**🎲 Singleton**

When a bean has a singleton scope, it means that **only a single instance of the bean is created per Spring application context**. This instance is stored in memory and reused each time the bean is requested in the future, rather than creating a new instance for each request.

@Component  
@Scope("singleton")  
public class SingletonBean {  
   
}

Here are some key features of the singleton scope in Spring:

* **Single instance**: Only one instance of the bean is created in the entire Spring application lifecycle. Each time the bean is requested, the same previously created instance is returned, if it already exists.
* **Stored in memory**: The bean instance is stored in memory, which means that it persists for the lifetime of the application and is available for use at any time.
* **Shared**: The same singleton bean is shared between different components or classes that inject it into their Spring application context. This can have implications for concurrency and bean state, so care must be taken when handling shared data in singleton beans.
* **Efficienc**y: Using singleton beans can improve the efficiency and performance of the application by avoiding the overhead of creating multiple instances of the same bean for each request.

🚩**It is important to note that the singleton scope in Spring is the DEFAULT scope for beans, so if no scope is specified, it is considered as singleton. However, care should be taken when using singleton beans in concurrent or stateful applications, as this can have implications for concurrency and state integrity.**