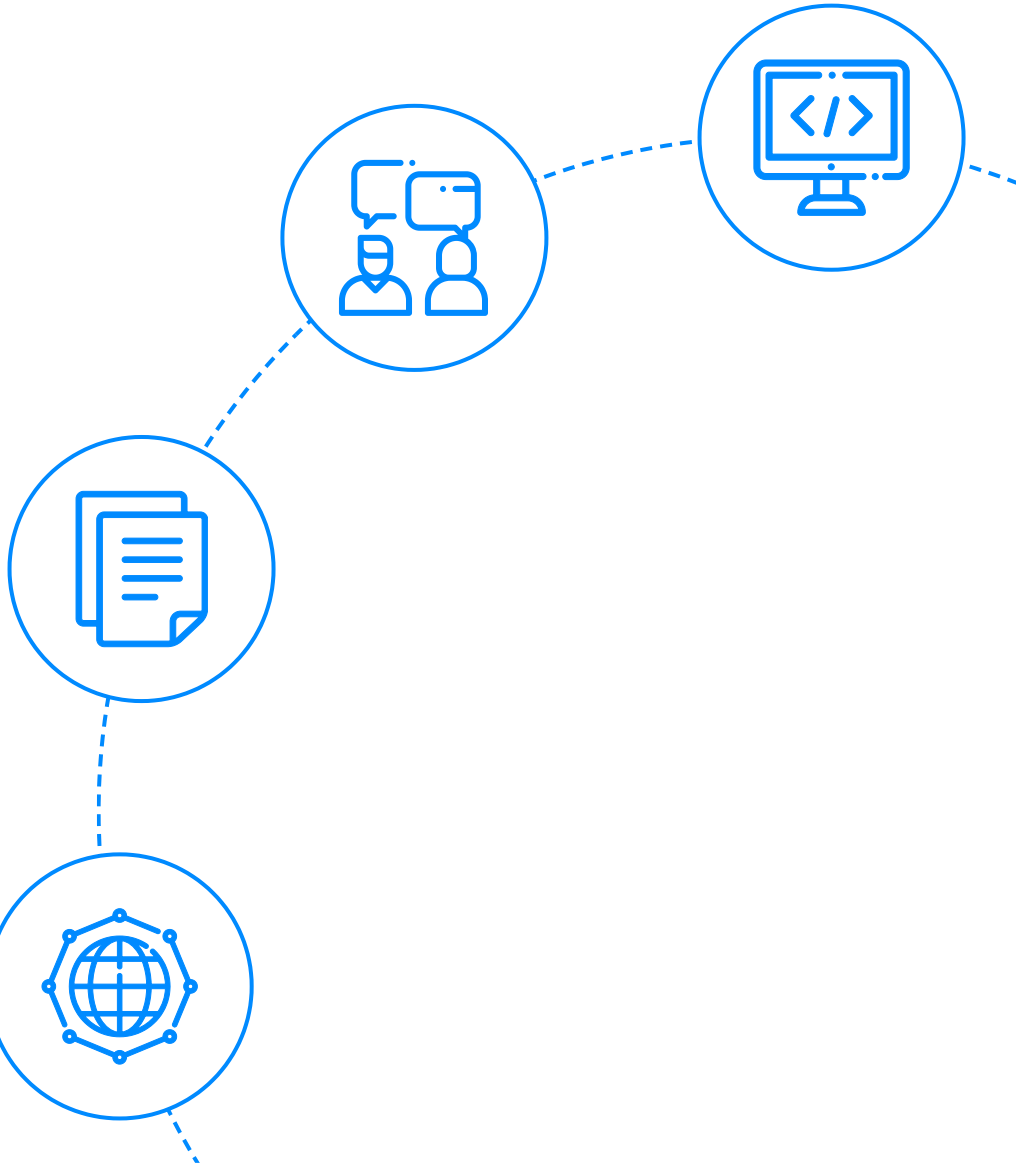
InterviewBit

J2EE Interview Questions



To view the live version of the page, click here.

Contents

J2EE Interview Questions for Freshers

J2EE Interview Questions for Experienced

J2EE Interview Questions for Experienced(.....Continued)

19. What do you understand by Connectors? Can you describe the Connector Architecture?
20. What do you understand by JRMP?
21. What happens if the database connected to the Java application via connection pool suddenly goes down?
22. How is 32-bit JVM different from 64-bit JVM?
23. How is a webserver different from an application server?
24. What is the purpose of heap dumps and how do you analyze a heap dump?
25. How can we take a heap dump of a Java process?
26. How is J2EE different from Spring?
27. What are EAR, WAR, and JAR?
28. What do you know about Hibernate?
29. What are deployment descriptors used for?
30. Can you describe the phases of the servlet lifecycle?
31. How does a servlet application work?
32. What do you understand by Java Message Service (JMS)?

Let's get Started

Introduction to J2EE

J2EE (Java Enterprise Edition) standards were first proposed by **Oracle** (Sun Microsystems) to help developers develop, build and deploy reusable, distributed, reliable, scalable, portable and secure enterprise-level business applications. In simple terms, J2EE constitutes a set of frameworks, a collection of APIs and various J2EE technologies like **JSP**, **Servlets** etc that are used as standards for simplifying the development and building of large scale applications.

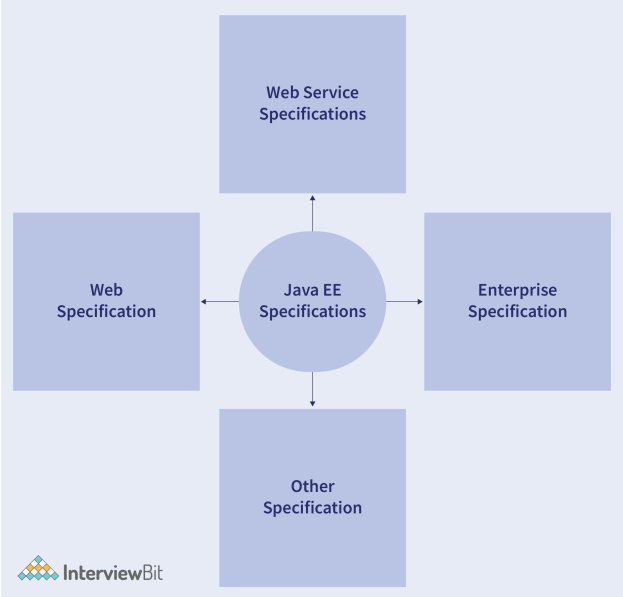
It is aimed at easing the development, build and deployment process of enterprise-level applications that can be run on different platforms which supports Java. J2EE remains the most popular standard followed by the Java developers community which is why it is important for developers to know about J2EE concepts and have hands-on experience in them.

In this article, we will see the most commonly asked interview questions on J2EE for both freshers and experienced professionals.

J2EE Interview Questions for Freshers

1. What is J2EE?

J2EE or Java Enterprise Edition is a **Java**-based platform that is a combination of services protocols and APIs (Application Programming Interfaces) that provides capabilities to develop multi-tier, secure, stable and fast enterprise-level applications. J2EE provides web, enterprise, web service and various other specifications for developing enterprise-level web applications.



