Spring Boot与Spring Cloud的对应关系

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

更详细的版本对应关系

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

<https://tool.lu/json>

Spring Cloud官方文档

<https://cloud.spring.io/spring-cloud-static/Hoxton.SR1/reference/htmlsingle/>

Spring Cloud翻译文档

<https://www.bilibili.com/video/BV18E411x7eT?p=4>

Spring Boot官方文档

<https://docs.spring.io/spring-boot/docs/2.2.2.RELEASE/reference/htmlsingle/>

1. 服务注册中心：1、Eureka（已停更）

2、Zookeeper（不常用）

3、Consul（Go语言）

4、Nacos（阿里巴巴）

二、服务调用： 1、Ribbon

2、LoadBalancer（技术未成熟）

三、服务调用： 1、Feign（已停更）

2、OpenFeign

四、服务降级 1、Hystrix（已停更）

2、resilience4j（国外常用）

3、sentienl（阿里巴巴）

五、服务网关 1、Zuul（已停更）

2、Zuul2（胎死腹中）

3、gateway

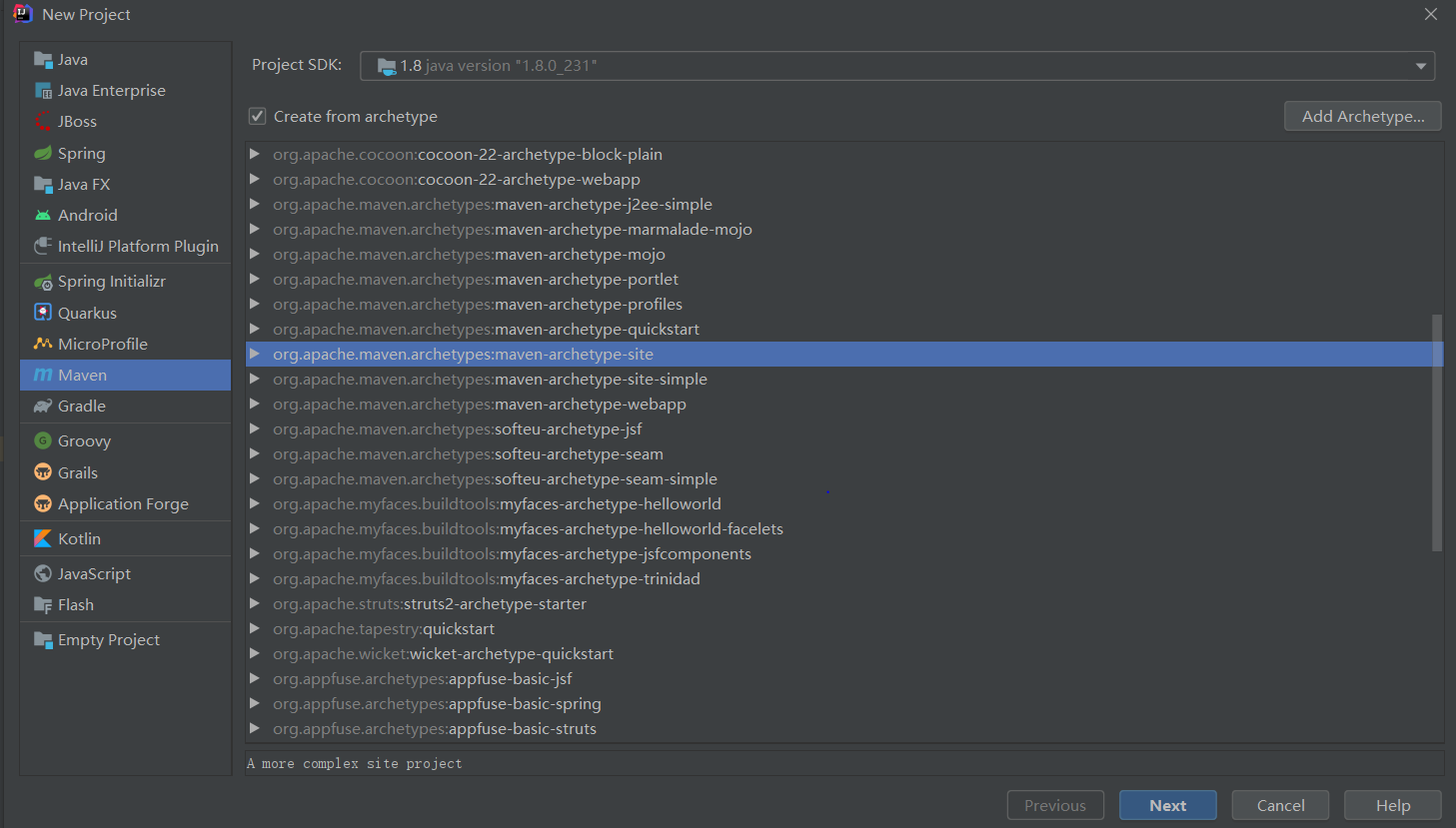
六、服务配置 1、Config（已停更）

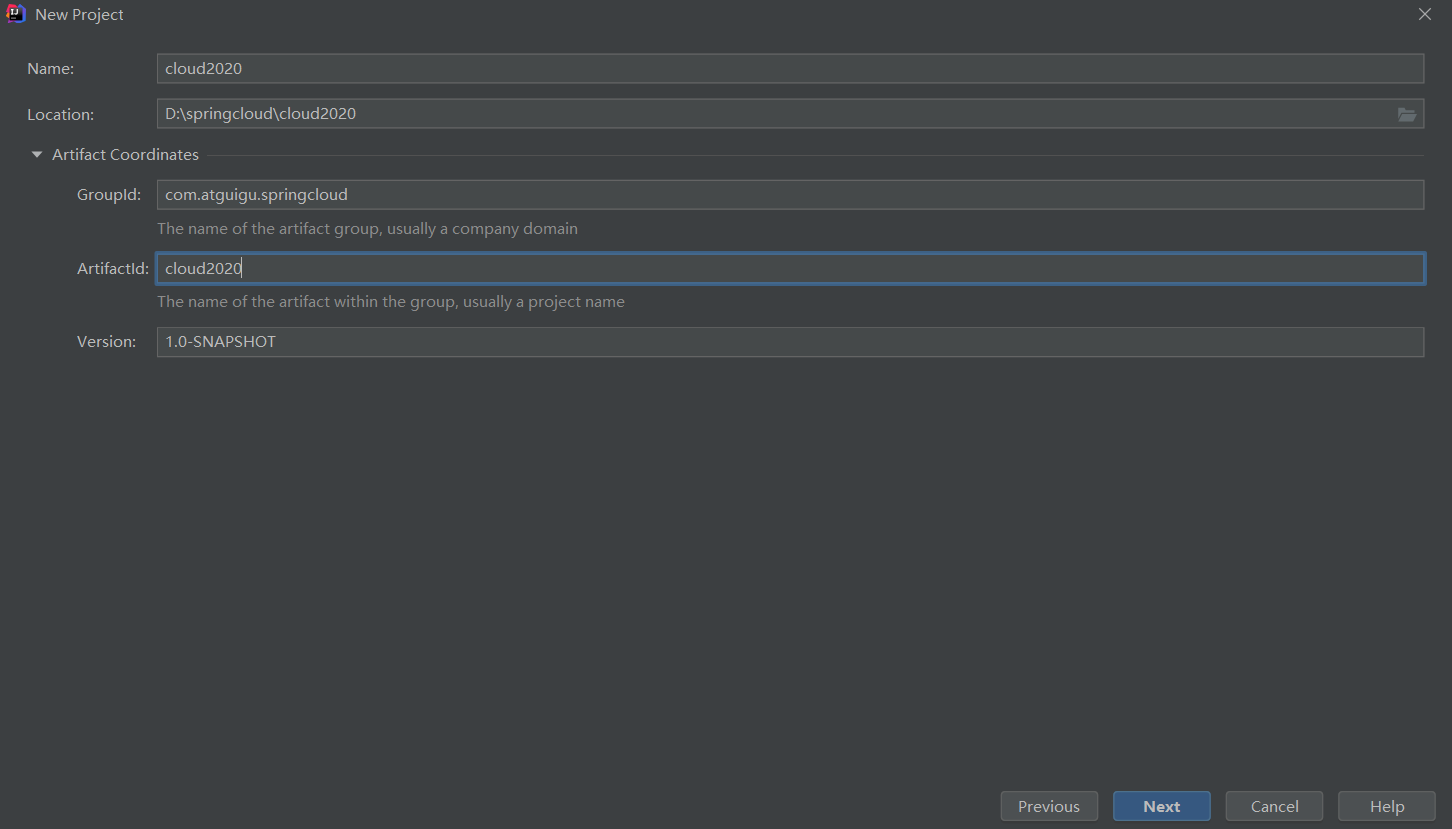
2、Nacos（阿里巴巴）

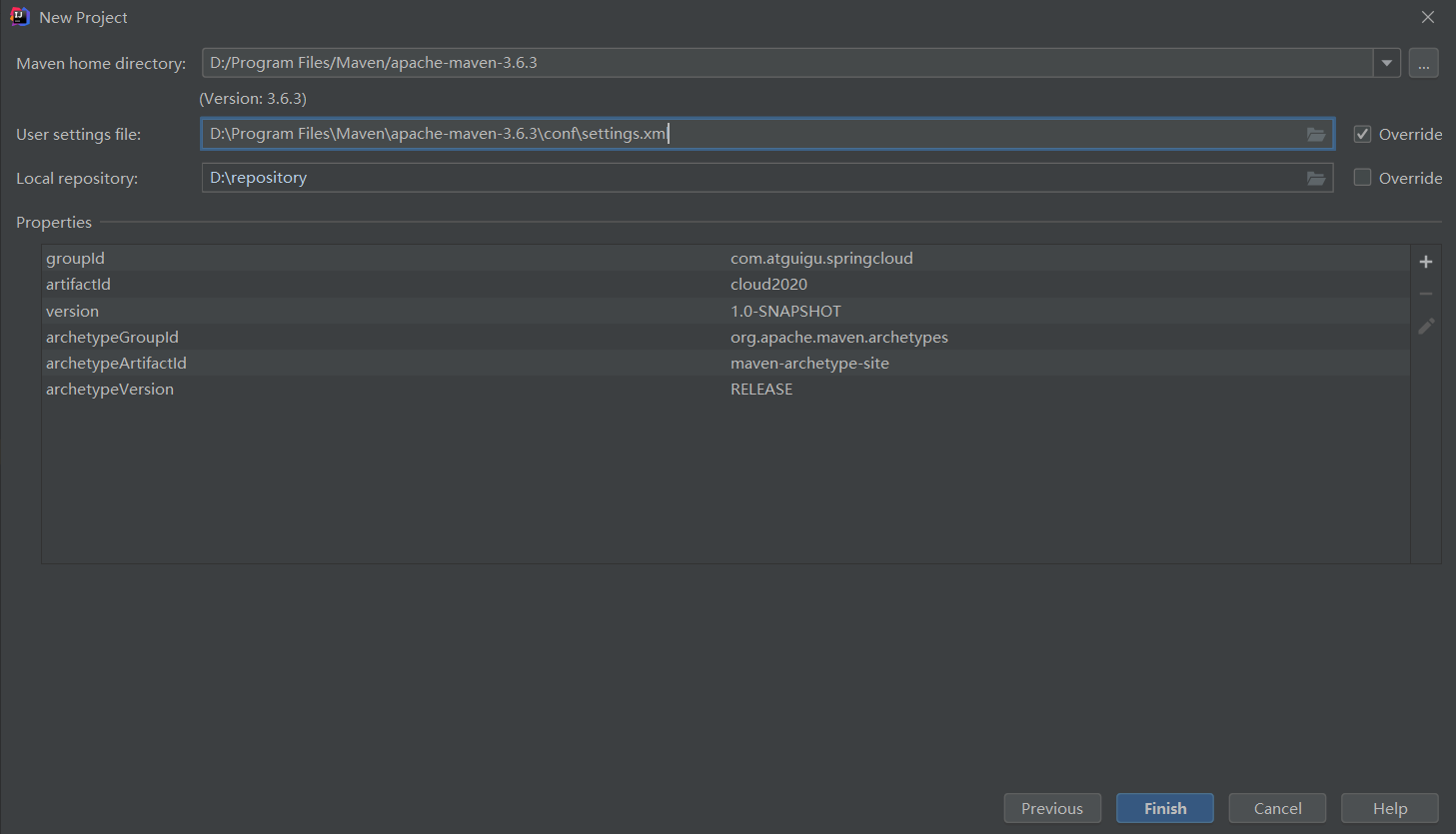
