

¿Qué ofrece Autentia Real Business Solutions S.L?

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

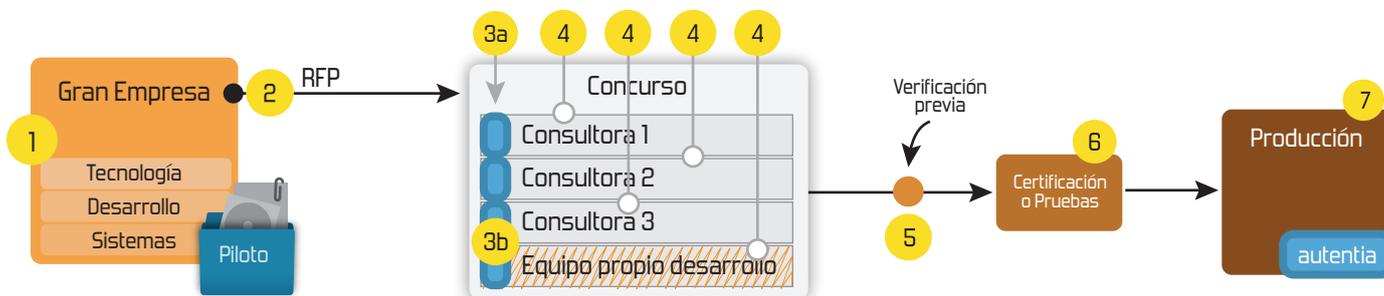
1. Desarrollo de componentes y proyectos a medida



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)



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

[Ver cursos que ofrece Autentia](#)

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



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



[iNUEVO!] 2008-12-01

2008-11-17

2008-09-01

2008-07-31

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



Germán Jiménez Centeno

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática *

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/J2EE

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

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

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



[Catálogo de cursos](#)

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

Fecha de creación del tutorial: 2009-01-29

StrutsTestCase

- [Introducción](#)
- [El código de ejemplo](#)
- [Conclusiones](#)

Introducción

StrutsTestCase es una extensión de junit que permite realizar pruebas sobre aplicaciones basadas en Struts.

Proporciona dos maneras diferentes de realizar las pruebas: mediante mock objects (las clases ServletRequest, ServletResponse y aquellas manejadas por el servidor de aplicaciones son mock objects), o mediante Cactus (un contenedor de servlets ligero). El cambiar entre cualquiera de las dos aproximaciones es tan sencillo como cambiar la clase de la que heredará nuestro test, sin que el test en sí cambie.

Personalmente prefiero usar mock objects en lugar de Cactus. Usar Cactus implica que nuestro proyecto tiene más dependencias y configuraciones. Con mock objects todo queda más sencillo, aunque dependiendo del tipo de aplicación necesitaremos realizar ciertas inicializaciones, como veremos en el ejemplo que sigue.

Podéis descargarlo y acceder a la documentación <http://strutstestcase.sourceforge.net/>.

Otras opciones para realizar este tipo de pruebas es utilizar otras herramientas que nos permiten simular navegaciones en cualquier aplicación web (ya no nos estaríamos casando con struts), y como lo normal es que cada acción funcional este asociada a un action de struts, realizaríamos pruebas sobre los action (con salvedades como que ya dependemos de un servidor de aplicaciones y la configuración que todo ellos conlleva). Una herramienta muy útil para estos menesteres, Selenium, la podéis ver en el tutorial de mi compañero Víctor: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=seleniumIDE>.

El código de ejemplo

Veremos cómo usar StrutsTestCase para una aplicación que usa Struts 1.3, contruida con Maven.

La aplicación trata de una sencilla librería por internet, donde se necesita estar logado para realizar las acciones, y finalmente se pueden realizar búsquedas filtradas de los libros.

Veamos como ejemplo una clase de acción de la que vamos a realizar tests:

Catálogo de servicios Autentia (PDF 6,2MB)



[En formato comic...](#)



Web
 www.adictosaltrabajo.com

Últimos tutoriales

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](#)

2009-01-20
[Pruebas Software con Junit 4 y Eclipse](#)

2009-01-19
[Executor : Un programa para ejecutarlos a todos.](#)

2009-01-18
[Soap Monitor: Monitorización de mensajes SOAP en Axis2](#)

