WEEKLY ACCESSMENT 06

NAME:VINAY REDDY SABBELLA

PRACTICAL QUESTIONS:

Q7. What is Spring Profile?

Answer:

Spring Profiles helps segregating your application configurations, and make them available only in certain environments. ﻿ An application run on many different environments. For example, Dev, QA, Test, Stage, Production etc.

Q10.List Annotations from springboot?

Answer:

Spring Boot Annotations Everyone Should Know

1. @Bean

The @Bean annotations are used at the method level and indicate that a method produces a bean that is to be managed by Spring container. It is an alternative to the XML<bean> tag.

Example:

@Bean

Public BeanExample beanExample ()

{

return new BeanExample (),

}

2. @Service

It is used at the class level. It shows that the annotated class is a service class, such as business basic logic, and call external APIs.

Example:

@Service

public class TestService

{

public void service1()

{

// business code

}

}

Read: Spring Boot Projects & Topics

3. @Repository

It is a Data Access Object (DAO) that accesses the database directly. It indicates that the annotated class is a repository.

Example:

@Repository

public class TestRepository

{

public void delete()

{

// persistence code

}

}

4. @Configuration

It is used as a source of bean definitions. It is a class-level annotation.

Example:

@Configuration

public class Bus

{

@BeanBus engine()

{

return new Bus();

}

}

5. @Controller

The annotation is used to indicate that the class is a web request handler. It is often used to present web pages. It is most commonly used with @RequestMapping annotation.

Example:

@Controller

@RequestMapping(“cars”)

public class CarsController

{

@RequestMapping(value= “/{name}”, method= RequestMethod.GET)

public Employee getCarsByName()

{

Return carsTemplate;

}

}

6. @RequestMapping

RequestMapping is used to map the HTTP request. It is used with the class as well as the method. It has many other optional elements like consumes, name, method, request, path, etc.

Example:

@Controller

public class FlowersController

{

@RequestMapping (“/red-colour/flowers”)

public String getAllFlowers(Model model)

{

//application code

return “flowerlist”;

}

7. @Autowired

This annotation is used to auto-wire spring bean on setter methods, constructor and instance variable. It injects object dependency implicitly. When we use this annotation, the spring container auto-wires the bean by its matching data type.

Example:

@Component

public class Employee

private Person person;

@Autowired

public Employee(Person person)

{

this.person=person

}

}

8. @Component

It is a class-level annotation that turns the class into Spring bean at the auto-scan time.

Example:

@Component

Public class Teachers

{

……

}

9. @SpringBootApplication

It consists of @Configuration, @ComponentScan, and @EnabeAutoConfiguration. The class annotated with @SpringBootApplication is kept in the base package. This annotation does the component scan. However, only the sub-packages are scanned.

10. @EnableAutoConfiguration

It is placed on the main application class. Based on classpath settings, other beans, and various property settings, this annotation instructs SpringBoot to start adding beans.

11. @ComponetScan

It is used to scan a package of beans. It is used with the annotation @Configuration to allow Spring to know the packages to be scanned for annotated components. This annotation is also used to specify base packages.

Example:

@ComponentScan(basePackages = “com.xyz”)

@Configuration

Public class ScanComponent

{

//…

}

12. @Required

This annotation is applied to bean setter methods. It indicates that the required property must be filled at the configuration time in the affected bean, or else it throws an exception: BeanInitializationException.

13. @Qualifier

It is used along with @Autowired annotation. It is used when more control is required over the dependency injection process. Individual constructor arguments or method parameters can be specified by using this annotation. Confusion arises when more than one bean of the same type is created, and only one of them is to be wired with a property, @Qualifier is used to get rid of the confusion.

14. @CookieValue

It is used at the method parameter level as an argument of the request mapping method. For a given cookie name, the HTTP cookie is bound to a @CookieValue parameter.

15. @Lazy

It is used in the component class. At startup, all auto-wired dependencies are created and configured. But a @Lazy annotation can be created if a bean is to be initialized lazily. This means that only if it is requested for a bean will be created. It can also be used on @Configuartion classes. It’s an indication that all @Bean methods within that @Configuration should be lazily initialized.

Q9.What is Junit and annotations used in junit?

Answer:

JUnit Annotations is a special form of syntactic meta-data that can be added to Java source code for better code readability and structure. Variables, parameters, packages, methods and classes can be annotated.