**1 What is hibernate in java?**

Hibernate ORM (Hibernate in short) is an object-relational mapping tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database. Hibernate also provides data query and retrieval facilities.

**2 Is hibernate better than JDBC?**

JDBC will always give better performance as compared to Hibernate for most of the database vendors. … The choice of hibernate over jdbc and sql queries is not because of the performance but because of reasons mainly object persistence and database independence in terms of not writing database specific queries.

**3 Why do we need hibernate in Java?**

So with JDBC, mapping between Java objects and database tables is done manually. Hibernate reduces lines of code by maintaining object-table mapping itself and returns result to application in form of Java objects. Hibernate, with Transparent Persistence, cache is set to application work space.

**4 What is the use of ORM in Java?**

ORM allows you to use java objects as representation of a relational database. It maps the two concepts (object-oriented and relational) Hibernate is an ORM framework – you describe how your objects are represented in your database, and hibernate handles the conversion.

**5 What is the difference between JPA and Hibernate?**

JPA is the interface, Hibernate is one implementation of that interface. JPA is a specification for accessing, persisting and managing the data between Java objects and the relational database. As the definition says its API, it is only the specification. Hibernate is a JPA provider.

**6 What is the use of Session in hibernate?**

The main runtime interface between a Java application and Hibernate. This is the central API class abstracting the notion of a persistence service. The lifecycle of a Session is bounded by the beginning and end of a logical transaction. (Long transactions might span several database transactions.)

**7 What is the architecture of hibernate?**

Hibernate makes use of the database and configuration data to provide persistence services (and persistent objects) to the application.

**8 What is ORM hibernate?**

Hibernate ORM (Hibernate in short) is an object-relational mapping tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database.

**9 Can we use only JPA without hibernate?**

You have to specify the persistence provider(Hibernate,EclipseLink) in order to use the JPA implementation. The persistence providers have the implementation classes for JPA specifications. You can’t just use JPA, because it is an API =), but there are plenty JPA implementations: EclipseLink.

**10 What are the fetching strategies supported by hibernate?**

Fetch strategies can be declared in the O/R mapping metadata, or over-ridden by a particular HQL or Criteria query. Hibernate defines the following fetching strategies:

Join fetching

Select fetching

Subselect fetching

Batch fetching

Immediate fetching

Lazy collection fetching

“Extra-lazy” collection fetching

Proxy fetching

“No-proxy” fetching

Lazy attribute fetching

**11 What is a polymorphic association?**

Polymorphic association is a term used in discussions of Object-Relational Mapping with respect to the problem of representing in the relational database domain, a relationship from one class to multiple classes. In statically typed languages such as Java these multiple classes are subclasses of the same superclass.

**12 What is the difference between session and Sessionfactory in hibernate?**

SessionFactory is Hibernate’s concept of a single datastore and is threadsafe so that many threads can access it concurrently and request for sessions and immutable cache of compiled mappings for a single database. Sessions are opened by a SessionFactory and then are closed when all work is complete.

**13 What is the difference between load and get method in hibernate?**

Session.load(): It will always return a “proxy” (Hibernate term) without hitting the database. In Hibernate, proxy is an object with the given identifier value, its properties are not initialized yet, it just look like a temporary fake object. If no row found , it will throws an ObjectNotFoundException.

**14 Is hibernate Sessionfactory Singleton?**

SessionFactory is also thread safe so only one thread can execute at a time its code. The instance of sessionFactory is heavyweighted because it contains connection, hibernate configuration, mapping files, location path so if you create number of instance of sessionFactory then your code becomes very heavy.

**15 What are the configuration files in hibernate?**

Hibernate also requires a set of configuration settings related to database and other related parameters. All such information is usually supplied as a standard Java properties file called hibernate.properties, or as an XML file named hibernate.cfg.xml.

**16 What is the use of dialect in hibernate?**

Dialect means “the variant of a language”. Hibernate, as we know, is database agnostic. It can work with different databases. However, databases have proprietary extensions/native SQL variations, and set/sub-set of SQL standard implementations. Therefore at some point hibernate has to use database specific SQL.

**17 What is the use of Show\_sql in hibernate?**

Hibernate has build-in a function to enable the logging of all the generated SQL statements to the console. You can enable it by add a “show\_sql” property in the Hibernate configuration file “ hibernate.cfg.xml “. This function is good for basic troubleshooting, and to see what’s Hibernate is doing behind.

**18 What is hibernate proxy and how it helps in lazy loading?**

A proxy is a subclass implemented at runtime. Hibernate creates a proxy (a subclass of the class being fetched) instead of querying the database directly, and this proxy will load the “real” object from the database whenever one of its methods is called.

**19 Is session is thread safe in hibernate?**

SessionFactory is Hibernates concept of a single datastore and is threadsafe so that many threads can access it concurrently and request for sessions and immutable cache of compiled mappings for a single database. Why to make session object thread safe if we already have a SessionFactory(immutable) object.

**20 What is the use of configuration in hibernate?**

The org.hibernate.cfg.Configuration is used to build an immutable org.hibernate.SessionFactory . The mappings are compiled from various XML mapping files. A org.hibernate.cfg.Configuration also allows you to specify configuration properties.

**21 What is criteria in hibernate?**

In Hibernate, the Criteria API helps us build criteria query objects dynamically. Criteria is a another technique of data retrieval apart from HQL and native SQL queries. The primary advantage of the Criteria API is that it is intuitively designed to manipulate data without using any hard-coded SQL statements.

**22 What is the difference between lazy and eager loading in hibernate?**

All data is fetched when eager marked data in the object when session is connected. However, in case of lazy loading strategy, lazy loading marked object does not retrieve data if session is disconnected (after session.close() statement). All that can be made by hibernate proxy.

**23 Is Sessionfactory immutable?**

The internal state of a SessionFactory is immutable. Most problems with concurrency occur due to sharing of objects with mutable state. Once the object is immutable, its internal state is setted on creation and cannot be changed. So many threads can access it concurrently and request for sessions.

**24 Is Hibernate configuration file mandatory?**

Basically you are setting all the required properties via your properties object so there is no real need to tell Hibernate to look for a hibernate.cfg.xml file which is exactly what the configure() method does. No, it’s not mandatory to use hibernate.cfg.xml. Just don’t use .configure().

**25 What is meant by annotation in hibernate?**

Hibernate annotations are the newest way to define mappings without the use of XML file. You can use annotations in addition to or as a replacement of XML mapping metadata. Hibernate Annotations is the powerful way to provide the metadata for the Object and Relational Table mapping.

**26 What does hibernate.hbm2ddl.auto create means?**

hibernate.hbm2ddl.auto. Automatically validates or exports schema DDL to the database when the SessionFactory is created. With create-drop , the database schema will be dropped when the SessionFactory is closed explicitly.

**27 What is the meaning of persistence in hibernate?**

When a POJO instance is in session scope, it is said to be persistent i.e hibernate detects any changes made to that object and synchronizes it with database when we close or flush the session. And about hibernate.properties and XML Mapping.

**28 How does Hibernate proxy work?**

When a method is invoked on the object, Hibernate will fetch the data from the column and populate the object. This is the proxy mechanism. To add this new behavior (the loading of the data when a method is invoked), Hibernate will create a dynamic subclass of Person using CGLib and add the desired functionality.

**29 What is first level cache in hibernate?**

First level cache is associated with “session” object. The scope of cache objects is of session. First level cache is enabled by default and you can not disable it. When we query an entity first time, it is retrieved from database and stored in first level cache associated with hibernate session.

**30 What is the use of Session in hibernate?**

The main runtime interface between a Java application and Hibernate. This is the central API class abstracting the notion of a persistence service. The lifecycle of a Session is bounded by the beginning and end of a logical transaction. (Long transactions might span several database transactions.)

**31 What is lazy fetching in Hibernate?**

Lazy fetching decides whether to load child objects while loading the Parent Object. You need to do this setting respective hibernate mapping file of the parent class. Lazy = true (means not to load child) By default the lazy loading of the child objects is true.

**32 What is an SQL dialect?**

SQL Dialect. The SQL dialect, derived from the Structured Query Language, uses human-readable expressions to define query statements. Use a SQL query statement with the following ADSI search interfaces: The ActiveX Data Object (ADO) interfaces, which are Automation interfaces that use OLE DB.

**33 What is in HQL?**

Hibernate Query Language (HQL) is an object-oriented query language, similar to SQL, but instead of operating on tables and columns, HQL works with persistent objects and their properties. HQL queries are translated by Hibernate into conventional SQL queries, which in turns perform action on database.

