周六：5-9

https://github.com/liusCoding/springcloud-2020/blob/master/pom.xml

分布式框架，一个技术总称。分布式，微服务的一站式服务

Eureka 所有服务在这注册

Server提供注册中心

Client(微服务的客户端) 服务生产者和消费者，启动时先注册

Feign 请求统一交给它

Hystrix 容错

Ribbon 负载均衡

Zuul 网关

Config 统一配置，GitHub上

Sleuth 追踪服务，

传参

**系统架构**：

浏览量少时，所有业务可以使用一个系统，不容易水平扩展，单点容错率低，并发处理差。

**微服务架构：**

**通过SpringCloud**

**如何自动注册于发现(手动固定不容易维护，挂掉就完了)**

**如何实现状态的监管**

**如何进行动态路由**

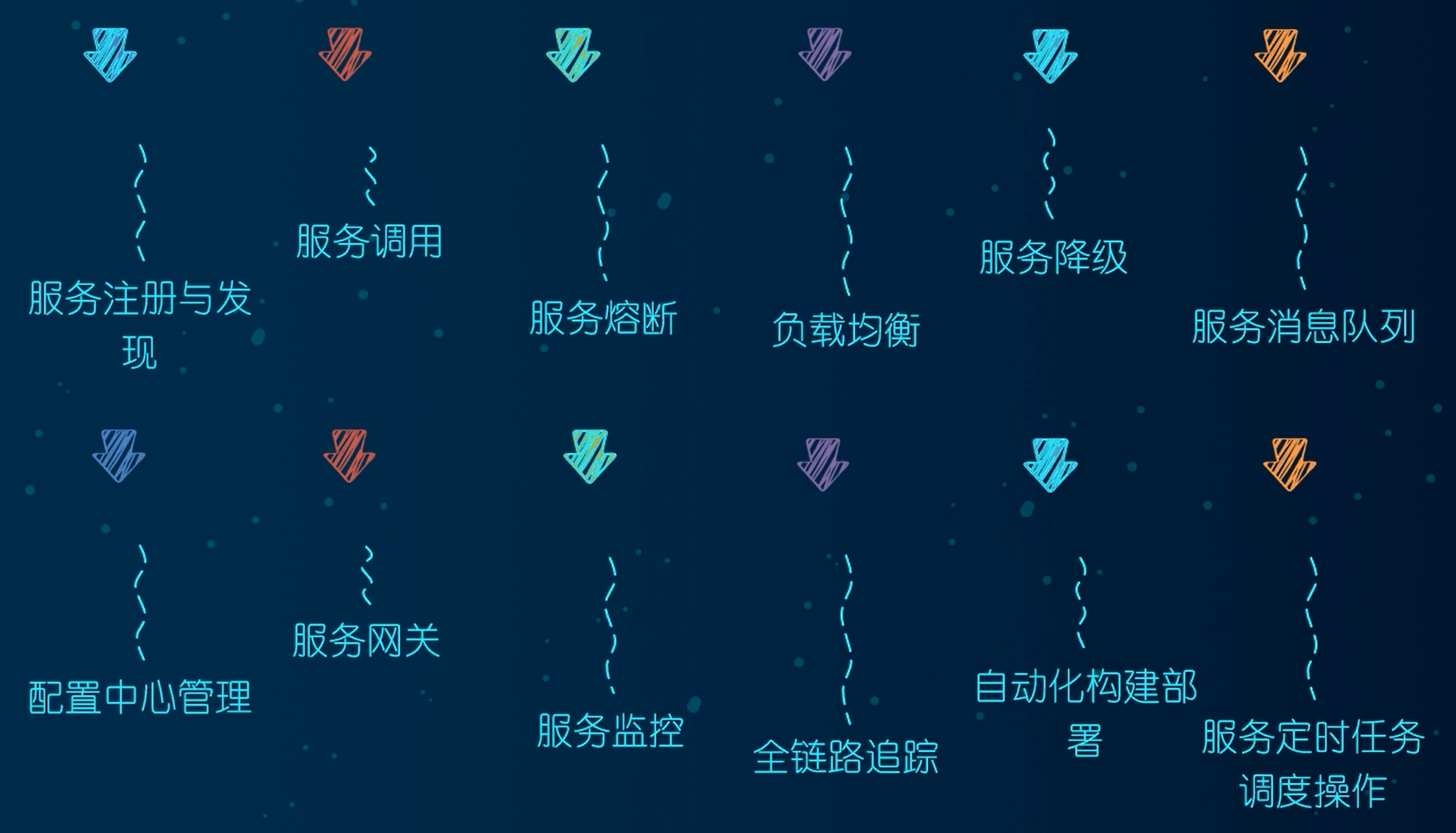
**如何容错**

**如何统一配置**

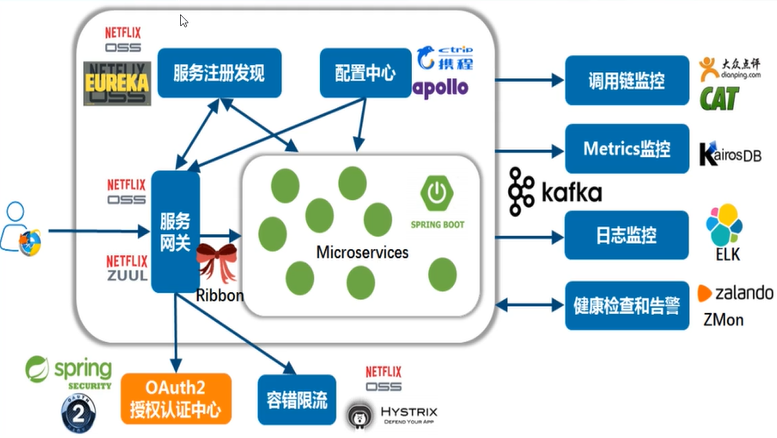
