Chap:6

EJB(Enterprise JavaBean)

Prepared: JINAL JOSHI

Syllabus Content

- Introduction
- Benefits of EJB
- Restriction on EJB
- Types of EJB
  - 1.Session Beans
  - 2.Entity Beans
  - 3.Message-driven beans
- Timer service

- **Introduction of EJB**

Introduction of EJB

- EJB (*Enterprise Java Bean*) is used to develop scalable, robust and secured enterprise applications in java.
- EJB stands for **Enterprise Java Beans**.
- EJB is an essential part of a J2EE platform.

- J2EE platform has component based architecture to provide multi-tiered, distributed and highly transactional features to enterprise level applications.

Introduction of EJB

- EJB stands for Enterprise Java Beans. EJB is an essential part of a J2EE platform. J2EE platform has component based architecture to provide multi-tiered, distributed and highly transactional features to enterprise level applications.

- EJB provides an architecture to develop and deploy component based enterprise applications considering robustness, high scalability, and high performance.

- An EJB application can be deployed on any of the application server compliant with the J2EE 1.3 standard specification. We'll be discussing EJB 3.0 in detail in this tutorial.

- Benefits(uses) of EJB

Benefits of EJB

- EJB provides an architecture to develop and deploy component based enterprise applications considering robustness, high scalability, and high performance.

- An EJB application can be deployed on any of the application server compliant with the J2EE 1.3 standard specification.
- To run EJB application, you need an *application server* (EJB Container) such as Jboss, Glassfish, Web logic, Web sphere etc. It performs:

- life cycle management,
- security,
- transaction management, and
- object pooling.
- EJB application is deployed on the server, so it is called server side component also.

Benefits of EJB

- Simplified development of large-scale enterprise level application.

- Application Server/EJB container provides most of the system level services like transaction handling, logging, load balancing, persistence mechanism, exception handling, and so on.

- Developer has to focus only on business logic of the application.

- EJB container manages life cycle of EJB instances, thus developer needs not to worry about when to create/delete EJB objects.

- Restriction(limits) on EJB

Restriction on EJB

- Requires application server
- Requires only java client. For other language client, you need to go for webservice.
- Complex to understand and develop EJB applications.

- Types of EJB

Types of EJB

- EJB is primarily divided into three categories; following table lists their names with brief descriptions:

- 1. Session Bean
- 2.Entity Bean
- 3. Message Driven Bean

Types of EJB

- **Session Bean:**

- Session bean encapsulates business logic only, it can be invoked by local, remote and webservice client.
- It can be used for calculations, database access etc.
- The life cycle of session bean is maintained by the application server (EJB Container).

Types of Session Bean

There are 3 types of session bean.

1) Stateless Session Bean: It doesn't maintain state of a client between multiple method calls.

2) Stateful Session Bean: It maintains state of a client across multiple requests.

3) Singleton Session Bean: One instance per application, it is shared between clients and supports concurrent access.

Types of EJB

- **2.Entity Bean:**
- Entity beans represent persistent data storage. User data can be saved to database via entity beans and later on can be retrieved from the database in the entity bean.
- Entity bean represents the persistent data stored in the database. It is a server-side component.

- In EJB 2.x, there was two types of entity beans:
- **1.bean managed persistence** (BMP) and
- 2.**container managed persistence** (CMP).

Types of EJB

- 3**. Message Driven Bean(MDB)**
- A message driven bean (MDB) is a bean that contains business logic. But, it is invoked by passing the message. So, it is like JMS Receiver.
- MDB asynchronously receives the message and processes it.

A message driven bean receives message from queue or topic, so you must have the knowledge of JMS API.

- **Timer service**

Timer service

- Timer Service is a mechanism by which scheduled application can be build. For example, salary slip generation on the 1st of every month.

- EJB 3.0 specification has specified @Timeout annotation, which helps in programming the EJB service in a stateless or message driven bean.
- EJB Container calls the method, which is annotated by @Timeout.

Timer service

- The TimerService interface provides enterprise bean components with access to the container-provided Timer Service.

- The EJB Timer Service allows stateless session beans, singleton session beans, message-driven beans . To create a timer, we need to create TimerService object and use one of its createTimer() method.

- The Timer Service is accessed via dependency injection, through the getTimerService method of the EJBContext interface, or through lookup in the JNDI namespace.

Timer service

- Example:

```
public class IntervalTimerDemo
{
   public void createTimer(long milliseconds);
}
```

Assignment Question

- Q.1 Write short note on EJB
- Q.2 Write uses of EJB
- Q.3 Write limitations of EJB
- Q.4 Explain types of EJB
- Q.5 Explain timer service in EJB.