



JBoss

JBoss For Java Developers with EJB3

QUALITY TRAINING
FROM THE SOURCE.

JBoss for Java Developers with EJB3 is targeted toward proficient Java developers who wish to extend their knowledge to J2EE middleware and who wish to get started with the new EJB3 component model to develop Java Middleware applications.

The class is an in-depth introduction to J2EE and EJB3 using the JBoss Application Server. It provides a hands-on approach to the development and deployment of multi-tier J2EE applications that will use EJB3.

Course Duration: This is a 4 day course.

Course Format: The format is 50% theory, 50% hands-on lab exercises.



Prerequisites and Course Modules

Prerequisites

Basic Java programming skills and knowledge of OOAD concepts are required. The student must have practical knowledge of, and/or experience with, the following:

- The object-oriented concepts of inheritance, polymorphism and encapsulation
- Java syntax, specifically for data types, variables, operators, statements and control flow
- Writing Java classes as well as using Java interfaces and abstract classes
- Using Java collections
- Handling Java exceptions
- Using the JDK and creating the necessary environment for compilation and execution of a Java executable from the command line.

No prior knowledge of J2EE or the JBoss Application Server is required. This training is based on the JBoss 4.x application server, J2EE 1.4 and EJB3.

Course Modules

1. **Introduction to J2EE and EJB3.** This module provides an overview of the J2EE platform architecture and EJB3 component model. It clearly defines J2EE, the motivation behind the design of the platform, the EJB3 components that comprise the J2EE 1.4 platform and provides an overview to the packaging and description of roles in J2EE development and management.

2. **Servlets.** This module provides the background and motivation that led to the creation of the Java Servlet specification as well as an overview of the servlet architecture. It covers basic HTTP request and parameter handling, HTTP Sessions, filters and web application lifecycle events. It also covers the Web Archive (WAR) packaging, how to use JBoss specific deployment descriptors for servlet configuration, and how to configure Tomcat connectors.

3. **Java Server Pages.** This presentation gives the background and motivation of the Java Server Pages specification and gives a foundation for JSP implementation. It covers JSP tags and directives, introduces MVC architecture concepts with JSP and EJB3 and how to make modifications to applications currently deployed with JSP pages with JBoss expanded directory deployments.

4. **Naming Services and JNDI.** This section describes the J2EE Naming Service, its role and how it is used. It explains the basics of the Java Naming and Directory Interface (JNDI) API and the configuration of the Naming Service. It also provides foundational details of JBossNS, the naming service implementation used by the JBoss application server.

5. **Introduction to EJB3.** This module provides a basic introduction to Enterprise Java Beans 3.0. It explains the case for server-side components, the declarative (via annotations and/or configuration files) programming approach of EJB components. It shows how EJB insulates “business code”

from system code. This section introduces the different kinds of EJB components: session, entity and message-driven beans and presents how to author with version 3.0 of this component specification. The student will learn how to specify a local and/or remote POJI (plain old Java interface) for the simple POJO-based (plain old Java object) component. The module will show how to author EJB3 life-cycle listeners and listener classes and exception classes that may be specified to determine the completion of a transaction. This section will introduce the student to the completely new persistence model for entity beans and their embeddable classes. The module will conclude with an explanation of several architectural innovations found in the JBoss Application Server EJB3 container such as smart proxy technology. Smart Proxy technology adds “ease-of-use” to EJB3 deployment and backward compatibility to EJB2.x. It is integral to the highly configurable, “interceptor-based” EJB3 container.

6. **EJB3: Stateless Session Beans.** This module covers the stateless session bean in detail. EJB3 bean implementation, packaging and deployment will be addressed. Bean lifecycle and lifecycle listener classes, pooling and concurrency issues are covered as well as how to configure session bean pool sizes and author custom interceptor stacks. The student will learn how to create custom container configurations in JBoss and how to use the interface class name or a global JNDI mapping for bean location.



