[](https://www.csdn.net/)

* [博客](https://blog.csdn.net/)
* [课程](https://edu.csdn.net/)
* [文库·商城](https://mall.csdn.net/)
* [问答](https://ask.csdn.net/)
* [社区](https://bbs.csdn.net/)
* [插件](file:///E:\Document\%E9%BB%91%E9%A9%ACSpringCloud\%20https:\plugin.csdn.net\?utm_source=toolbar)
* [认证](https://ac.csdn.net/landing1/?utm_source=crz_pc_toolbar1)
* [开源](https://git.csdn.net/)

 搜索

[登录/注册](https://passport.csdn.net/account/login)

[会员中心 [](https://mall.csdn.net/vip)](https://mall.csdn.net/vip)

[收藏](https://i.csdn.net/#/user-center/collection-list?type=1)

[动态](https://blink.csdn.net)

[创作](https://mp.csdn.net)

* [博客](https://blog.csdn.net/)
* [课程](https://edu.csdn.net/)
* [文库·商城](https://mall.csdn.net/)
* [问答](https://ask.csdn.net/)
* [社区](https://bbs.csdn.net/)
* [插件](https://plugin.csdn.net/?utm_source=toolbar)
* [认证](https://ac.csdn.net/landing1/?utm_source=crz_pc_toolbar1)
* [开源](https://git.csdn.net/)

 搜索

 搜索

[会员中心](https://mall.csdn.net/vip)

[收藏](https://i.csdn.net/#/user-center/collection-list?type=1)

[动态](https://blink.csdn.net/)

[消息](https://passport.csdn.net/account/login)

[创作](https://mp.csdn.net/)

**黑马2021最新版 SpringCloud基础篇全技术栈导学（RabbitMQ+Docker+Redis+搜索+分布式）**

置顶 [@大吉](https://blog.csdn.net/weixin_44757863) 2021-10-28 14:55:53  2297  收藏 20

分类专栏： [微服务](https://blog.csdn.net/weixin_44757863/category_11325074.html) [SpringBoot](https://blog.csdn.net/weixin_44757863/category_10495455.html) 文章标签： [spring cloud](https://so.csdn.net/so/search/s.do?q=spring%20cloud&t=blog&o=vip&s=&l=&f=&viparticle=) [黑马](https://so.csdn.net/so/search/s.do?q=%E9%BB%91%E9%A9%AC&t=blog&o=vip&s=&l=&f=&viparticle=) [java](https://so.csdn.net/so/search/s.do?q=java&t=blog&o=vip&s=&l=&f=&viparticle=)

版权

[微服务](https://blog.csdn.net/weixin_44757863/category_11325074.html)同时被 2 个专栏收录

2 篇文章1 订阅

订阅专栏

[SpringBoot](https://blog.csdn.net/weixin_44757863/category_10495455.html)

9 篇文章1 订阅

订阅专栏

**前言**

**基础篇全部代码和资料已上传到gitee，大家需要可自取:** <https://gitee.com/da-ji/spring-cloud-learning-notes>  
**点个Star，后续更新高级篇和面试篇不迷路** ⚆\_⚆

本笔记基于：  
1、 [尚硅谷 2020.3 *SpringCloud*(H版&alibaba)框架开发教程](https://www.bilibili.com/video/BV18E411x7eT?from=search&seid=14890308186329670911)  
2、 [黑马 2021.8 SpringCloud\*+RabbitMQ+Docker+Redis+搜索+分布式](https://www.bilibili.com/video/BV1LQ4y12%E4%BB%B7%E6%A0%BC7n4?from=search&seid=10991461656716219000)

代码和资料基于：  
[黑马 2021.8 SpringCloud\*+RabbitMQ+Docker+Redis+搜索+分布式](https://www.bilibili.com/video/BV1LQ4y12%E4%BB%B7%E6%A0%BC7n4?from=search&seid=10991461656716219000)

第一套教程是经典的尚硅谷阳哥的教程，好处是经过时间的沉淀，已经非常成熟，网上大神的笔记也多，只要是人类出现的问题网上一搜都有答案；非常适合自学。

第二套教程是黑马程序员的2021年8月份最新版教程，截止到发稿时应该是**全网最最新的教程**，在计算机技术日新月异的今天，尽可能往前学最新的技术至少没错。**而且该套教程有一个特点在于，将课程分为实用篇和高级篇：**

* 实用篇基本上以微课堂的形式出现，**平均视频时长也就10分钟左右，易于接受**，涵盖了80%开箱上手就能用的知识；
* 而高级篇就比较深入和复杂了，为应对企业的复杂工作设计，**每个视频长度都为一小时左右**，同时也是面试常问的地方。

由于本人已经工作了，为了在工作中快速拿起来就能用，我选择的学习路线是：先刷黑马程序员的实用篇，以最少的时间快速掌握SpringCloud的相关知识，然后视情况而定深入学尚硅谷的教程或是黑马程序员的高级篇。

最后，这两篇教程虽然都非常好，但是都没有推出面试篇(源码深入讲解)，如果大家经济上允许，可以支持一波培训机构内部课程；经济不允许也可以自学，当然我也会在博客和[Gitee](https://so.csdn.net/so/search?q=Gitee&spm=1001.2101.3001.7020)中陆续更新一些更高深的技术。

**为方便大家速查，后文中这种颜色的字体，代表知识点在**[**对应模块代码**](https://gitee.com/da-ji/spring-cloud-learning-notes)**中的位置**  
**为方便大家速查，后文中这种颜色的字体，代表知识点在**[**对应模块代码**](https://gitee.com/da-ji/spring-cloud-learning-notes)**中的位置**  
**为方便大家速查，后文中这种颜色的字体，代表知识点在**[**对应模块代码**](https://gitee.com/da-ji/spring-cloud-learning-notes)**中的位置**

**课程资料链接：https://pan.baidu.com/s/169SFtYEvel44hRJhmFTRTQ 提取码：1234**  
**课程资料链接：https://pan.baidu.com/s/169SFtYEvel44hRJhmFTRTQ 提取码：1234**  
**课程资料链接：https://pan.baidu.com/s/169SFtYEvel44hRJhmFTRTQ 提取码：1234**

**目录**

* [前言](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#_0)
* [一、微服务技术栈导学](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#_35)
* [二、Dubbo&Zookeeper](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#DubboZookeeper_58)
* [三、微服务远程调用Demo——RestTemplate基本使用](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#DemoRestTemplate_61)
* [四、Eureka注册中心](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#Eureka_73)
* [五、Ribbon负载均衡](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#Ribbon_109)
* [六、Nacos注册中心](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#Nacos_149)
  + [6.1 安装启动](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#61__153)
  + [6.2 Nacos自定义负载均衡策略](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#62_Nacos_173)
  + [6.3 Nacos实现配置热更新](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#63_Nacos_208)
  + [6.4 Nacos集群](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#64_Nacos_222)
* [七、Feign远程调用](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#Feign_243)
  + [7.1 还原事故现场](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#71__250)
  + [7.2 Feign自定义配置](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#72_Feign_269)
  + [7.3 Feign性能优化](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#73_Feign_278)
  + [7.4 Feign最佳实践](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#74_Feign_310)
* [八、统一Gateway网关](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#Gateway_330)
  + [8.1 概述](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#81__332)
  + [8.2 搭建网关服务](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#82__343)
  + [8.3 路由过滤](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#83__383)
    - [8.3.1 断言工厂：对请求进行过滤](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#831__384)
    - [8.3.2 过滤器GatewayFilter：对请求和响应进行过滤](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#832_GatewayFilter_391)
    - [8.3.3 全局过滤器GlobalFilter：可以自定义过滤逻辑代码实现](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#833_GlobalFilter_404)
    - [8.3.4 过滤器链执行顺序](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#834__420)
  + [8.4 网关跨域问题处理](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#84__426)
* [九、Docker](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#Docker_466)
  + [9.1 Docker概念](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#91_Docker_469)
  + [9.2 Docker常用命令](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#92_Docker_472)
* [十、MQ(Message Queue)消息队列](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#MQMessage_Queue_477)
  + [10.1 概述](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#101__478)
  + [10.2 RabbitMQ安装](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#102_RabbitMQ_500)
  + [10.3 常见消息模型](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#103__538)
    - [10.3.1 简单队列模型](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#1031__540)
  + [10.4 Spring AMQP](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#104_Spring_AMQP_545)
    - [10.3.2 WorkQueue模型](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#1032_WorkQueue_578)
    - [10.3.3 发布-订阅模型](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#1033__588)
      * [A. Fanout Exchange](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#A_Fanout_Exchange_600)
      * [B. Direct Exchange](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#B_Direct_Exchange_632)
      * [C. Topic Exchange](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#C_Topic_Exchange_670)
    - [10.3.4 消息转换器](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#1034__683)
* [十一、ElasticSearch分布式搜索](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#ElasticSearch_733)
  + [11.1 ES基础概念](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#111_ES_735)
  + [11.2 安装部署ES](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#112_ES_777)
  + [[Debug] 停止ES容器（或是重启Linux）后，如何恢复Docker网络：](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#Debug_ESLinuxDocker_832)
  + [11.3 索引库操作](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#113__835)
  + [11.4 文档操作](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#114__908)
  + [11.5 RestClient操作索引库和文档](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#115_RestClient_977)
  + [11.6 DSL查询语法](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#116_DSL_1064)
  + [11.7 Java RestClient查询语法](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#117_Java_RestClient_1347)
  + [11.8 ES综合案例：黑马旅游](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#118_ES_1358)
  + [11.9 ES数据聚合](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#119_ES_1398)
  + [11.10 ES数据补全](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#1110_ES_1479)
  + [11.11 ES与MySQL之间数据同步（面试常问）](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#1111_ESMySQL_1560)
  + [11.12 搭建高可用ES集群](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#1112_ES_1587)
* [后记](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#_1619)

**一、微服务**[技术栈](https://so.csdn.net/so/search?q=%E6%8A%80%E6%9C%AF%E6%A0%88&spm=1001.2101.3001.7020)**导学**

从单体架构过度到微服务架构，需要一系列中间技术支撑，其中重要的部分包括：

* 注册中心：Eureka 、Zookeeper、Nacos
* 服务网关：Zuul 、Gateway
* 微服务远程调用：RestTemplate、Feign
* 容器化技术 Docker
* 消息队列 MQ（多种实现方式）
* 负载均衡 Ribbon 、 Nginx
* 分布式搜索技术：ElasticSearch

**尚硅谷阳哥的SpringCloud版本选型：**

**黑马程序员的SpringCloud版本选型：**

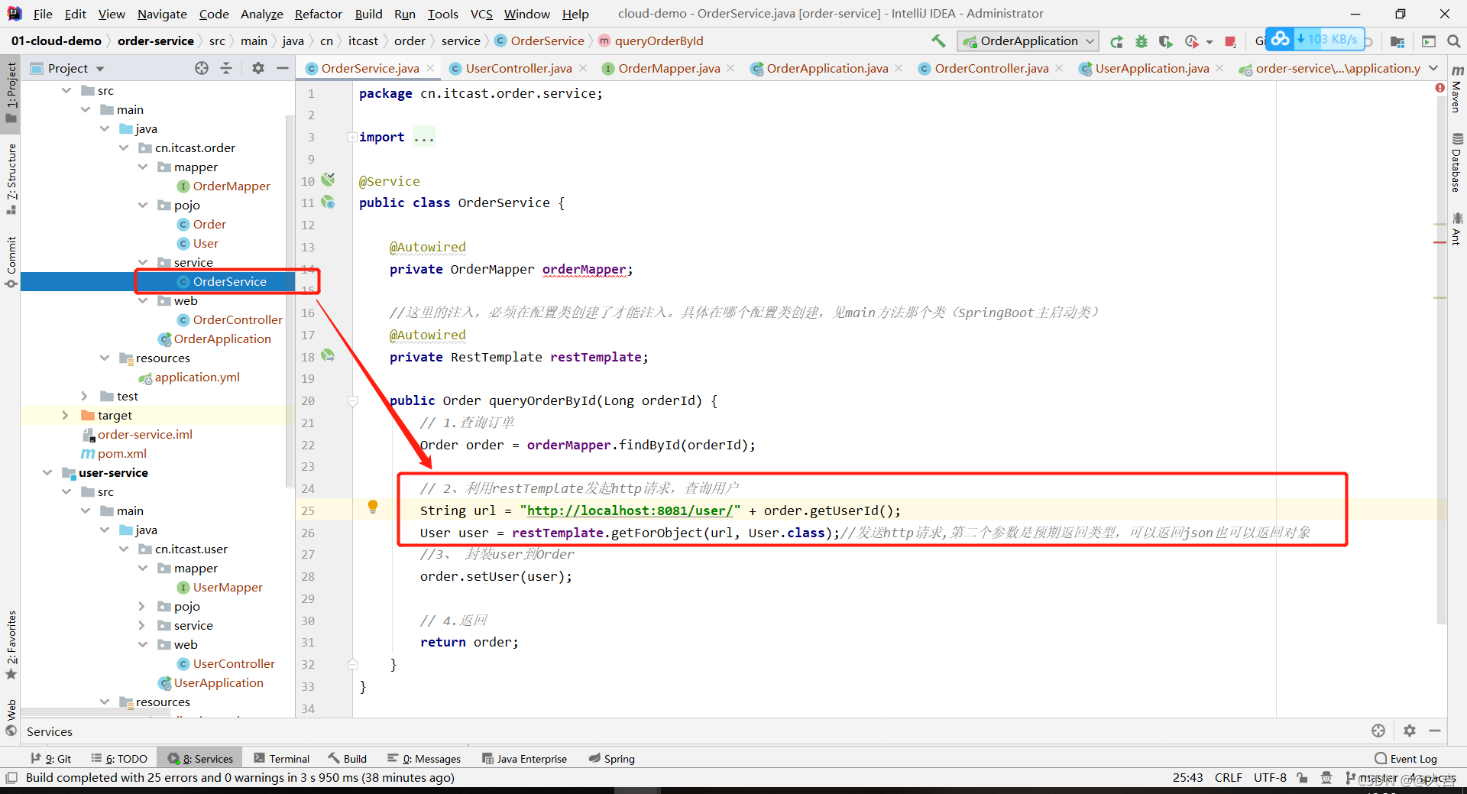
**可以看到，黑马的版本明显较新，本文采用黑马程序员的版本（Hoxton.SR10 + SpringBoot 2.3.x）**

