1. 微服务（微服务是一种架构风格）
2. 一系列微小的服务组成
3. 每个服务都跑在自己的进程里
4. 每个服务为独立的业务开发
5. 独立部署

单体架构：

1. 优点：容易测试、容易部署
2. 缺点：开发效率底、稳定性不高、代码维护难、部署不灵活、稳定性不高、扩展性不够

分布式架构：（旨在支持应用程序和服务的开发，可以利用物理架构 由多个自治的处理元素，不共享内存，但通过网络发送消息合作）（分布式是一个厨师，一个洗碗工。微服务是一堆厨师和一堆洗碗工）

Spring cloud是什么？

1. 是一个开源工具集，含了多个子项目。
2. 简化了分布式开发

Eureka

1. 基于Netflix Eureka的二次封装
2. 包含Eureka Server 注册中心，
3. Eureka Client 服务注册
4. <https://spring.io/projects/spring-cloud> 官网查看各个组件版本对应

基本配置

*# eureka.server.enable-self-preservation：是否开启自我保护  
# 会进入自动保护模式，注册中心并不会将该服务从注册中心删除掉。设置成false  
# eureka.instance.prefer-ip-address：true 是否以 IP 注册到注册中心  
# eureka.instance.instance-id：注册限制的实例 ID，即上图显示的 127.0.0.1:8761  
# eureka.instance.hostname 来修改 hostname 的值,Eureka 默认是以 hostname  
#  
# client.serviceUrl.defaultZone：注册中心默认地址。  
# eureka.server.peer-node-read-timeout-ms：微服务节点连接超时时间。  
#  
# eureka.client.register-with-eureka: false 是否向服务端注册自己  
# eureka.client.false.fetch-registry: false 表示是否从Eureka Server获取信息*

1. 高可用 多个Eureka Server 集群相互注册

坑点application.yml 使用拷贝的时候 有时候注解会引发错误，导致运行异常

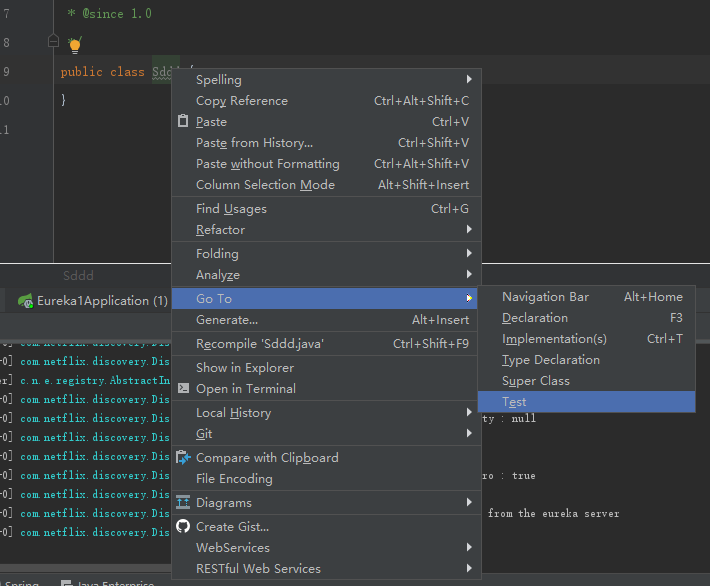
defaultZone: http://localhost:8761/eureka/,http://localhost:8762/eureka/

每一个server 需要在defaultZone填写其他注册中心地址

defaultZone: http://localhost:8761/eureka/,http://localhost:8762/eureka/,http://localhost:8763/eureka/

在client端填写所有注册中心地址

快速添加测试类





每个服务都有自己的数据库

其他服务需要这些数据，需要到指定服务去调用其Api获取，不能自己访问

**@NotEmpty 用在集合类上面**

**@NotBlank 用在String上面**

**@NotNull 用在基本类型上**

Ribbon

负载均衡器。

1. 服务发现（ServerList），根据服务名字，把相同的服务放在一起
2. 服务选择(IRule)，依据一些原则，选择一个服务处理业务
3. 服务监听(ServerListFilter)，服务开启关闭

默认规则轮询模式

Feign