

搜索

# 分布式链路追踪 SkyWalking 源码分析 —— Collector 初始化

点击上方“芋道源码”，选择“设为星标” (https://mp.weixin.qq.com/s?

\_\_biz=MzUzMtA2NTU2Ng%3D%3D&chksm=fa49755dcd3efc4bf4f566fbbb74c191d0b79f2d3222fd211bc52d80b5ef127f52b1158ed71&idx=3&mid=2247486188&scene=21&sn=f160d91ea23e5113e6077c500a2e30c4#wechat\_redirect)

做积极的人，而不是积极废人！

## 源码精品专栏

- 中文详细注释的开源项目 (https://mp.weixin.qq.com/s?\_\_biz=MzUzMtA2NTU2Ng%3D%3D&chksm=fa497489cd3efd9f7298f5da6aad0c443ae15f398436aff57cb2b734d6689e62ab43ae7857ac&idx=1&mid=2247486264&scene=21&sn=475ac3f1ef253a33daacf50477203c80#wechat\_redirect)
- 消息中间件 RocketMQ 源码解析 (https://blog.csdn.net/github\_38592071/article/details/72599545)
- 数据库中间件 Sharding-JDBC 和 MyCAT 源码解析 (https://mp.weixin.qq.com/s?\_\_biz=MzUzMtA2NTU2Ng%3D%3D&chksm=fa497480cd3efd96fe17975b0b8b141e87fd0a62673e6a30b501460de80b3eb997056f09de08&idx=1&mid=2247486257&scene=21&sn=4d3c9c675f8833157641a2e0b48e498c#wechat\_redirect)
- 作业调度中间件 Elastic-Job 源码解析 (https://mp.weixin.qq.com/s?\_\_biz=MzUzMtA2NTU2Ng%3D%3D&chksm=fa497483cd3efd950514da5a37160e7fd07f0a96f39265cf7ba3721985e5aadbdcbe7aafc34a&idx=1&mid=2247486258&scene=21&sn=ae5665ae9c3002b53f87cab44948a096#wechat\_redirect)
- 分布式事务中间件 TCC-Transaction 源码解析 (https://mp.weixin.qq.com/s?\_\_biz=MzUzMtA2NTU2Ng%3D%3D&chksm=fa497482cd3efd9402d71469f71863f71a6998b27e12ca2e00446b8178d79dcef0721d8e570a&idx=1&mid=2247486259&scene=21&sn=b023cf3dbf97e5da59db2f4ee632f5a6#wechat\_redirect)
- Eureka 和 Hystrix 源码解析 (https://mp.weixin.qq.com/s?\_\_biz=MzUzMtA2NTU2Ng%3D%3D&chksm=fa497485cd3efd93937143a648bc1b530bc7d1f6f8ad4bf2ec112ffe34dee80b474605c22db0&idx=1&mid=2247486260&scene=21&sn=8f14c0c191d6f8df6eb34202f4ad9708#wechat\_redirect)

- Java 并发源码 ([https://mp.weixin.qq.com/s?\\_\\_biz=MzUzMtA2NTU2Ng%3D%3D&chksm=fa497484cd3efd92352d6fb3d05ccbaebca2fafed6f18edbe5be70c99ba088db5c8a7a8080c1&idx=1&mid=2247486261&scene=21&sn=bd69f26aadfc826f6313ffbb95e44ee5#wechat\\_redirect](https://mp.weixin.qq.com/s?__biz=MzUzMtA2NTU2Ng%3D%3D&chksm=fa497484cd3efd92352d6fb3d05ccbaebca2fafed6f18edbe5be70c99ba088db5c8a7a8080c1&idx=1&mid=2247486261&scene=21&sn=bd69f26aadfc826f6313ffbb95e44ee5#wechat_redirect))

摘要: 原创出处 <http://www.iocoder.cn/SkyWalking/collector-init/> 「芋道源码」欢迎转载, 保留摘要, 谢谢!

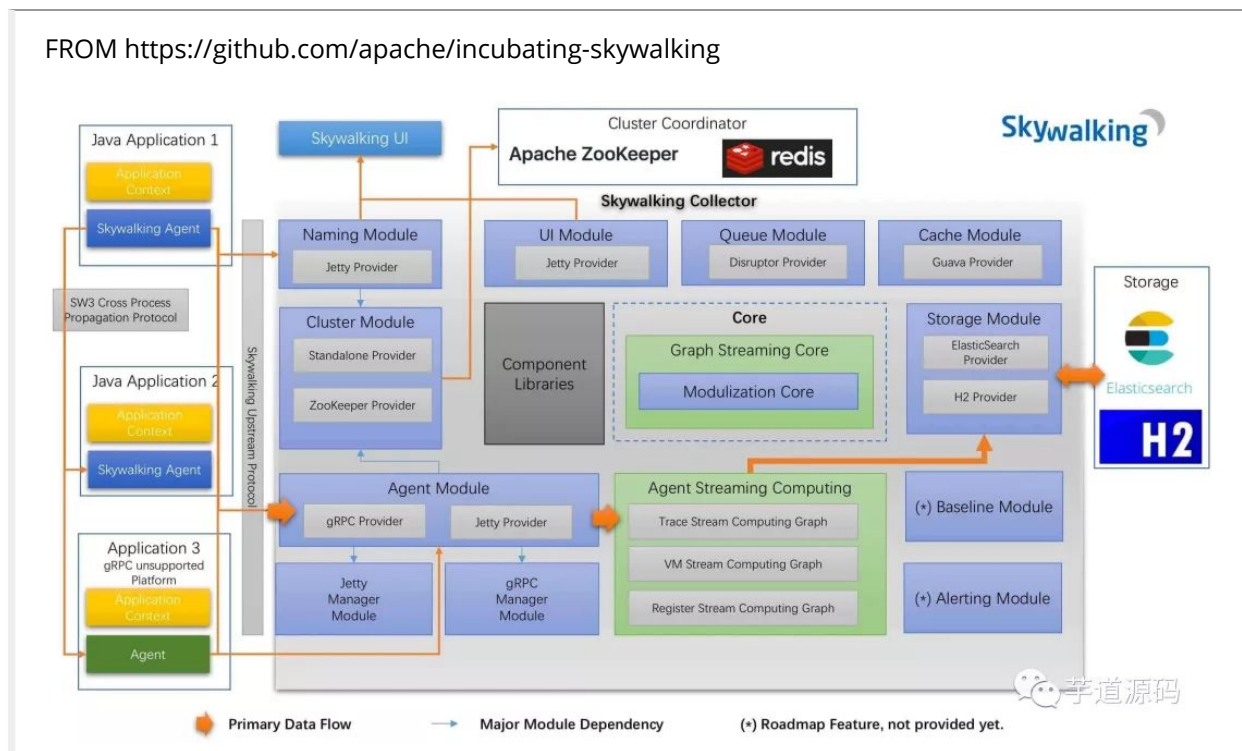
本文主要基于 SkyWalking 3.2.6 正式版

- 1. 概述
- 2. CollectorBootStartUp
- 2. ApplicationConfigLoader
- 3. ModuleManager
  - 3.1 Module
  - 3.2 ModuleProvider
  - 3.3 Service
  - 3.4 BootstrapFlow
- 4. Module 实现类简介

