



# DefineX Java Spring Boot Bootcamp

## HW-1

İlker KUŞ

# İÇİNDEKİLER

1. Java dünyasındaki Framework'ler ve çözdükleri problemler nedir? Kod Örneklerini de içermelidir. ....	3
1) Spring Framework: .....	3
2) Hibernate:.....	3
3) Apache Struts: .....	4
4) Apache Wicket:.....	4
5) Google Web Toolkit (GWT): .....	5
6) Dropwizard: .....	5
7) Play Framework:.....	5
8) Vaadin:.....	6
9) Blade:.....	6
10) Grails:.....	6
2. SOA - Web Service - Restful Service - HTTP methods kavramlarını örneklerle açıklayınız.....	8
1) Service Oriented Architecture (SOA):.....	8
2) Web Service:.....	8
3) RESTful Service: .....	9
4) HTTP Methods:.....	9

## 1. Java dünyasındaki Framework'ler ve çözdükleri problemler nedir? Kod Örneklerini de içermelidir. (5 Puan)

Java dünyasında birçok Framework bulunmaktadır ve her biri belirli bir problemi çözmek ya da belirli bir konuda geliştiriciye yardımcı olmak için tasarlanmıştır. Araştırmalarım sonucunda en çok kullanılan **Java Web Framework'leri** ile ilgili bilgi vermeye çalışacağım. Farklı kaynaklardan yola çıkarak 10 adet Java Web Framework seçimi yaptım. Bu Frameworkler ile ilgili kısa bir açıklama ile birlikte sektörel kullanımları ve basit bir kod örneği eklemesi yaptım.

### 1) Spring Framework:

- Ne işe yarar: Spring, Java tabanlı uygulamaların geliştirilmesi için kapsamlı bir framework'tür. Dependency Injection, Aspect-Oriented Programming, transaction management gibi birçok modülü içerir.
- Sektördeki kullanımı: Spring, birçok endüstride geniş bir kullanıma sahiptir, özellikle kurumsal uygulamalar ve büyük ölçekli projelerde sıkça tercih edilir.
- Kod örneği:

```
@RestController
public class HelloController {
    @GetMapping("/hello")
    public String hello() {
        return "Hello, World!";
    }
}
```

### 2) Hibernate:

- Ne işe yarar: Hibernate, Java tabanlı veritabanı işlemleri için kullanılan bir ORM (Object-Relational Mapping) framework'tür. Veritabanı işlemlerini doğrudan Java nesneleri üzerinden gerçekleştirmenizi sağlar.
- Sektördeki kullanımı: Hibernate, özellikle büyük ve karmaşık veritabanı işlemleri gerektiren projelerde sıkça tercih edilir.
- Kod örneği:

```
@Entity
@Table(name = "users")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String username;
    private String email;
}
```

### 3) Apache Struts:

- Ne işe yarar: Struts, Model-View-Controller (MVC) mimarisini destekleyen bir web uygulama framework'üdür. Web uygulamalarını modüler hale getirir ve ayrıntılı konfigürasyon seçenekleri sunar.
- Sektördeki kullanımı: Struts, özellikle eski ve büyük ölçekli kurumsal projelerde hala yaygın olarak kullanılmaktadır.
- Kod örneği:

```
public class HelloWorldAction extends ActionSupport {
    private String message;

    public String execute() throws Exception {
        setMessage("Hello, World!");
        return SUCCESS;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }
}
```

### 4) Apache Wicket:

- Ne işe yarar: Wicket, component-based web application development (bileşen tabanlı web uygulama geliştirme) için kullanılan bir framework'tür. Java kodunu HTML ile birleştirir ve web uygulamalarını oluşturmayı kolaylaştırır.
- Sektördeki kullanımı: Wicket, büyük ölçekli ve daha karmaşık kullanıcı arayüzlerine sahip web uygulamaları geliştirmek için tercih edilir.
- Kod örneği:

```
public class HomePage extends WebPage {
    public HomePage() {
        add(new Label("message", "Hello, World!"));
    }
}
```