**二、Dubbo&Zookeeper**

**核心代码位置：在模块 dubbo+zookeeper 下**  
这部分是跟狂神说Java学习的（黑马版直接跳过了这两个技术），Zookeeper与Eureka 、Nacos一样也是一种注册中心。

**三、微服务远程调用Demo——RestTemplate基本使用**

**核心代码位置：在模块 01-cloud-demo 下的order-service 和 user-service**

核心代码如下图：实现了跨服务远程调用  


实例

// 1.查询订单  
Order order = orderMapper.findById*(*orderId*)*;  
//2、利用RestTemplate对user\_service服务接口发起http请求，根据order里面的userId查询用户信息  
//2.1 url路径,拼接成完整的请求接口地址。  
String url = "http://userservice:8081/user/" + order.getUserId*()*;  
//发送http请求，实现远程调用  
User user = restTemplate.getForObject*(*url, User.class*)*;  
//3.封装user到order  
order.setUser*(*user*)*;  
// 4.返回  
return order;

关键语句User user = restTemplate.getForObject*(*url, 返回值类型*)*;

总结：RestTemplate微服务调用方式

基于RestTemplate发起的http请求实现远程调用

http请求做远程调用是与语言无关的调用，只要知道对方  
的ip、端口、接口路径、请求参数即可。

**四、Eureka注册中心**

**核心代码位置：在模块 01-cloud-demo 下的eureka-server（注册的是order-service 和 user-service）**

**Eureka的作用：**

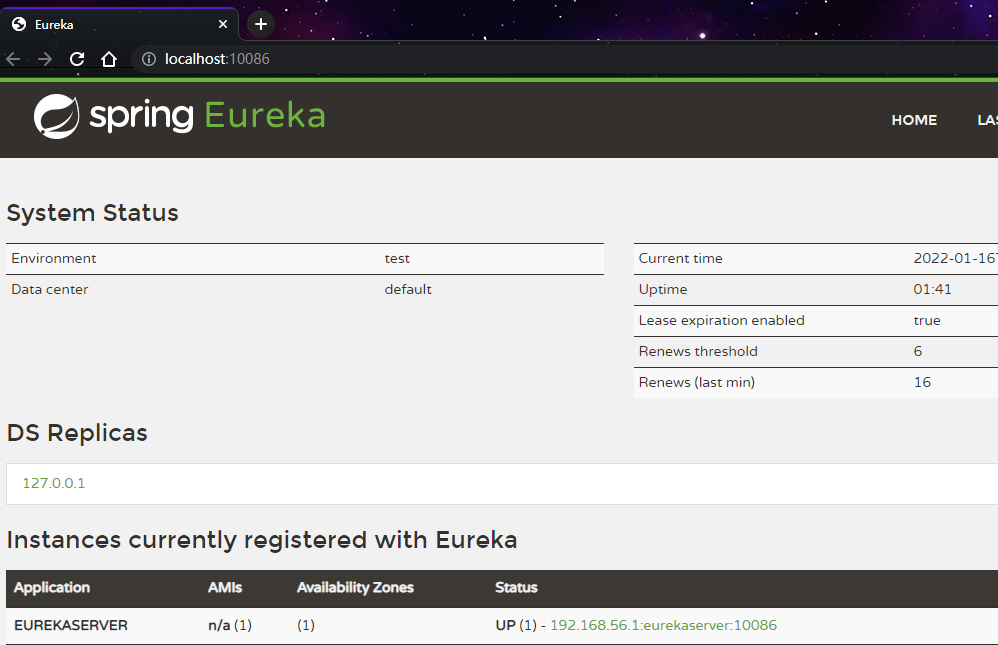
* 消费者该如何获取服务提供者具体信息?
  + 服务提供者启动时向eureka注册自己的信息
  + eureka保存这些信息
  + 消费者根据服务名称向eureka拉取提供者信息
* 如果有多个服务提供者,消费者该如何选择?
  + 服务消费者利用负载均衡算法，从服务列表中挑选一个
* 消费者如何感知服务提供者健康状态?
  + 服务提供者会每隔30秒向EurekaServer发送心跳请求，报告健康状态
  + eureka会更新记录服务列表信息，心跳不正常会被剔除
  + 消费者就可以拉取到最新的信息

**注意点：**

Eureka自己也是一个微服务，Eureka启动时，要把自己也注册进去。这是因为如果后续搭建Eureka集群时做数据交流，yml代码：

server:  
 port: 10086 *#eureka服务端口*spring:  
 application:  
 name: eurekaserver *# eureka的服务名称*eureka:  
 client:  
 service-url: *# eureka的地址信息* defaultZone: http://127.0.0.1:10086/eureka

上段代码块中，defaultZone，将自己也注册进去了。效果如下图：



**Eureka注册中心总结**

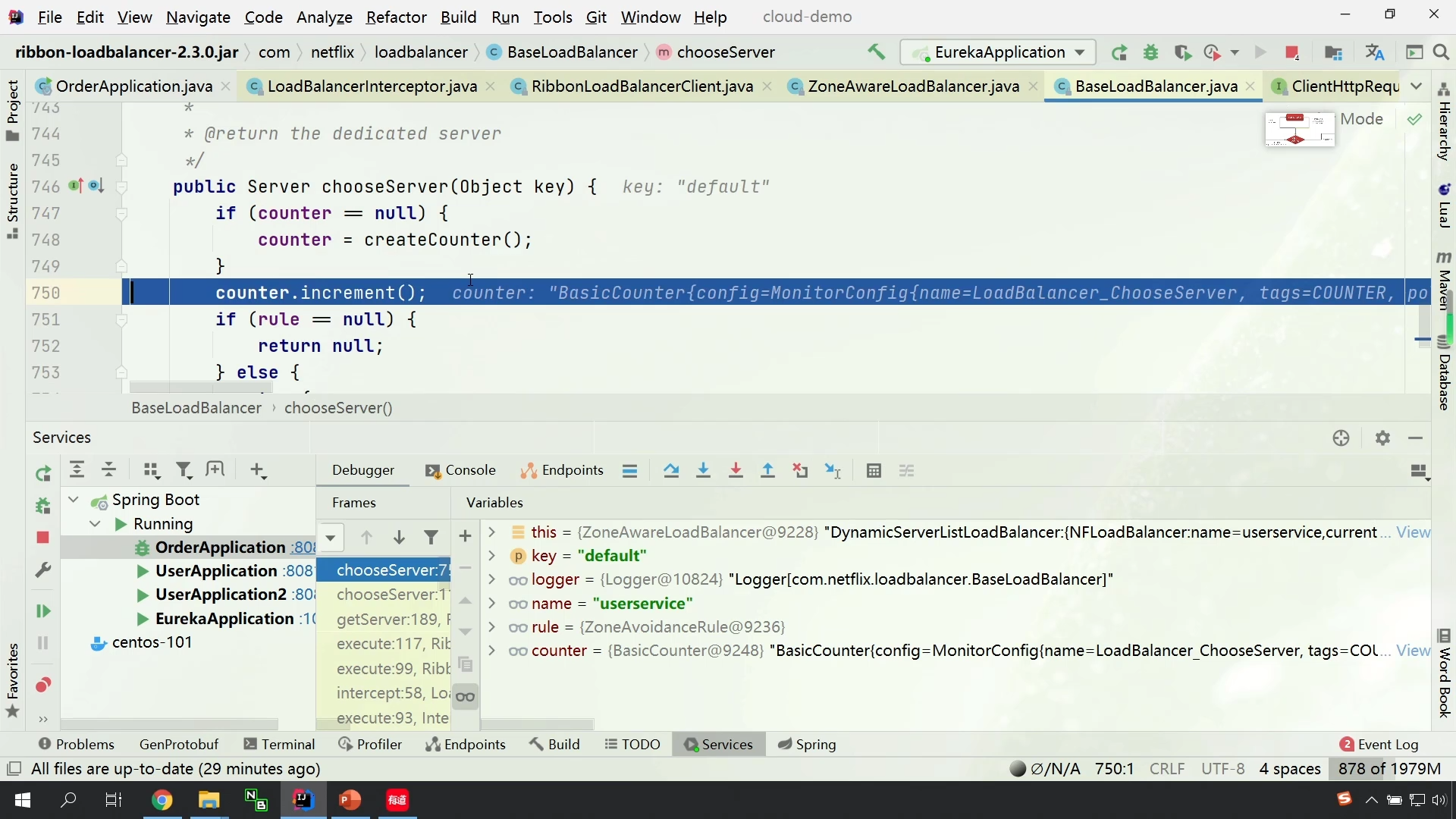
1. 搭建EurekaServer
   * 引入eureka-server依赖
   * 添加@EnableEurekaServer注解
   * 在application.yml中配置eureka地址
2. 服务注册
   * 引入eureka-client依赖
   * 在application.yml中配置eureka地址
3. 服务发现
   * 引入eureka-client依赖
   * 在application.yml中配置eureka地址
   * 给RestTemplate添加@LoadBalanced注解
   * 用服务提供者的服务名称远程调用