**34 What is the use of projection in hibernate?**

To put it simple, Hibernate Projections are used in order to query only a subset of the attributes of an entity or group of entities you’re querying with Criteria. You can also use Projections to specify distinct clauses and aggregate functions like max , sum and so on. Like modifying the select clause in an SQL query.

**35 What is Lazyinitializationexception in hibernate?**

Indicates access to unfetched data outside of a session context. For example, when an uninitialized proxy or collection is accessed after the session was closed.

**36 What are the different cascade types in hibernate?**

JPA CascadeType – Hibernate native CascadeType

EntityManager Session action – Hibernate native

action – JPA

detach(entity) DETACH evict(entity) DETACH or EVICT

merge(entity) MERGE merge(entity) MERGE

persist(entity) PERSIST persist(entity) PERSIST

refresh(entity) REFRESH refresh(entity) REFRESH

remove(entity) REMOVE delete(entity) REMOVE or DELETE saveOrUpdate(entity) SAVE\_UPDATE replicate(entity, replicationMode) REPLICATE

lock(entity, lockModeType) buildLockRequest(entity, lockOptions) LOCK

**37 What is the use of Show\_sql in hibernate?**

Hibernate has build-in a function to enable the logging of all the generated SQL statements to the console. You can enable it by add a “show\_sql” property in the Hibernate configuration file “ hibernate.cfg.xml “.

This function is good for basic troubleshooting, and to see what’s Hibernate is doing behind.

**38 What is hbm2ddl in hibernate?**

hibernate.hbm2ddl.auto Automatically validates or exports schema DDL to the database when the SessionFactory is created. With create-drop, the database schema will be dropped when the SessionFactory is closed explicitly.

**39 What is the use of dialect in hibernate?**

Dialect means “the variant of a language”. Hibernate, as we know, is database agnostic. It can work with different databases. However, databases have proprietary extensions/native SQL variations, and set/sub-set of SQL standard implementations. Therefore at some point hibernate has to use database specific SQL.

**40 Can I disable first level cache in hibernate?**

The scope of cache objects is of session. Once session is closed, cached objects are gone forever. First level cache is enabled by default and you can not disable it. When we query an entity first time, it is retrieved from database and stored in first level cache associated with hibernate session.

**41 What is criteria in hibernate?**

In Hibernate, the Criteria API helps us build criteria query objects dynamically. Criteria is a another technique of data retrieval apart from HQL and native SQL queries. The primary advantage of the Criteria API is that it is intuitively designed to manipulate data without using any hard-coded SQL statements.

**42 Which one is faster JDBC or hibernate?**

JDBC will always give better performance as compared to Hibernate for most of the database vendors. The choice of hibernate over jdbc and sql queries is not because of the performance but because of reasons mainly object persistence and database independence in terms of not writing database specific queries.

**43 What is the use of bag in hibernate?**

Hibernate Bag is a java collection that stores elements without caring about the sequencing, but allow duplicate elements in the list. A bag is a random grouping of the objects in the list.

**44 What is the use of MappedBy in hibernate?**

With the mappedBy , you directly tell Hibernate/JPA that one table owns the relationship, and therefore it is stored as a column of that table. Without, the relationship is external and Hibernate/JPA need to create another table to store the relationship.

**45 What is inverse true in hibernate?**

The real meaning is that it defines which side is the parent or the relationship owner for the two entities (parent or child). Hence, inverse=”true” in a Hibernate mapping shows that this class (the one with this XML definition) is the relationship owner; while the other class is the child.

**46 What is a bidirectional relationship?**

Bidirectional Relationships. In a bidirectional relationship, each entity has a relationship field or property that refers to the other entity. Through the relationship field or property, an entity class’s code can access its related object.

**47 What is the dirty checking in hibernate?**

Hibernate allows dirty checking feature.All persistent objects are monitored by hibernate.it detects which objects have been modified and then calls update statements on all updated objects.the process of updating the changed object is called automatic dirty checking.

**48 What are the important interfaces in hibernate?**

Session InterfaceSession Factory InterfaceConfiguration InterfaceTransaction InterfaceQuery and Criteria Interface

**49 What is hibernate mapping file?**

The mapping document is an XML document having <hibernate-mapping> as the root element, which contains all the <class> elements. The <class> elements are used to define specific mappings from a Java classes to the database tables.

**50 What is difference between openSession() and getCurrentSession()?**

If you set hibernate.current\_session\_context\_class to thread and then implement something like a servlet filter that opens the session – then you can access that session anywhere else by using the SessionFactory.getCurrentSession().

SessionFactory.openSession() always opens a new session that you have to close once you are done with the operations.

SessionFactory.getCurrentSession() returns a session bound to a context – you don’t need to

close this.

**51 What is difference between Session get() and load() method in Hibernate?**

Session.load() – It will always return a “proxy” (Hibernate term) without hitting the database. In Hibernate, proxy is an object with the given identifier value, its properties are not initialized yet, it just look like a temporary fake object. If no row found , it will throws an ObjectNotFoundException.

Session.get() – It always hit the database and return the real object, an object that represent the database row, not proxy.If no row found , it return null.

**52 What are different states of an entity bean in Hybernate?**

The Entity bean has three states:

Transient Persistent Detached

Transient: When ever we create a new object of Entity bean then we can say that is in Transient state,At that time any modification in the object state does not effect on database.

Persistent: When ever the Object of entity bean associated with session we can say that is in persistent state, if any change in the object state , then that modification effects in database.

Detached :When ever the object is removed from session then it enters in to detached state.Any modification on detached state object , does not effect in database.

**53 Difference between hibernate session merge() vs update()?**

Hibernate handles persisting any changes to objects in the session when the session is flushed. update can fail if an instance of the object is already in the session. Merge should be used in that case. It merges the changes of the detached object with an object in the session, if it exists.

**54 Difference between save() and saveorupdate() in hibernate?**

The important difference between the org.hibernate.Session class methods, save & saveOrUpdate is, save generates a new identifier and results in an INSERT query, whereas saveOrUpdate does an INSERT or an UPDATE. Save method stores an object into the database.

**55 What is the difference between save() and persist() in hibernate?**

persist() – Hibernate persist is similar to save (with transaction) and it adds the entity object to the persistent context, so any further changes are tracked. If the object properties are changed before the transaction is committed or session is flushed, it will also be saved into database.

persist() also guarantees that it will not execute an INSERT statement if it is called outside of transaction boundaries. This is useful in long-running conversations with an extended Session/persistence context.

save() does not guarantee the same, it returns an identifier, and if an INSERT has to be executed to get the identifier (e.g. “identity” generator, not “sequence”), this INSERT happens immediately, no matter if you are inside or outside of a transaction. This is not good in a long-running conversation with an extended Session/persistence context.

**56 What are the collection types in Hibernate?**

Hibernate collections types are:

java.util.List.java.util.Set.java.util.SortedSet.java.util.Map.java.util.SortedMap.java.util.Collection.or write the implementation of org.hibernate.usertype.UserCollectionType.

**57 How to implement Joins in Hibernate?**

Using HQL we can implement joins in hibernate.

HQL Joins – HQL supports inner join, left outer join, right outer join and full join. For example, select e.name, a.city from Employee e INNER JOIN e.address a . In this query, Employee class should have a variable named address.

**58 What is Named SQL Query?**

A named query is a SQL expression represented as a table. In a named query, you can specify an SQL expression to select rows and columns returned from one or more tables in one or more data sources.

**59 What is Hibernate Criteria API?**

In Hibernate, the Criteria API helps us build criteria query objects dynamically. Criteria is a another technique of data retrieval apart from HQL and native SQL queries. The primary advantage of the Criteria API is that it is intuitively designed to manipulate data without using any hard-coded SQL statements.

**60 What are callback interfaces in hibernate?**

Callback interface for Hibernate code. To be used with HibernateTemplate ‘s execution methods, often as anonymous classes within a method implementation. The typical implementation will call Session.load/find/update to perform some operations on persistent objects.

**61 What is transaction management in hibernate?**

In a non-managed environment, Hibernate is usually responsible for its own database connection pool. The application developer has to manually set transaction boundaries (begin, commit, or rollback database transactions) themselves.

**62 What are the mapping associations used in hibernate?**

There are two mapping associations used in hibernate, they are:

1) One-to-One Association

2) Many-to-Many Association

**63 What is Hibernate QBC API?**

Hibernate Query By Criteria (QBC) API is used to create queries by manipulation of criteria objects at runtime.