## 1. 概述

本文主要分享 SkyWalking Collector 启动初始化的过程。在分享的过程中, 我们会简单介绍 Collector 每个模块及其用途。

ps: Collector 是 SkyWalking 的 Server 端。整体如下图:



## 2. CollectorBootStartUp

`org.skywalking.apm.collector.boot.CollectorBootStartUp`, 在 `apm-sniffer/apm-agent` Maven 模块项目里, SkyWalking Collector 启动入口。

`#main(args)` 方法, 启动 Collector, 代码如下:

- 第 45 行：调用 `ApplicationConfiguration#load()` 方法，加载 Collector 配置。
- 第 47 行：调用 `ModuleManager#init(...)` 方法，初始化 Collector 组件们。
- 第 60 行：调用 `Thread#sleep(60000)` 方法，等待 Collector 内嵌的 Jetty Server 启动完成。

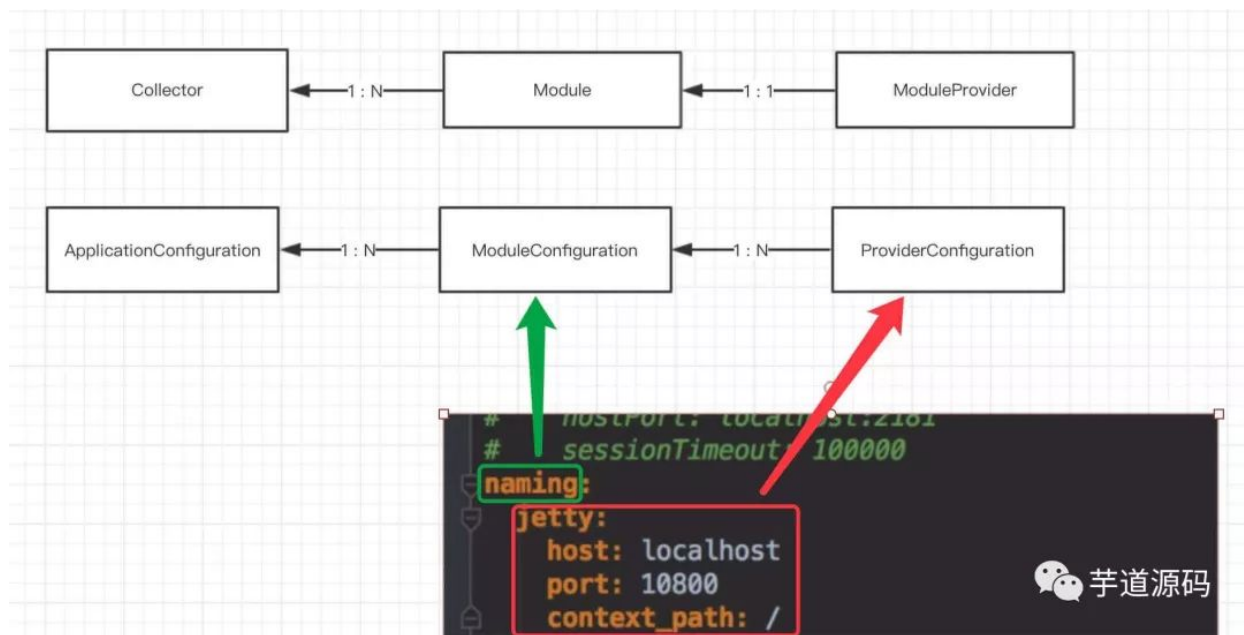
## 2. ApplicationConfigLoader

`org.skywalking.apm.collector.boot.config.ApplicationConfigLoader`，实

现 `org.skywalking.apm.collector.boot.config.ConfigLoader` 接口，Collector 配置

(`org.skywalking.apm.collector.core.module.ApplicationConfiguration`)加载器。

在看具体代码实现之前，我们先了解下 `ApplicationConfiguration` 整体类结构。如下图所示：



- Collector 使用组件管理器( `ModuleManager` )，管理多个组件( `Module` )。
  - 一个组件有多种组件服务提供者( `ModuleProvider` )，**同时**一个组件只允许使用**一个**组件服务提供者。这块下面会有代码解析说明。
- Collector 使用一个应用配置类( `ApplicationConfiguration` )。
  - 一个应用配置类包含多个组件配置类( `ModuleConfiguration` )。每个组件对应一个组件配置类。
  - 一个组件配置类包含多个组件服务提供者配置( `ProviderConfiguration` )。每个组件服务提供者对应一个组件配置类。**注意**：因为一个组件只允许**同时**使用**一个**组件服务提供者，所以一个组件配置类**只设置**一个组件服务提供者配置。
- 整个配置文件，对应应用配置类。**绿框**部分，对应一个组件配置类。**红框**部分，对应一个组件服务提供者配置类。

下面，我们来看看 `ApplicationConfigLoader#load()` 方法，代码如下：

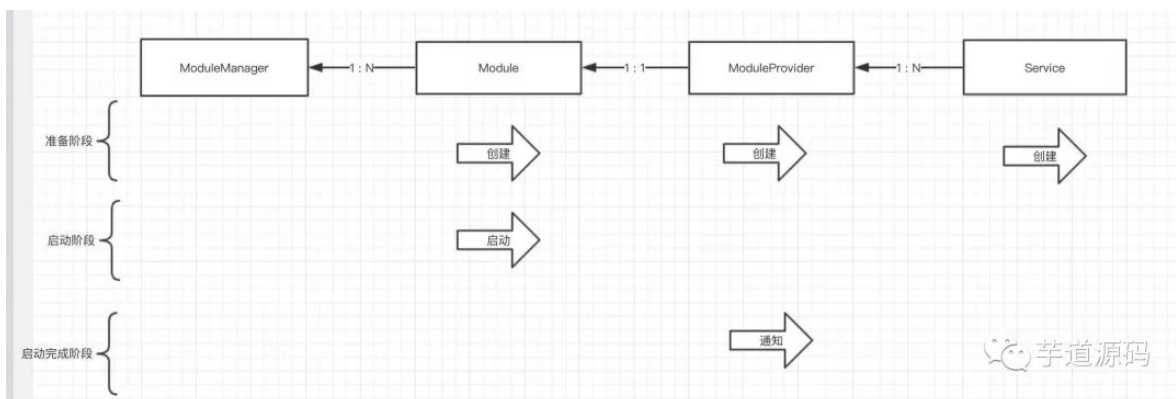
- 第 47 行：调用 `#loadConfig()` 方法，从 `apm-collector-core` 的 ``application.yml`` 加载自定义配置。
- 第 49 行：调用 `#loadDefaultConfig()` 方法，从 `apm-collector-core` 的 ``application-default.yml`` 加载默认配置。
- 两个方法逻辑基本一致，已经添加代码注释，胖友自己阅读理解。

