In the context of Spring Framework, bean scopes determine the lifecycle and visibility of the beans. The two commonly used bean scopes are Singleton and Prototype.

**Singleton Scope**

* **Definition**: A singleton-scoped bean means that the Spring container creates only one instance of the bean per Spring IoC container.
* **Lifecycle**: The single instance is created at the time of container startup or when it is first requested. Subsequent requests will return the same instance.
* **Usage**: Singleton is the default scope in Spring. It is useful for stateless beans or when you want to share the same instance across the application.
* **Configuration**: Can be explicitly declared using the **@Scope** annotation or in XML configuration.



### Prototype Scope

* **Definition**: A prototype-scoped bean means that a new instance of the bean is created every time it is requested from the Spring container.
* **Lifecycle**: The instance is created, initialized, and returned upon each request. The Spring container does not manage the complete lifecycle (e.g., it does not call destroy methods).
* **Usage**: Prototype scope is useful for stateful beans, where each client needs its own instance with its own state.
* **Configuration**: Can be declared using the **@Scope** annotation or in XML configuration.

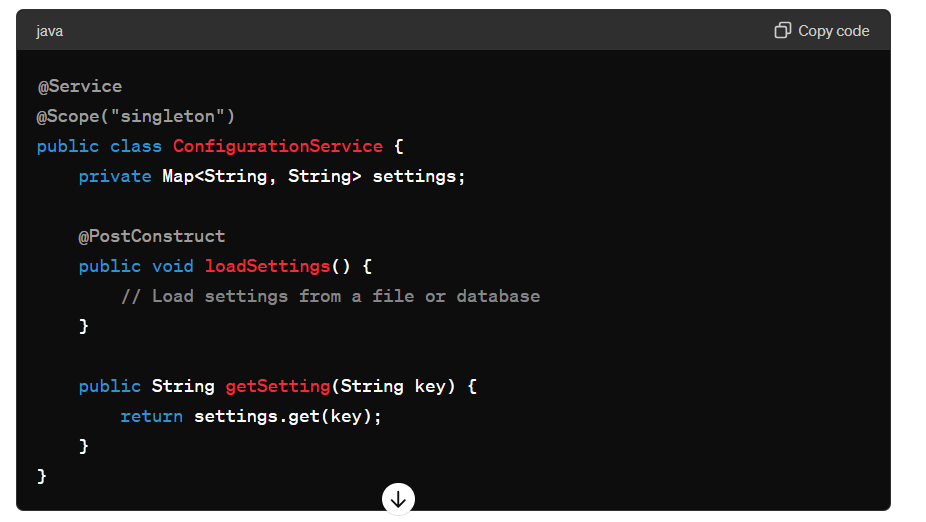


**Key Differences**

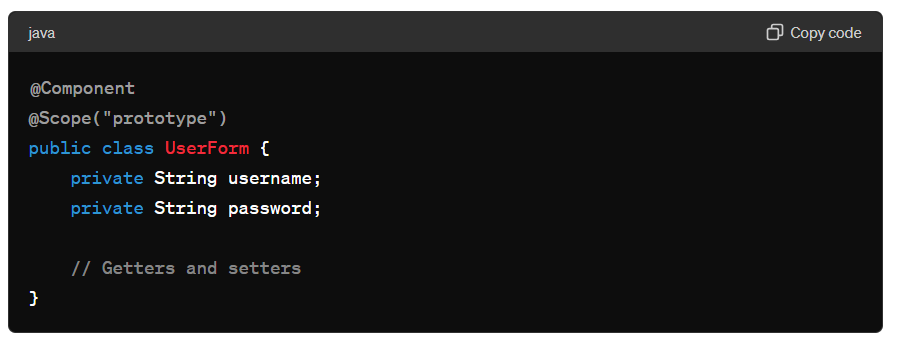
1. **Instance Count**:
   * **Singleton**: Only one instance per Spring container.
   * **Prototype**: Multiple instances, one per request.
2. **Lifecycle Management**:
   * **Singleton**: Fully managed by Spring; both initialization and destruction callbacks are called.
   * **Prototype**: Only initialization is managed; destruction callbacks are not called.
3. **Use Case**:
   * **Singleton**: Suitable for stateless, shared resources.
   * **Prototype**: Suitable for stateful objects where a new instance is needed for each operation.

**Example Scenarios**

* **Singleton Bean**: A service that holds configuration settings read from a file or database, which are shared across the application.



**Prototype Bean**: A form backing object in a web application, where each HTTP request needs its own instance.



In summary, the Singleton scope is ideal for beans that should be shared across the application and have a single instance, while the Prototype scope is suitable for beans that need to be instantiated multiple times with unique state per request.