# Payroll Execution Integration Guide

## HR System - Collection Relationships & Subsystem Dependencies

**Version:** 1.0
**Date:** December 15, 2025
**Module Owner:** Payroll Execution Team

---

## Table of Contents



---

## 1. System Overview

The HR System is divided into the following subsystems/modules:

| Module | Owner Team | Primary Responsibility |
|---|---|---|
| **Employee Profile** | Team A | Employee master data, system roles |
| **Organization Structure** | Team B | Departments, positions, assignments |
| **Recruitment** | Team C | Job requisitions, applications, interviews, offers |
| **Onboarding** | Team D | Contract signing, task checklists, document collection |
| **Offboarding** | Team E | Termination/resignation requests, clearance |
| **Leaves** | Team F | Leave types, requests, balances, entitlements |
| **Time Management** | Team G | Attendance, shifts, overtime, lateness |
| **Performance** | Team H | Appraisals, cycles, templates, records |
| **Payroll Configuration** | Team I | Tax rules, insurance, allowances, pay grades |
| **Payroll Execution** | **YOUR TEAM** | Payroll runs, payslips, signing bonuses, termination benefits |
| **Payroll Tracking** | Team K | Disputes, claims, refunds, employee payroll history |

---

# 2. Collection Ownership Map

## Collections by Module

```
┌─────────────────────────────────────────────────────────────────┐
│                        EMPLOYEE PROFILE (Team A)                │
├─────────────────────────────────────────────────────────────────┤
│ • employee_profiles        - Master employee data               │
│ • employee_system_roles    - Authentication & role assignments  │
│ • employee_qualifications  - Education, certifications          │
│ • employeedocuments        - Personal documents                 │
│ • employee_profile_change_requests - Profile update requests   │
└─────────────────────────────────────────────────────────────────┘

┌─────────────────────────────────────────────────────────────────┐
│                      ORGANIZATION STRUCTURE (Team B)            │
├─────────────────────────────────────────────────────────────────┤
│ • departments              - Company departments                │
│ • positions                - Job positions/titles               │
│ • position_assignments     - Employee-Position mappings         │
│ • structure_change_requests - Org change requests               │
│ • structure_change_logs    - Audit trail                        │
│ • structure_approvals      - Approval workflow                  │
└─────────────────────────────────────────────────────────────────┘

┌─────────────────────────────────────────────────────────────────┐
│                        RECRUITMENT (Team C)                     │
├─────────────────────────────────────────────────────────────────┤
│ • jobtemplates             - Reusable job descriptions          │
│ • jobrequisitions          - Open positions                     │
│ • candidates               - Applicant profiles                 │
│ • applications             - Job applications                   │
│ • applicationstatushistories - Application tracking             │
│ • interviews               - Interview schedules                │
│ • referrals                - Employee referrals                 │
│ • offers                   - Job offers                         │
└─────────────────────────────────────────────────────────────────┘

┌─────────────────────────────────────────────────────────────────┐
│                        ONBOARDING (Team D)                      │
├─────────────────────────────────────────────────────────────────┤
│ • onboardings              - Onboarding process tracking        │
│ • contracts                - Employment contracts               │
│ • documents                - Onboarding documents               │
└─────────────────────────────────────────────────────────────────┘

┌─────────────────────────────────────────────────────────────────┐
│                        OFFBOARDING (Team E)                     │
├─────────────────────────────────────────────────────────────────┤
│ • terminationrequests      - Termination/resignation requests   │
│ • clearancechecklists      - Exit clearance tracking            │
└─────────────────────────────────────────────────────────────────┘

┌─────────────────────────────────────────────────────────────────┐
│                          LEAVES (Team F)                        │
├─────────────────────────────────────────────────────────────────┤
│ • leavetypes               - Annual, sick, maternity, etc.      │
│ • leavecategories          - Category groupings                 │
│ • leavepolicies            - Leave rules and limits             │
│ • leaverequests            - Employee leave applications        │
│ • leavebalances            - Current balance tracking           │
│ • leaveentitlements        - Allocated leave days               │
│ • leaveadjustments         - Manual balance adjustments         │
│ • leave_deductions         - Deduction records for payroll      │
│ • attachments              - Medical certificates, etc.         │
│ • calendars                - Holiday calendars                  │
└─────────────────────────────────────────────────────────────────┘

┌─────────────────────────────────────────────────────────────────┐
│                      TIME MANAGEMENT (Team G)                   │
├─────────────────────────────────────────────────────────────────┤
│ • attendancerecords        - Daily punch in/out                 │
│ • shifts                   - Shift definitions                  │
│ • shifttypes               - Shift type configurations          │
│ • shiftassignments         - Employee shift schedules           │
```

```
| • schedulerules            - Scheduling rules                                |
| • overtimerules            - OT calculation rules                            |
| • latenessrules            - Late arrival penalties                          |
| • timeexceptions           - Attendance exceptions                           |
| • attendancecorrectionrequests - Correction requests                         |
| • holidays                 - Company holidays                                |
└──────────────────────────────────────────────────────────────────────────────┘

┌──────────────────────────────────────────────────────────────────────────────┐
│                          PERFORMANCE (Team H)                                │
├──────────────────────────────────────────────────────────────────────────────┤
| • appraisal_templates      - Review form templates                           |
| • appraisal_cycles         - Review periods                                  |
| • appraisal_assignments    - Employee-reviewer mappings                      |
| • appraisal_records        - Completed reviews                               |
| • appraisal_disputes       - Appeal/dispute records                          |
| • assessmentresults        - Assessment scores                               |
└──────────────────────────────────────────────────────────────────────────────┘

┌──────────────────────────────────────────────────────────────────────────────┐
│                      PAYROLL CONFIGURATION (Team I)                          │
├──────────────────────────────────────────────────────────────────────────────┤
| • taxrules                 - Tax calculation rules                           |
| • taxbrackets              - Tax rate brackets                               |
| • insurancebrackets        - Insurance contribution rates                    |
| • allowancetypes           - Allowance definitions                           |
| • allowances               - Configured allowances                           |
| • paytypes                 - Salary/hourly/commission                        |
| • paygrades                - Salary grades/bands                             |
| • deductiontypes           - Deduction categories                            |
| • benefittypes             - Benefit definitions                             |
| • payrollpolicies          - General payroll rules                           |
| • payrollconfigs           - System configurations                           |
| • bonusconfigs             - Bonus calculation configs                       |
| • signingbonus             - Signing bonus configurations                    |
| • signingbonuses           - Signing bonus policies                          |
| • terminationandresignationbenefits - Termination benefit configs            |
| • companywidesettings      - Global company settings                         |
└──────────────────────────────────────────────────────────────────────────────┘

┌──────────────────────────────────────────────────────────────────────────────┐
│                   ⭐ PAYROLL EXECUTION (YOUR TEAM) ⭐                        │
├──────────────────────────────────────────────────────────────────────────────┤
| • payrollruns              - Monthly payroll cycles                          |
| • employeepayrolldetails   - Individual employee calculations                |
| • payslips                 - Generated payslips                              |
| • employeesigningbonus     - Employee-specific signing bonuses               |
| • employeeterminationresignations - Employee termination benefits            |
| • employeeallowances       - Employee allowance assignments                  |
| • employeebenefits         - Employee benefit assignments                    |
└──────────────────────────────────────────────────────────────────────────────┘

┌──────────────────────────────────────────────────────────────────────────────┐
│                       PAYROLL TRACKING (Team K)                              │
├──────────────────────────────────────────────────────────────────────────────┤
| • disputes                 - Payroll disputes                                |
| • claims                   - Employee claims                                 |
| • refunds                  - Approved refunds                                |
| • employeepenalties        - Misconduct penalties                            |
| • notificationlogs         - System notifications                            |
└──────────────────────────────────────────────────────────────────────────────┘
```

