**JPA Repository Interface**

Extending these interfaces allows Spring to find your repository interfaces and create proxy objects for them, that you can inject later on into your beans.

It provides you with methods that allow you to perform some common operations without the need to declare these methods yourself.

**PagingAndSortingRepository** extends **CrudRepository**

**JpaRepository** extends **PagingAndSortingRepository**

The **CrudRepository** interface provides methods for CRUD operations, so it allows you to create, read, update and delete records without having to define your own methods. The **PagingAndSortingRepository** provides additional methods to retrieve entities using pagination and sorting. Finally the JpaRepository add some more functionality that is specific to JPA.

In short JpaRepository

* has additional JPA specific methods that support for example [Query By Example](http://docs.spring.io/spring-data/jpa/docs/current/reference/html/#query-by-example) , deleting in batches, manual flushing changes to database
* querying methods return List's instead of Iterable's

**Query methods** are methods that find information from the database and are declared on the repository interface. For example, if we want to create a database query that finds the Todo object that has a specific id, we can create the query method by adding the findById() method to the TodoRepositoryinterface. After we have done this, our repository interface looks as follows:

|  |  |
| --- | --- |
|  | interface TodoRepository extends Repository<Todo, Long> {      //This is a query method.      Todo findById(Long id);  } |

| **Keyword** | **Sample** | **JPQL snippet** |
| --- | --- | --- |
| And | findByLastnameAndFirstname | … where x.lastname = ?1 and x.firstname = ?2 |
| Or | findByLastnameOrFirstname | … where x.lastname = ?1 or x.firstname = ?2 |
| Is,Equals | findByFirstname,findByFirstnameIs,findByFirstnameEquals | … where x.firstname = ?1 |
| Between | findByStartDateBetween | … where x.startDate between ?1 and ?2 |
| LessThan | findByAgeLessThan | … where x.age < ?1 |
| LessThanEqual | findByAgeLessThanEqual | … where x.age <= ?1 |
| GreaterThan | findByAgeGreaterThan | … where x.age > ?1 |
| GreaterThanEqual | findByAgeGreaterThanEqual | … where x.age >= ?1 |
| After | findByStartDateAfter | … where x.startDate > ?1 |
| Before | findByStartDateBefore | … where x.startDate < ?1 |
| IsNull | findByAgeIsNull | … where x.age is null |
| IsNotNull,NotNull | findByAge(Is)NotNull | … where x.age not null |
| Like | findByFirstnameLike | … where x.firstname like ?1 |
| NotLike | findByFirstnameNotLike | … where x.firstname not like ?1 |
| StartingWith | findByFirstnameStartingWith | … where x.firstname like ?1(parameter bound with appended %) |
| EndingWith | findByFirstnameEndingWith | … where x.firstname like ?1(parameter bound with prepended %) |
| Containing | findByFirstnameContaining | … where x.firstname like ?1(parameter bound wrapped in %) |
| OrderBy | findByAgeOrderByLastnameDesc | … where x.age = ?1 order by x.lastname desc |
| Not | findByLastnameNot | … where x.lastname <> ?1 |
| In | findByAgeIn(Collection<Age> ages) | … where x.age in ?1 |
| NotIn | findByAgeNotIn(Collection<Age> ages) | … where x.age not in ?1 |
| True | findByActiveTrue() | … where x.active = true |
| False | findByActiveFalse() | … where x.active = false |
| IgnoreCase | findByFirstnameIgnoreCase | … where UPPER(x.firstame) = UPPER(?1) |