

¿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)



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Foros](#) | [Tutoriales](#) | [Servicios Gratuitos](#) | [Contacte](#)

Tutorial desarrollado por: [Miguel Angel Mena Sevilla](#)

Puedes encontrarme en [Autentia](#)



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

[Master Java J2ee Oracle](#)

Prácticas laborales 100% aseguradas Nuevo temario de Struts.
Trabaja ya
www.grupoatrium.com

[Curso Web J2EE](#)

Curso Avanzado en Desarrollo Web con J2EE
www.eps.mondragon.edu/caj2ee

Anuncios Goooooogle

Anúnciese en este sitio

Extensión de SSL para Struts

Prólogo (por [Roberto Canales Mora](#))

Struts es uno de los Frameworks elegidos dentro de www.autentia.com para nuestro trabajo diario. Hay que entender que Struts no es más que la base de lo que podría ser un Framework más completo y de más alto nivel (tipo [Spring](#)). Para que un Framework de este estilo sea realmente útil hay que invertir tiempo y recursos en mejorarlo y ampliarlo, dotándolo de sistemas sencillos de control de:

- Seguridad
- Velocidad
- Navegación
- Caché
- Trazas de CRM operacional
- Etc.

Entre las ventajas de Struts cabe destacar que, sin demasiada complejidad (*curva de aprendizaje reducida*), te permite extenderlo para adaptarlo a tus necesidades, fundamentalmente en empresas grandes. Hay además muchas extensiones estándar.

Una de las cosas que te permite (y que mucha gente obvia) es que si las piezas que hay que construir son las mismas (acciones, formularios, jsps, tags ...) **podemos crear una metodología de análisis y diseño orientado a objeto** (el análisis no es Orientado a Objeto ... aunque puede ser [amigoso](#)) para sacarle partido a este Framework (uno de nuestro cursos más demandados en Autentia.com <http://www.adictosaltrabajo.com/plantillas/autentiafor.php>)

Hoy Miguel nos va a contar como resolver de un modo estándar uno de los problemas con Struts forzar el uso de SSL para transacciones seguras ...

Introducción

En casi todas las aplicaciones Struts orientadas a comercio, es necesario el uso de conexiones seguras por SSL para garantizar la seguridad de las comunicaciones electrónicas. No obstante, dado el bajo porcentaje (un diez por ciento) de transmisiones por HTTPS, no resulta eficiente desplegar una aplicación Web completa en SSL. La solución con mayor rendimiento propone desplegar una aplicación en HTTP y emplear HTTPS únicamente con aquellas páginas que transmitan datos comprometidos.

La solución ideal es proponer un sistema que utilice de manera automática el protocolo adecuado en el momento de acceder a los recursos de la Red (en tiempo de ejecución).

En la actualidad, Struts 1.0 y superiores ya incluyen una extensión que implementa esta funcionalidad de redirección: `ssltext`. Define un marco de trabajo en el que los desarrolladores pueden especificar qué peticiones de acciones requieren HTTPS y cuáles HTTP, además de redirigir peticiones erróneas (este producto puede obtenerse del correspondiente [enlace](#) en SourceForge.net). Este tutorial únicamente pretende enseñar el funcionamiento de esta extensión mostrando su implementación.

Extensión de Struts para SSL

A continuación vamos a explicar el funcionamiento de SSL mediante un sencillo tutorial, en el que describimos todos los pasos para extender la funcionalidad de Struts para que pueda redirigir de forma dinámica las URL por

el protocolo correspondiente.

1. Actualización de propiedades en *struts-config.xml*

Para detectar si una petición de acción ha de ser transmitida por HTTP (puerto 80) o por HTTPS (puerto 443), debemos añadir en *struts-config.xml* una nueva propiedad que interprete esta especificación y transmita la petición por el puerto correspondiente.

Para poder ampliar la funcionalidad de la etiqueta `<action>`, Struts define la etiqueta `<set-property>`, que permite asignar un nombre y un valor de propiedad nuevos a la acción.

Procedemos a crear una propiedad de nombre `secure` en `MiAccion`:

```
<action path="/MiAccion" type="test.ssl.MiAccion" >
  <set-property property="secure" value="true"/>
  <forward name="success" path="/home.jsp" />
  <forward name="failure" path="/error.jsp" />
</action>
```

El valor de `secure` nos indica si la transmisión es por HTTP (`true`) o por HTTPS (`false`).

2. Creación de una extensión de *ActionMapping*

Para usar esta nueva propiedad, tenemos que extender la definición de `ActionMapping` para que la incluya. Para ello, definimos una extensión de `ActionMapping`, a la llamaremos `SecureActionMapping`, la cuál especificará nuestra propiedad.

