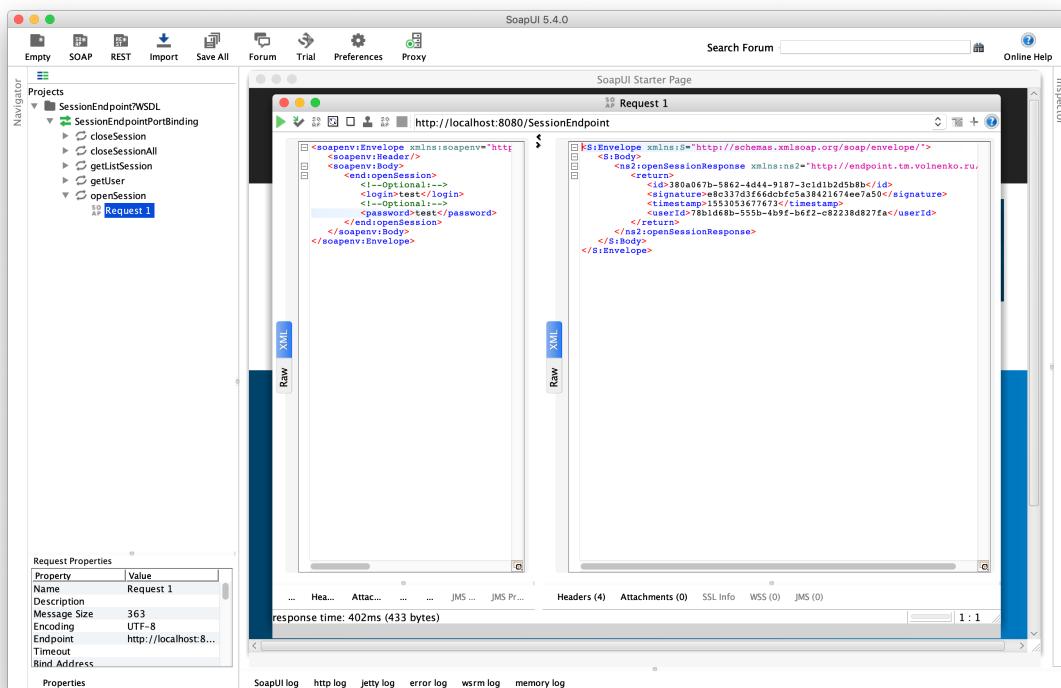


Рис. 33: ПРИМЕР ВЫЗОВА ОТКРЫТИЯ СЕССИИ

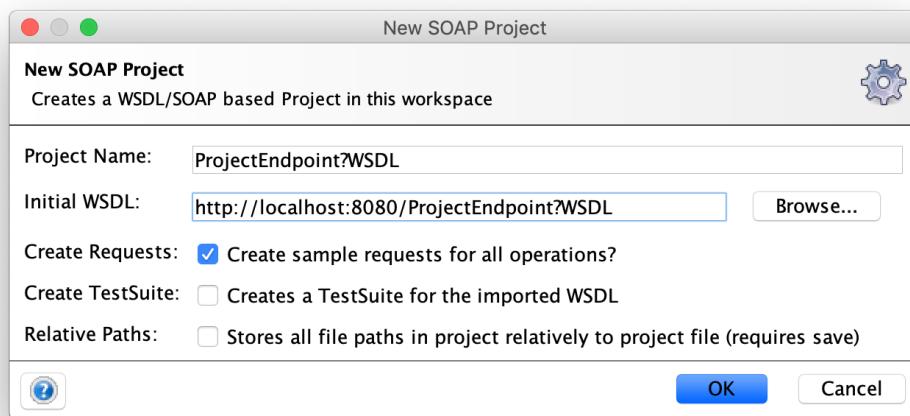


Рис. 34: ИМПОРТ WSDL-КОНТРАКТА ДЛЯ ВЕБ-СЕРВИСА ПРОЕКТОВ

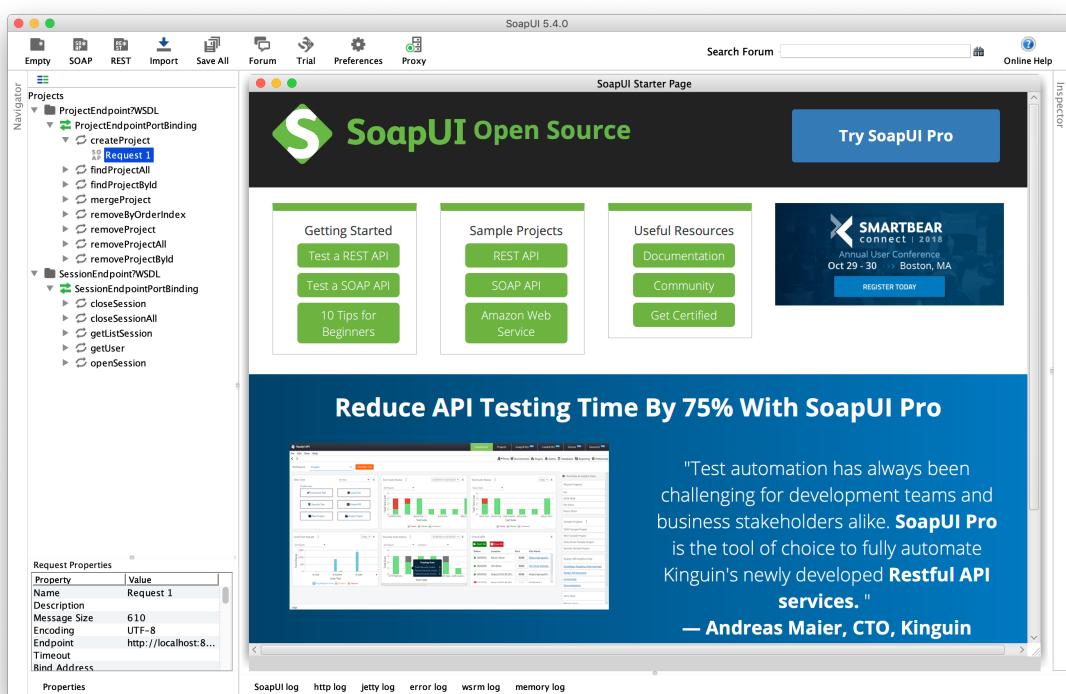


Рис. 35: ВЫЗОВ ВЕБ-ОПЕРАЦИИ СОЗДАНИЯ ПРОЕКТА

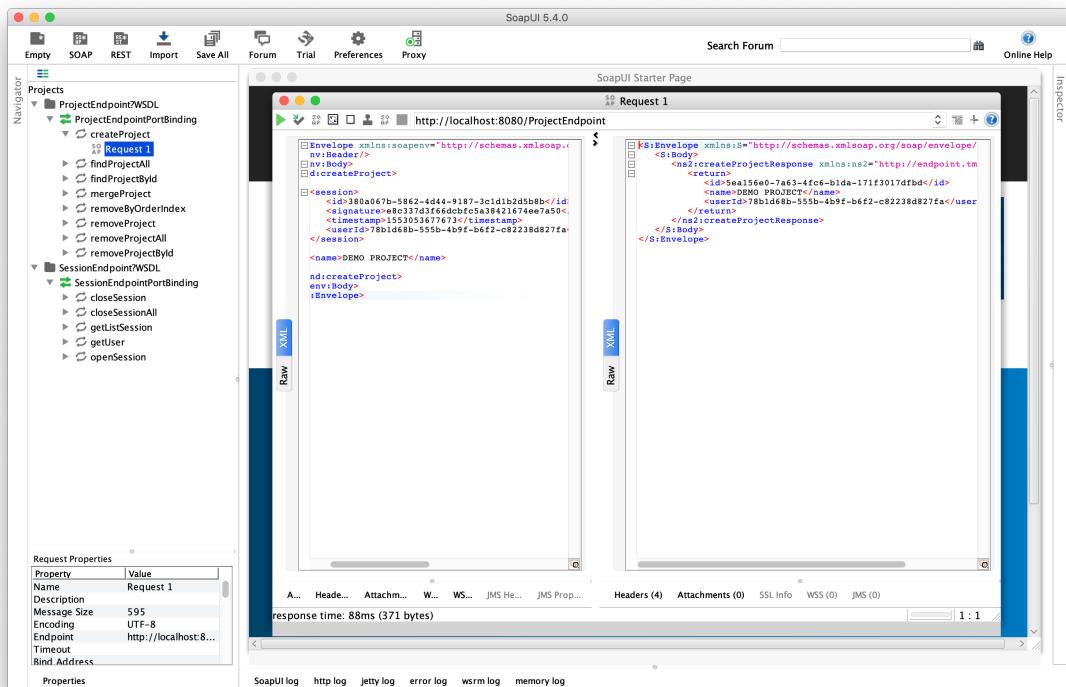


Рис. 36: ПРИМЕР ВЫЗОВА СОЗДАНИЯ ПРОЕКТА

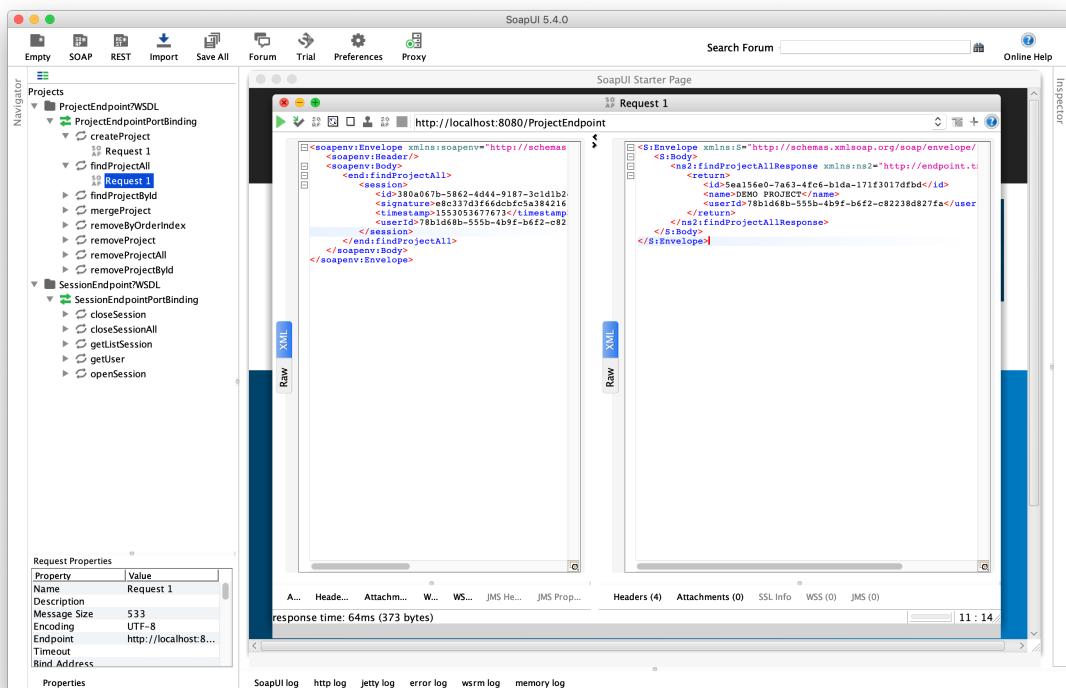


Рис. 37: ПРИМЕР ВЫЗОВА СПИСКА ПРОЕКТОВ ПОЛЬЗОВАТЕЛЯ

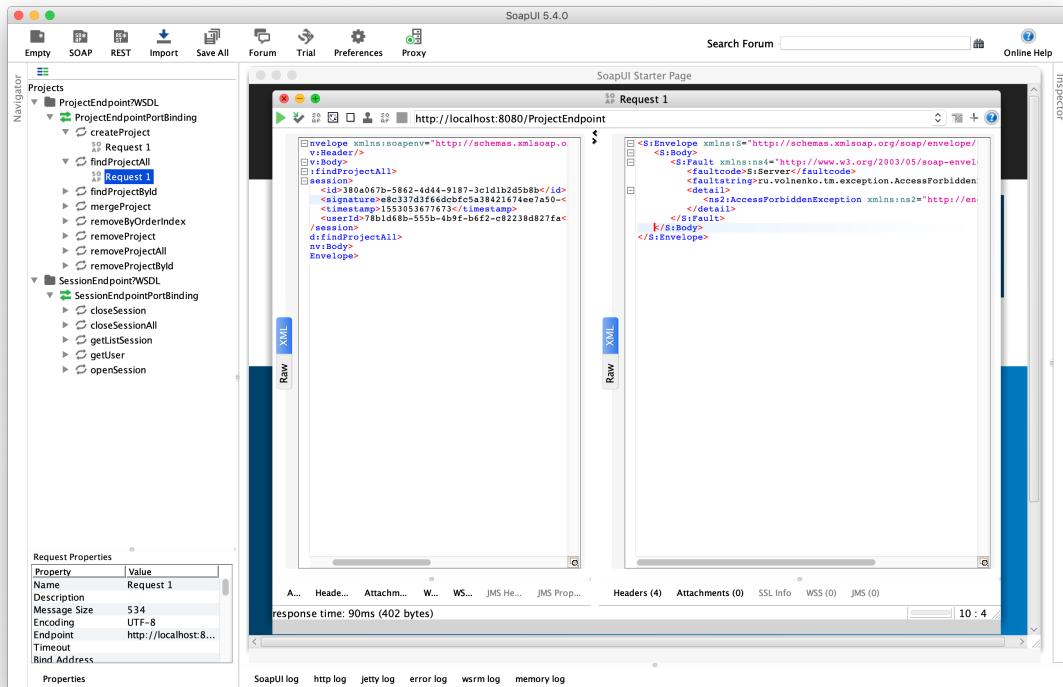


Рис. 38: ПРИМЕР ОТКАЗА ДОСТУПА ПО СЕССИИ

The screenshot shows a web-based interface for managing data types. The title bar reads "Task Manager Server: Data Types". The main content area is titled "XML" and contains a table with the following data:

type	description
fail	
result	
success	
abstractEntity	
project	
session	
task	
user	
role	

Рис. 39: ОПИСАНИЕ ТИПОВ ДАННЫХ ОТ ENUNCATE

The screenshot shows a web-based interface for managing services. The title bar reads "Task Manager Server: Services". The main content area is titled "Services" and contains two sections of tables:

Namespace <http://endpoint.api.tm.volnenko.ru/> (wsdl)

name	description
IAdminUserEndpoint	
IProjectEndpoint	
ISessionEndpoint	
ITaskEndpoint	
IUserEndpoint	

Namespace <http://endpoint.tm.volnenko.ru/> (wsdl)

name	description
------	-------------

Рис. 40: ОПИСАНИЕ ВЕБ-СЕРВИСОВ ОТ ENUNCATE

12 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-12

12.1 РЕАЛИЗОВАТЬ ХРАНЕНИЕ ДАННЫХ В СУБД

- 12.1.1 ПОДКЛЮЧИТЬ JDBC-КОННЕКТОР В КАЧЕСТВЕ ЗАВИСИМОСТИ В MAVEN-ПРОЕКТЕ
- 12.1.2 СОЗДАТЬ В СУБД ОТДЕЛЬНУЮ БД ДЛЯ ПРИЛОЖЕНИЯ
- 12.1.3 СОЗДАТЬ ТАБЛИЦЫ ДЛЯ ХРАНЕНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ (ПОЛЬЗОВАТЕЛЬ, ЗАДАЧА, ПРОЕКТ И СЕССИЯ)
- 12.1.4 ПРОИЗВЕСТИ РЕФАКТОРИНГ РЕПОЗИТОРИЯ ДЛЯ ПЕРЕКЛЮЧЕНИЯ НА РАБОТУ С СУБД
- 12.1.5 РЕАЛИЗОВАТЬ ХРАНЕНИЕ ТЕКУЩЕГО СОЕДИНЕНИЕ С СУБД В ЗАГРУЗЧИКЕ ПРИЛОЖЕНИЯ
- 12.1.6 РЕАЛИЗОВАТЬ УПРАВЛЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТЬЮ С ИСПОЛЬЗОВАНИЕМ SQL-КОНСТРУКЦИЙ
- 12.1.7 РЕАЛИЗОВАТЬ УТИЛИТНЫЙ КЛАСС ДЛЯ УСТАНОВЛЕНИЯ СОЕДИНЕНИЯ С СУБД

