



# Enterprise JavaBeans

EJB v1.0

©2006 WolfWare, Inc.

1

## What is an EJB?

Written in the Java programming language, an *enterprise bean* is a server-side component that encapsulates the business logic of an application.

- Sun Java EE 5 Tutorial

Note the word 'enterprise'.

EJB v1.0

©2006 WolfWare, Inc.

2

Website: <http://XMaLpha.com>

E-Mail: [Courses@XMaLpha.com](mailto:Courses@XMaLpha.com)

All original material copyright © 2007 Wolfware, Inc. All rights reserved.



## Why Use EJBs?

- Simplify large, distributed application development
  - Focus on business logic, not low-level services
- Separate presentation from business logic
- Potential for reuse of EJBs across applications

EJB v1.0

©2006 WolfWare, Inc.

3

## Applications for EJBs

- Situation requires scalability
  - Distribution across platforms
- Situation requires transactions on data
- Situation requires various clients
- Multiple applications using common components

EJB v1.0

©2006 WolfWare, Inc.

4

Website: <http://XMaLpha.com>

E-Mail: [Courses@XMaLpha.com](mailto:Courses@XMaLpha.com)

All original material copyright © 2007 Wolfware, Inc. All rights reserved.



## Pre-3.0 vs 3.0

- 3.0 takes advantage of Java 5 features
  - Persistence
  - Annotations
- Pre-3.0 EJBs regarded as too complex
  - Many 3<sup>rd</sup> party tools to simplify development
- 3.0 added considerable services, especially to support web services

EJB v1.0

©2006 WolfWare, Inc.

5

## EJB Types

- Session
  - Stateful
  - Stateless
- Message-driven
- Entity (pre-3.0) == Persistence (3.0)

EJB v1.0

©2006 WolfWare, Inc.

6



## Session Beans

- Represents a single client session
- Terminates when the session terminates
- Stateful – holds client state for duration of session
- Stateless – does not hold client state across invocations
- Use interfaces
- Synchronous

EJB v1.0

©2006 WolfWare, Inc.

7

## Session Bean Types

- Stateful
  - Use @Stateful annotation
  - Associated with an individual client session
  - Bean maintains state throughout session
- Stateless
  - Use @Stateless annotation
  - Does not maintain state

EJB v1.0

©2006 WolfWare, Inc.

8

Website: <http://XMaLpha.com>

E-Mail: [Courses@XMaLpha.com](mailto:Courses@XMaLpha.com)

All original material copyright © 2007 Wolfware, Inc. All rights reserved.



## Uses for Session Beans

- Stateless session beans can implement web services
- Bean instances do not need to be shared by multiple clients
- Bean state does not need to be stored (is not persistent)
- If bean doesn't need any specific client data, use stateless bean

EJB v1.0

©2006 WolfWare, Inc.

9

## Session Bean Interfaces

- Clients access session beans via interfaces
- Session Beans require:
  - Interface class
    - Annotated for type of client:
      - Local, Remote, or Web Service
  - EJB class that implements the interface
  - Client to access bean

EJB v1.0

©2006 WolfWare, Inc.

10

Website: <http://XMaLpha.com>

E-Mail: [Courses@XMaLpha.com](mailto:Courses@XMaLpha.com)

All original material copyright © 2007 Wolfware, Inc. All rights reserved.



## Remote Interface

### Remote Client:

- May run on different hardware (and different JVM) than the EJB
- Can be a web component, application client, or another EJB
- Location of EJB is transparent to Client
- Use @Remote annotation

EJB v1.0

©2006 WolfWare, Inc.

11

## Local Interface

### Local Client:

- Must be on same JVM
- Can be web component or another EJB
- Location is not transparent
- Use @Local annotation

EJB v1.0

©2006 WolfWare, Inc.

12



## Web Services Interface

### Web Service Client:

- Can invoke methods of a stateless session bean
  - Methods annotated with @WebMethod
- Can access EJB through JAX-WS
- Use @WebService annotation

EJB v1.0

©2006 WolfWare, Inc.

13

## Session Bean Steps

- Develop the Bean
  - Bean class (contains the business logic)
  - Interface (defines the bean's API)
- Deploy the Bean
- Develop the client that calls the bean

EJB v1.0

©2006 WolfWare, Inc.

14



## Message-driven Beans

- Analogous to Java GUI event listeners
- Receives JMS messages
- Can receive messages from a variety of sources
  - Other EJBs
  - Application Clients
  - Web Components
  - Other JMS application

EJB v1.0

©2006 WolfWare, Inc.

15

## Message-driven Beans

- Similar to session beans, except the way they are accessed (again, analogous to event handling)
- Asynchronous
- Can access data in a data repository (similar to entity beans)
- Use JMS, not interfaces

EJB v1.0

©2006 WolfWare, Inc.

16



## Uses for Message Beans

- Could use Session beans instead
- Asynchronous feature may relieve load on the server
- Different design paradigm

EJB v1.0

©2006 WolfWare, Inc.

