

Phase 3

Total session 9

Spring framework and Spring boot, Unit testing, Rest full web service and micro service using Spring framework/spring boot

Day 2

Constructor base and setter base di using annotation

1. Partial xml file configuration
2. Pure annotation base

```
@Component // <bean class="com.Address"></bean>
```

By default id name is class name in camel naming rules.

1. If class contains one word. Then id names is lower name of that class name. if class name is Address then id is **address**
2. If class contains more than one word then first word lower case and remaining word first letter upper case like EmployeeInfo then id name is **employeeInfo**.

By default @compoennt annotation is not enable we need to enable using xml file or with java class with few more annotation.

Inside xml file we need to write the one tag which help to enable @Compoennt annotation.

```
<context:component-scan base package="com"></context:component-scan>
```

Using **BeanFactory** as well as ApplicationContext interface we can pull the object from a container.

BeanFactory is core interface. ApplicationContext which internally extends BeanFactory which provide more features.

BeanFactory → GenericService

ApplicationContext → HttpServlet

1. @Component : on pojo or java bean class level annotation
2. @Value : default value for primitive property
3. @Autowired : on complex property or user defined object (that class also with @Component)
4. @Configuration : this annotation is equal to xml file configuration.
5. @ComponentScan : This annotation is equal to context component scan tag which help to enable @component annotation.

We can enable @component as well as another annotation using java class.

Spring JDBC

Spring JDBC modules provides DataSource features. Which help to maintain the database connection properly.

SpringWithDatabaseSource this project is use to improve Model layer : bean class, service class, dao class, resource class.

Spring JDBC module provided pre defined class ie DriverManagerDatasource. Which help to provide database connection using xml file configuration.

@Scope annotation : this annotation is use to provide you want singleton or new memory ie prototype.

@Repository : it is a dao layer specific annotation.

@Service : this annotation we need to use on service layer

JdbcTemplate : Spring framework provided pre defined API ie JdbcTemplate which internally wrap core jdbc code and provided extra functionality to improve Dao layer using JDBC concept.

Spring DAO provide pre defined class id JdbcTemplate to improve JDBC functionality.

Spring DAO layer.

RowMapper is a type of interface and this interface contains pre defined method which help to map each row from query.