

# Approval Workflow Guide

---

## Overview

---

The Genesis Provenance approval workflow enables multi-step approval processes for luxury assets. This allows team members to request reviews, authenticity checks, and verifications from designated approvers before assets are marked as verified.

---



## How the Approval Workflow Works

---

### 1. Approval Request Creation

Any team member can request an approval for an asset by:

1. **Navigate to Asset:** Go to `/vault/[id]` (asset detail page)
2. **Open Approvals Tab:** Click on the “Approvals” tab
3. **Click “Request Approval”**
4. **Fill out the request form:**
  - **Approval Type:** Select from:
    - Asset Verification
    - Authenticity Check
    - Value Assessment
    - Condition Review
    - Ownership Transfer
    - Documentation Review
    - Other
  - **Priority:** Low, Medium, High, or Urgent
  - **Request Notes:** Explain what needs to be reviewed
  - **Required Role:** (Optional) Specify which role can approve (Owner, Admin, Editor, Viewer)
  - **Specific Approver:** (Optional) Assign to a specific team member
  - **Due Date:** (Optional) Set a deadline

### 2. Notification System

When an approval request is created:

- **Notifications are sent to:**
  - The specific approver (if assigned)
  - All team members with the required role (if no specific approver)
  - All admins/owners (if no restrictions)
- **Notifications appear** in the bell icon on the dashboard top bar
- **Approval requests show as “Pending”** in the asset’s Approvals tab

### 3. Approval Response

Eligible approvers can respond to pending requests:

## Who Can Approve?

- The specific user assigned as approver
- Any user with the required role or higher
- Owners and admins can approve anything
- The requester **cannot** approve their own request

## Response Options:

- **Approve:** Accept the request
- **Reject:** Deny the request
- **Cancel:** (Requester only) Cancel the request

## Response Process:

1. Click on the pending approval in the Approvals tab
2. Click "Respond to Approval"
3. Select "Approve" or "Reject"
4. Add response notes (optional but recommended)
5. ☒ **Important:** Check "Update item status to verified" if:
  - The approval type is "Asset Verification" or "Authenticity Check"
  - You want the asset to be marked as verified upon approval

## 4. Status Updates - The Key to Verification

**This is the critical part:**

### When "Pending Review" Becomes "Verified"

An asset's status changes from "Pending Review" to "Verified" when:

1. ☒ **An approval request is created** with type:
  - "Asset Verification" OR
  - "Authenticity Check"
2. ☒ **An authorized approver responds** with "Approved"
3. ☒ **The approver checks the box:** "Update item status to verified"

### Example Workflow:

```

Asset Status: Pending Review
↓
User requests "Asset Verification" approval
↓
Admin/Approver reviews the asset
↓
Approver clicks "Approve" AND checks "Update item status to verified"
↓
Asset Status: Verified ✓
  
```

### Important Notes:

- If the approver **doesn't check** "Update item status to verified", the asset status remains unchanged
- Only "Asset Verification" and "Authenticity Check" approval types can change status to verified
- Other approval types (Value Assessment, Condition Review, etc.) do NOT automatically change the asset status

## 5. After Approval

Once an approval is processed:

- **✓ Status updates:** The approval shows as “Approved”, “Rejected”, or “Cancelled”
- **✓ Notifications sent:** The requester receives a notification about the outcome
- **✓ Provenance event created:** A “reviewed” event is added to the asset’s timeline
- **✓ Item status updated:** (If “Update item status” was checked and approval type was verification/authenticity)
- **✓ Approval history:** All approvals are permanently recorded in the asset’s history

## Common Use Cases

### Use Case 1: Initial Asset Verification

**Scenario:** A collector adds a new luxury watch and wants it verified.

1. **Collector uploads** the watch with photos and documentation
2. **Asset is created** with status “Pending Review”
3. **Collector requests** an “Asset Verification” approval
4. **Assigns it** to the organization’s admin/expert
5. **Admin reviews** the asset, photos, and documentation
6. **Admin approves** the request and **checks “Update item status to verified”**
7. **Asset status changes** to “Verified” ✓
8. **Asset is now** trusted and can be included in reports