---

# 3. Payroll Execution Collections (Your Module)

## 3.1 payrollruns

**Purpose:** Tracks monthly payroll processing cycles

```
{
  _id: ObjectId,
  runId: String,                    // Unique run identifier "PR-2025-12-1234567890"
  payrollPeriod: Date,              // End of payroll period
  status: String,                   // "draft" | "under review" | "pending finance approval" |
                                    // "approved" | "rejected" | "locked"

  // Department/Entity Info
  entity: String,                   // Department name
  entityId: ObjectId,               // → departments._id

  // Employee Count
  employees: Number,
  exceptions: Number,
  irregularitiesCount: Number,
  irregularities: [String],         // Array of flagged issues

  // Financial Totals
  totalBaseSalary: Number,
  totalAllowances: Number,
  totalOvertime: Number,
  totalGrossPay: Number,
  totalTaxDeductions: Number,
  totalInsuranceDeductions: Number,
  totalPenalties: Number,
  totalDeductions: Number,
  totalRefunds: Number,
  totalnetpay: Number,

  // Workflow
  payrollSpecialistId: ObjectId,    // → employee_system_roles._id (creator)
  payrollManagerId: ObjectId,       // → employee_system_roles._id (manager)

  // Approvals
  approvedByManager: Boolean,
  approvedByManagerAt: Date,
  managerApprovalDate: Date,
  approvedByFinance: Boolean,
  approvedByFinanceAt: Date,

  // Freeze/Lock
  frozen: Boolean,
  frozenAt: Date,
  frozenBy: ObjectId,
  frozenReason: String,
  unfrozenAt: Date,
  unfrozenBy: ObjectId,
  unfrozenReason: String,

  // Payslips
  payslipsGenerated: Boolean,
  payslipsGeneratedAt: Date,

  paymentStatus: String,            // "pending" | "processing" | "paid"

  createdAt: Date,
  updatedAt: Date
}
```

**Reference IDs:**

- `entityId` → `departments._id`
- `payrollSpecialistId` → `employee_system_roles._id`
- `payrollManagerId` → `employee_system_roles._id`

---

## 3.2 employeepayrolldetails

**Purpose:** Individual employee payroll calculations for a run

```
{
  _id: ObjectId,
  payrollRunId: ObjectId,           // → payrollruns._id
  employeeId: ObjectId,             // → employee_profiles._id

  // Time Period
  payrollPeriod: Date,
  daysInMonth: Number,
  daysWorked: Number,

  // Earnings
  baseSalary: Number,
  allowances: Number,
  overtime: Number,
  bonuses: Number,
  benefits: Number,
  refunds: Number,
  grossSalary: Number,

  // Deductions
  taxAmount: Number,
  insuranceAmount: Number,
  penalties: Number,
  deductions: Number,               // Total (tax + insurance) excluding penalties

  // Net
  netSalary: Number,                // grossSalary - deductions
  netPay: Number,                   // netSalary - penalties + refunds

  // Signing Bonus (if applicable)
  signingBonusId: ObjectId,         // → employeesigningbonus._id
  signingBonusAmount: Number,

  // Termination Benefit (if applicable)
  terminationBenefitId: ObjectId,   // → employeeterminationresignations._id
  terminationBenefitAmount: Number,

  // Time Management Integration
  attendanceData: {
    scheduledWorkMinutes: Number,
    actualWorkMinutes: Number,
    missingWorkMinutes: Number,
    latenessMinutes: Number,
    overtimeMinutes: Number
  },

  // Detailed Breakdown
  earningsDetails: {
    baseSalary: Number,
    allowances: [{
      name: String,
      amount: Number,
      _id: ObjectId
    }],
    bonuses: [Object],
    benefits: [Object],
    refunds: [Object]
  },

  deductionsDetails: {
    taxes: [{
      name: String,
      rate: Number,
      _id: ObjectId
    }],
    insurances: [{
      name: String,
      employeeRate: Number,
      amount: Number,
      _id: ObjectId
    }],
    taxAmount: Number,
    insuranceAmount: Number,
    penaltiesAmount: Number
  },

  // Exceptions
  exceptions: [String],
  hasExceptions: Boolean,
```

```
  status: String,                  // "calculated" | "reviewed" | "approved"

  createdAt: Date,
  updatedAt: Date
}
```

**Reference IDs:**

- `payrollRunId` → `payrollruns._id`
- `employeeId` → `employee_profiles._id`
- `signingBonusId` → `employeesigningbonus._id`
- `terminationBenefitId` → `employeeterminationresignations._id`

