**Advanced Java**

**Q.1: What is web-application?**

Web-application is a software which runs on remote server and can be accessed by a user via browser.

**Q.2: What is difference between web application and websites?**

Usually, websites contain static content whereas web applications are interactive with end user.

Websites usually contains simple functions whereas functions of web-applications are quite complex.

**Q.3: What do you mean by web server?**

A web server is a program which processes the network request of the user and serves them with the files and contents which creates web pages. And this exchange is takes place via HTTP protocol.

**Q. 4: How web server works?**

Whenever any browser send request to server. The request is sent in the form of small small packets considering HTTP protocol. These packets are then gathered and reassembled by the server. Then server unwraps these packets and read what is requested and according sends the response in the form of what is understood by the browser.

**Q.5: What is client-server architecture?**

Client: Client is a media which send request to the server and receives request from server. Our browser will act as a client.

Server: Server is a media which receives the request from the client. Reads the request and accordingly sends the dynamic response to client.

Protocol: The main job of the protocol in client server architecture is to carry the request data from client to server and response from server to client in predefined format.

**Q. 6: What is HTTP?**

HTTP stands for Hyper Text Transfer Protocol which is used to transfer the data over the web.

HTTP is a stateless protocol means that every request is treated as a new request on server.

HTTP unable (not able to manage) clients previous request data.

**Q. 7: What is difference between JSE and J2EE?**

Java SE stands for Java standard edition which is used to develop desktop applications whereas J2EE stands for Java Enterprise Application which is used to develop Enterprise applications. Enterprise application is a large software system platform design to operate in a corporate environment.

**Q. 8: What is J2EE?**

J2EE stands for Java Enterprise Edition. It consists of specifications only. Specifications required for primary services of any Enterprise Application. Such as servlet API, JSP API, Security, Connection pooling, Java Persistent API, webservices support etc. And implementation of these specifications is left to the vendors. Vendor such as Apache provided Tomcat web server and Tomee application server.

**Q. 9: What is the difference between web server and application server?**

Web server is comparatively small program which provide only primary services implementations whereas application server provides all services implementation and application server itself contains web server.

**Q. 10: What is web container?**

We can say it is a server side JVM resides within a web server. It provides a run-time environment for dynamic web components (Servlet & JSP)

* It can create the Http request and Http response objects.
* It controls and manages the lifecycle of Dynamic web components
* It provides ready made support for services like security, connection pooling etc.
* It can handle the concurrent request from multiple clients
* And can manage session tracking.

**Q. 11: What is servlet?**

Servlet is a java class, which is used to add dynamic nature to the web application. And its life cycle is managed by the web container.

**Q. 12: What is life cycle of Servlet?**

Whenever any request comes for a particular servlet. The web container first locate the servlet, load then instantiate it with default constructor and then lifecycle methods like init(), service() and destroy() gets called.

Init() method gets called after instantiation of the servlet. Service() method actually contains business logic and it gets called for every request. Destroy() method gets called redeployment of the application or when the container gets closed.

**Note:** @WebServlet(value=”/validate”)

**Q. 13: What is session?**

Session is a conversional state between the client and server.

**Q. 14: What is session tracking and why do we need session tracking?**

Session is a conversional state between the client and server. Since HTTP protocol and web servers are stateless it means every request coming from a client is a new request. But in some applications, we should know who the client is to process the request accordingly. EX. In shopping application we need client information to charge the amount from the correct client during checkout.

So, session tracking is needed to

* identify the client among multiple clients
* And to remember the conversational state of the client.

**Q. 15: How will you do session tracking in java?**

1. Plain cookie

2. Httpsession API

3. URL Rewriting

**Q. 16: What is Cookie?**

Cookie is a small amount of text data created by the server. Server stores the client specific information in this cookie and then send these cookies in the response header and these cookies are then downloaded by browser and will send it in every request. In this case the cookie management is done by the developer.