## 5) Google Web Toolkit (GWT):

- Ne işe yarar: GWT (Google Web Toolkit), web uygulamaları geliştirmek için kullanılan bir framework'tür. Java kodunu JavaScript'e dönüştürerek, web uygulamalarının geliştirilmesini kolaylaştırır.
- Sektördeki kullanımı: GWT, özellikle Google teknolojileriyle entegre projelerde ve büyük ölçekli web uygulamalarında sıkça kullanılır.
- Kod örneği:

```
public class HelloWorld implements EntryPoint {  
    public void onModuleLoad() {  
        RootPanel.get().add(new Label("Hello, World!"));  
    }  
}
```

## 6) Dropwizard:

- Ne işe yarar: Dropwizard, hızlı ve hafif RESTful web servisleri geliştirmek için kullanılan bir framework'tür. İçinde Jersey, Jackson, Metrics gibi popüler kütüphaneleri barındırır.
- Sektördeki kullanımı: Dropwizard, özellikle mikro-servis mimarisiyle çalışan ve hafif RESTful servislere ihtiyaç duyan projelerde tercih edilir.
- Kod örneği:

```
@Path("/hello")  
public class HelloWorldResource {  
    @GET  
    @Produces(MediaType.TEXT_PLAIN)  
    public String hello() {  
        return "Hello, World!";  
    }  
}
```

## 7) Play Framework:

- Ne işe yarar: Play Framework, hızlı ve ölçeklenebilir web uygulamaları geliştirmek için kullanılan bir framework'tür. Reactive ve asenkron programlama modellerini destekler.
- Sektördeki kullanımı: Play Framework, özellikle modern ve hızlı web uygulamaları geliştirmek için tercih edilir.
- Kod örneği:

```
public class HomeController extends Controller {  
    public Result index() {  
        return ok("Hello, World!");  
    }  
}
```

## 8) Vaadin:

- Ne işe yarar: Vaadin, Java tabanlı web uygulamaları geliştirmek için kullanılan bir framework'tür. Server-side Java kodu ile zengin kullanıcı arayüzleri oluşturmayı sağlar.
- Sektördeki kullanımı: Vaadin, özellikle şirket içi uygulamalar ve enterprise projelerde tercih edilir.
- Kod örneği:

```
@Route("")
public class MainView extends VerticalLayout {
    public MainView() {
        add(new Text("Hello, World!"));
    }
}
```

## 9) Blade:

- Ne işe yarar: Blade, hafif ve basit bir Java web framework'tür. Minimal yapıya sahiptir ve hızlı web uygulamaları geliştirmeyi sağlar.
- Sektördeki kullanımı: Blade, küçük ve orta ölçekli projelerde hızlı prototip oluşturmak için tercih edilir.
- Kod örneği:

```
@GetRoute("/hello")
public void hello(Context context) {
    context.html("Hello, World!");
}
```

## 10) Grails:

- Ne işe yarar: Grails, Groovy programlama dilini kullanarak Java tabanlı web uygulamaları geliştirmek için kullanılan bir framework'tür. Hızlı uygulama geliştirme ve Spring ile entegrasyonu kolaylaştırır.
- Sektördeki kullanımı: Grails, özellikle hızlı prototipler oluşturmak ve dinamik web uygulamaları geliştirmek için tercih edilir.
- Kod örneği:

```
class HelloController {
    def index() {
        render "Hello, World!"
    }
}
```

Peki , hangi framework hangi projede kullanılmalı sorusuna gelirse,

- Büyük ölçekli, kurumsal ve karmaşık projeler için Spring Framework ve Hibernate gibi kapsamlı ve güçlü framework'ler tercih edilir. Bu tür projelerde genellikle güvenilirlik, performans ve geniş işlevsellik ön plandadır.
- Hızlı prototipler oluşturmak veya küçük ve orta ölçekli projelerde hafiflik ve basitlik önemliyse, Blade veya Grails gibi daha hafif ve minimal framework'ler tercih edilebilir.
- Modern ve hızlı web uygulamaları geliştirmek için Play Framework veya Vaadin gibi framework'ler tercih edilirken, RESTful servisler geliştirmek için Dropwizard gibi spesifik çözümler tercih edilebilir.
- Her proje farklı gereksinimlere sahip olduğu için, tercih edilen framework projenin ihtiyaçlarına ve gereksinimlerine en iyi şekilde uyacak şekilde seçilmelidir.

