Bean Scope :

When defining a <bean> in Spring in xml file, you have the option for declaring a scope for that bean.

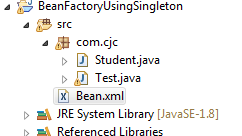
The Spring Framework supports following six scopes-

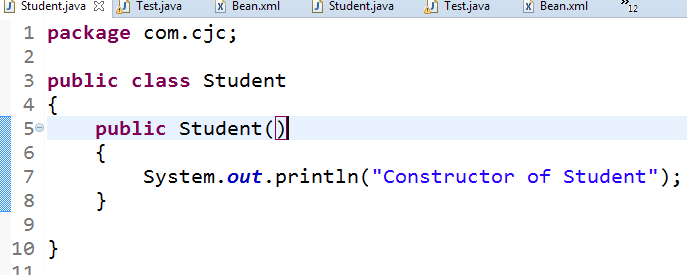
* + *singleton* : This scopes the bean definition to a single instance per Spring IoC container (default).
  + *prototype* : This scopes a single bean definition to have any number of object instances. Opposite to singleton, it produces a new instance each and every time a bean is requested.
  + *request* : This scopes a bean definition to an HTTP request. Only valid in the context of a web-aware Spring ApplicationContext.
  + *session* : This scopes a bean definition to an HTTP session. Only valid in the context of a web-aware Spring ApplicationContext.
  + *websocket*: A single instance will be created and available during complete lifecycle of WebSocket.

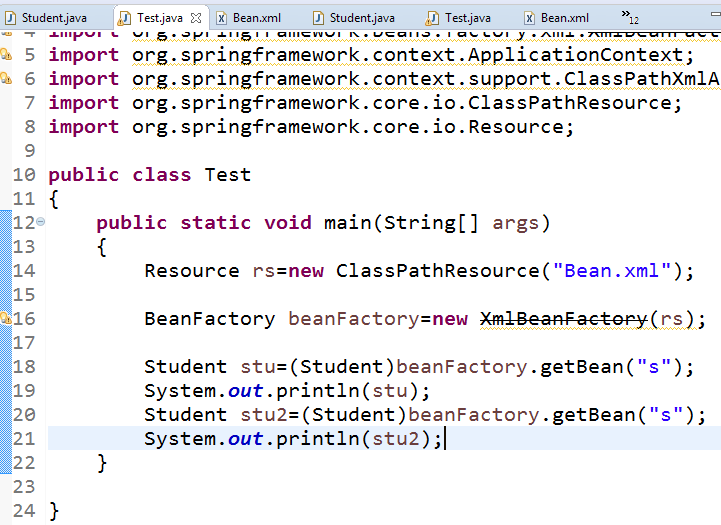
Only valid in web-aware Spring ApplicationContext

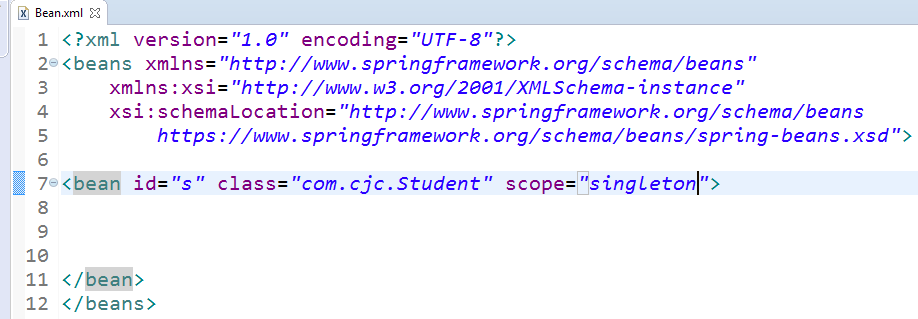
* + *application*: A single instance will be created and available during complete lifecycle of ServletContext. Only valid in web-aware Spring ApplicationContext.

