

Web Service客户端项目

项目概述

一、 项目介绍

本项目为南京大学软件学院2017届服务计算与SOA课程web service客户端项目。本项目为web service的客户端编写，调用了服务端项目中完成的学生成绩服务、学生信息服务与邮箱Login服务。

二、 项目结构与运行

本项目由三部分组成，目录分别对应为assignment11, infoAndGradeService, loginService。assignment11中为本次的web service客户端项目，infoAndGradeService中为assignment10中的学生信息与学生成绩服务项目，loginService中为assignment10的login服务项目。其中infoAndGradeService又由GradeService和PersonallInfoService两个项目组成。

运行本项目时，首先请启动loginService中的assignment10（login服务），与infoAndGradeService中的GradeService、PersonallInfoService。

loginService中的assignment10的启动类为DemoApplication，GradeService的启动类为ServicePublisher，PersonallInfoService的启动类为ServicePublisher。

三个服务都启动后，即可以运行客户端项目，启动类为webServiceClient包下的Client类，其中覆盖了包括学生信息服务、学生成绩服务、login权限验证服务以及错误处理的用例，详见Client类的主方法中的注释。

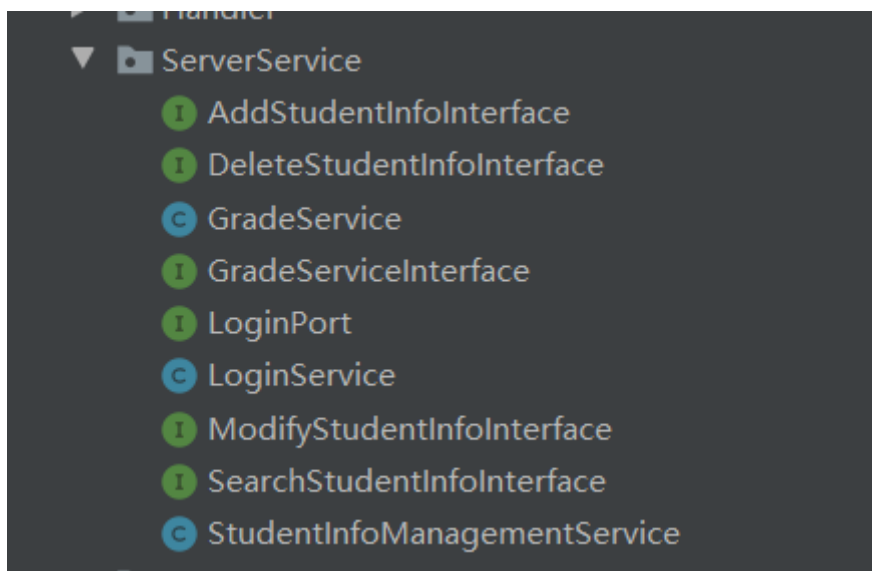
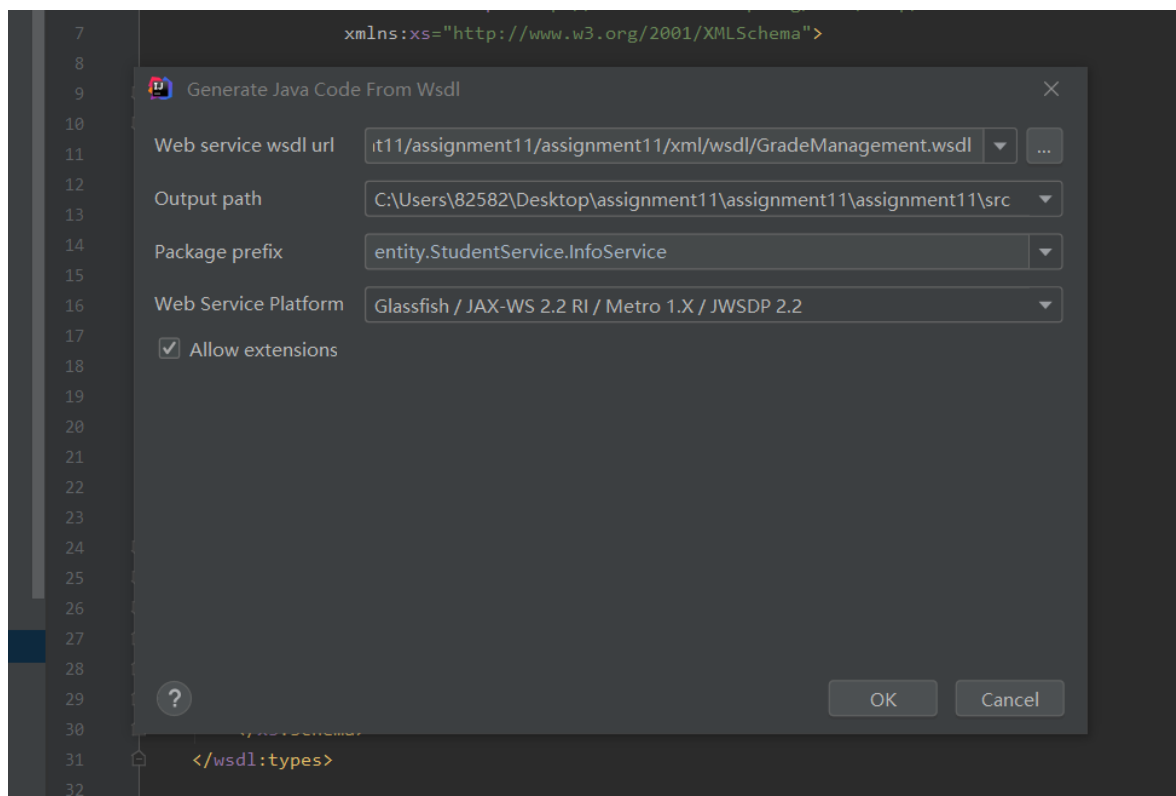
客户端与服务端的所有消息保存在assignment11项目目录下的message文件中。