---

## 3.3 payslips

**Purpose:** Generated payslips for employees

```
{
  _id: ObjectId,
  employeeId: ObjectId,             // → employee_profiles._id
  payrollRunId: ObjectId,           // → payrollruns._id

  // Earnings Details
  earningsDetails: {
    baseSalary: Number,
    allowances: [{
      name: String,
      amount: Number,
      status: String,
      _id: ObjectId
    }],
    bonuses: [Object],
    benefits: [Object],
    refunds: [Object]
  },

  // Deductions Details
  deductionsDetails: {
    taxes: [{
      name: String,
      rate: Number,
      status: String,
      _id: ObjectId
    }],
    insurances: [{
      name: String,
      amount: Number,
      employeeRate: Number,
      employerRate: Number,
      minSalary: Number,
      maxSalary: Number,
      status: String,
      _id: ObjectId
    }],
    taxAmount: Number,
    insuranceAmount: Number,
    penaltiesAmount: Number,
    unpaidLeaveAmount: Number,
    lateDeductionAmount: Number,
    otherDeductions: Number
  },

  // Totals
  totalGrossSalary: Number,
  totaDeductions: Number,           // Note: typo in schema, kept for compatibility
  netPay: Number,

  // Status
  paymentStatus: String,            // "pending" | "processing" | "paid"

  // Distribution
  distributedAt: Date,
  distributedVia: String,           // "portal" | "email" | "pdf"

  createdAt: Date,
  updatedAt: Date
}
```

### Reference IDs:

- `employeeId` → `employee_profiles._id`
- `payrollRunId` → `payrollruns._id`

---

## 3.4 employeesigningbonus

**Purpose:** Employee-specific signing bonus records

```
{
  _id: ObjectId,
  employeeId: ObjectId,             // → employee_profiles._id

  // From Onboarding/Contract
  contractId: ObjectId,             // → contracts._id
  onboardingId: ObjectId,           // → onboardings._id

  // Bonus Details
  amount: Number,                   // Original configured amount
  givenAmount: Number,              // Actual amount (can be edited)

  // Approval Workflow
  status: String,                   // "pending" | "approved" | "rejected" | "paid"
  approvedBy: ObjectId,             // → employee_system_roles._id
  approvedAt: Date,
  rejectedBy: ObjectId,
  rejectedAt: Date,
  rejectionReason: String,

  // Payment
  paymentDate: Date,
  payrollRunId: ObjectId,           // → payrollruns._id (when paid)

  createdBy: ObjectId,              // → employee_system_roles._id
  createdAt: Date,
  updatedAt: Date
}
```

### Reference IDs:

- `employeeId` → `employee_profiles._id`
- `contractId` → `contracts._id`
- `onboardingId` → `onboardings._id`
- `approvedBy` → `employee_system_roles._id`
- `payrollRunId` → `payrollruns._id`

---

## 3.5 employeeterminationresignations

**Purpose:** Employee termination/resignation benefit records

```
{
  _id: ObjectId,
  employeeId: ObjectId,             // → employee_profiles._id

  // From Offboarding
  terminationRequestId: ObjectId,   // → terminationrequests._id

  // Benefit Details
  type: String,                     // "termination" | "resignation"
  reason: String,
  lastWorkingDay: Date,
  yearsOfService: Number,

  // Financial
  amount: Number,                   // Calculated benefit amount
  givenAmount: Number,              // Actual amount (can be edited)

  // Components
  unpaidSalary: Number,
  unusedLeaveCompensation: Number,
  severancePay: Number,
  otherBenefits: Number,

  // Approval Workflow
  status: String,                   // "pending" | "approved" | "rejected" | "paid"
  approvedBy: ObjectId,             // → employee_system_roles._id
  approvedAt: Date,
  rejectedBy: ObjectId,
  rejectedAt: Date,
  rejectionReason: String,

  // Payment
  paymentDate: Date,
  payrollRunId: ObjectId,           // → payrollruns._id (when paid)

  createdBy: ObjectId,
  createdAt: Date,
  updatedAt: Date
}
```

**Reference IDs:**

- `employeeId` → `employee_profiles._id`
- `terminationRequestId` → `terminationrequests._id`
- `approvedBy` → `employee_system_roles._id`
- `payrollRunId` → `payrollruns._id`

---

## 3.6 employeeallowances

**Purpose:** Employee-specific allowance assignments

```
{
  _id: ObjectId,
  employeeId: ObjectId,             // → employee_profiles._id
  allowanceTypeId: ObjectId,        // → allowancetypes._id OR allowances._id

  name: String,
  amount: Number,
  effectiveDate: Date,
  endDate: Date,

  status: String,                   // "active" | "inactive"

  createdBy: ObjectId,
  createdAt: Date,
  updatedAt: Date
}
```

**Reference IDs:**

- `employeeId` → `employee_profiles._id`
- `allowanceTypeId` → `allowancetypes._id` OR `allowances._id`

---

### 3.7 employeebenefits

**Purpose:** Employee-specific benefit assignments

```
{
  _id: ObjectId,
  employeeId: ObjectId,             // → employee_profiles._id
  benefitTypeId: ObjectId,          // → benefittypes._id

  name: String,
  amount: Number,
  effectiveDate: Date,
  endDate: Date,

  status: String,                   // "active" | "inactive"

  createdBy: ObjectId,
  createdAt: Date,
  updatedAt: Date
}
```

**Reference IDs:**

- `employeeId` → `employee_profiles._id`
- `benefitTypeId` → `benefittypes._id`

---

## 4. Upstream Dependencies (Data You Need)

These are collections managed by OTHER teams that Payroll Execution READS from:

### 4.1 From Employee Profile (Team A)

| Collection | Fields Needed | Purpose |
|---|---|---|
| `employee_profiles` | `_id`, `firstName`, `lastName`, `workEmail`, `employeeNumber`, `status`, `hireDate`, `departmentId`, `positionId`, `payGradeCode`, `baseSalary` | Employee master data for payroll calculation |
| `employee_system_roles` | `_id`, `employeeProfileId`, `roles`, `isActive` | Authentication & role verification |

**Required API from Team A:**

```
// GET /employee-profile/active?departmentId=xxx
// Returns: Array of active employees in department

// GET /employee-profile/:id
// Returns: Full employee profile with salary info
```

---

### 4.2 From Organization Structure (Team B)

| Collection | Fields Needed | Purpose |
|---|---|---|
| `departments` | `_id`, `name`, `isActive` | Department filtering for payroll runs |
| `positions` | `_id`, `title`, `payGradeId` | Position information |

