财务 ERP - 进销存模块（采购与台账）详细开发文档

版本历史

版本	日期	修改内容	作者
v1.0	2026-06-25	基于 20260618 需求梳理，定义采购模块与台账功能	架构组

---

1. 模块定位与范围

1.1 模块目标

实现采购业务全流程的财务核算与台账管理，打通采购订单 → 付款 → 入库 → 发票的业务闭环，并提供多维度的台账查询与对账功能。本模块作为现有 `wimoor-finance` 微服务的扩展，复用已有凭证、科目、辅助核算等基础能力。

1.2 涉及功能清单

- 进销存配置：采购账户、供应商收款账户、库存核算方法、凭证科目映射
- 采购单（会计版）：创建、付款、退款、入库、退货，状态追踪
- 采购账户台账：多维度查询、台账付款（账期结算）、批量上传对账单、退款处理
- 供应商台账：按供应商汇总应付/已付/待付、库存价值、发票状态，支持发票入账
- 发票台账：数电发票同步、入账、查询、统计分析

- 本地仓库台账：移动加权平均核算，汇总账与明细账，与总账勾稽
- FBA仓库台账：与亚马逊库存报告对接，按月汇总核算（后续迭代）

1.3 依赖关系

- 复用 `wimoor-finance` 现有表：`fin_accounting_subjects`、`fin_vouchers`、`fin_voucher_entries`、`fin_auxiliary_*`、`fin_currency`
- 调用 `wimoor-erp` 模块：供应商基础信息、SKU信息、仓库信息
- 调用 `wimoor-amazon` 模块：FBA库存报告数据（后续）
- 调用 `wimoor-admin` 模块：飞书多维表格连接配置（用于发票同步等）

---

2. 数据库设计（新增表）

2.1 采购账户表 `fin_purchase_account`

已在原文档中定义，此处补充字段说明：

字段	类型	说明
`id`	bigint unsigned	主键
`groupid`	bigint unsigned	租户ID（账套）
`account_name`	varchar(100)	账户名称，如"公户现金""跨境直采"
`account_type`	tinyint	1-现金，2-账期
`related_subject_id`	bigint unsigned	对应会计科目ID（如"银行存款-浦发"）
`config_json`	json	完整配置：费用类型列表、归属标记、借贷科目映射、退款设置、支持的付款账户等
`is_enabled`	tinyint	是否启用
`created_time`	datetime	
`updated_time`	datetime	

config_json 结构示例：

json

```
{
  "feeTypes": [
    { "code": "GOODS", "name": "货款", "isSupplierRelated": true },
    { "code": "FREIGHT", "name": "运费", "isSupplierRelated": true },
    { "code": "SERVICE", "name": "跨境直采手续费", "isSupplierRelated": false }
  ],
  "paySubjectMapping": {
    "GOODS": { "debitSubjectId": 101, "creditSubjectId": 201 },
    "FREIGHT": { "debitSubjectId": 102, "creditSubjectId": 201 },
    "SERVICE": { "debitSubjectId": 501, "creditSubjectId": 201 }
  },
  "refundMapping": {
    "GOODS": { "debitSubjectId": 201, "creditSubjectId": 101 },
    "FREIGHT": { "debitSubjectId": 201, "creditSubjectId": 102 }
  },
  "creditPayAccounts": [  // 仅账期类型有效，可选的现金账户列表
    { "cashAccountId": 1, "subjectMapping": { "debit": 301, "credit": 201 } }
  ]
}
```

2.2 采购订单表 `fin_purchase_order`

已在原文档中定义，补充以下索引和约束：

sql

```
ALTER TABLE `fin_purchase_order`
ADD INDEX `idx_status_pay` (`status_pay`),
ADD INDEX `idx_status_inv` (`status_inv`),
ADD INDEX `idx_status_stock` (`status_stock`);
```

