Geekbrains

**Исследование особенностей разработки небольших масштабируемых приложений с использованием микросервисной архитектуры**

Программа: Разработчик - Программист

Специализация: Веб-разработка на JAVA

Зотов Алексей Викторович

Москва

2024

**Основные требования к оформлению дипломного проекта**

1. Текст проекта оформляйте только с одной стороны листа формата А4. При наборе используйте межстрочный интервал 1,5.  
   Объемные таблицы/иллюстрации возможно размещать на листах А3, А2 и выносить в приложения.
2. При наборе информации, используйте шрифт Times New Roman с выравниванием текста по ширине. Рекомендуем кегль — 12–14, для некоторых структурных единиц работы можно выбрать полужирное выделение.
3. Поля страницы, где размещается текст, выставляйте по определенным показателям: нижнее/верхнее – 20 мм, правое – 10 мм, левое (для последующей прошивки) – 30 мм.
4. Все листы дипломного проекта нумеруются по порядку, начиная с введения (используется сквозная нумерация). Номер указывайте в центре нижней части страницы без точки.
5. Заголовок раздела с одним подразделом или пунктом не нумеруется.
6. После текста перед следующим заголовком установите одиночный интервал.
7. После заголовка перед подзаголовком проставьте двойной интервал.
8. Имена собственные — наименования компаний/изделий, фамилии учёных и т. д., указывайте на языке оригинала.
9. Каждый структурный элемент начинается с нового листа.

*Вся работа должна составлять 50+ страниц формата А4, включая приложения.*

|  |
| --- |
| 💡 Инструкция по работе с шаблоном.   1. Создайте копию данного шаблона. Файл - Создать копию. 2. Изучите шаблон и начинайте работу. |

**Содержание**

[Содержание](#_heading=h.7mt9x4b37ce)

[Введение](#_heading=h.1fob9te)

[Теоретическая и практическая главы](#_heading=h.hst8x8f1r8tz)

[Заключение](#_heading=h.dhwmmf3wq1kk)

[Список используемой литературы](#_heading=h.m8jvbwhv7ujg)

[Приложения](#_heading=h.s6k7t2lwcp0k)

# Как оформить содержание?

В содержании дипломного проекта необходимо озаглавить основные блоки и подпункты работы с указанием страниц.

### Пример

*Данный пример не является эталонным, лишь один из вариаций.*

Введение (2–3 стр.)

Глава 1. Основы ручного тестирования веб-приложений (~15 стр.)

1.1 Что такое веб-приложение, его особенности

1.2 Что такое тестирование, кто такой тестировщик. Зачем нужно тестирование и когда оно начинается

1.3 Этапы ручного тестирования веб-приложений

1.4 Источники требований: что такое техническое задание и как проводить ручное тестирование, если техническое задание отсутствует

Глава 2. Подготовка к ручному тестированию веб-приложения (~15 стр)

2.1 Составление чек-листов

2.2 Составление тест-кейсов

2.3 Разработка тест-плана

2.4 Тест-дизайн и тест-аналитика

2.5 Баг-репорты и багтрекинговая система

Глава 3. Проведение тестирования и разработка предложений по улучшению тестирования Интернет-магазина РИВ ГОШ (~20 стр.)

3.1 Тестирование функциональности

3.2 Тестирование юзабилити

3.3 Тестирование интерфейса

3.4 Тестирование совместимости

3.5 Тестирование производительности

3.6 Тестирование безопасности

3.7 Подведение итогов. Разработка предложений по улучшению ручного тестирования веб-приложения РИВ ГОШ

Заключение (~ 4 стр.)

Список используемой литературы

Приложения

# Введение

**Тема проекта:** Исследование особенностей разработки небольших масштабируемых приложений с использованием микросервисной архитектуры

**Цель:** Рассмотреть основные подходы к дизайну и развертыванию приложений и используемому инструментарию. Создать шаблон приложения, предусматривающий легкую адаптацию к конкретным нуждам разработчиков при развертывании небольших приложений (pet projects)

**Какие задачи решает:** В процессе обучения или же в процессе разработки отдельных сервисов возникает потребность посмотреть на работу разработанного сервиса в условиях, кросс взаимодействия с другими сервисами. Будь то сервис уведомлений, аутентификации и т. п.

