

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

Cevap:

Model-View-Controller(MVC) yazılım mühendisliğinde kullanılan bir yazılım mimarisidir. 3 katmandan oluşur. Bu katmanlar bir birinden bağımsız olarak çalışmaktadır.

Projede tüm katmanları değiştirmek yerine tek bir katmanı değiştirerek farklı kullanıcı arayüzüne entegre edebiliriz. Projelerde projenin yönetim ve kontrolü daha rahat olduğu için tercih edilmektedir.

Java'da MVC kurgusu:

Model kısmında sınıflar ve nesneler şeklinde temsil edilir. Veritabanı işlemleri veya uygulama veri yapısı bu sınıflarda olabilir.

View kısmında Kullanıcı arayüzü genellikle Swing, JavaFX veya JSP gibi teknolojilerle oluşturulur.

Controller kısmında Servlet, Spring MVC gibi frameworkler veya kendi yazdığımız sınıflar aracılığıyla gerçekleştirilir.

OOP Katmanları:

Data Access(Veri erişim): Veritabanı veya dış kaynaklara erişimi yönetir.

Business Logic(İş Mantığı): İş kuralları ve iş mantığı bulunur. Verilerin işlenmesi, doğrulanması ve iş kurallarına uygun işlenmesi sağlanır.

User Interface(Kullanıcı Arayüzü): kullanıcı arayüzünün oluşturulduğu ve sunulduğu yerdir.

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.

Cevap:

Restful API veya SOAP gibi web servisleri kullanabilirler. İki platformda kendi dillerine uygun API'leri kullanarak http üzerinden istek gönderebilir ve alabilirler.

Veri alışverişinde JSON veya XML evrensel veri formatlarıyla veri alışverişi yapılabilirler. Verileri okuyarak işleyebilir.

Veritabanı entegrasyonu ile olabilir. İki platformun da erişebildiği bir veritabanı kullanarak iletişim kurulabilir. Bir uygulama veri yazarken diğer uygulama veritabanından verileri okuyabilir.

RPC protokolü kullanarak iletişim sağlanabilir. Bir uygulama, diğer uygulamadaki metodu çağırabilir ve sonucu alabilir.

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

Cevap:

WebSockets veya Server-Sent Events(SSE) gibi teknolojileri kullanabilinir. Bu teknolojiler sunucu ile gerçek zamanlı iletişim kurulmasını sağlarlar.

-WebSockets: Sunucu ile tarayıcı arasında çift yönlü iletişim kuran bir iletişim protokolüdür.

-Server-Sent Events(SSE): Sunucudan tek yönlü, devamlı ve anlık veri iletimi sağlayan mekanizmadır.

Sunucu tarafından belirli aralıklarla veya anlık olarak gönderilen verileri web sayfasına aktarır. Sunucu tarafında uygun SSE destekleyen yapı gereklidir.

4-Bir for döngüsü ile aşağıdaki çıktıyı yazar mısınız?

```
*
**
****
*****
*****
*****
*****
```

Cevap:

```
For ( int i = 1; i <=10; i++)
{
    if ( i == 1)
    {
        System.out.println("*");
    }
    Else if( i % 2 == 0)
    {
        For( int j = 1; j <= i; j++)
        {
            System.out.print("*");
        }
        System.out.println();
    }
}
```

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?

Cevap:

Terminal veya komut istemcisini kullanarak, verilen IP adresi ve port bilgisi ile sunucuya erişimi test ederim.

Örneğin:

telnet <IP_ADRESİ> <PORT>

Sunucuya Erişim:

SSH (Secure Shell) protokolünü kullanarak sunucuya erişim sağlarım.

Örneğin:

ssh kullanıcı_adi@IP_ADRESİ -p PORT

Dosya Aktarımı:

Dosya aktarımı için scp (Secure Copy) veya rsync gibi araçları kullanabilirim.

Sunucuya dosya gönderme örneği (scp):

scp local_file kullanıcı_adi@IP_ADRESİ:/remote/path/

Sunucudan dosya çekme örneği (scp):

scp kullanıcı_adi@IP_ADRESİ:/remote/path/remote_file local_path/

Dosya Çekme (wget kullanarak):

Eğer uzak bir dosyayı HTTP veya FTP üzerinden çekmek istiyorsam, wget kullanabilirim.

Örneğin:

wget http://example.com/remote_file

Dosya Çekme (curl kullanarak):

Alternatif olarak, curl kullanarak da dosya çekebilirim.

Örneğin:

curl -O http://example.com/remote_file

Bu adımlar, sunucuya erişimi test etmek, SSH ile bağlantı kurmak, dosya aktarımı yapmak ve uzaktan dosya çekmek için kullanabilirim.

6-Local database kurulumu(mysql,postgresql veya herhangi bir database)

-Java spring uygulaması ayağa kaldırılması,

-İki adet tablo yer almalı ve bu tabloların birbiriyle bağlı olmalıdır.(Örn: şirket ve çalışan gibi),

-Java spring uygulamasında ekleme, silme, güncelleme, listeleme gibi servisler yer almalıdır ve responseda yapılan işlen detayı return edilmelidir.

-Ekleme silme, güncelleme, listeleme işlemleri postman vb ile işlem yapılabilmelidir.

Bu adımlar sırasıyla izlenip java uygulaması üzerinden database' e kayıt atılmalı (Herhangi bir kayıt olabilir fark etmez. Database'de bir tablo açılıp o tabloya değer girilmesi java isteği üzerinden). Daha sonra aynı istek atılan uygulama ile (örnek postman ...) get ve list java spring endpointleri çağırılarak, database e atılan kayıt response olarak dönülmeli.

MVC deki model ve kontroller akışının güzel kurgulanması ve uygulama ayağı nasıl kaldırılıp servislerin nasıl kullanıldığına dair bir döküman hazırlanmalıdır. Bu proje için kaynak kodu ile iletilmesi gerekmektedir.

Cevap:

https://github.com/kadirkenan/backend_challenge

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 doldurmalıyım?

Cevap:

[http://example/select?q=updatedAt:\[2020-01-01T00:00:00Z TO *\]&wt=json&indent=true](http://example/select?q=updatedAt:[2020-01-01T00:00:00Z TO *]&wt=json&indent=true)

Bu sorgu şu şekilde açıklanabilir:

q: Sorgu alanı.

updatedAt:[2020-01-01T00:00:00Z TO *]: "updatedAt" alanının 2020 Ocak 1 tarihinden sonraki tarih aralığını içeren belgeleri getirir.

wt=json: Yanıtın JSON formatında olmasını sağlar.

indent=true: Yanıtın düzenli bir şekilde gösterilmesini sağlar.

Bu sorgu, "updatedAt" alanı içinde 2020 Ocak 1 tarihinden sonraki belgeleri getirecektir.

