

□□□□	□□□□	□□□□
ShipInboundPlanPlacementV2Controller	wimoor666\wimoor666\wimoor-amazon\amazon-boot\src\main\java\com\wimoor\amazon\inboundV2\controller\ShipInboundPlanPlacementV2Controller.java	□□□□□□□□□□□□□□□□ □
ShipFormController	wimoor666\wimoor666\wimoor-amazon\amazon-boot\src\main\java\com\wimoor\amazon\inbound\controller\ShipFormController.java	□□□□□□□□□□□□□□□□ □□

3.2 ?????

□□□□	□□□□	□□□□
IShipInboundShipmentService	□□□□□□□□□□□□□□□□	
ShipInboundShipmentServiceImpl	□□□□□□□□□□□□□□□□	
IFulfillmentInboundService	□□□□□□□□□□□□□□□□ □□□□	
FulfillmentInboundServiceImpl	□□□□□□□□□□□□□□□□ □□□□	

3.3 ?????

□□□	□□□□	□□□□
ShipInboundShipment	□□□□□□□□□□□□□□	
ShipInboundBox	□□□□□□□□□□□□□□	
ShipInboundCase	□□□□□□□□□□□□□□	
ShipInboundDestinationAddress	□□□□□□□□□□□□□□□□ □	

4. API ?????

4.1 ?? API ??

□□ API □□	□□	□□□□□
shipmenthandlingApi.requestInboundShipment()	□□□□□□□□	ShipFormController.requestInboundShipmentAction()

API		
shipmenthandlingApi.disableShipment()		ShipFormController.cancelShipmentAction()
shipmenthandlingApi.localDoneShipment()		
shipmentPlacementApi.getBaseInfo()		ShipInboundPlanPlacementV2Controller.getBaseInfoAction()
shipmentPlacementApi.saveDestinationBox()		ShipInboundPlanPlacementV2Controller.saveDestinationBoxAction()

4.2 ?? API ??

getBaseInfoAction()	/api/v2/shipInboundPlan/shipment/getBaseInfo	
saveDestinationBoxAction()	/api/v2/shipInboundPlan/shipment/saveDestinationBox	
requestInboundShipmentAction()	/api/v1/shipForm/requestInboundShipment	
cancelShipmentAction()	/api/v1/shipForm/cancelShipment	
getPkgLabelUrlAction()	/api/v2/shipInboundPlan/shipment/getPkgLabelUrl	URL

5. ?????

5.1 ??????????

1. 请求 ID
2. 请求 ID URL 请求 ID
3. 请求 ID 请求 shipmentPlacementApi.getBaseInfo() 请求 ID
4. 请求 ID 请求 ID
5. 请求 ID 请求 ID

5.2 ??????????

1. 请求 ID 请求 ID
2. 请求 ID 请求 shipmenthandlingApi.requestInboundShipment() 请求 ID
3. 请求 ID 请求 ID
4. 请求 ID 请求 ID
5. 请求 ID 请求 shipmenthandlingApi.disableShipment() 请求 ID
6. 请求 ID 请求 ID

5.3 ????????

1. `shipmentPlacementApi.saveDestinationBox()`
2. `shipmentPlacementApi.saveDestinationBox()`
3. `shipmentPlacementApi.saveDestinationBox()`
4. `shipmentPlacementApi.saveDestinationBox()`
5. `shipmentPlacementApi.saveDestinationBox()`

5.4 ????????

1. `shipmentPlacementApi.saveDestinationBox()`
2. `shipmentPlacementApi.saveDestinationBox()`
3. `shipmentPlacementApi.saveDestinationBox()`

5.5 ????????

1. `shipmentHandlingApi.localDoneShipment()`
2. `shipmentHandlingApi.localDoneShipment()`
3. `shipmentHandlingApi.localDoneShipment()`
4. `shipmentHandlingApi.localDoneShipment()`
5. `shipmentHandlingApi.localDoneShipment()`

6. ????????

6.1 ????????

6.1.1 ????????

```
function getBaseInfo(shipmentid) {
  shipmentPlacementApi.getBaseInfo({
    "shipmentid": shipmentid
  }).then(res => {
    if (res.data) {
      var data = res.data;
      shipmentInfo.value = data.shipmentAll;
      shipment.value = data.shipment;
      plan.value = data.plan;
      volume.value = data.totalBoxSize;
      shiptype.value = data.shipment.transtyle;
    }
  });
}
```

```

// 箱数
if (data.listbox && data.listbox.length > 0) {
    boxnum.value = data.listbox.length;
} else {
    boxnum.value = data.shipment.boxnum;
}
// 価格
if (data.detail) {
    totalprice.value = "¥" + getValue(parseFloat(data.detail.itemprice));
}
// 送り先
if (data["fromAddress"]) {
    fromAddress.value = data["fromAddress"];
}
if (data["toAddress"]) {
    toAddress.value = data["toAddress"];
}
if (data["toAddressBox"]) {
    toAddressBox.value = data["toAddressBox"];
}
// 送料
if (data.transinfo == null || data.transinfo == undefined) {
    totalfee.value = "0";
} else {
    if (!data.transinfo.otherfee) data.transinfo.otherfee = 0;
    if (!data.transinfo.transweight) data.transinfo.transweight = 0;
    if (!data.transinfo.singleprice) data.transinfo.singleprice = 0;
    totalfee.value = ("¥" + parseFloat(parseFloat(data.transinfo.transweight)
* parseFloat(data.transinfo.singleprice)) + parseFloat(data.transinfo.otherfee));
}
// 重量
boxweightvalue.value = data.totalweight;
if (data.transinfo && data.transinfo.transweight) {
    if (data.transinfo.wunit) {
        weightvalue.value = (data.transinfo.transweight + "" +
getValue(data.transinfo.wunit));
    } else {
        if (data.transchannel.priceunits == "weight") {
            weightvalue.value = (data.transinfo.transweight + "kg");
        } else {

