# 01尚硅谷——SpringCloud\_前提



## cloud技术的五大神兽

服务发现—NetFlix Eureka

客户端负载均衡—Netflix Ribbon

断路器—Netflix hystrix

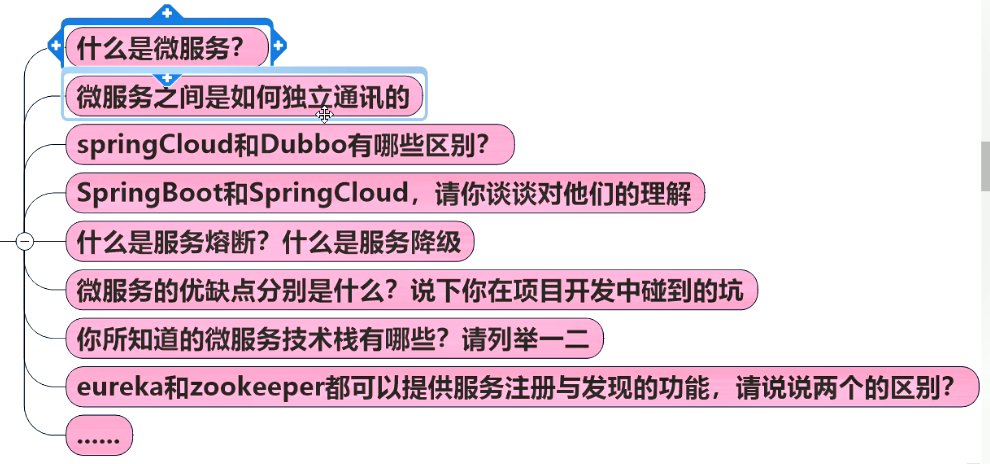
服务网关—Netflix zuul

分布式配置—Spring cloud config

本次课程只是cloud的第一季

# 02SpringCloud大纲

# 03SpringCloud从面试题开始



# 04微服务是什么



dubbo是基于rpc(Remote Procedure Call)远程过程调用。

cloud是基于RESTful API调用

## 技术维度理解

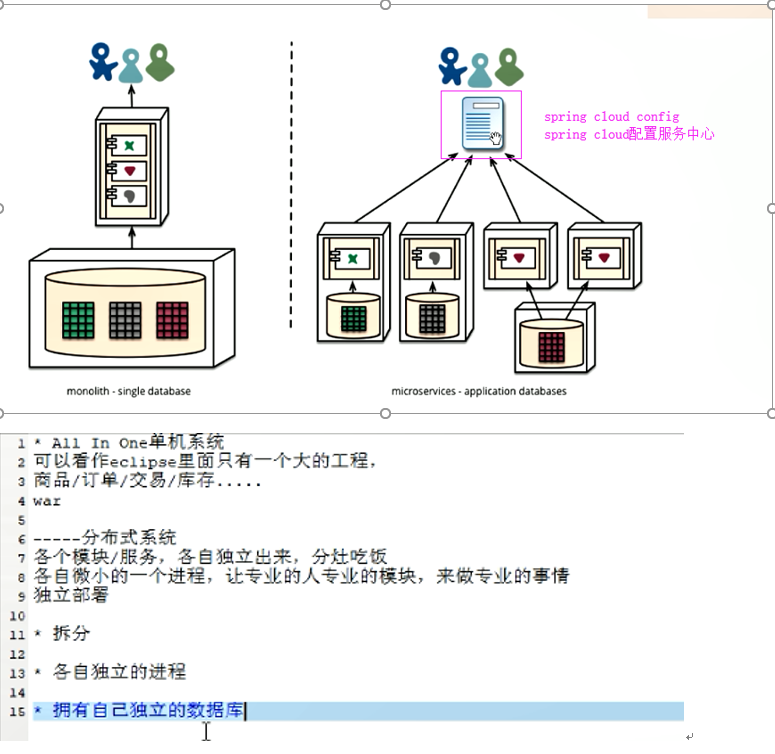
