面向服务软件工程 Assignment 2

1. Task 1: 认证服务 (Start from Java)

1.1 实现逻辑

采用 Bottom-Up 模式，先定义 Java 接口 (LoginService)，利用 JAX-WS 注解 (@WebService, @WebMethod) 暴露服务。服务通过邮箱后缀判断用户身份（老师/本科生/研究生），并考虑对非法输入抛出异常。
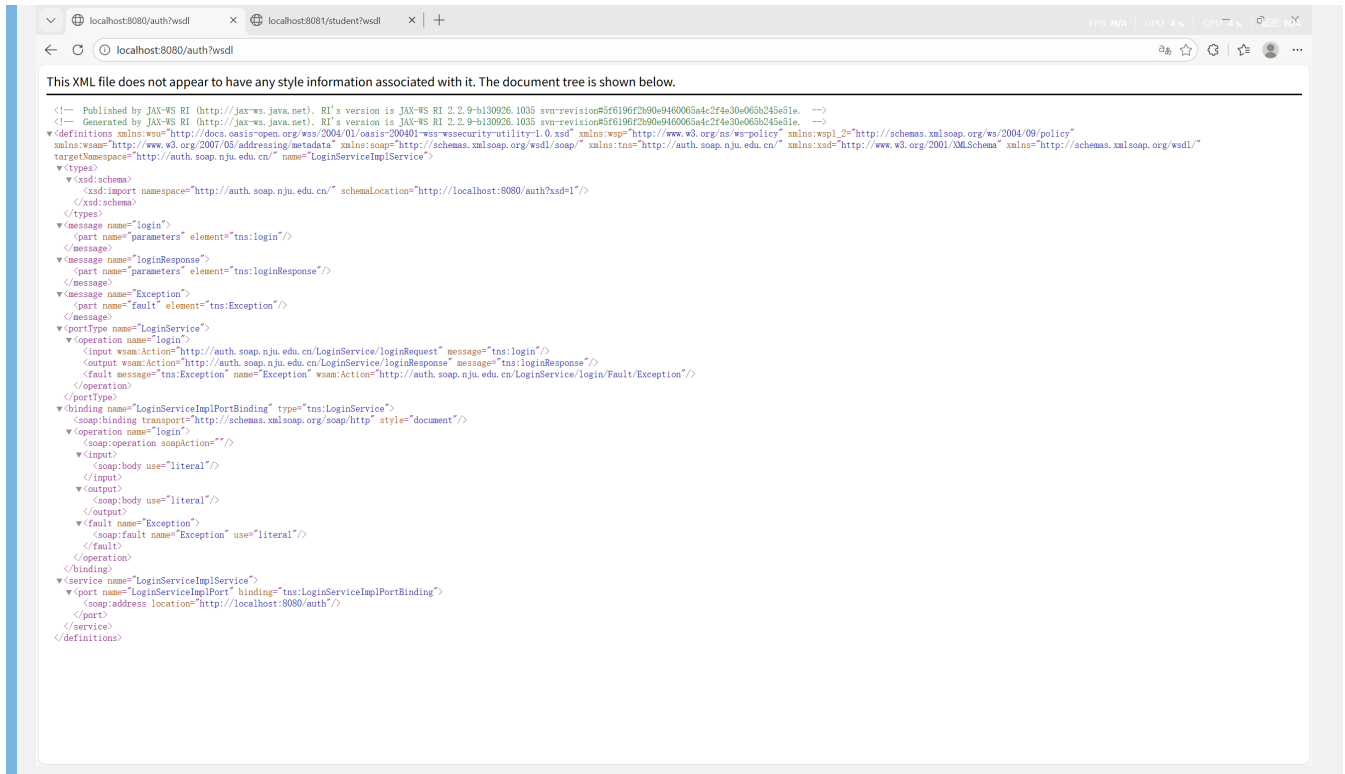
2.2 关键代码实现

接口定义:

```java
@WebService
public interface LoginService {
    @WebMethod
    String login(@WebParam(name = "email") String email,
                 @WebParam(name = "password") String password) throws Exception;
}
```

服务发布: 为了满足"学院内网可访问性"的要求，服务绑定地址设为 0.0.0.0。

```java
String address = "http://0.0.0.0:8080/auth";
Endpoint.publish(address, new LoginServiceImpl());
```

