Java Persistence Query Language

- JPQL
- CriteriaBuilder
- CriteriaQuery
- Root
- Query
- Examples

JPQL

The Java Persistence Query Language is a platform-independent object-oriented query language defined as part of the Java Persistence API specification.

JPQL is used to make queries against entities stored in a relational database.

It is heavily inspired by SQL, and its queries resemble SQL queries in syntax, but operate against JPA entity objects rather than directly with database tables.

In addition to retrieving objects (SELECT queries), JPQL supports set based UPDATE and DELETE queries.

javax.persistence.criteria.CriteriaBuilder

Used to construct criteria queries, compound selections, expressions, predicates, orderings.

Usage

```
CriteriaBuilder builder = session.getCriteriaBuilder();
```

javax.persistence.criteria.CriteriaQuery

CriteriaQuery is an object representation of query string.

So, instead of strings with queries you should use *criteria* to specify details of the query.

Usage

```
CriteriaQuery<UserDataSet> criteria = builder.createQuery(UserDataSet.class);
Root<UserDataSet> from = criteria.from(UserDataSet.class);
criteria.where(builder.equal(from.get("name"), name));
```

javax.persistence.criteria.Root<X>

A root type in the from clause. Query roots always reference entities.

<X> is the entity type referenced by the root.

Usage

```
Root<UserDataSet> from = criteria.from(UserDataSet.class);
criteria.where(builder.equal(from.get("name"), name));
```

org.hibernate.query.Query<R>

Represents an JPQL query or a compiled Criteria query.

Also acts as the Hibernate extension to the JPA Query contract.

Usage

```
Query<UserDataSet> query = session.createQuery(criteria);
UserDataSet dataSet = query.uniqueResult();
```