12.2 РЕКОМЕНДАЦИИ

- 12.2.1 ПРИ СОЗДАНИИ ТАБЛИЦ УВЕДИТЬСЯ В ИСПОЛЬЗОВАНИИ ТРАНЗАКЦИОННОГО ДВИЖКА СУБД (INNODB)
- 12.2.2 ПРИ СОЗДАНИИ ТАБЛИЦ УСТАНОВИТЬ НЕОБХОДИМЫЕ КЛЮЧИ ДЛЯ ОБЕСПЕЧЕНИЯ ЦЕЛОСТНОСТИ ДАННЫХ
- 12.2.3 ОРГАНИЗОВАТЬ ХРАНЕНИЕ НАСТРОЕК ПОДКЛЮЧЕНИЯ К СУБД В PROPERTY-ФАЙЛЕ
- 12.2.4 ИСПОЛЬЗОВАТЬ СУБД MYSQL ВЕРСИИ 5.5
- 12.2.5 ИСПОЛЬЗОВАТЬ ГРАФИЧЕСКИЙ КЛИЕНТ HEIDYSQL
- 12.2.6 ИСПОЛЬЗОВАТЬ КОННЕКТОР MYSQL 5.1.4
- 12.2.7 ИСПОЛЬЗОВАТЬ КОДИРОВКУ UTF8_GENERAL_CI
- 12.2.8 ПЕРЕЧИСЛЕНИЯ ХРАНИТЬ В СУБД В ВИДЕ СТРОКИ
- 12.2.9 КЛАСС DATE ХРАНИТЬ В СУБД В ВИДЕ DATETIME
- 12.2.10 ИСПОЛЬЗОВАТЬ ЗНАЧЕНИЕ UUID В КАЧЕСТВЕ КЛЮЧА
- 12.2.11 ВСЕ СУЩНОСТИ И ТАБЛИЦЫ ДОЛЖНЫ СОЕДИНЯТЬСЯ ПО ПОЛЮ "ID"
- 12.2.12 РЕАЛИЗОВАТЬ ХРАНЕНИЕ BOOLEAN НА БАЗЕ BIT(1) ИЛИ TINYINT(1)

12.3 ПРОГРАММНЫЙ КОД

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.47</version>
</dependency>
```

Рис. 41: MAVEN-ЗАВИСИМОСТЬ MYSQL JDBC CONNECTOR

```
@Nullable
@sneakyThrows
private Project fetch(@Nullable final ResultSet row) {
    if (row == null) return null;
    @NotNull final Project project = new Project();
    project.setId(row.getString(FieldConst.ID));
    project.setName(row.getString(FieldConst.NAME));
    project.setDescription(row.getString(FieldConst.DESCRIPTION));
    project.setDateBegin(row.getDate(FieldConst.DATE_BEGIN));
    project.setDateEnd(row.getDate(FieldConst.DATE_END));
    return project;
}

@NotNull
@sneakyThrows
public List<Project> getListProjectByUserId(final String userId) {
    @NotNull final String query =
        "SELECT * FROM `app_project` WHERE `user_id` = ?";
    @NotNull final PreparedStatement statement =
        getConnection().prepareStatement(query);
    statement.setString(1, userId);
    @NotNull final ResultSet resultSet = statement.executeQuery();
    @NotNull final List<Project> result = new ArrayList<>();
    while (resultSet.next()) result.add(fetch(resultSet));
    statement.close();
    return result;
}
```

Рис. 42: ПРИМЕР ВЫБОРКИ ЗАПИСЕЙ ИЗ MYSQL

```
CREATE TABLE `app_user` (
  `id` varchar(255) NOT NULL,
  `email` varchar(255) DEFAULT NULL,
  `firstName` varchar(255) DEFAULT NULL,
  `lastName` varchar(255) DEFAULT NULL,
  `locked` bit(1) DEFAULT NULL,
  `login` varchar(255) DEFAULT NULL,
  `middleName` varchar(255) DEFAULT NULL,
  `passwordHash` varchar(255) DEFAULT NULL,
  `phone` varchar(255) DEFAULT NULL,
  `role` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Рис. 43: DDL ТАБЛИЦЫ "ПОЛЬЗОВАТЕЛИ"

```
CREATE TABLE `app_session` (
  `id` varchar(255) NOT NULL,
  `signature` varchar(255) DEFAULT NULL,
  `timestamp` bigint(20) DEFAULT NULL,
  `user_id` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FKrrdhu5ryvprfqplat774p2n4t` (`user_id`),
  CONSTRAINT `FKrrdhu5ryvprfqplat774p2n4t` FOREIGN KEY (`user_id`)
    REFERENCES `app_user` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Рис. 44: DDL ТАБЛИЦЫ "СЕССИИ"

```

CREATE TABLE `app_project` (
  `id` varchar(255) NOT NULL,
  `dateBegin` datetime DEFAULT NULL,
  `dateEnd` datetime DEFAULT NULL,
  `description` varchar(255) DEFAULT NULL,
  `name` varchar(255) DEFAULT NULL,
  `user_id` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FKp9byv3k2r7rgg7svn3rx10a1u` (`user_id`),
  CONSTRAINT `FKp9byv3k2r7rgg7svn3rx10a1u` FOREIGN KEY (`user_id`)
    REFERENCES `app_user` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Рис. 45: DDL ТАБЛИЦЫ "ПРОЕКТЫ"

```

CREATE TABLE `app_task` (
  `id` varchar(255) NOT NULL,
  `dateBegin` datetime DEFAULT NULL,
  `dateEnd` datetime DEFAULT NULL,
  `description` varchar(255) DEFAULT NULL,
  `name` varchar(255) DEFAULT NULL,
  `project_id` varchar(255) DEFAULT NULL,
  `user_id` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FKsu3pcsyuwr6nmpcpufikq5u4` (`project_id`),
  KEY `FKkc5pwubxw7j4b0xprgdmgkrel` (`user_id`),
  CONSTRAINT `FKkc5pwubxw7j4b0xprgdmgkrel` FOREIGN KEY (`user_id`)
    REFERENCES `app_user` (`id`),
  CONSTRAINT `FKsu3pcsyuwr6nmpcpufikq5u4` FOREIGN KEY (`project_id`)
    REFERENCES `app_project` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Рис. 46: DDL ТАБЛИЦЫ "ЗАДАЧИ"

12.4 СКРИНШОТЫ

The screenshot shows the MySQL Workbench interface for the 'task-manager' database. The 'app_user' table is selected in the 'ТАБЛИЦЫ' (Tables) list. The main pane displays the table structure with columns: id, email, firstName, lastName, locked, login, middleName, passwordHash, phone, and role. The 'ИНДЕКСЫ' (Indexes) section shows a primary index on the 'id' column.

Поле	Тип	Длина	≥ 0	00...	Двчн	Null	К...	По у...	Дополн...	Кодир...	С К
id	VARCHAR	255					PRI	None	UTF-8	UTF-8	UTF-8
email	VARCHAR	255						NULL	None	UTF-8	UTF-8
firstName	VARCHAR	255						NULL	None	UTF-8	UTF-8
lastName	VARCHAR	255						NULL	None	UTF-8	UTF-8
locked	BIT	1						NULL	None		
login	VARCHAR	255						NULL	None	UTF-8	UTF-8
middleName	VARCHAR	255						NULL	None	UTF-8	UTF-8
passwordHash	VARCHAR	255						NULL	None	UTF-8	UTF-8
phone	VARCHAR	255						NULL	None	UTF-8	UTF-8
role	VARCHAR	255						NULL	None	UTF-8	UTF-8

Неуникаль...	Имя ключа	Порядок поля	Имя колонки	Сравне...	Размер	Обрабо...	Сжатие	Комментарий
0	PRIMARY	1	id	A	2	NULL	NULL	

Рис. 47: ТАБЛИЦА "ПОЛЬЗОВАТЕЛЬ"

The screenshot shows the MySQL Workbench interface for the 'task-manager' database. The 'app_session' table is selected in the 'ТАБЛИЦЫ' (Tables) list. The main pane displays the table structure with columns: id, signature, timestamp, and user_id. The 'ИНДЕКСЫ' (Indexes) section shows two indexes: a primary index on 'id' and a foreign key index on 'user_id'.

Поле	Тип	Длина	≥ 0	00...	Двчн	Null	К...	По у...	Дополн...	Кодир...	С К
id	VARCHAR	255					PRI	None	UTF-8	UTF-8	UTF-8
signature	VARCHAR	255						NULL	None	UTF-8	UTF-8
timestamp	BIGINT	20						NULL	None		
user_id	VARCHAR	255						M...	NULL	None	UTF-8

Неуникаль...	Имя ключа	Порядок поля	Имя колонки	Сравне...	Размер	Обрабо...	Сжатие	Комментарий
0	PRIMARY	1	id	A	0	NULL	NULL	
1	FKrrdhuh5...	1	user_id	A	0	NULL	NULL	

Рис. 48: ТАБЛИЦА "СЕССИЯ"

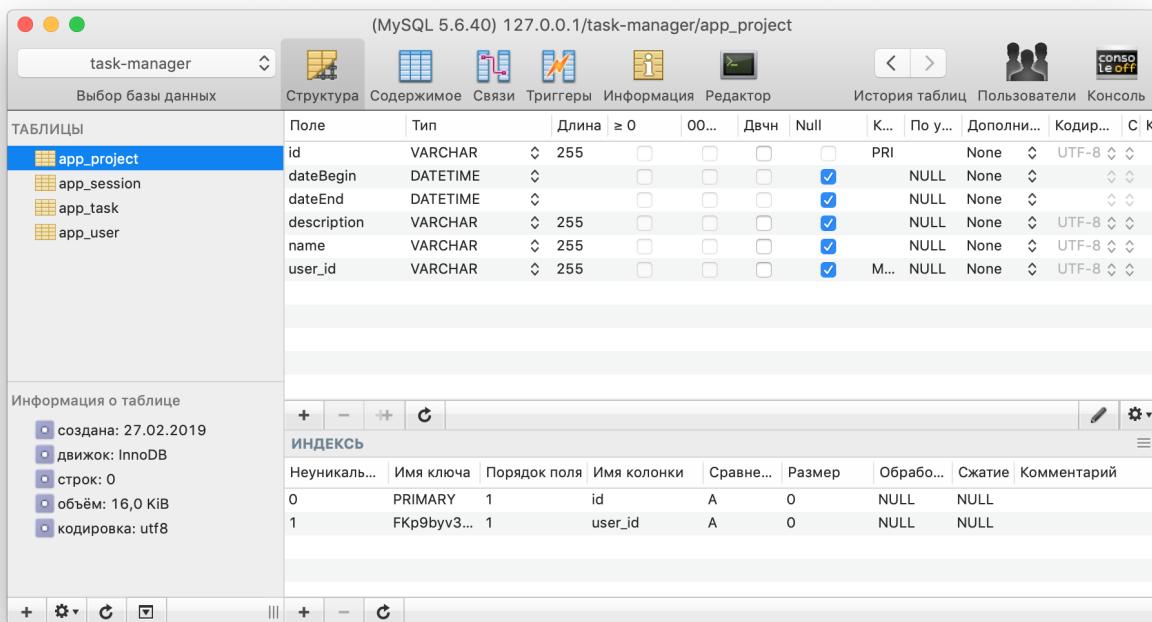


Рис. 49: ТАБЛИЦА "ПРОЕКТ"

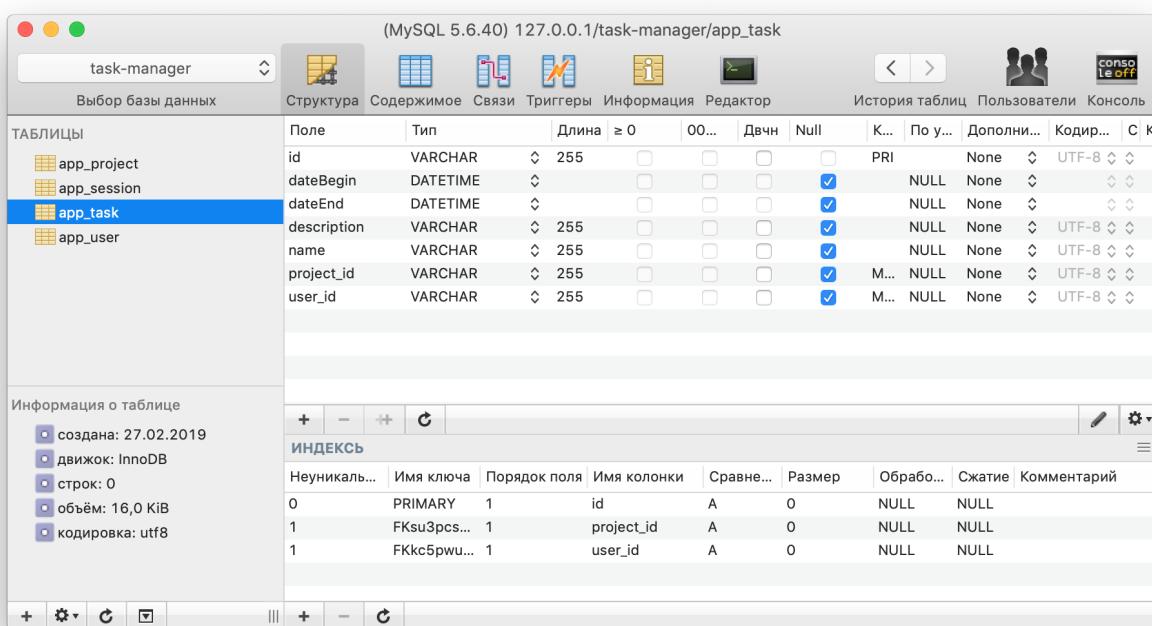


Рис. 50: ТАБЛИЦА "ЗАДАЧА"

13 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-13

13.1 ВНЕДРИТЬ ТЕХНОЛОГИЮ MYBATIS

- 13.1.1 ПОДКЛЮЧИТЬ ПОСЛЕДНЮЮ ВЕРСИЮ MYBATIS В КАЧЕСТВЕ ЗАВИСИМОСТИ
- 13.1.2 ИСПОЛЬЗОВАТЬ JAVA BASED КОНФИГУРАЦИЮ ДЛЯ СЕССИИ MYBATIS
- 13.1.3 РЕАЛИЗОВАТЬ РЕПОЗИТОРИИ НА БАЗЕ МАППЕРОВ MYBATIS
- 13.1.4 ПЕРЕНЕСТИ ВСЕ SQL-КОНСТРУКЦИИ В АННОТАЦИИ MYBATIS

13.2 РЕКОМЕНДАЦИИ

- 13.2.1 ВЫПОЛНЯТЬ COMMIT СЕССИИ MYBATIS ДЛЯ ВСЕХ ОПЕРАЦИЙ НА ЗАПИСЬ ДЛЯ ФИКСАЦИИ ИЗМЕНЕНИЙ
- 13.2.2 РЕАЛИЗОВАТЬ РУЧНОЕ УПРАВЛЕНИЕ ТРАНЗАКЦИЕЙ НА УРОВНЕ СЕРВИСОВ
- 13.2.3 ВЫПОЛНЯТЬ ROLLBACK ТРАНЗАКЦИИ ПРИ ВЫПАДЕНИИ ИСКЛЮЧЕНИЯ В БЛОКЕ TRY
- 13.2.4 ПРИ НЕСОВПАДЕНИИ НАЗВАНИЙ ПОЛЕЙ В СУЩНОСТЯХ И ТАБЛИЦАХ ИСПОЛЬЗОВАТЬ АННОТАЦИЮ RESULT
- 13.2.5 РЕПОЗИТОРИИ - МАППЕРЫ ДОЛЖНЫ БЫТЬ РЕАЛИЗОВАНЫ В ВИДЕ ИНТЕРФЕЙСОВ

13.3 ПРОГРАММНЫЙ КОД

