SOA 作业报告

作业 10

2017-6-15

南京大学软件学院
18 组

目录

1、作业简介

本次作业使用 SpringBoot + Apache CXF 为 webservice 框架，发布于 springboot 自带的 tomcat 服务器上，使用了 Mysql 数据库，并使用了 Spring Data JPA 进行 ORM，使用 maven 构建项目。

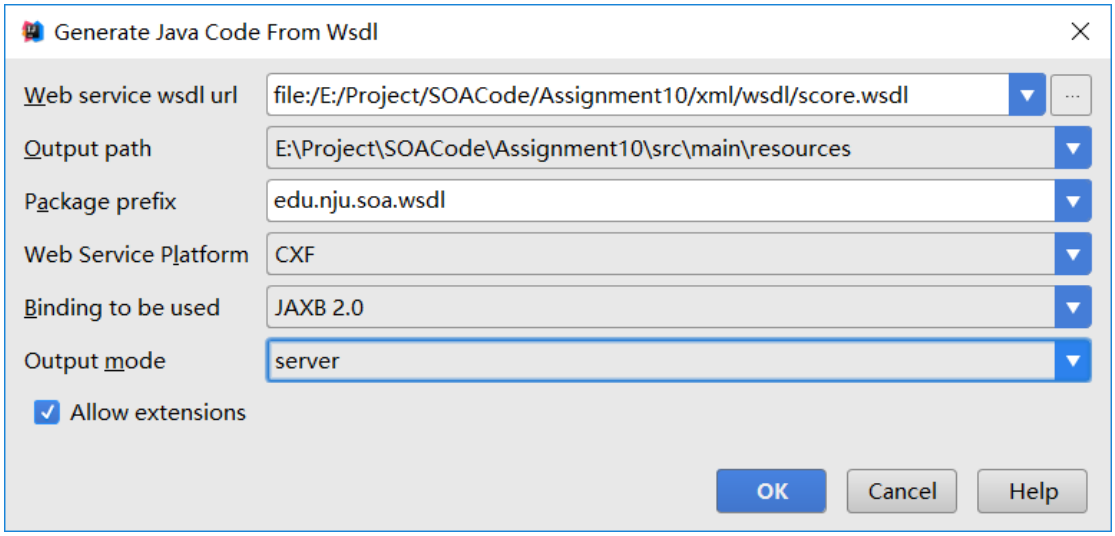
虽然 SOAP 是一种比较古老的技术，但仍然可以使用最新的平台和技术发挥它的作用。服务已经发布在服务器：http://106.15.183.234:8080/soap-api

2、具体步骤

（1）转换作业 7、8 的 wsdl

因为作业 7、8 使用的 wsdl 是 2.0 的，而 start from wsdl 转换工具 wsdl2java 不支持 2.0 的转换，因此首先需要将之前作业的 wsdl 文件转换成 1.2 的版本。
转换的文件位于/xml/wsdl 文件夹，为了转换需要新增了一些输入类型。

