MD065

Traning Inventory module

Technical design

Document author: Zeki

Creation Date: December 13, 2024

Update Date:

Document Number: TRAINING.FINAL.INV.MD065

Current Version： 1.0

**Approval signature:**

|  |  |
| --- | --- |
| 〈XXXX project manager〉 |  |
| 〈Implementing Party Project Manager〉 |  |

## Document control

**Change history**

1

| Date | Author | Version | Change Description |
| --- | --- | --- | --- |
|  |  |  |  |
| 2024-12-13 | Zeki | 1.0 | initialization |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Review**

| Name | Position |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Distribution**

| Copy Number | Name | Position |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Remark:**

For document management purposes, if you have received an electronic version of this document, please print it out and write your name on the cover in the appropriate space

For document management purposes, if you receive a paper version of this document, please write your name on the cover

Catalog

Document control 2

Overview 4

Development content overview 5

Overall demand 5

Modular architecture 5

Data relationships (ER diagram) 5

Development list 5

Value Config 5

Front-end page 8

INV01-Counting Order Management 10

Counting order save verification method 11

Counting order save method 12

Counting order remove method 14

Counting order query method 14

Counting order execute verification method 16

Counting order execute method 17

Counting order synchronization WMS method 19

Counting Result Synchronous method 20

Counting order submit verification method 21

Counting order submit method 22

Counting order report dataset method 24

Appendix 25

Open and closed issues 26

Open issues 26

Closed issues 26

## Overview

This article mainly describes the development and implementation of related businesses of the inventory module.

## Development content overview

### Overall demand

#### Requirements description

#### Business Process

### Modular architecture

#### Module architecture diagram

### Data relationships (ER diagram)

#### Data Relationship

### Development list

### Value Config

#### List of value definitions

| Code | INV.COUNTING.COUNT\_STATUS | | Parent Code | |  |
| --- | --- | --- | --- | --- | --- |
| Description | Maintain counting status value | |  | |  |
| value | meaning | description | Parent code | tag | |
| DRAFT | Draft | Draft |  |  | |
| INCOUNTING | In counting | In counting |  |  | |
| PROCESSING | Processing | Processing |  |  | |
| WITHDRAWN | Withdrawn | Withdrawn |  |  | |
| APPROVED | Approved | Approved |  |  | |
| REJECTED | Rejected | Rejected |  |  | |
| CONFIRMED | Confirmed | Confirmed |  |  | |

| Code | INV.COUNTING.COUNT\_DIMENSION | | Parent Code | |  |
| --- | --- | --- | --- | --- | --- |
| Description | Maintain counting demension value | |  | |  |
| value | meaning | description | Parent code | tag | |
| SKU | Sku | Sku |  |  | |
| LOT | Sku + Batch | Sku + Batch |  |  | |

