

# Workflow API Functions

Technical Manual - API Reference

SONARWORKS WORKFLOW SYSTEM

Version 1.1

# Table of Contents

1. Introduction
2. Workflow Endpoints Overview
3. List Workflows
4. Get Workflow Details
5. Create Workflow
6. Update Workflow
7. Publish/Unpublish Workflow
8. Workflow Instances
9. Submit Workflow Instance
10. Instance Operations
11. Error Handling
12. Code Examples

# 1. Introduction

This manual documents the Workflow API functions for the Sonar Workflow System. These APIs enable programmatic management of workflow templates and workflow instances (submissions).

## API Capabilities:

- Retrieve and manage workflow templates
- Submit and track workflow instances
- Manage workflow forms and fields
- Handle workflow attachments
- Query and filter workflow data

### 1.1 Authentication

All endpoints require JWT authentication:

[HTTP Header]  
Authorization: Bearer <token>

# 2. Workflow Endpoints Overview

## 2.1 Workflow Template Endpoints

Method	Endpoint	Description
GET	/api/workflows	List all workflows
GET	/api/workflows/active	List published workflows
GET	/api/workflows/{id}	Get workflow by ID
GET	/api/workflows/code/{code}	Get workflow by code
POST	/api/workflows	Create new workflow
PUT	/api/workflows/{id}	Update workflow
DELETE	/api/workflows/{id}	Delete workflow
POST	/api/workflows/{id}/publish	Publish workflow
POST	/api/workflows/{id}/unpublish	Unpublish workflow

## 2.2 Workflow Instance Endpoints

Method	Endpoint	Description
GET	/api/workflows/{code}/instances	List instances for workflow
POST	/api/workflows/{code}/new	Create/submit instance
GET	/api/workflows/{code}/instances/{id}	Get instance detail
PUT	/api/workflows/{code}/edit/{id}	Update draft instance
POST	/api/workflows/instances/{id}/submit	Submit draft for approval
POST	/api/workflows/instances/{id}/recall	Recall pending instance
POST	/api/workflows/instances/{id}/cancel	Cancel instance

## 3. List Workflows

### 3.1 Get All Workflows

[HTTP]

GET /api/workflows

**Query Parameters:**

Parameter	Type	Description
page	integer	Page number (0-indexed)
size	integer	Items per page
sort	string	Sort field and direction (e.g., 'name,asc')
search	string	Search in name and description
category	string	Filter by category (FINANCIAL, NON_FINANCIAL)

**Response:**

[JSON]

```
{  
  "success": true,  
  "data": {  
    "content": [  
      {  
        "id": "550e8400-e29b-41d4-a716-446655440000",  
        "code": "PURCHASE_REQUEST",  
        "name": "Purchase Request",  
        "description": "Submit purchase requests for office supplies",  
        "category": "FINANCIAL",  
        "icon": "shopping_cart",  
        "published": true,  
        "version": 3,  
        "createdAt": "2024-01-10T09:00:00Z",  
        "updatedAt": "2024-01-15T14:30:00Z"  
      },  
      ],  
      "totalElements": 25,  
      "totalPages": 3,  
      "number": 0,  
      "size": 10  
    }  
}
```

## 3.2 Get Active (Published) Workflows

[HTTP]

```
GET /api/workflows/active
```

Returns only workflows that are published and accessible to the current user based on organizational restrictions.

