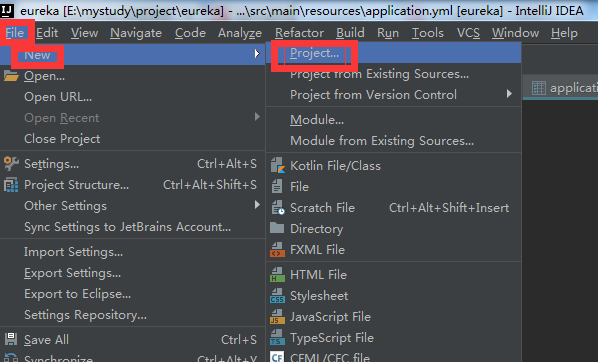
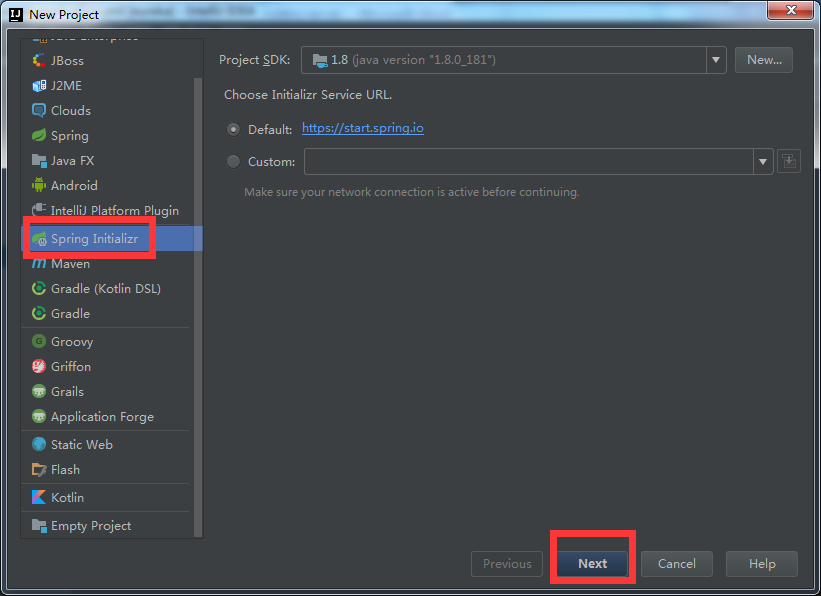
**Eureka Server 服务配置**