**64 What is hibernate criteria join?**

Hibernate Criteria JOIN API allows users to perform join operation.

Suppose you have to perform a operation like

SELECT S.\*, C.\* FROM STUDENT S, CONTACT C WHERE S.ROLL\_NO=C.ID;

Then you can write this statement using Criteria join in a very simple way

Criteria criteria = session.createCriteria(Student.class);criteria.setFetchMode(“Contact”, FetchMode.JOIN);List list = criteria.list();

**65 What is the is the default transaction factory in hibernate?**

JDBCTransactionFactory is the default transaction factory in hibernate.

**66 What is JMX in Hibernate?**

Java Management Extensions (JMX) is a Java technology that supplies tools for managing and monitoring applications, system objects, devices (such as printers) and service-oriented networks. Those resources are represented by objects called MBeans (for Managed Bean).

**67 How to bind hibernate session factory to JNDI?**

When binding the SessionFactory to JNDI, Hibernate will use the values of hibernate.jndi.url , hibernate.jndi.class to instantiate an initial context.

**68 What are the fetching strategies in hibernate?**

There are four fetching strategies

1. fetch-“join” = Disable the lazy loading, always load all the collections and entities.

2. fetch-“select” (default) = Lazy load all the collections and entities.

3. batch-size=”N” = Fetching up to ‘N’ collections or entities, \*Not record\*.

4. fetch-“subselect” = Group its collection into a sub select statement.

**69 What are derived properties in hibernate?**

In Hibernate a derived property (also called a calculated property) is a read-only property whose value is calculated at fetch time using SQL expressions.

**70 What is version property in hibernate?**

The <version> property (or @Version annotation) – We know that that Hibernate can provide optimistic locking through a version property on your persistent objects. Furthermore, the version property is automatically managed by Hibernate.

**71 What does session lock () method do in hibernate?**

The lock() method, with LockOptions.NONE, can be used to associate a detached object to a session and put the object back into a persistence context. On top is Hibernate code to reattach a detached object using a typical update method call. On the bottom is code to reattach a detached object using a lock method call.

**72 What does evict do in hibernate?**

evict() To detach the object from session cache, hibernate provides evict() method. After detaching the object from the session, any change to object will not be persisted. The associated objects will also be detached if the association is mapped with cascade=”evict”.

**73 What is implicit polymorphism in hibernate?**

Implicit polymorphism means if a class or interface is used in HQL, criteria or named queries, hibernate fetches the records from the table mapped to the used class along with all the tables mapped to its subclasses, at any hierarchy level. This is one of the great advantage for using hibernate.

**74 What is table per concrete class in hibernate?**

When we use Table Per Concrete class in hibernate, tables are created per class. So there are no nullable values in the table. Disadvantage of this approach is that duplicate columns are created in the subclass tables.

**75 What is light object mapping in hibernate?**

Light Object Mapping is one of the levels of ORM quality in which all entities are represented as classes and they are mapped manually in the Relational Tables.

**76 What is JPA ?**

JPA stands for Java Persistence API. It is a specification on how Java should support the mapping between objects and relational databases for persistence, which is called ORM (Object Relational Mapping). It can be seen as a framework for persistence that is very lightweight and based on POJOs.

Several vendors provide implementations for the JPA. The candidate should be able to point a few, like Hibernate, Toplink, Eclipselink and Apache OpenJPA.

**77 Please explain the concept of POJO Persistence in JPA?**

Historically, many frameworks and tools used for persistence in Java required that classes would extend some abstract class or interface or would impose some other very intrusive requirement.

POJO Persistence is about being able to make any Plain Old Java Object persistent in a non-intrusive way. This is made possible in JPA because it adopts a metadata-driven approach. This metadata can be very easily set using annotations in your POJOS or can be set externally using XML.

**78 What is the EntityManager in JPA?**

JPA is said to be “non-intrusive” but not “transparent persistence”. This means the entities don’t persist themselves magically. There is a need for an API which provides operations to persist and retrieve objects from the database. This is the reason of the EntityManager to exist, its “râison-dêtre”. EntityManager is the main API to provide methods to persist or retrieve data, methods such as persist or find.

The way to obtain an instance of the EntityManager is through an EntityManagerFactory or have it injected for you if you are using the Enterprise Edition.

**79 What is a Persistent Context in JPA ?**

Persistent objects are the ones being managed by the EntityManager. The Persistent Context is a set of entity instances being managed by the EntityManager. The identity is used to determine the uniqueness within the set because only one instance of an entity can exist within a Persistent context.

**80 Could you please explain the concept of Entity in the context of JPA?**

In the context of the Java Persistence API, an Entity means a POJO which has a state, can have relationships and would have its persistence to a relational database managed by some implementation of the JPA.

An Entity is an object that is persistable, it also must have a persistence identity, which means it has a primary key and a row of its own in a database table. There must be a mapping between a database table and the class, which can make extensive use of defaults to required the minimum necessary configuration.

The class can either be annotated with the @Entity annotation or be mapped in an external XML file. Another required annotation (or configuration in XML) is @Id, which marks the field or property (either the field itself or the getter method) responsible for holding the persistence identity (primary key).

**81 What is a Persistence Unit in JPA?**

Persistence Unit is the configuration settings of a logical grouping of user defined persistable classes that an EntityManager is supposed to manage.

It is in a file called persistence.xml that we can describe the persistence unit. One single file can contain one or more persistence units. The persistence unit is named to create an EntityFactory. We are required to provide the Persistence Unit name in order for that configuration to be used.

**82 What is the JPQL?**

JPQL stands for Java Persistence Query Language. It is a very powerful query language, similar to SQL, to execute queries across entities and their relationships instead of directly against the data model.

You might be asking yourself why in the world would you need a Query Language given you could be writing SQL directly?

The answer is that it has two major advantages over writing SQL directly: The first one is that JPQL has an object-oriented syntax. The second one, which is kind of a consequence from the first is that JPQL is also not bound to a database schema, it uses a schema abstraction based on the object model instead of the relational one.

This code assumes you have an instance of EntityManager in the variable entityManager. Than it creates a query instance and asks for the result:

TypedQuery<Person> query = entityManager.createQuery(“SELECT p FROM Person p”, Person.class);

List<Person> people = query.getResultList();

The JPQL presents some features (not exclusive):

It can produce Single or Multiple value result types

It has aggregate functions, sorting and grouping clauses

Support for join, both inner and outer

Update and delete

**83 What is a named query in the JPA? How to set and execute it?**

In the context of JPQL, Named queries are a way to organize query definitions associating them with names. It is important to have in mind that the names must be unique within a single Persistent Unit.

There are two ways to set Named queries, either using annotation or setting them in external files.

The easiest way to set them is by using the @NamedQuery annotation in the class definition of any Entity. The only drawback is that it somehow “pollutes” the class with the queries.

@NamedQuery(name=”School.findByName”, query=”SELECT s FROM School s WHERE s.name = :schoolName”)

The other option, the use of external file.

Example (I’ve removed a lot of XML details to make the main construct clearer):

<?xml version=”1.0″ encoding=”UTF-8″?>

<entity-mappings …>

<named-query name=”School.findByName”>

<query>SELECT s FROM School s WHERE s.name = :schoolName</query>

</named-query>

</entity-mappings>

And in the persistence.xml, using the “mapping-file” tag, the above file can be referenced.

To add several named queries is pretty straightforward in the external file, just add as many “named-query” tags as necessary. When using annotations, though, it is still very easy but you need to remember that the @NamedQuery annotation has to be nested within the @NamedQueries annotation:

@NamedQueries({

@NamedQuery(name=”…”, query=”…”),

@NamedQuery(name=”…”, query=”…”),

@NamedQuery(name=”…”, query=”…”)

})

The last point the candidate should mention is how to execute the named query. It is necessary to invoke the instance method createNamedQuery from the

EntityManager: em.createNamedQuery(“School.findByName”, School.class).setParameter(“name”, name).getSingleResult();

if the query is expected to return more than one value, use the method .getResultList() instead of the .getSingleResult()

**84 What is paging and how to achieve it in JPA?**

A common problem with large result sets is that they can be inefficient to display and or to load. The applications must be able to cope with this problem and a common technique is to present the data in fixed size batches called pages, the paging also provides a way to navigate the results, retrieving other pages.

Pagination is made very easy with JPA, the two interfaces related to queries, Query and TypedQuery have the methods setFirstResult() to set what is the first element of the given page and setMaxResults() to determine the page or size.