# 4. Get Workflow Details

## 4.1 Get by ID

[HTTP]

```
GET /api/workflows/{id}
```

## 4.2 Get by Code

[HTTP]

```
GET /api/workflows/code/{code}
```

**Response:**

[JSON]

```
{
  "success": true,
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "code": "PURCHASE_REQUEST",
    "name": "Purchase Request",
    "description": "Submit purchase requests",
    "category": "FINANCIAL",
    "icon": "shopping_cart",
    "published": true,
    "version": 3,
    "requireComment": true,
    "forms": [
      {
        "id": "form-uuid-1",
        "name": "Main Form",
        "displayOrder": 1,
        "isMainForm": true,
        "screens": [
          {
            "id": "screen-uuid-1",
            "title": "Basic Information",
            "displayOrder": 1
          }
        ],
        "fieldGroups": [

```

```
{
  "id": "group-uuid-1",
  "name": "Request Details",
  "columns": 2,
  "collapsible": true,
  "expanded": true,
  "displayOrder": 1
}
],
"fields": [
{
  "id": "field-uuid-1",
  "name": "itemDescription",
  "label": "Item Description",
  "fieldType": "TEXT",
  "dataType": "ALPHANUMERIC",
  "required": true,
  "displayOrder": 1,
  "columnSpan": 2,
  "helpText": "Enter description of items to purchase"
},
{
  "id": "field-uuid-2",
  "name": "amount",
  "label": "Total Amount",
  "fieldType": "CURRENCY",
  "dataType": "NUMBER",
  "required": true,
  "displayOrder": 2,
  "minValue": 0,
  "maxValue": 1000000
}
]
},
],
"approvers": [
{
  "id": "approver-uuid-1",
  "level": 1,
  "user": {
    "id": "user-uuid",
    "username": "manager1",
    "fullName": "Manager One"
  },
  "amountLimit": 5000,
  "sendEmailNotification": true,
  "sendEmailApprovalLink": true
}
]
```

```
        ],
        "corporates": [],
        "sbus": [],
        "branches": [],
        "departments": []
    }
}
```

## 5. Create Workflow

### 5.1 Endpoint

[HTTP]  
POST /api/workflows

### 5.2 Request Body

[JSON]

```
{
    "name": "Leave Request",
    "code": "LEAVE_REQUEST",
    "description": "Submit leave/vacation requests",
    "category": "NON_FINANCIAL",
    "icon": "event_busy",
    "requireComment": true,
    "forms": [
        {
            "name": "Leave Form",
            "displayOrder": 1,
            "isMainForm": true,
            "fieldGroups": [
                {
                    "name": "Leave Details",
                    "columns": 2,
                    "collapsible": false,
                    "displayOrder": 1
                }
            ],
            "fields": [
                {
                    "name": "leaveType",
                    "label": "Leave Type",
                    "fieldType": "SELECT",
                    "required": true,
                    "displayOrder": 1,
                    "options": [
                        {"label": "Annual Leave", "value": "ANNUAL"},
                        {"label": "Sick Leave", "value": "SICK"}
                    ]
                }
            ]
        }
    ]
}
```

```

        {"label": "Personal Leave", "value": "PERSONAL"}
    ],
},
{
    "name": "startDate",
    "label": "Start Date",
    "fieldType": "DATE",
    "required": true,
    "displayOrder": 2
},
{
    "name": "endDate",
    "label": "End Date",
    "fieldType": "DATE",
    "required": true,
    "displayOrder": 3
},
{
    "name": "reason",
    "label": "Reason",
    "fieldType": "TEXTAREA",
    "required": false,
    "displayOrder": 4,
    "columnSpan": 2
}
],
},
],
"approvers": [
{
    "level": 1,
    "userId": "manager-user-uuid",
    "sendEmailNotification": true,
    "sendEmailApprovalLink": true
}
]
}

```

## 5.3 Response

[JSON]

```
{
    "success": true,
    "message": "Workflow created successfully",
    "data": {
        "id": "new-workflow-uuid",
        "code": "LEAVE_REQUEST",
        "name": "Leave Request",
        "published": false,

```

```
        "version": 1
    }
}
```

## 6. Update Workflow

### 6.1 Endpoint

[HTTP]

```
PUT /api/workflows/{id}
```

### 6.2 Request Body

Same structure as create. Include all fields, not just changed ones.

