

??-?????

FBA ??????????????

1. ?????

```

[ ] FBA [ ] API
[ ] wimoor666\wimoor-
ui\src\views\erp\shipv2\shipment_add\approve\ [ ] FBA
[ ]

```

2. ???????

2.1 ?????

```

index.vue ( )
├─ [ ]
│  ├─ [ ]
│  │  ├─ [ ]
│  │  │  └─ [ ] ( / )
│  │  └─ [ ]
│  └─ [ ]
│     └─ [ ] (Table )
│        └─ [ ] (List )
└─ [ ]
   └─ [ ] (approveVisible)
      │  └─ [ ]
      │  └─ [ ] ( / )
      │  └─ [ ]
      └─ [ ] (SplitDialog)

```

2.2 ?????

文件	内容
wimoor666\wimoor-ui\src\views\erp\shipv2\shipment_add\approve\index.vue	XXXXXXXXXXXX
wimoor666\wimoor-ui\src\views\erp\shipv2\shipment_add\approve\components\table.vue	XXXXXX
wimoor666\wimoor-ui\src\views\erp\shipv2\shipment_add\approve\components\list.vue	XXXXXX
wimoor666\wimoor-ui\src\views\erp\shipv2\shipment_add\approve\components\split_dialog.vue	XXXXXX

3. API ?????

3.1 ?? API ??

API 名称	方法	参数	文件
shipmentplanApi.getPlanInfo({formid})	GET	{formid: planid}	[shipmentplanApi.js]
shipmentplanApi.approveInboundPlan({id})	POST	{id: plandata.value.id}	[shipmentplanApi.js]
shipmentplanApi.rejectInboundPlan({id})	POST	{id: plandata.value.id}	[shipmentplanApi.js]
shipmentplanApi.createShipment({shipmentid})	POST	{shipmentid: item.shipmentId}	[shipmentplanApi.js]
shipmentplanApi.cancelShipment({shipmentid})	POST	{shipmentid: item.shipmentId}	[shipmentplanApi.js]
calculateApi.inplacefee(paramlist)	POST	paramlist (SKU XXXXXXXXXX)	[calculateApi.js]

3.2 ????????

控制器	方法	URL	文件
ShipInboundPlanV2Controller.getPlanInfoAction()	GET	/api/v2/shipInboundPlan/getPlanInfo	ShipInboundPlanV2Controller.java
ShipInboundPlanV2Controller.approveInboundPlan()	POST	/api/v2/shipInboundPlan/approveInboundPlan	ShipInboundPlanV2Controller.java

클래스	메서드	URL	파일
ShipInboundPlanV2Controller	rejectInboundPlan()	/api/v2/shipInboundPlan/rejectInboundPlan	ShipInboundPlanV2Controller.java
ShipFormController	createShipmentAction()	/api/v1/shipForm/create	ShipFormController.java
ShipFormController	cancelShipmentAction()	/api/v1/shipForm/cancelShipment	ShipFormController.java

4. ???????

4.1 ??????

클래스	테이블	테이블명	파일
ShipInboundPlan	ShipInboundPlan	t_erp_ship_v2_inboundplan	ShipInboundPlan.java
ShipInboundItem	ShipInboundItem	t_erp_ship_v2_inbounditem	ShipInboundItem.java
ShipInboundShipment	ShipInboundShipment	t_erp_ship_v2_inboundshipment	ShipInboundShipment.java
ShipInboundOperation	ShipInboundOperation	t_erp_ship_v2_inboundoperation	ShipInboundOperation.java

4.2 ??????

ShipInboundPlan (1) — (N) ShipInboundItem
 ShipInboundPlan (1) — (N) ShipInboundShipment
 ShipInboundPlan (1) — (N) ShipInboundOperation

5. ???????

5.1 ??????

1. 클래스 `onMounted` 메서드에서 `loadData()` 호출
2. `loadData()` 메서드에서 `shipmentplanApi.getPlanInfo()` 호출
3. `loadData()` 메서드에서
 - `plandata.value` 반환
 - `tableRef.value.planData`에 `listRef.value.planData` 할당
 - `summaryTotalPrice(item)` 호출

- `tableRef.value.initData(item)` `|||||`
- `listRef.value.remark` `||||`

5.2 ?????

1. `|||||` `|||` "||" "||" → `||` `approvePlan()`
2. `|||||` `|||||` "||" "||" "||" "
3. `|||||` `||`
 - `||` "||" " → `||` `shipmentplanApi.approveInboundPlan()`
 - `||` "||" " → `||` `shipmentplanApi.rejectInboundPlan()`
4. `|||||` `|||||` `loadData()` `|||||`

5.3 ???????

