



# Bean Loading

By Rahul Barve



# Bean Loading

- In Spring, Bean Loading happens by 2 ways:
  - EAGER (DEFAULT)
  - LAZY



# Bean Loading

- The bean registered in the configuration unit gets instantiated as soon as the `ApplicationContext` is built.
- This is known as EAGER Loading.



# Bean Loading

- The bean registered in the configuration unit gets instantiated only when the client program makes a request for the same.
- This is known as LAZY Loading.



# **AutoWiring**

**By Rahul Barve**



# AutoWiring

- Rather than explicitly wiring all of your bean's properties, you can have Spring automatically figure out how to wire beans.
- It is done by setting the **autowire** property.



# AutoWiring

- Spring provides 3 types of autowiring:
  - byName
  - byType
  - constructor



# AutoWiring

- byName
  - Attempts to find a bean in the container whose name is the same as the name of the property being wired.





# AutoWiring

- byType
  - Attempts to find a single bean in the container whose type matches the type of the property being wired.



# AutoWiring

- constructor
  - Tries to match up a constructor of the autowired bean with beans whose types are assignable to the constructor arguments.



# Bean Scopes



## Bean Scopes

- Every bean registered in XML file has some scope.
- It is possible to modify scope of the bean using `scope` attribute of `<bean>` element.



# Bean Scopes

- There are 5 different types of scopes:
  - singleton
  - prototype
  - request
  - session
  - global-session



# Bean Scopes

- singleton
  - It is the default scope.
  - Indicates that the bean configuration is singleton.
  - If the same bean is requested multiple times, spring returns the same object.



# Bean Scopes

- prototype
  - Antonym of singleton.
  - If the same bean is requested multiple times, spring returns the a new object every time.



# Bean Scopes

- request
  - Applicable only in the context of Spring MVC.
  - The bean is alive until the response is generated.
  - For every new instance of `HttpServletRequest`, spring creates a new instance.





# Bean Scopes

- session
  - Applicable only in the context of Spring MVC.
  - The bean is alive until the session is over.
  - Bean can survive even if the response is generated.
  - For every new instance of HttpSession, spring creates a new instance.



# Bean Scopes

- global-session
  - Applicable in the context of Spring Portlet environment.
  - The bean is alive across multiple portlets.



## Let's Summarize

- Working with ApplicationContext
- Bean Life Cycle
- Understanding Dependency Injections
- Auto-wiring
- Bean Scopes