**Контрольные вопросы**   
Общий вопрос: что такое JDBC, JPA и Hibernate? В чем их разница?

**Hазницу между** JDBC**,** JPA**,** Hibernate в рамках выполнения запросов через Java, без учета [пулов соединений](https://java-master.com/%d0%bf%d1%83%d0%bb-%d1%81%d0%be%d0%b5%d0%b4%d0%b8%d0%bd%d0%b5%d0%bd%d0%b8%d0%b9-connection-pool/).

Основное различие между JPA и JDBC является уровнем абстракции.

JDBC - это стандарт низкого уровня для взаимодействия с базами данных. JPA - это стандарт более высокого уровня с той же целью. JPA позволяет использовать объектную модель в своем приложении, которая может сделать вашу жизнь намного проще. JDBC позволяет вам делать больше вещей с базой данных напрямую, но это требует большего внимания. Некоторые задачи не могут быть эффективно решены с использованием JPA, но могут быть решены более эффективно с JDBC.

JDBC является стандартом для непосредственного подключения к БД и выполнения SQL против него.

Hibernate - это легкий объектный пакет JDBC, который является отдельным объектом постоянным слоем рамы. Hibernate можно использовать в любом JDBC.

Главное отличие заключается в том, что Hibernate сначала извлекает отображение объектов в кэше (т.е. операции Hibernate - это объекты), а JDBC является прямой операционной базой данных.

**1. Как развернуть приложение Spring Boot на сервере (tomcat)?**

Нужно запустить сервер Tomcat и развернуть файл WAR в каталоге webapps. После этого в папке автоматически появится файл (напрмер spring). После развертывания нужно нажать на URL в нашем веб-браузере **http: // localhost: 8080 / spring /** .

**2. В чем разница между Spring Boot и Spring MVC?**

Прежде всего, это две разные области, и задачи, которые они решают, разные.

Spring MVC - это фреймворк MVC, основанный на сервлетах. Он решает проблему разработки WEB с помощью Dispatcher Servlet и ModelAndView. Но его конфигурация громоздка, с большим количеством файлов .xml и .properties. И при использовании maven для управления проектами легко возникают конфликты пакетов jar.

Spring Boot - это пакет для быстрой разработки и интеграции на основе Spring, который включает не только Spring mvc, но также Spring JPA и Spring Security.  
реализует автоматическую настройку и снижает сложность построения проекта. А самодостаточный tomcat.pom.xml может быть настроен как стартовый, что упрощает настройку и решает проблему конфликтов пакетов jar.

**3. Что такое War-файл и отличие от Jar-файла?**

jar и war – расширения платформо-независимых файлов-архивов Java-приложения.

Файлы JAR имеют расширение .jar, а файлы WAR - расширение .war. Но файл WAR - это особый тип файла JAR. Файлы JAR содержат файлы классов, библиотеки, ресурсы и файлы свойств. Файлы WAR содержат сервлеты, страницы JSP, страницы HTML, кодировку JavaScript. Файлы JAR используются для архивации всего приложения Java (настольного компьютера), а файлы WAR используются для развертывания веб-приложений.

**4. Зачем использовать Spring Boot?**

Spring - это очень популярная среда разработки на основе Java, которая используется для создания веб-приложений и приложений для предприятий. В отличие от многих других платформ, которые фокусируются только на одной области, среда Spring предоставляет широкий спектр функций для удовлетворения потребностей современного бизнеса через свои портфельные проекты.

Платформа Spring предоставляет гибкость для настройки bean-компонентов различными способами, такими как XML, аннотации и JavaConfig. По мере увеличения количества функций сложность возрастает, а настройка приложений Spring становится утомительной и подверженной ошибкам.

**5. Как можно запустить Spring Boot?**

1. Запускаем в IDE  
Есть два способа запустить непосредственно в Eclipse и IDEA.

Метод упаковки jar

По умолчанию Spring Boot использует метод встраивания Tomcat, Jetty и других серверов в war пакет jar и должен предоставить основной класс с основным методом. В это время запустите основной метод непосредственно в среде IDE, чтобы запустить приложение Spring Boot.

Если приложение преобразовано в развертывание war пакета, вам необходимо настроить сервер в IDE в это время, затем добавить свой проект Spring Boot на сервер и, наконец, запустить сервер.

2. Собираем и запускаем  
Когда ваш Spring Boot готов к тестированию или подключению к сети, вам нужно запустить его как jar-пакет или war пакет. Метод war пакета здесь не упоминается, и его можно запустить непосредственно на сервере. Вот как запустить jar-пакет напрямую.

$ java -jar javastack-0.0.1-SNAPSHOT.jar

Этот метод также поддерживает включение удаленной отладки, например:

$ java -Xdebug -Xrunjdwp:server=y,transport=dt\_socket,address=8000,suspend=n -jar javastack-0.0.1-SNAPSHOT.jar

3. Запускать с надстройками

Можно использовать плагины Maven и Gradle в среде IDE или в командной строке для запуска приложений Spring Boot.