**85 What happens if a query is executed against entities that have uncommitted changes?**

One might be tempted to think that the changes that are not yet committed would not visible to the queries. This possible explanation could stem from the fact that queries are executed directly against the database, not using the internal cache.

However, the persistence provider ensures, by default, that queries can see pending transactional changes in their resultset. To have more control, both EntityManager and Query interfaces provide a way to set the flush mode, which is used by the provider to determine how to handle pending changes and queries.

The options available are AUTO (default option) and COMMIT. The first tells the provider that the pending changes should be included in the results. The second option allows the provider to just ignore uncommitted changes.

**86 What is the Java Criteria API?**

Retrieving entities is one of the most important functions of an ORM technology. For that, JPA offers two different ways to express queries, the query languages (JPQL and native SQL) and the Criteria API.

This API, part of the JPA, allows the creation of dynamic queries. One of its advantages over JPQL is that it can be validated at compile time, so problems can be discovered much earlier.

**87 Please explain the Life cycle of an Entity within JPA ?**

Remember an Object is considered Managed if it is associated with an Entity Manager. So, once you create an object it is considered transient (not managed by an Entity Manager and no representation in the database). If this object is persisted, it becomes managed.

There are other ways to get to a managed state, like when you retrieve objects from the database, such as using find, or when you have a detached (we are going to see this one soon) that you merge.

Detached: These are objects that are not managed by the EntityManager, but they are persisted. In other words, they have are stored in the database. These objects can return to a MANAGED state if it is merged back. The EntityManager’s merge method is invoked with the object as a parameter.

**88 What is lazy loading in JPA?**

Performance is a very important topic when loading the relationships of an Entity. Lazy loading represents a mechanism used by many ORMs, such as JPA for Java or Entity Framework for .Net to load information just when they are actually demanded.

Although the objective of Lazy Loading is to improve performance, the wrong use of Lazy or Eager strategies can significantly be detrimental to it.

The strategies JPA uses by default are almost always the best choice. Unfortunately, this is not always the case. So one should know what are the defaults and its weakness.

Lazy strategy, on the other hand, is used by default with relationships of the types one-to-many or many-to-many. This means that the relationships will not be loaded immediately with the entity, it will wait for an accessor method to be invoked, such as a getter. Once the method is invoked, it executes a query against the database to finally load the data.

Look for the code below, where School has many students, in a one-to-many relationship, therefore using Lazy Loading by default. There is a performance problem with this code:

for(School school : schools) {

for(Student student : school.getStudents(){

System.out.println(student.getID());

}

}

It will be executed in what is known as N+1 queries, which is a known problem in the using JPA. In this particular case, having an Eager strategy could have had better performance.

**89 What are Embeddable classes? How to create them?**

An Embeddable class represents part of an Entity persistent data but it does not contain a persistent identity of their own. Instances of an embeddable class share the identity of the entity that owns it.

@Entity

public class Order {

@Id

protected long id

@Embedded

Payment payment;

}

@Embeddable

public class Payment {

private String type;

private String currency;

private double amount;

}

**90 What are Persistent Fields?**

In the case of JPA, it does not even require that the object being mapped follow the JavaBeans convention. The fields, except public ones, can be directly mapped to a column in a table in the database.

For field access to be used, it is just a matter of annotating the one single field of the entity for JPA to use field access. The other fields become persistent by default, which means they too get mapped to columns of the same table as the Entity. If a field should not be persistent, one must either annotate it with @Transient or simply use the Java transient modifier.

**91 What are Persistent Properties?**

Persistent properties are the mapping between setter/getter from an Object that follows the JavaBean conventions to columns on a table in a relational database.

It is required that the annotations be applied to the getter methods. The same rules about the @Transient annotation or transient modifier apply here.

**92 Please, briefly explain Multiplicity in Entity Relationships and what are its possible types?**

Entities have relationships with other entities, and a good ORM has to be possible to map these relationships to the database.

There are two important concepts related to Entity Relationships, namely Role, and Cardinality. A student has a relationship with a school, he or she plays the role of studying in that School. The cardinality refers to a number of entities exist on each side of a relationship. Let’s say that ordinarily, a student has one school, but a school has many students.

It is the cardinality of both source and target roles that are used to name the mappings.

We have, so, four types of multiplicities, that we can divide into two groups.

**Single-Valued Associations**

When the target role has a cardinality of one, it is about single-valued associations.

**One-to-one:**

The cardinality on both sides of the relationship is one. This would be the case for a School with a Principal, where the School has only one person playing the role of Principal, and this person plays this role for a single School.

In the School entity, there would be a field or property to store a Principal. The Principal probably also have a School entity field or property.

The easiest way to signal this association is by using the @OneToOne annotation on the persistent field or property.

**Many-to-one:**

The cardinality on the source side is many and on the target side of the relationship is one. From the perspective of a Student, the relationship with School is many to one.

The easiest way to signal this association is by using the @ManyToOne annotation on the persistent field or property.

**Collection-Valued Associations**

When the source role has a cardinality of one, it is about collection-valued associations.

**One-to-many:**

This is the case of source role having a cardinality of one and target source of many. Still, in the example of School, it has many Students or many ClassRooms.

The easiest way to signal this association is by using the @OneToMany annotation on the persistent field or property.

**Many-to-many:**

This is the case where both source and target roles having a cardinality of many. An example would be that each Teacher has many Students, and each Student has many Teachers.

The easiest way to signal this association is by using the @ManyToMany annotation on the persistent field or property.

**93 What is the difference between “@JoinColumn” and “mappedBy” Foreign Key Constraints?**

Imagine a relationship of one to many, like a School which has many Students.

The most common way to represent this in the database would be to use a foreign key like “schoolID” in the Students table.

If we follow this, then we would have:

@Entity

public class School {

@OneToMany(mappedBy = “student”)

private List<Student> departments;

}

@Entity

public class Student {

@ManyToOne

@JoinColumn(name = “schoolID”)

private School school;

}

The side of the relationship (considering the data scheme) that has the foreign key or the “join column”, is considered as “owner” of the relationship. It is important to understand this because the annotations are always defined on the owning side of the relationship. In our example, the Student owns the relationship, so it is in this class that the annotation @JoinColumn must be defined.

The mappedBy, on the other hand, is used in the non-owning or inverse side of the relationship.

**94 Please explain what are Compound Primary Keys.**

It is not always possible or desirable to have a single field to represent the identity of an entity. The primary key is composed of more than a single field. In the database, it means the Primary Key is composed of more than a single column.

To represent a compound primary key in an Entity it is necessary to have a separate class, called Primary Key Classes, containing the primary key fields. These classes must define the methods equals() and hashCode().

An example of the compound key might be from a Course entity, where it can be formed from the topic and the level, as CIVIL\_LAW-101 and CRIMINAL\_LAW-101

**95 What are the differences between EntityManager.merge() and EntityManager.persist()?**

The merge method can both update and existing entity and insert a new one, while the persist method only inserts an entity.

The question that arises is why would one choose to use the persist method, given that merge apparently can do both saving and updating.

The main difference is in the treatment your Entity will receive after the methods are invoked.

The persist method will insert your entity and make the entity instance itself managed, while merge will either insert or update your Entity, but it makes a copy of your instance and that copy becomes managed.

**96 When to override “equals() & hashcode()” method in an Entity class ?**

Whenever you need to have Entity objects within collections such as Set, List etc the identity of objects become an issue because once an object is in the collection, the behavior of changing its identity is probably not what you want.

Entities that represent the same Entity should be considered equal and return, therefore, the same hashcode. The most straightforward implementation would be to just use the id in the implementations of both equals() and hashcode(). The problem with this approach is that it would change identity once the object is persisted, after all, Entities in general just receive their ids upon insertion. If you implement an equals method that does something while id is null and something else after it is set, which would be a possible first solution that comes to mind, the identity of this Entity would change once it receives an id.

Like we said, identity change is very problematic for instances in collections. Hashcode should always return the same value for the same identity and collections make use of the hashes to store the items internally.

Let’s look at an example in code:

Set<School> schools = new HashSet<>();

School s = new School();

schools.add(s);

schools.contains(s); //returns true. Same hash as when inserted in set, right?

s.setID(5); //this would happen if it had been persisted

schools.contains(s); //returns false. Different hash now.

So it is fundamental that the implementation of equals() and hashCode() is valid both before and after the object is persisted.

The natural answer is to have an id generated as soon as the entity first created.

**97 Explain Directionality in JPA Relationships?**

Entities can have relationships with other entities. For this to happen, at least one side of the relationship must have an attribute that refers to the other entity.