1. `|||||` `|||||` "||" "||" "||" "
2. `||||` `|||` "||||" "||||" → `||` `changeAllCancel()` → `|||||` "||" "
3. `||||` `|||` "||" "||" → `||` `submitplan()`
4. `|||||` `||`
 - `||||` "||" "||||" → `||` `shipmentplanApi.createShipment()`
 - `||||` "||" "||||" → `||` `shipmentplanApi.cancelShipment()`
5. `||||` `|||||` `loadData()` `|||||`

5.4 ???????

1. `||||` `|||` "||" "||" → `||` `splitHandel()` → `|||||`
2. `||||` `|||` "||" "||" → `||` `closePage()` → `|||||`
3. `||||` `|||` "||" "||" → `||` `copyPlan()` → `|||||`

6. ???????

6.1 ???????

6.1.1 ???????

```
function loadData() {
  if(planid) {
    //|||||plan
    shipmentplanApi.getPlanInfo({"formid":planid}).then((res) => {
      if(res.data) {
        res.data.iserror=false;
      }
    });
  }
}
```

```

    plandata.value=res.data;
    var item=res.data;
    var nowDate=new Date();
    item.nameVis=false;
    item.remarkVis=false;
    if(item.name==' ' || item.name==undefined || item.name==null) {

item.name="FBA"+"("+(nowDate.getMonth()+1)+"/"+(nowDate.getDate()+"/"+nowDate.getFullYear()+
"+nowDate.getHours()+":"+nowDate.getMinutes()+")-"+(index+1)
    }
    if(res.data.auditstatus==3 && item.status==1) {
        res.data.iserror=true;
    }

    tableRef.value.planData=res.data;
    summaryTotalPrice(item);
    tableRef.value.initData(item);
    listRef.value.planData=res.data;
    listRef.value.remark=res.data.remark;

    }
});
}
}

```

6.1.2 ?????

```

function approvePlan() {
    ElMessageBox.confirm('?????????', '??', {
        distinguishCancelAndClose: true,
        confirmButtonText: '??',
        cancelButtonText: '??',
        type: 'warning',
        center: true,
    })
    .then(() => {
        submitLoading.value=true;
        shipmentplanApi.approveInboundPlan({"id":plandata.value.id}).then(res => {
            ElMessage.success('????');
            submitLoading.value=false;

```

```

        loadData();
    }).catch(error => {
        submitLoading.value=false;
        console.log(error);
    });
})
.catch((action) => {
    if(action=='cancel') {
        submitLoading.value=true;
        shipmentplanApi.rejectInboundPlan({"id":plandata.value.id}).then(res => {
            ElMessage.success('取消成功');
            loadData();
            submitLoading.value=false;
        }).catch(error => {
            submitLoading.value=false;
        });
    }
})
}
}

```

6.1.3 ??????

```

async function submitplan() {
    confirmloading.value=true;
    for(var i=0;i<shipmentList.value.length;i++) {
        var item=shipmentList.value[i];
        if(item.approve==true) {
            await shipmentplanApi.createShipment({"shipmentid":item.shipmentId}).then(res => {
                ElMessage.success( item.shipmentId+'创建成功');
            }).catch(error => {
            });
        } else {
            await shipmentplanApi.cancelShipment({"shipmentid":item.shipmentId}).then(res => {
                ElMessage.success(item.shipmentId+'取消成功');
            }).catch(error => {
            });
        }
    }
}
nextTick(() => {
    confirmloading.value=false;
}

```

```

})
nextTick(() => {
    approveVisible.value=false;
})
var timer=setTimeout(function() {
    nextTick(() => {
        loadData();
    })
}, 300);
}

```

6.2 ???????

