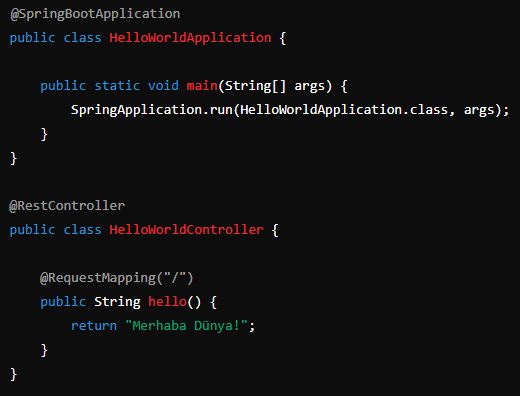
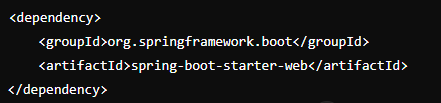
1)Java diline ait frameworkler gün geçtikçe artıyor. Yeni problemler oluştukça yeni çözümler de üretiliyor. Şimdilik en yaygın kullanılan Java Frameworkleri ve çözdükleri problemler şunlar:

* **Spring**: Geniş çapta güvenilir, düşük maliyetli ve esnek bir framework olarak kabul edilir. Kodlama verimliliğini artırır ve sistem kaynaklarını etkin bir şekilde kullanarak genel uygulama geliştirme süresini azaltır. Spring, sıkıcı yapılandırma işlerini ortadan kaldırır, böylece geliştiriciler iş mantığını yazmaya odaklanabilirler. Ayrıca, Spring altyapıyı ele alarak geliştiricilerin dağıtım ortamları ve diğer altyapısal konularla ilgilenmeden yüksek kaliteli, kullanışlı ve ölçeklenebilir kurumsal uygulamalar geliştirmesine olanak tanır.

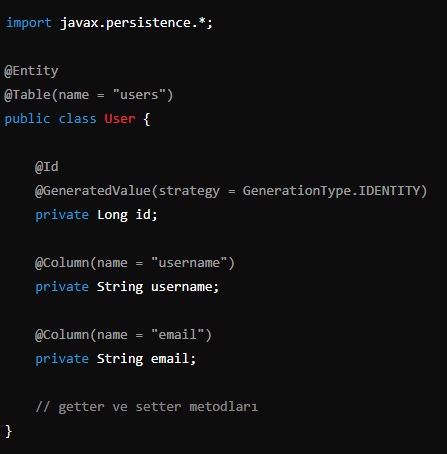


Merhaba Dünya! Yazdıran bir Spring Boot uygulaması



Bağımlılıkları yazdığımız pom.xml dosyası

* **Hibernate**: Tartışmalara konu olan bir framework olup olmadığı konusunda çeşitli görüşler bulunmaktadır. Temel olarak Hibernate, bir ORM (Object-Relational Mapping) kütüphanesidir. Bu kütüphane, Java sınıflarını veritabanı tablolarına dönüştürerek ve Object Oriented programlama modellerine uygun şekilde veritabanıyla ilişki kurarak, veritabanı işlemlerini kolaylaştırır ve kurulu yapıyı güçlendirir. Kullanıcıların veri sorgulama ve veri çekme işlemlerini Hibernate otomatik olarak gerçekleştirir. Dolayısıyla, Hibernate'i Java nesnelerini veritabanı tablolarına dönüştüren ve veritabanı işlemlerini kolaylaştıran bir ORM kütüphanesi olarak tanımlayabiliriz.