Últimas ofertas de empleo

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

```

view plain print ?
01. public class LoginAction extends GenericAction {
02.
03.     private static final Log log = LoggerFactory.getLog(LoginAction.class);
04.
05.     @Override
06.     public ActionForward executeAction(ActionMapping aM, ActionForm aF,
07.         HttpServletRequest request, HttpServletResponse response)
08.         throws Exception {
09.
10.         final LoginForm form = (LoginForm) aF;
11.         String forw = "error";
12.
13.         final String name = form.getName();
14.         final String pwd = form.getPwd();
15.
16.         final ActionErrors errors = new ActionErrors();
17.
18.         if (log.isDebugEnabled()) {
19.             log.debug("pasando por loginAction");
20.         }
21.
22.         if ("test".equals(name) && "test".equals(pwd)) {
23.             final User logged = new User();
24.
25.             logged.setLogin("test");
26.             logged.setPassword("test");
27.             logged.setNombre("German");
28.             logged.setApellido("Jimenez");
29.             logged.setEmail("gjimenez@autentia.com");
30.
31.             request.getSession().setAttribute("loggedUser", logged);
32.             forw = "success";
33.
34.         } else {
35.
36.             errors.add("errors.login.fail", new ActionMessage(
37.                 "errors.login.fail"));
38.             addErrors(request, errors);
39.             forw = "error";
40.         }
41.
42.         return aM.findForward(forw);
43.     }
44. }
45.

```

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

Y la clase de la que hereda, que verifica que un usuario esté o no logado para todas las acciones:

```

view plain print ?
01. public abstract class GenericAction extends Action {
02.
03.     private static final Log log = LoggerFactory.getLog(GenericAction.class);
04.
05.     private boolean validateUser(ActionMapping aM, HttpServletRequest req,
06.         HttpServletResponse res) throws Exception {
07.
08.         return req.getSession(false).getAttribute("loggedUser") != null;
09.     }
10.
11.     /*
12.     * (non-Javadoc)
13.     * @see org.apache.struts.action.Action#execute(org.apache.struts.action.
14.     * ActionMapping, org.apache.struts.action.ActionForm,
15.     * javax.servlet.http.HttpServletRequest,
16.     * javax.servlet.http.HttpServletResponse)
17.     */
18.     @Override
19.     public ActionForward execute(ActionMapping aM, ActionForm aF,
20.         HttpServletRequest request, HttpServletResponse response)
21.         throws Exception {
22.
23.         boolean isLogged = true;
24.
25.         if (!(request.getContextPath() + "/login.do").equals(request
26.             .getRequestURI())) {
27.             isLogged = validateUser(aM, request, response);
28.         }
29.
30.         if (log.isDebugEnabled()) {
31.             log.debug("Pasando por la validacion de usuario: " + isLogged);
32.         }
33.
34.         return isLogged ? executeAction(aM, aF, request, response)
35.             : new ActionForward("/pages/login.jsp");
36.     }
37.
38.     public abstract ActionForward executeAction(ActionMapping aM,
39.         ActionForm aF, HttpServletRequest request,
40.         HttpServletResponse response) throws Exception;
41.
42.
43. }

```

Por último nos quedaría ver la clase de test de junit con su extensión de StrutsTestCase (veréis que es un test de JUnit 3, ya que no podemos usar test de JUnit 4 con StrutsTestCase) (aunque JUnit 4 es capaz de ejecutar tests de JUnit 3):

