遗留系统封装说明

0. 封装使用说明

0.1 JDK 版本

需要确保 JDK 版本为 1.8，否则会报 Endpoint 类的 ClassNotFoundException 异常

0.2 本地调用方式

先在 Initiator 类中将服务通过 Endpoint 进行发布。

然后在本地通过创建*Service 类型的服务对象对服务进行调用，具体见 **3. 服务调用 & 测试结果**。

0.3 客户端调用方式

首先需要确保客户端类实例化之前，相应的服务在服务端已经发布，这样在客户端就能通过服务地址获取 WSDL 文档，进而结合 QName 来创建服务接口的具体服务实例，然后调用其中的服务。在测试客户端程序调用时，需要通过@Before 在每次单元测试前，将服务发布出去，然后获取到调用相应服务所需要的客户端对象，同时为了确保服务不会重复发布而导致绑定异常出现，@Before 的代码中加入了相关判断，具体见 4. 客户端代码 & 测试结果。

0.4 数据模拟方式

创建一个 Database 类，在其中通过静态类型的 Map 对象来模拟数据库中的表。

```java
package Service;

import ...

public class Database {
    //模拟personnel表，主键key是工号
    public static Map<String, PersonnelEntity> tbl_personnel;
    //模拟class表，主键key是班次代码
    public static Map<Integer, ClassEntity> tbl_class;
    //模拟order表，主键key是订单号
    public static Map<String, OrderEntity> tbl_order;
    //模拟material表，主键key是物料编码
    public static Map<String, MaterialEntity> tbl_material;
    //模拟resource表中的生产线资源信息，主键key是生产线资源id
    public static Map<String, LineEntity> tbl_lineResources;
    //模拟resource表，主键key是资源代码
    public static Map<String, ResourceEntity> tbl_resource;
    //模拟product表，主键key是物料编码
    public static Map<String, BOMEntity> tbl_product;
}
```

0.5 Xslx 表格的读取

读取过程中为了保证可扩展性，将数据为空，但表头不为空的内容作为空字符串读入，从而保证在未来 xslx 表格发生变更时，能够更容易的修改代码并进行扩展。

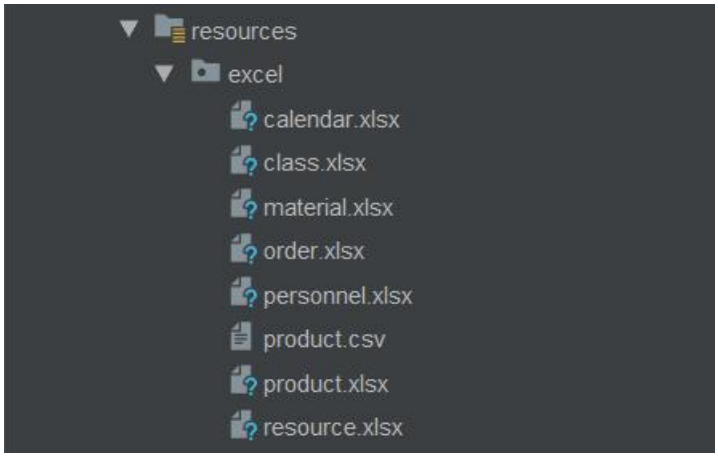
详见 util.ExcelReadUtil.java。

0.6 编译说明

在读取 Xlsx 表格文件的过程中，使用了相对路径，而该路径的获取是针对编译产生的 class 文件进行的，所以如果运行过程中出现因为读取文件导致的 NullPointerException 异常，可以将 target 文件夹删除，并重新编译运行。

0.7 数据表拆分

将原始多表 xlsx 拆分成多个 xlsx 并放在 resources.excel 文件夹下，从而能够更好的和 Database 类中模拟的数据库表 Map 对象进行映射。



0.8 开发方式

开发采用 Start-from-Java 的开发方式。通过先行开发 Java Web Service，通过@WebService 注释来让 JWS 来自动生成相应的 WSDL 文档。

1. 数据模型

1.1 员工信息

```java
package Service.model;

/**
 * 员工信息
 */
public class PersonnelEntity {
    //工号
    private String id;
    //姓名
    private String name;
    //职位
    private String position;
    //所属组
    private String groupId;
```

