

JAVA FRAMEWORS

Java dünyasında onlarca framework vardır. Ben bunlarda dört tanesini açıklayacağım ve çözdükleri problemleri anlatacağım. Açıklayacağım frameworkler şunlardır; Spring ve Hibernate.

SPRING

Spring, Java ve .NET için yazılmış açık kaynak kodlu bir frameworktür. Spring, bizlere birçok kütüphane ve modülleri kullanıma sunarak entegrasyon işlemini kolaylaştırmayı amaçlamaktadır. Spring, bizleri ağır bir iş yükünden kurtararak daha kolay kullanım sağlamaktadır.

Spring içerisinde, gruplar halinde bulunan 20' ye yakın modülden oluşmaktadır. Aşağıda bu gruplar ve modülleri verilmiştir.

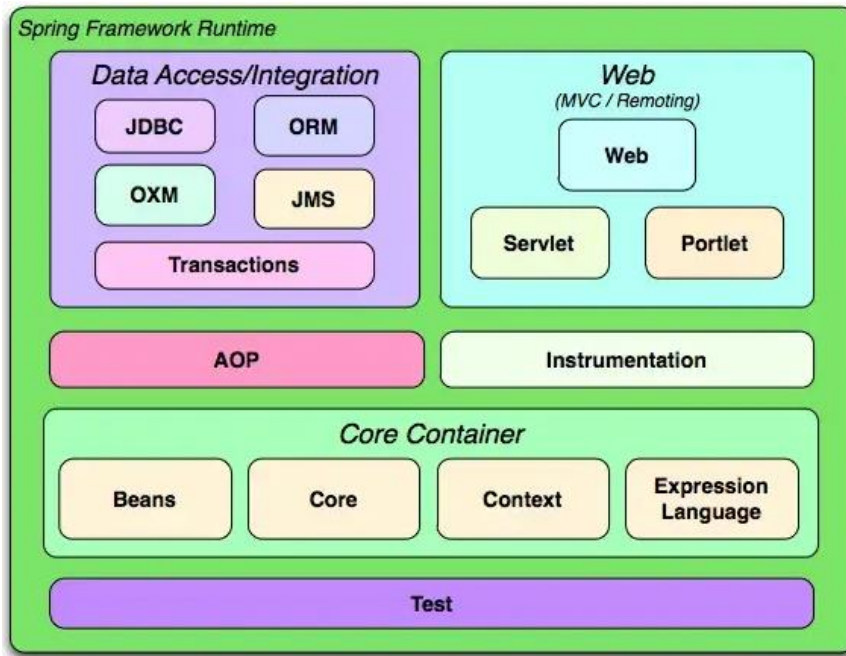


Figure 1 [1]

Spring Core Container

Beans, Core, Context, Expression Language olmak üzere 4 modülden oluştuğunu yukarıdaki görselde görmekteyiz.

- 1- Spring Core: Burada uygulamanın kontrolünü başka bir kod parçasına yani Spring Frameworke devretmiş oluyoruz. Konfigürasyonlar sayesinde springe uygulamanın mantığını tanımlayan yazdığımız kodun nasıl yönetileceği hakkında talimat veriyoruz.
- 2- Bean: Beans modülü tarafından sağlanan factory pattern'in karmaşık bir uygulaması olarak bilinmektedir.

- 3- Context: Kurulmuş ve yapılandırılmış bir nesneye erişim için araç sağlar. Core ve Beans modülleri tarafından sağlanan sağlam temel üzerine kuruludur. ApplicationContext interface ise Context modülünün odak noktasıdır.
- 4- Expression Language: Bir nesne grafiğini gerçek zamanlı olarak aramak ve değiştirmek için kullanılmaktadır.

Spring Web

Spring web ile genel olarak web uygulamaları yazılmaktadır. Günümüzde birçok web sayfası Spring MVC kullanmaktadır. Genel olarak web katmanı şunlardan oluşur; Web, Web-MVC, Web-Socket...

Spring Data

Spring data ile veritabanı projeleri yazabiliriz ve az sayıda kod satırı ile persistence katmanını kullanabiliriz. Veritabanıyla ilgili proje yazmak istiyorsak, bu proje hem SQL hem de NoSQL ile yazılabilmektedir.

Spring Boot

Bir frameworkün tüm konfigürasyonlarını kod ile ayarlamak yerine Spring Boot'un size gerektiği gibi özelleştirebileceğiniz varsayılan bir konfigürasyon sunmasıdır. Genel olarak, bilinen kurallara uyduğunuz için daha az kod yazmanız ve uygulamanızın diğerlerinden birkaç veya küçük açılardan farklılık göstermesidir. Bu nedenle, her uygulama için tüm yapılandırmaları yazmak yerine, varsayılan bir yapılandırmayla başlamak ve yalnızca kuraldan farklı olanı değiştirmek daha kolay ve verimli bir süreçtir.

Spring Framework'ün, önceden tanımlanmış templateler, kolay test aşaması, hızlı kod yazma süreci gibi avantajları bulunmaktadır.

Şimdi de Spring ile yapılmış basit bir uygulamayı sizlerle paylaşacağım.

