



Spring Data JPA

JX Java Back End

John Mitchell • Initial Version

JPA CrudRepository Standard Methods

<https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html>

Modifier and Type	Method and Description
long	<code>count()</code> Returns the number of entities available.
void	<code>delete(T entity)</code> Deletes a given entity.
void	<code>deleteAll()</code> Deletes all entities managed by the repository.
void	<code>deleteAll(Iterable<? extends T> entities)</code> Deletes the given entities.
void	<code>deleteById(ID id)</code> Deletes the entity with the given id.
boolean	<code>existsById(ID id)</code> Returns whether an entity with the given id exists.
<code>Iterable<T></code>	<code>findAll()</code> Returns all instances of the type.
<code>Iterable<T></code>	<code>findAllById(Iterable<ID> ids)</code> Returns all instances of the type with the given IDs.
<code>Optional<T></code>	<code>findById(ID id)</code> Retrieves an entity by its id.
<code><S extends T></code> <code>S</code>	<code>save(S entity)</code> Saves a given entity.
<code><S extends T></code> <code>Iterable<S></code>	<code>saveAll(Iterable<S> entities)</code> Saves all given entities.



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, findByFirstnames, 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> age)	... where x.age not in ?1
True	findByActiveTrue()	... where x.active = true
False	findByActiveFalse()	... where x.active = false
IgnoreCase	findByFirstnameIgnoreCase	... where UPPER(x.firstname) = UPPER(?1)

<https://docs.spring.io/spring-data/jpa/docs/1.5.1.RELEASE/reference/html/jpa.repositories.html#jpa.query-methods.query-creation>



Restaurant Layout

```
private long restaurantid;
```

```
private String name;
```

```
private String address;
```

```
private String city;
```

```
private String state;
```

```
private String telephone;
```

```
@OneToMany
```

```
private List<RestaurantPayments> restaurantPayments
```

```
@OneToMany
```

```
private List<Menu> menus
```



Repository

```
Restaurant findByName(String name);
```

```
List<Restaurant> findByNameContainingIgnoreCase (String name);
```

```
List<Restaurant> findByNameContainingIgnoreCaseAndCityContainingIgnoreCase  
    (String likename, String likecity);
```

Service

```
public Restaurant findRestaurantByName(String name) throws EntityNotFoundException
{
    Restaurant restaurant = restrepos.findByName(name);
    return restaurant;
}
```

```
public List<Restaurant> findRestaurantByNameLike(String name)
{
    List<Restaurant> list = restrepos.findByNameContainingIgnoreCase(name);
    return list;
}
```

```
public List<Restaurant> findNameCity(String name, String city)
{
    List<Restaurant> list =
        restrepos.findByNameContainingIgnoreCaseAndCityContainingIgnoreCase(name, city);
    return list;
}
```



API End Points

localhost:2019/restaurant/name/Apple

localhost:2019/restaurants/namelike/e

localhost:2019/restaurants/name/e/city/tow

Controller

```
@GetMapping(value = "/restaurant/name/{name}",
            produces = {"application/json"})
public ResponseEntity<?> getRestaurantByName(
    @PathVariable
    String name)
{
    Restaurant r = restaurantService.findRestaurantByName(name);
    return new ResponseEntity<>(r, HttpStatus.OK);
}

@GetMapping(value = "/restaurants/namelike/{name}",
            produces = {"application/json"})
public ResponseEntity<?> listRestaurantNameLike(@PathVariable String name)
{
    List<Restaurant> myRestaurants = restaurantService.findRestaurantByNameLike(name);
    return new ResponseEntity<>(myRestaurants, HttpStatus.OK);
}

@GetMapping(value = "/restaurants/name/{name}/city/{city}",
            produces = {"application/json"})
public ResponseEntity<?> listRestaurantNameCity(@PathVariable String name,
                                                @PathVariable String city)
{
    List<Restaurant> myRestaurants = restaurantService.findNameCity(name, city);
    return new ResponseEntity<>(myRestaurants, HttpStatus.OK);
}
```

