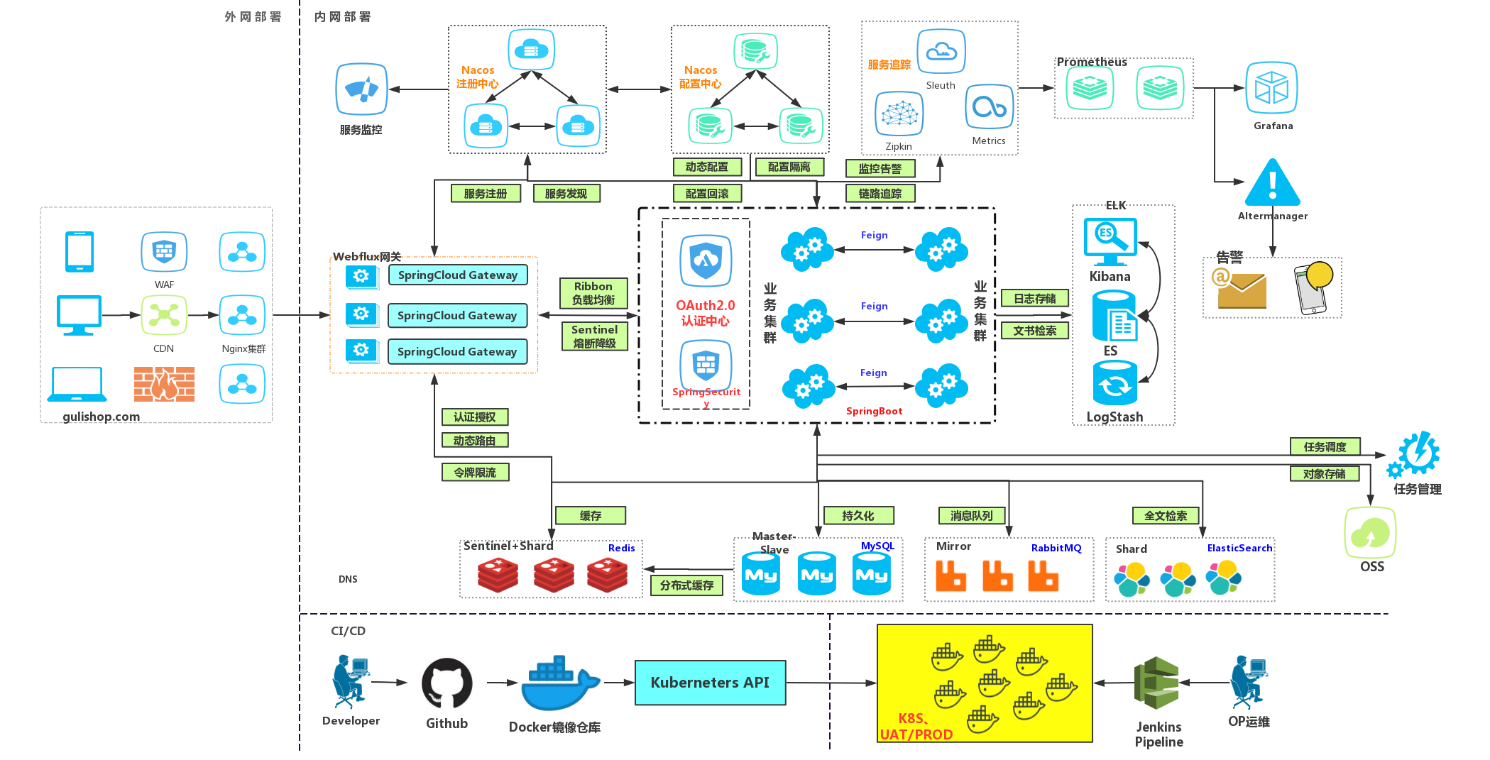
## 微服务（https://martinfowler.com）

微服务架构是一种架构模式，它提倡将单一应用程序划分成一组小的服务，服务之间相互协调、相互配合，为用户提供最终价值。每个服务运行在其独立的进程中，服务与服务间采用轻量级的通信机制互相沟通（通常是基于HTTP的RESTful API）。每个服务都围绕着具体业务进行构建，并且能够被独立地部署到生产环境、类生产环境等。另外，应尽量避免统一的、集中式的服务管理机制，对具体的一个服务而言，应根据业务上下文，选择合适的语言、工具对其进行构建。