Course Modules

7. **EJB3: Stateful Session Beans.** This module covers the stateful session bean in detail. Lifecycle listener classes, container annotations and interceptor classes for EJB3 stateful session beans are thoroughly examined. This module will provide a comparison between stateless and stateful session beans and why one would be chosen over the other. The student will learn how to configure a Stateful Session Bean cache and how to configure Stateless and Stateful Session Beans for clustering in JBoss.
8. **EJB3: Entity Beans.** This section covers the new persistence model for EJB3 (and J2SE) POJO components in detail. This includes addressing the interfaces, lifecycle and lifecycle listener classes, interceptor classes, simple and composite ids and queries in EJB3. Issues of concurrency, pooling and packaging and deployment are covered. The student will learn how to construct embeddable and composite id classes. The student will be introduced to the use of "dependency injection." The student will learn how to configure different commit options, mapping and fetching strategies, and how to manage basic datasource configuration for entity objects in EJB3.
9. **Transaction Assembly.** This module introduces the concept of transactional boundaries and how the transactional declarative tags (as annotations or tags in the configuration file) allow developers to control atomic units of work. A foundational overview of the two phase commit protocol and the handling of exceptions and rollbacks from within an EJB3 POJO is provided.
10. **J2EE Security and JBoss.** This section is an overview to the J2EE security topics. The student will be shown how to add access control to web applications, how to authenticate users and how to control access to EJB components. This module also shows how to setup and configure the numerous security domains, or login modules for a JBoss instance.
11. **EJB3: Message Driven Beans (MDB) and the Java Message Service (JMS)**
This module provides a comprehensive introduction to the Java Messaging Service and the JMS API. Point-to-point and publish-subscribe messaging domains will be covered as well as transactional JMS. The student will be shown how to work with an integrated JMS within the EJB3 component framework. An in-depth look in to Message Driven Beans (MDB) (for asynchronous communication with the EJB3 layer) will be provided. The student will learn how to create message driven beans within a transactional setting and how to deploy JMS connection factories, JMS destinations and message driven beans.
12. **EJB3 Interceptors.** This section describes the interceptors formalized in the EJB3 specification. Interceptors are stateless Java classes that work in the EJB3 container in a manner similar to the way Servlet 2.3 filters work in the Web container. EJB3 interceptors intercept the incoming business method invocations and the outgoing return message. This module describes how to author both global and class-specific interceptors and how the invocation state and an arbitrary payload can be shared among multiple interceptors via the InvocationContext object.
13. **Web Services with EJB3.** This section will provide an overview of the architecture and implementation of J2EE 1.4 web services. The student will learn how to expose EJB3 and Servlet components via a Web Services Definition Language (WSDL) document to enable remote XML RPC invocations to these components. The student will learn the basic approach of integrating the JBoss application server with legacy enterprise middleware and .NET.





Course Modules



14. **JMX and JBoss Microkernel.** This module provides a general overview to the Java Management Extensions (JMX) specification and to the JBoss Microkernel architecture. The student will learn how to create system level components that can be monitored and managed via any management client using the JMX API. The student will become capable of understanding the underlying mechanisms of the JBoss Application Server and will understand how to customize or augment the application server behaviors. The student will learn how to package and deploy a JMX component via JBoss Service Archives (SAR) or simple XML configuration files.

15. **Extensions to EJB3 – A preview of EJB3.1 (OPTIONAL).** JBoss, Inc is renowned as a leader in innovation. The JBoss 4.0.3 application server serves as the platform to preview the next version of EJB3, EJB3.1. The student will be shown the natural progression from EJB3 to enterprise POJOs. The Service Bean,

the next generation of JMX, and Message-Driven POJOs, the next generation of typed MDBs will be presented. As the final part of this module, the student will be introduced to how an EJB3, which by default is communicated with synchronously, can be communicated with asynchronously via the introduction of an asynchronous proxy for any EJB3 implementation.

16. **J2EE Tools (OPTIONAL).** This module is presents ideas on how to augment and facilitate J2EE and POJO development using the popular Open Source tool ANT. After mastering the basics of J2EE component development and deployment on the JBoss Application Server, the student will learn how to take advantage of this powerful tool to simplify, facilitate and automate development and deployment of J2EE applications on the JBoss application server.



Getting Started

Contact us to learn more about JBoss Inc. support services. Please visit us at www.jboss.com, e-mail us at sales@jboss.com, or contact one of our sales offices:

US Headquarters:

JBoss Inc.
3340 Peachtree Road, NE
Suite 1200
Atlanta, GA USA 30326
Phone (404) 467-8555
Fax (404) 948-1496

European Headquarters:

JBoss Europe
Place Numa-Droz 2,
CP 2522
CH - 2001 Neuchâtel
Switzerland
Phone +41 32 720 92 60
Fax +41 32 720 92 61

European Sales Office:

JBoss UK
Regents Place
338 Euston Road
London NW1 3BT
United Kingdom
Phone: +44 20 7543 6910
Fax: +44 20 7544 8401

