

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

Отчет по учебной практике
Тема НИР: Разработка, развёртка и поддержка приложения,
написанного на Java с использованием Spring.

Студентка гр. 5304

Фомченко О.А.

Санкт-Петербург
2019

СОДЕРЖАНИЕ

1. УЧАСТИЕ В УЧЕБНОМ ПРОЦЕССЕ БАКАЛАВРОВ.....	3
2. ОБЗОР ЛИТЕРАТУРЫ.....	4
2.1. Тема НИР.....	4
2.2. Основные определения	4
2.3. Введение	5
2.4. Укрупненное описание постановки задачи	6
2.5. Обзор литературы.....	6
2.6. План работы на весенний семестр.....	8
2.7. Выводы	8
2.8. Список литературы.....	9

1. УЧАСТИЕ В УЧЕБНОМ ПРОЦЕССЕ БАКАЛАВРОВ

В ходе прохождения учебной практической работы с 23.09.2019 по 23.12.2019 на занятиях по дисциплине «Информационные технологии и базы данных в прикладных коммуникациях» осуществлялось взаимодействие с бакалаврами и помощь преподавателям, ведущим практические занятия.

Во время учебных занятий проверялись теоретические и практические знания студентов, осуществлялся прием практических заданий, проверка контрольных работ и содействие учебному процессу в целом.

По итогу работы со студентами все поставленные задачи были выполнены и был получен важный опыт преподавательской деятельности.

Отзыв руководителя учебной практики прилагается.

2. ОБЗОР ЛИТЕРАТУРЫ

2.1. Тема НИР

Разработка, развёртка и поддержка приложения, написанного на Java с использованием Spring.

2.2. Основные определения.

Spring^[1] – это платформа для разработки приложений на языке Java, позволяющая быстро создавать эффективные приложения, работающие в сети;

Travis CI^[7] — распределённый веб-сервис для сборки и тестирования программного обеспечения, использующий GitHub в качестве хостинга исходного кода;

Непрерывная интеграция (Continuous Integration – CI)^[2] — это практика разработки программного обеспечения, которая заключается в слиянии рабочих копий в общую основную ветвь разработки несколько раз в день и выполнении частых автоматизированных сборок проекта для скорейшего выявления потенциальных дефектов и решения интеграционных проблем;

Непрерывная доставка (Continuous delivery – CD)^[2] – является продолжением непрерывной интеграции, дающая возможность быстро и эффективно выпускать новые изменения для своих клиентов. Это означает, что помимо автоматизации тестирования также возможность автоматизировать процесс выпуска и возможность в любой момент развернуть приложение, нажав кнопку;

Непрерывное развертывание (Continuous deployment – CD)^[7] – идет на один шаг дальше, чем непрерывная доставка. С этой практикой каждое изменение, которое проходит все стадии производственного процесса, передается клиентам. Вмешательство человека не требуется, и только неудачный тест не позволит внедрить новое изменение в производство;

Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации. Позволяет

«упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть перенесён на Linux-систему, а также предоставляет среду по управлению контейнерами ;

GitHub^[3] — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки;

Linux^[4] – семейство Unix-подобных операционных систем на базе ядра Linux. Как и ядро Linux, системы на его основе как правило создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения.

2.3. Введение

Разработка промышленных приложений на платформе J2EE всегда считалась достаточно непростым занятием. Spring Framework придуман специально для того, чтобы упростить разработку enterprise-приложений.

Spring - это, можно сказать, архитектурный фреймворк, потому что он придуман не столько для выполнения какой-то прикладной задачи, а для обеспечения лучшей масштабируемости, возможности более простого тестирования и более простой интеграции с другими фреймворками (например, уже упомянутым Struts или Hibernate). Благодаря этому писать большие приложения становится проще - разработчики просто избегают ряда проблем, связанных с созданием enterprise-приложений, вместо того, чтобы их решать.

Основные преимущества Spring-a:

- Возможность разработки приложения как набора слабосвязанных (loose-coupled) компонентов. Чем меньше компоненты приложения знают друг о друге, тем проще разрабатывать новый и поддерживать существующий функционал приложения.
- Spring поощряет модульность. Компоненты можно добавлять и удалять (почти) независимо друг от друга. В принципе, приложение можно разработать таким образом, что оно даже не будет знать, что управляется Spring'ом.
- Spring заметно упрощает модульное тестирование (unit-testing).

