Konecta ERP System

API Documentation

Development Team

November 11, 2025

Contents

1 API Overview

1.1 Base URLs

- **Development**: `http://localhost:8084/api` (via Gateway)
- **Direct Service Access**:
  - Auth Service: `http://localhost:8089/api/auth`
  - HR Service: `http://localhost:8088/api/hr`
  - Finance Service: `http://localhost:8087/api/finance`
  - Inventory Service: `http://localhost:8086/api/inventory`
  - Reporting Service: `http://localhost:5160/api/report`

1.2 Authentication

All authenticated endpoints require a JWT token in the Authorization header:

```
Authorization: Bearer <token>
```

1.3 Role-Based Access Control

The system supports the following roles:

- **ADMIN**: Full system access
- **HR**: Human resources management
- **FINANCE**: Financial operations
- **INVENTORY**: Inventory management
- **EMPLOYEE**: Self-service access

2 Authentication Service APIs

2.1 Base Path: /api/auth

2.1.1 1. User Invitation

- **Endpoint**: `POST /api/auth/users/invite`
- **Authorization**: ADMIN, HR
- **Purpose**: Invite a new user to the system
- **Request Body**:

```
{
  "email": "user@example.com",
  "name": "John Doe",
  "role": "EMPLOYEE"
}
```

- **Response**:

```
{
  "message": "invite sent",
  "id": 1,
  "email": "user@example.com"
}
```

2.1.2 2. Validate Registration Token

- **Endpoint**: `GET /api/auth/registration/validate?token={token}`
- **Authorization**: Public
- **Purpose**: Validate registration token before completing registration
- **Response**:

```
{
  "valid": true,
  "email": "user@example.com",
  "name": "John Doe",
  "role": "EMPLOYEE"
}
```

2.1.3 3. Complete Registration

- **Endpoint**: `POST /api/auth/registration/complete`
- **Authorization**: Public
- **Purpose**: Complete user registration with profile details
- **Request Body**:

```
{
  "token": "verification-token",
  "fullName": "John Doe",
  "username": "johndoe",
  "phone": "+1234567890",
  "password": "securePassword123",
  "Account": "1234 5678 9012 3456",
  "cardType": "visa or Mastercard"
}
```

- **Response**:

```
{
  "message": "otp_sent",
  "delivery": "email",
  "expiresIn": 300
}
```

2.1.4 4. Verify OTP

- **Endpoint**: `POST /api/auth/registration/verify-otp`
- **Authorization**: Public
- **Purpose**: Verify OTP and activate account

- **Request Body**:

```
{
  "token": "verification-token",
  "otp": "123456"
}
```

- **Response**:

```
{
  "message": "account_activated"
}
```

2.1.5 5. Login

- **Endpoint**: `POST /api/auth/login`
- **Authorization**: Public
- **Purpose**: Authenticate user and receive JWT token
- **Request Body**:

```
{
  "email": "user@example.com",
  "password": "password123"
}
```

- **Response**:

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

2.1.6 6. Get Current User Profile

- **Endpoint**: `GET /api/auth/me`
- **Authorization**: Authenticated
- **Response**:

```
{
  "id": 1,
  "username": "johndoe",
  "email": "user@example.com",
  "role": "EMPLOYEE"
}
```

2.1.7 7. Update User Profile

- **Endpoint**: `PUT /api/auth/me`
- **Authorization**: Authenticated
- **Request Body**:

```
{
  "username": "newusername",
  "password": "newPassword123",
  "confirmPassword": "newPassword123"
}
```

2.1.8 8. List Users

- **Endpoint**: `GET /api/auth/users?q={query}&role={role}`
- **Authorization**: ADMIN, HR
- **Query Parameters**:
  - `q`: Search query (optional)
  - `role`: Filter by role (optional)

2.1.9 9. Update User

- **Endpoint**: `PUT /api/auth/users/{id}`
- **Authorization**: ADMIN, HR
- **Request Body**:

```
{
  "fullName": "Updated Name",
  "phone": "+1234567890"
}
```

