

¿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)



Powered by **autentia** Hosting patrocinado por **enredados**

- [Inicio](#)
- [Quienes somos](#)
- [Tutoriales](#)
- [Formación](#)
- [Empleo](#)
- [Colabora](#)
- [Comunidad](#)
- [Libro de Visitas](#)
- [Comic](#)

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

[Ver cursos que ofrece Autentia](#)

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



[NUEVO!] 2008-03-11

2008-03-06

2008-03-02

2008-02-26

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

Tutorial desarrollado por



Raúl Expósito Díaz

Consultor de desarrollo tecnológico de proyectos informáticos.

Catálogo de servicios de Autentia

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

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

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



[Catálogo de cursos](#)

Ingeniero Técnico en Informática de Gestión por la Universidad de Alcalá e Ingeniero en Informática por la Universidad Carlos III de Madrid. [Perfil XING](#)

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/J2EE

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

Fecha de creación del tutorial: **2008-02-17**

Cómo realizar pruebas unitarias con Spring y JUnit4 utilizando Gienah

1. Introducción

Como venimos repitiendo desde [Autentia](#) constantemente, la realización de pruebas unitarias no debería ser opcional ya que, entre otras cosas, ayudan a pensar en qué debería hacer el código y qué no y nos da seguridad de que vamos a poder detectar errores ante cualquier cambio futuro en pocos segundos.

En [otros tutoriales](#) de [adictosaltrabajo](#) ya hemos hablado acerca de la importancia de realizar pruebas unitarias y también hemos dado diversas recetas para integrarlas en vuestros proyectos, así que no me voy a centrar en lo importante que es realizar este tipo de pruebas.

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](#).

Pero, ¿por qué utilizar [Gienah](#)?, ¿acaso no es posible realizar pruebas unitarias con [JUnit 4](#) y [Spring](#) directamente?. Por ser es posible, pero hay que montar cierta infraestructura para conseguirlo. Esta infraestructura ya está hecha en [Gienah](#) y además es muy sencilla de utilizar y muy potente, así que ¿para qué reinventar la rueda?

Catálogo de servicios Autentia (PDF 6,2MB)



En formato comic...



Web
 www.adictosaltrabajo.com

Últimos tutoriales

2008-03-10
[Visualización de código HTML como un grafo](#)

2008-03-04
[Introducción a JPivot](#)

2008-03-03
[Tablas dinámicas online](#)

2008-02-29
[Generación automática de gráficas en un web](#)

2008-02-28
[Manual de instalación de OpenCms 7](#)

2008-02-28
[Creación de un proyecto en SourceForge.net](#)

2008-02-22
[Lucene: Analyzers, stemming y búsqueda de documentos similares.](#)

2008-02-22
[Crear un logger utilizado a través de aspectos con Spring AOP.](#)

2008-02-20
[Primeros pasos con PostgreSQL en Debian](#)

2008-02-17
[Cómo realizar pruebas unitarias con Spring y JUnit4 utilizando Gienah](#)

Últimas ofertas de empleo

2008-02-28
[Otras - Arquitecto / Aparejador - MADRID.](#)


```

view plain print ?
01. import org.gienah.testing.junit.Configuration;
02. import org.gienah.testing.junit.SpringRunner;
03.
04. @RunWith(value = SpringRunner.class)
05. @Configuration(locations = {"resources/services.xml", "resources/persistence.xml"})
06. public class AutentiaTest
07.

```

Si solo necesitamos beans declarados en el fichero `services.xml`:

```

view plain print ?
01. @RunWith(value = SpringRunner.class)
02. @Configuration(locations = {"resources/services.xml"})
03. public class AutentiaTest
04.

```

Y, si el fichero estuviera en el classpath:

```

view plain print ?
01. @RunWith(value = SpringRunner.class)
02. @Configuration(locations = {"services.xml"})
03. public class AutentiaTest
04.

```

De momento ya tenemos acceso al contexto de Spring, con lo que podemos hacer uso de los beans declarados en él con sus dependencias ya inyectadas, pero ¿cómo recuperar los beans declarados?. Muy sencillo, sólo debemos hacer uso de la anotación `@Dependency`:

```

view plain print ?
01. @RunWith(value = SpringRunner.class)
02. @Configuration(locations = {"services.xml"})
03. public class AutentiaTest {
04.
05.     @Dependency
06.     private AutentiaBean autentiaBean;
07.