```
public SqlSessionFactory getSqlSessionFactory() {  
    @Nullable final String user = propertyService.getJdbcUsername();  
    @Nullable final String password = propertyService.getJdbcPassword();  
    @Nullable final String url = propertyService.getJdbcUrl();  
    @Nullable final String driver = propertyService.getJdbcDriver();  
  
    final DataSource dataSource =  
        new PooledDataSource(driver, url, user, password);  
  
    final TransactionFactory transactionFactory =  
        new JdbcTransactionFactory();  
  
    final Environment environment =  
        new Environment("development", transactionFactory, dataSource);  
  
    final Configuration configuration = new Configuration(environment);  
    configuration.addMapper(UserRepository.class);  
    configuration.addMapper(ProjectRepository.class);  
    configuration.addMapper(SessionRepository.class);  
    configuration.addMapper(TaskRepository.class);  
    return new SqlSessionFactoryBuilder().build(configuration);  
}
```

Рис. 51: JAVA BASED КОНФИГУРАЦИЯ MYBATIS

```
<dependency>  
    <groupId>org.mybatis</groupId>  
    <artifactId>mybatis</artifactId>  
    <version>RELEASE</version>  
</dependency>
```

Рис. 52: MAVEN-ЗАВИСИМОСТЬ MYBATIS

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration
    PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>

    <environments default="development">
        <environment id="development">
            <transactionManager type="JDBC"/>
            <dataSource type="POOLED">
                <property name="driver" value="com.mysql.jdbc.Driver"/>
                <property name="url"
                    value="jdbc:mysql://localhost:3306/task-manager"/>
                <property name="username" value="root"/>
                <property name="password" value="" />
            </dataSource>
        </environment>
    </environments>

    <mappers>
        <mapper class="ru.volnenko.tm.repository.TaskRepository"/>
        <mapper class="ru.volnenko.tm.repository.ProjectRepository"/>
    </mappers>

</configuration>
```

Рис. 53: XML BASED КОНФИГУРАЦИЯ MYBATIS

14 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-14

14.1 ВНЕДРИТЬ В ПРИЛОЖЕНИЕ HIBERNATE

- 14.1.1 ТЕКУЩИЕ СУЩНОСТИ ПЕРЕВЕСТИ В ОБЪЕКТЫ DTO ДЛЯ СОХРАНЕНИЯ ТРАНСПОРТА ДАННЫХ ПО SOAP
- 14.1.2 СОЗДАТЬ ГРАФЫ СУЩНОСТЕЙ С ИСПОЛЬЗОВАНИЕМ JPA
- 14.1.3 СОЗДАТЬ JAVA BASED КОНФИГУРАЦИЮ СЕССИИ HIBERNATE
- 14.1.4 РЕАЛИЗОВАТЬ КАСКАДНОЕ УДАЛЕНИЕ ГРАФА СУЩНОСТИ "ПРОЕКТ"
- 14.1.5 РЕАЛИЗОВАТЬ УПРАВЛЕНИЕ ТРАНЗАКЦИЕЙ HIBERNATE НА УРОВНЕ СЕРВИСОВ
- 14.1.6 РЕАЛИЗОВАТЬ ПРЕОБРАЗОВАНИЕ СУЩНОСТЕЙ В КЛАССЫ DTO НА СЛОЕ ПРЕЗЕНТАЦИИ ДАННЫХ

14.2 РЕКОМЕНДАЦИИ

- 14.2.1 ИСПОЛЬЗОВАТЬ ИНТЕРФЕЙСЫ JPA ВМЕСТО РЕАЛИЗАЦИИ HIBERNATE
- 14.2.2 ИСПОЛЬЗОВАТЬ ИНТЕРФЕЙС "EntityManagerFactory" ВМЕСТО "SessionFactory"
- 14.2.3 ИСПОЛЬЗОВАТЬ ИНТЕРФЕЙС "EntityManager" ВМЕСТО "Session"
- 14.2.4 РЕАЛИЗОВАТЬ ХРАНЕНИЕ ПЕРЕЧИСЛЕНИЙ В ВИДЕ СТРОКИ С ИСПОЛЬЗОВАНИЕМ АННОТАЦИИ "@Enumerated"
- 14.2.5 РЕАЛИЗОВАТЬ ПРОВЕРКУ НА УНИКАЛЬНОСТЬ ЛОГИНА С ИСПОЛЬЗОВАНИЕМ АННОТАЦИИ "@Column"
- 14.2.6 ВЫНЕСТИ ОБЩИЕ ПОЛЯ СУЩНОСТЕЙ В АБСТРАКТНЫЙ КЛАСС СУЩНОСТИ С ИСПОЛЬЗОВАНИЕМ "@MappedSuperclass"
- 14.2.7 ИСПОЛЬЗОВАТЬ СРЕДСТВА АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ СХЕМЫ СУБД ПО СТРАТЕГИИ ОБНОВЛЕНИЯ
- 14.2.8 ИСПОЛЬЗОВАТЬ ОТОБРАЖЕНИЕ SQL-КОНСТРУКЦИЙ ПРИ ВЫЗОВЕ HIBERNATE НА ЧТЕНИЕ / ЗАПИСЬ
- 14.2.9 СОЗДАВАТЬ РЕПОЗИТОРИЙ И ЭКЗЕМПЛЯР ENTITY MANAGER НА КАЖДЫЙ ВЫЗОВ СЕРВИСА

14.3 ПРОГРАММНЫЙ КОД

```
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.4.0.Final</version>
    <scope>compile</scope>
</dependency>
```

Рис. 54: MAVEN-ЗАВИСИМОСТЬ HIBERNATE CORE

```
private EntityManagerFactory factory() {
    final Map<String, String> settings = new HashMap<>();
    settings.put(Environment.DRIVER, propertyService.getJdbcDriver());
    settings.put(Environment.URL, propertyService.getJdbcUrl());
    settings.put(Environment.USER, propertyService.getJdbcUsername());
    settings.put(Environment.PASS, propertyService.getJdbcPassword());
    settings.put(Environment.DIALECT,
        "org.hibernate.dialect.MySQL5InnoDBDialect");
    settings.put(Environment.HBM2DDL_AUTO, "update");
    settings.put(Environment.SHOW_SQL, "true");

    final StandardServiceRegistryBuilder registryBuilder
        = new StandardServiceRegistryBuilder();
    registryBuilder.applySettings(settings);
    final StandardServiceRegistry registry = registryBuilder.build();
    final MetadataSources sources = new MetadataSources(registry);
    sources.addAnnotatedClass(Task.class);
    sources.addAnnotatedClass(Project.class);
    sources.addAnnotatedClass(User.class);
    sources.addAnnotatedClass(Session.class);
    sources.addAnnotatedClass(Cat.class);
    final Metadata metadata = sources.getMetadataBuilder().build();
    return metadata.getSessionFactoryBuilder().build();
}
```

Рис. 55: JAVA BASED КОНФИГУРАЦИЯ HIBERNATE

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
    http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd"
    version="2.1">
    <persistence-unit name="ENTERPRISE"
        transaction-type="RESOURCE_LOCAL">
        <provider>
            org.hibernate.jpa.HibernatePersistenceProvider
        </provider>
        <class>ru.volnenko.tm.model.User</class>
        <class>ru.volnenko.tm.model.Task</class>
        <class>ru.volnenko.tm.model.Project</class>
        <class>ru.volnenko.tm.model.Session</class>
        <properties>
            <property name="javax.persistence.jdbc.driver"
                value="com.mysql.jdbc.Driver"/>
            <property name="javax.persistence.jdbc.url"
                value="jdbc:mysql://localhost:3306/task-manager"/>
            <property name="javax.persistence.jdbc.user" value="root"/>
            <property name="javax.persistence.jdbc.password" value="" />
        </properties>
    </persistence-unit>
</persistence>
```

Рис. 56: КОНФИГУРАЦИЯ PERSISTENCE UNIT

```
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-c3p0</artifactId>
    <version>5.4.0.Final</version>
    <scope>compile</scope>
</dependency>
```

Рис. 57: MAVEN-ЗАВИСИМОСТЬ HIBERNATE C3P0

15 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-15

15.1 ВНЕДРИТЬ В ПРИЛОЖЕНИЕ SE WELD-CDI

15.1.1 РЕАЛИЗОВАТЬ ВНЕДРЕНИЕ ФАБРИКИ HIBERNATE С ИСПОЛЬЗОВАНИЕМ PRODUCER

15.1.2 РЕАЛИЗОВАТЬ ИНИЦИАЛИЗАЦИЮ СЕРВЕРНОГО ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ КОНТЕЙНЕРА CDI

15.1.3 РЕАЛИЗОВАТЬ ИНИЦИАЛИЗАЦИЮ КЛИЕНТСКОГО ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ КОНТЕЙНЕРА CDI

15.1.4 РЕАЛИЗОВАТЬ ВНЕДРЕНИЕ ЗАВИСИМОСТЕЙ МЕЖДУ КЛАССАМИ ПРИЛОЖЕНИЯ

15.2 РАЗРАБОТАТЬ ИНТЕГРАЦИОННЫЕ ТЕСТЫ

15.2.1 РЕАЛИЗОВАТЬ ИНТЕГРАЦИОННЫЕ ТЕСТЫ ДЛЯ CRUD ОПЕРАЦИЙ В КЛИЕНТСКОМ МОДУЛЕ С ИСПОЛЬЗОВАНИЕМ JUNIT

15.2.2 ИНТЕГРАЦИОННЫЕ ТЕСТЫ ДОЛЖНЫ РАБОТАТЬ ПОД ТЕСТОВОЙ УЧЕТНОЙ ЗАПИСЬЮ И УДАЛЯТЬ ЗА СОБОЙ ДАННЫЕ ПОСЛЕ ВЫПОЛНЕНИЯ ТЕСТИРОВАНИЯ ВЫЗОВОВ ТОЧЕК ДОСТУПА

15.2.3 ИНТЕГРАЦИОННЫЕ ТЕСТЫ ДОЛЖНЫ ЗАПУСКАТЬСЯ ПОД ОТДЕЛЬНЫМ MAVEN-ПРОФИЛЕМ ДЛЯ СОХРАНЕНИЯ СТАБИЛЬНОСТИ СБОРКИ БЕЗ ЗАПУЩЕННОГО СЕРВЕРНОГО ПРИЛОЖЕНИЯ

15.3 РЕКОМЕНДАЦИИ

15.3.1 ОТСУТСТВУЕТ ВОЗМОЖНОСТЬ РЕАЛИЗОВАТЬ ТОЧКИ ДОСТУПА С ОБЛАСТЬЮ ВИДИМОСТИ В РАМКАМ ПРИЛОЖЕНИЯ

15.3.2 РЕАЛИЗОВАТЬ КЛАССЫ СЕРВИСОВ И ЗАГРУЗЧИКИ ПРИЛОЖЕНИЯ В ВИДЕ ОДИНОЧЕК С ОБЛАСТЬЮ ВИДИМОСТИ В РАМКАХ ПРИЛОЖЕНИЯ

15.3.3 СОЗДАВАТЬ РЕПОЗИТОРИЙ И ЭКЗЕМПЛЯР ENTITY MANAGER НА КАЖДЫЙ ВЫЗОВ СЕРВИСА

15.3.4 ДОБАВИТЬ ДИСКРИПТОР РАЗВОРАЧИВАНИЯ ПРИЛОЖЕНИЙ "beans.xml" С ВКЛЮЧЕННЫМ СКАНИРОВАНИЕМ ВСЕХ КЛАССОВ БЕЗ АННОТАЦИЙ

15.3.5 РЕАЛИЗОВАТЬ СКВОЗНОЕ ВНЕДРЕНИЕ ЗАВИСИМОСТЕЙ В ТОЧКИ ДОСТУПА МОЖНО ЧЕРЕЗ ИХ ВНЕДРЕНИЕ В ЗАГРУЗЧИК ПРИЛОЖЕНИЯ ДЛЯ ПОСЛЕДУЮЩЕЙ ПУБЛИКАЦИИ

15.4 ПРОГРАММНЫЙ КОД

```
<beans xmlns="http://xmlns.jcp.org/xml/ns/javaee"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                           http://xmlns.jcp.org/xml/ns/javaee/beans_1_1.xsd"
       bean-discovery-mode="all">

</beans>
```

Рис. 58: КОНФИГУРАЦИЯ "META-INF/beans.xml"

```
public final class Application {

    public static void main(final String[] args) throws Exception {
        SeContainerInitializer.newInstance()
            .addPackages(Application.class).initialize()
            .select(Bootstrap.class).get().init();
    }

}
```

Рис. 59: ТОЧКА ВХОДА В CDI-ПРИЛОЖЕНИЕ

```
<dependency>
    <groupId>org.jboss.weld.se</groupId>
    <artifactId>weld-se-core</artifactId>
    <version>3.1.0.Final</version>
    <scope>compile</scope>
</dependency>
```

Рис. 60: MAVEN-ЗАВИСИМОСТЬ WELD-CDI

16 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-16

16.1 ВНЕДРИТЬ АПАЧЕ DELTA SPIKE

- 16.1.1 ДОБАВИТЬ ЗАВИСИМОСТИ АПАЧЕ DELTA SPIKE С ОФИЦИАЛЬНОГО ВЕБ-САЙТА В MAVEN-ПРОЕКТ
 - 16.1.2 РЕАЛИЗОВАТЬ АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ ТРАНЗАКЦИЕЙ НА УРОВНЕ СЕРВИСОВ С ИСПОЛЬЗОВАНИЕМ АННОТАЦИИ "@Transactional"
 - 16.1.3 РЕАЛИЗОВАТЬ РЕПОЗИТОРИИ В ВИДЕ ИНТЕРФЕЙСОВ С АННОТАЦИЕЙ "@Repository"
 - 16.1.4 РЕАЛИЗОВАТЬ ВНЕДРЕНИЕ "@EntityManager" С ОБЛАСТЬЮ ВИДИМОСТИ "@TransactionalScope"
- ### 16.2 РАЗРАБОТАТЬ ТЕСТЫ СЛОЕВ ПРИЛОЖЕНИЯ
- 16.2.1 РЕАЛИЗОВАТЬ ПАКЕТ ТЕСТОВ ДЛЯ РЕПОЗИТОРИЕВ И СЕРВИСОВ С ИСПОЛЬЗОВАНИЕМ H2 СУБД
- ### 16.3 ПРОГРАММНЫЙ КОД

```
<dependency>
    <groupId>org.apache.deltaspike.modules</groupId>
    <artifactId>deltaspike-data-module-impl</artifactId>
    <version>RELEASE</version>
    <scope>compile</scope>
</dependency>
```

Рис. 61: MAVEN-ЗАВИСИМОСТЬ АПАЧЕ DELTASPIKE

```
import org.apache.deltaspike.jpa.api.entitymanager.PersistenceUnitName;
import org.apache.deltaspike.jpa.api.transaction.TransactionScoped;
import org.jetbrains.annotations.NotNull;

import javax.enterprise.context.ApplicationScoped;
import javax.enterprise.inject.Disposes;
import javax.enterprise.inject.Produces;
import javax.inject.Inject;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;

@ApplicationScoped
public class EntityManagerProducer {

    @NotNull
    private static final String UNIT_NAME = "ENTERPRISE";

    @Inject
    @PersistenceUnitName(UNIT_NAME)
    private EntityManagerFactory entityManagerFactory;

    @NotNull
    @Produces
    @TransactionScoped
    public EntityManager create() {
        return this.entityManagerFactory.createEntityManager();
    }

    public void dispose(@Disposes EntityManager entityManager) {
        if (entityManager.isOpen()) entityManager.close();
    }
}
```

Рис. 62: КОНФИГУРАЦИЯ ВНЕДРЕНИЯ ENTITY MANAGER В CDI

17 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-17

17.1 ВНЕДРИТЬ В ПРОЕКТ HAZELCAST

17.1.1 ДОБАВИТЬ MAVEN-ЗАВИСИМОСТИ ДЛЯ HAZELCAST

17.1.2 ДОБАВИТЬ MAVEN-ЗАВИСИМОСТИ ДЛЯ HIBERNATE

17.2 ВНЕДРИТЬ В ПРОЕКТ JPA SECOND LEVEL CACHE

17.2.1 ДОБАВИТЬ АННОТАЦИИ ДЛЯ КЕШИРОВАНИЯ СУЩНОСТЕЙ

17.2.2 ПРОТЕСТИРОВАТЬ КЕШ ВТОРОГО УРОВНЯ

17.2.3 ПРОТЕСТИРОВАТЬ ОБЪЕДИНЕНИЕ HAZELCAST НОД В КЛАСТЕР ДЛЯ НЕСКОЛЬКИХ ЭКЗЕМПЛЯРОВ ПРИЛОЖЕНИЯ

17.3 ПРОГРАММНЫЙ КОД

