crazyhzm 265

Dubbo源码解析（一）Hello,Dubbo

java dubbo 阅读约 15 分钟

Hello,Dubbo

你好，dubbo，初次见面，我想和你交个朋友。

Dubbo你到底是什么？

先给出一套官方的说法：Apache Dubbo是一款高性能、轻量级基于Java的RPC开源框架。

■ 那么什么是RPC？

> 文档地址：http://dubbo.apache.org/zh-cn...

文档简短形象的对单一应用架构、垂直应用架构、分布式服务架构、流动计算架构做了一个对比，可以很明白的看出这四个架构所适用的场景，因为业务需求越来越复杂，才会有这一系列的演变。

RPC英文全名为Remote Procedure Call，也叫远程过程调用，其实就是一个计算机通信协议，它是一种通过网络从远程计算机程序上请求服务,而不需要了解底层网络技术的协议。计算机通信协议有很多种，对于开发来说，很多熟悉的是HTTP协议，我这里就做个简单的比较，HTTP协议是属于应用层的，而RPC跨越了传输层和应用层。HTTP本身的三次握手协议，每发送一次请求，都会有一次建立连接的过程，就会带来一定的延迟，并且HTTP本身的报文庞大，而RPC可以按需连接，调用结束后就断掉，也可以是长链接，多个远程过程调用共享同一个链接，可以看出来RPC的效率要高于HTTP，但是相对于开发简单快速的HTTP服务,RPC服务就会显得复杂一些。

回到原先的话题，继续来聊聊dubbo。关于dubbo 的特点分别有连通性、健壮性、伸缩性、以及向未来架构的升级性。特点的详细介绍也可以参考上述链接的官方文档。官方文档拥有的内容我在这就不一一进行阐述了。

因为接下来需要对dubbo各个模块的源码以及原理进行解析，所以介绍一下dubbo的源码库，dubbo框架已经交由Apache基金会进行孵化，被挂在github开源。

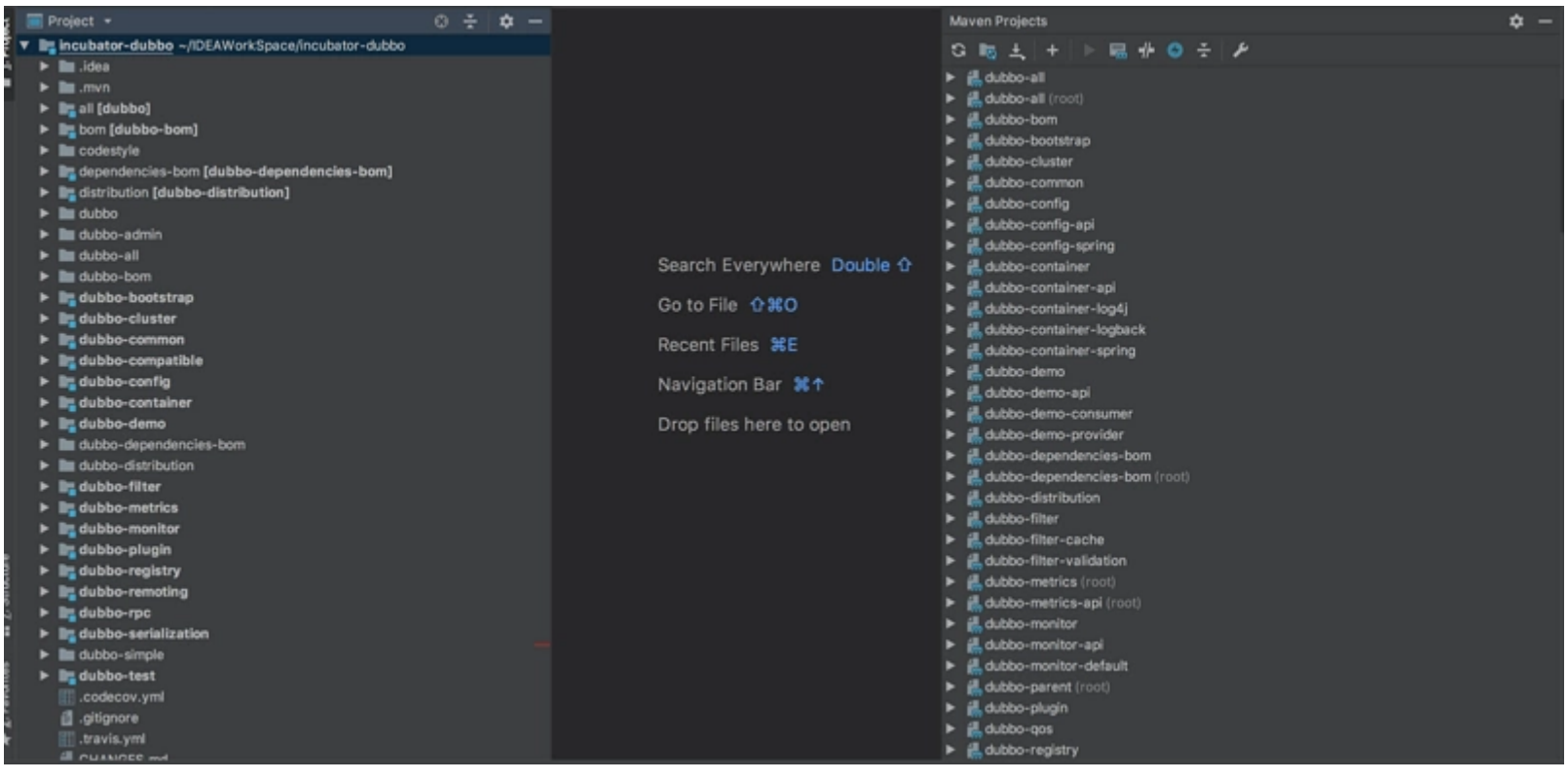
> github地址：https://github.com/apache/inc...