七、服务总线 1、Bus（已停更）

2、Nacos（阿里巴巴）

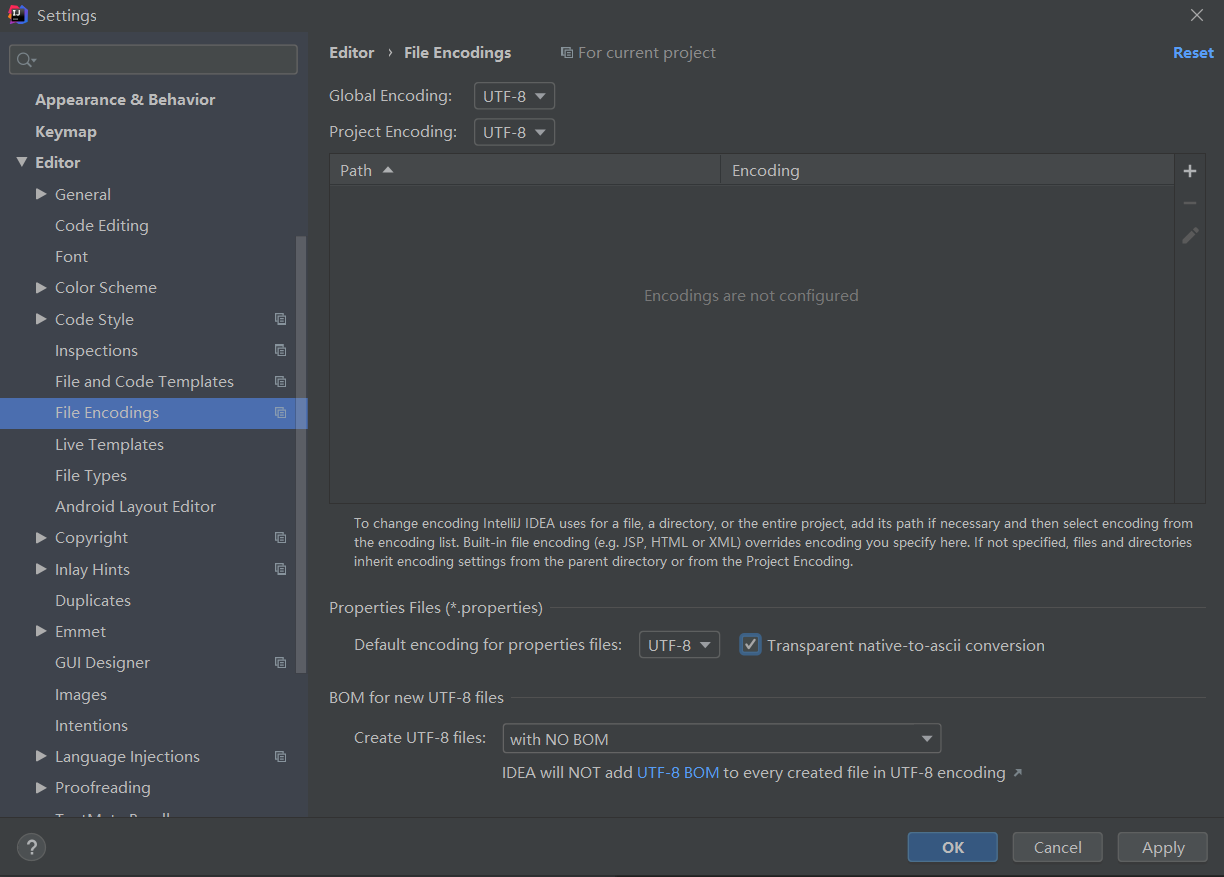
新建工程



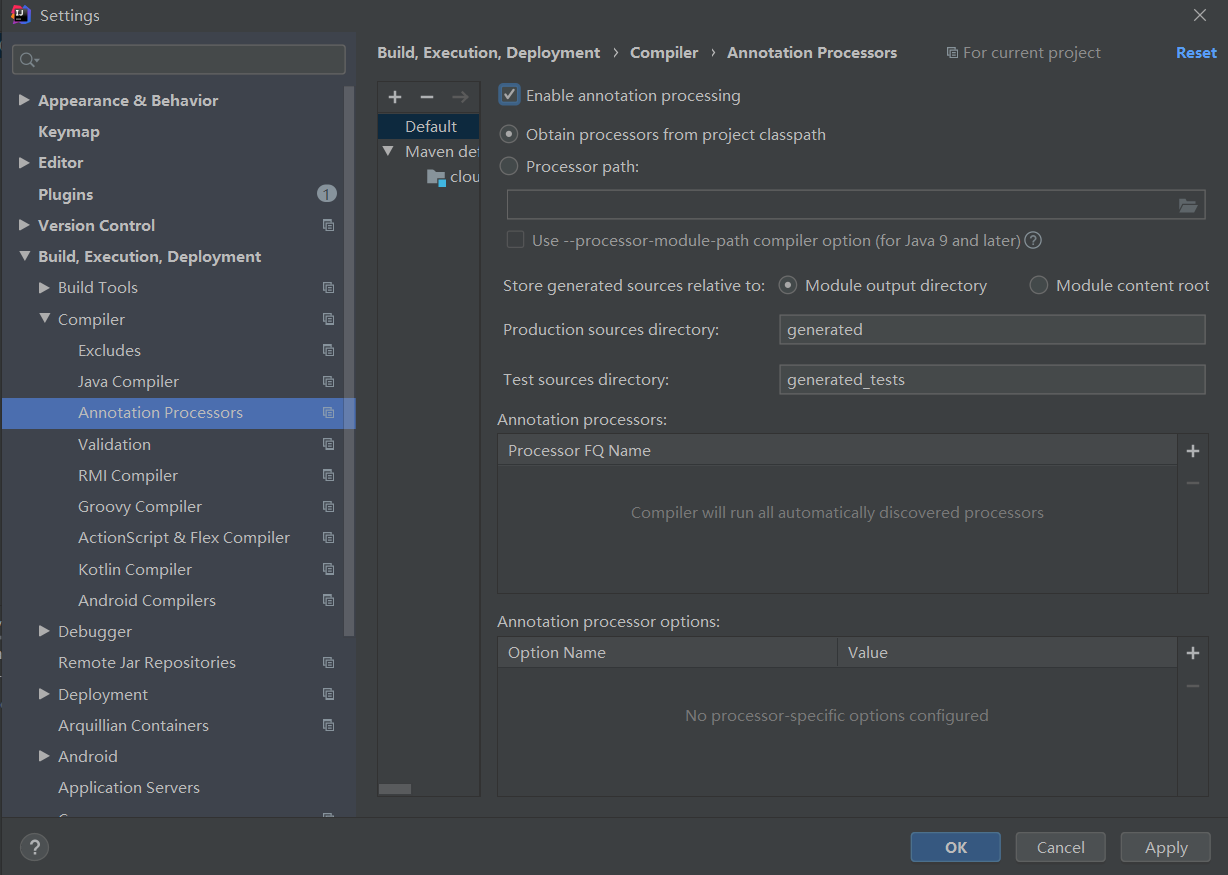




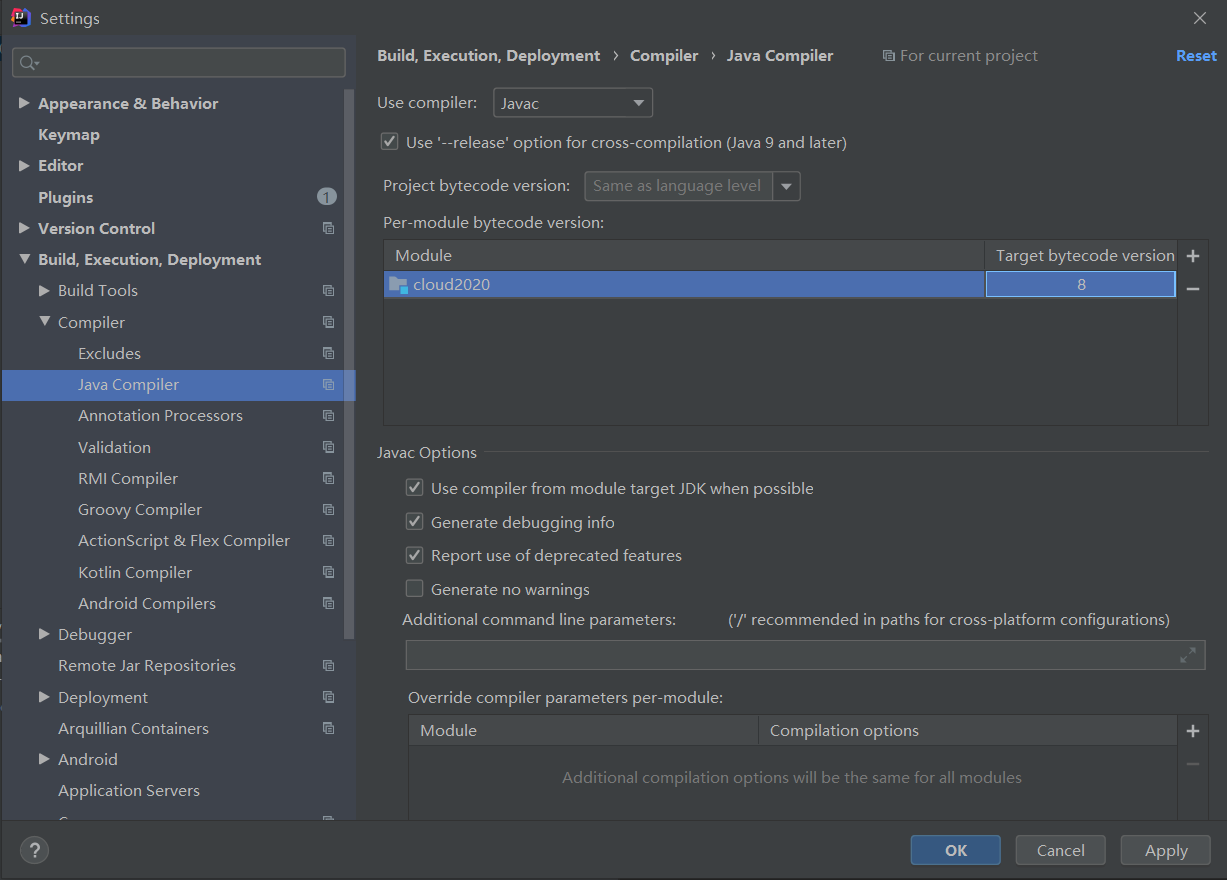
约定字符编码



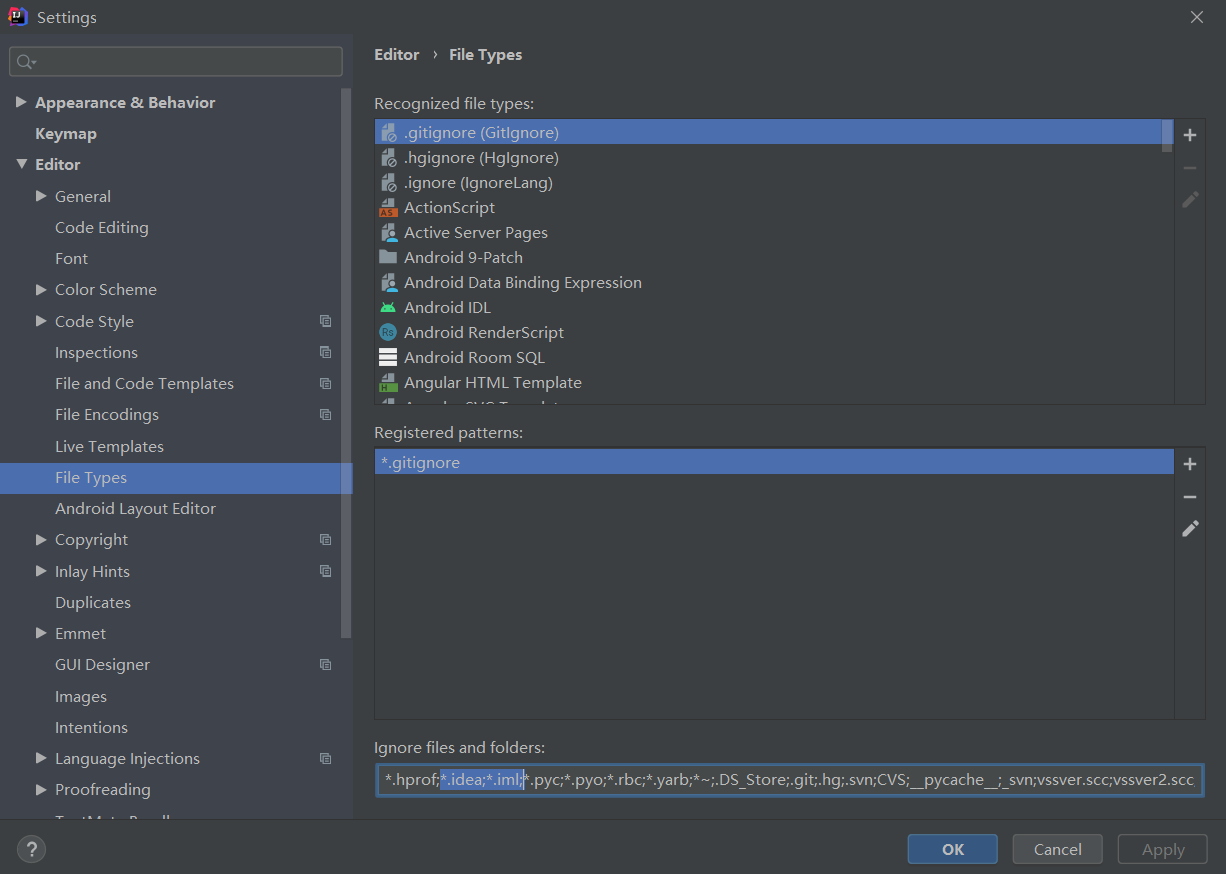
支持注解



设置Java编译版本



文件过滤



父工程pom配置

在父pom中使用dependencyManagement元素能让所有子项目在引用一个依赖时不用显式的列出版本号。

因为，Maven会沿着父子层次向上走，直到找到一个拥有dependencyManagement元素的项目，然后它就会使用这个dependencyManagement元素中指定的版本号。