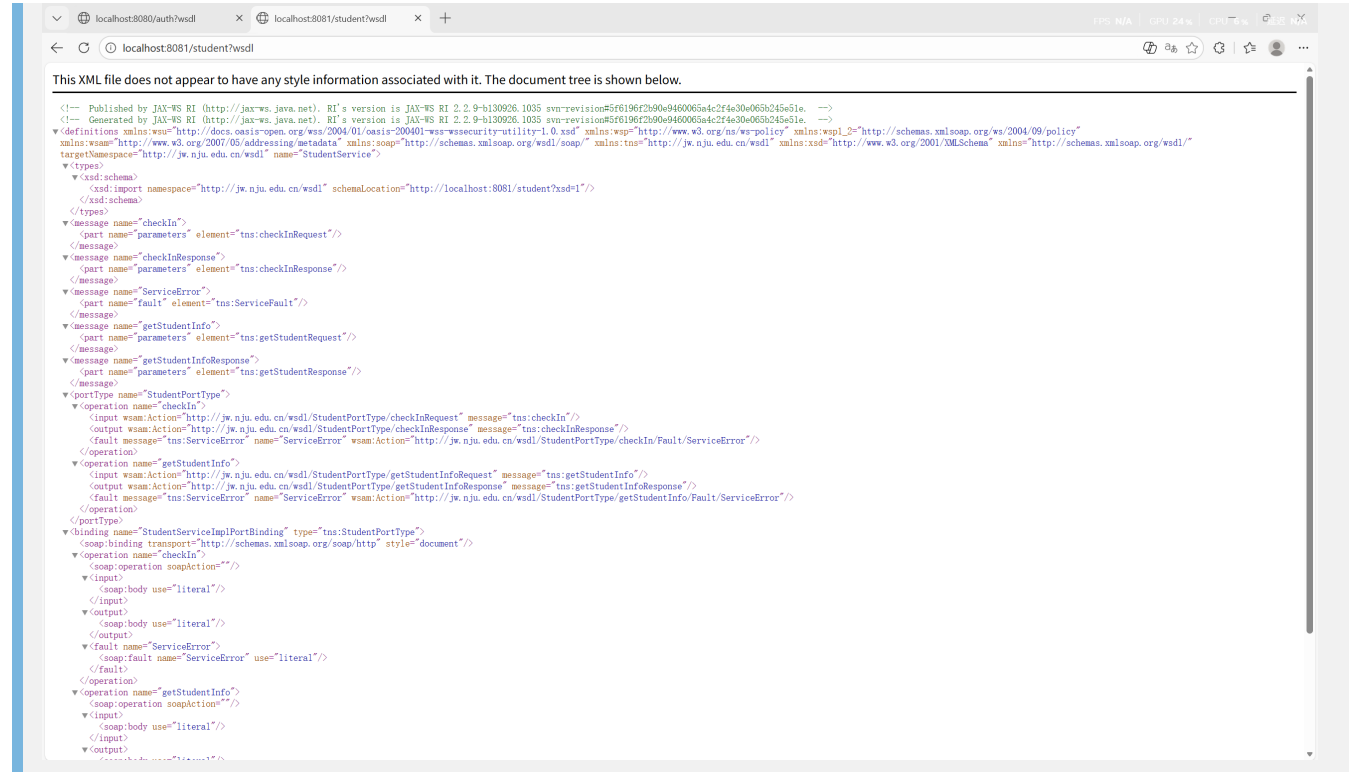### 2.3 运行结果验证

2. Task 2: 业务服务 (Start from WSDL)

2.1 实现逻辑

采用 Top-Down 模式，首先编写 WSDL 文档，定义了 `StudentType` 数据结构以及 `getStudentInfo` (档案获取) 和 `checkIn` (签到) 两个操作。

2.2 代码生成与实现

使用 JDK 自带工具 `wsimport` 生成服务端接口代码： `wsimport -keep -p cn.edu.nju.soap.biz -d . ../xml_data/StudentService.wsdl`

3.3 运行结果验证

3. 客户端测试 (Client)

3.1 测试用例设计

1. **Login 测试**: 验证正确邮箱返回角色，错误邮箱抛出异常。
2. **档案测试**: 使用老师邮箱（高权限）成功获取 Mock 学生数据。
3. **签到测试**: 模拟学生完成签到操作。

3.2 控制台输出结果

程序输出结果如下，符合预期。

```
>>> 1. 测试 Login 服务
Testing Teacher: TEACHER
Testing Student: UNDERGRADUATE
Testing Error:
Expected Exception: AuthError: Invalid email domain. Use nju.edu.cn or
smail.nju.edu.cn

>>> 2. 测试 Archive & CheckIn 服务
尝试获取档案(使用老师邮箱)...
Success: Got student Mock Student
尝试签到...
Student 1001 checked in for SOA_2024
```

```
"C:\Program Files (x86)\Java\jdk-1.8\bin\java.exe" ...
>>> 1. 测试 Login 服务
Testing Teacher: TEACHER
Testing Student: UNDERGRADUATE
Testing Error:
Expected Exception: AuthError: Invalid email domain. Use nju.edu.cn or smail.nju.edu.cn

>>> 2. 测试 Archive & CheckIn 服务
尝试获取档案(使用老师邮箱)...
Success: Got student Mock Student
尝试签到...
Student 1001 checked in for SOA_2024


进程已结束，退出代码为 0
```