### 6.3 Response

[JSON]

```
{
  "success": true,
  "message": "Workflow updated successfully",
  "data": {
    "id": "workflow-uuid",
    "code": "LEAVE_REQUEST",
    "name": "Leave Request",
    "published": true,
    "version": 2
  }
}
```

## 7. Publish/Unpublish Workflow

### 7.1 Publish Workflow

[HTTP]

```
POST /api/workflows/{id}/publish
```

**Response:**

[JSON]

```
{
  "success": true,
  "message": "Workflow published successfully",
  "data": {
    "id": "workflow-uuid",
    "published": true
  }
}
```

## 7.2 Unpublish Workflow

[HTTP]  
POST /api/workflows/{id}/unpublish

### Response:

[JSON]

```
{  
  "success": true,  
  "message": "Workflow unpublished successfully",  
  "data": {  
    "id": "workflow-uuid",  
    "published": false  
  }  
}
```

# 8. Workflow Instances

## 8.1 List Instances

[HTTP]  
GET /api/workflows/{code}/instances

### Query Parameters:

Parameter	Type	Description
page	integer	Page number
size	integer	Items per page
status	string	Filter by status (DRAFT, PENDING, APPROVED, REJECTED)
initiator	string	Filter by initiator user ID
fromDate	date	Submission date from
toDate	date	Submission date to

### Response:

[JSON]

```
{  
  "success": true,  
  "data": {  
    "content": [  
      {  
        "id": "instance-uuid-1",  
        "referenceNumber": "WF-2024-00001",  
        "workflow": {  
          "id": "workflow-uuid",  
          "code": "PURCHASE_REQUEST",  
          "name": "Purchase Request"  
        }  
      }  
    ]  
  }  
}
```

```

        "name": "Purchase Request"
    },
    "status": "PENDING",
    "initiator": {
        "id": "user-uuid",
        "username": "john.doe",
        "fullName": "John Doe"
    },
    "currentApprover": {
        "id": "approver-uuid",
        "username": "manager1",
        "fullName": "Manager One"
    },
    "amount": 2500.00,
    "submittedAt": "2024-01-15T10:30:00Z",
    "currentLevel": 1
},
],
"totalElements": 45,
"totalPages": 5,
"number": 0
}
}

```

## 8.2 Get Instance Detail

[HTTP]  
 GET /api/workflows/{code}/instances/{instanceId}

**Response:**

[JSON]

```
{
    "success": true,
    "data": {
        "id": "instance-uuid",
        "referenceNumber": "WF-2024-00001",
        "workflow": {
            "id": "workflow-uuid",
            "code": "PURCHASE_REQUEST",
            "name": "Purchase Request"
        },
        "status": "PENDING",
        "initiator": {
            "id": "user-uuid",
            "username": "john.doe",
            "fullName": "John Doe"
        },
        "fieldValues": [

```

```
{
  "field": {
    "id": "field-uuid-1",
    "name": "itemDescription",
    "label": "Item Description"
  },
  "value": "Office supplies for Q1",
  "displayValue": "Office supplies for Q1"
},
{
  "field": {
    "id": "field-uuid-2",
    "name": "amount",
    "label": "Total Amount"
  },
  "value": "2500.00",
  "displayValue": "$2,500.00"
}
],
"attachments": [
  {
    "id": "attachment-uuid",
    "fileName": "quotation.pdf",
    "contentType": "application/pdf",
    "fileSize": 102400,
    "uploadedAt": "2024-01-15T10:25:00Z"
  }
],
"approvalHistory": [
  {
    "id": "history-uuid-1",
    "action": "SUBMITTED",
    "actor": {
      "id": "user-uuid",
      "fullName": "John Doe"
    },
    "actionDate": "2024-01-15T10:30:00Z",
    "comments": "Initial submission"
  }
],
"submittedAt": "2024-01-15T10:30:00Z",
"currentLevel": 1,
"amount": 2500.00
}
}
```