1.2 班次信息

```java
package Service.model;

/**
 * 班次信息
 */
public class ClassEntity {
    //班次代码 (0:全天, 1: 早班, 2: 晚班, 3: 休息)
    private int classCode;
    //班次名称
    private ClassName className;
    //工作时间段
    private String workingHours;
```

1.2.1 班次名称

```java
/**
 * 班次名称
 */
enum ClassName {
    WHOLE,      //全天
    DAY,        //早班
    NIGHT,      //晚班
    REST        //休息
}
```

1.3 生产线信息

```
package Service.model;

public class LineEntity {
    //生产线资源id
    private String id;
    //资源名称
    private String resourceName;
    //资源类型
    private String resourceType;
    //资源量
    private int count;
```

1.4 人力资源班组信息

```
package Service.model;

/**
 *人力资源（班组）信息
 */
public class ResourceEntity {
    //项目
    private String project;
    //资源代码
    private String resourceId;
    //资源名称
    private String resourceName;
    //所属资源
    private String resourceBl;
    //资源类型
    private String resourceType;
```

1.5 人力资源排班信息

```
package Service.model;

import ...

public class CalendarEntity {
    private String resourceCode;
    private List<WorkDay> workDays;
    private int classCode;
    private String remarks;
```

1.5.1 工作日

```
enum WorkDay {
    SUN,    //星期日
    MON,    //星期一
    TUE,    //星期二
    WED,    //星期三
    THU,    //星期四
    FRI,    //星期五
    SAT     //星期六
}
```

1.6 物品信息

```
package Service.model;

/**
 * 物品信息
 */
public class MaterialEntity {
    //物料编码
    private String id;
    //物料描述
    private String description;
    //物品属性
    private MaterialAttr materialAttr;
    //计量单位
    private Measurement measurement;
    //备料方式
    private Preparation preparation;
```

1.6.1 物品属性

```
/**
 * 物品属性
 */
enum MaterialAttr{
    FINISHED,       //成品
    SEMI_FINISHED,  //半成品
    RAW             //原材料
}
```

1.6.2 计量单位

```
/**
 * 计量单位
 */
enum Measurement{
    PCS,
    KG
}
```

1.6.3 准备方式

```
/**
 * 备料方式
 */
enum Preparation{
    MAKE, //自制
    BUY   //外购
}
```

1.7 BOM 信息

```java
package Service.model;

import java.util.List;

public class BOMEntity {
    private String id;
    //物品代码
    private List<String> materials;
    //物品对应的数量
    private List<Double> materialCount;
    //主资源
    private List<String>  mainResource;
    //副资源
    private List<String>  lineResource;
    //换线时间
    private String changeTime;
    //标准产能
    private String standardOutput;
    //产品规定生产人员
    private int workerCount;
```

1.8 订单信息

```java
package Service.model;

import java.util.Date;

/**
 * 订单信息
 */
public class OrderEntity {
    //订单号
    private String id;
    //物料号码
    private String materialId;
    //订单数量
    private Long number;
    //交期
    private Date ddl;
```

2. 遗留系统服务接口 & WSDL

2.1 人事系统

2.1.1 服务接口

```java
package Service;

import ...

/**
 * 人事系统
 */
@WebService(
        serviceName = "PersonnelService",
        name="PersonnelServiceSoap"
)
public interface PersonnelService {

    //获取员工信息
    PersonnelEntity getStaffInfoById(String id);

    //进行身份认证
    String idAuthentication(String id);
}
```

2.1.2 WSDL 文档

```xml
<!--
    Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e.
-->
<!--
    Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e.
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
 xmlns:wspl_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:tns="http://impl.Service/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://impl.Service/" name="PersonnelService">
  <import namespace="http://Service/" location="http://localhost:8080/PersonnelService?wsdl=1"/>
  <binding xmlns:ns1="http://Service/" name="PersonnelServiceSoapPortBinding" type="ns1:PersonnelServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="getStaffInfoById">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="idAuthentication">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="PersonnelService">
    <port name="PersonnelServiceSoapPort" binding="tns:PersonnelServiceSoapPortBinding">
      <soap:address location="http://localhost:8080/PersonnelService"/>
    </port>
  </service>
</definitions>
```

2.2 考勤系统

2.2.1 服务接口

```java
package Service;

import ...

/**
 * 考勤系统
 */
@WebService(
        serviceName = "AttendanceService",
        name = "AttendanceServiceSoap"
)
public interface AttendanceService {
    //获取班次信息
    List<ClassEntity> getClassInfo();

    //获取人力资源（班组）排班信息
    List<CalendarEntity> getCalendarInfo();
}
```