- Spring сильно облегчает инициализацию и настройку компонентов приложения, позволяя гибко настраивать приложение без существенных изменений Java-кода.

Именно поэтому данная работа ставит перед собой задачи исследования фреймворка Spring в качестве инструмента для разработки приложения, а также для автоматизации процессов тестирования, построения и развертывания приложения.

2.4. Укрупненное описание постановки задачи

Целью работы является исследование и разработка web-приложения на Java с использованием современного фреймворка Spring.

Поставлены следующие задачи:

- обзор аналогичных методов и продуктов;
- реализация в web-проекте;
- тестирование в реальных условиях;
- разработка итогового решения.

2.5. Обзор литературы

В результате обзора литературы были найдены следующие основные составляющие фреймворка, которые будут раскрыты в НИР:

- IoC (Inversion of Control) контейнер;
- AOP-фреймворк (включая интеграцию с AspectJ);
- Data Access фреймворк;
- Transaction management;
- MVC-фреймворк;
- Remote Access фреймворк;
- Batch processing;
- Фреймворк аутентификации и авторизации;
- Remote Management;

- Messaging-фреймворк;
- Testing-фреймворк.

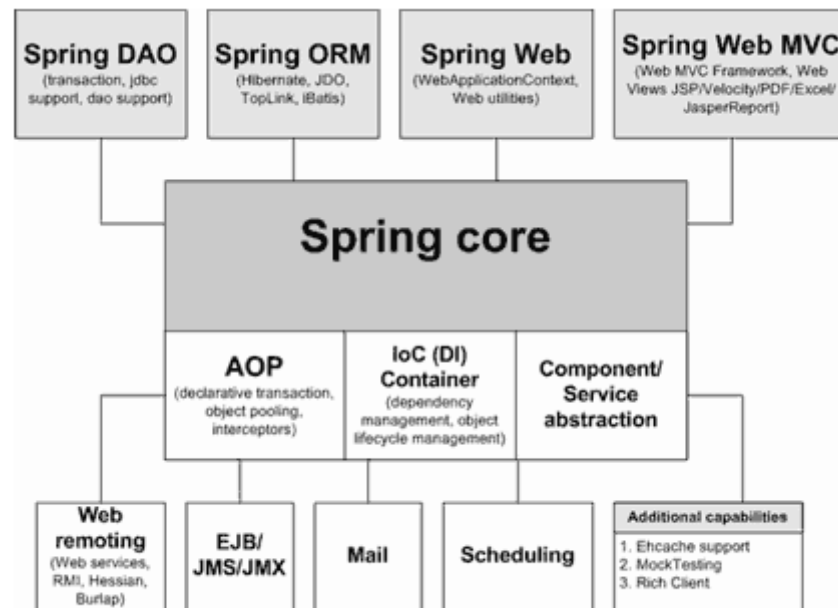


Рисунок 1. Схема фреймворков, включенных в Spring.

Для развёртки приложения и реализации в нем CI/CD мы будем использовать Travis CI.

