

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

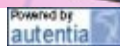
Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)



Hosting patrocinado por



[Inicio](#)

[Quienes somos](#)

[Tutoriales](#)

[Formación](#)

[Colabora](#)

[Comunidad](#)

[Comic](#)

[Charlas](#)

[Más](#)

NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha aquí...

[Ver cursos que ofrece Autentia](#)

[Descargar comics en PDF y alta resolución](#)

Catálogo de servicios Autentia (PDF 6,2MB)



[comic...](#)



[¡NUEVO!] 2008-12-01

2008-11-17

2008-09-01

¿Por qué no se aplican los mismos criterios de la vida a la informática?



google™

Web
2008-07-31
www.adictosaltrabajo.com

Buscar

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la sección [comic](#).

Tutorial desarrollado por

Catálogo de servicios de Autentia



Carlos Pérez **García**

Creador de **MobileTest**, un complemento educativo para los profesores y sus alumnos.

Consultor tecnológico en el desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática *

[Descargar \(6,2 MB\)](#)

[Descargar en versión comic \(17 MB\)](#)

[AdictosAlTrabajo.com](#) es el Web de difusión de conocimiento de [Autentia](#).



[Catálogo de cursos](#)

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/J2EE

Descargar este documento en formato PDF: [embeddedJBoss3_Java6.pdf](#)

Fecha de creación del tutorial: 2009-02-09

EJB 3.0 y pruebas unitarias con Maven, JUnit 4 y Embedded JBoss sobre Java 6

1. [Introducción](#)
2. [Emdebbed JBoss \(Beta 3\)](#)
3. [Ejemplo](#)
 1. Lógica de Negocio
 1. [Interfaz EJB Local.](#)
 2. [Interfaz EJB Remota.](#)
 3. [EJB de ejemplo.](#)
 2. [Invocando la interfaz local y remota.](#)
 3. [Configuración de Maven: pom.xml](#)
4. [Conclusiones](#)

Introducción

Este tutorial es un complemento a otro tutorial (<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ejb3JUnitWithJBoss>) publicado anteriormente, y en el cual se documenta de forma más extensa los conceptos expuestos aquí.

El problema es que el anterior tutorial fallaría si lo intentáramos ejecutar bajo la versión 6 de la máquina virtual de Java (Java 6), en este tutorial se aporta el conocimiento necesario para solventar este problema.

Además, como diferencia en vez de usar la versión JBoss Embedded Beta 2, en este tutorial usaremos la JBoss Embedded Beta 3 (Kernel de JBoss 5).

Emdebbed JBoss (Beta 3)

Últimos tutoriales

2009-02-09
[EJB 3.0 y pruebas unitarias con Maven, JUnit 4 y Embedded JBoss sobre Java 6](#)

2009-02-09
[Web Service Security](#)

2009-02-09
[Manual Avanzado de Firebug](#)

2009-01-29
[Ejemplo con Mockito](#)

2009-01-29
[Uso de Mock objects en pruebas con Mockito](#)

2009-01-29
[StrutsTestCase](#)

2009-01-28
[Eventos en Hibernate \(parte III\)](#)

2009-01-28
[Eventos en Hibernate \(parte II\)](#)

2009-01-27
[Eventos en Hibernate \(parte I\)](#)

2009-01-25
[Aprendiendo XMLSchema a través de ejemplos](#)

Últimas ofertas de empleo

Emdebbbed JBoss es un versión limitada de el servidor de aplicaciones JBoss, pensada para ser embebida en aplicaciones de escritorio, tests, Tomcat, etc..

El peso que tiene esta versión es muy inferior al que tiene la versión no embebida del servidor de aplicaciones JBoss y entre otras cosas es ideal para ejecutar nuestros tests de forma más eficiente.

Es decir, esta versión está pensada para ser capaz de ejecutar el Kernel de JBoss 5 y los servicios JEMS sin usar el classloader de la versión no embebida de JBoss.

- Puede ver un listado de sus características y limitaciones [haciendo clic aquí](#).
- Puedes descargarlo [haciendo clic aquí](#).

