

CONFIGURAR HTTPS EN TOMCAT

DESCRIPCIÓN

Explicación para configurar SSL en Tomcat y acceder desde <https://localhost:8443/AcmeBnB>

REQUISITOS

- Java SDK
- Tomcat 7.0

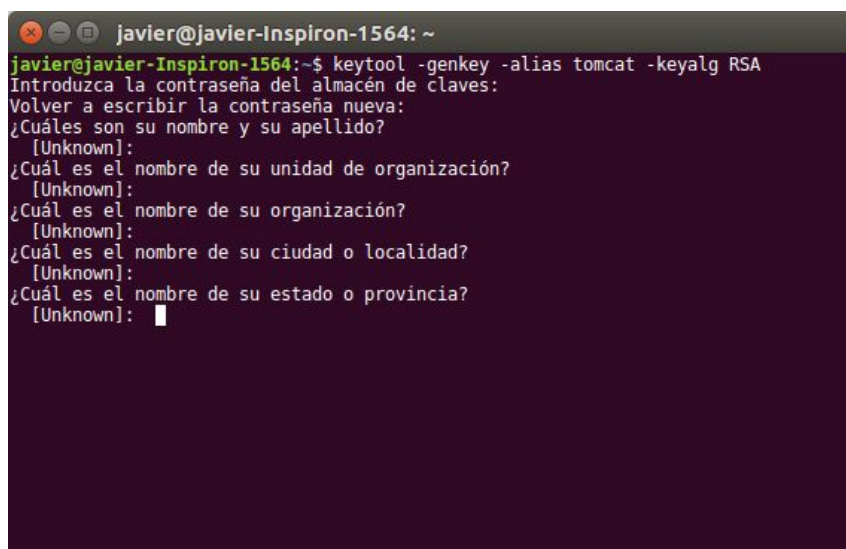
Creación de la Keystore

Crear una Keystore es obligatorio para configurar HTTPS en nuestro servidor. Podemos generarlo en cualquier sistema operativo, en este caso la explicación se basará en Linux.

- Abrimos una nueva ventana del terminal y escribimos:

```
keytool -genkey -alias tomcat -keyalg RSA
```

Nos mostrará por ventana una serie de mensajes que tendremos que ir rellenando con información. Únicamente será necesario rellenar la contraseña y recordarla, lo demás podremos dejarlo en blanco.



```
javier@javier-Inspiron-1564: ~  
javier@javier-Inspiron-1564:~$ keytool -genkey -alias tomcat -keyalg RSA  
Introduzca la contraseña del almacén de claves:  
Volver a escribir la contraseña nueva:  
¿Cuáles son su nombre y su apellido?  
[Unknown]:  
¿Cuál es el nombre de su unidad de organización?  
[Unknown]:  
¿Cuál es el nombre de su organización?  
[Unknown]:  
¿Cuál es el nombre de su ciudad o localidad?  
[Unknown]:  
¿Cuál es el nombre de su estado o provincia?  
[Unknown]:
```

Una vez completado y generada la llave, tendremos un nuevo fichero con extensión .keystore en el directorio donde se encontrase la ventana de comandos, por defecto la carpeta personal del usuario logueado.

Configurar SSL en nuestra instalación de Tomcat

Lo siguiente será incluir dicho certificado en nuestra instalación de Tomcat. Nos vamos para ello a nuestra máquina virtual de Windows, **accedemos como Administrador del sistema** y copiamos el certificado .keystore que hemos generado anteriormente en alguna carpeta accesible del sistema. Por comodidad es aconsejable incluirla dentro de los archivos de configuración de Tomcat: C:\Program Files\Apache Software Foundation\Tomcat 7.0\conf

Pasamos ahora a configurar Tomcat. Lo primero será hacer una copia del archivo server.xml para volver a nuestra configuración anterior en caso de romper la existente o querer simplemente volver al estado anterior. Posteriormente abrimos el archivo server.xml con Notepad o algún programa de edición de archivos.

Dentro del archivo observamos la siguiente línea:

```
<!--  
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"  
    maxThreads="150" scheme="https" secure="true"  
    clientAuth="false" sslProtocol="TLS" />  
-->
```

La sustituimos por la siguiente (incluidos los caracteres para comentarlo):

```
<Connector SSLEnabled="true" acceptCount="100" clientAuth="false"  
    disableUploadTimeout="true" enableLookups="false" maxThreads="25"  
    port="8443" keystoreFile="C:\Program Files\Apache Software Foundation\Tomcat  
7.0\conf\keystore" keystorePass="password"  
    protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https"  
    secure="true" sslProtocol="TLS" />
```

Observa el apartado de **keystoreFile** y **keystorePass**, ya que tendrás que modificarlo con sus respectivos valores de la localización del certificado y la contraseña que le hayamos asignado en su creación.

```
<Connector SSLEnabled="true" acceptCount="100" clientAuth="false"  
    disableUploadTimeout="true" enableLookups="false" maxThreads="25"  
    port="8443" keystoreFile="C:\Program Files\Apache Software Foundation\Tomcat 7.0\conf\dp.keystore" keystorePass="123456"  
    protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https"  
    secure="true" sslProtocol="TLS" />
```

Guardamos el archivo.

Forzar redirección a HTTPS

Para forzar que nuestro proyecto siempre funcione con SSL tendremos que crear una regla para redirigir las páginas HTTP a HTTPS. Para ello nos vamos al directorio de configuración de Tomcat (indicado anteriormente), crear una copia del archivo web.xml por seguridad y abrir el archivo web.xml con cualquier editor de texto.

Una vez abierto añadimos las siguientes líneas antes del cierre de la etiqueta <web-app>.

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>AcmeBnB</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

Configurando *url-pattern* a */** y *transport-guarantee* a *CONFIDENTIAL* obligaremos a Tomcat a redirigir todas las páginas HTTP a HTTPS.

Configurar servidor en Eclipse

Iniciamos eclipse con nuestro proyecto Spring. Ya que hemos editado la configuración de Tomcat, lo primero será borrar la configuración del servidor que tengamos en ese momento. Hacemos clic derecho sobre el servidor y pulsamos en Delete para eliminar su configuración. Asegúrate de marcar Delete unused server configuration(s) para evitar que queden archivos de configuración.

Una vez borrado creamos el servidor como normalmente y lo iniciamos. Ahora ya podrás acceder al proyecto desde <https://localhost:8443/AcmeBnB>