（2）从 wsdl 生成 java 文件



使用本地文件生成 wsdl 文件

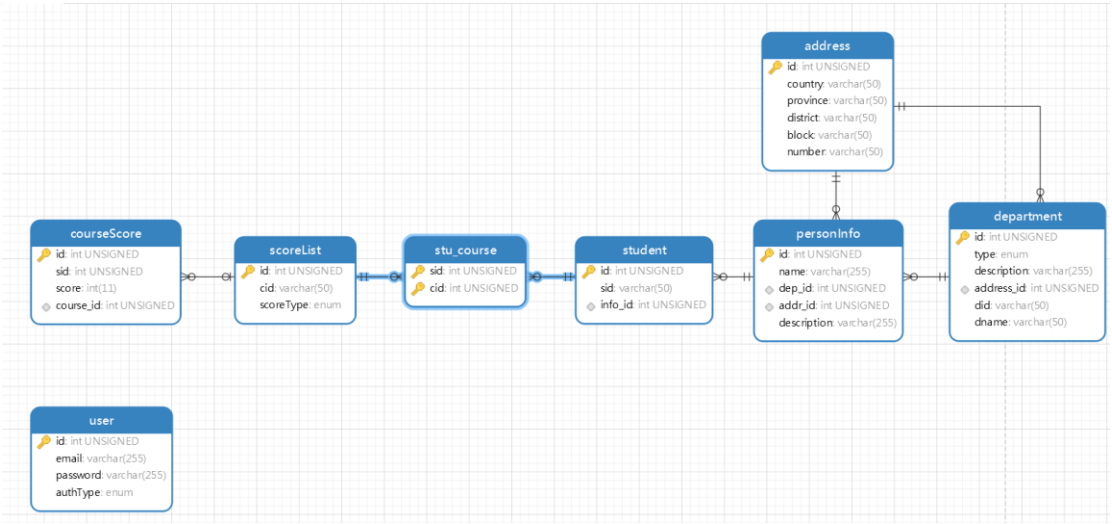
（3）完善 model 与 entity

a、refactor

因为之前定义的 schema 为中文，但是因为编码不方便，所以将类和属性的名字改成中文，但是保持定义 schema 的中文名字不变。

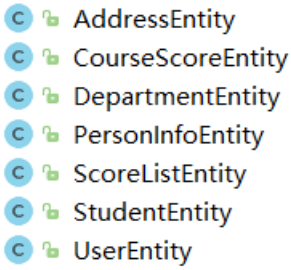
b、数据库建表

为了映射方便，应当保持表结构与 schema 定义类似。



c、生成映射 entity

自动生成一部分代码，手动添加表关联（One to many, Many to Many 等）注解。




```
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
@Column(name = "id", nullable = false)
public int getId() { return id; }

public void setId(int id) { this.id = id; }

@ManyToOne(fetch = FetchType.EAGER)
@JoinColumn(name = "info_id",referencedColumnName = "id")
public PersonInfoEntity getPersonInfo() { return personInfo; }

public void setPersonInfo(PersonInfoEntity personInfo) { this.personInfo = personInfo; }

@ManyToMany
@JoinTable(name = "stu_course", schema = "soa",
        joinColumns = @JoinColumn(name = "sid", referencedColumnName = "id",nullable = false),
        inverseJoinColumns = @JoinColumn(name = "cid", referencedColumnName = "id",nullable = false))
public List<ScoreListEntity> getScoreList() { return scoreList; }
```

（4）数据库 JPA

定义 Spring Data JPA 接口，比如：

```
/**
 * Created by cuihao on 2017-06-13.
 * Auth repo definition
 */
public interface UserRepository extends JpaRepository<UserEntity,Integer> {

    UserEntity findByEmail(String email);

}
```

定义并实现 DAO

```
@Repository
public class UserDaoImpl implements UserDao {

    @Resource
    private UserRepository userRepository;

    @Override
    public UserEntity findById(int id) { return userRepository.findOne(id); }

    @Override
    public UserEntity findByEmail(String email) { return userRepository.findByEmail(email); }

    @Override
    public void delete(int id) { userRepository.delete(id); }

    @Override
    public UserEntity save(UserEntity userEntity) { return userRepository.save(userEntity); }
}
```

```java
@Override
@Transactional
public StudentEntity save(StudentEntity studentEntity) {
    PersonInfoEntity personInfo = studentEntity.getPersonInfo();
    DepartmentEntity departmentEntity = personInfo.getDepartment();
    departmentEntity.setAddress(addressRepository.save(departmentEntity.getAddress()));
    personInfo.setDepartment(departmentRepository.save(departmentEntity));
    personInfo.setAddress(addressRepository.save(personInfo.getAddress()));
    studentEntity.setPersonInfo(personInfoRepository.save(personInfo));
    List<ScoreListEntity> scoreListEntities = studentEntity.getScoreList();
    if (scoreListEntities!=null && !scoreListEntities.isEmpty()) {
        studentEntity.setScoreList(scoreListEntities.stream().map(s->scoreDao.save(s)).collect(Collectors.toList()));
    }
    return studentRepository.save(studentEntity);
}
```

