

J2EE Interview Questions

1) What is J2EE?

J2EE is an environment for developing and deploying enterprise applications. The J2EE platform consists of a set of services, application programming interfaces (APIs), and protocols that provide the functionality for developing multi tiered, and web-based applications.

2) What is the J2EE module?

A J2EE module consists of one or more J2EE components for the same container type and one component deployment descriptor of that type.

3) What are the components of J2EE application?

A J2EE component is a self-contained functional software unit that is assembled into a J2EE application with its related classes and files and communicates with other components. The J2EE specification defines the following J2EE components:

- Application clients and applets are client components.
- Java Servlets and Java Server Pages™ (JSP™) technology components are web components.
- Enterprise JavaBeans™ (EJB™) components (enterprise beans) are business components.
- Resource adapter components provided by EIS and tool vendors.

4) What are the four types of J2EE modules?

1. Application client module
2. Web module
3. Enterprise JavaBeans module
4. Resource adapter module

5) What does application client module contain?

The application client module contains:

- class files,
- an application client deployment descriptor.

Application client modules are packaged as JAR files with a .jar extension.

6) What does Enterprise JavaBeans module contain?

The Enterprise JavaBeans module contains:

- class files for enterprise beans
- An EJB deployment descriptor.

EJB modules are packaged as JAR files with a .jar extension.

7) What does resource adapt module contain?

The resource adapt module contains:

- all Java interfaces,
- classes,
- native libraries,
- other documentation,
- A resource adapter deployment descriptor.

Resource adapter modules are packages as JAR files with a .rar (Resource adapter Archive) extension.

8) How many development roles are involved in J2EE application?

There are at least 5 roles involved:

1. Enterprise Bean Developer
 - Writes and compiles the source code
 - Specifies the deployment descriptor
 - Bundles the .class files and deployment descriptor into an EJB JAR file
2. Web Component Developer
 - Writes and compiles Servlets source code
 - Writes JSP and HTML files
 - Specifies the deployment descriptor for the Web component
 - Bundles the .class, .jsp, .html, and deployment descriptor files in the WAR file
3. J2EE Application Client Developer
 - Writes and compiles the source code
 - Specifies the deployment descriptor for the client
 - Bundles the .class files and deployment descriptor into the JAR file
4. Application Assembler The application assembler is the company or person who receives application component JAR files from component providers and assembles them into a J2EE application EAR file. The assembler or deployer can edit the deployment descriptor directly or use tools that correctly add XML tags according to interactive selections. A software developer performs the following tasks to deliver an EAR file containing the J2EE application:
 - Assembles EJB JAR and WAR files created in the previous phases into a J2EE application (EAR) file
 - Specifies the deployment descriptor for the J2EE application
 - Verifies that the contents of the EAR file are well formed and comply with the J2EE specification
5. Application Deployer and Administrator
 - Configures and deploys the J2EE application
 - Resolves external dependencies
 - Specifies security settings & attributes
 - Assigns transaction attributes and sets transaction controls
 - Specifies connections to databases
 - Deploys or installs the J2EE application EAR file into the J2EE server
 - Administers the computing and networking infrastructure where J2EE applications run
 - Oversees the runtime environment

But a developer role depends on the job assignment. For a small company, one developer may take these 5 roles altogether.

9) What is difference between J2EE 1.3 and J2EE 1.4?

J2EE 1.4 is an enhancement version of J2EE 1.3. It is the most complete Web services platform ever.

J2EE 1.4 includes:

- Java API for XML-Based RPC (JAX-RPC 1.1)
- SOAP with Attachments API for Java (SAAJ),
- Web Services for J2EE(JSR 921)
- J2EE Management Model(1.0)
- J2EE Deployment API(1.1)
- Java Management Extensions (JMX),
- Java Authorization Contract for Containers(JavaACC)
- Java API for XML Registries (JAXR)
- Servlet 2.4
- JSP 2.0
- EJB 2.1
- JMS 1.1
- J2EE Connector 1.5

The J2EE 1.4 features complete Web services support through the new JAX-RPC 1.1 API, which supports service endpoints based on Servlets and enterprise beans. JAX-RPC 1.1 provides interoperability with Web services based on the WSDL and SOAP protocols. The J2EE 1.4 platform also supports the Web Services for J2EE specification (JSR 921), which defines deployment requirements for Web services and utilizes the JAX-RPC programming model.

