



链滴

SpringBoot2.1.9+dubbo2.7.3+Nacos1.1. 4 构建你的微服务体系

作者: [jianzh5](#)

原文链接: <https://ld246.com/article/1574660752999>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



简单几步使用最新版本的DUBBO构建你的微服务体系

NACOS注册中心

- 从 [github](#)下载最新版本的nacos
- 上传至服务器并解压
- 单机启动 `sh startup.sh -m standalone`
- nacos 控制台访问地址 <http://192.168.136.129:8848/nacos>, 使用账号nacos/nacos登录并访问

A screenshot of the Nacos 1.1.4 configuration management interface. The top navigation bar includes links for Dashboard, Document, Examples, Community, and English. The left sidebar has sections for Configuration Management (包括配罝列表, 配罝版本, 配罝差異), Service Management (包括服务列表, 服务消费者, 服务空间), and Cluster Management (包括节点列表). The main content area is titled '配置管理 | public' and shows a table for managing configuration items. The table columns include Data ID, Group, 值和数据, and 操作. There is a search bar at the top of the table.

项目框架

本次案例包含三个组件

- 公共接口层 dubbo-api
- 生产者 dubbo-provider
- 消费者 dubbo-consumer

代码目录如下：



父目录

父目录主要是定义公共组件依赖，版本号，**pom**文件如下

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <packaging>pom</packaging>

    <modules>
        <module>dubbo-api</module>
        <module>dubbo-provider</module>
        <module>dubbo-consumer</module>
    </modules>

    <groupId>com.jianzh5</groupId>
    <artifactId>dubbo-demo</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <java.version>1.8</java.version>
        <netty-all.version>4.0.35.Final</netty-all.version>
        <spring-boot.version>2.1.9.RELEASE</spring-boot.version>
        <dubbo.version>2.7.3</dubbo.version>
        <nacos-client.version>1.1.4</nacos-client.version>
    </properties>

    <dependencyManagement>
        <dependencies>
            <!--SpringBoot-->
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-dependencies</artifactId>
                <version>${spring-boot.version}</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>
```

```
<!-- Apache Dubbo -->
<dependency>
    <groupId>org.apache.dubbo</groupId>
    <artifactId>dubbo-dependencies-bom</artifactId>
    <version>${dubbo.version}</version>
    <type>pom</type>
    <scope>import</scope>
</dependency>

<!-- Dubbo Registry Nacos -->
<dependency>
    <groupId>org.apache.dubbo</groupId>
    <artifactId>dubbo-registry-nacos</artifactId>
    <version>${dubbo.version}</version>
</dependency>

<dependency>
    <groupId>com.alibaba.nacos</groupId>
    <artifactId>nacos-client</artifactId>
    <version>${nacos-client.version}</version>
</dependency>

<!-- Dubbo Spring Boot Starter -->
<dependency>
    <groupId>org.apache.dubbo</groupId>
    <artifactId>dubbo-spring-boot-starter</artifactId>
    <version>${dubbo.version}</version>
</dependency>

<dependency>
    <groupId>org.apache.dubbo</groupId>
    <artifactId>dubbo</artifactId>
    <version>${dubbo.version}</version>
    <exclusions>
        <exclusion>
            <groupId>org.springframework</groupId>
            <artifactId>spring</artifactId>
        </exclusion>
        <exclusion>
            <groupId>javax.servlet</groupId>
            <artifactId>servlet-api</artifactId>
        </exclusion>
        <exclusion>
            <groupId>log4j</groupId>
            <artifactId>log4j</artifactId>
        </exclusion>
    </exclusions>
</dependency>

<!-- 接口-->
<dependency>
    <groupId>com.jianzh5</groupId>
    <artifactId>dubbo-api</artifactId>
```

```

        <version>1.0-SNAPSHOT</version>
    </dependency>

</dependencies>
</dependencyManagement>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <version>${spring-boot.version}</version>
            <executions>
                <execution>
                    <goals>
                        <goal>repackage</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>

</project>

```

在pom文件中直接依赖接口，这样生产者和消费者都可以使用了

接口层dubbo-api定义

- pom

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd
maven-4.0.0.xsd">
    <parent>
        <artifactId>dubbo-demo</artifactId>
        <groupId>com.jianzh5</groupId>
        <version>1.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.jianzh5</groupId>
    <artifactId>dubbo-api</artifactId>
    <packaging>jar</packaging>
</project>

```

- 定义真实的接口

```

/**
 * @author jianzh5
 * @date 2019/11/5 10:45
 */
public interface HelloService {

```

```
    String sayHello();
}
```

生产者dubbo-provider

- pom

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd
maven-4.0.0.xsd">
  <parent>
    <artifactId>dubbo-demo</artifactId>
    <groupId>com.jianzh5</groupId>
    <version>1.0-SNAPSHOT</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.jianzh5</groupId>
  <artifactId>dubbo-provider</artifactId>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter</artifactId>
    </dependency>