```

Una vez tengamos el bean podemos utilizar sus métodos declarados en la interfaz `AutentiaBean` y, de este modo, realizar los test unitarios para la clase que implementa la interfaz

```

view plain print ?
01. @RunWith(value = SpringRunner.class)
02. @Configuration(locations = {"services.xml"})
03. public class AutentiaTest {
04.
05.     @Dependency
06.     private AutentiaBean autentiaBean;
07.
08.     @Test
09.     public void testURLAutentia() {
10.         Assert.assertTrue(autentiaBean.getUrl().equals( "http://www.autentia.com/" ));
11.     }
12.

```

Si no conocéis las anotaciones de JUnit 4 os invito a que leáis este [estupendo tutorial](#) de [Carlos García](#)

5. Probando DAO

Otra funcionalidad muy útil de [Gienah](#) es que permite englobar un método en una transacción y, una vez esta acabe, hacerle un rollback para dejar la base de datos tal y como estaba antes de iniciar el test. De este modo nos aseguramos de que no haya dependencias entre test y que todas las pruebas se realizan sobre una base de datos con la misma carga de datos. Para ello sólo hemos de añadir la anotación `@Transactional`

```

view plain print ?
01. @RunWith(value = SpringRunner.class)
02. @Configuration(locations = {"persistence.xml"})
03. public class AutentiaTest {
04.
05.     @Dependency
06.     private AutentiaDao autentiaDao;
07.
08.     @Test
09.     @Transactional
10.     public void testDeleteAutentiaTable() {
11.         autentiaDao.deleteAll();
12.         Assert.assertTrue(autentiaDao.findAll().size() == 0);
13.     }
14.

```

En este ejemplo, al finalizar el test, la tabla `Autentia` volverá a tener la carga que tenía antes de iniciar el test y borrar todos sus datos, ya que se hace rollback al finalizar el método.

6. Conclusiones

En mi opinión [Gienah](#) es una librería muy útil porque nos ahorra montar una infraestructura algo engorrosa y, además, daros cuenta de lo limpio que queda el código y de lo sumamente sencillo que se hace el realizar pruebas unitarias para beans declarados en Spring. No sólo eso, sino que poder declarar los métodos transaccionales también juega una baza importante en pruebas de integración con bases de datos.

En este tutorial me he limitado a compartir con vosotros mi conocimiento sobre [Gienah](#), no obstante, también se pueda hacer alguna cosa más con esta librería:

<http://groups.google.com/group/gienah-testing/web/project-5-minutes-tutorial>

Tras este tutorial espero que tengáis un motivo más para usar test unitarios en vuestros proyectos con Spring :-)

Espero que os sea de utilidad.

- 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 suscripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Pruebas Web con JWebUnit	Os mostramos como automatizar las pruebas de caja negra (desde el punto de vista de usuario final) de vuestro Web con el Framework gratuito JWebUnit. Esta técnica es perfecta para crear test de regresión de aplicaciones Web complejas.	2004-06-30	8409	pdf
Pruebas de integración con Maven	Este tutorial nos muestra un ejemplo para lanzar las pruebas de integración "engañando" a Maven para que no se lanzen en la fase de test teniendo únicamente un módulo para ambas	2007-02-08	2574	pdf
SpringIDE, plugin de Spring para Eclipse	En adictosaltrabajo os hemos ido presentando diversos plugins para Eclipse. Esta vez le toca el turno a SpringIDE, un plugin que os ayudará a desarrollar aplicaciones que utilicen Spring.	2008-01-19	1198	pdf
JUnit 4. Pruebas de Software Java	Tutorial que describe como utilizar la herramienta JUnit 4 para realizar pruebas de integridad y errores sobre Java.	2006-06-02	8639	pdf
Creación de un repositorio maven interno accesible por SSH	En este tutorial os enseñamos a hacer un repositorio maven privado para vuestro equipo de desarrollo	2008-01-03	904	pdf
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	2708	pdf
Spring: definición dinámica de Beans	Este tutorial habla sobre la modificación dinámica de los beans del contexto para simplificar la configuración de Spring	2007-05-09	3059	pdf
JUnit (3.8 y 4) y como ejecutar en un orden determinado los métodos de test de una clase de test	En este tutorial Alejandro Pérez nos enseña a ejecutar test de junit (v 3 y 4) segun un cierto orden.)	2008-02-11	436	pdf
Creación de una aplicación con Spring e Hibernate desde 0	Este tutorial vamos a explicar paso a paso cómo crear una pequeña aplicación usando Spring e Hibernate con anotaciones partiendo desde 0	2008-02-15	1612	pdf
Test con JUnit	Cuando se hacen desarrollo profesionales, no basta con hacer los programas, hay que asegurarse de que van a funcionar. Una de las técnicas más seguras es crear aplicaciones que incluyan el código para autoprobarse. Os mostramos como usar JUnit	2003-06-21	16956	pdf

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.