A continuación, presentamos la definición de `SecureActionMapping`:

```
/**
 * SecureActionMapping class
 */
package org.apache.struts.action;

public class SecureActionMapping extends ActionMapping {

    protected boolean secure;

    public void setSecure(boolean b){
        this.secure = b;
    }

    public boolean getSecure(){
        return this.secure;
    }
}
```

3. Actualización de mapping en *web.xml*

Para poder utilizar nuestra nueva clase, debemos indicarle al `ActionServlet` que debe utilizar `SecureActionMapping` en lugar de `ActionMapping`. Para realizar esto, hay que actualizar la entrada de servlet de `ActionServlet` en el fichero `web.xml`: basta con añadir otro parámetro de inicialización, de nombre `mapping`, que indique a nuestro `ActionServlet` qué clase usar para mapear (nota: tiene que ser una clase que herede de `ActionMapping`).

Nuestro fichero `web.xml` debería mostrar algo así:

```
<servlet id="Servlet_1">
  <servlet-name>action</servlet-name>
  <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>

  <!-- Struts Config -->
  <init-param>
    <param-name>config</param-name>
    <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>

  <init-param>
    <param-name>mapping</param-name>
    <param-value>org.apache.struts.action.SecureActionMapping</param-value>
  </init-param>
```

```

<!-- other initialization parameters -->
<load-on-startup>1</load-on-startup>
</servlet>

```

Ahora, cuando `ActionServlet` parsea **struts-config.xml**, crea una instancia de `SecureActionMapping` para `MiAccion`. Cuando capta la nueva propiedad `secure`, encuentra el método `getSecure()`.

4. Actualización de servlet en web.xml

Ahora, hay que especificar dos nuevas propiedades que contengan los números de puerto de los protocolos HTTP y HTTPS. Estas propiedades son definidas como cualquier otro parámetro de inicialización. Las llamamos `http-port` y `https-port`: para estos nombres, el sistema les asigna por defecto los valores de puerto 80 y 443 si no son introducidos (nosotros optamos por hacerlo).

Al igual que en los dos pasos anteriores, para poder usar estas propiedades hay que añadirlas en una extensión del fichero `ActionServlet`, a la que vamos a llamar `SecureActionServlet`; después actualizamos la entrada correspondiente en `web.xml` para que detecte nuestro nuevo fichero de configuración de servlets.

En este caso vamos a empezar por la definición de las nuevas propiedades y la captura del nuevo servlet de acción en `web.xml`, dejando para más adelante el código de `secureActionServlet` (por ser bastante extenso).

Este es el aspecto de `web.xml`, tras actualizarlo:

```

<servlet id="Servlet_1">
  <servlet-name>action</servlet-name>
  <servlet-class>org.apache.struts.action.SecureActionServlet</servlet-class>

  <!-- Struts Config -->
  <init-param>
    <param-name>config</param-name>
    <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
  <init-param>
    <param-name>mapping</param-name>
    <param-value>org.apache.struts.action.SecureActionMapping</param-value>
  </init-param>

  <!-- other initialization parameters -->
  <init-param>
    <param-name>http-port</param-name>
    <param-value>8080</param-value>
  </init-param>
  <init-param>
    <param-name>https-port</param-name>
    <param-value>8443</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

```

5. Creación de una extensión de ActionServlet

`SecureActionServlet` implementa la lógica de negocio necesaria para redirigir una petición de acción al protocolo apropiado. En primer lugar encuentra el objeto `SecureActionMapping`, necesario para captar el objeto que contiene la propiedad `secure`. A continuación, comprueba el valor de `secure` en ese objeto para determinar qué protocolo utilizar.

Este es el aspecto del archivo `SecureActionServlet`, una vez implementado:

```

package org.apache.struts.action;

import org.apache.struts.util.SecureRequestUtils;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

/**
 * An extension of the ActionServlet class to add properties for
 * http and https ports. Performs logic to determine if
 * requests should be redirected to http or to https protocol.
 */
public class SecureActionServlet extends ActionServlet {

