# API Documentation

## Equipment Management System

Technical Implementation Guide

# API Overview and Authentication

## API Structure

Base URL: `https://api.equipment-management.com/v1`
All requests should be made over HTTPS.

## Authentication

**API Key:** `Authorization: ApiKey YOUR_API_KEY`
**Bearer Token:** `Authorization: Bearer YOUR_TOKEN`

## Common Response Format

```
// Success Response
{
  "status": "success",
  "data": { ... },
  "meta": { ... }
}
```

## Error Response

```
// Error Response
{
  "status": "error",
  "error": {
    "code": "ERROR_CODE",
    "message": "Error message"
  }
}
```

## Common Status Codes

| Status Code | Description |
|---|---|
| 200 OK | Request succeeded |
| 201 Created | Resource created successfully |
| 400 Bad Request | Invalid request parameters |
| 401 Unauthorized | Authentication failed |
| 404 Not Found | Resource not found |

# Equipment Registry API

## Endpoints

| | |
|---|---|
| GET | /equipment |
| GET | /equipment/{id} |
| POST | /equipment |
| PUT | /equipment/{id} |
| DELETE | /equipment/{id} |

## Request Example

```
// Create new equipment
POST /equipment
{
  "name": "Excavator XL2000",
  "model": "XL2000",
  "category": "heavy",
  "status": "active",
  "location": "Site A",
  "operator_id": "op-123",
  "project_id": "proj-456"
}
```

## Response Example

```
// Response for GET /equipment/{id}
{
  "status": "success",
  "data": {
    "id": "eq-123",
    "name": "Excavator XL2000",
    "model": "XL2000",
    "category": "heavy",
    "status": "active",
    "compliance_score": 92,
    "last_inspection": "2025-06-15",
    "next_inspection": "2025-07-15",
    "location": "Site A",
    "operator": {
      "id": "op-123",
      "name": "John Smith"
    }
  }
}
```

## Query Parameters

| Parameter | Description |
|---|---|
| status | Filter by equipment status |
| category | Filter by equipment category |
| project_id | Filter by assigned project |

# Lifting Tools API

## Endpoints

| Method | Endpoint | Description |
|---|---|---|
| GET | /lifting-tools | List all lifting accessories |
| GET | /lifting-tools/{id} | Get specific lifting accessory |
| POST | /lifting-tools | Create new lifting accessory |
| PUT | /lifting-tools/{id} | Update lifting accessory |
| DELETE | /lifting-tools/{id} | Delete lifting accessory |

## Request Example

```
// Create Lifting Tool
POST /v1/lifting-tools
{
  "name": "Chain Hoist 2T",
  "type": "hoist",
  "manufacturer": "SafeLift",
  "model": "CH-2000",
  "serial_number": "SL-5678",
  "safe_working_load": 2000,
  "load_unit": "kg",
  "certification_number": "CERT-1234",
  "certification_date": "2023-01-15",
  "next_inspection_date": "2023-07-15",
  "project_id": "proj-789"
}
```

## Response Example

```
// Success Response
{
  "status": "success",
  "data": {
    "id": "lift-1234",
    "name": "Chain Hoist 2T",
    "type": "hoist",
    "manufacturer": "SafeLift",
    "model": "CH-2000",
    "serial_number": "SL-5678",
    "safe_working_load": 2000,
    "load_unit": "kg",
    "certification_date": "2023-01-15",
    "next_inspection_date": "2023-07-15",
    "status": "active"
  }
}
```

# Power Tools API

## Endpoints

| Method | Endpoint | Description |
|---|---|---|
| GET | /power-tools | List all power tools |
| GET | /power-tools/{id} | Get power tool details |
| POST | /power-tools | Create new power tool |
| PUT | /power-tools/{id} | Update power tool |
| DELETE | /power-tools/{id} | Delete power tool |

## Request Example

```
// POST /power-tools
{
  "name": "Cordless Drill",
  "type": "Drill",
  "manufacturer": "DeWalt",
  "model": "DCD778",
  "serial_number": "DW20220512",
  "inspection_frequency": 30,
  "status": "active",
  "project_id": "proj_12345"
}
```

## Response Example

```
// Response
{
  "status": "success",
  "data": {
    "id": "pt_67890",
    "name": "Cordless Drill",
    "type": "Drill",
    "manufacturer": "DeWalt",
    "model": "DCD778",
    "serial_number": "DW20220512",
    "inspection_frequency": 30,
    "status": "active",
    "project_id": "proj_12345",
    "last_inspection": "2025-06-15",
    "next_inspection": "2025-07-15",
    "created_at": "2025-06-10T14:30:00Z",
    "updated_at": "2025-06-10T14:30:00Z"
  }
}
```