2.2.2 WSDL 文档

2.3 ERP 系统

2.3.1 服务接口

```java
@WebService(
        serviceName = "ERPService",
        name="ERPServiceSoap"
)
public interface ERPService {
    //获取人力资源（班组）信息
    List<ResourceEntity> getResourceTeamInfo();

    //获取物品信息
    MaterialEntity getMaterialInfoById(String id);

    //获取生产线/设备资源信息
    LineEntity getLineResourceById(String id);

    //获取生产线/设备资源信息
    List<LineEntity> getAllLineResources();

    //获取BOM信息
    BOMEntity getBOMById(String id);

    //获取BOM信息
    List<BOMEntity> getAllBOMs();
}
```

2.3.2 WSDL 文档

2.4 订单管理系统

2.4.1 服务接口

```java
package Service;

import ...

/**
 * 订单管理系统
 */
@WebService(
        serviceName = "OrderService",
        name="OrderServiceSoap"
)
public interface OrderService {

    //获取订单信息
    OrderEntity getOrderInfoById(String id);
}
```

2.4.2 WSDL 文档

3. 服务调用 & 测试结果

3.1 人事系统

```java
package Service;

import ...

public class PersonnelServiceTest {

    PersonnelServiceImpl personnelService = new PersonnelServiceImpl();

    @Test
    public void getStaffInfoById() {
        PersonnelEntity staffInfo = personnelService.getStaffInfoById("1");
        assertEquals( expected: "1", staffInfo.getId());
        assertEquals( expected: "童玲", staffInfo.getName());
        assertEquals( expected: "组长", staffInfo.getPosition());
        assertEquals( expected: "5", staffInfo.getGroupId());
    }

    @Test
    public void idAuthentication() {
        String actualPosition = personnelService.idAuthentication("1");
        assertEquals( expected: "组长", actualPosition);
    }
}
```

3.2 考勤系统

```java
package Service;

import ...

public class AttendanceServiceTest {

    AttendanceServiceImpl attendanceService = new AttendanceServiceImpl();

    @Test
    public void getClassInfo() {
        List<ClassEntity> classInfo = attendanceService.getClassInfo();
        assertEquals( expected: 4, classInfo.size());
    }

    @Test
    public void getCalendarInfo() {
        List<CalendarEntity> calendarInfo = attendanceService.getCalendarInfo();
        assertEquals( expected: 65, calendarInfo.size());
    }
}
```

3.3 ERP 系统

```java
package Service;

import ...

public class ERPServiceTest {

    ERPServiceImpl erpService = new ERPServiceImpl();

    @Test
    public void getResourceTeamInfoTest() {
        List<ResourceEntity> resourceInfo = erpService.getResourceTeamInfo();
        assertEquals( expected: 65, resourceInfo.size());
    }

    @Test
    public void getMaterialInfoById() {
        MaterialEntity material = erpService.getMaterialInfoById("3000608");
        assertEquals( expected: "3000608", material.getId());
        assertEquals( expected: "UT 4-TWIN HV", material.getDescription());
        assertEquals(material.computeAttr( str: "成品"), material.getMaterialAttr());
        assertEquals(material.computeMeasurement( str: "PCS"), material.getMeasurement());
        assertEquals(material.computePrep( str: "自制"), material.getPreparation());
    }
}
```

```java
    @Test
    public void getBOMById(){
        BOMEntity bom=erpService.getBOMById("3036466");
        assertEquals( expected: "3036466", bom.getId());
        assertEquals( expected: "9761340", bom.getMaterials().get(0));
        assertEquals( expected: 5, bom.getMaterials().size());
        assertEquals( expected: 5, bom.getMaterialCount().size());
        assertEquals( expected: 9, bom.getMainResource().size());
        assertEquals( expected: 3, bom.getLineResource().size());
        assertEquals( expected: "0.1H", bom.getChangeTime());
        assertEquals( expected: "178个/小时", bom.getStandardOutput());
        assertEquals( expected: 4, bom.getWorkerCount());


    }
```

