Bean Scopes

Explain the different bean scopes supported by Spring (singleton, prototype, request, session, etc.). Describe each scope and provide an example scenario where each scope would be appropriate. Show how to define bean scopes using XML configuration.

**Different Bean Scopes in Spring:**

1. **Singleton Scope**:
   * **Description**: Only one instance of the bean is created for the entire application context (per container). It is the default scope in Spring.
   * **Example Scenario**: When you want to share the same instance of a bean across all clients, ensuring that changes to the bean's state are visible to all.

**XML Configuration**:

<bean id="singletonBean" class="com.example.SingletonBean" scope="singleton"/>

1. **Prototype Scope**:

* **Description**: A new instance of the bean is created each time it is requested (using dependency injection or getBean() method).
* **Example Scenario**: When you want a new instance of the bean each time it is injected or requested, especially if the bean maintains state specific to the caller.

**XML Configuration**:

<bean id="prototypeBean" class="com.example.PrototypeBean" scope="prototype"/>

**3. Request Scope**:

* **Description**: A new instance of the bean is created for each HTTP request.
* **Example Scenario**: In web applications, when you want to store data specific to a user's request, such as form data or user session details.

**XML Configuration**:

<bean id="requestScopedBean" class="com.example.RequestScopedBean" scope="request"/>

**4. Session Scope**:

* **Description**: A new instance of the bean is created for each HTTP session.
* **Example Scenario**: In web applications, when you want to store user-specific data across multiple requests within the same session, such as user preferences or shopping cart items.

**XML Configuration**

<bean id="sessionScopedBean" class="com.example.SessionScopedBean" scope="session"/>

**5. Global Session Scope**:

* **Description**: Similar to session scope but specifically used for applications deployed in a portlet context.
* **Example Scenario**: When developing portlets and needing to maintain session state across multiple portlets within the same application.

**XML Configuration**

<bean id="globalSessionScopedBean" class="com.example.GlobalSessionScopedBean" scope="globalSession"/>

**6. Application Scope**:

* **Description**: One instance of the bean for each Spring ServletContext.
* **Example Scenario**: When you want to store application-wide data or configuration that needs to be shared across all sessions and requests within the same application context.

**XML Configuration**

<bean id="applicationScopedBean" class="com.example.ApplicationScopedBean" scope="application"/>

Ex:-

<!-- Singleton scope (default) -->

<bean id="singletonBean" class="com.example.SingletonBean" />

<!-- Prototype scope -->

<bean id="prototypeBean" class="com.example.PrototypeBean" scope="prototype" />

<!-- Request scope (for web applications) -->

<bean id="requestScopedBean" class="com.example.RequestScopedBean" scope="request" />

<!-- Session scope (for web applications) -->

<bean id="sessionScopedBean" class="com.example.SessionScopedBean" scope="session" />

<!-- Global session scope (for portlet applications) -->

<bean id="globalSessionScopedBean" class="com.example.GlobalSessionScopedBean" scope="globalSession" />

<!-- Application scope (for web applications) -->

<bean id="applicationScopedBean" class="com.example.ApplicationScopedBean" scope="application" />

 singletonBean and prototypeBean are defined with scope="singleton" and scope="prototype" explicitly.

 requestScopedBean, sessionScopedBean, globalSessionScopedBean, and applicationScopedBean are defined with their respective scopes (request, session, globalSession, application) suitable for web applications.