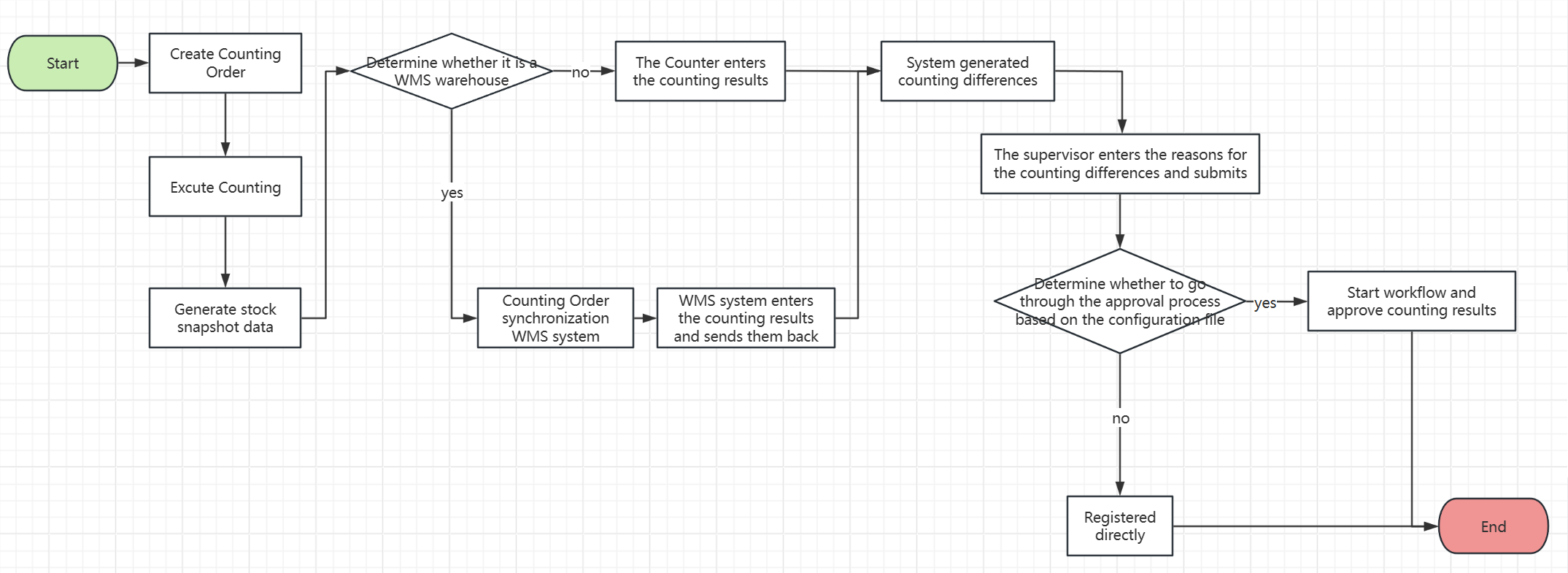
| Code | INV.COUNTING.COUNT\_TYPE | | Parent Code | |  |
| --- | --- | --- | --- | --- | --- |
| Description | Maintain counting type value | |  | |  |
| value | meaning | description | Parent code | tag | |
| MONTH | Counting per month | Counting per month |  |  | |
| YEAR | Counting per year | Counting per year |  |  | |

| Code | INV.COUNTING.COUNT\_MODE | | Parent Code | |  |
| --- | --- | --- | --- | --- | --- |
| Description | Maintain counting mode value | |  | |  |
| value | meaning | description | Parent code | tag | |
| VISIBLE\_COUNT | Visible count | Visible count |  |  | |
| UNVISIBLE\_COUNT | Unvisible count | Unvisible count |  |  | |

#### Configuration maintenance(ProfileClient)

| Configuration code | Configuration description | Configuration value | value |
| --- | --- | --- | --- |
| FEXAM95.INV.COUNTING.ISWORKFLOW | Determine whether to go through the approval process | Global | 1 |
|  |  |  |  |
|  |  |  |  |

#### Flow chart



图示

描述已自动生成

## Front-end page

#### Counting Order Query

图形用户界面, 应用程序

描述已自动生成

#### Counting Order Create

社交网络的手机截图

描述已自动生成

#### Counting Order Preview

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

## INV01-Counting Order Management

#### Background

For materials with physical inventory, can counting through the system counting function.

#### PRD position

Omit

#### API(14)

**Counting order save(API name: orderSave):** Counting order save verification method -> Counting order save method

**Counting order remove(API name: orderRemove):** Counting order remove method

**Counting order query(API name: list, detail):** Counting order query method

**Counting order execution(API name: orderExecution):** Counting order save verification method -> Counting order save method -> Counting order execute verification method -> Counting order execute method -> Counting order synchronization WMS method

**Submit counting results for approval(API name: orderSubmit):** Counting order save verification method -> Counting order save method -> Counting order submit verification method -> Counting order submit method

**Counting result synchronous(API name: countResultSync):** Counting result synchronous method

**Counting order report dataset method(API name: countingOrderReportDs):** Counting order report dataset method

#### Basic DTO

InvCountInfoDTO

|  |
| --- |
| public class InvCountInfoDTO {  private String totalErrorMsg;  @ApiModelProperty(value = "Verification passed list")  private List<InvCountHeaderDTO> successList;  @ApiModelProperty(value = "Verification failed list")  private List<InvCountHeaderDTO> errorList;  } |