## 9. Submit Workflow Instance

### 9.1 Create and Submit Instance

[HTTP]

POST /api/workflows/{code}/new

#### Request Body:

```
[JSON]
{
    "status": "PENDING",
    "fieldValues": {
        "itemDescription": "Office supplies for Q1 2024",
        "amount": 2500.00,
        "justification": "Required for team operations",
        "vendor": "Office Depot"
    },
    "sbuId": "sbu-uuid",
    "comments": "Please expedite if possible"
}
```

#### Response:

```
[JSON]
{
    "success": true,
    "message": "Workflow submitted successfully",
    "data": {
        "id": "new-instance-uuid",
        "referenceNumber": "WF-2024-0002",
        "status": "PENDING"
    }
}
```

### 9.2 Save as Draft

To save as draft, set status to DRAFT:

```
[JSON]
{
    "status": "DRAFT",
    "fieldValues": {
        "itemDescription": "Partial data..."
    }
}
```

### 9.3 Submit with Attachments

Use multipart/form-data for file uploads:

```
[HTTP]
POST /api/workflows/{code}/new
Content-Type: multipart/form-data

--boundary
Content-Disposition: form-data; name="data"
Content-Type: application/json

{"status": "PENDING", "fieldValues": {...}}
--boundary
Content-Disposition: form-data; name="files"; filename="quote.pdf"
Content-Type: application/pdf

<binary file content>
--boundary--
```

## 10. Instance Operations

### 10.1 Update Draft Instance

```
[HTTP]
PUT /api/workflows/{code}/edit/{instanceId}
```

### 10.2 Submit Draft for Approval

```
[HTTP]
POST /api/workflows/instances/{id}/submit
```

### 10.3 Recall Instance

```
[HTTP]
POST /api/workflows/instances/{id}/recall
```

#### Request Body:

```
[JSON]
{
  "reason": "Need to update the vendor information"
}
```

### 10.4 Cancel Instance

```
[HTTP]
POST /api/workflows/instances/{id}/cancel
```

#### Request Body:

```
[JSON]
{
```

```

    "reason": "Request no longer needed"
}

```

## 11. Error Handling

Error Code	HTTP Status	Description
WORKFLOW_NOT_FOUND	404	Workflow does not exist
INSTANCE_NOT_FOUND	404	Instance does not exist
VALIDATION_ERROR	422	Field validation failed
WORKFLOW_NOT_PUBLISHED	400	Cannot submit to unpublished workflow
CANNOT_RECALL	400	Instance cannot be recalled in current state
ACCESS_DENIED	403	No permission to access this workflow

## 12. Code Examples

### 12.1 List and Submit Workflow (JavaScript)

```
[JavaScript]
// Get active workflows
const workflows = await fetch('/api/workflows/active', {
  headers: { 'Authorization': `Bearer ${token}` }
}).then(r => r.json());

// Submit a new instance
const submission = await fetch('/api/workflows/PURCHASE_REQUEST/new', {
  method: 'POST',
  headers: {
    'Authorization': `Bearer ${token}`,
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    status: 'PENDING',
    fieldValues: {
      itemDescription: 'Office Supplies',
      amount: 500.00,
      vendor: 'Amazon'
    }
  })
}).then(r => r.json());

console.log('Reference:', submission.data.referenceNumber);
```

## 12.2 Python Example

```
[Python]
import requests

# List workflows
response = requests.get(
    "https://api.server.com/api/workflows/active",
    headers={"Authorization": f"Bearer {token}"}
)
workflows = response.json()

# Submit workflow
submission = requests.post(
    "https://api.server.com/api/workflows/PURCHASE_REQUEST/new",
    headers={
        "Authorization": f"Bearer {token}",
        "Content-Type": "application/json"
    },
    json={
        "status": "PENDING",
        "fieldValues": {
            "itemDescription": "Office Supplies",
            "amount": 500.00
        }
    }
)
print(submission.json())
```