```
<dependency>
    <groupId>com.hazelcast</groupId>
    <artifactId>hazelcast</artifactId>
    <version>3.12</version>
</dependency>
<dependency>
    <groupId>com.hazelcast</groupId>
    <artifactId>hazelcast-hibernate53</artifactId>
    <version>1.3.2</version>
</dependency>
```

Рис. 63: MAVEN-ЗАВИСИМОСТЬ HAZELCAST & HIBERNATE

```
@Getter
@Setter
@Entity
@Cacheable
@NoArgsConstructor
@Table(name = "app_project")
@Cache(usage = CacheConcurrencyStrategy.READ_WRITE)
public class Project extends AbstractEntity
```

Рис. 64: КОНФИГУРАЦИЯ СУЩНОСТИ JPA HIBERNATE

```

<hazelcast xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.hazelcast.com/schema/config
    http://www.hazelcast.com/schema/config/hazelcast-config-3.11.xsd"
    xmlns="http://www.hazelcast.com/schema/config">
    <network>
        <port auto-increment="true" port-count="100">5701</port>
        <join>
            <multicast enabled="false"></multicast>
            <tcp-ip enabled="true">
                <member>127.0.0.1</member>
            </tcp-ip>
        </join>
    </network>
</hazelcast>

```

Рис. 65: XML-КОНФИГУРАЦИЯ HAZELCAST

```

<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
    http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd"
    version="2.1">
    <persistence-unit name="ENTERPRISE" transaction-type="RESOURCE_LOCAL">
        <properties>
            <property name="hibernate.cache.use_second_level_cache" value="true" />
            <property name="hibernate.cache.use_query_cache" value="true" />
            <property name="hibernate.cache.use_minimal_puts" value="true" />
            <property name="hibernate.cache.hazelcast.use_lite_member" value="true" />
            <property name="hibernate.cache.region_prefix" value="task-manager"/>
            <property name="hibernate.cache.provider_configuration_file_resource_path"
                value="hazelcast.xml" />
            <property name="hibernate.cache.region.factory_class"
                value="com.hazelcast.hibernate.HazelcastLocalCacheRegionFactory" />
        </properties>
    </persistence-unit>
</persistence>

```

Рис. 66: КОНФИГУРАЦИЯ SECOND LEVEL CACHE

18 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-18

- 18.1 РЕАЛИЗОВАТЬ КОМАНДУ SERVER INFO
- 18.2 НАСТРОИТЬ АПАЧЕ НА ПОРТУ 80 В КАЧЕСТВЕ ПРОКСИ БАЛАНСИРОВЩИКА ВЕБ-ЗАПРОСОВ
- 18.3 ОБНОВИТЬ КЛИЕНТ ДЛЯ РАБОТЫ С БАЛАНСИРОВЩИКОМ ВЕБ-ЗАПРОСОВ
- 18.4 ПРОТЕСТИРОВАТЬ РАБОТУ HAZELCAST JPA SECOND LEVEL CACHE В КЛАСТЕРЕ

```
LoadModule proxy_module libexec/apache2/mod_proxy.so
LoadModule proxy_http_module libexec/apache2/mod_proxy_http.so
LoadModule proxy_balancer_module libexec/apache2/mod_proxy_balancer.so
LoadModule lbmethod_byrequests_module
libexec/apache2/mod_lbmethod_byrequests.so
```

Рис. 67: МОДУЛИ АПАЧЕ ДЛЯ БАЛАНСИРОВКИ НАГРУЗКИ

```
<VirtualHost 127.0.0.1:80>

    ServerName cluster
    ProxyRequests Off
    ProxyPreserveHost On

    <Proxy balancer://mycluster>
        BalancerMember http://127.0.0.1:7070
        BalancerMember http://127.0.0.1:8080
        BalancerMember http://127.0.0.1:9090
        ProxySet lbmethod=byrequests
    </Proxy>

    ProxyPass / balancer://mycluster/
    ProxyPassReverse / balancer://mycluster/

</VirtualHost>
```

Рис. 68: КОНФИГУРАЦИЯ ВИРТУАЛЬНОГО ХОСТА АПАЧЕ

```
java -jar -Dserver.port=7070 ./tm-server.jar  
java -jar -Dserver.port=8080 ./tm-server.jar  
java -jar -Dserver.port=9090 ./tm-server.jar
```

Рис. 69: ЗАПУСК НОД СЕРВЕРА НА СМЕЩЕННЫХ ПОРТАХ

```
server-info  
[server-info]  
HOST: localhost  
PORT: 8080  
[OK]
```

```
server-info  
[server-info]  
HOST: localhost  
PORT: 7070  
[OK]
```

```
server-info  
[server-info]  
HOST: localhost  
PORT: 9090  
[OK]
```

Рис. 70: ОТВЕТ ОТ КЛАСТЕРА НА ЗАПРОС ИНФОРМАЦИИ О НОДЕ

```
Members {size:3, ver:3} [  
    Member [127.0.0.1]:5701 - be5414ed-4b51-43ac-ad5e-0d9451ee590e  
    Member [127.0.0.1]:5702 - 0ac9971a-e85a-4678-9987-71a993931837  
    Member [127.0.0.1]:5703 - 6ccc7971-6bc1-4e9d-9435-ad838ef7fcb0 this  
]
```

Рис. 71: ЛОГИ ОБЪЕДИНЕНИЯ НОД HAZELCAST В КЛАСТЕР

18.5 СКРИНШОТЫ

```

edit httpd.conf - Far 3.0.5354 x64
C:\Apache24\conf\httpd.conf
#LoadModule lmbmethod_bybusiness module modules/mod_lmbmethod_bybusiness.so
#LoadModule lmbmethod_bytraffic module modules/mod_lmbmethod_bytraffic.so
#LoadModule lmbmethod_heartbeat module modules/mod_lmbmethod_heartbeat.so
#LoadModule ldap_module modules/mod_ldap.so
#LoadModule logio_module modules/mod_logio.so
LoadModule log_config_module modules/mod_log_config.so
#LoadModule log_debug module modules/mod_log_debug.so
#LoadModule log_forensic module modules/mod_log_forensic.so
#LoadModule lua_module modules/mod_lua.so
#LoadModule macro_module modules/mod_macro.so
#LoadModule md_module modules/mod_md.so
LoadModule mime_module modules/mod_mime.so
#LoadModule mime_magic module modules/mod_mime_magic.so
LoadModule negotiation module modules/mod_negotiation.so
#LoadModule proxy_ajp module modules/mod_proxy_ajp.so
#LoadModule proxy_connect module modules/mod_proxy_connect.so
#LoadModule proxy_express module modules/mod_proxy_express.so
#LoadModule proxy_fcgi module modules/mod_proxy_fcgi.so
#LoadModule proxy_ftp module modules/mod_proxy_ftp.so
#LoadModule proxy_html module modules/mod_proxy_html.so
LoadModule proxy module modules/mod_proxy.so
LoadModule proxy_balancer module modules/mod_proxy_balancer.so
LoadModule proxy_http module modules/mod_proxy_http.so
LoadModule slotmem_shm module modules/mod_slotmem_shm.so
LoadModule lmbmethod_byrequests module modules/mod_1bmethod_byrequests.so
#LoadModule proxy_http2 module modules/mod_proxy_http2.so
#LoadModule proxy_scgi module modules/mod_proxy_scgi.so
#LoadModule proxy_uwsgi module modules/mod_proxy_uwsgi.so
#LoadModule proxy_wstunnel module modules/mod_proxy_wstunnel.so
#LoadModule ratelimit module modules/mod_ratelimit.so
#LoadModule reflector module modules/mod_reflector.so
#LoadModule remoteip module modules/mod_remoteip.so
#LoadModule request module modules/mod_request.so
#LoadModule reqtimeout module modules/mod_reqtimeout.so
#LoadModule rewrite module modules/mod_rewrite.so
#LoadModule sed module modules/mod_sed.so
#LoadModule session module modules/mod_session.so
#LoadModule session_cookie module modules/mod_session_cookie.so
#LoadModule session_crypto module modules/mod_session_crypto.so

```

Рис. 72: МОДУЛИ АРАЧЕ ДЛЯ БАЛАНСИРОВЩИКА

```

edit httpd.conf - Far 3.0.5354 x64
1251 Ln 552/552 Col 1 Ch 1 10:58
C:\Apache24\conf\httpd.conf
<IfModule proxy_html_module>
Include conf/extra/httpd-proxy-html.conf
</IfModule>

# Secure (SSL/TLS) connections
# Note: The following must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
Include conf/extra/httpd-ssl.conf
Include conf/extra/httpd-ssl-vhosts.conf
SSIRandomSeed startup builtin
SSIRandomSeed connect builtin
</IfModule>
<IfModule http2_module>
    ProtocolsHonorOrder On
    Protocols h2 h2c http/1.1
</IfModule>

<VirtualHost 127.0.0.1:80>
    ServerName cluster
    ProxyRequests Off
    ProxyPreserveHost On

    <Proxy balancer://mycluster>
        BalancerMember http://127.0.0.1:7079
        BalancerMember http://127.0.0.1:8080
        BalancerMember http://127.0.0.1:9090
        ProxySet lmbmethod-byrequests
    </Proxy>

    ProxyPass / balancer://mycluster/
    ProxyPassReverse / balancer://mycluster/
</VirtualHost>

```

Рис. 73: ВИРТУАЛЬНЫХ ХОСТ АРАЧЕ ДЛЯ БАЛАНСИРОВЩИКА

The screenshot shows a window titled "edit hosts - Far 3.0.5354 x64" displaying the Windows hosts file located at C:\Windows\System32\drivers\etc\hosts. The file contains comments and examples of how to map IP addresses to host names. It includes entries for source servers (102.54.94.97, 38.25.63.10) and client hosts (x.acme.com). It also defines localhost mappings for 127.0.0.1 and ::1, and a cluster entry.

```
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97    rhino.acme.com    # source server
#      38.25.63.10    x.acme.com        # x client host
#
# localhost name resolution is handled within DNS itself.
#      127.0.0.1          localhost
#      ::1               localhost
127.0.0.1      cluster
```

Рис. 74: КОНФИГУРАЦИЯ ХОСТОВ WINDOWS

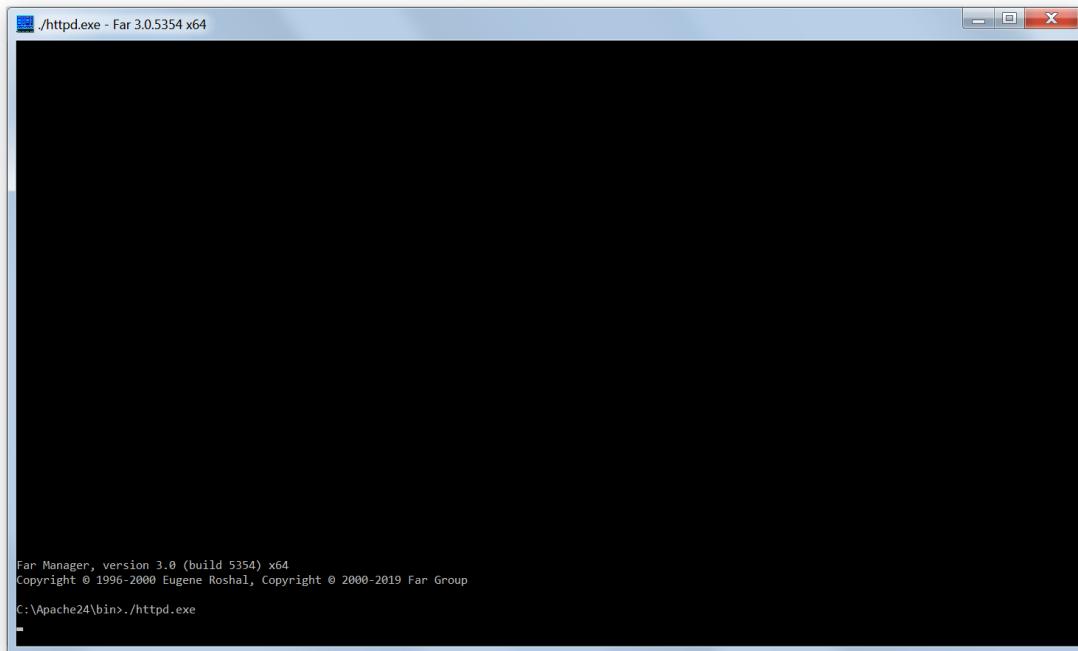
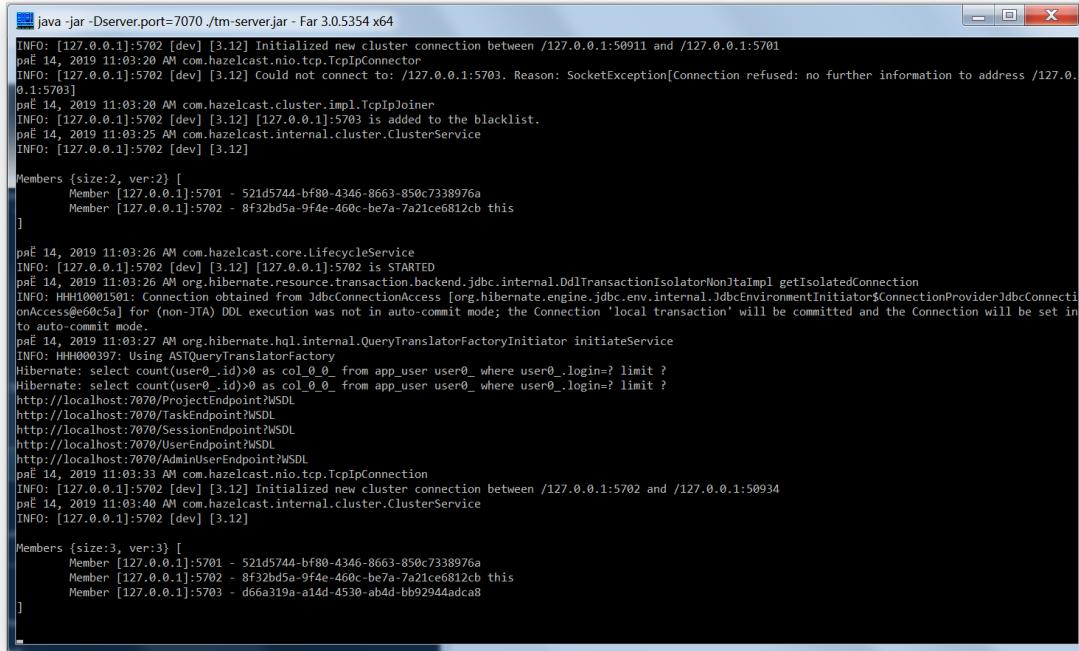


Рис. 75: АПАЧЕ ВЕБ-СЕРВЕР ДЛЯ ОС WINDOWS



```

java -jar -Dserver.port=7070 /tm-server.jar - Far 3.0.5354 x64
INFO: [127.0.0.1]:5702 [dev] [3.12] Initialized new cluster connection between /127.0.0.1:50911 and /127.0.0.1:5701
pr@14 2019 11:03:20 AM com.hazelcast.nio.tcp.TcpIpConnector
INFO: [127.0.0.1]:5702 [dev] [3.12] Could not connect to: /127.0.0.1:5703. Reason: SocketException[Connection refused: no further information to address /127.0.0.1:5703]
pr@14 2019 11:03:20 AM com.hazelcast.cluster.impl.TcpIpJoiner
INFO: [127.0.0.1]:5702 [dev] [3.12] [127.0.0.1]:5703 is added to the blacklist.
pr@14 2019 11:03:25 AM com.hazelcast.internal.cluster.ClusterService
INFO: [127.0.0.1]:5702 [dev] [3.12]

