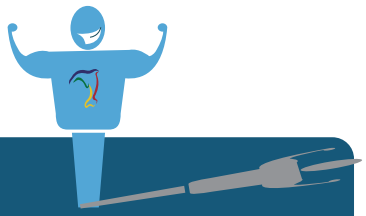
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

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

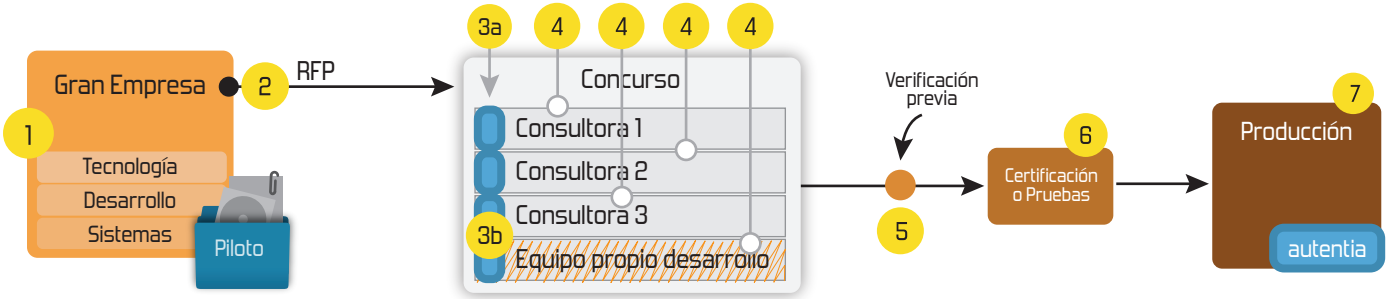
1. Desarrollo de componentes y proyectos a medida



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
- 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
- Tareas programadas (Quartz)
- Gestor documental (Alfresco)
- Inversión de control (Spring)
- BPM (jBPM o Bonita)
- Generación de informes (JasperReport)
- ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

DESARROLLADO POR:
Jose Manuel Sánchez Suárez

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

Anuncios Google Spring SMTP Server Swing Class Java Class Tutorial

Fecha de publicación del tutorial: 2011-02-20

192

Share | Regístrate para votar

Pruebas de integración del envío de Email con el soporte de Spring.

0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno.
- 3. Adaptando las librerías a la última versión de Spring.
- 4. Añadiendo aserciones al test con el soporte del servidor SMTP.
- 5. Haciendo uso del soporte de Spring para inyectar el servidor SMTP.
- 6. Referencias.
- 7. Conclusiones.

1. Introducción

Ya vimos hace algún tiempo las posibilidades que teníamos con el soporte de Spring de enviar emails simples, con attachments, basados en plantillas y con recursos en línea:

- Integración de Spring con el envío de emails.
- Integración de Spring con el envío de emails: técnicas avanzadas (I).
- Integración de Spring con el envío de emails: técnicas avanzadas (II)

En estos tutoriales se explican los pasos a dar para configurar el servicio y se realiza una prueba de envío a través de un test de junit. Si bien, dicho test no es tal, puesto que carece de aserciones, el objetivo del mismo era el de levantar el contenedor de Spring y probar el envío efectivo a través de un servidor real, pero como digo, la comprobación del resultado era manual y a posteriori de la ejecución del test.

Los test de integración son siempre los más complejos de testar puesto que emular el sistema de terceros con el que nos integramos en producción en un entorno de pruebas no es trivial.

¿Alguien ha participado de un proyecto en el que haya comenzado a fallar la build porque en producción se ha modificado la password del servidor de correo?, no hay que llegar a estos extremos, hacer un test de envío de mail que envíe un correo real no es de recibo.

Es aquí donde entra el objeto de este tutorial, realizar una prueba de concepto de un mock que emula un servidor de correo SMTP local.

Este tutorial también sería válido como complemento del de envío de correo electrónico con el soporte de Jboss Seam, pero nos vamos a centrar en el soporte de Spring.

2. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 17' (2.93 GHz Intel Core 2 Duo, 4GB DDR3 SDRAM).
- Sistema Operativo: Mac OS X Snow Leopard 10.6.1
- Spring 3.0.5.RELEASE
- Dumbster 1.6

3. Adaptando las librerías a la última versión de Spring.

Lo primero que hemos hecho es adaptar nuestras librerías a las últimas versiones e incluir la dependencia a librería dumbster, que tiene el mock que nos interesa.

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context-support</artifactId>
    <version>3.0.5.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>3.0.5.RELEASE</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>dumbster</groupId>
    <artifactId>dumbster</artifactId>
    <version>1.6</version>
    <scope>test</scope>
</dependency>
```

El uso de dicha librería fue propuesto por Sam Brannen en el #springio de este año, como ejemplo de una buena práctica de test de integración, en su charla sobre "Effective out-of-container Integration Testing"

4. Añadiendo aserciones al test con el soporte del servidor SMTP.

El código de nuestro test en relación con el envío del mail es el mismo, lo que difiere es la necesidad de arrancar y parar el servidor de SMTP y las aserciones que podemos realizar con el soporte del mismo.

```

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = { "/applicationContext.xml" })
public class MailServiceTest {

    private static final Log log = LogFactory.getLog(MailServiceTest.class);

    private static final String EMAIL = "jmsanchez@autentia.com";
```

Catálogo de servicios Autentia

Últimas Noticias

- Hablando de coaching ágil, milagro nocturo y pruebas de vida
- XIII Charla Autentia - AOS y TDD - Vídeos y Material
- Las metodologías ágiles como el catalizador del cambio
- XIV Charla Autentia - ZK
- Informática profesional: Las reglas no escritas para triunfar en la empresa. 2ª EDICIÓN ACTUALIZADA.

Histórico de NOTICIAS

Últimos Tutoriales

- CRUD con Spring MVC Portlet (II)
- ZK - Añadir versiones de ZK al ZK Studio en Eclipse y cambiarle la versión de ZK a un proyecto.
- ZK - Instalar Studio en Eclipse
- ZK - ¿Cómo crear tu primer proyecto con ZK?
- CRUD con Spring MVC Portlet

Últimos Tutoriales del Autor

- Selección múltiple de filas en un datatable con JSF: haciendo uso de librerías de componentes.
- Selección múltiple de filas en un datatable con JSF: haciéndolo nosotros mismos.
- Gestión del evento de cambio de valor en JSF2: valueChangeListener.
- Obtención de la fila seleccionada en un datatable con JSF2
- Eclipse custom templates

Síguenos a través de:

Últimas ofertas de empleo

2010-10-11
Comercial - Ventas - SEVILLA.

2010-08-30
Otras - Electricidad - BARCELONA.

```
09
10      private static final String SUBJECT = "Entregable del #springio.";
11
12      private static final String BODY = "Probando Fake SMTP Server en el envío de email con el soporte de Spring.
By #autentia.";
13
14      @Resource
15      private MailService mailService;
16
17      private SimpleSmtpServer server;
18
19      @Before
20      public void beforeTest(){
21          server = SimpleSmtpServer.start(2525);
22      }
23
24      @Test
25      public void sendSimpleEmail() {
26
27          mailService.send(EMAIL, SUBJECT, BODY);
28
29          server.stop();
30
31          Assert.assertEquals("There must be a single email.",1,server.getReceivedEmailSize());
32
33          final Iterator<?> emailIter = server.getReceivedEmail();
34          final SmtpMessage email = (SmtpMessage)emailIter.next();
35
36          Assert.assertEquals(SUBJECT, email.getHeaderValue("Subject"));
37          Assert.assertEquals(EMAIL, email.getHeaderValue("To"));
38
39          Assert.assertTrue(email.getBody().contains("By #autentia."));
40
41      }
42
43  }
```

Hemos añadido un método beforeTest para arrancar el servidor antes de la ejecución de cada test.

Después del envío del email, paramos el servidor y primero comprobamos que el número de emails recibidos en el mismo sea igual a uno.

Sabiendo que hay uno, lo obtenemos de la iteración y comprobamos su subject, el campo to y que contenga un texto predefinido en el body.

5. Haciendo uso del soporte de Spring para inyectar el servidor SMTP.

Para que funcione correctamente, hemos tenido que modificar el fichero de configuración de nuestra aplicación, en el entorno de test, para que las propiedades del servidor del correo, sean

- mail.host=localhost
- mail.port=2525

Y mientras lo cambiaba, la inercia me ha llevado a replantearme la configuración puesto que estaba escribiendo en dos sitios distintos la misma cosa, la configuración del puerto.

La solución pasa por publicar el servidor de SMTP en el contexto de Spring, como sigue:

```
1  <bean id="simpleSmtpServer" class="com.dumbster.smtp.SimpleSmtpServer" factory-method="start" scope="prototype">
2      <constructor-arg value="${mail.port}"/>
3  </bean>
```

Con mail.port se inicializa tanto el servicio como el servidor, con lo que no me repito (DRY) y puedo inyectar ambos en cualquier punto manejado por el contenedor, incluso en los tests.

```
01  @RunWith(SpringJUnit4ClassRunner.class)
02  @ContextConfiguration(locations = { "/applicationContext.xml" })
03  public class MailServiceTest {
04
05      private static final Log log = LogFactory.getLog(MailServiceTest.class);
06
07      private static final String EMAIL = "jmsanchez@autentia.com";
08
09      private static final String SUBJECT = "Entregable del #springio.";
10
11      private static final String BODY = "Probando Fake SMTP Server en el envío de email con el soporte de Spring.
By #autentia.";
12
13      @Resource
14      private MailService mailService;
15
16      @Resource
17      private SimpleSmtpServer server;
18
19      @Test
20      public void sendSimpleEmail() {
21
22          mailService.send(EMAIL, SUBJECT, BODY);
23
24          server.stop();
25
26          Assert.assertEquals("There must be a single email.",1,server.getReceivedEmailSize());
27
28          final Iterator<?> emailIter = server.getReceivedEmail();
29          final SmtpMessage email = (SmtpMessage)emailIter.next();
30
31          Assert.assertEquals(SUBJECT, email.getHeaderValue("Subject"));
32          Assert.assertEquals(EMAIL, email.getHeaderValue("To"));
33
34          Assert.assertTrue(email.getBody().contains("By #autentia."));
35
36      }
37
38  }
```

El bean lo publicamos como prototype para que cada vez que se inyecte se cree una nueva instancia del mismo y se pueda parar el servidor en los tests sin problemas.

6. Referencias.

- http://quintanasoft.com/dumbster/

7. Conclusiones.

Desde la publicación de los tutoriales sobre envío de emails he recibido algún que otro mensaje de correo electrónico a mi buzón porque quien andaba probando el código de los mismos no cambiaba mi dirección de email del campo to, espero que con este tutorial dejen de llegar ;)

Ahora en serio, me tomo este tutorial como un entregable después del #springio de este año. Cuando te aportan tanto y de tanta calidad te sientes en deuda y quería realizar yo también una pequeña aportación a la comunidad.

Un saludo.

Jose

jmsanchez@autentia.com

Anímate y coméntanos lo que pienses sobre este **TUTORIAL:**
Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

2010-08-24
Otras Sin catalogar - LUGO.

2010-06-25
T. Información - Analista / Programador - BARCELONA.

Jose Manuel Sánchez
sanchezsuarezj

tras el #springio y gracias a la charla de @sam_brannen sobre Effective out-of-container Integration Testing, un tuto > http://ow.ly/3ZP7i
yesterday · reply

@miguelinlas3 muy buena la exposición y muy interesante la tecnología. #springio #hadoop
2 days ago · reply

@gimenete gracias a ti, salimos con ganas de probarlo!
2 days ago · reply

#springio #wurfl un viejo conocido, ahora con el soporte de #springMobile.
2 days ago · reply

Join the conversation

Enviar comentario

(Sólo para usuarios registrados)

» **Registrate** y accede a esta y otras ventajas «

Comentarios

franferri *2011-02-20 - 11:33:01*

Más simple y claro imposible.

SOME RIGHTS RESERVED Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

Copyright 2003-2011 © All Rights Reserved | Texto legal y condiciones de uso | Banners | Powered by Autentia | Contacto

W3C XHTML 1.0 W3C CSS XML RSS XML ATOM