In addition to numerous Web services APIs, J2EE 1.4 platform also features support for the WS-I Basic Profile 1.0. This means that in addition to platform independence and complete Web services support, J2EE 1.4 offers platform Web services interoperability.

The J2EE 1.4 platform also introduces the J2EE Management 1.0 API, which defines the information model for J2EE management, including the standard Management EJB (MEJB). The J2EE Management 1.0 API uses the Java Management Extensions API (JMX).

The J2EE 1.4 platform also introduces the J2EE Deployment 1.1 API, which provides a standard API for deployment of J2EE applications.

The J2EE 1.4 platform includes security enhancements via the introduction of the Java Authorization Contract for Containers (JavaACC). The JavaACC API improves security by standardizing how authentication mechanisms are integrated into J2EE containers.

The J2EE platform now makes it easier to develop web front ends with enhancements to Java Servlet and JavaServer Pages (JSP) technologies. Servlets now support request listeners and enhanced filters. JSP technology has simplified the page and extension development models with the introduction of a simple expression language, tag files, and a simpler tag extension API, among other features. This makes it easier than ever for developers to build JSP-enabled pages, especially those who are familiar with scripting languages.

Other enhancements to the J2EE platform include the J2EE Connector Architecture, which provides incoming resource adapter and Java Message Service (JMS) plug ability. New features in Enterprise JavaBeans (EJB) technology include Web service endpoints, a timer service, and enhancements to EJB QL and message-driven beans.

The J2EE 1.4 platform also includes enhancements to deployment descriptors. They are now defined using XML Schema which can also be used by developers to validate their XML structures.

Note: The above information comes from SUN released notes.

10) Is J2EE application only a web-based?

NO. A J2EE application can be web-based or non-web-based. If an application client executes on the client machine, it is a non-web-based J2EE application. The J2EE application can provide a way for users to handle tasks such as J2EE system or application administration. It typically has a graphical user interface created from Swing or AWT APIs, or a command-line interface. When user request, it can open an HTTP connection to establish communication with a Servlet running in the web tier.

11) Are JavaBeans J2EE components?

NO. JavaBeans components are not considered J2EE components by the J2EE specification. JavaBeans components written for the J2EE platform have instance variables and get and set methods for accessing the data in the instance variables. JavaBeans components used in this way are typically simple in design and implementation, but should conform to the naming and design conventions outlined in the JavaBeans component architecture.

12) Is HTML page a web component?

NO. Static HTML pages and applets are bundled with web components during application assembly, but are not considered web components by the J2EE specification. Even the server-side utility classes are not considered web components, either.

13) What is the container?

A container is a runtime support of a system-level entity. Containers provide components with services such as lifecycle management, security, deployment, and threading.

14) What is the web container?

Servlet and JSP containers are collectively referred to as Web containers.

15) What is the thin client?

A thin client is a lightweight interface to the application that does not have such operations like query databases, execute complex business rules, or connect to legacy applications.

16) What are types of J2EE clients?

- Applets
- Application clients
- Java Web Start-enabled rich clients, powered by Java Web Start technology.
- Wireless clients, based on Mobile Information Device Profile (MIDP) technology.

17) What is deployment descriptor?

A deployment descriptor is an Extensible Markup Language (XML) text-based file with an .xml extension that describes a component's deployment settings. A J2EE application and each of its modules has its own deployment descriptor.

18) What is the EAR file?

An EAR file is a standard JAR file with an .ear extension, named from Enterprise Archive file. A J2EE application with all of its modules is delivered in EAR file.

