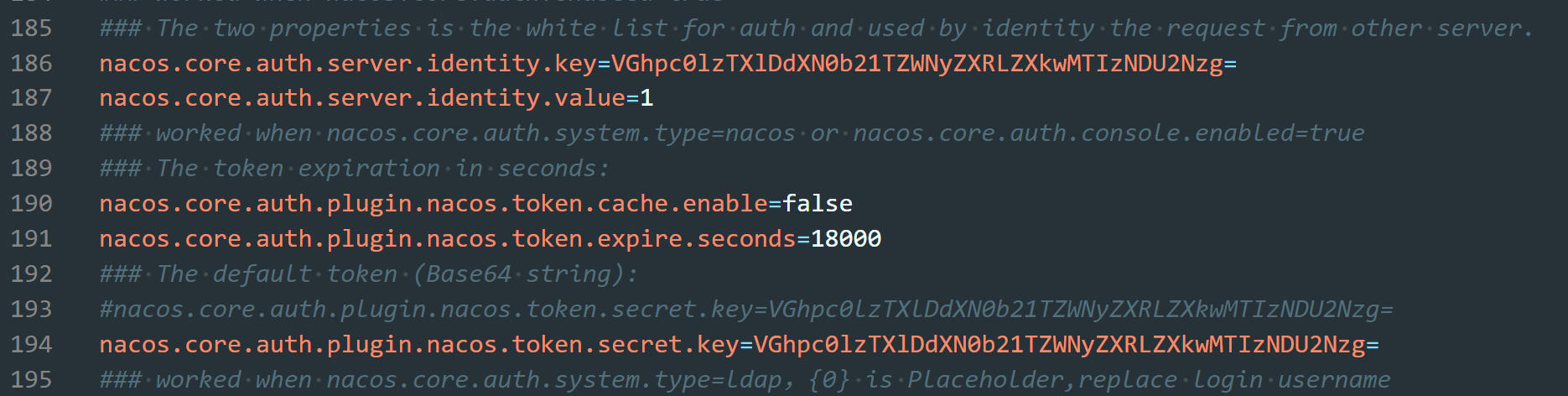
1. 根据教程安装配置nacos

[Nacos 安装教程（史上最详细保姆级教程）\_nacos安装-CSDN博客](https://blog.csdn.net/weixin_62079735/article/details/138094336)

