

新Eureka系统建设指南

- 想拆分大系统oms ?
- 要使用GM-TOOLS ?
- 需要升级基础框架(SpringBoot、Cloud等) 版本?
- 自动生成案例项目 ?
- 自动生成基础代码 ?

没错, 本文将带你完成以上全部事项。

〇、名词解释

名词	解释	举例
系统	系统是多个项目共用的技术体系	eurake: 采用SpringCloud体系架构的系统; nacos: 采用SpringCloud Alibaba体系架构的系统;
项目	项目是多个微服务组成的业务工程	oms: eurake系统下的代表项目; pos: nacos系统下的先驱项目;
模块	模块是业务代码所在的工程, 也称微服务。	oms-crm: oms项目下的客户关系管理服务; pos-oder: pos项目下的订单服务;

一、操作步骤

1.1 新系统建设

此步骤是建立一套完整新系统的操作步骤, 已有系统请从1.2开始阅读..

1. 按照2.1.1操作生成系统
2. 清理示例工程
 - eureka-one、eureka-two
3. 进入系统根目录
4. 按照2.1.2操作生成项目
5. 调整示例工程
 - one-api (需要再重建, 这是定义远程接口的父工程)
 - one-micro1、one-micro2
6. 进入项目根目录
7. 按照2.1.3操作生成模块
8. 调整代码生成器配置: main方法上的变量
9. 执行代码生成器

1.2 新项目建设

此步骤是基于已有系统建设新项目的操作步骤, 已有项目请从1.3开始阅读..

按新系统建设步骤 3-9 即可完成项目建设

1.3 新模块建设

此步骤是基于已有项目建设新模块/微服务的操作步骤, 已有模块只是了解该自动化工程, 请随意!

按新系统建设步骤 6-9 即可完成模块建设

二、自动生成

2.1 项目生成器

- **项目生成器:** 生成一个空系统/项目/微服务, 以下 **x** 参数表示需要调整 (非正式项目除外)。
- -DgroupId: 坐标组, ✓
- -DartifactId: 坐标名, x
- -Dversion: 坐标版本号, x
- -Dpackage: 代码所在包名, x

2.1.1 生成系统代码

执行命令即可完成

```
mvn archetype:generate -DarchetypeCatalog=remote -DinteractiveMode=false -
DarchetypeGroupId=cn.huolala -DarchetypeArtifactId=eureka-sys-demo-archetype -
DarchetypeVersion=1.0.0 -DgroupId=cn.huolala -DartifactId=eureka-sys-demo -
Dversion=1.0.0 -Dpackage=cn.huolala.eureka.sys.demo.one.micro1
```

2.1.2 生成项目代码

执行命令即可完成

注意: 生成项目之前请确保当前在系统根路径下, 或者系统已经部署到私服 (否则报错), 非正式项目建议生成系统。

```
mvn archetype:generate -DarchetypeCatalog=remote -DinteractiveMode=false -
DarchetypeGroupId=cn.huolala -DarchetypeArtifactId=eureka-one-archetype -
DarchetypeVersion=1.0.0 -DgroupId=cn.huolala -DartifactId=eureka-one -
Dversion=1.0.0 -Dpackage=cn.huolala.eureka.sys.demo.one.micro1
```

2.1.3 生成模块代码

执行命令即可完成

注意: 生成模块之前请确保当前在项目根路径下, 或者系统已经部署到私服 (否则报错), 非正式项目建议生成系统。

```
mvn archetype:generate -DarchetypeCatalog=remote -DinteractiveMode=false -
DarchetypeGroupId=cn.huolala -DarchetypeArtifactId=one-micro1-archetype -
DarchetypeVersion=1.0.0 -DgroupId=cn.huolala -DartifactId=one-micro1 -
Dversion=1.0.0 -Dpackage=cn.huolala.eureka.sys.demo.one.micro1
```

2.2 代码生成器

- **代码生成器**: 生成常见Controller、Service、Mapper..