## 3. ModuleManager

`org.skywalking.apm.collector.core.module.ModuleManager`，组件管理器，负责组件的管理与初始化。

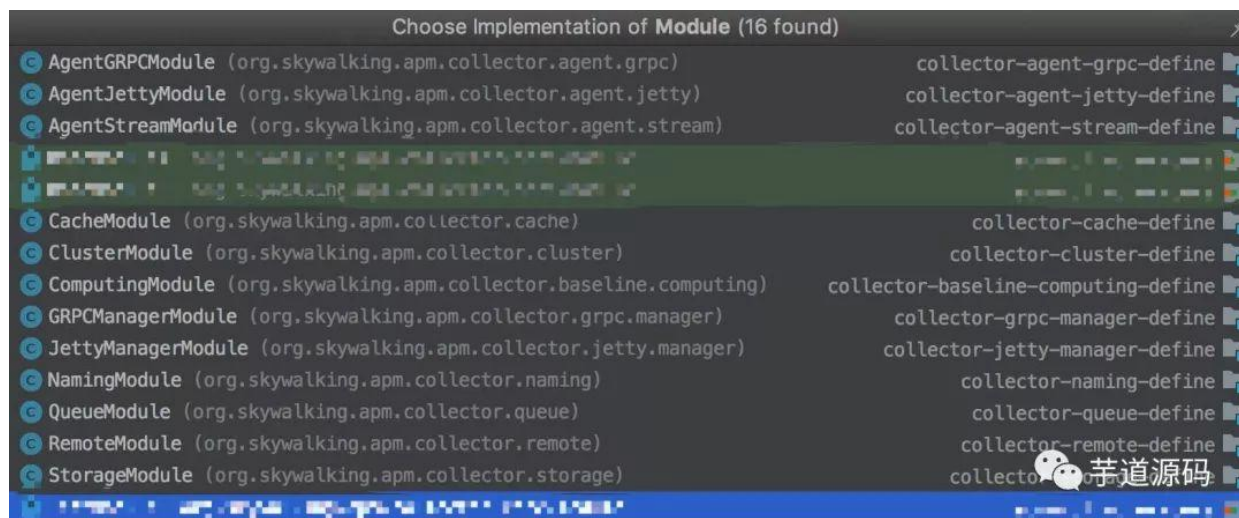
`#init()` 方法，初始化组件们，代码如下：

- 第 51 至 53 行：调用 `java.util.ServiceLoader#load(Module.class)` 方法，加载所有 `Module` 实现类的实例数组。  
 ServiceManager 基于 SPI (Service Provider Interface) 机制，在每个 `apm-collector-xxx-define` 项目的 `/resources/META-INF/services/org.skywalking.apm.collector.core.module.Module` 文件里，定义了该项目 `Module` 的实现类。如果胖友对 SPI 机制不熟悉，可以看下如下文章：
  - 《SPI 和 ServiceLoader》
  - 《跟我学Dubbo系列之Java SPI机制简介》
- 第 55 至 75 行：遍历所有 `Module` 实现类的实例数组，创建在配置中的 `Module` 实现类的实例，并执行 `Module` 准备阶段的逻辑，后添加到加载的组件实例的映射( `loadedModules` )。
  - 第 59 至 67 行：创建 `Module` 对象。
  - 第 69 行：调用 `Module#prepare(...)` 方法，执行 `Module` 准备阶段的逻辑。在改方法内部，会创建 `Module` 对应的 `ModuleProvider`。在「3.1 Module」详细解析。
  - 第 71 行：添加到 `loadedModules`。
- 第 77 至 80 行：校验在配置中的 `Module` 实现类的实例都创建了，否则抛出异常。
- 第 84 行：调用 `BootstrapFlow#start(...)` 方法，执行 `Module` 启动逻辑。「3.4 BootstrapFlow」详细解析。
- 第 86 行：调用 `BootstrapFlow#notifyAfterCompleted()` 方法，执行 `Module` 启动完成，通知 `ModuleProvider`。「3.4 BootstrapFlow」详细解析。
- 总的来说，`Module` 初始化的过程，可以理解成三个阶段，如下图所示：



## 3.1 Module

`org.skywalking.apm.collector.core.module.Module`，组件抽象类。通过实现 `Module` 抽象类，实现不同功能的组件。目前 Collector 的 `Module` 实现类如下图：



`#name()` 抽象方法，获得组件名。目前组件名有：





`#providers()` 方法，获得 `ModuleProvider` 数组。实际上，一个 `Module` 同时只能有一个 `ModuleProvider`，参见 `#provider()` 方法。

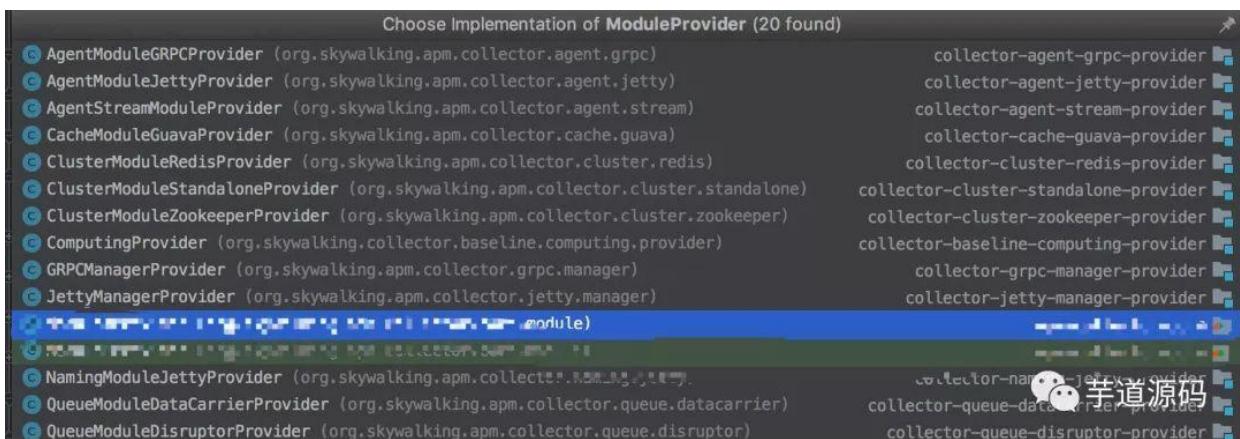
`#services()` 抽象方法，获得 `Service` 类数组。具体 `Service` 对象，在 `ModuleProvider` 对象里获取，参见 `#getService(serviceType)` 方法。

`#prepare(...)` 方法，执行 `Module` 准备阶段的逻辑，代码如下：

- 第 69 行：调用 `java.util.ServiceLoader#load(ModuleProvider.class)` 方法，加载所有 `ModuleProvider` 实现类的实例数组。  
ServiceManager 基于 SPI (Service Provider Interface) 机制，在每个 `apm-collector-xxx-yyy-provider` 项目的 `/resources/META-INF.services/org.skywalking.apm.collector.core.module.ModuleProvider` 文件里，定义了该项目的 `ModuleProvider` 的实现类。
- 第 72 至 93 行：遍历所有 `ModuleProvider` 实现类的实例数组，创建在配置中的 `ModuleProvider` 实现类的实例，后添加到加载的组件服务提供者实例的映射 (`loadedProviders`)。
- 第 95 至 98 行：校验有 `ModuleProvider` 初始化，否则抛出异常。
- 第 100 至 104 行：调用 `ModuleProvider#prepare(...)` 方法，执行 `ModuleProvider` 准备阶段的逻辑。在改方法内部，会创建 `ModuleProvider` 对应的 `Service`。在「3.2 ModuleProvider」详细解析。