InvCountHeaderDTO

|  |
| --- |
| public class InvCountHeaderDTO extends InvCountHeader {  @ApiModelProperty(value = "Error Message")  private String errorMsg;  private List<InvCountLineDTO> invCountLineDTOList;  ...  } |

### Counting order save verification method

#### Program1(8)：Counting order save verification method(InvCountHeaderController)

Parameter: List<InvCountHeaderDTO>

Return: InvCountInfoDTO

Method name: manualSaveCheck

1. Not null verification, value sets, and other business verifications are processed before the counting order save
2. Pre-verification(pseudocode)

|  |
| --- |
| // NotNull /NotBlank verification  validObject(invCountHeader, validateClass);  // other business verifications  // Foreach input data  For(invCountHeader : invCountHeaders) {  // 1. Insert new data  // skip validation  If(headerId == null) {return;}  // 2. Update data  // a. not allow update status field  // b. Judge whether the document status is "Draft, In Counting, Rejected, Withdrawn" status  if(status != Draft, In Counting, Rejected, withdrawn) {  // verification failed , the error message is: only draft, in counting, rejected, and withdrawn status can be modified  }  // c. Judge the current user  if(status = draft and creator != current user) {  // verification failed, the error message is:Document in draft status can only be modified by the document creator.  }  if(status = "in counting, rejected, withdrawn") {  if(warehouse is WMS warehouse and current operator isn’t supervisor) {  // verification failed, the error message is: The current warehouse is a WMS warehouse, and only the supervisor is allowed to operate.  }  if(current operator not in Head counter list, supervisor list, document creator) {  // verification failed, the error message is: only the document creator, counter, and supervisor can modify the document for the status of in counting, rejected, withdrawn.  }  }  }  // Collect error information and throw |

### Counting order save method

#### Program1(5): Counting order save method(InvCountHeaderController)

Parameter: List<InvCountHeaderDTO>

Return: List<InvCountHeaderDTO>

Method name: manualSave

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field logic** | | | | | |
| **Counting Order Header** | | | | | |
| **Field Name** | **Description** |  | **Is necessary**  **(from input)** | **Value logic** |  |
| tenantId | Tenant ID |  | Yes | current tenant id |  |
| countHeaderId | Primary key |  | No | Auto increment |  |
| countNumber | document number |  | No | Code rule: INV.COUNTING95.COUNT\_NUMBER  Updates not allowed  Rule: INV-Counting-currentTenantId-20241122-001 |  |
| countStatus | document status |  | Yes | Default value: DRAFT  Lov: INV.COUNTING.COUNT\_STATUS  Updates not allowed |  |
| companyId | Company id |  | Yes | Only draft status allows updates |  |
| departmentId | Department id |  | No | Only draft status allows updates |  |
| warehouseId | Warehouse id |  | Yes | Only draft status allows updates |  |
| countDimension | Counting dimension |  | No | Only draft status allows updates |  |
| countType | Counting type |  | No | Only draft status allows updates |  |
| countMode | Counting mode |  | No | Only draft status allows updates |  |
| countTimeStr | Counting time |  | No | Only draft status allows updates |  |
| counterIds | Counter ids |  | Yes | Only draft status allows updates |  |
| supervisorIds | Supervisor ids |  | Yes | Only draft status allows updates |  |
| snapshotMaterialIds | Snapshot material ids |  | No | Only draft status allows updates |  |
| snapshotBatchIds | Snapshot batch ids |  | No | Only draft status allows updates |  |
| remark | Remark |  | No | Only draft, in counting status allows updates |  |
| reason | Explanation of the reasons for the counting discrepancy |  | No | Only in counting, rejected status allows updates |  |
| delFlag | Delete flag, 1 for deletion/0 for normal |  | No | Default value is 0 |  |
| **Counting Order Line** | | | | | |
| **Field Name** | **Description** |  | **Is necessary**  **(from input)** | **Value logic** |  |
| tenantId | Tenant ID |  | Yes | current tenant id |  |
| countLineId | Primary key |  | Yes | Auto increment |  |
| countHeaderId | Count header ID, association  fexam\_inv\_count\_header |  | Yes | From header data |  |
| lineNumber | Line number |  | Yes | Natural serial number, start from 1, generated by code |  |
| snapshotUnitQty | On hand quantity |  | Yes | Get from stock table(unit quantity) |  |
| unitQty | Counting result quantity |  | No | Only in counting status allows updates, and only counter can modify |  |
| unitDiffQty | The main unit difference quantity after counting |  | No | Automatic calculation, only the in counting status is allowed to be updated |  |
| counterIds | Counter ids |  | No | Initially use the value on the counting order header. Only the status is in counting and the document creator allows update. And accounting the counting order header counter list.  When the counter finish the row unit qty and save, use current use id cover the value.  For example,  header.counterIds=1,2,3  After execute, line.counterIds=1,2,3  Document creator can modify this field but limit in (1,2,3)  After counter(eg. 2) finish the unit qty, modify value(eg. 2) |  |
| remark | Remark |  | No | Only in counting status allows updates |  |