1. 打开conf/application.properties,修改以下地方：



（取消掉两个key的注释）

cd /d D:\nacos\nacos-server-3.0.0\nacos\bin

1. 启动nacos：进入 bin 目录，打开cmd 输入startup.cmd -m standalone

（这个时候input的地方直接回车就行）

OYaNGR2XHQ

数据库 115.29.227.47 SQLPASSWORD

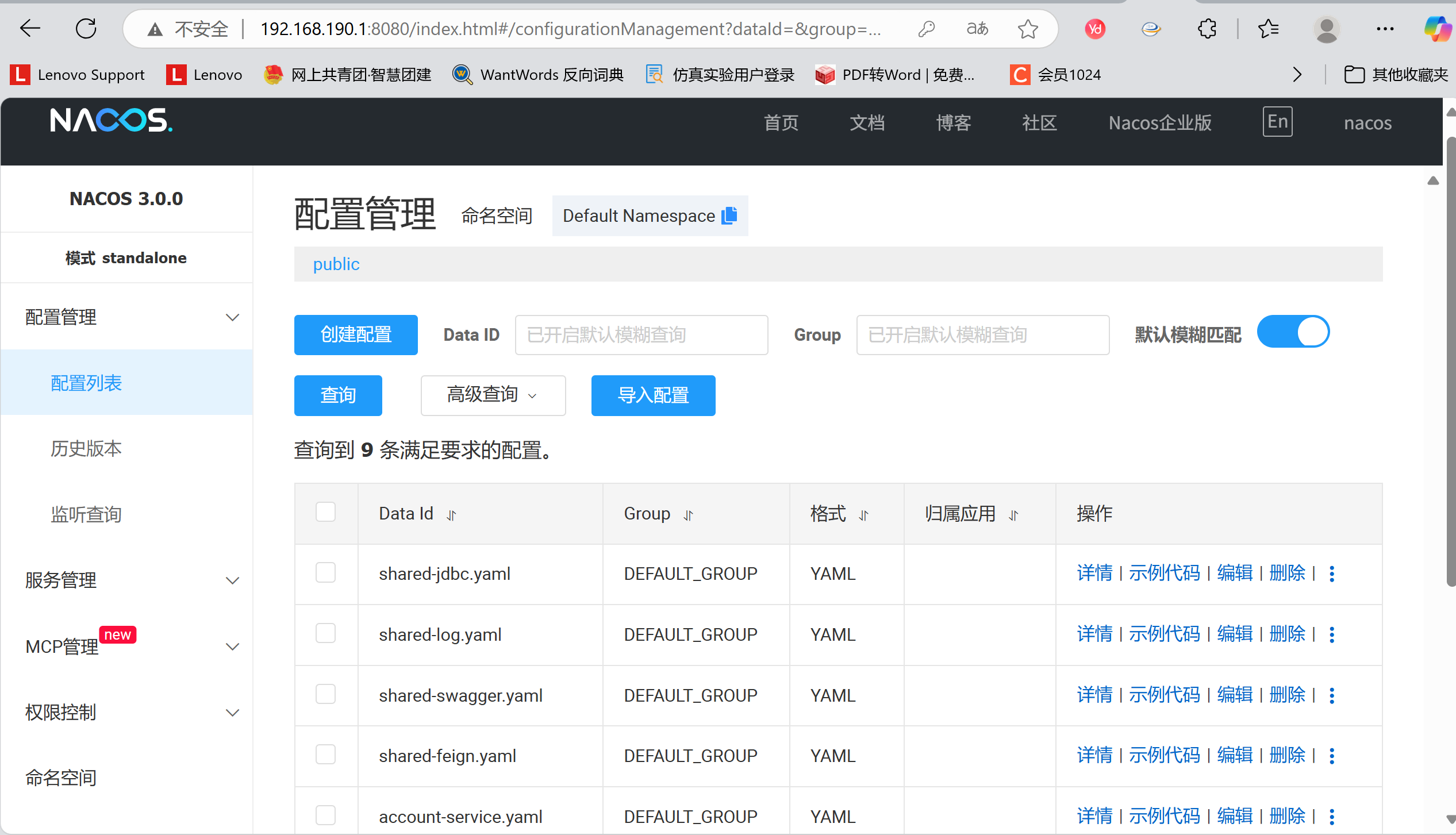
1. 在浏览器打开Console



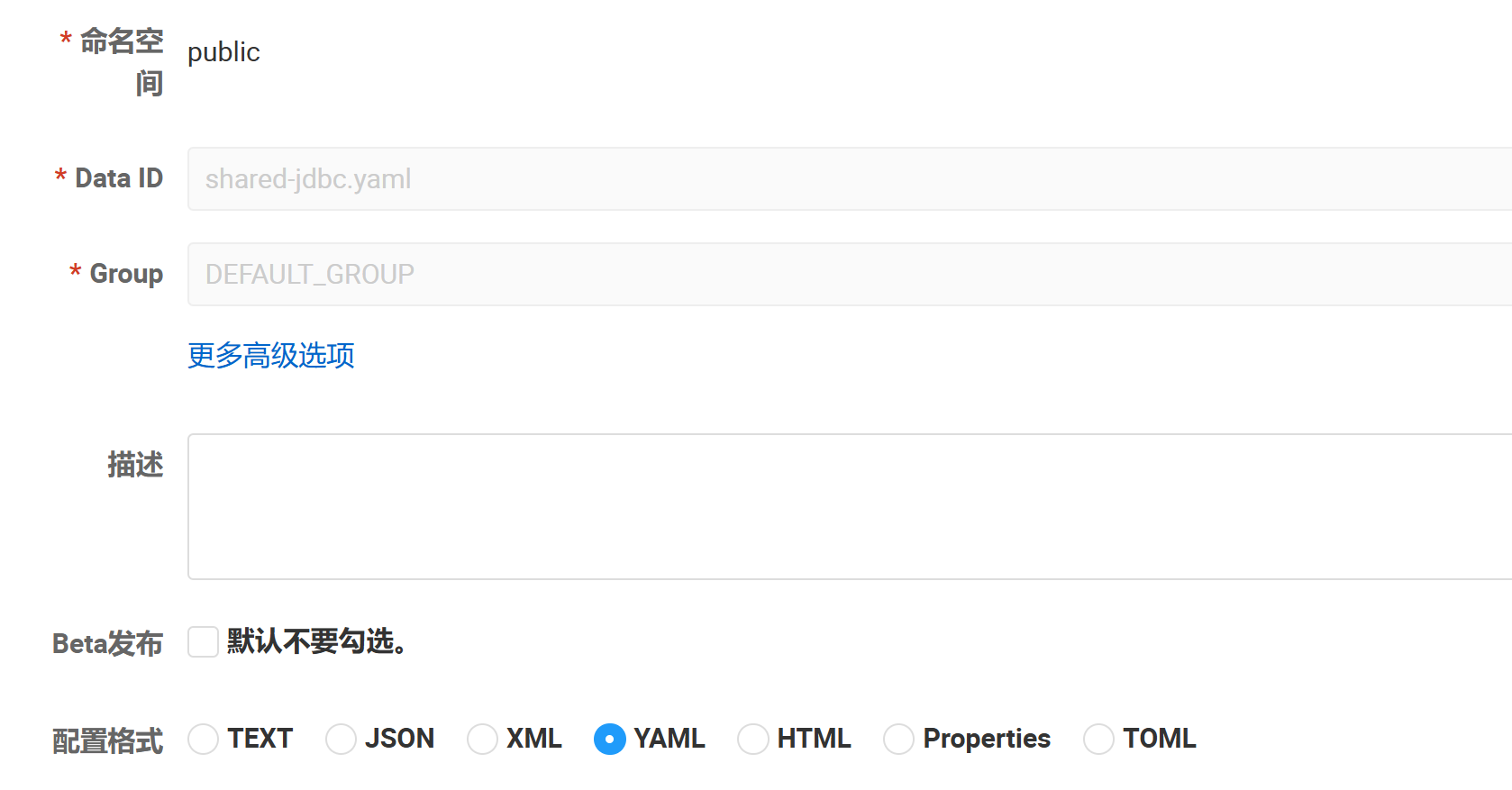
1. 应该会让注册登录，要记住账号和密码
2. 导航到这个页面：



1. 需要添加新的配置：



配置的时候都只改dataID，配置格式为YAML，其他的不变



然后复制粘贴代码到各个yaml

shared-jdbc.yaml:

spring:

datasource:

driver-class-name: com.mysql.cj.jdbc.Driver # MySQL 8+ 的驱动

url: jdbc:mysql://47.111.147.130:3306/

username: root # 替换为您的数据库用户名

password: SQLPASSWORD # 替换为您的数据库密码

# 如果您使用 MyBatis-Plus

mybatis-plus:

mapper-locations: classpath\*:/mapper/\*\*/\*.xml # 检查 mapper 文件路径是否正确

# type-aliases-package: com.plugin.vivien.entity # 指定实体类包路径

configuration:

map-underscore-to-camel-case: true

log-impl: org.apache.ibatis.logging.stdout.StdOutImpl # 开发时可以开启 SQL 日志打印

shared-log.yaml:

logging:

level:

# 设置根日志级别为 INFO

root: info

# 设置项目自身包的日志级别为 DEBUG

com.plugin.vivien: debug

file:

# 指定日志文件输出路径，./logs 表示当前服务运行目录下的 logs 文件夹

path: ./logs

# 可以定义日志输出格式等，例如：

# pattern:

# console: "%d{yyyy-MM-dd HH:mm:ss.SSS} [%thread] %-5level %logger{50} - %msg%n"

# file: "%d{yyyy-MM-dd HH:mm:ss.SSS} [%thread] %-5level %logger{50} - %msg%n"

# Logback specific configuration (if needed)

# logging:

# config: classpath:logback-spring.xml

shared-swagger.yaml:

# Knife4j (Swagger 增强 UI) 配置

knife4j:

# 开启 Knife4j 功能

enable: true

# 配置 Swagger 文档基本信息 (也可以在代码中使用 @ApiInfoBuilder 配置)

# openapi:

# title: "WiseInvest API 文档"

# description: "WiseInvest 微服务项目接口文档"

# version: "1.0"

# contact:

# name: "Your Name"

# url: "http://example.com"

# email: "your.email@example.com"

# terms-of-service-url: "http://example.com/terms/"

# license: "Apache 2.0"

# license-url: "https://www.apache.org/licenses/LICENSE-2.0.html"

# Springfox (原生 Swagger) 配置 (如果未使用 Knife4j 或需要同时配置)

# springfox:

# documentation:

# swagger-ui:

# enabled: true # 是否开启 Swagger UI

# swagger:

# v2:

# path: /v2/api-docs # API Json 路径

# # 可以定义分组等

# # docket:

# # base-package: com.plugin.vivien # 指定扫描的包

Shared-feign.yaml:

feign:

# 启用 OkHttp 作为 Feign 的 HTTP 客户端

okhttp:

enabled: true

# 启用 HttpClient 作为 Feign 的 HTTP 客户端 (如果不用 OkHttp)

# httpclient:

# enabled: false

# max-connections: 200

# max-connections-per-route: 50

client:

config:

# default 应用于所有 Feign 客户端的默认配置

default:

# 连接超时时间，单位毫秒

connectTimeout: 5000

# 读取超时时间，单位毫秒

readTimeout: 5000

# Feign 的日志级别 (NONE, BASIC, HEADERS, FULL)

loggerLevel: basic

# 开启 Feign 对 Sentinel 的整合 (如果使用了 Sentinel)

# sentinel:

# enabled: true

# 开启 Gzip 压缩

# compression:

# request:

# enabled: true

# mime-types: text/xml,application/xml,application/json # 指定压缩的 MIME 类型

# min-request-size: 2048 # 最小压缩大小阈值

# response:

# enabled: true

account-service.yaml:

# Nacos 中的 account-service.yaml

spring:

datasource:

url: jdbc:mysql://47.111.147.130:3306/fts-account

username: root

password: SQLPASSWORD

product-service.yaml:

spring:

datasource:

url: jdbc:mysql://47.111.147.130:3306/fts-product

username: root

password: SQLPASSWORD

settle-service.yaml:

spring:

datasource:

url: jdbc:mysql://47.111.147.130:3306/fts-settle

username: root

password: SQLPASSWORD

transaction-service.yaml:

spring:

datasource:

url: jdbc:mysql://47.111.147.130:3306/fts-transaction

username: root

password: SQLPASSWORD

gateway-routes.json:

[

{

"id": "account-service-route",

"uri": "lb://account-service",

"predicates": [

{

"name": "Path",

"args": {

"\_genkey\_0": "/account/\*\*"

}

}

],

"filters": [

{

"name": "StripPrefix",

"args": {

"\_genkey\_0": "1"

}

}

]

},

{

"id": "product-service-route",

"uri": "lb://product-service",

"predicates": [

{

"name": "Path",

"args": {

"\_genkey\_0": "/product/\*\*"

}

}

],

"filters": [

{

"name": "StripPrefix",

"args": {

"\_genkey\_0": "1"

}

}

]

},

{

"id": "transaction-service-route",

"uri": "lb://transaction-service",

"predicates": [

{

"name": "Path",

"args": {

"\_genkey\_0": "/transaction/\*\*"

}

}

],

"filters": [

{

"name": "StripPrefix",

"args": {

"\_genkey\_0": "1"

}

}

]

},

{

"id": "settle-service-route",

"uri": "lb://settle-service",

"predicates": [

{

"name": "Path",

"args": {

"\_genkey\_0": "/settle/\*\*"

}

}

],

"filters": [

{

"name": "StripPrefix",

"args": {

"\_genkey\_0": "1"

}

}

]

}

]

配置好了之后可以运行项目，连接数据库，单独运行各个微服务