Homework 3- Mustafa YAŞAR

1. SOAP ve Restful servisleri arasındaki farklar nelerdir?

REST (Representational State Transfer): İstemci-sunucu arasında hızlı ve kolay şekilde iletişim kurulmasını sağlayan bir servis yapısıdır. REST standartlarına uygun yazılan web servislerine RESTful servisler denir.

SOAP (Simple Access Protocol): En temel anlamda, internet üzerinden küçük miktarda bilgileri ya da mesajları aktarma protokolüdür.

REST vs SOAP

SOAP	REST
SOAP bir protokoldür, bir web servisi oluştururken uyulması gereken kurallar bütünüdür.	REST bir mimari stildir.
Yalnızca XML mesaj biçimini destekler.	Veri formatında herhangi bir kısıtlama getirmez, çoklu veri formatlarını destekler – JSON, XML, CSV vb.
SOAP, bir istemci tarafından tüketilmek üzere çağrılmış yöntemlerini ortaya çıkarır.	REST kaynakları, URI ve HTTP Verbs-GET, PUT, POST ve DELETE hizmeti tarafından ortaya çıkar.
Ayrıntılı XML biçiminin kullanılması nedeniyle REST'ten daha yavaştır. Ancak daha güvenli olarak kabul edilir. Bunun nedeni, web hizmeti güvenliğini iyileştirmek için Web Hizmeti Belirtiminin bir bileşeni olan WS-security'yi kullanmasıdır.	REST, hafif ve daha hızlı olarak kabul edilir (XML ayrıştırması gerekmez).
SOAP Web servisleri istekleriyle birlikte SOAP başlıklarını kullanır. Bu başlıklar, istek hakkında meta/ek bilgiler içerir.	REST API'leri daha az bant genişliği tüketir. Bunun nedeni, isteğin her iletide SOAP başlıkları gerektirmemesidir. REST, herhangi bir meta bilgi için HTTP başlıklarını kullanır.
SOAP web hizmeti bir sözleşmeyi takip eder ve karmaşık mantığı uygulamak için tercih edilir.	RESTful web servislerinin oluşturulması daha kolay ve hızlıdır.

2. Kabul Testi ve Fonksiyonel Test arasındaki farklar nelerdir?

Kabul Testi	Fonksiyonel Test
Kabul testi, yazılımı müşteri beklentilerine göre doğrular. Son kullanıcı, yazılımın kabul edilebilir olup olmadığına karar vermek için yazılımın işlevselliğini kontrol eder. Beta testi veya son kullanıcı testi olarak da bilinir.	Yazılımın işlevselliğini doğrulayan bir tür yazılım testidir. Test işlemi sırasında kaynak kodu dikkate alınmadığından, esas olarak kara kutu tipi bir testtir. Bu yazılım testi biçiminin temel amacı, belirli girdiler sağlayarak ve çıktıları işlevsel gereksinimlere göre doğrulayarak uygulamanın her bir işlevselliğini test etmektir. Tamamen program spesifikasyonlarına dayalı olduğu için spesifikasyon bazlı test olarak da bilinir.
Bu, işlevsellik ve regresyon testinden sonra gerçekleştirilen en son adımdır.	Kabul testinden önce fonksiyonel testler (unit testing, smoke testing, integration testing, regression testing vb.) yapılır.
UAT Test Sürecinde iş analisti, QA lideri veya Test Yöneticisi vardır. Gereksinimler doğrultusunda işletme veya ürün sahibi yer alır.	Yazılım veya QA mühendisleri tarafından gerçekleştirilir

3. Mocklama nedir?

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.

Mock'ların genel kullanım şekli, yerine geçtiği bağımlılık üzerinde, test edilen sistemin yapması beklenen işlemlerin yapılıp yapılmadığını doğrulamak olarak tanımlanabilir. Mock nesnelerinin casus nesnelerinden (spy) farkı, her ikisi de üzerlerinde yapılan işlemleri takip ederken, mocklar bu işlemi testlerin doğrulama (assert) kısmına da entegre ederler.

Mocklama işlemi genelde kütüphaneler yardımıyla, test metodlarının içinde veya tekrar eden test ayarlarının yapıldığı bir özelleştirme metodunu, satır arası kodlar ile yapılır. Yani çoğu zaman mocklanan tipten devralan bir tip yazılmaz. Mock kütüphaneleri genelde bu işi, dilin reflection kütüphanesinden faydalanarak, çalışma zamanında, ayarlanan kurulumu sağlayacak vekil tipler üreterek sağlarlar. Bunun getirdiği bir avantaj, bütün ön koşulları yerine getiren bir dublör birimi yazmadan, her test metodu için yalnızca beklenen davranışı sağlayan satır içi ayarlamalar yapılmasına olanak sağlamasıdır.

Mock nesnelerini doğru yerlerde kullanmanın çeşitli avantajları olduğu gibi, dezavantajları da olabiliyor.Örneğin test ortamında mock kullanılacak bir sistemde neredeyse her şeyin birer arayüz (interface) üzerine inşa edilmesi gerekebiliyor. Bu da kimi zaman over-engineering olarak nitelendirilen probleme yol açabiliyor.

Java'nın Mocking Kütüphaneleri

- JMock
- Mockito
- EasyMock

4. Birim testi için makul kod kapsamı % kaçtır ve bunun nedeni nedir?

Birçok proje için makul kod kapsamı hedefi %70-80 dir. %100 kapsamı genellikle zaman alıcı ve yüksek maliyetlidir. %100 kod kapsamının bile bir sistemdeki hataların yalnızca yarısını ortaya çıkardığı tahmin edilmektedir. Düşük kod kapsamı yetersiz test anlamına gelir, ancak yüksek kod kapsamı hiçbir şeyi garanti etmez.

5. Http/Post ve Http/Put karşılaştırması yapınız.

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

PUT	POST
PUT metodu "idempotent(etkisiz)" olarak tanımlanmıştır. Yani aynı resource path üzerinde gerçekleştirilen birden fazla PUT request'i server tarafında ilk PUT request'inin yaptığından başka bir state değişikliğine neden olmaz.	İdempotent değildir. POST ile aynı resource path'ine yapılan birden fazla request her seferinde server tarafında bir state değişikliğine neden olacaktır.
PUT yanıtları önbelleğe alınamaz.	POST yanıtları önbelleğe alınamaz.
Nesne için id 'yi client veriyorsa put kullanılır.	İd 'yi Server veriyorsa o zaman post kullanılır.

6. Http'nin güvenli ve güvenli olmayan metotları nelerdir?

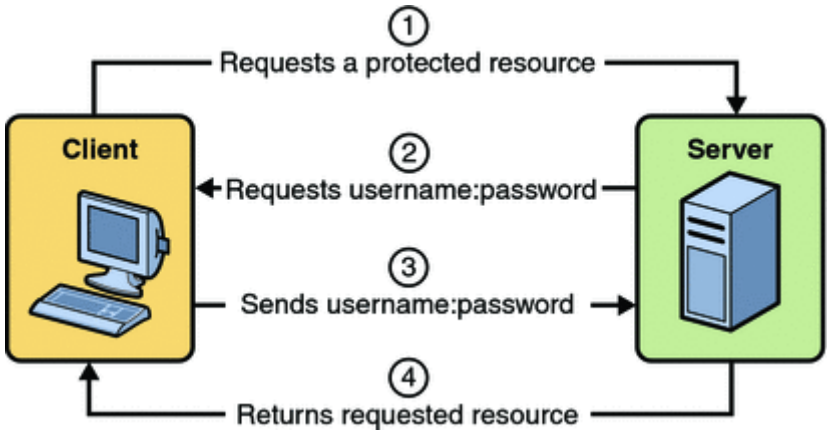
Bir HTTP metodu, sunucunun durumunu değiştirmiyorsa güvenlidir. Başka bir deyişle, bir metot salt okunur bir işleme yol açıyorsa güvenlidir. Birkaç yaygın HTTP metodu güvenlidir: GET, HEAD veya OPTIONS. Tüm güvenli metotlar aynı zamanda idempotentdır, ancak tüm idempotent yöntemler güvenli değildir. Örneğin, PUT ve DELETE her ikisi de idempotentdır ancak güvenli değillerdir.

HTTP Metotları	Güvenli mi?	Idempotent mı?
GET	EVET	EVET
HEAD	EVET	EVET
OPTIONS	EVET	EVET
TRACE	EVET	EVET
PUT	HAYIR	EVET
DELETE	HAYIR	EVET
POST	HAYIR	HAYIR
PATCH	HAYIR	HAYIR

7. Http Temel Kimlik Doğrulama nasıl çalışır?

HTTP Temel Kimlik Doğrulaması, sunucunun web istemcisinden bir kullanıcı adı ve parola istemesini ve bunları yetkili kullanıcılardan oluşan bir veri tabanıyla karşılaştırarak kullanıcı adı ve parolanın geçerli olduğunu doğrulamasını gerektirir. Temel kimlik doğrulama bildirildiğinde, aşağıdaki eylemler gerçekleşir:

1. Bir istemci, korunan bir kaynağa erişim ister.
2. Web sunucusu, kullanıcı adı ve parola isteyen bir iletişim kutusu döndürür.
3. İstemci, kullanıcı adını ve parolayı sunucuya gönderir.
4. Sunucu, belirtilen alanda kullanıcının kimliğini doğrular ve başarılı olursa istenen kaynağı döndürür.



HTTP temel kimlik doğrulaması, güvenli bir kimlik doğrulama mekanizması değildir. Temel kimlik doğrulama, kullanıcı adlarını ve parolaları İnternet üzerinden Base64 kodlu metin olarak gönderir ve hedef sunucunun kimliği doğrulanmaz. Bu kimlik doğrulama biçimi, kullanıcı adlarını ve parolaları açığa çıkarabilir. Birisi iletimi kesebilirse, kullanıcı adı ve şifre bilgilerinin kodu kolayca çözülebilir. Ancak, SSL gibi güvenli bir aktarım mekanizması veya IPSEC protokolü veya VPN stratejileri gibi ağ düzeyindeki güvenlik önlemleri ile birlikte kullanıldığında, bu risklerin bazıları önlene bilinir.

8. RestTemplate nedir ve Spring de nasıl tanımlanır?

RestTemplate bir Spring uygulamasında senkronize HTTP isteklerinin nasıl oluşturulacağını belirler. 3 şekilde tanımlanabilir.

- Varsayılan RestTemplateBuilder'ı kullanarak
- RestTemplateCustomizer kullanarak
- Kendi RestTemplateBuilder'ımızı oluşturarak

9. "Idempotent" nedir ve hangi HTTP metotları "idempotent"dır?

Matematikte Idempotent bir fonksiyon bir defa çağırdığında etki gösterir ama bu ilk çağırmanın ardından n defa daha çağırılır ise ilk çağrılmaya gösterdiği etkiden sonra başka bir etki göstermez. Bilgisayar bilimlerinde ilk uygulamadan sonra sonucu değiştirmeden birden çok kez uygulanabilen metotlar elde ederiz.

Idempotent http metotları:

HTTP Metotları	Idempotent mı?
GET	EVET
HEAD	EVET
OPTIONS	EVET
TRACE	EVET
PUT	EVET
DELETE	EVET
POST	HAYIR
PATCH	HAYIR

10. "DNS Spoofing" nedir ve nasıl önlenir?

Normalde, ağa bağlı bir bilgisayar, İnternet servis sağlayıcısı (ISS) veya kullanıcının bilgisayarı tarafından sağlanan bir DNS sunucusunu kullanır. Bir kuruluşun ağında kullanılan DNS sunucuları, daha önce elde edilen sorgu sonuçlarını önbelleğe alarak çözünürlük yanıt performansını artırır. Tek bir DNS sunucusuna yapılan zehirlenme saldırıları, kullanıcıları tehlikeye girmiş sunucu ile direkt olarak veya mümkünse tehlikeye girmiş sunucunun alt sunucuları ile dolaylı olarak etkileyebilir.

Önbellek zehirlenmesi saldırısı gerçekleştirmek için saldırgan, DNS yazılımındaki kusurlardan yararlanır. Sunucu, yetkili bir kaynaktan geldiğinden emin olmak için DNS yanıtlarını doğrulamalıdır (örneğin, DNSSEC kullanarak); aksi halde, sunucu yanlış girdileri önbelleğe alabilir ve aynı isteği yapan diğer kullanıcılara sunabilir.

Bu saldırı, kullanıcıları bir web sitesinden, saldırganın seçtiği başka bir web sitesine yönlendirmek için kullanılabilir. Örneğin, saldırgan, DNS sunucusundaki hedef web sitesinin IP adresini, kontrolü altındaki bir sunucunun IP adresi ile değiştirir. Saldırgan daha sonra kendi sunucusundaki dosyaları, hedef sunucudakilerle eşleşecek şekilde oluşturur. Bu dosyalar genellikle bilgisayar solucanları veya bilgisayar virüsleri gibi zararlı içerikler içerir. Böylece, bilgisayarı zehirli DNS sunucusuna bağlanan bir kullanıcı, orijinal olmayan bir sunucudan gelen içeriği kabul etmeye kandırılır ve kötü niyetli içeriği

bilmeden indirir. Bu teknik ayrıca, banka ve kredi kartı bilgileri gibi kişisel bilgileri toplamak için, orijinal bir web sitesinin sahte bir sürümünün oluşturulduğu kimlik avı saldırılarında da kullanılabilir.

Alınabilecek Önlemeler

DNS zehirlenmesi saldırıları tespit edilmesi ve çözümü zor olabileceği için çok tehlikelidir. DNS servis sağlayıcısı veya web sitesi sahipleri, tehditleri yönetmek için çeşitli araçlar ve protokoller kullanarak kullanıcıları korumak adına adımlar atmalıdır. Bu tip saldırılardan korumanın bilinen en iyi yollarını şu şekilde sıralayabiliriz:

1. DNSSEC'i tanıtmak, DNS zehirlenmesi saldırılarına karşı korunmak için alabileceğiniz en değerli önlemlerden biridir. DNSSEC, mevcut internet protokollerinde standart olmayan DNS verilerini doğrulamayı mümkün kılmak için ortak anahtar şifrelemesine güvenir. Spesifik olarak, bir isteğe yanıt veren herhangi bir DNS'in kök alan adını doğrulamak ve bunu yapmaya yetkili olduğundan emin olmak için sertifika tabanlı kimlik doğrulamasını kullanır. Ayrıca, yanıtın içeriğine güvenilip güvenilmeyeceğini ve bu içeriklerin aktarım sırasında değiştirilip değiştirilmediğini değerlendirir.
2. Bir diğer önemli adım, DNS istek ve yanıtlarında yer alan verileri her zaman şifrelemektir. Bu, verilere müdahale edebilecek siber suçlulara karşı ek bir koruma katmanı sunar. Örneğin, bir saldırgan şifrelenmiş verileri ele geçirmeyi başarsa bile, gelecekteki yanıtlarda kullanmak üzere çoğaltmak için ihtiyaç duyduğu bilgileri almak için okuyamaz.
3. Kuruluşlar, DNS'e ek bir koruma katmanı sağlayan yapılandırmalar için adımlar da atabilir. DNS sunucularını, diğer DNS sunucularıyla olan ilişkilere büyük ölçüde güvenmeyecek şekilde yapılandırabilirler. Bu, bilgisayar korsanlarının kendi DNS sunucuları aracılığıyla bağlantı kurmasını zorlaştırır. Ek olarak kuruluşlar, DNS sunucularını yalnızca belirli hizmetlerin çalışmasına izin verecek şekilde daha sınırlı veri kümelerini depolayacak şekilde yapılandırabilirler.
4. Sistem güncellemeleri genellikle yeni güvenlik protokolleri ve tanımlanmış güvenlik açıklarına yönelik düzeltmeler içerdiğinden DNS'in en son sürümünü kullanmak da son derece önemlidir.
5. DNS zehirlenmesi saldırısının gerçekleşmesi durumunda güçlü algılama protokollerinin olması sonucu değiştirebilir. En iyi algılama protokolleri, düzenli izleme kullanır. En büyük uyarı işaretlerinden biri tek bir alan adı hakkında tek bir kaynaktan DNS etkinliğinde artış olması ve tek bir kaynaktan birden fazla alan adı hakkında DNS etkinliğinde artış olmasıdır. Bunlar, DNS zehirlenmesi için bir giriş noktası bulma girişimlerinin göstergesidir.
6. Potansiyel risklerin farkına varmak için siber güvenlik eğitimleri önerilir. Fark etmesi çok zor olsa çalışanların siber tehditler konusunda eğitilmesi kuruluşların itibar ve finans kaybı yaşama ihtimallerini zayıflatır.

11. "Content negotiation" nedir?

Content Negotiation işleyişi HTTP protokolüne özgü bir kavramdır. Anlam olarak tercüme edecek olursak, client ve server arasında yapılan bir içerik anlaşması veya müzakeresidir diyebiliriz. Amacı, aynı URI ile farklı doküman türlerinde içerik sunabilmektir. Yani daha genel bir ifadeyle kaynak gösterim şeklinin kullanıcılar tarafından belirlenmesi diyebiliriz.

12. RESTful Servislerde “statelessness” nedir?

REST (Representational “State” Transfer) mimarisine göre sunucu, sunucu tarafında istemci oturumu ile ilgili herhangi bir durumu saklamaz. Bu kısıtlamaya “statelessness” denir. İstemciden sunucuya yapılan her istek, isteği anlamak için gerekli tüm bilgileri içermelidir. Sunucu, sunucuda saklanan herhangi bir bağlamdan yararlanamaz. Uygulamanın oturum durumu bu nedenle tamamen istemcide tutulur. Müşteri, oturumla ilgili bilgileri kendi tarafında depolamaktan ve kullanmaktan sorumludur.

13. CSRF saldırısı nedir? Nasıl önlenir?

CSRF saldırısı, daha önce kimliği doğrulanmış başka bir web sitesi aracılığıyla bir web uygulamasına istek gönderen kötü amaçlı bir bağlantı içerir. Elde edilen kimlik bilgileriyle mağdur kimliğine bürünür ve kötü amaçlı faaliyetlerde kimlik doğrulama bilgisi atlanılmış olur. Örneğin, bankacılık sistemine giriş sayfası tarayıcıda açık bulunduğu bir durumda, mail adresine gelen tehlikeli bir bağlantı tıklanarak saldırgana kullanıcı bilgileri verilmiş olur. Saldırgan bu bilgilerle bankacılık sistemine girip para transferi gerçekleştirebilir. Bu tür saldırılar genellikle bankacılık, sosyal medya ve ağ cihazları için kullanılan web ara yüzlerine karşı gerçekleştirilir.

CSRF Zafiyetinde Alınabilecek Önlemler

a) Token Kullanımı

Kullanıcıya her oturum için random ve benzersiz “token” bilgisi verilir.

b) CAPTCHA Kullanımı

Bir web formunda captcha (Completely Automated Public Turing test to tell Computers and Humans Apart) bilgisi doğru girilmediği sürece işlem gerçekleştirilemeyeceği için “CSRF” saldırısına karşı alınacak bir önlem niteliğindedir.

14. HTTP isteğinin ve HTTP yanıtının temel bileşenleri nelerdir?

HTTP isteğinin 5 ana bileşeni vardır. Ve bu bileşenler;

- Yöntem
- URI
- HTTP Sürümü
- İstek Başlığı
- İstek Gövdesi

HTTP Yanıtı, istemcinin kolayca anlayabilmesi için takip edilen özel bir yapıya sahiptir. İnsanlar arasında iletişim kopukluğu olmaması için herkesin takip ettiği bir Evrensel Dil vardır. HTTP Yanıtı genel olarak 3 ana bileşene sahiptir:

- Durum Satırı
- Başlıklar
- Gövde (Opsiyonel)