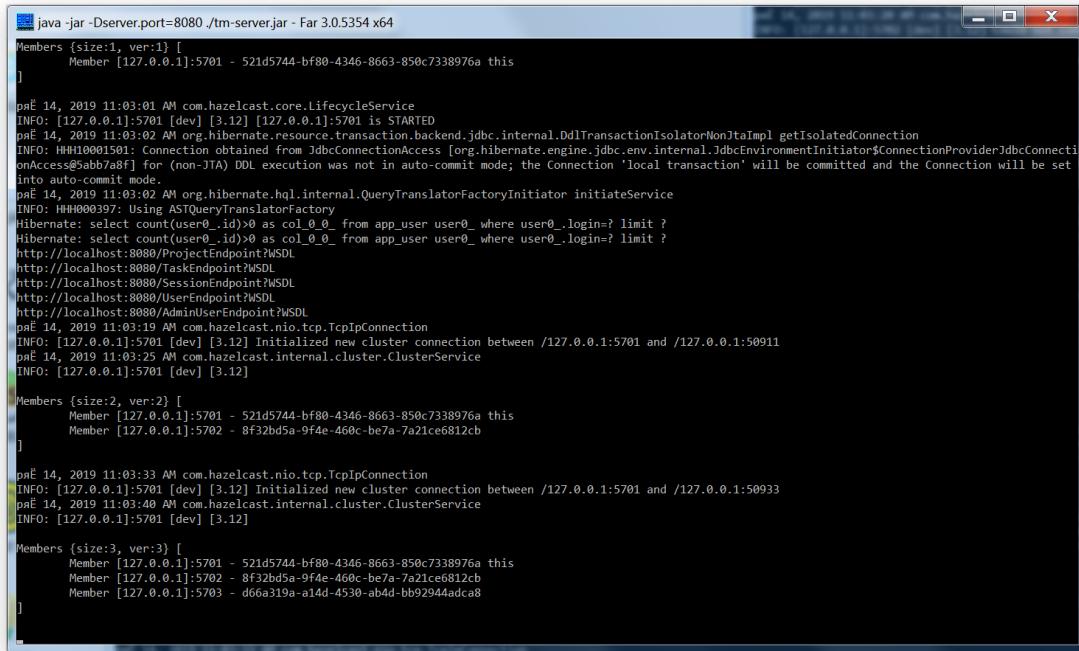
Members {size:2, ver:2} [
    Member [127.0.0.1]:5701 - 521d5744-bf80-4346-8663-850c7338976a
    Member [127.0.0.1]:5702 - 8f32bd5a-9f4e-460c-be7a-7a21ce6812cb this
]

pr@14 2019 11:03:26 AM com.hazelcast.core.LifecycleService
INFO: [127.0.0.1]:5702 [dev] [3.12] [127.0.0.1]:5702 is STARTED
pr@14 2019 11:03:26 AM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH010001501: Connection obtained from JDBCConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProvider$JdbcConnectionAccess@6e0c5a] for (non-JTA) DDL execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set in to auto-commit mode.
pr@14 2019 11:03:27 AM org.hibernate.hql.internal.QueryTranslatorFactoryInitiator initiateService
INFO: HHH000397: Using ASTQueryTranslatorFactory
Hibernate: select count(user0_.id)>0 as col_0_0 from app_user user0_ where user0_.login=? limit ?
Hibernate: select count(user0_.id)>0 as col_0_0 from app_user user0_ where user0_.login=? limit ?
http://localhost:7070/ProjectEndpoint?WSDL
http://localhost:7070/TaskEndpoint?WSDL
http://localhost:7070/SessionEndpoint?WSDL
http://localhost:7070/UserEndpoint?WSDL
http://localhost:7070/AdminUserEndpoint?WSDL
pr@14 2019 11:03:33 AM com.hazelcast.nio.tcp.TcpIpConnection
INFO: [127.0.0.1]:5702 [dev] [3.12] Initialized new cluster connection between /127.0.0.1:5702 and /127.0.0.1:50934
pr@14 2019 11:03:40 AM com.hazelcast.internal.cluster.ClusterService
INFO: [127.0.0.1]:5702 [dev] [3.12]

Members {size:3, ver:3} [
    Member [127.0.0.1]:5701 - 521d5744-bf80-4346-8663-850c7338976a
    Member [127.0.0.1]:5702 - 8f32bd5a-9f4e-460c-be7a-7a21ce6812cb this
    Member [127.0.0.1]:5703 - d66a319a-a14d-4530-ab4d-bb92944adca8
]

```

Рис. 76: НОДА СЕРВЕРА ПРИЛОЖЕНИЯ НА ПОРТЕ 7070



```

java -jar -Dserver.port=8080 /tm-server.jar - Far 3.0.5354 x64
Members {size:1, ver:1} [
    Member [127.0.0.1]:5701 - 521d5744-bf80-4346-8663-850c7338976a this
]

pr@14 2019 11:03:01 AM com.hazelcast.core.LifecycleService
INFO: [127.0.0.1]:5701 [dev] [3.12] [127.0.0.1]:5701 is STARTED
pr@14 2019 11:03:02 AM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH010001501: Connection obtained from JDBCConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProvider$JdbcConnectionAccess@5abb7a8f] for (non-JTA) DDL execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.
pr@14 2019 11:03:02 AM org.hibernate.hql.internal.QueryTranslatorFactoryInitiator initiateService
INFO: HHH000397: Using ASTQueryTranslatorFactory
Hibernate: select count(user0_.id)>0 as col_0_0 from app_user user0_ where user0_.login=? limit ?
Hibernate: select count(user0_.id)>0 as col_0_0 from app_user user0_ where user0_.login=? limit ?
http://localhost:8080/ProjectEndpoint?WSDL
http://localhost:8080/TaskEndpoint?WSDL
http://localhost:8080/SessionEndpoint?WSDL
http://localhost:8080/UserEndpoint?WSDL
http://localhost:8080/AdminUserEndpoint?WSDL
pr@14 2019 11:03:19 AM com.hazelcast.nio.tcp.TcpIpConnection
INFO: [127.0.0.1]:5701 [dev] [3.12] Initialized new cluster connection between /127.0.0.1:5701 and /127.0.0.1:50911
pr@14 2019 11:03:25 AM com.hazelcast.internal.cluster.ClusterService
INFO: [127.0.0.1]:5701 [dev] [3.12]

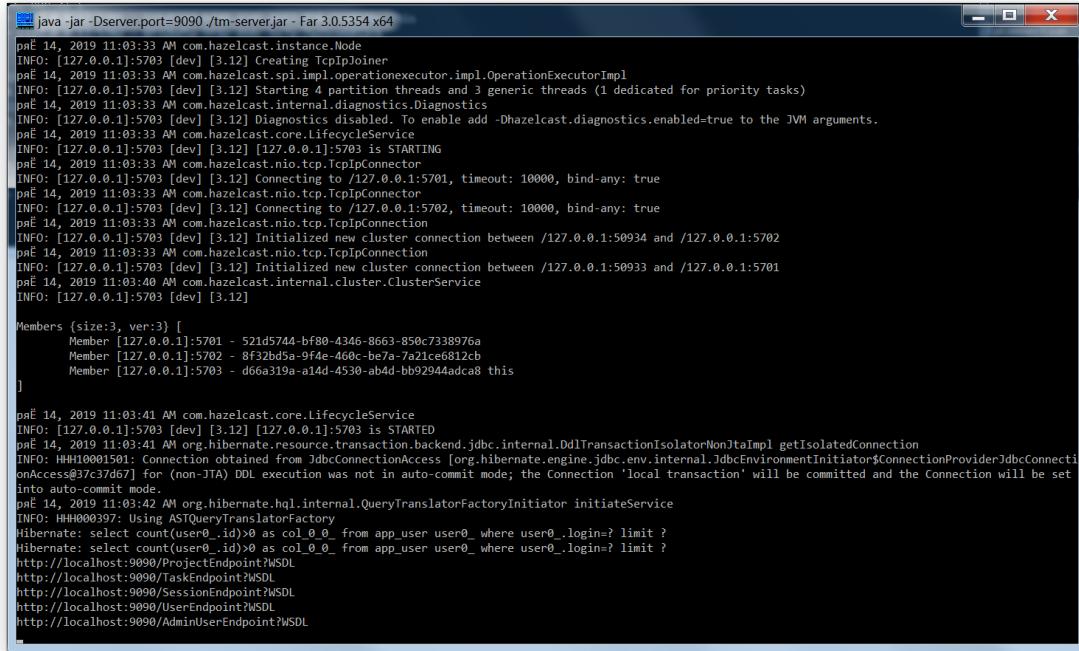
Members {size:2, ver:2} [
    Member [127.0.0.1]:5701 - 521d5744-bf80-4346-8663-850c7338976a this
    Member [127.0.0.1]:5702 - 8f32bd5a-9f4e-460c-be7a-7a21ce6812cb
]

pr@14 2019 11:03:33 AM com.hazelcast.nio.tcp.TcpIpConnection
INFO: [127.0.0.1]:5701 [dev] [3.12] Initialized new cluster connection between /127.0.0.1:5701 and /127.0.0.1:50933
pr@14 2019 11:03:40 AM com.hazelcast.internal.cluster.ClusterService
INFO: [127.0.0.1]:5701 [dev] [3.12]

Members {size:3, ver:3} [
    Member [127.0.0.1]:5701 - 521d5744-bf80-4346-8663-850c7338976a this
    Member [127.0.0.1]:5702 - 8f32bd5a-9f4e-460c-be7a-7a21ce6812cb
    Member [127.0.0.1]:5703 - d66a319a-a14d-4530-ab4d-bb92944adca8
]

```

Рис. 77: НОДА СЕРВЕРА ПРИЛОЖЕНИЯ НА ПОРТЕ 8080



```

java -jar D-server.jar -port=9090 /tm-server.jar -Far 3.0.5354 x64
...
INFO: [127.0.0.1]:5703 [dev] [3.12] Diagnostics disabled. To enable add -Dhazelcast.diagnostics.enabled=true to the JVM arguments.
...
INFO: [127.0.0.1]:5703 [dev] [3.12] [127.0.0.1]:5703 is STARTING
...
INFO: [127.0.0.1]:5703 [dev] [3.12] [127.0.0.1]:5703 is STARTED
...
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProvider$JdbcConnectionAccess@37c37d67] for (non-JTA) DDL execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.
...
INFO: WELD-000000: 3.1.0 (Final)
...
INFO: WELD-000101: Transactional services not available. Injection of @Inject UserTransaction not available. Transactional observers will be invoked synchronously.
...
INFO: WELD-ENV-002003: Weld SE container 7671c593-0776-4b67-a629-5057d3ec0de1 initialized
*** WELCOME TO TASK MANAGER ***
...
[login]
ENTER LOGIN:
admin
ENTER PASSWORD:
admin
[OK]

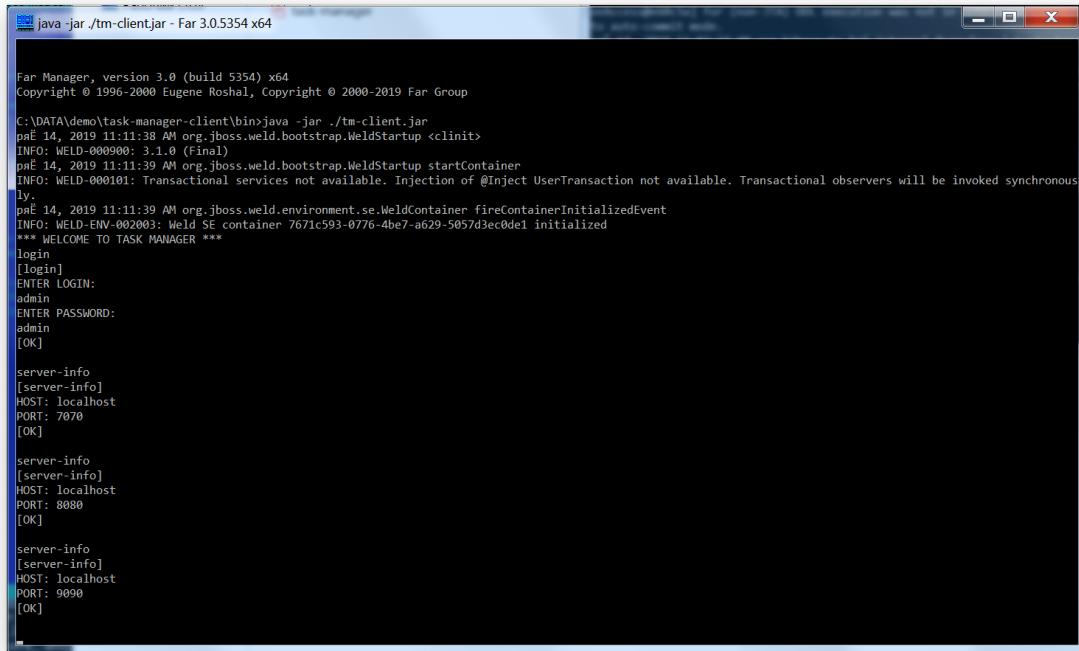
server-info
[server-info]
HOST: localhost
PORT: 7070
[OK]

server-info
[server-info]
HOST: localhost
PORT: 8080
[OK]

server-info
[server-info]
HOST: localhost
PORT: 9090
[OK]

```

Рис. 78: НОДА СЕРВЕРА ПРИЛОЖЕНИЯ НА ПОРТЕ 9090



```

java -jar /tm-client.jar -Far 3.0.5354 x64
...
Far Manager, version 3.0 (build 5354) x64
Copyright © 1996-2000 Eugene Roshal, Copyright © 2000-2019 Far Group
C:\DATA\demo\task-manager-client\bin>java -jar ./tm-client.jar
...
INFO: WELD-000000: 3.1.0 (Final)
...
INFO: WELD-000101: Transactional services not available. Injection of @Inject UserTransaction not available. Transactional observers will be invoked synchronously.
...
INFO: WELD-ENV-002003: Weld SE container 7671c593-0776-4b67-a629-5057d3ec0de1 initialized
*** WELCOME TO TASK MANAGER ***
...
[login]
ENTER LOGIN:
admin
ENTER PASSWORD:
admin
[OK]

server-info
[server-info]
HOST: localhost
PORT: 7070
[OK]

server-info
[server-info]
HOST: localhost
PORT: 8080
[OK]

server-info
[server-info]
HOST: localhost
PORT: 9090
[OK]

```

Рис. 79: ИНФОРМАЦИЯ О НОДАХ КЛАСТЕРА

19 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SE-19

19.1 ПЕРЕВЕСТИ ПРИЛОЖЕНИЕ НА SPRING

- 19.1.1 ЗАМЕНИТЬ ЗАВИСИМОСТИ И АННОТАЦИИ WELD CDI НА SPRING CONTEXT
- 19.1.2 ЗАМЕНИТЬ РЕПОЗИТОРИИ APACHE DELTA SPIKE НА SPRING DATA JPA
- 19.1.3 ЗАМЕНИТЬ ТЕСТЫ WELD - APACHE DELTA SPIKE НА SPRING TESTING

19.2 РЕКОМЕНДАЦИИ

- 19.2.1 УДАЛИТЬ ВСЕ ЗАВИСИМОСТИ WELD И APACHE DELTA SPIKE ИЗ MAVEN-ПРОЕКТА
- 19.2.2 РЕАЛИЗОВАТЬ SPRING CONFIG С ИСПОЛЬЗОВАНИЕМ JAVA BASED ПОДХОДА И АННОТАЦИЙ

19.3 ПРОГРАММНЫЙ КОД

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.0.8.RELEASE</version>
</dependency>
```

Рис. 80: MAVEN-ЗАВИСИМОСТЬ SPRING CONTEXT

```
<dependency>
    <groupId>org.springframework.data</groupId>
    <artifactId>spring-data-jpa</artifactId>
    <version>2.0.7.RELEASE</version>
</dependency>
```

