Php和java

1）2017鹿晗关晓彤，新浪微博瘫痪了，临时加1000 redis

2）2018 赵丽颖和冯绍峰结婚，新浪微博又瘫痪了。

为什么会造成这种瘫痪？高并发，预案

1. 预估业务峰值，新浪微博支持4个明星出轨（王宝强，超过常规峰值2倍）
2. 领导，钱。不能快速响应。

知识回顾：

1. SpringCloud微服务框架
   1. Eureka注册中心
   2. Ribbon前端负载均衡
   3. Feign声明式（接口）支持RESTFul形式
   4. Hystrix断路器
   5. Zuul API网关
   6. Sidecar异构开发语言支持
   7. SpringCloud ConfigServer配置中心
   8. Dubbo官宣它是SpringCloud子集，专注于解决RPC通讯解决方案
2. ConfigServer

Why?

有这样个别配置项，有变化？

举例：税率，出口退税20%，骗税。

5%

动态的配置项：数据库+Redis

Redis LRU算法，最近最久未使用就驱逐

单独给它一个redis节点（集群）

传统方案：

1. 写死属性配置文件jdbc.properties
2. 数据库+Redis

配置中心：完全支持分布式，依赖git来完成数据存储

和项目结合：

1. 搭建Config-Server
   1. 创建SpringBoot项目
   2. 增加依赖config-server
   3. 配置关联git保存数据，自动维护多个版本
2. 搭建客户端，最终是消费者Config-client

客户端启动时，扫描@Value，去hashmap中找对应key，找不到报错。

1. 动态刷新配置（手工刷新一个连接地址）

http://localhost:7020/refresh 访问config-server端，必须以post请求提交

部署在内网，部署到一个需要登录系统上，对shiro安全校验

在Controller上增加一个注解@RefreshScope

pom.xml增加一个依赖starter-actuator监控需要jar依赖

4）@Value处理在application.yml文件处理之前处理。必须换一个配置文件bootstrap.properties。bootstrap启动时间是在application之前。

20%，5%，按每年第一天

2018年税率20%，2019年1月1日开始按5%来计算

什么时间程序调用对应的税率？

做一个程序来判断一下当前时间，如果19.1.1就切换新的税率。

问题：

1. 是否能保证计算机是19.1.1（手工）
2. 未完成业务

给配置项加一个版本号

20%,v1.0

5%,v1.1

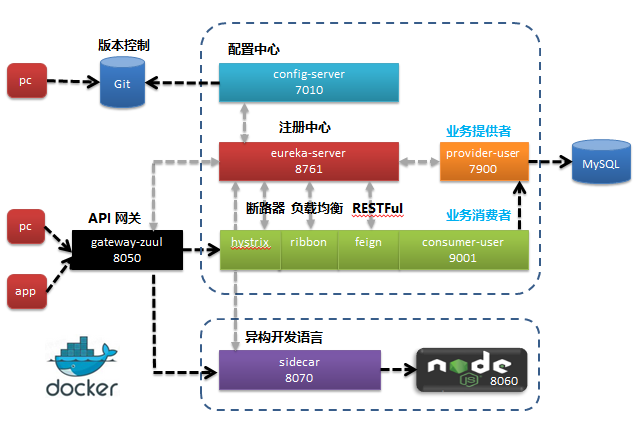
版本控制，Git完成版本控制，

系统可以删除以前配置项，数据库可以比较随机删除，但是Git记录所有的版本，保证历史数据都不会错。

配置中心，ZooKeeper

Zk存储数据，数据量非常小，数据字典，树形结构

整合User表CRUD整体操作



1. 实现Eureka注册中心（复用）
2. 实现业务提供者（SSM，京淘使用通用Mapper，升级MP MybatisPlus）
3. 实现消费者（hystrix+ribbon+feign）
4. 实现网关（复用，只需修改配置文件，增加一个映射）

开发步骤：

1. 服务提供者s
2. 服务消费者

MybatisPlus



项目框架：springboot+SSM+MP

通用Mapper MybatisPlus

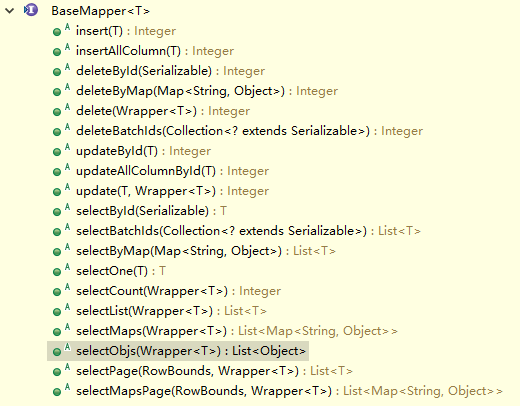
1. 继承SysMapper接口，扩展批量删除 继承BaseMapper
2. 在pojo上JPA注解，不用SQL语句 自定义注解@TableName
3. 支持分页PageHelper，Mybatis拦截器 在加载时处理
4. 不很好支持查询条件、排序、分组 EntityWrapper包装

QBC( Query By Criteria) 出处hibernate QBC，鸡肋，号称面向对象方式

Select(map) birthdayStart,birthdayEnd

Select id,name from user where id=1 order by id

Select \* from user where year(birthday) between 1900 and 2000 and id=1



动态生成SQL

SELECT id,NAME,birthday,address FROM user

分析：

1. 关键字：SELECT FROM（selectList）
2. 字段：pojo属性，
3. 表：@Table(“tb\_user”)JPA

架构：springmvc+spring+mybatis+myabtisPlus

1. pojo+mp注解
2. UserMapper接口，继承BaseMapper
3. UserService、UserServiceImpl
4. UserController

注册权法，规定：70%以上代码，文字，就是抄袭



消费者

1. UserFeign接口
2. UserController调用UserFeign
3. 启动

小结：

1. 找最简单一个业务（查询所有），把项目架构整个搭建起来
2. 整个架构：zuul调用consumer，consumer调用user，user基于SSM+mp
3. 整个开发过程倒着做。SpringBoot+SSM+MP。
   1. 提供者pojo+MybatisPlus提供注解@TableName、@TableId
   2. UserMapper接口，继承BaseMapper。单表SQL
   3. 业务层：UserServiceImpl、UserService接口
   4. 提供者控制层：UserController，使用RESTFul形式
   5. 消费者pojo，无需写mp注解；UserFeign接口（1.参数不能为对象，2.@RequestMapping不能拆分 3.@PathVariable(“xxx”)）;对外暴露Controller
   6. Zuul配置映射（路由）