**如何负载均衡**

# 错误

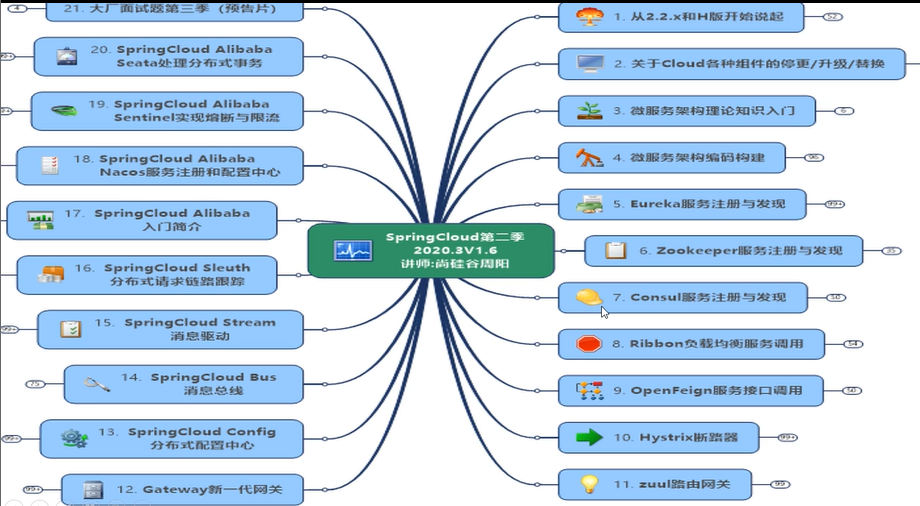






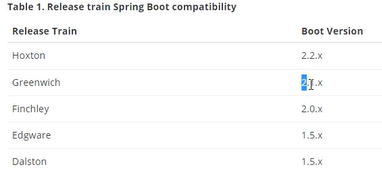




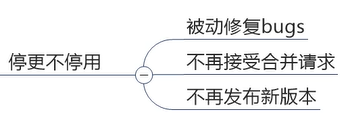


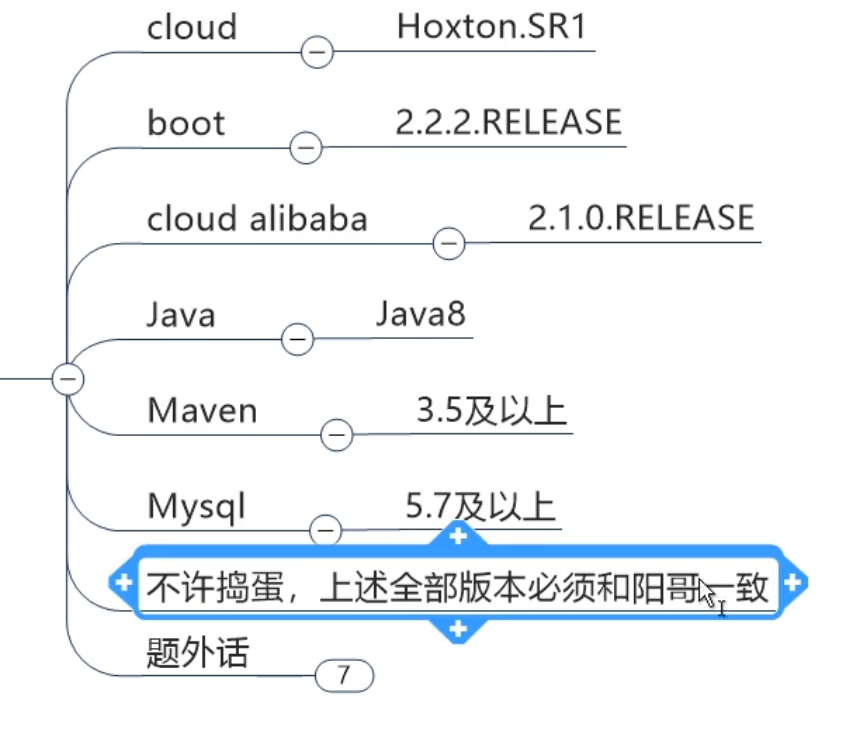


版本问题









停更升级与替换

# 服务注册中心：

Eureka：停了

Zookeeper，接近，容易使Eureka替换成ZK，保守

Cansul：不推荐

**Nacos**：完美

分成服务提供者和服务调用者，注册中心来记录服务，不提供服务,注册中心可以有多个，多个之间相互注册，一个服务可以注册到多个中心。相互注册的服务中心数据共享。1

分成多个，怎么管理？

服务的注册发现于监控。

服务提供方于Eureka之间通过心跳机制进行监控，当某个服务出现问题时会将它从服务列表中剔除。

注册中心：

注册发现：

根据服务名称获取服务地址列表

启动引导类中添加注解@EnableDiscoverClient开启eurake客户端发现功能

配置文件中指定应用名称和eurake的地址。

上述两部就可以将该服务注册到eurake

服务提供者在启动时，会检测配置文件中的eurake.client.register-with-eurake=true是否正确，默认时true，会向注册中心发送一个Rest请求，并携带自己的元数据，注册中心将这些信息保存到一个双层Map结构中。

**失效剔除：**

注册完成后，服务提供者会维持一个心跳，告诉注册中心我还活着。

配置：

自我保护

配置默认打开

注册中心不确定服务是否时可用的，

# 服务调用：

Ribbon：不成熟，停更了

LoadBalancer：很小，

**Feign：**停更了，彻底死了

OpenFeign：推荐

在发送服务地址请求的时候，使用负载均衡拦截器拦截，**根据服务名获得服务列表**，使用对应的负载均衡算法选择一个地址去访问服务。

用

服务器端

客户端(服务器的客户端)