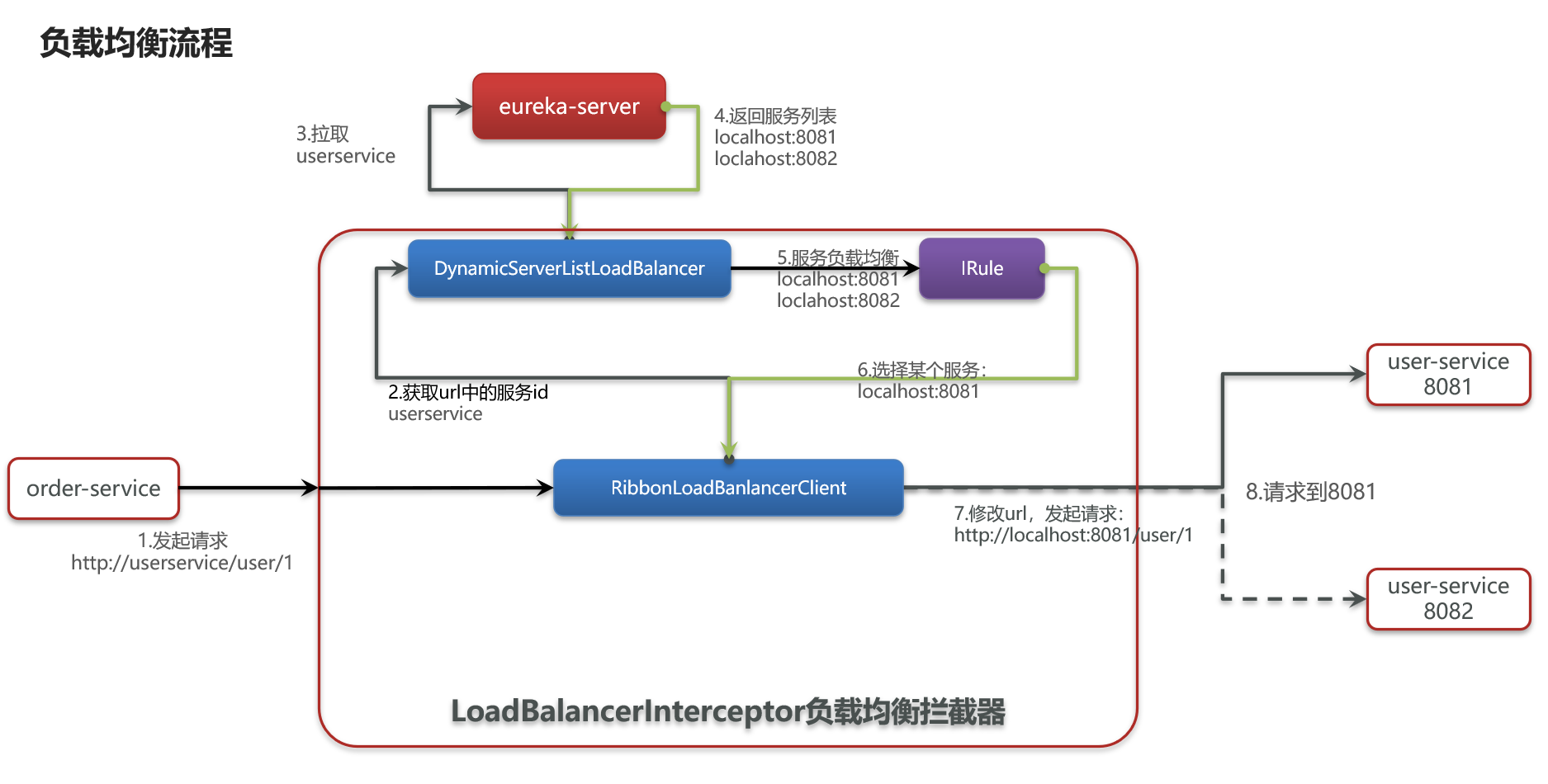
**五、Ribbon负载均衡**

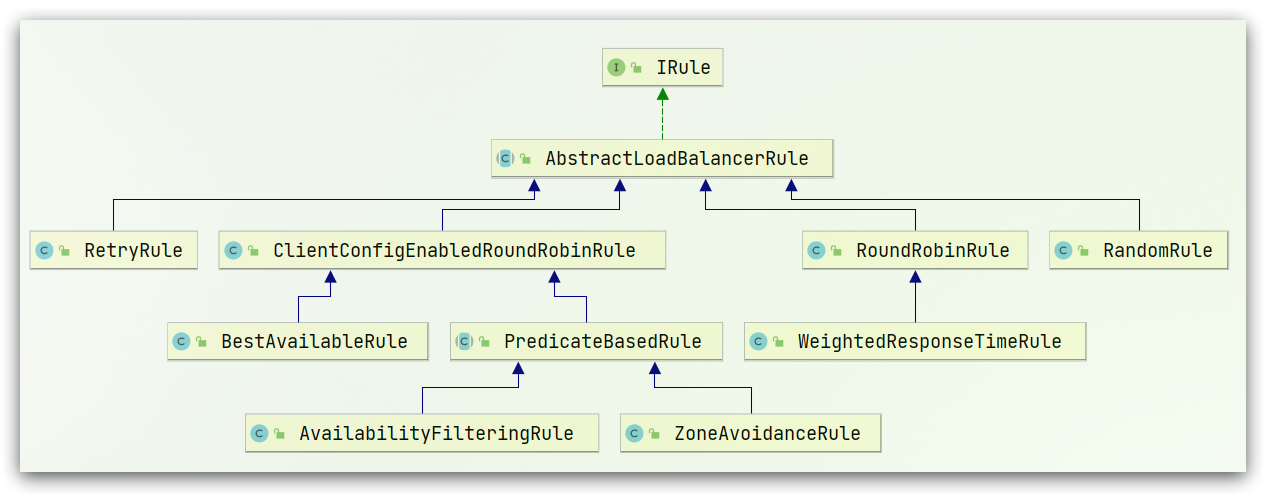
**两个疑问：**

* 如果有多个服务提供者，服务调用者如何知道究竟调用哪个服务呢？
* 而且服务调用者为何不用写死服务提供者的链接（ip和端口），只需要写服务名称即可？为什么我们只输入了服务名称就可以访问了呢？  
  （String url = "http://userservice/user/" + order.getUserId(); //由于已经在Eureka里面配置了服务，这里只需要写配置的服务名即可）

这都是Ribbon的负载均衡做到的，**针对问题一**，通过跟断点得知，Ribbon是通过几种不同的负载均衡算法实现的这一个机制（比如[轮询算法](https://blog.csdn.net/jasonliuvip/article/details/25725541)）；针对问题二，Ribbon会根据服务名称去Eureka注册中心拉取服务，如下两个图所示：

  
**Ribbon 负载均衡策略**





RoundRobin —— 意为轮询，操作系统也有类似的概念（CPU时间片轮转）

可以使用如下代码配置对某个服务的负载均衡策略(在 application.yml里配置)

userservice: # 给某个微服务配置负载均衡规则，这里是userservice服务为例

ribbon:

NFLoadBalancerRuleClassName: com.netflix.loadbalancer.RandomRule # 负载均衡规则

* 1
* 2
* 3
* 4

**Ribbon开启饥饿加载**  
Ribbon默认是采用懒加载，即第一次访问时才会去创建LoadBalanceClient，因此会发现第一次请求时间会比较长。

而饥饿加载则会在项目启动那一刻完成服务加载，降低第一次访问的耗时，通过下面配置开启饥饿加载：

ribbon:

eager-load:

enabled: true # 开启饥饿加载

clients:

- userservice # 指定饥饿加载的服务名称

- xxxxservice # 如果需要指定多个，需要这么写

**六、Nacos注册中心**

和前面的Eureka、Zookeeper是同类型技术，Nacos功能更强大，国内受欢迎

**6.1 安装启动**

下载地址：https://github.com/alibaba/nacos/releases  
本文选用1.4.1版本

解压完成后，cd到nacos的bin目录下，然后输入命令：  
startup.cmd -m standalone

关闭的话，如果是linux系统，就运行shutdown.sh即可

出现如上图所示界面，说明启动成功。通过上图也可知它的默认端口是8848（国人做的注册中心果然不一样 8848氪金手机~）

输入地址http://127.0.0.1:8848/nacos 即可访问主页，用户名和密码都是nacos  
**核心代码位置：在模块 01-cloud-demo 下注册了order-service 和 user-service，同时注释掉了两个模块的Eureka代码（包括pom.xml也注释了，毕竟是同类技术）**

**注意**，必须将之前的Eureka代码和pom都注释掉，而且把SpringCloud也注释掉（因为已经用了SpringCloudAlibaba），否则有可能报：APPLICATION FAILED TO START这个错误

**对比之前的Eureka，我们是在idea里面专门启动了一个Eureka的工程，所以 Eureka不需要下载，就可以通过端口号访问Eureka的注册中心。而Nacos是 下载并运行的，所以不需要在idea启动某个模块，直接通过运行Nacos的startup.cmd即可通过端口号访问Nacos的注册中心。**

**6.2 Nacos负载均衡**

### 自定义负载均衡策略

也是使用的Ribbon，下面一个例子将Nacos配置成同集群优先的负载均衡策略：

默认的ZoneAvoidanceRule并不能实现根据同集群优先来实现负载均衡。

Nacos中提供了一个NacosRule的实现，可以优先从同集群中挑选实例。

1. 修改order-service中的application.yml，设置集群为HZ：

cloud:  
 nacos:  
 server-addr: localhost:8848 *# nacos服务地址* discovery:  
 cluster-name: HZ *#设置集群名字*

2）修改负载均衡规则

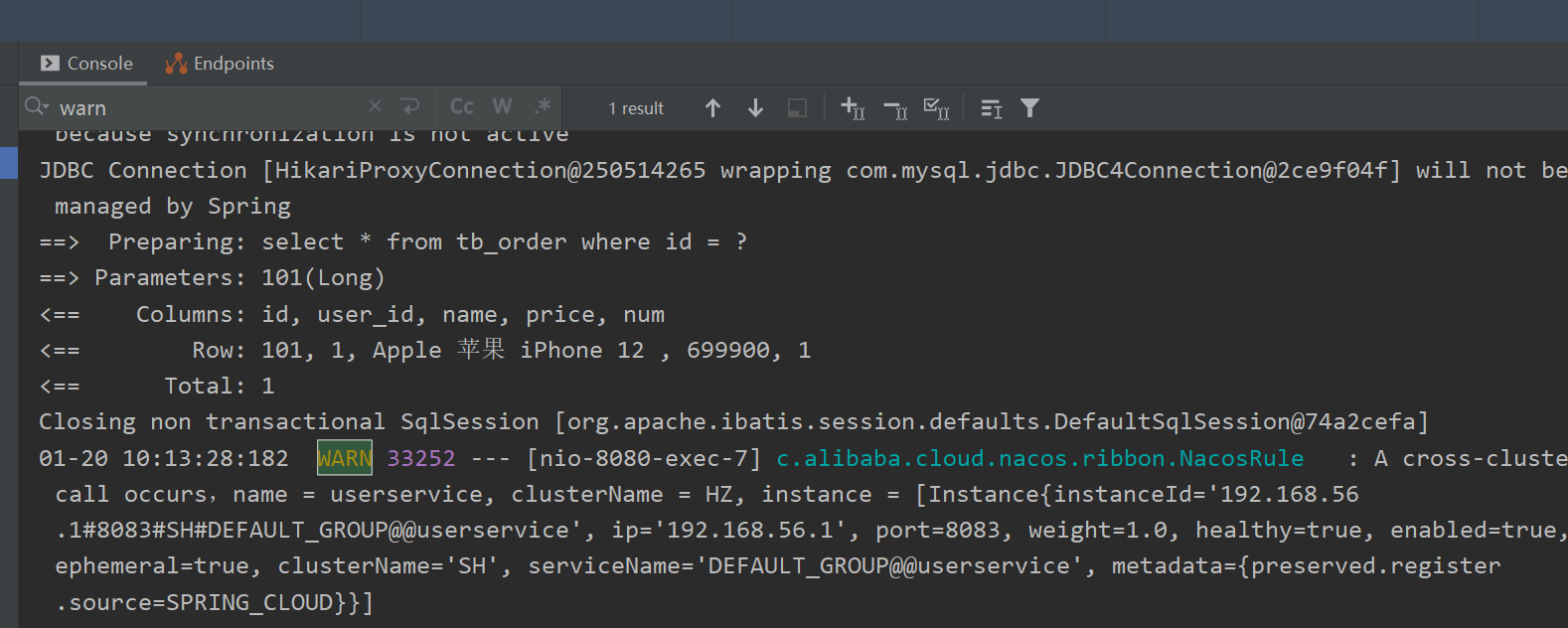
修改order-service的application.yml文件，修改负载均衡规则：

配置完成之后，就可以实现同集群优先的 负载均衡了

在order-service中设置负载均衡的IRule为NacosRule，这个规则优先会寻找与自己同集群的服务

userservice: *# 要做配置的微服务* ribbon:  
 NFLoadBalancerRuleClassName: com.alibaba.cloud.nacos.ribbon.NacosRule *# 负载均衡规则*

3) 如果没找到同集群则会跨集群访问,同时控制台日志会显示WARN提示



总结：实例权限控制

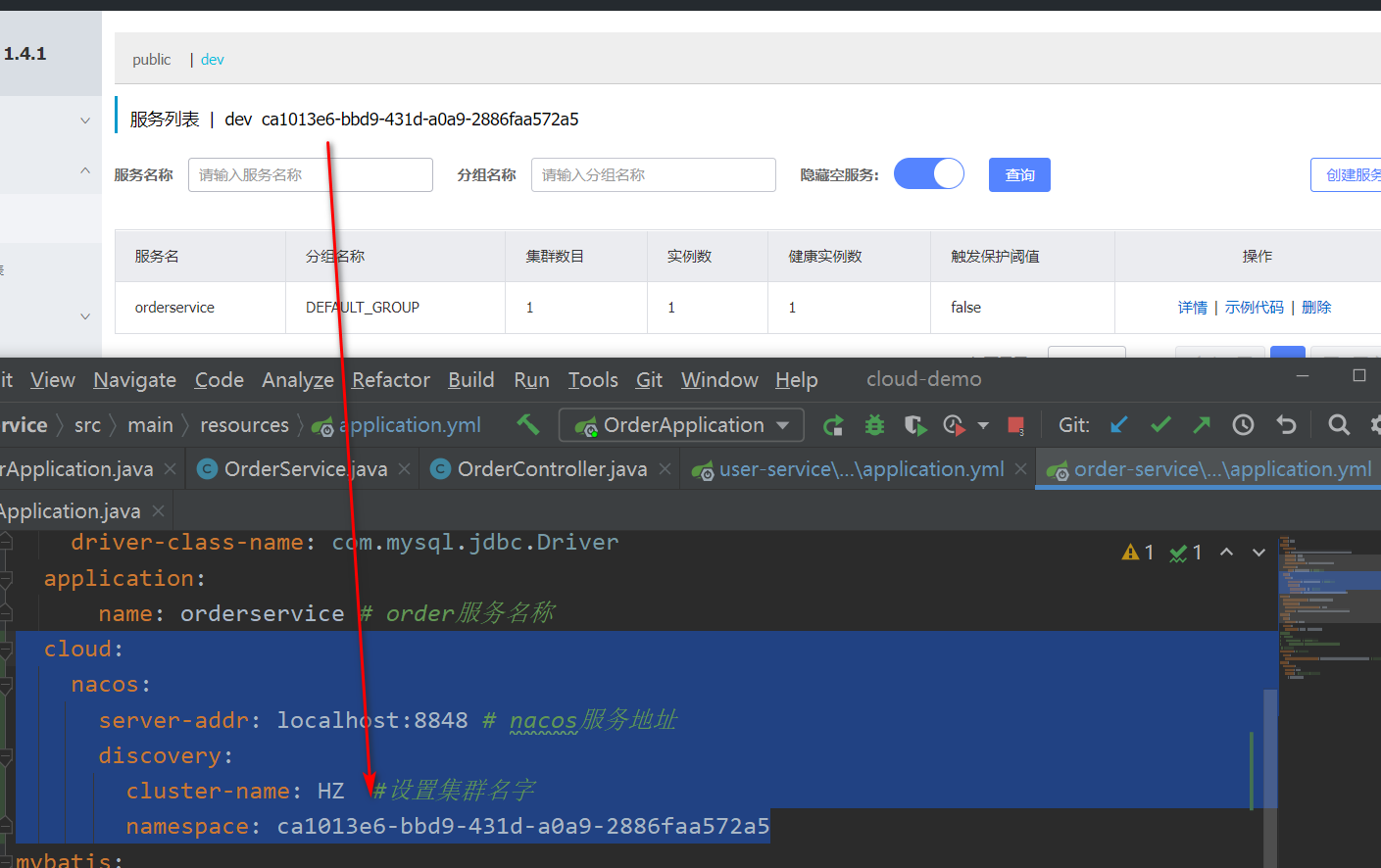
1、Nacos控制台可以设置实例的权重值，0~1之间

2、同集群内的多个实例，权重越高被访问的频率越高

3、权重设置为0则完全不会被访问

### Nacos环境隔离

对应填到yml文件中的namespace中。



1. 总结：Nacos环境隔离
   1. 每个namespace都有唯一id
   2. 服务设置namespace时要写id而不是名称
   3. 不同namespace下的服务互相不可见