## 创建Spring Cloud项目（idea）

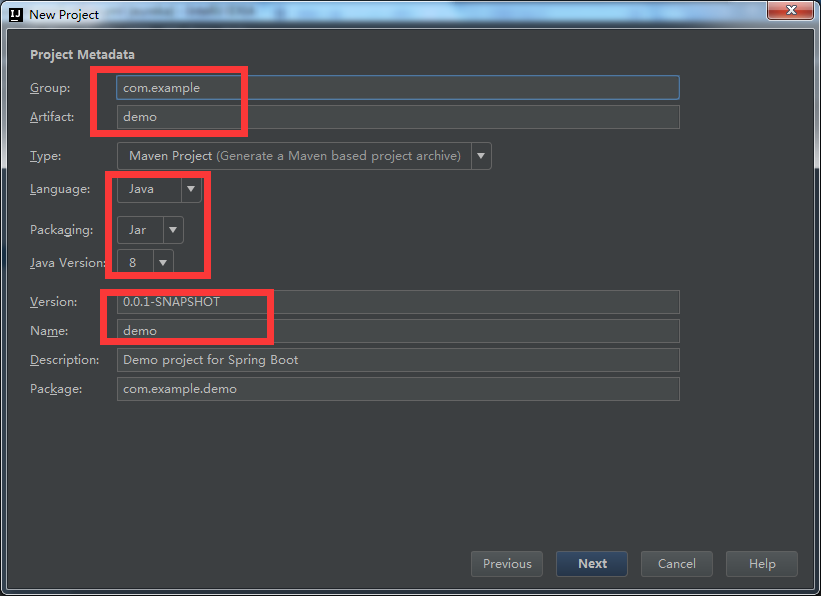
1. 点击创建



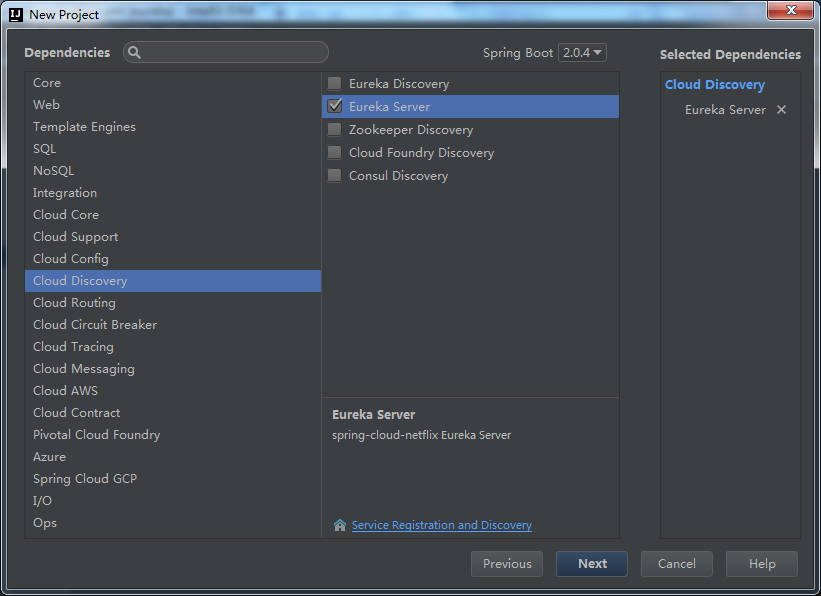
1. 选择模板



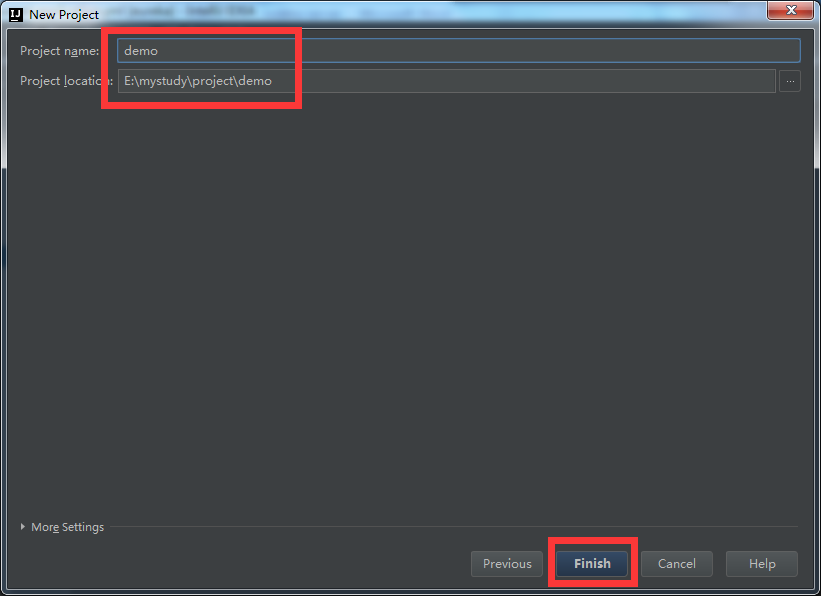
1. 补充基本信息



1. 选择服务模板



1. 完成



## 代码相关

1. pom文件

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>   <groupId>com.sys</groupId>  <artifactId>eureka</artifactId>  <version>0.0.1-SNAPSHOT</version>  <packaging>jar</packaging>   <name>eureka</name>  <description>Demo project for Spring Boot</description>   <parent>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-parent</artifactId>  <version>2.0.4.RELEASE</version>  <relativePath/> <!-- lookup parent from repository -->  </parent>   <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  <java.version>1.8</java.version>  <spring-cloud.version>Finchley.SR1</spring-cloud.version>  </properties>   <dependencies>  <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>  </dependency>   <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-test</artifactId>  <scope>test</scope>  </dependency>  </dependencies>   <dependencyManagement>  <dependencies>  <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-dependencies</artifactId>  <version>${spring-cloud.version}</version>  <type>pom</type>  <scope>import</scope>  </dependency>  </dependencies>  </dependencyManagement>   <build>  <plugins>  <plugin>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-maven-plugin</artifactId>  </plugin>  </plugins>  </build>   </project> |

1. 配置文件

|  |
| --- |
| **server:  port:** 8761  **spring:  application:  name:** eureka  **eureka:  instance:  hostname:** localhost  **server:  service-url:** http://${**eureka.instance.hostname**}:${**server.port**}/eureka  **client:  register-with-eureka: false  fetch-registry: false** |

1. 启动类

|  |
| --- |
| package com.sys.eureka;  import org.springframework.boot.SpringApplication; import org.springframework.boot.autoconfigure.SpringBootApplication; import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;  @SpringBootApplication @EnableEurekaServer public class EurekaApplication {   public static void main(String[] args) {  SpringApplication.*run*(EurekaApplication.class, args);  } } |

## 用户认证配置（Spring Security）

1. 修改配置文件：

|  |
| --- |
| **security:  basic:  enabled: true  user:  name:** ldvv  **password:** ldvv |

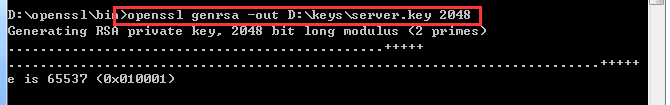
1. 引入相关依赖：

|  |
| --- |
| <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-security</artifactId> </dependency> |