针对某个服务注册负载均衡

RPC(速度快，效率高，基于Sorket)或HTTP(应用广泛，基于TCP协议)

RrstTemplate

httpclient，okhttp，JDK原生URLConnection三种http客户调研工具类包乐意方便的进行http服务调用，spring提供的Spring RestTemplate对上述三个进行了封装

# 服务降级

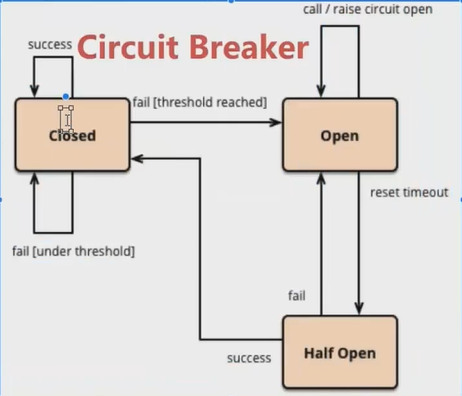
就是**熔断**，自身的保护，防止级联失败

雪崩：某个服务阻塞了，请求一直增加，服务器资源耗尽，导致宕机。

Hystrix：挂了，思想还在用

resilience4j:国外流行

sentienl：阿里的，推荐



# 服务网关

Zuul：分裂了

Zuul2：人跑了，做不出来了

**gateway**：spring推荐

网关是一个服务器，是系统的唯一入口，提供路由功能，还可以进行身份验证，监控，负载均衡，缓存，请求分片与管理，静态资源相应处理等。

转发请求。

进制过滤器

过滤器通信

# 服务配置

Config：挂了

**Nacos**：阿里的，推荐

热部署，及时更新。

配置服务器中读取配置，

配置服务器读git

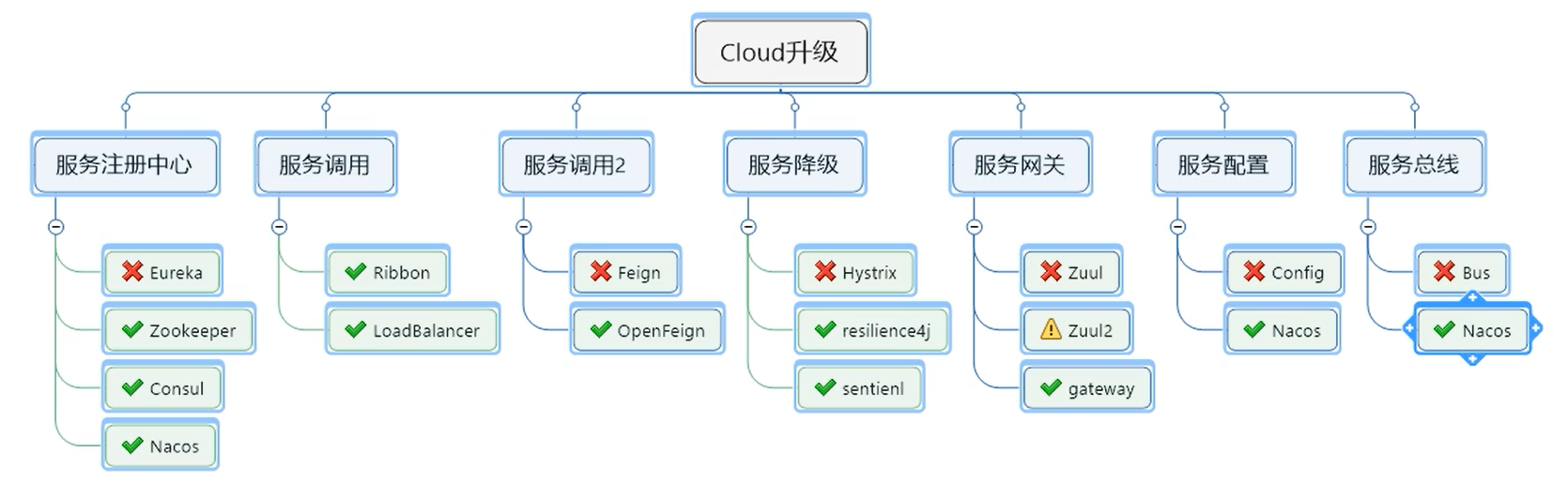
热部署

加密

# 服务总线

Bus：挂了

