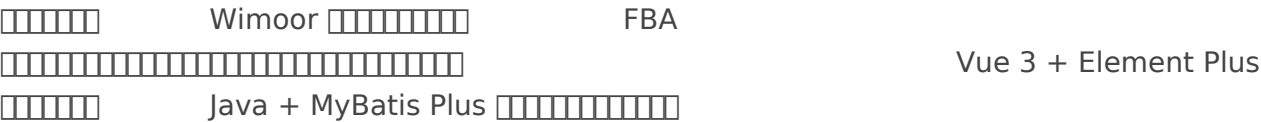


1.



1.1

flowchart TD

A[

B --> C[

C --> D[API]

D --> E[

C --> F[

E --> D

1.2

		
		Vue 3 + Element Plus
		Vue 3 + Element Plus
		Vue 3 + Element Plus
		Java + Spring Boot
		RESTful API
		MySQL

2. 数据表

2.1 数据表

2.1.1 数据表 (index.vue)

- 数据表
 - 数据表
 - 数据表
 - 数据表
- 数据表
 - 数据表
 - activeName
 - orderStatus
 - 数据表
 - getData
 - tableRef.value.getshipmentData(obj)
 - 数据表
 - handleClick

```
// 数据表
function handleClick(){
  if(activeName.value=="0"){
    obj.orderStatus = ""
    obj.hasexceptionnum=null;
  }else if(activeName.value=="1"){
    obj.orderStatus = 7
    obj.hasexceptionnum=null;
  }else if(activeName.value=="2"){
    obj.orderStatus = 5
    obj.hasexceptionnum=null;
  }else if(activeName.value=="3"){
    obj.orderStatus = 55
    obj.hasexceptionnum=null;
  }else if(activeName.value=="4"){
    obj.orderStatus = 6
    obj.hasexceptionnum=null;
  }else if(activeName.value=="5"){
    obj.orderStatus = 0
    obj.hasexceptionnum=null;
  }else if(activeName.value=="6"){
    obj.hasexceptionnum='ok';
  }
}
```

```

        obj.orderStatus = 6
    }
    tableRef.value.getshipmentData(obj);
    headerRef.value.statusChange(obj);
}

```

2.1.2 数据接口 (table.vue)

- 数据接口
 - 获取数据
 - 获取数据
 - 获取数据
- 数据接口
 - 获取数据
 - 获取数据
 - 获取数据
 - 获取数据

API

```

// 数据接口
function loadtableData(params){
    params.groupid = paramashead.value.store;
    params.marketplaceid =paramashead.value.country;
    params.warehouseid =paramashead.value.warehouse;
    params.fbacenter=paramashead.value.fbacenter;
    if(paramashead.value.start!==undefined){
        params.fromdate = paramashead.value.start;
        params.enddate =paramashead.value.end;
    }else{
        const end = new Date()
        const start = new Date()
        start.setTime(start.getTime() - 3600 * 1000 * 24 * 7)
        params.fromdate =dateFormat(start);
        params.enddate =dateFormat(end);
    }
    params.auditstatus = paramashead.value.orderStatus;
    if(paramashead.value.seachtype!==undefined){
        params.searchtype =paramashead.value.seachtype;
    }else{
        params.searchtype ="sku";
    }
    params.search = paramashead.value.searchwords;
}

```

```

params.company =paramashead.value.company;
params.channel= paramashead.value.channel;
params.transtype =paramashead.value.transtype;
params.checkdown=paramashead.value.checkdown;
params.checkinv=paramashead.value.checkinv;
params.areCasesRequired=paramashead.value.areCasesRequired;
params.hasexceptionnum=paramashead.value.hasexceptionnum;
params.hasreferenceid=paramashead.value.hasreferenceid;
var tagtypes=["primary","success","info","warning","danger"];
shipmenthandlingApi.getshipList(params).then((res)=>{
    var indexv=0;
    res.data.records.forEach(itemv=>{
        if(oldcheckinv[itemv.checkinv]==undefined){
            indexv=(indexv+1)%5;
            itemv.tagtype=tagtypes[indexv];
            oldcheckinv[itemv.checkinv]=indexv;
        }else{
            itemv.tagtype=tagtypes[oldcheckinv[itemv.checkinv]];
        }
    });
    tableData.records = res.data.records;
    tableData.total =res.data.total;
})
}