然后讲一下dubbo的版本策略：两个大版本并行发展，2.5.x是稳定版本，2.6.x是新功能实验版本。2.6上实验都稳定了以后，会迁移到2.5，所以如果你想了解dubbo最新的牛逼特性，就选择2.6，否则用2.5版本。我接下来介绍都是基于2.6.x版本。

dubbo框架设计介绍

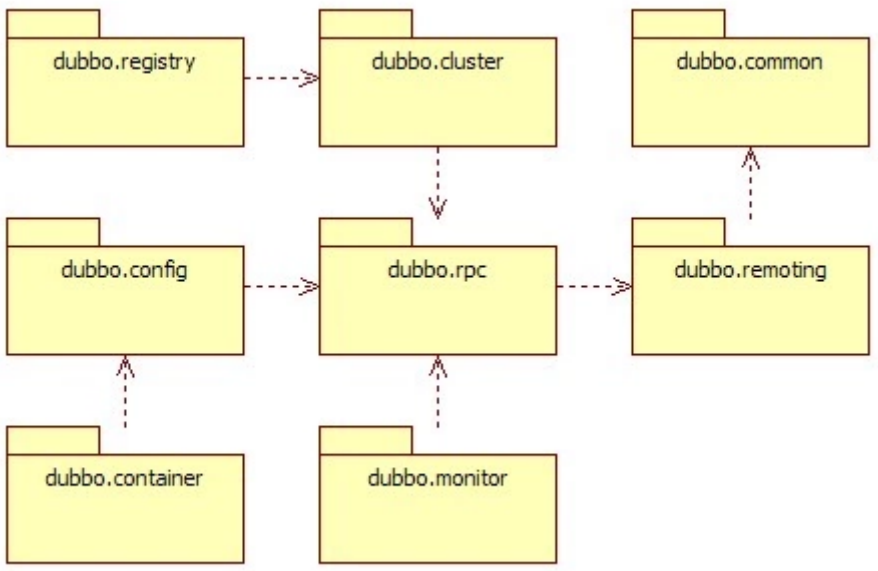
dubbo的官方文档真的写的太好了，我在这里还是要夸赞一下。接下来我对整个框架设计的介绍是基于官方文档的基础上进行扩充，尽量做到更加通俗易懂。

■ 我们先来看看把源码clone下来后的结构：



可以看到Dubbo被拆分成很多的Maven项目（右边的我还没有截全）接下来我会介绍左边每个模块的大致作用。

如果看过dubbo官方文档的朋友肯定看到过以下这个图：



从以上这个图我们可以清晰的看到各个模块之间依赖关系,其实以上的图只是展示了关键的模块依赖关系，还有部分模块比如dubbo-bootstrap清理模块等，下面我会对各个模块做个简单的介绍，至少弄明白各个模块的作用。

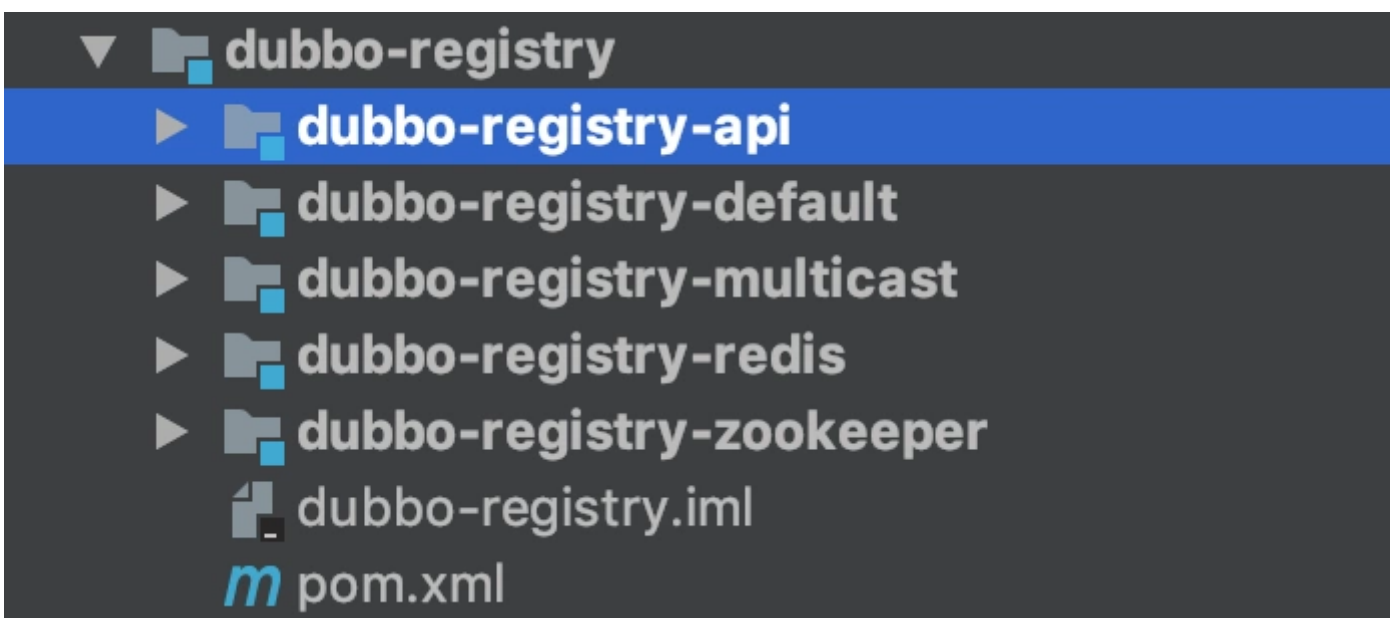
先来讲讲上图的几个模块：

■（一）dubbo-registry——注册中心模块

官方文档的解释：基于注册中心下发地址的集群方式，以及对各种注册中心的抽象。

我的理解是：dubbo的注册中心实现有Multicast注册中心、Zookeeper注册中心、Redis注册中心、Simple注册中心（具体怎么实现我在后面文章中会介绍），这个模块就是封装了dubbo所支持的注册中心的实现。

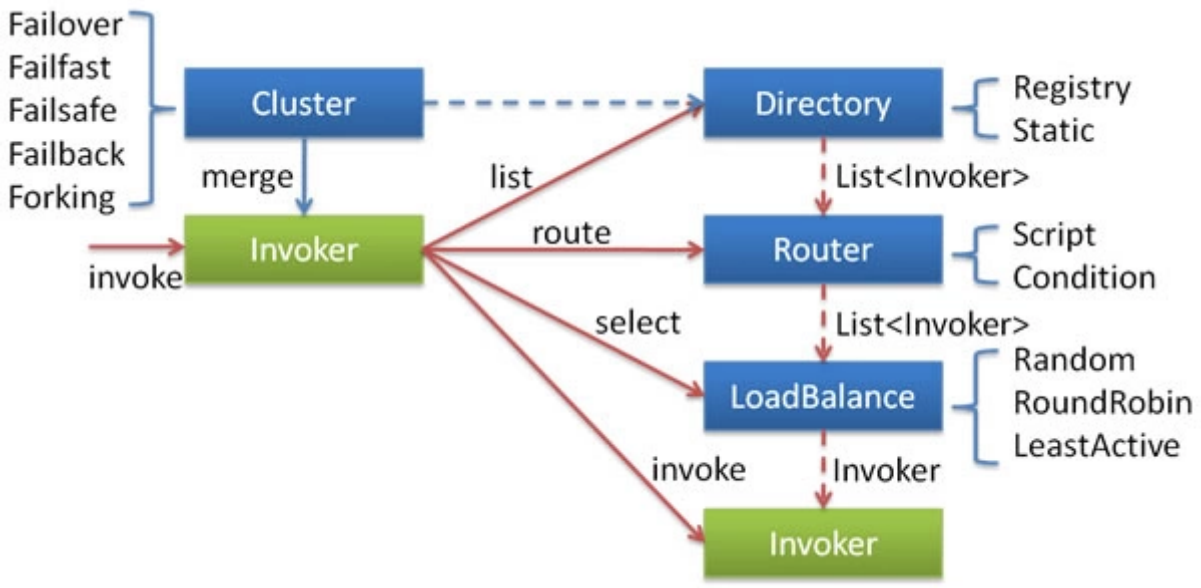
看看registry目录结构：



1. dubbo-registry-api：抽象了注册中心的注册和发现，实现了一些公用的方法，让子类只关注部分关键方法。
2. 以下四个包是分别是四种注册中心实现方法的封装，其中dubbo-registry-default就是官方文档里面的Simple注册中心。