2.1.10 10. Delete User

- **Endpoint**: `DELETE /api/auth/users/{id}`
- **Authorization**: ADMIN, HR

2.1.11 11. Forgot Password - Start

- **Endpoint**: `POST /api/auth/forgot/start`
- **Authorization**: Public
- **Request Body**:

```
{
  "email": "user@example.com"
}
```

2.1.12 12. Forgot Password - Complete

- **Endpoint**: `POST /api/auth/forgot/complete`
- **Authorization**: Public
- **Request Body**:

```
{
  "email": "user@example.com",
  "otp": "123456",
  "password": "newPassword123",
  "confirmPassword": "newPassword123"
}
```

2.1.13 13. Validate JWT Token

- **Endpoint**: `POST /api/auth/validate`
- **Authorization**: Public
- **Request Body**:

```
{
  "token": "jwt-token-string"
}
```

3 HR Service APIs

3.1 Base Path: `/api/hr`

3.2 Employee Management

3.2.1 1. List All Employees

- **Endpoint**: `GET /api/hr/employees`
- **Authorization**: Authenticated
- **Response**: Array of `EmployeeResponse`

3.2.2 2. Get Employee by ID

- **Endpoint**: `GET /api/hr/employees/{id}`
- **Authorization**: Authenticated
- **Response**: `EmployeeResponse`

3.2.3 3. Create Employee

- **Endpoint**: `POST /api/hr/employees`
- **Authorization**: ADMIN
- **Request Body**:

```
{
  "firstName": "John",
  "lastName": "Doe",
  "email": "john.doe@example.com",
  "phone": "+1234567890",
  "position": "Software Engineer",
  "hireDate": "2024-01-15",
  "salary": 75000.00,
  "workingHours": 40.0,
  "departmentId": 1
}
```

3.2.4 4. Update Employee

- **Endpoint**: `PUT /api/hr/employees/{id}`
- **Authorization**: ADMIN, HR
- **Request Body**: Same as Create

3.2.5 5. Delete Employee

- **Endpoint**: `DELETE /api/hr/employees/{id}`
- **Authorization**: ADMIN, HR

3.2.6 6. Ensure Employee Exists

- **Endpoint**: `POST /api/hr/employees/ensure`
- **Authorization**: ADMIN, HR
- **Purpose**: Create employee if not exists, or return existing
- **Request Body**:

```
{
  "email": "john.doe@example.com",
  "fullName": "John Doe",
  "phone": "+1234567890",
  "position": "Software Engineer",
  "departmentId": 1,
  "salary": 75000.00,
  "workingHours": 40.0
}
```

3.2.7 7. Get My Employee ID

- **Endpoint**: `GET /api/hr/employees/me-id`
- **Authorization**: EMPLOYEE
- **Response**: Employee ID (Long)

3.3 Department Management

3.3.1 1. List All Departments

- **Endpoint**: `GET /api/hr/departments`
- **Authorization**: Authenticated

3.3.2 2. Get Department by ID

- **Endpoint**: `GET /api/hr/departments/{id}`
- **Authorization**: Authenticated

3.3.3 3. Create Department

- **Endpoint**: `POST /api/hr/departments`
- **Authorization**: ADMIN, HR
- **Request Body**:

```
{
  "name": "Engineering",
  "description": "Software development department"
}
```

3.3.4 4. Update Department

- **Endpoint**: `PUT /api/hr/departments/{id}`
- **Authorization**: ADMIN, HR

3.3.5 5. Delete Department

- **Endpoint**: `DELETE /api/hr/departments/{id}`
- **Authorization**: ADMIN, HR

3.4 Attendance Management

3.4.1 1. Mark Attendance

- **Endpoint**: `POST /api/hr/attendance`
- **Authorization**: EMPLOYEE
- **Request Body**:

```
{
  "date": "2024-01-15",
  "present": true,
  "workingHours": 8.0
}
```

