Understanding and Designing with EJB

B.Ramamurthy

Review

- Request/Response Model
- Distributed Objects: stubs and skeleton providing location transparency
- Naming and Lookup: registry and binding
- Server-side technology: servlets (project1)
- Web applications: can be written entirely using Java Server Pages (static and dynamic content and data access can be provided); JSP is wrapper on servlet technology.
- Concept of initial context:The starting point for resolution of names for naming and directory operations.
- Data base access: using Java Data Base Connectivity

When to use EJB

- For large scale applications: where resources and data are distributed.
- When the application is run on servers at many locations.
- Where scalability is critical.
- Where transactions are required to ensure data integrity
- When a variety of clients need to handled.

Types of Enterprise Bean: Session

- **Session bean**: represents a single client inside the J2EE server. Session represents an interactive session. When a client terminates the session bean terminates/is no longer associated with the client.
- Stateful session bean: maintains a conversational state for the duration of a session. Ex: items reviewed in a session at some sites
- Stateless session bean: does not maintain a conversational state. Ex: computing a formula for a given value

Types of Enterprise Bean: Entity

- An entity bean represents a business object in a persistent storage mechanism. Ex: customers, orders, and products.
- Each entity bean typically has an underlying table in a relational database (business data), and each instance of the bean corresponds to a row in that table.
- Transactional and recoverable on a server crash.

Types of Enterprise Bean: Message-Driven

- A message driven bean is an enterprise bean that allows J2EE applications to process messages asynchronously.
- It acts as a JMS listener, which is similar to an event listener except that it receives messages instead of events.
- The messages can be sent by any J2EE component: an application client, another enterprise bean, or a web component, or a non-J2EE system using JMS.
- Retain no data or conversational state.

Contents of an Enterprise Bean

- Interfaces: The remote and home interface for remote access. Local and local home accesses for local access.
- Enterprise bean class: Implements the methods defined in the above interfaces.
- Deployment descriptor: An XML file that specifies information about the bean such as its type, transaction attributes, etc.
- Helper classes: non-bean classes needed by the enterprise bean class such as utility and exception classes.

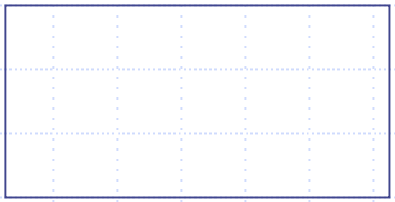
Naming Conventions

Item	Syntax	Example
Enterprise Bean Name (DD)	**<name>EJB**	**AccountEJB**
EJB JAR display name (DD)	**<name>EJB**	**AccountJAR**
Enterprise bean class	**<name>Bean**	**AccountBean**
Home interface	**<name>Home**	**AccountHome**
Remote interface	**<name>**	**Account**
Local home interface	**Local<name>Home**	**LocalAccountHome**
Local interface	**Local<name>**	**LocalAccount**
Abstract Schema (DD)	**<name>**	**Account**

The life cycles of enterprise beans

- An enterprise bean goes through various stages during its lifetime. Each type has different life cycle.

Session bean



Entity and Message-driven Bean Lifecycle

Designing an application

- Start with Remote interface methods.
- For completion write the Home interface.
- Implement these methods in a (session) bean class.
- Update build.xml “ant” file and compile using ant <target>
- Use the deploy tool to deploy the application on your j2EE server and set up the configuration.
- Write a client (preferably a web client) to test your enterprise application.
- Lets go through converter application. Your assignment is to make it a more meaningful and useful converter.