**Required API from Team B:**

```
// GET /organization-structure/departments?isActive=true
// Returns: Array of active departments for payroll initiation dropdown
```

---

### 4.3 From Onboarding (Team D)

| Collection | Fields Needed | Purpose |
|---|---|---|
| `contracts` | `_id`, `employeeId`, `signingBonusAmount`, `signingBonusFlag` | Signing bonus trigger |
| `onboardings` | `_id`, `employeeId`, `status`, `startDate` | New hire detection |

**Required API from Team D:**

```
// POST /onboarding/contracts/:contractId/process-signing-bonus
// Triggers: Creates employeesigningbonus record for payroll execution
```

---

## 4.4 From Offboarding (Team E)

| Collection | Fields Needed | Purpose |
|---|---|---|
| `terminationrequests` | `_id`, `employeeId`, `status`, `terminationType`, `lastWorkingDay` | Termination detection |

**Required API from Team E:**

```
// GET /offboarding/termination-requests?status=approved
// Returns: Approved terminations needing benefit calculation

// POST /offboarding/trigger-final-settlement
// Triggers: Creates employeeterminationresignations for payroll execution
```

---

## 4.5 From Leaves (Team F)

| Collection | Fields Needed | Purpose |
|---|---|---|
| `leaverequests` | `employeeId`, `status`, `startDate`, `endDate`, `leaveTypeId`, `isPaid` | Unpaid leave deductions |
| `leavebalances` | `employeeId`, `balance`, `used` | Unused leave compensation |

**Required API from Team F:**

```
// GET /leaves/employees/:employeeId/balances
// Returns: Current leave balances for encashment calculation

// POST /leaves/payroll/calculate-unpaid-deduction
// Body: { employeeId, startDate, endDate }
// Returns: Unpaid leave days and deduction amount
```

---

## 4.6 From Time Management (Team G)

| Collection | Fields Needed | Purpose |
|---|---|---|
| `attendancerecords` | `employeeId`, `date`, `punchIn`, `punchOut`, `workMinutes`, `lateMinutes`, `overtimeMinutes` | Actual time worked |
| `overtimerules` | `multiplier`, `status` | OT pay calculation |
| `latenessrules` | `deductionRate` | Late penalty calculation |

**Required API from Team G:**

```
// GET /attendance/payroll?employeeId=xxx&startDate=xxx&endDate=xxx
// Returns: Attendance summary for payroll period
{
  scheduledWorkMinutes: number,
  actualWorkMinutes: number,
  missingWorkMinutes: number,
  latenessMinutes: number,
  overtimeMinutes: number
}
```

---

## 4.7 From Payroll Configuration (Team I)

| Collection | Fields Needed | Purpose |
|---|---|---|
| `taxrules` | `name`, `rate`, `status` | Tax calculation |
| `insurancebrackets` | `minSalary`, `maxSalary`, `employeeRate`, `employerRate`, `status` | Insurance deduction |
| `allowances` | `_id`, `name`, `amount`, `status` | Allowance configuration |
| `paygrades` | `code`, `baseSalary`, `minSalary`, `maxSalary` | Salary bands |

### Required API from Team I:

```
// GET /payroll-configuration-requirements/tax-rules?status=approved
// Returns: Active tax rules for calculation

// GET /payroll-configuration-requirements/insurance-brackets?status=approved
// Returns: Active insurance brackets

// GET /payroll-configuration-requirements/allowances/all?status=approved
// Returns: Active allowances

// GET /payroll-configuration-requirements/pay-grades
// Returns: All pay grades for salary lookup
```

---

## 4.8 From Payroll Tracking (Team K)

| Collection | Fields Needed | Purpose |
|---|---|---|
| `refunds` | `employeeId`, `amount`, `status`, `payrollPeriod` | Refunds to add to payroll |
| `employeepenalties` | `employeeId`, `amount`, `type` | Misconduct penalties |

### Required API from Team K:

```
// GET /payroll/tracking/refunds/pending?employeeId=xxx
// Returns: Approved refunds to include in payroll

// GET /payroll/tracking/employee/:employeeId/misconduct-deductions
// Returns: Active penalties to deduct
```

## 5. Downstream Dependencies (Who Needs Your Data)

These modules READ from Payroll Execution collections:

### 5.1 Payroll Tracking (Team K)

**Collections They Read:**

- `payrollruns` - For payroll history display
- `payslips` - For employee payslip viewing
- `employeepayrolldetails` - For salary history

**APIs You Must Provide:**

```
// GET /payroll-execution/runs
// Returns: List of payroll runs (for tracking dashboards)

// GET /payroll-execution/:id/payslips
// Returns: Payslips for a specific run

// GET /payroll-execution/payslips/:payslipId
// Returns: Single payslip details
```

---

### 5.2 Employee Self-Service (Team A)

**Collections They Read:**

- `payslips` - Employees view their own payslips

**APIs You Must Provide:**

```
// GET /payroll/tracking/employee/:employeeId/payslips
// Returns: Employee's payslip history
```