2. What are the main advantages of J2EE?

Following are the advantages of the J2EE platform:

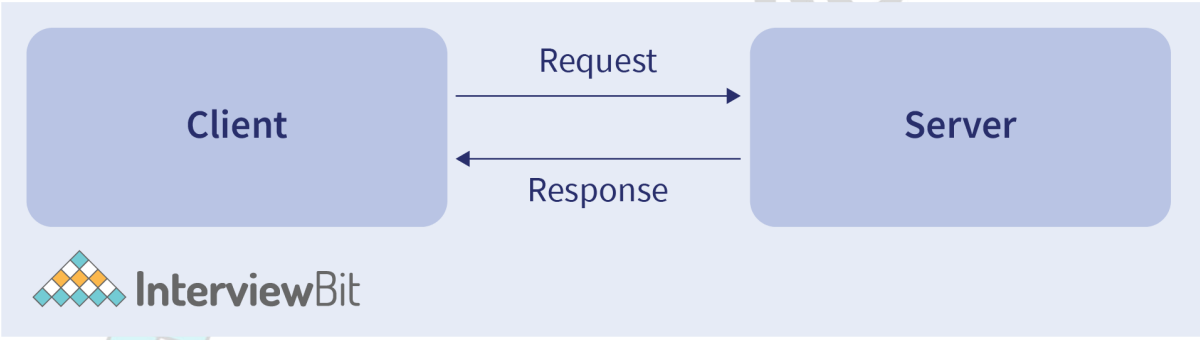
- **Support for Web Services:** J2EE provides a platform to develop and deploy web services. The JAX-RPC (Java API for XML based Remote Procedure Call) helps developers develop SOAP-based portable and interoperable web services, clients and endpoints.
- **Faster Time to Market:** J2EE uses the concept of containers for simplifying the development. This helps in business logic separation from lifecycle management and resources which aids developers to focus on business logic than on the infrastructure. For instance, the EJB (Enterprise JavaBeans) container takes care of threading, distributed communication, transaction management, scaling etc and provides a necessary abstraction to the developers.
- **Compatibility:** J2EE platform follows the principle of "Write Once, Run Anywhere". It provides comprehensive standards and APIs that ensures compatibility among different application vendors resulting in the portability of applications.
- **Simplified connectivity:** J2EE helps in easier applications connectivity which allows utilizing the capabilities of different devices. It also provides JMS (Java Message Service) to integrate diverse applications in asynchronous and loosely coupled ways. It also provides CORBA (Common Object Request Broker Architecture) support for linking systems tightly via remote calls.

Due to all the above benefits packed in one technology, it helps the developers to reduce the TCO (Total Cost of Ownership) and also focus more on actual business logic implementation.

3. What are some of the technologies provided by the J2EE platform?

Some of the important technologies provided by J2EE are:

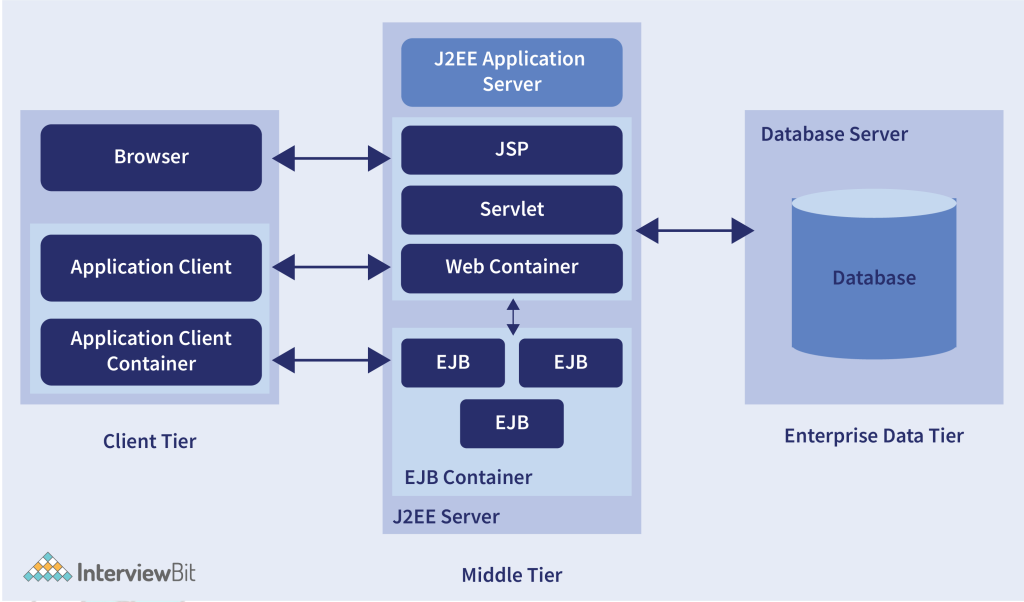
- **Java API for XML-Based RPC (JAX-RPC):** This is used to build web services and clients that make use of XML and Remote Procedure Calls.
- **Java Server Pages (JSP):** This is used for delivering XML and HTML documents. Apart from these, we can make use of OutputStream for delivering other data types as well.
- **Java Servlets:** Servlets are classes used for extending the server capabilities which hosts applications and can be accessed using the request-response model.



- **Enterprise Java Beans (EJB):** This is a server-side component that is used for encapsulating the application's business logic by providing runtime environment, security, servlet lifecycle management, transaction management and other services.
- **J2EE Connector Architecture:** This defines standard architecture to connect J2EE platforms to different EIS (Enterprise Information Systems) such as mainframe processes, database systems and different legacy applications coded in another language.
- **J2EE Deployment API:** Provides specifications for web services deployment
- **Java Management Extensions (JMX):** They are used for supplying tools for monitoring and managing applications, objects, devices and networks.
- **J2EE Authorization Contract for Containers (JACC):** This is used to define security contracts between authorization policy modules and application servers.
- **Java API for XML Registries (JAXR):** This provides standard API to access different XML Registries to enable infrastructure for the building and deployment of web services.
- **Java Message Service (JMS):** This is a messaging standard for allowing different JEE components for creating, sending, receiving and reading messages by enabling communication in a distributed, loosely coupled, asynchronous and reliable manner.
- **Java Naming and Directory Interface (JNDI):** This is an API that provides naming and directory functionality for Java-based applications.
- **Java Transaction API (JTA):** This is used for specifying Java standard interfaces between transaction systems and managers.
- **Common Object Request Broker Architecture (CORBA):** This provides a standard for defining Object Management Group designed for facilitating system communication deployed on diverse platforms.
- **JDBC data access API:** This provides API for getting data from any data sources like flat files, spreadsheets, relational databases etc.

4. What are the various components of J2EE application architecture?

J2EE is made up of 3 main components (tiers) - Client tier, Middle tier, Enterprise data tier as shown in the below image:



- **Client Tier:** This tier has programs and applications which interact with the user and they are generally located in different machines from the server. Here, different inputs are taken from the user and these requests are forwarded to the server for processing and the response will be sent back to the client.
- **Middle Tier:** This tier comprises of Web components and EJB containers. The web components are either servlet or JSP pages that process the request and generate the response. At the time of the application's assembly, the client's static HTML codes, programs and applets along with the server's components are bundled within the web components. The EJB components are present for processing inputs from the user which are sent to the Enterprise Bean that is running in the business tier.
- **Enterprise Data Tier:** This tier includes database servers, resource planning systems and various other data sources that are located on a separate machine which are accessed by different components by the business tier. Technologies like JPA, JDBC, Java transaction API, Java Connector Architecture etc are used in this tier.

