

Criteria API

The Criteria API allows you to build up a criteria query object programmatically; the `org.hibernate.Criteria` interface defines the available methods for one of these objects.

The Hibernate Session interface contains several Overloaded `createCriteria()` methods.

The commonly used methods of Criteria interface are as follows:

- `public Criteria add(Criterion c)` is used to add restrictions.
- `public Criteria addOrder(Order o)` specifies ordering.
- `public Criteria setFirstResult(int firstResult)` specifies the first number of record to be retrieved.
- `public Criteria setMaxResult(int totalResult)` specifies the total number of records to be retrieved.
- `public List list()` returns list containing object.
- `public Criteria setProjection(Projection projection)` specifies the projection.

Examples:

```
Criteria cr = session.createCriteria(Employee.class);  
cr.add(Restrictions.eq("salary", 2000));  
List results = cr.list();
```

Or

```
Criteria crit = session.createCriteria(Product.class);  
List<Product> results = crit.list();
```

Restrictions:

The Criteria API makes it easy to use restrictions in your queries to selectively retrieve objects;

You may add these restrictions to a Criteria object with the add() method.

The add() method takes an org.hibernate.criterion.Criterion object that represents an individual restriction.

Example:

```
Criteria crit = session.createCriteria(Product.class);
crit.add(Restrictions.eq("description", "Mouse"));
List<Product> results = crit.list();
```

Pagination:

One common application pattern that criteria can address is pagination through the result set of a database query.

There are two methods on the Criteria interface for paging, just as there are for Query: setFirstResult() and setMaxResults().

```
Criteria crit = session.createCriteria(Product.class);
crit.setFirstResult(1);
crit.setMaxResults(20);
List<Product> results = crit.list();
```

Distinct Results

If you would like to work with distinct results from a criteria query, Hibernate provides a result transformer for distinct entities, org.hibernate.transform.DistinctRootEntityResultTransformer, which ensures that no duplicates will be in your query's result set.

Example:

```
Criteria crit = session.createCriteria(Product.class);
```

```
Criterion price = Restrictions.gt("price",new Double(25.0));  
crit.setResultTransformer( DistinctRootEntityResultTransformer.INSTANCE )  
List<Product> results = crit.list();
```