## 3.2 ModuleProvider

`org.skywalking.apm.collector.core.module.ModuleProvider`，组件服务提供者抽象类。通过实现 `ModuleProvider` 抽象类，实现不同功能的组件服务提供者。目前 Collector 的 `ModuleProvider` 实现类如下图：



`#name()` 抽象方法，获得组件服务提供者名。目前组件服务提供者名有：



`#module()` **抽象方法**，获得 `ModuleProvider` 对应的 `Module` 类。注意，`ModuleProvider` 的名字可以重复，例如上图的 `jetty`，通过对应的 `Module` 类来区分。

`#requiredModules()` **抽象方法**，获得 `ModuleProvider` 依赖的 `Module` 名字数组。

----- Service 相关方法 Begin -----

`#registerServiceImplementation(Class<? extends Service>, Service)` 方法，注册 `Service` 对象。一个 `ModuleProvider` 可以有 0 到 N 个 `Service` 对象。

`#getService(Class<T>)` 方法，获得 `Service` 对象。

`#requiredCheck(...)` 方法，**校验** `ModuleProvider` 包含的 `Service` 们都创建成功。

- **方法参数**，从 `Module#services()` 方法获得。
- 该方法会被 `BootstrapFlow#start()` 方法调用，在「3.4 BootstrapFlow」详细解析。

----- Service 相关方法 End -----

`#prepare(Properties)` **抽象方法**，执行 `ModuleProvider` 准备阶段的逻辑：`Service` 的创建，私有变量的创建等等。例如，`StorageModuleH2Provider#prepare(Properties)`。

`#start(Properties)` **抽象方法**，执行 `ModuleProvider` 启动阶段的逻辑：私有变量的初始化等等。例如，`StorageModuleH2Provider#start(Properties)`。

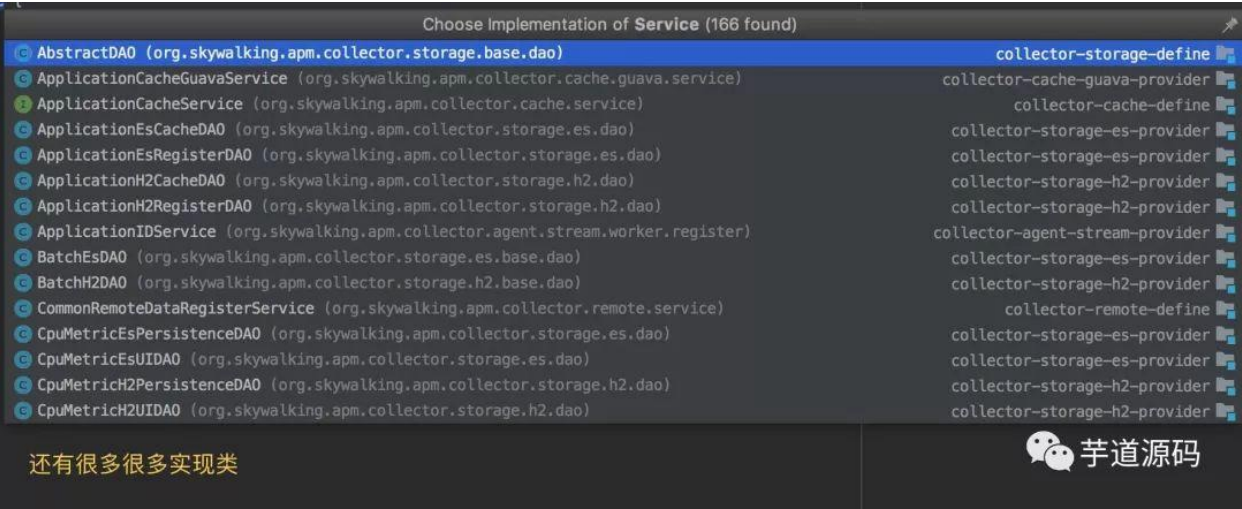
- 该方法会被 `BootstrapFlow#start()` 方法调用，在「3.4 BootstrapFlow」详细解析。

`#notifyAfterCompleted()` **抽象方法**，执行 `ModuleProvider` 启动完成阶段的逻辑：私有变量的初始化等等。例如，`StorageModuleEsProvider#notifyAfterCompleted(Properties)`。

