**Springboot With MongoDB -2022**

**Spring Boot Version: 2.6.6 (Upgraded due to security reason)**

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>**spring-boot-starter-data-mongodb**</artifactId> 🡸 **Spring Data Mongo**

</dependency>

**Main Application**

**@SpringBootApplication**

**@EnableMongoRepositories 🡸 This is important**

**@EnableAutoConfiguration(exclude = EmbeddedMongoAutoConfiguration.class)**

**public** **class** SampleMainApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(SampleMainApplication.**class**, args);

}

}

\*\* **The below MongoDB Configuration is not necessarily required. You can also provide the MongoDB details in application.properties file**.

# MongoDB Details 🡸 **In application.properties**

**spring.data.mongodb.uri=mongodb://localhost:27017/test**

**spring.data.mongodb.database=test**

**MongoDB Configuration**

**@Configuration**

**public** **class** SimpleMongoConfig {

**private** String mongoDBDatabaseName = "test";

@Bean

**public** MongoClient mongo() {

ConnectionString connectionString =

**new** **ConnectionString("mongodb://localhost:27017/test");**

**MongoClientSettings mongoClientSettings =** **MongoClientSettings**

**.*builder*()**

**.applyConnectionString(connectionString)**

**.build();**

**return** MongoClients.*create*(mongoClientSettings);

}

**@Bean**

**public MongoTemplate mongoTemplate() throws Exception {**

**return new MongoTemplate(mongo(), mongoDBDatabaseName);**

**}**

}

**Repository Layer**

**@Repository**

**public interface SampleDataRepository extends MongoRepository<Employee, String>** {

🡺 You can also use **CrudRepository** 🡸

}

**@Repository**

**public interface SampleDataPagingViewRepository extends PagingAndSortingRepository<Employee, Long>** {

}

**Service Implementation Class**

@Slf4j

**@Service**

**public** **class** SampleServiceImpl **implements** SampleService {

**@Autowired**

**private SampleDataRepository dataRepo;**

**@Autowired**

**private SampleDataPagingViewRepository viewPagingRepo;**

@Override

**public** **void** storeData(Employee emp) {

dataRepo.save(emp);

}

@Override

**public** **void** storeAllEmployees(List<Employee> emps) {

dataRepo.saveAll(emps);

}

@Override

**public** Employee getEmployeeById(String id) {

Optional<Employee> optEmp = dataRepo.findById(id);

**if** (optEmp.isPresent())

**return** optEmp.get();

**return** **null**;

}

@Override

**public** List<Employee> getDataByPage(Integer pageNo, Integer pageSize) {

**Pageable paging = PageRequest.*of*(pageNo, pageSize);**

**Page<Employee> pagedResult = viewPagingRepo.findAll(paging);**

**if (pagedResult.hasContent()) {**

**return pagedResult.getContent();**

**} else {**

**return new ArrayList<Employee>();**

**}**

}

}

**Entity Layer**

@Data

@Document("employee") 🡸 **Mark it**

**public** **class** Employee {

@Id

**private** String id;

**private** String firstName;

**private** String lastName;

}