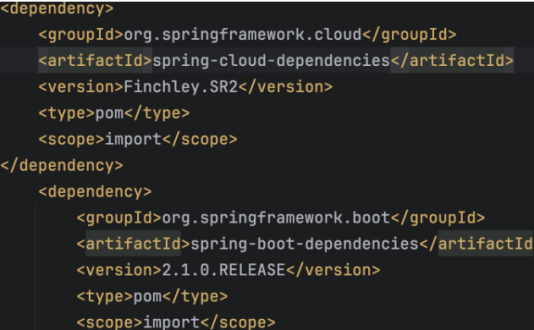
# 初化项目

首先我们的项目是一个springboot的项目，你可以选择sprinbgoot或者自己建立都是可以的，这里比较重要的是pom引入，

通过这个pom，可以规范我们和springboot和cloud的版本是一致的，不会有版本错误，同时这个项目的pom引入十分的干净，不会有很乱的pom存在，大家也可以下载以后自己看看。子模块需要什么就自己去继承，不需要版本号，会继承统一的版本管理。子模块之间互相引用会用到子的所有包，所以越上面的越什么都不用引入。

这里就可以启动项目了，不过你需要注释所有配置和功能。

# 然后我们引入mysql

这里推荐是下载绿色版，也就是不需要安装只是解压使用的这种

下载地址

<https://dev.mysql.com/downloads/mysql/>

请下载8.4.5版本

下载以后解压，然后初始化你的mysql，在bin目录下

mysqld --initialize --console

然后就会生成root的默认密码

记下来，然后安装mysql，同目录下执行

Mysql --install MYSQL8

然后启动net start mysql8

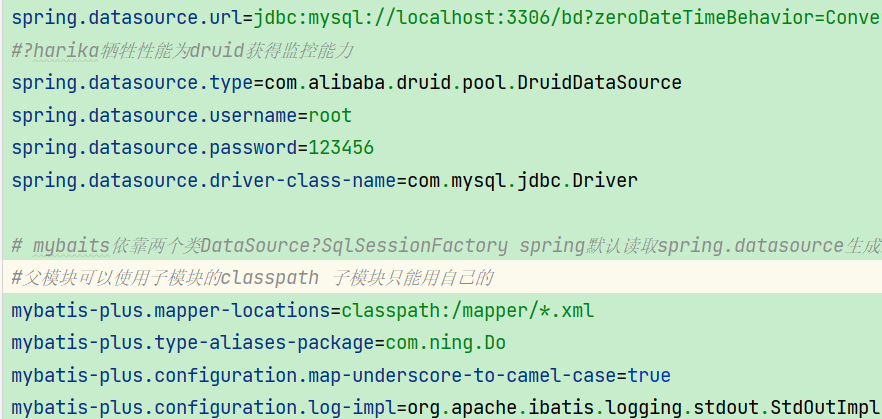
然后登录

Mysql -uroot -p 复制上你刚才的那个密码

登录以后修改密码

set password='123456';