1. 使用开发工具打开生成的项目
2. 进入模块项目的test包
3. 类: CodeGenerator.java
4. 执行main方法

main方法上面有些配置需要根据自己的实际情况进行变更。

5. 按提示完成基础代码生成

三、目录说明

本次目录以生成的新系统代码结构为例, 开发新系统请认真阅读系统目录说明。

```
└─eureka-sys-demo # 系统根目录, demo一般为系统名称: eureka、nacos (也不一定要以注册中心为
名, 只要具有代表性即可)
|   pom.xml # 系统根目录, 要求继承: tools-parent 最新版
|   README.md
|
└─eureka-one # 项目名称, demo含义同上, one是项目名称: oms、pos等
|   |   pom.xml # 项目根目录, 要求继承: eureka-sys-demo; 并且依赖项目所用到的所有技术
|   |   (好处: 开发者不需要关注项目用到哪些技术/框架/组件)。
|   |   README.md
|   |
|   |   └─doc # 项目文档根目录
|   |   |   └─dev # 开发文档、技术摘要..
|   |   |   └─micro # 微服务分布图、微服务职责说明..
|   |   |       POS微服务分布图.jpg
|   |   |   └─sql # 数据库建表语句、更新语句..
|   |   |
|   |   └─one-api # 项目远程服务根目录, one含义同上是项目名称: oms、pos等
|   |   |   |   pom.xml
|   |   |   |   README.md
|   |   |   |
|   |   |   └─one-micro1-api # 微服务1提供的远程接口, 微服务1是生产着, 依赖该模块的服务是消
费者。
|   |   |       |   pom.xml
|   |   |       |
|   |   |       └─one-micro2-api # 微服务2提供的远程接口, 微服务2是生产着, 依赖该模块的服务是消
费者。
|   |   |           |   pom.xml
|   |   |           |
|   |   |           └─one-cite # 微服务one为了兼容旧系统版本、代码以及依赖所编写的冗余代码(用于被微服务
one继承)
|   |   |               |   pom.xml
|   |   |               |
|   |   |               └─one-micro1 # 项目中的一个微服务, one含义同上是项目名称: oms、pos等, micro1是微
服务名称: user、order、store..
```

```

| | | pom.xml # 微服务根目录，要求继承：eureka-one;
| | | README.md
| | |
| | | └─src # 开发包
| | |     └─main
| | |         └─java
| | |             └─cn
| | |                 └─huolala
| | |                     └─eureka # 系统名
| | |                         └─sys
| | |                             └─demo
| | |                                 └─one # 项目名
| | |                                     └─micro1 # 模块名/微服务名
| | |                                         Micro1App.java # 启动类
| | |
| | |     └─resources
| | |         application-dev.properties # 开发环境配置，其他环境复制改名即可
| | |
| | |         application.properties # 主配置：各环境均生效
| | |         bootstrap-dev.properties # 开发环境配置，其他环境复制改名即可
| | |         config.properties # 暂未启用
| | |         logback-dev.xml # 开发环境日志配置，其他环境复制改名即可
| | |         README.md
| | |
| | |     └─test # 测试包
| | |         └─java
| | |             └─cn
| | |                 └─huolala
| | |                     └─eureka
| | |                         └─sys
| | |                             └─demo
| | |                                 └─one
| | |                                     └─micro1
| | |                                         CodeGenerator.java # 恭喜，你成功
| | |
| | |         └─resources
| | |             └─templates # 自定义模板目录（无需关注）
| | |                 customController.java.ftl
| | |                 customService.java.ftl
| | |                 customServiceImpl.java.ftl
| | |                 customVo.java.ftl
| | |
| | |     └─one-micro2 # 项目中的一个微服务，one含义同上是项目名称：oms、pos等，micro1是微
| | |         服务名称：user、order、store..
| | |             pom.xml
| | |
| | |     └─demo-two # 项目名称，demo含义同上，one是项目名称：oms、pos等
| | |         pom.xml
| | |
| | |     └─doc # 系统文档
| | |         └─system # 系统架构图、系统说明..
| | |             Nacos系统架构图示例.jpg
| | |         └─middleware # 中间件说明，部署手册..
| | |
| | |     └─middleware # 中间件项目，有些中间件需要开发系统的人自己拉取源码进行编译，甚至进行修改
| | |         和配置：代表性项目：dubbo..
| | |             pom.xml

```

可

的找到了基础代码生成器

```
|
|─middleware-one # 系统中的一个中间件根目录
|   pom.xml
|
|─middleware-two # 系统中的一个中间件根目录
|   pom.xml
```