```

```

        weightvalue.value = (data.transinfo.transweight + "cbm");
    }
}
}
// 重量
if (data.detail) {
    readyweightvalue.value = (getValue(data.detail.readweight));
}
emit("change", data);
}
})
}

```

6.1.2 ?????

```

function deleteShipment() {
    // 显示modal 对话框
    dialogVisible.value = true;
    statusLoading.value = true;
    var status = shipDatas.value.shipmentstatus;
    shipmenthandlingApi.requestInboundShipment({
        "shipmentid": shipmentid
    }).then(res => {
        if (res.data != "fail") {
            status = res.data;
            shipstatus.value = res.data;
            if (status != "CANCELLED") {
                visibleBtn.value = "";
                canceltitle.value = "删除" + status + ",请确认";
            } else {
                canceltitle.value = "删除" + status + ",请确认";
            }
        } else {
            visibleBtn.value = "";
            canceltitle.value = "删除,请确认";
        }
        statusLoading.value = false;
        emit("change");
    })
}

```

```

function confirmDelete(ftype) {
  var nowstatus = "";
  if (ftype == "local") {
    nowstatus = "DELETED";
  } else {
    nowstatus = shipstatus.value;
  }
  confirmCancelLoading.value = true;
  shipmenthandlingApi.disableShipment({
    "shipmentid": shipmentid,
    "shipmentStatus": nowstatus,
    "disableShipment": "1"
  }).then(res => {
    ElMessage.success('成功');
    confirmCancelLoading.value = false;
    dialogVisible.value = false;
    context.emit("change");
  }).catch(error => {
    confirmCancelLoading.value = false;
  })
}

```

6.2 ????????

6.2.1 ??????????

```

@GetMapping("/getBaseInfo")
public Result<Map<String, Object>> getBaseInfoAction(@ApiParam("ID")@RequestParam String
shipmentid) {
  ShipInboundShipment ship = null;
  if(shipmentid.contains("FBA")){

  ship=shipInboundShipmentV2Service.lambdaQuery().eq(ShipInboundShipment::getShipmentConfirmatio
nId,shipmentid).one();
  }else{

  ship=shipInboundShipmentV2Service.lambdaQuery().eq(ShipInboundShipment::getShipmentid,shipment
id).one();

```

```

}
ShipInboundShipmenSummarytVo data =
shipInboundShipmentV2Service.summaryShipmentItemWithoutItem(ship.getShipmentid());
BeanUtil.copyProperties(ship, data,"itemList");
ShipInboundPlan plan = shipInboundPlanV2Service.getById(ship.getFormid());
if(plan!=null) {
    data.setMarketplaceid(plan.getMarketplaceid());
    data.setTranstyle(ship.getTranstyle());

data.setCountryCode(marketplaceService.findMapByMarketplaceId().get(plan.getMarketplaceid()).g
etMarket());
}
List<ShipInboundShipment> shipments =
shipInboundShipmentV2Service.lambdaQuery().eq(ShipInboundShipment::getFormid,
ship.getFormid()).list();
List<String> shipmentids=new LinkedList<String>();
for(ShipInboundShipment shipment:shipments) {
    shipmentids.add(shipment.getShipmentid());
}
plan.setShipmentids(shipmentids);
Map<String, Object> map = getItemPriceAction(ship.getShipmentid());
SummaryShipmentVo detail = shipInboundShipmentV2Service.showPlanListByPlanid(
ship.getShipmentid());
map.put("detail", detail);
map.put("plan", plan);

ship.setShipmentstatus(shipInboundShipmentV2Service.getShipmentStatusName(ship.getShipmentstat
us()));
map.put("shipment", ship);
map.put("shipmentid", ship.getShipmentid());
ShipAddress fromAddress = shipAddressService.getById(plan.getSourceAddress());
ShipInboundDestinationAddress toAddress =
shipInboundShipmentV2Service.getToAddress(ship.getDestination());
if(ship.getDestinationbox()!=null){
    ShipInboundDestinationAddress toAddressBox =
shipInboundShipmentV2Service.getToAddress(ship.getDestinationbox());
    map.put("toAddressBox", toAddressBox);
}
map.put("toAddress", toAddress);
map.put("fromAddress", fromAddress);