    <!-- Dubbo -->
    <dependency>
      <groupId>org.apache.dubbo</groupId>
      <artifactId>dubbo-spring-boot-starter</artifactId>
    </dependency>
    <dependency>
      <groupId>org.apache.dubbo</groupId>
      <artifactId>dubbo</artifactId>
    </dependency>

    <!-- Dubbo Registry Nacos -->
    <dependency>
      <groupId>org.apache.dubbo</groupId>
      <artifactId>dubbo-registry-nacos</artifactId>
    </dependency>

    <dependency>
      <groupId>com.alibaba.nacos</groupId>
      <artifactId>nacos-client</artifactId>
    </dependency>

    <dependency>
      <groupId>com.jianzh5</groupId>
      <artifactId>dubbo-api</artifactId>
    </dependency>
  </dependencies>

```

```
</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
</project>
```

- 配置文件 `application.xml`

```
dubbo.application.name = dubbo-provider
dubbo.registry.address = nacos://192.168.136.129:8848
dubbo.scan.base-packages=com.jianzh5.provider.service.impl
dubbo.protocol.port=20881
dubbo.protocol.name=dubbo
```

- 接口实现

```
@Service
public class HelloServiceImpl implements HelloService {

  public String sayHello() {
    return "欢迎关注微信公众号：JAVA日知录";
  }

}
```

注意这里的`@service`引用的是`org.apache.dubbo.config.annotation.Service`,不要引用错了

- 启动类

```
@SpringBootApplication
@EnableDubbo
public class ProviderBootstrap {
  public static void main(String[] args) {
    SpringApplication.run(ProviderBootstrap.class, args);
  }
}
```

需要引入`@EnableDubbo`注解

消费者dubbo-consumer

- pom

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd
maven-4.0.0.xsd">
  <parent>
```

```
<artifactId>dubbo-demo</artifactId>
<groupId>com.jianzh5</groupId>
<version>1.0-SNAPSHOT</version>
</parent>
<modelVersion>4.0.0</modelVersion>

<groupId>com.jianzh5</groupId>
<artifactId>dubbo-consumer</artifactId>

<dependencies>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!-- Dubbo -->
    <dependency>
        <groupId>org.apache.dubbo</groupId>
        <artifactId>dubbo-spring-boot-starter</artifactId>
    </dependency>
    <dependency>
        <groupId>org.apache.dubbo</groupId>
        <artifactId>dubbo</artifactId>
    </dependency>

    <!-- Dubbo Registry Nacos -->
    <dependency>
        <groupId>org.apache.dubbo</groupId>
        <artifactId>dubbo-registry-nacos</artifactId>
    </dependency>

    <dependency>
        <groupId>com.alibaba.nacos</groupId>
        <artifactId>nacos-client</artifactId>
    </dependency>

    <dependency>
        <groupId>com.jianzh5</groupId>
        <artifactId>dubbo-api</artifactId>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>

</project>
```

- 配置文件application.properties

```
spring.application.name=dubbo-consumer
dubbo.registry.address = nacos://192.168.136.129:8848
server.port=9090
```

- 消费者 HelloController

```
/**
 * @author jianzh5
 * @date 2019/11/5 11:29
 */
@RestController
public class HelloController {

    @Reference
    private HelloService helloService;

    @GetMapping("/sayHello")
    public String sayHello(){
        return helloService.sayHello();
    }

}
```

使用@Reference注解注入HelloService

- 接口启动类

```
@SpringBootApplication
public class ConsumerBootstrap {

    public static void main(String[] args) {
        SpringApplication.run(ConsumerBootstrap.class, args);
    }

}
```

项目启动

- 启动生产者 dubbo-provider
- 启动消费者 dubbo-consumer
- 查看nacos控制台，观察是否注册

NACOS 1.1.4

public

服务列表 | public

服务名称: consumer:com.jianshi5.ap.HelloService: 分组名称: provider:com.jianshi5.ap.HelloService: 检查间隔: 10s 搜索

| 服务名 | 分组名称 | 集群数 | 实例数 | 健康状态数 | 耗时保护阈值 | 操作 |
|--|---------------|-----|-----|-------|--------|--|
| consumer:com.jianshi5.ap.HelloService: | DEFAULT_GROUP | 1 | 1 | 1 | true | 详情 示例代码 删除 |
| provider:com.jianshi5.ap.HelloService: | DEFAULT_GROUP | 1 | 1 | 1 | true | 详情 示例代码 删除 |

- 访问浏览器 <http://localhost:9090/sayHello> 查看接口返回

← → ⌂ ⓘ localhost:9090/sayHello

欢迎关注微信公众号：JAVA日知录