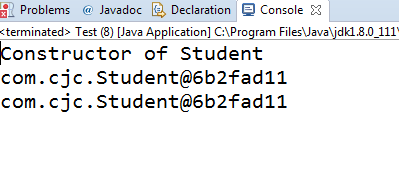
**Singleton example:**



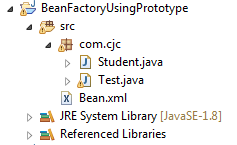


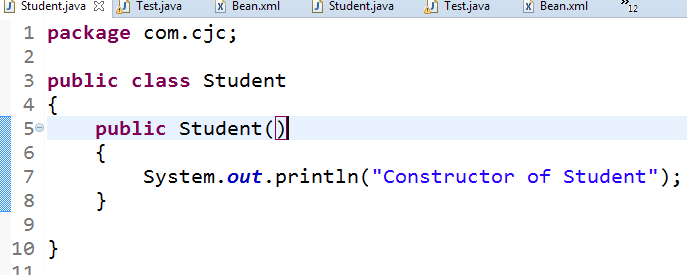


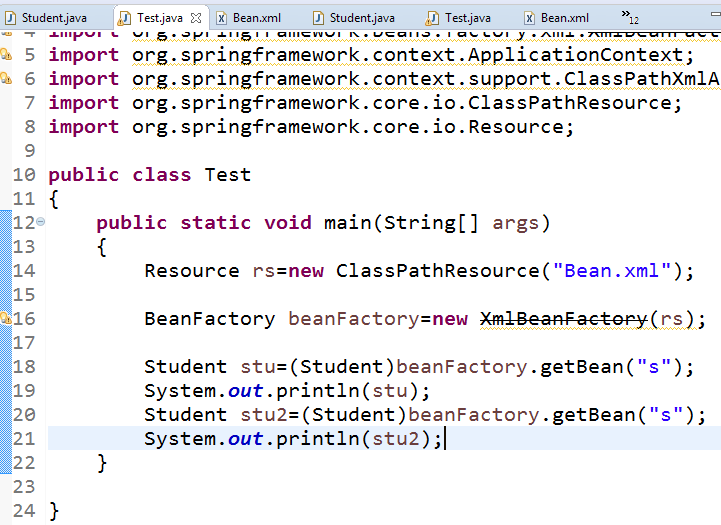


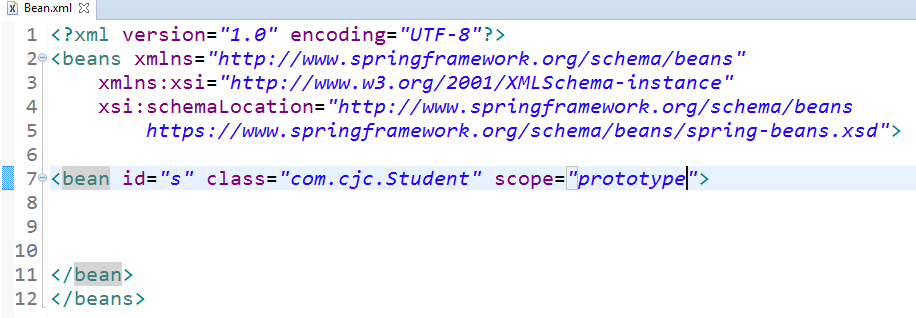
****

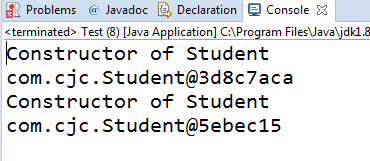
**2)prototype Scope:**











Lazy & Eager Loading :

For Singleton scope : only one object for bean will be created

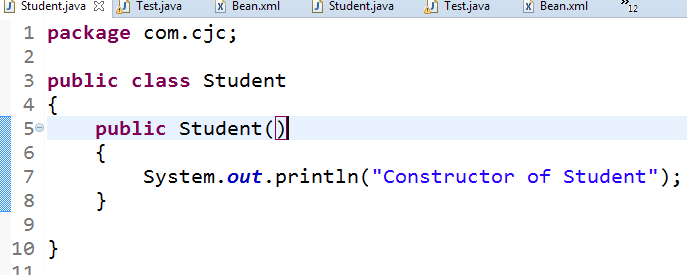
* BeanFactory : lazy-loading (i.e. object is created on request)
* ApplicationContext : eager- loading (i.e. object is created when xml file loads)

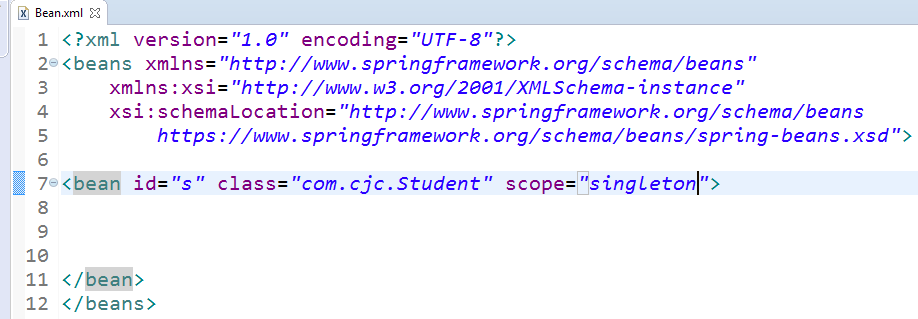
Let us see above concept practically...