（5）发布 webservice

作业 7、8 的接口使用生成的 java 文件中的接口，邮箱验证接口自己定义，然后实现这个 web service 接口。
需要注意的是，同一门课程并且成绩类型相同时应该将很多成绩放在一个课程的标签里。
示例：

```java
/**
 * Add score
 * @param courseScoreListTypeHolder Mode: in and out
 * @throws IdNotFoundException cannot find id
 * @throws ScoreTypeException score type error
 * @throws ScoreModifyException score modify error
 * @throws AuthorityException do not have authority
 */
@WebMethod(action = "score/addScore")
@RequestWrapper(localName = "addScore", targetNamespace = "http://jw.nju.edu.cn/schema", className = "cn.edu.nju.jw.
@ResponseWrapper(localName = "modifyScore", targetNamespace = "http://jw.nju.edu.cn/schema", className = "cn.edu.nju
@Action(input = "score/addScoreRequest", output = "score/addScoreResponse", fault = {
    @FaultAction(className = IdNotFoundException.class, value = "score/addScore/Fault/IdNotFoundException"),
    @FaultAction(className = ScoreTypeException.class, value = "score/addScore/Fault/ScoreTypeException"),
    @FaultAction(className = ScoreModifyException.class, value = "score/addScore/Fault/ScoreModifyException"),
    @FaultAction(className = AuthorityException.class, value = "score/addScore/Fault/AuthorityException")
})
void addScore(
        @WebParam(name = "课程成绩列表", targetNamespace = "http://jw.nju.edu.cn/schema", mode = WebParam.Mode.INOUT)
                Holder<CourseScoreListType> courseScoreListTypeHolder)
    throws AuthorityException, IdNotFoundException, ScoreModifyException, ScoreTypeException
;
```

```java
@Override
@Transactional
public void addScore(Holder<CourseScoreListType> courseScoreListTypeHolder)
        throws AuthorityException, IdNotFoundException, ScoreModifyException, ScoreTypeException {
    List<CourseScore> courseScores = courseScoreListTypeHolder.value.getCourseScores();
    validate(courseScores);
    courseScoreListTypeHolder.value.setCourseScores(courseScores.stream()
            .map(ScoreListEntity::new)
            .map(scoreListEntity -> scoreDao.save(scoreListEntity))
            .map(CourseScore::new).collect(Collectors.toList()));
}
```

（6）配置 webservice

新建 config 类（@Configuration）注解，，添加以下配置：

```
@Bean
public ServletRegistrationBean dispatcherServlet() {
    return new ServletRegistrationBean(new CXFServlet(),  ...urlMappings: "/soap-api/*");
}

@Bean(name= Bus.DEFAULT_BUS_ID)
public SpringBus springBus() { return new SpringBus(); }
```

随后注入刚刚写的 service

```
@Bean(name = "authEndpoint")
public Endpoint endpoint1() {
    EndpointImpl endpoint = new EndpointImpl(springBus(), authController);
    endpoint.publish( addr: "/auth");
    return endpoint;
}

@Bean(name = "scoreEndpoint")
public Endpoint endpoint2() {
    EndpointImpl endpoint = new EndpointImpl(springBus(), scoreController);
    endpoint.publish( addr: "/score");
    return endpoint;
}

@Bean(name = "studentEndpoint")
public Endpoint endpoint3() {
    EndpointImpl endpoint = new EndpointImpl(springBus(), studentController);
    endpoint.publish( addr: "/student");
    return endpoint;
}
```