```

2.1.3 数据展示 (transinfo.vue)

- 数据展示
 - 数据表头 ZH ZM
 - 数据表体
 - 数据表尾
 - 数据表脚
- 数据加载
 - 数据加载函数 loadTransDetialInfo
 - ZH数据加载 loadZhApiDetail
 - ZM数据加载 loadZmApiDetail

```

// 数据加载函数
function loadTransDetialInfo(companyid,shipmentid,ordernum){
    var html="";

```

```

loading.value=true;
shipment.value="";
zmData.value="";
transportationApi.shipTransDetial({"companyid":
companyid,"shipmentid":shipmentid,"ordernum":ordernum}).then(res=>{
    loading.value=false;
    if(res && res.data.ftype=="ZH"){
        systemType.value=res.data.ftype;
        loadZhApiDetail(res.data,companyid,shipmentid);
    }
    if(res && res.data.ftype=="ZM"){
        systemType.value=res.data.ftype;
        loadZmApiDetail(res.data);
    }
})
dialogTransVisible.value=true;
}

```

2.2 接口 API

2.2.1 接口 API

- `shipmenthandlingApi.getshipList(params)` 接口
 - 接口 接口
 - 接口 接口

2.2.2 接口 API

- `transportationApi.shipTransDetial(params)` 接口
 - 接口 接口 ID 接口 ID 接口
 - 接口 接口

2.2.3 接口 API

- `shipmentApi.refreshShipmentRec(params)` 接口
 - 接口 接口 ID
 - 接口 接口
- `shipmentApi.ignoreShipment(params)` 接口
 - 接口 接口 ID
 - 接口 接口

2.3 项目搭建

- 创建项目：使用 Vue 3 的 `ref` 和 `reactive` 来管理状态
- 配置路由：使用 `props` 和 `emit` 来传递数据
- 安装依赖：使用 `Vue Router` 来管理路由

3. 项目结构

3.1 目录结构

3.1.1 项目结构 (Shipment.java)

- 项目结构
 - `src` 目录
 - `main` 目录
 - `java` 目录
 - `com` 目录
 - `example` 目录
 - `Shipment.java`

3.2 模块结构

3.2.1 模块结构

- 模块结构
 - `src` 目录
 - `main` 目录
 - `java` 目录
 - `com` 目录
 - `example` 目录
 - `Shipment.java`

3.2.2 模块结构

- 模块结构
 - `src` 目录
 - `main` 目录
 - `java` 目录
 - `com` 目录
 - `example` 目录
 - `Shipment.java`

3.3 项目 API 接口

3.3.1 船隻清單

- 地址 `/api/erp/ship/shipmenthandling/getshipList`
- 方法 GET
- 參數 船隻清單
- 返回 船隻清單

3.3.2 船隻運輸詳情

- 地址 `/api/erp/ship/transportation/shipTransDetial`
- 方法 GET
- 參數 船隻 ID 船隻 ID
- 返回 船隻運輸詳情

3.3.3 船隻清單刷新

- 地址 `/api/erp/ship/shipment/refreshShipmentRec`
- 方法 GET
- 參數 船隻 ID
- 返回 船隻清單

4. 船隻清單

4.1 船隻清單

- ZH 船隻清單
 - 船隻清單
 - 船隻清單
 - 船隻清單
- ZM 船隻清單
 - 船隻清單
 - 船隻清單

4.2 船隻清單 API 地址

4.2.1 API 地址

1. 船隻清單
2. 船隻清單 API
3. 船隻清單

- 4.
- 5.
- 6.