### Counting order remove method

#### Program1(5): Counting order remove(InvCountHeaderController)

Parameter: List<InvCountHeaderDTO>

Return: InvCountInfoDTO

Method name: checkAndRemove

1. Pre-verification(pseudocode)

|  |
| --- |
| For(invCountHeader : invCountHeaders) {  // status verification, Only allow draft status to be deleted  If(status != draft) {setting error message}  // current user verification, Only current user is document creator allow delete document  If(current user != header.createdBy) { setting error message }  }  // Collect error information and throw |

1. Delete documents

|  |
| --- |
| Physical removal |

### Counting order query method

#### Program1(4): List query(InvCountHeaderController)

Parameter: InvCountHeaderDTO

Return: Page<List<InvCountHeaderDTO>>

Method name: list

1. Data Sorting：Sort by document creation time in descending order(use @SortDefault)

|  |  |  |  |
| --- | --- | --- | --- |
| **Query conditions** | | | |
| **Field Name** | **Description** |  | **Value logic** |
| countNumber | Document number |  | Support fuzzy search |
| countStatus | Document status |  | LOV: INV.COUNTING.COUNT\_STATUS  Pass lov code |
| countDimension | Counting Dimensions |  | LOV: INV.COUNTING.COUNT\_DIMENSION  Pass lov code |
| countType | Counting type |  | LOV: INV.COUNTING.COUNT\_TYPE  Pass lov code |
| countMode | Counting mode |  | LOV: INV.COUNTING.COUNT\_MODE  Pass lov code |
| supervisor | Supervisor |  | Type String, pass user id, support single, not multi |

#### Program2(6): Detail query(InvCountHeaderController)

Parameter: Long countHeaderId

Return: InvCountHeaderDTO

Method name: detail

1. Data Sorting：Sort by document creation time in descending order

|  |  |  |  |
| --- | --- | --- | --- |
| **Special processing logic** | | | |
| **Field Name** | **Description** |  | **Value logic** |
| counterList | Counter list |  | Convert String to List  Eg.  Database value: 1,2,3  Return: counterList:[{id:1,realName:ZH1},{id:2,realName:ZH2},{id:3,realName:ZH3}] |
| supervisorList | Supervisor list |  | Convert String to List  Eg.  Database value: 1,2,3  Return: counterList:[{id:1,realName:ZH1},{id:2,realName:ZH2},{id:3,realName:ZH3}] |
| snapshotMaterialList | Snapshot Material List |  | Convert String to List  Eg.  Database value: 1\_item1,2\_item2,3\_item3  Return: counterList:[{id:1,code:item1},{id:2, code:item2},{id:3, code:item3}] |
| snapshotBatchList | Snapshot Batch List |  | Convert String to List  Eg.  Database value:1,2,3  Return: counterList:[{batchId:1,batchCode:b1},{batchId:2,batchCode:b2},{batchId:3,batchCode:b3}] |
| isWMSwarehouse | Is WMS warehouse |  | Get info from fexam\_inv\_warehouse table |
| invCountLineDTOList | Inv Counting Line DTO List |  | List, query from counting line table by countHeaderId |

#### Program3(6): Data permission rule

1. Restrictions on the counting order header table:

* Rule code: your service name.INVOICE\_HEADER eg.FEXAM-21995.INVOICE\_HEADER
* Only the document creator can view the draft status
* Non-draft status documents can be viewed by relevant persons(creator,counter,supervisor)
* The tenant administrator can view all data of the corresponding tenant.