微服务化的核心就是将传统的一站式应用，根据业务拆分成一个一个的服务，彻底

地去耦合,每一个微服务提供单个业务功能的服务，一个服务做一件事，

从技术角度看就是一种小而独立的处理过程，类似进程概念，能够自行单独启动

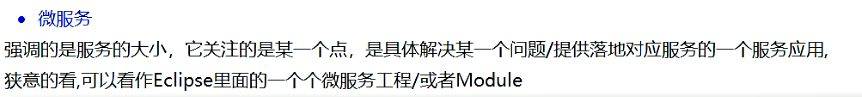
或销毁，拥有自己独立的数据库。

# 05微服务是什么2



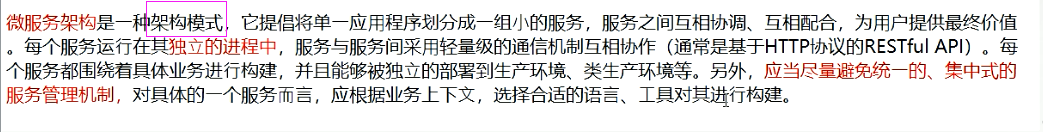
# 06微服务和微服务架构

## 什么是微服务

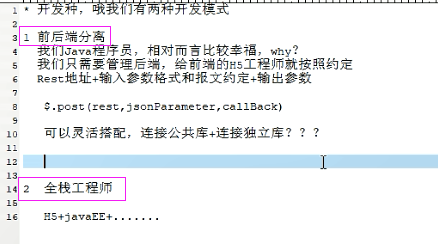


eclipse工具里面用maven开发的一个个独立的小moudle,它具体是使用springboot开发的一个小模块，专业的事情交给专业的模块来做，一个模块就做一件事情，强调的是一个个的个体，每个个体完成一个具体的任务或者功能。