6.2.1 ???????

```

@ApiOperation(value = "??????")
@GetMapping("/getPlanInfo")
public Result<ShipPlanVo> getPlanInfoAction(String formid) {
    UserInfo user=UserInfoContext.get();
    try {
        if(StrUtil.isEmptyIfStr(formid)) {
            throw new BizException("??ID????");
        }

        ShipPlanVo vo = shipInboundPlanV2Service.getPlanBaseInfo(formid,user);
        return Result.success(vo);
    } catch(Exception e) {
        e.printStackTrace();
    }
    return Result.success(null);
}

```

6.2.2 ???????

```

@ApiOperation(value = "??????")
@GetMapping("/approveInboundPlan")
@SystemControllerLog("??")
@Transactional
public Result<String> approveInboundPlan(String id) {
    UserInfo user=UserInfoContext.get();
}

```



```

inplan.setAuditime(new Date());
inplan.setAuditor(user.getId());
shipInboundPlanV2Service.updateById(inplan);
shipInboundV2ShipmentRecordService.saveRecord(inplan);
return Result.success(inplan.getId());
}

```

6.2.4 ????

```

@ApiOperation(value = "取消发货")
@GetMapping("/cancelShipment")
public Result<Boolean> cancelShipmentAction(String shipmentid) {
    ShipInboundShipment shipment = shipInboundShipmentService.getById(shipmentid);
    if(shipment!=null) {
        shipment.setStatus(-1);
        ShipInboundPlan plan = shipInboundPlanService.getById(shipment.getInboundplanid());
        if(plan.getAuditstatus()==1) {
            plan.setAuditstatus(2);
            shipInboundPlanService.updateById(plan);
        }
        shipment.setShipmentstatus(ShipmentStatus.CANCELLED.getValue());
        boolean isupdate = shipInboundShipmentService.updateById(shipment);
        if(isupdate) {
            AmazonAuthority auth =
amazonAuthorityService.selectByGroupAndMarket(plan.getAmazongroupid(),
plan.getMarketplaceid());
            iFulfillmentInboundService.checkCancel(auth,plan,shipment);
        }
        return Result.judge(isupdate);
    } else {
        throw new BizException("取消发货失败");
    }
}
}

```

6. ????

□□	□□ /□□	□□	□□
□□	Vue	3.x	□□□□

이름	이름 / 설명	기능	기술
이름	Element Plus	기능	UI 라이브러리
이름	Icon Park	기능	아이콘
이름	Axios	기능	HTTP 클라이언트
이름	Spring Boot	기능	프레임워크
이름	MyBatis Plus	기능	ORM 라이브러리
이름	Swagger	기능	API 문서화
이름	MySQL	기능	데이터베이스

7. ???????

7.1 ?????

```
// 테스트용
const { data } = await shipmentplanApi.getPlanInfo({ formid: "plan123" });

// 테스트용
const { data } = await shipmentplanApi.approveInboundPlan({ id: "plan123" });

// 테스트용
const { data } = await shipmentplanApi.rejectInboundPlan({ id: "plan123" });

// 테스트용
const { data } = await shipmentplanApi.createShipment({ shipmentid: "shipment123" });

// 테스트용
const { data } = await shipmentplanApi.cancelShipment({ shipmentid: "shipment123" });
```

7.2 ?????

```
// 테스트용
{
  "code": 0,
  "msg": "",
  "data": {
```

```
"id": "plan123",
"number": "SF202601220001",
"name": "FBA(1/22/2026 14:30)-1",
"groupid": "group1",
"warehouseid": "warehouse1",
"marketplaceid": "US",
"amazonauthid": "auth1",
"sourceAddress": "address1",
"remark": "□□□□□□",
"auditstatus": 1,
"itemlist": [
  {
    "sku": "SKU001",
    "msku": "MSKU001",
    "quantity": 10,
    "labelOwner": "SELLER",
    "prepOwner": "SELLER"
  }
],
"shipmentList": [
  {
    "shipmentId": "shipment123",
    "name": "□□1",
    "toquantity": 10,
    "weight": 5.5,
    "readweight": 6.0,
    "boxvolume": "0.1",
    "itemprice": 100,
    "addressTo": {
      "countrycode": "US"
    },
    "destinationFulfillmentCenterId": "ABE2"
  }
]
}
}

// □□□□□□
{
```

```
"code": 0,  
"msg": "",  
"data": "plan123"  
}  
  
// 测试  
{  
  "code": 0,  
  "msg": "",  
  "data": "plan123"  
}
```

8. ??????

```
flowchart TD  
  A[ ] --> B[ ]  
  B --> C[ ]  
  C --> D{ }  
  D -->| | E[ ]  
  D -->| | F[ ]  
  D -->| | G[ ]  
  E --> H[ ]  
  H --> I{ }  
  I -->| | J[approveInboundPlan ]  
  I -->| | K[rejectInboundPlan ]  
  J --> L[ ]  
  K --> M[ ]  
  L --> N[ ]  
  M --> N  
  N --> O[ ]  
  
subgraph  
  P[ ]  
  Q[ ]  
  R[ ]  
  S[ ]  
  T[ ]  
end
```