5. How is JDK different from JIT?

JDK (Java Development Kit) is a cross-platformed software development environment offering various collections of libraries and tools required for developing Java applications and applets. It also consists of JRE that provides tools and libraries which aids in byte code execution. JDK is needed for writing and running programs in Java. Whereas JIT stands for Just In Time Compiler which is a module inside JVM (which is inside JRE). JIT compiler is used for compiling some parts of byte code having similar functionality at the same time to machine code for optimising the compilation time and performance.

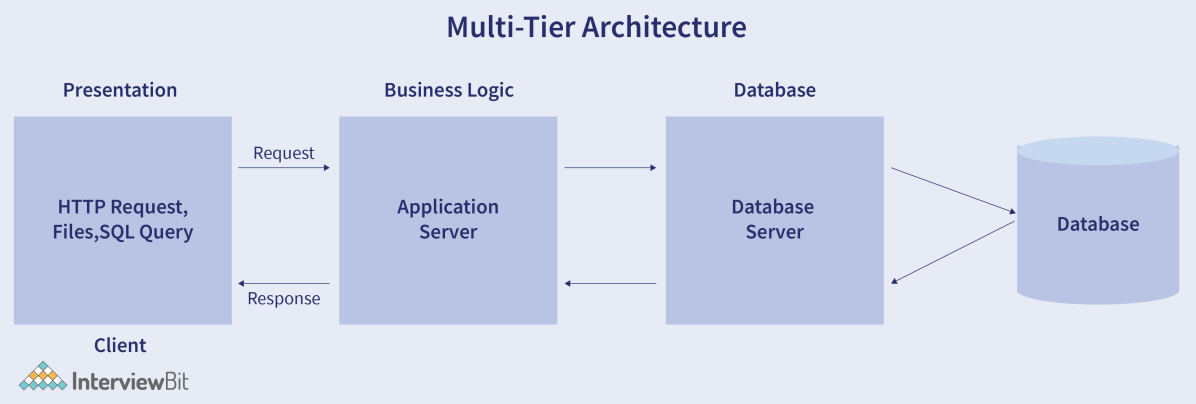
6. How are PATH and CLASSPATH different from each other in terms of J2EE?

PATH and CLASSPATH are key environmental variables used by Java platforms.

- The PATH variable points to JDK binaries or native libraries like java.exe. Whereas The CLASSPATH variable points to the binaries of Java such as JAR files that consist of bytecode.
- PATH is a system-level variable independent of Java being present in the system or not. Whereas CLASSPATH is purely Java-specific which is used by JVM for loading classes required by Java applications while running.

7. How is multi-tier client-server architectural model advantageous?

Multi-tier client-server architectural model consists of various components known as tiers that interact with each other. The below image represents the three-tier application model which has client/presentation tier, business logic tier and the database tier which interact with each other to process a request and send a response:



In the multi-tier system, we have the following advantages:

- Any changes to the user interface or business logic can be done independently.
- It introduces abstraction between the components. For example, the client tier can access data without knowing from where and how the response comes from, what is the server infrastructure available in the backend etc.
- Each tier can be coded or developed independently. For example, the middle tier can be coded in Java or python, the client tier can be coded in Angular or React etc.
- The database can have pooled connections for sharing data among multiple users without the need for creating a new connection for every user.

8. What do you understand by build file?

A build file is used for automating various steps involved in software development. Along with this, the build file also specifies libraries and their versions that need to be included. It also includes the type of optimizations required for the project. Whenever the project size increases, build provides a standard way to build the project.

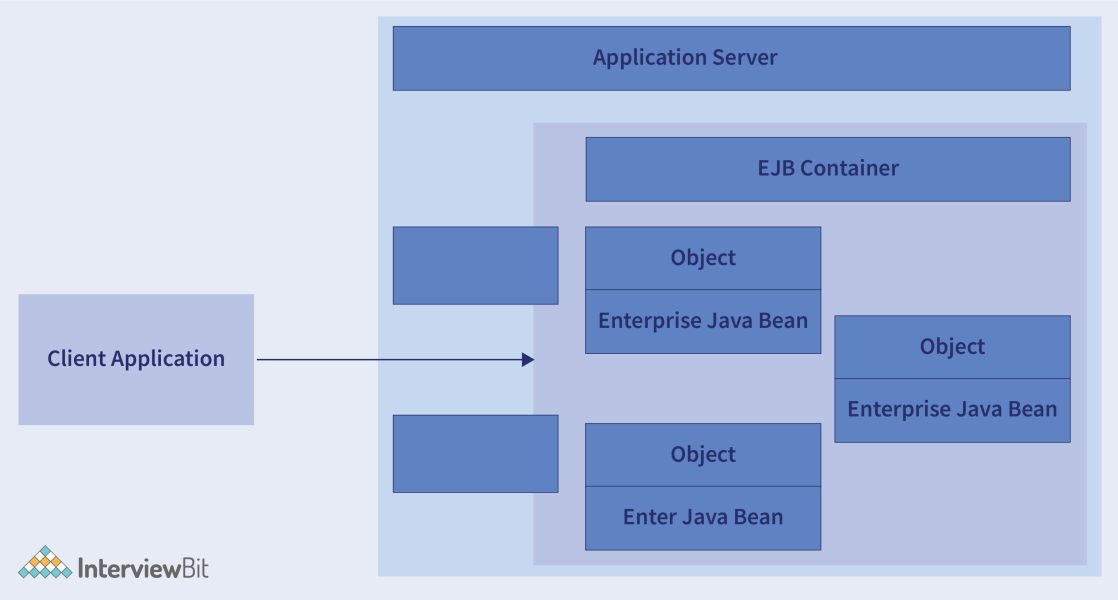
9. Why do we have JDBC and JNDI in J2EE? How are they different from each other?

JDBC or Java Database Connectivity provides guidelines and APIs for connecting databases from different vendors like MySQL, Oracle, PostgreSQL etc for getting data. JNDI (Java Naming and Directory Interface) helps in providing logical structure to retrieve a resource from the database, EJB beans, messaging queues etc without knowing the actual host address or port. A resource can be registered with JNDI and then those registered application components can be accessed using the JNDI name.

10. What is an EJB? How can you use it in J2EE?

EJB or Enterprise Java Beans is one of the most important parts of the J2EE platform that helps to develop enterprise-level multi-tiered applications and deploy them by keeping in mind performance, scalability and robustness. EJBs can be used when we want to achieve the following:

- **Clustering**: For deploying the application in a cluster environment, EJBs can be used to achieve high availability and fault tolerance.
- **Concurrency without using Threads**: EJBs can be used to achieve concurrency without using actual threads since they are instantiated using the object pool and are available in the EJB container. This helps in achieving performance without involving complexities around Threads.



