# Spring Boot Complete Tutorial - Master Class

<https://www.youtube.com/watch?v=zvR-Oif_nxg>

## Beans scopes

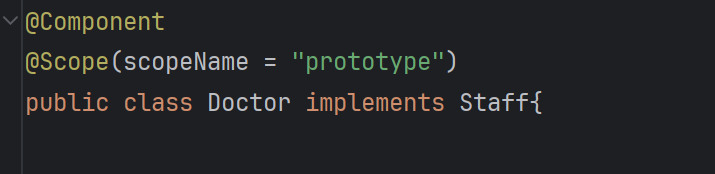
<https://docs.spring.io/spring-framework/reference/core/beans/factory-scopes.html>

### Singleton

Only one shared instance of a singleton bean is managed, and all requests for beans with an ID or IDs that match that bean definition result in that one specific bean instance being returned by the Spring container.

### Prototype

The non-singleton prototype scope of bean deployment results in the creation of a new bean instance every time a request for that specific bean is made. That is, the bean is injected into another bean or you request it through a getBean() method call on the container. As a rule, you should use the prototype scope for all stateful beans and the singleton scope for stateless beans.



### Request, Session, Application, and WebSocket Scopes

The request, session, application, and websocket scopes are available only if you use a web-aware Spring ApplicationContext implementation (such as XmlWebApplicationContext). If you use these scopes with regular Spring IoC containers, such as the ClassPathXmlApplicationContext, an IllegalStateException that complains about an unknown bean scope is thrown.

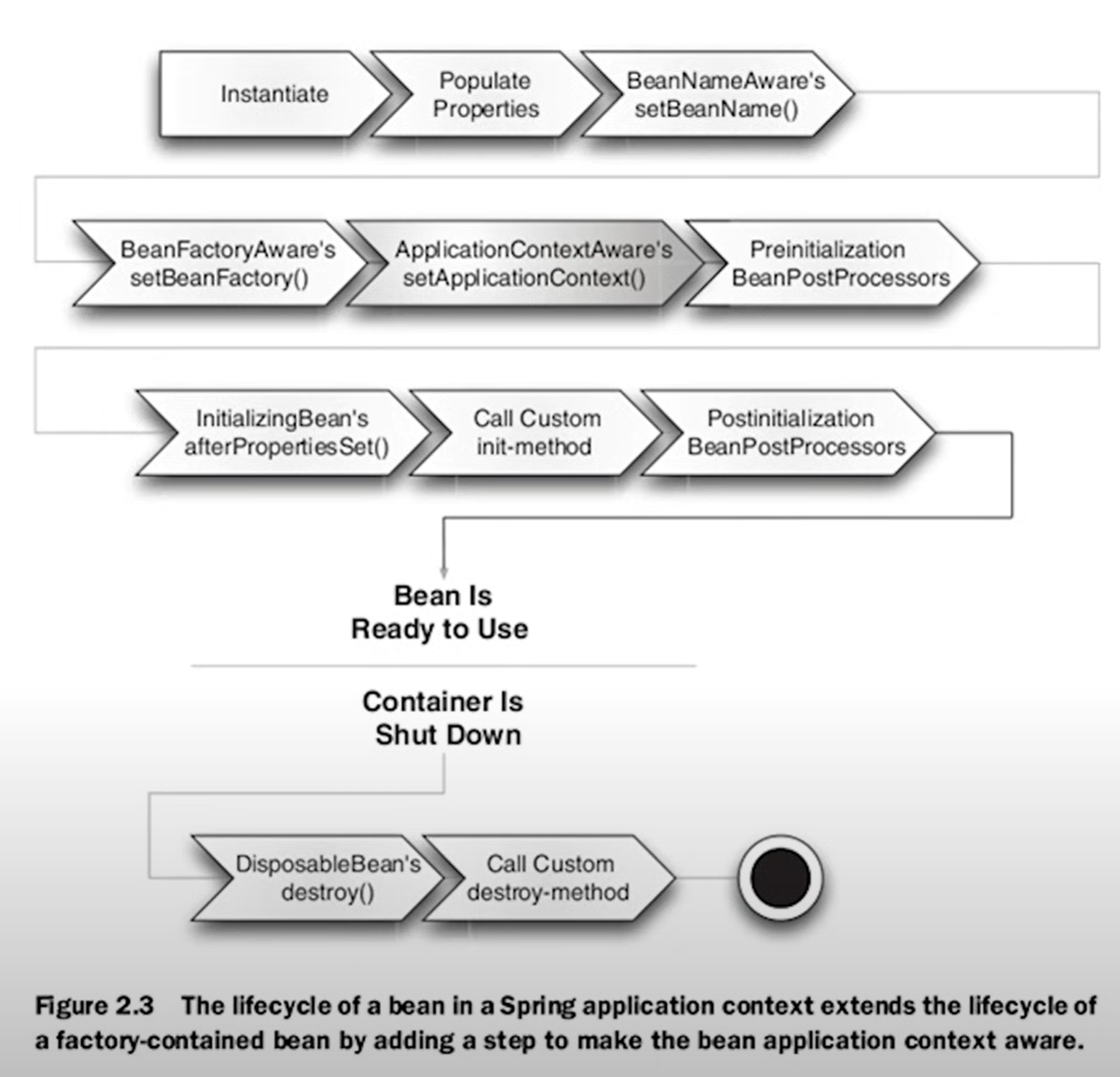
Request: Scopes a single bean definition to the lifecycle of a single HTTP request.

Session: Scopes a single bean definition to the lifecycle of an HTTP session.

Application: Scopes a single bean definition to the lifecycle of a ServletContext.

Websocket: Scopes a single bean definition to the lifecycle of a WebSocket.

### Beans Lifecycle



## Spring AOP

Aspect-oriented programming (AOP) is a programming paradigm that aims to modularize cross-cutting concerns, which are functionalities that affect multiple parts of a software application. It allows the separation of such concerns from the main application logic, promoting better modularity and reusability. AOP achieves this by creating entities called "aspects" that encapsulate these cross-cutting concerns and can be applied across different parts of the application.

i.e.

* Logging
* Authorization and authentication