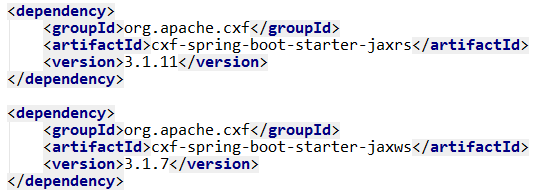
项目中用到webservice提供对外服务，选用框架时主要用Axis和CXF。Axis跨语言，CXF与springboot无缝对接

1.添加依赖



**一、编写服务端**

**2.服务端 向客户端提供服务**

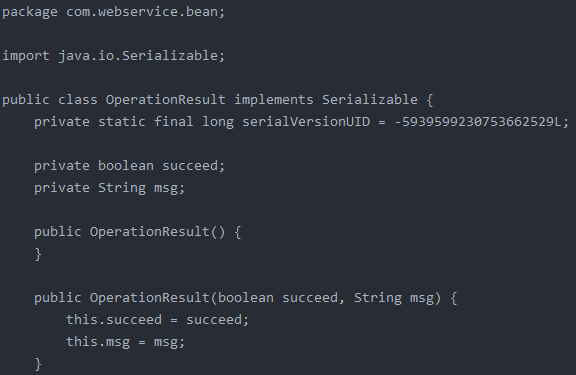
（1）服务端接口



用到的注解：@WebService @WebMethod @WebParam

targetNamespace是包名的倒序

（2）写一个类OperationResult，记录操作结果



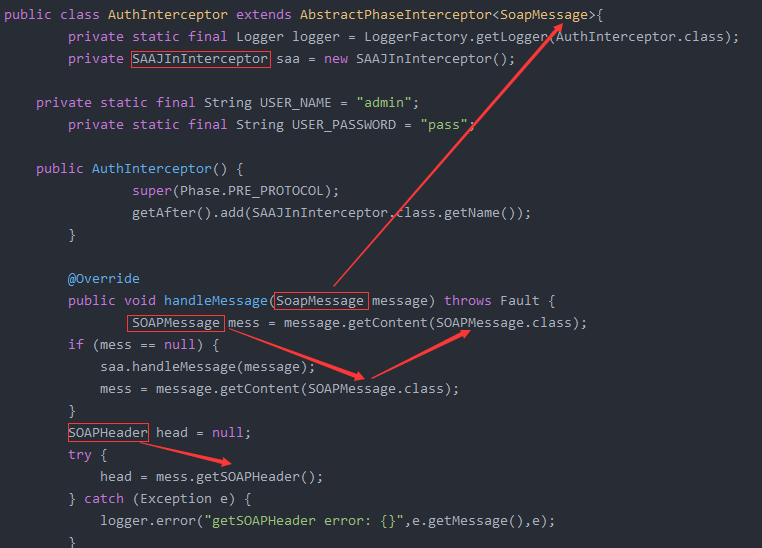
（3）服务端接口的实现类



**3.配置过滤器**

Interceptor是CXF中一个模式，可以动态添加许多功能

例子：用户验证拦截器





**4.配置（将刚配的过滤器放入interceptors中）**

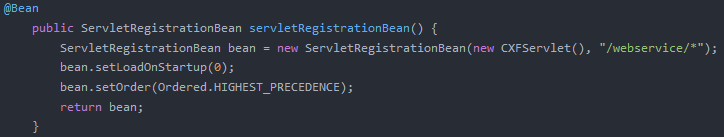


springBoot自动注册了servlet，默认为/services，如果要修改成webservice，两种方式：

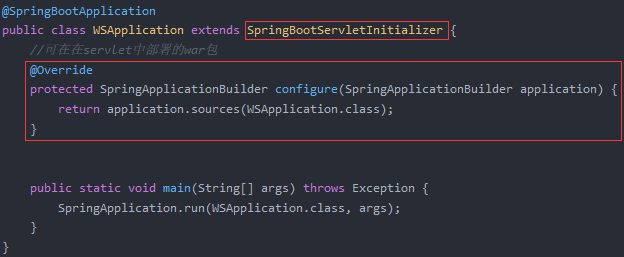
（1）在resources文件夹下新建application.properties文件

