**Approval Workflows Module - Installation Guide**

**📋 Overview**

The Approval Workflows module enables:

* Multi-level approvals for PR, PO, GRN, and other documents
* Configurable workflow steps
* Role-based and user-based approvers
* Parallel and sequential approvals
* Timeout and escalation handling
* Complete audit trail
* Conditional approvals based on value thresholds

**🚀 Installation Steps**

**1. Create the Approvals App Directory Structure**

mkdir -p approvals/migrations

touch approvals/\_\_init\_\_.py

touch approvals/apps.py

touch approvals/models.py

touch approvals/admin.py

touch approvals/signals.py

touch approvals/tests.py

touch approvals/views.py

**2. Copy the Code Files**

Copy the following files from the artifacts:

1. **approvals/models.py** - Core models
2. **approvals/admin.py** - Django admin configuration
3. **approvals/apps.py** - App configuration
4. **approvals/signals.py** - Auto-create approvals
5. **approvals/init.py** - Package initialization

**3. Update Django Settings**

Edit pdl\_ims\_backend/settings.py:

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

# Third-party apps

'rest\_framework',

'corsheaders',

'django\_filters',

# Our apps

'users.apps.UsersConfig',

'core.apps.CoreConfig',

'inventory.apps.InventoryConfig',

'procurement.apps.ProcurementConfig',

'production.apps.ProductionConfig',

'assets.apps.AssetsConfig',

'approvals.apps.ApprovalsConfig', # ← ADD THIS

'reports.apps.ReportsConfig',

]

**4. Create and Run Migrations**

# Create migration files

python manage.py makemigrations approvals

# Review the migration file

# It should create 4 tables:

# - approval\_workflows

# - approval\_workflow\_steps

# - approvals

# - approval\_history

# Apply migrations

python manage.py migrate approvals

**5. Verify Installation**

# Check if models are registered

python manage.py shell

from approvals.models import ApprovalWorkflow, Approval

from django.contrib.contenttypes.models import ContentType

# Should return 4 content types

ContentType.objects.filter(app\_label='approvals').count()

# Expected: 4

exit()

**6. Test Admin Interface**

python manage.py runserver

Navigate to: http://127.0.0.1:8000/admin/approvals/

You should see:

* Approval workflows
* Approval workflow steps
* Approvals
* Approval histories

**🧪 Testing the Module**

**Test 1: Create a Simple Workflow**

**Go to: Admin → Approvals → Approval workflows → Add**

Create a simple PR workflow:

* **Workflow name**: Standard PR Approval
* **Entity type**: Purchase Request
* **Is default**: ✅ Check this
* **Is active**: ✅ Check this
* Click **Save and continue editing**

Now add steps in the inline section at the bottom:

**Step 1:**

* Step sequence: 1
* Step name: Department Head Approval
* Approver role: Select "Department Head" (or any role you have)
* Is mandatory: ✅ Check
* Timeout hours: 24

**Step 2:**

* Step sequence: 2
* Step name: Manager Approval
* Approver role: Select "Factory Manager" (or any role)
* Is mandatory: ✅ Check
* Timeout hours: 48

Click **Save**

✅ **Success!** You've created your first approval workflow!

**Test 2: Create Workflows for Other Entities**

Create workflows for:

**PO Workflow (High Value):**

* Workflow name: High Value PO Approval
* Entity type: Purchase Order
* Min value: 100000 (100K BDT)
* Is default: Leave unchecked
* Steps:
  + Step 1: Procurement Officer
  + Step 2: Finance Manager
  + Step 3: Factory Manager

**GRN Workflow:**

* Workflow name: Standard GRN Approval
* Entity type: Goods Received Note
* Is default: ✅ Check
* Steps:
  + Step 1: QC Inspector
  + Step 2: Store Manager

**Test 3: Trigger Auto-Approval**

Now test if approvals are auto-created:

**Create a Purchase Request:**

1. Go to: **Procurement → Purchase requests → Add**
2. Fill in all required fields
3. **Important**: Set status to DRAFT first
4. Click **Save and continue editing**
5. Now change status to SUBMITTED
6. Click **Save**

**Check if approval was created:**

Go to: **Approvals → Approvals**

You should see a new approval for your PR!