此外，由于GradeService与PersonallInfoService使用的是简单的内存存储，所以每次重新启动GradeService与PersonallInfoService都相当于将学生信息和学生成绩数据清空。

三、 项目详细介绍

本项目为web service的客户端。

首先使用学生信息服务、学生成绩服务、邮箱login服务的wsdl，使用jetbrain intellij idea的web service工具生成客户端接口类、实现类以及相关的实体对象（见下图）。



然后编写对应于上述每种服务实现类的客户端的服务类，即将学生信息服务、学生成绩服务、邮箱login服务再封装一层。服务类防御webServiceClient包下，为EmailLoginService类、InfoService类、ScoreService类。以ScoreService类举例，其调用了上图中生成的GradeService与GradeServiceInterface，并指定其请求地址（与学生成绩服务项目的服务地址相同）。而因为成绩服务需要用soap消息头部携带的邮箱与密码进行权限验证，故还需要在ScoreService类的对应方法中将email与password两个字段加入soap消息的首部。（见下图）

指定请求地址：

```
public 课程成绩列表类型 addScore(课程成绩列表类型 scoreList,String email,String password) throws MalformedURLException {
    URL url = new URL( "http://localhost:8088/StudentGradeManagement" + "?wsdl");
    GradeService gradeService = new GradeService(url);
    GradeServiceInterface gradeServiceInterface = gradeService.getGradeServiceEndpoint();
    WSBindingProvider bp = (WSBindingProvider)gradeServiceInterface;
    bp.setOutboundHeaders(Headers.create(new QName( namespaceURI: "http://www.nju.edu.cn/schema", localPart: "email"),email), Headers.c
    try{
        return gradeServiceInterface.addGrade(scoreList);
    }
    catch (InvalidRequestFault e){
        System.out.println("错误: "+e.getFaultInfo());
        e.printStackTrace();
    }
}
```

向Header中添加Email与password：

```

URL url = new URL("http://localhost:8088/StudentGradeManagement" + "?wsdl");
GradeService gradeService = new GradeService(url);
GradeServiceInterface gradeServiceInterface = gradeService.getGradeServiceEndpoint();
WSBindingProvider bp = (WSBindingProvider)gradeServiceInterface;
bp.setOutboundHeaders(Headers.create(new QName("http://www.nju.edu.cn/schema", "email"), email), Headers.create(n
try{
    return gradeServiceInterface.addGrade(scoreList);
}