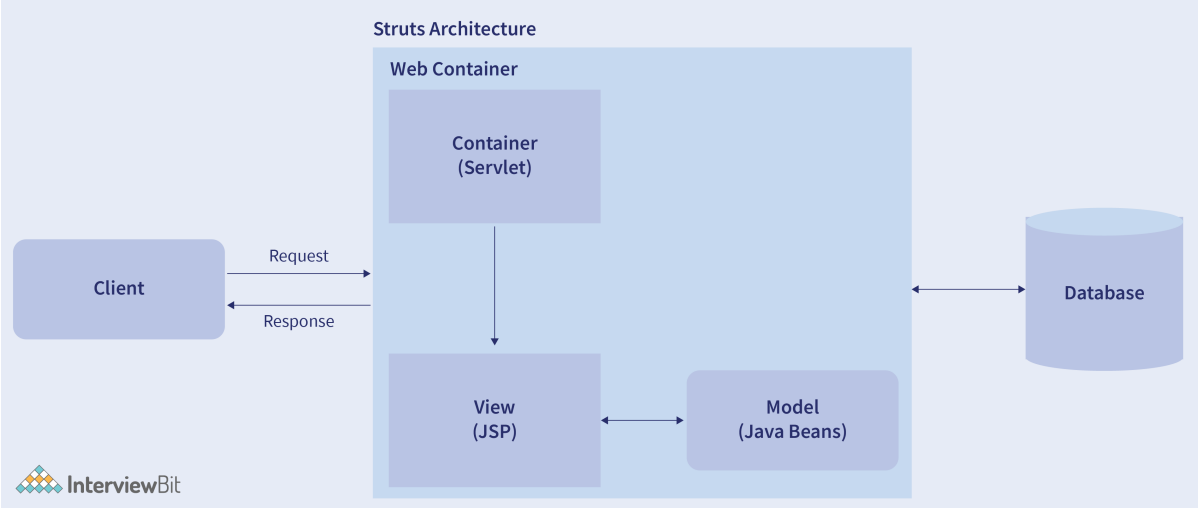
- **Transaction management**: EJBs can be used for achieving transaction management for databases by using annotations provided by EJB.
- **Database Connection Pool**: EJB can access connection pools defined in the J2EE server. The connection pools help in achieving abstraction of database connectivity and operations.
- **Security**: EJBs use JAAS (Java Authentication and Authorization Service) to develop secure applications. EJB methods can be authenticated and authorised with only configuration changes.
- **Scheduler Service**: EJBs can be used in the Timer Service which enables task implementation for further execution or repetitive execution.

11. What are the J2EE applets? Why can we use it?

Applets are J2EE client components that are written in Java and are executed in a web browser or a variety of other devices which supports the applet programming model. They are used for providing interactive features to web apps and help in providing small, portable embedded Java programs in HTML pages which will be run automatically when we view the pages.

12. What is the architecture model of Struts?

Strut is a combination of JSP, Java Servlets, messages and custom tags that together form an application development framework for developing enterprise-level applications. It is based on MVC (Model-View-Controller) architecture.

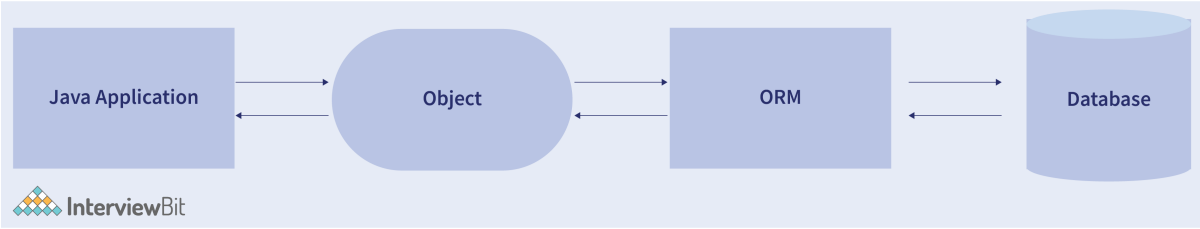


- **Model:** This component defines the internal system state. It can either be a Java Beans cluster or a single bean depending on the application architecture.
- **View:** Struts make use of JSP technology for designing views of enterprise-level applications.
- **Controller**: It is a servlet and is used for managing user actions that process requests and respond to them.

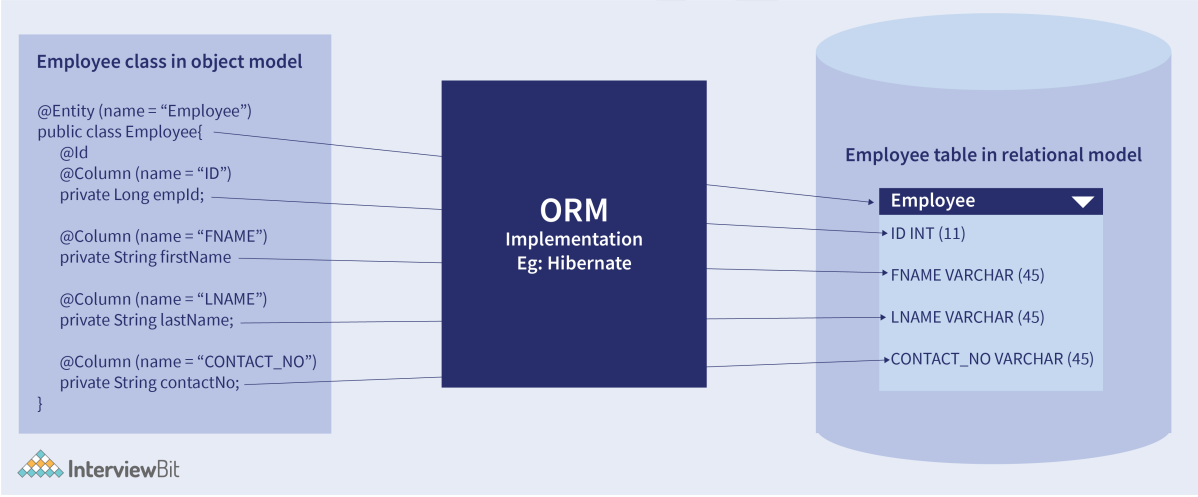
13. What do you understand by ORM?

ORM stands for Object-Relational Mapping that transforms objects of Java class to tables in relational databases and vice versa using metadata describing the mapping between the database and the objects.