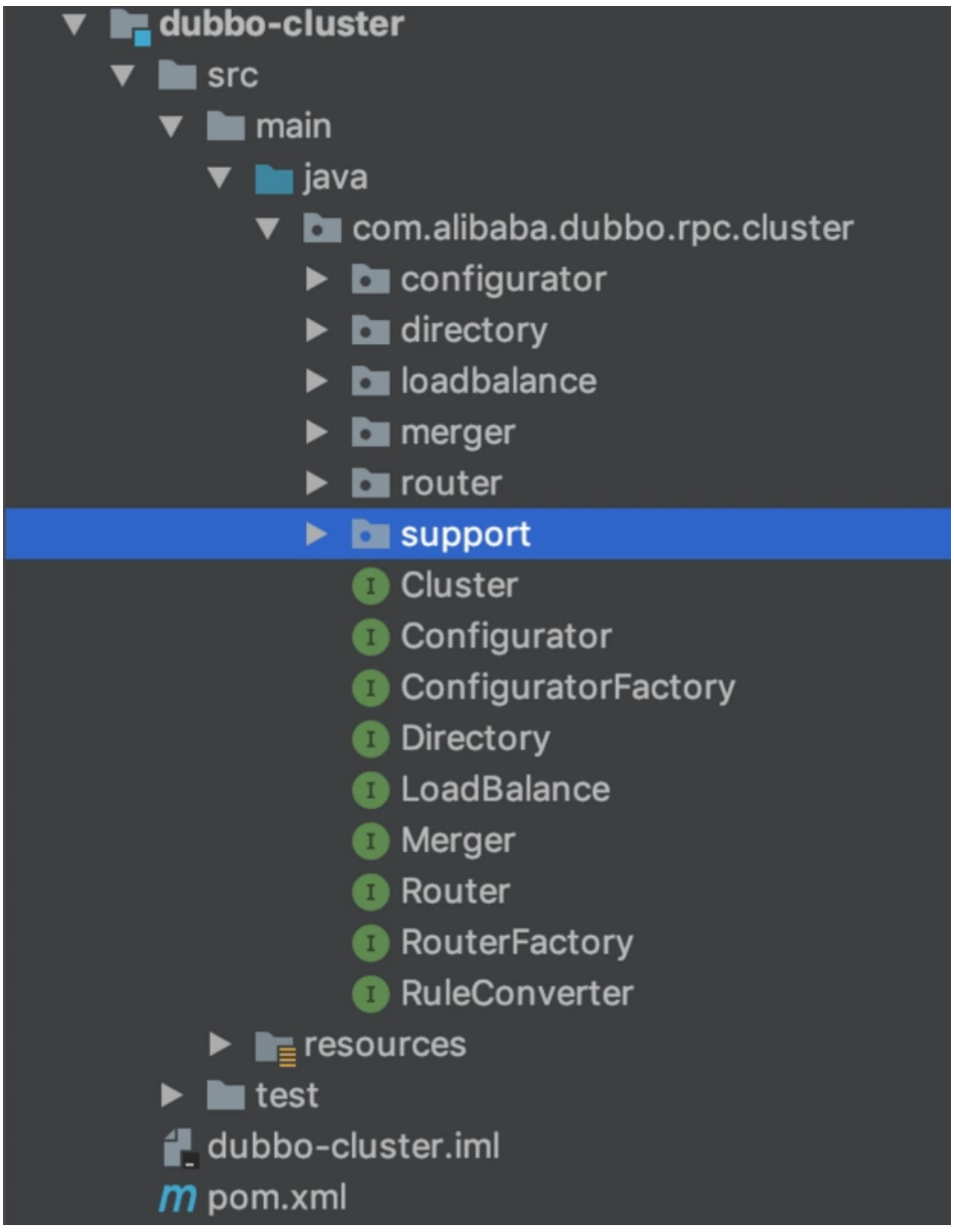
■（二）dubbo-cluster——集群模块

官方文档的解释：将多个服务提供方伪装为一个提供方，包括：负载均衡, 容错，路由等，集群的地址列表可以是静态配置的，也可以是由注册中心下发。



我的理解：它就是一个解决出错情况采用的策略，这个模块里面封装了多种策略的实现方法，并且也支持自己扩展集群容错策略，cluster把多个Invoker伪装成一个Invoker，并且在伪装过程中加入了容错逻辑，失败了，重试下一个。

