

Theoretical Homework #2

1) SOAP vs Restful?

SOAP ve Rest iki farklı online data transfer yaklaşımıdır. İkisi de data ve web uygulamaları arasında bağlantıyı sağlayan API arayüzlerinin nasıl oluşturulduğuyla alakalıdır. Birçok eski sistem hala SOAP kullanmaktadır. REST daha yeni bir teknolojidir ve daha hızlı bir alternatif olarak görülür. REST, daha esnek bir yapıya sahipken SOAP ise XML mesajlaşma gibi belirli kuralları olan bir protokoldür. Günümüz teknolojileri için SOAP yükü fazla olduğu için pek tercih edilmez.

2) Difference between acceptance test and functional test?

Fonksiyonel testler ürünün fonksiyonlarının test edildiği, ürünün her türlü olayı senaryoya karşı kapsamının denetlendiği, çalışma prensiplerinin tetkik edildiği testlerdir. Kabul testleri ise daha çok müşteri odaklı olup, belli kabullerle ürünün müşterinin isteklerini karşılayıp karşılanmadığı testlerdir. Kabul testlerinin kapsamı müşterinin istekleriyle doğru orantılı olduğundan kapsamının daha dar olduğunu kabul edebiliriz.

3) What is Mocking?

Birim testlerde tercih edilen bir yöntemdir. Test edilmek istenen birimin bağlı olduğu sınıflar mocking ile simüle edilir ve bu simüle edilen sınıflarla birimin bağımlılıklarının önüne geçilerek test edilmek istenen birimin bağımsız çalışması sağlanır. Böylece birim bağımsız olarak test edilmiş olur.

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

Kod kapsamının %100 olmasını beklemek yanıltıcı bir ölçüt olarak kabul edilir. Bu tarz durumlarda %80 üzeri makul kabul edilip en çok kullanılan test yollarını karşılamak hedeflenmektedir. %90+ olması gereken değil, olursa iyi olur olarak görülmelidir.

5) HTTP/POST vs HTTP/PUT?

HTTP/PUT	HTTP/POST
PUT pratikte UPDATE operasyonudur.	POST Create operasyonu için kullanılır.
PUT isteği belirli bir kaynağa yapılır. Eğer istek URI'yı halihazırda var ise güncelleme işlemi yapılır.	POST yöntemi, kaynak sunucusunun altında yeni bir alt öge yaratır. Bu da POST istek URI'sının belli bir kaynağa ait olmasını gerektirir.
Birden fazla PUT isteği göndermek sadece bir işleme karşılık gelir.	Gönderilen POST isteği kadar kaynak altında yeni URI oluşturulur.

6) What are the Safe and Unsafe methods of HTTP?

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

7) How does HTTP Basic Authentication work?

HTTP Basic Authentication, bir sunucunun bir istemciden kimlik doğrulama bilgisi (bir kullanıcı kimliği ve parola) isteyebileceği basit bir sorgulama ve yanıt mekanizmasıdır. İstemci, kimlik doğrulama bilgilerini bir Authorization başlığında sunucuya iletir. Kimlik doğrulama bilgileri, base-64 kodlamasındadır. Bir istemci, sunucunun kimlik doğrulama bilgisi beklediği bir istekte bulunursa, sunucu bir 401 durum kodu, bir kimlik doğrulama hatasını gösteren bir neden ifadesi ve bir WWW-Authenticate başlığı ile bir HTTP yanıtı gönderir. Çoğu web istemcisi, bu yanıtı son kullanıcıdan bir kullanıcı kimliği ve parola isteyerek gerçekleştirir.

8) Define RestTemplate in Spring?

RestTemplate, istemci tarafı HTTP erişimi için merkezi Spring sınıfıdır. Kavramsal olarak, JdbcTemplate, JmsTemplate ve Spring Framework ve diğer portföy projelerinde bulunan diğer çeşitli şablonlara çok benzer. Bu, örneğin, RestTemplate oluşturulduktan sonra iş parçacığı için güvenli olduğu ve işlemlerini özelleştirmek için geri aramaları kullanabileceğiniz anlamına gelir.

9) What is the difference between @Controller and @RestController?