Nacos：阿里的

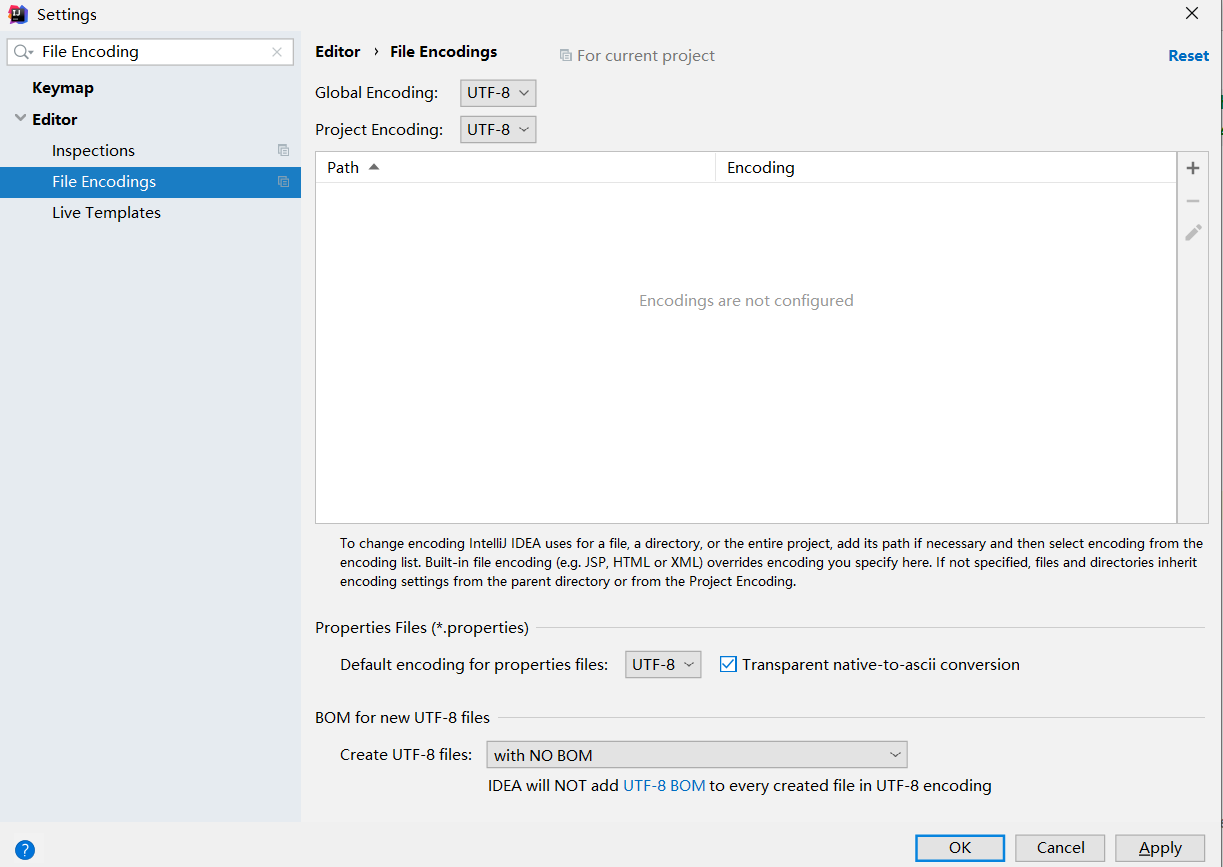


父工程的创建：

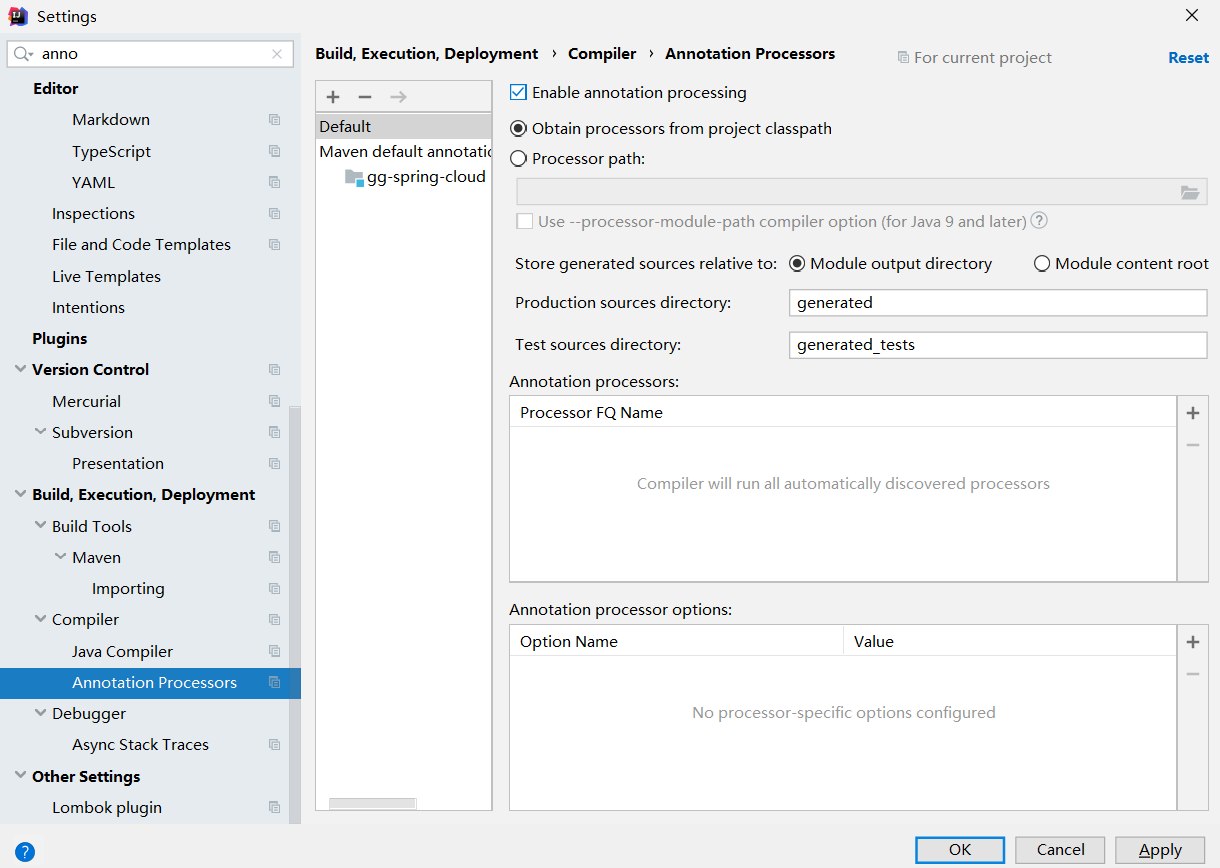
创建一个maven



设置File Encoding



enable



file type filter 不在IDEA中显示不想要的文件

父pom

maven的聚合，继承，依赖，传递依赖

父工程有总的pom文件，通过dependenceManager来进行统一版本号。只留一个pom文件，一般是父pom，它里面声明的依赖，子pom中的依赖可以不再声明版本号，也可以写，它只是声明依赖，并没有引入依赖。



