

Spring frameworkünün kullandığı design patternlar neler?

- **Factory Method:** BeanFactory bu pattern üzerine inşa edilmiştir. Spring managed beanların yaratılması ve bağımlılıkların sağlanmasında kullanılır.
- **Singleton:** Yaratılan bean'ler default olarak container genelinde tek bir instance'a sahip olmaktadır. Spring'in Singleton implemantasyonu GOF Singleton Pattern'ından scope olarak biraz daha farklıdır.
- **Prototype:** Yazılım geliştiricisi istediği takdirde ApplicationContext herhangi bir bean tanımından her `getBean('beanName')` ile erişimde yeni bir instance yaratmaktadır. Bean tanımı burada tam bir prototype pattern'dir.
- **Proxy:** AOP'de yaygın olarak uygulanmaktadır. Spring Application Framework'ün en çok faydalandığı pattern diyebiliriz. Scoped bean oluşturmada, Spring AOP kabiliyetinin sunulmasında, TX kabiliyetinin implemantasyonunda hep bu pattern kullanılmaktadır.
- **Template Method:** Spring'in veri erişim altyapısı bu pattern üzerine kuruludur. JdbcTemplate, HibernateTemplate...
- **Observer:** ApplicationContext'in event yönetimi tam bir publish subscribe örneğidir. ApplicationContext bir event medium rolündedir. Bir grup spring managed bean ApplicationContext vasıtası ile ApplicationListener tipindeki diğer bir grup bean'lere event notifikasyonunda bulunabilirler.
- **Mediator:** Sadece bildikleri ApplicationContext'in kendisidir, yani mediator.
- **Front Controller:** Spring MVC Framework'ün iskeletini oluşturan DispatcherServlet bu pattern'ın bire bir karşılığıdır.