# Audit Task Management Panel APIs

## GET /audit/work-tasks

### CURL

curl --location --request GET 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/work-tasks' \

--header 'warehouse\_id: 7' \

--header 'user\_id: 101'

### Request

#### Headers

* **user\_id**
* **warehouse\_id**

#### Query Parameters

* **status** [optional]: Only tasks of this status will be returned
* **assigned\_to** [optional]: Only tasks assigned to this user will be returned

### Response

{

status: string; (‘CREATED’, ‘ASSIGNED’, ‘IN\_PROGRESS’, ‘COMPLETED’)

message: string;

data: {

workTasks: Array<{

id: number;

status: string;

assignedToName?: string;

workOrders: Array<{

workOrderId: number;

rackName: string;

}>;

createdAt: number;

}>;

auditors: Array<{

id: number;

name: string;

}>;

};

};

### Access

All users of the same warehouse

## POST /audit/assign-task

Returns an error if the auditor already has the maximum number of work tasks assigned to them. The maximum number of tasks is read from table purplle\_purplle2.ext\_config from the key max\_work\_tasks\_per\_auditor

### CURL

curl --location 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/assign-task' \

--header 'warehouse\_id: 7' \

--header 'user\_id: 811114008' \

--header 'Content-Type: application/json' \

--data '{

"work\_task\_id": 4,

"auditor\_id": 535

}'

### Request

#### Headers

* **userId**
* **warehouseId**

#### Request Body

* **work\_task\_id**
* **auditor\_id**

### Response

{

status: string;

message: string;

data: {

workTaskId: number;

auditorId: number;

};

}

### Access

User with supervisor role of this warehouse

Note:

# 

# **📦 HDD APIs Documentation**

## **1. GET /audit/work-task**

### **Description**

Retrieves an audit work task assigned to the auditor based on status priority (IN\_PROGRESS, ASSIGNED) from audit\_wt\_master.

If the work task status is 'ASSIGNED', all RACKS associated with that task must have a 'FREE' status in warehouse\_bulk\_rack. If the work task status is 'IN\_PROGRESS', the RACKS' status should be 'AUDITING\_IN\_PROGRESS'.

### **Tables**

audit\_wt\_master, audit\_wo\_master, warehouse\_bulk\_rack

### **Request**

**Headers**

| **Key** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| user\_id | string | Yes | Auditor's user ID |
| warehouse\_id | string | Yes | Warehouse identifier |

**Query Parameters:** None

### **CURL**

curl --location 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/work-task' \

--header 'warehouse\_id: 7' \

--header 'user\_id: 811114008'

### **Response**

****{

"status": "success",

"message": "Task fetched successfully",

"data": {

"workTaskId": 123,

"auditor": {

"id": 456,

"name": "Alex"

}

"racks": ["R1", "R2"]

}

}

### If there are no pending tasks:

****{

"status": "success",

"message": "There are no pending work tasks assigned to you"

}

### If all tasks are blocked:

****{

"status": "success",

"message": "All your work tasks are blocked"

}



## **2. POST /audit/work-task/start**

### **Description**

* The status of all racks in warehouse\_bulk\_rack related to the work task must be "FREE" if the task status is "ASSIGNED." If the task status is "IN\_PROGRESS," the rack status must be "AUDITING\_IN\_PROGRESS." Otherwise, an error will be returned.
* All racks associated with the task must have their status updated to "AUDITING\_IN\_PROGRESS."
* The task status in audit\_wt\_master will be updated to "IN\_PROGRESS."
* Retrieves all work orders associated with the task based on status priority (IN\_PROGRESS, WT\_ASSIGNED, QUEUED) and then based on sequence (column from the table warehouse\_bulk\_rack) from audit\_wo\_master.