2.3 采购付款明细表 `fin_purchase_payment`

已在原文档定义，确保与 `purchase_order_id` 和 `voucher_id` 关联。

2.4 供应商收款账户表 `fin_supplier_bank`

（新增）用于维护供应商的多个收款账户信息：

sql

```
CREATE TABLE `fin_supplier_bank` (
  `id` bigint unsigned NOT NULL AUTO_INCREMENT,
  `groupid` bigint unsigned NOT NULL,
  `supplier_id` bigint unsigned NOT NULL,
```

```
  `bank_name` varchar(100) COLLATE utf8mb4_bin DEFAULT NULL,
  `account_name` varchar(100) COLLATE utf8mb4_bin NOT NULL,
  `account_no` varchar(50) COLLATE utf8mb4_bin NOT NULL,
  `is_default` tinyint DEFAULT '0',
  `is_enabled` tinyint DEFAULT '1',
  `created_time` datetime DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_supplier` (`supplier_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin COMMENT='供应商收款账户';
```

2.5 库存余额表 `fin_warehouse_stock`

已在原文档定义，新增 `warehouse_type` 字段区分本地/FBA:

sql

```
ALTER TABLE `fin_warehouse_stock` ADD COLUMN `warehouse_type` tinyint DEFAULT '1' COMMENT '1-本地，2-FBA';
```

2.6 库存变动明细表 `fin_inventory_transaction`

已在原文档定义。

2.7 采购发票关联表（可选）

为支持“一张发票对应多笔采购订单”，可新增中间表，但当前简化设计，发票直接关联凭证，通过凭证分录的辅助核算追踪订单，暂不建中间表。

---

3. 核心业务逻辑

3.1 采购付款凭证生成规则（完整版）

根据需求，采购付款涉及三个环节：付款、在途库存确认、发票入账。系统按以下规则自动生成凭证。

3.1.1 现金类账户付款（公户、跨境直采等）

- 付款时（采购单点击“付款”）：
    - 凭证1（付款）：

      ```text
      借：预付账款-ERP采购供应商（供应商辅助核算）
      贷：银行存款-浦发银行(2384) / 其他货币资金-连连跨境直采
      ```

    - 凭证2（在途库存）：

      ```text
      借：在途库存_XX本地仓库（存货辅助核算）
      贷：预付账款-ERP在途发票
      ```

- 收到发票时（发票入账操作）：

  ```text
  借：预付账款-ERP在途发票
  贷：预付账款-ERP采购供应商
  ```

- 入库时（采购单收货）：

  ```text
  借：库存商品_XX本地仓库（存货辅助核算）
  贷：在途库存_XX本地仓库
  ```

3.1.2 账期类账户付款（先采后付、公户请款等）

- 采购单点击“付款”（即发起请款）：

  ```text
  借：预付账款-ERP采购供应商
  贷：应付账款_1688先采后付（或应付账款_公户请款即付等）
  ```

- 在途库存确认（同3.1.1）：

  ```text
  借：在途库存_XX本地仓库
  贷：预付账款-ERP在途发票
  ```

- 台账付款（实际还款）：
    - 借：应付账款_1688先采后付
    - 贷：银行存款-浦发 / 其他货币资金-连连跨境直采
    - 若有跨境直采手续费，另加一笔：

      ```text
      借：财务费用_跨境直采
      贷：其他货币资金-连连跨境直采
      ```

3.1.3 退款处理

退款时，根据已付款、已入库、已开票的不同状态，系统自动生成红字冲销凭证（方向相反），并更新台账。

具体规则已在需求中列出，此处不再赘述。

3.1.4 辅助核算要求

所有涉及供应商的科目必须启用 “供应商” 辅助核算；涉及存货的科目启用 “存货” 辅助核算（即 SKU）。系统在生成凭证时自动填充 `fin_voucher_entries_auxiliary` 表。