If only one side knows about the relationship, then the relationship has only one direction, it is called unidirectional.

If on the other hand, both sides of the relationships have attributes pointing to each other, this relationship is called a bi-directional. In the end, a Bidirectional relationship is nothing more than a pair of unidirectional relationships.

The relationship between Student and Locker can be modeled as a bi-directional one-on-one relationship.

A bi-directional role has to follow some rules (check them all at here at the Oracle’s tutorial page). The most important one to remember is that the owned side of the relationship must use the “mappedBy” element to point to the owning side (the one which contains the foreign key).

**98 What are the different types of cascade Operations for Entities?**

The following are the different types of cascade operations available for Entities:

CascadeType.PERSIST – Triggered when the method persist from EntityManager is invoked and an

Entity is persisted;

CascadeType.DETACH – Triggered when an entity is removed from Persistence Context. The related

entity will be detached as well.

CascadeType.MERGE – Triggered when a change is made to an Entity, either when the transaction in

which the change happened ends or when the merge() method is invoked. The related entity will be merged as well.

CascadeType.REFRESH – Triggered when an entity is making up-to-date by receiving information

from the database; The related entity will be refreshed as well.

CascadeType.REMOVE – Triggered when an entity is removed from the database. The marked

relationships will be removed as well.

CascadeType.ALL – All prior triggers will be reflected on the related entities.

**99 Explain Web Services?**

A Web Service can be defined as an application component for communication or say exchanging information between two applications over the network. Web services basically work on client server model where web services are easily accessible to client applications over the network.

To enable communication between various applications, web services take the help of open standards like XML (for data tagging), SOAP (for message transferring and WSDL (to denote service availability).

**100 What are the components of web service?**

The different components of web services are

• SOAP- Simple Object Access Protocol

• UDDI- Universal Description, Discovery, and Integration

• WSDL- Web Service Description language

• RDF- Resource Description Framework

• XML- Extensible Markup Language

**101 Explain the term Interoperability with respect of Web services?**

The term ‘Interoperability’ is widely used in product marketing description which defines the ability of different products or systems working together without any special effort from the customer part.

This is applicable in the same way when we talk about ‘Interoperability’ in terms of web services. Here it determines the communication between various applications, sharing of data as well as services among themselves. There is no restriction on the type of application to be in communication. If any code is written, it will be treated as generic code that will be understood by all application. Thus, the cost of writing specific codes for each application is reduced.

There is no restriction on the type of application to be in communication. If any code is written, it will be treated as generic code that will be understood by all application. Thus, the cost of writing specific codes for each application is reduced.

**102 Define web service protocol stack and its layers?**

Web service protocol stack consists of 4 layers. This can be described as follows

1 Service transport: This is the first layer which helps in transporting XML messages between various client applications. This layer commonly uses the below-mentioned protocols:

• HTTP(Hypertext Transport Protocol)

• SMTP(Simple Mail Transport Protocol)

• FTP(File Transfer Protocol)

• BEEP(Block Extensible Exchange Protocol)

2 XML messaging: This layer is based on the XML model where messages are encoded in common XML format which is easily understood by others. This layer includes

• XML-RPC

• SOAP(Simple Object Access Protocol)

3 Service description: This layer contains description like location, available functions, and data types for XML messaging which describes the public interface to a specific web service. This layer includes:

• WSDL(Web Service Description Language)

4 Service discovery: This layer is responsible for providing a way to publish and find web services over the web. This layer includes:

• UDDI(Universal Description, Discovery, and Integration)

**103 Explain web service architecture?**

Web service framework consists of an architecture which consists of three different layers. The roles of these layers are defined as below

• Service Provider: As the name denotes, service provider role is to create the web service and makes it accessible to the client applications over the internet for their usage.

• Service Requestor: Service requestor is basically any consumer of web service like any client application. Client applications are written in any language contact web service for any type of functionality by sending XML request over the available network connection.

• Service Registry: Service registry is the centralized directory which helps locate web services for client applications. Here we can find the existing web services, as well as developers, can also create the new one.

The Service Provider uses the ‘Publish’ interface of Service Registry to make the existing web services available to client applications. With all the information provided by the service registry, service requestor is able to bind or invoke services.

**104 What do you understand by XML-RPC?**

RPC is Remote Procedure Call and as the name suggests, it is the method of calling a procedure or function available on any remote computer.

XML stands for Extensible Markup Language. Thus XML-RPC represents a simple protocol that performs RPCs by using XML messaging. This has been considered as an excellent tool for connecting different environments and also establishing connections between wide varieties of computers.

**105 Explain features of XML-RPC?**

The major features of XML-RPC are enlisted below

• RPCs are performed using simple XML language.

• XML encoded Requests are sent via HTTP POST.

• XML Response is embedded in HTTP response.

• It is considered as platform-independent.

• It allows communication between diverse applications.

• It uses HTTP protocol for passing information between client and server computers.

• It has small XML vocabulary for describing request and response’s nature.

**106 Enlist few advantages of web services?**

We have already discussed web services, its architecture, components. Now, let us see some its advantages

* Every application is now on the internet and it the web service which provides some sort of required functionality to the client applications.
* Web services help in exposing the existing functionalities over the network to help other applications to use in their programs.
* It has features like ‘Interoperability’ which determines the communication between various applications, sharing of data as well as services among themselves.
* Web services use the standardized web service protocol stack for communication which consists of 4 layers namely, Service Transport, XML messaging, Service description and Service discovery.
* It has the feature of the low cost of communication because of the usage of SOAP (Simple Object Access Protocol) over HTTP protocol.
* Easy to deploy, integrate and is reusable.
* Allows simple integration between different feature as a part of loose coupling feature.

**107 Explain the term UDDI with its features?**

UDDI is an XML-based standard in the service discovery layer of web service protocol stack. It is used for publishing and finding web services over the web as it acts like a directory. Some of the features of UDDI are explained below

• It is an open framework and is platform independent.

• SOAP, COBRA, and Java RMI protocols are used for communication.

• It helps businesses to discover each other and enable interaction between them over the internet.

• It acts as a database containing all WSDL files.

**108 Which language is used by UDDI?**

UDDI uses the language known as WSDL (Web Service Description Language)

**109 Explain BEEP?**

BEEP stands for Blocks Extensible Exchange Protocol. BEEP is determined as building new protocols for the variety of applications such as instant messaging, network management, file transfer etc. It is termed as new Internet Engineering Task Force (IETF which is layered directly over TCP. It has some built-in features like

• Authentication

• Security

• Error handling

• Handshake Protocol

**110 Enlist few tools used to test web services?**

To test Web services, below-mentioned tools are used

• SoapUI

• REST client

• JMeter

**111 Do we require any special application to access web service?**

The only requirement for accessing web services from any application is that it must support XML-based request and response. There is no need or say the requirement of installing any application for accessing web services.

**112 What do you know about RESTful Web Services?**

REST stands for Representational State Transfer. REST is defined as the stateless client-server architectural style for developing application accessed over the web. When web services use HTTP methods to implement the concept of REST architecture, then it is known as RESTful Web services. In this architectural style, data and functionality are served as resources and is accessed by URI (Uniform Resource Identifiers).

RESTful web services enable web services to work best by inducing properties like

• Performance

• Scalability

• Modifiability

**113 Explain the advantages of RESTful web services?**

Enlisted below are the advantages of RESTful web services

* They are considered as language and platform independent as these can be written in any programming language and can be executed on any platform.
* REST is lightweight protocol and is considered as fast because of less consumption of bandwidth and resources.
* It supports multiple technologies and different data formats like plain text, XML, JSON, etc.
* It has loosely coupled implementation and can be tested easily over browsers.

**114 Differentiate SOAP and REST?**

Difference between SOAP and REST can be easily understood from the below table

**SOAP**

Simple Object Access Protocol (SOAP serves as a standard protocol for web service creation.

Web services and clients are tightly coupled and define some standards that are to be strictly followed.

It requires more bandwidth and resource as well as uses service interfaces for exposing business logic.

It is usually less preferred and permits XML data format only.

Java API for SOAP web service is JAX-WS.

SOAPUI can be used for testing SOAP web services.

It defines its own security and uses WSDL contract for binding web services and client programs.

**REST**

Representational State Transfer (REST is an architectural style for web service creation.

It does not follow too many standards and is loosely coupled.

It requires less bandwidth and resource as well as uses URI (Uniform Resource Identifiers for exposing business logic.

