

Q1. What is Jakarta EE and how does it relate to Java EE?

A: Jakarta EE is the community-governed evolution of Java EE after Oracle transferred stewardship to the Eclipse Foundation. The namespace changed from `javax.*` to `jakarta.*` in EE 9. It standardizes enterprise APIs like CDI, JAX-RS, JPA, and JMS.

Q2. What is CDI and what are its main bean scopes?

A: CDI (Contexts and Dependency Injection) manages the lifecycle and injection of beans. `@RequestScoped` creates one instance per HTTP request. `@SessionScoped` per HTTP session. `@ApplicationScoped` is a singleton shared across the application.

Q3. How do you build a RESTful endpoint with JAX-RS?

A: Annotate the class with `@Path('/users')` and method with `@GET/@POST` plus `@Produces(MediaType.APPLICATION_JSON)`. `@PathParam`, `@QueryParam`, and `@RequestBody` bind URL parts and request data. Extend `Application` or use a Jakarta EE server's auto-scan.

Q4. When would you use a `@Stateless` EJB vs `@Stateful` vs `@Singleton`?

A: `@Stateless` handles requests without client-specific state and is pooled ideal for services. `@Stateful` maintains conversational state per client session use for shopping carts. `@Singleton` has one instance per JVM and supports concurrent access control.

Q5. How does JMS work and what is the difference between Queue and Topic?

A: A JMS Queue delivers each message to exactly one consumer (point-to-point). A Topic delivers the message to all subscribers (publish-subscribe). `@MessageDriven` EJBs or JMS listeners process messages asynchronously from the broker.

Q6. How do you configure JPA persistence in a Jakarta EE application?

A: Define `META-INF/persistence.xml` listing the persistence unit, JTA data source, and entity classes. Inject the `EntityManager` with `@PersistenceContext(unitName='myPU')`. The container manages transactions, so use `@Transactional` or container-managed TX.

Q7. What is JNDI and how do you look up a resource by name?

A: JNDI (Java Naming and Directory Interface) is a registry for server resources. `@Resource(lookup='java:comp/env/jdbc/myDS')` injects a `DataSource` directly. Programmatic lookup uses `InitialContext ctx = new InitialContext(); ctx.lookup('java:...')`.

Q8. How do Jakarta EE security mechanisms protect a JAX-RS endpoint?

A: `@RolesAllowed('ADMIN')` on a resource method restricts access to principals with that role. The security domain, identity store (`@DatabaseIdentityStoreDefinition`), and authentication mechanism (`@FormAuthenticationMechanismDefinition`) are configured via annotations.

Q9. What is the difference between WAR, EAR, and executable JAR packaging?

A: A WAR is a web archive deployed to a servlet container or app server. An EAR bundles multiple WARs and EJB JARs for a full app server. Modern Jakarta EE apps often use executable JARs with embedded servers (WildFly Bootable JAR, Open Liberty) for cloud deployment.

Q10. How do CDI interceptors differ from EJB interceptors?

A: CDI interceptors are bound to beans via `@InterceptorBinding`, a custom annotation that marks methods to intercept. EJB interceptors use `@Interceptors(MyInterceptor.class)` on the bean class. CDI interceptors work on any CDI-managed bean, not just EJBs.