微服务架构案例（可保存后，放大图片浏览）



微服务架构

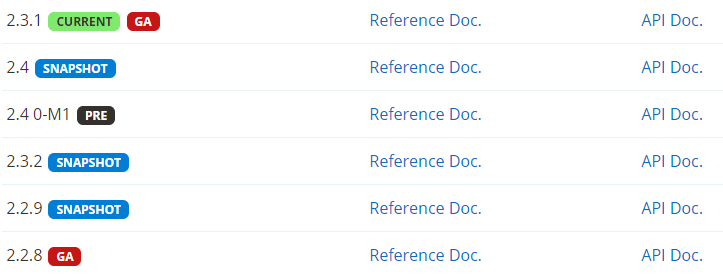
<https://microservices.io/>

为什么大家都在谈微服务？

<https://yq.aliyun.com/articles/175350?spm=a2c4e.11153959.0.0.1f5f389bvyxq4F>

## SpringBoot

1. springboot 版本号解释说明



* Release: 最终版本,Release不会以单词形式出现在软件封面上，取而代之的是符号(R)
* GA: General Availability,正式版本，官方推荐使用此版本,在国外都是用GA来说明release版本的
* RC: 该版本已经相当成熟了，基本上不存在导致错误的BUG，与即将发行的正式版相差无几。
* M: 又叫里程碑版本，表示该版本较之前版本有功能上的重大更新。
* SNAPSHOT(优先使用Lv2): 快照版，可以稳定使用，且仍在继续改进版本。

1. Official doc

<https://docs.spring.io/spring-boot/docs/2.3.1.RELEASE/reference/html/>

1. spring-boot/wiki

<https://github.com/spring-projects/spring-boot/wiki>

1. spring-boot/starter

<https://docs.spring.io/spring-boot/docs/2.3.1.RELEASE/reference/html/using-spring-boot.html#using-boot-starter>

1. spring-boot/knowledge-sharing

<http://www.ityouknow.com/spring-boot.html>

## SpringCloud

<https://spring.io/projects/spring-cloud> 

### Spring Cloud版本

#### 3.1.1 version命名说明

这些版本名称的命名方式采用了伦敦地铁站的名称，同时根据字母表的顺序来对应版本时间顺序，比如：最早的Release版本：Angel，第二个Release版本：Brixton，然后是Camden、Dalston、Edgware、Finchley、Greenwich目前最新的是Hoxton版本。

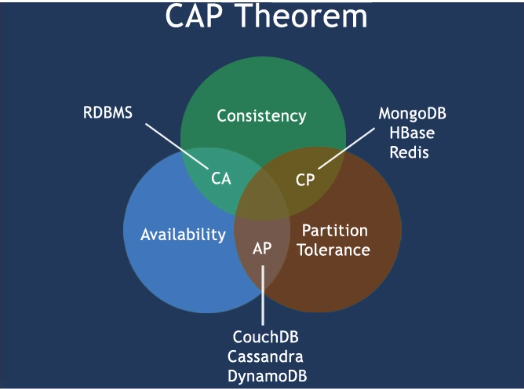
当一个版本的Spring Cloud项目的发布内容积累到临界点或者解决了一个严重bug后，就会发布一个“service releases”版本，简称SRX版本，其中X是一个递增数字。当前官网上最新的稳定版本是Edgware.SR3，里程碑版本是Finchley.M9。下表列出了这两个版本所包含的子项目及各子项目的版本号。

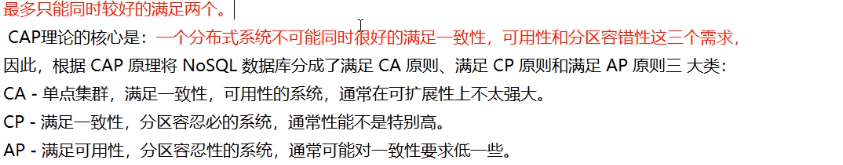
#### 3.1.2 Spring各组件的版本匹配

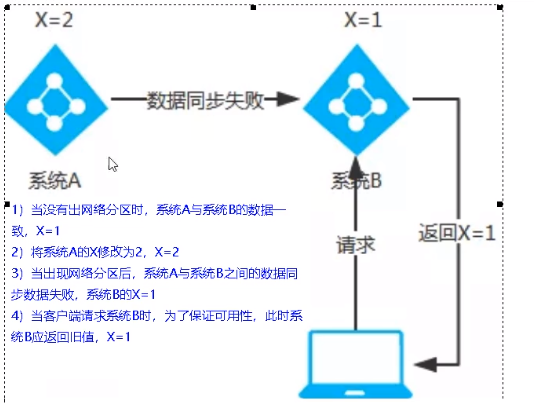
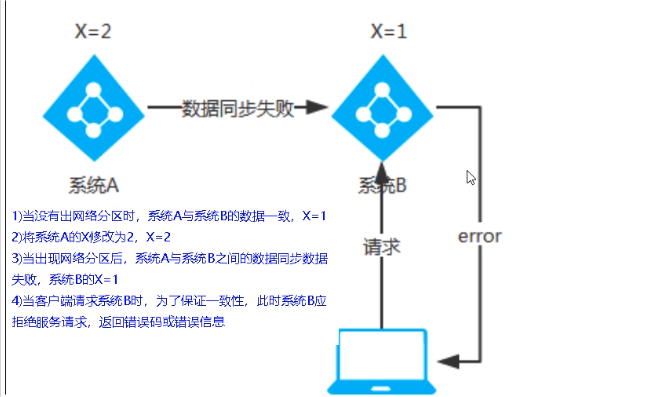
<https://start.spring.io/actuator/info>

<https://cloud.spring.io/spring-cloud-static/Hoxton.SR5/reference/html/>

### Service Registry Center





AP(Eureka)CP(ZP/Consul)

#### 3.2.1 Eureka

What is Eureka?