看看cluster的目录结构：



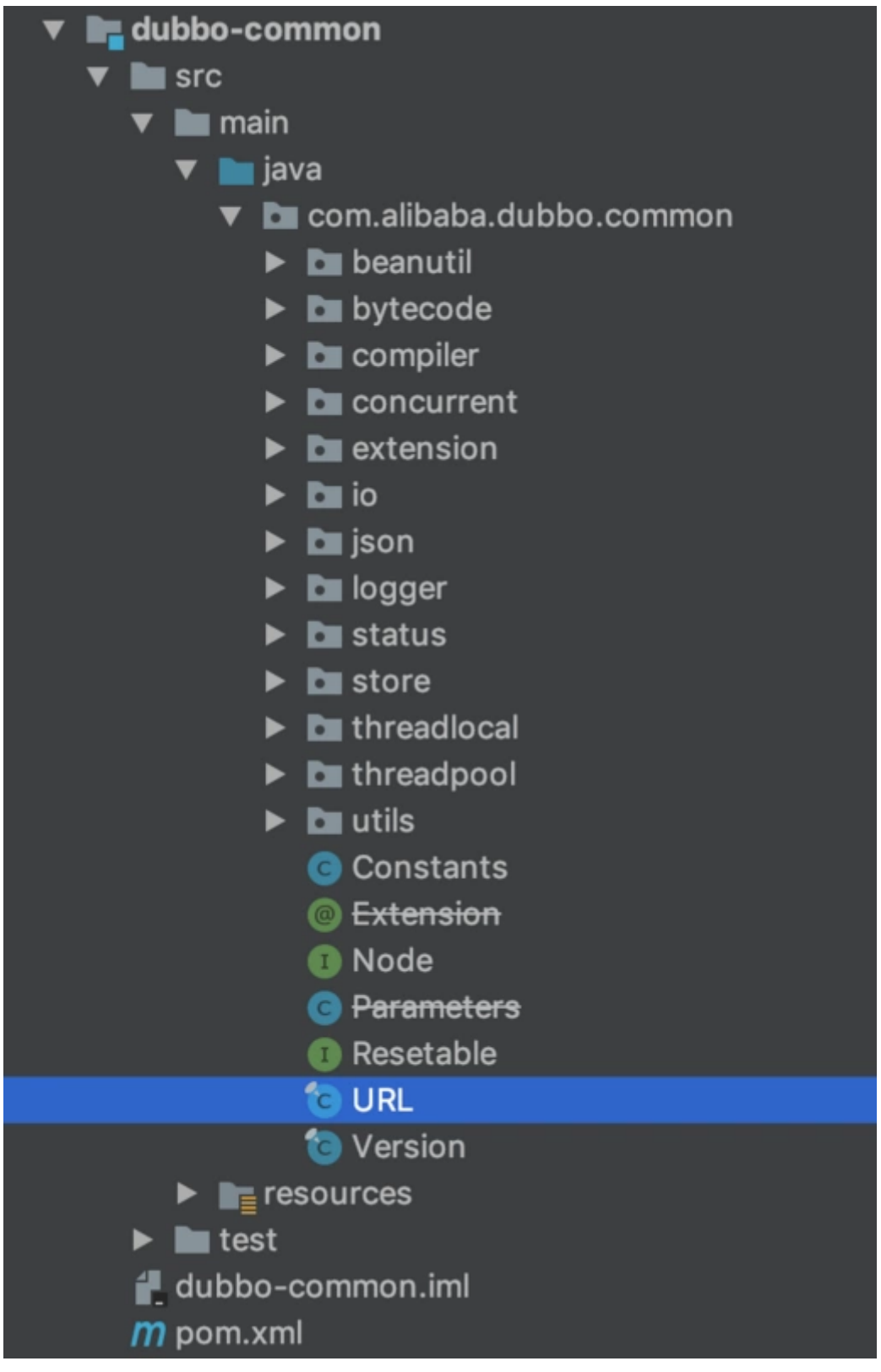
1. configurator包：配置包，dubbo的基本设计原则是采用URL作为配置信息的统一格式，所有拓展点都通过传递URL携带配置信息，这个包就是用来根据统一的配置规则生成配置信息。
2. directory包：Directory 代表了多个 Invoker，并且它的值会随着注册中心的服务变更推送而变化 。这里介绍一下Invoker，Invoker是Provider的一个调用Service的抽象，Invoker封装了Provider地址以及Service接口信息。
3. loadbalance包：封装了负载均衡的实现，负责利用负载均衡算法从多个Invoker中选出具体的一个Invoker用于此次的调用，如果调用失败，则需要重新选择。
4. merger包：封装了合并返回结果，分组聚合到方法，支持多种数据结构类型。
5. router包：封装了路由规则的实现，路由规则决定了一次dubbo服务调用的目标服务器，路由规则分两种：条件路由规则和脚本路由规则，并且支持可拓展。
6. support包：封装了各类Invoker和cluster，包括集群容错模式和分组聚合的cluster以及相关的Invoker。

■（三）dubbo-common——公共逻辑模块

官方文档的解释：包括 Util 类和通用模型。

我的理解：这个应该很通俗易懂，工具类就是一些公用的方法，通用模型就是贯穿整个项目的统一格式的模型，比如URL，上述就提到了URL贯穿了整个项目。

看看common的目录：



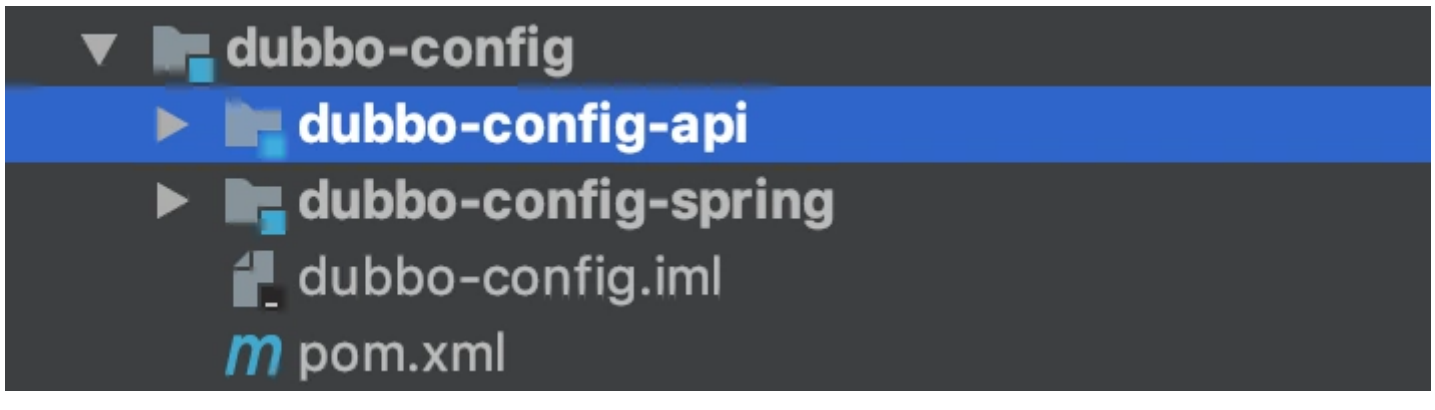
这个类中的包含义我就不一一讲了，具体的介绍会穿插在后续文章中，因为这些都是工具类的一些实现，包的含义也很明显。

■（四）dubbo-config——配置模块

官方文档的解释：是 Dubbo 对外的 API，用户通过 Config 使用Dubbo，隐藏 Dubbo 所有细节。

我的理解：用户都是使用配置来使用dubbo，dubbo也提供了四种配置方式，包括XML配置、属性配置、API配置、注解配置，配置模块就是实现了这四种配置的功能。

看看config的目录：



1. dubbo-config-api：实现了API配置和属性配置的功能。
2. dubbo-config-spring：实现了XML配置和注解配置的功能。

■（五）dubbo-rpc——远程调用模块

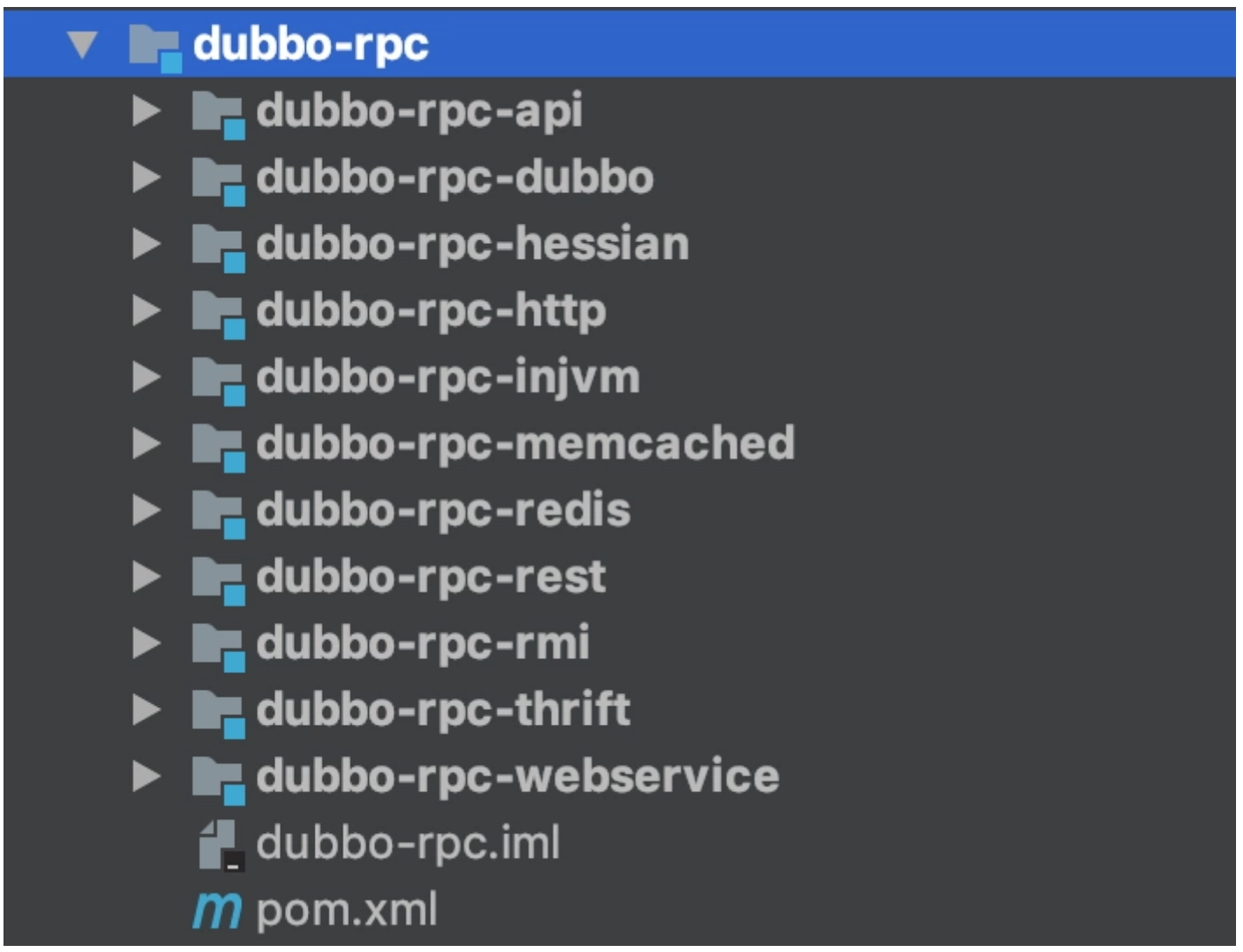
官方文档的解释：抽象各种协议，以及动态代理，只包含一对一的调用，不关心集群的管理。

我的理解：远程调用，最主要的肯定是协议，dubbo提供了许许多多的协议实现，不过官方推荐时使用dubbo自己的协议，还给出了一份性能测试报告。

> 性能测试报告地址：http://dubbo.apache.org/zh-cn...

这个模块依赖于dubbo-remoting模块，抽象了各类的协议。

看看rpc的目录：



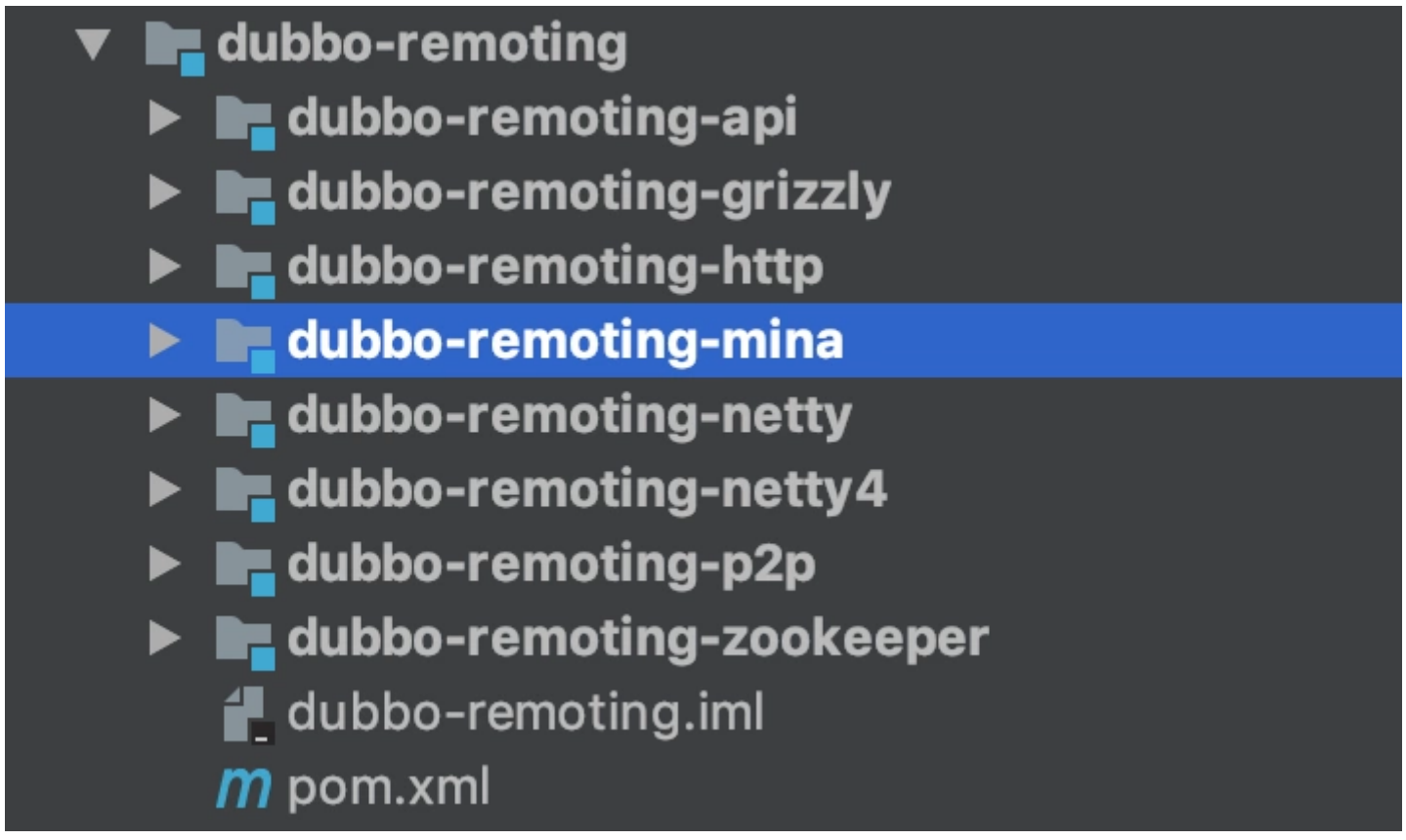
1. dubbo-rpc-api：抽象了动态代理和各类协议，实现一对一的调用
2. 另外的包都是各个协议的实现。

■（六）dubbo-remoting——远程通信模块

官方文档的解释：相当于 Dubbo 协议的实现，如果 RPC 用 RMI协议则不需要使用此包。

我的理解：提供了多种客户端和服务端通信功能，比如基于Grizzly、Netty、Tomcat等等，RPC用除了RMI的协议都要用到此模块。

看看remoting的目录：



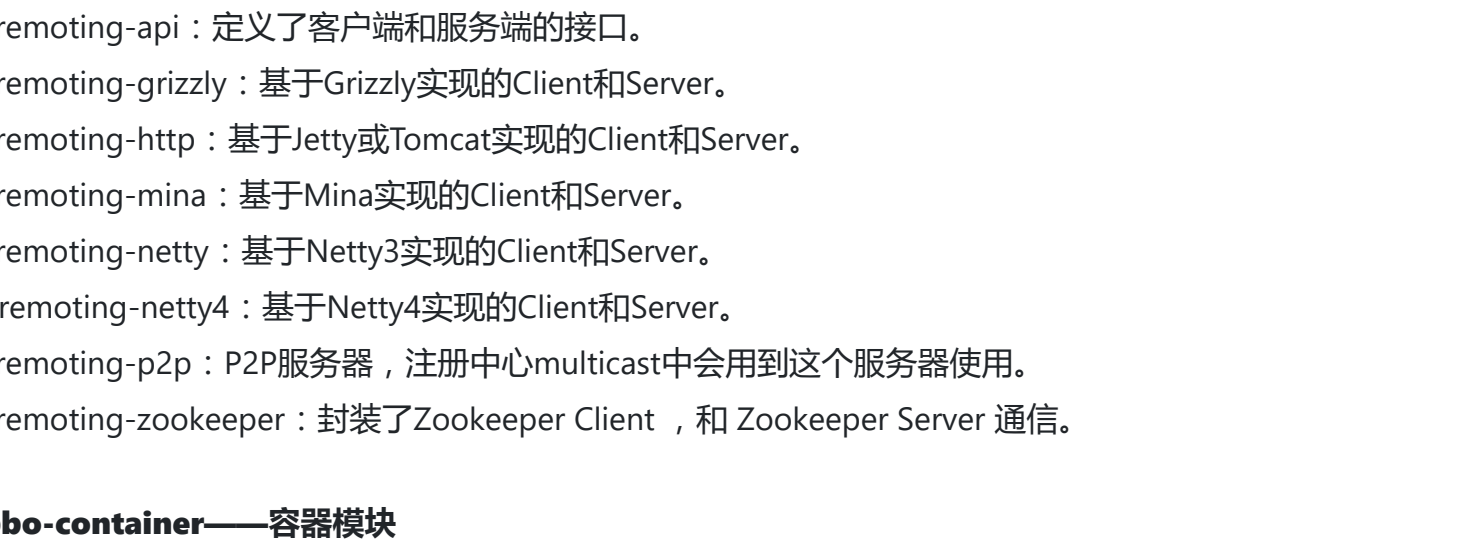
1. dubbo-remoting-api：定义了客户端和服务端的接口。
2. dubbo-remoting-grizzly：基于Grizzly实现的Client和Server。
3. dubbo-remoting-http：基于Jetty或Tomcat实现的Client和Server。
4. dubbo-remoting-mina：基于Mina实现的Client和Server。
5. dubbo-remoting-netty：基于Netty3实现的Client和Server。
6. Dubbo-remoting-netty4：基于Netty4实现的Client和Server。
7. dubbo-remoting-p2p：P2P服务器，注册中心multicast中会用到这个服务器使用。
8. dubbo-remoting-zookeeper：封装了Zookeeper Client ，和 Zookeeper Server 通信。

■（七）dubbo-container——容器模块

官方文档的解释：是一个 Standlone 的容器，以简单的 Main 加载 Spring 启动，因为服务通常不需要 Tomcat/JBoss 等 Web 容器的特性，没必要用 Web 容器去加载服务。

我的理解：因为后台服务不需要Tomcat/JBoss 等 Web 容器的功能，不需要用这些厚实的容器去加载服务提供方，既资源浪费，又增加复杂度。服务容器只是一个简单的Main方法，加载一些内置的容器，也支持扩展容器。

看看container的目录：



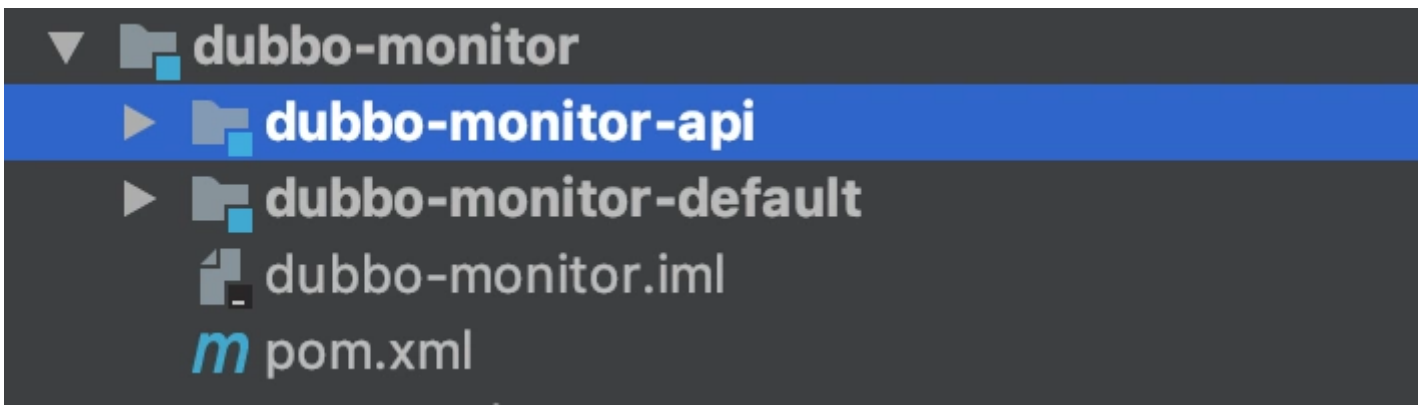
1. dubbo-container-api：定义了Container接口，实现了服务加载的Main方法。
2. 其他三个分别提供了对应的容器，供Main方法加载。