3.4.2 2. Get Attendance by Employee

- **Endpoint**: `GET /api/hr/attendance/{employeeId}`
- **Authorization**: ADMIN, HR, EMPLOYEE (own records only)

3.4.3 3. Get My Attendance

- **Endpoint**: `GET /api/hr/attendance/me`
- **Authorization**: EMPLOYEE

3.5 Leave Management

3.5.1 1. Create Leave Request

- **Endpoint**: `POST /api/hr/leaves`
- **Authorization**: EMPLOYEE
- **Request Body**:

```
{
    "startDate": "2024-02-01",
    "endDate": "2024-02-05",
    "reason": "Vacation"
}
```

3.5.2 2. List All Leave Requests

- **Endpoint**: `GET /api/hr/leaves`
- **Authorization**: ADMIN, HR

3.5.3 3. Get Leaves by Employee

- **Endpoint**: `GET /api/hr/leaves/{employeeId}`
- **Authorization**: ADMIN, HR, EMPLOYEE (own records only)

3.5.4 4. Get My Leaves

- **Endpoint**: `GET /api/hr/leaves/me`
- **Authorization**: EMPLOYEE

3.5.5 5. Approve Leave Request

- **Endpoint**: `PUT /api/hr/leaves/{id}/approve`
- **Authorization**: ADMIN, HR

3.5.6 6. Reject Leave Request

- **Endpoint**: `PUT /api/hr/leaves/{id}/reject`
- **Authorization**: ADMIN, HR

3.6 Job Management

3.6.1 1. List All Jobs

- **Endpoint**: `GET /api/hr/jobs`
- **Authorization**: Authenticated

3.6.2 2. Get Job by ID

- **Endpoint**: `GET /api/hr/jobs/{id}`
- **Authorization**: Authenticated

3.6.3 3. Create Job Posting

- **Endpoint**: `POST /api/hr/jobs`
- **Authorization**: ADMIN, HR
- **Request Body**:

```
{
  "title": "Senior Software Engineer",
  "description": "Job description...",
  "departmentId": 1,
  "status": "OPEN"
}
```

3.6.4 4. Update Job

- **Endpoint**: `PUT /api/hr/jobs/{id}`
- **Authorization**: ADMIN, HR
- **Request Body**: Same as Create

3.6.5 5. Delete Job

- **Endpoint**: `DELETE /api/hr/jobs/{id}`
- **Authorization**: ADMIN, HR

3.7 Performance Management

3.7.1 1. Create Performance Review

- **Endpoint**: `POST /api/hr/performance`
- **Authorization**: ADMIN, HR
- **Request Body**:

```
{
  "employeeId": 1,
  "rating": 4.5,
  "feedback": "Excellent performance",
  "reviewDate": "2024-01-15"
}
```

3.7.2 2. Get Performance by Employee

- **Endpoint**: `GET /api/hr/performance/{employeeId}`
- **Authorization**: ADMIN, HR, EMPLOYEE (own records only)

3.7.3 3. Get My Performance

- **Endpoint**: `GET /api/hr/performance/me`
- **Authorization**: EMPLOYEE

3.8 Offboarding Management

3.8.1 1. Initiate Offboarding

- **Endpoint**: `POST /api/hr/offboarding/initiate`
- **Authorization**: ADMIN, HR
- **Request Body**:

```
{
  "employeeId": 1,
  "lastWorkingDay": "2024-03-01"
}
```

3.8.2 2. Get Offboarding Status

- **Endpoint**: `GET /api/hr/offboarding/{employeeId}`
- **Authorization**: Authenticated

3.8.3 3. Schedule Exit Interview

- **Endpoint**: `POST /api/hr/offboarding/interview`
- **Authorization**: ADMIN, HR

3.8.4 4. Complete Clearance

- **Endpoint**: `PUT /api/hr/offboarding/clearance`
- **Authorization**: ADMIN, HR