否则会报500错误

**6.3 Nacos实现配置热更新**

有两种方式，都在代码中配置了，具体位置在：  
**核心代码位置：在模块 01-cloud-demo 下 user-service，第一种方式是通过配置文件方式(PatternProperties.java)；第二种方式是通过注解@Value("${yaml里定义的键值对}")的方式**

* **热更新注意点：**  
  你在Nacos中配置的：  
  你在bootstrap.yaml里配置的：  
  **这两张图应该是一致的，注意 -和.的区别！！！**
* **热更新优先级**  
  Nacos带环境的配置 > Nacos不带环境的配置 > 本地yaml文件配置  
    
  很好理解，Nacos带环境可以理解为专属化配置(开发环境和生产环境)、肯定优先于Nacos不带环境的全局配置；本地yaml文件配置则肯定低于Nacos的配置。

**6.4 Nacos集群**

**位置：在模块 01-cloud-demo 下根目录，有一个叫Nacos集群搭建.md的文件**

**注意点：修改两个配置文件：**

* 修改cluster.conf
* 修改Nacos的application.properties（不是你的application.properties）

修改完成后保存即可。

* **使用Nginx对Nacos做反向代理**  
  这里需要Nginx前置知识，可以看我以下这一篇文章：[Nginx入门：通俗理解反向代理和负载均衡，简单配置Nginx](https://blog.csdn.net/weixin_44757863/article/details/120117645?spm=1001.2014.3001.5501)

**如果你的Nacos配置集群死活报下图的错误：**  
  
请检查你的MySQL版本，需要在5.7及以上，而且在8.0以下（比较苛刻）

## 6.5 Nacos和Eureka对比

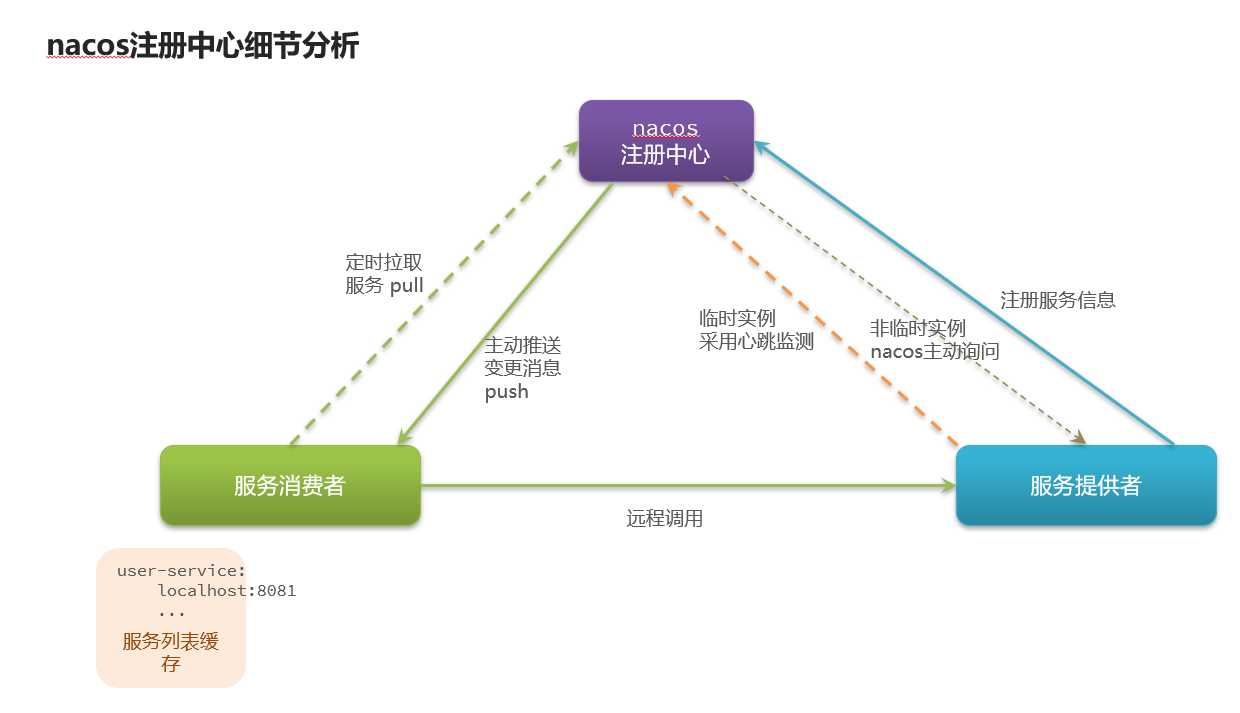
不管是怎样的注册中心啊，我们的服务都会把注册服务信息提交给注册中心，然后服务消费者就会找Nacos拉取注册服务信息，拉取之后会将服务信息存在服务列表缓存(每30S拉取更新)

Nacos会将实例分为临时实例和非临时实例

临时实例会每隔一段时间先Nacos，非临时实例会由Nacos主动询问，如果非临时实例挂了，nacos不会将其剔除而仅仅是标记为”不健康了”等待其恢复健康，这差别。。。

Nacos发现服务挂了则会立即主动推送变更消息给消费者，因此比Eureka时效性更好，及时性更高。

1. Nacos与eureka的共同点
   1. 都支持服务注册和服务拉取
   2. 都支持服务提供者心跳方式做健康检测
2. Nacos与Eureka的区别
   1. Nacos支持服务端主动检测提供者状态：临时实例采用心跳模式，非临时实例采用主动检测模式(会有性能消耗)
   2. 临时实例心跳不正常会被剔除，非临时实例则不会被剔除
   3. Nacos支持服务列表变更的消息推送模式，服务列表更新更及时
   4. Nacos集群默认采用AP方式，当集群中存在非临时实例时，采用CP模式(强调数据可靠性和一致性)；Eureka采用AP方式,不支持切换



**七、Feign远程调用**

**核心代码位置：如下图所示：**

**order-service会引入上图的feign-api，实现远程调用**

**7.1 还原事故现场**

由于上一章（第六章）做了Nacos集群，但是整个第七章是基于单体的注册中心。所以要把集群恢复成单体。

* nacos不使用集群启动，恢复你standalone环境，主要是修改配置文件的nacos端口
* 这样做的目的是让微服务注册进注册中心。你用nacos还原事故现场也行，用eureka还原事故现场也行。反正能还原即可。
* 打开你的数据库服务

**引入feign版本报错bug问题解决：**  
我手工指定了一个版本，版本号是：

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-openfeign</artifactId>

<version>2.2.9.RELEASE</version>

</dependency>

* 1
* 2
* 3
* 4
* 5

**7.2 Feign自定义配置**

分为配置文件方式和代码方式。

* 配置文件方式：
* 代码方式（新建一个配置类）：

**我们采用的是代码方式，并全局生效（新建一个配置类）**

**7.3 Feign性能优化**

底层的客户端实现是：

* URLConnection:默认实现，不支持连接池
* Apache HttpClient: 支持连接池（常用）
* OKHttp：支持连接池

第一种方式是默认的，不支持连接池。所以这里的性能优化指的是：换成第二种方式或者第三种方式。

**其中第二种方式 Apache HttpClient 常用于模拟postman的样式，发送一个form-data样式的post请求，也可在这个post请求里上传文件。我们也采用的是这种方式**

性能优化步骤：  
1、引入jar包：

<!--HttpClient依赖-->

<dependency>

<groupId>io.github.openfeign</groupId>

<artifactId>feign-httpclient</artifactId>

<version>10.1.0</version>

</dependency>

* 1
* 2
* 3
* 4
* 5
* 6

2、在配置文件yml中配置：

feign:

httpclient:

enabled: true # 支持HttpClient的开关

max-connections: 200 # 最大连接数

max-connections-per-route: 50 # 单个路径的最大连接数

* 1
* 2
* 3
* 4
* 5

**这里的改动都是在order-service模块下**

**7.4 Feign最佳实践**

打成jar包方式：

[java中的JAR包](https://yuanyu.blog.csdn.net/article/details/87880736)

**1、在项目中添加单独的jar包步骤：**

写好自己的maven项目后，执行clean package，即可得到一个jar包

**2、在项目中引入单独的jar包图解：**

上图其实是在项目的根目录创建了一个叫lib的文件夹，里面存着自定义jar包。然后即可引入。  
**3、针对1和2的补充，有的时候没必要非得打jar包，可以写一个子模块引入呀，如下图所示：**

这块看不懂，可以自行搜索maven的jar包引入方式和顺序

**八、统一Gateway网关**

**8.1 概述**

三大功能：

* 身份认证和权限校验
* 服务路由、负载均衡
* 请求限流

在SpringCloud中网关技术包括两种：gateway和zuul  
其中Zuul是基于Servlet的实现，属于阻塞式编程，而Gateway则是基于SPring5中提供的WebFlux，属于响应式编程的实现，具备更好的性能。

**8.2 搭建网关服务**

**核心代码位置：如下图**

步骤：

* 新建模块
* 编写配置文件yml：
  + 注册进nacos的配置
  + 网关自身的端口号
  + 网关路由配置

server:

port: 10010

spring:

application:

name: gateway

cloud:

nacos:

server-addr: 127.0.0.1:8848 # nacos地址

gateway:

routes:

- id: user-service # 路由标识，必须唯一

uri: lb://userservice # 路由的目标地址 lb就是负载均衡，后面跟着是服务名称

predicates: # 路由断言，判断请求是否符合规则

- Path=/user/\*\* # 路径断言，判断路径是否是以/user开头，如果是则符合规则

- id: order-service

uri: lb://orderservice

predicates:

- Path=/order/\*\*

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18

除了上面这些，还可以配置路由过滤器。后面会讲到。

配置完毕后，启动你的网关服务和你的user-service和order-service服务，即可通过网关访问到user-service和order-service

**工作原理总结**

**8.3 路由过滤**

**8.3.1 断言工厂：对请求进行过滤**

如果你不会写匹配表达式，可以去spring官网查：

如果你的请求不符合路由断言，那你的请求就会被拒绝，返回一个404. 我们可以通过配置路由断言工厂的方式来过滤某些请求。

**8.3.2 过滤器GatewayFilter：对请求和响应进行过滤**

**它和8.3.1讲述的断言工厂一样，都配置在yaml里**

* GatewayFilter 和 8.3.1讲述的断言工厂的区别：
* 与断言工厂类似，spring也为我们提供了过滤器工厂：

GatewayFilter可以针对某一类路由标识单独配置，也可以配置成全局配置（所有路由id都生效），具体可自行百度，但是过滤器链执行顺序有变化，可以看8.8.4详解

**8.3.3 全局过滤器GlobalFilter：可以自定义过滤逻辑代码实现**

**案例正确执行的效果图：**  
不加参数被过滤器拦截：  
  
加了参数，不被拦截，正确获得响应！

**8.3.4 过滤器链执行顺序**

原理：关键词 适配器模式

顺序：

**8.4 网关跨域问题处理**

域名不一致就是跨域：

* 域名不同 比如www.baidu.com 和 www.bilibili.com
* 域名相同，端口不同

跨域是一个前端的概念，浏览器禁止请求的发起者和服务端发生跨域ajax请求，该请求会被浏览器拦截。  
解决方案：CORS

之所以之前的user-service调用order-service不存在跨域，是因为不是ajax请求。因为这是一个浏览器行为，只有ajax请求会被拦截

**处理方法：**  
简单配置即可：

spring:

cloud:

gateway:

globalcors: # 全局的跨域处理

add-to-simple-url-handler-mapping: true # 解决options请求被拦截问题

corsConfigurations:

'[/\*\*]':

allowedOrigins: # 允许哪些网站的跨域请求

- "http://localhost:8090"

- "http://www.leyou.com"

allowedMethods: # 允许的跨域ajax的请求方式

- "GET"

- "POST"

- "DELETE"

- "PUT"

- "OPTIONS"

allowedHeaders: "\*" # 允许在请求中携带的头信息

allowCredentials: true # 是否允许携带cookie

maxAge: 360000 # 这次跨域检测的有效期

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19

如果想要演示，需要启动一个前端工程模拟一个ajax请求。

**九、**[Docker](https://so.csdn.net/so/search?q=Docker&spm=1001.2101.3001.7020)

Docker命令居多，可以看我下面两张思维导图，包含了概念理解和常用命令。

**9.1 Docker概念**

**9.2 Docker常用命令**

**十、MQ(Message Queue)消息队列**

**10.1 概述**

* **事件驱动架构的概念**：  
  MQ是事件驱动架构的实现形式，MQ其实就是事件驱动架构的Broker。
* **异步应用场景：**  
  如果是传统软件行业：虽然不需要太高并发，但是涉及到和其它系统做对接，我方系统处理速度(50ms)远快于对方系统处理速度(1-3s)，为了兼顾用户的体验，加快单据处理速度，故引入MQ。  
  用户只用点击我方系统的按钮，我方按钮发送到MQ即可给用户返回处理成功信息。背后交由对方系统做处理即可。至于处理失败，补偿机制就不是用户体验要考虑的事情了，这样可以大大提升用户体验。
* **异步通讯优缺点：**
  + **优点：**
    - 耦合度低
    - 吞吐量提升
    - 故障隔离
    - 流量削峰
  + **缺点：**
    - 依赖于MQ的可靠性，安全性，吞吐能力（因为加了一层MQ，当然高度依赖它）
    - 业务复杂了，业务没有明显的流程线，不好追踪管理
* **MQ常见技术介绍：**