```

```

protected String httpPort = "80";
protected String httpsPort = "443";

public void init() throws ServletException {
    super.init();
    initPorts();
}

/**
 * Initializes the http and https port properties
 * from init parameters specified in the web.xml file.
 * If not found there, default values remain.
 */
protected void initPorts(){
    String value = getServletConfig().getInitParameter("http-port");

    if( value != null ){
        // Other validation code here.
        httpPort = value;
    }

    value = getServletConfig().getInitParameter("https-port");

    if( value != null ){
        // Other validation code here.
        httpsPort = value;
    }
}

public void setHttpsPort( String s ){
    this.httpsPort = s;
}
public void setHttpPort( String s ){
    this.httpPort = s;
}
public String getHttpsPort( ){
    return this.httpsPort;
}
public String getHttpPort( ){
    return this.httpPort;
}

/**
 * Checks whether servlet should redirect to another protocol
 * before calling the superclass process method.
 * @param aRequest is the current request object.
 * @param aResponse is the current response object.
 * @throws IOException if an input/output error occurs.
 * @throws ServletException if a servlet error occurs.
 */
protected void process(HttpServletRequest request,
    HttpServletResponse response
    ) throws IOException, ServletException {

    // Identify the path component we will use to select a mapping.
    String path = processPath(request);
    if (path == null) {
        if (debug >= 1)
            log(" No path available for request URI " +
                request.getRequestURI());
        response.sendError(HttpServletResponse.SC_BAD_REQUEST,
            internal.getMessage("processPath"));
        return;
    }

    // Look up the corresponding mapping.
    SecureActionMapping mapping = (SecureActionMapping)processMapping(path, request);
    if (mapping == null) {
        if (debug >= 1)
            log(" No mapping available for path " + path);
        response.sendError(HttpServletResponse.SC_BAD_REQUEST,
            internal.getMessage("processInvalid", path));
        return;
    }

    // Redirect to https/http if necessary.
    if( checkSsl(mapping, request, response) ){
        return ;
    }
}

```

```

    }
    super.process(request, response);
}

/** Checks to see if SSL should be toggled for this
 * action.
 * @param aMapping is the mapping object for this Action.
 * @param aRequest is the current request object.
 * @param aResponse is the current response object.
 */
private boolean checkSsl(SecureActionMapping aMapping,
    HttpServletRequest aRequest,
    HttpServletResponse aResponse){

    String redirectString =
        SecureRequestUtils.getRedirectString(aRequest,
            getHttpPort(),
            getHttpsPort(),
            aMapping.getSecure());

    if( redirectString != null ){
        try{
            // Redirect the page to the desired URL.
            aResponse.sendRedirect(aResponse.encodeRedirectURL( redirectString));
            return true;
        }catch(Exception ioe){
            System.out.println("IOException in redirect" + ioe.getMessage());
        }
    }
    return false;
}
}
}

```

Referencias

La mayor parte del contenido de este tutorial ha sido realizado a partir de la página “*Mix protocols transparently in Web applications*”, de Steve Ditlinger, en la página oficial de de Java World. Se puede acceder a su artículo desde este [enlace](#).

Conclusiones

No olvidéis que en www.autentia.com os podemos ayudar a trabajar de un modo profesional con Struts o formar a vuestros equipos en "Técnicas Avanzadas de Análisis y Diseño Orientado a Objeto"

Si desea contratar formación, consultoría o desarrollo de piezas a medida puede contactar con

[Autentia S.L.](http://www.autentia.com) Somos expertos en:
J2EE, C++ , OOP, UML, Vignette, Creatividad ..
 y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
<i>e-mail</i>	
	<input type="button" value="Enviar"/>

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto	Descripción
JSF y comparativa con Struts	Os mostramos los pasos necesarios para empezar a utilizar JSF (Java Server Faces) y su comparación / relación con Struts
¿Es criar cerdos tan distinto a trabajar con nuevas tecnologías?	Os proponemos un cuento sobre la evolución de una empresa de crianza de cerdos y distintos problemas en su crecimiento. Seguro que, leyendo entre líneas, podéis sacar conclusiones (podéis no compartir las nuestras) enriquecedoras para vuestro contexto.
Desarrollo Struts con XDoclet	Alejandro Perez nos enseña como simplificar el desarrollo de aplicaciones J2EE basadas en Struts, automatizando la generación de código con XDoclet
Uso de Tiles en Struts	Os mostramos como utilizar el sistema de plantillas proporcionado por Struts (tiles)
Seguridad en Tomcat	Os mostramos como proteger de un modo básico el acceso a recursos dentro de vuestro servidor de componentes Tomcat
Introducción a Hibernate	Cesar Crespo nos enseña como utilizar unos de los sistemas más extendidos de mapeo de objetos a estructuras relacionales (tablas de base de datos)
Consola de administración de Struts	En este tutorial aprenderemos a simplificar la gestión de Struts a través de una consola gráfica gratuita
Pool de conexiones a BBDD con struts	Os mostramos como configurar un pool de conexiones a base de datos en vuestras aplicaciones construidas con struts
Struts Jakarta	Cuando se ha trabajado creando aplicaciones Java poco a poco se va viendo la necesidad de normalizar los desarrollo. Uno de los Framework (entornos) más extendidos es Struts
Patrones de diseño J2EE	Os mostramos una interpretación particular de los patrones de diseño J2EE

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.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)



www.AdictosAlTrabajo.com Optimizado 800X600