3.2 移动加权平均成本计算

在每次采购入库、盘盈入库时，重新计算库存单价。出库时，出库成本 = 出库数量 × 当前移动单价。

Java 核心实现：

java

```java
@Component
public class MovingAverageCostCalculator {

    /**
     * 计算新移动平均单价
     * @param currentQty 当前数量
     * @param currentAmount 当前金额
     * @param inQty 入库数量（正数）
     * @param inAmount 入库金额（正数）
     * @return 新单价，保留4位小数
     */
    public BigDecimal calculateNewUnitCost(BigDecimal currentQty, BigDecimal currentAmount,
                                           BigDecimal inQty, BigDecimal inAmount) {
        if (inQty.compareTo(BigDecimal.ZERO) == 0) {
            return BigDecimal.ZERO;
        }
        BigDecimal newQty = currentQty.add(inQty);
        if (newQty.compareTo(BigDecimal.ZERO) == 0) {
            return BigDecimal.ZERO;
        }
        BigDecimal newAmount = currentAmount.add(inAmount);
        return newAmount.divide(newQty, 4, RoundingMode.HALF_UP);
    }

    /**
     * 计算出库成本（金额）
     */
    public BigDecimal calculateOutCost(BigDecimal qty, BigDecimal unitCost) {
        return qty.multiply(unitCost).setScale(2, RoundingMode.HALF_UP);
    }
}
```

库存余额更新（确保不反算）：

java

```java
@Transactional
public void updateStockForTransaction(InventoryTransactionDTO dto) {
    // 1. 获取当前期间库存
    WarehouseStock stock = stockMapper.selectBySkuPeriod(...);
    BigDecimal oldQty = stock.getCurrentQty();
    BigDecimal oldAmount = stock.getCurrentAmount();
    BigDecimal oldUnitCost = stock.getUnitCost();

    if (dto.getTransType() == PURCHASE_IN) {
        // 入库: 重新计算单价
        BigDecimal newUnitCost = calculateNewUnitCost(oldQty, oldAmount, dto.getQty(), dto.getAm
        stock.setUnitCost(newUnitCost);
        stock.setCurrentQty(oldQty.add(dto.getQty()));
        stock.setCurrentAmount(oldAmount.add(dto.getAmount()));
    } else if (dto.getTransType() == SALE_OUT) {
        // 出库: 使用当前单价计算成本
        BigDecimal cost = calculateOutCost(dto.getQty().abs(), oldUnitCost);
        dto.setAmountChange(cost.negate()); // 记录变动金额
        stock.setCurrentQty(oldQty.subtract(dto.getQty().abs()));
        stock.setCurrentAmount(oldAmount.subtract(cost));
        // 单价不变
    }
    // 其他类型（调拨、盘点等）类似...
    stockMapper.updateById(stock);
    // 插入变动明细
    inventoryTransactionMapper.insert(buildTransaction(dto, oldUnitCost));
}
```

3.3 采购账户台账查询与批量付款

3.3.1 台账列表查询

支持多维度筛选（采购账户、账套、时间、状态），返回以下核心字段:

- 账簿、采购账户、采购订单号、SKU、供应商、收款账号
- 付款状态、库存状态、发票状态
- 金额字段：订单金额、请款金额（归属供应商）、已付金额、待付金额
- 库存字段：库存价值、已入库价值、待入库价值
- 发票字段：应开票金额、已开票金额、待开票金额

SQL 查询核心（**MyBatis XML**）：

xml

```xml
<select id="queryLedgerList" resultType="com.wimoor.finance.vo.PurchaseLedgerVO">
    SELECT
```

```
        po.groupid,
        po.id as orderId,
        po.sku,
        s.name as supplierName,
        sb.account_no as bankAccount,
        po.total_amount,
        po.paid_amount,
        po.inventory_value,
        po.invoiced_amount,
        -- 计算待付、待入库、待开票
        (po.total_amount - po.paid_amount) as pending_pay,
        (po.inventory_value - IFNULL(SUM(ir.received_value),0)) as pending_stock,
        (po.total_amount - po.invoiced_amount) as pending_invoice,
        po.status_pay,
        po.status_stock,
        po.status_inv
    FROM fin_purchase_order po
    LEFT JOIN t_erp_supplier s ON po.supplier_id = s.id
    LEFT JOIN fin_supplier_bank sb ON s.id = sb.supplier_id AND sb.is_default=1
    LEFT JOIN fin_inventory_transaction ir ON po.id = ir.source_doc_id AND ir.trans_type='RECEIV
    WHERE po.groupid = #{groupid}
    <if test="accountIds != null and accountIds.size()>0">
        AND po.purchase_account_id IN
    </if>
    <if test="startDate != null"> AND po.created_time >= #{startDate} </if>
    <if test="endDate != null"> AND po.created_time < #{endDate} </if>
    <if test="payStatus != null"> AND po.status_pay IN ... </if>
    GROUP BY po.id
    ORDER BY po.created_time DESC
</select>
```