This is represented as shown in the below image:



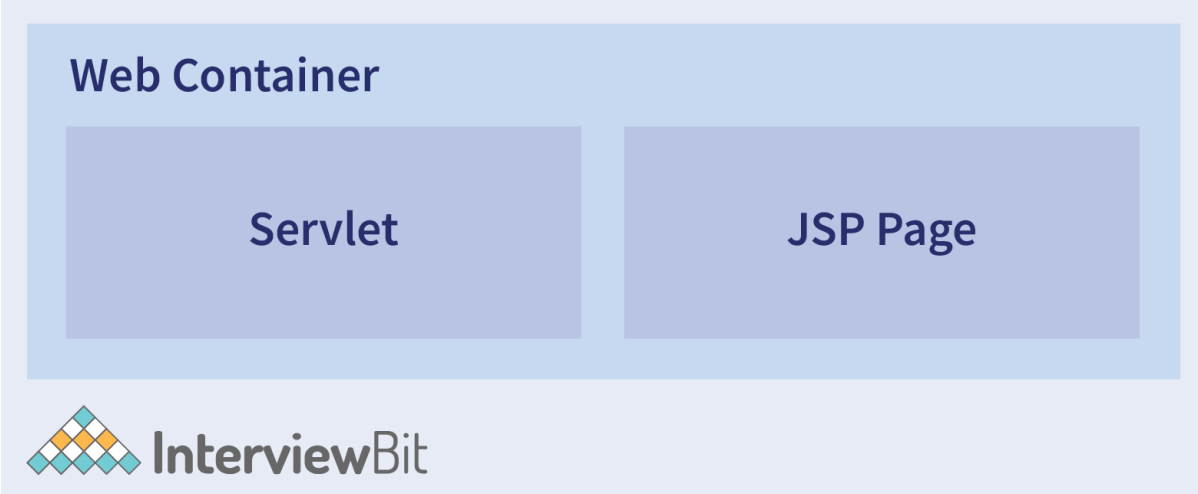
Consider an example where we have an Employee class having employeeId, firstName, lastName, contactNo as attributes. Consider we also have an Employee table that has ID, FNAME, LNAME and CONTACT_NO as columns. If we want to send data from our Java application and save it in the database, we cannot do it straightforwardly by simply saving the Java objects in the database directly. We need some sort of a mapper that maps the Java objects to the records that are compatible to be saved in the database table. This is where ORM comes into the picture. ORM helps in this transformation while writing data to the database as described in the below image:



The database records cannot be directly consumed by the Java applications as Java only deals with objects. ORM again plays a major role in the transformation of database records to Java objects.

14. What constitutes web components?

Java Servlets and Java Server Pages (JSP) components together constitute web components.



Java Servlets dynamically process requests and responses. JSP pages are used for executing servlets that allow a natural approach to creating static content.

15. What do you understand by JSF?

JSF stands for Java Server Faces which is a web framework that is intended for simplifying the development process for user interfaces. It is a standardized display technology for Java-based web applications. It is based on MVC (Model-View-Controller) pattern and provides reusable UI components.

16. What factors should a J2EE application possess for operating in a global economy?

Following are the factors that a J2EE application should possess to operate globally:

- **Financial Considerations:** Each country has its taxes, restrictions and tariffs depending on the government. All these factors should be considered while developing the J2EE application.
- **Language Requirements:** An application developed should support regional languages of the country for wider user coverage.
- **Legal Differences:** Every government has their custom laws, privacy laws and requirements for each country. An application developed should abide by all the rules of the land.

17. What are the differences between JVM vs JIT vs JDK vs JRE?

JVM	JIT	JDK	JRE
Java Virtual Machine: Introduced for managing system memory and also to provide a portable execution environment for Java applications.	Just in Time Compilation: This is a part of JVM and was developed for improving JVM performance.	Java Development Kit: JDK is a cross-platformed software development environment offering various collections of libraries and tools required for developing Java applications and applets.	Java Runtime Environment: JRE is part of JDK that consists of JVM, core classes and support libraries. It is used for providing a runtime environment for running Java programs.
Used for compiling byte code to machine code completely.	This is used for compiling only reusable byte code to machine code.	JDK is essential for writing and running programs in Java.	JRE is a subset of JDK and is like a container that consists of JVM, supporting libraries and other files. It doesn't have development tools such as compilers and debuggers.

J2EE Interview Questions for Experienced

18. What are the design goals of J2EE architecture?

The design goals of J2EE architecture are as follows:

- **Service Availability:** To ensure that the application is available 24*7 to achieve required business goals.
- **Data Connectivity:** The connection between a J2EE application and legacy systems should remain compatible enough for ensuring business functions.
- **Ease of Accessibility:** The user should be able to connect to applications using any device and from anywhere.
- **User Interaction:** The user interaction should be seamless and should be able to connect to different devices like desktops, mobiles, laptops etc.
- **Abstraction and Flexibility:** The developer should focus on business logic and the configuration details should be handled by the server.

19. What do you understand by Connectors? Can you describe the Connector Architecture?

Connectors are used for providing standard extension mechanisms to provide connectivity to different enterprise information systems. A connector architecture consists of resource adapters and system-level contracts, both of which are specific to enterprise information systems. The resource adapters are plugged into the container. The connector architecture defines certain contracts which a resource adapter must support for plugging into J2EE applications like security, transaction, resource management etc.

20. What do you understand by JRMP?

JRMP stands for Java Remote Method Protocol which is used for Remote Method Invocation (RMI) for passing Java objects as arguments. It is an underlying protocol used by RMI for marshalling objects as a stream during object serialization for transferring objects from one JVM to other.

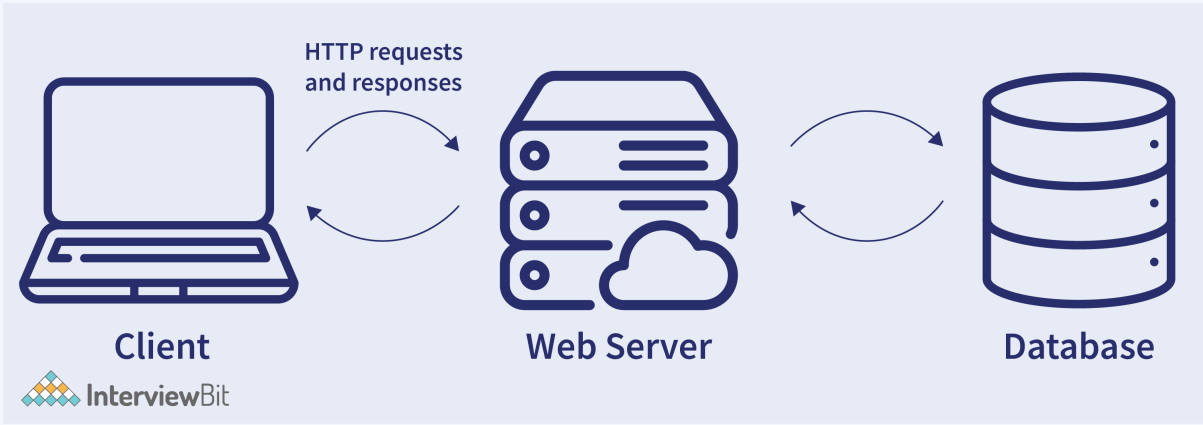
21. What happens if the database connected to the Java application via connection pool suddenly goes down?

Since the Java application uses a connection pool, it has active connections that would get disconnected if the database goes down. When the queries are executed to retrieve or modify data, then we will get a Socket exception.