<https://cloud.spring.io/spring-cloud-static/spring-cloud-netflix/2.2.3.RELEASE/reference/html/#service-discovery-eureka-clients>

<https://github.com/Netflix/eureka/wiki>

How to use Eureka?

<http://www.ityouknow.com/springcloud/2017/05/10/springcloud-eureka.html>

#### 3.2.2 Zookeeper

What is Zookeeper?

How to use Zookeeper?

#### 3.2.3 Consul

What is Consul ?

How to use Consul ?

#### 3.2.4 Nacos

What is Nacos ?

How to use Nacos ?

### 3.3 Service Invocation

#### 3.3.1 Ribbon

What is Ribbon ?

<https://cloud.spring.io/spring-cloud-static/spring-cloud-netflix/2.2.3.RELEASE/reference/html/#spring-cloud-ribbon>

<https://github.com/Netflix/ribbon>

How to use Ribbon ?

#### 3.3.2 OpenFeign

What is OpenFeign ?

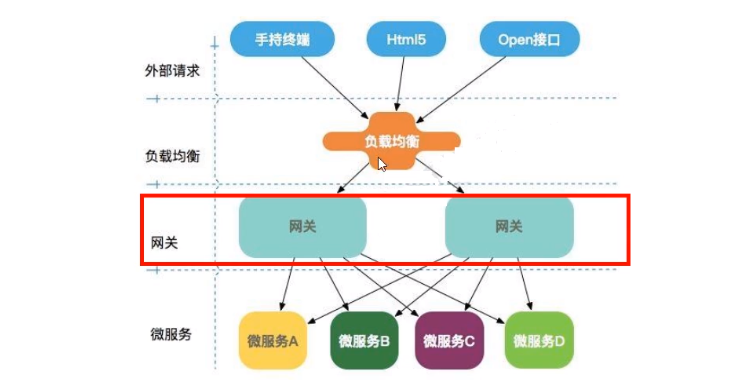
<https://cloud.spring.io/spring-cloud-static/spring-cloud-openfeign/2.2.3.RELEASE/reference/html/>

<https://github.com/OpenFeign/feign>

How to use OpenFeign ?

<http://www.ityouknow.com/springcloud/2017/05/12/eureka-provider-constomer.html>

### 3.4 Service Gateway



#### 3.4.1 Zuul

What is Zuul ?

<https://cloud.spring.io/spring-cloud-static/spring-cloud-netflix/2.2.3.RELEASE/reference/html/#router-and-filter-zuul>

<https://github.com/Netflix/zuul>

How to use Zuul ?

<http://www.ityouknow.com/springcloud/2017/06/01/gateway-service-zuul.html>

<http://www.ityouknow.com/springcloud/2018/01/20/spring-cloud-zuul.html>

#### 3.4.2 Spring Gateway

What is Spring Gateway ?

<https://cloud.spring.io/spring-cloud-static/spring-cloud-gateway/2.2.3.RELEASE/reference/html/>

<https://github.com/spring-cloud/spring-cloud-gateway>

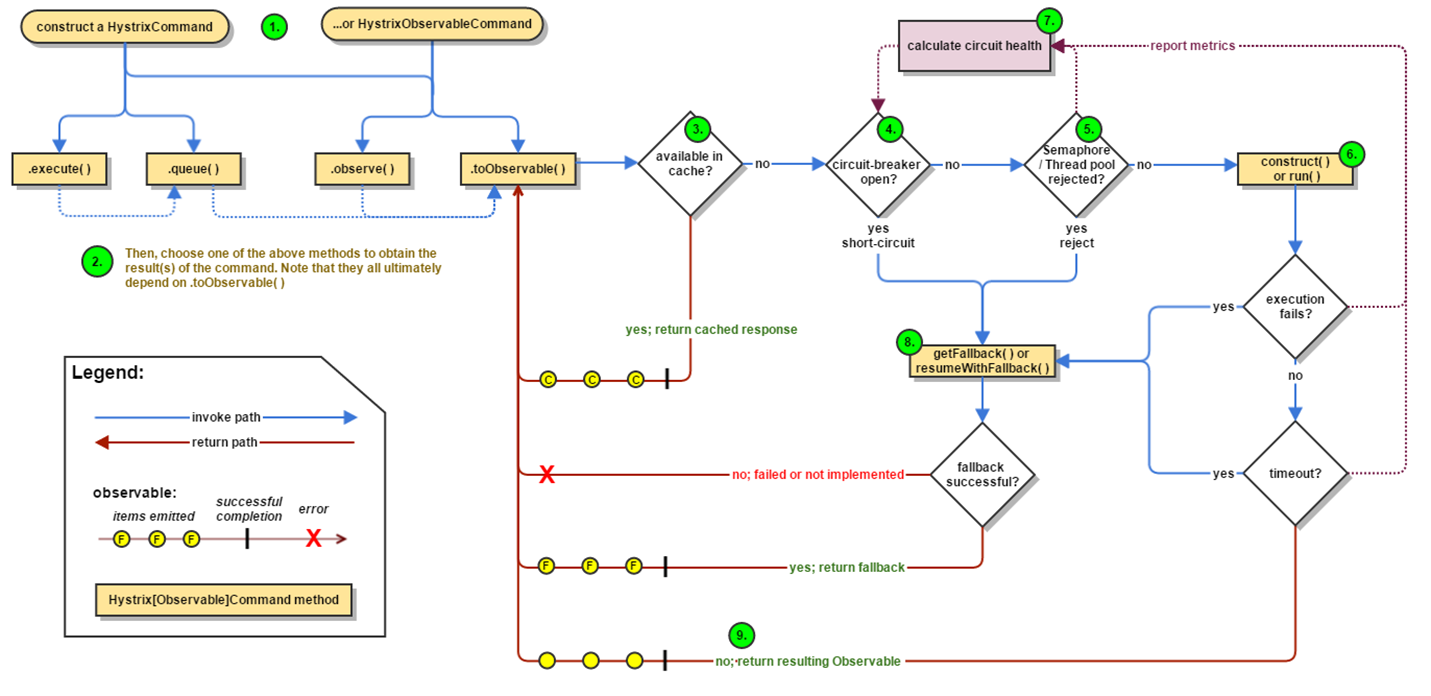
How to use Spring Gateway ?