（7）运行项目

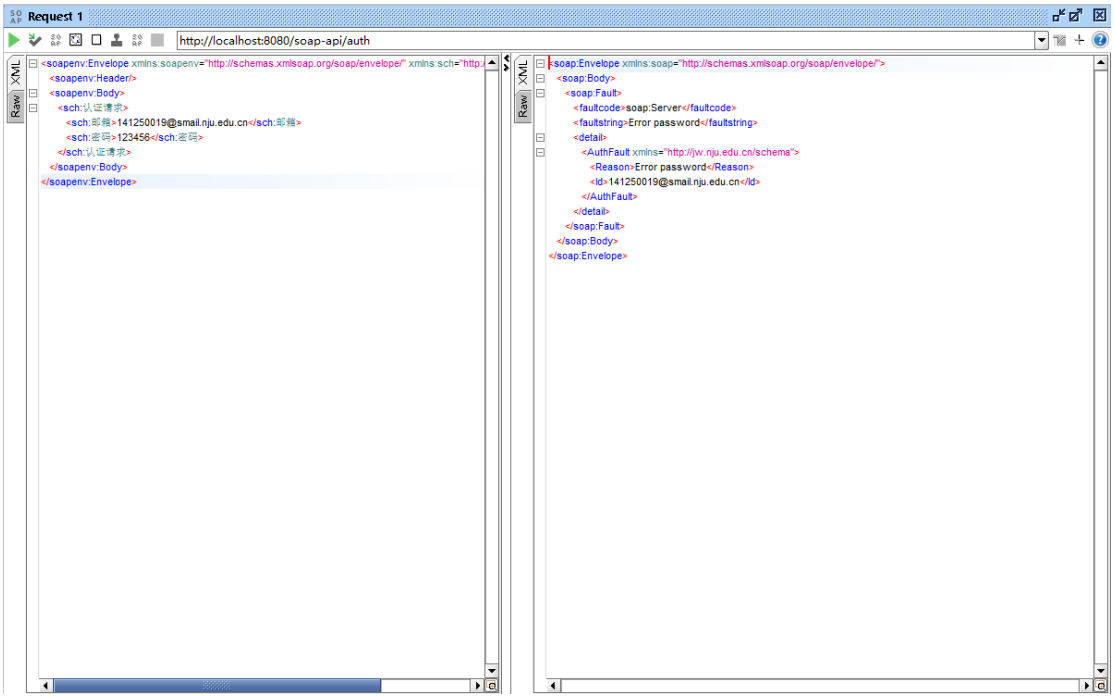
直接运行 main 方法即可

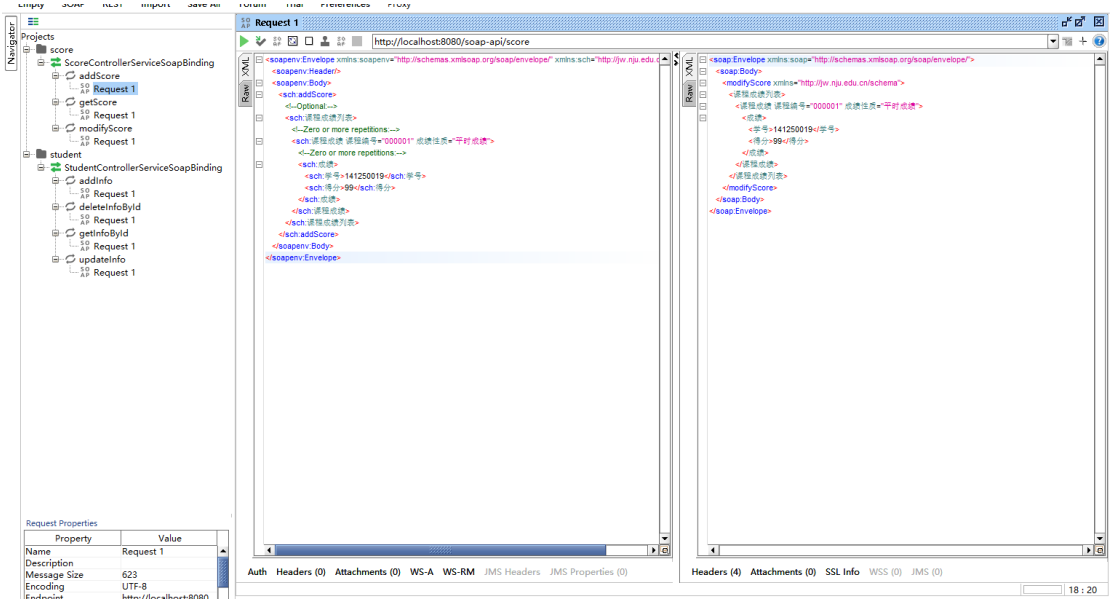
```
/**
 * Created by cuihao on 2017-06-11.
 * Main entrance of student manage application
 */
@SpringBootApplication
public class StudentApplication {

    public static void main(String[] args) { SpringApplication.run(StudentApplication.class, args); }

}
```

