

1-MVC kavramını açıklar mısınız ? Neden ihtiyaç duyuluyor. Java'da nasıl kurgulanıyor.

Object Oriented katmanları nelerdir ?

MVC'nin Model View Controller olmak üzere üç katmandan oluşur. Model katmanı verilerin işlendiği, işin yürütüldüğü katmandır. View katmanı kullanıcı ara yüzünün olduğu katmandır, frontend'dir. Controller katmanı view dan gelen istekleri karşılayan ve view katmanına geri dönüş yapan katmandır.

MVC karmaşıklığı azaltıp projenin sürdürülebilir olmasını sağlar, kullanıcıdan bağımsız test edilmesine olanak sağlar ve seo açısından avantajlıdır.

Java'da mvc Spring MVC ile kurgulanır model katmanı rest apiyle birbirlerine benzer controllerlarda @Controller anotasyonu kullanılır ve html sayfalar templates dosyasında bulunur.Server-side templating engine olarak Tymeleaf kullanır.

Rest API 'de ise controller katmanında @RestController anotasyonu kullanılır view katmanı tamamen backend uygulamasından ayrılır.

Object Oriented katmanları

Abstraction:

Soyutlamadır. Abstract classlardan oluşur. Amaç ortak özelliği olan nesneleri tek bir çatı altına almaktır.

Encapsulation:

Public private ve protected access modifierlarını kullanarak metotlarımıza erişim engeli koymuş oluyoruz.

Public her class'a, protected yalnızca packagelar içindeki classlara, private yalnızca kendi clasına erişime açıktır.

Inheritance:

Kalıtımdır. Super clasın sub classlara mirasını yani metot ve değişkenlerini aktarmasıdır

Polymorphism:

Çok biçimlilik. Super classın referanslarının, subclasstan türeyen nesneri tutabilmesidir.Sub classlar super classın statik ve final metotları hariç tüm metotlarına sahiptirler.

2- Birbirinden bağımsız iki platformun birbiriyle haberleşmesi nasıl sağlanabilir. Örneğin, X platformu Java ile yazılmış olsun, Y platform u C# ile. Bu iki platformun bir biri ile iletişim halinde request-response ilişkisi kurması gerekiyor. Bu yapıyı nasıl sağlarız.

Java’da iletişimi sağlamak için RestTemplate sınıfını kullanabiliriz. RestTemplate ile Get, Post, Delete ve Update metodlarını kullanabiliriz, eğer bir callback beklenmiyor ise RabbitMq gibi messaging uygulamalarını kullanabiliriz hem hız açısında hem de veri kaybı güvenliği açısından daha avantajlıdır.

3-Bir web sayfasında ekran sürekli Backend’ den veya bir başka yapı tarafından güncelleniyor. Siz, web sayfasını refresh etmeden bu güncel bilgiyi anlık ekrana nasıl yansıtırsınız.

Eğer Angular,React veya Vue.js gibi frameworkler kullanıyorsak backend’den gelen veri değiştiğinde değişiklik component bazlı olacağından sadece değişen veri değişir sayfa yenilenmeden yeni gelen veri gösterilmiş olur . framework kullanmıyorsak gelen veriyi \$('#id').html() sürekli güncellememiz gerekir bu gibi durumlarda frameworkler daha avantajlıdır.

5-Firmada çalışan için sana remote bir linux server verildi. Elinde ip adresi port bilgisi kullanıcı adı ve şifren var. Server a erişimi nasıl test edersin, Server a nasıl erişirsin, Server a nasıl dosya atarsın, Serverdan nasıl dosya çekersin.

Server’a erişimi ping [ip adresi] ile erişimi test ederim

Bunun için şuan kullandığım FileZilla’yı kullanabilirim. Port bilgisi, ip adresi, kullanıcı adı, şifreyi ve protocol bilgisini ftp olarak girdikten sonra Server’a bağlanmış olurum. Ya da vs code gibi idelerde ftp extension(ftp-simple vb.) kullanarak bağlanabilirim.

7-Apache Solr servisine yazılacak bir query örneği Apache Solr kullanılan sql programlarından daha farklı runtime bir database. Solr a hali hazırda kayıtlı bir alan olduğunu düşünelim. Alanın ismi “updatedAt” long tipinde tutulan bir alan. Ben 2020 Ocak ayından sonraki verileri getir dediğimde solr a nasıl bir query yazılmalı. http://example?query=kısmını nasıl doldurmaliyim?

Jan 31 2022 23:59:59 GMT+0000 =1643673599

http://example?query=time=[1643673599%20TO%*]