1. Restrictions on the counting order line table:

* Rule code: your service name.INVOICE\_LINE eg.FEXAM-21995.INVOICE\_ LINE
* The counter can only see the rows they need to counting
* The supervisor and the document creator can see all rows
* The tenant administrator can view all data of the corresponding tenant.

### Counting order execute verification method

#### Program1(5): counting order execute verification method(InvCountHeaderController)

Parameter: List<InvCountHeaderDTO>

Return: InvCountInfoDTO

Method name: executeCheck

1. Mandatory input verification(Reference table - field logic)
2. Pre-verification(pseudocode)

|  |
| --- |
| Requery the database based on the input document ID  For(invCountHeader : invCountHeaders) {  // a. document status validation: Only draft status can execute  // b. current login user validation: Only the document creator can execute  // c. value set validation  // d. company, department, warehouse validation  // e. on hand quantity validation  Query the stock data that the on hand quantity is not 0 according to the tenantId + companyId + departmentId + warehouseId + snapshotMaterialIds + snapshotBatchIds on the table header, if no data is queried, error message: Unable to query on hand quantity data.  } |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Counting Order Header** | | | | | |
| **Field Name** | **Description** |  | **Is necessary**  **(from input)** | **Value logic** |  |
| tenantId | Tenant id |  | Yes |  |  |
| countStatus | Document status |  | Yes | LOV: INV.COUNTING.COUNT\_STATUS |  |
| companyId | Company id |  | Yes | - |  |
| warehouseId | Warehouse id |  | Yes | - |  |
| countDimension | Counting dimension |  | Yes | LOV: INV.COUNTING.COUNT\_DIMENSION |  |
| countType | Counting type |  | Yes | LOV: INV.COUNTING.COUNT\_TYPE |  |
| countTimeStr | Counting time |  | Yes | When countType=MONTH, limit format: yyyy-mm; when countType=YEAR, limit format: yyyy |  |
| countMode | Counting mode |  | Yes | LOV: INV.COUNTING.COUNT\_MODE |  |
| counterIds | Counter ids |  | Yes |  |  |
| supervisor | supervisor |  | Yes |  |  |

### Counting order execute method

#### Program1(5): counting order execute method(InvCountHeaderController)

Parameter: List<InvCountHeaderDTO>

Return: List<InvCountHeaderDTO>

Method name: execute

|  |
| --- |
| 1. Update the counting order status to "In counting"  2. Get stock data to generate row data: Query the stock data that the on hand quantity is not 0 according to the tenantId + companyId + departmentId + warehouseId + snapshotMaterialIds + snapshotBatchIds on the table header, Summarize according to the counting dimension and save the data in the counting order line table. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field logic** | | | | | |
| **Counting Order Header** | | | | | |
| **Field Name** | **Description** |  | **Is necessary**  **(from input)** | **Value logic** |  |
| countStatus | Document status |  | - | The status is updated to "in counting" |  |
| **Counting Order Line** | | | | | |
| **Field Name** | **Description** |  | **Is necessary**  **(from stock)** | **Value logic** |  |
| tenantId | Tenant ID |  | - | current tenant id |  |
| countLineId | Table ID |  | - | Auto increment |  |
| countHeaderId | Count header ID, association  fexam\_inv\_count\_header |  | - | Get from header data |  |
| lineNumber | Line number |  | - | Natural serial number, start from 1, generated by code |  |
| warehouseId | Warehouse ID |  | - | If the counting dimension = SKU, summarize by material;  If the counting dimension = LOT, summarize by material + batch; |  |
| materialId | Material ID |  | - |  |
| materialCode | Material code |  | - |  |
| unitCode | Unit code |  | - |  |
| batchId | Batch ID |  | - |  |
| snapshotUnitQty | Snapshot unit quantity |  | - | Stock table summary value |  |
| unitQty | Counting result unit quantity |  | - | Default is null |  |
| unitDiffQty | Counting result unit difference quantity |  | - | Default is null |  |
| counterIds | Counter ids |  | - | Default is the counterIds on the counting order header. |  |