```

此外还需要添加异常处理：

```

try{
    return gradeServiceInterface.addGrade(scoreList);
}
catch (InvalidRequestFault e){
    System.out.println("错误: "+e.getFaultInfo());
    e.printStackTrace();
    return null;
}
catch (AlreadyExistFault e){
    System.out.println("错误: "+e.getFaultInfo());
    e.printStackTrace();
    return null;
}

```

客户端服务类编写完成后编写Handler，放于Handler包下。PersistenceHandler负责将每个消息保存到文件（assignment11项目目录下的message文件）中，LoginHandler负责对成绩服务的相关消息调用邮箱login服务进行权限验证。InfoModifyHandler负责在学生成绩发生更新时，同步更新学生信息中的成绩列表。

LoginHandler中对Header中email与password的获取以及对Login服务的调用和权限验证:

```

try{
    SOAPMessage msg = context.getMessage();

    SOAPEnvelope env = msg.getSOAPPart().getEnvelope();
    SOAPHeader hdr = env.getHeader();
    SOAPBody body = env.getBody();
    EmailLoginService emailLoginService = new EmailLoginService();
    if(hdr!=null){
        NodeList nodeList = hdr.getChildNodes();
        String email = nodeList.item( index: 0).getTextContent();
        String password = nodeList.item( index: 1).getTextContent();
        EmailAuthenticationInfoType param = new EmailAuthenticationInfoType();
        param.setEmail(email);
        param.setPwd(password);
        EmailAuthenticationResultType res = emailLoginService.login(param);

        if(res.getEmailIdentity().toString().equals("老师")){
            return true;
        }
        else if(res.getEmailIdentity().toString().equals("本科生")||res.getEmailIdentity().toString().equals("研究生")){

            System.out.println();
            //-----此操作为查询成绩操作则通过权限验证-----
            if(body.getElementsByTagNameNS( "http://jw.nju.edu.cn/schema", "localName: "searchRequest").getLength()==1){
                System.out.println("您的身份为: "+res.getEmailIdentity().toString()+"，没有执行此操作的权限。");
                return false;
            }
            else{
                return true;
            }
        }
    }
}

```

InfoModifyHandler中对学生信息的同步修改:

```

SOAPMessage msg = context.getMessage();

SOAPEnvelope env = msg.getSOAPPart().getEnvelope();
SOAPHeader hdr = env.getHeader();
SOAPBody body = env.getBody();
NodeList nodeList = body.getChildNodes();
String kind = nodeList.item( index: 0).getLocalName();
//判断是否为增加、删除或修改成绩操作
if(kind.equals("addResponse")||kind.equals("deleteResponse")||kind.equals("modifyResponse")){
    NodeList headerNodeList = hdr.getChildNodes();
    String id = body.getElementsByTagNameNS( namespaceURI: "http://jw.nju.edu.cn/schema", localName: "学号").item( index: 0).getLocalName();

    ScoreService scoreService = new ScoreService();

    课程成绩列表类型 scoreList = scoreService.searchScore(id,email,password);

    InfoService infoService = new InfoService();
    学生类型 student = infoService.searchStudentInfo(id);
    student.set课程成绩列表(scoreList);
    infoService.modifyStudentInfo(student);
}
}

```

编写完Handler后，编写xml配置文件并在对应的服务实现类上添加注解，以将编写好的Handler配置到每个服务上。xml配置文件放在allocation包下。其中，login-handler.xml配置了InfoModifyHandler、LoginHandler、PersistenceHandler，用于学生成绩服务，进行权限验证、学生信息的同步修改以及消息的持久化。

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<javaee:handler-chains
    xmlns:javaee="http://java.sun.com/xml/ns/javaee"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <javaee:handler-chain>
        <javaee:handler>
            <javaee:handler-class>Handler.LoginHandler</javaee:handler-class>
        </javaee:handler>
        <javaee:handler>
            <javaee:handler-class>Handler.PersistenceHandler</javaee:handler-class>
        </javaee:handler>
        <javaee:handler>
            <javaee:handler-class>Handler.InfoModifyHandler</javaee:handler-class>
        </javaee:handler>
    </javaee:handler-chain>
</javaee:handler-chains>

```

message-handler.xml中配置了PersistenceHandler，用于Login服务与学生信息服务的消息持久化。

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<javaee:handler-chains
    xmlns:javaee="http://java.sun.com/xml/ns/javaee"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <javaee:handler-chain>
        <javaee:handler>
            <javaee:handler-class>Handler.PersistenceHandler</javaee:handler-class>
        </javaee:handler>
    </javaee:handler-chain>
</javaee:handler-chains>

