实验1：RPC服务开发与调用

作者： BY2221105 王靖皓

0、运行环境

- OS：64bit windows10
- 编辑器： Idea 2022.1.3
- Java : 1.8.0_331

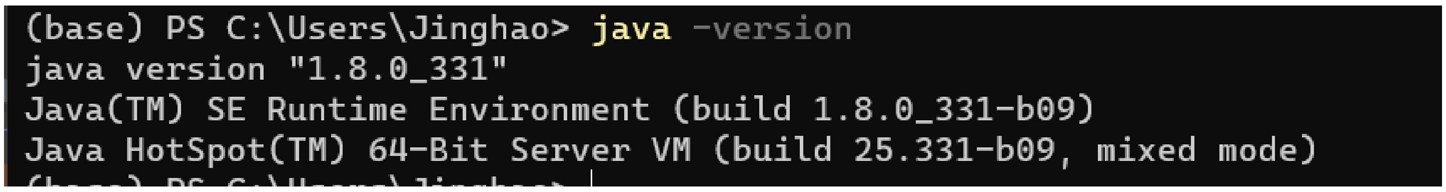
1、开发环境配置

安装 JDK 环境

下载安装 JDK8 环境，下载地址：https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html

在下载页面中，找到对应的平台进行下载，并完成安装

安装之后在终端检查 java 版本：



```
(base) PS C:\Users\Jinghao> java -version
java version "1.8.0_331"
Java(TM) SE Runtime Environment (build 1.8.0_331-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.331-b09, mixed mode)
(base) PS C:\Users\Jinghao>
```

安装 ZooKeeper

由于 dubbo 需要在 zookeeper 中完成注册，因此需要搭建 zookeeper

zookeeper 下载地址：https://zookeeper.apache.org/ 下载 ZooKeeper

Windows10安装Zookeeper教程_JMzz的博客-CSDN博客_win10安装zookeeper

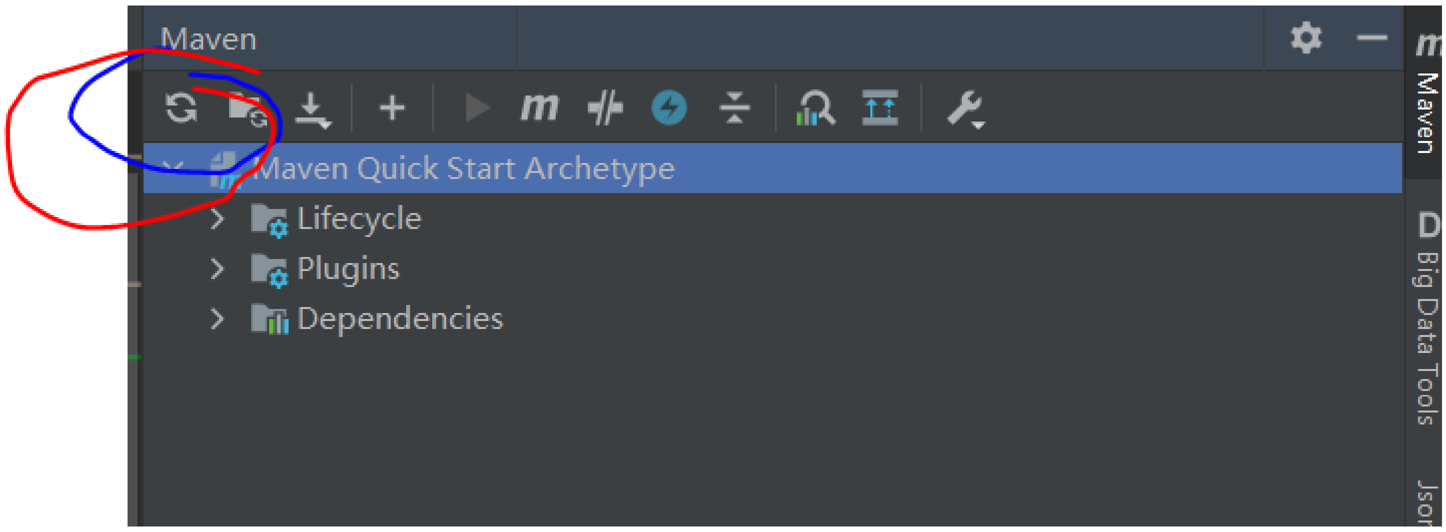
创建 Dubbo 工程项目

前往 dubbo 官方 github 仓库：https://github.com/apache/dubbo ReadMe 中的 Getting started 说明，引入 dubbo 依赖：

