**Spring**

**Beans** – obiectele care formeaza “coloana vertebrala” a aplicatiei si sunt gestionate de containerele IoC din Spring

|  |  |
| --- | --- |
| Proprietati | Descriere |
| class | Se specifica clasa care este folosita pentru a crea un bean. |
| name | Reprezinta numele de identificare a bean-ului creat. |
| scope | Reprezinta scopul pentru care a fost creat un bean. |
| contruct-arg | Folosit pentru a injecta dependinta |
| properties | Folosit pentru a injecta dependinta |
| autowiring mode | Folosit pentru a injecta dependinta |
| lazy-initialization mode | Transmite container-ului IoC sa creeze bean-ul doar atunci cand este cerut prima data. |
| initialization method | Se apeleaza inainte ca toate proprietatile necesare unui bean sa fie setate |
| descruction method | Se apeleaza la distrugerea unui bean. |

**Scopul unui bean**

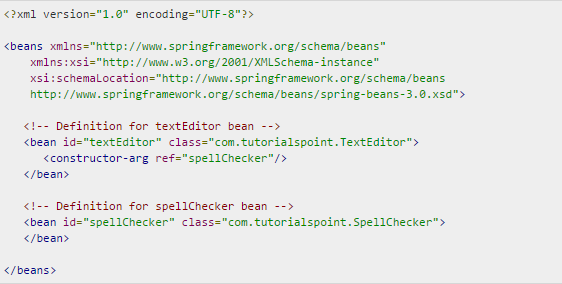
* Singleton ( o singura instant per intreg container-ul IoC )
* Prototype ( o singura definitie a unui bean ii permite sa aibe oricate instante )
* Request ( definit pe o cerere HTTP )
* Session (definiti pe o cerere HTTP session )
* Global-Session ( definit pe o sesiune HTTP globala )

**Dependency Injection**

* atunci cand scriem o aplicatie Java complexa, clasele trebuie sa fie cat mai independente posibil
* aceasta metoda ne ajuta sa unim aceste clase impreuna, iar in acelasi timp sa le pastram independente
* se poate implementa prin “class constructor” sau prin “setter methods”

**DI folosind class constructor**

* configuram fisierul .xml

****

**DI folosind setter Methods**

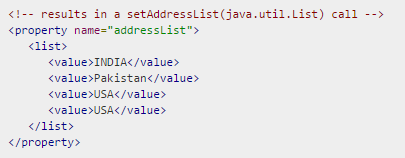
* configuram fisierul .xml

****

Id-ul “spellChecker” se refera la numele obiectului care trebuie instantiat, iar “ref” face legatura cu bean-ul care are legatura cu aceasta clasa.

**“Injectarea” de colectii**

* **<list>**



* **<set>**



* **<map>**

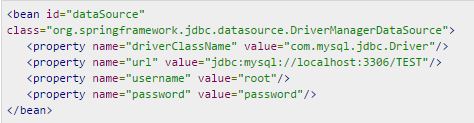


* **<property>**



**JDBC Framework**

* se foloseste clasa JdbcTemplate, care executa interogari SQL.
* instantierea clasei JdbcTemplate este thread safe odata ce a fost configurata. Deci putem configure o singura instant a acestei clase si apoi sa o injectam in fiecare DAO.
* o practica folosita este configurarea DataSource in fisierul de configurare Spring ( .xml)
* configurarea DataSource se realizeaza astfel:

****