- 该方法会被 `BootstrapFlow#notifyAfterCompleted()` 方法调用，在「3.4 BootstrapFlow」详细解析。

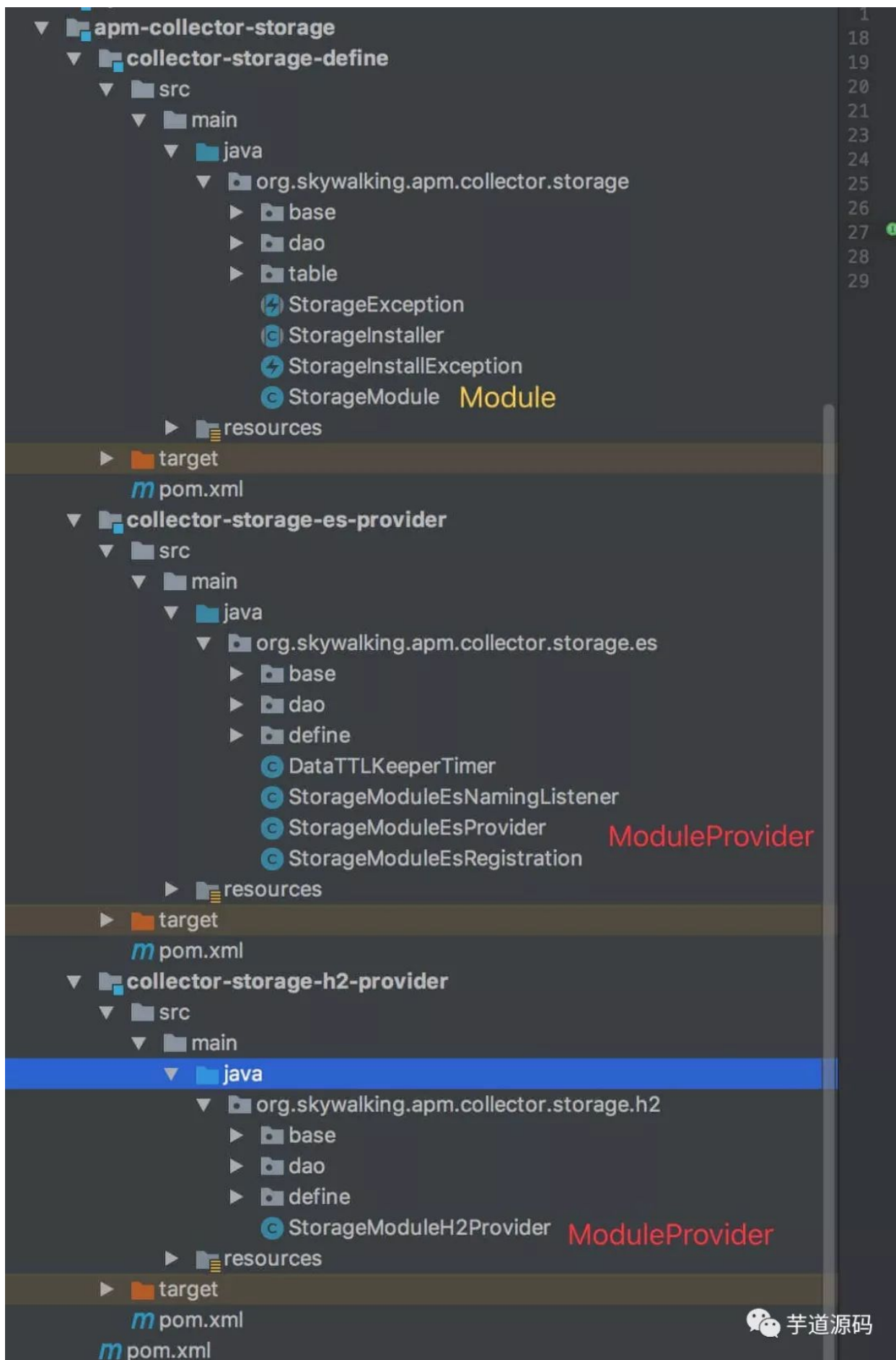
### 3.3 Service

`org.skywalking.apm.collector.core.module.Service` ，服务接口。通过实现 Service 接口，实现不同功能的服务。目前 Collector 的 Service 实现类如下图：



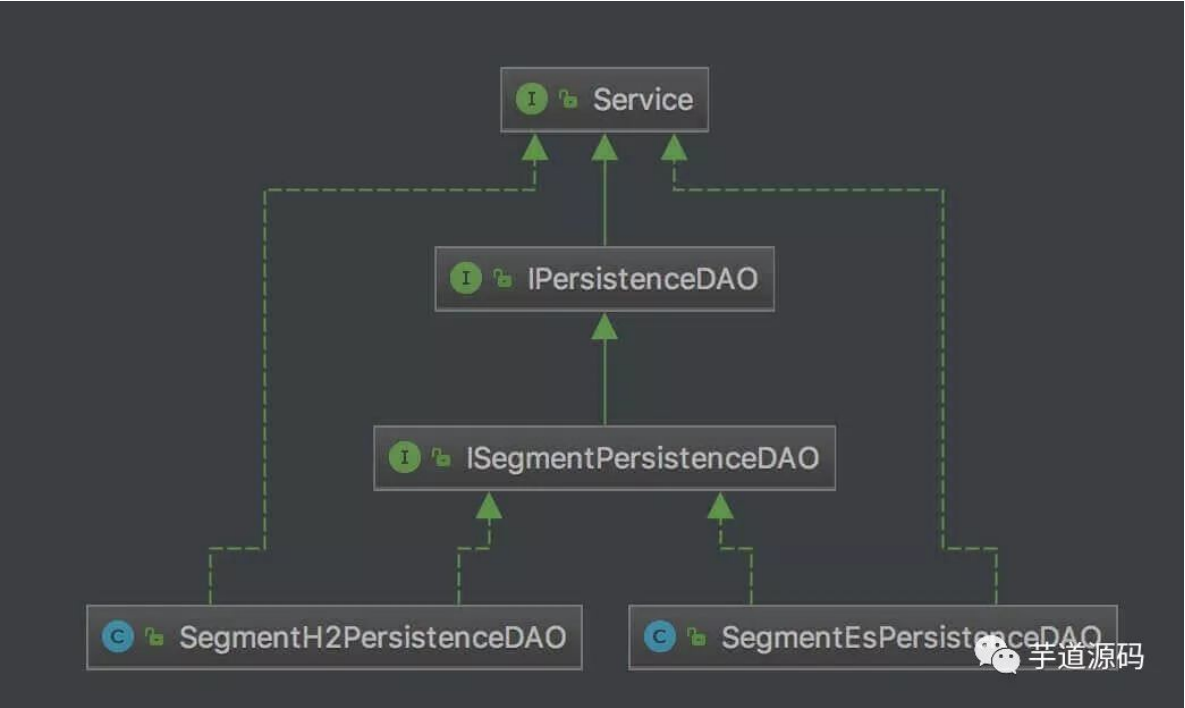
这里有一点要注意下，实际上 Module 是与 Service "直接" 一对多的关系。中间 有一层 ModuleProvider 存在的原因是，相同 Module 可以有多种 ModuleProvider 实现，而 ModuleProvider 提供提供相同功能的 Service ，但是实现不同。