**Задачи:**

1. Изучение литературы

2. Создать базовый проект

3. Выбрать инструментарий для разработки, мониторинга и управления микросервисным приложением

4. Изучить инструментарий принципы внедрения и внедрить в базовое приложение

5. Изучить процесс развертывания таких приложений с использованием Docker

**Инструменты:** Postman, FastRequest, PostgreSQL, Docker, Docker-Compose, Spring Cloud, Eureka, Open Feign, Zipkin, API Gateway, RabbitMQ

Состав команды: Зотов Алексей Викторович (программист)

# Теоретическая и практическая главы

**Основная часть** составляет 80% от всего объёма написанного. В нём последовательно рассказывается, как были решены вопросы, перечисленные во вступлении, какие цели достигнуты и с каким результатом.

Содержит 2 главы — теоретическую и практическую, где автор выражает свою позицию относительно гипотез.

В практической части студент должен пошагово описать:

* как он делал проект;
* какие инструменты использовал в работе;
* на каком этапе использовал инструменты и как они помогли в достижении цели работы.

В основной части должны быть практические примеры, результаты и выводы после каждого элемента исследования.

# **Глава 2. Разработка приложения**

**2.1 Подготовка базовой архитектуры приложения**

Разработка приложения будет вестись с использованием IntellyJ IDEA Community edition с использованием Maven framework.

Приложение будет разрабатываться как мультимодульное, в составе которого будет «родительский» pom.xml файл с основными зависимостями. Сами сервисы будут включены в проект в качестве отдельных модулей. Каждый из модулей будет включать в себя как собственные, необходимые зависимости, так и «родительские» зависимости. Проект будет обновляться и совершенствоваться в процессе разработки, поэтому было принято решение создать пустой Maven проект без использования “spring initializr” – [https://start.spring.io](https://start.spring.io/)

**2.1.1 Развертывание базового приложения и настройка pom.xml**

В папке с проектом через консольную команду создаем базовый пустой maven проект (рис. 1).

Рисунок 1

Далее настраиваем проект. Версия Java: Oracle open jdk 17.0.10. Поскольку данный проект будет являться «родительским» для остальных модулей, удаляем папку src, а также добавляем 2 собственные папки к проекту а именно: папка с проектом диплома и папка с диаграммами.

«Родительский» pom файл настраиваем с использованием dependency Management блока в который подключим артефакт spring dependency и зафиксируем версию 3.2.5 на уровне свойств. Основная задача, фиксирование версий используемых зависимостей и внедрение на уровень проекта основных зависимостей, который абсолютно точно понадобятся во всех дочерних модулях. Для удобства работы подключим на уровень проекта Lombok и spring boot test. В подраздел plugin подключим арефакт maven-plugin.

На этом, этап подготовки базового проекта завершен, собираем pom файл, подгружаем зависимости и размещаем первый коммит на GitHub -[*Initial commit*](https://github.com/AVZotov/Tech-Specialization-diploma-microservices/commit/df8f748734800529b78367ae7ebf1ee3d667ce10)

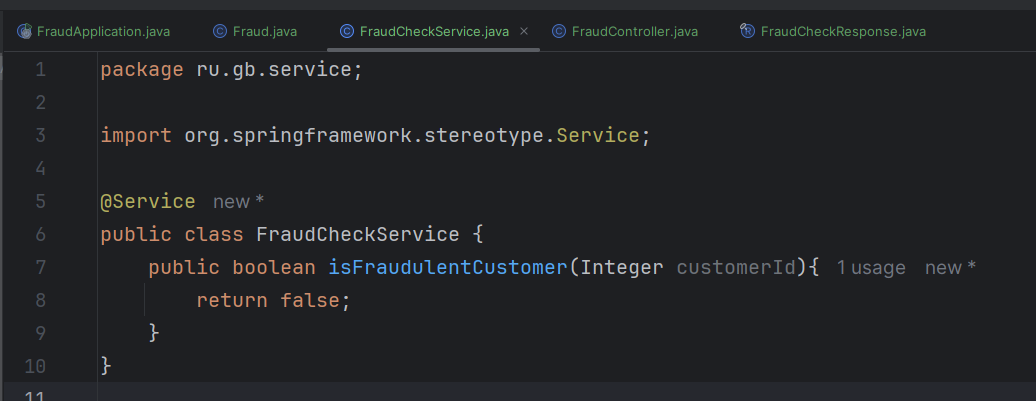
***2.1.2 Создание сервисных приложений***

