http://www.lycato.com

Spring Data JPA

Sort & Paginate

Nguyễn Nghiệm

Sort

Agenda

- Sort
- Pageable, PageRequest
- Page

https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html

AGENDA

Sort

|< << 5/8 >> >|

Paginate

SORT & PAGINATE JPAREPOSITORY API

Sort, Pageable, Page ?

SORTING & PAGINNATING

No sort & paginate

```
@Autowired
ProductDAO dao;
@RequestMapping("/find“)
public String find() {
        List<Product> list = dao.findAll();
        …
}
```

Paginating

```
@Autowired
ProductDAO dao;
@RequestMapping("/page")
public String page() {
        Pageable pageable = PageRequest.of(0, 10);
        Page<Product> page = dao.findAll(pageable);
        …
}
```

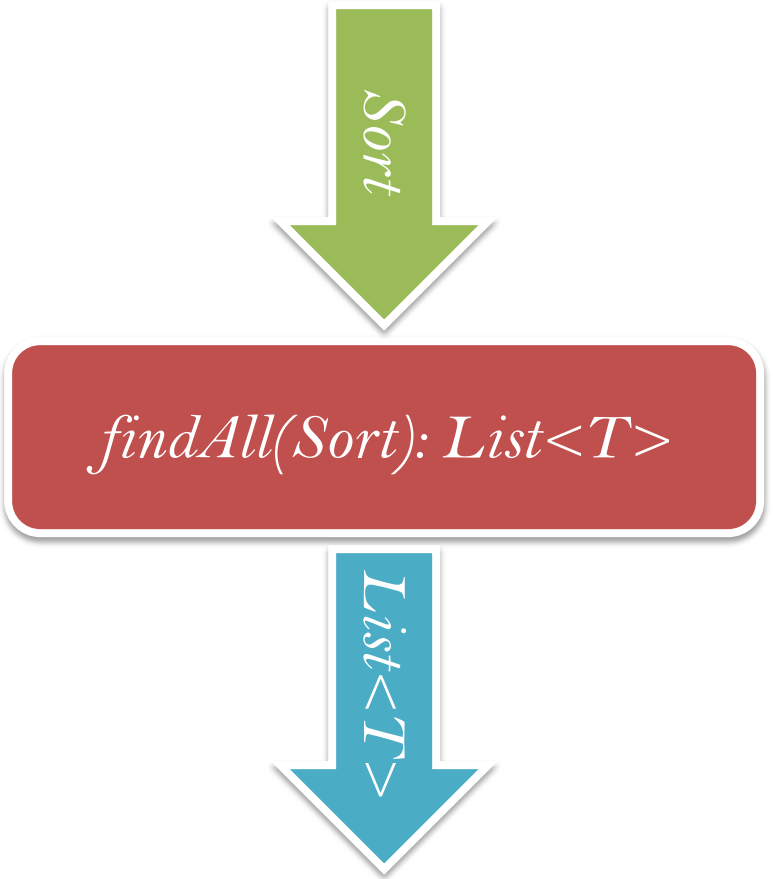
Sort ASC

```
@Autowired
ProductDAO dao;
@RequestMapping("/sort“)
public String sort() {
        Sort sort = Sort.by("unitPrice");
        List<Product> list = dao.findAll(sort);
        …
}
```

Sort DESC and paginate

```
@Autowired
ProductDAO dao;
@RequestMapping("/page-and-sort")
public String page() {
        Sort sort = Sort.by(Direction.DESC, "unitPrice");
        Pageable pageable = PageRequest.of(0, 10, sort );
        Page<Product> page = dao.findAll(pageable);
        …
}
```

