

# Web Programlama 2


## Ders 08: Spring

# Slaytlar Hakkında

- Bu slaytlar oldukça yüksek seviyede genel bir bakış açısı sunmaktadır
- Detaylar Git repository'sindeki yorum satırlarında bulunmaktadır

# Spring Framework

- Açık kaynak, ilk olarak 2002 yılında yayınlandı
- Enterprise/web uygulamalar geliştirmek için framework
- *Pivotal Software* tarafından geliştirilip/destekleniyor
- JEE'nin daha hafif alternatifi olarak ortaya çıktı
- Şu anda oldukça karmaşıktır ancak pek çok JEE bakış açısını paylaşır/kullanır
- Muhtemelen en iyi dokümantasyona sahip frameworklerden biridir
  - Bakınız <https://spring.io>



# Neden adı “Spring”?

JEE kışından (winter) sonra gelen *Spring*...

---

# Spring vs SpringBoot

- Spring'in veritabanları, web sayfaları, web servisleri vs. barındıran pek çok modülü bulunmaktadır.
- Spring uygulamasını birbirine bağlamak için çok fazla ayar yapılması gerekir
- *SpringBoot* (2012): Spring'in bir parçası
- Ayarlamalar arasında ortak bir yaklaşım sunar
  - ör, varsayılan, makul ayarlamalar
  - ör, eğer veritabanı olarak embeded H2 veritabanı kullanıyorsanız SpringBoot otomatik olarak başlatacak ve Hibernate ayarlamalarını yapacaktır
- Tam işlevsel enterprise bir uygulama çok hızlı bir şekilde yazılabilir

# Popülerlik

- Spring backend'de enterprise uygulama geliştirmek için tartışmasız en popüler framework'tür
- Ancak, JEE gibi, bir öğrenme eğrisine sahiptir çünkü dependency injection ve Proxy sınıf mantığını kavramanız gerekir
- Java *DropWizard* gibi daha hafif alternatiflere de sahiptir
  - Daha az “sihir”, ancak daha tek düze ve az fonksiyonellik...



# Deployment

- Spring ile yine WAR dosyaları oluşturabilir ve WildFly gibi JEE container'larına deploy edebilirsiniz
  - Çünkü Spring JEE'nin «Servlet» API'ını (HTTP istekleri için kullanılan) kullanır
  - Ancak, en yaygın durum kendi kendine çalışabilen oldukça büyük bir JAR dosyası oluşturulmasıdır (ihtiyaç duyulan bütün bağımlılıkları barındırır)
    - Dahili bir servlet containerı (Tomcat veya Jetty gibi) barındırır
    - Bu ikisi, "Servlet" API'sini destekleyen, ancak JEE'nin diğer özelliklerini desteklemeyen sunuculardır.
-

# JEE'den Spring'e

- Spring'te, JEE'den esinlenilmiş ve pek çok yapı takip edilmiştir
  - Ve aslında JEE'nin pek çok bakış açısı da Spring ile birlikte gelmiştir
- Spring oldukça büyüktür, bu yüzden yalnızca şu ana kadar ki gördüğümüz yapıları (ör: *JPA*, *EJB* ve *JSF*) tartışacağız
- Pek çok benzerlikleri olduğu için birini diğerine aktarmak çok da karmaşık değildir





# JPA

- Spring, JPA'yı doğrudan kullanabilir
  - Varsayılan olarak, Spring JPA için Hibernate kullanır
  - *@Entity* veya *EntityManager*'da herhangi bir değişiklik yapmaya gerek yoktur
  - *persistence.xml*'e ihtiyaç yoktur çünkü Hibernate temel Spring dosyasından ayarları alır (ör., *application.yml*)
  - Spring JPA'da daha fazla fonksiyonellik sağlar (*@Repository* gibi)
-