**Q. 17: What is Httpsession API?**

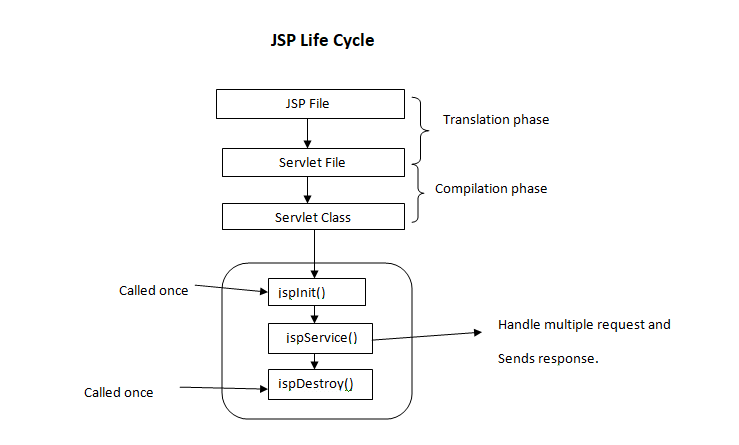
In this technique entire state of the client is not saved on the client side instead saved on the server-side data structure in Http session object. But the key to this Http session object is STILL send to the client in the form of cookie. In this case the cookie management is done by the web container.

In case of URL Rewriting additional parameter is appended with URL.

**Q. 18: What is JSP?**

JSP stands for Java Server Pages. It its server-side technology and used to create web application with dynamic web content. It is advanced version of servlet technology where JSP is first converted into servlet by JSP container before processing the clients request.

**JSP Life Cycle?**



**Q. 19: What is Maven?**

Maven is a Project Management Tool or we can say it’s a build tool based on pom.xml. So, when we create a project, we need lots of dependencies, so instead of we downloading the dependencies maven will download the dependencies for us based on pom.xml file. It downloads the dependencies from maven repository.

**Q. 20: What are scripting elements?**

Scripting elements are JSP tags that let you insert the java code into JSP pages,

**Q. 21: What is JDBC?**

JDBC stands for Java Database Connectivity. Using JDBC driver we can connect our java application to database and then execute the query with database. JDBC Drivers are the implementations of the java supplies specifications provided by the 3rd party vendors in the form of JAR files.

**NOTE:** JAR (Java archival format: compressed bundle of packaged Java classes)

* Java supplies specifications: java.sql.Connection --i/f sun supplied(Java SE)
* Imple class provided by MySQL -- com.mysql.cj.jdbc.ConnectionImpl.class
* Imple class provided by Oracle -- oracle.jdbc.OracleConnection

**Q. 22: What is Hibernate?**

Java Persistence API i.e., JPA is a specification of java. It is used to persist (Store) data between java object and relational database. JPA acts as a bridge between the object oriented world and relational database world. However, JPA are only specifications, it doesn’t perform any operation by itself. It requires an implemetation. So, Hibernate is one such ORM (Object to Relational Mapping) tool or framework which implements the JPA specifications for data persistence. As a ORM tool it helps us to simplify data creation, data manupulation and data access.

**NOTE:** The **javax.persistence** package contains the JPA classes and interfaces.

**Q. 23: Do you know any other ORM tools available?**

Yes, apart from Hibernate, TopLink and iBatis also implements JPA specifications for data persistence.

**Q. 24: Why did you choose hibernate for your project?**

- Hibernate is open source and lightweight framework.

- Hibernate uses chache internally, hence it is fast.

- we can write database independent queries

- Fetching data from multiple tables is easy in hibernate

- Hibernate supports Query cache and provide statistics about query and database status

*hibernate.generate\_statistics* to *true* in properties.xml

**Q. 26: What are the advantages of using ORM over JDBC?**

* Application development is fast.
* Management of transaction is easy.
* Generates key automatically.
* Details of SQL queries are hidden.

**Q.25: What are the core interfaces of hibernate?**

1. Configuration
2. Session Factory
3. Session
4. Query
5. Transaction

**Q. 26: What is Configuration file?**

Configuration is a class given by hibernate people to load the hibernate XML configuration file.

Configuration cfg = new Configuration ();

**Q. 27: What is session?**

Session represents a wrapper around a pooled out JDBC connection. It provides methods to store, update, delete or fetch data from the database such as persist(), update(), delete(), load(), get() etc.

**Q. 28: What is connection Pooling?**

A JDBC connection pool is **a group of reusable connections for a particular database**. Because creating each new physical connection is time consuming, the server maintains a pool of available connections to increase performance. When an application requests a connection, it obtains one from the pool

**Q. 29: What is Session Factory?**

We use Session Factory object to create session object. It is Singleton in nature that means we get single object per DB per application.

**Q. 30: What is Configuration?**

Configuration object is used to create the SessionFactory object.

**NOTE**: The org.hibernate.Transaction interface provides methods for transaction management.

**Q. 31: What is Hibernate caching?**

Caching is a facility provided by ORM frameworks which help users to get fast running web application, while help framework itself to reduce number of queries made to database in a single transaction.