**10.2**[RabbitMQ](https://so.csdn.net/so/search?q=RabbitMQ&spm=1001.2101.3001.7020)**安装**

**如何安装，见下图文件：RabbitMQ部署指南.md**

执行MQ容器的命令和简单说明：

docker run \

-e RABBITMQ\_DEFAULT\_USER=root \ #用户名

-e RABBITMQ\_DEFAULT\_PASS=root \ # 密码

--name mq \

--hostname mq1 \ # 主机名，将来做集群部署要用

-p 15672:15672 \ # 端口映射，映射RabbitMQ管理平台端口

-p 5672:5672 \ # 端口映射，消息通信端口

-d \ # 后台运行

rabbitmq:3-management # 镜像名称

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9

#号不被识别，下面提供一个没有#的版本

docker run \

-e RABBITMQ\_DEFAULT\_USER=root \

-e RABBITMQ\_DEFAULT\_PASS=root \

--name mq \

--hostname mq1 \

-p 15672:15672 \

-p 5672:5672 \

-d \

rabbitmq:3-management

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9

最后在浏览器地址栏输入：你的端口号:15672  
  
如果看到上图页面，就说明成功了！

**虚拟主机，租户隔离的概念，重要！！！**  
vitural host：虚拟主机，是对queue、exchange等资源的逻辑分组

**10.3 常见消息模型**

**10.3.1 简单队列模型**

**核心代码位置：下图所示**

**10.4 Spring AMQP**

**概述**

AMQP（Advanced Message Queuing Protocol），是用于在应用程序之间传递业务信息的开放标准，该协议与语言和平台无关，更符合微服务中独立性的要求

SpringAMQP就是Spring基于AMQP定义的一套API规范。

**使用Spring AMQP实现简单队列模型步骤：**

**以生产者为例：**

由于这玩意已被spring托管了，所以对比之前rabbitmq demo的方式，不需要在代码里写配置了，直接在spring的application.yml里写配置文件即可.

配置如下：

# 1.1.设置连接参数，分别是：主机名、端口号、用户名、密码、vhost

spring:

rabbitmq:

host: 127.0.0.1

port: 5672

username: root

password: root

virtual-host: /

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8

然后编写测试类，以及测试代码,位置如下图所示：

**消费者一侧，和生产者类似。不再赘述，如下图进行配置即可：**  
  
至于如何启动消费者 一侧？如下图所示：

**10.3.2 WorkQueue模型**

之所以 10.3.2 放在 10.4章，因为demo模型的演示，今后就是以 Spring AMQP为例了

**概述**  
其实就是一个队列，绑定了多个消费者，一条消息只能由一个消费者进行消费，默认情况下，每个消费者是轮询消费的。**区别于下文的发布-订阅模型（该模型允许将同一消息发给多消费者）**

**案例：**

**10.3.3 发布-订阅模型**

**概念**  
允许将同一个消息发给多个消费者。  
其实就是加了一层交换机而已，如下图所示：  
  
**交换机类型有很多，下文逐一介绍。下图表示了各交换机类型的继承关系**

**最后，交换机只能做消息的转发而不是存储，如果将来路由（交换机和消息队列queue的连接称作路由）没有成功，消息会丢失**

**A. Fanout Exchange**

**位置如下图，注意一定要放在consumer包下，因为是消费者消费行为：**  
  
**生产者添加代码位置如下图：**

**队列绑定成功后，打开mq可视化页面，会看到如下图所示：**

**写好代码后，分别启动生产方，消费方，即可看到调试成功信息输出：**

**概念**：  
这种模型中生产者发送的消息所有消费者都可以消费。

**案例：**

**总结：workQueue模式和FanoutQueue模式区别：**  
**P代表生产者，C代表消费者 X代表交换机，红色部分代表消息队列**  
workQueue:  
  
FanoutQUeue:  
  
可以发现，FanoutQueue增加了一层交换机，可以多个队列对应多个消费者。**而且比起WorkQueue，FanoutQueue生产者是先发送到交换机; 而WorkQueue是直接发送到队列**

**B. Direct Exchange**

**概念**：DirectExchange 会将接收到的消息根据规则路由到指定的queue，因此称为路由模式，如下图所示：

**P代表生产者，C代表消费者 X代表交换机，红色部分代表消息队列**

* 每一个queue都会与Exchange设置一个BindingKey
* 将来发布者发布消息时，会指定消息的RoutingKey
* Exchange将消息路由到BingingKey与RoutingKey一致的队列
* 实际应用时，可以绑定多个key。
* **如果所有queue和所有Exchange绑定了一样的key，那生产者所有符合key的消息消费者都会消费。如果这样做，那DirectExchange就相当于FanoutExchange了（Direct可以模拟Fanout的全部功能）**

案例如图：

**消费者添加代码位置如下图：**  
  
**发送队列添加代码位置如下图：**

**这次的案例，我们用注解的方式声明队列和绑定交换机，之前Fanout的Demo是手写了个配置类。** 直接在监听队列里面声明如下图注解即可：  
  
上图的@QueueBinding点进去：  
  
上面的key是个数组，可以写多个key。

写完代码后启动消费者的SpiringBoot主启动类（报错信息不用管），然后进入rabbitMQ可视化控制台，出现下图则说明配置成功：  
  
随后运行发送队列的Test代码，打开消费者的控制台，出现如下图输出，则说明案例测试通过：

**C. Topic Exchange**

**概念： 和上面的Direct Exchange及其相似：**

（下图来源于Java旅途 ，作者大尧）  
  
**案例：**  
  
**发送队列、消费者的添加代码位置和上面的DirectExchange位置一致，就在DirectExchange代码下面。**

写完代码后启动消费者的SpiringBoot主启动类（报错信息不用管），然后进入rabbitMQ可视化控制台，出现下图则说明配置成功：

**10.3.4 消息转换器**

**引入：**

在之前的案例中，我们发送到队列的都是String类型，但是实际上，我们可以往消息队列中扔进去任何类型。我们看下图，convertAndSend这个方法，第三个参数也是Object。这说明可以发送任何类型给消息队列:

**案例：**  
创建一个队列，向该队列扔一个任意对象（Object类型）

**创建队列位置、发送队列的添加代码位置如下图**  
创建队列位置：  
  
发送：

**写完代码后启动发送的Test，去看RabbitMQ控制台，发现我们发过来的对象在内部被序列化（ObjectOutPutStream）了，如下图所示：**

如果不知道什么是ObjectOutPutStream可自行百度：

上面说的ObjectOutPutStream这个序列化方式，缺点很多（性能差、长度太长、安全性有问题）。我们可以在这里调优一下，推荐JSON的序列化方式。于是引出了这一节的正文：自定义消息转换器(覆盖了原有的Bean配置)：

**声明配置位置如下图**

**配置了消息转换器转换成json，然后重复之前的步骤，使用发送者发送一条消息到队列，发送完成后打开RabbitMQ控制台，出现如下图所示：**  
  
**该对象被成功序列为json格式了！！！！！**

* 对刚才发送过来的json格式消息进行接收，需要修改消费者一侧的代码。并不复杂，如下图所示：  
    
  **消费者配置、监听消息位置如下2图：**

**总结**

* 消息序列化和反序列化使用MessageConverter实现
* SpringAMQP的消息序列化默认底层是使用JDK的序列化
* 我们可以手动配置成其它的序列化方式（覆盖MessageConverter配置Bean），推荐json
* **发送方和接收方必须使用相同的MessageConverter**

**十一、ElasticSearch分布式搜索**

**11.1 ES基础概念**

**ES概述：**

ELK（Elastic Stack）是以Elastic为核心的技术栈，如下图所示：

ElasticSearch底层是Lucene（侧面说明了ES和Hadoop千丝万缕的关系）

推荐下面一篇文章：深入浅出大数据（From Zhihu）https://zhuanlan.zhihu.com/p/54994736  
  
这个Lucene使用java写成的，其实就是个jar包，我们引入之后就可以使用这个Lucene的API。而ES就是基于Lucene的二次开发，对其API进行进一步封装：

**倒排索引基础概念：**

先了解传统MySQL的正向索引：  
  
倒排索引基本概念：

这个倒排索引其实和生活中字典相当像，你拿到一本字典的目录，肯定不会傻到先找页码，你肯定是先大略看一眼目录的关键字，然后找到关键字之后，去看关键字旁边的页码，最后再根据页码翻到书对应的那一页。

**倒排索引其实就是上面的例子。**

然而MySQL这种正向索引，就是基于文档id创建索引，查询词条的时候必须先找到文档，然后根据文档内容判断是否包含词条。

倒排索引正式一点的说法就是：对文档内容分词，对词条创建索引，并记录词条所在文档的信息，查询时先根据词条查询文档id，然后根据id找到该文档。

**文档和词条的概念：**

每一条数据就是一个文档，对文档的内容分词，得到的词语就是词条。

**ES 和 MySQL 概念对比**

**11.2 安装部署ES**

**见课前资料的：安装elasticsearch.md**

使用docker容器化部署，这里针对启动容器命令解析一下：

-e “ES\_JAVA\_OPTS=-Xms512m -Xmx512m” 配置堆内存（JVM）。因为ES底层是java实现的，所以要配置jvm内存大小。默认值是1T，对于轻量级服务器太大了，所以适当减少为512M(但是不能再弄少了，再少的话可能跟着视频走，会出现内存不足的问题)

-e “discovery.type=single-node” 单点模式运行（区别于集群模式运行）

两个-v参数：数据卷挂载，分别是数据保存目录(data)，和插件目录(plugins)

–network es-net 将ES容器加入到刚刚创建的docker网络中

-p 9200:9200 和 -p 9300:9300 是暴露的端口，9200是用户访问的http协议端口，9300是ES容器节点互联的端口

elasticsearch:7.12.1 镜像名称

docker run -d \

--name es \

-e "ES\_JAVA\_OPTS=-Xms512m -Xmx512m" \

-e "discovery.type=single-node" \

-v es-data:/usr/share/elasticsearch/data \

-v es-plugins:/usr/share/elasticsearch/plugins \

--privileged \

--network es-net \

-p 9200:9200 \

-p 9300:9300 \

elasticsearch:7.12.1

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* **安装部署kibana（数据可视化界面）**

黑马官方的kibana的tar包有问题，建议自己从docker hub拉下来镜像。但是拉下来之前要注意 [ES 和 kibana的版本对应关系](https://www.elastic.co/cn/support/matrix#matrix_compatibility)：  
找到对应版本后（我已经找好了），执行命令：

docker pull kibana:7.12.1

从官网拉下来，这个过程比较慢，慢慢等

* **什么是分词器？为什么要安装分词器？**  
    
    
  分词器我们选择IK分词器（来源于github，专门适配了中文）  
    
  该分词器的具体安装也在文档里有写。
* **分词器总结**

**[Debug] 停止ES容器（或是重启Linux）后，如何恢复Docker网络：**

**11.3 索引库操作**

先给出ES官方帮助文档地址：  
https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html

索引库相当于MySQL中的Table。具体操作有两个：

* Mapping映射属性
* 索引库的CRUD

**先介绍Mapping映射属性：**

* **创建索引库**  
    
  **一个简单的创建索引库的语句：**

# 创建索引库

PUT /heima

{

"mappings": {

"properties": {

"info": {

"type": "text",

"analyzer": "ik\_smart"

},

"email": {

"type": "keyword",

"index": false

},

"name": {

"type": "object",

"properties": {

"firstName": {

"type": "keyword"

},

"lastName": {

"type": "keyword"

}

}

}

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* **查看、修改、删除索引库**

查看索引库：GET /索引库名  
删除索引库：DELETE /索引库名

修改索引库从设计上被禁止了，索引库和mapping一旦创建无法修改，但是可以添加新的字段 **(该字段必须是全新的字段)** 。

它们的语法如下：

# 查询

GET /heima

# 修改（必须添加一个全新的字段）

PUT /heima/\_mapping

{

"properties":{

"age":{

"type": "integer"

}

}

}

# 删除

DELETE /heima

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16

**11.4 文档操作**

索引库相当于数据库的table，文档就相当于数据库的行。

* **添加文档**

# 插入一个文档

POST /heima/\_doc/1

{

"info": "黑马程序员java讲师",

"email": "112837@qq.com",

"name":{

"firstName":"云",

"lastName":"赵"

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* **查看、删除文档**

# 查询

GET /heima/\_doc/1

# 删除

DELETE /heima/\_doc/1

* 1
* 2
* 3
* 4
* 5

每次写操作的时候，都会使得文档的"\_version"字段+1

* **修改文档方式1 全量修改**  
  它会删除旧文档，新增新文档

语法：和新增的语法完全一致，只不过新增是POST，全量修改是PUT  
示例：

# 插入一个文档

PUT /heima/\_doc/1

{

"info": "黑马程序员java讲师",

"email": "112837@qq.com",

"name":{

"firstName":"云",

"lastName":"赵"