It is usually more preferred and permits data formats like Plain text, HTML, JSON, etc.

Java API for RESTFUL web service is JAX-RS.

Browsers and extensions such as Chrome postman are used for testing REST web services.

It does not have any defined contract as well as does not have its own security methods.

**115 Explain different HTTP methods supported by RESTful web services?**

Enlisted below are some common HTTP methods along with their functions that are supported by RESTful web services

• GET: Read-only access to the resource.

• PUT: Creation of new resource.

• DELETE: Removal of a resource.

• POST: Update of an existing resource.

• OPTIONS: Get supported operations on the resource.

• HEAD: Returns HTTP header only, nobody.

**116 What is a resource in RESTful web service and how it is represented?**

Resource is said to be a fundamental concept having a type and relationship with other resources. In REST architecture, each content is considered as the resource and they are identified by their URIs.

Resources are represented with the help of XML, JSON, text etc in RESTful architecture.

**117 What are the core components of HTTP request and HTTP response?**

**HTTP request has following 5 major components**

HTTP Requests Meaning/work

Verb Indicate HTTP methods like GET, PUT, POST, etc

URI Identifies the resource on server

HTTP Version Indicates version.

Request Header

Contains metadata like client type, cache settings, message body format, etc for HTTP request message.

Request Body Represents content of the message.

**HTTP response has following 4 major components**

HTTP Response Meaning/work

Status/Response code Indicates the status of the server for requested resource.

HTTP version Represents HTTP version.

Response Header

Consists of metadata like content length, content type, server length, etc for HTTP response message.

Response Body Represents response message content.

**118 What is the purpose and format of URI in REST architecture?**

Purpose of URI is to locate resources on the server that are hosting web services.

Format of URI

<protocol>://<service-name>/<ResourceType>/<ResourceID>

**119 Explain the term statelessness in terms of RESTful web services?**

In REST architecture, there is a restriction where a REST web service is not allowed to keep a client state on the server. Such condition is known as ‘Statelessness’. In such situation, the client passes its context to the server and in turn, the server stores the context in order to process client’s further requests.

**120 Enlist advantages and disadvantages of statelessness?**

The advantages of statelessness include

Each and every method requests are treated independently.

Application design is simplified as it does not maintain client’s previous interaction.

Works with HTTP protocol as it shares the feature of being statelessness.

The disadvantage of statelessness includes

Every time client interaction takes place, web services are to be provided with extra information about each request so that they can interpret the client’s state.

**121 For designing a secure RESTful web service, what are the best factors that should be followed?**

As HTTP URL paths are used as a part of RESTful web service, so they need to be secured. Some of the best practices include the following

* Perform validation of all inputs on the server from SQL injection attacks.
* Perform user’s session based authentication whenever a request is made.
* Never use sensitive data like username, session token password, etc through URL. These should be passed via POST method.
* Methods like GET, POST, PUT, DELETE, etc should be executed with proper restrictions.
* HTTP generic error message should be invoked wherever required.

**122 Define SOAP web services?**

Simple Object Access Protocol (SOAP is defined as the XML based protocol which is known for designing and developing web services as well as enabling communication between applications developed on different platforms with different programming languages over the internet. It is both platform and language independent.

**123 What are the various approaches available for developing SOAP based web services?**

There are basically 2 different approaches available for developing SOAP-based web services. These are explained as follows

* Contract-first approach: In this approach, the contract is defined first by XML and WSDL and then java classes are derived from the contract.
* Contract-last approach: In this approach, java classes are defined first and then the contract is generated which is usually the WSDL file from the java class.

“Contract-first” method is the most preferred approach.

**124 Explain the major obstacle faced by SOAP users?**

One of the major hindrance observed by users of SOAP is the ‘Firewall security mechanism’. In this case, all the HTTP ports except those which bypass firewall are locked. In some cases, a technical issue of mixing specification of message transport with message structure is also observed.

**125 What are the advantages and disadvantages of SOAP?**

**Enlisted below are advantages of SOAP web services**

* SOAP allows communications between various applications and it is both language and platform independent.
* It is very simple as well as uses standard HTTP protocol and XML for sending and receiving messages.
* It defines and uses its own security known as WS security.
* It decouples the encoding and communication protocol from the runtime environment.
* It eradicates firewall problems and is vendor neutral.
* It allows circulation of messages in distributed and decentralized environment.

**Enlisted below are disadvantages of SOAP web services**

* Lightweight formats other than XML are not supported.
* Not easily testable on browsers.
* Security facilities are not present.
* SOAP is slow and cannot be easily tested on the browser.
* Web services and clients are tightly coupled and define some standards that are to be strictly followed.

**126 What are the elements of a SOAP message?**

SOAP is just like other XML document and has following elements

* Envelope: This element is defined as the mandatory root element. It translates the XML document and determines the start and end of the SOAP message.
* Header: This element contains the optional header attributes of the message that contains specific information of the application. This element can occur multiple times and are intended to add new features and functionalities.
* Body: This element is mandatory and contains the call and response messages. It is also defined as the child element of the envelope containing all the application derived XML data that has been exchanged as a part of SOAP message.
* Fault element: Errors that occur during processing of the messages are handled by the fault element. If the error is present, then this element appears as a child element of the body. However, there can only be one fault block.

**127 What are the important characteristics of SOAP envelope element?**

We have seen the basic work of a SOAP envelope element in the previous answer, now let us see some of its characteristics

* SOAP envelope is a packaging mechanism.
* Every Soap message has a mandatory root envelope message.
* Only one body element is allowed for each envelope element.
* As the SOAP version changes, envelope changes.
* If the header element is present, it should appear as the first child.
* Prefix ENV and envelope element is used for specification.
* A namespace and an optional encoding style are used in case of optional SOAP encoding.

**128 Enlist few syntax rules applicable for SOAP message?**

Enlisted below are some important syntax rules that are applicable for SOAP message

A SOAP message

• Must be encoded using XML.

• Must use the SOAP envelope namespace.

• Must use the SOAP encoding namespace.

• Must not contain the DTD reference.

• Must not contain XML processing instructions.

**129 Define SOA?**

A Service Oriented Architecture (SOA is basically defined as an architectural pattern consisting of services. Here application components provide services to the other components using communication protocol over the network. This communication involves data exchanging or some coordination activity between services.

Some of the key principles on which SOA is based are mentioned below

* The service contract should be standardized containing all the description of the services.
* There is loose coupling defining the less dependency between the web services and the client.
* It should follow Service Abstraction rule, which says the service should not expose the way functionality has been executed to the client application.
* Services should be reusable in order to work with various application types.
* Services should be stateless having the feature of discoverability.
* Services break big problems into little problems and allow diverse subscribers to use the services.

**130 Explain the actions performed by SOAPUI?**

SOAPUI is an open-source, free and cross-platform functional testing solution. Mentioned below are some actions performed by SOAPUI

* It can help create functional, security and load testing test suites.
* Data driven testing and scenario based testing is also performed.
* It has the ability to impersonate web services as well as has got built-in reporting abilities.
* Web Services Security

**131 What are the primary security issues of web service?**

To ensure reliable transactions and secure confidential information, web services requires very high level of security which can be only achieved through Entrust Secure Transaction Platform. Security issues for web services are broadly divided into three sections as described below

1 Confidentiality: A single web service can have multiple applications and their service path contains a potential weak link at its nodes. Whenever messages or say XML requests are sent by the client along with the service path to the server, they must be encrypted. Thus, maintaining the confidentiality of the communication is a must.

2 Authentication: Authentication is basically performed to verify the identity of the users as well as ensuring that the user using the web service has the right to use or not? Authentication is also done to track user’s activity. There are several options that can be considered for this purpose

• Application level authentication

• HTTP digest and HTTP basic authentication

• Client certificates

3 Network Security: This is a serious issue which requires tools to filter web service traffic.

**132 What do you know about foundation security services?**

Foundation security services consist of the following

• Integration

• Authentication

• Authorization

• Digital Signatures

• Encryption processes

**133 What is Entrust Identification Service?**

Entrust Identification Service is categorized under Entrust Secure Transaction Platform which provides essential security capabilities to ensure secure transactions. This usually allows companies to fully control the identities that are trusted to perform web service transactions.

**134 What is Entrust Entitlements Service?**

Entrust Entitlement service is those whose task is to verify the services that are attempting to access the web services. It basically ensures security in business operations as well as some authentication services.