# EJB

- Spring'te desteklenmez
- Spring kendi eşdeğer bean mekanizmasına sahiptir
- Sınıflar *@Service* ile annotate edilir ve injection *@Autowired* ile yapılır
- Spring servisleri varsayılan olarak *singleton*'dur
- Transaction kullanmak isterseniz, varsayılan değildir ve *@Transactional* annotation'ını kullanmanız gerekir
- Sıklıkla EJB'den Spring Service'e dönüştürme işlemi yalnızca annotation değişikliği ile gerçekleşir

# JSF

---

- Spring tarafından doğrudan desteklenmez
- Spring, Spring MVC adı verilen kendi server-side HTML rendering yapısına sahiptir
- Ancak, JSF (the controller beans and XHTML templates) harici kütüphanelerle (JoinFaces gibi) kullanılabilir
  - Bu Spring bean'leri içinde JSF controller bean'lerinin kullanılabilmesini sağlar ve Spring'in HTTP isteklerinde JSF servlet kullanabilmesi için gerekli ayarlamaları yapar
- Hatırlatma: bugün, gerçekten bir frontend framework/kütüphane'si öğrenmek istiyorsanız React öğrenin. Bu derste JSF kullanıyoruz çünkü hem JEE hem de Spring'te kullanılabilir ve JavaScript olmadan bir web uygulaması nasıl gerçekleştirilebilir bunu inceliyoruz



# JoinFaces

- SpringBoot'ta JSF çalıştırmak için kullanılan kütüphane
  - Web.xml yapılandırmasına gerek yok, bir WAR oluşturmaya veya "src/main/webapp" altında JSF dosyalarına sahip olmaya gerek yok
  - JSF dosyaları "*src/main/resources*" altında diğer statik kaynakların arasında bulunmaktadır
  - Özellikle "**src/main/resources/META-INF/resources**" altındaki bütün dosyalar HTTP asset'leri olarak ele alınacaktır
-

# Test

- Spring'in JEE'yi tamamen yok ettiği nokta burasıdır
- WildFly gibi JEE container'larının indirilmesine ihtiyaç duymaz
  - Spring bağımlılık olarak içe aktarılan bir başka JAR kütüphanesidir
- Bir Spring uygulaması, JVM içindeki testleri ile doğrudan başlatabilir.
  - Anında bir WAR dosyası oluşturmak için Arquillian kullanmaya ve ardından JEE container'ını çalıştıran harici bir işlemi kontrol etmeye gerek yok

# Entry Point

- *@SpringBootApplication* ile annotate edilmiş bir sınıfa ihtiyaç var
  - Bu uygulamanın entry point'i (başlangıç noktası) olacaktır
- SpringBoot hangi *@Service* bean'inin başlatılacağını kontrol etmek için otomatik olarak classpath'deki bütün dosyaları tarar
- Package adları verirken dikkatli ol: varsayılan olarak, eğer *@SpringBootApplication* uygulaması X.Y.Z package'i içindeyse, yalnızca X.Y.Z ve alt paketleri taranır
  - ör, X.Y.Z.W taranırken X.Y taranmaz

# Ayarlamalar

- *SpringBoot*, classpath'de neyin mevcut olduğuna bağlı olarak mantıklı varsayılan yapılandırmalar sağlar
- Eğer değişiklik yapmanız gerekirse, *application.properties* veya *application.yml* dosyalarında gerçekleşir
  - Bunlar aynıdır yalnızca farklı formattadır
  - “.properties”: <property-name>=<value> ikilisi
  - “.yml”/“.yaml”: YAML (YAML Ain't Markup Language)
- Ayarlama listesi için bakınız:
  - <https://docs.spring.io/spring-boot/docs/current/reference/html/common-application-properties.html>

# Git Repository Modülü

- *NOT: açıklamaların büyük bir çoğunluğu kod içerisinde yorum satırı olarak bulunmaktadır, burada slaytlarda bulunmamaktadır*
- **intro/spring/bean/service**
- **intro/spring/bean/jpa**
- **intro/spring/bean/repository**
- **intro/spring/bean/profile**
- **intro/spring/bean/configuration**
- **intro/spring/bean/async**
- **intro/spring/jsf**
- Ders 08 alıştırması (dokümantasyona bakınız)