3.8.5 5. Generate Exit Documents

- **Endpoint**: `POST /api/hr/offboarding/exit-documents`
- **Authorization**: ADMIN, HR

3.9 AI Integration

3.9.1 1. AI Query

- **Endpoint**: `POST /api/hr/ai/query`
- **Authorization**: ADMIN, HR
- **Purpose**: Query external AI service for HR-related assistance
- **Request Body**:

```
{
  "query": "What are the best practices for employee onboarding?"
}
```

4 Finance Service APIs

4.1 Base Path: `/api/finance`

4.2 Payroll Management

4.2.1 1. Calculate Payroll

- **Endpoint**: `POST /api/finance/payroll/calculate`
- **Authorization**: ADMIN, FINANCE
- **Request Body**:

```
{
  "employeeId": 1,
  "period": "2024-01",
  "baseSalary": 75000.00,
  "bonuses": 5000.00,
  "deductions": 2000.00,
  "Account":"1234 5678 9012 3456  / visa or mastercard"
}
```

- **Response**: `PayrollResponse` with calculated net salary

4.2.2 2. Calculate Payroll (Alternative)

- **Endpoint**: `POST /api/finance/payroll`
- **Authorization**: ADMIN, FINANCE
- Same as above

4.2.3 3. Get Payroll by Employee and Period

- **Endpoint**: `GET /api/finance/payroll/employee/{employeeId}?period={period}`
- **Authorization**: ADMIN, FINANCE, EMPLOYEE (own records only)

4.2.4 4. Get My Payroll

- **Endpoint**: `GET /api/finance/payroll/me?period={period}`
- **Authorization**: EMPLOYEE

4.2.5 5. Get Payrolls by Period

- **Endpoint**: `GET /api/finance/payroll?period={period}`
- **Authorization**: ADMIN, FINANCE

4.3 Invoice Management

4.3.1 1. Create Invoice

- **Endpoint**: `POST /api/finance/invoices`
- **Authorization**: ADMIN, FINANCE
- **Request Body**:

```
{
  "clientName": "Acme Corporation",
  "invoiceDate": "2024-01-15",
  "amount": 50000.00
}
```

4.3.2 2. Get Invoice by ID

- **Endpoint**: `GET /api/finance/invoices/{id}`
- **Authorization**: ADMIN, FINANCE

4.3.3 3. Update Invoice

- **Endpoint**: `PUT /api/finance/invoices/{id}`
- **Authorization**: ADMIN, FINANCE
- **Request Body**: Same as Create

4.3.4 4. Send Invoice

- **Endpoint**: `PUT /api/finance/invoices/{id}/send`
- **Authorization**: ADMIN, FINANCE
- **Purpose**: Mark invoice as sent

4.3.5 5. Mark Invoice as Paid

- **Endpoint**: `PUT /api/finance/invoices/{id}/mark-paid`
- **Authorization**: ADMIN, FINANCE

4.3.6 6. List Invoices

- **Endpoint**: `GET /api/finance/invoices?status={status}`
- **Authorization**: ADMIN, FINANCE
- **Query Parameters**:
  - `status`: Filter by status (DRAFT, SENT, PAID) - optional

4.3.7 7. Delete Invoice

- **Endpoint**: `DELETE /api/finance/invoices/{id}`
- **Authorization**: ADMIN, FINANCE

4.4 Expense Management

4.4.1 1. Submit Expense

- **Endpoint**: `POST /api/finance/expenses`
- **Authorization**: ADMIN, FINANCE, EMPLOYEE
- **Request Body**:

```
{
  "submittedBy": 1,
  "category": "Travel",
  "amount": 500.00,
  "description": "Business trip to client site"
}
```

4.4.2 2. Approve Expense

- **Endpoint**: `PUT /api/finance/expenses/{id}/approve?approverId={approverId}`
- **Authorization**: ADMIN, FINANCE

4.4.3 3. Reject Expense

- **Endpoint**: `PUT /api/finance/expenses/{id}/reject?approverId={approverId}`
- **Authorization**: ADMIN, FINANCE