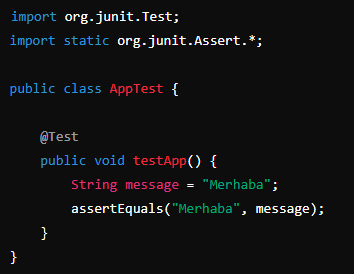


User adında bi entity sınıfı



Veritabanı işlemleri

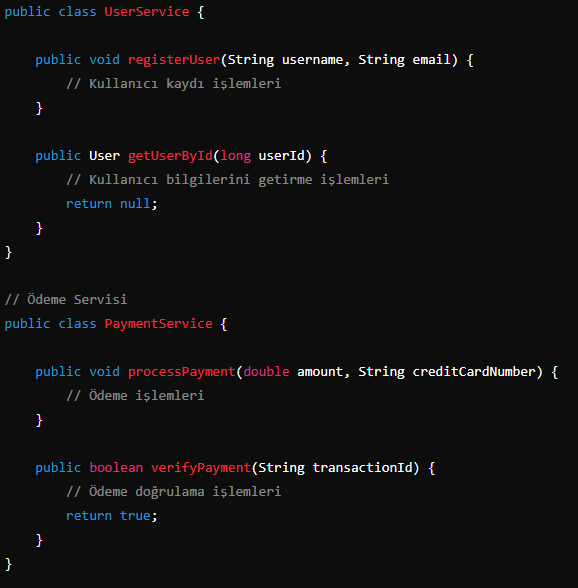
* **Maven**, projelerde standart oluşturmayı, geliştirme sürecini basitleştirmeyi, belgeleri etkin bir şekilde oluşturmayı ve kütüphane bağımlılıklarını ile IDE bağımlılıklarını ortadan kaldırmayı sağlayan bir araçtır. IDE bağımlılığı olmadığı için farklı IDE'lerde de kullanılabilen bir standart oluşturur. JAR dosyalarını tek tek elle eklemek zorunda kalmadan, bu dosyaların proje tarafından yönetilmesini sağlayarak bu işlemleri kolaylaştırır. Bu yönetim işlemi pom.xml dosyası üzerinden gerçekleştirilir. Build sürecini tektipleştirir.



Maven ile bir JUnit test örneği

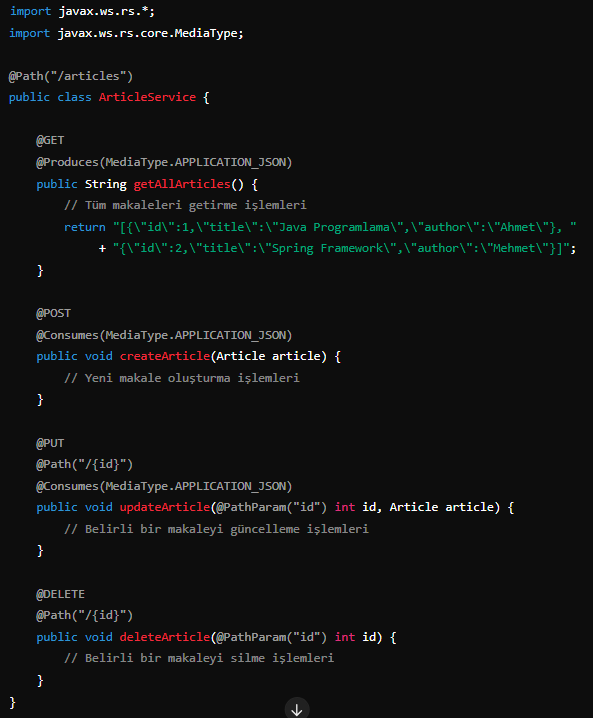
2) **Service-Oriented Architecture (SOA)**: SOA, bir yazılım mimarisi yaklaşımıdır ve hizmet odaklı bir yapı sağlar. Bu yaklaşım, hizmetleri modüler parçalara ayırır ve bu hizmetlerin birbirleriyle iletişim kurabilecekleri standart arayüzler üzerinden entegre olmasını sağlar.

SOA kullanarak bir e-ticaret platformunda farklı servislerin nasıl oluşturulabileceğine bir örnek:



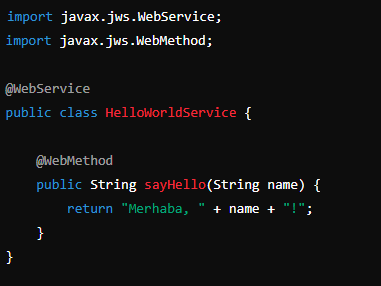
**RESTful Service**: REST (Representational State Transfer), HTTP protokolü üzerinden kaynaklara erişim sağlamak için bir mimari stildir. RESTful Servisler, kaynakları (resource) temsil eden URI'lar (Uniform Resource Identifier) üzerinden erişir ve HTTP metodlarını kullanarak bu kaynaklarla etkileşim sağlar.

Örnek: Makale oluşturma silme görme ve güncelleme



**Web Services**: Web Servisleri, farklı sistemler arasında iletişim sağlamak için kullanılır. SOAP (Simple Object Access Protocol) veya REST (Representational State Transfer) tabanlı Web Servisler kullanılabilir. Bir SOAP Web Servisi, XML tabanlı mesajlarla iletişim sağlar ve genellikle WSDL (Web Services Description Language) ile tanımlanır.

SOAP tabanlı bir Web Servis örneği:



**HTTP Metodları**: HTTP metodları, bir web sunucusuna gönderilen isteklerin türlerini belirtir. En yaygın kullanılan HTTP metotları şunlardır:

* GET: Bir kaynağın alınmasını sağlar.
* POST: Bir kaynağa yeni bir veri ekler.
* PUT: Bir kaynağın tamamen güncellenmesini sağlar.
* DELETE: Bir kaynağı siler.

Örnek:

