

developer.com

GAMELAN™

Search

EARTHWEB®

CodeGuru | Gamelan | Jars | Wireless | Discussions

Navigate developer.com

Architecture & Design

Database

Java ▶

Languages & Tools ▶

Microsoft & .NET ▶

Open Source

Project Management

Security

Techniques

Voice

Web Services

Wireless/Mobile ▶

XML

Technology Jobs

Developer.com
Webcasts:

[Whitepaper: Enabling Technology for Blade I/O Virtualization.](#) In this technology brief, IDC examines HP Virtual Connect & the benefits & challenges with using this technology to virtualize I/O with HP BladeSystems.

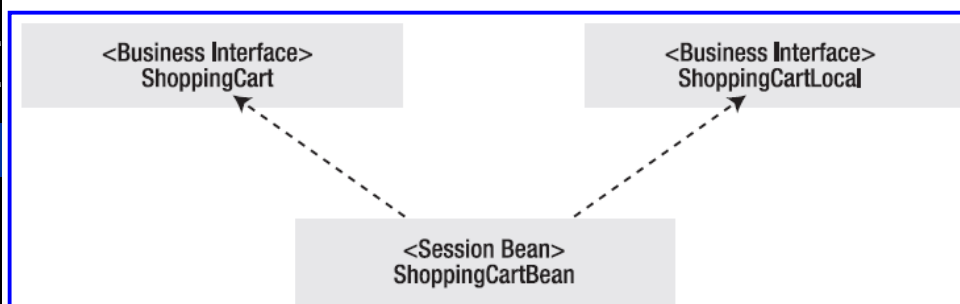
EJB 3 Session Beans

By [Raghu R. Kodali](#), [Jonathan R. Wetherbee](#), and [Peter Zadrozny](#)

Go to page: [Prev](#) [1](#) [2](#) [3](#) [4](#) [5](#) [Next](#)

The Business Interface

Business interfaces for stateful session beans are similar to those for stateless session beans, and are annotated in the same way, using `@Local` and `@Remote` annotations. The `ShoppingCart` session bean has both remote and local interfaces, as shown in Figure 6.



Power Consumption and Cooling Resource Center

Webcast: Blades and Virtualization: A Perfect Match?

July 12, 2007 (2pm EDT, 11am PDT)

Blades have been promoted as the path to IT nirvana in their own right. However, blades by themselves only simplify the physical infrastructure. The real win comes from combining blades with a variety of software technologies to simplify management and improve operational efficiency. Virtualization technology helps deliver on blade promises such as manageability, availability and flexibility. >

On Demand Webcast: Chilling Changes in the Server Room

The more your organization depends upon the network and your servers, the more heat they produce and the more power they consume. Learn about technological and data center changes that are driving your cooling bills through the roof, forcing you to drag in even more power to your server room that's already resonating at 60 Hz. >

Whitepaper: Enabling Technology for Blade I/O Virtualization

HP's BladeSystem c-Class portfolio was designed to address some of the key total cost of ownership issues facing today's datacenter, including server management costs, utilization, and power and cooling. In this technology brief, IDC examines HP Virtual Connect and the benefits and challenges associated with using this technology to virtualize I/O with HP BladeSystems. >

Whitepaper: Enabling Technologies for Blade Management

The Impact of Coding Standards and Code Reviews

Project Management for the Developer

Defining Your Own Software Development Methodology

more Webcasts...



Developer Jobs ▶

internet.commerce

[Be a Commerce Partner](#)
[Mortgage Refinance](#)
[Marketing Products](#)
[Buy Text Link Ads](#)
[Baby Photo Contest](#)
[Online Meetings](#)
[Cheap Plasma TVs](#)
[Web Hosting](#)
[Web Site Design](#)
[Merchant Accounts](#)
[Franchise Directory](#)
[Build a Server Rack](#)
[Promotional Gifts](#)
[Memory](#)
[Server Racks](#)

[Click here for a larger image.](#)

Figure 6. Business interfaces for ShoppingCart

You will primarily use the local interface from our web application. The remote interface is added to facilitate unit testing of the bean in this article.

Listings 10 and 11 show the remote and local `shoppingCart` business interfaces, with `@Remote` and `@Local` annotations, respectively.

Listing 10. ShoppingCart.java

```
package com.apress.ejb3.chapter02;

import javax.ejb.Remote;

@Remote
public interface ShoppingCart {
}
```

Listing 11. ShoppingCartLocal.java

```
package com.apress.ejb3.chapter02;

import javax.ejb.Local;

@Local
public interface ShoppingCartLocal {
}
```

Alternatively, you can use the coding style shown in Listing 2-12, in which you can specify the `@Local` and `@Remote` annotations before specifying `@Stateful` or `@Stateless` with the name of the business interface.

Listing 12. ShoppingCartBean.java

```
package com.apress.ejb3.chapter02;

import javax.ejb.Local;
import javax.ejb.Remote;
import javax.ejb.Stateful;

@Local({ShoppingCartLocal.class})
@Remote({ShoppingCart.class})
@Stateful(name="ShoppingCart")

public class ShoppingCartBean implements ShoppingCart,
    ShoppingCartLocal {
    public ShoppingCartBean() {
    }
}
```

HP's Insight Control Management plays a central role in reducing overall datacenter operating expenses and helps differentiate the BladeSystem c-Class system, both from competitive blade offerings and from rack-optimized servers. In this technical brief, IDC examines HP Insight Control and the importance of manageability in the selection of a blade platform. >

Whitepaper: Enabling Technologies for Power and Cooling

In recent years, the rate of server technology advancement has outpaced the datacenters ability to support these systems, especially in terms of power and cooling. In this technology brief, IDC focuses on HP Thermal Logic, a complete system solution to current power and cooling challenges. >



Developer News

- [Adobe Gives ColdFusion A Big Overhaul](#) May 30, 2007
- [Complex Events? BEA Has 'Em Covered](#) May 28, 2007
- [For The Love of Multi-Core Pain](#) May 28, 2007
- [Standards Group OKs E-Mail Validation Spec](#) May 25, 2007

Free Tech Newsletter



}

Note: In this article, you will follow the earlier convention, in which `@Local` and `@Remote` annotations are marked on the business interfaces.

Business Methods

Business methods in stateful session beans are similar to those in stateless session beans. You will augment the `ShoppingCart` bean by adding business methods that will add and remove wines from the shopping cart, and return a list of cart items.

Listing 13 shows the `ShoppingCart` bean implementing the `addWineItem()`, `removeWineItem()`, and `getCartItems()` methods.

Listing 13. ShoppingCartBean.java

```
package com.apress.ejb3.chapter02;
import java.util.ArrayList;

import javax.ejb.Stateful;

@Stateful(name="ShoppingCart")

public class ShoppingCartBean implements ShoppingCart,
    ShoppingCartLocal {
    public ShoppingCartBean() {
    }
    public ArrayList cartItems;

    public void addWineItem(String wine) {
        cartItems.add(wine);
    }

    public void removeWineItem(String wine) {
        cartItems.remove(wine);
    }

    public void setCartItems(ArrayList cartItems) {
        this.cartItems = cartItems;
    }

    public ArrayList getCartItems() {
        return cartItems;
    }
}
```

Callback Methods

Stateful session beans support callback events for construction, destruction, activation, and passivation. Following are the callbacks that map to the preceding events:

- **PostConstruct:** Denoted with the `@PostConstruct` annotation. Any method in the bean class can be marked with this annotation.

- **PreDestroy**: Denoted with the `@PreDestroy` annotation.
- **PreActivate**: Denoted with the `@PreActivate` annotation.
- **PrePassivate**: Denoted with the `@PrePassivate` annotation.

The `PostConstruct` callback happens after a bean instance is instantiated in the EJB container. If the bean is using any dependency injection mechanism for acquiring references to resources or other objects in its environment, the `PostConstruct` event happens after injection is performed and before the first business method in the bean class is called.

In the case of the `ShoppingCart` session bean, you could have a business method called `initialize()` that initializes the `cartItems` list, as show in Listing 14.

Listing 14. The PostConstruct Method

```
@PostConstruct
public void initialize()
{
    cartItems = new ArrayList();
}
```

The `PreDestroy` callback happens after any method with an `@Remove` annotation has been completed. In the case of the `ShoppingCart` session bean, you could have a business method called `exit()` that writes the `cartItems` list into a database. In this article, you will just print out a message to the system console to illustrate the callback. Listing 15 shows the code for the `exit()` method, which has the `@PreDestroy` annotation.

Listing 15. The PreDestroy Method

```
@PreDestroy
public void exit()
{
    // items list into the database.
    System.out.println("Saved items list into database");
}
```

The `@Remove` annotation is a useful life cycle method for stateful session beans. When the method with the `@Remove` annotation is called, the container will remove the bean instance from the object pool after the method is executed. Listing 16 shows the code for the `stopSession()` method, which has the `@Remove` annotation.

Listing 16. The Remove Method

```
@Remove
public void stopSession()
{
    // The method body can be empty.
    System.out.println("From stopSession method with @Remove
        annotation");
}
```

}

The `PrePassivate` callback kicks in when a stateful session bean instance is idle for too long. During this event, the container might passivate and store its state to a cache. The method tagged with `@PrePassivate` is called before the container passivates the bean instance.

The `PostActivate` event gets raised when the client application uses a passivated stateful session bean again. A new instance with restored state is created. The method with the `@PostActivate` annotation is called when the bean instance is ready.

Interceptors

There are some minor differences between interceptors for stateless and stateful session beans. `AroundInvoke` methods can be used with stateful session beans. For stateful session beans that implement `SessionSynchronization`, `afterBegin` occurs before any methods that have `AroundInvoke` annotations, and before the `beforeCompletion()` callback method.

Go to page: [Prev](#) [1](#) [2](#) [3](#) [4](#) [5](#) [Next](#)




Tools:  [Email](#)  [Print](#)  [Digg This Story](#)

 [del.icio.us](#)

Add [www.developer.com](#) to your favorites

 [MY YAHOO!](#)  [Windows Favorites](#)

Add [www.developer.com](#) to your browser search box

 [IE 7](#) |  [Firefox 2.0](#) |  [Firefox 1.5.x](#)

Receive news via our XML/RSS [feed](#) [XML](#) [RSS](#)

[EJB/Components Archives](#)

DEVELOPER SOLUTIONS

- ▶ Developing Intelligent Communications? Visit the Avaya Developer Connection Center on DevX.
- ▶ **Windows Server "Longhorn" Showcase Roadshow: Developer Essentials.**
- ▶ Generate Complete .NET Web Apps in Minutes . Download Iron Speed Designer today.
- ▶ Is it time to make your move to the multi-threaded and parallel processing world? Find out!
- ▶ New Article! Take a Sneak Peek--Microsoft .NET Developers Test Drive Avaya's Latest SDK

Jupiter Online Media:



Search:

[Jupitermedia Corporation](#) has two divisions: [Jupiterimages](#) and [Jupiter Online Media](#)

[Jupitermedia Corporate Info](#)

Copyright 2007 Jupitermedia Corporation All Rights Reserved.
[Legal Notices](#), [Licensing](#), [Reprints](#), & [Permissions](#), [Privacy Policy](#).

[Web Hosting](#) | [Newsletters](#) | [Tech Jobs](#) | [Shopping](#) | [E-mail Offers](#)