---

### 5.3 Finance Reporting

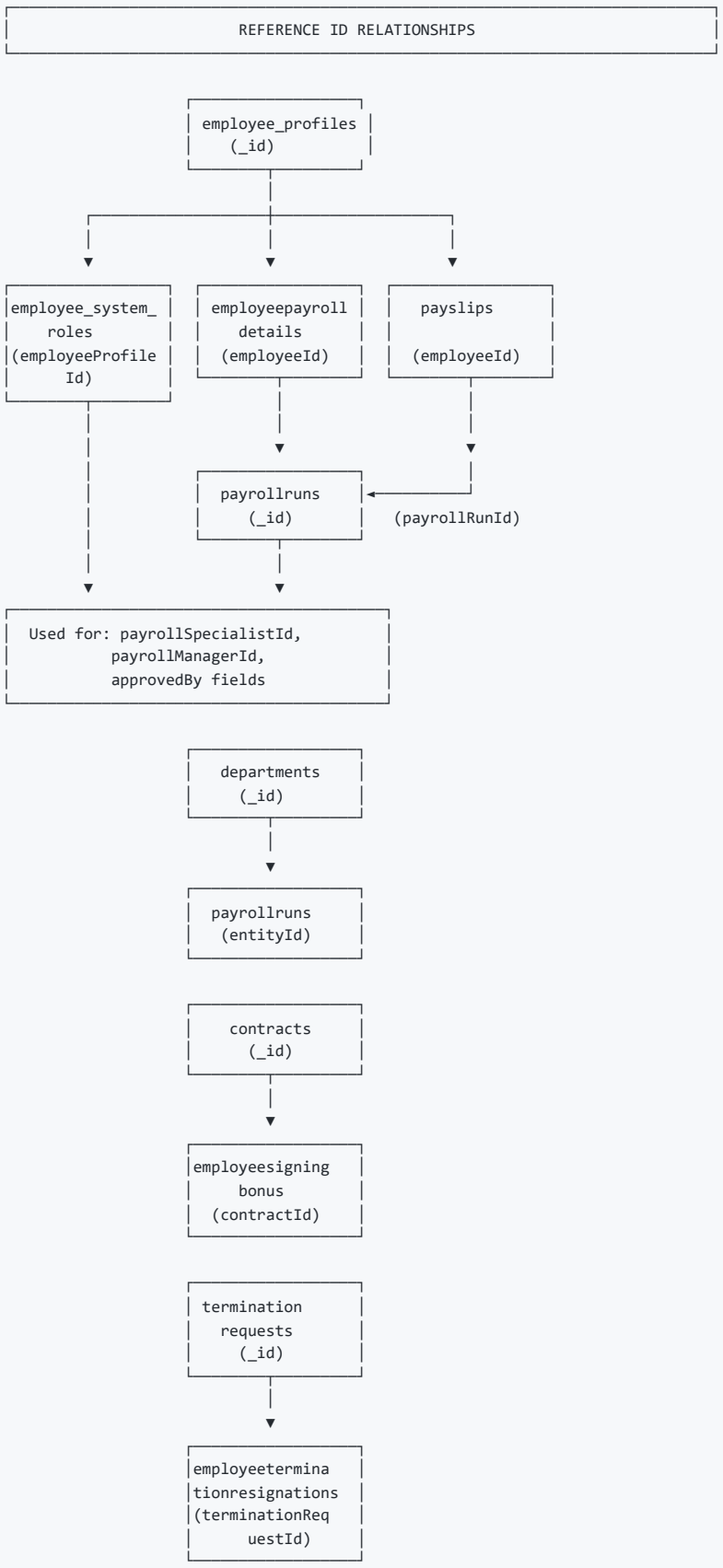
**Collections They Read:**

- `payrollruns` - For financial reports
- `payslips` - For tax/insurance reports

---

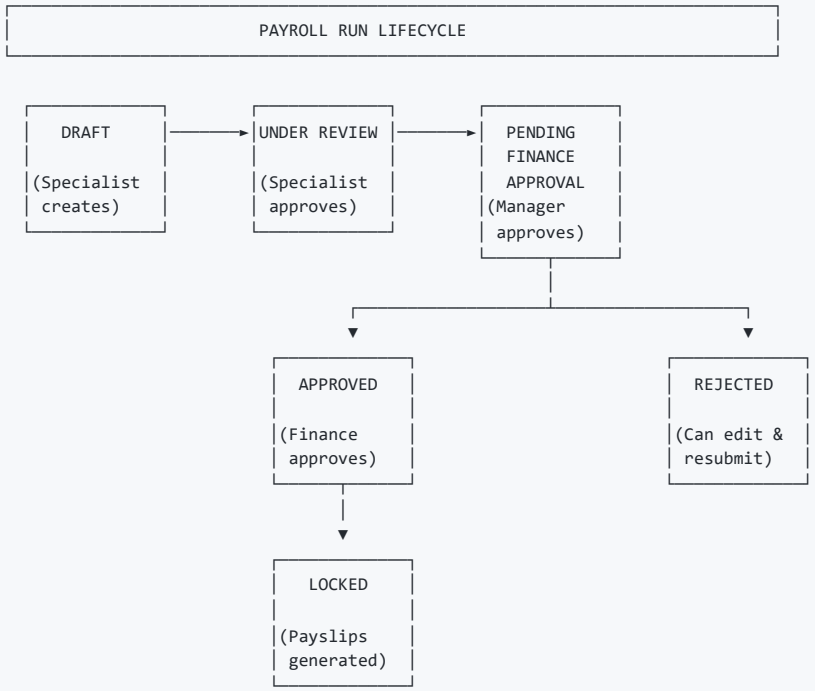
# 6. Collection Schemas & Reference IDs

## Complete Reference ID Map

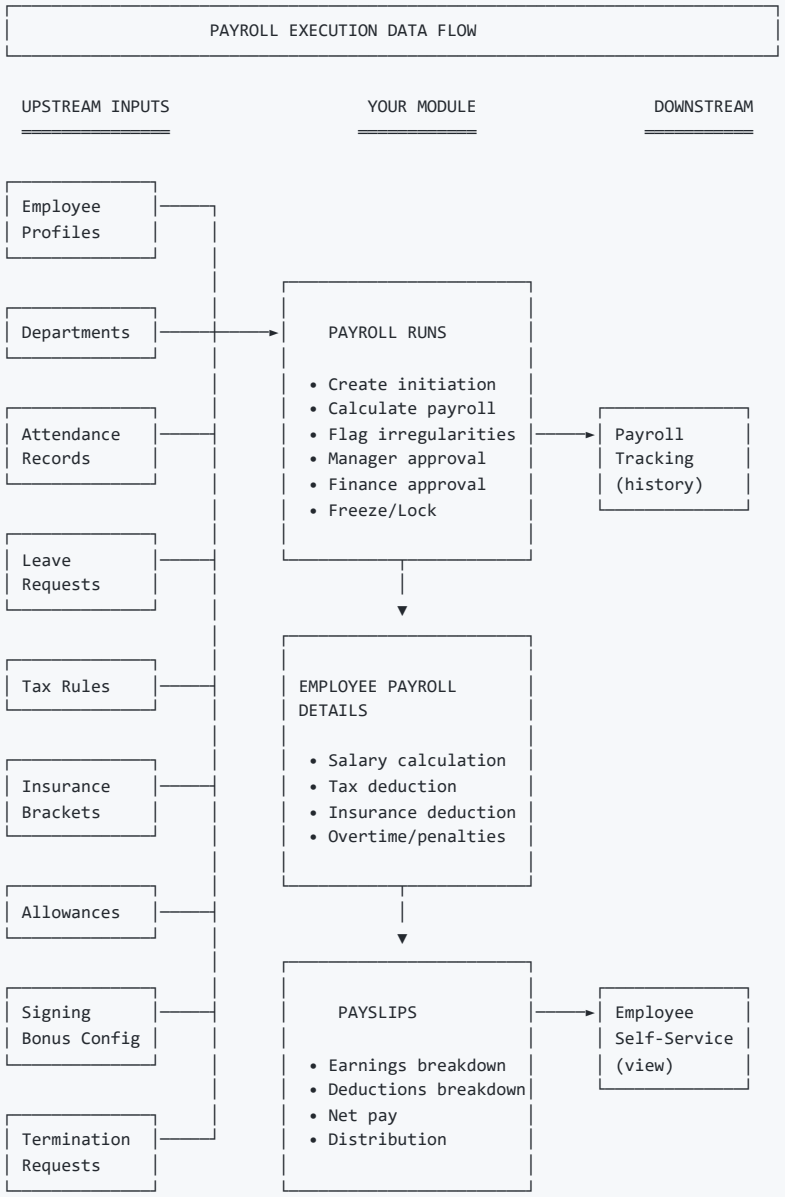


# 7. Data Flow Diagrams

## 7.1 Payroll Run Lifecycle



### 7.2 Data Integration Flow




---

## 8. UI Components Required

### 8.1 Payroll Specialist Dashboard

**Page:** `/dashboard/payroll-specialist/runs`

| Component | Fields | Source Data |
|---|---|---|
| **Create Initiation Modal** | | |
| - Department Dropdown | `departments[]` | `GET /organization-structure/departments` |
| - Period Date Picker | `payrollPeriod` | Date input |
| - Employee Count | `employees` | Auto-calculated |
| | | |
| **Payroll Runs List** | | |
| - Run ID | `runId` | `payrollruns.runId` |
| - Department | `entity` | `payrollruns.entity` |
| - Period | `payrollPeriod` | `payrollruns.payrollPeriod` |
| - Employees | `employees` | `payrollruns.employees` |
| - Status Badge | `status` | `payrollruns.status` |
| - Total Net Pay | `totalnetpay` | `payrollruns.totalnetpay` |
| | | |
| **Run Details View** | | |
| - Financial Summary | `totalGrossPay`, `totalDeductions`, `totalnetpay` | `payrollruns` |
| - Taxes | `totalTaxDeductions` | `payrollruns` |
| - Insurance | `totalInsuranceDeductions` | `payrollruns` |
| - Irregularities | `irregularities[]` | `payrollruns` |
| | | |
| **Signing Bonuses Tab** | | |
| - Employee Name | `employeeName` | Lookup from `employee_profiles` |
| - Amount | `givenAmount` | `employeesigningbonus.givenAmount` |
| - Status | `status` | `employeesigningbonus.status` |
| - Edit Button | | Only if status = “pending” |
| - Approve/Reject | | Action buttons |
| | | |
| **Termination Benefits Tab** | | |

| Component | Fields | Source Data |
|---|---|---|
| - Employee Name | `employeeName` | Lookup from `employee_profiles` |
