

HTTPS 配置说明

1、生成根证书

1、查看根证书是否存在

---查看 jre 中所有信任的证书信息

```
keytool -list -keystore %JAVA_HOME%/jre/lib/security/cacerts -storepass changeit
```

---查看别名为 root 的证书信息

```
keytool -v -list -alias root -keystore %JAVA_HOME%/jre/lib/security/cacerts -storepass changeit
```

2、删除根证书

---删除 jre 中别名为 root 的证书

```
keytool -delete -alias root -keystore %JAVA_HOME%/jre/lib/security/cacerts -storepass changeit
```

2、生成根证书

---生成根证书

```
keytool -genkey -keyalg RSA -alias root -dname "CN=唐宁, OU=ce, O=creditease, L=北京, ST=北京, C=CN" -storepass changeit -keystore root.keystore -validity 180
```

---导出根证书

```
keytool -export -alias root -file root.crt -storepass changeit -keystore root.keystore
```

---导入根证书

```
keytool -import -alias root -file root.crt -keystore F:\jdk1.6.0_06\jre\lib\security\cacerts -storepass changeit
```

---查看是否导入根证书

```
keytool -v -list -alias root -keystore F:\jdk1.6.0_06\jre\lib\security\cacerts -storepass changeit
```

3、需要双向认证时用到

---生成客户端证书

```
keytool -genkey -keyalg RSA -alias client -storetype PKCS12 -dname "CN=测试, OU=ce, O=creditease, L=北京, ST=北京, C=CN" -storepass 123456 -keystore client.p12 -validity 180
```

---导出客户端证书

```
keytool -export -alias client -file client.cer -keystore client.p12 -storetype PKCS12 -storepass 123456
```

---导入客户端证书

```
keytool -import -alias client -file client.cer -keystore client.keystore -storepass 123456
```

2、配置 Tomcat

1、注释掉 8080 端口

```
<Connector executor="tomcatThreadPool"
```

```
port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
```

2、新增 8443 端口，并添加证书

```
<Connector SSLEnabled="true" clientAuth="false"
    keystoreFile="F:/jdk1.6.0_06/bin/root.keystore" keystorePass="changeit"
    maxSpareThreads="10" maxThreads="150" minSpareThreads="2" port="8443"
    protocol="HTTP/1.1" scheme="https" secure="true" sslProtocol="TLS"
    truststoreFile="F:/jdk1.6.0_06/jre/lib/security/cacerts" />
```

keystoreFile 指向密钥库文件 root.keystore

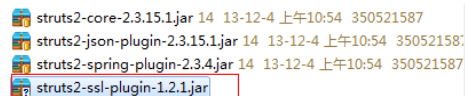
keystorePass 默认为 changeit

truststoreFile 指向 jre 信任的证书库文件

3、整合 struts2

利用 struts2-ssl-plugin 插件配置安全访问协议

1、添加 jar 包，struts 必须是 2.1.18 及以上版本的



struts2-core-2.3.15.1.jar 14 13-12-4 上午10:54 350521587
struts2-json-plugin-2.3.15.1.jar 14 13-12-4 上午10:54 350521587
struts2-spring-plugin-2.3.4.jar 14 13-12-4 上午10:54 350521587
struts2-ssl-plugin-1.2.1.jar

2、配置 struts.xml 文件，添加以下配置

```
<constant name="struts2.sslplugin.httpPort" value="8085"/>
<constant name="struts2.sslplugin.httpsPort" value="8443"/>
<package name="customer" extends="ssl-default" namespace="/app/customer">
    <action name="customerIndex" class="customerAction"
        method="customerIndex">
        <result name="success">/app/customer/listCustomer.jsp</result>
    </action>
</package>
```

注意：default 里可以继承这个 ssl-default，因为其实这个 ssl-default 也是继承 struts-default 的，之后在你需要用某个 SSL 的方法或类前，用注解 @Secured，就行了：整个类都用 HTTPS：

```
@Secured
public class CustomerAction extends EGridBaseAction {
    private CustomerService customerService;
    private Long customerId;
    private String customerName;
```

方法用 HTTPS：

```
@Secured
public String addCustomer() throws Exception
{
    Customer c = new Customer();
    c.setCustomerName(customerName);
    c.setCreateDate(createDate);
    c.setEndDate(endDate);
    c.setRealName(realName);
    //c.setCreator(creator); //当前登录用户的user_code
    if(customerId == null || customerId == 01){ //增加用户
        c.setPassword(MD5Util.md5(password));
    }
}
```

4、 重启 Tomcat

访问 <https://localhost:8443/web>