

Beans

pom -> project object model

A Spring Bean is simply a Java object that is:

- Instantiated by Spring
- Configured by Spring
- Managed throughout its lifecycle by Spring

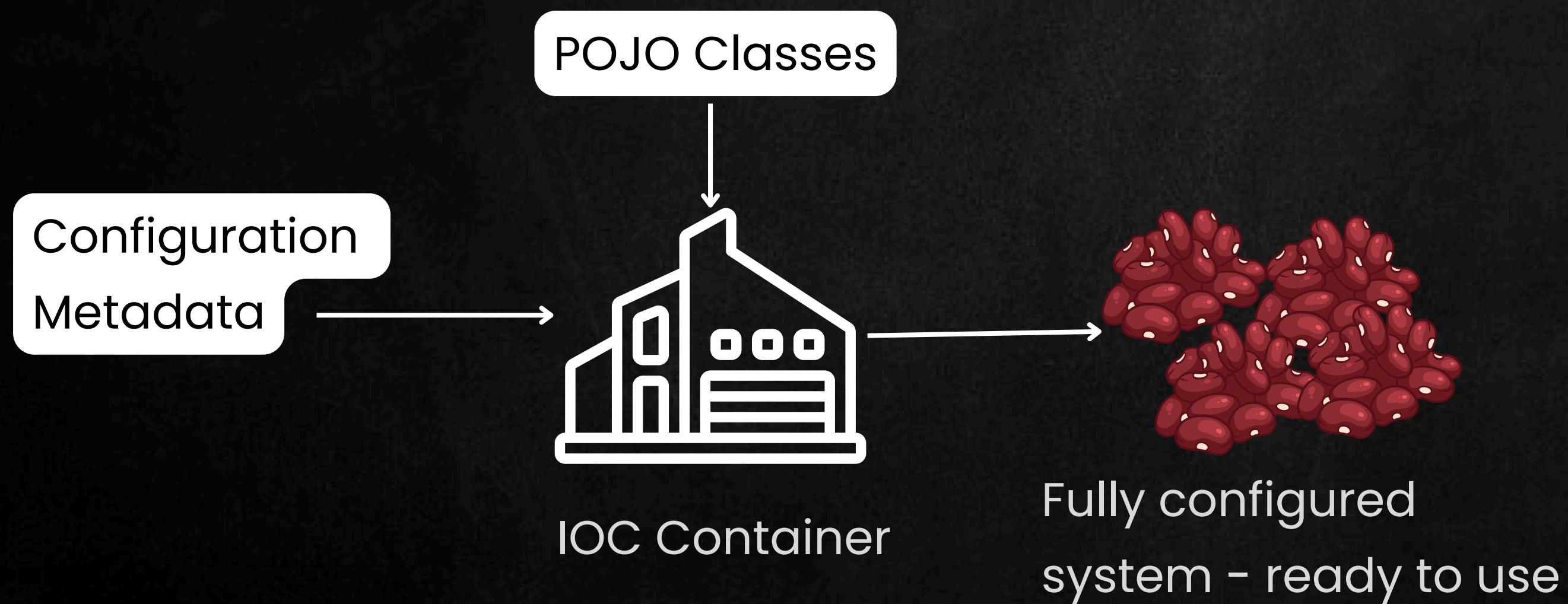
Beans form the backbone of a Spring application and are the core building blocks that are wired together to create the application.



IOC Container aka Application Context

IOC Container - The Bean Factory

At the center of everything is the ApplicationContext – Spring's IoC (Inversion of Control) container.



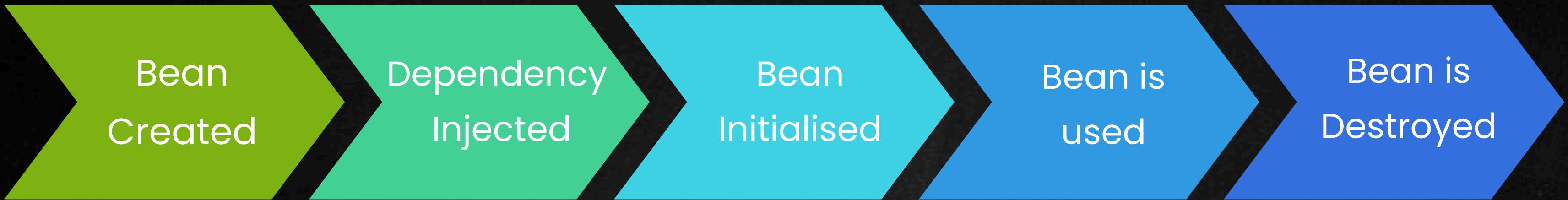
How to create Beans?