| - Type | `type` | `employeeterminationresignations.type` |
| - Amount | `givenAmount` | `employeeterminationresignations.givenAmount` |
| - Status | `status` | `employeeterminationresignations.status` |
| | | |
| **Payslips View** | | |
| - Employee Name | `employeeName` | Lookup |
| - Gross Salary | `totalGrossSalary` | `payslips.totalGrossSalary` |
| - Deductions | `totaDeductions` | `payslips.totaDeductions` |
| - Net Pay | `netPay` | `payslips.netPay` |
| - Status Badge | `paymentStatus` | `payslips.paymentStatus` |

---

## 8.2 Payroll Manager Dashboard

**Page:** `/dashboard/payroll-manager/runs`

| Component | Fields | Source Data |
|---|---|---|
| **Runs Pending Approval** | | |
| - Same as Specialist | | Filter: `status = 'under review'` |
| | | |
| **Approval Actions** | | |
| - Approve Button | | Sets `approvedByManager = true` |
| - Reject Button | | Sets `status = 'rejected'` |
| - Rejection Reason | `rejectionReason` | Text input |
| | | |
| **Freeze/Unfreeze** | | |
| - Freeze Button | | Sets `frozen = true` |
| - Unfreeze Button | | Requires `unfreezeReason` |
| - Reason Input | `frozenReason` / `unfrozenReason` | Text input |

---

## 8.3 Finance Staff Dashboard

**Page:** `/dashboard/finance-staff/runs`

| Component | Fields | Source Data |
|---|---|---|
| **Runs Pending Finance Approval** | | |
| - Same as Specialist | | Filter: `status = 'pending finance approval'` |
| | | |
| **Financial Breakdown** | | |
| - Gross Pay | `totalGrossPay` | `payrollruns` |
| - Taxes | `totalTaxDeductions` | `payrollruns` |
| - Insurance | `totalInsuranceDeductions` | `payrollruns` |
| - Other Deductions | calculated | `totalDeductions - tax - insurance` |
| - Net Pay | `totalnetpay` | `payrollruns` |
| | | |
| **Approval Workflow** | | |
| - Step 1: Manager | `approvedByManager` | Boolean display |
| - Step 2: Finance | `approvedByFinance` | Boolean display |
| - Step 3: Payslips | `payslipsGenerated` | Boolean display |
| | | |
| **Generate Payslips Button** | | Only if `status = 'approved'` |

