Spring Boot

Spring boot is an open source java framework used to create a microservice

We are using spring boot version 2

Spring boot 2 is nothing but Spring 5 only but with auto configuration

For example if we are using hibernate with spring then we need to write lot of configuration but most of the people write similar around 90-95% similar configuration code so spring boot provide lot of default configuration so we don’t need to write lot of configuration we only need to write configuration that we want to add extra or want to override

# Spring boot 2

🡪If we want to run spring boot application then we can mention

mvn spring-boot:run

-->we can create the spring boot application manually or using [spring initializer](https://start.spring.io/)

here spring-boot is plugin and run is the goal

🡪What is @SpringBootApplication – it is equal to @Configuration+@ComponentScan+@EnableAutoConfiguration

🡪What is @Service🡪 it is @Component only but it also says that this is the service class

🡪What is @Repository🡪 it is @Component only but it also says that this is the dao or Repository class

🡪if we want to provide the configuration on data or for example on hibernate ,sql then here we have standard file (app.properties) which is under resources directory

In spring if we want to create the IOC Container when we have java annotation configuration class(for example Config.java) then we use to do like

ApplicationContext context=new AnnottaionConfigApplicationContext(Config.class, args);

In spring boot also we can create the container in this way but we can create using another way like

ApplicationContext context=SpringApplication.run(Config.class, args);

## Integrate Hibernate with Spring-boot

We have already discussed the hibernate now it’s time to integrate it with spring boot

First we will add the hibernate and sql dependencies in pom.xml file

[mysql-connector-java](https://mvnrepository.com/artifact/mysql/mysql-connector-java) and [spring-boot-starter-data-jpa](https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-data-jpa)

now we will add the configuration in app.properties file inside resources directory

now we are using spring boot so get lot of default configuration but we also need some of the extra configuration that we will write inside the app.properties file

# ===============================  
# = DATA SOURCE  
# ===============================  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.url=jdbc:mysql://localhost:3306/practice  
spring.datasource.username =root  
spring.datasource.password=scooby  
  
# ===============================  
# JPA / HIBERNATE / with mysql  
# ===============================  
spring.jpa.show-sql = false  
spring.jpa.hibernate.ddl-auto = create  
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8Dialect  
hibernate.format\_sql=true;

Database name

Previously if we need to use CRUD operation then we need EntityManger for that we used to create EntityManagerFactory and then we get EntityManager now in spring-boot we can do this in different way or in easy way

For that, inside dao implementation class we can Write

@PersistenceContext or

//Autowired

private EntityManager entityManager;

now we can do CRUD operation

@PersistenceContext is similar to @Autowired , we can use any of the annotation

Now we have persist(object),find(id),merge(object), remove method but for other methods like if we want to find by name or findAll() then directly we don’t have any methods

So for that we need HQL(hibernate query language) similar to sql query language