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

如果id在索引库里面不存在，并不会报错，而是直接新增，如果索引库存在该记录，就会先删掉该记录，然后增加一个全新的。

* **修改文档方式2 增量修改**  
  只修改某记录的指定字段值  
  语法：

# 局部修改文档字段

# 第三行，必须跟一个doc

POST /heima/\_update/1

{

"doc": {

"email":"lbwnb@qq.com"

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9

**文档操作总结**

**11.5 RestClient操作索引库和文档**

* **概念**  
  ES官方为各种语言操作ES提供了客户端API，用来操作ES。其实本质都是组装ES语句，通过http请求发送给ES。 官方文档地址：<https://www.elastic.co/guide/en/elasticsearch/client/index.html>  
    
  可以看到有很多语言的版本。
* **案例和代码位置**

**代码位置(大量代码写在测试类中)，该案例需要导入数据库，数据库执行脚本位置同代码目录：**

* **编写DSL语句，创建索引库（相当与MySQL中建表）**

语句如下：

# 酒店的mapping

PUT /hotel

{

"mappings": {

"properties": {

"id":{

"type": "keyword"

},

"name":{

"type": "text"

, "analyzer": "ik\_max\_word",

"copy\_to": "all"

},

"address":{

"type": "keyword"

, "index": false

},

"price":{

"type": "integer"

},

"score":{

"type": "integer"

},

"brand":{

"type": "keyword",

"copy\_to": "all"

},

"city":{

"type": "keyword"

},

"starName":{

"type": "keyword"

},

"business":{

"type": "keyword",

"copy\_to": "all"

},

"location":{

"type": "geo\_point"

},

"pic":{

"type": "keyword"

, "index": false

},

"all":{

"type": "text",

"analyzer": "ik\_max\_word"

}

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51

有时候可能会疑惑，同样的一个文本型字段，有的用text，有的用keyword。到底怎么选择呢？首先要了解索引和分词的概念：

* 索引(参与搜索，排序筛选等操作)
* 分词（把词看作一个整体还是把词用某种规则分开）
  + 比如 ： 上海，北京这种字段，不需要分词（这种字段在一个整体才有意义，分词就乱套了）
  + “震惊！卢bw将于2022年复出” 这种就需要分词搜索，既然要分词了，肯定要选择分词器。  
    了解了上面的概念，再看一下下图（[图来源于博客园——瘦风的南墙](https://www.cnblogs.com/shoufeng/p/10692113.html)）：  
      
    **备注1：index如果设置成false，则既不参与索引也不参与分词。  
    备注2：索引库的id总是被要求成keyword（也就是String）类型，即使数据库的主键id可能是int**

字段参数（用于聚合）：copy to ;  
地理位置特殊数据类型：geo\_point

**使用RestClient操作文档（索引库相当于数据库的table，文档就相当于数据库的行。），全都写在demo代码中，还是那句话：Java的API本质都是组装ES语句，通过http请求发送给ES。**

**11.6 DSL查询语法**

先给出帮助文档，帮助文档永远是学东西最准确的方式：  
<https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>

* **快速入门—简单查询：**

全文检索查询例：

# match 和 multi\_match

GET /hotel/\_search

{

"query": {

"match": {

"address": "如家外滩"

}

}

}

GET /hotel/\_search

{

"query": {

"multi\_match": {

"query": "外滩如家",

"fields": ["brand","name","business"]

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19

精确查询例：

# 精确查询（term查询）

GET /hotel/\_search

{

"query": {

"term": {

"city": {

"value": "上海"

}

}

}

}

# 精确查询(范围range)

GET /hotel/\_search

{

"query": {

"range": {

"price": {

"gte": 100,

"lte": 300

}

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24

地理查询例：

# distance查询

GET /hotel/\_search

{

"query": {

"geo\_distance":{

"distance": "5km",

"location": "31.21, 121.5"

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* **快速入门—打分算法：**

**打分算法（重点）：**  
  
对默认算分方式进行修改：  
  
**组合查询-function score 对应的Java RestClient代码：**

上面例子的查询语句：

GET /hotel/\_search

{

"query": {

"function\_score": {

"query": {

"match": {

"address": "外滩"

}

},

"functions": [

{

"filter": {

"term": {

"brand": "如家"

}

},"weight": 10

}

]

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* **快速入门—复合查询：**

复合查询可以将其它简单查询组合起来，实现更复杂的搜索逻辑。

**Boolean Query**

**注意，算分条件越多，性能就会越差。所以能使用filter的就别使用must，能不算分就不算分**

**案例：搜索名字包含“如家”，价格不高于400，在坐标31.21，121.5周围10km范围内的酒店**  
参考答案：

GET /hotel/\_search

{

"query": {

"bool": {

"must": [

{"match": {

"name": "如家"

}}

],

"must\_not": [

{"range": {

"price": {

"gt": 400

}

}}

],

"filter": [

{

"geo\_distance": {

"distance": "100km",

"location": {

"lat": 31.21,

"lon": 121.5

}

}

}

]

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* **快速入门—搜索结果处理：**

搜索结果的处理主要包括**排序、分页、高亮**。默认ES是根据得分排序的，但是你如果指定了按某种字段排序，就会按你指定的方法排序。

**A.排序**  
  
案例：  
  
查询语句实现：

# sort排序

GET /hotel/\_search

{

"query": {

"match\_all": {}

},

"sort": [

{

"score": "desc"

},

{

"price": "asc"

}

]组合查询-function score 对应的Java RestClient代码：

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16

案例2：

查询语句实现2：

GET /hotel/\_search

{

"query": {

"match\_all": {}

},

"sort": [

{

"\_geo\_distance": {

"location": {

"lat": 31.03,

"lon": 121.61

},

"order": "asc"

, "unit": "km"

}

}

]

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20

**地理位置排序对应的java restclient代码：**

注意：一旦指定了某种排序之后，ES就会放弃打分。因为打分没意义了：  
  
**B.分页**

ES默认情况只返回10条数据，如果想返回更多条数据，则需修改分页参数。

分页语法（有点像MySQL的limit）：

示例：

GET /hotel/\_search

{

"query": {

"match\_all": {}

},

"sort": [

{

"price": "asc"

}

],

"from": 20

, "size": 5

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15

**分页出现的问题：ES底层是倒排索引，不利于分页，所以分页查询是一种逻辑上的分页。比如现在要查从990开始，截取10条数据（990～1000这10条），对ES来讲，是先查出来0～1000条数据，查出来之后逻辑分页截取10条给你。这么做如果是单体，最多只是效率问题，但是如果是集群，就会坏事。如下图所示：**

针对只能查询10000条结果的解决方案：  
  
**C.高亮**

示例：

# 高亮查询,默认情况下ES搜索字段必须与高亮字段一致

GET /hotel/\_search

{

"query": {

"match": {

"name": "如家"

}

},"highlight": {

"fields": {

"name": {

}

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16

**总结：**

**11.7 Java RestClient查询语法**

要构建查询条件，只要记住一个类：QueryBuilders。  
要构建搜索DSL，只需记住一个API：SearchRequest的source()方法（支持链式编程）

**核心代码位置：**

这里只有一个注意点：高亮结果的解析，比较麻烦。代码要配合下图理解：

**11.8 ES综合案例：黑马旅游**

**代码位置：就是11.7那个类，直接启动SpringBoot主启动类，然后访问localhost:8089即可访问到前端页面**

**要实现的功能：**

* 酒店搜索和分页
* 酒店结果过滤
* 我周边的酒店
* 酒店竞价排名

视频可能出现的bug：

**bug1 : 如果前端显示异常（搜索不生效），根据前端debug信息，修改index.html的第417行代码修改成如下图所示：**  
  
**bug2: 黑马旅游网的酒店竞价排名实现不了**

由于在视频里创建索引库里并没有创建isAD这个字段，我们需要手动追加该字段。在kibana控制台执行如下代码即可修复：

# 给索引库新增一个叫isAD的字段，类型是布尔类型

PUT /hotel/\_mapping

{

"properties":{

"isAD":{

"type": "boolean"

}

}

}

# 给索引库id为45845的记录赋值，让其isAD字段为true（用于测试广告竞价排名，该记录会靠前）

POST /hotel/\_update/45845

{

"doc": {

"isAD":true

}

}

GET hotel/\_doc/45845

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20

**11.9 ES数据聚合**

聚合，类似于MySQL的group by（对数据的统计分析和计算）。聚合不能是text类型，不能分词

聚合一共有几十种，在官方文档可以查到，但是主要分为三大类：  
  
管道聚合 可以理解为linux的 |

**1、Bucket聚合**

查询实例：  
  
**上图图例的结果是由count进行降序排列的，如果想让其升序排列，只需如下代码：**

# 聚合功能

GET hotel/\_search

{

"size": 0,

"aggs": {

"brandAgg": {

"terms": {

"field": "brand",

"size": 10,

"order": {

"\_count": "asc" #结果按照count升序排列

}

}

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16

**限定聚合范围：**

**2、Metrics聚合**

示例：

# 嵌套聚合metric

GET hotel/\_search

{

"size": 0,

"aggs": {

"brandAgg": {

"terms": {

"field": "brand",

"size": 10,

"order": {

"scoreAgg.avg": "asc" # 根据下面的子聚合结果的avg进行升序排序

}

},

"aggs": {

"scoreAgg": {

"stats": {

"field": "score"

}

}

}

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23

**使用Java Restclient实现上面几种聚合方式，位置如下：**

Java Restclient对应Json的图例：  
  
Java代码对应结果解析的图例：  
  
**3、聚合案例：**  
  
**案例位置同上面的 ES综合案例：黑马旅游**

**11.10 ES数据补全**

比如你在京东输入 sj 这两个字母，搜索框就会猜测出你想输入手机。这个就是数据补全

**安装数据补全分词器：**

分词器在课前资料里有  
  
测试你的分词器是否生效：

POST \_analyze

{

"text": ["卢本伟"],

"analyzer": "pinyin"

}

* 1
* 2
* 3
* 4
* 5

**自定义配置分词器：**  
概念：

将下图位置的自定义配置分词器的第一段粘贴至kibana控制台，即可完成自定义配置：

**Completion Suggester查询实现自动补全：**

Completion Suggester语法：

// 自动补全查询

GET /test/\_search

{

"suggest": {

"title\_suggest": {

"text": "s", // 关键字

"completion": {

"field": "title", // 补全字段

"skip\_duplicates": true, // 跳过重复的

"size": 10 // 获取前10条结果

}

}

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15

**总结：**  
自动补全对字段的要求：  
类型是completion类型；字段值是多词条的数组。

**案例：实现hotel索引库的自动补全、拼音搜索功能：**  
  
找到下图位置，复制粘贴进kibana控制台并且执行（这一步是重建酒店数据索引库，在此之前要删掉原有的酒店数据索引库）：  
  
**注意事项：**  
  
**在Java代码中重新定义转换实体的操作，定义一个新的字段suggestion，并且在kibana控制台进行测试：**

**经过上面一番操作后，类型为completion类型的suggestion字段就有了我们想要自动补全的例子，然后执行下面的查询语句：**  
  
**至此，自动补全、拼音搜索的demo已成功展示！**

**对上图的DSL语句在Java RestAPI里面进行发送：**

**使用Java Restclient实现上面自动补全方式，位置如下：**

案例效果：

**11.11 ES与MySQL之间数据同步（面试常问）**

概念

ES中的酒店数据来自于MySQL索引库，因此mysql数据发生改变时，ES的值也会跟着改变，这个就是ES和MySQL的数据同步。

思考：在微服务中，操作MySQL的业务和操作ES的业务可能在不同的微服务上，这种情况应该怎么实现数据同步呢？

**解决方案：**

**案例：利用MQ实现mysql与es的数据同步**  
  
**思路：**

**数据同步案例后台管理页面代码位置如下图（数据库就用之前的ES综合案例：黑马旅游）：**

**数据同步案例前端显示代码就是之前的ES综合案例：黑马旅游。前后端的微服务是分离的，端口号也不同。**  
**实际上，这个项目hotel-admin项目相当于生产者，负责发送数据库增删改消息；hotel-demo(之前的黑马旅游前端项目)相当于消费者，负责监听消息并更新ES中的数据。**

**这样就实现了在微服务中，操作MySQL的业务和操作ES的业务在不同的微服务上的跨服务数据同步**

用心跟着代码走，这个案例是完全可以做完并实现**视频全部功能**的，没有一句废话多余。

**11.12 搭建高可用ES集群**

**概念**

**搭建ES集群**

**位置同之前的elasticsearch.md，找到该文档第四节:部署ES集群**

**集群脑裂问题**  
  
  
脑裂问题：一个集群出现了2个主节点：

**集群分布式存储和分布式查询**

**集群故障转移**

**集群故障转移总结：**

* Master挂掉后，EligibleMaster选举为新的主节点
* master节点监控分片，节点状态，将故障节点的分片转移到正常节点，确保数据安全。

**后记**

黑马 SpringCloud 2021 基础篇笔记和代码已更新完毕，不得不说黑马的这套课程的确是良心之作，而且官方居然还开源出来让大家都可以学习，实在是难能可贵。

**如果大家在学习基础篇的同时有疑问，欢迎在评论区讨论和留言，也可以关注我，日后我还会陆续更新完高级篇和面试篇。**

[**@大吉**](https://blog.csdn.net/weixin_44757863)

[关注](javascript:;)

* 6
* [1](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#commentBox)
* [20](javascript:;)

专栏目录

[SpringCloud教程完整讲义](http://download.csdn.net/download/qq_21231347/11804567" \t "_blank)

