

## 1. Java dünyasındaki framework'ler ve çözdükleri problemler nedir?

a

## 2. SOA - Web Service - Restful Service - HTTP methods kavramlarını örneklerle açıklayınız.

### a. Service-Oriented-Architecture (SOA)

SOA, uygulamaların ayrı ve gevşek bir şekilde bağlanmış hizmetlerden oluştuğu bir yazılım tasarım yaklaşımıdır. Bu hizmetler, belirli bir işlevi yerine getirmek veya bir iş sürecini tamamlamak için genellikle bir ağ üzerinden birbirleriyle iletişim kurar. SOA'nın arkasındaki temel fikir, uygulamaları bağımsız olarak geliştirilebilen, konuşlandırılabilen ve bakımı yapılabilen daha küçük, yönetilebilir parçalara ayırmaktır.

Müşterilerin ürünlere göz atabildiği, bunları sepetlerine ekleyebildiği ve alışveriş yapabildiği, bir web sitesi üzerinden çalışan bir çevrimiçi mağaza hayal edelim. Mağazanın İT altyapısı SOA ilkeleri üzerine inşa edilmiş ve birkaç bağımsız hizmetten oluşuyor:

- Ürün Kataloğu Hizmeti: Adlar, açıklamalar ve fiyatlar gibi ürün bilgilerini yönetir. Müşteriler mağazaya göz attığında, bu hizmet gerekli ürün ayrıntılarını sağlar.
- Envanter Hizmeti: Stok seviyelerini takip eder. Bir müşteri satın alma işlemini tamamlamadan önce, bu hizmet ürünlerin stokta olup olmadığını kontrol eder.
- Sipariş İşleme Hizmeti: Müşteri siparişlerinin oluşturulmasını ve yönetilmesini sağlar. Stok seviyelerini güncellemek için Envanter Hizmeti ile ve ödemeleri işlemek için Ödeme Hizmeti ile iletişim kurar.
- Ödeme Hizmeti: Ödeme işlemlerini yönetir. Müşteri ödeme bilgilerini güvenli bir şekilde işler ve harici ödeme ağ geçitleriyle (kredi kartı işlemcileri gibi) iletişim kurar.
- Müşteri Profili Hizmeti: Kişisel bilgiler, gönderim adresleri ve sipariş geçmişi dahil olmak üzere müşteri hesaplarını korur.

Bir müşteri sipariş verdiğinde, bu hizmetler işlemi tamamlamak için birlikte çalışır. Sipariş İşleme Hizmeti süreci düzenler, ürünleri rezerve etmek için Envanter Hizmetini, ödemeyi işlemek için Ödeme Hizmetini çağırır ve Müşteri Profili Hizmetini sipariş ayrıntılarıyla günceller.

### b. Web Service

Bir web servisi, bir ağ üzerinden iki cihaz arasında standartlaştırılmış bir iletişim yolu veya ortamıdır. Bir ağ üzerinden birlikte çalışabilir makineler arası etkileşimi desteklemek için tasarlanmış bir yazılım sistemidir. Web servisleri TCP/IP, HTTP, Java, HTML ve XML gibi açık standartlar üzerine inşa edilmiştir ve üzerine inşa edildikleri programlama dili veya işletim sisteminden bağımsız olarak farklı sistemler ve platformlar arasında veri alışverişini ve entegrasyonu kolaylaştırır.

### c. Restful Service

RESTful servisi, "representational state transfer" (temsili durum aktarımı) (REST) teknolojisine dayanan, ağa bağlı uygulamalar tasarlamaya yönelik bir mimari stildir. CRUD (Create, Read, Update, Delete) işlemlerini yansıtan GET, POST, PUT, DELETE ve PATCH gibi işlemlerle verilere erişmek ve bunları kullanmak için HTTP metodlarını kullanır. RESTful servisleri durum bilgisi içermez, yani çağrılar birbirinden bağımsız olarak yapılabilir ve her çağrı kendini başarıyla tamamlamak için gerekli tüm verileri içerir.

Örnek olarak Sosyal Medya Profil Hizmeti:

GET /users/5599: 5599 ID'li kullanıcının profil bilgilerini alır.

POST /users: İstek gövdesinde sağlanan bilgilerle yeni bir kullanıcı profili oluşturur.

PUT /users/5599: 5599 ID'li kullanıcının profil bilgilerini istek gövdesinde sağlanan verilerle günceller.

DELETE /users/5599: 5599 ID'li kullanıcının profilini siler.

#### **d. HTTP Methods**

HTTP metotları, bir URL tarafından tanımlanan belirli bir kaynak üzerinde gerçekleştirilecek eylemi tanımlar. Her metot belirli bir işlem türünü ve amacını belirtir. İşte en yaygın kullanılan HTTP yöntemlerine ve tipik kullanım durumlarına genel bir bakış:

##### **1. GET**

Belirtilen bir kaynaktan veri alır. GET istekleri yalnızca veri getirmeli ve başka bir etkiye sahip olmamalıdır.

##### **2. POST**

Bir kaynak oluşturmak veya güncellemek için sunucuya veri gönderir. Veriler isteğin gövdesine dahil edilir. POST istekleri veri depolama dışındaki işlemleri de tetikleyebilir.

##### **3. PUT**

Hedef kaynağın mevcut tüm temsillerini yüklenen içerikle değiştirir.

##### **4. DELETE**

Belirtilen kaynağı kaldırır. Bir URL tarafından tanımlanan bir kaynağı silmek için DELETE isteği kullanılır.

##### **5. PATCH**

Bir kaynağı kısmen günceller. PATCH, bir kaynağı tamamen güncelleyen PUT'un aksine, bir kaynakta kısmi değişiklik yapmak için kullanılır.

Bu HTTP yöntemlerinin her biri, tek tip ve önceden tanımlanmış bir dizi durumsuz işlem sağlayacak şekilde "create, read, update, and delete" (CRUD) işlemlerine karşılık geldiği REST mimari stilinde çok önemli bir rol oynar. Bu, geliştiricilerin HTTP isteklerinin davranışını anlamasını ve tahmin etmesini ve evrensel olarak tüketilebilecek web hizmetleri tasarlamasını kolaylaştırır.