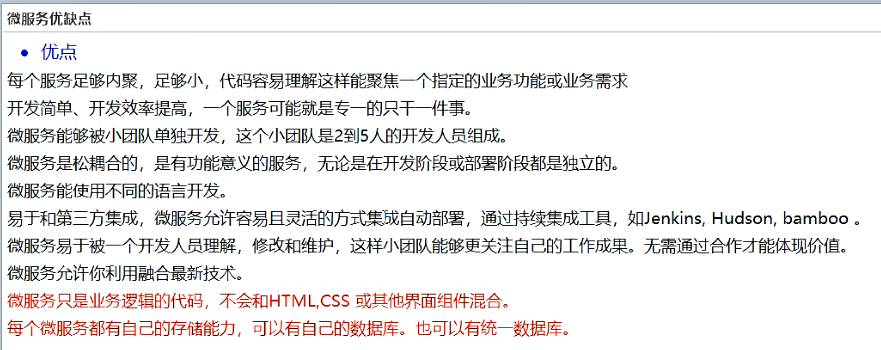
## 什么是微服务架构



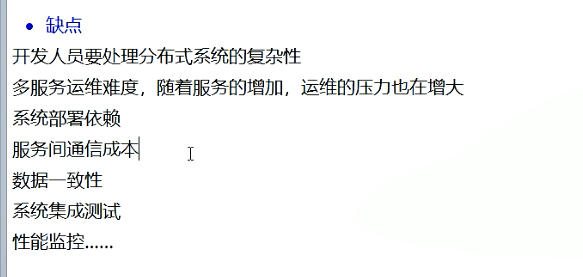
# 07微服务优缺点



## 优点



## 缺点



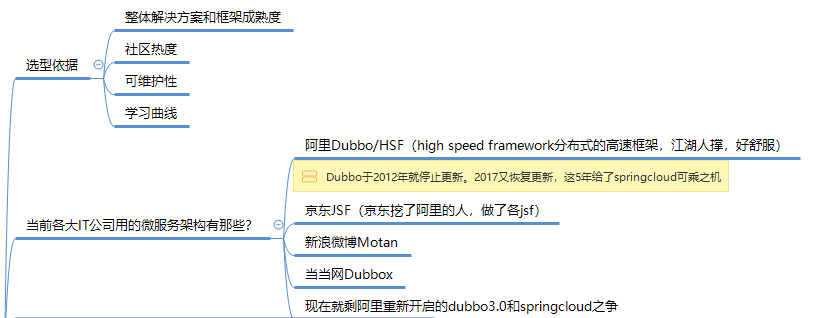
# 08微服务技术栈有哪些（这个栈指的多种技术的集合体）

举个例子





# 09为什么选择springcloud作为微服务架构（老系统用dubbo,）



# 10springcloud是什么

SpringCloud，基于SpringBoot提供了一套微服务解决方案，包括服务注册与发现，配置中心，全链路监控，服务网关，负载均衡，熔断器等组件，除了基于NetFlix的开源组件做高度抽象封装之外，还有一些选型中立的开源组件。

SpringCloud利用SpringBoot的开发便利性巧妙地简化了分布式系统基础设施的开发，SpringCloud为开发人员提供了快速构建分布式系统的一些工具，包括配置管理、服务发现、断路器、路由、微代理、事件总线、全局锁、决策竞选、分布式会话等等,它们都可以用SpringBoot的开发风格做到一键启动和部署。

SpringBoot并没有重复制造轮子，它只是将目前各家公司开发的比较成熟、经得起实际考验的服务框架组合起来，通过SpringBoot风格进行再封装屏蔽掉了复杂的配置和实现原理，最终给开发者留出了一套简单易懂、易部署和易维护的分布式系统开发工具包

# 11SpringCloud和springboot的区别

springboot是微观的，（相当于医院的一个一个科室）

springcloud是宏观的（相当于对外的一个医院）

springcloud依赖于springboot.