### Use Case 2: Authenticity Check for High-Value Item

**Scenario:** A dealer wants a second opinion on a \$50k item.

1. **Dealer has** an asset marked as “Verified”
2. **Dealer requests** an “Authenticity Check” approval
3. **Sets priority** to “High” and adds detailed notes
4. **Assigns to** a specific team member with expertise
5. **Expert reviews** and may request additional photos via comments
6. **Expert approves** and **checks “Update item status to verified”** (reconfirms verification)
7. **Approval is recorded** in the provenance timeline




### Use Case 3: Pre-Sale Condition Review

**Scenario:** Before selling an asset, an owner wants a condition review.





1. **Owner requests** a “Condition Review” approval
2. **Sets due date** for 7 days before the planned sale
3. **Assigns to** an editor with required role “Editor”
4. **Editor provides** detailed condition notes in the response
5. **Editor approves** (does **not** check “Update item status” - not applicable)
6. **Asset status** remains “Verified”
7. **Condition review** is documented in approval history

## Permission Levels

### Requester Permissions

-  Can create approval requests for any asset in their organization
-  Can cancel their own pending requests
-  Cannot approve their own requests

### Approver Permissions

-  Can approve/reject if they are the assigned approver
-  Can approve/reject if they have the required role
-  Owners and admins can approve any request
-  Cannot approve requests from themselves





### Role Hierarchy (for required roles)

```

Owner (highest)
  ↓
Admin
  ↓
Editor
  ↓
Viewer (lowest)
  
```

If a request requires “Editor” role, both Editors, Admins, and Owners can approve it.

## Approval Status Reference

Status	Icon	Description	Actions Available
<b>Pending</b>	 Clock	Awaiting approval	Approve, Reject, Cancel (requester)
<b>Approved</b>	 Check	Request was approved	View details, Response notes
<b>Rejected</b>	 X	Request was rejected	View details, Response notes
<b>Cancelled</b>	 Alert	Requester cancelled	View details

## Step-by-Step: Making an Asset “Verified”

### Starting Point: Asset is “Pending Review”

#### Step 1: Request Verification Approval

1. Go to asset detail page (/vault/[asset-id])
2. Click "Approvals" tab
3. Click "Request Approval" button
4. Select approval type: "Asset Verification"
5. Set priority (e.g., "Medium")
6. Add request notes: "Please verify authenticity and documentation"
7. (Optional) Assign to specific team member
8. Click "Submit Request"

## Step 2: Approver Reviews Asset

1. Approver receives **notification**
2. Approver navigates **to** asset
3. Reviews:
  - Photos **and** media
  - Provenance timeline
  - AI authentication results (**if** any)
  - VIN/serial number
  - Documentation

## Step 3: Approver Approves Request

1. In Approvals **tab**, click pending approval
2. Click **"Respond to Approval"**
3. Select **"Approve"**
4. Add response **notes**: **"Verified based on documentation and AI analysis"**
5. ☒ Check **"Update item status to verified"** ☐ **CRITICAL STEP**
6. Click **"Submit Response"**

## Step 4: Asset is Now Verified

- ☒ Asset status: "Verified"
- ☒ Green badge shows in vault
- ☒ Approval recorded in history
- ☒ Provenance event created
- ☒ Requester notified

## API Endpoints

### Create Approval Request

POST /api/items/[id]/approvals

Body:

```
{
  "approvalType": "verification",
  "priority": "medium",
  "requestNotes": "Please verify this asset",
  "requiredRole": "admin", // optional
  "approverUserId": "user-id", // optional
  "dueDate": "2024-12-31" // optional
}
```

## Respond to Approval

PATCH /api/items/[id]/approvals/[approvalId]

