

JPA是什么JavaEE对象持久化标准浅析Java认证考试 PDF转换
可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/594/2021_2022_JPA_E6_98_AF_E4_BB_80_E4_c104_594707.htm JPA通过JDK 5.0注解或XML描述对象 - 关系表的映射关系，并将运行期的实体对象持久化到数据库中。起源 Sun引入新的JPA ORM规范出于两个原因：其一，简化现有Java EE和Java SE应用的对象持久化的开发工作；其二，Sun希望整合对ORM技术，实现天下归一。JPA由EJB 3.0软件专家组开发，作为JSR-220实现的一部分。但它不囿于EJB 3.0，你可以在Web应用、甚至桌面应用中使用。JPA的宗旨是为POJO提供持久化标准规范，由此可见，经过这几年的实践探索，能够脱离容器独立运行，方便开发和测试的理念已经深入人心了。目前Hibernate 3.2、TopLink 10.1.3以及OpenJPA都提供了JPA的实现。JPA的总体思想和现有Hibernate、TopLink，JDO等ORM框架大体一致。总的来说，JPA包括以下3方面的技术：ORM映射元数据 JPA支持XML和JDK 5.0注解两种元数据的形式，元数据描述对象和表之间的映射关系，框架据此将实体对象持久化到数据库表中；JPA的API用来操作实体对象，执行CRUD操作，框架在后台替我们完成所有的事情，开发者从繁琐的JDBC和SQL代码中解脱出来。查询语言 这是持久化操作中很重要的一个方面，通过面向对象而非面向数据库的查询语言查询数据，避免程序的SQL语句紧密耦合。JPA的优势 1 标准化 JPA是JCP组织发布的Java EE标准之一，因此任何声称符合JPA标准的框架都遵循同样的架构，提供相同的访问API，这保证了基于JPA开发的企业应用能够经过少量的修改就能够在不同的JPA框架

下运行。2 对容器级特性的支持 JPA 框架中支持大数据集、事务、并发等容器级事务，这使得 JPA 超越了简单持久化框架的局限，在企业应用发挥更大的作用。3 简单易用，集成方便 JPA 的主要目标之一就是提供更加简单的编程模型：在 JPA 框架下创建实体和创建 Java 类一样简单，没有任何的约束和限制，只需要使用 `javax.persistence.Entity` 进行注释；JPA 的框架和接口也都非常简单，没有太多特别的规则和设计模式的要求，开发者可以很容易的掌握。JPA 基于非侵入式原则设计，因此可以很容易的和其它框架或者容器集成。4 可媲美 JDBC 的查询能力 JPA 的查询语言是面向对象而非面向数据库的，它以面向对象的自然语法构造查询语句，可以看成是 Hibernate HQL 的等价物。JPA 定义了独特的 JPQL (Java Persistence Query Language)，JPQL 是 EJB QL 的一种扩展，它是针对实体的一种查询语言，操作对象是实体，而不是关系数据库的表，而且能够支持批量更新和修改、JOIN、GROUP BY、HAVING 等通常只有 SQL 才能够提供的高级查询特性，甚至还能够支持子查询。5 支持面向对象的高级特性 JPA 中能够支持面向对象的高级特性，如类之间的继承、多态和类之间的复杂关系，这样的支持能够让开发者最大限度的使用面向对象的模型设计企业应用，而不需要自行处理这些特性在关系数据库的持久化。JPA 的供应商 JPA 的目标之一是制定一个可以由很多供应商实现的 API，并且开发人员可以编码来实现该 API，而不是使用私有供应商特有的 API。因此开发人员只需使用供应商特有的 API 来获得 JPA 规范没有解决但应用程序中需要的功能。尽可能地使用 JPA API，但是当需要供应商公开但是规范中没有提供的功能时，则使用供应商特有

的API。 1 Hibernate JPA是需要Provider来实现其功能的，Hibernate就是JPA Provider中很强的一个，目前来说应该无人能出其右。从功能上来说，JPA现在就是Hibernate功能的一个子集。Hibernate从3.2开始，就开始兼容JPA。Hibernate3.2获得了Sun TCK的JPA(Java Persistence API)兼容认证。只要熟悉Hibernate或者其他ORM框架，在使用JPA时会发现其实非常容易上手。例如实体对象的状态，在Hibernate有自由、持久、游离三种，JPA里有new，managed，detached，removed，明眼人一看就知道，这些状态都是一一对应的。再如flush方法，都是对应的，而其他的再如说Query query = manager.createQuery(sql)，它在Hibernate里写法上是session，而在JPA中变成了manager，所以从Hibernate到JPA的代价应该是非常小的。同样，JDO，也开始兼容JPA。在ORM的领域中，看来JPA已经是王道，规范就是规范。在各大厂商的支持下，JPA的使用开始变得广泛。