22. How is 32-bit JVM different from 64-bit JVM?

64-bit JVM is used in 64-bit operating systems whereas 32-bit JVM is used for 32-bit operating systems. In 64-bit JVM, we can specify more heap size memory up to 100G when compared to the 4G limit of 32-bit JVM. Java applications take more memory while running in 64-bit JVM when compared to running the same application in 32-bit JVM. This is because of the increased size of the Ordinary Object Pointer. However, this can be bypassed by making use of the -XXCompressedOOP option of the JVM for telling to use 32-bit pointers. Additionally, 64-bit JVM uses 12 bytes object header size and a maximum of 8 bytes of internal references whereas the 32-bit JVM uses 8 bytes headers and a maximum of 4 bytes of internal references.

23. How is a webserver different from an application server?

Web Server	A
Web servers are computer programs that accept requests and returns responses based on that.  	
It constitutes the web container.	I
These are useful for getting static content for the applications.	T
This consumes and utilizes fewer resources.	I
Provide an environment for running web applications.	P
Web servers don't support multithreading.	M
This server makes use of HTML and HTTP protocols.	T s

24. What is the purpose of heap dumps and how do you analyze a heap dump?

Heap dumps consist of a snapshot of all live objects on Java heap memory that are used by running Java applications. Detailed information for each object like type, class name, address, size and references to other objects can be obtained in the heap dump. Various tools help in analyzing heap dumps in Java. For instance, JDK itself provides jhat tool for analysing heap dump. Heap dumps are also used for analysing memory leaks which is a phenomenon that occurs when there are objects that are not used by the application anymore and the garbage collection is not able to free that memory as they are still shown as referenced objects. Following are the causes that result in memory leaks:

- Continuously instantiating objects without releasing them.
- Unclosed connection objects (such as connections to the database) post the required operation.
- Static variables holding on to references of objects.
- Adding objects in HashMap without overriding hashCode() equals() method. If these methods are not included, then the hashmap will continuously grow without ignoring the duplicates.
- Unbounded caches.
- Listener methods that are uninvoked.

Due to this, the application keeps consuming more and more memory and eventually this leads to OutOfMemory Errors and can ultimately crash the application. We can make use of the Eclipse Memory Analyzer or jvisualVM tool for analysing heap dump to identify memory leaks.

25. How can we take a heap dump of a Java process?

There are multiple ways for taking heap dump of Java process. Tools like jCmd, jVisualVM, jmap are available for this purpose. For example, if we are using jmap, then heap dump can be taken by running the below command:

```
$ jmap -dump:live, file=/path/of/heap_dump.hprof  PID
```

This heap dump contains live objects that are stored in heap_dump.hprof file. Process ID (PID) of the Java process is needed to get the dump that can be obtained by using `ps` or `grep` commands.

26. How is J2EE different from Spring?

J2EE	Spring
J2EE is a standard or specification defined by Sun/Oracle which is used for web development.	Spring is a framework used for designing templates for an application.
J2EE has an Oracle-based license.	Spring is an open-source framework.
J2EE is based on a 3D framework- Logical Tiers, Client Tiers, and Presentation Tiers.	Spring is based on layered architecture having many modules that are made on top of the core container.
J2EE makes use of high-level object-oriented languages like Java.	Spring doesn't have a specific programming model.
J2EE is faster.	Spring is slower than J2EE.
Makes use of JTA API with execution.	Spring provides a layer of abstraction to help JTA execution merchants.

27. What are EAR, WAR, and JAR?

EAR stands for Enterprise Archive file and it consists of web, EJB and client components all compressed and packed into a file called .ear file. EAR files allow us to deploy different modules onto the application server simultaneously.

WAR stands for Web Archive file and consists of all web components packed and compressed in a .war file. This file allows testing and deploying web applications easily in a single request.

JAR stands for Java Archive file. It consists of all libraries and class files that constitute APIs. These are packed and compressed in a file called the .jar file. These are used for deploying the entire application including classes and resources in a single request.

28. What do you know about Hibernate?

Hibernate is an Object Relational Mapper framework in Java that provides a layer of abstraction for retrieving or modifying data in the database. It handles all the implementations internally and the developer need not worry about how the connections to the databases are made, how the data translation from Java application to Database and vice versa happens. Hibernate supports powerful object-oriented concepts like inheritance, association, polymorphism, compositions, collections that help in making queries using the Java approach by using HQL (Hibernate Query Language).

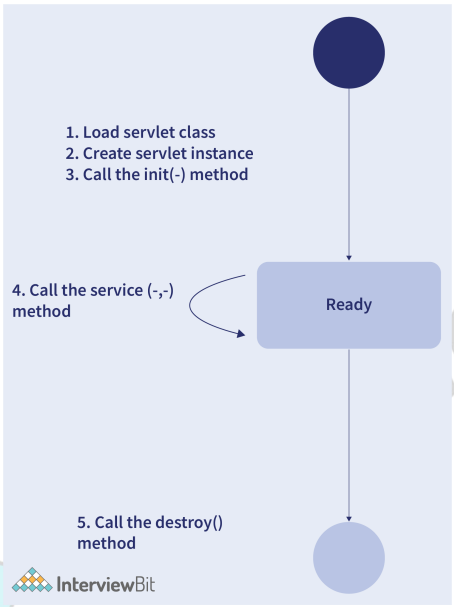
29. What are deployment descriptors used for?

Servlets are server-side components that aid in developing powerful server-side applications. Ther are servers that are platform-independent and follow various protocols as per the application design. The most commonly used protocol is the HTTP protocol. In Java, we can create servlets by implementing the Servlet interface that has 3 lifecycle methods - init, service and destroy - and we can use the below classes for implementing servlets:

- javax.servlet.http.HttpServletRequest
- javax.servlet.http.HttpServletResponse
- javax.servlet.http.HttpSession.

30. Can you describe the phases of the servlet lifecycle?

The below image describes the different phases of the servlet lifecycle:



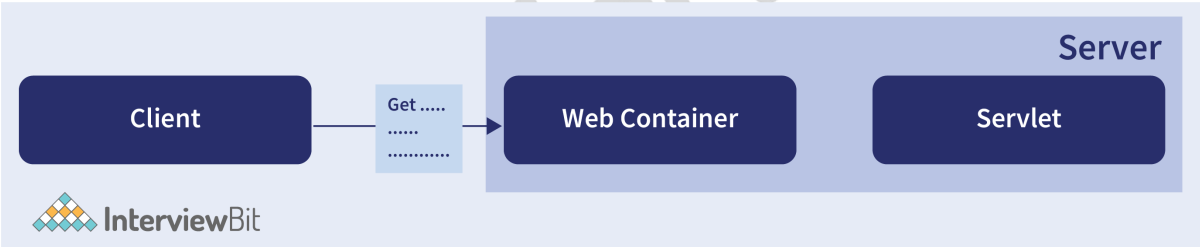
There are five phases, are as follows:

- **Classloading phase:** The first step is to load the servlet class file (.class extension) by the web container.
- **Instantiation phase:** Next step is to instantiate the servlet by calling the default constructor.
- **Initialize phase:** In this phase, the init() method of the servlet is run where the servlet configuration will be assigned to the servlet. This is a lifecycle method provided by the Servlet interface which is run only once in the servlet lifetime.
- **Request Handling phase**: Here, the servlets provide services to different requests by making use of the service() method of the Servlet interface.
- **Removal phase:** In this phase, the destroy() lifecycle method of the Servlet interface will be called that is used for clearing the configuration and closing resources before servlet destruction. Post this, the garbage collection will take place.