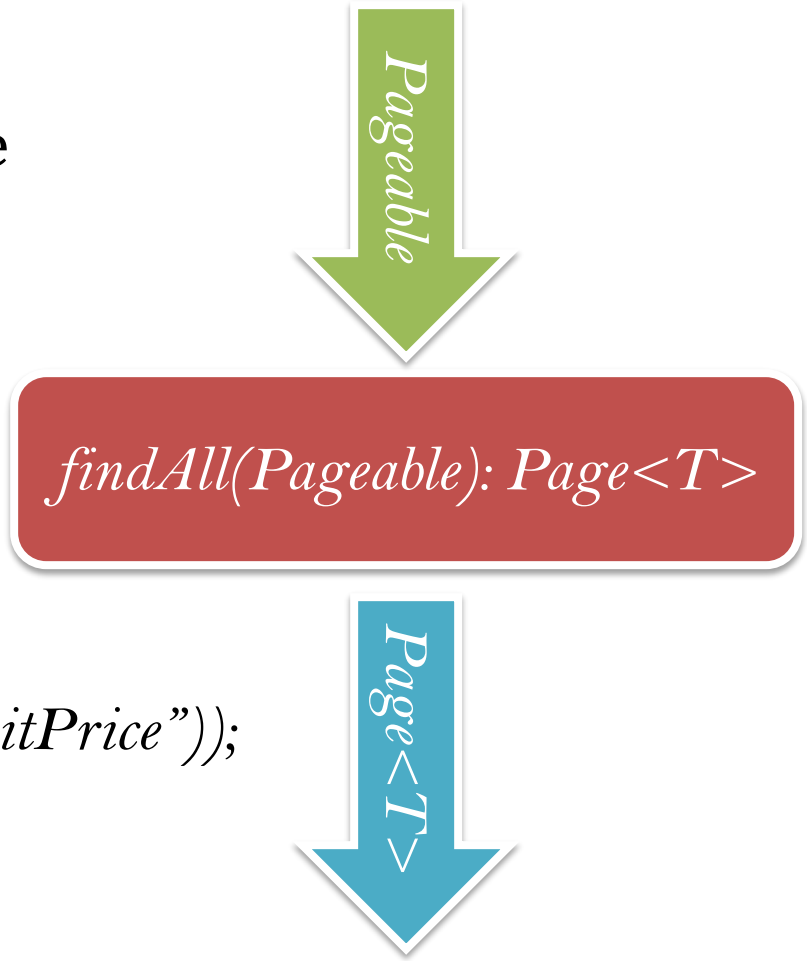
- Sort
  - *Sort.by(String...properties)*: Sort
    - Sắp xếp tăng dần theo các thuộc tính
  - *Sort.by(Direction, String...properties)*: Sort
    - Sắp xếp tùy hướng (*Direction.ASC*: tăng, *Direction.DESC*: giảm) theo các thuộc tính
- Ví dụ:
  - *Sort sort = Sort.by(Direction.DESC, "unitPrice")*
  - *Sort sort = Sort.by("productDate", "unitPrice")*
  - *List<Product> list = dao.findAll(sort)*



Paginate

PAGINATING

- Pageable
  - *PageRequest.of(int page, int size)*: Pageable
  - *PageRequest.of(int page, int size, Sort sort)* : Pageable
- Page<T>
  - *getContent()*: Iterable<T> hoặc *toList()*: List<T>
  - *getPageable()*: Pageable
- Ví dụ
  - *Pageable pageable = PageRequest.of(1, 20);*
  - *Pageable pageable = PageRequest.of(1, 20, Sort.by("unitPrice"));*
  - *Page<Product> page = dao.findAll(pageable);*
  - *Iterable<Product> list = page.getContent();*
  - *List<Product> list = page.toList();*



Page<T> API

	Methods	Return Type	Description
ĐỌC THÔNG TIN	*getNumber()*	*int*	*Trang số*
	getSize()	*int*	*Kích thước trang*
	getTotalPages()	*int*	*Số lượng trang*
	getTotalElements()	*int*	*Tổng số phần tử*
	getNumberOfElements()	*int*	*Số lượng phần tử hiện tại*
	getContent() hoặc toList()	*List<T>*	*Danh sách phần tử*
KIỂM TRA	*hasNext()*	*boolean*	*Có trang sau?*
	hasPrevious()	*boolean*	*Có trang trước?*
	isFirst()	*boolean*	*Trang đầu?*
	isLast()	*boolean*	*Trang cuối?*
ĐIỀU HƯỚNG	*nextPageable()*	*Pageable*	*Trang sau*
	previousPageable()	*Pageable*	*Trang trước*
	nextOrLastPageable()	*Pageable*	*Trang sau hoặc trang cuối*
	previousOrFirstPageable()	*Pageable*	*Trang trước hoặc trang đầu*

- Duyệt dữ liệu

```
<any th:each="item: ${page.content}">
    //....
</any>
```

- Hiển thị kết quả phân trang
  - Trang số: *${page.number}*
  - Kích thước trang: *${page.size}*
  - Tổng số trang: *${page.totalPages}*
  - Số phần tử hiện tại: *${page.numberOfElements}*
  - Tổng số phần tử: *${page.totalElements}*

Summary

- ☑ Truy vấn có sắp xếp và phân trang
  - ✓ findAll(Sort): List<T>
  - ✓ findAll(Pageable): Page<T>
- ☑ Các lớp liên quan
  - ✓ Sort
    - ✓ Sort.of(Direction, String…)
    - ✓ Sort.of(String…)
  - ✓ Pageable
    - ✓ PageRequest.of(int, int)
    - ✓ PageRequest.of(int, int, Sort)
  - ✓ Page
    - ✓ toList(): List<T>
    - ✓ getContent(): Iterable<T>
    - ✓ get<PageableInfo>()