有了mysql以后对项目进行修改，配置中增加数据源配置，启动类增加xml文件扫描即可。



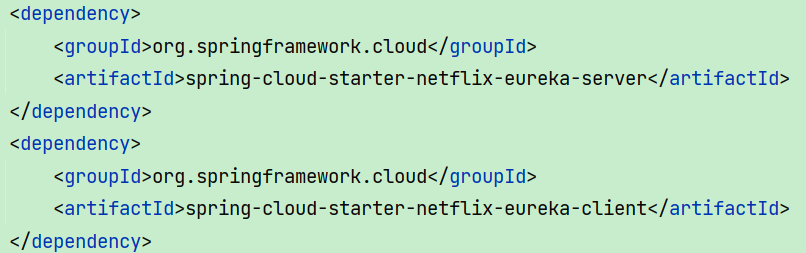


实际使用时会集群部署

# 引入eureka

Eureka没有外部启动，所以我们需要作为server同时也作为client注册上去。

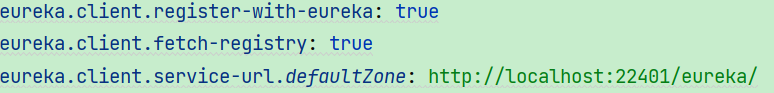
首先引入pom，在api包引入



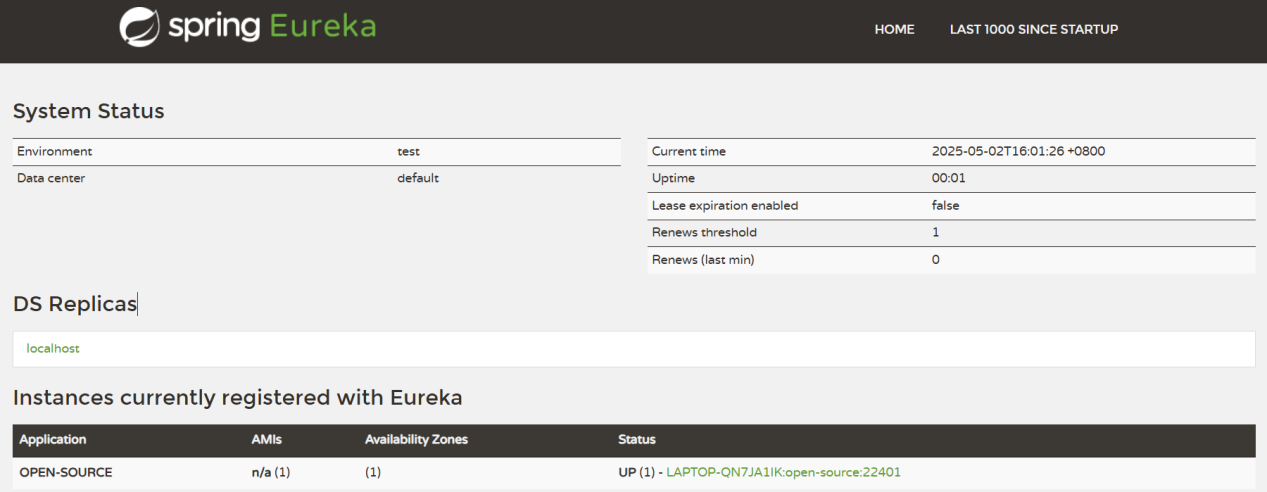
服务端只要在启动类上增加注解即可



客户端则是需要在配置中写明，是否要注册上去，以及注册的地址



成功以后的界面如下，登录UI为我们自己的端口，唯一的服务也是我们注册上去的



和eureka交互，可以使用client



eureka去除实例是通过心跳续约和我们之前一样，定时任务删除

不同的是有保护机制，即大量失效的时候，会停止删除，判断此时网络并不好

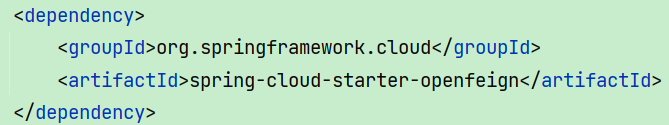
有主动把自己剔除的接口，也可以自己直接调取api如果获取某个服务的地址端口号那些

配置中需要配置你的注册地址在哪里，如果你是服务端，那么端口号就是你的项目启动号

没有密码配置，也就是说如果需要安全验证，需要自己拦截去做

# 引入feign

一个pom即可



然后就是直接使用了



正常我们应该是把第三方的api引入进来，但是这里是我们一个项目，所以只能自己引用自己了

子模块互相依赖确实是不可以的，解决方式是提取出去，然后共同依赖那一个模块

和mq一样可能会有循环调用 需要自己业务避免

唯一需要注意的是 你的服务名字应该和eureka上面的对应上即可

# 引入canal

MySQL 的权限是基于用户名和主机名的组合进行管理的。例如，'root'@'localhost' 和 'root'@'%' 是两个不同的用户。

账号需要权限，同时mysql需要启用binlog并且格式是row的。

默认端口3306

下载deployer，

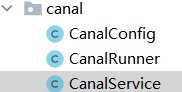
<https://github.com/alibaba/canal/releases?spm=5176.28103460.0.0.2f2e1db83S5glh>