3.3.2 台账付款操作

- 前端勾选多条记录 → 调用 `/api/finance/ledger/purchase-account/ledger-pay`
- 后端校验：所有记录必须属于同一个账期采购账户；必须选择现金类付款账户
- 生成凭证（借贷科目按配置）并插入 `fin_purchase_payment`，更新 `paid_amount` 和状态
- 支持分摊：若付款金额与应付总额不一致，按比例分摊到尾差处理

批量上传对账单（**Excel**）流程：

1. 下载模板（含列：采购订单号、SKU、应付金额、本次付款金额、费用类型）
2. 用户填写后上传
3. 后端解析，校验订单号有效、未超额付款
4. 预览确认，用户提交
5. 批量生成付款记录和凭证

3.4 供应商台账与发票入账

3.4.1 供应商台账查询

与采购账户台账类似，但按供应商维度汇总，展示：

- 供应商、账簿、采购订单明细
- 应付/已付/待付（归属供应商）
- 库存价值、已入库、待入库
- 应开票、已开票、待开票

3.4.2 发票入账操作

- 前端勾选多条未开票记录 → 选择数电发票（从发票台账中选）
- 调用 `/api/finance/ledger/invoice/post`
- 后端：生成凭证：

  text

  ```text
  借：预付账款-ERP在途发票
  贷：预付账款-ERP采购供应商
  ```

- 同时更新采购订单的 `invoiced_amount` 和 `status_inv`，更新发票的 `posting_status=1`、`voucher_id`
- 支持发票分摊到多条订单（按订单待开票金额比例）

3.5 发票台账同步与入账

3.5.1 发票数据同步

- 定时任务调用税局电子发票 API（或从金蝶拉取），将数电发票信息写入 `fin_invoice`
- 同步时，根据发票销方税号匹配内部供应商（`supplier_id`），若无法匹配则记录为“待匹配”
- 同步后，发票状态默认为“正常”

3.5.2 发票台账查询

提供多维度筛选（账簿、销方、日期、入账状态、发票状态），展示发票号码、开票日期、价税合计、税额、入账状态等。

3.5.3 手工入账（未匹配自动入账）

用户可在发票台账页面勾选发票，选择“入账”，系统生成凭证（借：预付账款-ERP在途发票 贷：预付账款-ERP采购供应商），并关联供应商（若之前未匹配）。

3.6 本地仓库台账（移动加权平均核算）

3.6.1 汇总账

按“期间 + 账簿 + SKU + 仓库”展示期初、入库、出库、期末的数量和金额。

- 期初 = 上一期间期末
- 入库 = 所有 `PURCHASE_IN` 类型变动汇总
- 出库 = 所有 `SALE_OUT` 类型变动汇总
- 期末 = 期初 + 入库 - 出库

勾稽校验：期末金额应与对应会计科目（库存商品）余额一致，若不一致，系统提示差异。

3.6.2 明细账

展示每笔库存变动明细：操作时间、类型、数量、单价、金额变动、关联单据号、凭证号。支持按 SKU、仓库、时间筛选。

---

4. API 接口详细设计

4.1 采购账户管理

方法	路径	说明	请求参数
POST	`/api/finance/purchase/account/create`	创建采购账户	`PurchaseAccountSaveDTO`
PUT	`/api/finance/purchase/account/update`	更新配置	`PurchaseAccountUpdateDTO`
GET	`/api/finance/purchase/account/list`	查询可用账户	`groupid`
GET	`/api/finance/purchase/account/detail/{id}`	获取详情	`id`