# 12SpirngCloudVSDubbo区别对比（烂熟于心）

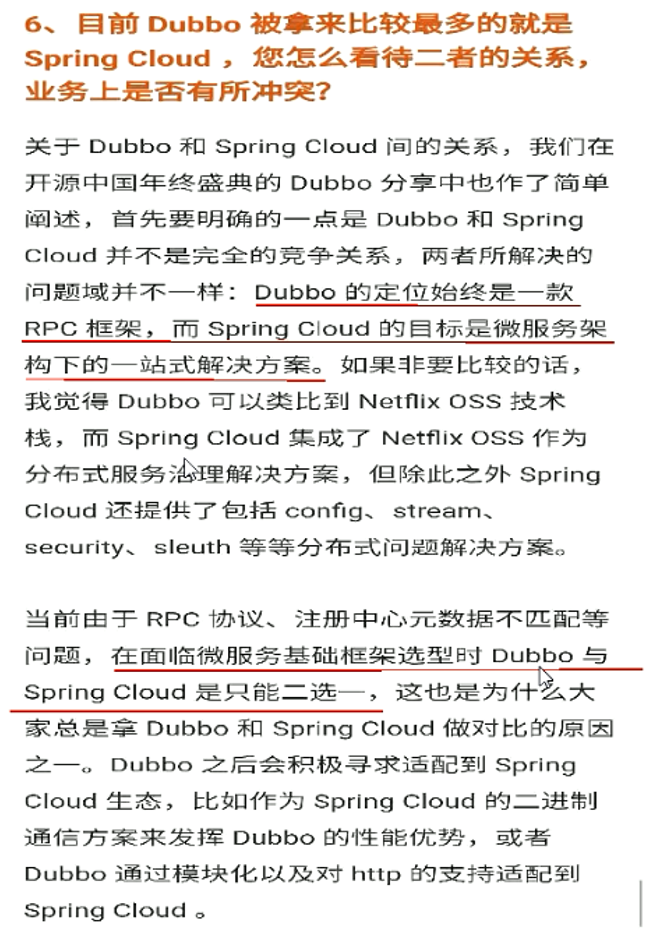


LVS+Keepalived实现高可用

## 两者对比结果



## 刘军

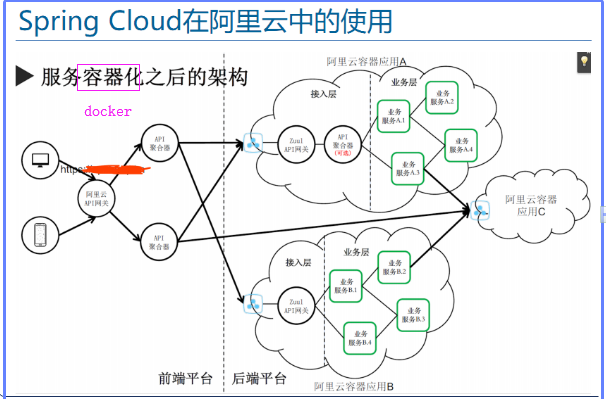


# 13SpringCloud功能和官网资料介绍



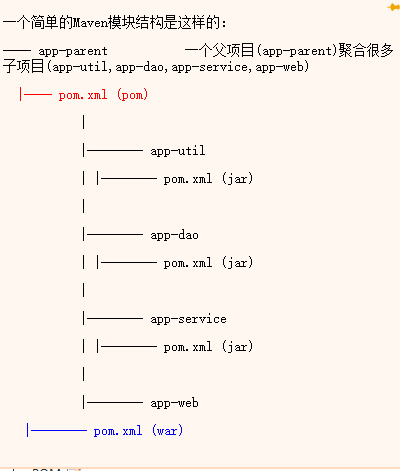
# 14springcloud国内使用情况





# 15Springcloud\_Rest微服务案例-总体概述

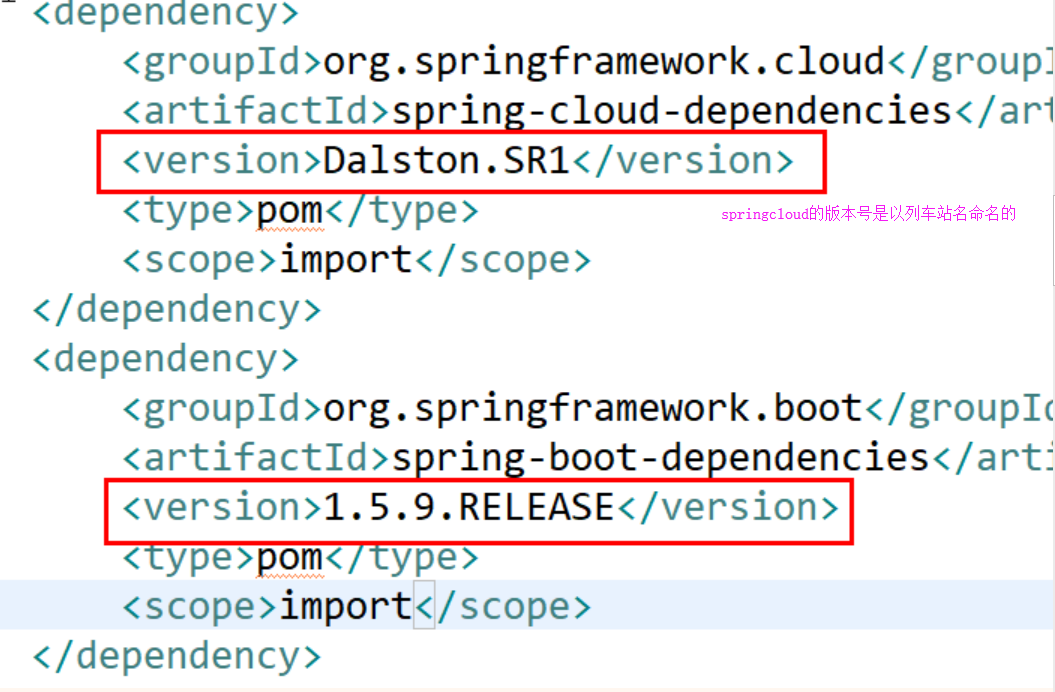
springmvc+springboot+mybatis整合的案例