```

在对应的服务实现类上添加@HandlerChain注解，利用xml配置文件将Handler配置到服务上。下图以ServerService包的GradeService类举例。

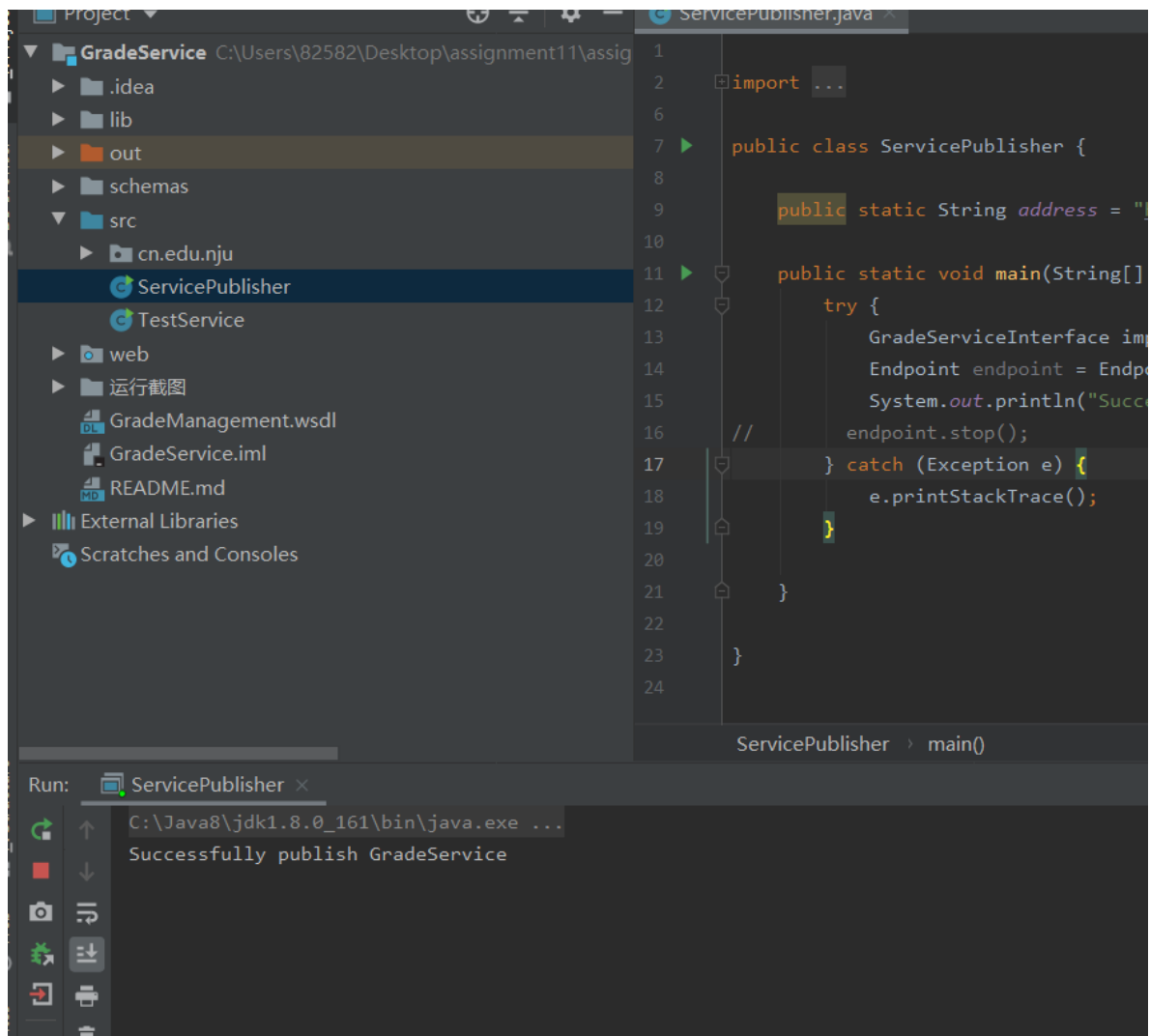
```
ScoreService.java x InfoModifyHandler.java x login-handler.xml x LoginService.java x GradeService.java x LoginHandler.java x
21 @HandlerChain(file = "allocation/login-handler.xml")
22 @WebServiceClient(name = "GradeService", targetNamespace = "http://jw.nju.edu.cn/wsdl", wsdlLocation = "file:/C:/Users/825
23 public class GradeService
24     extends Service
25 {
26
27     private final static URL GRADESERVICE_WSDL_LOCATION;
```