2 Spring Spring Hibernate 常常被称为Java Web应用人气最旺的框架组合。而在JCP通过的Web Beans JSR，却欲将JSF EJB JPA、来自JBoss Seam（Spring除外）的一些组件和EJB 3（目前能够提供有基本拦截和依赖注入功能的简化Session Bean框架）的一个Web组合进行标准化。如今的Spring 2.0为JPA提供了完整的EJB容器契约，允许JPA在任何环境内可以在Spring管理的服务层使用（包括Spring的所有AOP和DI增强）。同时，关于下一个Web应用组合会是EJB、Spring Hibernate还是Spring JPA的论战，早已充斥于耳。在Spring 2.0.1中，正式提供对JPA的支持，这也促成了JPA的发展，要知道JPA的好处在于可以分离于容器运行，变得更加的简洁。

3 OpenJPA OpenJPA是Apache组织

提供的开源项目，它实现了 EJB 3.0 中的 JPA 标准，为开发者提供功能强大、使用简单的持久化数据管理框架。OpenJPA 封装了和关系型数据库交互的操作，让开发者把注意力集中在编写业务逻辑上。OpenJPA 可以作为独立的持久层框架发挥作用，也可以轻松的与其它 Java EE 应用框架或者符合 EJB 3.0 标准的容器集成。4 其它目前支持的实现包括 Toplink、Hibernate Entitymanager 等。TopLink 以前需要收费，如今开源了。OpenJPA 虽然免费，但功能、性能、普及性等方面更加需要加大力度。对于 EJB 来说，实体 Bean 一直是被批评的对象，由于其太复杂和庞大。JPA 的出现，很大程度的分离了复杂性。这让 EJB 的推广也变得容易。总而言之，JPA 规范主要关注的仅是 API 的行为方面，而由各种实现完成大多数性能有关的调优。尽管如此，所有可靠的实现都应该拥有某种数据缓存，以作为选择。但愿不久的将来，JPA 能成为真正的标准。

小结 EJB 3.0 和 JPA 毫无疑问将是 Java EE 5 的主要卖点。在某些领域中，它们给 Java 社区带来了竞争优势，并使 Java 在其他领域与竞争对手不分伯仲（因为，不可否认，目前某些领域尚不存在基于标准的方法）。过去数年来，Spring Framework 一直是 EJB 在企业领域的主要竞争对手。EJB 3.0 规范解决了很多促进 Spring 兴起的问题。随着它的出现，EJB 3.0 毫无疑问比 Spring 提供了更好的开发体验最引人注目的优势是它不需要配置文件。JPA 提供一种标准的 OR 映射解决方案，该解决方案完全集成到 EJB 3.0 兼容的容器中。JPA 的前辈将会继续稳定发展，但是业务应用程序中的 raw 使用将可能会减少。实现 JPA 兼容的实体管理器似乎很可能是此类技术的发展方向。Java EE 系列规范的较大问题与 JPA 没有任何关系。Java EE

系列规范的问题涉及到 Web和EJB容器之间的集成。Spring在此领域仍然具有主要竞争优势。JBoss的Seam项目尝试使用自定义的方法来解决这一问题。Caucho Resin应用服务器试图扩展容器边界并支持在Web容器中使用@EJB注释。我们希望Java EE 5.1将解决层集成的问题，为我们提供一个全面而标准的依赖性注入方法。在不久的将来，Sun可能会将JPA作为一个单独的JSR对待，同时JPA还可能作为Java SE的一部分。不过这些都不太重要，重要的是，我们现在已经可以在脱离容器的情况下、在Java SE应用中使用JPA了。JPA已经作为一项对象持久化的标准，不但可以获得Java EE应用服务器的支持，还可以直接在Java SE中使用。开发者将无需在现有多种ORM框架中艰难地选择，按照Sun的预想，现有ORM框架头顶的光环将渐渐暗淡，不再具有以往的吸引力。更多优质资料尽在百考试题论坛 百考试题在线题库 java认证更多详细资料 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com