■（八）dubbo-monitor——监控模块

官方文档的解释：统计服务调用次数，调用时间的，调用链跟踪的服务。

我的理解：这个模块很清楚，就是对服务的监控。

看看monitor的目录：



1. dubbo-monitor-api：定义了monitor相关的接口，实现了监控所需要的过滤器。
2. dubbo-monitor-default：实现了dubbo监控相关的功能。

■（九）dubbo-bootstrap——清理模块

这个模块只有一个类，是作为dubbo的引导类，并且在停止期间进行清理资源。具体的介绍我在后续文章中讲解。

■（十）dubbo-demo——示例模块

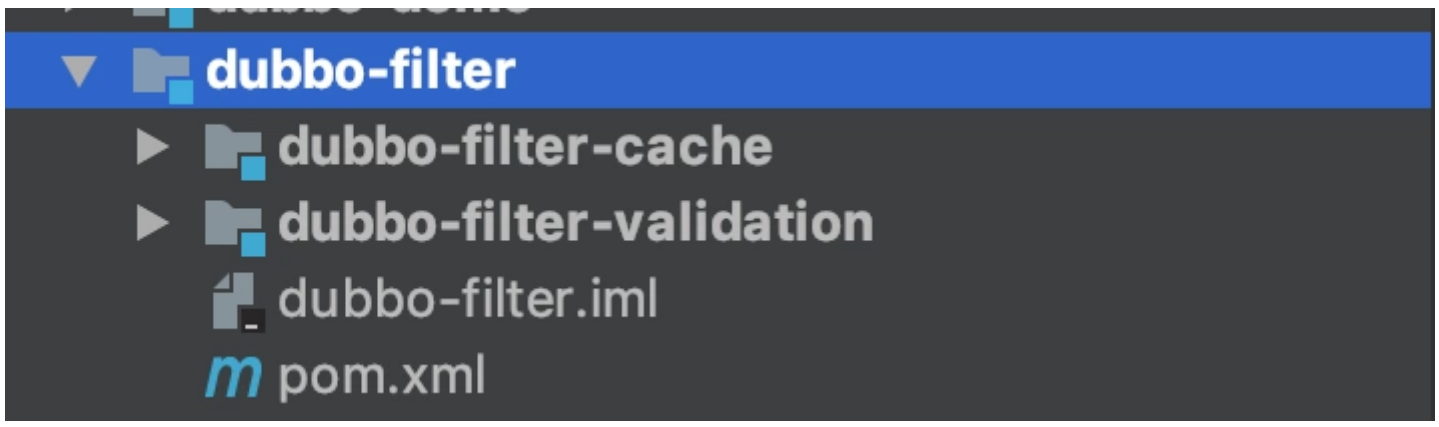
这个模块是快速启动示例，其中包含了服务提供方和调用方，注册中心用的是multicast，用XML配置方法，具体的介绍可以看官方文档。

> 示例介绍地址：http://dubbo.apache.org/zh-cn...