Hibernate achieves this by implementing first level cache.

First level cache in hibernate is enabled by default and it is asscociated with session object.

|  |  |
| --- | --- |
| Annotations | Use |
| @Entity | Used for declaring any POJO class as an Entity for database |
| @Table | Used to change table details   * name: to override the table name |
| @ID | Used for declaring primary key inside our POJO class |
| @GeneratedValue  @GeneratedValue(strategy = GenerationType.***IDENTITY***) | Used to generate the values automaticaly, we don’t need to set values manually. |
| @Column | It is used to specify column mapping. In case we don’t need to specify name of table as POJO we can change it using name attribute.   * name: we can change name of the entity for database * length: the size of the column mostly used in string * unique: to add unique constraint to colum * nullable: to mark the column values can be NULL or NOT |
| @Transient | To tell hibernate not to add this particular collumn |
| @Temporal | Used to format the date for storing to database |

**Q. 32: What is JPQL?**

It is Java Persistence Query Language

**Q. 33: POJO/Entity life cyle?**

1. Transient

An object is said to be in transient state if it is not associated with the session,and has no matching record in the database table.

2. Persistent

An object is said to be in persistent state if it is associated with session object (L1 cache) and will result into a matching record in the databse table.

Ex. After Save() method it is in L1 cache and after commit its record gets created or updated in Database. However during get() method call we will get already pesistent object in L1 cache.

3. Detached

Object is not associated with session but has matching record in the database table. If we make any changes to the state of detached object it will NOT be reflected in the database.

4. Removed

Object was earlier associated with session and also had a record in database table. However after removed state it is neigther assoociated with session nor it has record in database table.

**Q. 34: What is eager and Lazy loading? What is difference between them?**

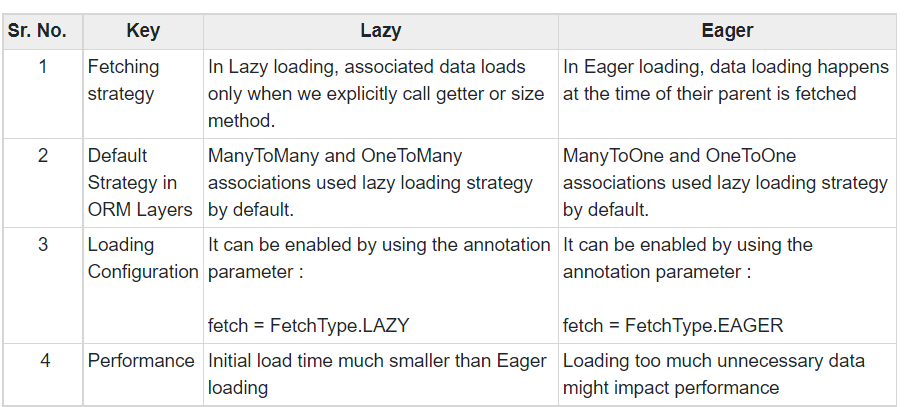
Eager and Lazy are data loading startegies in Hibernate ORM. These strategies are used when one entity class contains reference of other entity class. Means whenever there relation exists between entities we use these data loading strategies.

**Lazy Loading** − Associated data loads only when we explicitly call getter or size method.

* Use Lazy Loading when you are using one-to-many collections.
* Use Lazy Loading when you are sure that you are not using related entities.

**Eager Loading** − Data loading happens at the time of their parent is fetched.

* Use Eager Loading when the relations are not too much. Thus, Eager Loading is a good practice to reduce further queries on the Server.
* Use Eager Loading when you are sure that you will be using related entities with the main entity everywhere.



**Q. 34: What is difference between session.save() and session.persist()?**

|  |  |
| --- | --- |
| **save** | **persist** |
| Return serializable ID of the instance | Return nothing since return type is void |
| public Serializable save(Object o) | public void persist(Object o) |
| If you give not NULL Id no matter existing or non-existing, it doesn’t throw an exception. It simply ignores the passed Id and create a new record. | If we pass not NULL Id it will throw PersistentObjectException i.e., detached entity passed to persist. |

**Q. 35: What is Spring?**

Spring is both container and framework.

Container is because it manages life cycle of the spring beans. Spring beans are the java objects (e.g., controller, service, Dao) whose life cycle is completely managed by the Spring container. And the framework is because it provides readymade implementations of standard patterns like (MVC, proxy, singleton, factory, ORM). Spring does not implement any of the J2EE specifications however it is created to make developing complex J2EE applications easier.