09-23

[SpringCloud的完整讲义，讲的很详细，非常容易理解，从大局讲到细节；通过对比讲清springcloud的应用](http://download.csdn.net/download/qq_21231347/11804567" \t "_blank)

[2021年新版移动支付，SpringBoot，SpringCloud，RabbitMQ等，大型分布式商城系统，含源码...](https://blog.csdn.net/qq_17231297/article/details/120806694" \t "_blank)

[最新发布](https://blog.csdn.net/qq_17231297/article/details/120806694" \t "_blank)

[公众号-老炮说Java](https://blog.csdn.net/qq_17231297)

 168

[最近有一些读者让我帮他们找，完整的商城项目，这不昨天发现整套?分布式，SpringBoot，SpringCloud，Redis，RabbitMQ，MyCat，分布式事物，支付宝支付，微信支...](https://blog.csdn.net/qq_17231297/article/details/120806694" \t "_blank)

**评论**1

窗体顶端



窗体底端

[我叫李花花](https://blog.csdn.net/qq_37323956)

高级篇出了吗？

回复10 天前

[...SpringCloud微服务项目资源\_脚印\_黑马springcloud项目](https://blog.csdn.net/tian_ci/article/details/109391187" \t "_blank)

1-8

[黑马十次方SpringCloud微服务项目资源 资源易失效,请各位移步我的个人博客「xiongtianci.com/blog.xiongtianci.com」进行获取,通过右上角的搜索框搜索十次方:](https://blog.csdn.net/tian_ci/article/details/109391187" \t "_blank)

[b站黑马springCloud-常见面试题,多多三连\_橘·灬的博客](https://blog.csdn.net/qq_40780198/article/details/121372344" \t "_blank)

12-17

[b站黑马springCloud-常见面试题,多多三连 常见面试题 1.微服务篇 1.1.SpringCloud常见组件有哪些? 问题说明:这个题目主要考察对SpringCloud的组件基本了解 难易程度:简单 参考话术: SpringCloud包含的组件很多,有很多功能是重复的。其中...](https://blog.csdn.net/qq_40780198/article/details/121372344" \t "_blank)

[史上最简单的SpringCloud教程 | 第十一篇: docker部署spring cloud项目](https://blog.csdn.net/weixin_30569001/article/details/98955847" \t "_blank)

[weixin\_30569001的博客](https://blog.csdn.net/weixin_30569001)

 3098

[转载请标明出处： 原文首发于：https://www.fangzhipeng.com/springcloud/2017/07/12/sc11-docker/ 本文出自方志朋的博客 一、docker简介 Docker是一个开源的引擎，可以轻松的为任何应用创建一个轻量级的、可移植的、自给自足的容器。开发者在笔记本上编译测试通过的容器可以批量...](https://blog.csdn.net/weixin_30569001/article/details/98955847" \t "_blank)

[2021最新版RabbitMQ完整教程学习笔记](https://blog.csdn.net/weixin_47257749/article/details/116400508" \t "_blank)

[super\_song的博客](https://blog.csdn.net/weixin_47257749)

 861

[文章目录一、中间件1. 什么是中间件2. 中间件技术及架构的概述3. 基于消息中间件的分布式系统的架构4. 消息队列协议5. 消息队列持久化6. 消息的分发策略7. 消息队列高可用和高可靠二、入门及安装 RabbitMQ1. RabbitMQ入门及安装概述下载RabbitMQ安装Erlang安装socat安装rabbitmq2. RabbitMQWeb管理界面及授权操作RabbitMQ管理界面授权账号和密码小结：3. RabbitMQ之Docker安装Dokcer安装RabbitMQ4. RabbitMQ的](https://blog.csdn.net/weixin_47257749/article/details/116400508" \t "_blank)

[Springcloud2020.0.3学习笔记(黑马+尚硅谷)(一:介绍)](https://blog.csdn.net/qq_45886944/article/details/120414609" \t "_blank)

1-3

[spring cloud2020.0.3最大的变化就是移除了spring cloud netflix模块,仅仅包括了eureka模块。 既然把Netflix组件remove掉了,那么需要有替代方案,Spring Cloud团队给我们推荐了用于替代的产品:](https://blog.csdn.net/qq_45886944/article/details/120414609" \t "_blank)

[...之SpringCloud的介绍\_Java持续实践\_黑马springcloud...](https://blog.csdn.net/qq_33229669/article/details/87456220" \t "_blank)

1-8

[黑马十次方项目day07-01之SpringCloud的介绍 1. 1什么是SpringCloud Spring Cloud是一系列框架的有序集合。它利用Spring Boot的开发便利性巧妙地简 化了分布式系统基础设施的开发,如服务发现注册、配置中心、消息总线、负载均衡、...](https://blog.csdn.net/qq_33229669/article/details/87456220" \t "_blank)

[SpringCloud整合Redis](http://download.csdn.net/download/cadn_jueying/10486599" \t "_blank)

06-19

[SpringCloud整合Redis缓存；版本：SpringCloud ：Dalston.SR4；SpringBoot：1.5.14.RELEASE；Redis:3.2.0;Maven :3.5.3.代码下载下来即可用](http://download.csdn.net/download/cadn_jueying/10486599" \t "_blank)

[【良心好物】SpringCloud+RabbitMQ+Docker+Redis+搜索+分布式](https://blog.csdn.net/qq_41997592/article/details/120123964" \t "_blank)

[qq\_41997592的博客](https://blog.csdn.net/qq_41997592)

 56

[目录前言正文内容结束 前言 找一个适合自己的网课可太难了！！ 于是记录一下，对比过某些付费课程，对于个人来说的确更适合这个！ 放在这里，有需要的兄弟可以看看，不要再被看的难受的课折磨了！ 正文内容 不是广告！！不是广告！！不是广告！！ 讲解清楚，图文并茂！！难以相信这样的课居然免费！！冲！ 课程链接 结束 B站真棒，听着这样的课学习效率都高出来不少。 ...](https://blog.csdn.net/qq_41997592/article/details/120123964" \t "_blank)

[SpringCloud学习笔记(1):Eureka注册中心\_黑马程序员广...](https://blog.csdn.net/u010395024/article/details/106186261" \t "_blank)

1-2

[4.创建配置文件/src/main/resources/application.yml:server: port: 8081spring: application: name:sc-provider#注册到Eureka注册中心上的服务名称,对应Eureka界面上的Application列 eureka: client: serviceUrl: defaultZone: http://local...](https://blog.csdn.net/u010395024/article/details/106186261" \t "_blank)

[黑马mysql百度云盘\_黑马-百度云搜索引擎](https://blog.csdn.net/weixin_39966765/article/details/113460504" \t "_blank)

[weixin\_39966765的博客](https://blog.csdn.net/weixin_39966765)

 341

[资源内容：黑马|\_\_\_\_阶段6 微服务社交平台【十次方】|\_\_\_\_17 项目完结|\_\_\_\_16 dockfile|\_\_\_\_15 SpringCloud之一统天下|\_\_\_\_14 SpringCloud之初入江湖|\_\_\_\_13 密码加密与微服务鉴权JWT|\_\_\_\_12 rabbitm...](https://blog.csdn.net/weixin_39966765/article/details/113460504" \t "_blank)

[2019年分布式架构教程 SpringCloud+Docker基础入门到实战](https://download.csdn.net/download/weixin_27038261/19551013" \t "_blank)

06-10

[2019年分布式架构教程 SpringCloud+Docker基础入门到实战](https://download.csdn.net/download/weixin_27038261/19551013" \t "_blank)

[使用springcloud2021版本修改黑马的教程](https://blog.csdn.net/qq_45886944/article/details/120740515" \t "_blank)

[大胆刁民的博客](https://blog.csdn.net/qq_45886944)

 268

[Gateway服务网关 Spring Cloud Gateway 是 Spring Cloud 的一个全新项目，该项目是基于 Spring 5.0，Spring Boot 2.0 和 Project Reactor 等响应式编程和事件流技术开发的网关，它旨在为微服务架构提供一种简单有效的统一的 API 路由管理方式。 Spring Cloud Gateway 作为 Spring Cloud 生态系统中的网关，目标是替代 Netflix Zuul，其不仅提供统一的路由方式，并且基于 Filter 链的方式提供](https://blog.csdn.net/qq_45886944/article/details/120740515" \t "_blank)

[spring-cloud-learning:Springcloud + rabbitmq + kafka + slueth + zipkin + lcn + redis-源码](https://download.csdn.net/download/weixin_42117340/16035877" \t "_blank)

03-22

[Spring云学习 注册表，redis，mq，分布式事务，互连追踪等](https://download.csdn.net/download/weixin_42117340/16035877" \t "_blank)

[SpringBoot+Cloud实战项目—黑马乐优商城(19天全套+视频笔记)](http://download.csdn.net/download/weixin_44011409/10866225" \t "_blank)

12-21

[从第50天到68天 视频、 笔记、源码都有 十分齐全 springboot+springcloud](http://download.csdn.net/download/weixin_44011409/10866225" \t "_blank)

[最新黑马程序员十次方项目](https://download.csdn.net/download/u010865991/11076722" \t "_blank)

03-31

[微服务开发，前端系统开发，人工智能，共计20天 培训时间比较新，2018年10月份的项目，貌似最近很多人都在找 目前只找到了微服务开发的10天，这是最完整的高清版本，拿完记得点赞。如链接失效，请留言提示](https://download.csdn.net/download/u010865991/11076722" \t "_blank)

[黑马程序员代码生成器](https://download.csdn.net/download/weixin_42112635/11021433" \t "_blank)

03-15

[黑马程序员代码生成器](https://download.csdn.net/download/weixin_42112635/11021433" \t "_blank)

[RabbitMQ 整合 SpringCloud实战](https://blog.csdn.net/u010395024/article/details/104550135" \t "_blank)

[黑马程序员广州中心的专栏](https://blog.csdn.net/u010395024)

 85

[RabbitMQ 整合 SpringCloud实战 RabbitMQ 整合 SpringCloud实战 rabbitmq-common 子项目 rabbitmq-springcloud-consumer 子项目 rabbitmq-springcloud-producer 子项目 RabbitMQ 整合 SpringCloud实战 ...](https://blog.csdn.net/u010395024/article/details/104550135" \t "_blank)

[传智播客管家婆java答案\_传智播客黑马19年Java全(IDEA) [230G]|百度云|天翼云|GD|OD...](https://blog.csdn.net/weixin_39819393/article/details/111839076" \t "_blank)

[weixin\_39819393的博客](https://blog.csdn.net/weixin_39819393)

 348

[目录：/黑马19年Java全(IDEA) [412.2M]┣━━01-java语言入门 [0B]┃ ┣━━01\_Java基础语法┃ ┗━━02\_面向对象与常用类┣━━02-Java语进阶 [0B]┃ ┣━━day01\_Object类、常用API┃ ┣━━day02\_Collection、泛型┃ ┣━━day03\_List、Set、数据结构、Collections┃ ...](https://blog.csdn.net/weixin_39819393/article/details/111839076" \t "_blank)

[【Docker】容器技术黑马Docker（一）——了解容器技术，了解Docker](https://blog.csdn.net/kisscatforever/article/details/78896625" \t "_blank)

[哈士奇](https://blog.csdn.net/kisscatforever)

 1万+

[一、前言&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;说到容器，小编最先接触的容器还是tomcat这些web容器，感觉容器就是把应用程序装进去，应用程序在容器中运行，我们可以通过访问容器暴露的端口来访问容器中的应用程序。&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;再说到容器技术，这个就厉害了。 这里小编就重点介绍一下容器技术的发展以及容器技术的黑马Docker。二、容器技术发展&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;小编觉得这张图非常棒，涵盖了容器技术的演变。&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;容器技术最早可以追溯](https://blog.csdn.net/kisscatforever/article/details/78896625" \t "_blank)

[黑马 Docker 笔记](https://sixsixsix516.blog.csdn.net/article/details/108919697" \t "_blank)

[伟大的计算机科学家](https://blog.csdn.net/weixin_42195284)

 373

[# 1、yum 包更新到最新 yum update # 2、安装需要的软件包， yum-util 提供yum-config-manager功能，另外两个是devicemapper驱动依赖的 yum install -y yum-utils device-mapper-persistent-data lvm2 # 3、 设置yum源 sudo yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/d.](https://sixsixsix516.blog.csdn.net/article/details/108919697" \t "_blank)

[Docker redis 集群搭建](https://blog.csdn.net/u010395024/article/details/104746474" \t "_blank)

[黑马程序员广州中心的专栏](https://blog.csdn.net/u010395024)

 183

[Docker redis 集群搭建 Redis 集群是一个提供在多个 Redis 节点间共享数据的程序集。 Redis 集群并不支持处理多个 keys 的命令,因为这需要在不同的节点间移动数据，从而达不到像 Redis 那样的性能，在高负载的情况下可能会导致不可预料的错误. Redis 集群通过分区来提供一定程...](https://blog.csdn.net/u010395024/article/details/104746474" \t "_blank)

[spring-cloud学习教程/视频（最新最全）](https://blog.csdn.net/c_zhouwei084/article/details/81812204" \t "_blank)

[热门推荐](https://blog.csdn.net/c_zhouwei084/article/details/81812204" \t "_blank)

[c\_zhouwei084的博客](https://blog.csdn.net/c_zhouwei084)