■（十一）dubbo-filter——过滤器模块

这个模块提供了内置的一些过滤器。

看看filter的目录：

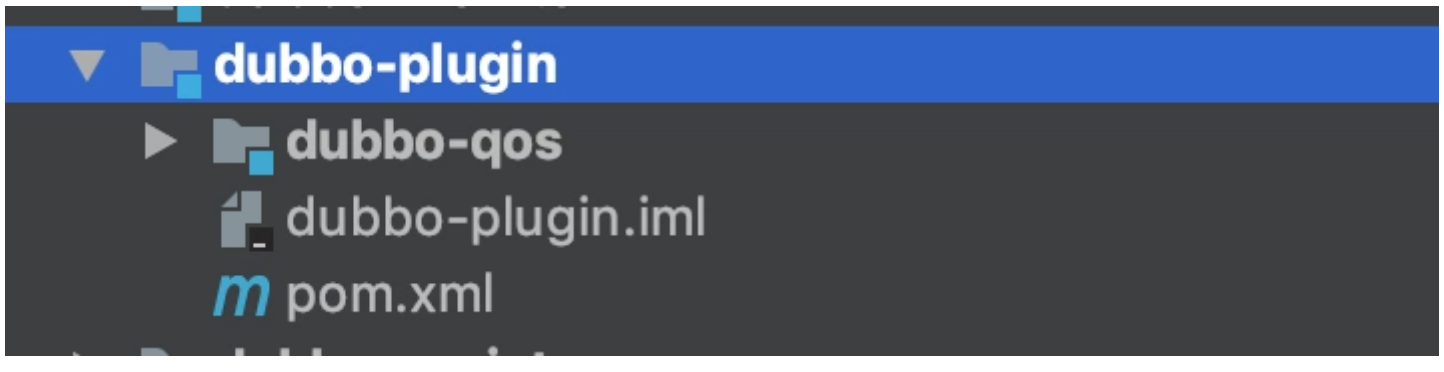


1. dubbo-filter-cache：提供缓存过滤器。
2. dubbo-filter-validation：提供参数验证过滤器。

■（十二）dubbo-plugin——插件模块

该模块提供了内置的插件。

看看plugin的目录：

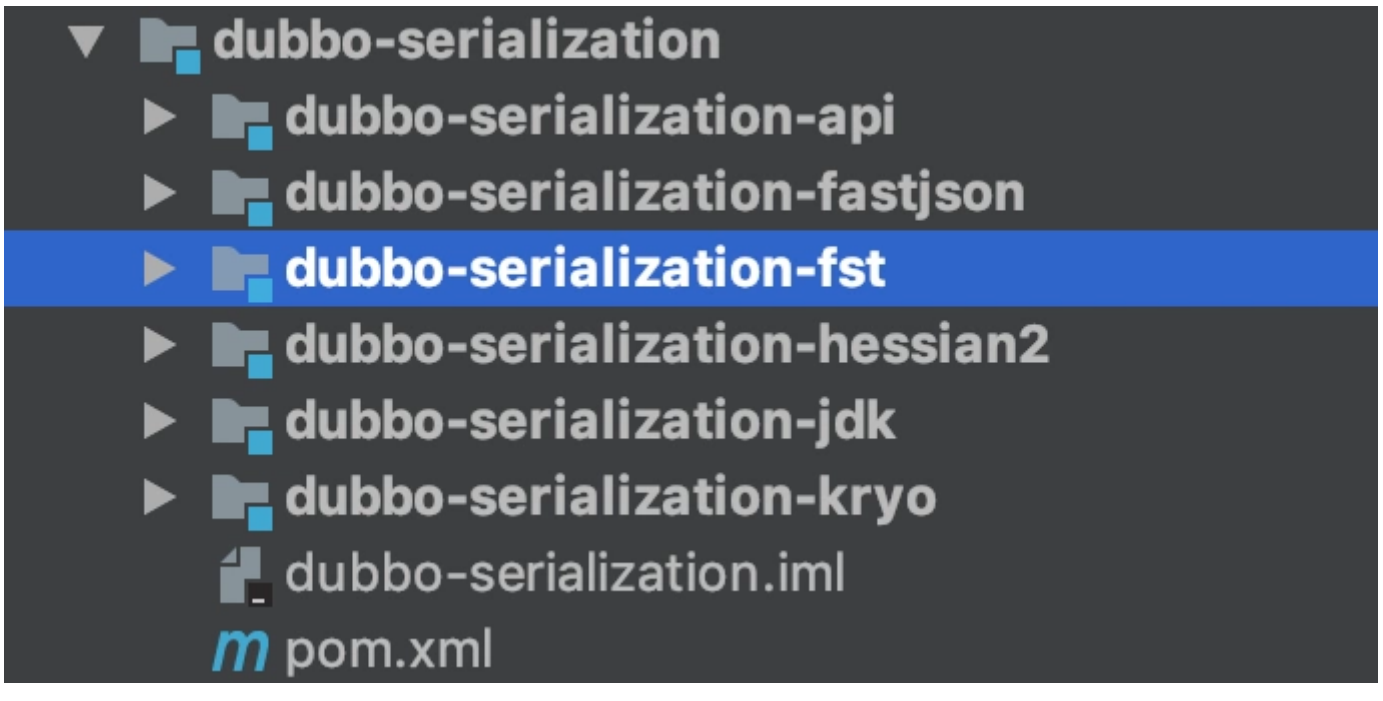


1. dubbo-qos：提供了在线运维的命令。

■（十三）dubbo-serialization——序列化模块

该模块中封装了各类序列化框架的支持实现。

看看serialization的目录：

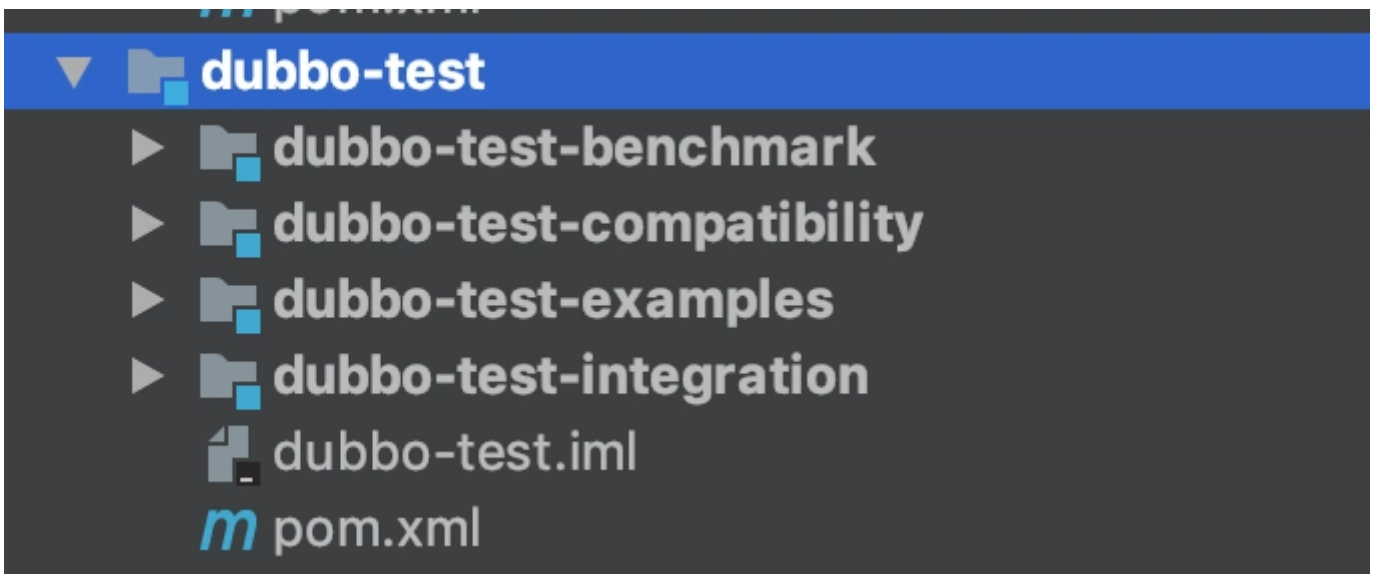


1. dubbo-serialization-api：定义了Serialization的接口以及数据输入输出的接口。
2. 其他的包都是实现了对应的序列化框架的方法。dubbo内置的就是这几类的序列化框架，序列化也支持扩展。

■（十四）dubbo-test——测试模块

这个模块封装了针对dubbo的性能测试、兼容性测试等功能。

看看test的目录：



1. dubbo-test-benchmark：对性能的测试。
2. dubbo-test-compatibility：对兼容性的测试，对spring3对兼容性测试。
3. dubbo-test-examples：测试所使用的示例。
4. dubbo-test-integration：测试所需的pom文件

下面我来讲讲dubbo中Maven相关的pom文件

1. dubbo-bom/pom.xml，利用Maven BOM统一定义了dubbo的版本号。dubbo-test和dubbo-demo的pom文件中都会引用dubbo-bom/pom.xml，以dubbo-demo都pom举例子：

```xml
<description>The demo module of dubbo project</description>
<properties>
    <skip_maven_deploy>true</skip_maven_deploy>
</properties>
<modules>
    <module>dubbo-demo-api</module>
    <module>dubbo-demo-provider</module>
    <module>dubbo-demo-consumer</module>
</modules>

<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>com.alibaba</groupId>
            <artifactId>dubbo-bom</artifactId>
            <version>${project.parent.version}</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>

</project>
```

2. dubbo-dependencies-bom/pom.xml：利用Maven BOM统一定义了dubbo依赖的第三方库的版本号。dubbo-parent会引入该bom：

```xml
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>com.alibaba</groupId>
            <artifactId>dubbo-dependencies-bom</artifactId>
            <version>2.6.5-SNAPSHOT</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
```

3. all/pow.xml：定义了dubbo的打包脚本，使用dubbo库的时候，需要引入改pom文件。
4. dubbo-parent：是dubbo的父pom，dubbo的maven模块都会引入该pom文件。

后记

读完整篇文章后不知道对你踏入dubbo源码解读是否有几分帮助，后续我将会逐步更新dubbo每个模块的解读，如果我在哪一部分写的不够到位或者写错了，欢迎给我提意见，我的私人微信号码：HUA799695226。