**Q. 36: Why Spring?**

1. Spring framework provides us predefined templates, templates for JDBC, Hibernate, JPA etc. technologies. Thus, we don’t have to write the too much code again and again.
2. Spring applications are loosely coupled because of dependency injection.
3. And because of the dependency injection it makes easier to test the application
4. Spring framework doesn’t force programmer to inherit any class or to implement any interface. That is why it is said non-invasive.
5. Dependency injection feature of spring framework and its support to many frameworks makes the development of Java EE application fast.

**Q. 37: What is dependency injection? And how do you achieve it?**

Dependency injection is a process through which Spring container injects objects into other objects. The object to be injected is called dependency of the dependent object. It makes our application loosely coupled, so when anytime the nature of dependency changes it won’t affect the dependent object.

There are two ways:

1. Constructor based DI
2. Setter based DI

**Q. 38: What is Spring IOC?**

IOC means inversion of control is a design pattern to provide loose coupling. Spring IOC is a container in a core spring framework. It creates the object, configure and assemble their dependencies and manages their life cycle. The container uses Dependency Injection to manage the component.

**Q. 39: What is Spring boot?**

Spring boot is a framework which is implemented on top of the existing spring framework to ease bootstrapping (self-starting process) and development of new Spring application.

1. So, it is very easy to develop spring-based application with java
2. It reduces lots of development time by providing auto configuration and hence increases productivity
3. It avoids writing lots of boilerplate code and annotations.
4. It follows "Opinionated Defaults Configuration" Approach to reduce Developer effort
5. It provides embedded HTTP servers like Tomcat. To develop and test our application very easily.

**Q. 40: Why did you choose Spring boot for you project as a backend technology?**

1. The very first and super cool feature of the spring boot is auto configuration. Spring boot automatically configure everything that is required for our application.
2. Another amazing feature of spring boot is the spring boot starter. These are the dependency descriptors added under the **<dependencies>** section in pom.xml. So, before spring boot was introduced developers used to include all those dependencies. But now spring boot starter provides all those with just a single dependency.
3. Third feature is that we don’t have to worry about deploying our application to external container. By default, Spring boot contains embedded Tomcat server. So, we can simply right click and run our project on embedded tomcat server.

Using all these features, application development becomes faster.

**Q. 41: Explain life cycle of the spring java beans?**

Spring beans are the java objects whose life cycle is managed by the spring container.

Like every living being has a life cycle means when and how it is born, how it behaves throughout his life, and when & how it dies. Similarly, the bean life cycle refers to when & how the bean is instantiated, what actions it performs until it lives, and when & how it is destroyed.

//So, when the spring container gets started. After that it locate, load and create instance of //the bean as per request.

So, if we want to execute some piece of code on bean instantiation and just before the container destroyed bean, we can use custom init() and destroy() methods.

To provide these custom methods using programmatic approach, we need to implement our bean with two interfaces.

1. InitializingBean

2. DisposableBean interfaces. And then need to override afterPropertiesSet() and destroy() methods.

And using annotation approach, we can write @PostContructor and @Prdestroy annotation on custom methods.

**Q. 42: What is Autowiring?**

Autowiring is a feature of Spring Framework, using which it automatically selects the bean object to be injected.

**Q. How FE different from BE?**

**Q. What is difference between query parameter & Path variable?**

**Q. What is CORS?**

CORS stands for Cross Origin Resource Sharing.

Means if two servers are running on different ports they may be on same m/c or on different m/c’s. EX. We have FE React server running on port 3000 and BE Spring boot application running on port no. 8080. So, whenever FE sends request to BE via Browser, Browser first send a preflight request to server to tell that the request is coming from different application having different origin i.e., running on different server would you like to permit it or not.

And in case of CORS failure, error will be displayed.

Q. How did you used CORS in your project to deal with CORS Issue?

Spring provides annotation support for CORS. We have used **@CrossOrigin** annotation on controller

**NOTE:** we can use this annotation at method level as well

**Q. What is marshalling and unmarshalling?**

Marshalling also known as serialization is a process of converting state of the object into stream of different forms. Ex. Converting a state of java object into JSON format is called marshalling.

Whereas regaining state of the object is called Unmarshalling. Ex. Converting the JSON object into Java object.

**Q. What do you mean by REST API?**

REST stands for Representational State Transfer. It is an architectural style and uses HTTP protocol for data communication.