## Inspection Endpoints

| Method | Endpoint | Description |
|---|---|---|
| POST | /power-tools/{id}/inspections | Record new inspection |
| GET | /power-tools/{id}/inspections | Get inspection history |
| POST | /power-tools/{id}/maintenance | Record maintenance |
| GET | /power-tools/{id}/maintenance | Get maintenance history |

# Vehicles API

## Endpoints

| Method | Endpoint | Description |
|---|---|---|
| GET | /vehicles | List all vehicles |
| GET | /vehicles/{id} | Get vehicle details |
| POST | /vehicles | Create new vehicle |
| PUT | /vehicles/{id} | Update vehicle |
| DELETE | /vehicles/{id} | Delete vehicle |

## Request Example

```
// Create Vehicle
POST /api/v1/vehicles
{
  "name": "Truck 101",
  "model": "Volvo FH16",
  "plate_number": "ABC123",
  "category": "heavy",
  "status": "active",
  "location": "Site A",
  "driver_id": "d12345",
  "project_id": "p789"
}
```

## Response Example

```
// Vehicle Response
{
  "status": "success",
  "data": {
    "id": "v56789",
    "name": "Truck 101",
    "model": "Volvo FH16",
    "plate_number": "ABC123",
    "category": "heavy",
    "status": "active",
    "compliance_score": 95,
    "last_inspection": "2025-06-15",
    "next_inspection": "2025-07-15",
    "location": "Site A",
    "driver": {
      "id": "d12345",
      "name": "John Smith"
    },
    "project": {
      "id": "p789",
      "name": "Highway Construction"
    }
  }
}
```

## Additional Endpoints

| Method | Endpoint | Description |
|---|---|---|
| GET | /vehicles/{id}/documents | List vehicle documents |
| POST | /vehicles/{id}/documents | Upload document |
| POST | /vehicles/{id}/share | Share vehicle details |
| GET | /vehicles/categories | List vehicle categories |

# Operator Management API

## Endpoints

| Method | Endpoint | Description |
|---|---|---|
| GET | /operators | List all operators |
| GET | /operators/{id} | Get operator details |
| POST | /operators | Create new operator |
| PUT | /operators/{id} | Update operator |

## Request Example

```
// Create Operator
POST /api/v1/operators
{
  "name": "John Smith",
  "email": "john.smith@example.com",
  "nationality": "US",
  "type": "Heavy Equipment",
  "project_id": "proj-123"
}
```

## Response Example

```
// Operator Details Response
{
  "status": "success",
  "data": {
    "id": "op-789",
    "name": "John Smith",
    "email": "john.smith@example.com",
    "nationality": "US",
    "type": "Heavy Equipment",
    "project_id": "proj-123",
    "certifications": [
      {
        "id": "cert-456",
        "type": "License",
        "number": "OP-12345",
        "expiry_date": "2025-12-31"
      }
    ]
  }
}
```

## Document Management

| Method | Endpoint | Description |
|---|---|---|
| POST | /operators/{id}/documents | Upload document |
| GET | /operators/{id}/documents | List documents |
| GET | /operators/{id}/documents/{doc_id} | Download document |

# Maintenance API

## Endpoints

| Method | Endpoint | Description |
|---|---|---|
| GET | /maintenance | List all maintenance records |
| GET | /maintenance/{id} | Get maintenance details |
| POST | /maintenance | Create maintenance record |
| PUT | /maintenance/{id} | Update maintenance record |

## Request Example

```
// Create Maintenance Record
POST /api/v1/maintenance
{
  "asset_type": "equipment",
  "asset_id": "eq-123",
  "maintenance_type": "preventive",
  "description": "Regular oil change",
  "scheduled_date": "2025-07-20",
  "assigned_to": "tech-456",
  "priority": "medium"
}
```

## Response Example

```
// Maintenance Response
{
  "status": "success",
  "data": {
    "id": "maint-789",
    "asset_type": "equipment",
    "asset_id": "eq-123",
    "asset_name": "Excavator XL2000",
    "maintenance_type": "preventive",
    "description": "Regular oil change",
    "scheduled_date": "2025-07-20",
    "status": "scheduled",
    "assigned_to": {
      "id": "tech-456",
      "name": "Mike Johnson"
    },
    "priority": "medium"
  }
}
```

## Filter Parameters

| Parameter | Description |
|---|---|
| asset_type | Filter by asset type (equipment, vehicle, tool) |
| asset_id | Filter by specific asset ID |
| status | Filter by status (scheduled, in_progress, completed) |
| date_range | Filter by scheduled date range |

