

- 1、Dubbo简介
- 2、user模块开发Dubbo服务案例
- 3、功能拓展
  - 搭建Dubbo-Admin监控服务
  - 了解Dubbo的高级特性和用法

### 三、Dubbo基础服务流程

阶段目标：打通用户模块的Dubbo服务远程调用流程。

## 1、Dubbo简介

---

参见官网：<https://cn.dubbo.apache.org/zh-cn/overview/what/overview/>

Dubbo不是微服务架构的竞争者，而是微服务架构中一个重要的环节。

## 2、user模块开发Dubbo服务案例

---

tl-live-user-provider的核心pom.xml依赖：(tl-live-api模块依赖也一样)

```

<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>
    <parent>
        <groupId>com.tl</groupId>
        <artifactId>tl-live-user</artifactId>
        <version>1.1</version>
    </parent>

    <artifactId>tl-live-user-provider</artifactId>
    <packaging>jar</packaging>

    <name>tl-live-user-provider</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.apache.dubbo</groupId>
            <artifactId>dubbo-spring-boot-starter</artifactId>
            <version>${dubbo.version}</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-bootstrap</artifactId>
            <version>${spring-cloud-bootstrap.version}</version>
        </dependency>
        <dependency>
            <groupId>com.alibaba.cloud</groupId>
            <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
        </dependency>
        <dependency>
            <groupId>com.tl.user</groupId>
            <artifactId>tl-live-user-interface</artifactId>
            <version>${project.version}</version>
        </dependency>
    </dependencies>

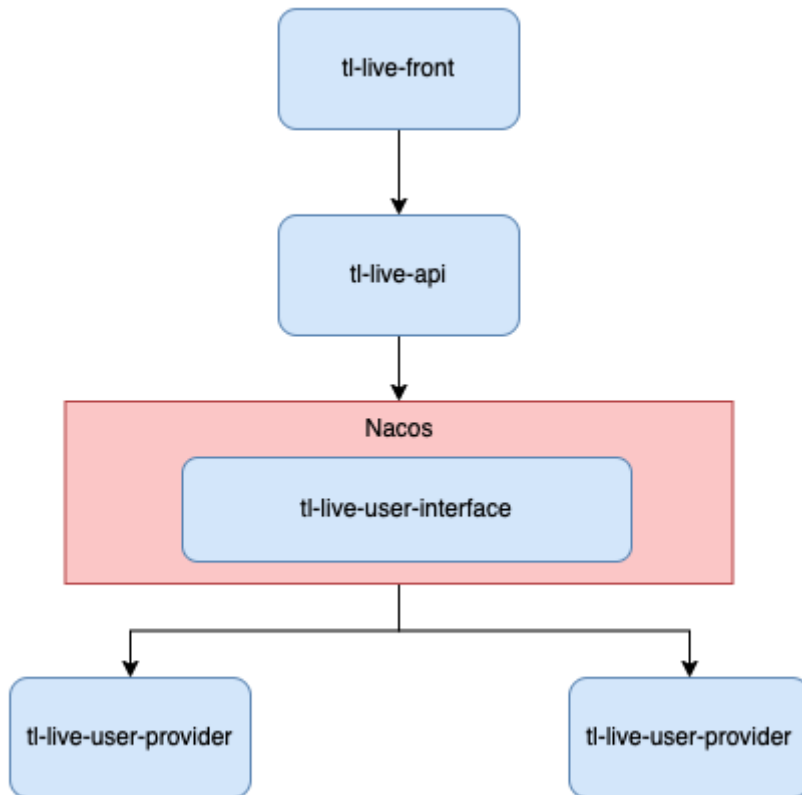
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
                <version>3.0.4</version>
                <configuration>
                    <classifier>exec</classifier>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>

```

注意点：1、Dubbo3.2.0支持的SpringBoot版本需要是3.0.X

2、使用当前版本的dubbo-spring-boot-starter组件时，需要加入spring-cloud-starter-bootstrap依赖支持，否则会报错。这是一个关于解析配置文件的扩展模块。

核心业务实现流程：



### 3、功能拓展

#### 搭建Dubbo-Admin监控服务

```
# 搜索镜像
docker search dubbo-admin
# 拉取镜像
docker pull apache/dubbo-admin
# 查看镜像结构
docker inspect 843213c7e1c8
# 获取一些重要信息
# config.ExposedPorts 暴露的端口。这个端口就可以映射到宿主机
# config.Env 镜像的环境信息。这些信息可以在创建镜像时，通过-e 指定。
# 启动镜像 需要提前启动nacos，否则无法连接注册中心，镜像会停止。
docker run -d --name dubbo-admin \
  -e admin.registry.address=nacos://192.168.65.210:8848?namespace=tl-live \
  -e admin.config-center=nacos://192.168.65.210:8848?namespace=tl-live \
  -e admin.metadata-report.address=nacos://192.168.65.210:8848?namespace=tl-live \
  -e admin.root.user.name=root \
  -e admin.root.user.password=root \
  -p 7080:8080 \
  apache/dubbo-admin
# 查看镜像进程
docker ps
# 检查镜像日志
docker logs 843213c7e1c8
```

如果启动正常，就可以访问 <http://192.168.65.210:7080>访问dubbo admin的管理页面。用户名和密码都是指定的root。

## 了解Dubbo的高级特性和用法

参见官网 <https://cn.dubbo.apache.org/zh-cn/overview/manual/java-sdk/advanced-features-and-usage/service/>

重点理解：服务分版本、服务分组、集群容错