The approval should show:

* Entity number: Your PR number
* Entity type: Purchase Request
* Status: IN\_PROGRESS (or PENDING)
* Current step: Step 1

✅ **Success!** Auto-approval creation is working!

**Test 4: Approve a Step**

**Manual approval using Python script:**

Create test\_approval.py in your project root:

import os

import django

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'pdl\_ims\_backend.settings')

django.setup()

from approvals.models import Approval, approve\_step

from users.models import User

# Get the pending approval

approval = Approval.objects.filter(status='IN\_PROGRESS').first()

if not approval:

print("No pending approvals found")

exit()

print(f"Approval: {approval.entity\_number}")

print(f"Current Step: {approval.get\_current\_step().step\_name}")

print(f"Status: {approval.status}")

# Get a user who can approve (you need to be in the approver role)

user = User.objects.filter(is\_superuser=True).first()

# Approve the step

try:

workflow\_complete = approve\_step(

approval=approval,

user=user,

comments="Approved for testing purposes"

)

if workflow\_complete:

print(f"✓ Approval COMPLETE! Status: {approval.status}")

else:

print(f"✓ Step approved. Next step: {approval.get\_current\_step().step\_name}")

except Exception as e:

print(f"✗ Error: {e}")

Run it:

python test\_approval.py

**Expected Output:**

Approval: PR-2025-0001

Current Step: Department Head Approval

Status: IN\_PROGRESS

✓ Step approved. Next step: Manager Approval

Run again to approve second step:

python test\_approval.py

**Expected Output:**

Approval: PR-2025-0001

Current Step: Manager Approval

Status: IN\_PROGRESS

✓ Approval COMPLETE! Status: APPROVED

**🎯 Usage Examples**

**Example 1: Check Pending Approvals for User**

from approvals.models import Approval

from users.models import User

# Get user

user = User.objects.get(username='john.doe')

# Find approvals pending this user's action

pending\_approvals = Approval.objects.filter(

status\_\_in=['PENDING', 'IN\_PROGRESS']

)

for approval in pending\_approvals:

step = approval.get\_current\_step()

if step and user in step.get\_approvers():

print(f"Pending: {approval.entity\_number} - {step.step\_name}")

**Example 2: Create Approval Programmatically**

from approvals.models import create\_approval

from procurement.models import PurchaseRequest

pr = PurchaseRequest.objects.get(pr\_number='PR-2025-0001')

approval = create\_approval(

entity\_type='PR',

entity\_id=pr.id,

entity\_number=pr.pr\_number,

requested\_by=pr.requester,

value=pr.total\_estimated\_value,

remarks=pr.justification

)

print(f"Created approval: {approval}")

**Example 3: Approve with Custom Logic**

from approvals.models import Approval, approve\_step

approval = Approval.objects.get(entity\_number='PR-2025-0001')

user = User.objects.get(username='manager')

# Check if user can approve

if user in approval.get\_pending\_approvers():

complete = approve\_step(

approval=approval,

user=user,

comments="Approved with budget allocation XYZ",

ip\_address='192.168.1.100'

)

if complete:

# Update the original PR status

pr = PurchaseRequest.objects.get(id=approval.entity\_id)

pr.status = 'APPROVED'

pr.approved\_by = user

pr.approved\_at = timezone.now()

pr.save()

**Example 4: Reject an Approval**

from approvals.models import Approval

approval = Approval.objects.get(entity\_number='PR-2025-0002')

user = User.objects.get(username='manager')

# Reject

approval.reject(

user=user,

comments="Budget constraints - please revise"

)

# Update original document

pr = PurchaseRequest.objects.get(id=approval.entity\_id)

pr.status = 'REJECTED'

pr.rejection\_reason = "Budget constraints"

pr.save()

**Example 5: Check for Timeouts**

from approvals.models import Approval

# Find timed-out approvals

pending\_approvals = Approval.objects.filter(

status\_\_in=['PENDING', 'IN\_PROGRESS']