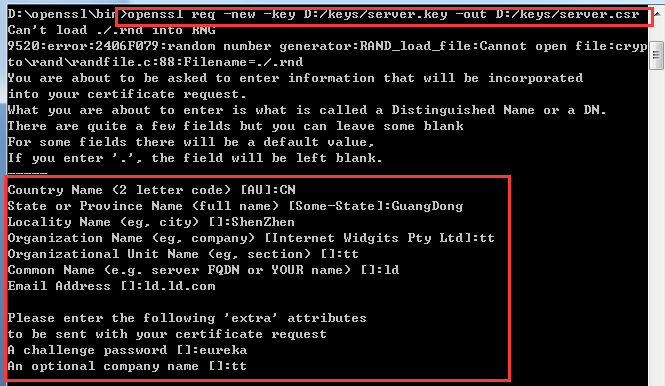
## SSL配置

1. 生成相关证书文件

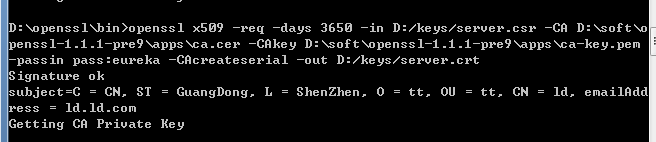
a. openssl genrsa -out D:\keys\server.key 2048



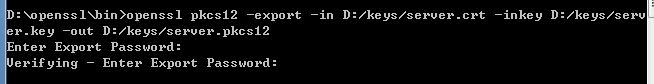
b. openssl req -new -key D:/keys/server.key -out D:/keys/server.csr



c. openssl x509 -req -days 3650 -in D:/keys/server.csr -CA D:\soft\openssl-1.1.1-pre9\apps\ca.cer -CAkey D:\soft\openssl-1.1.1-pre9\apps\ca-key.pem -passin pass:eureka -CAcreateserial -out D:/keys/server.crt



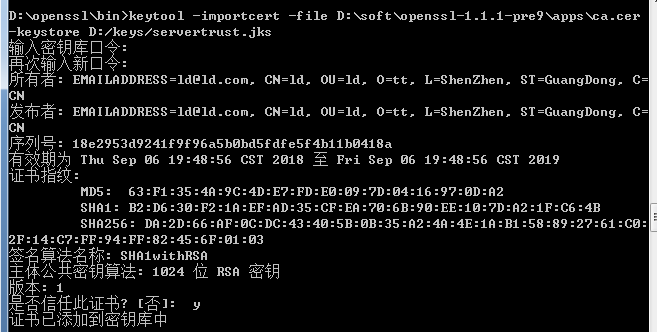
d. openssl pkcs12 -export -in D:/keys/server.crt -inkey D:/keys/server.key -out D:/keys/server.pkcs12

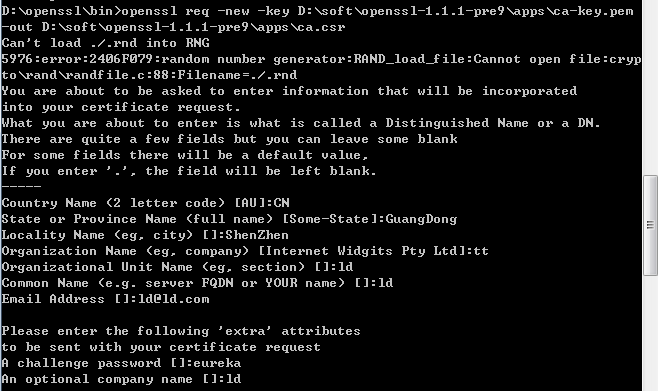


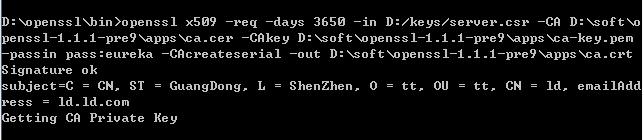
e. keytool -importkeystore -srckeystore D:/keys/server.pkcs12 -destkeystore D:/keys/server.jks -srcstoretype pkcs12



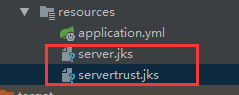
f. keytool -importcert -file D:\soft\openssl-1.1.1-pre9\apps\ca.cer -keystore D:/keys/servertrust.jks



ps: 生成根证书：



1. 拷贝证书到项目中



1. 修改配置文件：

|  |
| --- |
| **server:  port:** 8761  **ssl:  enabled: true  client-auth:** *want* **key-store:** classpath:server.jks  **key-store-password:** eureka  **trust-store:** classpath:servertrust.jks  **trust-store-password:** eureka |

## 完成