<http://www.ityouknow.com/springcloud/2018/12/12/spring-cloud-gateway-start.html>

<http://www.ityouknow.com/springcloud/2019/01/19/spring-cloud-gateway-service.html>

http://www.ityouknow.com/springcloud/2019/01/26/spring-cloud-gateway-limit.html

### 3.5 Service Circuit Breaker



Hystrix工作流程分析 <https://www.cnblogs.com/holly8/p/11226177.html>

#### 3.5.1 Hystrix

What is Hystrix ?

<https://cloud.spring.io/spring-cloud-static/spring-cloud-netflix/2.2.3.RELEASE/reference/html/#circuit-breaker-hystrix-clients>

<https://github.com/Netflix/Hystrix/wiki>

<https://github.com/Netflix-Skunkworks/hystrix-dashboard/wiki>

How to use Hystrix ?

<https://github.com/Netflix/Hystrix/wiki>

<http://www.ityouknow.com/springcloud/2017/05/18/hystrix-dashboard-turbine.html>

#### 3.5.2 Sentinel

What is Sentinel ?

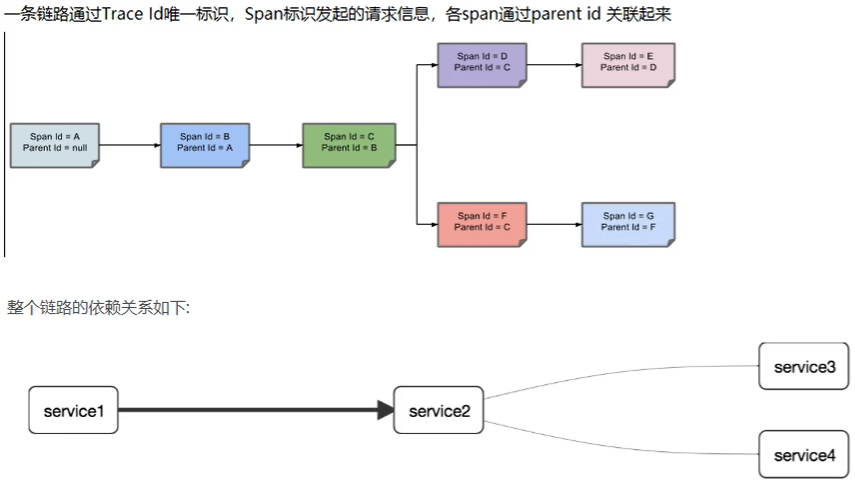
<https://spring-cloud-alibaba-group.github.io/github-pages/greenwich/spring-cloud-alibaba.html#_spring_cloud_alibaba_sentinel>

<https://github.com/alibaba/Sentinel/wiki>

How to use Sentinel ?

<https://github.com/alibaba/Sentinel/wiki>

### 3.6 Service Tracing



#### 3.6.1 Sleuth & Zipkin

下载Zipkin : https://dl.bintray.com/openzipkin/maven/io/zipkin/java/zipkin-server/

What is Sleuth ?

<https://cloud.spring.io/spring-cloud-static/spring-cloud-sleuth/2.2.3.RELEASE/reference/html/>

<https://github.com/spring-cloud/spring-cloud-sleuth/wiki>

What is Zipkin ?

