5分钟内搞定 Tomcat 的 SSL 配置 热

红薯 发布于 2011/07/04 00:07　阅读 85K+　收藏 130　评论 26　补充话题

Tomcat　SSL　HTTPS　精华

本教程使用 JDK 6 和 Tomcat 7，其他版本类似。

基本步骤：

1. 使用 java 创建一个 **keystore** 文件
2. 配置 Tomcat 以使用该 keystore 文件
3. 测试
4. 配置应用以便使用 SSL， 例如 https://localhost:8443/yourApp

1. 创建 keystore 文件

执行 **keytool -genkey -alias tomcat -keyalg RSA** 结果如下

```
loiane:bin loiane$ keytool -genkey -alias tomcat -keyalg RSA
Enter keystore password:  password
Re-enter new password: password
What is your first and last name?
  [Unknown]:  Loiane Groner
What is the name of your organizational unit?
  [Unknown]:  home
What is the name of your organization?
  [Unknown]:  home
What is the name of your City or Locality?
  [Unknown]:  Sao Paulo
What is the name of your State or Province?
  [Unknown]:  SP
```

```
What is the two-letter country code for this unit?
  [Unknown]:  BR
Is CN=Loiane Groner, OU=home, O=home, L=Sao Paulo, ST=SP, C=BR correct?
  [no]:  yes

Enter key password for
    (RETURN if same as keystore password):  password
Re-enter new password: password
```

这样就在用户的主目录下创建了一个 .keystore 文件

2. 配置 Tomcat 以使用 keystore 文件

打开 server.xml 找到下面被注释的这段

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

干掉注释，并将内容改为

```
Connector SSLEnabled="true" acceptCount="100" clientAuth="false"
    disableUploadTimeout="true" enableLookups="false" maxThreads="25"
    port="8443" keystoreFile="/Users/loiane/.keystore" keystorePass="password"
    protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https"
    secure="true" sslProtocol="TLS" />
```

3. 测试

启动 Tomcat 并访问 **https://localhost:8443**. 你将看到 Tomcat 默认的首页。

需要注意的是，如果你访问默认的 8080 端口，还是有效的。

4. 配置应用使用 SSL

打开应用的 web.xml 文件，增加配置如下：

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

将 URL 映射设为 /*，这样你的整个应用都要求是 HTTPS 访问，而 transport-guarantee 标签设置为 CONFIDENTIAL 以便使应用支持 SSL。

如果你希望关闭 SSL，只需要将 CONFIDENTIAL 改为 NONE 即可。