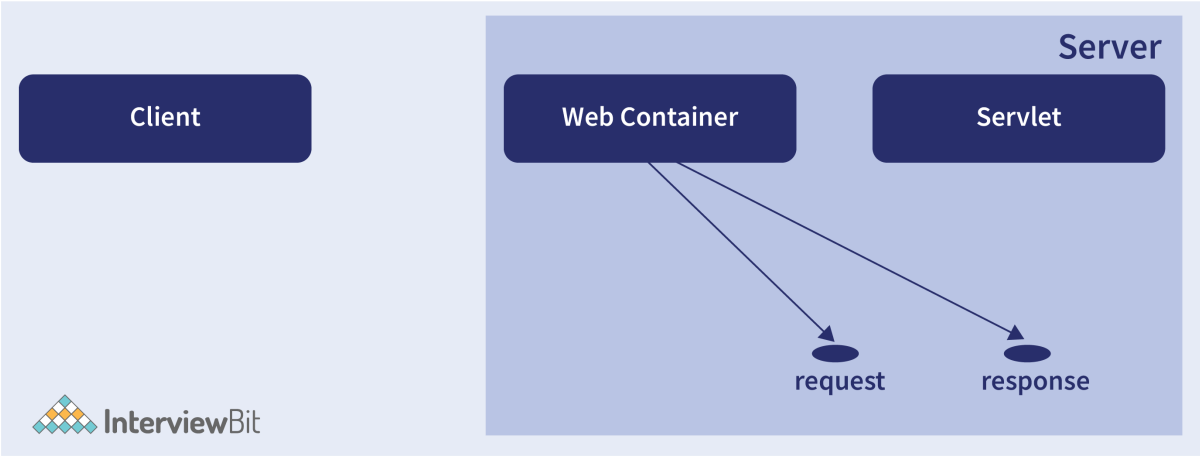
31. How does a servlet application work?

A Java servlet is typically multithreaded. This means that multiple requests can be sent to the same servlet and they can be executed at the same time. All the local variables (not pointing to shared resources) inside the servlet are automatically thread-safe and are request specific. Care has to be taken when the servlet is accessing or modifying the global shared variable. The servlet instance lifecycle for different requests are managed by the web container as follows:

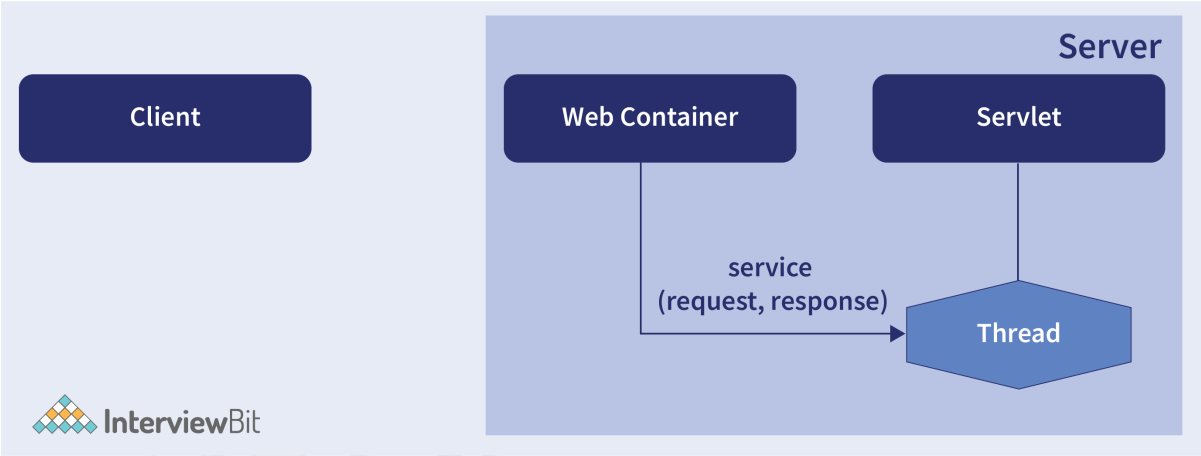
- User clicks on a link in a client that requests a response from a server. In this instance, consider that the client performs GET request to the server as shown in the image below:



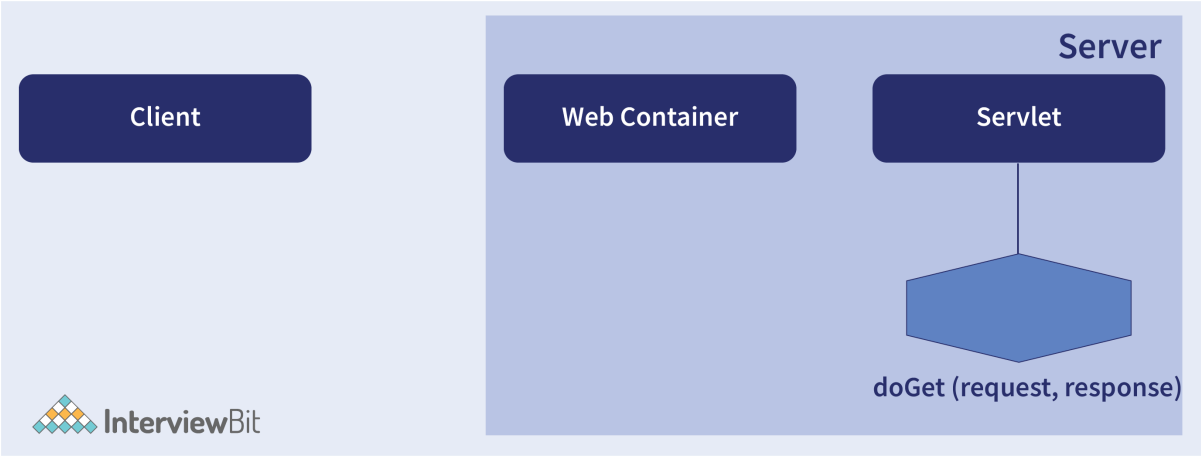
- The web container intercepts the request and identifies which servlet has to serve the request by using the deployment descriptor file and then creates two objects as shown below-
  - HttpServletRequest - to send servlet request
  - HttpServletResponse - to get the servlet response