在项目根目录下的 pom.xml 文件中加入以下依赖项：

```
<properties>
    <dubbo.version>3.0.7</dubbo.version>
</properties>

<dependencies>
    <dependency>
        <groupId>org.apache.dubbo</groupId>
        <artifactId>dubbo</artifactId>
        <version>${dubbo.version}</version>
    </dependency>
    <dependency>
        <groupId>org.apache.dubbo</groupId>
        <artifactId>dubbo-dependencies-zookeeper</artifactId>
        <version>${dubbo.version}</version>
        <type>pom</type>
    </dependency>
</dependencies>
```

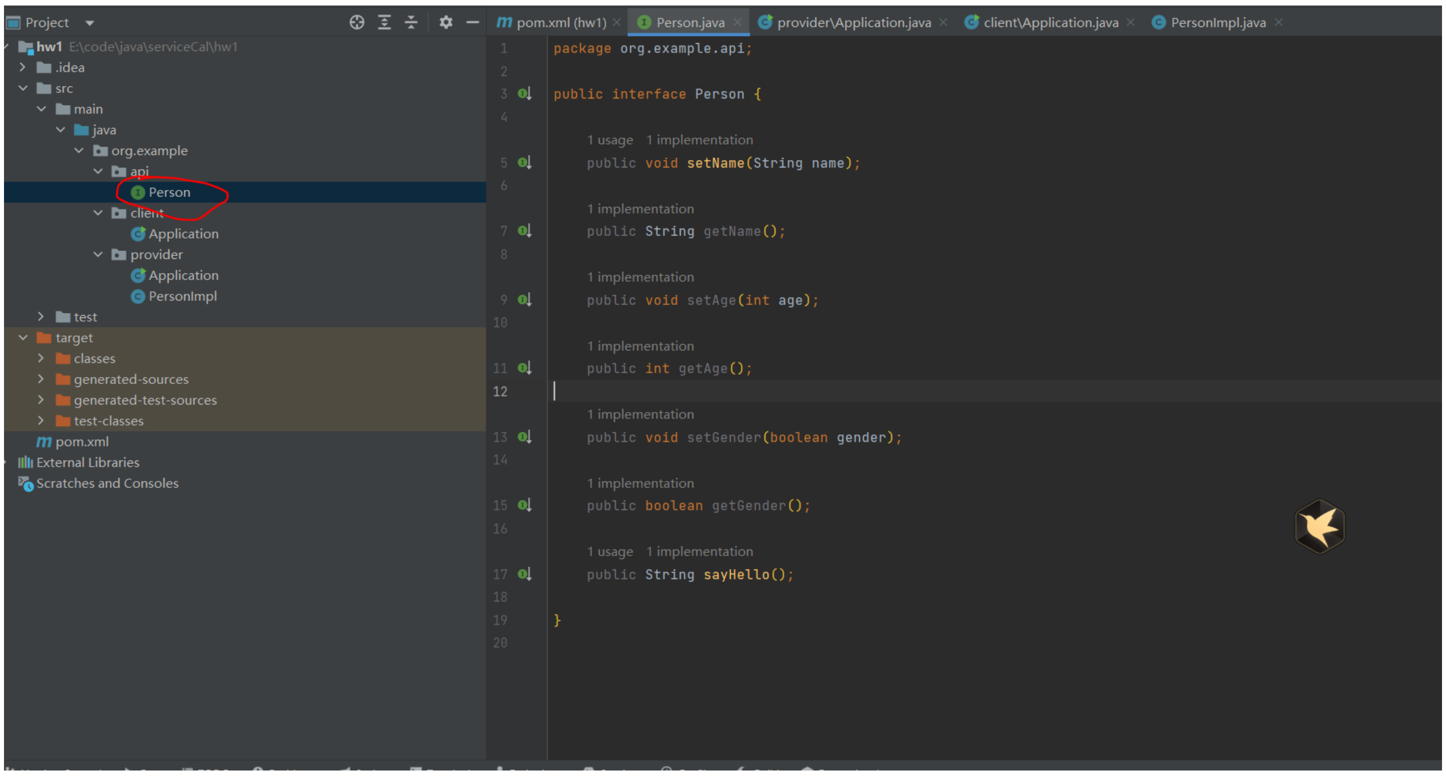
下载maven配置



2. 服务接口定义、实现

创建文件 Person.java 定义 Person 接口

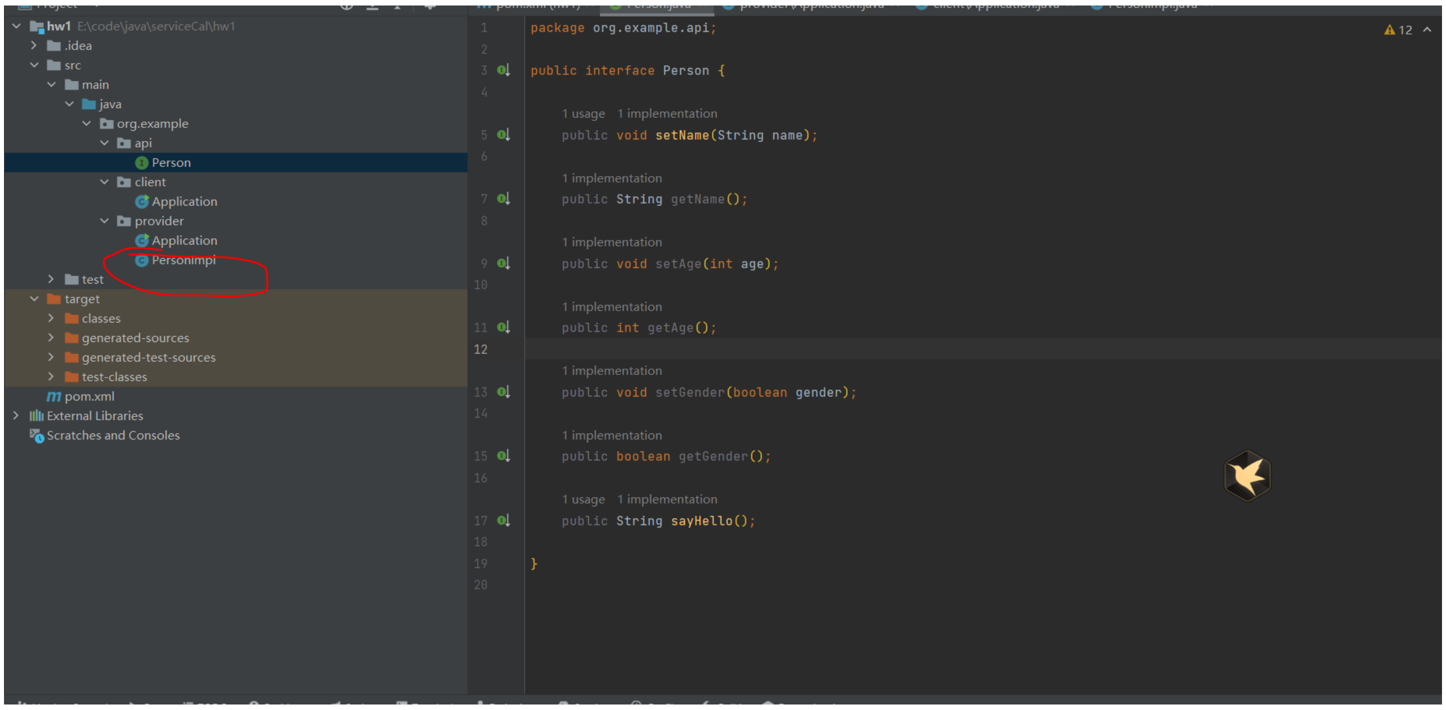
```
package org.example.api;

public interface Person {

    public void setName(String name);

    public String getName();

    public void setAge(int age);

    public int getAge();

    public void setGender(boolean gender);

    public boolean getGender();

    public String sayHello();

}
```



```java
package org.example.api;

public interface Person {

    1 usage   1 implementation
    public void setName(String name);

    1 implementation
    public String getName();

    1 implementation
    public void setAge(int age);

    1 implementation
    public int getAge();

    1 implementation
    public void setGender(boolean gender);

    1 implementation
    public boolean getGender();

    1 usage   1 implementation
    public String sayHello();

}
```

