

## 1 – SOAP vs Restful ?

Rest, JSON, HTML ve XML gibi veri tiplerini desteklediğinden sadece XML tabanlı SOAP'a göre daha esnek ve kullanışlıdır.

Soap, isteklerini oluşturmak için W3C standardı olan bir WSDL dili kullanmak zorundadır. Rest'te buna gerek yoktur istekler URI üzerinden oluşturulabilir. Bir dile ihtiyaç duymadığından REST'in kullanımı ve tasarlanması daha kolaydır.

Güvenlik açısından SOAP için hazır fonksiyonlar bulunmaktadır ve daha kompleks oluşu onu bu konuda daha avantajlı yapar.

Her ikisi de HTTP protokolünü kullanır ve server-client arasındaki haberleşmeyi sağlar. Ancak REST için HTTP şarttır SOAP ise TCP/IP STMP gibi protokollere de destek verebilir.

Cachleme isteği için REST basit HTTP metodunu kullanır ve SOAP'a göre daha kolaydır. SOAP, bu işlem için karışık XML requestleri yapmaktadır.

## 2 - Difference between acceptance test and functional test ?

Fonksiyonel testler, yazılımın fonksiyonlarını, ne yaptığını test eden testlerdir. Örneğin, açtığınız tarayıcıdaki her menü öğesi amaçlanan işi yapıyorsa.

Kabul testleri, bazı kabul kriterlerine göre yapılan testlerdir. Bazı ekiplerin yazılımı, sözleşme gerekliliklerini karşılayıp karşılamadığını görmek için sipariş eden adına test etmesi veya yazılım testlerini gönderen şirket içindeki bazı ekibin, sevkiyata hazır olma kabul kriterlerine uygun olup olmadığını test etmesi gibi.

Kabul testi, “biz başlattık” ile “müşteriler kabul etti” arasında, ikincisine yakın bir adımdır.

İşlevsel test, test türüdür (birkaçını belirtmek için diğer türler yük testi, UX testi, erişilebilirlik testi olabilir).

## 3 - What is Mocking ?

Mocking(mocklama), popüler yazılım metodolojisi olan TDD ve özelde birim testlerinin (unit test), test ettikleri sistemi izole etmede kullandığı yöntemlerden biridir. Bu yöntemler, geniş anlamıyla test dublörleri (test double) olarak tanımlanabilir. Test dublörleri, test edilen sistemin bağımlı olduğu diğer birimlerin yerini tutar. Bu izolasyona birim testlerinde ihtiyaç duyulmasının temelde iki sebebi vardır:

1. Birim testleri, genelde test ettikleri sistemin kendisi ile ilgili varsayımları doğrulamak için yazılır.
2. Test dublörleri, davranış ve kullanım şekillerine göre çeşitlenir. Bunlardan en çok kullanılanları dummy, fake, stub, spy ve mock'tur denebilir. Bu çeşitliliğe sebep olan genel faktörler, bu dublörlerin beklenen işi yapıp yapmadığı ve yaparken nasıl bir davranış gösterdiği ile ilgilidir.

#### 4 - What is a reasonable code coverage % for unit tests (and why) ?

Yeni uygulama kodu için unit testleri ile %95 veya daha yüksek kapsamı hedeflemek makuldür.

Geliştiriciler programlarken unit testi yaptıklarında, kod tabanının ömrünü ve kalitesini artırır. Bir geliştirme ekibinin unit testlerine yatırım yaptığı süre, kusurları gidermek ve sorunları daha sonra analiz etmek için daha az zaman harcar. Ayrıca, regresyon testi için bir dizi unit testi ile ekipler, yeni bir özellik mevcut işlevselliği bozduğunda daha az örnekle karşılaşır.

#### 5 – HTTP/POST vs HTTP/PUT ?

POST kaynağa veri göndermek için kullanılır. PUT ise aynı kaynağa aynı adres ile erişilir ve eğer içerik var ise gelen veriler ile değiştirilir, eğer içerik yok ise yeni içerik yaratılır. Kısaca PUT veri güncellemek için kullanılır.

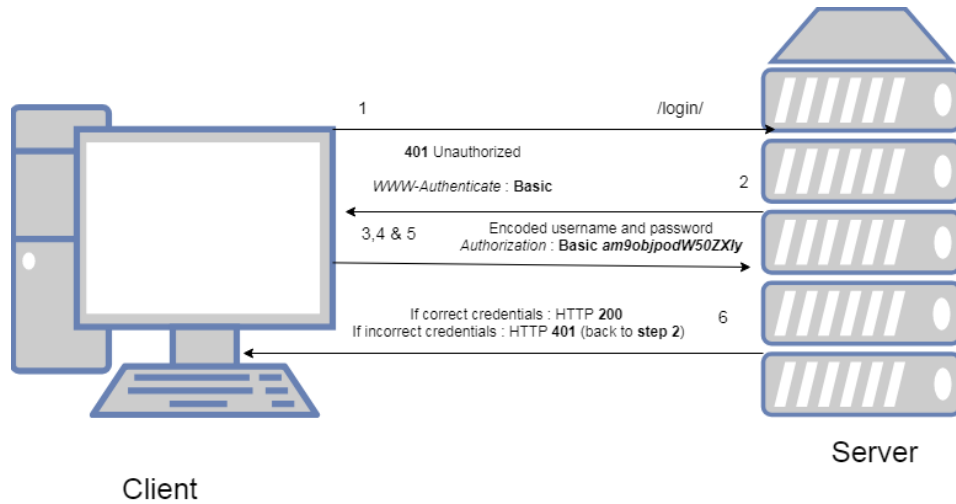
#### 6 - What are the Safe and Unsafe methods of HTTP ?