Рис. 81: MAVEN-ЗАВИСИМОСТЬ SPRING DATA JPA

```
@Bean
public DataSource dataSource() {
    final DriverManagerDataSource dataSource =
        new DriverManagerDataSource();
    dataSource.setDriverClassName("com.mysql.jdbc.Driver");
    dataSource.setUrl("jdbc:mysql://127.0.0.1:3306/example");
    dataSource.setUsername("root");
    dataSource.setPassword("");
    return dataSource;
}

@Bean
public LocalContainerEntityManagerFactoryBean entityManagerFactory(
    final DataSource dataSource
) {
    final LocalContainerEntityManagerFactoryBean factoryBean;
    factoryBean = new LocalContainerEntityManagerFactoryBean();
    factoryBean.setDataSource(dataSource);
    factoryBean.setJpaVendorAdapter(new HibernateJpaVendorAdapter());
    factoryBean.setPackagesToScan("ru.volnenko.spring.model");
    final Properties properties = new Properties();
    properties.put("hibernate.show_sql", "true");
    properties.put("hibernate.hbm2ddl.auto", "update");
    properties.put("hibernate.dialect",
        "org.hibernate.dialect.MySQL5Dialect");
    factoryBean.setJpaProperties(properties);
    return factoryBean;
}

@Bean
public PlatformTransactionManager transactionManager(
    final LocalContainerEntityManagerFactoryBean emf
) {
    final JpaTransactionManager transactionManager =
        new JpaTransactionManager();
    transactionManager.setEntityManagerFactory(emf.getObject());
    return transactionManager;
}
```

Рис. 82: JAVA BASED КОНФИГУРАЦИЯ SPRING JPA

Часть II

JAVA EE

20 ПРАКТИЧЕСКОЕ ЗАДАНИЕ ЕЕ-01

20.1 СОЗДАТЬ ВЕБ-ПРИЛОЖЕНИЕ

20.1.1 СОЗДАТЬ WAR-ПРИЛОЖЕНИЕ С ИСПОЛЬЗОВАНИЕМ MAVEN-АРХИТИПА ЧЕРЕЗ IDE

20.1.2 ОБНОВИТЬ ВЕРСИЮ ДИСКРИПТОРА "web.xml" ДО 3.1

20.2 НАСТРОИТЬ SERVLET-КОНТЕЙНЕР

20.2.1 НАСТРОИТЬ СБОРКУ И ЗАПУСК ВЕБ-ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ ТОМСАТ MAVEN-ПЛАГИНА

20.2.2 НАСТРОИТЬ СБОРКУ И ЗАПУСК ВЕБ-ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ ІЕТТУ-ПЛАГИНА

20.3 РЕАЛИЗОВАТЬ CRUD ДЛЯ ДОМЕНА

20.3.1 РАЗРАБОТАТЬ JSP-СТРАНИЦЫ ЛИСТИНГА С ТАБЛИЧНОЙ ВЕРСТКОЙ ДЛЯ ОТОБРАЖЕНИЯ ЗАПИСЕЙ

20.3.2 РАЗРАБОТАТЬ JSP-СТРАНИЦЫ С ФОРМОЙ РЕДАКТИРОВАНИЯ ДЛЯ ИЗМЕНЕНИЯ ЗАПИСЕЙ

20.4 РЕАЛИЗОВАТЬ АВТОРИЗАЦИЮ С ИСПОЛЬЗОВАНИЕМ SERVLET HTTP-SESSION

20.4.1 РЕАЛИЗОВАТЬ СТРАНИЦУ АВТОРИЗАЦИИ

20.4.2 РЕАЛИЗОВАТЬ СТРАНИЦУ РЕГИСТРАЦИИ

20.4.3 РЕАЛИЗОВАТЬ ЗАВЕРШЕНИЯ СЕАНСА ПОЛЬЗОВАТЕЛЯ

20.5 РЕКОМЕНДАЦИИ

20.5.1 КАЖДАЯ СУЩНОСТЬ ДОЛЖНА ИМЕТЬ МИНИМУМ ДВЕ JSP-СТРАНИЦЫ ДЛЯ ЛИСТИНГА И РЕДАКТИРОВАНИЯ

20.5.2 РЕАЛИЗОВАТЬ БИЗНЕС ЛОГИКУ НА СТОРОНЕ SERVLET-КЛАССА С ПЕРЕДАЧЕЙ УПРАВЛЕНИЯ JSP-СТРАНИЦЕ

20.5.3 РЕАЛИЗОВАТЬ ОТДЕЛЬНЫЕ SERVLET-КЛАССЫ ДЛЯ КАЖДОЙ CRUD-ОПЕРАЦИИ

20.5.4 РЕАЛИЗОВАТЬ РЕГИСТРАЦИЮ SERVLET-КЛАССОВ С ИСПОЛЬЗОВАНИЕМ АННОТАЦИЙ

20.5.5 РЕАЛИЗОВАТЬ ХРАНЕНИЕ ДАННЫХ В РЕПОЗИТОРИЯХ В ВИДЕ ОДИНОЧЕК НА БАЗЕ СТРУКТУРЫ "МАР"

20.5.6 РАЗМЕСТИТЬ JSP-СТРАНИЦЫ В ПАПКЕ "WEB-INF/views"

20.5.7 ИСПОЛЬЗОВАТЬ ТЕХНОЛОГИЮ JSTL ДЛЯ ОТРИСОВКИ ЭЛЕМЕНТОВ В JSP-СТРАНИЦАХ

20.6 ПРОГРАММНЫЙ КОД

```
<plugin>
    <groupId>org.apache.tomcat.maven</groupId>
    <artifactId>tomcat7-maven-plugin</artifactId>
    <version>2.2</version>
    <configuration>
        <path>/</path>
    </configuration>
</plugin>
```

Рис. 83: КОНФИГУРАЦИЯ ТОМСАТ MAVEN-PLUGIN

```
<plugin>
    <groupId>org.eclipse.jetty</groupId>
    <artifactId>jetty-maven-plugin</artifactId>
    <version>9.4.14.v20181114</version>
    <configuration>
        <webApp>
            <contextPath>/</contextPath>
        </webApp>
    </configuration>
</plugin>
```

