Annotations in Spring

Discuss the usage of annotations like @Component, @Autowired, @Qualifier, and @Scope in Spring. Provide examples of how these annotations are used to configure Spring beans and manage dependencies without XML configuration.

In Spring framework, annotations provide a more concise and flexible way to configure beans and manage dependencies compared to XML configuration. Let's discuss the usage of several key annotations: @Component, @Autowired, @Qualifier, and @Scope.

**1. @Component**

The @Component annotation is used to mark a Java class as a bean so that the Spring container can manage it. It's a generic stereotype annotation for any Spring-managed component.

Ex:

@Component

public class UserService {

public void doSomething() {

System.out.println("UserService: doing something...");

}

}

* @Component marks the UserService class as a Spring bean.
* Spring automatically detects and registers UserService in the application context.

**2. @Autowired**

The @Autowired annotation is used to inject dependencies automatically into a Spring bean.

**Example:**

@Component

public class UserController {

private UserService userService;

@Autowired

public UserController(UserService userService) {

this.userService = userService;

}

public void invokeUserService() {

userService.doSomething();

}

}

 @Autowired injects the UserService bean into the UserController constructor.

 Spring resolves and provides the UserService bean dependency automatically.

### 3. @Qualifier

The @Qualifier annotation is used along with @Autowired to specify which bean should be injected when multiple beans of the same type are present.

**Example:**

@Component

public class UserController {

private UserService adminService;

@Autowired

public UserController(@Qualifier("adminService") UserService adminService) {

this.adminService = adminService;

}

public void invokeAdminService() {

adminService.doSomething();

}

}

@Component

@Qualifier("adminService")

public class AdminService implements UserService {

public void doSomething() {

System.out.println("AdminService: doing something...");

}

}

* @Qualifier("adminService") specifies that the AdminService bean should be injected into UserController.
* This resolves ambiguity when there are multiple beans of the same type (UserService).

**4. @Scope**

The @Scope annotation defines the scope of a Spring bean.

**Example:**

@Component

@Scope("prototype")

public class PrototypeBean {

// Bean definition

}

* @Scope("prototype") specifies that instances of PrototypeBean should be scoped as prototypes.
* Prototype scope means that a new instance of the bean is created each time it is requested.

**Using Annotations for Dependency Injection**

Annotations like @Component, @Autowired, @Qualifier, and @Scope eliminate the need for XML configuration in many cases. They provide a more concise and readable way to define beans and manage dependencies in Spring applications.