4.2.2

- ZH

```
{
  "service_name": "测试",
  "shipment_id": "123456",
  "client_reference": "REF789",
  "charge_amount": "100.00",
  "status": "成功",
  "parcel_count": "2",
  "sumqty": "10",
  "sumprice": "500.00",
  "to_address": {
    "name": "测试",
    "address_1": "测试1",
    "city_1": "测试",
    "postcode": "100000",
    "country": "CN"
  },
  "from_address": {
    "name": "测试",
    "address_1": "测试1",
    "city_1": "测试",
    "postcode": "200000",
    "country": "CN"
  },
  "traces": [
    {
      "time": 1674567890,
      "info": "测试"
    },
    {
      "time": 1674567900,
      "info": "测试"
    }
  ]
}
```

```
],
"parcels": [
  {
    "number": "P123",
    "ext_number": "EP123",
    "client_weight": "5.0",
    "client_length": "30",
    "client_width": "20",
    "client_height": "10",
    "carrier_code": "SF",
    "tracking_number": "SF1234567890",
    "chargeable_weight": "5.0",
    "actual_weight": "4.8",
    "chargeable_length": "30",
    "chargeable_width": "20",
    "chargeable_height": "10",
    "picking_time": 1674567890,
    "declarations": [
      {
        "sku": "SKU123",
        "name_zh": "产品名称",
        "name_en": "Product Name",
        "unit_value": "50.00",
        "qty": "2",
        "material": "材料",
        "usage": "用途",
        "brand": "品牌",
        "size": "尺寸",
        "sale_url": "https://example.com",
        "weight": "1.0",
        "hscode": "12345678"
      }
    ]
  }
]
}
```

• ZM

```
{
  "jobno": "ZM1234567890",
  "podInfoDTOList": [
    {
      "scanTime": "2026-01-23 10:00:00",
      "remark": "□□□□□"
    },
    {
      "scanTime": "2026-01-23 14:00:00",
      "remark": "□□□□□□"
    }
  ]
}
```

5. □□□□□

5.1 □□□□□□□

flowchart TD

```
A[□□□□□□□] --> B[□□□□□]
B --> C[□□□□□API]
C --> D[□□□□□]
D --> E[□□□□□]
E --> F[□□□□□□□]
F --> G[□□□□□]
```

5.2 □□□□□□□

flowchart TD

```
A[□□□□□□□□□] --> B[□□□□□]
B --> C[□□□□□API]
C --> D[□□□□□]
D --> E[□□□□□API]
E --> F[□□□□□]
F --> G[□□□□□]
```

G --> H[]

H --> I[]

5.3

flowchart TD

A[] --> B[]

B --> C{ }

C -->| | D[API]

C -->| | E[API]

D --> F[]

E --> F

F --> G[]

G --> H[]

H --> I[]

6.

6.1

- **Vue 3 Composition API** Vue 3 Composition API
- **Element Plus** Element Plus
-
-
-
-

6.2

- **Spring Boot** Spring Boot
- **MyBatis Plus** MyBatis Plus
- **RESTful API** RESTful API
-
-
-

6.3 目录

- 目录 ZH ZM
- 目录
- 目录 API
- 目录
- 目录

7. 目录

7.1 目录

1. 目录
 - 目录
 - 目录 API
 - 目录
2. 目录
 - 目录
 - 目录 Pinia Vuex
 - 目录
3. 目录
 - 目录
 - 目录
 - 目录

7.2 目录

1. 目录
 - 目录
 - 目录
 - 目录
2. 目录
 - 目录
 - 目录 API
 - 目录
3. 目录
 - 目录
 - 目录 API
 - 目录

7.3 数据库

1. 数据库
 - 数据库
 - 数据库
2. **API** 接口
 - 接口 API 数据库 API
 - 数据库
3. 数据库
 - 数据库 NoSQL 数据库
 - 数据库

8. 数据库

8.1 数据库

1. 数据库
 - 数据库
 - 数据库
2. 数据库
 - 数据库 FedEx UPS DHL
 - 数据库 API 数据库
3. 数据库
 - 数据库
 - 数据库
4. 数据库
 - 数据库
 - 数据库
5. 数据库
 - 数据库
 - 数据库

8.2 数据库

1. 数据库 **WebSocket**
 - 数据库
 - 数据库
2. 数据库 **AI** 接口
 - 数据库 AI 数据库