最后编写webServiceClient包下的Client类的主方法以覆盖每个用例。

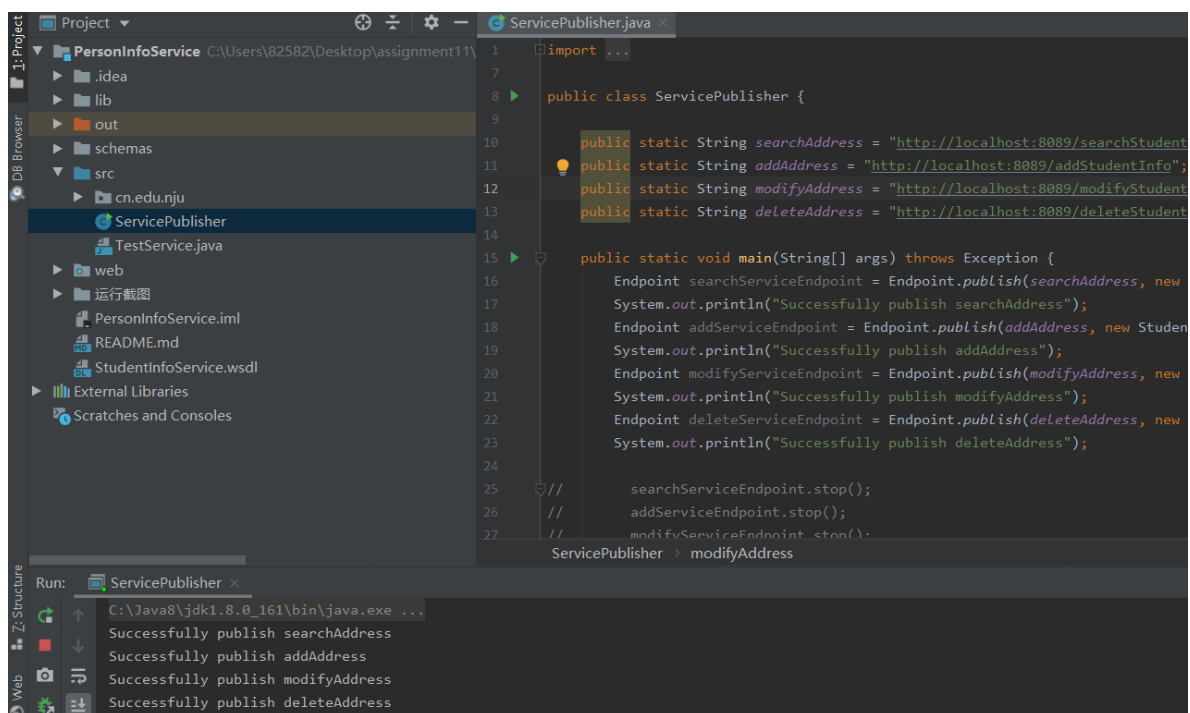
```
ScoreService.java x InfoModifyHandler.java x Client.java x login-handler.xml x LoginService
5 import java.math.BigInteger;
6 import java.net.MalformedURLException;
7 import java.util.List;
8
9 public class Client {
10     public static void main(String[] args) throws MalformedURLException {
11         InfoService infoService = new InfoService();
12         //-----查找不存在的学生信息，返回错误-----
13         System.out.println("查找不存在的学生信息，返回错误：");
14         infoService.searchStudentInfo( id: "171250640");
15
16         //-----新增学生信息（初始成绩列表为空,个人信息填写姓名性别和电话）-----
17         System.out.println("新增学生信息（初始成绩列表为空,个人信息填写姓名性别和电话）：");
18         学生类型 student = new 学生类型();
19         student.set学号("171250640");
20         课程成绩列表类型 scoreList = new 课程成绩列表类型();
21         个人信息类型 info = new 个人信息类型();
```

