

## Scope

Spring beans have different scopes that define their lifecycle. The two most commonly used scopes are:

- **Singleton:** (Default) Only one instance of the bean is created and shared across the application.
- **Prototype:** A new instance of the bean is created every time it is requested.

### Steps:

1. By default, Spring beans are singleton-scoped.
2. To change the scope, use the @Scope annotation and specify the scope type (e.g., "prototype").

### Configuration class:

```
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Scope;

@Configuration
public class AppConfig {

    // Singleton scope (default)
    @Bean
    public Desktop desktopSingleton() {
        return new Desktop();
    }

    // Prototype scope (new instance each time)
    @Bean
    @Scope("prototype")
    public Desktop desktopPrototype() {
```

```
    return new Desktop();  
    }  
}
```

## Main Class:

```
// Main class to demonstrate scope  
public class App {  
    public static void main(String[] args) {  
        ApplicationContext context = new  
        AnnotationConfigApplicationContext(AppConfig.class);  
  
        // Singleton scope  
        Desktop dt1 = context.getBean("desktopSingleton", Desktop.class);  
        Desktop dt2 = context.getBean("desktopSingleton", Desktop.class);  
        System.out.println(dt1 == dt2); // Output: true (Same instance)  
  
        // Prototype scope  
        Desktop dt3 = context.getBean("desktopPrototype", Desktop.class);  
        Desktop dt4 = context.getBean("desktopPrototype", Desktop.class);  
        System.out.println(dt3 == dt4); // Output: false (Different instances)  
    }  
}
```

## Key Points:

- **Singleton Scope:** One shared instance for the entire application.
- **Prototype Scope:** A new instance is created each time the bean is requested.

