Spring Boot guide:

<https://o7planning.org/ru/11267/spring-boot-tutorial-for-beginners>

<https://o7planning.org/ru/11669/spring-boot>

**Spring Boo**t  является  проектом на уровне  **IO Execution** (уровень выполнения) **IO Spring Framework**   
**Spring Boot**это следуюший шаг Spring, чтобы сделать его легче в настройке и развитии  приложений.

С **Spring Boot** конфигурации Spring  минимизируются максимально. **Spring Boot** поддерживает встроенный контейнер (embedded containers), который  позволяет веб-приложениям работать независимо и без необходимости применения на **Web Server**.  
Вы можете использовать **spring Boot** чтобы создать Java Web приложение, работающее через команду line " **Java -jar"** или экспортировать War файл для применения  на Web Server как обычно.

**Разбор  pom.xml файла для Spring Boot:**

**Spring-boot-starter-parent** готовый проект в **Spring Boot**. Стандартные зависимые библиотеки объявлены в **spring-boot-starter-paren**t, ваш проект только  унаследовал их. Нужно просто объявить **<parent>** в файле **pom.xml**вашего проекта:

<!-- <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-parent> -->

<parent>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-parent</artifactId>

    <version>2.0.0.RELEASE</version>

    <relativePath/> <!-- lookup parent from repository -->

</parent>

### spring-boot-starter-web

Другие “ **Starter”** только обеспечивают зависимости, которые  вам могут понадобятся при развитии конкретного типа приложения. Например, при развитии веб-приложения, вам нужна зависимость **spring-boot-starter-web:**

<dependencies>

   <dependency>

       <groupId>org.springframework.boot</groupId>

       <artifactId>spring-boot-starter-web</artifactId>

   </dependency>

   <dependency>

       <groupId>org.springframework.boot</groupId>

       <artifactId>spring-boot-starter-test</artifactId>

       <scope>test</scope>

   </dependency>

</dependencies>

**spring-boot-starter-test**

Импортирует библиотеки необходимые для тестирования:

* [JUnit](http://junit.org/): The de-facto standard for unit testing Java applications.
* [Spring Test](https://docs.spring.io/spring/docs/5.0.8.RELEASE/spring-framework-reference/testing.html#integration-testing) & Spring Boot Test: Utilities and integration test support for Spring Boot applications.
* [AssertJ](https://joel-costigliola.github.io/assertj/): A fluent assertion library.
* [Hamcrest](http://hamcrest.org/JavaHamcrest/): A library of matcher objects (also known as constraints or predicates).
* [Mockito](http://mockito.org/): A Java mocking framework.
* [JSONassert](https://github.com/skyscreamer/JSONassert): An assertion library for JSON.
* [JsonPath](https://github.com/jayway/JsonPath): XPath for JSON.

**Spring-boot-maven-plugin** является плагином, который обеспечивает  необходимые библиотеки, помогает вашему проекту работать напрямю без необходимости применять на Web Server. Это помогает создать file jar, который может быть выполнен (Executable):

<plugins>

    <plugin>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-maven-plugin</artifactId>

    </plugin>

</plugins>

Общие свойства **application.properties**

<https://o7planning.org/ru/11685/spring-boot-common-properties>

# Thymeleaf