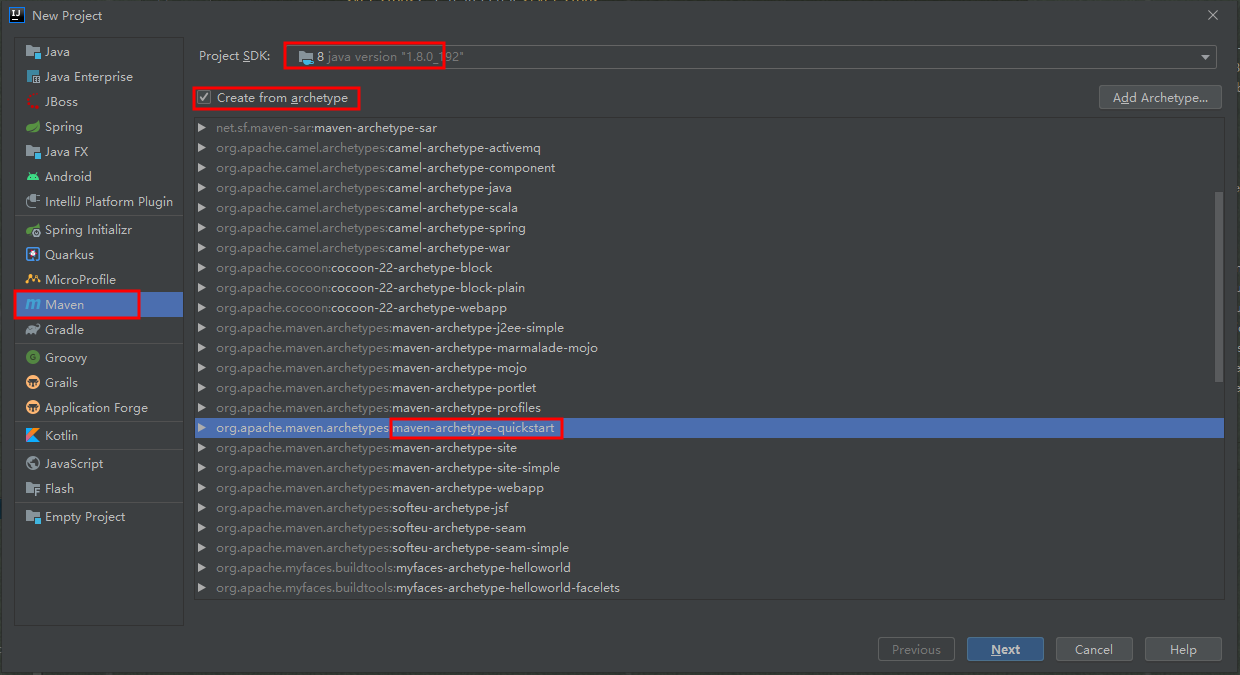
<https://zipkin.io/>

How to use Sleuth & Zipkin ?

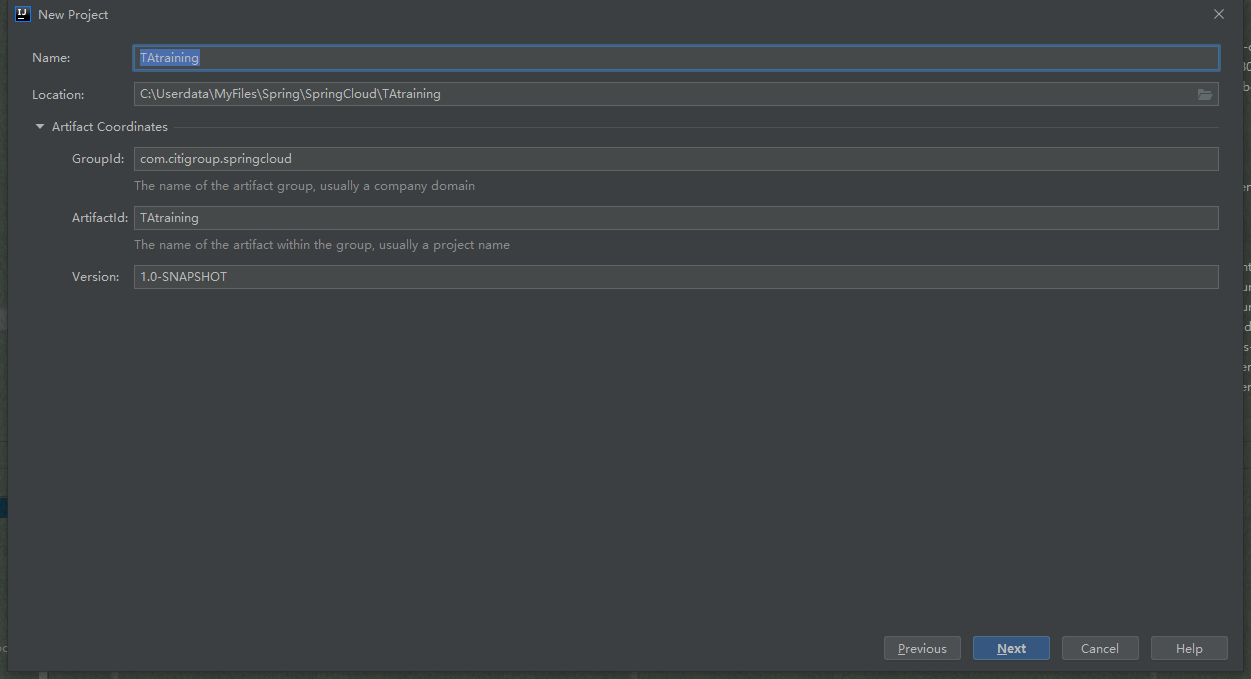
<http://www.ityouknow.com/springcloud/2018/02/02/spring-cloud-sleuth-zipkin.html>