)

for approval in pending\_approvals:

if approval.check\_timeout():

print(f"⚠ Timed out: {approval.entity\_number}")

approval.escalate()

**🔧 Integration with Existing Models**

**Update Purchase Request Model**

Edit procurement/models.py - add this method to PurchaseRequest:

def submit\_for\_approval(self):

"""Submit PR for approval."""

if self.status != 'DRAFT':

raise ValidationError("Can only submit draft PRs")

self.status = 'SUBMITTED'

self.save()

# Approval will be auto-created by signal

from approvals.models import Approval

approval = Approval.objects.filter(

entity\_type='PR',

entity\_id=self.id

).first()

return approval

**Update Purchase Order Model**

Edit procurement/models.py - add to PurchaseOrder:

def get\_approval\_status(self):

"""Get current approval status."""

from approvals.models import Approval

approval = Approval.objects.filter(

entity\_type='PO',

entity\_id=self.id

).first()

if not approval:

return None

return {

'status': approval.status,

'current\_step': approval.get\_current\_step().step\_name if approval.get\_current\_step() else None,

'pending\_approvers': [u.full\_name for u in approval.get\_pending\_approvers()]

}

**📊 Expected Results After Testing**

After completing all tests, you should have:

**In Workflows:**

* 3+ workflows created (PR, PO, GRN)
* Each workflow has 2-3 steps
* Steps assigned to roles

**In Approvals:**

* Test approvals created for PR, PO, GRN
* Some approvals completed (APPROVED)
* Some approvals in progress (IN\_PROGRESS)

**In History:**

* Multiple history entries showing who approved what
* Comments recorded
* Action dates logged

**🐛 Troubleshooting**

**Issue: Approval Not Auto-Creating**

**Solution:**

# Check if signals are loaded

from approvals import signals

# Verify workflow exists

from approvals.models import ApprovalWorkflow

workflows = ApprovalWorkflow.objects.filter(

entity\_type='PR',

is\_active=True

)

print(f"Found {workflows.count()} PR workflows")

# Manually create if needed

from approvals.models import create\_approval

approval = create\_approval(

entity\_type='PR',

entity\_id=pr.id,

entity\_number=pr.pr\_number,

requested\_by=pr.requester

)

**Issue: "No active workflow found"**

**Solution:** Make sure you have:

1. Created a workflow for that entity type
2. Set it as **is\_active = True**
3. Set it as **is\_default = True** (or configure value ranges)

**Issue: User Can't Approve**

**Solution:**

# Check if user has the required role

from users.models import User, Role

user = User.objects.get(username='john.doe')

required\_role = Role.objects.get(role\_name='Department Head')

# Check user's roles

user\_roles = user.user\_roles.all()

print(f"User roles: {[ur.role.role\_name for ur in user\_roles]}")

# Add role if missing

if required\_role not in [ur.role for ur in user\_roles]:

from users.models import UserRole

UserRole.objects.create(user=user, role=required\_role)

**✅ Verification Checklist**

* [ ] Approvals app added to INSTALLED\_APPS
* [ ] Migrations created and applied
* [ ] Admin interface accessible
* [ ] Can create workflows
* [ ] Can create workflow steps
* [ ] Approvals auto-create when documents submitted
* [ ] Can approve steps manually
* [ ] History records created
* [ ] Workflow completes correctly
* [ ] Rejection works
* [ ] Timeout detection works

**🎉 Success!**

If all tests pass, your Approval Workflows module is ready!

**What's Next?**

1. Create workflows for all entity types
2. Assign users to appropriate roles
3. Test full approval cycles
4. Move to Reports Module

**📚 Next Steps**

After Approval Workflows:

1. **Reports Module** - Generate stock, procurement, and asset reports
2. **Notifications** - Email/SMS alerts for approvals
3. **RMG Operations** - Yarn/fabric operations
4. **API Layer** - REST API for all modules
5. **Frontend** - User interface

**Ready to continue?** Let me know! 🚀