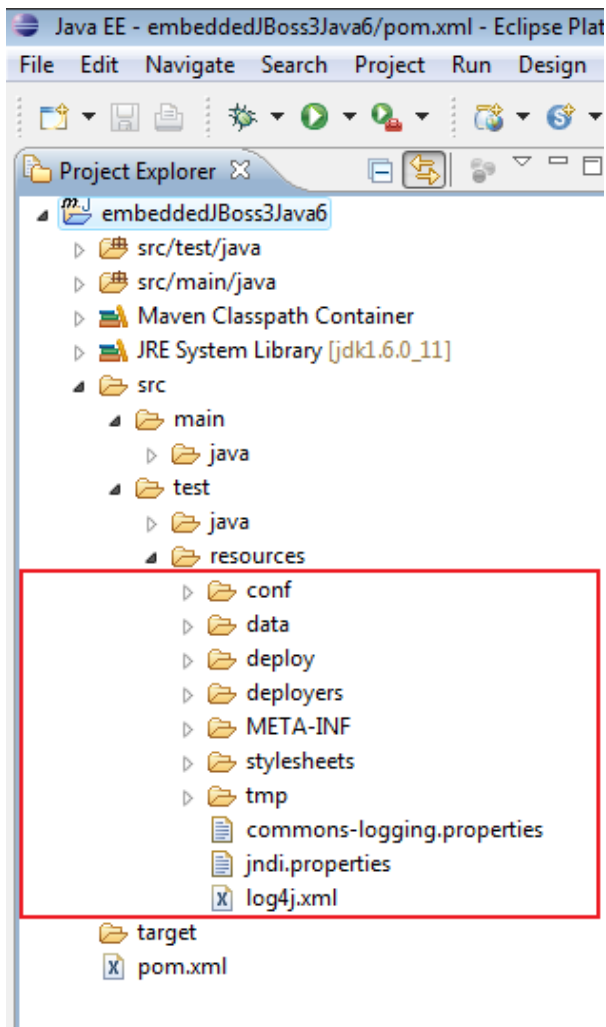
Ejemplo

A continuación vamos a ver un ejemplo completo en el cual crearemos un [EJB sin estado](#) que expone funcionalidad que puede ser invocada de forma tanto [local](#) como [remota](#). Posteriormente [creamos un test JUnit](#) que despliega el EJB en el microcontenedor JBoss Emdebbbed 3 e invoca el método que expone.

A continuación vamos a ver un completo ejemplo en el cual:

1. Definiremos una interfaz que exponga la funcionalidad [local](#) del EJB.
2. Definiremos una interfaz que exponga la funcionalidad [remota](#) del EJB.
3. Crearemos un [EJB sin estado](#) que expone el servicio local y remoto definido anteriormente.
4. Creamos un par de [tests JUnit](#) que despliegue el EJB en el microcontenedor JBoss Emdebbbed 3 e invoque el método local y remoto.
5. Veremos que modificaciones tenemos que hacer en el archivo de [configuración de Maven](#) para ejecutar todo de manera automática.

El código fuente de este tutorial puede ser [descargado desde aquí](#).



Importante:

Hay que copiar los archivos de configuración de JBoss Embedded (%JBoss_EMDEBBED_DIR%/bootstrap) al directorio `src/test/resources` de nuestro proyecto Maven, para que esta configuración esté disponible (via classpath) para el servidor.

Si observa la imagen, verá que son los archivos y directorios resaltados por el cuadrado rojo.

Interfaz EJB Local:

2008-12-22
[Otras - Mecánica - SEVILLA.](#)

2008-11-27
[Comercial - Ventas - ALICANTE.](#)

2008-10-30
[Comercial - Ventas - BARCELONA.](#)

2008-10-30
[T. Información - Analista / Programador - BARCELONA.](#)

2008-10-27
[T. Información - Analista / Programador - CIUDAD REAL.](#)

Anuncios Google
[Project 2002](#)
[Juegos Java](#)
[Java Architecture](#)
[Database for Java](#)

```

view plain print ?
01. package com.autentia.embedded3java6;
02.
03. /**
04.  * Interfaz Local de un saludador.
05.  * Sólo se puede usar desde La JVM donde está ejecutandose el servidor de apl
06.  */
07. @javax.ejb.Local
08. public interface Greeter {
09.     public String sayHi();
10. }

```

Interfaz EJB Remota:

```

view plain print ?
01. package com.autentia.embedded3java6;
02.
03. /**
04.  * Interfaz remota de un saludador.
05.  * Para poder invocar el EJB desde fuera de La JVM del servidor de aplicacion
06.  */
07. @javax.ejb.Remote
08. public interface GreeterRemote extends Greeter {
09.     // Aunque no añade nada nuevo a la interfaz que extiende, es mejor separa
10.     // y así se evitarán problemas futuros en caso de ampliar funcionalidad.
11. }

```

EJB de ejemplo:

EJB de ejemplo: No tiene estado y puede ser invocado local o remotamente:

```

view plain print ?
01. package com.autentia.embedded3java6;
02.
03. /**
04.  * EJB sin estado.
05.  */
06. @javax.ejb.Stateless
07. public class DummyGreeterBean implements Greeter, GreeterRemote {
08.     public static final String DEFAULT_GREETING = "Autentia: ¡ Hola Mundo !";
09.
10.     public String sayHi() {
11.         return DummyGreeterBean.DEFAULT_GREETING;
12.     }
13. }

```

Test a realizar:

view plain print ?

```
01. package com.autentia.embedded3java6;
02.
03. import javax.naming.InitialContext;
04. import junit.framework.JUnit4TestAdapter;
05. import org.jboss.deployers.spi.DeploymentException;
06. import org.jboss.embedded.Bootstrap;
07. import org.jboss.virtual.plugins.context.vfs.AssembledContextFactory;
08. import org.jboss.virtual.plugins.context.vfs.AssembledDirectory;
09. import org.junit.Assert;
10.
11. /**
12.  * Tests que deseamos realizar de nuestra lógica de negocio
13.  * @author Carlos García. Autentia.
14.  */
15. public class EJBsTest {
16.
17.     /**
18.      * Simula un jar con todos los elementos que vamos a desplegar en el micr
19.      */
20.     private static AssembledDirectory jar;
21.
22.     /**
23.      * En caso de ejecutar los tests en un IDE.
24.      */
25.     public static void main(String[] args) {
26.         junit.textui.TestRunner.run(EJBsTest.suite());
27.     }
28.
29.     /**
30.      * Despliega todos los elementos que necesitamos para ejecutar los test.
31.      * Simula la construcción de un jar, y como si luego echáramos este jar a
32.      /deploy del JBoss.
33.      * @throws DeploymentException si no se puede hacer el despliegue.
34.      */
35.     private static void deploy() throws DeploymentException {
36.         jar = AssembledContextFactory.getInstance().create("ejbTestCase.jar");
37.         jar.addClass(Greeter.class);
38.         jar.addClass(GreeterRemote.class);
39.         jar.addClass(DummyGreeterBean.class);
40.         Bootstrap.getInstance().deploy(jar);
41.     }
42.
43.     /**
44.      * Elimina los elementos que se habían desplegado con deploy();
45.      * @throws DeploymentException si no se puede hacer la desinstalación.
46.      */
47.     private static void undeploy() throws DeploymentException {
48.         Bootstrap.getInstance().undeploy(jar);
49.         AssembledContextFactory.getInstance().remove(jar);
50.     }
51.
52.     /**
53.      * Este método determina los métodos que se van a ejecutar con el test.
54.      * @return EL conjunto de test que hay que ejecutar.
55.      */
56.     public static junit.framework.Test suite() {
57.         final junit.framework.Test suite = new JUnit4TestAdapter(EJBsTest.class);
58.         return suite;
59.     }
60.
61.     /**
62.      * La inicialización de Embedded JBoss tarda varios segundos así que nos
63.      * @throws DeploymentException si no se puede hacer el despliegue de los
64.      */
65.     @org.junit.BeforeClass
66.     public static void setUpBeforeAllTest() throws Exception {
67.         if (! Bootstrap.getInstance().isStarted()) {
68.             Bootstrap.getInstance().bootstrap();
69.         }
70.
71.         EJBsTest.deploy();
72.     }
73.
74.     /**
75.      * Parada del Embedded JBoss. Nos aseguramos que se haga una única vez de
76.      * @throws DeploymentException En caso de error en el proceso de desinsta
77.      */
78.     @org.junit.AfterClass
```

Archivo pom.xml:

Está autocomentado.

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.autentia.tutoriales.jbossembeddedjava6</groupId>
  <artifactId>embeddedJBoss3Java6</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>embeddedJBoss3Java6</name>
  <url>http://maven.apache.org</url>

  <!-- Damos de alta el repositorio de JBoss -->
  <repositories>
    <repository>
      <id>jboss-repository</id>
      <name>JBoss Repository</name>
      <url>http://repository.jboss.com/maven2</url>
      <layout>default</layout>
    </repository>
  </repositories>

  <build>
    <plugins>
      <!-- Modificamos el comportamiento de plugin que compila las clases para indicar
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <configuration>
          <source>1.6</source>
          <target>1.6</target>
        </configuration>
      </plugin>

      <!-- Modificamos el comportamiento de plugin que ejecuta los tests (Surefire) -->
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>2.4.2</version>