3.4 订单管理系统

```java
package Service;

import ...

public class OrderServiceTest {

    OrderServiceImpl orderService = new OrderServiceImpl();

    @Test
    public void getOrderInfoById() {
        OrderEntity order = orderService.getOrderInfoById("418458");
        assertEquals( expected: "418458", order.getId());
        assertEquals( expected: "3040339", order.getMaterialId());
        assertEquals(Long.valueOf(100), order.getNumber());
        assertEquals( expected: "Wed Nov 28 14:00:00 CST 2018", order.getDdl().toString());
    }
}
```

4. 客户端代码 & 测试结果

4.1 人事系统

4.1.1 客户端程序

```
import javax.xml.namespace.QName;
import javax.xml.ws.Service;
import java.net.MalformedURLException;
import java.net.URL;

public class PersonnelClient {
    private PersonnelService personnelService;
    public PersonnelClient(){
        URL personnelWsdlDocumentLocation = null;
        try {
            personnelWsdlDocumentLocation = new URL( spec: "http://localhost:8080/PersonnelService");
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        QName sNamePersonnel = new QName( namespaceURI: "http://impl.Service/", localPart: "PersonnelService");
        Service servicePersonnel = Service.create(personnelWsdlDocumentLocation, sNamePersonnel);
        personnelService = servicePersonnel.getPort(PersonnelService.class);
    }
    public PersonnelEntity getStaffInfoById(String id){
        return personnelService.getStaffInfoById(id);
    }
    public String idAuthentication(String id){
        return personnelService.idAuthentication(id);
    }
}
```

4.1.2 客户端测试

```
import static org.junit.Assert.assertEquals;

public class PersonnelClientTest {
    private static PersonnelClient client;
    @Before
    public void setup(){
        if(client!=null) return;
        String personnelAddr = "http://localhost:8080/PersonnelService";
        Endpoint.publish(personnelAddr, new PersonnelServiceImpl());
        client = new PersonnelClient();
    }
    @Test
    public void getStaffInfoById() {
        PersonnelEntity staffInfo = client.getStaffInfoById("1");
        assertEquals( expected: "1", staffInfo.getId());
        assertEquals( expected: "童玲", staffInfo.getName());
        assertEquals( expected: "组长", staffInfo.getPosition());
        assertEquals( expected: "5", staffInfo.getGroupId());
    }
    @Test
    public void idAuthentication() {
        String actualPosition = client.idAuthentication("1");
        assertEquals( expected: "组长", actualPosition);
    }
}
```

4.2 考勤系统

4.2.1 客户端程序

```
import javax.xml.ws.Service;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.List;

public class AttendanceClient {
    private AttendanceService attendanceService;
    public AttendanceClient(){
        URL attendanceWsdlDocumentLocation = null;
        try {
            attendanceWsdlDocumentLocation = new URL( spec: "http://localhost:8080/AttendanceService");
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        QName sNameAttendance = new QName( namespaceURI: "http://impl.Service/",  localPart: "AttendanceService");
        Service serviceAttendance = Service.create(attendanceWsdlDocumentLocation, sNameAttendance);
        attendanceService = serviceAttendance.getPort(AttendanceService.class);
    }
    public List<ClassEntity> getClassInfo(){
        return attendanceService.getClassInfo();
    }
    public List<CalendarEntity> getCalendarInfo(){
        return attendanceService.getCalendarInfo();
    }
}
```

4.2.2 客户端测试

```
import javax.xml.ws.Endpoint;
import java.util.List;

import static org.junit.Assert.assertEquals;

public class AttendanceClientTest {
    private static AttendanceClient client;
    @Before
    public void setup(){
        if(client!=null) return;
        String attendanceAddr = "http://localhost:8080/AttendanceService";
        Endpoint.publish(attendanceAddr, new AttendanceServiceImpl());
        client = new AttendanceClient();
    }
    @Test
    public void getClassInfo() {
        List<ClassEntity> classInfo = client.getClassInfo();
        assertEquals( expected: 4, classInfo.size());
    }
    @Test
    public void getCalendarInfo() {
        List<CalendarEntity> calendarInfo = client.getCalendarInfo();
        assertEquals( expected: 65, calendarInfo.size());
    }
}
```

