

# 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 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 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。

```
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;
}
```

### 0.4 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。

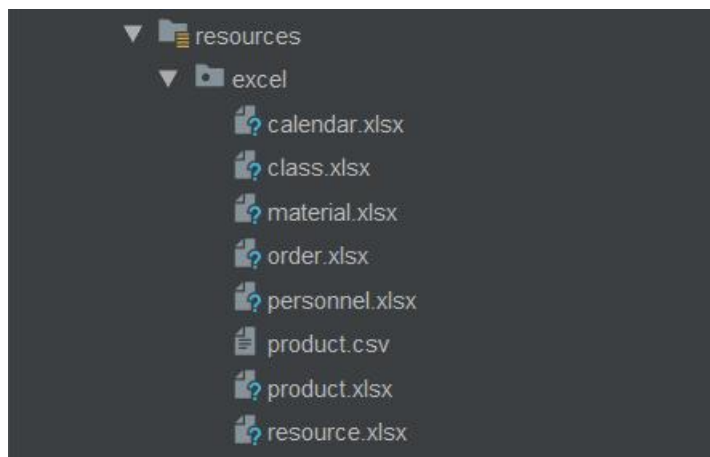
详见 [util.ExcelReadUtil.java](#)。

## 0.5 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。

## 0.6 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。



## 0.7 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

```
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

```
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;
    //工作时段
    private String workingHours;
}
```

### 1.2.1 班次名称

```
/**
 * 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

```
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 int changeTime;
    //标准产能
    private int standardOutput;
```

## 1.8 订单信息

```
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

```

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

```

<!--
    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:wsp1_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" />
    <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

```

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

```
▼<!--
  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:wsp1_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="AttendanceService">
  <import namespace="http://Service/" location="http://localhost:8080/AttendanceService?wsdl=1"/>
  ▼<binding xmlns:ns1="http://Service/" name="AttendanceServiceSoapPortBinding" type="tns:AttendanceServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    ▼<operation name="getClassInfo">
      <soap:operation soapAction=""/>
      ▼<input>
        <soap:body use="literal"/>
      </input>
      ▼<output>
        <soap:body use="literal"/>
      </output>
    </operation>
    ▼<operation name="getCalendarInfo">
      <soap:operation soapAction=""/>
      ▼<input>
        <soap:body use="literal"/>
      </input>
      ▼<output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  ▼<service name="AttendanceService">
    ▼<port name="AttendanceServiceSoapPort" binding="tns:AttendanceServiceSoapPortBinding">
      <soap:address location="http://localhost:8080/AttendanceService"/>
    </port>
  </service>
</definitions>
```

## 2.3 ERP 系统

### 2.3.1 服务接口

```
@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



[illegible]

## 2.4 订单管理系统

### 2.4.1 服务接口

```
package Service;

import ...

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

public interface OrderService {

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

### 2.4.2 WSDL 文档

```

[...
  <!--
Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#f6196f2b90e460005a4c2f4c30a065b245e51a.
-->
<!--
Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#f6196f2b90e460005a4c2f4c30a065b245e51a.
-->
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<definitions xmlns:wsa="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-security-utility-1.0.xsd" xmlns:wsdp="http://www.w3.org/ns/ws-policy" xmlns:wsp_1_2="http://schemas.xmlsoap.org/wss/2004/09/policy"
xmlns:wsse="http://www.w3.org/2003/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/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="OrderService">
  <import namespace="http://Service/" location="http://localhost:8080/OrderService.wsdl" />
  <binding name="tns:OrderService" type="tns:OrderServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
    <operation name="getOrderInfoById">
      <soap:operation soapAction="" />
      <input>
        <soap:body use="literal" />
      </input>
      <output>
        <soap:body use="literal" />
      </output>
    </operation>
  </binding>
  <service name="OrderService">
    <port name="OrderServiceSoapPort" binding="tns:OrderServiceSoapPortbinding">
      <soap:address location="http://localhost:8080/OrderService/" />
    </port>
  </service>
</definitions>

```

## 3. 服务调用 & 测试结果

### 3.1 人事系统

```
1 package Service;
2
3 import ...
4
5
6
7
8
9 public class PersonnelServiceTest {
10
11     PersonnelServiceImpl personnelService = new PersonnelServiceImpl();
12
13     @Test
14     public void getStaffInfoById() {
15         PersonnelEntity staffInfo = personnelService.getStaffInfoById("1");
16         assertEquals( expected: "1", staffInfo.getId());
17         assertEquals( expected: "童玲", staffInfo.getName());
18         assertEquals( expected: "组长", staffInfo.getPosition());
19         assertEquals( expected: "5", staffInfo.getGroupId());
20     }
21
22     @Test
23     public void idAuthentication() {
24         String actualPosition = personnelService.idAuthentication("1");
25         assertEquals( expected: "组长", actualPosition);
26     }
27 }
28
```

### 3.2 考勤系统

```
1 package Service;
2
3 import ...
4
5
6
7
8
9
10
11
12 public class AttendanceServiceTest {
13
14     AttendanceServiceImpl attendanceService = new AttendanceServiceImpl();
15
16     @Test
17     public void getClassInfo() {
18         List<ClassEntity> classInfo = attendanceService.getClassInfo();
19         assertEquals( expected: 4, classInfo.size());
20     }
21
22     @Test
23     public void getCalendarInfo() {
24         List<CalendarEntity> calendarInfo = attendanceService.getCalendarInfo();
25         assertEquals( expected: 65, calendarInfo.size());
26     }
27 }
28
```

### 3.3 ERP 系统

```
1 package Service;
2
3 import ...
4
5
6
7
8
9
10
11
12
13 public class ERPServiceTest {
14
15     ERPServiceImpl erpService = new ERPServiceImpl();
16
17     @Test
18     public void getResourceTeamInfoTest() {
19         List<ResourceEntity> resourceInfo = erpService.getResourceTeamInfo();
20         assertEquals( expected: 65, resourceInfo.size());
21     }
22
23     @Test
24     public void getMaterialInfoById() {
25         MaterialEntity material = erpService.getMaterialInfoById("3000608");
26         assertEquals( expected: "3000608", material.getId());
27         assertEquals( expected: "UT 4-TWIN HV", material.getDescription());
28         assertEquals(material.computeAttr( str: "成品"), material.getMaterialAttr());
29         assertEquals(material.computeMeasurement( str: "PCS"), material.getMeasurement());
30         assertEquals(material.computePrep( str: "自制"), material.getPreparation());
31     }
32 }
```

### 3.4 订单管理系统

```
1 package Service;
2
3 import ...
4
5
6
7
8
9 public class OrderServiceTest {
10
11     OrderServiceImpl orderService = new OrderServiceImpl();
12
13     @Test
14     public void getOrderInfoById() {
15         OrderEntity order = orderService.getOrderInfoById("418458");
16         assertEquals( expected: "418458", order.getId());
17         assertEquals( expected: "3040339", order.getMaterialId());
18         assertEquals(Long.valueOf(100), order.getNumber());
19         assertEquals( expected: "Wed Nov 28 14:00:00 CST 2018", order.getDdl().toString());
20     }
21 }
22
```