



MEHMET EROL

JAVA CHALLENGE

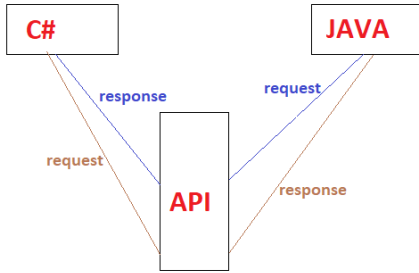
1- SAP Commerce (Hybris) nedir? Hangi amaçlarla kullanılır? Kullandığı teknolojiler nelerdir? Kısaca açıklayınız.

Daha önce SAP Hybris ile çalışmadım fakat tanımını yapabilirim.

SAP Commerce (Hybris), müşteri deneyimini ve yönetimini geliştirmek için kullanılan, geniş bir teknoloji yelpazesini kapsayan bir e-ticaret platformudur. Bu platform, işletmelerin müşteri etkileşimlerini yönetmelerine ve optimize etmelerine yardımcı olur.

Teknik olarak bakıldığında, SAP Commerce (Hybris) bir Java VM'de, bir Servlet Container'da veya bir J2EE uyumlu uygulama sunucusunda çalışır. MySQL, Oracle ve ya SAP HANA gibi harici bir veritabanına bağlanır.

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



Birbirinden bağımsız iki platformun birbiriyle haberleşmesi genellikle bir tür ara katman veya hizmet kullanılarak sağlanır. Bu, genellikle bir API veya web hizmeti şeklinde olur. Örneğin, X platformu Java ile yazılmış ve Y platformu C# ile yazılmış olsaydı, her iki platform da HTTP üzerinden JSON veya XML gibi bir veri formatını kullanarak bir RESTful API üzerinden iletişim kurabilir.

3- SOLR Nedir? Kullanım alanlarını araştırınız. Kurumsal bir projede kullanılabilecek iki farklı kullanım alanı örneği veriniz.

Apache Solr, Apache Lucene üzerine kurulu popüler, hızlı, açık kaynaklı bir kurumsal arama platformudur. Solr, yüksek güvenilirlik, ölçeklenebilirlik ve hata toleransı sağlar ve dağıtılmış indeksleme, replikasyon ve yük dengelemeli sorgulama, otomatik hata düzeltme ve kurtarma, merkezi konfigürasyon gibi özellikler sunar. Kullanım Alanları; Kurumsal Arama: Solr, kurumların belgeleri ve e-posta eklerini arayıp analiz etmek için kullanılabileceği bir platform sağlar. Sağlık Araştırmaları: Solr, araştırmacıların sayısız DNA modelini eşleştirmek için kullanılabileceği bir araçtır.

Kurumsal bir projede Solr'un kullanımına örnek olarak, Instagram ve Netflix'i verebiliriz

Instagram: Instagram, coğrafi arama API'sini desteklemek için Solr kullanır.

Netflix: Solr, bu yoğun site üzerinde temel film aramasını destekler

4- Aşağıdaki algoritma için uygun çözümü üretin.

- Java'da 100 adet random sayıya sahip bir liste oluşturun.
- Daha sonra bu listenin bir kopyasını oluşturun.
- 0 ile 100 arasında rastgele bir sayı üretin.
- Kopya listedeki bu random sayının olduğu indisteski değeri silin.
- Şimdi elinizde iki adet liste var ve kopya listede orjinal listeye göre bir eleman eksik.
- Hangi elemanın eksik olduğunu bulan bir metot oluşturun.

Aşağıda geliştirdiğim yöntem ve kod içerisinde gerekli olan açıklamalar bulunmaktadır. Bu yöntemim parametre olarak ister dizi gönderelim istersek te liste gönderelim doğru sonuçları verecektir.

```
public static void main(String[] args) {
    ArrayList<Integer> orjinalListe = new ArrayList<>();
    //orjinal liste
    Random rand = new Random();
    for (int i = 0; i < 100; i++) {
        orjinalListe.add(i: rand.nextInt(bound:101));
        //orjinal listeye 100 tane random eleman ekledim
    }
    ArrayList<Integer> kopyaListe = new ArrayList<>(orjinalListe);
    //kopya listesi oluşturup orjinal listenin elemanlarını aldım.
    int randomIndex = rand.nextInt(bound:100);
    //rastgele bir sayı silmesi için rastgele bir index seçtim
    kopyaListe.remove(index:randomIndex);
    //gelen rastgele indeks kopya listeden sildim.
    System.out.println("Original Liste:");
    for (int i : orjinalListe) {
        System.out.print(i + " ");
    }
    System.out.println("\n-----");
    System.out.println("Kopya Liste:");
    for (int i : kopyaListe) {
        System.out.print(i + " ");
    }
    System.out.println("\n");
    //yukarıda orjinal liste ve kopya listeyi kullanıcı görmesi için bastırdım.
    int eksikEleman = eksikOlanElemaniBul(orjinalListe, kopyaListe);
    System.out.println("Eksik olan eleman: " + eksikEleman);
}
```

```
public static int eksikOlanElemaniBul(ArrayList<Integer> orjinalListe, ArrayList<Integer> kopyaListe) {
    //
    parametre olarak gelen orjinalListe ve kopyaListeye göre methodumu oluşturdum.
    Şöyle bir algoritma geliştirdim: orjinal listedeki tüm sayıları topladım,
    daha sonra kopya listedeki elemanlardan çıkartarak en son kalan sayı benim eksik
    elemanın olarak düğündüm.
    */
    int eksikEleman = 0;
    for (int i : orjinalListe) {
        eksikEleman += i;
    }
    for (int i : kopyaListe) {
        eksikEleman -= i;
    }
    return eksikEleman;
}
```

Output - Run (EnocaTeknikSoru) x

```
Building EnocaTeknikSoru 1.0-SNAPSHOT
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ EnocaTeknikSoru ---
Original Liste:
84 33 16 60 14 53 31 92 38 54 84 98 99 52 41 5 81 35 76 82 100 55 67 31 10
-----
Kopya Liste:
84 33 16 60 14 53 31 38 54 84 98 99 52 41 5 81 35 76 82 100 55 67 31 10 91
Eksik olan eleman: 92
-----
BUILD SUCCESS
-----
Total time: 2.648 s
Finished at: 2024-03-31T15:26:23+03:00
-----
```

5- Cevap ve Github linki maildeki diğer dökümandır.