# start

# 16SpringCloud\_Rest微服务案例（父工程构建步骤）

## 本次springcloud版本和spingboot版本

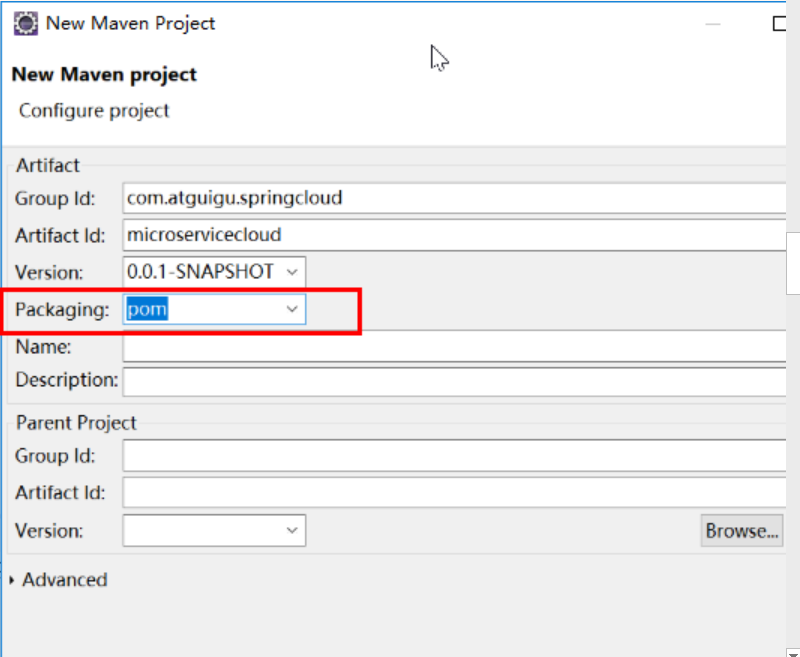


注意：cloud和boot的对应

## 第一步创建work set

在 package explorer 空白处,右击,new->Other->Java Working Set,输入 Working Set 的名字,选择相应的Projects(按住Ctrl,一次选多个;按住Shift,连续选择),Finish.创建好 working set 之后 , 选择 package explorer 的 倒三角 ,Top Level Elements->Working Sets.