Рис. 84: КОНФИГУРАЦИЯ JETTY MAVEN-PLUGIN

20.7 СКРИНШОТЫ

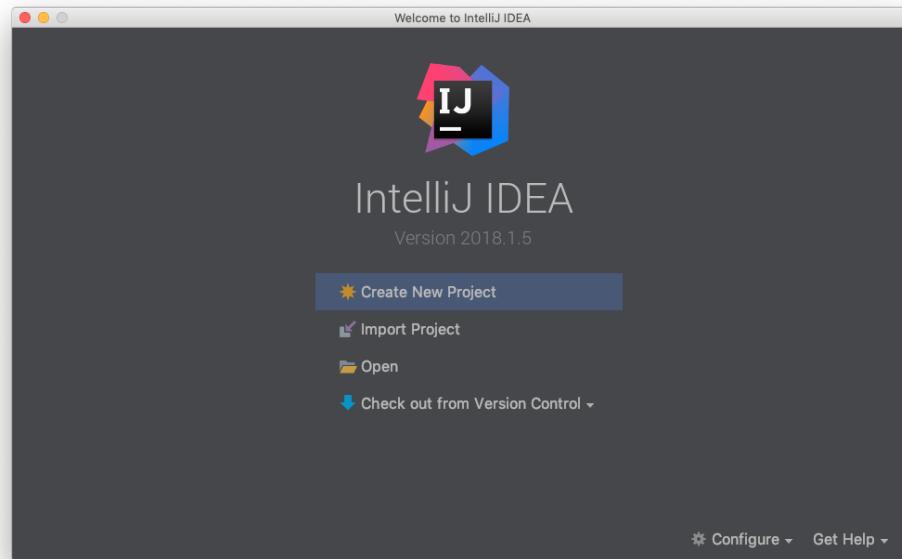


Рис. 85: ПРИВЕТСТВИЕ IDE ДЛЯ СОЗДАНИЯ ПРОЕКТА

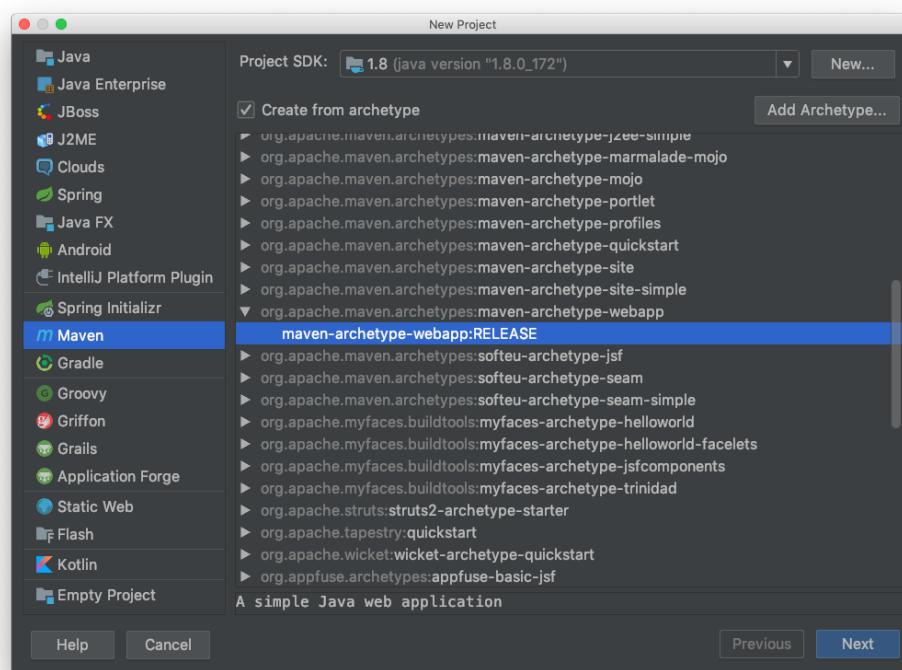


Рис. 86: ВЫБОР ШАБЛОНА MAVEN-ПРОЕКТА ДЛЯ ВЕБ-ПРИЛОЖЕНИЯ

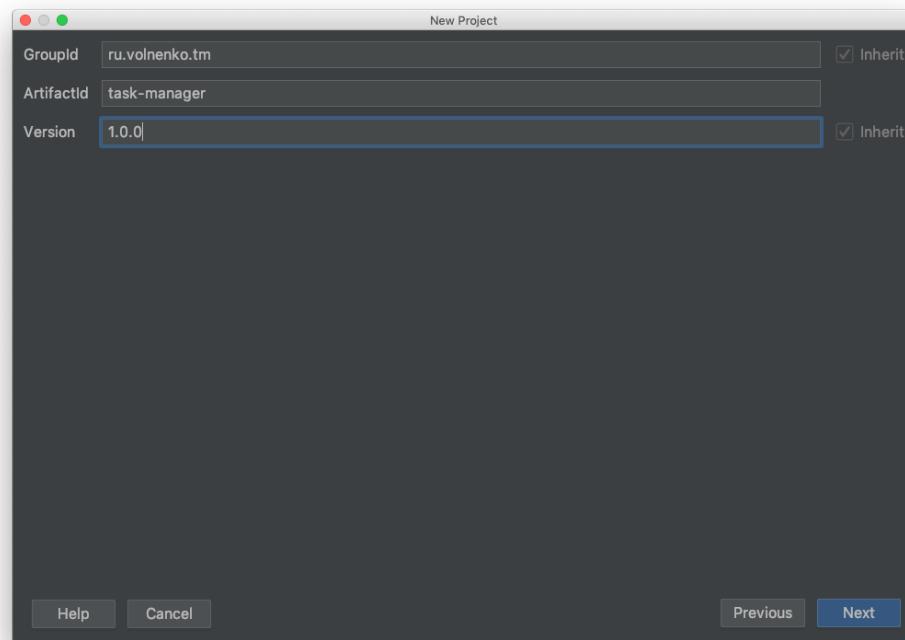


Рис. 87: НАСТРОЙКА ШАБЛОНА MAVEN-ПРОЕКТА

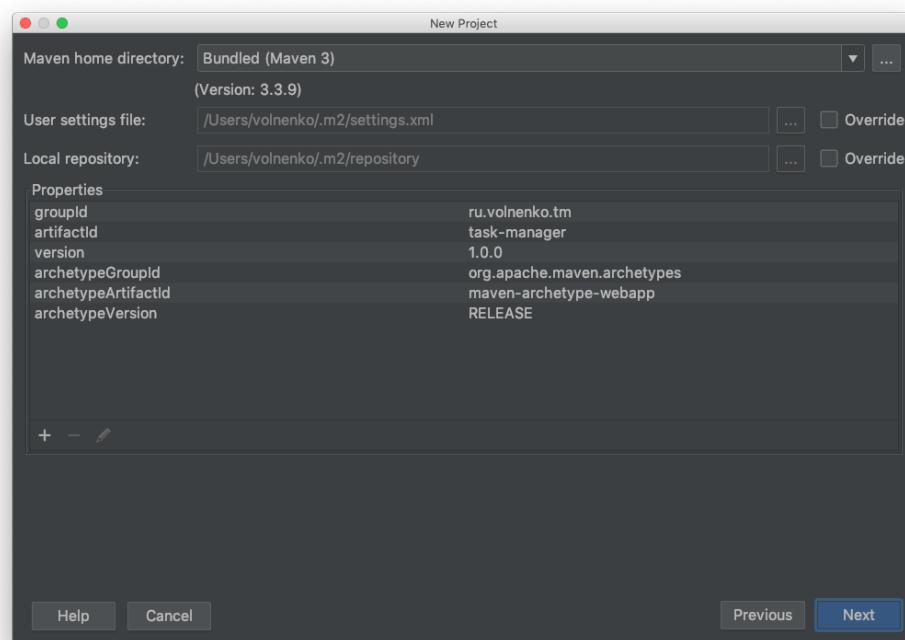


Рис. 88: ПОДТВЕРЖДЕНИЕ НАСТРОЕК MAVEN-ПРОЕКТА

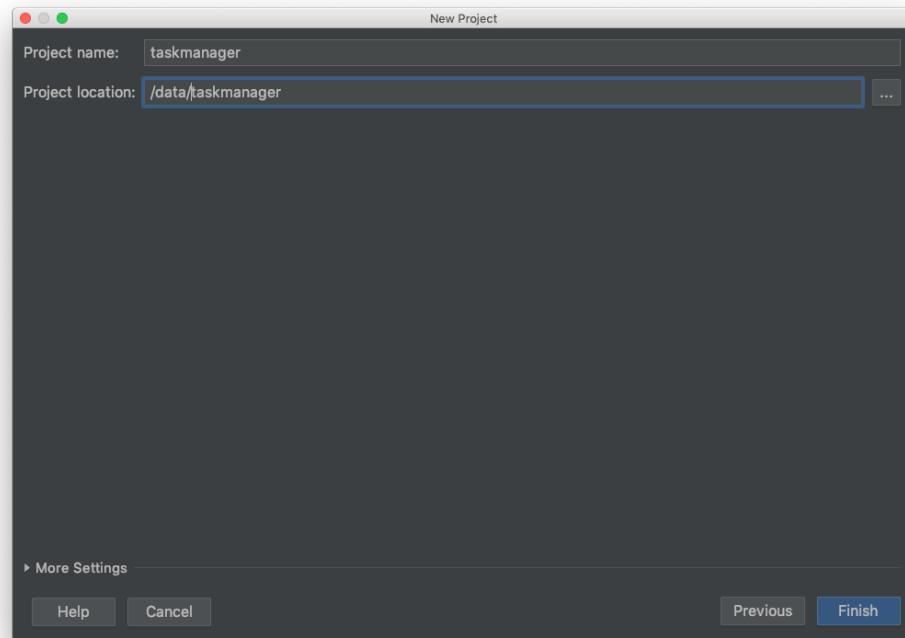


Рис. 89: НАСТРОЙКА РАБОЧЕГО ПРОСТРАНСТВА ПРОЕКТА

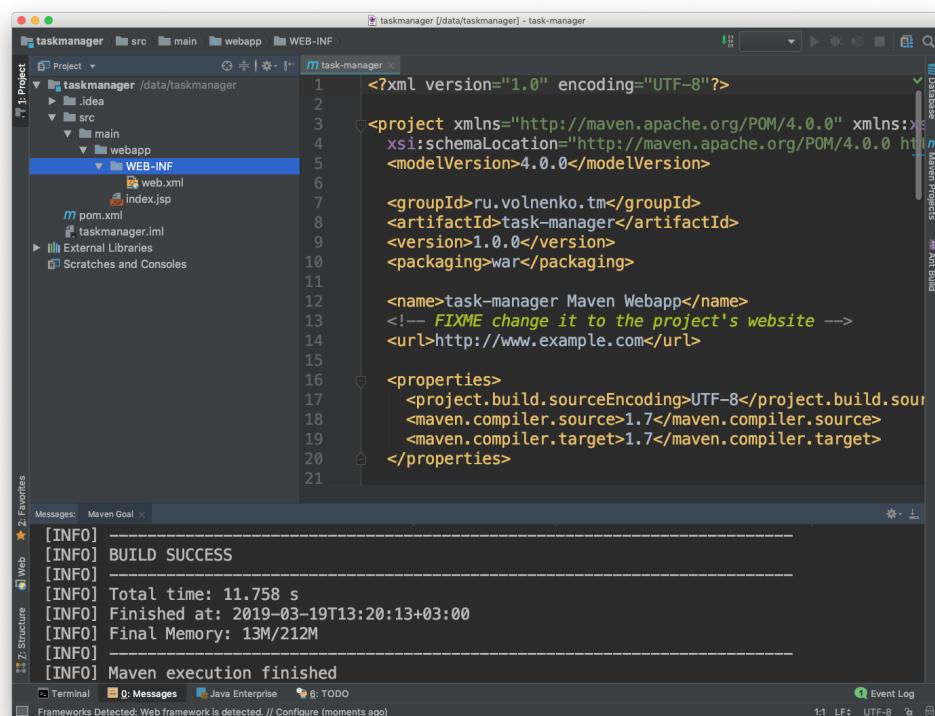


Рис. 90: ОТОБРАЖЕНИЕ СТРУКТУРЫ WAR-ПРОЕКТА В IDE

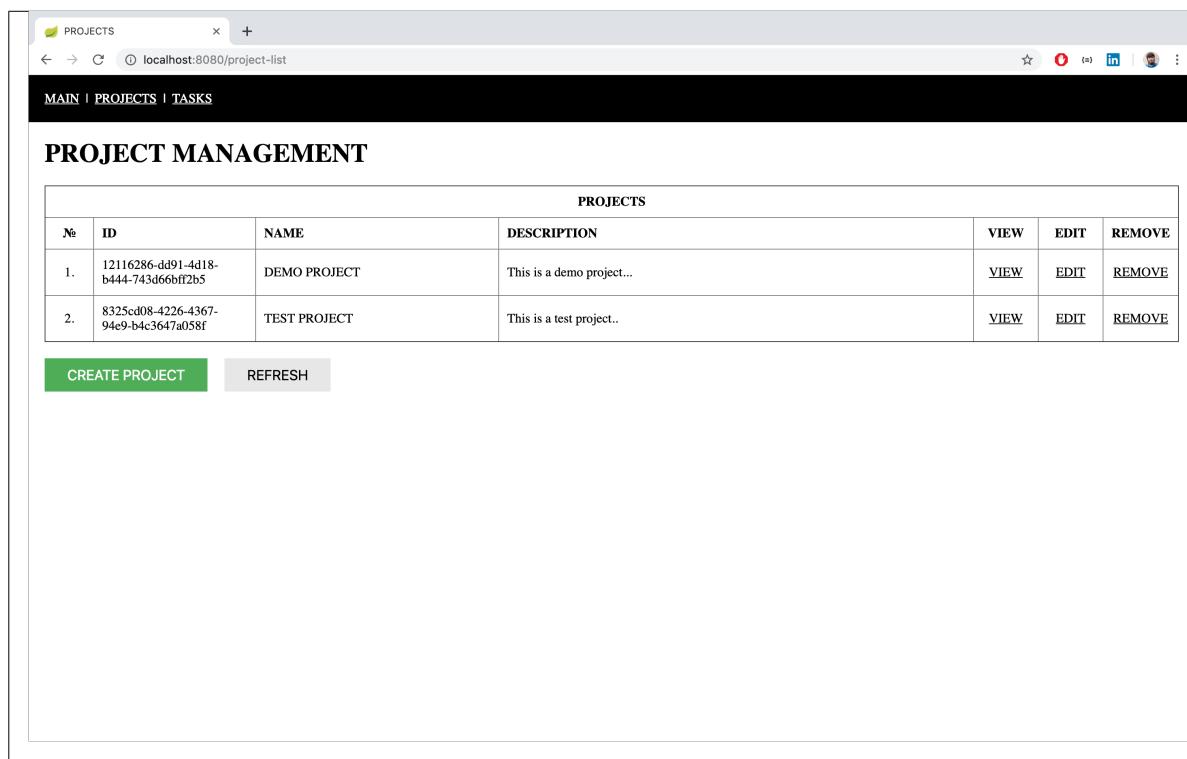


Рис. 91: JSP-СТРАНИЦА ДЛЯ ОТОБРАЖЕНИЯ СПИСКА ПРОЕКТОВ

NAME

DESCRIPTION

DATE BEGIN

DATE END

STATUS

[SAVE](#)

Рис. 92: JSP-СТРАНИЦА ДЛЯ РЕДАКТИРОВАНИЯ ПРОЕКТА

Рис. 93: JSP-СТРАНИЦА ДЛЯ ОТОБРАЖЕНИЯ СПИСКА ЗАДАЧ

Рис. 94: JSP-СТРАНИЦА ДЛЯ РЕДАКТИРОВАНИЯ ЗАДАЧИ

21 ПРАКТИЧЕСКОЕ ЗАДАНИЕ ЕЕ-02

21.1 ВНЕДРИТЬ CDI В WAR-ПРИЛОЖЕНИЕ

- 21.1.1 РЕАЛИЗОВАТЬ ОДИНОЧКИ ДЛЯ РЕПОЗИТОРИЕВ С ИСПОЛЬЗОВАНИЕМ CDI АННОТАЦИЙ
- 21.1.2 ИСПОЛЬЗОВАТЬ ВНЕДРЕНИЕ ЗАВИСИМОСТЕЙ В SERVLET-КЛАССЫ ВМЕСТО ВЫЗОВОВ СТАТИЧЕСКИХ МЕТОДОВ

21.2 НАСТРОИТЬ СЕРВЕР WILDFLY

- 21.2.1 ЗАГРУЗИТЬ ПОСЛЕДНЮЮ ВЕРСИЮ СЕРВЕРА ПРИЛОЖЕНИЙ WILDFLY

- 21.2.2 НАСТРОИТЬ ДЕПЛОЙМЕНТ ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ MAVEN-ПЛАГИНА

21.3 РАЗРАБОТАТЬ SOAP-СЕРВИСЫ JAX-WS

- 21.3.1 РЕАЛИЗОВАТЬ CRUD-СЕРВИСЫ С ИСПОЛЬЗОВАНИЕМ JAX-WS ДЛЯ УПРАВЛЕНИЯ ПРЕДМЕТНОЙ ОБЛАСТЬЮ

- 21.3.2 РЕАЛИЗОВАТЬ ВЕБ-СЕРВИС SOAP ДЛЯ АВТОРИЗАЦИИ И РЕГИСТРАЦИИ ЧЕРЕЗ JAX-WS

- 21.3.3 РЕАЛИЗОВАТЬ ГЕНЕРАЦИЮ КЛИЕНТА В ОТДЕЛЬНОМ MAVEN-МОДУЛЕ ДЛЯ ВЕБ-СЕРВИСОВ

- 21.3.4 РАЗРАБОТАТЬ ПАКЕТ ИНТЕГРАЦИОННЫХ ТЕСТОВ ДЛЯ АВТОРИЗАЦИИ И CRUD-ОПЕРАЦИЙ

22 ПРАКТИЧЕСКОЕ ЗАДАНИЕ ЕЕ-03

22.1 ВНЕДРИТЬ JPA В WAR-ПРИЛОЖЕНИЕ

- 22.1.1 СОЗДАТЬ МОДУЛЬ ДЛЯ КОННЕКТОРА MYSQL 5.1.4 В СЕРВЕРЕ ПРИЛОЖЕНИЙ WILDFLY

- 22.1.2 СОЗДАТЬ DATASOURCE ДЛЯ БД ПРОЕКТА В КОНФИГУРАЦИИ СЕРВЕРА ПРИЛОЖЕНИЙ WILDFLY

- 22.1.3 РАЗРАБОТАТЬ ДИСКРИПТОР "persistence.xml" ДЛЯ СОЕДИНЕНИЯ ПРИЛОЖЕНИЯ С НАСТРОЕННЫМ DATASOURCE

- 22.1.4 ИСПОЛЬЗОВАТЬ В РЕПОЗИТОРИЯХ JPA ВМЕСТО СТРУКТУР "МАР"

- 22.1.5 РЕАЛИЗОВАТЬ РЕПОЗИТОРИИ С ИСПОЛЬЗОВАНИЕМ ТРАНЗАКЦИОННЫХ КЛАССОВ CDI

23 ПРАКТИЧЕСКОЕ ЗАДАНИЕ ЕЕ-04

- 23.1 ВНЕДРИТЬ В ПРИЛОЖЕНИЕ APACHE DELTA SPIKE
 - 23.1.1 ПОДКЛЮЧИТЬ ЗАВИСИМОСТИ MAVEN-ПРОЕКТА ДЛЯ APACHE DELTA SPIKE С ОФИЦИАЛЬНОГО САЙТА РАЗРАБОТЧИКОВ
 - 23.1.2 РЕАЛИЗОВАТЬ РЕПОЗИТОРИИ С ИСПОЛЬЗОВАНИЕМ APACHE DELTA SPIKE В ВИДЕ ИНТЕРФЕЙСОВ
- 23.2 РЕКОМЕНДАЦИИ
 - 23.2.1 РЕАЛИЗОВАТЬ УПРАВЛЕНИЕ ТРАНЗАКЦИЯМИ С ИСПОЛЬЗОВАНИЕМ ОКРУЖЕНИЯ СЕРВЕРА ПРИЛОЖЕНИЯ WILDFLY

24 ПРАКТИЧЕСКОЕ ЗАДАНИЕ ЕЕ-05

24.1 ВНЕДРИТЬ В ПРИЛОЖЕНИЕ PRIMEFACES

- 24.1.1 РЕАЛИЗОВАТЬ CRUD-СТРАНИЦЫ С ИСПОЛЬЗОВАНИЕМ JSF ВМЕСТО JSP
- 24.1.2 ИСПОЛЬЗОВАТЬ КОМПОНЕНТЫ PRIMEFACES ДЛЯ РЕАЛИЗАЦИИ ВЕБ-СТРАНИЦ

24.2 ВНЕДРИТЬ В ПРИЛОЖЕНИЕ PRETTY FACES

- 24.2.1 ИСПОЛЬЗОВАТЬ ПЕРЕОПРЕДЕЛЕНИЕ URL ДЛЯ JSF-СТРАНИЦ С ИСПОЛЬЗОВАНИЕМ АННОТАЦИЙ PRETTY FACES
 - 24.2.2 РАЗМЕСТИТЬ АННОТАЦИИ PRETTY FACES В СЕРВЕРНЫХ БИНАХ JSF-СТРАНИЦ
- ### 24.3 РЕКОМЕНДАЦИИ
- 24.3.1 РАЗМЕСТИТЬ JSF-СТРАНИЦЫ В ПАПКЕ "WEB-INF/views"
 - 24.3.2 ИСПОЛЬЗОВАТЬ ОБЛАСТЬ ВИДИМОСТИ "ViewScope" ДЛЯ СЕРВЕРНЫХ БИНОВ JSF-СТРАНИЦ

24.4 СКРИНШОТЫ

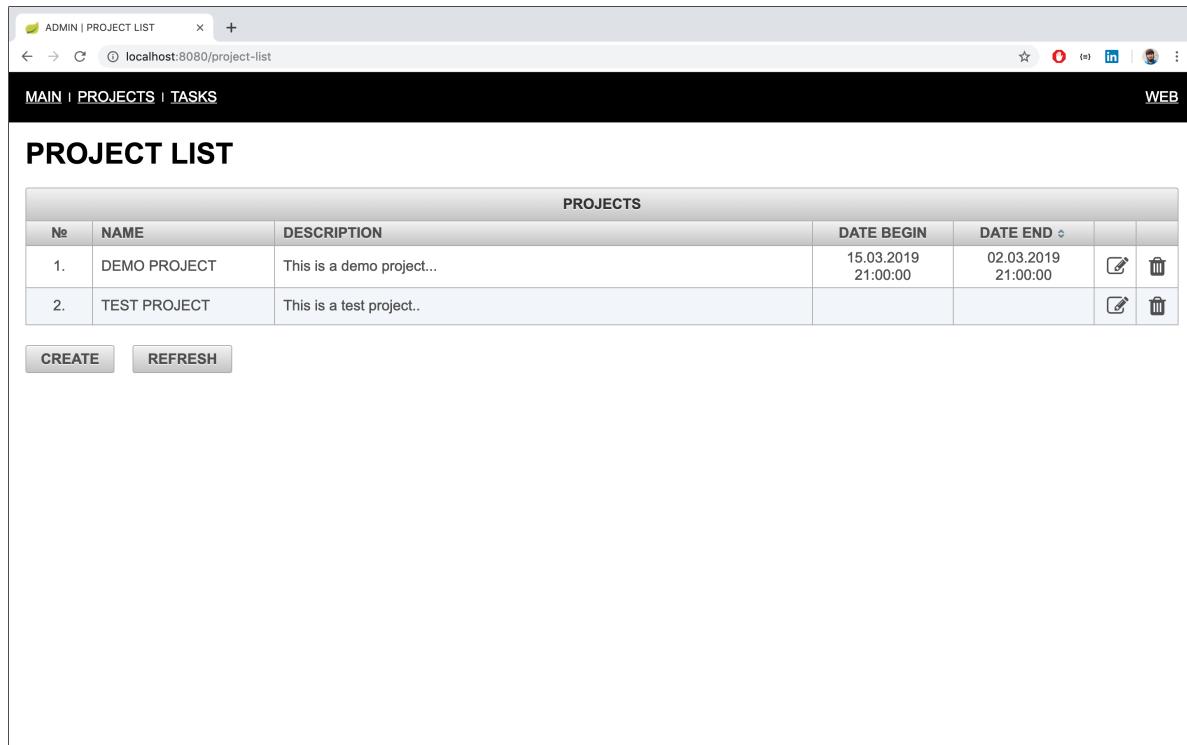


Рис. 95: JSF-СТРАНИЦА ДЛЯ ОТОБРАЖЕНИЯ СПИСКА ПРОЕКТОВ

Рис. 96: JSF-СТРАНИЦА ДЛЯ РЕДАКТИРОВАНИЯ ПРОЕКТА

The screenshot shows a web browser window titled "ADMIN | TASK LIST". The URL is "localhost:8080/task-list". The page has a header with "MAIN | PROJECTS | TASKS" and a "WEB" tab. Below the header is a section titled "TASK LIST" containing a table with three rows of task data. At the bottom are "CREATE" and "REFRESH" buttons.

Nº	PROJECT	NAME	DESCRIPTION	DATE BEGIN	DATE END		
1.	TEST PROJECT	Prepare demo data	Parse XML from web-service...	01.03.2019 21:00:00	16.03.2019 21:00:00		
2.	DEMO PROJECT	JPA Config	Make JPA connection for RDMS...	07.03.2019 21:00:00	22.03.2019 21:00:00		
3.	DEMO PROJECT	First task	Make project layout...	01.03.2019 21:00:00	08.03.2019 21:00:00		

Рис. 97: JSF-СТРАНИЦА ДЛЯ ОТОБРАЖЕНИЯ СПИСКА ЗАДАЧ

The screenshot shows a web browser window titled "ADMIN | ADMIN". The URL is "localhost:8080/task-edit?id=8c37d535-281a-46cf-bf5d-68783da11034". The page has a header with "MAIN | PROJECTS | TASKS" and a "WEB" tab. Below the header is a section titled "EDIT TASK" with form fields for "PROJECT" (set to "TEST PROJECT"), "NAME" (set to "Prepare demo data"), "DESCRIPTION" (set to "Parse XML from web-service..."), "DATE BEGIN" (set to "02.03.19"), "DATE END" (set to "17.03.19"), and "STATUS" (set to "Not started"). At the bottom is a "SAVE" button.

