Logging in spring boot:

Spring-boot-starter-logging Extends from the Spring-starter-web. Log uses Log4j2 for adding the log .

Syntax :

//adding a logger to the class

Logger logger = (Logger) LoggerFactory.*getLogger*(CategoryController.**class**);

logger.trace("cstrgoty crested!");

Validating request body:

* Hibernate validator jar is already present in the spring starter web.
* It follows various validation like string, email etc
* - notnull vs notblank vs notempty
* *@NotBlank:* a constrained *String* is valid as long as it's not null and the trimmed length is greater than zero
* @NotBlank: a constrained String is valid as long as it's not null and the trimmed length is greater than zero
* @NotBlank: a constrained String is valid as long as it's not null and the trimmed length is greater than zero

1. Add not blank or not empty annotation on model .Add max min and error message on the fields
2. Add valid annotation on the request body

//https://www.baeldung.com/spring-data-jpa-pagination-sorting

1. Create a repository for pagination(can work with the same repo). This extends PagingAndSortingRepository .Although JpaRepository extends this but still we will distinctly use it because it will allow us to add our own sort and find fundtion
2. Create a method that returns Page as a return type.

In service :

Page<CategoryModel>findPaginated(Pageable pageRequest);

Impl :

@Override

**public** Page<CategoryModel> findPaginated(Pageable pageRequest) {

// **TODO** Auto-generated method stub

Page<CategoryModel>allCategories=categoryRepository.findAll(pageRequest);

**return** allCategories;

}

Controller :

@GetMapping(Category.***PAGENO***)

**public** Page<CategoryModel>getCategoryDetails(@PageableDefault(page=0,size=3)Pageable pageRequest) {

Page<CategoryModel>categoryPages = categoryService.findPaginated(pageRequest);

List<CategoryModel>categoryResult =categoryPages.getContent();

Page<CategoryModel>categoryList = **new** PageImpl(categoryResult,pageRequest,categoryPages.getTotalElements());

**return** categoryList;

}

Post man request : localhost:8080/api/page**?page=0&size=2**

**Login : Using generated security password: 3cd25120-30a9-479f-be3a-e48f7009f139**