## 第二步新建父工程（记住是pom模式）



作用：主要是定义POM文件，将后续各个子模块公用的jar包等统一提出来，类似一个抽象类

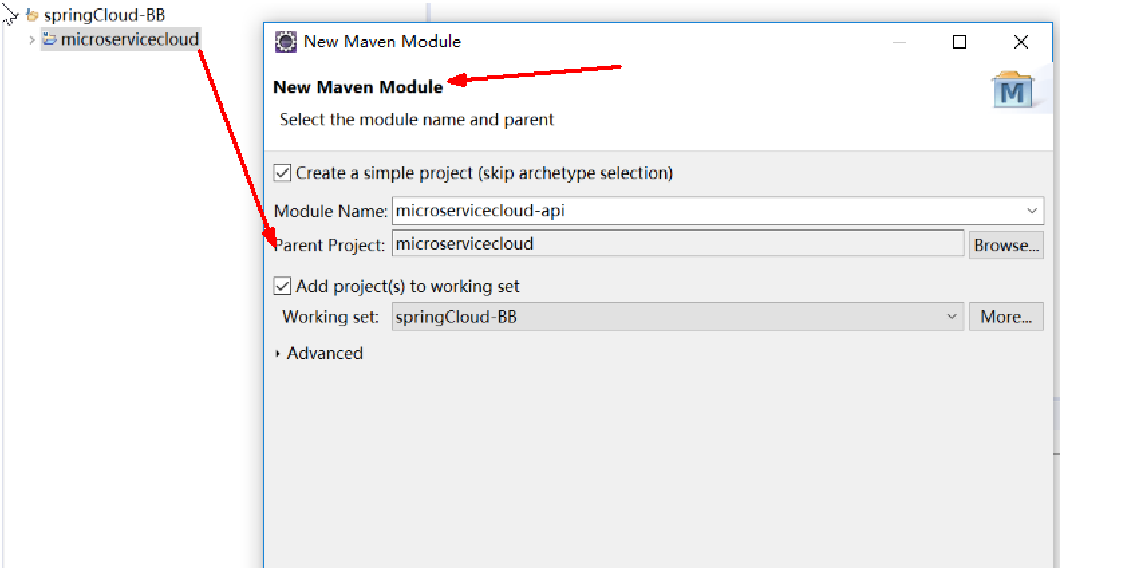
### pom.xml文件



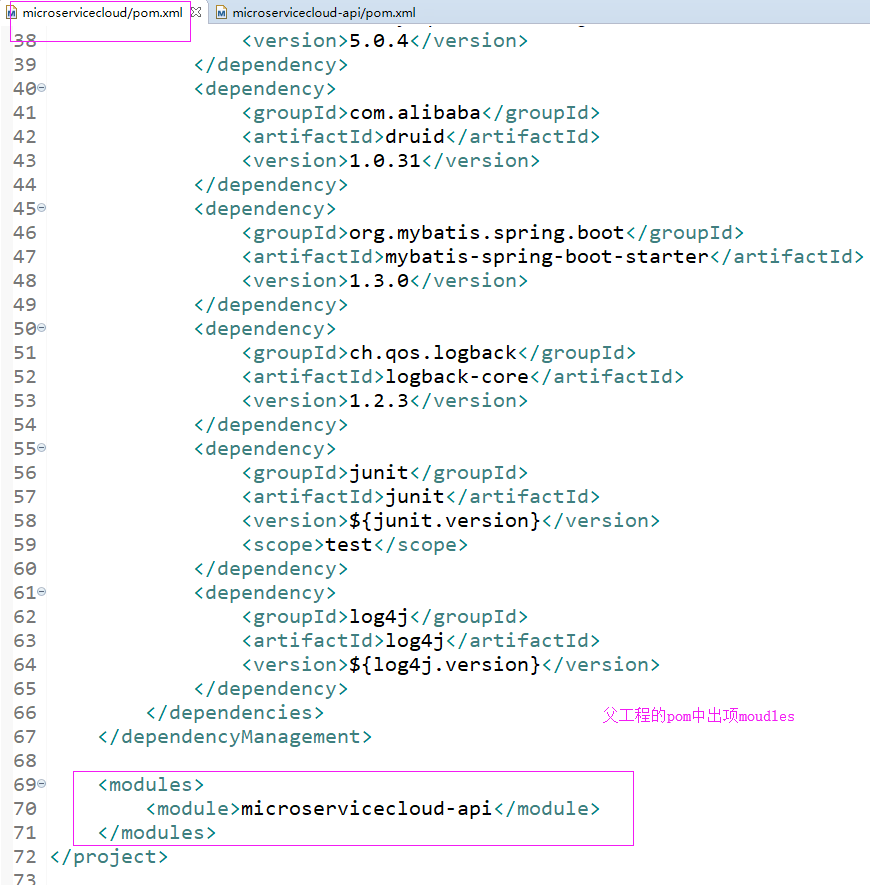
# 17SpringCloud\_Rest微服务案例-API公共模块和部门Entity步骤