view plain print ?

```
01. public class LibraryTest extends MockStrutsTestCase {
02.
03.     @Override
04.     protected void setUp() throws Exception {
05.         super.setUp();
06.
07.         final DAO<Book> dao = new BookDAO();
08.         final List<Editorial> editoriales = Arrays.asList(Editorial.EDITORIALS);
09.         final List<Author> author = Arrays.asList(Author.AUTHORS);
10.
11.         // Debemos establecer para cada prueba el contextPath, puesto que L
12.         // request es un
13.         // mock object y nadie la establece
14.         getMockRequest().setContextPath("/library");
15.
16.         // establecemos ciertas variables en el contexto de aplicacion que s
17.         // supone
18.         // que inicializa un listener
19.         getSession().getServletContext().setAttribute("editorialesSelect",
20.             editoriales);
21.         getSession().getServletContext().setAttribute("autoresSelect", author);
22.
23.     try {
24.         // Anadimos los datos de prueba a nuestro DAO
25.         // Lo ideal seria utilizar mock objects e integrar con Spring
26.         dao.update(new Book("El fuego", "", Integer.valueOf(1), "", "",
27.             Float.valueOf(0), editoriales.get(0), author.get(0)));
28.         dao.update(new Book("Luna nueva", "", Integer.valueOf(1), "", "",
29.             Float.valueOf(0), editoriales.get(1), author.get(1)));
30.         dao.update(new Book("Crepusculo", "", Integer.valueOf(1), "", "",
31.             Float.valueOf(0), editoriales.get(2), author.get(1)));
32.         dao.update(new Book("El club de las malas madres", "", Integer
33.             .valueOf(1), "", "", Float.valueOf(0), editoriales.get(5),
34.             author.get(3)));
35.         dao.update(new Book(
36.             "El seÑor de los anillos: la comunidad del anillo", "",
37.             Integer.valueOf(1), "", "", Float.valueOf(0), editoriales
38.             .get(3), author.get(4)));
39.         dao.update(new Book("El seÑor de los anillo: las dos torres", "",
40.             Integer.valueOf(1), "", "", Float.valueOf(0), editoriales
41.             .get(3), author.get(4)));
42.         dao.update(new Book("El seÑor de los anillo: el retorno del Rey",
43.             "", Integer.valueOf(1), "", "", Float.valueOf(0),
44.             editoriales.get(3), author.get(4)));
45.     } catch (DAOException e) {
46.
47.     }
48.
49.     // el dao se encuentra en el contexto de la aplicacion
50.     getSession().getServletContext().setAttribute("bookDao", dao);
51. }
52.
53. public void testShouldLogin() {
54.
55.     // Establecemos la url de la request nosotros mismos puesto
56.     // que la request es un mock object y nadie lo establece
57.     getMockRequest().setRequestURI("/library/login.do");
58.
59.     // indicamos el action a probar
60.     setRequestPathInfo("/login");
61.     // andamos los parametros necesarios, que sera como
62.     // si enviaramos en un formulario
63.     addRequestParameter("name", "test");
64.     addRequestParameter("pwd", "test");
65.
66.     // ejecutamos la accion
67.     actionPerform();
68.
69.     // verificamos que recibimos el forward succes:
70.     verifyForward("success");
71.
72.     // obtenemos el usuario logado en sesion y comprobamos su existencia
73.     final User logged = (User) getSession().getAttribute("loggedUser");
74.     assertNotNull(logged);
75.
76.     // comprobamos que sea valida
77.     assertEquals("test", logged.getLogin());
78.     assertEquals("test", logged.getPassword());
79.     assertEquals("German", logged.getNombre());
80.     assertEquals("Jimenez", logged.getApellido());
81.     assertEquals("gjimenez@autentia.com", logged.getEmail());
82.
83.     // verificamos que no existan ActionErrors:
84.     verifyNoActionErrors();
85.
86. }
87.
88. public void testShouldLoginFail() {
89.
90.     // Establecemos la url de la request nosotros mismos puesto
91.     // que la request es un mock object y nadie lo establece
92.     getMockRequest().setRequestURI("/library/login.do");
93.
94.     // indicamos el action a probar
95.     setRequestPathInfo("/login");
96.
97.     // anadimos los parametros necesarios, que sera como
```

Veamos ciertas características del test:

- Si reescribimos el método setUp, debemos **explícitamente llamar a super.setUp()**. Si no lo hacemos StrutsTestCase no se inicializará correctamente.
- Hereda de MockStrutsTestCase (si quisieramos Cactus debe heredar de CactusStrutsTestCase, y configurar Cactus).
- Al usar mock objects en la ServletRequest y ServletResponse, hay ciertos atributos que nadie setea (como contextPath o requestURI), pero que podemos o tendremos que establecer nosotros, accediendo a los objetos derequest y response mediante getMockRequest() y getMockResponse() y realizar las operaciones necesarias.
- Sin embargo todos aquellos atributos que nuestro action ponga en sesión o cualquier otro contexto se mantendrá durante toda la prueba (cada uno de los métodos del test de JUnit), por tanto si queremos probar una búsqueda y necesitamos estar logados, en la prueba de búsqueda debemos logarnos antes.
- Por defecto busca el fichero struts-config.xml. Si tuviéramos diferentes nombres se pueden indicar con el método setConfigFile(pathname).
- Los ficheros web.xml y struts-config.xml **deben** estar en el classpath para que StrutsTestCase los pueda encontrar y procesar. Una solución, si se realiza desde eclipse, es añadir la carpeta webapp al build path (StrutsTestCase busca realmente WEB-INF/web.xml y WEB-INF/struts-config.xml). Desde maven hemos de configurar el plugin surefire, como veremos a continuación

