Hibernate Tip: Select multiple scalar values

Question:

How can I select a list of scalar values in a Criteria query?

Solution:

The *CriteriaQuery* interface provides the *multiselect()* method which allows you to select multiple scalar values. The following code snippet shows an example for such a query.

```
// Prepare query

CriteriaBuilder cb = em.getCriteriaBuilder();

CriteriaQuery<Tuple> q = cb.createTupleQuery();

Root<Author> author = q.from(Author.class);


// Select multiple scalar values

q.multiselect(author.get(Author_.firstName).alias("firstName"),
        author.get(Author_.lastName).alias("lastName"));


List<Tuple> authorNames = em.createQuery(q)
        .getResultList();


for (Tuple authorName : authorNames) {
        log.info(authorName.get("firstName") + " "
                + authorName.get("lastName"));
}
```

The *multiselect()* method expects a *List* or an array of *Selection* interfaces which define the entity attributes which shall be fetched

from the database. In this example, I use the JPA metamodel to reference the attributes in a type-safe way.

When you execute such a *CriteriaQuery*, it returns a *List* of *Tuple* interface implementations. The *Tuple* interface provides convenient access to the selected values based on its position or its alias. In the code snippet, I defined an alias for each attribute in the query and use it to get them from the *Tuple* result.

Learn more

You can not only use the Criteria API to create database queries, you can also write update and delete statements, as I show in this post: Criteria Update/Delete – The easy way to implement bulk operations with JPA2.1.

And if you want to use the Criteria API in your project, you should also have a look at the JPA metamodel. It provides a great way to create queries in a type-safe way. I explain it in detail in this post: Create type-safe queries with the JPA static metamodel.