

Spring

Java framework based on Dependency Injection that provides comprehensive infrastructure support for developing apps.

Create & store dependencies in the Spring container & injects them into registered objects (aka “beans”, which are simply objects managed by the spring container).

Dependency Injection (D.I.) is often referred to as a “flavor” of Inversion of Control (IoC), in which control of object creation is given to the container as opposed to being managed explicitly by the developer.

D.I. decouples execution from implementation

Spring Bean Lifecycle

- 1.) Instantiate
- 2.) Populate Properties
- 3.) Set BeanNameAware
- 4.) Set BeanFactoryAware
- 5.) Set ApplicationContext
- 6.) Pre-initialization BeanPostProcessor
- 7.) InitializingBeanAfterPropertiesSet
- 8.) Call custom init method : bean is ready
- 9.) Post-initialization BeanPostProcessor
- 10.) Disposable Beans : destroy()
- 11.) custom destroy() : container shut down