        <!-- La siguiente configuración es necesaria para que funcione JBoss Embedde
        <configuration>
          <!-- Debido a un bug de Java 6 (http://bugs.sun.com/bugdatabase
/view_bug.do?bug_id=6434149)
          debemos indicar a surefire que inicie la JVM con la que ejecuta los
          <argLine>-Dsun.lang.ClassLoader.allowArraySyntax=true</argLine>
        </configuration>
      </plugin>
    </plugins>
  </build>

  <dependencies>
    <!-- Dependencias para JBoss Embedded 3 -->
    <dependency>
      <groupId>org.jboss.embedded</groupId>
      <artifactId>jboss-embedded-all</artifactId>
      <version>beta3</version>
      <scope>test</scope>
    </dependency>

    <dependency>
      <groupId>org.jboss.embedded</groupId>
      <artifactId>jboss-embedded</artifactId>
      <version>beta3</version>
      <scope>test</scope>
    </dependency>

    <dependency>
      <groupId>org.jboss.embedded</groupId>
      <artifactId>thirdparty-all</artifactId>
      <version>beta3</version>
      <scope>test</scope>
    </dependency>

    <dependency>
      <groupId>org.jboss.embedded</groupId>
      <artifactId>hibernate-all</artifactId>
      <version>beta3</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>

```

```

<!--

```


Conclusiones

Por desgracia a veces cuando sale una nueva versión al mercado de algún elemento (en este caso Java 6) puede aparecer algún problema el cual requiera realizar acciones adicionales para su funcionamiento, en este caso modificar los archivos de configuración y parámetros de ejecución de la JVM.

Un saludo.

Carlos García. Creador de **MobileTest**, un complemento educativo para los profesores y sus alumnos.

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ivota!

Muy malo Malo Regular Bueno Muy bueno



Votar

- Puedes opinar sobre este tutorial [haciendo clic aquí](#).
- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo AdictosAlTrabajo en XING [haciendo clic aquí](#).
- Añadir a favoritos Technorati.



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido (**Ver todos los tutoriales**). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com



Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de subscripción a novedades:

E-mail

Aceptar

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
Uso de Mock objects en pruebas con Mockito	En este tutorial que nos presenta German vemos que es mockito y aprenderemos como usar su API.	2009-01-29	402	Muy bueno	1	
Anotaciones en EJB 3.0	Este tutorial nos va a enseñar algunas características del API de EJB 3.0 y las mejoras introducidas en la nueva version 3.0	2007-05-25	12781	Muy bueno	2	
Interceptando un EJB en JBoss	En este tutorial os vamos a enseñar la arquitectura de EJBs en JBoss y a como modificarla, insertando un interceptor propio dentro de la cadena de interceptores del Proxy Cliente.	2007-03-26	7017	Bueno	1	
EJB 3.0: Resurrection	Este tutorial nos va a presentar las nuevas funcionalidades que nos aportan los EJB 3.0.	2007-05-07	7113	Regular	5	
Hibernate y las anotaciones de EJB 3.0	En este tutorial Alejandro Pérez nos muestra las ventajas que nos aporta Hibernate y las anotaciones de EJB 3.0	2007-06-25	9053	Regular	2	
Despliegue gráfico de EJBs	Os mostramos como crear y desplegar de un modo gráfico un EJB de sesión en el servidor de aplicaciones de referencia de Sun	2003-09-21	23737	-	-	
EJB 3.0, un ejemplo práctico con Maven y JBoss	Este tutorial presenta un ejemplo sencillo donde se verá como desarrollar EJBs de sesión y de entidad, inyección de dependencias, llamar a los EJBs desde una aplicación Web, definición de un DataSource, y como configurarlo y hacerlo funcionar en JBoss, y	2007-08-06	7824	-	-	
EJB 3.0 y pruebas unitarias con Maven, JUnit y Embedded JBoss	En este tutorial Alejandro Pérez nos enseña como realizar test unitarios sobre EJB 3.0. Para ello se usará Maven, JUnit y Embedded JBoss	2007-08-09	5590	-	-	
Cómo realizar pruebas unitarias con Spring y JUnit4 utilizando Gienah	En este tutorial vamos a presentaros Gienah, una tecnología que os permitirá de una forma muy cómoda y sencilla utilizar componentes de Spring en vuestros test unitarios realizados con JUnit 4	2008-02-17	2104	-	-	

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

Anuncios Google

[Java Architecture](#)

[Database for Java](#)

[JSP Databases](#)

[Servlet Database](#)

[Eclipse J2EE IDE](#)

