

# Hibernate Tip: How to use pagination with JPQL

## Question:

JPQL doesn't support the *LIMIT* keyword. How can I use pagination with Hibernate?

## Solution:

With JPA and Hibernate, you have to set the pagination information on the *Query* interface and not in the query *String* as you know it from SQL. You can do that by calling the *setFirstResult(int startPosition)* and *setMaxResults(int maxResults)* methods.

The following code snippet shows a simple example that returns the first 5 Authors from the database. The result set index is 0 based and you, therefore, need to provide 0 as a *startPosition* to begin with the first element.

```
List<Author> authors = em.createQuery(
    "SELECT a FROM Author a", Author.class)
    .setMaxResults(5)
    .setFirstResult(0)
    .getResultList();
```

To select the next 5 *Authors* from the database, you only need to change the *startPosition* to 5.

```
List<Author> authors = em.createQuery(
    "SELECT a FROM Author a", Author.class)
    .setMaxResults(5)
    .setFirstResult(0)
    .getResultList();
```

# Hibernate Tip: How to use pagination with JPQL

## Further reading:

If you like to learn more about JPQL, have a look at my recent post in which I explain several JPQL capabilities: [Is your query too complex for JPA and Hibernate?](#)

And don't forget, if you can't implement your query with JPQL, you can still use a native query to execute plain SQL statements. I get into more details about that in my free [Native Queries with Hibernate](#) [ebook](#).