dependencyManagement里只是声明依赖，**并不实现引入**，因此子项目需要显式的声明需要用的依赖。

如果子项目中指定了版本号，那么会使用子项目中指定的jar版本。

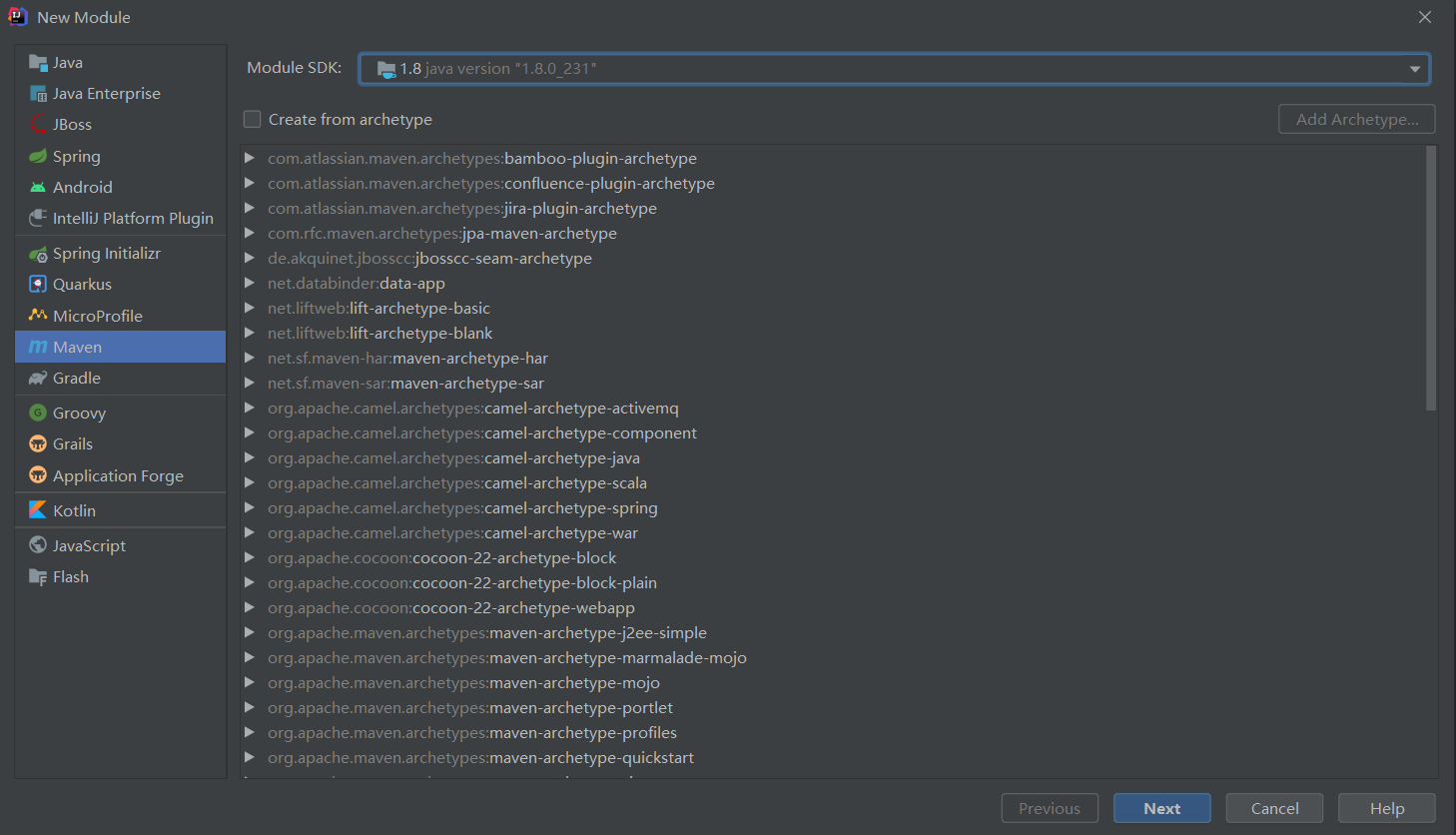
跳过单元测试

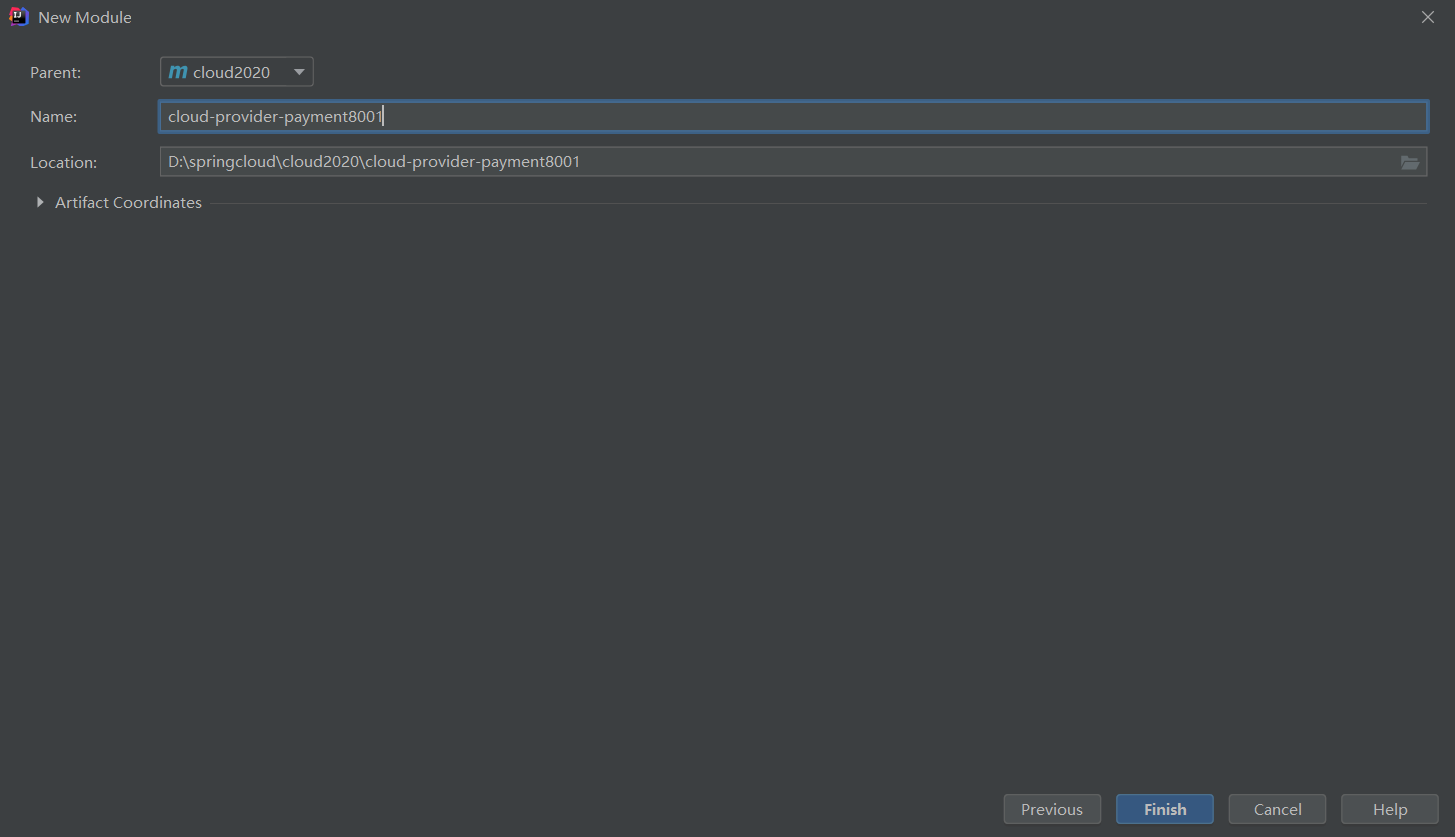
 （到P7）

模块构建步骤

1. 建module
2. 改pom
3. 写yml
4. 主启动
5. 业务类

支付模块构建





业务类构建步骤

1. 建表
2. entities
3. dao
4. service
5. controller

工程重构

maven clean->install（打包）

（到P14）

yml文件编写时要注意层次缩进和空格！

集群环境构建

找到C:\Windows\System32\drivers\etc中的hosts文件

在hosts文件最下方添加如下内容

########################SpringCloud########################

127.0.0.1 eureka7001.com

127.0.0.1 eureka7002.com

<http://localhost:8001/actuator/health> 健康检查

Consul下载地址<https://www.consul.io/downloads.html> （国内已被禁用）

Consul中文文档<https://www.springcloud.cc/spring-cloud-consul.html>

CAP理论

C: Consistency（强一致性）

A: Availability（可用性）

P: Partition Tolerance （分区容错性）（所有分布式架构都要保证这一点）

一个分布式系统只能满足其中两个需求

CAP理论关注的粒度是数据，而不是整体系统设计的策略

Eureka: AP（高可用）

Consul: CP

Redis: CP

MySQL: CA

（到P35）