### 

### **Tables**

audit\_wt\_master, audit\_wo\_master, warehouse\_bulk\_rack

### **Request**

**Headers**

| **Key** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| user\_id | string | Yes | Auditor's user ID |
| warehouse\_id | string | Yes | Warehouse identifier |

**Body**

| **Field** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| work\_task\_id | number | Yes | ID of the task |

### **CURL**

curl --location 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/work-task/start' \

--header 'warehouse\_id: 7' \

--header 'user\_id: 811114008' \

--header 'Content-Type: application/json' \

--header 'Cookie: GCLB=CMqpx-CippWaDBAD' \

--data '{

"work\_task\_id": 31

}'

### **Response**

****{

"status": "success",

"message": "Task started",

"data": {

"workOrders": [

{

"status": "WT\_ASSIGNED",

"work\_order\_id": 101,

"rack\_name": "R1"

}

]

}

}



## **3. POST /audit/work-order/skip**

### **Description**

Status of Work order in audit\_wo\_master is updated to “QUEUED”

### **Tables**

audit\_wo\_master

### **Request**

**Headers**

| **Key** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| user\_id | string | Yes | Auditor's user ID |
| warehouse\_id | string | Yes | Warehouse identifier |

**Body**

| **Field** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| work\_order\_id | number | Yes | ID of the work order |

### **CURL**

curl --location 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/work-order/skip' \

--header 'warehouse\_id: 2' \

--header 'user\_id: 811114008' \

--header 'Content-Type: application/json' \

--data '{

"work\_order\_id": 64

}'

### **Response**

****{

"status": "success",

"message": "Work order skipped"

}



## **4. POST /audit/work-order/start**

### **Description**

* Update the status of work order in audit\_wo\_master to “IN\_PROGRESS”
* Create entry for all the boxes associated with the work order in audit\_wo\_box table with status “PENDING” using warehouse\_bulk\_rack\_inout and procurement\_bulk\_inventoryitem table
* All the boxes associated with the work order will be returned in the response.

### **Tables**

audit\_wo\_master, warehouse\_bulk\_rack\_inout, procurement\_bulk\_inventoryitem, audit\_wo\_box

### **Request**

**Headers**

| **Key** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| user\_id | string | Yes | Auditor's user ID |
| warehouse\_id | string | Yes | Warehouse identifier |

**Body**

| **Field** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| workTaskId | number | Yes | ID of the task |
| workOrderId | number | Yes | ID of the work order |

### **CURL**

curl --location 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/work-order/start' \

--header 'warehouse\_id: 5' \

--header 'user\_id: 811114008' \

--header 'Content-Type: application/json' \

--data '{

"workTaskId": 31,

"workOrderId": 66

}'

### **Response**

****{

"status": "success",

"message": "Work order started",

"data": {

"boxDetails": [

{

"boxCode": "BX001",

"status": "pending"

}

]

}

}



## **5. POST /audit/work-order/box/scan**

### **Description**

* Validates if the box is valid or not in the procurement\_bulk\_inventoryitem table using box\_code.
* If the box is associated with work order and there is entry in work\_order\_box table then retrieve the details using warehouse\_bulk\_rack\_inout and procurement\_bulk\_inventoryitem and return in response
* If box is not associated with work order it should satisfy the config “RACK\_TYPE\_CONFIG” in ext\_config\_table which is an object with key that is “type” from warehouse\_bulk\_rack table and value is object with keys “skuLimit” and “isEnabled”, “skuLimit” value is integer which is the the maximum count of boxes of that type that can be there in work order, so need to check in work\_order\_box table if the count of same type box should not exceed the skuLimit and this check only needs to be performed if “isEnabled” value is true, error will be returned only if “isEnabled” value is true and skuLimit is exceeded else an entry of the box is is created in work\_order\_box and isExtra is set to 1
* If valid, retrieve details using warehouse\_bulk\_rack\_inout and procurement\_bulk\_inventoryitem
* Also fetch isEanVerify key from config “dc-audit-config” in ext\_config table whose value is an object with key “isEanVerify” and value boolean and return in response

### **Tables**

audit\_wo\_master, warehouse\_bulk\_rack\_inout, procurement\_bulk\_inventoryitem, warehouse\_bulk\_rack

### **Request**

**Headers**

| **Key** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| user\_id | string | Yes | Auditor's user ID |
| warehouse\_id | string | Yes | Warehouse identifier |

**Body**

| **Field** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| boxCode | string | Yes | Box identifier |
| workOrderId | number | Yes | Work order ID |
| workTaskId | number | Yes | Task ID |

### **CURL**

curl --location 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/work-order/box/scan' \

--header 'warehouse\_id: 7' \

--header 'Content-Type: application/json' \

--data '{

"boxCode": "ZZ5",

"workOrderId": 28,

"workTaskId": 13

}'