## IDEA新建project工作空间

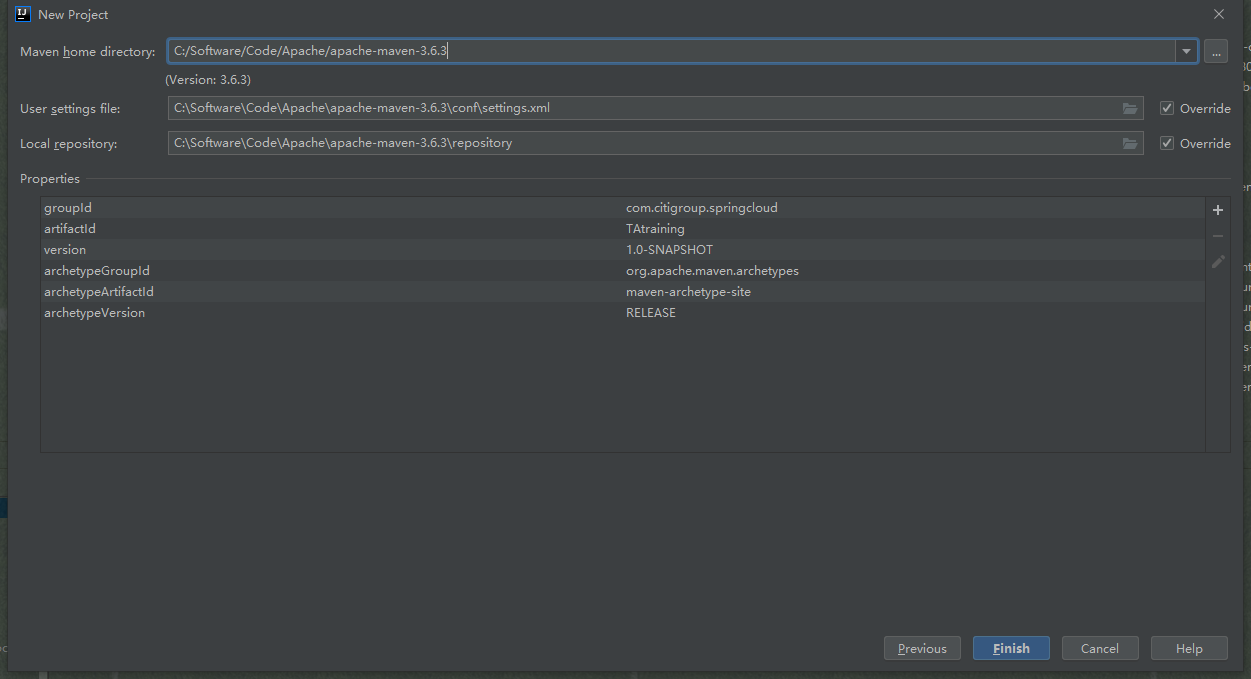
### New Project



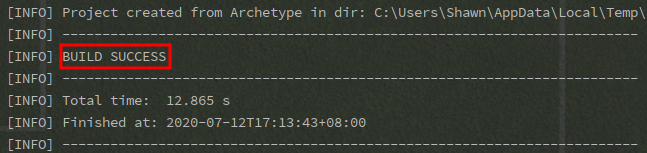
### 聚合总父工程名字



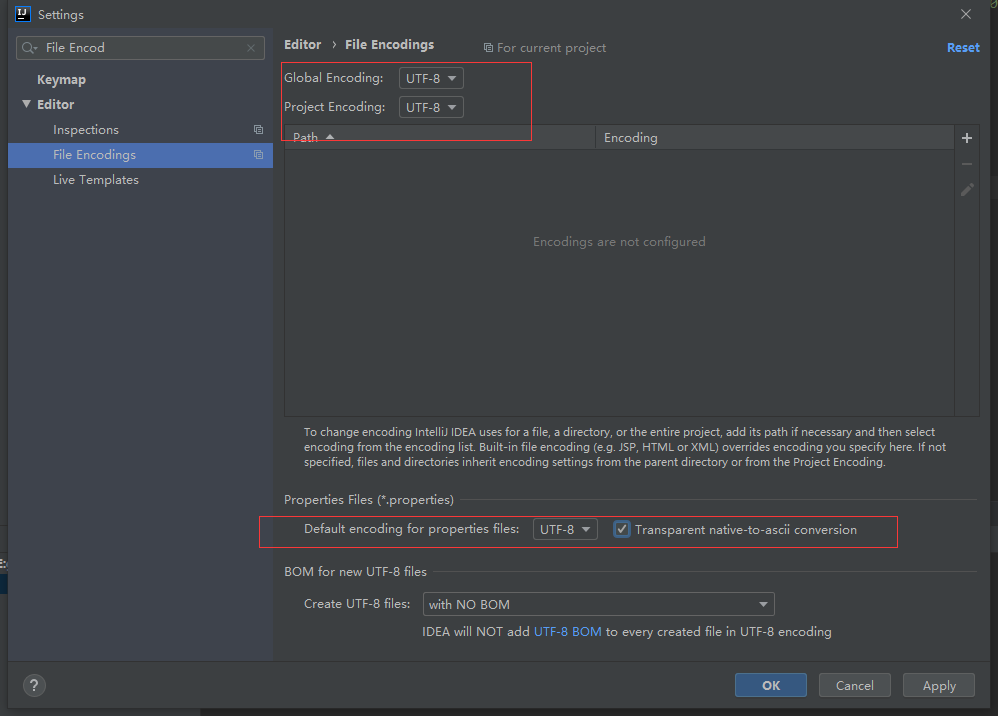
### Maven选版本



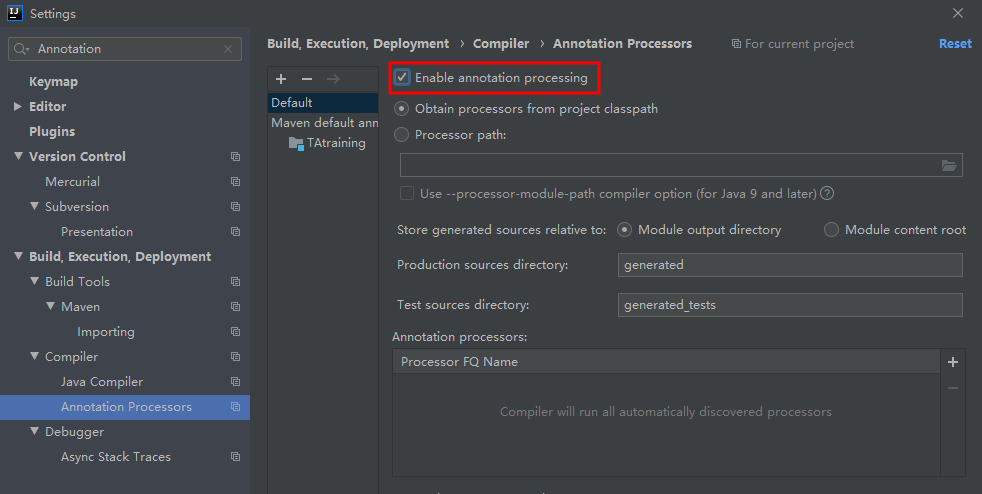
### 构建成功



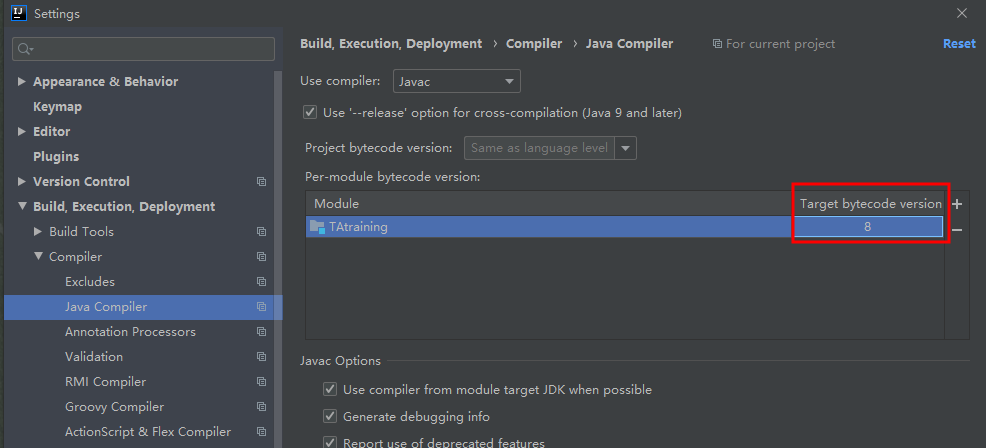
### 字符编码



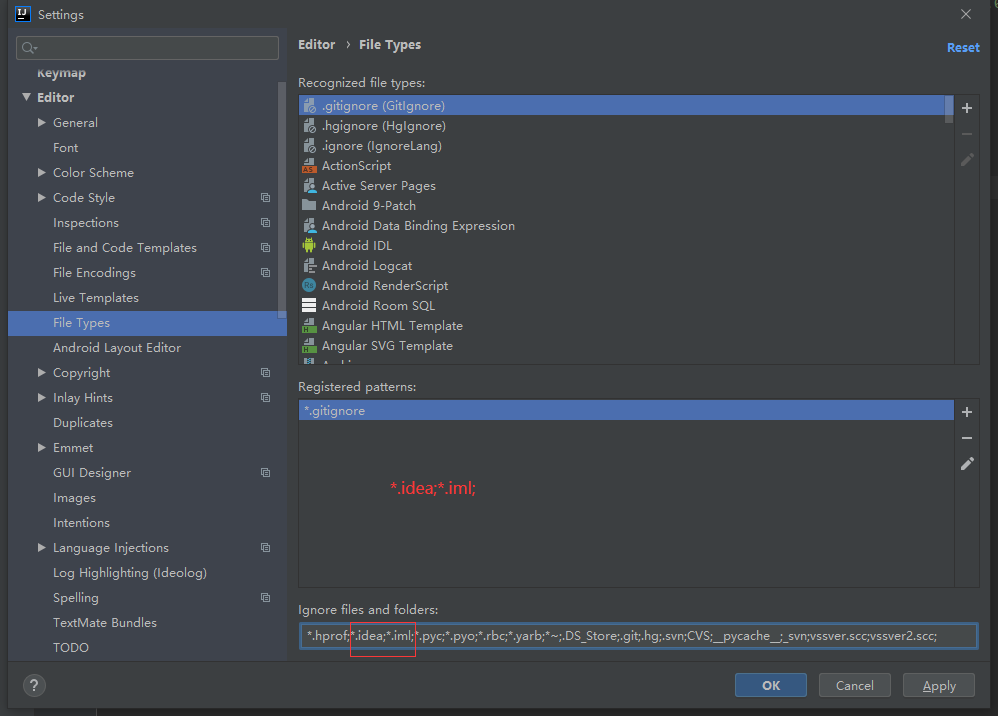
### 注解生效激活



### Java编译版本选8

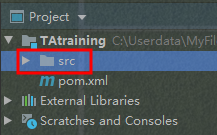


### 文件过滤（可选）



## 父工程POM和子模块Module

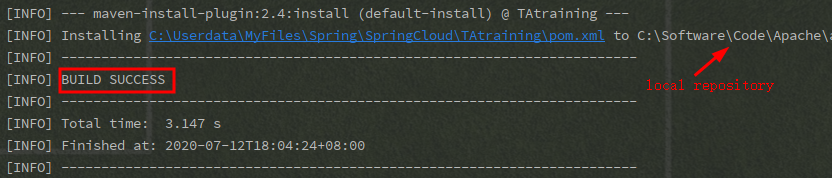
### 删除project的src目录



### 修改pom.xml

<?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.citigroup.springcloud</groupId>  
 <artifactId>TAtraining</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <packaging>pom</packaging>  
  
 <name>TAtraining</name>  
 <!-- *FIXME change it to the project's website* -->  
 <url>http://www.example.com</url>  
  
 <!--统一管理jar包版本-->  