Server durumunu değiştirmezlerse HTTP methodları güvenli kabul edilir. Bu nedenle güvenli methodlar yalnızca salt okunur işlemler için kullanılabilir. HTTP RFC, güvenli olmak için aşağıdaki yöntemleri tanımlar: GET, HEAD, OPTIONS ve TRACE.

HTTP Method	Safe
GET	Yes
HEAD	Yes
OPTIONS	Yes
TRACE	Yes
PUT	No
DELETE	No
POST	No
PATCH	No

#### 7 - How does HTTP Basic Authentication work ?

HTTP Basic Authentication (Temel kimlik doğrulaması), istemcinin istekte bulunurken kullanıcı adı ve parola sağlaması için kullanılan bir yöntemdir. Sunucu tarafında belirli bir dosya veya tüm çalışma alanı için kurulabilen bu doğrulama yönteminde istemcinin doğru giriş bilgilerini girmesi gerekmektedir.



### 8 - Define RestTemplate in Spring ?

Resmi dökümantasyona göre RestTemplate, HTTP isteklerini gerçekleştirmek için senkronize bir istemcidir.

### 9 – What is the difference between @Controller and @RestController ?

@Controller, classları Spring MVC Controller olarak işaretlemek için kullanılır.

@RestController anotasyonu, RESTful Web hizmetlerinde kullanılan özel bir controller'dır ve @Controller ile @ResponseBody ek anotasyonlarının birleşimidir.

### 10 – What is DNS Spoofing ? How to prevent ?

DNS Spoofing, DNS zehirlenmesi ya da DNS önbellek zehirlenmesi anlamına gelir. DNS zehirlenmesi DNS alan adı sunucusundaki önbellek veritabanının içerdiği veriler ile gerçekleşen bir çeşit bilgisayar saldırısıdır.

DNS zehirlenmesi, Domain Ad Sistemi sunucusunun ön bellek veritabanına veri eklenmesi ya da oradaki verilerin değiştirilmesiyle oluşur. Bu durum oluşurken; yanlış IP adreslerinin dönmesi veya trafiğin başka bir bilgisayara yönlendirilmesi gerçekleşir. Genellikle, trafiğin; saldırıyı yapanın bilgisayarına yönlendirilmesine neden olur.

DNS zehirlenmesinden korunabilmek için, DNS ön belleğinizi aralıklarla temizlemeniz tavsiye edilmektedir.

### 11 – What is content negotiation ?

Content Negotiation istemci ile sunucu arasındaki içerik anlaşması anlamına gelir. HTTP protokolünün inşa edildiği şekle bakıldığında kullanıcının ne tür veri göndereceği ve sunucunun ne tür veri kabul edeceğinin özelleştirilmesidir. JAX-RS içerisinde kabul edilecek içerik türü @Consumes notasyonu ile, üretilecek içerik de @Produces notasyonu ile belirleniyor.

### 12 – What is statelessness in RESTful Web Services ?

Bir RESTful Web Servisi, sunucuda bir client durumu tutmamalıdır. Bu kısıtlamaya statelessness denir. Contextini sunucuya iletmek istemcinin sorumluluğundadır ve ardından sunucu, müşterinin daha sonraki isteğini işlemek için bu contexti saklayabilir. Örneğin, sunucu tarafından sürdürülen oturum, istemci tarafından geçirilen oturum tanımlayıcısı tarafından tanımlanır.

### **13 - What is CSRF attack? How to prevent ?**

CSRF (Cross Site Request Forgery) genel yapı olarak sitenin açığından faydalanarak siteye sanki o kullanıcıymış gibi erişerek işlem yapmasını sağlar.

Genellikle GET requestleri ve SESSION işlemlerinin doğru kontrol edilememesi durumlarındaki açıklardan saldırganların faydalanmasını sağlamaktadır.

Bu tür açıkları kapatmak için en pratik yol ise token kullanımıdır.

### **14 - What are the core components of the HTTP request and HTTP response?**

HTTP Request:

- Verb – GET, POST, DELETE, PUT vb. HTTP yöntemlerini belirtir.
- URI – Uniform Resource Identifier (URI) sunucudaki kaynağı tanımlamayı sağlar.
- HTTP Version – HTTP versionunu belirtir. Örneğin HTTP v1.1 .
- Request Header – Key/value çiftleri olarak HTTP isteği request mesajı için meta verileri içerir. Örneğin, client (veya tarayıcı) türü, client tarafından desteklenen biçim, ileti gövdesi biçimi, önbellek ayarları vb.
- Request Body – Content veya kaynak gösterimi.

HTTP Response:

- Status Line
- Headers
- Body (opsiyonel)