### **Response**

****{

"status": "success",

"message": "Box scanned",

"data": {

"isEanVerify": true,

"Sku": "SKU123",

"desc": "Sample Product",

"ean": "8901234567890",

"batch": "B123",

"mrp": 199.0,

"mfg": 202401,

"exp": 202501,

"qty": 10

}

}



## **6. POST /audit/work-order/box/confirm**

### **Description**

For each box:

* **Wrong Location:** If the box does not belong to the work order, an entry will be created in `audit\_wo\_box` with isExtra 1 and an entry will be generated in `audit\_flags\_log` table with the type "WRONG\_LOCATION", old value will be the rack\_id in warehouse\_bulk\_rack\_inout which we can get using id obtained from procurement\_bulk\_inventoryitem using box\_code and then querying against inventoryitem\_id in warehouse\_bulk\_rack\_inout table else blank and new value will be the rack\_id of the work order from audit\_wo\_master.
* **Batch Discrepancy:** Any discrepancies in MRP or manufacturing date, or expiry date will be flagged as "BATCH\_DISCREPANCY" where one entry will be created for each change in MRP, mfg and expiry, old value will be the value that exist in procurement\_bulk\_inventoryitem and new value will be the value entered by user
* **Quantity Discrepancy:**
  + If the quantity is less than expected, it will be flagged as "LOST" where old value will be qty\_left in procurement\_bulk\_inventoryitem table and new value will be the value by user.
  + If the quantity is more than expected, it will be flagged as "OVERAGE\_EXPECTED\_SKU\_TO\_CRATE" where old value will be qty\_left in procurement\_bulk\_inventoryitem table and new value will be the value by user.
* **Damaged Quantity:** If there is any damaged quantity, a "DAMAGED\_TO\_CRATE" flag will be created where old value will be 0 and new value will be entered by user.
* **EAN Mismatch:** An EAN mismatch will generate a "WRONG\_SKU\_PRODUCT" with old value from ean in product\_product table and new value entered by the user flag.
* The status of the box in audit\_bo\_box will be update to “COMPLETED”

### **Tables**

audit\_wo\_master, warehouse\_bulk\_rack\_inout, procurement\_bulk\_inventoryitem, audit\_flags\_log, audit\_wo\_box, product\_product

### **Request**

**Headers**

| **Key** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| user\_id | string | Yes | Auditor's user ID |
| warehouse\_id | string | Yes | Warehouse identifier |

**Body**

| **Field** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| boxCode | string | Yes | Box identifier |
| ean | string | Yes | Product EAN |
| qty | number | Yes | Confirmed quantity |
| damagedQty | number | Yes | Damaged quantity |
| mrp | number | Yes | MRP value |
| mfg | number | Yes | Manufacturing date |
| exp | number | Yes | Expiry date |

### **CURL**

curl --location 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/work-order/box/confirm' \

--header 'warehouse\_id: 7' \

--header 'user\_id: 535' \

--header 'Content-Type: application/json' \

--data '{

"boxCode": "ZZ5",

"workOrderId": 28,

"workTaskId": 13,

"isEanVerify": true,

"eanCode": "PPLB1296NYBAE97",

"qtyLeft": 200,

"mrp": 110,

"mfgDate": 1682879400,

"expiryDate": 1753986600,

"damagedQty": 0,

"crate": "C001"

}'

### **Response**

****{

"status": "success",

"message": "Box confirmed"

}



## **7. POST /audit/work-order/submit**

### Description

* Verify the status of all the boxes associated with work order in audit\_wo\_box boxes are “COMPLETED” for the work order, If any box status is still “PENDING” mark it “MISSING”, also flag “WRONG\_LOCATION” will be generated for each box missing in audit\_flag\_log table with old value will be the rack\_id in audit\_wo\_master table for that work order
* Update the status of work order in audit\_wo\_master table to “COMPLETED”
* Get the work\_task of the work order from audit\_wt\_master table and then get all the work orders of that task from audit\_wo\_master, Checks if status of all the work orders of the task is “COMPLETED”, If yes, updates the status of work task in audit\_wt\_master table

### **Tables**

audit\_wo\_master, audit\_wt\_master, audit\_flags\_log, audit\_wo\_box

### **Request**

**Headers**

| **Key** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| user\_id | string | Yes | Auditor's user ID |
| warehouse\_id | string | Yes | Warehouse identifier |

**Body**

| **Field** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| workTaskId | number | Yes | Task ID |
| workOrderId | number | Yes | Work order ID |