 <properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 <maven.compiler.source>1.8</maven.compiler.source>  
 <maven.compiler.target>1.8</maven.compiler.target>  
 <junit.version>4.12</junit.version>  
 <log4j.version>1.2.17</log4j.version>  
 <lombok.version>1.16.18</lombok.version>  
 <spring.boot.version>2.2.2.RELEASE</spring.boot.version>  
 <spring.cloud.version>Hoxton.SR1</spring.cloud.version>  
 <spring.cloud.alibaba.version>2.1.0.RELEASE</spring.cloud.alibaba.version>  
 </properties>  
  
 <!--子模块继承后,提供作用:锁定版本+子module不用groupId和version-->  
 <dependencyManagement>  
 <dependencies>  
 <!--springboot 2.2.2-->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-dependencies</artifactId>  
 <version>${spring.boot.version}</version>  
 <type>pom</type>  
 <scope>import</scope>  
 </dependency>  
 <!--Spring cloud Hoxton.SR1-->  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-dependencies</artifactId>  
 <version>${spring.cloud.version}</version>  
 <type>pom</type>  
 <scope>import</scope>  
 </dependency>  
 <!--Spring cloud alibaba 2.1.0.RELEASE-->  
 <dependency>  
 <groupId>com.alibaba.cloud</groupId>  
 <artifactId>spring-cloud-alibaba-dependencies</artifactId>  
 <version>${spring.cloud.alibaba.version}</version>  
 <type>pom</type>  
 <scope>import</scope>  
 </dependency>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>${junit.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>log4j</groupId>  
 <artifactId>log4j</artifactId>  
 <version>${log4j.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 </dependency>  
 </dependencies>  
 </dependencyManagement>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 <configuration>  
 <fork>true</fork>  
 <addResources>true</addResources>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

### 父工程创建完成执行mvn:insall将父工程发布到仓库方便子工程继承



### 5步创建子模块

建module，改POM，写YML，主启动，业务类