**135 What is Entrust Privacy Service?**

As the name suggests, Entrust Privacy Service perform encryption of the data so that only concerned parties are able to access the data. It basically deals with two factors

• Confidentiality

• Security

**136 Explain WSDL?**

WSDL stands for Web service Description Language. It is a simple XML document which comes under the Service Description layer of Web Service Protocol Stock and describes the technical details or locates the user interface to web service. Few of the important information present in WSDL document are

• Method name

• Port types

• Service end point

• Method parameters

• Header information

• Origin, etc

**137 What are the different elements of WSDL documents?**

The different elements of WSDL document along with brief description is enlisted below

* Types: This defines the message data types, which are in the form of XML schema, used by the web services.
* Message: This defines the data elements for each operation where messages could be the entire document or an argument that is to be mapped.
* Port Type: There are multiple services present in WSDL. Port type defines the collection of operations that can be performed for binding.
* Binding: Determines and defines the protocol and data format for each port type.
* Operations: This defines the operations performed for a message to process the message.

**138 Explain the message element in WSDL?**

Message element describes the data that has been exchanged between the consumer and the web service providers. Every web service consists of two messages and each message has zero or more <part> parameters.

The two messages are

• Input: Describes the parameter for the web service

• Output: Describes the return data from the web service.

**139 Enlist the operation types response used in WSDL?**

WSDL basically defines 4 types of Operation type responses. These are enlisted below

* One-way: Receives a message but does not return response.
* Request-Response: Receives a request and return a response.
* Solicit-Response: Sends a request and wait for a response.
* Notification: Sends a message but does not wait for a response.

Among these, Request-Response is the most common operation type.

**140 Is binding between SOAP and WSDL possible?**

Yes, it is possible to bind WSDL to SOAP. The binding is possible by basically two attributes

• Name: Defines the name of the binding.

• Type: Defines the port for the binding.

For SOAP binding, two attributes need to be declared

• Transport: Defines the SOAP protocol to be used i.e. HTTP.

• Style: This attribute can be ‘rpc’ or ‘document’.

**141 Explain <definition> element?**

Definition element is described as the root of WSDL document which defines the name of the web service as well as act as a container for all the other elements.

**143 What are the two attributes of <Port> element in WSDL?**

Every port element is related to a specific binding by defining an individual endpoint. The port element has following two attributes

* Name: This attribute provides the unique name within the WSDL document.
* Binding: This attribute refers to the process of binding which has to be performed as per the linking rules defined by WSDL.

**144 What are the points that should be considered by ports while binding?**

WSDL allows extensibility elements which are used to specify binding information. Below are few important points that should be kept in consideration while binding.

A port must not

• Specify more than one address.

• Specify any binding information other than address information.

**145 What are web services ?**

According to oracle docs, web services can be defined as

Web services are client and server applications that communicate over the World Wide Web’s (WWW HyperText Transfer Protocol (HTTP). Web services provide a standard means of inter operating between software applications running on a variety of platforms and frameworks.

Main characteristics of the Web Services are :

1. Interoperability

2. Extensibility

3. Machine processable descriptions.

for example in simple words , when we call somebody so the person dialing and calling is the client application , while person receiving the call is server application and "hello" word is the protocol as similar to HTTP request .

**146 What is the difference between SOA and a web service?**

SOA (Service-Oriented Architecture is an architectural pattern that makes possible for

services to interact with one another independently.

Web Services is a realization of SOA concept, that leverages XML, JSON, etc. and common Internet protocols such as HTTP(S), SMTP, etc.

SOA is a system-level architectural style that tries to expose business. WOA is an interface-level architectural style that focuses on the means by which these service capabilities are exposed to consumers.

**147 What is SOAP?**

SOAP (Simple Object Access Protocol is a transport protocol for sending and receiving requests and responses on XML format, which can be used on top of transport protocols such as HTTP, SMTP, UDP, etc.

**148 What is REST?**

REST (REpresentational State Transfer is an architectural style by which data can be transmitted over transport protocol such as HTTP(S).

**149 What is the difference between a REST web service and a SOAP web service?**

Below are the main differences between REST and SOAP web service

* REST supports different formats like text, JSON and XML; SOAP only supports XML;
* REST works only over HTTP(S on a transport layer; SOAP can be used different protocols on a transport layer;
* REST works with resources, each unique URL is some representation of a resource; SOAP works with operations, which implement some business logic through different interfaces;
* SOAP based reads can’t be cached, for SOAP need to provide caching; REST based reads can be cached;
* SOAP supports SSL security and WS-security(Web Service-security); REST only supports SSL security;
* SOAP supports ACID (Atomicity, Consistency, Isolation, Durability); REST supports transactions, but it is neither ACID compliant nor can provide two phase commit.

**150 How to decide which one of web service to use REST or SOAP?**

“REST vs SOAP” we can rephrased to "Simplicity vs Standard". Of course, "Simplicity" with REST at most cases wins, it wins in performance, scalability and support for multiple data formats, but SOAP is favored where service requires comprehensive support for security (WS-security and transactional safety (ACID).

**151 What is WSDL?**

WSDL (Web Services Description Language is an XML format for describing web services and how to access them.

**152 What is JAX-WS?**

JAX-WS (Java API for XML Web Services is a set of APIs for creating web services in XML format.

**153 What is JAXB?**

JAXB (Java Architecture for XML Binding is a Java standard that defines how Java objects are converted from and to XML. It makes reading and writing of XML via Java relatively easy.

**154 Can we send soap messages with attachments?**

Yes, we can send different formats such as PDF document, image or other binary file with soap messages as an attachment. Messages send using the binary data. SOAP messages is attached with MIME extensions that come in multipart/related.

An example:

MIME-Version: 1.0

Content-Type: Multipart/Related; boundary=MIME\_boundary; type=text/xml;

start="<claim061400a.xml@ javahungry.com>"

Content-Description: This is the optional message description.

<?xml version='1.0' ?>

<SOAP-ENV:Envelope

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Body>

..

<theSignedForm href="cid:claim061400a.tiff@javahungry.com"/>

..

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

--MIME\_boundary

Content-Type: image/tiff

Content-Transfer-Encoding: binary

Content-ID: <claim061400a.tiff@javahungry.com>

...binary TIFF image...

--MIME\_boundary—

**155 What is MTOM?**

MTOM (Message Transmission Optimization Mechanism is a mechanism for transmitting large binary attachments with SOAP messages as raw bytes, allowing for smaller messages.

**156 What is XOP?**

XOP (XML-binary Optimized Packaging is a mechanism defined for the serialization of XML Information Sets that contain binary data, as well as deserialization back into the XML Information Set.

**157 What is a SOAP envelope element?**

SOAP envelop element is the root element of a SOAP message which defines the XML document as a SOAP message.

An example:

<?xml version="1.0"?>

<soap:Envelope

xmlns:soap="http://www.w3.org/2001/12/soap-envelope"

soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

...

Message information

...

</soap:Envelope>

**158 What does a SOAP namespace defines?**

SOAP namespace defines the Envelope as a SOAP Envelope.

An example:

xmlns:soap=http://www.w3.org/2001/12/soap-envelope

**159 What is the SOAP encoding?**

SOAP encoding is a method for structuring the request which is suggested within the SOAP specification, known as the SOAP serialization.

**160 What does SOAP encodingStyle attribute defines?**

SOAP encodingStyle defines the serialization rules used in a SOAP message. This attribute may appear on any element, and is scoped to that element's contents and all child elements not themselves containing such an attribute. There is no default encoding defined for a SOAP message.

An example: SOAP-ENV:encodingStyle="http://www.w3.org/2001/12/soap-encoding"

**161 What are 2 styles web service’s endpoint by using JAX-WS?**

* RPC (remote procedure call style web service in JAX-WS;
* document style web service in JAX-WS.

**162 What is encoding rules for header entries?**

* a header entry is identified by its fully qualified element name, which consists of the namespace URI and the local name. All immediate child elements of the SOAP Header element must be namespace-qualified.
* the SOAP encodingStyle attribute may be used to indicate the encoding style used for the header entries.
* the SOAP mustUnderstand attribute and SOAP actor attribute may be used to indicate how to process the entry and by whom.