---

## 8.4 Payslip Detail Modal

**Component:** Payslip detail popup (all dashboards)

| Section | Fields | Source |
| --- | --- | --- |
| **Header** | | |
| - Employee Name | `employeeName` | Lookup |
| - Employee Number | `employeeNumber` | Lookup |
| - Period | `payrollPeriod` | `payslips.payrollRunId` → `payrollruns.payrollPeriod` |
| - Entity | `entity` | `payrollruns.entity` |
| - Status Badge | `paymentStatus` | `payslips.paymentStatus` |
| | | |
| **Earnings Section** | | |
| - Base Salary | `earningsDetails.baseSalary` | |
| - Allowances (expandable) | `earningsDetails.allowances[]` | Each with name & amount |
| - Bonuses (expandable) | `earningsDetails.bonuses[]` | |
| - Benefits (expandable) | `earningsDetails.benefits[]` | |
| - Refunds (expandable) | `earningsDetails.refunds[]` | |
| - **Total Gross** | `totalGrossSalary` | |
| | | |
| **Deductions Section** | | |
| - Taxes (expandable) | `deductionsDetails.taxes[]` | Each with name & rate |
| - Tax Amount | `deductionsDetails.taxAmount` | |
| - Insurance (expandable) | `deductionsDetails.insurances[]` | Each with name & rate |
| - Insurance Amount | `deductionsDetails.insuranceAmount` | |
| - Penalties | `deductionsDetails.penaltiesAmount` | |
| - Unpaid Leave | `deductionsDetails.unpaidLeaveAmount` | |
| - Late/Early Leave | `deductionsDetails.lateDeductionAmount` | |
| - **Total Deductions** | `totaDeductions` | |
| | | |

| Section | Fields | Source |
| --- | --- | --- |
| **Net Pay** | `netPay` | Highlighted |
| | | |
| **Metadata** | | |
| - Payslip ID | `_id` | |
| - Generated Date | `createdAt` | |
| - Distributed Date | `distributedAt` | |

# 9. Integration API Contracts

## 9.1 APIs Your Module PROVIDES

```
// ═══════════════════════════════════════════════════════════════════
// PAYROLL EXECUTION APIs
// Base URL: /payroll-execution
// ═══════════════════════════════════════════════════════════════════

// ─────────────────────────────────────────────────────────────────────
// PAYROLL INITIATION
// ─────────────────────────────────────────────────────────────────────

POST /payroll-execution/initiation
// Creates a new payroll run
// Body: { entityId, entity, payrollPeriod, payrollManagerId }
// Returns: Created payroll run

GET /payroll-execution/initiation/:id
// Get payroll run details
// Returns: Full payroll run with all fields

PATCH /payroll-execution/initiation/:id
// Update payroll run (only if DRAFT or REJECTED)
// Body: { payrollPeriod?, entity?, employees? }

POST /payroll-execution/initiation/:id/approve
// Specialist approves → triggers calculation → status = "under review"
// Query: ?approvedBy=userId

POST /payroll-execution/initiation/:id/reject
// Reject payroll run
// Body: { reason }

// ─────────────────────────────────────────────────────────────────────
// MANAGER & FINANCE APPROVAL
// ─────────────────────────────────────────────────────────────────────

POST /payroll-execution/:id/approve
// Manager approval → status = "pending finance approval"
// Query: ?approvedBy=userId

POST /payroll-execution/:id/approve-finance
// Finance approval → status = "approved"
// Query: ?approvedBy=userId

// ─────────────────────────────────────────────────────────────────────
// FREEZE / UNFREEZE
// ─────────────────────────────────────────────────────────────────────

POST /payroll-execution/:id/freeze
// Freeze/lock payroll run
// Query: ?frozenBy=userId

POST /payroll-execution/:id/unfreeze
// Unfreeze payroll run (requires reason)
// Body: { reason }
// Query: ?unfrozenBy=userId

// ─────────────────────────────────────────────────────────────────────
// PAYSLIPS
// ─────────────────────────────────────────────────────────────────────

POST /payroll-execution/:id/generate-payslips
// Generate payslips for all employees in run
// Requires: status = "approved" or "locked"

GET /payroll-execution/:id/payslips
// Get all payslips for a payroll run

GET /payroll-execution/payslips/:payslipId
// Get single payslip details

// ─────────────────────────────────────────────────────────────────────
// SIGNING BONUSES
```

```
// ──────────────────────────────────────────────────────────────────────────

GET /payroll-execution/signing-bonuses
// List all signing bonuses

GET /payroll-execution/signing-bonuses/:id
// Get signing bonus details

POST /payroll-execution/signing-bonuses/:id/edit
// Edit signing bonus amount
// Body: { amount }

POST /payroll-execution/signing-bonuses/:id/approve
// Approve signing bonus
// Query: ?approvedBy=userId

POST /payroll-execution/signing-bonuses/:id/reject
// Reject signing bonus
// Body: { reason }

// ──────────────────────────────────────────────────────────────────────────
// TERMINATION BENEFITS
// ──────────────────────────────────────────────────────────────────────────

GET /payroll-execution/termination-benefits
// List all termination benefits

GET /payroll-execution/termination-benefits/:id
// Get termination benefit details

POST /payroll-execution/termination-benefits/:id/edit
// Edit termination benefit amount
// Body: { amount }

POST /payroll-execution/termination-benefits/:id/approve
// Approve termination benefit

POST /payroll-execution/termination-benefits/:id/reject
// Reject termination benefit
// Body: { reason }

// ──────────────────────────────────────────────────────────────────────────
// SUPPORTING
// ──────────────────────────────────────────────────────────────────────────

GET /payroll-execution/runs
// List all payroll runs (with filtering)
// Query: ?status=xxx&entityId=xxx&page=1&limit=10

GET /payroll-execution/departments
// Get departments for dropdown (proxied from org structure)
```

### 9.2 APIs Your Module CONSUMES

```
// ═══════════════════════════════════════════════════════════════
// REQUIRED FROM OTHER MODULES
// ═══════════════════════════════════════════════════════════════

// From Employee Profile (Team A)
GET /employee-profile/:id
GET /employee-profile/admin/employees?departmentId=xxx&status=active

// From Organization Structure (Team B)
GET /organization-structure/departments?isActive=true

// From Time Management (Team G)
GET /attendance/payroll?employeeId=xxx&startDate=xxx&endDate=xxx

// From Leaves (Team F)
GET /leaves/employees/:employeeId/balances
POST /leaves/payroll/calculate-unpaid-deduction

// From Payroll Configuration (Team I)
GET /payroll-configuration-requirements/tax-rules?status=approved
GET /payroll-configuration-requirements/insurance-brackets?status=approved
GET /payroll-configuration-requirements/allowances/all?status=approved
GET /payroll-configuration-requirements/pay-grades

// From Payroll Tracking (Team K)
GET /payroll/tracking/refunds/pending?employeeId=xxx
GET /payroll/tracking/employee/:employeeId/misconduct-deductions
```