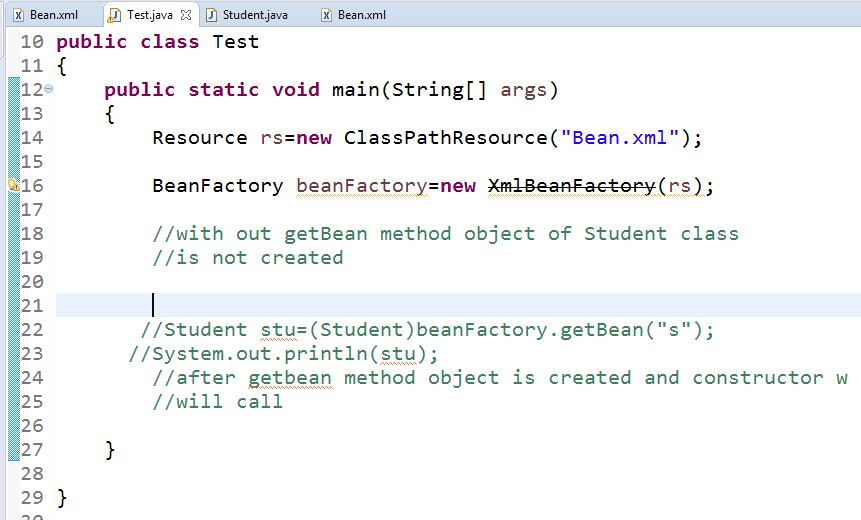
For prototype Scope:

* BeanFactory : lazy-loading (i.e. object is created on request)
* ApplicationContext : Lazy- loading (i.e. object is created on request)

Let us see above concept practically...







If BeanFactory scope is prototype then also lazy loading will Occure

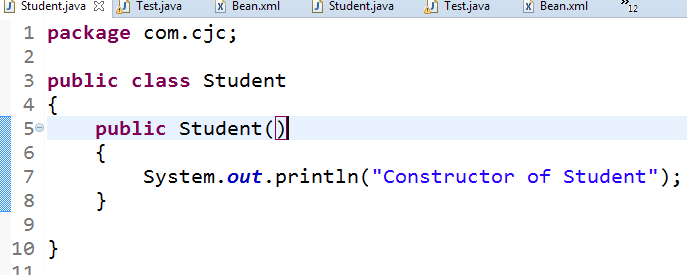
All program is same as previous program

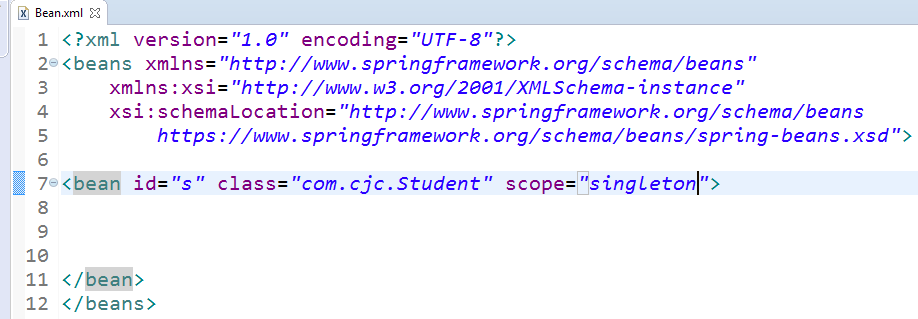
Only change scope=prototype

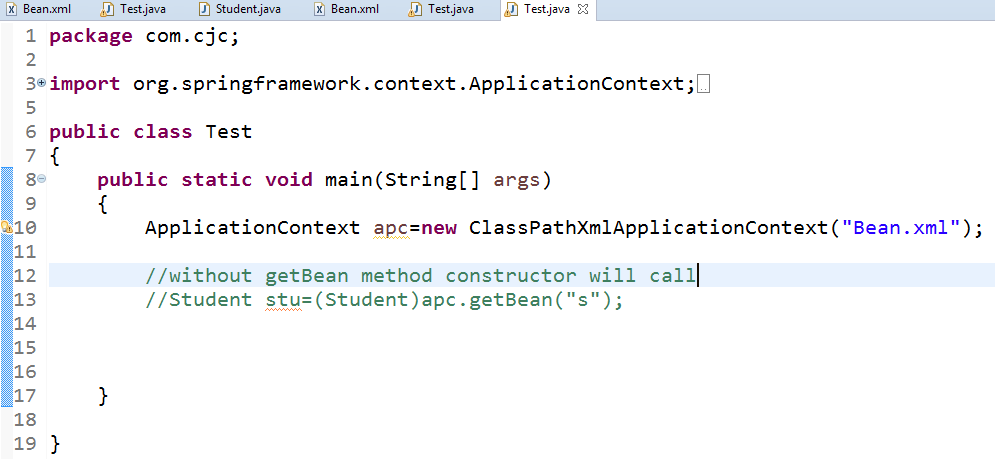
* ApplicationContext : eager- loading (i.e. object is created when xml file loads)

Let us see above concept practically...

For this scenario eager loading will occure







LazyLoading And EagerLoading both container:

|  |  |  |
| --- | --- | --- |
| Containers | Singleton | Prototype |
| BeanFactory | LazyLoading | LazyLoading |
| ApplicationContext | EagerLoading | LazyLoading |