**Hibernate Stuff By Ashok:-**

JPA and Hibernate Difference

JPA is a specification for accessing, persisting and managing the data between Java objects and the relational database. As the definition says its API, it is only the specification. There is no implementation for the API. JPA specifies the set of rules and guidelines for developing the interfaces that follows standard. Straight to the point **: JPA is just guidelines to implement the Object Relational Mapping (ORM) and there is no underlying code for the implementation**.

Where as, Hibernate is the actual implementation of JPA guidelines. When hibernate implements the JPA specification, this will be certified by the JPA group upon following all the standards mentioned in the specification. For example, JPA guidelines would provide information of mandatory and optional features to be implemented as part of the JPA implementation.

Hibernate is a JPA provider. When there is new changes to the specification, hibernate would release its updated implementation for the JPA specification. Other popular JPA providers are Eclipse Link (Reference Implementation), OpenJPA, etc. You can find the latest [release of JPA providers](http://www.javabeat.net/jpa-releases/).

In summary, JPA is not an implementation, it will not provide any concrete functionality to your application. Its purpose is to provide a set of rules and guidelines that can be followed by JPA implementation vendors to create an ORM implementation in a standardized manner. This allows the underlying JPA implementation to be swapped and for developers to easily transition (think knowledge wise) from one implementation to another. Hibernate is the most popular **JPA provider**.

**NULL and NOT NULL comparison in the Hibernate API**

**null and not null check**

While checking for the *null* and *not null* values in the *Hibernate API*, we have to be careful and there is chance for misunderstanding. This tips explains how to compare the null values in the *Hibernate* query.

In Java language, if object==null will return the proper value if it is null or not. That means null==null will returns true in the programming. But, the same scenario is differenet in the database world. if you check for null==null in the query, it will retrun false. That is why database provide a “IS NULL” keyword to check the null values. *Hibernate* provides equaivalent methods to check the null values. Look into the following example code for more details:

JavaBeatHibernateExample.java

package javabeat.net.hibernate;

import java.util.List;

import org.hibernate.Criteria;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import org.hibernate.criterion.Expression;

/\*\*

\* source : www.javabeat.net

\*/

public class JavaBeatHibernateExample {

public static void main(String args[]) {

Configuration configuration = new Configuration();

// configuring hibernate

SessionFactory sessionFactory = configuration.configure().buildSessionFactory();

Session session = sessionFactory.openSession();

Criteria criteria = session.createCriteria(Student.class);

criteria.add(Expression.isNotNull(&quot;name&quot;));

List&lt;Student&gt; list = criteria.list();

for (Student student : list){

System.out.println(student.getName());

}

}

}

### In the above code,Expression.isNotNull(“name”) is used for checking the particular colimn for the null values. All the non null rows will be returned for the above code. In the same way you can check for the null values like this:Expression.isNull(“name”)

**24.6 Using YAML instead of Properties**

[YAML](http://yaml.org/) is a superset of JSON, and as such is a very convenient format for specifying hierarchical configuration data. The SpringApplication class will automatically support YAML as an alternative to properties whenever you have the [SnakeYAML](http://www.snakeyaml.org/) library on your classpath.

|  |
| --- |
|  |
| If you use Starters’ SnakeYAML will be automatically provided via spring-boot-starter. |

**24.6.1 Loading YAML**

Spring Framework provides two convenient classes that can be used to load YAML documents. The YamlPropertiesFactoryBean will load YAML as Propertiesand the YamlMapFactoryBean will load YAML as a Map.