1）Maven Plugin

$ mvn spring-boot:run

2）Gradle Plugin

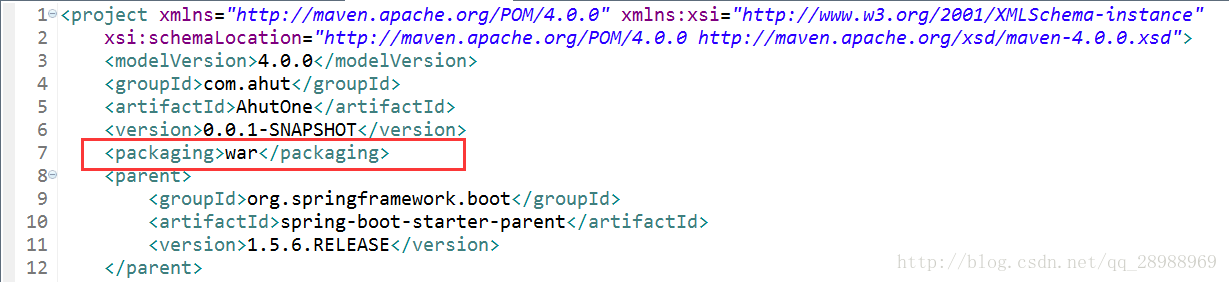
$ gradle bootRun

Рекомендуется использовать подключаемые модули для работы в среде IDE, поскольку вы можете интегрировать больше функций подключаемых модулей, таких как горячее развертывание, различение различных конфигураций среды и т. д.

**6. Требуется ли для запуска Spring Boot отдельный контейнер?**

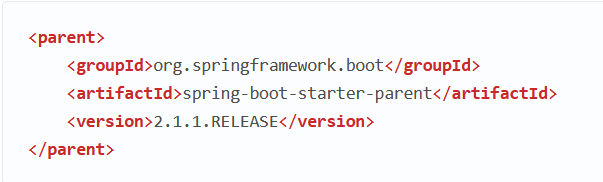
Да, требуется отдельный контейнер. Проект Spring Boot будет иметь встроенный tomcat, проект можно запустить только с помощью простой команды, но в производственной среде мы по-прежнему используем публикацию проекта в третьем контейнере сервлета. Далее будет рассказано, как использовать Spring Boot.

Изменение способа упаковки проекта на war:

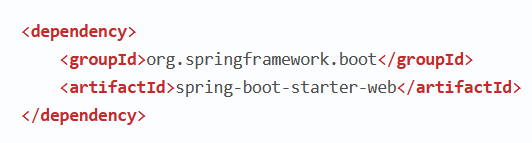


**7. Как настроить приложение Spring Boot с помощью maven?**

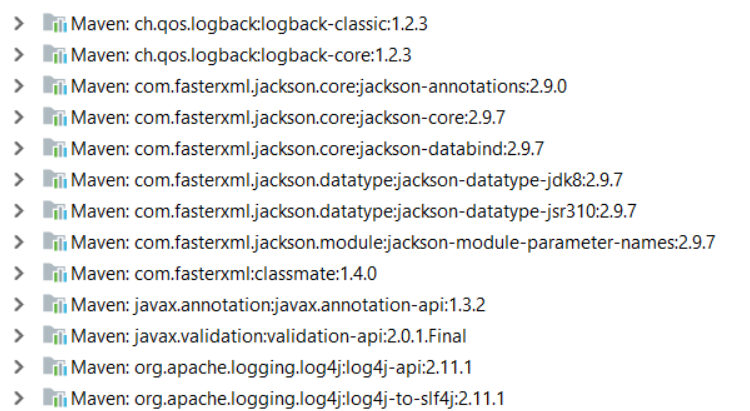
1. Создать Maven-проект в используемой вами IDE, назвав его SpringBootRestService.
2. Вторым шагом необходимо настроить Spring Boot в файле **pom.xml.** Все приложения Spring Boot конфигурируются от **spring-boot-starter-parent**, поэтому перед дальнейшим определением зависимостей, добавьте **starter-parent** следующим образом:



Т.к. мы создаем REST API, то необходимо в качестве зависимости использовать **spring-boot-starter-web**, которая неявно определяет все остальные зависимости, такие как **spring-core**, **spring-web**, **spring-webmvc**, **servlet api**, и библиотеку **jackson-databind**, поэтому просто добавьте в pom.xml следующее:



Теперь следующие jar-библиотеки автоматически импортируются в ваш проект:



Следующий шаг — добавляем Spring Boot плагин:



Последний шаг — сделать так, чтобы Maven генерировал исполняемый jar-файл при сборке:



Используя одну зависимость, мы можем создать полностью функциональное web-приложение.

**8. Какие встроенные серверы поддерживает Spring Boot?**

Каждое Spring Boot web-приложение включает встроенный web-сервер.

Разработчикам теперь не надо беспокоиться о настройке контейнера сервлетов и развертывании приложения на нем. Теперь приложение может запускаться само, как исполняемый jar-файл с использованием встроенного сервера.

Если нужно использовать отдельный HTTP-сервер, для этого достаточно исключить зависимости по умолчанию. Spring Boot предоставляет отдельные starter-пакеты для разных HTTP-серверов.

Создание автономных web-приложений со встроенными серверами не только удобно для разработки, но и является допустимым решением для приложений корпоративного уровня и становится все более полезно в мире микросервисов.

**9. Для чего используются аннотации @RestController и @Controller?**

Классы контроллеров в Spring аннотируются аннотацией @Controller или @RestController. Они помечают классы контроллеров как обработчики запросов, чтобы Spring мог распознать их как службу RESTful во время выполнения.

**10. Что такое Thymeleaf?**

Thymeleaf — современный серверный механизм Java-шаблонов для веб- и автономных сред, способный обрабатывать HTML, XML, JavaScript, CSS и даже простой текст.

Основной целью Thymeleaf является создание элегантного и удобного способа шаблонизации.