

Pagination

**import** org.springframework.data.domain.Page;

**import** org.springframework.data.domain.PageRequest

Controller

@GetMapping("/getStudentsByPagination/{pageNumber}/{pageSize}")

**public** Page<Student> getSortedStudents(@PathVariable("pageNumber") **int** pageNumber,@PathVariable("pageSize") **int** pageSize)

{

**return** studservice.getStudentsByPagination(pageNumber, pageSize);

}

Service

**public** Page<Student> getStudentsByPagination(**int** pageNumber,**int** pageSize) {

Page<Student> studentPage= studrepo.findAll(PageRequest.*of*(pageNumber, pageSize));

**return** studentPage;

}

Postman

<http://localhost:8085/getStudentsByPagination/0/2>

offset 0 and limit 2 , offset is calculated as pageNumber\*pagesize

<http://localhost:8085/getStudentsByPagination/1/2>

offset 2 and limit 2

Pagination and Sorting

**Will first sort on the fieldname and then create the pages**

Controller

@GetMapping("/getStudentsByPaginationAndSorting/{pageNumber}/{pageSize}/{field}")

**public** Page<Student> getStudentsByPaginationAndSorting(@PathVariable("offset") **int** offset,@PathVariable("pageSize") **int** pageSize,@PathVariable("field") String field)

{

**return** studservice.getStudentsByPaginationAndSorting(pageNumber, pageSize,field);

}

Service

**public** Page<Student> getStudentsByPaginationAndSorting(**int** pageNumber,**int** pageSize,String field) {

Page<Student> studentPage= studrepo.findAll(PageRequest.*of*(pageNumber, pageSize).withSort(Sort.*by*(Sort.Direction.***ASC***, field)));

**return** studentPage;

}

Postman

<http://localhost:8085/getStudentsByPaginationAndSorting/0/3/per>

public Page getCustomers(int pageNo, int pageSize, String sortBy)

{

Pageable pageable = PageRequest.of(pageNo, pageSize, Sort.by(sortBy));

return customerRepository.findAll(pageable);

}