4.2 采购订单（会计版）

方法	路径	说明	请求参数

POST	`/api/finance/purchase/order/create`	创建采购订单（从 ERP 业务单转换）	`PurchaseOrderSaveDTO`
POST	`/api/finance/purchase/order/pay`	付款（生成凭证）	`PurchasePayDTO`（含付款账户、费用明细）
POST	`/api/finance/purchase/order/refund`	退款	`RefundDTO`
POST	`/api/finance/purchase/order/receive`	入库	`ReceiveDTO`（收货数量 / 金额）
GET	`/api/finance/purchase/order/page`	订单列表	`PageQuery` + 筛选条件

4.3 采购账户台账

方法	路径	说明
GET	`/api/finance/ledger/purchase/page`	台账列表（分页）
POST	`/api/finance/ledger/purchase/ledger-pay`	台账付款（批量）
POST	`/api/finance/ledger/purchase/upload-pay`	上传对账单批量付款
GET	`/api/finance/ledger/purchase/statistics`	统计图表数据

4.4 供应商台账

方法	路径	说明
GET	`/api/finance/ledger/supplier/page`	供应商台账列表
POST	`/api/finance/ledger/supplier/invoice-post`	发票入账
GET	`/api/finance/ledger/supplier/export-uninvoiced`	导出未开票明细

4.5 发票台账

方法	路径	说明
GET	`/api/finance/ledger/invoice/page`	发票台账列表
POST	`/api/finance/ledger/invoice/sync`	手动触发同步（定时任务自动）
POST	`/api/finance/ledger/invoice/post`	发票入账（勾选后）
GET	`/api/finance/ledger/invoice/statistics`	发票统计

4.6 库存台账

方法	路径	说明
GET	`/api/finance/ledger/stock/summary`	汇总账（按期间）
GET	`/api/finance/ledger/stock/detail`	明细账（按 SKU）
GET	`/api/finance/ledger/stock/check`	勾稽校验（与总账对比）

---

5. 服务层实现要点

5.1 采购付款服务（`PurchasePaymentService`）

- 职责：处理采购单付款、退款、台账付款
- 主要方法：
  - `payOrder(PurchasePayDTO)`：校验账户余额（现金类）、生成凭证、更新订单状态、更新台账
  - `ledgerPay(LedgerPayDTO)`：批量处理账期还款，支持分摊
  - `refund(RefundDTO)`：退款，生成红字凭证，更新订单状态

5.2 库存服务（`InventoryService`）

- 职责：管理库存余额、变动明细、成本计算
- 主要方法：
  - `onPurchaseReceive(PurchaseOrder order, ReceiveDTO)`：入库，调用成本计算器更新库存
  - `onSaleOut(SaleOutDTO)`：销售出库，计算成本并更新库存
  - `getStockSummary(String period)`：汇总账查询
  - `getStockDetail(String sku)`：明细账查询

5.3 发票服务（`InvoiceService`）

- 职责：同步税局发票、匹配供应商、入账
- 主要方法：
  - `syncFromTaxAPI()`：调用外部 API拉取发票
  - `matchSupplier()`：根据税号匹配供应商
  - `postInvoice(InvoicePostingDTO)`：生成入账凭证，关联采购订单

5.4 台账查询服务（`LedgerQueryService`）

- 职责：提供采购账户、供应商、发票、库存的统一查询视图
- 使用 MyBatis-Plus 分页 + 复杂 SQL关联查询

---

6. 前端页面交互设计

6.1 采购单（会计版）页面

- 列表页：展示订单号、SKU、供应商、金额、付款/库存/发票状态；支持筛选、排序
- 详情/操作页：显示完整信息，提供“付款”“退款”“收货”“退货”按钮，点击弹出对应表单
- 付款表单：选择采购账户（带出对应费用类型）、填写金额、选择供应商收款账户