然后启动，默认端口1111，配置需要监听的数据库地址和账号，启动参数修改的小一些

然后是项目中

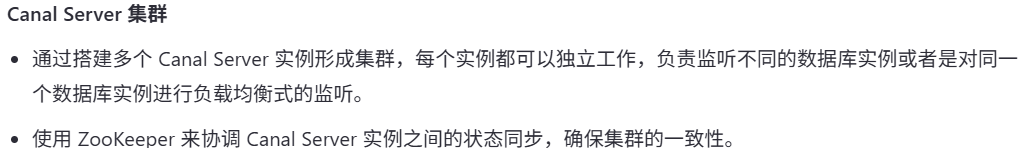


通过这里



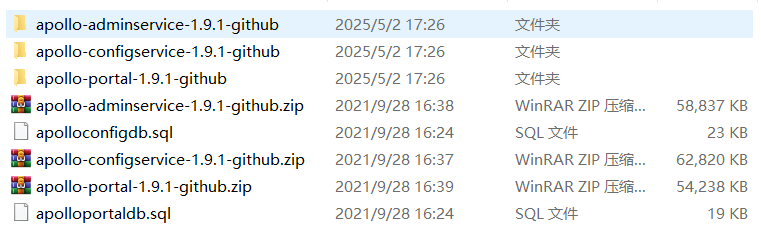
完成canal客户端的初始化，启动监听器，监听数据并处理

实际使用时是集群化部署，一个负责几张表，不会多个负责一个，有顺序问题



# 引入apollo

Apollo是依赖mysql的所以需要执行初始化mysql，同时需要启动三个组件



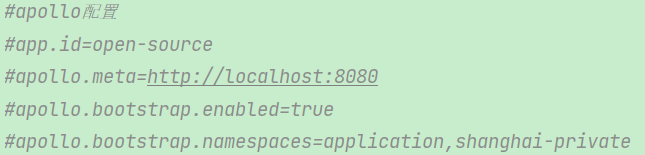
apollo的配置分级如下 环境--集群-namespace

环境就是dev这些，集群应该是按照位置（上海）或者逻辑划分来决定，namespace可以说是最小的单位，和appid绑定，不和集群是绑定。

对于本应用来说创建一个namespace，每个集群都能看到，但是内容是各自独立的。一个应用的公共namespace是可以被其他应用使用的。如果想要用其他应用的配置则是通过去关联那个应用的公共namespace来完成。