## 2. SOA - Web Service - Restful Service - HTTP methods kavramlarını örneklerle açıklayınız.(15 Puan)

### 1) Service Oriented Architecture (SOA):

Hizmet odaklı mimari (SOA), iş uygulamaları oluşturmak için hizmetler adı verilen yazılım bileşenlerini kullanan bir yazılım geliştirme yöntemidir. Her hizmet, bir iş özelliği sunar ve hizmetler, farklı platform ve diller arasında birbiriyle iletişim de kurabilir. Geliştiriciler, hizmetleri farklı sistemlerde yeniden kullanmak veya birkaç bağımsız hizmeti karmaşık görevleri gerçekleştirmek amacıyla birleştirmek için hizmet odaklı mimariyi kullanır.

Örneğin, bir e-ticaret platformu düşünelim. Bu platformda müşteri yönetimi, sipariş işleme, envanter yönetimi gibi farklı işlevler bulunmaktadır. SOA mimarisi kullanılarak, her bir işlev ayrı bir hizmet olarak tasarlanır. Bu şekilde, sipariş işleme hizmeti, envanter yönetimi hizmetiyle iletişim kurarak bir ürünün stok durumunu kontrol edebilir.

Service Oriented Architecture Bazı Faydaları şunlardır:

- Geliştiriciler, zamandan ve maliyetten tasarruf etmek için hizmetleri farklı iş süreçlerinde yeniden kullanır. Hizmet odaklı mimari sayesinde, uygulamaları kod yazmaktan ve entegrasyonları sıfırdan yapmaktan daha hızlı şekilde bir araya getirebilirler.
- Hizmet odaklı mimarideki herhangi bir hizmeti değiştirmek, iş sürecinin genel işlevselliğini etkilemez.
- Hizmet odaklı mimari, teknolojideki gelişmelere daha uyumludur. Uygulamalarınızı verimli ve uygun maliyetli şekilde modernize edebilirsiniz.

### 2) Web Service:

Web servisleri, farklı ağlardaki uygulamaların birbirleriyle haberleşebilmesi için standart protokoller kullanarak sunulan hizmetlerdir. XML veya JSON gibi formatlar aracılığıyla veri alışverişi yapılır. Web servisleri, SOA mimarisinin bir uygulamasıdır.

Örneğin, bir hava durumu web servisi düşünün. Bu servis, belirli bir şehrin güncel hava durumu bilgisini XML veya JSON formatında sağlar. Bir mobil uygulama, bu web servisine HTTP istekleri göndererek hava durumu bilgisini alabilir ve kullanıcılarına sunabilir.



### 3) RESTful Service:

RESTful servisler, REST (Representational State Transfer) mimarisine dayalı web servisleridir. REST, web servislerinin nasıl tasarlanması gerektiğine dair bir dizi kural ve kısıtlama sunar. RESTful servisler, HTTP protokolünü kullanarak kaynaklara erişim sağlar ve genellikle JSON formatında veri alışverişi yapar.

Örneğin, bir kitaplık yönetim sistemi düşünün. Bu sistemde kitaplar, RESTful servis aracılığıyla yönetilebilir. HTTP metodları kullanılarak farklı işlemler yapılır:

- **GET /books** - Tüm kitapların listesini alır.
- **POST /books** - Yeni bir kitap ekler.
- **PUT /books/{id}** - Belirli bir kitabın bilgilerini günceller.
- **DELETE /books/{id}** - Belirli bir kitabı siler.

### 4) HTTP Methods:

RESTful servislerde kaynaklarla etkileşimde bulunmak için kullanılan HTTP metodlarıdır. En yaygın kullanılan HTTP metodları şunlardır:

- **GET:** Bir kaynağın temsilini almak için kullanılır.
- **POST:** Yeni bir kaynak oluşturmak için kullanılır.
- **PUT:** Bir kaynağın tamamını güncellemek için kullanılır.
- **DELETE:** Bir kaynağı silmek için kullanılır.
- **PATCH:** Bir kaynağın bir kısmını güncellemek için kullanılır.

Örneğin, bir kullanıcı yönetim sistemi düşünün. Bu sistemde kullanıcılarla ilgili işlemler HTTP metodları aracılığıyla yapılır:

- **GET /users** - Tüm kullanıcıların listesini alır.
- **POST /users** - Yeni bir kullanıcı ekler (kullanıcı bilgileri ile birlikte).
- **PUT /users/{id}** - Belirli bir kullanıcının tüm bilgilerini günceller.
- **DELETE /users/{id}** - Belirli bir kullanıcıyı siler.