## 第一步

在父工程中即（microservicecloud中）新建microservicecloud-api



## 第一步创建完成后



## 第二步

在microservicecloud-api中田间lombok依赖

## 第三步

新建部门



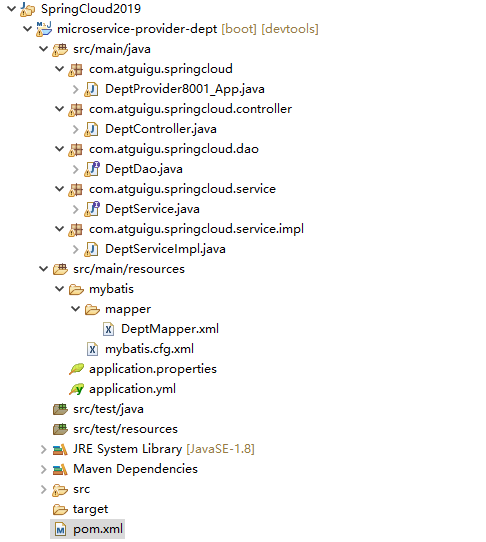
## 第四步

mvn clean install后给其它模块引用，达到通用目的。

也即需要用到部门实体的话，不用每个工程都定义一份，直接引用本模块即可。

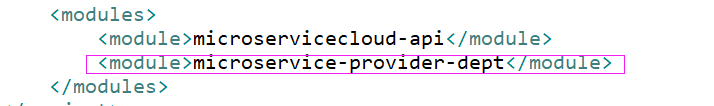
# 18SpringCloud\_Rest微服务案例-部门服务提供者

## 最终工程展现



## 1新建microservicecloud-provider-dept-8001maven module

建立成功后，父工程会多个子模块



## 2.配置pom



## 3.yml和properties

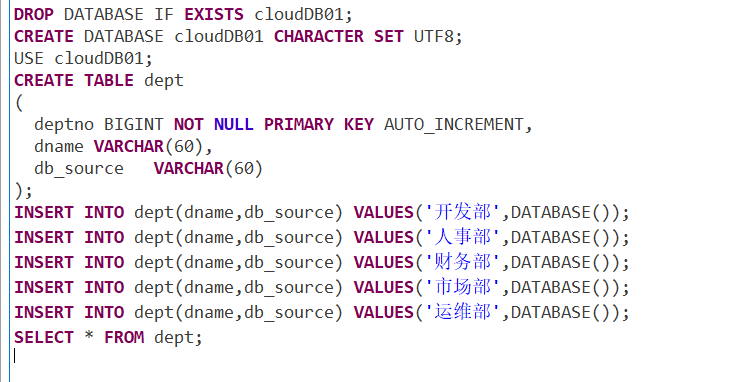




## 4.工程src/main/resources目录下新建mybatis文件夹后新建mybatis.cfg.xml文件



## 5.MySQL创建部门数据库脚本



## 6.DeptDao部门类



## 7.工程src/main/resources/mybatis目录下新建mapper文件夹后再建DeptMapper.xml



## 8.DeptService部门服务接口



## 9.DeptServiceImpl部门服务接口实现类



## 10.DeptController部门微服务提供者REST

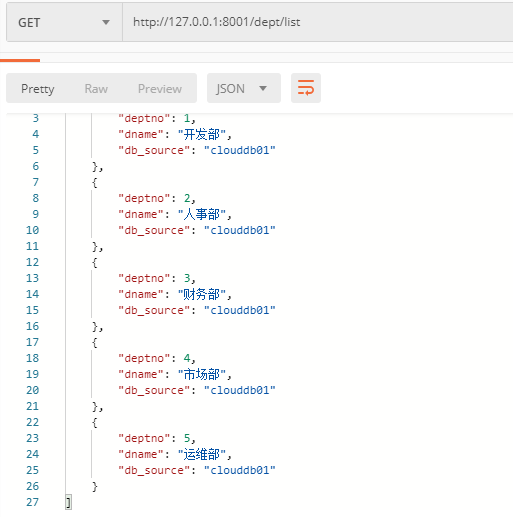


## 11.DeptProvider8001\_APP主启动类DeptProvider8001\_App



## 12测试





# 19SpringCloud\_Rest微服务案例-部门服务消费者