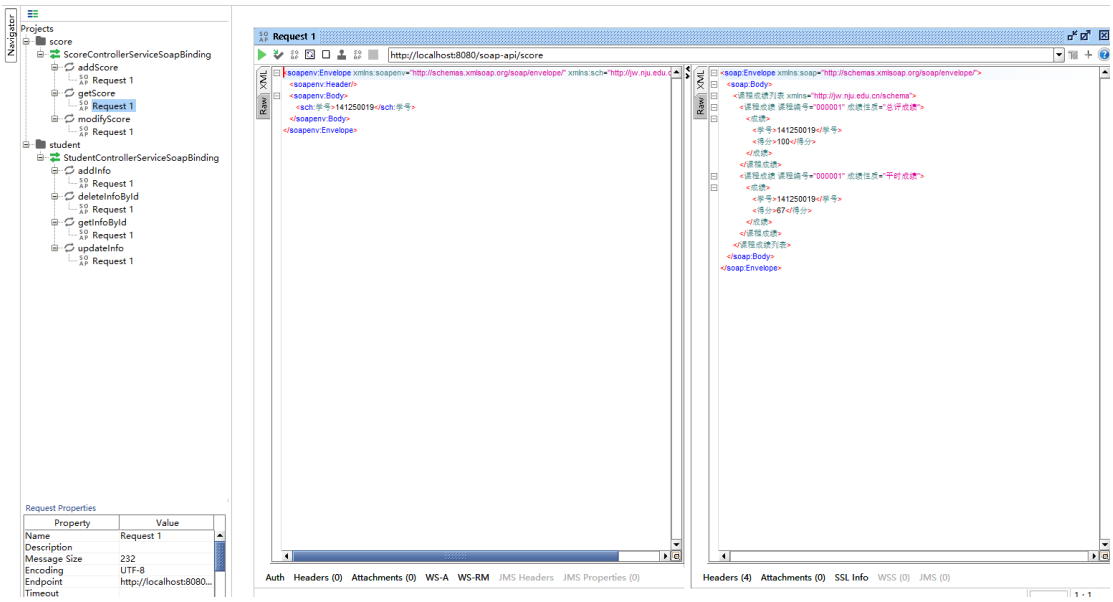
（8）测试接口

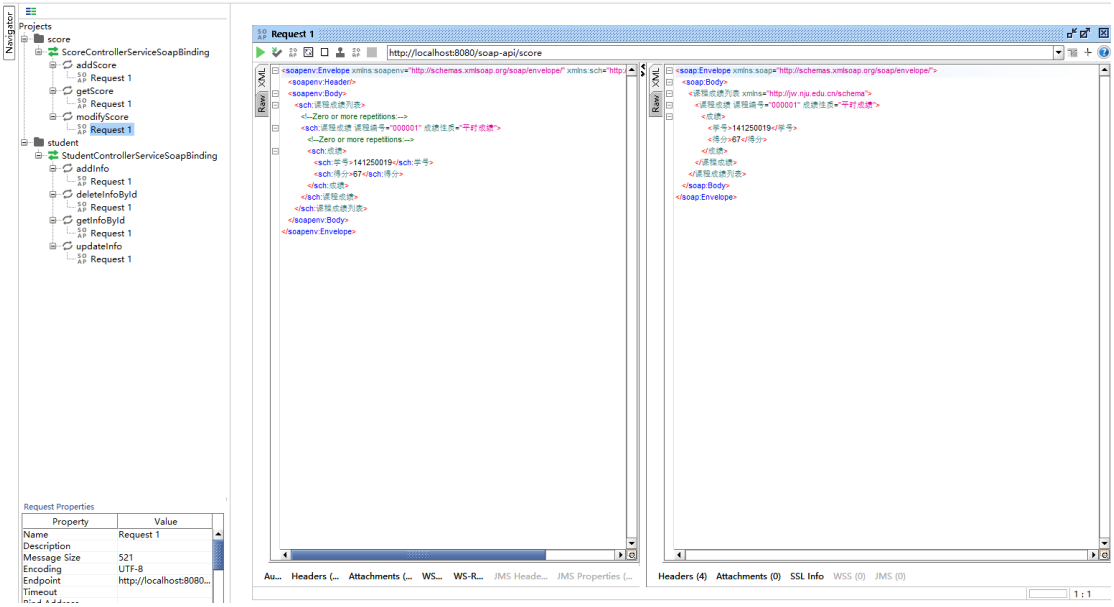
使用 SoapUI 测试服务发布的接口：

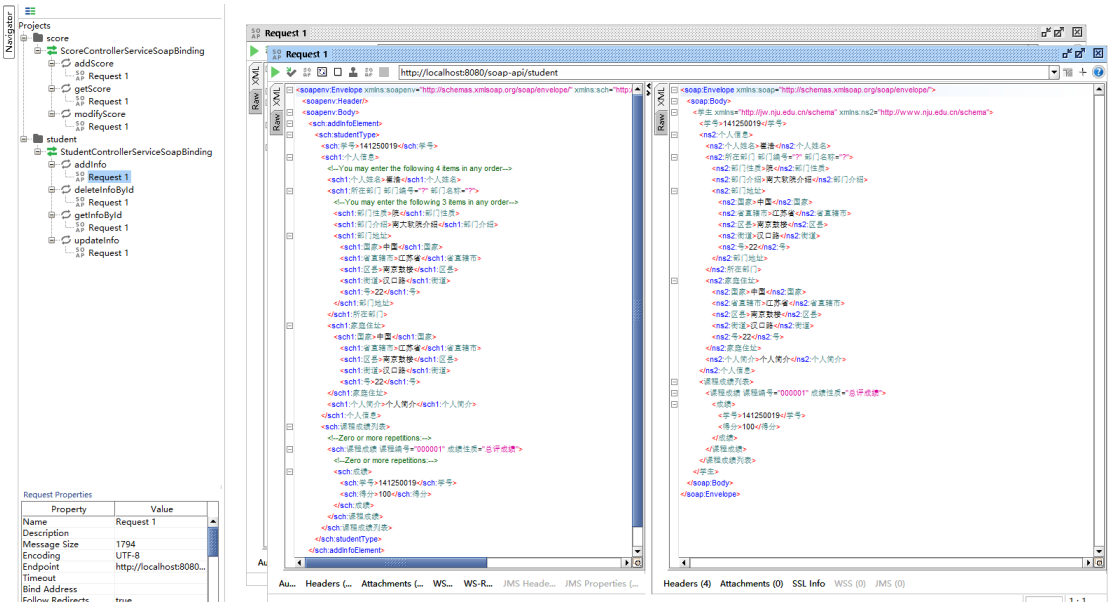
测试 fault：

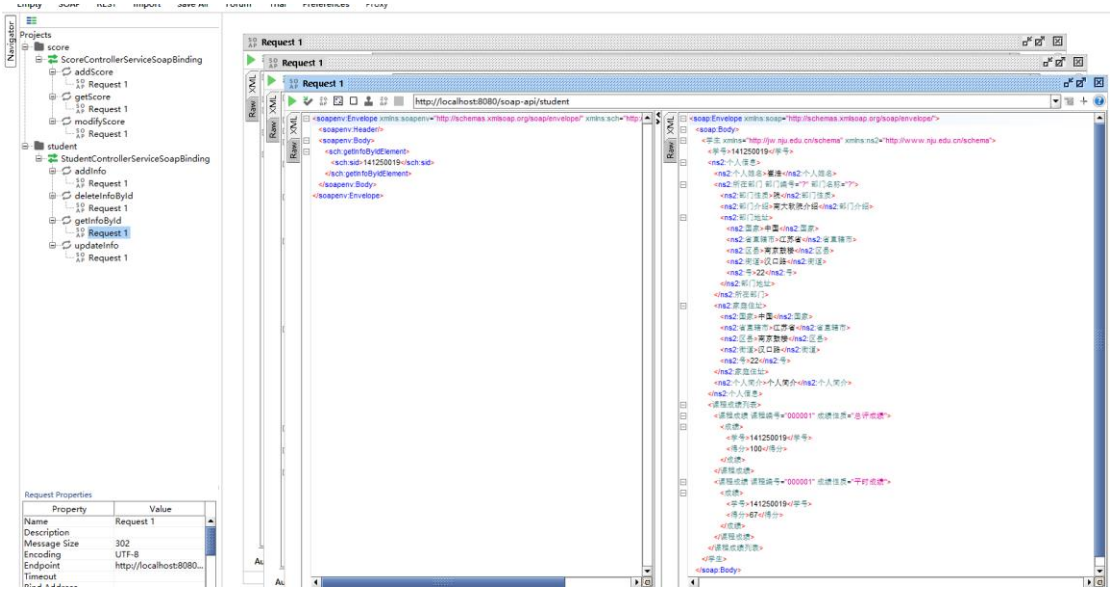

Request 1
http://localhost:8080/soap-api/auth
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http:/
<soapenv:Header/>
<soapenv:Body>
<sch:认证请求>
<sch:邮箱>141250019@smail.nju.edu.cn</sch:邮箱>
<sch:密码>123456</sch:密码>
</sch:认证请求>
</soapenv:Body>
</soapenv:Envelope>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<soap:Fault>
<faultcode>soap:Server</faultcode>
<faultstring>Error password</faultstring>
<detail>
<AuthFault xmlns="http://jw.nju.edu.cn/schema">
<Reason>Error password</Reason>
<Id>141250019@smail.nju.edu.cn</Id>
</AuthFault>
</detail>
</soap:Fault>
</soap:Body>
</soap:Envelope>