创建订单支付模块的微服务

流程

1. 建moudle
2. 写pom
3. 该yml
4. 写启动类
5. 业务类

web和actuator连用(图形化显示和坐标监控)

resource下建yml，编程绿叶子

创建完子模块时，父pom会多出modules标签

订单表

主键，订单流水

@Slf4j

Controller去调业务，成功返回。。。失败返回。。。

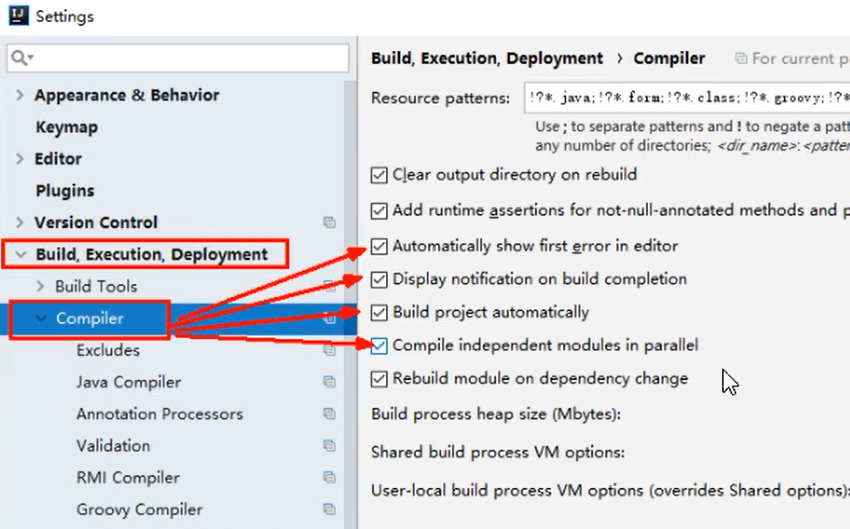
热部署devtools(开发阶段用)

1. 添加jar包

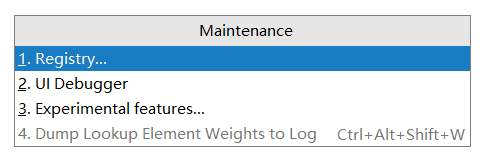
<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-devtools</**artifactId**>  
 <**scope**>runtime</**scope**>  
 <**optional**>true</**optional**>  
</**dependency**>

<**plugins**>  
 <**plugin**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-maven-plugin</**artifactId**>  
 <**configuration**>  
 <**fork**>true</**fork**>  
 <**addResources**>true</**addResources**>  
 </**configuration**>  
 </**plugin**>  
</**plugins**>

1. 