Body:

```
{
  "status": "approved", // or "rejected" or "cancelled"
  "responseNotes": "Asset verified",
  "updateItemStatus": true // Set to true to update item status
}
```

## Get All Approvals for Item

GET /api/items/[id]/approvals

Response:

```
{
  "approvals": [
    {
      "id": "...",
      "approvalType": "verification",
      "status": "pending",
      "priority": "medium",
      "requestedBy": { "fullName": "John Doe" },
      "canApprove": true // Dynamic based on current user
    }
  ]
}
```



## Troubleshooting

### Issue: “I approved the request but the asset is still ‘Pending Review’”

#### Solution:

- Make sure you **checked the box** “Update item status to verified” when approving
- Verify the approval type was “Asset Verification” or “Authenticity Check”
- Other approval types do not automatically change the asset status

### Issue: “I can’t see the approval request”

#### Solution:

- Check if you have the required role or are the assigned approver
- Navigate to the asset’s detail page → Approvals tab
- Check your notifications (bell icon) for approval requests

### Issue: “I can’t approve my own request”

#### Solution:

- This is by design for security and accountability
- Assign the approval to another team member
- Ask an admin/owner to approve it

## Issue: “The ‘Respond to Approval’ button is disabled”

### Possible reasons:

- The approval is already processed (approved/rejected)
  - You are the requester (cannot approve own request)
  - You don't have the required role
  - You are not the assigned approver
- 



## Best Practices

---

### For Requesters

1. **✓ Be specific** in request notes - explain what needs verification
2. **✓ Set appropriate priority** - urgent for time-sensitive matters
3. **✓ Assign to experts** - use specific approvers for specialized items
4. **✓ Set due dates** - especially for pre-sale verifications
5. **✓ Add comments** if approver requests more information

### For Approvers

1. **✓ Review thoroughly** - check all photos, documents, and AI results
2. **✓ Add detailed response notes** - future reference and audit trail
3. **✓ Check “Update status”** when verifying assets (for verification types)
4. **✓ Request more info** via comments if needed before approving
5. **✓ Act promptly** on urgent and high-priority requests

### For Organizations

1. **✓ Define clear approval types** for your workflow
  2. **✓ Assign roles appropriately** - ensure qualified approvers
  3. **✓ Use required roles** for consistency (e.g., “Admin” for verifications)
  4. **✓ Monitor pending approvals** regularly
  5. **✓ Train team** on when to use each approval type
-

## Summary

---

### The Complete Verification Flow:

```
Asset Created (Pending Review)
↓
Request "Asset Verification" Approval
↓
Assign to Approver (or Role)
↓
Approver Reviews Asset
↓
Approver Approves + Checks "Update Status"
↓
Asset Status → Verified ✓
↓
Asset Shows Green Badge in Vault
```

### Key Takeaways:

1. **Approval requests enable structured review processes**
  2. **Multiple approval types support different workflows**
  3. **“Asset Verification” and “Authenticity Check” can update status**
  4. **Must check “Update item status” box when approving**
  5. **All approvals are permanently recorded in provenance**
  6. **Role-based permissions ensure proper authorization**
  7. **Notifications keep team informed of approval status**
- 

## Related Documentation

---

- [Team Management Guide](/TEAM_MANAGEMENT_GUIDE.md) (/TEAM\_MANAGEMENT\_GUIDE.md)
  - [Collaboration Features](/COLLABORATION_FEATURES_COMPLETE.md) (/COLLABORATION\_FEATURES\_COMPLETE.md)
  - [Phase 3B Roadmap](/PHASE_3B_ROADMAP.md) (/PHASE\_3B\_ROADMAP.md)
  - [Notification System](/NOTIFICATION_SYSTEM_GUIDE.md) (/NOTIFICATION\_SYSTEM\_GUIDE.md)
- 

### Questions or Issues?

If you encounter any problems with the approval workflow, check the troubleshooting section above or contact your organization administrator.