# 极速开发框架SpringBoot及源码分析

主讲：Cat 老师

**北京动力节点教育科技有限公司**  
**2020 • 北京**

**动力节点版权所有，禁止私自传播**

## SpringBoot配置SSL（https）

SpringBoot可以通过在application.properties或application.yml配置文件中配置各种server.ssl.\*属性来声明性使用SSL（https），比如下面的例子在application.properties中设置SSL属性：

*#配置https的端口*server.port=8443  
*#配置https的证书*server.ssl.key-store=classpath:tomcat.pkcs12  
*#证书的密码（在生成证书的时候会指定一个密码）*server.ssl.key-store-password=123456  
*#执行证书的类型*server.ssl.key-store-type=pkcs12  
*#server.ssl.key-store-type=jks*

如果使用了上面的配置就表示springboot应用程序不再在端口8080上支持HTTP连接请求，SpringBoot不能通过配置application.properties来实现既支持HTTP连接又支持HTTPS连接，这是做不到的，如果要同时支持HTTP和HTTPS，则需要以编程方式配置其中的一个，建议使用application.properties文件来配置HTTPS，以编程方式配置HTTP，这是比较容易的方法；

**SpringBoot支持配置https具体步骤：**

1. 生成证书，可以使自签名证书（平时测试的时候）或者从SSL证书授权中心购买证书（上线）；

平时生成证书进行测试的话，有两种生成证书的方式：

（1）利用JDK工具生成证书

keytool是jdk的工具，安装好jdk可以找到，用于生成jks专有格式

keytool -genkey -keyalg RSA -keystore tomcat.jks

转换为pkcs12行业标准格式：（生成这个格式后java才识别）

keytool -importkeystore -srckeystore tomcat.jks -destkeystore tomcat.pkcs12 -deststoretype pkcs12

证书生成后我们可以验证下jks是否包含了完整的证书链：

keytool -list -v -keystore server.jks

keytool -list -v -keystore server.pkcs12

（2）利用Openssl工具生成证书

通过openssl来生成，如果linux中没有安装openssl，需要安装一下，执行：

yum install openssl openssl-devel -y

1、#生成一个RSA密钥 （私钥）

openssl genrsa -out server.key 2048

2、#生成一个证书请求

openssl req -new -key server.key -out server.csr -subj "/C=CN/ST=Beijing/L=Beijing/O=power Inc./OU=Web Security/CN=power.com"

3、#自己签发证书

openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

**字段解读**

C字段：Country，单位所在国家，为两位数的国家缩写，如：CN 表示中国；

ST 字段：State/Province，单位所在州或省；

L 字段：Locality，单位所在城市/或县区；

O 字段：Organization，此网站的单位名称；

OU 字段：Organization Unit，下属部门名称，也常常用于显示其他证书相关信息，如证书类型，证书产品名称或身份验证类型或验证内容等；

CN 字段：Common Name，网站的域名；

转换为pkcs12格式（因为在Java中使用证书，需要转换一下格式）

openssl pkcs12 -export -clcerts -in **server.crt** -inkey server.key -out server.p12

openssl pkcs12 -export -clcerts -in **server.crt** -inkey server.key -out server.pkcs12

转换为jks格式（因为在Java中使用证书，需要转换一下格式，jks是java独有的）

keytool -importkeystore -srckeystore server.pkcs12 -destkeystore server.jks -srcstoretype pkcs12 -deststoretype jks

1. 配置或编写代码；

注意我们是在Java环境中，常用的证书形式有p12、pkcs12格式、jks格式，如果不是该格式，需要转换；p12、pkcs12是同一个，只是证书的后缀不同而已；

作业：单独有一个Tomcat，配置一下https；

## SpringBoot全局异常处理

在项目的开发中，如果报错了，直接在页面显示500错误，输出一大堆的异常信息，这对应用户来说体验不友好，所以在企业里面对这些异常一般都会统一捕获，由一个专门的异常处理类来统一处理。（原来在ssm框架中，我们是在tomcat定义错误页或者web.xml定义错误页）

具体参考课程演示和代码；

## SpringBoot 404页面处理

当输入地址有误，会进入springboot默认的白板404页面，对用户不太友好，我们可以统一定义一个全局的404错误处理；

@Bean  
**public** WebServerFactoryCustomizer<ConfigurableWebServerFactory> webServerFactoryCustomizer() {  
 **return** (factory->{  
 ErrorPage error404Page = **new** ErrorPage(HttpStatus.***NOT\_FOUND***, **"/404.html"**);  
 factory.addErrorPages(error404Page);  
 });  
}

具体参考课程演示和代码；

SpringBoot更多话题，同学们可以学习一下：

SpringBoot restTemplate

SpringBoot webClient

Validation 验证（springmvc 303验证） @Vilidate 注解验证

SpringBoot 多数据源  
SpringBoot 服务监控 springboot actuator、SpringBoot admin；

springboot整合thymeleaf

Springboot统一缓存处理

SpringBoot 定时任务quartz