RestTemplate

提供了许多边界访问Http服务的方法，Spring提供的基于Rest的工具集

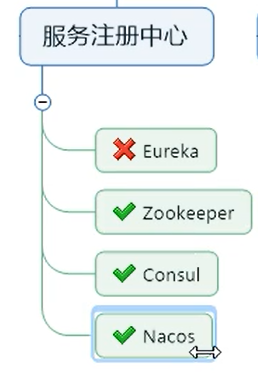
工程重构

将entity打包用一份

新建一个公用的工程，将所有公用资源放到该工程

clean install，install到本地库，其他项目引入这个项目的gav





eureka

为很么不直接调，而是使用注册中心

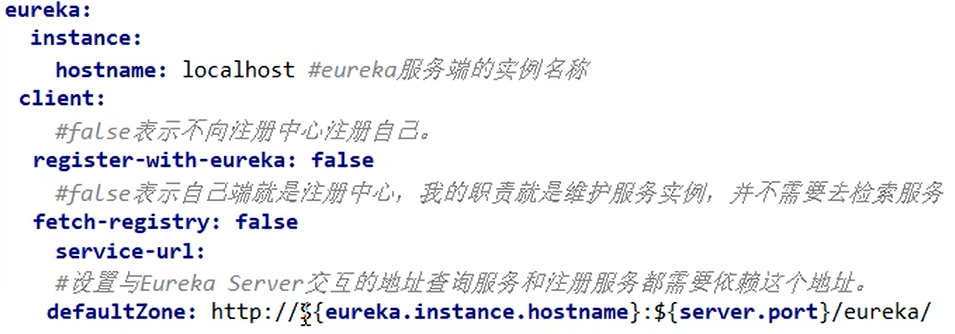
服务多的时候，方便调度协调和管理，调用均衡和容错等

其他服务使用eureka client连接到注册中心，并维持心跳，可以通过eureka server来监控其他微服务工作是否正常

eureka clien会将自己的服务地址，服务名等信息注册到eureka server

其他eureka clien用服务名去eureka server查找服务地址，

心跳的默认周期是30s，如果一个eureka server多个周期内都没收到eureka clien的心跳，就会移除它



SpringBoot->eureka clien

注解+配置

微服务的名称



注册中心爆红，eureka的自我保护机制

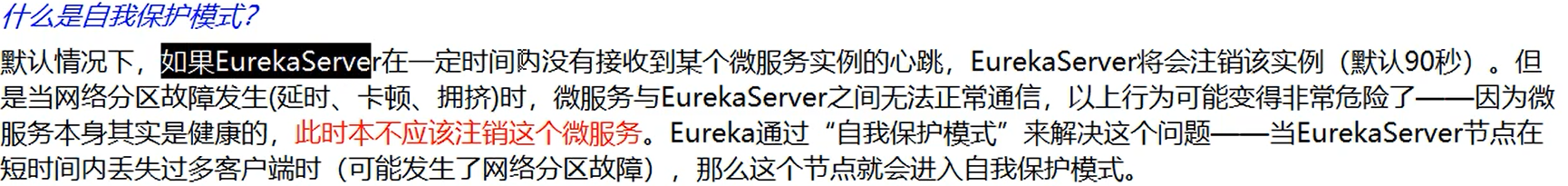
不再删除任何服务 CAP中的AP

**集群的高可用**

eureka集群：互相注册，相互守望，对外暴露出一个整体

不需要加client注解，因为server也可以注册发现



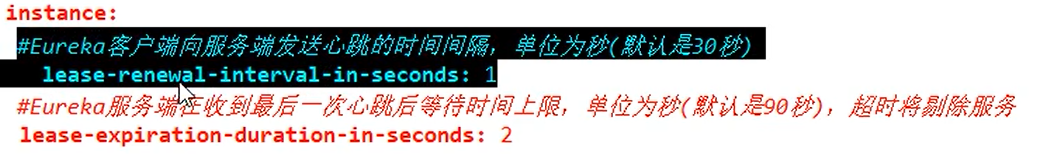


如何禁用eureka的自我保护机制



默认是开启的

为了看效果，其他服务的设置



1. 搭建eureka集群
2. 将微服务注册进集群

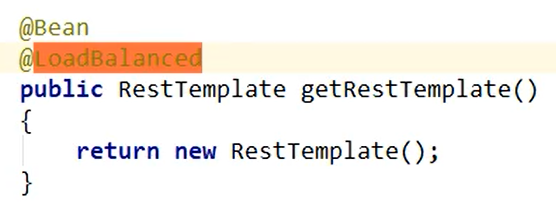


调用服务时，地址不再写死，而是只写服务的名称



但是不知道时这个服务的哪个地址，需要加restTemplate的一个注解





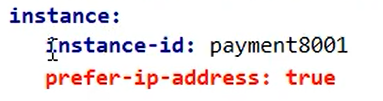




主机名称：

注册中心的显示：去掉主机名称，只显示服务名

在配置文件中添加instance属性，来改变显示的instance名称



为true标识可以显示Ip

健康检查

几号机，服务器的端口，服务名称

服务发现Discover

拿到注册中心中服务的信息

**用来获得服务的信息**