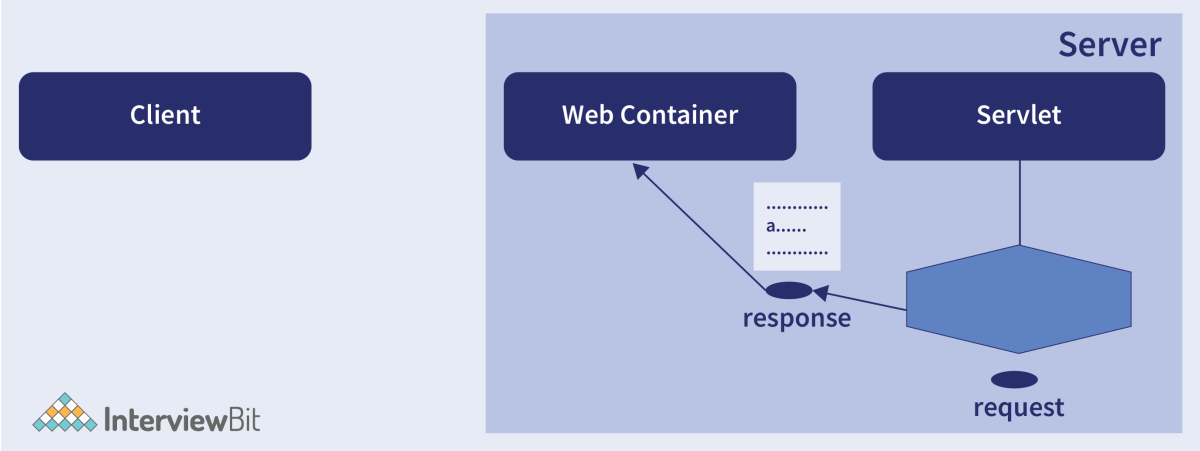
- The web container then creates and allocates a thread that inturn creates a request that calls the service() lifecycle method of the servlet and passes the request and response objects as parameters as shown below:



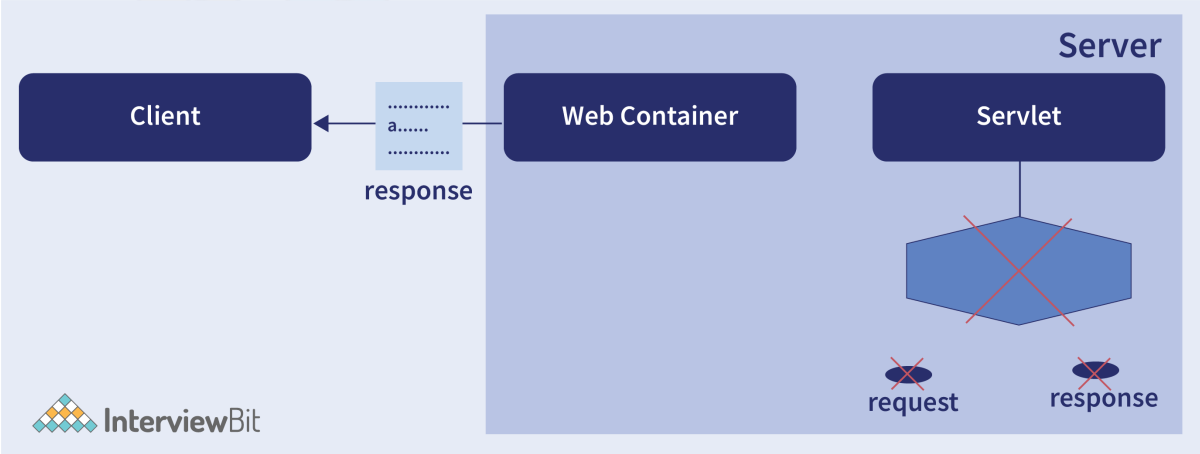
- service() method decides which servlet method - doGet() or doPost() or doPut() or doDelete()- depending on the HTTP requests received from the client. In our case, we got the GET request from the client and hence the servlet will call the doGet() method as described below.



- Servlet makes use of the response object obtained from the servlet method for writing the response to the client.



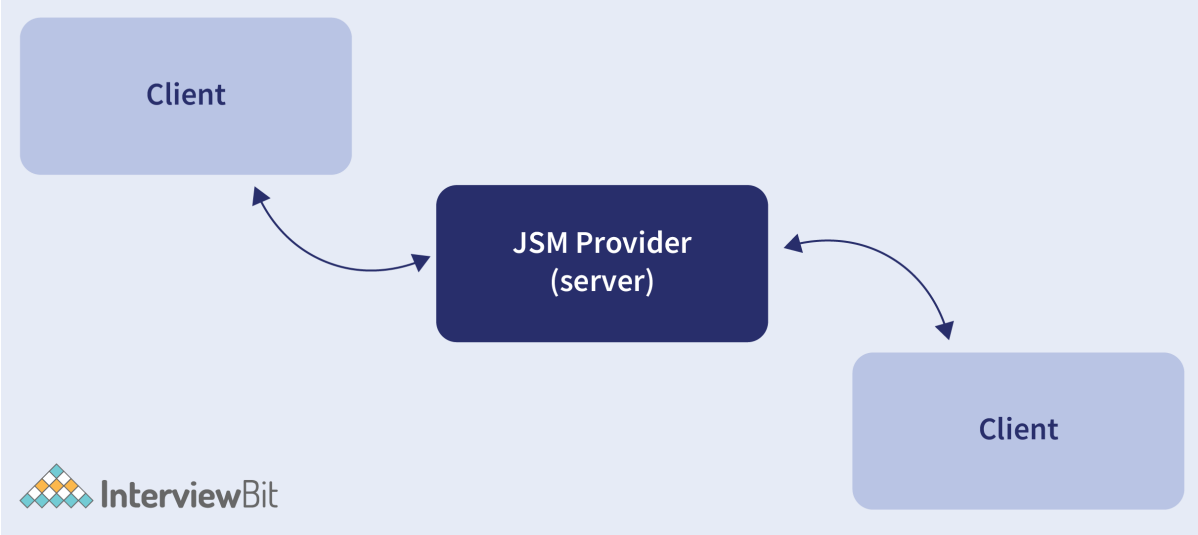
- Once the request is served completely, the thread dies and the objects are made ready for garbage collection.



32. What do you understand by Java Message Service (JMS)?

JMS is a Java-based API that is like a gateway to the message-oriented middleware like SonicMQ, IBM MQSeries etc. It provides a mechanism for sending and receiving messages by making use of the publishing/subscribe (1 message multiple receivers) model or point-to-point (1 message 1 receiver) paradigm from one client to another.

The following image describes how 2 clients can communicate with each other utilizing JMS providers.



Conclusion:

J2EE defines standards and specifications for various components such as e-mailing, database connectivity, security, XML parsing, CORBA communication etc that help in developing complex, reliable, secure and distributed servlets and applications that follow the client-server model. It provides various API interfaces that act as standards between different vendor adapters and J2EE components. This ensures that the application components are not dependent on vendor codes. Due to this, J2EE has been very popular among Java developers in the field of software development.

Links to More Interview Questions

C Interview Questions	Php Interview Questions	C Sharp Interview Questions
Web Api Interview Questions	Hibernate Interview Questions	Node Js Interview Questions
Cpp Interview Questions	Oops Interview Questions	Devops Interview Questions
Machine Learning Interview Questions	Docker Interview Questions	Mysql Interview Questions
Css Interview Questions	Laravel Interview Questions	Asp Net Interview Questions
Django Interview Questions	Dot Net Interview Questions	Kubernetes Interview Questions
Operating System Interview Questions	React Native Interview Questions	Aws Interview Questions
Git Interview Questions	Java 8 Interview Questions	Mongodb Interview Questions
Dbms Interview Questions	Spring Boot Interview Questions	Power Bi Interview Questions
Pl Sql Interview Questions	Tableau Interview Questions	Linux Interview Questions
Ansible Interview Questions	Java Interview Questions	Jenkins Interview Questions