以 `apm-collector-storage` 举例子，如下图所示：

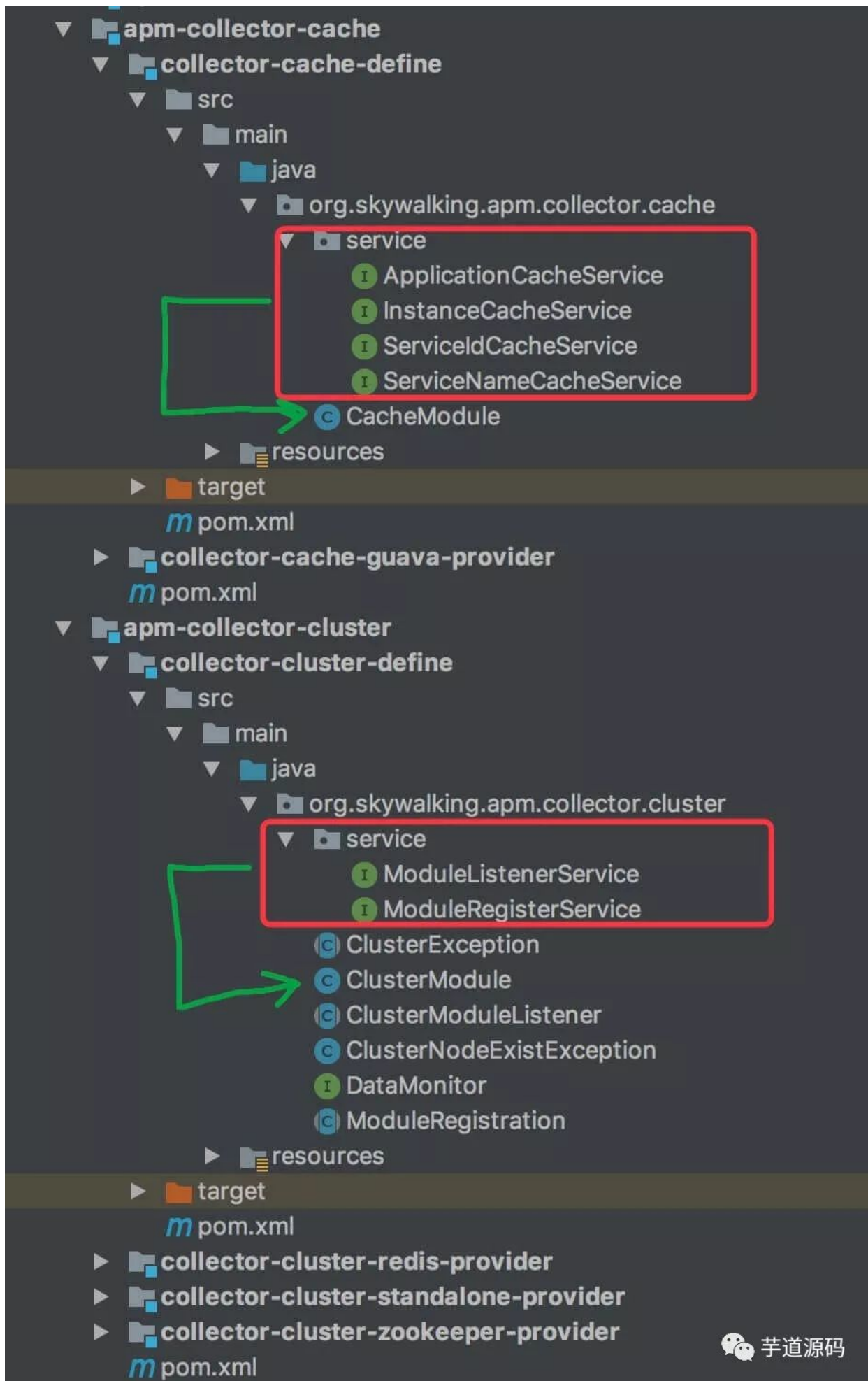


- StorageModuleEsProvider / StorageModuleH2Provider 分别基于 ES / H2 实现，其提供存储相同数据的不同实现。例如：





一般 collector-xxx-define 的 service 包下，会定义当前模块提供的 Service 接口，如下图所示：



这也是为什么有 `Module#services()` 和 `#requiredCheck(Class<? extends Service>[])` 这样的方法涉及的原因。

另外，如下是 Service 接口的解释：

The Service implementation is a service provided by its own modules.

And every {@link ModuleProvider} must provide all the given services of the {@link Module}.

## 3.4 BootstrapFlow

org.skywalking.apm.collector.core.module.BootstrapFlow , 组件启动流程。

BootstrapFlow **构造方法**, 调用 #makeSequence() 方法, 获得 ModuleProvider 启动顺序, 这个是该类的**重点**。

#start() 方法, 执行 Module 启动逻辑。

- 第 54 至 63 行: 校验**依赖** Module 已经都存在。
- 第 67 行: 校验 ModuleProvider 包含的 Service 们都**创建成功**。
- 第 70 行: 调用 ModuleProvider#start(...) 方法, 执行 ModuleProvider 启动阶段逻辑。

#notifyAfterCompleted() 方法, 调用 ModuleProvider#notifyAfterCompleted() 方法, 执行 ModuleProvider 启动完成阶段的逻辑。

## 4. Module 实现类简介

- Naming Module : 《SkyWalking 源码分析 —— Collector Naming Server 命名服务》
- UI Module :
  - 《SkyWalking 源码分析 —— 运维界面 (一) 之应用视角》
  - 《SkyWalking 源码分析 —— 运维界面 (二) 之应用实例视角》
  - 《SkyWalking 源码分析 —— 运维界面 (三) 之链路追踪视角》
  - 《SkyWalking 源码分析 —— 运维界面 (四) 之操作视角》
- Queue Module : 《SkyWalking 源码分析 —— Collector Queue 队列组件》
- Cache Module : 《SkyWalking 源码分析 —— Collector Cache 缓存组件》
- Cluster Module : 《SkyWalking 源码分析 —— Collector Cluster 集群管理》
- Component Libraries : 《SkyWalking 源码分析 —— Collector Client Component 客户端组件》、《SkyWalking 源码分析 —— Collector Server Component 服务器组件》
- Core :
  - 《SkyWalking 源码分析 —— Collector Storage 存储组件》 [2. apm-collector-core]
  - 《SkyWalking 源码分析 —— Collector 初始化》 [3. ModuleManager]
- Storage Module : 《SkyWalking 源码分析 —— Collector Storage 存储组件》
  - 《SkyWalking 源码分析 —— Collector Streaming Computing 流式处理 (一) 》 [2. apm-collector-core/graph]
  - 《SkyWalking 源码分析 —— Collector Streaming Computing 流式处理 (二) 》 [2. Data]
- Agent Module : 参见 Agent Streaming Computing 。
- Jetty Manager Module : 《SkyWalking 源码分析 —— Collector Jetty Server Manager》
- gRPC Manager Module : 《SkyWalking 源码分析 —— Collector gRPC Server Manager》
- Agent Streaming Computing :
  - 《SkyWalking 源码分析 —— Collector Streaming Computing 流式处理 (一) 》 [2. apm-collector-core/graph]
  - 《SkyWalking 源码分析 —— Collector Streaming Computing 流式处理 (二) 》 [2. Data]

- 《SkyWalking 源码分析 —— Collector Remote 远程通信服务》
  - 《SkyWalking 源码分析 —— Agent 收集 Trace 数据》
  - 《SkyWalking 源码分析 —— Agent 发送 Trace 数据》
  - 《SkyWalking 源码分析 —— Collector 接收 Trace 数据》
  - 《SkyWalking 源码分析 —— Collector 存储 Trace 数据》
- Baseline Module : TODO 【4001】
  - Alerting Module : TODO 【4001】

欢迎加入我的知识星球，一起探讨架构，交流源码。加入方式，**长按下方二维码**噢：



已在知识星球更新源码解析如下：



《精尽面试题（附答案）》	《精尽学习指南（附视频）》
01. Dubbo 面试题 02. Netty 面试题 03. Spring 面试题 04. Spring MVC 面试题 05. Spring Boot 面试题 06. Spring Cloud 面试题 07. MyBatis 面试题 08. 消息队列面试题 09. RocketMQ 面试题 10. RabbitMQ 面试题 11. Kafka 面试题 12. 缓存面试题 13. Redis 面试题 14. MySQL 面试题 15. 【分库分表】面试题 16. 【分布式事务】面试题 17. Elasticsearch 面试题 18. MongoDB 面试题 19. 设计模式面试题 20. Java 【基础】面试题 21. Java 【集合】面试题 22. Java 【并发】面试题 23. Java 【虚拟机】面试题 24. Linux 面试题 25. Git 面试题 26. 计算机网络面试题 27. Maven 面试题 28. Jenkins 面试题 29. Zookeeper 面试题 30. Nginx 面试题	00. 精尽学习指南 —— 路线 01. Dubbo 学习指南 02. Netty 学习指南 03. Spring 学习指南 04. Spring MVC 学习指南 05. Spring Boot 学习指南 06. Spring Cloud 学习指南 07. MyBatis 学习指南 08. RocketMQ 学习指南 09. RabbitMQ 学习指南 10. Kafka 学习指南 11. Redis 学习指南 12. MySQL 学习指南 13. MongoDB 学习指南 14. Elasticsearch 学习指南 15. 设计模式学习指南 16. Java 【基础】学习指南 17. Java 【并发】学习指南 18. Java 【虚拟机】学习指南 19. Linux 学习指南 20. 数据结构与算法学习指南 21. 计算机网络学习指南 22. Maven 学习指南 23. Jenkins 学习指南 24. Git 学习指南 25. IntelliJ IDEA 学习指南 26. Docker 学习指南 27. Kubernetes 学习指南 28. Zookeeper 学习指南 29. Nginx 学习指南