Por último veamos cómo quedaría nuestro fichero de pom, para ver las dependencias necesarias y la configuración del plugin de surefire para añadir la carpeta webapp al classpath:

```
view plain print ?
01. <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
02.   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd"
03.   <modelVersion>4.0.0</modelVersion>
04.   <groupId>com.autentia.training</groupId>
05.   <artifactId>library</artifactId>
06.   <packaging>war</packaging>
07.   <version>1.0-SNAPSHOT</version>
08.   <name>library-core Maven Webapp</name>
09.   <url>http://maven.apache.org</url>
10.
11.   <build>
12.     <plugins>
13.       <plugin>
14.         <groupId>org.apache.maven.plugins</groupId>
15.         <artifactId>maven-surefire-plugin</artifactId>
16.         <configuration>
17.           <additionalClasspathElements>
18.             <additionalClasspathElement>/webapp</additionalClasspathElement>
19.           </additionalClasspathElements>
20.         </configuration>
21.       </plugin>
22.     </plugins>
23.   </build>
24.
25.   <dependencies>
26.
27.     <!-- Dependencias internas necesarias -->
28.   <dependency>
29.     <groupId>org.apache.struts</groupId>
30.     <artifactId>struts-core</artifactId>
31.     <version>1.3.10</version>
32.   </dependency>
33.   <dependency>
34.     <groupId>org.apache.struts</groupId>
35.     <artifactId>struts-taglib</artifactId>
36.     <version>1.3.10</version>
37.   </dependency>
38.   <dependency>
39.     <groupId>org.apache.struts</groupId>
40.     <artifactId>struts-tiles</artifactId>
41.     <version>1.3.10</version>
42.   </dependency>
43.   <dependency>
44.     <groupId>commons-collections</groupId>
45.     <artifactId>commons-collections</artifactId>
46.     <version>3.2.1</version>
47.   </dependency>
48.   <dependency>
49.     <groupId>strutstestcase</groupId>
50.     <artifactId>strutstestcase</artifactId>
51.     <version>2.1.4-1.2-2.4</version>
52.   </dependency>
53. </dependencies>
54. </project>
```

Conclusiones

Ya veis que en [Autentia](#) nos gusta trabajar bien, y con las últimas herramientas, que nos ahorren tiempo y esfuerzo, así que no dudéis en contactar con nosotros para más información.

¿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 AdictosAITrabajo en XING haciendo clic [aquí](#).



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

soluciones reales para su negocio

Servicio de notificaciones:

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

Formulario de subscripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	pdf
Struts y EL en Netbeans 4	Os mostramos como configurar la última contribución de Struts que se integra con el lenguaje de expresiones EL.	2004-11-15	20367	-	pdf
Introducción a Struts Flow	Struts Flow es un módulo de extensión del conocido framework Struts, que facilita la implementación del flujo de páginas de una aplicación web	2006-01-02	8823	-	pdf
Uso de Tiles en Struts	Os mostramos como utilizar el sistema de plantillas proporcionado por Struts (tiles)	2004-11-14	19297	-	pdf
Consola de administración de Struts	En este tutorial aprenderemos a simplificar la gestión de Struts a través de una consola gráfica gratuita	2003-06-26	17569	-	pdf
Integración de Struts y eclipse	Alejandro Perez nos enseña como construir un entorno de alta eficiencia de desarrollo on Struts a través de plugins de eclipse	2003-11-28	53787	-	pdf
Activar el soporte SSL en Struts	Os mostramos las particularidades de uso y configuración de Struts para trabajar con SSL	2005-12-06	11009	-	pdf
Evitar doble-click en JSPs y Struts	Os mostramos como construir unas librerías de TAGs para evitar el problema de doble-click en aplicaciones JSP y como se soluciona (qué teneis que hacer) con el framework Struts	2004-12-06	13341	-	pdf
Plantear una aplicación Web y Struts	Os mostramos un posible modo de plantear una aplicación Web (análisis) y darla forma. El FrameWork utilizado es struts y tratamos de identificar qué depende de este FrameWork y qué no.	2004-11-05	41111	-	pdf
Extender la validación en Struts	Os mostramos con un ejemplo como extender los mecanismos de validación en Struts, utilizando el framework Commons Validator	2006-02-17	14133	-	pdf
Múltiples struts-config.xml e internacionalización de Jasper Report	En este tutorial queremos mostraros como dividir el struts-config.xml en dos o más ficheros, pero sin usar la capacidad de módulos que tiene struts y como generar informes con la ayuda de Jasper Report con un contenido diferente dependiendo del idioma del	2007-03-19	13984	-	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.