---

## 10. Seed Data Requirements

### 10.1 Pre-requisite Data (Must Exist First)

Before Payroll Execution can function, these collections must be populated:

| Order | Collection | Minimum Records | Owner Team |
|---|---|---|---|
| 1 | `employee_profiles` | At least 2-3 per department | Team A |
| 2 | `employee_system_roles` | 1 per employee + role users | Team A |
| 3 | `departments` | At least 3 (with `isActive: true`) | Team B |
| 4 | `positions` | At least 5 | Team B |
| 5 | `taxrules` | 3-5 tax brackets (status: approved) | Team I |
| 6 | `insurancebrackets` | 2-3 brackets (status: approved) | Team I |
| 7 | `allowances` | 3-5 types (status: approved) | Team I |
| 8 | `paygrades` | 3-5 grades with baseSalary | Team I |

## 10.2 Sample Seed Script for Payroll Execution

```
// scripts/seed_payroll_execution.js

const { MongoClient, ObjectId } = require('mongodb');
require('dotenv').config();

async function seedPayrollExecution() {
  const client = new MongoClient(process.env.MONGODB_URI);
  await client.connect();
  const db = client.db();

  // Get required references
  const departments = await db.collection('departments').find({ isActive: true }).toArray();
  const employees = await db.collection('employee_profiles').find({ status: 'active' }).toArray();
  const systemRoles = await db.collection('employee_system_roles').find({}).toArray();

  const payrollSpecialist = systemRoles.find(r => r.roles?.includes('PAYROLL_SPECIALIST'));
  const payrollManager = systemRoles.find(r => r.roles?.includes('PAYROLL_MANAGER'));

  if (!payrollSpecialist || !payrollManager) {
    console.error('Missing payroll specialist or manager roles!');
    return;
  }

  // 1. Create Payroll Run
  const payrollRun = {
    _id: new ObjectId(),
    runId: `PR-2025-03-${Date.now()}`,
    payrollPeriod: new Date('2025-03-31'),
    status: 'draft',
    entity: departments[0].name,
    entityId: departments[0]._id,
    employees: 0,
    exceptions: 0,
    irregularitiesCount: 0,
    irregularities: [],
    totalBaseSalary: 0,
    totalAllowances: 0,
    totalOvertime: 0,
    totalGrossPay: 0,
    totalTaxDeductions: 0,
    totalInsuranceDeductions: 0,
    totalPenalties: 0,
    totalDeductions: 0,
    totalRefunds: 0,
    totalnetpay: 0,
    payrollSpecialistId: payrollSpecialist._id,
    payrollManagerId: payrollManager._id,
    approvedByManager: false,
    approvedByFinance: false,
    frozen: false,
    payslipsGenerated: false,
    paymentStatus: 'pending',
    createdAt: new Date(),
    updatedAt: new Date()
  };

  await db.collection('payrollruns').insertOne(payrollRun);
  console.log('Created payroll run:', payrollRun.runId);

  // 2. Create Employee Signing Bonus (example)
  const newHire = employees[0];
  if (newHire) {
    const signingBonus = {
      _id: new ObjectId(),
      employeeId: newHire._id,
      amount: 5000,
      givenAmount: 5000,
      status: 'pending',
      createdBy: payrollSpecialist._id,
      createdAt: new Date(),
      updatedAt: new Date()
    };

    await db.collection('employeesigningbonus').insertOne(signingBonus);
    console.log('Created signing bonus for:', newHire.firstName);
  }
```

```
  // 3. Create Employee Allowances
  const allowances = await db.collection('allowances').find({ status: 'approved' }).toArray();

  for (const emp of employees.slice(0, 3)) {
    for (const allowance of allowances.slice(0, 2)) {
      await db.collection('employeeallowances').insertOne({
        _id: new ObjectId(),
        employeeId: emp._id,
        allowanceTypeId: allowance._id,
        name: allowance.name,
        amount: allowance.amount,
        effectiveDate: new Date('2025-01-01'),
        status: 'active',
        createdAt: new Date(),
        updatedAt: new Date()
      });
    }
  }
  console.log('Created employee allowances');

  await client.close();
  console.log('Payroll execution seed complete!');
}

seedPayrollExecution().catch(console.error);
```

---

# Appendix A: Status Enums

## Payroll Run Status

```
enum PayRollStatus {
  DRAFT = 'draft',
  UNDER_REVIEW = 'under review',
  PENDING_FINANCE_APPROVAL = 'pending finance approval',
  APPROVED = 'approved',
  REJECTED = 'rejected',
  LOCKED = 'locked'
}
```

## Bonus/Benefit Status

```
enum BonusStatus {
  PENDING = 'pending',
  APPROVED = 'approved',
  REJECTED = 'rejected',
  PAID = 'paid'
}
```

## Payment Status

```
enum PaymentStatus {
  PENDING = 'pending',
  PROCESSING = 'processing',
  PAID = 'paid'
}
```

---

## Appendix B: Business Rules Reference

| BR ID | Rule | Implementation |
|---|---|---|
| BR 17 | Auto-generated payslip with clear breakdown | `generatePayslips()` creates detailed payslips |
| BR 24 | Signing bonus review workflow | `approveSigningBonus()`, `rejectSigningBonus()` |
| BR 25 | Signing bonus edit capability | `editSigningBonus()` |
| BR 26 | Termination benefits review | `approveTerminationBenefit()`, `rejectTerminationBenefit()` |
| BR 27 | Termination benefits edit | `editTerminationBenefit()` |
| BR 28 | Auto-process signing bonus for new hire | Triggered by onboarding |
| BR 29 | Auto-process termination benefits | Triggered by offboarding |
| BR 30 | Manager and finance approval workflow | Multi-step approval |
| BR 31 | Salary calculations | `calculateEmployeePayroll()` |
| BR 34 | Deductions after gross calculation | Tax + Insurance in `deductionsDetails` |
| BR 63 | Validation checks on initiation | `validateNoDuplicatePayrollPeriod()` |
| BR 64 | Draft version generation | `status = 'draft'` on creation |

---

**Document End**

*For questions or clarifications, contact the Payroll Execution Team.*