**What are the Libraries Inside JEE?**

Java EE provides a rich set of **APIs (libraries/frameworks)** to build scalable, secure, and enterprise-grade applications.

**🔹 1. Servlet API**

**📌 Purpose:**

Handles HTTP **requests and responses** between the **client and server**.

**🛠 Features:**

* Request processing
* Session management
* Form handling

**🏦 Banking Example:**

A servlet receives login credentials from a **net banking** form and forwards it to the authentication logic.

**🛒 E-commerce Example:**

Handles checkout form submission with delivery address and payment method.

**🔹 2. JSP (JavaServer Pages)**

**📌 Purpose:**

Create **dynamic web content** by embedding Java code in HTML.

**🛠 Features:**

* Easy UI generation
* Works with Servlets
* Supports tag libraries (JSTL)

**🏦 Banking Example:**

Displays user account summary or transaction history in HTML after retrieving data from the backend.

**🛒 E-commerce Example:**

Generates dynamic product listings or order confirmation page.

**🔹 3. JSF (JavaServer Faces)**

**📌 Purpose:**

Component-based **UI framework** for building rich web interfaces.

**🛠 Features:**

* Reusable UI components
* Event handling
* Data binding

**🏥 Healthcare Example:**

Patient portal with UI components like date pickers for appointment scheduling.

**🛒 E-commerce Example:**

Drag-and-drop shopping cart with AJAX support.

**🔹 4. EJB (Enterprise JavaBeans)**

**📌 Purpose:**

Encapsulates **business logic**, transactions, and security.

**🛠 Features:**

* Declarative transactions
* Remote access
* Dependency injection

**🏦 Banking Example:**

An EJB handles **fund transfer**, applying interest, and maintaining logs—all within a single transaction.

**📞 Telecom Example:**

Manages mobile plan upgrades and customer usage billing in the backend.

**🔹 5. JPA (Java Persistence API)**

**📌 Purpose:**

Maps **Java objects to database tables** (ORM - Object Relational Mapping).

**🛠 Features:**

* Entity classes
* JPQL queries
* Relationships (OneToMany, etc.)

**🏦 Banking Example:**

A CustomerEntity maps to the CUSTOMER table, allowing account info to be queried or updated easily.

**🛒 E-commerce Example:**

Maps Order, Product, and Cart Java classes to relational database tables.

**🔹 6. JDBC (Java Database Connectivity)**

**📌 Purpose:**

Allows **direct access to relational databases** using SQL.

**🛠 Features:**

* Execute raw SQL queries
* Database connection management
* Manual transaction control

**🏦 Banking Example:**

Used to run a **stored procedure** that calculates credit card interest.

**📞 Telecom Example:**

Retrieve mobile usage records using raw SQL from Oracle DB.

**🔹 7. JMS (Java Message Service)**

**📌 Purpose:**

Handles **asynchronous messaging** between components or systems.

**🛠 Features:**

* Queue and Topic models
* Durable subscriptions
* Message persistence

**🏦 Banking Example:**

When a large batch transaction is initiated, a **JMS message** triggers downstream validation or notification systems.

**📞 Telecom Example:**

After recharging, a **message is published** to notify the billing system and SMS service.

**🔹 8. JAX-RS and JAX-WS**

**📌 Purpose:**

Build **web services** – RESTful (JAX-RS) or SOAP (JAX-WS)

**🛠 Features:**

* Annotation-based APIs
* JSON/XML support
* Stateless communication

**🏦 Banking Example:**

REST API that returns **account balance** to a mobile banking app.

**🛒 E-commerce Example:**

REST endpoint provides product catalog for front-end mobile/web clients.

**🔹 9. JTA (Java Transaction API)**

**📌 Purpose:**

Provides **distributed transaction management** across multiple databases or systems.

**🛠 Features:**

* Commit/rollback
* XA transactions (multi-resource)
* Declarative transaction support

**🏦 Banking Example:**

During fund transfer, if debit from one account and credit to another fail midway, JTA rolls back the whole transaction.

**🔹 10. Bean Validation (JSR 380)**

**📌 Purpose:**

Validates JavaBeans using annotations.

**🛠 Features:**

* @NotNull, @Size, @Email, etc.
* Integrates with JPA and JSF

**🛒 E-commerce Example:**

Ensures customer enters a valid email and mobile number during checkout.

**✅ Summary Table of JEE Libraries with Real-Time Use Cases**

| **JEE Library** | **Purpose** | **Banking Example** | **E-commerce Example** |
| --- | --- | --- | --- |
| Servlet | Request Handling | Login Authentication | Order Form Submission |
| JSP | Dynamic Web Pages | Account Summary Page | Product Listing Page |
| JSF | UI Components | Loan Application UI | Shopping Cart UI |
| EJB | Business Logic | Fund Transfer, Interest | Promo Code Calculation |
| JPA | ORM Mapping | Customer-Account Tables | Order-Product Mapping |
| JDBC | DB Access via SQL | Stored Procedures | Fetch Cart Items |
| JMS | Messaging System | Batch Transaction Updates | Order Status Notifications |
| JAX-RS / JAX-WS | Web Services | REST API for Balance | REST API for Product Info |
| JTA | Transaction Management | Atomic Fund Transfer | Inventory + Payment Sync |
| Bean Validation | Data Validation | Validate PAN/Phone | Validate Email, Zipcode |