1. Using @Component (and Stereotype annotations)

- `@Component` // generic bean
- `@Service` // business logic
- `@Repository` // database access
- `@Controller` // web layer
- `@RestController`

2. Using @Bean in a @Configuration Class

Lifecycle of a Bean

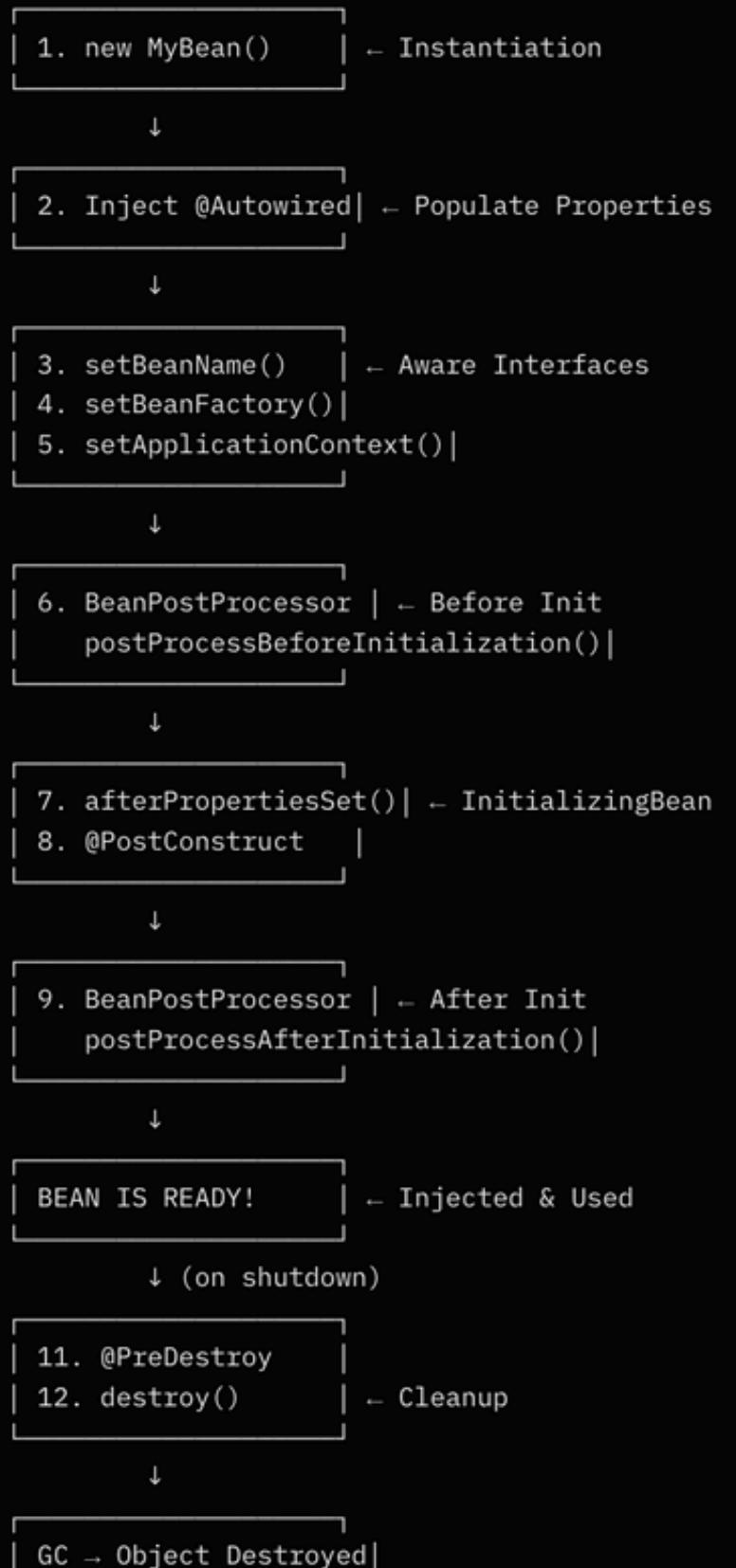


Bean Lifecycle Methods

The **@PostConstruct** annotation is used to mark a method that should be invoked immediately after a bean has been constructed and all of its dependencies have been injected.

The **@PreDestroy** annotation is used to mark a method that should be invoked just before a bean is destroyed by the container. This method can perform any necessary cleanup or resource releasing tasks.

Bean Lifecycle Methods



Bean scopes

Scope	use-case
singleton	One instance per app, use for Services, repositories
prototype	New instance every time, use for Stateful objects, request data
request	One per HTTP request, use for Web apps only
application	One per ServletContext, use for Global config
session	One per user session, use for Shopping cart