For example, the following YAML document:

environments:

dev:

url: http://dev.bar.com

name: Developer Setup

prod:

url: http://foo.bar.com

name: My Cool App

**18. How to implement Spring web using Spring boot?**  
**Web Application Convenience**   
• Boot automatically configures  
– A DispatcherServlet & ContextLoaderListener  
– Spring MVC using same defaults as @EnableWebMvc  
• Plus many useful extra features:  
– Static resources served from classpath  
• /static, /public, /resources or /META-INF/resources  
– Templates served from /templates  
• If Velocity, Freemarker, Thymeleaf, or Groovy on classpath  
– Provides default /error mapping  
• Easily overridden  
– Default MessageSource for I18N

**12. Can you control logging with Spring Boot? How?**  
Yes, we can control logging with spring boot.  
 **Customizing default Configuration for Logging:**  
By adding logback.xml file to the application we can override the default logging configuration providing by the Spring Boot. This file place in the classpath (src/main/resources) of the application for Spring Boot to pick the custom configuration.

**Spring Boot can control the logging level**  
– Just set it in application.properties  
• Works with most logging frameworks  
– Java Util Logging, Logback, Log4J, Log4J2

logging.level.org.springframework=DEBUG

logging.level.com.acme.your.code=INFO

**10. What is a Spring Boot starter POM? Why is it useful?**  
Starters are a set of convenient dependency descriptors that you can include in your application. The starters contain a lot of the dependencies that you need to get a project up and running quickly and with a consistent, supported set of managed transitive dependencies.

The starter POMs are convenient dependency descriptors that can be added to your application’s Maven. In simple words, if you are developing a project that uses Spring Batch for batch processing, you just have to include spring-boot-starter-batch that will import all the required dependencies for the Spring Batch application. This reduces the burden of searching and configuring all the dependencies required for a framework.

**7. What is the difference between an embedded container and a WAR?**  
There is no force to go container less  
– Embedded container is just one feature of Spring Boot  
• Traditional WAR also benefits a lot from Spring Boot  
– Automatic Spring MVC setup, including DispatcherServlet  
– Sensible defaults based on the classpath content  
– Embedded container can be used during development

**5. How does Spring Boot work? How does it know what to configure?**  
• Auto-configuration works by analyzing the classpath  
– If you forget a dependency, Spring Boot can’t configure it  
– A dependency management tool is recommended  
– Spring Boot parent and starters make it much easier  
• Spring Boot works with Maven, Gradle, Ant/Ivy  
– Our content here will show Maven

**2. What are the advantages of using Spring Boot?**

◾ It is very easy to develop Spring Based applications with Java or Groovy.

◾ It reduces lots of development time and increases productivity.

◾ It avoids writing lots of boilerplate Code, Annotations and XML Configuration.

◾ It is very easy to integrate Spring Boot Application with its Spring Ecosystem like Spring JDBC, Spring ORM, Spring Data, Spring Security etc.

◾ It follows “Opinionated Defaults Configuration” Approach to reduce Developer effort

◾ It provides Embedded HTTP servers like Tomcat, Jetty etc. to develop and test our web applications very easily.

◾ It provides CLI (Command Line Interface) tool to develop and test Spring Boot (Java or Groovy) Applications from command prompt very easily and quickly.

◾ It provides lots of plugins to develop and test Spring Boot Applications very easily using Build Tools like Maven and Gradle

◾ It provides lots of plugins to work with embedded and in-memory Databases very easily.

**9. What is DevTools in Spring boot?**

Spring boot comes with DevTools which is introduced to increase the productivity of developer. You don’t need to redeploy your application every time you make the changes.Developer can simply reload the changes without restart of the server. It avoids pain of redeploying application every time when you make any change. This module will be disabled in production environment.

**10. What is actuator in Spring boot?**

Spring boot actuator is one of the most important features of Spring boot. It is used to access current state of running application in production environment. There are various metrics which you can use to check current state of the application.

Spring boot actuator provides restful web services end points which you can simply use and check various metrics.  
For example:  
**/metrics :** This restful end point will show you metrics such as free memory, processors, uptime and many more properties,

Spring boot actuator will help you to monitor your application in production environment.Restful end points can be sensitive, it means it will have restricted access and will be shown only to authenticated users. You can change this property by overriding it in application.properties.