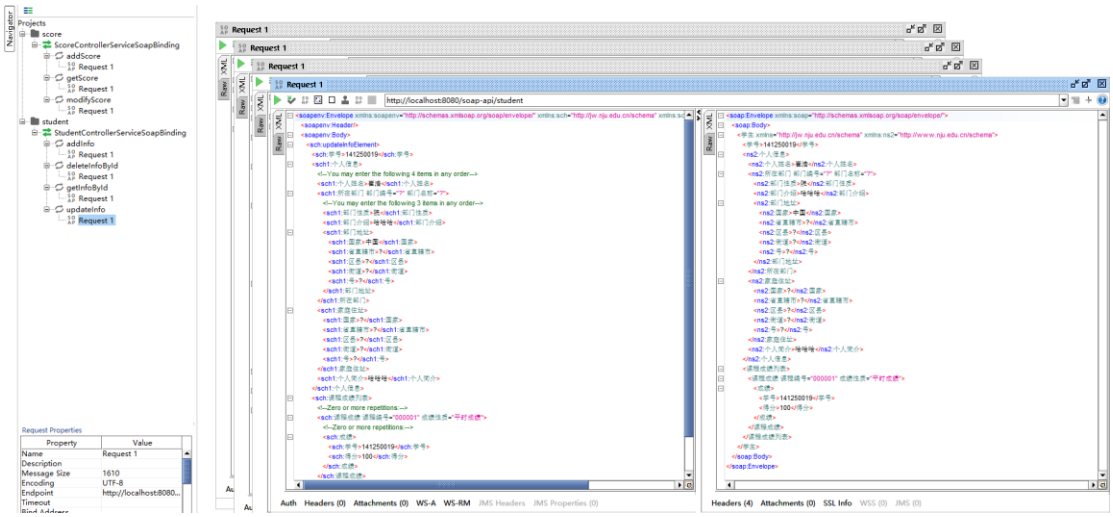

Navigator
Projects
score
ScoreControllerServiceSoapBinding
addScore
Request 1
getScore
Request 1
modifyScore
Request 1
student
StudentControllerServiceSoapBinding
addInfo
Request 1
deleteInfoById
Request 1
getInfoById
Request 1
updateInfo
Request 1
Request Properties
Property
Value
Name
Request 1
Description
Message Size
623
Encoding
UTF-8
Request 1
http://localhost:8080/soap-api/score
Auth
Headers (0)
Attachments (0)
WS-A
WS-RM
JMS Headers
JMS Properties (0)
Headers (4)
Attachments (0)
SSL Info
WSS (0)
JMS (0)
18 : 20


Navigator
Projects
score
ScoreControllerServiceSoapBinding
addScore
Request 1
getScore
Request 1
modifyScore
Request 1
student
StudentControllerServiceSoapBinding
addInfo
Request 1
deleteInfoById
Request 1
getInfoById
Request 1
updateInfo
Request 1
Request 1
http://localhost:8080/soap-api/score
Request Properties
Property
Value
Name
Request 1
Description
Message Size
232
Encoding
UTF-8
Endpoint
http://localhost:8080...
Timeout


Navigator
Projects
score
ScoreControllerServiceSoapBinding
addScore
Request 1
getScore
Request 1
modifyScore
Request 1
student
StudentControllerServiceSoapBinding
addInfo
Request 1
deleteInfoById
Request 1
getInfoById
Request 1
updateInfo
Request 1
Request 1
http://localhost:8080/soap-api/score
Request Properties
Property
Value
Name
Request 1
Description
Message Size
521
Encoding
UTF-8
Endpoint
http://localhost:8080...
Timeout