19) What are JTA and JTS?

JTA is the abbreviation for the Java Transaction API. JTS is the abbreviation for the Java Transaction Service. JTA provides a standard interface and allows you to demarcate transactions in a manner that is independent of the transaction manager implementation. The J2EE SDK implements the transaction manager with JTS. But your code doesn't call the JTS methods directly. Instead, it invokes the JTA methods, which then call the lower-level JTS routines.

Therefore, JTA is a high level transaction interface that your application uses to control transaction. And JTS is a low level transaction interface and EJBs uses behind the scenes (client code doesn't directly interact with JTS. It is based on object transaction service (OTS) which is part of CORBA.

20) What is JAXP?

JAXP stands for Java API for XML. XML is a language for representing and describing text-based data which can be read and handled by any program or tool that uses XML APIs.

21) What is J2EE Connector?

The J2EE Connector API is used by J2EE tools vendors and system integrators to create resource adapters that support access to enterprise information systems that can be plugged into any J2EE product. Each type of database or EIS has a different resource adapter.

22) What is JAAS?

The Java Authentication and Authorization Service (JAAS) provide a way for a J2EE application to authenticate and authorize a specific user or group of users to run it. It is a standard Pluggable Authentication Module (PAM) framework that extends the Java 2 platform security architecture to support user-based authorization.

23) What is Model 1?

Using JSP technology alone to develop Web page. Such term is used in the earlier JSP specification. Model 1 architecture is suitable for applications that have very simple page

flow, have little need for centralized security control or logging, and change little over time. Model 1 applications can often be refactored to Model 2 when application requirements change.

24) What is Model 2?

Using JSP and Servlet together to develop Web page. Model 2 applications are easier to maintain and extend, because views do not refer to each other directly.

25) What is Struts?

A Web page development framework. Struts combine Java Servlets, Java Server Pages, custom tags, and message resources into a unified framework. It is a cooperative, synergistic platform, suitable for development teams, independent developers, and everyone between.

26) How is the MVC design pattern used in Struts framework?

In the MVC design pattern, application flow is mediated by a central Controller. The Controller delegates requests to an appropriate handler. The handlers are tied to a Model, and each handler acts as an adapter between the request and the Model. The Model represents, or encapsulates, an application's business logic or state. Control is usually then forwarded back through the Controller to the appropriate View. The forwarding can be determined by consulting a set of mappings, usually loaded from a database or configuration file. This provides a loose coupling between the View and Model, which can make an application significantly easier to create and maintain.

Controller--Servlet controller which supplied by Struts itself; View --- what you can see on the screen, a JSP page and presentation components; Model --- System state and a business logic JavaBeans.

27) Do you have to use design pattern in J2EE project?

Yes. If I do it, I will use it. Learning design pattern will boost my coding skill.

28) Is J2EE a super set of J2SE?

Yes

29) What does web module contain?

The web module contains:

- JSP files,
- class files for Servlets,
- GIF and HTML files, and
- A Web deployment descriptor.

Web modules are packaged as JAR files with a .war (Web Archive) extension.

30) What APIs are available for developing a J2EE application?

- Enterprise JavaBeans Technology(3 beans: Session Beans, Entity Beans and Message-Driven Beans)
- JDBC API(application level interface and service provider interface or driver)
- Java Servlets Technology(Servlet)
- Java ServerPage Technology(JSP)
- Java Message Service(JMS)
- Java Naming and Directory Interface(JNDI)
- Java Transaction API(JTA)
- JavaMail API
- JavaBeans Activation Framework(JAF used by JavaMail)
- Java API for XML Processing(JAXP,SAX, DOM, XSLT)
- Java API for XML Registries(JAXR)
- Java API for XML-Based RPC(JAX-RPC)-SOAP standard and HTTP
- SOAP with Attachments API for Java(SAAJ)-- low-level API upon which JAX-RPC depends
- J2EE Connector Architecture
- Java Authentication and Authorization Service(JAAS)