Öncelikle yeni bir java projesi oluşturalım. Ardından HelloWorld adlı bir classı oluşturup içeriğini görseldeki gibi dolduralım.

```
package com.tutorialspoint;

public class HelloWorld {
    private String message;

    public void setMessage(String message){
        this.message = message;
    }
    public void getMessage(){
        System.out.println("Your Message : " + message);
    }
}
```

Main classı da görseldeki gibi dolduralım.

```
package com.tutorialspoint;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {
    public static void main(String[] args) {
        ApplicationContext context = new ClassPathXmlApplicationContext("Beans.xml");
        HelloWorld obj = (HelloWorld) context.getBean("helloWorld");
        obj.getMessage();
    }
}
```

ClassPathXmlApplicationContext() API'si bean yapılandırma dosyasını yükler ve sonunda sağlanan API'ye dayalı olarak, tüm nesneleri, yani yapılandırma dosyasında bahsedilen bean'leri oluşturmayı ve başlatmayı işini yapar.

getBean() metodu kullanarak en sonunda gerçek nesneye dönüştürülebilen genel bir nesneyi döndürmek için kullanır.

Ardından bean yapılandırma dosyası oluştururuz.

```
<?xml version = "1.0" encoding = "UTF-8"?>

<beans xmlns = "http://www.springframework.org/schema/beans"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

    <bean id = "helloWorld" class = "com.tutorialspoint.HelloWorld">
        <property name = "message" value = "Hello World!"/>
    </bean>

</beans>
```

<bean> etiketinde tanımlandığı gibi benzersiz bir kimlik atar. Nesne oluşturma sırasında kullanılan farklı değişkenlerin değerlerini iletmek için **<property>** etiketini kullanırız.

Son olarak projeyi çalıştıralım ve çıktı görselini aşağıda yer verdim.

```
Your Message : Hello World!
```

Spring' in öne çıkan özellikleri ve Spring' i neden tercih etmeliyim?

Kontrolün tersine çevrilmesi (IoC) – Nesne eşleştirme, derleme zamanından ziyade çalışma zamanında yapılması ve Bağımlılık Enjeksiyonu – IoC ile birlikte çalışır. Nesnelerin bağımlılığını sağlayarak gevşek bağlantıya izin verir.

HIBERNATE

Hibernate, JDBC de önemli bir kavram olan nesne düzeyindeki ilişkilerde sorunlarla karşılaşılmasının ardından bu sorunları çözerek Java ile veritabanlarının işlenmesini kolaylaştıran bir Nesne İlişkisel Eşlemedir (ORM). Nesne yönelimli deyimli takip eden kalıcı sınıflar geliştirmenize izin verir.

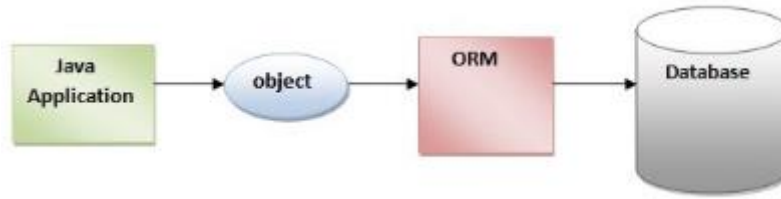


Figure 2 [2]

Kodun gevşek bir şekilde birleştirilmesi için bir soyutlama katmanı oluşturarak veritabanı tablolarına doğrudan karşılık gelen Java sınıflarıyla eşleşme sağlar. Hibernate ile, geliştiricilerin veritabanı bağlantıları ile ilgili işlemlerle uğraşmaları veya çıkan sorunları çözmekle zaman harcamaları gerekmez, bu işlemler Hibernate'in kendisi tarafından halledilir. hibernate.cfg.xml dosyası, veritabanı yapılandırması ve eşlemesi hakkında bizlere bilgi vermektedir.

Avantajları; sürdürülebilirlik, ücretsiz ve açık kaynaklı, JDBC API'sinden çok sayıda tekrar eden kodu kaldırır.

GOOGLE WEB TOOLKIT (GWT)

Google Web Toolkit (GWT), yazılım geliştiricilerin client tarafı Java kodu yazmalarına ve bunu JavaScript olarak oluşturmalarına yardımcı olan tamamen ücretsiz, açık kaynaklıdır. AdSense, Google Cüzdan ve Blogger gibi birçok Google ürünü GWT kullanılarak yazılır. Yazılım geliştiriciler, Google Web Toolkit kullanarak karmaşık tarayıcı uygulamalarını hızlı bir şekilde kodlayabilir. Google Web Toolkit Java'da Ajax uygulamaları geliştirmenize ve hata ayıklamanıza olanak tanır. Google Web Toolkit ile ilgili harika olan şey, JavaScript optimizasyonu veya duyarlı tasarım gibi ön uç teknolojilerde uzman olmadan karmaşık tarayıcı tabanlı uygulamalar yazabilmemizdir ki bu gerçekten çok iyi bir özelliktir.

Google Web Toolkit, bizlere Google API lerinin kullanımını sunar ve karmaşık JS uygulamalarının oluşturulup, çalıştırılmasına olanak sağlamaktadır.

Google Web Toolkit, yaygın web geliştirme görevleri için yeniden kullanılabilirliği açısından yazılım geliştiricilerine avantaj sağlamaktadır. Ayrıca Google API' leri Google Web Toolkit uygulamalarında kullanabiliriz.

KAYNAKÇA

1. <https://buseodaci.medium.com/spring-framework-nedir-fe46c9ce3456>
2. <https://www.javatpoint.com/hibernate-tutorial>