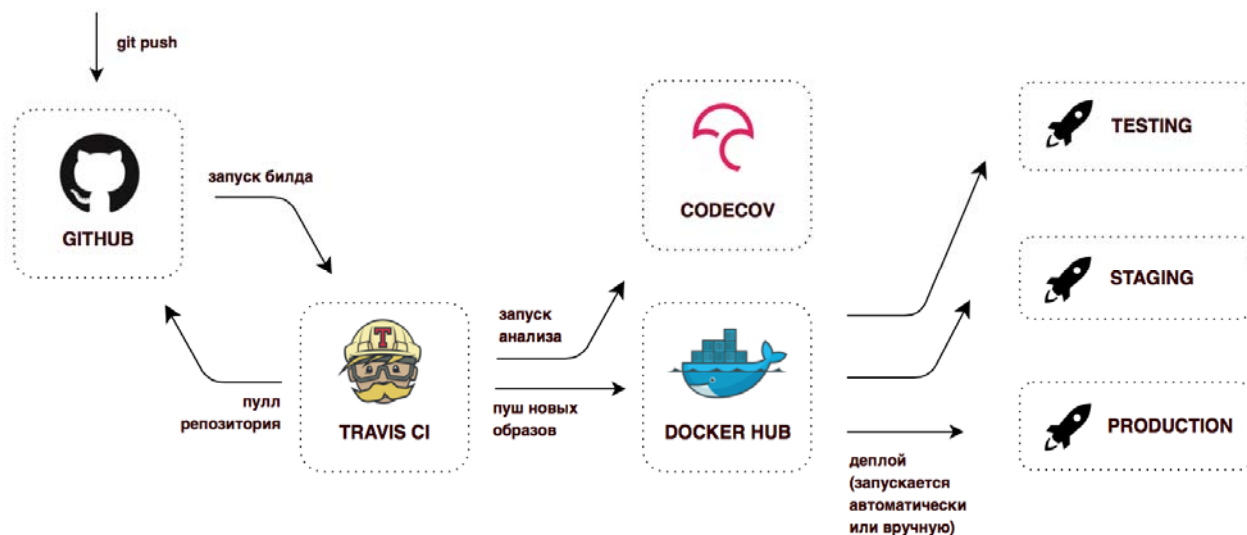


Рисунок 2. Схема простейшего Continuous Delivery workflow.

Развертывание микросервисной системы с большим количеством движущихся частей и взаимосвязанностью — задача очевидно более комплексная, нежели деплой монолитного приложения. Без

автоматизированной инфраструктуры вся история превратится в бесконечную боль и трату времени.

2.6. План работы на весенний семестр

План работы на весенний семестр расписан в таблице 1.

Таблица 1. План работы на весенний семестр

№ п/п	Наименование работ	Срок выполнения
1	Изучение предметной области и инструментов, детализация поставленных задач, обоснование актуальности разработки	03.02 – 29.02
2	Выбор метода достижения поставленных задач	01.03-15.03
3	Настройка реального приложения для возможности проведения испытаний с выбранным решением	16.03-31.03
4	Настройка сервера, настройка приложения для конкретного сервера, реализация выбранного решения	01.04-31.04
5	Подведение результатов работы и оформление НИР	01.05-31.05

2.7. Выводы

Уже тяжело найти компанию, которая разрабатывает Java Enterprise приложения, и не использует Spring. А все потому, что это огромный фреймворк, который позволяет быстро создавать большие и качественные приложения, не нанося при этом удар по безопасности данных или функциональности.

Spring имеет встроенный Testing-фреймворк, позволяющий контролировать корректность написанного кода, а при помощи таких инструментов, как Docker, Git и Travis CI мы сможем поддерживать непрерывную интеграцию, непрерывную доставку и непрерывное развертывание приложения.

2.8.Список литературы

1. Spring documentation // . URL : <https://spring.io/docs>.
2. How to set up CI/CD Pipeline // Medium. URL: <https://medium.com/@mosheezderman/how-to-set-up-ci-cd-pipeline-for-a-node-js-app-with-jenkins-c51581cc783c> (дата обращения: 14.12.2019).
3. How to CI and CD Application Using GitHub Actions // Medium. URL: <https://blog.bitsrc.io/https-medium-com-adhasmana-how-to-do-ci-and-cd-of-node-js-application-using-github-actions-860007bebae6> (дата обращения: 14.12.2019).
4. Now lets add some steps to our pipeline! Find problems in your code with ESLint // DEV. URL: <https://dev.to/paulasantamaria/improving-your-ci-pipeline-for-node-js-2aab> (дата обращения: 14.12.2019).
5. Continuous Integration для новичков // Хабр. URL: <https://habr.com/ru/post/352282/> (дата обращения: 14.12.2019).
6. Applying CI/CD to Java Apps Using Spring Boot .URL: <https://dzone.com/articles/applying-cicd-to-java-apps-using-spring-boot>.
7. Микросервисная архитектура, Spring Cloud и Docker. URL: <https://habr.com/ru/post/280786/>.
8. Знакомимся со Spring Framework // . URL: <https://www.kv.by/archive/index2009291108.htm>.
9. Spring Boot // . URL: <https://spring-projects.ru/projects/spring-boot/>.

Приложение(На руках есть подписанный оригинал):

**Отзыв
о прохождении учебной практики
студентом гр. 5304**

Фомченко Олегом Андреевичем,
в период прохождения учебной практики, выступая помощником в
принятии лабораторных работ по дисциплине “Информационные
технологии и базы данных в прикладных коммуникациях” у студентов 1-
ого курса бакалавриата, студент
Фомченко О.А. показал себя ответственным и внимательным
исполнителем.
Студент успешно справился со всеми поставленными задачами и
заслуживает оценку “отлично”.

_____ Бушмакин А.Л.