在provider目录下 创建文件 PersonImpl.java，对定义的 Person 接口完成实现

```java
package org.example.provider;

import org.apache.dubbo.config.annotation.DubboService;
import org.example.api.Person;

@DubboService
public class PersonImpl implements Person {
    String name;
    int age;
    boolean gender;

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public int getAge() {
        return age;
    }

    public void setGender(boolean gender) {
        this.gender = gender;
    }

    public boolean getGender() {
        return gender;
    }

    @Override
    public String sayHello() {
        return (" RPC Service: Hello world!" + this.name);
    }
}
```



```java
package org.example.api;

public interface Person {

    public void setName(String name);

    public String getName();

    public void setAge(int age);

    public int getAge();

    public void setGender(boolean gender);

    public boolean getGender();

    public String sayHello();

}
```

3、provider定义及实现

编写 Dubbo Service 代码

定义 zookeeper Host 地址：

```java
private static String zookeeperHost = System
        .getProperty("zookeeper.address", "127.0.0.1");

private static String zookeeperPort = System.getProperty("zookeeper.port",
        "2181");
```

编写 main 函数，先进行注册中心的配置：

```java
public static void main(String[] args) throws Exception {
    ServiceConfig<Person> service = new ServiceConfig<>();
    service.setApplication(new ApplicationConfig("first-dubbo-provider"));
    service.setRegistry(new RegistryConfig(
            "zookeeper://" + zookeeperHost + ":" + zookeeperPort));
    service.setInterface(Person.class);
    service.setRef(new PersonImpl());
    service.export();

    System.out.println("dubbo service started");
    new CountDownLatch(1).await();
}
```

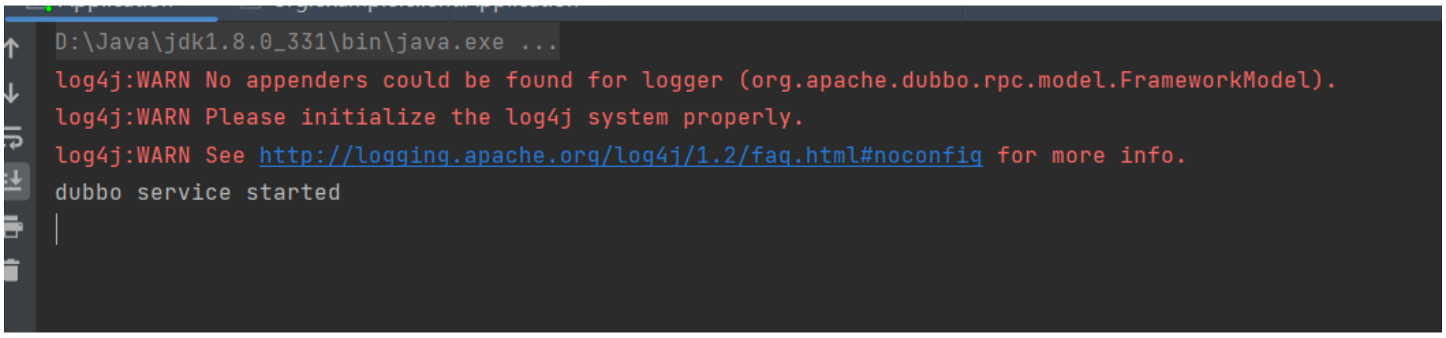
定义接口和具体实现：

```java
service.setInterface(Person.class);
service.setRef(new PersonImpl());
```