4.4.4 4. Get Expenses by Submitter

- **Endpoint**: `GET /api/finance/expenses/by-submitter/{employeeId}`
- **Authorization**: ADMIN, FINANCE, EMPLOYEE (own records only)

4.4.5 5. List All Expenses

- **Endpoint**: `GET /api/finance/expenses?status={status}`
- **Authorization**: ADMIN, FINANCE
- **Query Parameters**:
  - `status`: Filter by status (PENDING, APPROVED, REJECTED) - optional

4.4.6 6. Import Expenses (Multipart)

- **Endpoint**: `POST /api/finance/expenses/import`
- **Authorization**: ADMIN, FINANCE, HR
- **Content-Type**: `multipart/form-data`
- **Parameters**:
  - `file`: CSV/Excel file
  - `status`: Default status (default: APPROVED)
  - `dateFormat`: Date format (default: M/d/yyyy)
  - `mode`: Import mode - upsert or insert (default: upsert)

4.4.7 7. Import Expenses (Binary)

- **Endpoint**: `POST /api/finance/expenses/import-bin`
- **Authorization**: ADMIN, FINANCE, HR
- **Content-Type**: `application/octet-stream`, `text/csv`, or Excel MIME type
- **Headers**:
  - `X-Filename`: Optional filename
- **Query Parameters**: Same as multipart endpoint

5 Inventory Service APIs

5.1 Base Path: `/api/inventory`

5.2 Item Management

5.2.1 1. List All Items

- **Endpoint**: `GET /api/inventory/items`
- **Authorization**: Authenticated

5.2.2 2. Get Item by ID

- **Endpoint**: `GET /api/inventory/items/{id}`
- **Authorization**: Authenticated

5.2.3 3. Create Item

- **Endpoint**: `POST /api/inventory/items`
- **Authorization**: ADMIN, INVENTORY
- **Request Body**:

```
{
  "sku": "ITEM-001",
  "name": "Laptop",
  "description": "15-inch laptop",
  "unit": "pcs",
  "reorderLevel": 10.0
}
```

5.2.4 4. Update Item

- **Endpoint**: `PUT /api/inventory/items/{id}`
- **Authorization**: ADMIN, INVENTORY

5.2.5 5. Delete Item

- **Endpoint**: `DELETE /api/inventory/items/{id}`
- **Authorization**: ADMIN

5.3 Warehouse Management

5.3.1 1. List All Warehouses

- **Endpoint**: `GET /api/inventory/warehouses`
- **Authorization**: Authenticated

5.3.2 2. Get Warehouse by ID

- **Endpoint**: `GET /api/inventory/warehouses/{id}`
- **Authorization**: Authenticated

5.3.3 3. Create Warehouse

- **Endpoint**: `POST /api/inventory/warehouses`
- **Authorization**: ADMIN, INVENTORY
- **Request Body**:

```
{
  "name": "Main Warehouse",
  "location": "Building A, Floor 2"
}
```

5.3.4 4. Update Warehouse

- **Endpoint**: `PUT /api/inventory/warehouses/{id}`
- **Authorization**: ADMIN, INVENTORY

5.3.5 5. Delete Warehouse

- **Endpoint**: `DELETE /api/inventory/warehouses/{id}`
- **Authorization**: ADMIN

5.4 Inventory Operations

5.4.1 1. Record Movement

- **Endpoint**: `POST /api/inventory/movements`
- **Authorization**: ADMIN, INVENTORY
- **Request Body**:

```
{
  "itemId": 1,
  "warehouseId": 1,
  "type": "IN",
  "quantity": 100.0,
  "reference": "PO–12345"
}
```

- **Movement Types**: `IN`, `OUT`

5.4.2 2. Get Stock Level

- **Endpoint**: `GET /api/inventory/stock?itemId={itemId}&warehouseId={warehouseId}`
- **Authorization**: ADMIN, INVENTORY
- **Query Parameters**:
  - `itemId`: Required
  - `warehouseId`: Optional (if omitted, returns total across all warehouses)

