JPA 2.0 Feature Requests

By Roger Keays, 7 August 2007

I've just come across a blog by David Van Couvering enumerating the plans for JPA 2.0 and was pretty disappointed that it didn't really include any of the features I've been hanging out for in JPA. Fortunately there was a feedback address: persistence-feature-requests@sun.com, so I hastily collated my ideas and sent them there. Will it make any difference? Good question.

Here is my list of feature requests which I sent to that address.

1. em.getObjectId(o): Having a method to fetch any entity's primary key allows you to write some useful tools such as a generic Entity converter for JSF. *This feature is available in JDO 2.0*.
2. Mutable primary keys: JPA 1.0 specifically restricts the primary key of an entity being modified (section 2.1.4). The reason for this is unclear to be, and only means more unnecessary surrogate primary keys. *This feature is available in JDO 2.0*.
3. Generated values: JPA 1.0 only supports generated values for primary key fields. Allowing generated values for any field would be useful. *This feature is available in JDO 2.0*.
4. More Collection mappings: Particularly, I'm looking for support for mapping arrays, Collections and Maps of basic types (e.g. int[], List<String>, Map<String, Date>). *This feature is available in JDO 2.0*.
5. Dependent Elements: Support for dependent elements in collections like OpenJPA's @ElementDependent. i.e. removing an element from the collection deletes it from the database. *This feature is available in JDO 2.0*.
6. Indexed fields: Indexes are important in relational databases, so I'm surprised JPA doesn't offer an @Index annotation / metadata to indicate that the database column should be indexed. *This feature is available in JDO 2.0*.
7. JPQL LIMIT keyword: If JPQL supported a LIMIT x,y keyword as a part of the query string, this would reduce the necessity to build queries with Java code. *This feature is available in JDO 2.0*.
8. Automatic relationship management: Some JPA vendors support the automatic of relationships when entities are persisted (e.g. by creating a bidirectional link). I'd use these features if they were a part of the standard.
9. Automatic attach/merge: This is really my number one feature request, but I've included it last because I haven't given much thought to the viability of the idea. The problem is common - entities are detached between requests and need to be manually merged to access lazy loaded

fields. For me, this is consistently a cause of defects and annoyance. I'd be very happy if entities could be automatically merged into the persistence context when it is available, analogous to the way they are automatically detached when the persistence context ends.

David's blog does mention some changes to the management of the persistence context which might help alleviate the problems mentioned in item 9. Let's hope so.

It's great that JPA put an end to the EJB 2.0 persistence debacle, but as you can see, I still feel it is behind the 8-ball when compared to JDO 2.0. Please let me know if I've made any mistakes about what is and isn't in JDO 2.0 / JPA 1.0.

About Roger Keays

Roger Keays is an artist, an engineer, and a student of life. He has no fixed addressand has leftfootprints on 40-something different countries around the world. Roger is addicted to surfing. His other interests are music, psychology, languages, the proper use of semicolons, and finding good food.