### Counting order synchronization WMS method

#### Program1(12): Counting order synchronization WMS method(InvCountHeaderController)

Parameter: List<InvCountHeaderDTO>

Return: InvCountInfoDTO

Method name: countSyncWms

1. Pre-verification(pseudocode)

|  |
| --- |
| // Query warehouse data based on warehouseId  For(invCountHeader:invCountHeaderList) {  // a. Determine whether the warehouse data exists by tenantId and warehouseCode  // b. Get the extended table data based on the counting header ID  extraRepository.select(sourceId=countHeaderId and enabledFlag=1);  // if can not get the result, need to initialize  Extra syncStatusExtra = new Extra(tenantId, sourceId=countHeaderId, enableFlag=1, programKey="wms\_sync\_status");  Extra syncMsgExtra = new Extra(tenantId, sourceId=countHeaderId, enableFlag=1, programKey="wms\_sync\_error\_message");  // c. Determine whether it is a WMS warehouse and call the WMS interface to synchronize the counting order  If(isWMSwarehouse) {  // Construct WMS system interface parameters  {  "countHeaderId":1,  …  "countOrderLineList":[{  "countLineId":1,  …  },{  "countLineId":2,  …  }],  "employeeNumber":"21995"  }  // Call interface and get result status  callWmsApiPushCountOrder(namespace="HZERO", serverCode="FEXAM\_WMS", interfaceCode="fexam-wms-api.thirdAddCounting");  if(response.getString("returnStatus") = "S") {  syncStatusExtra.setProgramValue("SUCCESS");  syncMsgExtra.setProgramValue("");  // record WMS document number  invCountHeader.setRelatedWmsOrderCode(response.getString("code"));  }else {  syncStatusExtra.setProgramValue("ERROR");  syncMsgExtra.setProgramValue(response.getString("returnMsg"));  }  } else {  syncStatusExtra.setProgramValue("SKIP");  }  ~~// d. record WMS document number~~  ~~invCountHeader.setRelatedWmsOrderCode(response.getString("code"));~~  }  // Construct the return value InvCountInfoDTO  // If errorList not empty, need to rollback(equals not execute the document), at the same time, need to submit the transaction about insert or update extraList data. |

### Counting Result Synchronous method

#### Program1(5): Counting result synchronous method(InvCountHeaderController)

Parameter: InvCountHeaderDTO(Request Body)

Return: InvCountHeaderDTO

Method name: countResultSync

1. Mandatory input verification(Reference table - field logic)
2. Pre-verification(pseudocode)

|  |
| --- |
| // input  {  "countHeaderId":1,  …  "countOrderLineList":[{  "countLineId":1,  …  },{  "countLineId":2,  …  }]  }  // Query the database based on counting order header data  // Determine whether the warehouse is a WMS warehouse  If(not a WMS warehouse) {  errorMsg = "The current warehouse is not a WMS warehouse, operations are not allowed";  status="E";  }  // Check data consistency  If(The number of rows is inconsistent or lineId can not match) {  errorMsg = "The counting order line data is inconsistent with the INV system, please check the data";  status="E";  }  // if not error, set status = "S";  // Update the line data  updateLines(including unitQty,unitDiffQty,remark) |

#### Program2(2): Register external interface

service code: FEXAM\_INV

request mode: POST

interface code: fexam-<yourEmployeeId>-inv-api.resultSync

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field logic** | | | | | |
| **Counting Order Header** | | | | | |
| **Field Name** | **Description** |  | **Is necessary**  **(from input)** | **Value logic** |  |
| tenantId | Tenant Id |  | Yes |  |  |
| countNumber | Document number |  | Yes |  |  |
| **Counting Order Line** | | | | | |
| **Field Name** | **Description** |  | **Is necessary**  **(from stock)** | **Value logic** |  |
| tenantId | Tenant ID |  | Yes |  |  |
| countLineId | Table ID |  | Yes |  |  |
| unitQty | Counting result unit quantity |  | Yes | Allow update |  |
| unitDiffQty | The main unit difference quantity after counting |  | No | Automatic calculation(unitQty - snapshotUnitQty) |  |
| remark | Remark |  | No | Allow update |  |