**163 What is the wsimport tool?**

The wsimport tool is used to parse an existing Web Services Description Language (WSDL file and generate required files (JAX-WS portable artifacts for web service client to access the published web services: https://docs.oracle.com/javase/6/docs/technotes/tools/share/wsimport.html

**164 What is the wsgen tool?**

The wsgen tool is used to parse an existing web service implementation class and generates required files (JAX-WS portable artifacts for web service deployment: http://docs.oracle.com/javase/6/docs/technotes/tools/share/wsgen.html

**165 What is the difference between SOAP and other remote access techniques?**

* SOAP is simple to use and it is non - symmetrical unlike DCOM or CORBA is highly popular and usually have complexity in them.
* SOAP provides greater platform independent with the language independence unlike DCOM or CORBA doesn't provide any of these.
* SOAP uses HTTP as its transport protocol and the data are being saved in XML format that can be ready by human, whereas DCOM or CORBA have their own binary formats that are used to transport the data in complicated manner.
* SOAP identify the object other than URL endpoint. SOAP objects are stateless and it is hard to maintain that. Whereas, it is not hard to maintain in case of other remote access techniques.

**166 What is a resource in a REST?**

A resource is a unique URL with representation of an object which we can get contents via GET and modify via PUT, POST, DELETE.

**167 What are HTTP methods supported by REST?**

• GET;

• POST;

• PUT;

• DELETE;

• OPTIONS;

• HEAD.

**168 Whether can use GET request instead of POST to create a resource?**

It is not possibly, because GET can’t change a resource.

**169 What is the difference between PUT and POST?**

Need to use PUT when can update a resource completely through a specific resource. For example, if know that an article resides at http://javahungry.blogspot.com/article/123, can PUT a new resource representation of this article through a PUT on this URL. If do not know the actual resource location for instance, when add a new article, can use POST.

PUT is idempotent, while POST is not. It means if use PUT an object twice, it has no effect.

**170 What is WADL?**

WADL (Web Application Description Language is a XML description of a deployed RESTful web application.

**171 What are frameworks available to implement REST web services?**

Jersey, Restlet, EasyRest, etc.

**172 What is the Restlet framework?**

Restlet is a lightweight, comprehensive, open source RESTful web API framework for the Java platform.

It has advantages such as

* websocket and server-sent events support;
* HTTP/2 support;
* transparent HTTP PATCH support;
* client cache service;
* fluent APIs.

http://restlet.com/

**173 What is the Jersey framework?**

Jersey is open source framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339 Reference Implementation. It has

advantages such as

* contains support for Web Application Description Language (WADL);
* contains Jersey Test Framework which lets run and test Jersey REST services inside JUnit;
* supports for the REST MVC pattern, which would allow to return a View from Jersey services rather than just data.

https://jersey.java.net/

**174 What is the RESTeasy framework?**

RESTeasy is a JBoss project, which implements of the JAX-RS specification. It has benefits such as

* fully certified JAX-RS implementation; supports HTTP 1.1 caching semantics including cache revalidation;
* JAXB marshalling into XML, JSON, Jackson, Fastinfoset, and Atom as well as wrappers for maps, arrays, lists, and sets of JAXB Objects;
* OAuth2 and Distributed SSO with JBoss AS7;
* rich set of providers for: XML, JSON, YAML, Fastinfoset, Multipart, XOP, Atom, etc.

http://resteasy.jboss.org/

**175 What is the difference between AJAX and REST?**

* in Ajax, the request are sent to the server by using XMLHttpRequest objects; REST have a URL structure and a request/response pattern the revolve around the use of resources;
* Ajax eliminates the interaction between the customer and server asynchronously; REST requires the interaction between the customer and server;
* Ajax is a set of technology; REST is a type of software architecture and a method for users to request data or information from servers.

**176 What tool are required to test REST services?**

Firefox “poster” plugin for RESTFUL services. https://addons.mozilla.org/en-us/firefox/addon/poster/

**177 What does a @Path annotation do?**

@Path annotation binds URI pattern to a Java method.

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.core.Response;

@Path("/persons")

public class PersonRestService {

@GET

public Response getPerson( {

return Response.status(200).entity("getPerson is called").build();

}

@GET

@Path("/vip")

public Response getPersonVIP( {

return Response.status(200).entity("getPersonVIP is called").build();

}

}

On calling URI: “/persons” result: getPerson is called

On calling URI: “/persons/vip” result: getPersonVIP is called

**178 What does a @PathParam do?**

@PathParam annotation injects the value of URI parameter that defined in @Path expression.

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.PathParam;

import javax.ws.rs.core.Response;

@Path("/persons")

public class PersonRestService {

@GET

@Path("{id}")

public Response getPersonById(@PathParam("id" String id {

return Response.status(200).entity("getPersonById is called, id : " + id).build();

}

}

On calling URI: “/persons/1” result: getPersonById is called, id : 1

**179 What does a @QueryParam do?**

@QueryParam annotation injects URI query parameter into Java method.

import java.util.List;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.QueryParam;

import javax.ws.rs.core.Response;

@Path("/persons")

public class PersonService {

@GET

@Path("/query")

public Response getPersons(

@QueryParam("from" int from,

@QueryParam("to" int to,

@QueryParam("orderBy" List&lt;String&gt; orderBy {

return Response

.status(200)

.entity("getPersons is called, from : " + from + ", to : " + to

+ ", orderBy" + orderBy.toString()).build();

}

}

On calling URI: “/persons/query?from=10&to=20&orderBy=age&orderBy=name” result: getPersons is called, from : 10, to : 20, orderBy[age, name]

**180 What does a @MatrixParam do?**

@MatrixParam are a set of “name=value” in URI path.

import javax.ws.rs.GET;

import javax.ws.rs.MatrixParam;

import javax.ws.rs.Path;

import javax.ws.rs.PathParam;

import javax.ws.rs.core.Response;

@Path("/books")

public class BookService {

@GET

@Path("{year}")

public Response getBooks(@PathParam("year" String year,

@MatrixParam("author" String author,

@MatrixParam("country" String country {

return Response

.status(200)

.entity("getBooks is called, year : " + year

+ ", author : " + author + ", country : " + country)

.build();

}

}

On calling URI: “/books/2015” result: getBooks is called, year : 2015, author : null, country : null

On calling URI: “/books/2015;author= doyle;country=scotland” result: getBooks is called, year : 2015,

author : doyle, country : scotland

**181 What does a @FormParam do?**

@FormParam bind HTML form parameters value to a Java method.

import javax.ws.rs.FormParam;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.core.Response;

@Path("/persons")

public class PersonService {

@POST

@Path("/add")

public Response addPerson(

@FormParam("name" String name,

@FormParam("age" int age {

return Response.status(200)

.entity("addPerson is called, name : " + name + ", age : " + age)

.build();

}

}

HTML form:

<html>

<body>

<form action="/persons/add" method="post">

<p>

Name : <input type="text" name="name" />

</p>

<p>

Age : <input type="text" name="age" />

</p>

<input type="submit" value="Add Person" />

</form>

</body>

</html>

**182 How to get HTTP request header in JAX-RS (2 ways)?**

• inject directly with @HeaderParam;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.HeaderParam;

import javax.ws.rs.core.Response;

@Path("/persons")

public class PersonService {

@GET

@Path("/get")

public Response getPerson(

@HeaderParam("person-agent" String personAgent {

return Response.status(200)

.entity("getPerson is called, personAgent : " + personAgent)

.build();

}

}

On calling URI: “/persons/get” result: getPerson is called, personAgent : Mozilla/5.0 (Windows NT 6.1; rv:5.0 Gecko/20100101 Firefox/5.0

• pragmatically via @Context.

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.core.Context;

import javax.ws.rs.core.HttpHeaders;

import javax.ws.rs.core.Response;

@Path("/persons")

public class PersonService {

@GET

@Path("/get")

public Response getPerson(@Context HttpHeaders headers {

String personAgent = headers.getRequestHeader("person-agent").get(0);

return Response.status(200)

.entity("getPerson is called, personAgent : " + personAgent)

.build();

}

}

On calling URI: “/persons/get” result: getPerson is called, personAgent : Mozilla/5.0 (Windows NT 6.1; rv:5.0 Gecko/20100101 Firefox/5.0

**183 How to download file in JAX-RS?**

• put @Produces(“?” on service method, with a Response return type. Instead “?” write a type text/plain, image/png, etc.

• set “Content-Disposition” in Response header to tell browser pop up a download box for user to download.

import java.io.File;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.Response;

import javax.ws.rs.core.Response.ResponseBuilder;

@Path("/image")

public class ImageService {

private static final String FILE\_PATH = "c:\\my.png";

@GET

@Path("/get")

@Produces("image/png")

public Response getFile( {

File file = new File(FILE\_PATH);

ResponseBuilder response = Response.ok((Object file);

response.header("Content-Disposition",

"attachment; filename=image\_from\_server.png");

return response.build();

}

}