

UNIVERSIDADE FEDERAL DA BAHIA
INSTITUTO DE MATEMÁTICA
CIÊNCIA DA COMPUTAÇÃO
LINGUAGENS PARA APLICAÇÃO COMERCIAL

Java Persistence API 1.0

Salvador-Ba

Março/2009

RAMON PEREIRA LOPES

Java Persistence API 1.0

Atividade apresentada como requisito para avaliação no curso de Ciência da Computação da Universidade Federal da Bahia na disciplina Linguagens Para Aplicação Comercial orientada pelo professor Adonai Estrela Medrado.

Salvador-Ba

Março/2009

SUMÁRIO

INTRODUÇÃO	3
JAVA PERSISTENCE API	3
2.1 Motivação para Desenvolvimento	3
2.2 Principais características, características e restrições	3
2.3 Licença	4
CONCLUSÃO	5

INTRODUÇÃO

Em meio ao contexto da especificação do EJB 3.0, o Java Persistence API - JPA surgiu como uma padronização para o modelo de mapeamento objeto-relacional na plataforma Java, permitindo sua utilização fora de um servidor de aplicações, como por exemplo em simples aplicações Desktop.

Atuando como uma abstração sobre JDBC, o JPA é a camada de persistência responsável por mapear objetos para o banco de dados, através de uma simples linguagem de anotações permite ao desenvolvedor facilmente executar consultas, inserções, atualizações e remoção dos dados, além de recursos mais avançados.

JAVA PERSISTENCE API

2.1 Motivação para Desenvolvimento

O JPA 1.0 nasceu em meio à especificação do EJB 3.0, JSR 220, tendo como maior objetivo simplificar o modelo de persistência referente ao EJB 2.1, trazendo produtividade, facilidade de uso e aprendizado, além do desacoplamento referente ao uso de servidor de aplicações e certas interfaces necessárias na versão 2.1.

2.2 Principais características, características e restrições

- Uso de anotações para mapeamento de classes e definição de relacionamentos.
- Permite uso de SQL nativo ou EJB QL.
- Definição de *named queries*.
- Suporte a herança e consultas polimórficas.
- Permite uso de *callbacks* e *listeners*.
- Definição de operações em cascata.
- Configuração de conexões em arquivo XML.
- Criação das tabelas no banco de dados a partir das entidades mapeadas.
- Por ser uma abstração sobre o JDBC, necessita-se do *driver* do Sistema Gerenciado de Banco de Dados.
- Estratégias de concorrência de dados, exemplo: *Lock* otimista.
- Pode ser usado fora de um servidor de aplicações.

- Necessidade do uso de um provedor JPA, tais como Hibernate e TopLink Essentials.
- Não possui recurso semelhante à *Criteria* API pertencente ao Hibernate.

2.3 Licença

Como o Java Persistence API não passa de uma especificação para persistência na plataforma Java, vamos utilizar a implementação de referência - RI TopLink Essentials para este tópico. De acordo com o *site* do fabricante, Oracle, o TopLink Essentials está licenciado sobre *Common Development and Distributor License* - CDDL versão 1.0.

2.3.1 Restrições e exigências

- Fonte de modificação deve estar disponível sob licença *open source*.
- As modificações são regidas pela licença.
- Identificar as suas modificações no código como contribuinte.
- Licença pode ser aplicada ao programa como um todo ou apenas em arquivos e métodos individualmente.

2.3.2 Obrigações do desenvolvedor

- Possibilidade de distribuir o executável sob licença diferente.
- Pode-se utilizar arquivos, já que a licença é baseada em arquivo, de outras licenças.

2.4 Plataformas e sistemas operacionais suportados

- Plataformas: X86, x64, Intel Itanium, SPARC, Java Platform Standard Edition.
- Sistemas Operacionais: Linux, Solaris, Windows.

CONCLUSÃO

O Java Persistence API, que simplesmente nasceu em meio à especificação do EJB 3.0, vem ganhando força a ponto de se tornar independente e ter a sua própria JSR. A JSR – 317 é referente à segunda versão do padrão de persistência, prometendo adicionar novos recursos e melhorias.

Pode-se perceber que tal especificação apresenta um padrão para a camada de persistência na plataforma Java. Permitindo o uso fora de um servidor de aplicações, tem-se como consequência a substituição gradativa do uso de JDBC puramente, já que os provedores JPA atuam como uma abstração sobre esse. Em relação ao EJB 2.1, traz uma série de melhorias que se refletem em menos uso de interfaces, *look-up*, mapeamento em XML e novas funcionalidades.

Por executar sobre a plataforma Java Standard Edition, o provedor JPA se privilegia por transição, da característica de multiplataforma do Java, isto é, qualquer sistema operacional que suporte o JDK poderá ser utilizado para desenvolvimento.

Percebe-se que o JPA tem crescido, conquistado espaço e facilitado bastante o processo de desenvolvimento de software. Atualmente encontra-se na sua segunda versão da especificação, tendo grandes nomes da indústria como Oracle, JBoss, Sybase, Apache como membros atuantes na especificação.

REFERÊNCIAS

JSR 220: Enterprise JavaBeans. Disponível em: <<http://jcp.org/en/jsr/detail?id=220>>. Acessado em: 07 de Março de 2009.

JSR 317: Java Persistence 2.0. Disponível em: <<http://jcp.org/en/jsr/detail?id=317>>. Acessado em: 07 de Março de 2009.

BISWAS, Rahul; ORT, Ed. The Java Persistence API - A Simpler Programming Model for Entity Persistence. 2006. Disponível em: <<http://java.sun.com/developer/technicalArticles/J2EE/jpa>>. Acessado em: 07 de Março de 2009.

TopLink Essentials. Disponível em: <http://www.oracle.com/technology/products/ias/toplink/jpa/download.html>>. Acessado em: 07 de Março de 2009.

CDDL Versão 1.0. Disponível em: <<https://glassfish.dev.java.net/public/CDDLv1.0.html>>. Acessado em: 07 de Março de 2009.

BURKE, Bill; MONSON-HAEFEL, Richard. Enterprise JavaBeans 3.0. 5^a ed. p. 41-69. United States: Prentice-Hall, 2006.