### **CURL**

curl --location 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/work-order/submit' \

--header 'warehouse\_id: 7' \

--header 'user\_id: 535' \

--header 'Content-Type: application/json' \

--header 'Cookie: GCLB=CL\_Vib37943hDhAD' \

--data '{

"workOrderId": 28,

"workTaskId": 13

}'

### **Response**

****{

"status": "success",

"message": "Work order submitted"

}



# **📦 Flag Panel APIs Documentation**

## **1. GET /audit/flags**

### Description

* Will return entries from table audit\_flag\_logs with additional information.
* Will not return entries of status "RESOLVED" or "REJECTED" that were created more than 2 days ago.
* Will only return the flags of the user’s warehouse

### Tables

audit\_flags\_log, audit\_wo\_master, audit\_wo\_box

### Request

**Headers**

| **Key** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| user\_id | string | Yes | Auditor's user ID |
| warehouse\_id | string | Yes | Warehouse identifier |

**Query**

| **Field** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| type | string | No | Filter by flag type |
| status | string | No | Filter by status |
| crateId | string | No | Filter by crateId |
| timeStart | number | No | Filter flags created after this time. It should be a Unix timestamp in seconds |
| timeEnd | number | No | Filter flags created before this time. It should be a Unix timestamp in seconds |

### CURL

curl --location 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/flags' \

--header 'warehouse\_id: 7' \

--header 'user\_id: 535' \

### Response

****{

"status": "success",

"message": "Flags retrieved successfully",

"data": {

"flags": [

{

"id": 1,

"type": "LOST",

"identifier": "WT-001",

"sku": "S1-XYZ123",

"details": {

"qty": 2,

"boxCode": "B1"

},

"createdAt": 1757097000,

"status": "RESOLVED"

},

{

"id": 2,

"type": "OVERAGE\_EXPECTED\_SKU\_TO\_CRATE",

"identifier": "WT-002",

"sku": "S2-XSLFDKL",

"details": {

"qty": 3,

"isRecoveryCandidate": true

},

"createdAt": 1757097000,

"status": "PENDING"

},

{

"id": 3,

"type": "DAMAGED\_TO\_CRATE",

"identifier": "WT-003",

"sku": "S3-S238GF8D",

"details": {

"qty": 7

},

"createdAt": 1757097000,

"status": "PENDING"

},

{

"id": 4,

"type": "BATCH\_DISCREPANCY",

"identifier": "WT-004",

"sku": "S4-SDFSIU",

"details": {

"expected": {

"mrp": 129,

"mfgDate": 1722513600,

"expiryDate": "1816948800"

},

"found": {

"mrp": 110,

"mfgDate": 1682879400,

"expiryDate": 1753986600

}

},

"createdAt": 1757097000,

"status": "REJECTED"

}

]

}

}

Note: data.flags[\*].status can be one of "PENDING", "RESOLVED" or "REJECTED".

## **2. POST /audit/flags/reject**

### Description

* Update the status of flag in audit\_flag\_logs to “REJECTED”
* Add the reason in an array of object with key reason because there can be multiple reasons for a flag while resolving

### Tables

audit\_flag\_logs

### Request

**Headers**

| **Key** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| user\_id | string | Yes | Auditor's user ID |
| warehouse\_id | string | Yes | Warehouse identifier |

**Body**

| **Field** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| flagId | number | Yes | ID of the flag |
| reason | string | Yes | Reason for rejecting |

### CURL

curl --location 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/flags/reject' \

--header 'user\_id: 535' \

--header 'warehouse\_id: 7' \

--header 'Content-Type: application/json' \

--data '{"flagId": 6, "reason": "reason"}'

### Response

****{

"status": "success",

"message": "Flag Rejected",

}



## **3. POST /audit/flags/resolve**

### Description

For each flag type:

* **LOST:** If the flag type is “LOST”, Following Functions to be called sequentially File name: gon-outward.service.ts Functions: createGon, submitGonItcV2 (they need to be called sequentially)
  + In creatGon, picklist\_id is optional so we need not send also gon\_type will be “lost”, gon\_comment can be blank for now.
  + In submitGonItcV2, picklist\_id and skip\_sap is optional so we need not send
  + After the success of both functions we can update the flag status to “RESOLVED” in audit\_flag\_log
* **DAMAGED\_TO\_CRATE:** If the flag type is “DAMAGED\_TO\_CRATE” then in input we will be getting reasons array which will be array of objects with each object having key “reason” and “quantity” where “reason” will be string and “quantity” will be number.
  + We will sum up the quantity and following function to be called similar to in “LOST” case with gon\_type “DAMAGED” i.e.., createGon and submitGonItcV2 and no\_of\_qty will be the total quantity calculated from the reasons array
  + If There is difference in the quantity calculated and new\_value of flag entry, then for remaining quantity we will create a work order in audit\_wo\_master similar to work order for this flag which we can find using wo\_box\_id which is id in audit\_wo\_box table and from there we can use wo\_id to get the work\_order

### Tables

audit\_flag\_logs, audit\_wo\_box, audit\_wo\_master

### Request

**Headers**

| **Key** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| user\_id | string | Yes | Auditor's user ID |
| warehouse\_id | string | Yes | Warehouse identifier |

**Body**

| **Field** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| flagId | number | Yes | ID of the flag |
| reasons | Array of object | Yes | Reasons if any |
| batchResolution |  |  |  |

### CURL

* LOST

curl -X POST 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/flags/resolve' \

-H "Content-Type: application/json" \

-H "user\_id: 456" \

-H "warehouse\_id: 789" \

-d '{

"flagId": 123

}'



* DAMAGED\_TO\_CRATE

curl -X POST 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/flags/resolve' \

-H "Content-Type: application/json" \

-H "user\_id: 456" \

-H "warehouse\_id: 789" \

-d '{

"flagId": 123,

"reasons": [

{

"reason": "Damaged during transit",

"quantity": 1

},

{

"reason": "Expired product",

"quantity": 1

},

{

"reason": "Water damage",

"quantity": 1

}

]

}'



* OVERAGE\_EXPECTED\_SKU\_TO\_CRATE(EXCESS)

curl -X POST 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/flags/resolve' \

-H "Content-Type: application/json" \

-H "user\_id: 456" \

-H "warehouse\_id: 789" \

-d '{

"flagId": 123,

"quantity": 15

}'



* BATCH\_DISCREPANCY
  + Create New Batch

curl -X POST 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/flags/resolve' \

-H "Content-Type: application/json" \

-H "user\_id: 456" \

-H "warehouse\_id: 789" \

-d '{

"flagId": 123,

"batchResolution": {

"type": "CREATE",

"newBatchId": "BATCH789",

"mfgDate": 1640995200,

"expiryDate": 1704067200,

"remarks": "New batch created due to MRP/date discrepancy"

}

}'



* + Update Batch SWAT

curl -X POST 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/flags/resolve' \

-H "Content-Type: application/json" \

-H "user\_id: 456" \

-H "warehouse\_id: 789" \

-d '{

"flagId": 123,

"batchResolution": {

"type": "UPDATE",

}

}'



* WRONG\_SKU\_PRODUCT

curl -X POST 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/flags/resolve' \

-H "Content-Type: application/json" \

-H "user\_id: 456" \

-H "warehouse\_id: 789" \

-d '{

"flagId": 123,

"product\_sku": "SKU123456",

"batch\_id": "BATCH789"

}'



curl --location 'https://sandbox.purplle.com/wms/api/v1/dc-be/audit/flags/resolve' \

--header 'user\_id: 535' \

--header 'warehouse\_id: 7' \

--header 'Content-Type: application/json' \

--data '{"flagId": 6, "reasons": [

{"reason": "near to expiry", "quantity": 3}, {"reason": "physically damaged", "quantity": 4}]}'

### Response

****{

"status": "success",

"message": "Flag Resolved",

}



## **4. POST /audit/recovery-gon**

### Description

* Fetchs the quantity for gon recovery

### Tables

audit\_flag\_logs

### Request

**Headers**

| **Key** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| user\_id | string | Yes | Auditor's user ID |
| warehouse\_id | string | Yes | Warehouse identifier |

**Body**

| **Field** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| flagId | number | Yes | ID of the flag |
|  |  |  |  |

### CURL

curl -X POST "https://sandbox.purplle.com/wms/api/v1/dc-be/audit/recovery-gon" \

-H "Content-Type: application/json" \

-d '{

"flag\_id": 123

}'

### Response

****{

"httpStatus": 200,

"status": "success",

"message": "Recovery GON processed successfully",

"data": {

"quantity": 0

}

}

