

Степанов П.А.

Технологии разработки серверных информационных систем

Методическое пособие

ГУАП

Санкт-Петербург

-2017-

Оглавление

Введение	3
График выполнения лабораторных работ.....	3
Технические требования	4
Требования к отчету	4
Лабораторная работа №1. Разработка простого серверного приложения J2EE с использованием сервлетов	4
Лабораторная работа №2. Разработка ресурса REST/JSON сервиса.....	4
Лабораторная работа №3. Разработка простого AJAX приложения Spring.....	5
Лабораторная работа №4. Разработка формы логина	5
Лабораторная работа №5. Разработка приложения с использованием thymeleaf	5
Лабораторная работа №6. Разработка приложения с использованием Hibernate	5
Лабораторная работа №7. Разработка приложения с асинхронной очередью сообщений	5
Лабораторная работа №8. Разработка микросервиса.....	6
Варианты заданий	6

Введение

Предмет “Технологии разработки серверных информационных систем” призван дать обучающимся представление о методах построения современных приложений, размещаемый в сети Интернет (интранет). Лабораторные работы по этому предмету знакомят обучающихся со следующими технологиями:

- сервлеты
- инъекция зависимостей (DI, IoC, Spring)
- аспектно-ориентированное программирование (AOP, AspectJ)
- генерация страниц на сервере (Thymeleaf)
- генерация страниц на клиенте (AJAX)
- объектно-реляционные отображения (ORM, Hibernate, JPA)
- Управление транзакциями (JTA, XA)
- архитектура, управляемая событиями (EDA)
- Облачная архитектура
- Контейнерное развертывание

Страница с материалами учебного курса <http://stepanoff.info/trsis/index.html>

График выполнения лабораторных работ

Максимальное количество баллов за лабораторные работы – 72.

Для допуска к экзамену либо зачету требуется сдать ВСЕ лабораторные работы.

В таблице представлен максимальный рейтинг за лабораторную работу в зависимости от недели сдачи

	Лаб. 1	Лаб. 2	Лаб. 3	Лаб. 4	Лаб. 5	Лаб. 6	Лаб. 7	Лаб. 8
Нед. 1	9	9	9	9	9	9	9	9
Нед. 2	9	9	9	9	9	9	9	9
Нед. 3	8	9	9	9	9	9	9	9
Нед. 4	8	9	9	9	9	9	9	9
Нед. 5	7	8	9	9	9	9	9	9
Нед. 6	7	8	9	9	9	9	9	9
Нед. 7	6	7	8	9	9	9	9	9
Нед. 8	6	7	8	9	9	9	9	9
Нед. 9	5	6	7	8	9	9	9	9
Нед. 10	5	6	7	8	9	9	9	9
Нед. 11	4	5	6	7	8	9	9	9
Нед. 12	4	5	6	7	8	9	9	9
Нед. 13	3	4	5	6	7	8	9	9
Нед. 14	3	4	5	6	7	8	9	9
Нед. 15	2	3	4	5	6	7	8	9
Нед. 16	2	3	4	5	6	7	8	9
Нед. 17	1	2	3	4	5	6	7	8
Нед. 18	1	2	3	4	5	6	7	8

Технические требования

Рекомендуется разрабатывать программное обеспечение на платформе Java SE 8 <http://www.oracle.com/technetwork/java/javase/downloads/index.html> и Spring 4. Spring рекомендуется использовать в варианте spring-boot.

В качестве базы данных рекомендуется использовать СУБД, не требующее установки, как например Derby или H2.

В качестве билд-системы рекомендуется использовать Maven 2 <https://maven.apache.org/>

В качестве сервера приложений рекомендуется использовать встроенный в приложение tomcat.

Выполнение вышеуказанных рекомендаций позволит сократить набор устанавливаемых приложений до только платформы Java SE а набор загружаемых библиотек только до Maven 2. Все остальные библиотеки и ресурсы Maven загрузит и установит самостоятельно.

В качестве IDE рекомендуется использовать IntelliJ Idea, Eclipse или Netbeans

Использование других платформ разрешается, но Вы должны твердо осознавать, что можете остаться без технической поддержки преподавателем.

Требования к отчету

Отчет должен содержать

- Титульный лист
- Текст и вариант задания
- Описание разрабатываемого продукта
- Текст основных фрагментов кода
- Отчет должен быть отправлен в личный кабинет студента через систему документооборота ГУАП (информация о конкретном процессе будет уточнена)

Лабораторная работа №1. Разработка простого серверного приложения J2EE с использованием сервлетов

Выполните следующие задачи.

- 1 В соответствии со своим вариантом разработайте набор экранных форм приложения (порядка 5)
- 2 Соберите проект веб-приложения (war) на Maven (можно без использования Spring)
- 3 реализуйте формы средствами сервлетов. Проект должен как минимум содержать формы просмотра, добавления и удаления данных.
- 4 Аргументируйте почему были выбраны HTTP методы GET, POST или оба.
- 5 Использовать базу данных можно, но не обязательно

Лабораторная работа №2. Разработка ресурса REST/JSON сервиса

- 1 Подключите к проекту Maven фреймворк Spring (spring-boot)

- 2 Определите перечень Rest-сервисов, выполняющих те же действия, что и в лабораторной работе 1. Внимательно отнеситесь к вопросу какой HTTP метод использует тот или иной сервис и какие коды HTTP он может возвращать. Реализуйте эти сервисы.

Методы HTTP: <https://ru.wikipedia.org/wiki/HTTP#.D0.9C.D0.B5.D1.82.D0.BE.D0.B4.D1.8B>

Коды состояний HTTP:

https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BA%D0%BE%D0%B4%D0%BE%D0%B2_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D1%8F_HTTP

Лабораторная работа №3. Разработка простого AJAX приложения Spring

- 1 Создайте один или несколько форм и контроллеров для страниц приложения. Подключите библиотеку angularJS.
- 2 Реализуйте задачи из лабораторной работы 1 средствами AJAX и REST-сервисов, разработанных в предыдущей лабораторной работе.

Лабораторная работа №4. Разработка формы логина

- 1 Добавьте в приложение поддержку одной из моделей безопасности средствами spring-security
- 2 Защитите ресурсы, требующие запись и введите аудит.

Лабораторная работа №5. Разработка приложения с использованием thymeleaf

- 1 Реализуйте задачи из лабораторной работы 1 средствами Thymeleaf
- 2 Обратите внимание на локализацию приложения. Русский язык должен поддерживаться.

Лабораторная работа №6. Разработка приложения с использованием Hibernate

- 1 Добавьте базу данных, модель данных, JPA репозитории и сервисы.
- 2 Реализуйте операции чтения/записи из одной из предыдущих лабораторных работ (3 или 5) с помощью JPA

Лабораторная работа №7. Разработка приложения с асинхронной очередью сообщений

Целью работы является реализация простой системы распределенной репликации (“писатели-читатели”).

- 1 Скачайте и разверните Apache Kafka
- 2 Модифицируйте свое приложение со встраиваемой базой данных так, чтобы его можно было запустить в нескольких экземплярах на разных портах
- 3 Реализуйте в рамках своего приложения Producer и Consumer такие, что

- a. Producer при каждой операции записи оповещает соответствующий топик
 - b. Consumer при получении информации из топика записывает обновление в локальную (встроенную в приложение) базу
- 4 Продемонстрируйте, что информация, записанная одним приложением, доступна второму приложению.

Лабораторная работа №8. Разработка микросервиса

Подготовьте Ваше приложение к разворачиванию в облачном сервисе или компоненте Docker.

Используйте инструкции проекта spring-boot <https://spring.io/guides/gs/spring-boot-docker/>

Скачайте демонстрационный проект командой `git clone https://github.com/spring-guides/gs-spring-boot-docker.git`

В папке complete модифицируйте текст программы и pom файл так, чтобы строилось Ваше приложение из лабораторной работы номер 3. Убедитесь что оно по прежнему строится.

ОПЦИОНАЛЬНО

Установите докер, постройте и запустите контейнер. Кто справится - тот большой молодец.

Варианты заданий

1. Книжный магазин либо библиотека
2. Поликлиника (запись на прием к врачу)
3. Расписание занятий в институте
4. Расписание поездов, самолетов, кораблей
5. Планирование покупок в магазине
6. Учет оценок студентов за лабораторные работы
7. Ведение списка литературы согласно последнему актуальному ГОСТу
8. Учет трат в бюджете семьи
9. Складской учет
10. Магазин электроники
11. Спортивные соревнования
12. Продажа автомобилей
13. Торговля акциями на бирже
14. Сдача недвижимости в аренду
15. Коллекционирование (нумизматика, филателия и пр).
16. Система безопасности предприятия (помещения, люди, права на вход)
17. Учет конфигурации сетевого оборудования сети предприятия
18. Учет показателей потребления электроэнергии и водоснабжения
19. База данных с рецептами блюд
20. База данных предметов живописи, скульптуры и пр. (музей)
21. Магазин компьютерных игр
22. Магазин музыкальных композиций
23. Учет содержимого холодильника (со сроками годности)
24. Учет медицинских лабораторных показателей пациента поликлиники (изменение с течением времени гемоглобина в крови, сахара, АЛТ, АСТ и тому подобных параметров)

25. Ведение списка группы (ФИО, вариант задания, число сданных лабораторных, рейтинг) с учетом постоянно меняющегося количества студентов