4.3 ERP 系统

4.3.1 客户端程序

```java
public class ERPClient {
    private ERPService erpService;
    public ERPClient(){
        URL erpWsdlDocumentLocation = null;
        try {
            erpWsdlDocumentLocation = new URL( spec: "http://localhost:8080/ERPService");
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        QName sNameERP = new QName( namespaceURI: "http://impl.Service/", localPart: "ERPService");
        Service serviceERP = Service.create(erpWsdlDocumentLocation, sNameERP);
        erpService = serviceERP.getPort(ERPService.class);
    }
    public List<ResourceEntity> getResourceTeamInfo(){
        return erpService.getResourceTeamInfo();
    }
    public MaterialEntity getMaterialInfoById(String id){
        return erpService.getMaterialInfoById(id);
    }
    public LineEntity getLineResourceById(String id){
        return erpService.getLineResourceById(id);
    }
    public List<LineEntity> getAllLineResources() { return erpService.getAllLineResources(); }
    public BOMEntity getBOMById(String id) { return erpService.getBOMById(id); }
    public List<BOMEntity> getAllBOMs() { return erpService.getAllBOMs(); }
```

4.3.2 客户端测试

```java
public class ERPClientTest {
    private static ERPClient client;
    @Before
    public void setup(){
        if(client!=null) return;
        String erpAddr = "http://localhost:8080/ERPService";
        Endpoint.publish(erpAddr, new ERPServiceImpl());
        client = new ERPClient();
    }
    @Test
    public void getResourceTeamInfoTest() {
        List<ResourceEntity> resourceInfo = client.getResourceTeamInfo();
        assertEquals( expected: 65, resourceInfo.size());
    }
    @Test
    public void getMaterialInfoById() {
        MaterialEntity material = client.getMaterialInfoById("3000608");
        assertEquals( expected: "3000608", material.getId());
        assertEquals( expected: "UT 4-TWIN HV", material.getDescription());
        assertEquals(material.computeAttr( str: "成品"), material.getMaterialAttr());
        assertEquals(material.computeMeasurement( str: "PCS"), material.getMeasurement());
        assertEquals(material.computePrep( str: "自制"), material.getPreparation());
    }
```

```java
    @Test
    public void getBOMById(){
        BOMEntity bom=client.getBOMById("3036466");
        assertEquals( expected: "3036466", bom.getId());
        assertEquals( expected: "9761340", bom.getMaterials().get(0));
        assertEquals( expected: 5, bom.getMaterials().size());
        assertEquals( expected: 5, bom.getMaterialCount().size());
        assertEquals( expected: 9, bom.getMainResource().size());
        assertEquals( expected: 3, bom.getLineResource().size());
        assertEquals( expected: "0.1H", bom.getChangeTime());
        assertEquals( expected: "178个/小时", bom.getStandardOutput());
        assertEquals( expected: 4, bom.getWorkerCount());
    }
```

4.4 订单管理系统

4.4.1 客户端程序

```
import Service.OrderService;
import Service.model.OrderEntity;

import javax.xml.namespace.QName;
import javax.xml.ws.Service;
import java.net.MalformedURLException;
import java.net.URL;

public class OrderClient {
    private OrderService orderService;
    public OrderClient(){
        URL orderWsdlDocumentLocation = null;
        try {
            orderWsdlDocumentLocation = new URL( spec: "http://localhost:8080/OrderService");
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        QName sNameOrder = new QName( namespaceURI: "http://impl.Service/", localPart: "OrderService");
        Service serviceOrder = Service.create(orderWsdlDocumentLocation, sNameOrder);
        orderService = serviceOrder.getPort(OrderService.class);
    }
    public OrderEntity getOrderInfoById(String id){
        return orderService.getOrderInfoById(id);
    }
}
```

4.4.2 客户端测试

```
8
9       import javax.xml.ws.Endpoint;
10
11      import static org.junit.Assert.assertEquals;
12
13      public class OrderClientTest {
14          private static OrderClient client;
15          @Before
16          public void setup(){
17              if(client!=null) return;
18              String orderAddr = "http://localhost:8080/OrderService";
19              Endpoint.publish(orderAddr, new OrderServiceImpl());
20              client = new OrderClient();
21          }
22          @Test
23          public void getOrderInfoById() {
24              OrderEntity order = client.getOrderInfoById("418458");
25              assertEquals( expected: "418458", order.getId());
26              assertEquals( expected: "3040339", order.getMaterialId());
27              assertEquals(Long.valueOf(100), order.getNumber());
28              assertEquals( expected: "Wed Nov 28 14:00:00 CST 2018", order.getDdl().toString());
29          }
30      }
```