# Certifications API

## Endpoints

| Method | Endpoint | Description |
|---|---|---|
| GET | /certifications | List all certifications |
| GET | /certifications/{id} | Get certification details |
| POST | /certifications | Create new certification |
| PUT | /certifications/{id} | Update certification |

## Request Example

```
// Create Certification
POST /api/v1/certifications
{
  "name": "Crane Operator License",
  "number": "COL-2023-456",
  "type": "operator",
  "entity_id": "op-789",
  "issuer": "National Safety Council",
  "issue_date": "2023-01-15",
  "expiry_date": "2025-01-15"
}
```

## Response Example

```
// Certification Response
{
  "status": "success",
  "data": {
    "id": "cert-456",
    "name": "Crane Operator License",
    "number": "COL-2023-456",
    "type": "operator",
    "entity_id": "op-789",
    "entity_name": "John Smith",
    "issuer": "National Safety Council",
    "issue_date": "2023-01-15",
    "expiry_date": "2025-01-15",
    "status": "valid"
  }
}
```

## Filter Parameters

| Parameter | Description |
|---|---|
| type | Filter by certification type |
| entity_id | Filter by associated entity ID |
| status | Filter by status (valid, expired, expiring) |

# Projects API

## Endpoints

| Method | Endpoint | Description |
|---|---|---|
| GET | /projects | List all projects |
| GET | /projects/{id} | Get project details |
| POST | /projects | Create new project |
| PUT | /projects/{id} | Update project |

## Request Example

```
// Create Project
POST /api/v1/projects
{
  "name": "Downtown Construction",
  "code": "DC-2025",
  "description": "Commercial building construction",
  "location": "123 Main St, Downtown",
  "start_date": "2025-08-01",
  "end_date": "2026-03-31",
  "status": "planning"
}
```

## Response Example

```
// Project Response
{
  "status": "success",
  "data": {
    "id": "proj-456",
    "name": "Downtown Construction",
    "code": "DC-2025",
    "description": "Commercial building construction",
    "location": "123 Main St, Downtown",
    "start_date": "2025-08-01",
    "end_date": "2026-03-31",
    "status": "planning"
  }
}
```

## Resource Assignment

| Method | Endpoint | Description |
|---|---|---|
| POST | /projects/{id}/equipment | Assign equipment to project |
| GET | /projects/{id}/equipment | List assigned equipment |
| POST | /projects/{id}/operators | Assign operators to project |
| GET | /projects/{id}/operators | List assigned operators |

# Documents API

## Endpoints

| Method | Endpoint | Description |
|---|---|---|
| GET | /documents | List all documents |
| GET | /documents/{id} | Download document |
| POST | /documents | Upload new document |
| PUT | /documents/{id} | Update document metadata |

## Request Example

```
// Upload Document (multipart/form-data)
POST /api/v1/documents
{
  "name": "Equipment Manual",
  "type": "manual",
  "entity_type": "equipment",
  "entity_id": "eq-123",
  "description": "User manual for Excavator XL2000"
  // File data in multipart form
}
```

## Response Example

```
// Document Response
{
  "status": "success",
  "data": {
    "id": "doc-456",
    "name": "Equipment Manual",
    "type": "manual",
    "entity_type": "equipment",
    "entity_id": "eq-123",
    "description": "User manual for Excavator XL2000",
    "filename": "excavator_xl2000_manual.pdf",
    "file_size": 2458621,
    "mime_type": "application/pdf",
    "url": "/api/v1/documents/doc-456"
  }
}
```

## Filter Parameters

| Parameter | Description |
|---|---|
| type | Filter by document type |
| entity_type | Filter by related entity type |
| entity_id | Filter by specific entity ID |
| uploaded_after | Filter by upload date (after) |

# API Integration

## API Key Management

| Method | Endpoint | Description |
|---|---|---|
| GET | /api-keys | List all API keys |
| POST | /api-keys | Generate new API key |
| DELETE | /api-keys/{id} | Revoke API key |

## Request Example

```
// Generate API Key
POST /api/v1/api-keys
{
  "name": "Mobile App Integration",
  "permissions": ["read:equipment", "read:operators"],
  "expires_in": 31536000
}
```

## Webhook Management

| Method | Endpoint | Description |
|---|---|---|
| GET | /webhooks | List all webhooks |
| POST | /webhooks | Create new webhook |
| PUT | /webhooks/{id} | Update webhook |

## Webhook Example

```
// Create Webhook
POST /api/v1/webhooks
{
  "name": "Equipment Status Change",
  "url": "https://example.com/webhook/equipment",
  "events": ["equipment.created", "equipment.updated"],
  "active": true
}
```