17

## Bean Use Examples

- Converting
  - Currency, temperature, etc.
- Customize text with received data
  - Letters, calculations, etc.
- Shopping Cart

EJB v1.0

©2006 WolfWare, Inc.

18

Website: <http://XMALpha.com>

E-Mail: [Courses@XMALpha.com](mailto:Courses@XMALpha.com)

All original material copyright © 2007 Wolfware, Inc. All rights reserved.



## Entity Beans

- Sun no longer recognizes these as a type of EJB
- This feature has changed dramatically from prior versions of EJB
- Sun now refers to Entity Beans as part of Persistence (javax.persistence)
- Concept is still to represent persistent data

EJB v1.0

©2006 WolfWare, Inc.

19

## Uses for Entity Beans

- Inventory systems
- Membership systems
- Order systems
- Status systems
- Basically anything tied to data, although you can also use persistence from web applications

EJB v1.0

©2006 WolfWare, Inc.

20

Website: <http://XMaLpha.com>

E-Mail: [Courses@XMaLpha.com](mailto:Courses@XMaLpha.com)

All original material copyright © 2007 Wolfware, Inc. All rights reserved.



## Entities

- Map to data in a data repository
- Entity class typically maps to a table
- Entity instance typically maps to a row in the table
- Table columns map to instance variables or to properties

EJB v1.0

©2006 WolfWare, Inc.

21

## Entity Classes

- Use the @Entity Annotation
- Must have a no-argument constructor
- Cannot be final
- Can be part of an inheritance hierarchy
- Instance variables mapped to persistent data cannot be public

EJB v1.0

©2006 WolfWare, Inc.

22



## Persistent Fields

- Primitives
- String
- Serializable data types
- Enumerated data types
- Entities
- Collections of Entities

EJB v1.0

©2006 WolfWare, Inc.

23

## Persistent Properties

- Must follow JavaBean conventions
- Single-value requires getter and setter
- Multiple-value can use
  - Collections (Collection, Set, List, Map)
  - Generics

EJB v1.0

©2006 WolfWare, Inc.

24

Website: <http://XMaLpha.com>

E-Mail: [Courses@XMaLpha.com](mailto:Courses@XMaLpha.com)

All original material copyright © 2007 Wolfware, Inc. All rights reserved.



## Primary Keys

- Each Entity must have a unique identifier  
== Primary Key
- Can be simple or composite
- Simple use the @Id annotation
- Composite use the @EmbeddedId and IdClass annotations
- Use @GeneratedValue if key is to be generated

EJB v1.0

©2006 WolfWare, Inc.

25

## EntityManager

- Entities are managed by the EntityManager
- Responsible for creating and deleting entities
- Responsible for locating entities (via primary key)
- Manages querying of entities

EJB v1.0

©2006 WolfWare, Inc.

26

Website: <http://XMALpha.com>

E-Mail: [Courses@XMALpha.com](mailto:Courses@XMALpha.com)

All original material copyright © 2007 Wolfware, Inc. All rights reserved.



## Getting an EntityManager

```
import javax.persistence.*;
...
@PersistentContext
EntityManager em;
```

EJB v1.0

©2006 WolfWare, Inc.

27

## Locating Entities

- Use the find() method:  
em.find(ClassName.class, key)
- Example:

```
public Program getProgram(int programId) {
    return em.find(Program.class, programId);
}
```

EJB v1.0

©2006 WolfWare, Inc.

28



## Persisting Entities

- Tying a java object to a database item
- Typically done when creating the object
- Use the persist() method:  
em.persist(program)

Example:

```
public Program createProgram() {  
    Program program = new Program();  
    em.persist(program);  
    return program;  
}
```

EJB v1.0

©2006 WolfWare, Inc.

29

## Entity Lifecycle

Entities have states:

- New – no identity or persistent context yet
- Managed – have an identity and context
- Detached – have an identity but not a context currently
- Removed – have an identity and context, and are slated to be removed from the data repository

EJB v1.0

©2006 WolfWare, Inc.

30



## Querying Entities

- The Persistence API includes a query language
- Somewhat like SQL, but the syntax is different
- Queries are portable, do not rely on a particular data repository

EJB v1.0

©2006 WolfWare, Inc.

31

## SELECT Statement

```
QL_statement ::= select_clause
                from_clause
                [where_clause]
                [groupby_clause]
                [having_clause]
                [orderby_clause]
```

EJB v1.0

©2006 WolfWare, Inc.

32



## SELECT Statement

```
SELECT m  
FROM Member m
```

```
SELECT m  
FROM Member m  
WHERE m.id == :123
```

EJB v1.0

©2006 WolfWare, Inc.

33

## UPDATE and DELETE

```
update_statement :: = update_clause  
                    [where_clause]
```

```
delete_statement :: = delete_clause  
                    [where_clause]
```

EJB v1.0

©2006 WolfWare, Inc.

34