

API Testing Guide - Purchase Requisition System

Quick Start Testing

Prerequisites

1. Server running on `http://localhost:3001`
2. Login to get JWT token
3. Use token in Authorization header for protected endpoints

Step 1: Login

```
curl -X POST http://localhost:3001/api/auth/login \
-H "Content-Type: application/json" \
-d '{
  "username": "john.banda",
  "password": "password123"
}'
```

Save the token from response:

```
{{
  "success": true,
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refreshToken": "...",
  "user": { "id": 1, "role": "initiator", ... }
}}
```

For subsequent requests, use:

```
TOKEN="your token here"
```

Step 2: Create Requisition (Draft)

```
curl -X POST http://localhost:3001/api/requisitions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $TOKEN" \
-d '{
  "description": "Laptops and accessories for IT team",
  "delivery_location": "IT Office, 3rd Floor",
  "urgency": "High",
  "required_date": "2025-11-15",
  "account_code": "IT-2025-Q4",
  "created_by": 1,
  "items": [
    {
      "item_name": "Dell Latitude 5520 Laptop",
      "quantity": 3,
      "specifications": "i7-11th Gen, 16GB RAM, 512GB SSD"
    },
    {
      "item_name": "Wireless Mouse",
      "quantity": 3,
      "specifications": "Logitech M330"
    }
  ]
}'
```

Response:

```
{{
  "success": true,
  "requisition_id": 1,
  "req_number": "KSB-IT-JB-20251023124500"
}}
```

Save the requisition_id:

```
REQ_ID=1
```

Step 3: View Requisition

```
curl -X GET http://localhost:3001/api/requisitions/$REQ_ID \
-H "Authorization: Bearer $TOKEN"
```

Step 4: Add Another Item

```
curl -X POST http://localhost:3001/api/requisitions/$REQ_ID/items \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $TOKEN" \
-d '{
  "item_name": "USB-C Docking Station",
  "quantity": 2,
  "specifications": "Dell WD19, 130W"
}'
```

Step 5: Submit for Approval

```
curl -X PUT http://localhost:3001/api/requisitions/$REQ_ID/submit \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $TOKEN" \
-d '{
  "user_id": 1
}'
```

Status changes from draft to pending_hod

Step 6: HOD Login

```
curl -X POST http://localhost:3001/api/auth/login \
-H "Content-Type: application/json" \
-d '{
  "username": "mary.mwanza",
  "password": "password123"
}'
```

Save HOD token:

```
HOD_TOKEN="hod_token_here"
```

Step 7a: HOD Approves

```
curl -X PUT http://localhost:3001/api/requisitions/$REQ_ID/hod-approve \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $HOD_TOKEN" \
-d '{
  "user_id": 2,
  "approved": true,
  "comments": "Approved. Budget allocated for Q4 IT upgrades."
}'
```

Status changes to hod_approved

Step 7b: HOD Rejects (Alternative)

```
curl -X PUT http://localhost:3001/api/requisitions/$REQ_ID/hod-approve \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $HOD_TOKEN" \
-d '{
  "user_id": 2,
  "approved": false,
  "comments": "Budget constraints. Please reduce quantity or find cheaper alternatives."
}'
```

Status changes to rejected

Step 8: Admin Redirection (Optional)

Login as admin:

```
curl -X POST http://localhost:3001/api/auth/login \
-H "Content-Type: application/json" \
-d '{
  "username": "admin",
  "password": "admin123"
}'
```

Redirect requisition:

```

ADMIN_TOKEN="admin_token_here"

curl -X POST http://localhost:3001/api/requisitions/$REQ_ID/redirect \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $ADMIN_TOKEN" \
-d '{
  "from_user_id": 2,
  "to_user_id": 4,
  "reason": "Original HOD on medical leave. Redirecting to Acting HOD (Sarah Banda).",
  "stage": "hod_approval"
}'

```

Step 9: Procurement Login

```

curl -X POST http://localhost:3001/api/auth/login \
-H "Content-Type: application/json" \
-d '{
  "username": "james.phiri",
  "password": "password123"
}'

```

Save procurement token:

```
PROC_TOKEN="procurement_token_here"
```

Step 10: Get Vendors List

```

curl -X GET http://localhost:3001/api/vendors \
-H "Authorization: Bearer $PROC_TOKEN"

