**Spring Stereotype Annotations**

By:- Vikas

In Spring, @Autowired annotation handles wiring only. We have to define bean separately so spring container can identify them and injects them.

To overcome above issue, Spring provides stereo type annotation, that will automatically import the beans into the container and inject to dependencies. So we don’t need to define them using XML separately and that bean will be available for entire application to use. We can inject those bean in any layer of the application. All four annotations are used at class level and should be used for concrete classes not for interfaces.

These annotations are present in the **org.springframework.stereotype** package.

There are four Stereotype annotations, these are

1. **@Repository**
2. **@Component**
3. **@Service**
4. **@Controller**

**@Component type is generic stereotype for any Spring-managed component, @controller, @Service and @Repository are specialization of it.**

**@Component**

@Repository

@Controller

@Service

**Presentation Layer Business Layer DAO Layer**

Let’s discuss these four annotations in details.

**@Repository :-** This annotation is firstly introduced in spring 2.0 , this annotation is used for repository layer components from where we can directly access database.

These types of classes should be annotated with @Repository annotation for auto-detection through classpath scanning. **It also makes the unchecked exceptions** DataAccessException.

public interface EmployeeDAO

{

public Employee createNewEmployee();

}

@Repository ("employeeDao")

public class EmployeeDAOImpl implements EmployeeDAO

{

public Employee createNewEmployee()

{

Employee emp = new Employee();

emp.setId(101);

emp.setName("Vikas");

emp.setMobileNo("7416057807");

return emp;

}

}

#### @Component:-

@Component annotation is introduced in spring 2.5 version, this annotation marks class as a bean class so spring container can pick it up and pull into spring bean configuration file.

@Component

public class EmployeeDAOImpl implements EmployeeDAO

{

//……

}

**@Service:-**

@Service annotation exposes business logic which is used by repository class. If we give @Component instead of @Service, there is no issue because @Service is specialization of @Component but for better readability we should always use @Service for business layer.

@Service

public class EmployeeServiceImpl implements EmployeeService

{

//……

}

**@Controller:-**

@Controller annotation is used to make class as spring mvc controller. Whenever we use @Controller at class level then we can use another annotation like @RequestMapping to map URLs to instance methods of a class.

@Controller

public class EmployeeController

{

@RequestMapping(value = "/createEmployee", method = RequestMethod.POST)

public String createNewEmployee(@Model Employee emp)

{

return "success";

}

}

**Enable component scanning:-**

Above four annotations will be scanned by spring container when we use “context:component-scan” tag in our application-context.xml file.

application-context.xml

<context: component-scan base-package=”com.jb.stereo-demo.controller”/>

<context: component-scan base-package=”com.jb.stereo-demo.service”/>

<context: component-scan base-package=”com.jb.stereo-demo.dao”/>