Contents

[Java 2](#_Toc171427999)

[Stream Api 2](#_Toc171428000)

[Collection 2](#_Toc171428001)

[Spring Boot 2](#_Toc171428002)

[Rest Controller 2](#_Toc171428003)

[Bean 2](#_Toc171428004)

[Componen 2](#_Toc171428005)

[Service 2](#_Toc171428006)

[JPA 2](#_Toc171428007)

[Composite PK : Create separate class 2](#_Toc171428008)

[Pagination 2](#_Toc171428009)

[Native Query 2](#_Toc171428010)

[JDBC template 2](#_Toc171428011)

[Entity manager 2](#_Toc171428012)

[Virtual Entity class 3](#_Toc171428013)

[Batch Job 3](#_Toc171428014)

[Others 3](#_Toc171428015)

[Mapper 3](#_Toc171428016)

[Date handling 3](#_Toc171428017)

[Testing 3](#_Toc171428018)

[Unit testing 3](#_Toc171428019)

[Controller 3](#_Toc171428020)

[Service 3](#_Toc171428021)

[JPA 4](#_Toc171428022)

[Entity manager 4](#_Toc171428023)

[Utility Class 4](#_Toc171428024)

[Integration test 4](#_Toc171428025)

[Wire Mock 4](#_Toc171428026)

[Kafka 4](#_Toc171428027)

[NOS SQL DB 4](#_Toc171428028)

[Dev Ops 4](#_Toc171428029)

[Architecture 4](#_Toc171428030)

[Micro services 4](#_Toc171428031)

[12 factors App 4](#_Toc171428032)

# Java

## Stream Api

### Filter

### Map

### Grouping function

## Collection

# Spring Boot

## Rest Controller

## Bean

Component

## Service

## JPA

### Composite PK : Create separate class

In main Entity Class use @EmbeddedId rather than @ID

### Pagination

findByStatus(Long status,Pagable page); //2,PageRequest.of(pagenumber\_0,page\_size\_50,sortbt(Sorts.Direction.ASC,”status”)

## Native Query

Repository with native ,

JDBC template : Batch update ,Insert

### Entity manager

use in complex query like Virtual Entitity class with native query

Virtual Entity class: is class With only[ @Entity @Data /Lombok]no @Table and use this in Native query response , native query can be based on n number of table just use in any Real JPA repository

## Batch Job

## Others

### Mapper

### Date handling

# Testing

## Unit testing

Whenever possible use @Springtest & @MockBean as some time Mockito Extention does not mock class in child level .

Mocito is good for utility class or non SpringContext level

## Controller

@WebMVC @ContextConfiguration @MockBeans @MockMultipartFile

MainApplication(SpringBoot)

@ExtendsWith (SpringExtens.java)

## Service

@ExtendsWith (MokitoExtention.java)

@InjectMocks mainService

@Mocks /~~@MockBean~~ Other Service

MockedStatic,ReflectiontestUtil

Exception :arretThrows(Exceptionxyz.class()->{})

Envvaribale

1: Create application.yml in test and add env variable with dummy value

2: @TestPropertySource(properties = {"myproperty = foo"})

Or

static {

System.setProperty("myproperty", "foo");//direct in test method withour static

}

@Capture

ArgumentCapture<ABC.Class> classAbc;

File : Path fileclass;

@SnekyThrows

## JPA

## Entity manager

## Utility Class

@SpringBootTest

## Integration test

## Wire Mock

# Kafka

# NOS SQL DB

# Dev Ops

# Architecture

# Micro services

## 12 factors App

# Angular

Node

Component

Service

Directives

3rd party Lib

Tabular data

Model window /popup

Test case :

Unit test case