

## Site development **Tutorials World** via cloud platform **OpenShift**

**Chernova E.<sup>1</sup>, Popova Ju.<sup>2</sup>, Molchanova N.<sup>3</sup>, Molchanova N.<sup>4</sup>**

### Разработка сайта **TutorialsWorld** посредством облачной платформы **OpenShift**

**Чернова Е. В.<sup>1</sup>, Попова Ю. В.<sup>2</sup>, Молчанова Нат. Н.<sup>3</sup>, Молчанова Над. Н.<sup>4</sup>**

<sup>1</sup>Чернова Екатерина Владимировна / *Chernova Ekaterina Vladimirovna* – студент;

<sup>2</sup>Попова Юлия Валерьевна / *Popova Julija Valer'evna* – студент;

<sup>3</sup>Молчанова Наталия Николаевна / *Molchanova Natalija Nikolaevna* – студент;

<sup>4</sup>Молчанова Надежда Николаевна / *Molchanova Nadezhda Nikolaevna* – студент,

кафедра компьютерной безопасности и математического обеспечения информационных систем,  
факультет математики и информационных технологий,  
Оренбургский государственный университет, г. Оренбург

**Аннотация:** в статье описывается, с помощью каких средств и технологий был разработан сайт *TutorialsWorld*.

**Abstract:** the article describes, by what means and technologies have developed a site *TutorialsWorld*.

**Ключевые слова:** web-приложение, облачная платформа, *OpenShift*, *Apache Maven*.

**Keywords:** web-application, cloud platform, *OpenShift*, *Apache Maven*.

Целью данной работы является разработка сайта, содержащего краткие руководства по технологиям программирования. Перед выполнением данной работы были поставлены следующие задачи:

- изучить современные и актуальные технологии для разработки **web**-приложений на Java;
- изучить возможности современных облачных решений для хостинга web-приложений на Java;
- изучить современные технологии и подходы к реализации «frontend» части приложения с Java «backend»-ом;
- использовать полученные знания для реализации небольшого **web**-приложения – *TutorialsWorld*.

В современном мире технологии программирования быстро приходят и уходят. Реальность такова, что начав изучать одну технологию по толстой книге с полным руководством, к тому моменту, когда вы дочитаете её до середины, эта технология может уже устареть. Поэтому если вы, например, новичок или разработчик, которому нужно выбрать одну из технологий, вам очень важно получить краткий, но содержательный обзор данной технологии, не вдаваясь в технические детали. В англоязычном IT сегменте очень популярны сайты с IT tutorial-ами, чего нельзя сказать о русскоязычном сегменте сети. Поэтому было решено заполнить эту нишу и разработать подобного рода сайт.

Основными сущностями данного приложения являются сложные иерархии категорий и статьи в этих категориях. Так же имеется административная часть, которая позволяет управлять всем контентом сайта.

Данное приложение размещено на облачной платформе *OpenShift*. *OpenShift* это PaaS продукт от фирмы *Red Hat*, который также является IaaS, сравним с такими облачными решениями, как *Google Storage* и *Amazon S3*. Следует отметить, что *OpenShift* предоставляет на бесплатном плане до трех картриджей с базовым набором ресурсов.

*OpenShift* предоставляет готовые сконфигурированные решения на многих языках и под многие технологии и фреймворки. Для нашей технологии готового решения не было, и был собран свой DIY картридж, на котором с помощью *bash* скриптов развернута вся необходимая нам инфраструктура приложения.

Сборкой приложения занимается *Apache Maven* [1]. Он делает *resolving dependency*. Написанное приложение имеет 2 профиля – профиль разработчика поднимается с *InMemoryDb hsqldb* и сконфигурирован на разработку и отладку приложения, и *openshift* профиль, который, по сути, является production-ом. На production-е приложение конфигурируется с *mysql* базой данных. Автоконфигурацией занимается *Spring boot*, а единый интерфейс для работы с разными базами данных предоставляет имплементация *JPA2 - Spring Data JPA*.

В качестве основного Java фреймворка для реализации нашего проекта использован *Spring Framework*. Он предоставляет большое количество возможностей для реализации JavaEE приложения. В приложении ядро *Spring* выступает как DI и IoC контейнер, предоставляет доступ к бинам через *Application Context* (который имплементирует в себе *Bean Factory*).

Классический *Spring* иногда критикуют за большой объём и сложность всевозможных конфигураций. Данную проблему решает один из свежих проектов *Spring Boot* - он предоставляет нам дефолтную конфигурацию основных модулей, а также пакет наше приложение с контейнером сервлетов в *executable jar* файл.

Реализацией *Java Persistence API* занимается *Spring Data JPA*, что дает нам унифицированный интерфейс доступа и хранения данных, имплементирует за нас *Data Access Layer* и предоставляет удобный механизм доступа к *Entities* через репозитории.

*Spring Security* отвечает за аутентификацию и авторизацию в приложении, проверяет возможности пользователя, имплементирует *CSRF* защиту *Rest* запросов.

Реалии современного веба таковы, что классические подходы к построению фронтенда остались в прошлом. Сейчас популярно использовать JS MVC фреймворк на фронтенде, обращающийся с restFull backend-ом.

Разрешением зависимостей на фронтенде занимается Bower - стандартный менеджер пакетов для JS, самый популярный содержит более 11 тысяч пакетов. В качестве фронтенд фреймворка мы использовали AngularJs от компании Google. AngularJS позволяет создать нам Single Page application с поддержкой MVC, DI и предоставляет двусторонний Data Binding.

Одна из лучших возможностей AngularJS - это создание директив, которые являются повторно используемыми web компонентами [2]. Это дает возможность создавать новые HTML теги и атрибуты, которые могут динамично отображать контент в ответ на изменение данных и обновлять сами данные в случае необходимости. На фронтенде использовались такие технологии, как html5 и twitter bootstrap 3 для построения красивых и отзывчивых интерфейсов.

В результате получили простое приложение, но с очень интересной архитектурой, которая может быть использована для построения web-приложений любой сложности. Мы решили все поставленные задачи и получили массу положительных эмоций от работы с современными технологиями.

### *Литература*

1. Apache Maven Project [Электронный ресурс]. URL: <http://www.apache-maven.ru/> (дата обращения 12.01.2016).
2. Контроллеры AngularJS [Электронный ресурс]. URL: <http://metanit.com/web/angular/2.1.php> (дата обращения 10.01.2016).