### Counting order submit verification method

#### Program1(3): counting order submit verification method(InvCountHeaderController)

Parameter: List<InvCountHeaderDTO>

Return: InvCountInfoDTO

Method name: submitCheck

1. Pre-verification(pseudocode)

|  |
| --- |
| Requery the database based on the input document ID  // 1. Check document status  The operation is allowed only when the status in in counting, processing, rejected, withdrawn.  // 2. current login user validation  Only the current login user is the supervisor can submit document.  // 3. Data integrity check  Determine whether there is data with null value unit quantity field   * When exists, verification failed, prompt: "There are data rows with empty count quantity. Please check the data." * When not exists, verification passed   When there is a difference in counting, the reason field must be entered. |

### Counting order submit method

#### Program1(3): Counting order submit method

Parameter: List<InvCountHeaderDTO>

Return: List<InvCountHeaderDTO>

Method name: submit

1. Pre-verification(pseudocode)

|  |
| --- |
| // Get configuration file value  String workflowFlag = profileClient.getProfileValueByOptions(tenantId, null, null,"FEXAM95.INV.COUNTING.ISWORKFLOW");  // Determine whether to start workflow  If(workflowFlag) {  // start workflow  }else {  // update document status to confirmed  } |

#### Program2(3): Process configuration

|  |  |  |  |
| --- | --- | --- | --- |
| **Process classification definition** | | | |
| **Type** | **Code** | **Name** | **Description** |
| directory | FEXAM\_INV\_202411 | final exam 202411 |  |
| Process classification | INV\_COUNT95 | Counting order 95 |  |
| **Events** | | | |
| **Coding** | **Name** | **Interface code** | **Interface name** |
| START\_CALLBACK | Start callback | INV\_COUNT95\_CALLBACK | INV\_COUNT95\_CALLBACK |
| FINISH\_CALLBACK | Finish callback | INV\_COUNT95\_CALLBACK | INV\_COUNT95\_CALLBACK |
| CANCEL\_CALLBACK | Cancel callback | INV\_COUNT95\_CALLBACK | INV\_COUNT95\_CALLBACK |
| **Process definition** | | | |
| **Code** | **Name** | **Classification** | **Description** |
| INV\_COUNT95\_RESULT\_SUBMIT | Counting result submit 95 | final exam 202411 |  |

#### Program3(3): Counting order approval callback

Parameter: WorkFlowEventDTO

Return: InvCountHeaderDTO

|  |
| --- |
| 1. Define the workflow event input DTO  /\*\*  \* Workflow event input dto  \*/ @Data public class WorkFlowEventDTO {  /\*\*  \* business key  \*/  private String businessKey;   /\*\*  \* document status  \*/  private String docStatus;   /\*\*  \* workflow ID  \*/  private Long workflowId;   /\*\*  \* approved time  \*/  private Date approvedTime; }  2. Define approval callback API  Reference table for input parameters, update documents according to input parameters, update supervisor to current operator when process starts. |

|  |  |  |
| --- | --- | --- |
| **Field** | **Description** | **Value** |
| businessKey | Business key | Process variable: Business key |
| docStatus | Document status | Starting: PROCESSING  Ending: APROVED/REJECTED  Withdraw: WITHDRAWN |
| workflowId | Workflow Id | Process variable: Instance Id |
| approvedTime | Approved time | Process variable: CurrentDateTime |

#### Program4(3): Flow chart design

When departmentCode=D02, two approval nodes are required, including yourself and me; When departmentCode=D03, one approval nodes are required, including only yourself; in other cases, one approval nodes are required, including only yourself, and self-approval is enabled

### Counting order report dataset method

#### Program1(5): Counting order report dataset method

Parameter: InvCountHeaderDTO

Return: List<InvCountHeaderDTO>

Method name: countingOrderReportDs

#### Template(3)

表格

描述已自动生成

## Appendix

Omit

## Open and closed issues

### Open issues

| Seq | Issues | Solution | person in charge | target date | actual date |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Closed issues

| Seq | Issues | Solution | person in charge | target date | actual date |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |