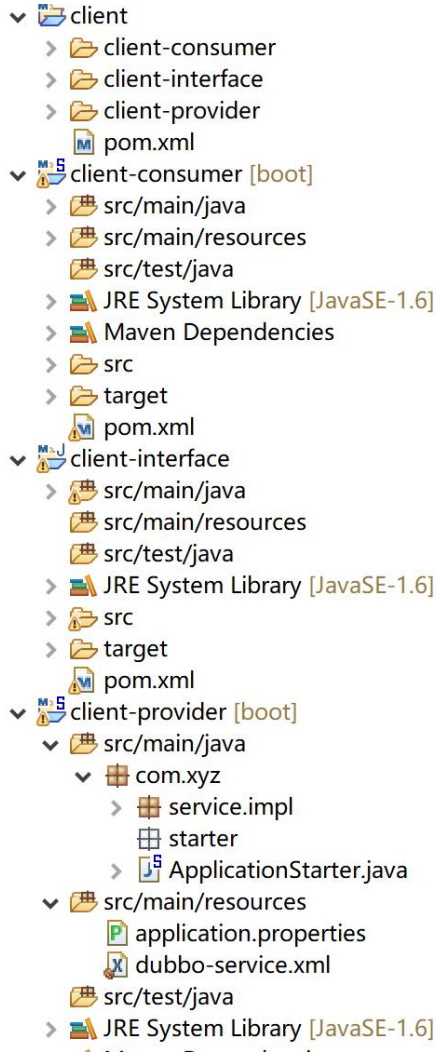
先展示一下项目



client作为父级别项目，下面是三个模块，如名字，一个接口项目，一个服务提供者项目，一个消费者项目

client的pom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>

	<!-- 从Spring Boot继承默认配置 -->
	<parent>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-starter-parent</artifactId>
		<version>1.5.10.RELEASE</version>
	</parent>
```

```
    <groupId>com.xyz</groupId>
    <artifactId>client</artifactId>
    <version>1.0.0</version>
    <packaging>pom</packaging>
    <name>client</name>
    <modules>
        <module>client-interface</module>
        <module>client-provider</module>
        <module>client-consumer</module>
    </modules>
</project>
```

client-interface是一个普通的项目，client-provider和client-consumer都是boot项目

因为只是集成简单集成dubbo，两个boot项目的pom依赖是一样的，这里以client-provider的pom为例

```
<?xml version="1.0"?>
<project
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd"
    xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>com.xyz</groupId>
        <artifactId>client</artifactId>
        <version>1.0.0</version>
    </parent>
    <groupId>com.xyz</groupId>
    <artifactId>client-provider</artifactId>
    <version>1.0.0</version>
    <name>client-provider</name>
    <url>http://maven.apache.org</url>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <!-- Spring boot启动支持 -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <!-- Spring boot 集成dubbo -->
        <dependency>
            <groupId>com.alibaba.spring.boot</groupId>
            <artifactId>dubbo-spring-boot-starter</artifactId>
```

```
            <version>1.0.0</version>
        </dependency>

        <!-- zookeeper 客户端jar -->
        <dependency>
            <groupId>com.101tec</groupId>
            <artifactId>zkclient</artifactId>
            <version>0.10</version>
        </dependency>

        <!-- client-interface -->
        <dependency>
            <groupId>com.xyz</groupId>
            <artifactId>client-interface</artifactId>
            <version>1.0.0</version>
        </dependency>

    </dependencies>

    <build>
        <finalName>springBootDemo</finalName>
        <!-- 打包成可执行jar文件 -->
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

因为注解集成dubbo的方式没成功而且使用dubbo的注解会增加项目和dubbo的耦合性，不利于项目后期的变更，所以这里采用xml的方式配置以及注册。

client-provider中增加dubbo-service.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://code.alibabatech.com/schema/dubbo
http://code.alibabatech.com/schema/dubbo/dubbo.xsd">
    <!-- 配置可参考 http://dubbo.io/User+Guide-zh.htm -->
    <!-- 服务提供方应用名，用于计算依赖关系 -->
    <dubbo:application name="client-provider" owner="client-provider" />
```

```
    <!-- 定义 zookeeper 注册中心地址及协议 -->
    <dubbo:registry protocol="zookeeper" address="127.0.0.1:2181"
        client="zkclient" />
    <!-- 定义 Dubbo 协议名称及使用的端口，dubbo 协议缺省端口为 20880，如果配置为 -1 或者没有配置 port，则会分配一个没有被占用的端口 -->
    <dubbo:protocol name="dubbo" port="-1" />
    <!-- 声明需要暴露的服务接口 -->
    <dubbo:service interface="com.xyz.service.UserService"
        ref="userService" timeout="10000" />
    <!-- 和本地 bean 一样实现服务 -->
    <bean id="userService" class="com.xyz.service.impl.UserServiceImpl" />
</beans>
```

并在Spring boot启动类添加如下注解

```
@ImportResource(value = {"classpath:dubbo-service.xml"})
```

client-consumer中增加dubbo.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://code.alibabatech.com/schema/dubbo
http://code.alibabatech.com/schema/dubbo/dubbo.xsd">
    <!-- 配置可参考 http://dubbo.io/User+Guide-zh.htm -->
    <!-- 服务提供方应用名，用于计算依赖关系 -->
    <dubbo:application name="client-consumer" owner="client-consumer" />
    <!-- 定义 zookeeper 注册中心地址及协议 -->
    <dubbo:registry protocol="zookeeper" address="127.0.0.1:2181"
        client="zkclient" />


    <dubbo:reference id="userService" interface="com.xyz.service.UserService" />
</beans>
```

并在Spring boot启动类添加如下注解

```
@ImportResource(value = {"classpath:dubbo.xml"})
```

这样就可以在client-consumer调用client-provider中的服务了。