```

```

map.put("shipmentAll",data);
if(plan.getAreCasesRequired()!=null&&plan.getAreCasesRequired()==true){
    if((plan.getSubmitbox()==null||plan.getSubmitbox()==false)&&ship.getStatus()==3) {
        this.getEditBoxDetialCaseAction(map, ship, plan);
    }else {
        getBoxDetialCaseAction(map,ship,plan);
    }
}else{
    if((plan.getSubmitbox()==null||plan.getSubmitbox()==false)&&ship.getStatus()==3) {
        this.getEditBoxDetialAction(map, ship, plan);
    }else {
        getBoxDetialAction(map,ship,plan);
    }
}
getShipAmazonInfoAction(map,ship);
return Result.success(map);
}

```

6.2.2 ????????

```

@ApiOperation(value = "????????????")
@SystemControllerLog("??????")
@GetMapping("/saveDestinationBox")
public Result<String> saveDestinationBoxAction(String shipmentid,String destinationBox) {
    UserInfo user=UserInfoContext.get();
    ShipInboundShipment shipment = shipInboundShipmentV2Service.getById(shipmentid);
    if(destinationBox.equals("NA")){
        shipment.setDestinationbox(null);
    }else{
        shipment.setDestinationbox(destinationBox);
    }
    shipInboundShipmentV2Service.updateById(shipment);
    return Result.success();
}

```

7. ?????

7.1 ?????

名称 / 库	版本	用途
Vue	3.x	前端框架
Element Plus	3.x	UI 组件库
Icon Park	3.x	图标库
Axios	3.x	HTTP 客户端

7.2 后端技术栈

名称 / 库	版本	用途
Spring Boot	3.x	后端框架
MyBatis Plus	3.x	ORM 框架
Swagger	3.x	API 文档
MySQL	3.x	数据库

8. 项目部署

8.1 部署到生产环境

1. 配置环境变量

```
const { data } = await shipmentPlacementApi.getBaseInfo({ shipmentid: "shipment123" });
```

2. 打包项目

```
{
  "code": 0,
  "msg": "",
  "data": {
    "shipmentAll": {
      "number": "FBA202601230001",
      "name": "测试单",
      "shipmentConfirmationId": "FBA1234567890",
      "referenceid": "REF123",
      "shipmentstatus": "WORKING",
      "groupname": "测试组",
      "country": "US",
    }
  }
}
```

```
"warehouse": "0000",
"remark": "0000",
"skuamount": 2,
"sumQuantity": 100
},
"shipment": {
  "shipmentid": "shipment123",
  "shipmentConfirmationId": "FBA1234567890",
  "destination": "destination1",
  "destinationbox": "destination2",
  "transtyle": "SP",
  "status": 3
},
"plan": {
  "id": "plan123",
  "sourceAddress": "address1",
  "marketplaceid": "US",
  "areCasesRequired": false
},
"detail": {
  "itemprice": 5000
},
"fromAddress": {
  "name": "0000",
  "addressline1": "000000",
  "city": "00",
  "stateorprovincecode": "00",
  "postalcode": "518000",
  "countrycode": "CN"
},
"toAddress": {
  "name": "0000",
  "addressLine1": "123 Main St",
  "city": "Los Angeles",
  "stateOrProvinceCode": "CA",
  "postalCode": "90001",
  "countryCode": "US"
},
"toAddressBox": {
```

```
    "name": "□□□□□□",
    "addressLine1": "456 Oak St",
    "city": "New York",
    "stateOrProvinceCode": "NY",
    "postalCode": "10001",
    "countryCode": "US"
  },
  "totalweight": 25,
  "totalBoxSize": 0.5,
  "transinfo": {
    "transweight": 25,
    "singleprice": 10,
    "otherfee": 50
  }
}
```

8.2 ??????????

□□ □

```
const { data } = await shipmentPlacementApi.saveDestinationBox({
  shipmentid: "shipment123",
  destinationBox: "destination2"
});
```

□□ □

```
{
  "code": 0,
  "msg": "",
  "data": ""
}
```

8.3 ?????

□□ □

```
const { data } = await shipmenthandlingApi.disableShipment({
  shipmentid: "shipment123",
  shipmentStatus: "CANCELLED",
  disableShipment: "1"
});
```

□ □

```
{
  "code": 0,
  "msg": "",
  "data": true
}
```

9. ??????

9.1 ??????????

flowchart TD

```
A[????????] --> B[????ID]
B --> C[??getBaseInfo API]
C --> D[??????]
D --> E[??????]
E --> F[??????]
F --> G[??????]
G --> H[??????]
H --> I[????????]
I --> J[??????]
J --> K[??????]
```

9.2 ????????

flowchart TD

```
A[????????] --> B[??requestInboundShipment API]
B --> C[????????]
C --> D[????????]
D --> E{????????}
```