《Dubbo 源码解析》	《Netty 源码解析》
01. 调试环境搭建 02. 项目结构一览 03. 配置 Configuration 04. 核心流程一览 05. 拓展机制 SPI 06. 线程池 ThreadPool 07. 服务暴露 Export 08. 服务引用 Refer 09. 注册中心 Registry 10. 动态编译 Compile 11. 动态代理 Proxy 12. 服务调用 Invoke 13. 调用特性 14. 过滤器 Filter 15. NIO 服务器 16. P2P 服务器 17. HTTP 服务器 18. 序列化 Serialization 19. 集群容错 Cluster 20. 优雅停机 Shutdown 21. 日志适配 Logging 22. 状态检查 Status 23. 监控中心 Monitor 24. 管理中心 Admin 25. 运维命令 QOS 26. 链路追踪 Tracing 27. Spring Boot 集成 28. Spring Cloud 集成 ... 一共 <b>73+</b> 篇	01. 调试环境搭建 02. NIO 基础 03. Netty 简介 04. 启动 Bootstrap 05. 事件轮询 EventLoop 06. 通道管道 ChannelPipeline 07. 通道 Channel 08. 字节缓冲区 ByteBuf 09. 通道处理器 ChannelHandler 10. 编解码 Codec 11. 工具类 Util ... 一共 <b>61+</b> 篇
	《MyBatis 源码解析》
	01. 调试环境搭建 02. 项目结构一览 03. MyBatis 初始化 04. SQL 初始化 05. SQL 执行 06. 插件体系 07. Spring 集成 ... 一共 <b>34+</b> 篇
	芋道源码

《Spring 源码解析》	《Spring MVC 源码解析》
01. 调试环境搭建 02. IoC Resource 定位 03. IoC BeanDefinition 载入 04. IoC BeanDefinition 注册 05. IoC Bean 获取 06. IoC Bean 生命周期 07. AOP 源码导读 08. Transaction 源码导读 ... 一共 <b>46+</b> 篇	01. 调试环境搭建 02. 容器的初始化 03. 组件一览 04. 请求处理一览 05. HandlerMapping 组件 06. HandlerAdapter 组件 07. HandlerExceptionResolver 组件 08. RequestToViewNameTranslator 组件 09. LocaleResolver 组件 10. ThemeResolver 组件 11. ViewResolver 组件 12. MultipartResolver 组件 13. FlashMapManager 组件 ... 一共 <b>24+</b> 篇
《Spring Boot 源码解析》	《数据库实体设计》
01. 调试环境搭建 02. 项目结构一览 03. SpringApplication 04. 自动配置 05. Condition 06. ServletWebServerApplicationContext 07. ReactiveWebServerApplicationContext 08. ApplicationContextInitializer 09. ApplicationListener ... 一共 <b>15+</b> 篇 (努力更新中)	01. 商品模块 02. 交易模块 03. 营销模块 04. 公用模块 ... 一共 <b>17+</b> 篇
	芋道源码

如果你喜欢这篇文章，喜欢，转发。

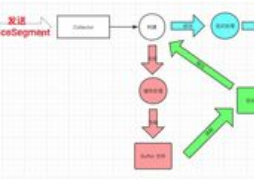
生活很美好，明天见(。ω。)/♡

(<https://creativecommons.org/licenses/by-sa/4.0/>) 版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA (<https://creativecommons.org/licenses/by-sa/4.0/>)版权协议，转载请附上原文出处链接和本声明。

本文链接：[https://blog.csdn.net/github\\_38592071/article/details/103371204](https://blog.csdn.net/github_38592071/article/details/103371204) ([https://blog.csdn.net/github\\_38592071/article/details/103371204](https://blog.csdn.net/github_38592071/article/details/103371204))

原作者删帖 (<https://www.pianshen.com/copyright.html#del>) 不实内容删帖 (<https://www.pianshen.com/copyright.html#others>) 广告或垃圾文章投诉 (<mailto:pianshen@gmx.com?subject=投诉本文含广告或垃圾信息> (请附上违规链接地址) )

智能推荐

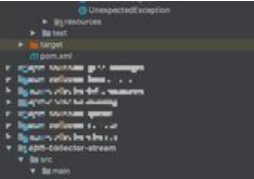


(</article/74331466420/>)

分布式追踪 SkyWalking 源码分析六 Collector 接收和发送 trace 数据 (/article/74331466420/)

分布式链路追踪系统，链路的追踪大体流程如下：Agent 收集 Trace 数据。Agent 发送 Trace 数据给 Collector 。Collector 接收 Trace 数据。Collector 存储 Trace 数据到存储器，例如，数据库。Collector 接收到 TraceSegment 数据后，进行构建。【蓝色流程】构建成功，进行流式

处理，最终存储到存储器( 例如，ES...



(</article/7745987682/>)

链路追踪 SkyWalking 源码分析 —— Collector Streamin g Computing 流式处理 (二) (/article/7745987682/)

点击上方“芋道源码”，选择“设为星标” 做积极的人，而不是积极废人！ 源码精品专栏 中文详细注释的开源项目 RPC 框架 Dubbo 源码解析 网络应用框架 Netty 源码解析 消息中间件 RocketMQ 源码解析 数据库中间件 Sharding-JDBC 和 MyCAT 源码解析 作业调度中间件 Elastic-Job 源码解析

分...



(</article/9983987748/>)

链路追踪 SkyWalking 源码分析 —— Collector Naming Server 命名服务 (/article/9983987748/)

点击上方“Java地基”，选择“设为星标” 做积极的人，而不是积极废人！ 源码精品专栏 中文详细注释的开源项目 RPC 框架 Dubbo 源码解析 网络应用框架 Netty 源码解析 消息中间件 RocketMQ 源码解析 数据库中间件 Sharding-JDBC 和 MyCAT 源码解析 作业调度中间件 Elastic-Job 源码解

析...



(</article/1229987810/>)

链路追踪 SkyWalking 源码分析 —— Collector Server C omponent 服务器组件 (/article/1229987810/)

点击上方“芋道源码”，选择“设为星标” 做积极的人，而不是积极废人！ 源码精品专栏 中文详细注释的开源项目 RPC 框架 Dubbo 源码解析 网络应用框架 Netty 源码解析 消息中间件 RocketMQ 源码解析 数据库中间件 Sharding-JDBC 和 MyCAT 源码解析 作业调度中间件 Elastic-Job 源码解析

分...

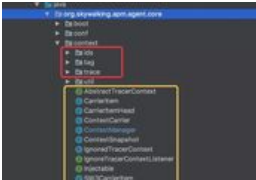
链路追踪 SkyWalking 源码分析 —— Collector Client Component 客户端组件 (/article/9542987830/)



(/article/9542987830/)

点击上方“芋道源码”，选择“设为星标” 做积极的人，而不是积极废人！ 源码精品专栏 中文详细注释的开源项目 RPC 框架 Dubbo 源码解析 网络应用框架 Netty 源码解析 消息中间件 RocketMQ 源码解析 数据库中间件 Sharding-JDBC 和 MyCAT 源码解析 作业调度中间件 Elastic-Job 源码解析 分...

猜你喜欢

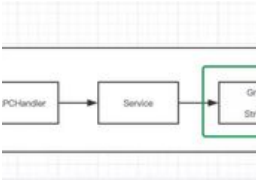


(/article/9446987572/)

分布式链路追踪 SkyWalking 源码分析 —— Agent 收集 Trace 数据 (/article/9446987572/)

点击上方“芋道源码”，选择“设为星标” 做积极的人，而不是积极废人！ 源码精品专栏 中文详细注释的开源项目 RPC 框架 Dubbo 源码解析 网络应用框架 Netty 源码解析 消息中间件 RocketMQ 源码解析 数据库中间件 Sharding-JDBC 和 MyCAT 源码解析 作业调度中间件 Elastic-Job 源码解析

分...

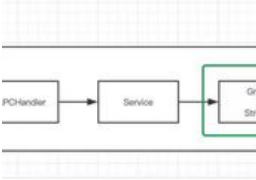


(/article/8919987593/)

分布式链路追踪 SkyWalking 源码分析 —— Agent DictionaryManager 字典管理 (/article/8919987593/)

点击上方“芋道源码”，选择“设为星标” 做积极的人，而不是积极废人！ 源码精品专栏 中文详细注释的开源项目 RPC 框架 Dubbo 源码解析 网络应用框架 Netty 源码解析 消息中间件 RocketMQ 源码解析 数据库中间件 Sharding-JDBC 和 MyCAT 源码解析 作业调度中间件 Elastic-Job 源码解析

分...



(/article/7242987602/)

分布式链路追踪 SkyWalking 源码分析 —— 应用于应用实例的注册 (/article/7242987602/)

点击上方“芋道源码”，选择“设为星标” 做积极的人，而不是积极废人！ 源码精品专栏 中文详细注释的开源项目 RPC 框架 Dubbo 源码解析 网络应用框架 Netty 源码解析 消息中间件 RocketMQ 源码解析 数据库中间件 Sharding-JDBC 和 MyCAT 源码解析 作业调度中间件 Elastic-Job 源码解析

分...

$$Y=c_k=\frac{\sum_{i=1}^N I(y_i=c_k)}{N}, \quad k=1,2,\cdots,K$$
$$X^{(j)}=a_{\beta} \{Y=c_k\}=\frac{\sum_{i=1}^N I(x_i^{(j)}=a_{\beta} y_i=c_k)}{\sum_{i=1}^N I(y_i=c_k)}$$
$$=1,2,\cdots,n; \quad l=1,2,\cdots,S_j; \quad k=1,2,\cdots,K$$

(/article/26251197687/)

统计学习方法读书笔记（四） (/article/26251197687/)

朴素贝叶斯法的学习与分类 基本方法： 假设输入空间是n维向量的集合， 输出空间x是定义在输入空间X上的随机变量， y是定义在输出空间Y的随机变量， P (X,Y) 是X,Y的联合分布。朴素贝叶斯就是通过训练数据集学习联合分布。实质就是学习先验概率和条件概率。先验概率如下： . 条件概率如下： 。通过这种方式就学习到了联合概率分布。朴素贝叶斯法对条件概率做了独立性

的假设， 具体独立性假设如下： 对于给定的输入x...



(/article/2994818442/)

zabbix添加触发器 (/article/2994818442/)

写触发器（一般和邮件一起用）报警邮件点我 某个分区容量少于10G提示你 下面的问题事件生成模式选多重（如果选单个， 触发器Top 100里就只存在几分钟） 低于70% 本身是有的， 是20% 复制一个， 改70%就行 &nbs...



# 赞助商广告

在百万程序员中推广你的产品 (mailto:pianshen@gmx.com?subject=申请广告合作)

## 相关文章

链路追踪 SkyWalking 源码分析 —— Agent 初始化 (/article/4315987861/)
链路追踪 SkyWalking 源码分析 —— Collector Queue 队列组件 (/article/6011408613/)
链路追踪 SkyWalking 源码分析 —— Collector Jetty Server Manager (/article/7967419332/)
链路追踪 SkyWalking 源码分析 —— Collector Storage 存储组件 (/article/8755987705/)
链路追踪 SkyWalking 源码分析 —— Collector gRPC Server Manager (/article/5459987760/)
链路追踪 SkyWalking 源码分析 —— Collector Jetty Server Manager (/article/5420987777/)
分布式追踪 SkyWalking 源码分析— Agent初始化 (/article/8789781211/)
skywalking原理_链路追踪 SkyWalking 源码分析——Collector Naming Server 命名服务 (/article/55302188178/)
分布式追踪 & APM 系统 SkyWalking 源码分析 —— Collector Streaming Computing 流式处理（二） (/article/9652750056/)
分布式追踪 & APM 系统 SkyWalking 源码分析 —— Collector Server Component 服务器组件 (/article/2592811670/)

## 热门文章

ubuntu制作usb启动盘 (/article/6748558624/)
发现最新的区块链应用-8月16日 (/article/4182609091/)
梯度下降法的三种形式批量梯度下降法、随机梯度下降以及小批量梯度下降法 (/article/613520162/)
java自定义登录_JavaWeb-SpringSecurity自定义登陆页面 (/article/66532474772/)
navicat找mysql的代码_本文为大家分享了使用navicat将csv文件导入mysql的具体代码，供大家参考，具体内容如下1.打开navicat，连接到数据库并找到自己想要导入数据的表... (/article/84662403510/)
Mac 命令行启动并连接Redis (/article/71061053078/)
vs2017解决scanf函数报错的问题 (/article/9791483757/)
MTOP2015双11整体网络拓扑 (/article/914251554/)
Python-S9—Day86-ORM项目实战之会议室预定相关 (/article/7451520829/)
项目进度计划的基本方法 (/article/91931024893/)

推荐文章

Oracle与MySQL的区别2 (/article/5665621117/)
Android学习笔记三之Android基础 (/article/73881575925/)
三维全景地图是怎么实现的？ 三维全景图制作教程 (/article/37142101076/)
个推php ios端教程,个推 -- iOS SDK 1.2.0 集成步骤 (/article/66532565698/)
2018云栖大会深圳峰会-企业级互联网架构专场看点提前大放送！ (/article/9634391345/)
看完泪奔：程序猿苦逼的一生   每日趣闻 (/article/52201655171/)
学校运动会主题的微信公众号图文排版有哪些技巧？ (/article/96501211491/)
询问HTG：白噪声屏幕保护程序，有效的文件命名以及从密码泄露中恢复 (/article/65712017722/)
java swing mysql 物资管理系统 (/article/61612721422/)
孝庄秘史第四集 (/article/4833928161/)

相关标签

分布式链路追踪 (/tag/%E5%88%86%E5%B8%83%E5%BC%8F%E9%93%BE%E8%B7%AF%E8%BF%BD%E8%B8%AA/)
skywalking原理 (/tag/skywalking%E5%8E%9F%E7%90%86/)
zabbix (/tag/zabbix/)
centos (/tag/centos/)