# 生成证书

## 手动生成weblogic需要配置的两个jks文件identity.jks和trust.jks

### 生成标识identity.jks密钥库

keytool -genkey -alias weblogicHL -keyalg RSA -keypass keypass123 –keystore identity.jks -storepass storepass123 -validity 3650

简单说明：

weblogicHL为“SSL—私有密钥别名”；

keypass123为“SSL—私有密钥密码短语”；

storepass123为“密钥—标识(identity)—定制标识密钥库密码短语”；

storepass123为“密钥库—信任(trust)—定制标识密钥库密码短语”；

3650为自定义证书的有效期，单位为天；

这条命令会要求输入一些值。自己输入就可以。

### 导出公钥证书的cer文件

keytool -export -alias weblogicHL -file root.cer -keystore identity.jks

需要输入密码

### 生成信任trust.jks密钥库

keytool -import -alias weblogicHL -trustcacerts -file root.cer -keystoretrust.jks

输入密码

### 根据cer生成.trustStore文件

keytool -import -file root.cer -storepass fs1358319 -keystore trust.trustStore

## weblogic配置过程

1. 生成结束后，需要把上面生成的identity.jks和trust.jks拷贝到base\_domain根目录下面。否则下面配置的时候就需要写绝对路径了。



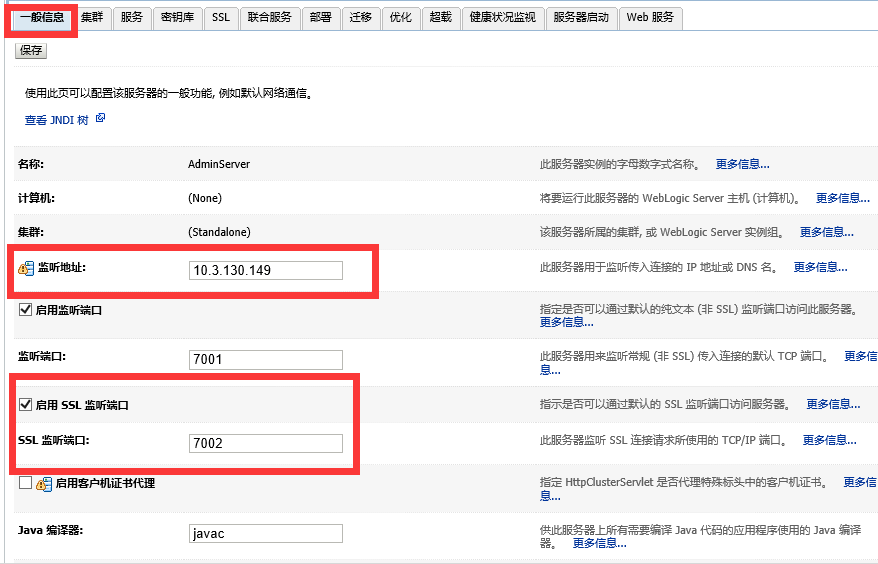
1. 配置密钥库。密钥库需要修改为“定制标识和定制信任”。定制标识密钥库和定制信任密钥库都是上面生成的。



1. 配置SSL。私有密钥的别名等，就是上面生成时设置的。

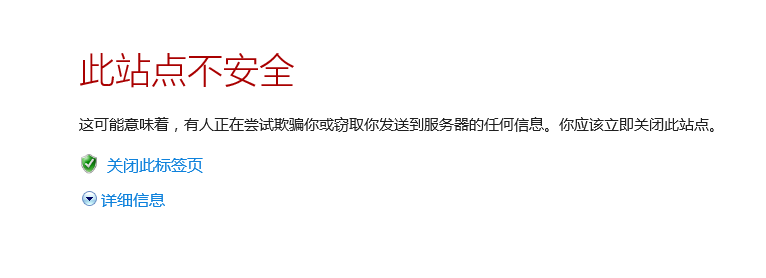


1. 设置SSL监听。监听地址自己试一下吧，需不需要设置。



### 浏览器报警以及非IE不能访问的方法

按照上面这几步，配置结束后，IE就可以正常访问了。一开始会显示为：



点击详细信息，就可以访问了。

此时可以在IE中导入相应的证书，也就是我们上面生成的root.cer。此时再访问，其实还是提示上面的画面。那时因为我们生成的证书是针对域名的，而我们一般是用IP地址访问。所以我们需要在hosts文件里，配置一下域名和IP的映射，然后再访问就可以了。

域名就是我们在第一步生成identity.jks时输入的那个名字（具体哪个我忘记了，自己试试吧）

## 配置禁用HTTP

### weblogic

### 应用

应用的配置，就在自己的项目下，知道到web.xml,在web-app之间添加：

<!-- 强制SSL配置，即普通的请求也会重定向为SSL请求 -->

<security-constraint>

<web-resource-collection>

<web-resource-name>SSL</web-resource-name>

<!-- <url-pattern>/servicesMs/\*</url-pattern> --><!-- 全站使用SSL <url-pattern>/\*</url-pattern>-->

<url-pattern>/\*</url-pattern>

</web-resource-collection>

<user-data-constraint>

<description>SSL required</description>

<!-- CONFIDENTIAL: 要保证服务器和客户端之间传输的数据不能够被修改，且不能被第三方查看到 -->

<!-- INTEGRAL: 要保证服务器和client之间传输的数据不能够被修改 -->

<!-- NONE: 指示容器必须能够在任一的连接上提供数据。（即用HTTP或HTTPS，由客户端来决定）-->

<transport-guarantee>CONFIDENTIAL</transport-guarantee>

</user-data-constraint>

</security-constraint>

## SSL验证的WebService接口的Java调用方法

调用HTTPS接口前，设置如下两句：

// trust.trustStore是上面生成的。fs1358319是对应的密码

System.setProperty("javax.net.ssl.trustStore", "trust.trustStore");

System.setProperty("javax.net.ssl.trustStorePassword", "fs1358319");