启动 Dubbo Service：

```java
service.export();
System.out.println("dubbo service started");
new CountDownLatch(1).await();
```

完成 zookeeper 配置后，启动服务端，可以看到提示 dubbo server 成功启动：

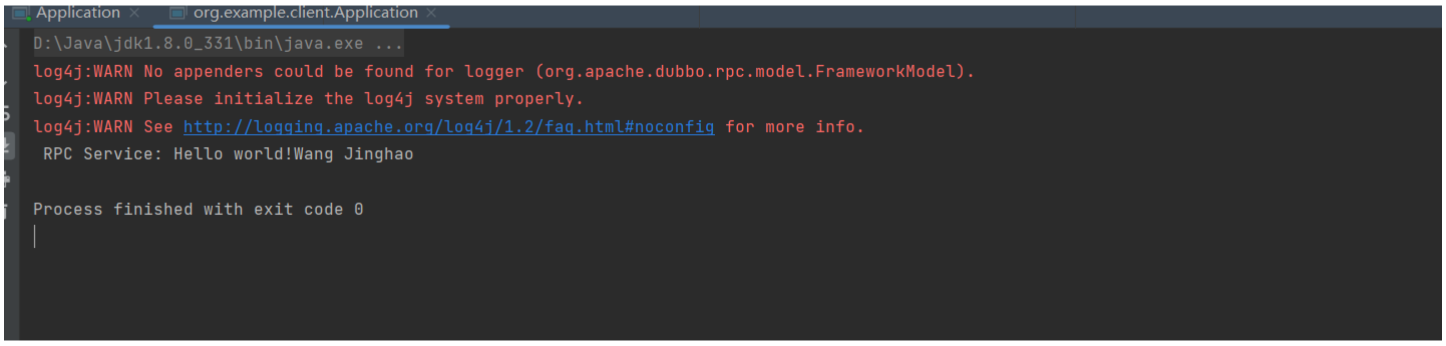


```
D:\Java\jdk1.8.0_331\bin\java.exe ...
log4j:WARN No appenders could be found for logger (org.apache.dubbo.rpc.model.FrameworkModel).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
dubbo service started
```

4、 编写 Dubbo 调用端，发起 rpc 调用

```java
package org.example.client;

import org.apache.dubbo.config.ApplicationConfig;
import org.apache.dubbo.config.ReferenceConfig;
import org.apache.dubbo.config.RegistryConfig;
import org.example.api.Person;

public class Application {
    private static String zookeeperHost = System
            .getProperty("zookeeper.address", "127.0.0.1");
    private static String zookeeperPort = System.getProperty("zookeeper.port",
            "2181");

    public static void main(String[] args) {
        ReferenceConfig<Person> reference = new ReferenceConfig<>();
        reference.setApplication(new ApplicationConfig("first-dubbo-consumer"));
        reference.setRegistry(new RegistryConfig(
                "zookeeper://" + zookeeperHost + ":" + zookeeperPort));
        reference.setInterface(Person.class);
        Person service = reference.get();
        service.setName("Wang Jinghao");
        String message = service.sayHello();
        System.out.println(message);
    }
}
```

- 调用端首先通过 reference 获得远端对象 `service`
- 调用远端对象的 `setName` 方法，设置对象姓名：wangjinghao
- 之后调用对象的 `sayHello` 方法，获得远端对象的返回值，并在终端打印：



```
D:\Java\jdk1.8.0_331\bin\java.exe ...
log4j:WARN No appenders could be found for logger (org.apache.dubbo.rpc.model.FrameworkModel).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
 RPC Service: Hello world!Wang Jinghao

Process finished with exit code 0
```

Code : buaa-service-computing/homework1 at main · Jinghao-coding/buaa-service-computing