```

Response:

```
[
  { "id": 1, "name": "Tech Solutions Ltd", ... },
  { "id": 2, "name": "Office Supplies Co", ... },
  { "id": 3, "name": "Hardware Plus", ... }
]
```

Step 11: Update Item with Vendor and Pricing

Get item IDs first:

```

curl -X GET http://localhost:3001/api/requisitions/$REQ_ID \
-H "Authorization: Bearer $PROC_TOKEN"

```

Update first item:

```

curl -X PUT http://localhost:3001/api/requisitions/$REQ_ID/items/1 \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $PROC_TOKEN" \
-d '{
  "unit_price": 5500.00,
  "vendor_id": 1
}'

```

Total Price calculated automatically: $3 \times 5500 = K16,500.00$

Update second item:

```

curl -X PUT http://localhost:3001/api/requisitions/$REQ_ID/items/2 \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $PROC_TOKEN" \
-d '{
  "unit_price": 150.00,
  "vendor_id": 2
}'

```

Total Price: $3 \times 150 = K450.00$

Update third item:

```

curl -X PUT http://localhost:3001/api/requisitions/$REQ_ID/items/3 \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $PROC_TOKEN" \
-d '{
  "unit_price": 850.00,
  "vendor_id": 1
}'

```

Total Price: $2 \times 850 = K1,700.00$

Grand Total: $16,500 + 450 + 1,700 = K18,650.00$

Step 12: Complete Procurement

```
curl -X PUT http://localhost:3001/api/requisitions/$REQ_ID/procurement \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $PROC_TOKEN" \
-d '{
  "user_id": 3,
  "comments": "All vendors confirmed. Delivery scheduled for Nov 10, 2025."
}'
```

Status changes to procurement_completed

Step 13: Generate PDF

Any authenticated user can download:

```
curl -X GET http://localhost:3001/api/requisitions/$REQ_ID/pdf \
-H "Authorization: Bearer $TOKEN" \
--output "Requisition_${REQ_ID}.pdf"
```

File downloaded: Requisition_1.pdf

Testing Error Cases

1. Reject without comments (should fail)

```
curl -X PUT http://localhost:3001/api/requisitions/$REQ_ID/hod-approve \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $HOD_TOKEN" \
-d '{
  "user_id": 2,
  "approved": false
}'
```

Expected Error:

```
{
  "error": "Comments required when rejecting a requisition"
}
```

2. PDF on draft (should fail)

```
curl -X GET http://localhost:3001/api/requisitions/$REQ_ID/pdf \
-H "Authorization: Bearer $TOKEN"
```

Expected Error:

```
{
  "error": "PDF can only be generated for approved requisitions",
  "current_status": "draft"
}
```

3. Missing required fields

```
curl -X POST http://localhost:3001/api/requisitions/$REQ_ID/items \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $TOKEN" \
-d '{
  "specifications": "Some specs"
}'
```