 2万+

[下载地址： 链接：https://pan.baidu.com/s/1JU0PUZNHF1t7fwIlgnh7sA 密码：y5vw 课程目录 &nbsp; &nbsp;&nbsp; &nbsp;&nbsp;&nbsp;第1篇 分布式开发与SpringCloud简介 &nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp; 小节1： 分布式开发简介 &nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp; 小节2： 线上笔记 分布式开发简介 &nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp;&nbsp; &nbsp; 小节3： SpringCloud开...](https://blog.csdn.net/c_zhouwei084/article/details/81812204" \t "_blank)

©️2021 CSDN 皮肤主题: 1024 设计师:白松林 [返回首页](https://blog.csdn.net/)

* [关于我们](https://www.csdn.net/company/index.html#about)
* [招贤纳士](https://www.csdn.net/company/index.html#recruit)
* [广告服务](https://www.csdn.net/company/index.html#advertisement)
* [开发助手](https://plugin.csdn.net/)
* 400-660-0108
* [kefu@csdn.net](mailto:webmaster@csdn.net)
* [在线客服](https://csdn.s2.udesk.cn/im_client/?web_plugin_id=29181)
* 工作时间 8:30-22:00
* [公安备案号11010502030143](http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=11010502030143)
* [京ICP备19004658号](http://beian.miit.gov.cn/publish/query/indexFirst.action)
* [京网文〔2020〕1039-165号](https://csdnimg.cn/release/live_fe/culture_license.png)
* [经营性网站备案信息](https://csdnimg.cn/cdn/content-toolbar/csdn-ICP.png)
* [北京互联网违法和不良信息举报中心](http://www.bjjubao.org/)
* [网络110报警服务](http://www.cyberpolice.cn/)
* [中国互联网举报中心](http://www.12377.cn/)
* [家长监护](https://download.csdn.net/index.php/tutelage/)
* [Chrome商店下载](https://chrome.google.com/webstore/detail/csdn%E5%BC%80%E5%8F%91%E8%80%85%E5%8A%A9%E6%89%8B/kfkdboecolemdjodhmhmcibjocfopejo?hl=zh-CN)
* ©1999-2022北京创新乐知网络技术有限公司
* [版权与免责声明](https://www.csdn.net/company/index.html#statement)
* [版权申诉](https://blog.csdn.net/blogdevteam/article/details/90369522)
* [出版物许可证](https://img-home.csdnimg.cn/images/20210414021151.jpg)
* [营业执照](https://img-home.csdnimg.cn/images/20210414021142.jpg)
* [关于我们](file:///\\www.csdn.net\company\index.html#about)
* [招贤纳士](file:///\\www.csdn.net\company\index.html#recruit)
* [广告服务](file:///\\www.csdn.net\company\index.html#advertisement)
* [开发助手](https://plugin.csdn.net/)
* 400-660-0108
* [kefu@csdn.net](mailto:webmaster@csdn.net)
* [在线客服](https://csdn.s2.udesk.cn/im_client/?web_plugin_id=29181)
* 工作时间 8:30-22:00
* [公安备案号11010502030143](http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=11010502030143)
* [京ICP备19004658号](http://beian.miit.gov.cn/publish/query/indexFirst.action)
* [京网文〔2020〕1039-165号](https://csdnimg.cn/release/live_fe/culture_license.png)
* [经营性网站备案信息](https://csdnimg.cn/cdn/content-toolbar/csdn-ICP.png)
* [北京互联网违法和不良信息举报中心](http://www.bjjubao.org/)
* [网络110报警服务](http://www.cyberpolice.cn/)
* [中国互联网举报中心](http://www.12377.cn/)
* [家长监护](https://download.csdn.net/index.php/tutelage/)
* [Chrome商店下载](https://chrome.google.com/webstore/detail/csdn%E5%BC%80%E5%8F%91%E8%80%85%E5%8A%A9%E6%89%8B/kfkdboecolemdjodhmhmcibjocfopejo?hl=zh-CN)
* ©1999-2022北京创新乐知网络技术有限公司
* [版权与免责声明](https://www.csdn.net/company/index.html#statement)
* [版权申诉](https://blog.csdn.net/blogdevteam/article/details/90369522)
* [出版物许可证](https://img-home.csdnimg.cn/images/20210414021151.jpg)
* [营业执照](https://img-home.csdnimg.cn/images/20210414021142.jpg)

[@大吉](https://blog.csdn.net/weixin_44757863)

码龄3年[山东农业大学](https://i.csdn.net/#/uc/profile?utm_source=14998968)

[81](https://blog.csdn.net/weixin_44757863)

[原创](https://blog.csdn.net/weixin_44757863)

[2万+](https://blog.csdn.net/rank/list/weekly" \t "_blank)

[周排名](https://blog.csdn.net/rank/list/weekly" \t "_blank)

[1万+](https://blog.csdn.net/rank/list/total" \t "_blank)

[总排名](https://blog.csdn.net/rank/list/total" \t "_blank)

14万+

访问

等级

1241

积分

24

粉丝

203

获赞

77

评论

283

收藏

[私信](https://im.csdn.net/chat/weixin_44757863)

关注



**热门文章**

* [解决org.apache.ibatis.binding.BindingException: Invalid bound statement (not found)问题  87892](https://blog.csdn.net/weixin_44757863/article/details/109301707)
* [在MyBatis中使用log4j.  3947](https://blog.csdn.net/weixin_44757863/article/details/109039696)
* [SpringBoot模块中,resources目录下各文件夹作用  3559](https://blog.csdn.net/weixin_44757863/article/details/109394145)
* [MVC架构和SSM框架的个人理解  3374](https://blog.csdn.net/weixin_44757863/article/details/109086022)
* [thymeleaf的th:each双层嵌套循环（一个最简单的Demo实现）  2513](https://blog.csdn.net/weixin_44757863/article/details/110785167)

**最新评论**

* [黑马2021最新版 SpringCloud基础篇全技术栈导学（RabbitMQ+Docker+Redis+搜索+分布式）](https://blog.csdn.net/weixin_44757863/article/details/120959505#comments_19679202)

[我叫李花花:](https://blog.csdn.net/qq_37323956)高级篇出了吗？

* [解决org.apache.ibatis.binding.BindingException: Invalid bound statement (not found)问题](https://blog.csdn.net/weixin_44757863/article/details/109301707#comments_19496975)

[courage191006:](https://blog.csdn.net/wddcz0523)牛逼

* [MVC架构和SSM框架的个人理解](https://blog.csdn.net/weixin_44757863/article/details/109086022#comments_19464672)

[m0\_57195751:](https://blog.csdn.net/m0_57195751)心中疑惑全解开了

* [thymeleaf的局部刷新(一个最简单的Demo实现，包括Controller和html)](https://blog.csdn.net/weixin_44757863/article/details/110913571#comments_19449818)

[weixin\_51624404:](https://blog.csdn.net/weixin_51624404)我想问下，那个评论区内容是有三级或三级以上的评论，前端该怎么使用fragment进行渲染。

* [解决org.apache.ibatis.binding.BindingException: Invalid bound statement (not found)问题](https://blog.csdn.net/weixin_44757863/article/details/109301707#comments_19214119)

[七月想养猫:](https://blog.csdn.net/qq_38081145)哥，叫你亲哥，给我解决的大问题

**您愿意向朋友推荐“博客详情页”吗？**

强烈不推荐

不推荐

一般般

推荐

强烈推荐

**最新文章**

* [深拷贝解决线程间数据互相覆盖问题（避坑记录）](https://blog.csdn.net/weixin_44757863/article/details/122378799)
* [项目实战：使用JUC的CompletableFuture执行任务，并根据返回值执行异步回调](https://blog.csdn.net/weixin_44757863/article/details/122370050)
* [记一次高并发下的JVM性能优化（MemoryAnalyzer，jstat，jmap）](https://blog.csdn.net/weixin_44757863/article/details/122330016)

[2022年3篇](https://blog.csdn.net/weixin_44757863?type=blog&year=2022&month=01)

[2021年29篇](https://blog.csdn.net/weixin_44757863?type=blog&year=2021&month=12)

[2020年51篇](https://blog.csdn.net/weixin_44757863?type=blog&year=2020&month=12)

**目录**

1. [前言](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t0)
   1. [目录](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t1)
2. [一、微服务技术栈导学](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t2)
3. [二、Dubbo&Zookeeper](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t3)
4. [三、微服务远程调用Demo——RestTemplate基本使用](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t4)
5. [四、Eureka注册中心](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t5)
6. [五、Ribbon负载均衡](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t6)
7. [六、Nacos注册中心](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t7)
   1. [6.1 安装启动](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t8)
   2. [6.2 Nacos自定义负载均衡策略](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t9)
   3. [6.3 Nacos实现配置热更新](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t10)
   4. [6.4 Nacos集群](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t11)
8. [七、Feign远程调用](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t12)
   1. [7.1 还原事故现场](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t13)
   2. [7.2 Feign自定义配置](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t14)
   3. [7.3 Feign性能优化](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t15)
   4. [7.4 Feign最佳实践](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t16)
9. [八、统一Gateway网关](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t17)
   1. [8.1 概述](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t18)
   2. [8.2 搭建网关服务](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t19)
   3. [8.3 路由过滤](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t20)
      1. [8.3.1 断言工厂：对请求进行过滤](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t21)
      2. [8.3.2 过滤器GatewayFilter：对请求和响应进行过滤](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t22)
      3. [8.3.3 全局过滤器GlobalFilter：可以自定义过滤逻辑代码实现](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t23)
      4. [8.3.4 过滤器链执行顺序](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t24)
   4. [8.4 网关跨域问题处理](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t25)
10. [九、Docker](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t26)
    1. [9.1 Docker概念](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t27)
    2. [9.2 Docker常用命令](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t28)
11. [十、MQ(Message Queue)消息队列](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t29)
    1. [10.1 概述](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t30)
    2. [10.2 RabbitMQ安装](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t31)
    3. [10.3 常见消息模型](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t32)
       1. [10.3.1 简单队列模型](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t33)
    4. [10.4 Spring AMQP](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t34)
       1. [10.3.2 WorkQueue模型](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t35)
       2. [10.3.3 发布-订阅模型](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t36)
       3. [10.3.4 消息转换器](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t37)
12. [十一、ElasticSearch分布式搜索](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t38)
    1. [11.1 ES基础概念](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t39)
    2. [11.2 安装部署ES](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t40)
    3. [[Debug] 停止ES容器（或是重启Linux）后，如何恢复Docker网络：](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t41)
    4. [11.3 索引库操作](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t42)
    5. [11.4 文档操作](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t43)
    6. [11.5 RestClient操作索引库和文档](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t44)
    7. [11.6 DSL查询语法](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t45)
    8. [11.7 Java RestClient查询语法](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t46)
    9. [11.8 ES综合案例：黑马旅游](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t47)
    10. [11.9 ES数据聚合](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t48)
    11. [11.10 ES数据补全](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t49)
    12. [11.11 ES与MySQL之间数据同步（面试常问）](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t50)
    13. [11.12 搭建高可用ES集群](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t51)
13. [后记](https://blog.csdn.net/weixin_44757863/article/details/120959505?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164231659816780357258116%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=164231659816780357258116&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_ecpm_v1~rank_v31_ecpm-4-120959505.nonecase&utm_term=%E9%BB%91%E9%A9%ACspringcloud&spm=1018.2226.3001.4450#t52)



**分类专栏**

* [debug专题16篇](https://blog.csdn.net/weixin_44757863/category_10469385.html)
* [JavaSE3篇](https://blog.csdn.net/weixin_44757863/category_11518437.html)
* [JVM1篇](https://blog.csdn.net/weixin_44757863/category_11575504.html)
* [Linux11篇](https://blog.csdn.net/weixin_44757863/category_10482692.html)
* [数据结构4篇](https://blog.csdn.net/weixin_44757863/category_10683144.html)
* [Git2篇](https://blog.csdn.net/weixin_44757863/category_11508212.html)
* [Go2篇](https://blog.csdn.net/weixin_44757863/category_11365239.html)
* [微服务2篇](https://blog.csdn.net/weixin_44757863/category_11325074.html)
* [SpringBoot9篇](https://blog.csdn.net/weixin_44757863/category_10495455.html)
* [Manjaro3篇](https://blog.csdn.net/weixin_44757863/category_11363825.html)
* [实用工具插件5篇](https://blog.csdn.net/weixin_44757863/category_10584508.html)
* [Docker3篇](https://blog.csdn.net/weixin_44757863/category_11307838.html)
* [中间件1篇](https://blog.csdn.net/weixin_44757863/category_11336609.html)
* [前端19篇](https://blog.csdn.net/weixin_44757863/category_10473045.html)
* [效率工作2篇](https://blog.csdn.net/weixin_44757863/category_10932996.html)
* [设计模式，架构3篇](https://blog.csdn.net/weixin_44757863/category_10476352.html)
* [数据库12篇](https://blog.csdn.net/weixin_44757863/category_10519969.html)
* [个人总结1篇](https://blog.csdn.net/weixin_44757863/category_10692791.html)
* [开发工具兼容问题(版本冲突)5篇](https://blog.csdn.net/weixin_44757863/category_10461649.html)
* [SSM9篇](https://blog.csdn.net/weixin_44757863/category_10465673.html)
* [安全框架2篇](https://blog.csdn.net/weixin_44757863/category_10616288.html)
* [Maven4篇](https://blog.csdn.net/weixin_44757863/category_10458227.html)
* [学习方法2篇](https://blog.csdn.net/weixin_44757863/category_10509349.html)

举报