Рис. 98: JSF-СТРАНИЦА ДЛЯ РЕДАКТИРОВАНИЯ ЗАДАЧИ

25 ПРАКТИЧЕСКОЕ ЗАДАНИЕ ЕЕ-06

25.1 РАЗРАБОТАТЬ REST-СЕРВИСЫ JAX-RS

25.1.1 РЕАЛИЗОВАТЬ CRUD-ОПЕРАЦИИ С ИСПОЛЬЗОВАНИЕМ JAX-RS НА БАЗЕ ВЕБ-СЕРВИСОВ JAX-WS

25.1.2 РЕАЛИЗОВАТЬ REST-СЕРВИС АВТОРИЗАЦИИ

Часть III

SPRING

26 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-01

26.1 ПОДКЛЮЧИТЬ К ПРОЕКТУ SPRING MVC

26.1.1 ПОДКЛЮЧИТЬ MAVEN-ЗАВИСИМОСТИ ОТ SPRING 5

26.1.2 РЕАЛИЗОВАТЬ РЕПОЗИТОРИИ НА БАЗЕ БИНОВ ОДИНОЧЕК С ИСПОЛЬЗОВАНИЕМ СТРУКТУРЫ "МАР"

26.1.3 РАЗРАБОТАТЬ КОНТРОЛЕРЫ ДЛЯ ОТРИСОВКИ JSP-СТРАНИЦ

26.1.4 УДАЛИТЬ ИЗ ПРОЕКТА НЕНУЖНЫЕ SERVLET-КЛАССЫ

26.2 РЕКОМЕНДАЦИИ

26.2.1 ИСПОЛЬЗОВАТЬ JAVA BASED КОНФИГУРАЦИЮ SPRING

26.2.2 РЕАЛИЗОВАТЬ ПЕРЕНАПРАВЛЕНИЕ НА СТРАНИЦЫ ЛИСТИНГА ДЛЯ ОПЕРАЦИЙ СОХРАНЕНИЯ И УДАЛЕНИЯ ЗАПИСЕЙ

27 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-02

27.1 ПОДКЛЮЧИТЬ К ПРОЕКТУ SPRING ORM

27.1.1 ПОДКЛЮЧИТЬ В КАЧЕСТВЕ ORM БИБЛИОТЕКУ HIBERNATE

27.1.2 ПОДКЛЮЧИТЬ В MYSQL CONNECTOR ВЕРСИИ 5.1.4

27.1.3 РЕАЛИЗОВАТЬ РЕПОЗИТОРИИ С ИСПОЛЬЗОВАНИЕМ ВНЕДРЕНИЯ ENTITY MANAGER

27.2 РЕКОМЕНДАЦИИ

27.2.1 ХРАНИТЬ НАСТРОЙКИ ПОДКЛЮЧЕНИЯ К СУБД В ФАЙЛЕ "application.properties"

27.2.2 ИСПОЛЬЗОВАТЬ JAVA BASED КОНФИГУРАЦИЮ JPA И JTA

27.2.3 ИСПОЛЬЗОВАТЬ ТИПИЗИРОВАННЫЕ JPQL-ЗАПРОСЫ

27.2.4 ОПЕРАЦИИ ЗАПИСИ ДАННЫХ ПРОИЗВОДИТЬ ЧЕРЕЗ МЕТОДЫ ENTITY MANAGER

27.2.5 РЕАЛИЗОВАТЬ АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ ТРАНЗАКЦИЯМИ НА СЛОЕ СЕРВИСОВ

28 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-03

28.1 ПОДКЛЮЧИТЬ К ПРОЕКТУ SPRING DATA JPA

28.1.1 РЕАЛИЗОВАТЬ СЛОЙ РЕПОЗИТОРИЕВ В ВИДЕ ИНТЕРФЕЙСОВ SPRING DATA JPA

29 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-04

29.1 ПОДКЛЮЧИТЬ К ПРОЕКТУ PRIMEFACES

29.1.1 ИСПОЛЬЗОВАТЬ SPRING BEANS ДЛЯ JSF

30 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-05

30.1 РЕАЛИЗОВАТЬ SOAP ТОЧКИ ДОСТУПА

30.1.1 ИСПОЛЬЗОВАТЬ APACHE CXF ДЛЯ РЕАЛИЗАЦИИ JAX-WS ПОД SERVLET-КОНТЕЙНЕРОМ

31 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-06

31.1 РЕАЛИЗОВАТЬ REST-КОНТРОЛЛЕРЫ

31.2 РЕАЛИЗОВАТЬ REST-КЛИЕНТЫ

31.2.1 РЕАЛИЗОВАТЬ КЛИЕНТЫ К ВЕБ-СЕРВИСАМ С ИСПОЛЬЗОВАНИЕМ REST TEMPLATE

32 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-07

32.1 ПОДКЛЮЧИТЬ К ПРОЕКТУ SPRING SECURITY

32.1.1 РЕАЛИЗОВАТЬ СТРАНИЦУ АВТОРИЗАЦИИ С НЕСТАНДАРТНЫМ ДИЗАЙНОМ

32.1.2 НАСТРОИТЬ ХРАНЕНИЕ СЕССИЙ ПОЛЬЗОВАТЕЛЯ В СУБД

32.1.3 РЕАЛИЗОВАТЬ АВТОРИЗАЦИЮ ДЛЯ REST-КОНТРОЛЛЕРОВ

32.1.4 РЕАЛИЗОВАТЬ АВТОРИЗАЦИЮ ДЛЯ SOAP ENDPOINT

33 ПРАКТИЧЕСКОЕ ЗАДАНИЕ SP-08

33.1 ПЕРЕВЕСТИ ПРОЕКТ НА SPRING BOOT

- 33.1.1 РАЗРАБОТАТЬ ФАЙЛ КОНФИГУРАЦИИ "application.properties"
- 33.1.2 ДОБАВИТЬ SPRING BOOT STARTER WEB
- 33.1.3 ДОБАВИТЬ SPRING BOOT STARTER JAX-WS
- 33.1.4 ДОБАВИТЬ SPRING BOOT STARTER DATA JPA
- 33.1.5 ДОБАВИТЬ SPRING BOOT STARTER JSF
- 33.1.6 ДОБАВИТЬ SPRING BOOT TESTING

Список иллюстраций

1	ПРИВЕТСТВИЕ IDE INTELLIJ IDEA	10
2	СОЗДАНИЕ ПРОЕКТА ИЗ MAVEN ARCHETYPE	10
3	КОНФИГУРИРОВАНИЕ MAVEN ARCHETYPE	11
4	ПОДТВЕРЖДЕНИЕ КОНФИГУРАЦИИ MAVEN ARCHETYPE	11
5	КОНФИГУРИРОВАНИЕ PROJECT WORKSPACE	12
6	СОЗДАННАЯ ТОЧКА ВХОДА В ПРИЛОЖЕНИЕ	12
7	СОЗДАННЫЙ ИЗ ШАБЛОНА MAVEN-ПРОЕКТ	13
8	СТРУКТУРА MAVEN-ПРОЕКТА	14
9	ЗАПУСК ПРИЛОЖЕНИЯ ИЗ ТЕРМИНАЛА	15
10	ЗАПУСКАЕМЫЙ КЛАСС В МАНИФЕСТЕ ПРИЛОЖЕНИЯ	15
11	РЕЛИЗ ПРИЛОЖЕНИЯ С ЗАВИСИМОСТЯМИ	22
12	РЕГИСТРАЦИЯ КЛАССОВ КОМАНД	23
13	GET И SET МЕТОДЫ ОТ LOMBOK	23
14	АННОТАЦИЯ NULLABLE ОТ JETBRAINS	24
15	АННОТАЦИЯ NOTNULL ОТ JETBRAINS	24
16	MAVEN-ЗАВИСИМОСТИ ДЛЯ FASTERXML	27
17	MAVEN-ЗАВИСИМОСТИ ДЛЯ JAX-B JSON	27
18	ДАМП ДАННЫХ В ТРАНСПОРТНОМ ФОРМАТЕ ДАННЫХ XML . .	28
19	ДАМП ДАННЫХ В ТРАНСПОРТНОМ ФОРМАТЕ ДАННЫХ JSON .	28
20	КОНФИГУРАЦИЯ MAVEN SUPER POM	30
21	КОНФИГУРАЦИЯ ENUNCATE MAVEN-PLUGIN	30
22	КОНФИГУРАЦИЯ ГЕНЕРАЦИИ КЛИЕНТА ПО WSDL	31
23	ФУНКЦИЯ MD5-ХЕШИРОВАНИЯ	32
24	ФУНКЦИЯ ЦИФРОВОЙ ПОДПИСИ ДАННЫХ	33
25	СЕРВЕРНОЕ КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ	34
26	КЛИЕНТСКОЕ КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ	34
27	РЕЛИЗНЫЕ ПАПКИ КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ . .	35
28	СЕАНС ТЕСТОВОГО ПОЛЬЗОВАТЕЛЯ	36
29	СЕАНС АДМИНИСТРАТОРА ПРИЛОЖЕНИЯ	36
30	ВНЕШНИЙ КЛИЕНТ SOAPUI	37
31	ИМПОРТ WSDL-КОНТРАКТА ДЛЯ ВЕБ-СЕРВИСА СЕССИЙ . . .	37
32	ВЫЗОВ ВЕБ-ОПЕРАЦИИ ОТКРЫТИЯ СЕССИИ	38

33	ПРИМЕР ВЫЗОВА ОТКРЫТИЯ СЕССИИ	38
34	ИМПОРТ WSDL-КОНТРАКТА ДЛЯ ВЕБ-СЕРВИСА ПРОЕКТОВ	39
35	ВЫЗОВ ВЕБ-ОПЕРАЦИИ СОЗДАНИЯ ПРОЕКТА	39
36	ПРИМЕР ВЫЗОВА СОЗДАНИЯ ПРОЕКТА	40
37	ПРИМЕР ВЫЗОВА СПИСКА ПРОЕКТОВ ПОЛЬЗОВАТЕЛЯ	40
38	ПРИМЕР ОТКАЗА ДОСТУПА ПО СЕССИИ	41
39	ОПИСАНИЕ ТИПОВ ДАННЫХ ОТ ENUNCATE	42
40	ОПИСАНИЕ ВЕБ-СЕРВИСОВ ОТ ENUNCATE	42
41	MAVEN-ЗАВИСИМОСТЬ MYSQL JDBC CONNECTOR	44
42	ПРИМЕР ВЫБОРКИ ЗАПИСЕЙ ИЗ MYSQL	44
43	DDL ТАБЛИЦЫ "ПОЛЬЗОВАТЕЛИ"	45
44	DDL ТАБЛИЦЫ "СЕССИИ"	45
45	DDL ТАБЛИЦЫ "ПРОЕКТЫ"	46
46	DDL ТАБЛИЦЫ "ЗАДАЧИ"	46
47	ТАБЛИЦА "ПОЛЬЗОВАТЕЛЬ"	47
48	ТАБЛИЦА "СЕССИЯ"	47
49	ТАБЛИЦА "ПРОЕКТ"	48
50	ТАБЛИЦА "ЗАДАЧА"	48
51	JAVA BASED КОНФИГУРАЦИЯ MYBATIS	50
52	MAVEN-ЗАВИСИМОСТЬ MYBATIS	50
53	XML BASED КОНФИГУРАЦИЯ MYBATIS	51
54	MAVEN-ЗАВИСИМОСТЬ HIBERNATE CORE	53
55	JAVA BASED КОНФИГУРАЦИЯ HIBERNATE	53
56	КОНФИГУРАЦИЯ PERSISTENCE UNIT	54
57	MAVEN-ЗАВИСИМОСТЬ HIBERNATE C3P0	54
58	КОНФИГУРАЦИЯ "META-INF/beans.xml"	56
59	ТОЧКА ВХОДА В CDI-ПРИЛОЖЕНИЕ	56
60	MAVEN-ЗАВИСИМОСТЬ WELD-CDI	56
61	MAVEN-ЗАВИСИМОСТЬ APACHE DELTASPIKE	57
62	КОНФИГУРАЦИЯ ВНЕДРЕНИЯ ENTITY MANAGER В CDI	58
63	MAVEN-ЗАВИСИМОСТЬ HAZELCAST & HIBERNATE	59
64	КОНФИГУРАЦИЯ СУЩНОСТИ JPA HIBERNATE	59
65	XML-КОНФИГУРАЦИЯ HAZELCAST	60

66	КОНФИГУРАЦИЯ SECOND LEVEL CACHE	60
67	МОДУЛИ АПАСНЕ ДЛЯ БАЛАНСИРОВКИ НАГРУЗКИ	61
68	КОНФИГУРАЦИЯ ВИРТУАЛЬНОГО ХОСТА АПАСНЕ	61
69	ЗАПУСК НОД СЕРВЕРА НА СМЕЩЕННЫХ ПОРТАХ	62
70	ОТВЕТ ОТ КЛАСТЕРА НА ЗАПРОС ИНФОРМАЦИИ О НОДЕ	62
71	ЛОГИ ОБЪЕДИНЕНИЯ НОД HAZELCAST В КЛАСТЕР	62
72	МОДУЛИ АПАСНЕ ДЛЯ БАЛАНСИРОВЩИКА	63
73	ВИРТУАЛЬНЫХ ХОСТ АПАСНЕ ДЛЯ БАЛАНСИРОВЩИКА	63
74	КОНФИГУРАЦИЯ ХОСТОВ WINDOWS	64
75	АПАСНЕ ВЕБ-СЕРВЕР ДЛЯ ОС WINDOWS	64
76	НОДА СЕРВЕРА ПРИЛОЖЕНИЯ НА ПОРТЕ 7070	65
77	НОДА СЕРВЕРА ПРИЛОЖЕНИЯ НА ПОРТЕ 8080	65
78	НОДА СЕРВЕРА ПРИЛОЖЕНИЯ НА ПОРТЕ 9090	66
79	ИНФОРМАЦИЯ О НОДАХ КЛАСТЕРА	66
80	MAVEN-ЗАВИСИМОСТЬ SPRING CONTEXT	67
81	MAVEN-ЗАВИСИМОСТЬ SPRING DATA JPA	67
82	JAVA BASED КОНФИГУРАЦИЯ SPRING JPA	68
83	КОНФИГУРАЦИЯ ТОМСАТ MAVEN-PLUGIN	71
84	КОНФИГУРАЦИЯ JETTY MAVEN-PLUGIN	71
85	ПРИВЕТСТВИЕ IDE ДЛЯ СОЗДАНИЯ ПРОЕКТА	72
86	ВЫБОР ШАБЛОНА MAVEN-ПРОЕКТА ДЛЯ ВЕБ-ПРИЛОЖЕНИЯ .	72
87	НАСТРОЙКА ШАБЛОНА MAVEN-ПРОЕКТА	73
88	ПОДТВЕРЖДЕНИЕ НАСТРОЕК MAVEN-ПРОЕКТА	73
89	НАСТРОЙКА РАБОЧЕГО ПРОСТРАНСТВА ПРОЕКТА	74
90	ОТОБРАЖЕНИЕ СТРУКТУРЫ WAR-ПРОЕКТА В IDE	74
91	JSP-СТРАНИЦА ДЛЯ ОТОБРАЖЕНИЯ СПИСКА ПРОЕКТОВ	75
92	JSP-СТРАНИЦА ДЛЯ РЕДАКТИРОВАНИЯ ПРОЕКТА	75
93	JSP-СТРАНИЦА ДЛЯ ОТОБРАЖЕНИЯ СПИСКА ЗАДАЧ	76
94	JSP-СТРАНИЦА ДЛЯ РЕДАКТИРОВАНИЯ ЗАДАЧИ	76
95	JSF-СТРАНИЦА ДЛЯ ОТОБРАЖЕНИЯ СПИСКА ПРОЕКТОВ	80
96	JSF-СТРАНИЦА ДЛЯ РЕДАКТИРОВАНИЯ ПРОЕКТА	80
97	JSF-СТРАНИЦА ДЛЯ ОТОБРАЖЕНИЯ СПИСКА ЗАДАЧ	81
98	JSF-СТРАНИЦА ДЛЯ РЕДАКТИРОВАНИЯ ЗАДАЧИ	81