1. что такое JDBC, JPA и Hibernate? В чем их разница?

JDBC, JPA и Hibernate - это технологии, используемые для взаимодействия с базами данных в приложениях Java. Вот краткое описание каждой из них и их разницы:

1. JDBC (Java Database Connectivity):

JDBC - это Java API, предназначенное для установления соединения с базами данных и выполнения SQL-запросов из Java-приложений.

Работа с JDBC включает создание соединения с базой данных, создание SQL-запросов, выполнение запросов и обработку результатов. Это требует множества кода и ручной обработки исключений.

2. JPA (Java Persistence API):

JPA - это стандарт Java EE API для управления отображением объектов Java на базу данных. Он предоставляет абстракцию уровня объектов для работы с данными.

JPA позволяет разработчикам работать с объектами Java, а не с SQL-запросами напрямую. Он предоставляет аннотации и API для сопоставления объектов с таблицами базы данных и автоматически создает SQL-запросы. JPA работает поверх JDBC.

3. Hibernate:

Hibernate - это одна из реализаций JPA (или можно сказать, что это JPA-провайдер). Он предоставляет реализацию спецификации JPA и дополнительные функции для управления объектами и базой данных.

Hibernate обеспечивает удобство и уровень абстракции, аналогичный JPA, но также предоставляет свои собственные функции, такие как кэширование, управление транзакциями и маппинг объектов на таблицы.

Разница между ними:

JDBC - это низкоуровневая технология, которая требует ручного написания SQL-запросов и управления соединениями. Он предоставляет полный контроль, но требует больше кода и подвержен ошибкам.

JPA - это стандарт Java EE для работы с базами данных на уровне объектов. Он упрощает доступ к данным, позволяя разработчикам работать с объектами Java, но требует JPA-провайдера, такого как Hibernate, для реализации.

Hibernate - это один из JPA-провайдеров. Он предоставляет дополнительные функции и упрощает работу с базами данных. Hibernate можно использовать как JPA-реализацию или самостоятельно без JPA.

2) Как настроить приложение Spring Boot с помощью maven?

1. Создать новый проект Maven с помощью Maven Archetype для Spring Boot:

mvn archetype:generate -DgroupId=com.example -DartifactId=myproject -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false -DarchetypeCatalog=local -DarchetypeGroupId=org.springframework.boot -DarchetypeVersion=2.5.4

2. Добавить зависимости для Spring Boot в файл pom.xml:

xml

<dependencies>

<dependency>

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

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

<version>2.5.4</version>

</dependency>

</dependencies>

3. Создать основной класс приложения и аннотировать его с помощью @SpringBootApplication:

java

@SpringBootApplication

public class MyApplication {

public static void main(String[] args) {

SpringApplication.run(MyApplication.class, args);

}

}

4. Настроить конфигурацию приложения в файле application.properties:server.port=8080

5. Запустить приложение с помощью команды:

mvn spring-boot:run

После выполнения этих шагов, приложение Spring Boot будет запущено и будет доступно по адресу <http://localhost:8080/>.