Как упоминалось в первой главе, дизайн приложения будет состоять из 3 сервисов и ряда сопутствующих модулей, с минимальным наполнением самих сервисов, предполагая что эти сервисы будут использоваться как шаблоны для тестирования или подмены необходимыми разработчику сервисами. Для эмуляции каждый из сервисов будет работать с собственной базой данных развернутой на хостовой машине в docker контейнере

Первым сервисом будет сервис *customer-service*. Мы хотим использовать базовое наполнение самого сервиса и реализовать базовый набор функционала для работы с репозиторием развернутом на PostgreSQL, с JPA драйвером, а также контроллер. Для этого мы добавим новый модуль в проект. В зависимости мы добавляем Spring web. Для удобства идентификации запущенных сервисов, добавим в ресурсы проекта файл *banner.txt* с названием сервиса, а также файл *application.yml* с базовыми настройками названия проекта, портом, в дальнейшем же добавим сюда настройки для работы сервиса с собственной базой данных.

Второй сервис, который мы также добавим в проект *fraud-service* будет максимально похож на первый сервис. Более того тут важно подчеркнуть что оба эти сервиса будут зависеть друг от друга и будут взаимодействовать только синхронно. Первый сервис будет регистрировать покупателей в собственной базе данных и направлять запрос во второй сервис, который будет имитировать проверку, является ли добавленный пользователь «настоящим» или валидным, либо это «мошенник» и дополнительно фиксировать данные в своей базе данных.

Поскольку данная работа в первую очередь направленна не на обработку персональных данных, а разработку взаимодействия между микросервисами, было решено максимально абстрагировать процесс проверки (рис. 2)

Рисунок 2

Помимо базовой настройки самих сервисов, мы добавим каждому микросервису отдельный порт, что бы избежать конфликтов.

# Заключение

В заключение необходимо включить следующее:

1. Краткие и ёмкие теоретические и практические выводы, которые были получены во время анализа теоретической базы и практического исследования.
2. Оценка проведённого исследования, описание его результатов.
3. Практическая значимость работы, рекомендации и планы на дальнейшие исследования.
4. Общий итог — достижение цели, выполнение задач, доказательство гипотезы.
5. Предложения по совершенствованию объекта исследования.

# Список используемой литературы

Здесь нужно будет указатьсписок используемой литературы, ссылки на все ресурсы, которые нужны были для создания проектной работы.

Основные правила оформления использованной литературы и ресурсов:

1. Каждый источник упоминается единожды, независимо от того, насколько часто на него ссылаются.
2. Список литературы оформляется в алфавитном порядке по фамилии автора, сначала русскоязычная литература, затем иностранная, далее интернет-сайты.
3. Библиографическая запись обязательно включает:
   * Фамилию автора или фамилии их группы, инициалы (при наличии).
   * Название статьи, книги, справочника, закона, иного документа.
   * Населённый пункт, в котором был издан источник, наименование издательства.
   * Год публикации.
   * Число страниц.

## Пример

* *Книга: Автор. Название книги. Город: Издательство, Год.*
* *Статья: Автор. "Заголовок статьи." Название журнала Том, номер (Год): страницы.*

# Приложения

В **приложения** обычно входят артефакты, получившиеся в процессе создания проекта:

1. Объёмные графики и таблицы, которые не помещаются на лист А4.
2. Длинные математические формулы и расчёты по ним.
3. Характеристики аппаратуры, которая использовалась для проведения исследования.
4. Авторские методики.
5. Вспомогательный материал: тесты, карточки, схемы, рисунки.
6. Материалы, полученные на предприятии: отчёты, прочие документы.