Expected Error:

```
{
  "error": "Item name and quantity are required"
}
```

Complete Workflow Test Script

Save this as test_workflow.sh:

```
#!/bin/bash

BASE_URL="http://localhost:3001"

# Colors
GREEN='\033[0;32m'
RED='\033[0;31m'
NC='\033[0m' # No Color

echo "==== Purchase Requisition System - Full Workflow Test ==="

# 1. Login as Initiator
echo -e "${GREEN}1. Login as Initiator${NC}"
LOGIN_RESP=$(curl -s -X POST $BASE_URL/api/auth/login \
```

```

-H "Content-Type: application/json" \
-d '{"username":"john.banda","password":"password123"})'

TOKEN=$(echo $LOGIN_RESP | grep -o '"token":[^"]*' | cut -d'"' -f4)
echo "Token: ${TOKEN:0:20}..."

# 2. Create Requisition
echo -e "\n${GREEN}2. Create Requisition${NC}"
CREATE_RESP=$(curl -s -X POST $BASE_URL/api/requisitions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $TOKEN" \
-d '{
  "description": "Test requisition",
  "delivery_location": "IT Office",
  "urgency": "High",
  "required_date": "2025-11-15",
  "account_code": "IT-2025-TEST",
  "created_by": 1,
  "items": [
    {"item_name": "Laptop", "quantity": 2, "specifications": "Dell i7"),
    {"item_name": "Mouse", "quantity": 2, "specifications": "Wireless"}
  ]
}')
REQ_ID=$(echo $CREATE_RESP | grep -o '"requisition_id":[^"]*' | cut -d':' -f2)
echo "Created Requisition ID: $REQ_ID"

# 3. Submit
echo -e "\n${GREEN}3. Submit for Approval${NC}"
curl -s -X PUT $BASE_URL/api/requisitions/$REQ_ID/submit \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $TOKEN" \
-d '{"user_id": 1}' | jq

# 4. Login as HOD
echo -e "\n${GREEN}4. Login as HOD${NC}"
HOD_LOGIN=$(curl -s -X POST $BASE_URL/api/auth/login \
-H "Content-Type: application/json" \
-d '{"username":"mary.mwanza","password":"password123"})')

HOD_TOKEN=$(echo $HOD_LOGIN | grep -o '"token":[^"]*' | cut -d'"' -f4)
echo "HOD Token: ${HOD_TOKEN:0:20}..."

# 5. HOD Approve
echo -e "\n${GREEN}5. HOD Approve${NC}"
curl -s -X PUT $BASE_URL/api/requisitions/$REQ_ID/hod-approve \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $HOD_TOKEN" \
-d '{"user_id": 2, "approved": true, "comments": "Test approval"}' | jq

# 6. Login as Procurement
echo -e "\n${GREEN}6. Login as Procurement${NC}"
PROC_LOGIN=$(curl -s -X POST $BASE_URL/api/auth/login \
-H "Content-Type: application/json" \
-d '{"username":"james.phiri","password":"password123"})')

PROC_TOKEN=$(echo $PROC_LOGIN | grep -o '"token":[^"]*' | cut -d'"' -f4)
echo "Procurement Token: ${PROC_TOKEN:0:20}..."

# 7. Add pricing
echo -e "\n${GREEN}7. Add Pricing to Items${NC}"
curl -s -X PUT $BASE_URL/api/requisitions/$REQ_ID/items/1 \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $PROC_TOKEN" \
-d '{"unit_price": 5500.00, "vendor_id": 1}' | jq

curl -s -X PUT $BASE_URL/api/requisitions/$REQ_ID/items/2 \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $PROC_TOKEN" \
-d '{"unit_price": 150.00, "vendor_id": 2}' | jq

# 8. Complete procurement
echo -e "\n${GREEN}8. Complete Procurement${NC}"
curl -s -X PUT $BASE_URL/api/requisitions/$REQ_ID/procurement \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $PROC_TOKEN" \
-d '{"user_id": 3, "comments": "All vendors confirmed"}' | jq

# 9. Generate PDF
echo -e "\n${GREEN}9. Generate PDF${NC}"
curl -X GET $BASE_URL/api/requisitions/$REQ_ID/pdf \
-H "Authorization: Bearer $TOKEN" \
--output "Test_Requisition_$REQ_ID.pdf"

echo -e "\n${GREEN}== Test Complete ==${NC}"
echo "PDF saved as: Test_Requisition_$REQ_ID.pdf"

```

Run the script:

```
chmod +x test_workflow.sh  
./test_workflow.sh
```

Postman Collection

Import this JSON into Postman:

```
{
  "info": {
    "name": "Purchase Requisition System",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
  },
  "variable": [
    {
      "key": "base_url",
      "value": "http://localhost:3001"
    },
    {
      "key": "token",
      "value": ""
    },
    {
      "key": "req_id",
      "value": ""
    }
  ],
  "item": [
    {
      "name": "Auth",
      "item": [
        {
          "name": "Login - Initiator",
          "request": {
            "method": "POST",
            "header": [],
            "body": {
              "mode": "raw",
              "raw": "{\n    \"username\": \"john.banda\",\n    \"password\": \"password123\"\n}",
              "options": { "raw": { "language": "json" } }
            },
            "url": { "raw": "{{base_url}}/api/auth/login" }
          }
        }
      ]
    }
  ]
}
```

For complete API documentation, see: WORKFLOW_GUIDE.md