@Controller sınıfları Spring MVC Controller sınıfları olarak belirtmeye yarar. @RestController anotasyonu ise RESTful Web servisleri için kullanılan @Controller ve @ResponseBody anotasyonlarının birleşimi olarak kullanılır. @RestController, @Controller'ın özelleştirilmesiyle oluşturulan yeni bir versiyonu da diyebiliriz. @Controller bir Spring Web MVC görüntüsü oluştururken @RestController buna ihtiyaç duymaz.

10) What is DNS Spoofing? How to prevent?

DNS Spoofing, Domain Name System'inin (DNS) verisini bozarak, DNS çözümlemesine bozuk veriyi dahil eden bir güvenlik saldırısıdır. Böylece saldırgan, veri trafiğini istediği bilgisayara yönlendirebilir.

11) What is content negotiation?

HTTP'de içerik anlaşması, kullanıcı aracısının hangi gösterimin kullanıcı için en uygun olduğunu belirlemesine yardımcı olmak için bir kaynağın farklı temsillerini aynı URI'ye sunmak için kullanılan mekanizmadır.

12) What is statelessness in RESTful Web Services?

REST mimarisine göre, bir RESTful Web Hizmeti, sunucuda bir istemci durumu tutmamalıdır. Bu kısıtlamaya "Statelessness" denir. Bağlamını sunucuya iletmek istemcinin sorumluluğundadır ve ardından sunucu, müşterinin daha sonraki isteğini işlemek için bu bağlamı saklayabilir. Örneğin, sunucu tarafından sürdürülen oturum, istemci tarafından geçirilen oturum tanımlayıcısı tarafından tanımlanır.

13) What is CSRF attack? How to prevent?

Cross-site Request Forgery (CSRF) siteler arası istek sahteciliği olarak tanımlanır. Bu, bir saldırganın bir kurbanı kendi adına işlemler yaptırdığı bir saldırı türüdür. Kurbanın sistem içerisinde erişim kapsamı ne kadar geniş ise saldırının boyutu da o kadar büyük olur.

Bu saldırıları önlemek kolay değildir ancak önlenmesi adına uygulanması gereken birkaç adım vardır.

Bunlar;

- Farkındalık oluşturmak ve eğitim vermek
- Risk değerlendirmesi yapmak (Bu saldırılar yalnızca kimlik doğrulamaya gerek duyan sitelerde olur, eğer kimlik doğrulamaya ihtiyacınız olan bir siteniz yoksa bu durumu göz ardı edebilirsiniz.)
- Anti-CSRF token kullanımı
- SameSite çerez kullanımı
- Acunetix ile düzenli tarama

14) What are the core components of the HTTP request and HTTP response?

HTTP Request için 5 ana bileşen vardır.

Verb - GET, POST, DELETE, PUT vb. gibi HTTP yöntemlerini belirtin.

URI - Sunucudaki kaynağı tanımlamak için Tekdüzen Kaynak Tanımlayıcısı (URI).

HTTP Version – HTTP sürümünü belirtir, örneğin HTTP v1.1 .

Request Header – Anahtar/değer çiftleri olarak HTTP İstek mesajı için meta verileri içerir. Örneğin, istemci (veya tarayıcı) türü, istemci tarafından desteklenen biçim, ileti gövdesi biçimi, önbellek ayarları vb.

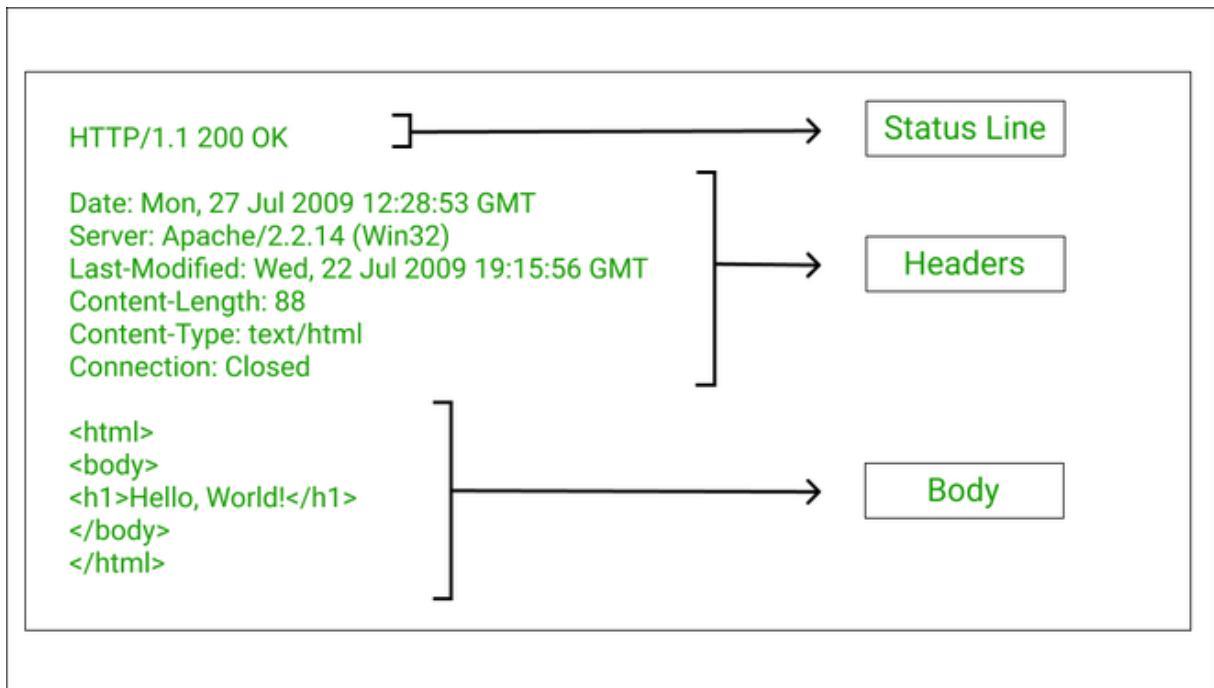
Request Body – Mesaj içeriği veya Kaynak gösterimi

HTTP Response: HTTP Response, istemcinin kolayca anlayabilmesi için takip edilen özel bir yapıya sahiptir. Herkesin takip ettiği bir Evrensel Dil vardır ki insanlar arasında iletişim kopukluğu olmasın. HTTP Response genel olarak 3 ana bileşene sahiptir:

Status Line (Durum Satırı)

Headers (Başlıklar)

Body (Optional)



15) What is JPA?

Java Persistence API (JPA), Java'nın bir özelliğidir. Java nesnesi ile ilişkisel veri tabanı arasındaki verileri kalıcı hale getirmek için kullanılır. JPA, nesne yönelimli etki alanı modelleri ve ilişkisel veri tabanı sistemleri arasında bir köprü görevi görür. JPA sadece bir spesifikasyon olduğu için kendi başına herhangi bir işlem gerçekleştirmez. Bir uygulama gerektirir. Bu nedenle, Hibernate, TopLink ve iBatis gibi ORM araçları, veri kalıcılığı için JPA spesifikasyonlarını uygular.

16) What is the naming convention for finder methods in the Spring data repository interface?

limit – Query'nin sonucu "first/top" anahtar kelimeleri kullanılarak sınırlandırılabilir

- findFirst10ByLastname
- findFirstOrderByLastnameAsc
- findTop3ByLastname
- findTopOrderByAgeDesc

property/properties expression – Sonuç Entity'nin özelliğine göre filtrelenebilir, and veya or anahtar kelimeleri kullanılarak seçimler çoğaltılabilir.

- findByLastnameAndFirstname
- findByLastnameOrFirstname
- findByFirstname

comparison – özellik belirtildikten sonra karşılaştırma operatörleri kullanılabilir.

- findByFirstnames
- findByFirstnameEquals
- findByStartDateBetween
- findByAgeLessThan, findByAgeLessThanEqual
- findByAgeGreaterThan, findByAgeGreaterThanEqual
- findByStartDateBefore, findByStartDateAfter
- findByAgeIsNull, findByAgeIsNotNull
- findByFirstnameLike, findByFirstnameNotLike
- findByFirstnameStartingWith, findByFirstnameEndingWith
- findByFirstnameContaining
- findByLastnameNot
- findByAgeIn(Collection ages), findByAgeNotIn(Collection ages)
- findByActiveTrue, findByActiveFalse
- findByFirstnameIgnoreCase

ordering operator – Metot operatörlerinin sonunda sıralama operatörleri kullanılabilir.

- findByLastnameOrderByFirstnameAsc
- findByLastnameOrderByFirstnameDesc

17) What is PagingAndSortingRepository?

PagingAndSortingRepository, paging ve sorting abstraction kullanarak varlıkları almak için ek yöntemler sağlayan CrudRepository'nin bir uzantısıdır.

İki metot içerir:

Page findAll(Pageable pageable) – Pageable bir nesne üzerinden üretilen bir entity sayfası döndürür.

Iterable findAll(Sort sort) – Verilen seçeneklere göre sıralanmış tüm varlıkları döndürür. Burada sayfalama uygulanmaz.

18) Differentiate between findById() and findOne()?

Her iki metot da veri tabanından verilerin döndürülmesi için kullanılır. Ancak findOne() metodu veri tabanına direkt ulaşmadan Entity.Manager.getReference() metodu yardımıyla Proxy geri döndürerek bu işi yaparken, findById() metodu direkt veri tabanına ulaşır, nesne eşleşmelerini gerçekleştirir ve eşleşmenin gerçekleştiği satırı döndürür. Eğer bulamazsa, null değer atar.

19) What is @Query used for?

@Query anotasyonu, bulucu sorgularını doğrudan repository metotlarıyla bildirir. Domain sınıflarında benzer @NamedQuery kullanılırken, Repository interface'inde Spring Data JPA @Query anotasyonu kullanılır. Bu, domain sınıflarını kalıcılığa özgü bilgilerden kurtarır, ki bu iyi bir şeydir.

20) What is lazy loading in hibernate?

Lazy loading, Hibernate'deki tüm entityler için kullanılan bir fetch tekniğidir. Üst sınıf nesnesini yüklerken bir alt sınıf nesnesinin yüklenip yüklenmeyeceğine karar verir. Mapping ile ilişki eşleme kullandığımızda, getirme tekniğini tanımlamamız gerekir. Lazy loading'in temel amacı, sadece gerekli nesneleri veri tabanından getirmektir.

21) What is SQL injection attack? Is Hibernate open to SQL injection attack?

SQL Injection (SQLi), kötü niyetli SQL ifadelerini çalıştırmayı mümkün kılan bir enjeksiyon saldırısı türüdür. Bu ifadeler, bir web uygulamasının arkasındaki bir veri tabanı sunucusunu kontrol eder. Saldırganlar, uygulama güvenlik önlemlerini atlamak için SQL Injection güvenlik açıklarını kullanabilir. Bir web sayfasının veya web uygulamasının kimlik doğrulamasını ve yetkilendirmesini dolaşabilir ve tüm SQL veri tabanının içeriğini alabilirler. Veri tabanına kayıt eklemek, değiştirmek ve silmek için SQL Injection'ı da kullanabilirler.

Hibernat, SQL Injection'a bağışıklık sağlamaz, api'yi istediği gibi kötüye kullanabilir. HQL (SQL'in Hazırda Bekletme alt kümesi) hakkında onu daha fazla veya daha az duyarlı hale getiren özel bir şey yoktur.

22) What is criteria API in hibernate?

Hibernate'de, Criteria API dinamik olarak ölçüt sorgu nesneleri oluşturmamıza yardımcı olur. Kriterler, HQL ve yerel SQL sorgularından ayrı olarak başka bir veri alma tekniğidir. Criteria API'nin birincil avantajı, herhangi bir sabit kodlanmış SQL ifadesi kullanmadan verileri işlemek için sezgisel olarak tasarlanmış olmasıdır. Programlı davranış, derleme zamanı sözdizimi denetimi sunar; sonuç olarak, çoğu hata mesajı derleme sırasında teslim edilir. Uygun olmasına rağmen, bu, HQL veya yerel SQL sorguları üzerinde herhangi bir performans geliştirme yeteneği sağlamaz. Aslında, geliştiricilere zevke ve ihtiyaca göre seçim yapmaları için verilen bir tercihtir. Üçünü de kullanabilir veya birine bağlı kalabilirsiniz.

23) What is Erlang? Why is it <<required>> for RabbitMQ?

Erlang, yüksek kullanılabilirlik gereksinimleri olan büyük ölçüde ölçeklenebilir yumuşak gerçek zamanlı sistemler oluşturmak için kullanılan bir programlama dilidir. Kullanımlarından bazıları telekom, bankacılık, e-ticaret, bilgisayar telefonu ve anlık mesajlaşmadır. Erlang'ın çalışma zamanı sistemi, eşzamanlılık, dağıtım ve hata toleransı için yerleşik desteğe sahiptir.

RabbitMQ, Erlang sanal makinesinde ve çalışma zamanında çalışır. RabbitMQ'yu çalıştırmak için uyumlu bir Erlang sürümü kurulmalıdır.

Erlang çalışma zamanı, RabbitMQ tarafından kullanılan bir dizi bileşen içerir.

En önemlileri:

- Erlang sanal makinesi kodu yürütür
- epmd, bir ana bilgisayardaki düğüm adlarını düğümler arası bir iletişim bağlantı noktasına çözümler

24) What is the JPQL?

JPQL, varlık modelinize göre veri tabanı sorguları tanımlamanıza izin veren güçlü bir sorgu dilidir. Yapısı ve sözdizimi SQL'e çok benzer. Ancak size bir JPQL sorgusunun farklı bölümlerinden geçmeden önce belirtmek istediğim önemli bir fark var.

JPQL, bir sorgu tanımlamak için veri tabanı tabloları yerine varlık nesne modelini kullanır. Bu, biz Java geliştiricileri için oldukça rahattır, ancak veri tabanının hala SQL kullandığını unutmamalısınız. Hibernate veya başka herhangi bir JPA uygulaması, JPQL sorgusunu SQL'e dönüştürmek zorundadır. Bu nedenle, oluşturulan SQL ifadelerini kontrol etmek için geliştirme sırasında SQL ifadelerinin günlüğe kaydedilmesini etkinleştirmek iyi bir uygulamadır.

25) What are the steps to persist an entity object?

- Entity Yöneten Factory Object yaratmak

```
EntityManagerFactory emf=Persistence.createEntityManagerFactory("Student_details");
```

- Factory'den entity manager edinme

```
EntityManager em=emf.createEntityManager();
```

- Entity manager'i başlatma

```
em.getTransaction().begin();
```

- Bir veriyi ilişkisel veri tabanında kalıcı kılmak

```
em.persist(s1);
```

- Transaction metodu kapatma

```
em.getTransaction().commit();
```

- Factory kaynaklarının bırakılması

```
emf.close();  
em.close();
```

26) What are the different types of entity mapping?

@OneToOne: Bu tür bir ilişkide, bir attribute, yalnızca bire bir eşleme korunacak şekilde başka bir attribute ile eşlenir. Bu, bir örnek yardımıyla daha iyi anlaşılabilir. Örneğin, bir kişi yalnızca bir departman için çalışıyorsa. Aynı kişi başka bir departmanda çalıştırılmaz, o zaman o mapping birebir olarak adlandırılır.

@OneToMany: Bu tür bir ilişkide, bir attribute başka bir attribute ile eşleştirilir, öyle ki bir attribute diğer birçok attribute ile eşlenir. Bu, bir örnek yardımıyla daha iyi anlaşılabilir. Örneğin, bir öğrenci farklı grupların üyesiyken aynı zamanda kültürel bir grup, spor kulübü, robotik kulübü gibi kulüplere de üye olduğunu düşünelim. Bu durumda öğrenci ve grup ilişkisine **@ManyToOne** bir ilişki denir. @OneToMany ve @ManyToOne karşılıklı çalışır.

27) What are the properties of an entity?

Tüm entityler bir property grubuna sahiptir. Bunlar key-value prensibiyle oluşturulmuş, her key'in karşılığı olacak bir value olarak tanımlanmış sistemlerdir. Örnek olarak bir öğrenciyi ele alırsak bir öğrenci "studentName", "studentNumber" gibi propertylere sahip olacaktır. Bu propertylerin karşılığı ise value yani "değer" olarak isimlendirilir.

28) Difference between CrudRepository and JpaRepository in Spring Data JPA?

JpaRepository CrudRepository'den, CrudRepository ise PagingAndSortingRepository'den türetilmiştir. CrudRepository temel olarak CRUD işlemlerini sağlar. PagingAndSortingRepository, kayıtların sayfalandırılması ve sıralanması için yöntemler sağlar. JpaRepository, kalıcılık bağlamını temizleme ve toplu olarak kayıtların silinmesi gibi JPA ile ilgili yöntemler sağlar. Miras yapıları nedeniyle JpaRepository, CrudRepository ve PagingAndSortingRepository'nin tüm davranışlarına sahip olacaktır. Bu nedenle, JpaRepository ve PagingAndSortingRepository tarafından sağlanan işlevlere sahip olmak için depoya ihtiyacınız yoksa, CrudRepository kullanmak daha mantıklı olabilmektedir.