cxf.path= /webservice

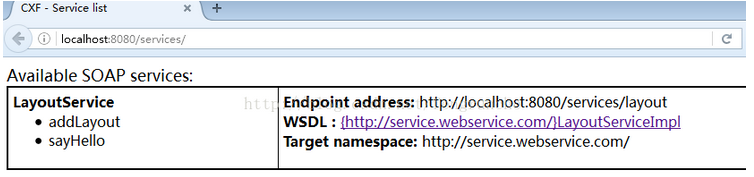
（2）在CXFConfig类中添加



**5.主程序入口**



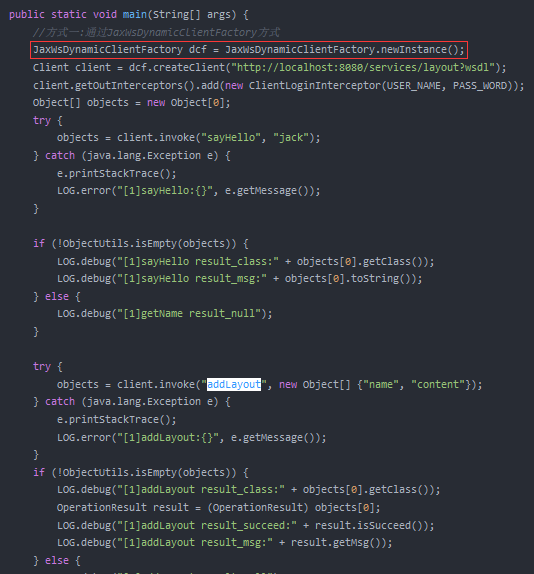
6.服务端配置完成，访问http://lcoalhost:8080/webservice



**二、编写客户端**

客户端有两种方式调用服务端

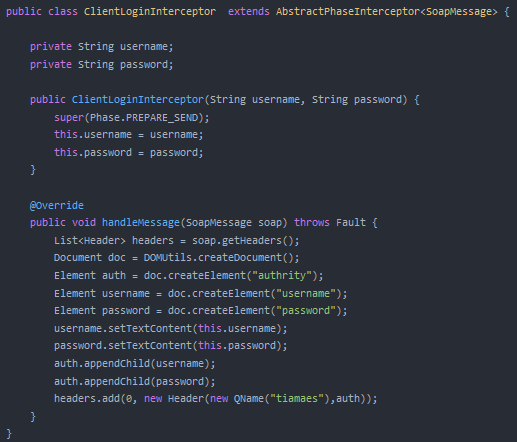
（1）通过JaxWsDynamicClientFactory



这种方式要注意的就是，如果调用的服务接口返回的是一个自定义对象，那么结果Object[]中的数据类型就成了这个自定义对象(组件帮你自动生成了这个对象)，

但是你本地可能并没有这个类，所以需要自行转换处理，最简单的是新建一个跟返回结果一模一样的类进行强转。

客户端需要在header中添加用户信息，这样才能通过服务器的验证。



（2）通过wsimport生成客户端代码

通过jdk自带的wsimport生成客户端代码。进入$JAVA\_HOME/bin下，新建bin和src两个文件夹，执行以下命令：

**wsimport -d ./bin -s ./src -keep http://localhost:8080/services/layout?wsdl**

其中几个参数有以下几个，

-d:生成客户端执行类的class文件存放目录，

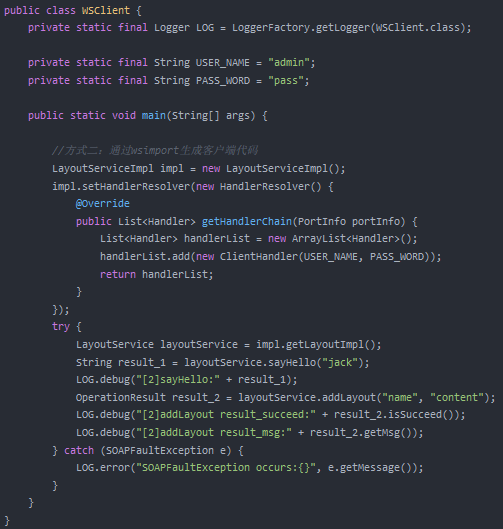
-s:生成客户端执行类的源文件存放目录，

-p:定义生成类的包名

  -verbose:显示生成过程

需要注意的是：无论-d或是-s后的目录必须在文件系统中存在，否则报错。

将src下连同包一起复制到程序中，编写客户端入口类：



用户的头信息设置类：