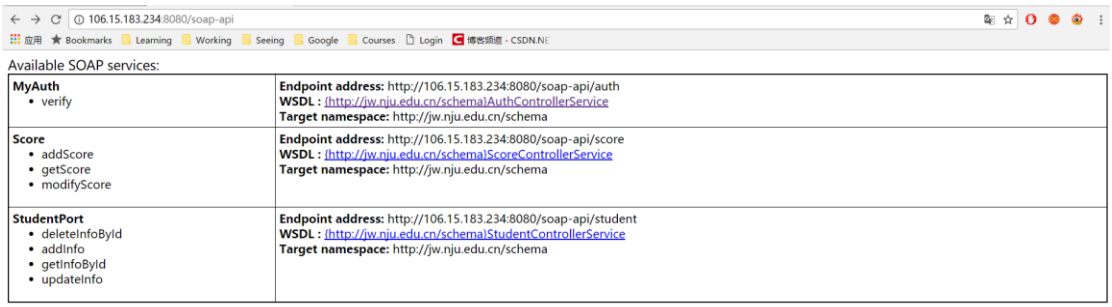

Navigator
Projects
score
ScoreControllerServiceSoapBinding
addScore
Request 1
getScore
Request 1
modifyScore
Request 1
student
StudentControllerServiceSoapBinding
addInfo
Request 1
deleteInfoById
Request 1
getInfoById
Request 1
updateInfo
Request 1
Request 1
Request 1
http://localhost:8080/soap-api/student
Request Properties
Property
Value
Name
Request 1
Description
Message Size
1794
Encoding
UTF-8
Endpoint
http://localhost:8080...
Timeout
Bind Address
Follow Redirects
true



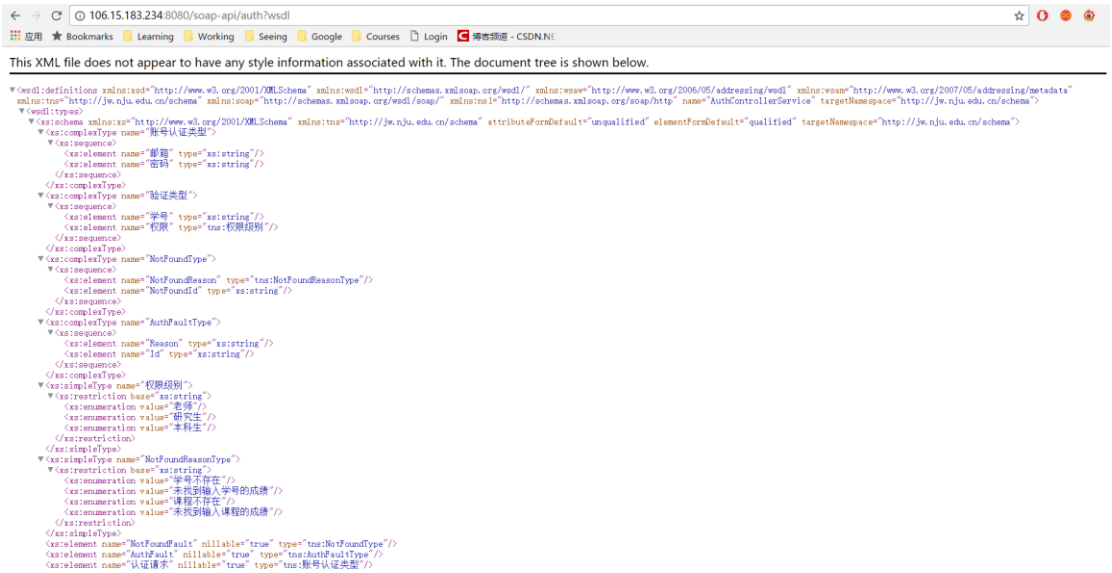


（8）发布服务

使用 mvn package -Dmaven.test.skip=true 生成 jar 包，在部署服务器上输入命令：nohup java –jar soa.jar &运行该 springboot 项目：



MyAuth • verify	**Endpoint address:** http://106.15.183.234:8080/soap-api/auth **WSDL :** {http://jw.nju.edu.cn/schema}AuthControllerService **Target namespace:** http://jw.nju.edu.cn/schema
Score • addScore • getScore • modifyScore	**Endpoint address:** http://106.15.183.234:8080/soap-api/score **WSDL :** {http://jw.nju.edu.cn/schema}ScoreControllerService **Target namespace:** http://jw.nju.edu.cn/schema
StudentPort • deleteInfoById • addInfo • getInfoById • updateInfo	**Endpoint address:** http://106.15.183.234:8080/soap-api/student **WSDL :** {http://jw.nju.edu.cn/schema}StudentControllerService **Target namespace:** http://jw.nju.edu.cn/schema



可以看到 start from java 自动生成的 wsdl 文件，其他两个服务的 wsdl 文件与原来的本地文件基本保持一致。