6.2 采购账户台账页面

- 筛选区：采购账户下拉、账套、时间、状态
- 表格：展示上述金额字段，底部汇总行
- 操作区：“台账付款”按钮（勾选后弹出付款对话框），“导出对账单”按钮
- 图表页签：按日期展示付款趋势图

6.3 供应商台账页面

- 筛选区：供应商下拉、账套、时间
- 表格：展示应付/已付/待付、库存价值、应开票/已开票/待开票
- 操作：“发票入账”按钮（勾选后弹出选择发票对话框）

6.4 发票台账页面

- 筛选区：账簿、销方、日期、入账状态
- 表格：发票号码、金额、税额、状态
- 操作：“同步”按钮（手动触发），“入账”按钮

6.5 本地仓库台账

- 汇总账：期间选择、SKU搜索，展示期初/入库/出库/期末数量和金额，提供校验标识
- 明细账：点击SKU跳转明细，展示每笔变动，联查凭证

7. 定时任务与消息队列

7.1 定时任务（XXL-JOB）

任务名称	Cron	说明
`InvoiceSyncJob`	`0 0 2 * * ?`	同步税局发票数据
`StockCheckJob`	`0 0 3 1 * ?`	每月1号执行库存与总账勾稽校验
`AutoPostInvoiceJob`	`0 0/30 9-18 * * ?`	工作时间内自动匹配未入账发票（可选）

7.2 消息队列（RabbitMQ）

- 采购订单状态变更时，发送事件（如 `ORDER_PAID`、`ORDER_RECEIVED`）到 exchange，库存服务、台账服务监听更新。
- 避免同步调用，提高响应速度。

8. 异常处理与日志

8.1 异常场景

- 付款金额超过订单未付金额 → 提示“超额付款”
- 发票金额与订单待开票金额不一致 → 提示“金额不符，请确认分摊”

- 供应商未维护收款账户 → 提示 “请先维护供应商收款信息”
- 科目配置缺失 → 提示 “采购账户科目映射不完整，请联系管理员”

8.2 日志规范

- 关键操作（付款、退款、入库、发票入账）记录 INFO 日志，包含操作人、订单号、金额
- 异常堆栈记录 ERROR 日志，并发送飞书告警（通过 `wimoor-common` 的告警组件）

---

9. 测试策略

9.1 单元测试（JUnit + Mockito）

- 移动加权平均计算器
- 凭证生成逻辑（各场景）
- 分摊算法

9.2 集成测试（@SpringBootTest + Testcontainers）

- 完整流程：创建订单 → 现金付款 → 入库 → 发票入账 → 查询台账
- 账期流程：创建订单 → 账期付款 → 台账付款 → 查询供应商待付
- 退款流程：部分退款、全额退款（不同状态组合）

9.3 前端测试（Vue Test Utils）

- 页面渲染、表单交互、数据校验

---

10. 部署与配置

10.1 依赖环境

- MySQL 8.0（已有）
- Redis（用于缓存科目、辅助核算项）
- RabbitMQ（事件驱动）
- XXL-JOB（调度中心）

10.2 配置示例（application.yml）

yaml

```
wimoor:
  finance:
    inventory:
      cost-method: moving_average    # 仅支持移动加权
    invoice:
      sync-api: https://api.tax.gov/invoice
      sync-cron: 0 0 2 * * ?
    ledger:
      batch-size: 1000    # 批量导入单次最大行数
```

---

11. 后续扩展规划

- **FBA**仓库台账：对接亚马逊库存报告，按月汇总，与本地仓库成本联动
- 关务台账：报关单与FBA货件关联，统计出口数据
- 无票成本统计：在利润表中增加无票成本明细
- 银行**API**对接：自动获取对公账户流水，与采购账户对账

> 开发周期估计：
>
> - 数据库设计与基础服务：3天
> - 核心业务逻辑（凭证生成、成本计算、台账查询）：5天
> - 接口开发与联调：4天
> - 前端页面开发：5天
> - 测试与修复：3天
>
> 总计约**20**人天

Revision #2
Created 2026-06-25 01:19:35 UTC by Admin
Updated 2026-06-25 02:52:06 UTC by Admin