四、运行截图

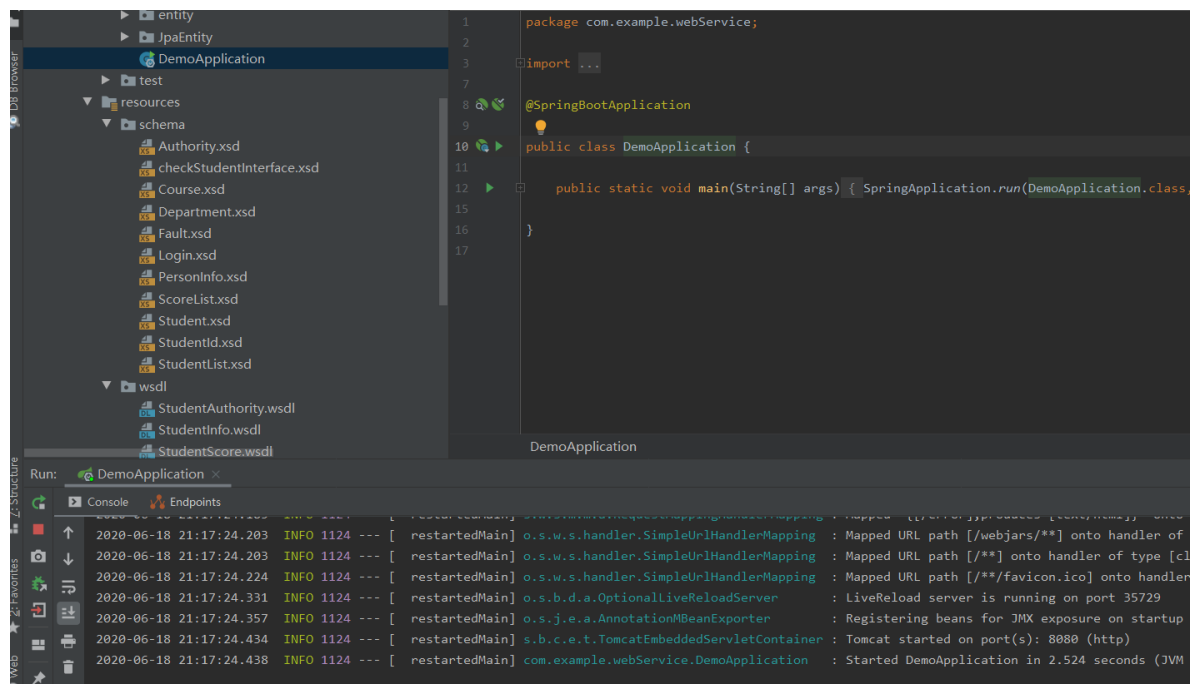
启动GradeService项目：



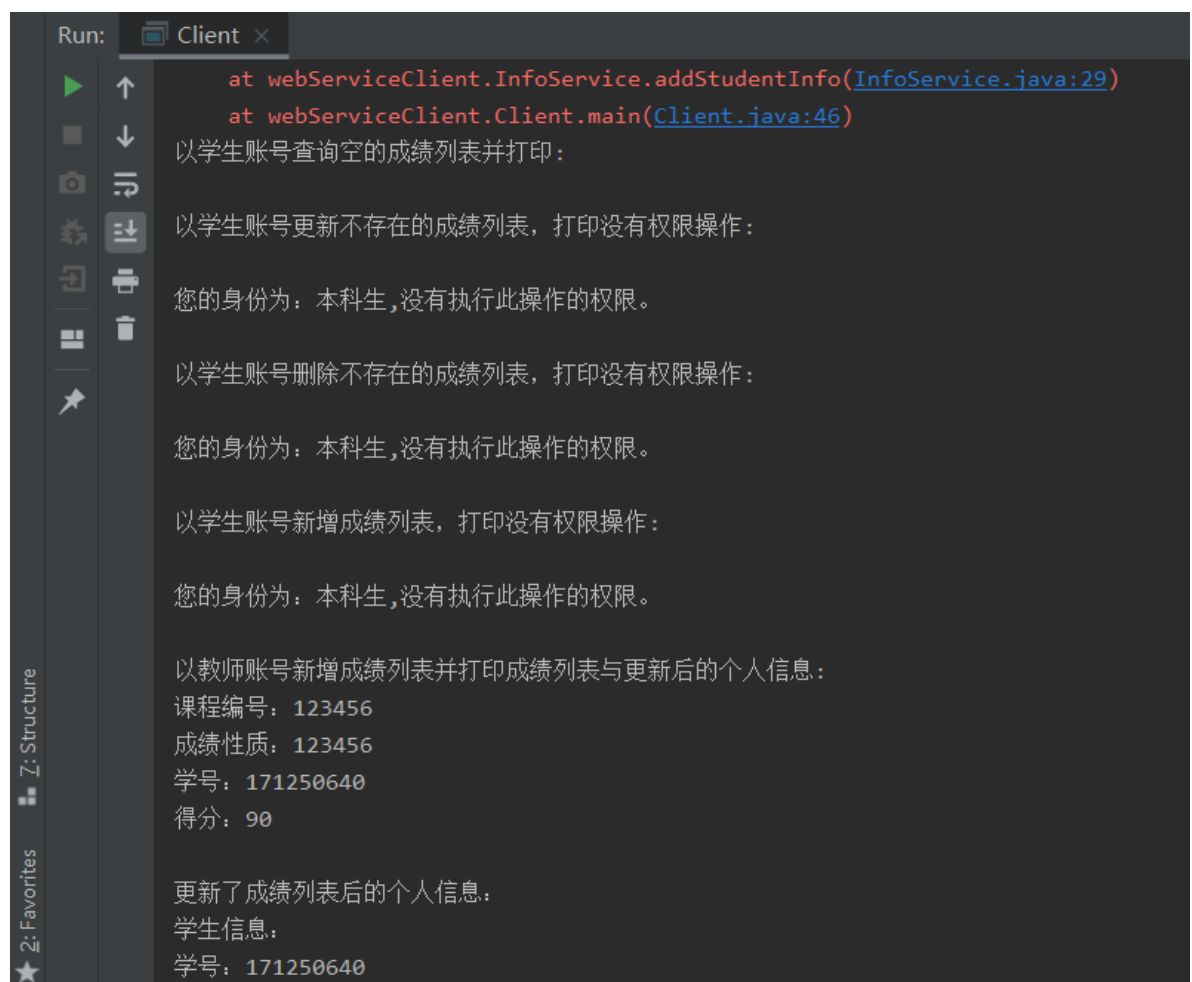
启动PersonalInfoService项目：



启动Login服务：



运行assignment11的Client类main方法：



message文件中的消息记录：

```
ScoreService.java x InfoModifyHandler.java x Client.java x message-handler.xml x message x login-handler.xml x LoginService.java
1 REQUEST:
2 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
3 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
4   <S:Header>
5     <op>searchInfo</op>
6   </S:Header>
7   <S:Body xmlns="http://jw.nju.edu.cn/schema" xmlns:ns2="http://www.nju.edu.cn/schema">
8     <学号>171250640</学号>
9   </S:Body>
10 </S:Envelope>
11
12 RESPONSE:
13 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
14 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
15   <SOAP-ENV:Header/>
16   <S:Body xmlns:ns4="http://www.w3.org/2003/05/soap-envelope">
17     <S:Fault>
18       <faultcode>S:Server</faultcode>
19       <faultstring>cn.edu.nju.jw.wsdl.CannotFindFault</faultstring>
20       <detail xmlns="" xmlns:faultReason="http://jw.nju.edu.cn/schema" xmlns:ns2="http://www.nju.edu.cn/schema">
21         <faultReason:faultReason>Cannot Find</faultReason:faultReason>
22       </detail>
23     </S:Fault>
24   </S:Body>
25 </S:Envelope>
26
27 REQUEST:
28 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
29 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
30   <S:Header>
```