5.4.3 3. Get Low Stock Items

- **Endpoint**: `GET /api/inventory/low-stock`
- **Authorization**: ADMIN, INVENTORY
- **Response**: Array of items below reorder level

5.4.4 4. List All Movements

- **Endpoint**: `GET /api/inventory/movements`
- **Authorization**: ADMIN, INVENTORY

6 Reporting Service APIs

6.1 Base Path: `/api/report`

6.1.1 1. Service Health Check

- **Endpoint**: `GET /api/report/test`
- **Authorization**: Public
- **Response**: Service status message

6.1.2 2. Get Dashboard Analytics

- **Endpoint**: `GET /api/report/dashboard?startDate={date}&endDate={date}&period={period}`
- **Authorization**: ADMIN, MANAGER, HR, FINANCE
- **Query Parameters**:
  - `startDate`: Optional start date filter
  - `endDate`: Optional end date filter
  - `period`: Optional period identifier
- **Response**: `DashboardAnalyticsDto` with aggregated metrics

6.1.3 3. Export Report (Generic)

- **Endpoint**: `POST /api/report/export`
- **Authorization**: ADMIN, MANAGER, HR, FINANCE
- **Request Body**:

```
{
  "type": "PayrollSummary",
  "format": "Pdf",
  "period": "2024-01",
  "startDate": "2024-01-01",
  "endDate": "2024-01-31",
  "employeeId": 1
}
```

- **Report Types**: `PayrollSummary`, `AttendanceReport`, `FinancialSummary`, `EmployeeReport`
- **Export Formats**: `Pdf`, `Excel`
- **Response**: File download (PDF or Excel)

6.1.4 4. Export Payroll Report

- **Endpoint**: `GET /api/report/export/payroll?period={period}&format={format}`
- **Authorization**: ADMIN, MANAGER, FINANCE
- **Query Parameters**:
  - `period`: Required (e.g., "2024-01")
  - `format`: Pdf or Excel (default: Pdf)

6.1.5 5. Export Attendance Report

- **Endpoint**: `GET /api/report/export/attendance?startDate={date}&endDate={date}&employeeId={`
- **Authorization**: ADMIN, MANAGER, HR
- **Query Parameters**:
  - `startDate`: Optional (default: 1 month ago)
  - `endDate`: Optional (default: today)
  - `employeeId`: Optional
  - `format`: Pdf or Excel (default: Pdf)

6.1.6 6. Export Financial Report

- **Endpoint**: `GET /api/report/export/financial?startDate={date}&endDate={date}&format={forma`
- **Authorization**: ADMIN, MANAGER, FINANCE
- **Query Parameters**:
  - `startDate`: Optional (default: 1 month ago)
  - `endDate`: Optional (default: today)
  - `format`: Pdf or Excel (default: Pdf)

6.1.7 7. Export Employee Report

- **Endpoint**: `GET /api/report/export/employee/{employeeId}?startDate={date}&endDate={date}&f`
- **Authorization**: ADMIN, MANAGER, HR, FINANCE, EMPLOYEE (own records only)
- **Query Parameters**:
  - `startDate`: Optional (default: 6 months ago)
  - `endDate`: Optional (default: today)
  - `format`: Pdf or Excel (default: Pdf)

7 Summary

This API documentation provides comprehensive details for all endpoints available in the Konecta ERP System. The system implements RESTful APIs with JWT-based authentication and role-based access control, ensuring secure access to various business functions across HR, Finance, Inventory, and Reporting domains.

All endpoints follow consistent patterns:

- **Authentication**: JWT tokens in Authorization headers
- **Authorization**: Role-based access control (RBAC)
- **Data Format**: JSON for request and response bodies
- **Error Handling**: Standard HTTP status codes
- **Versioning**: API versioning through path prefixes