加载哪个环境、哪个集群通过jvm启动参数控制,配置的优先级控制是在配置文件中谁在前谁优先，我们也可以通过指定namespace去获取特定的配置

页面端口号8070，springboot连接config组件，默认端口为8080



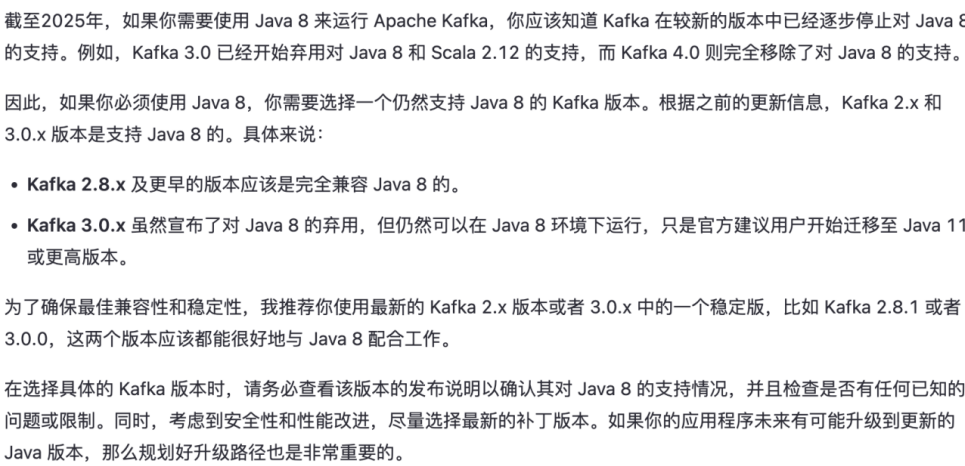




# 引入kafka

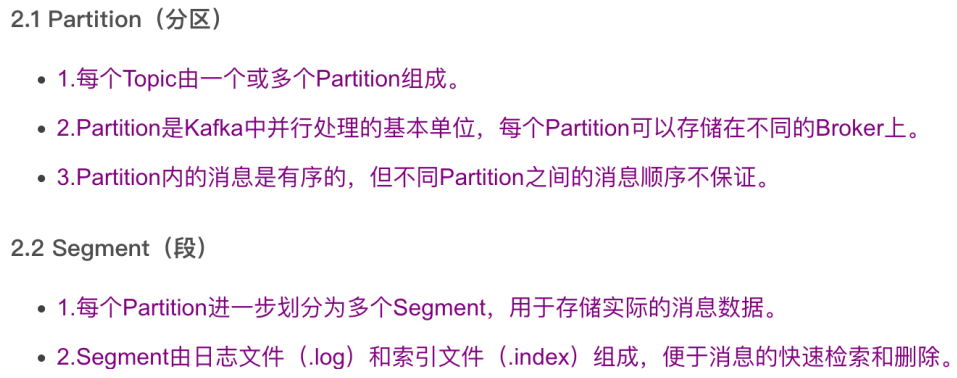
<https://kafka.apache.org/documentation/#quickstart>

Kafka依赖zookeeper，所以在启动时候需要先启动zookeeper



关于消息队列的一些问题：

kafka的分区是保证顺序的，同时kafka的一个分区只能有一个消费者



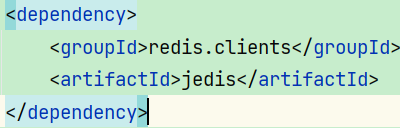
# 引入redis

下载安装redis 5.0.14.1

启动命令

redis-server.exe redis.windows.conf

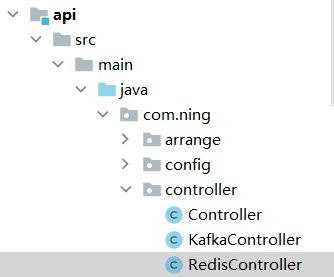
选择使用jedis进行通讯



使用lua 示例在这里



主要解决原子性问题，性能问题，复杂操作问题如获取多值、数据聚合等



Lua示例

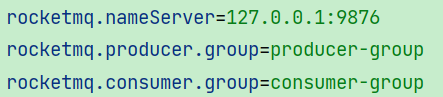
# 引入rocketMQ

下载rocketMQ 4.9.3

引入pom



写入配置



启动mqnamesrv，启动broker

示例位置



实际部署时一般是两个namesrv，多个broker，每个broker都和所有的namesrv进行连接

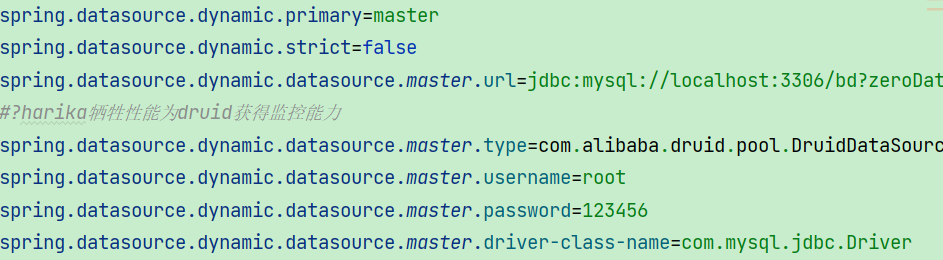
# 多数据源整合

整合多个数据源

一般是生成多个datasource、然后设置sqlSession、共享事务等等



我们这里使用了dynamic-datasource组件



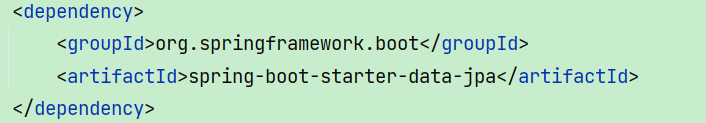
配合注解@DS可以直接开始使用

如果还需要集成es、ck等数据源，则可以通过client的方式来整合

# 整合jpa

经常jpa和mybatis是会一起使用的，jpa完成简单操作，mybatis完成复杂sql，又或者是历史原因等等

引入pom



配置类





配置中的内容请根据自己的实际路径进行修改，然后就可以进行使用了

# 整合es

你来补充

# 整合k8s

你来补充