EPORDO

EPORDO | Cloud Attendance | Payroll

EPORDO Payroll Complete Package

All-in-one guide for international customers

EPORDO Cloud Attendance - Payroll Module (Complete Package)

For overseas customers | English edition | September 2026

This document combines the full Payroll documentation set:

- Documentation Index
- Overview and Architecture Framework
- Quick Start Guide
- Complete User Guide

EPORDO Cloud Attendance - Payroll Module

Documentation Index (English)

Product: EPORDO Cloud Attendance | Payroll

Audience: HR managers, payroll administrators, implementation partners, overseas customers

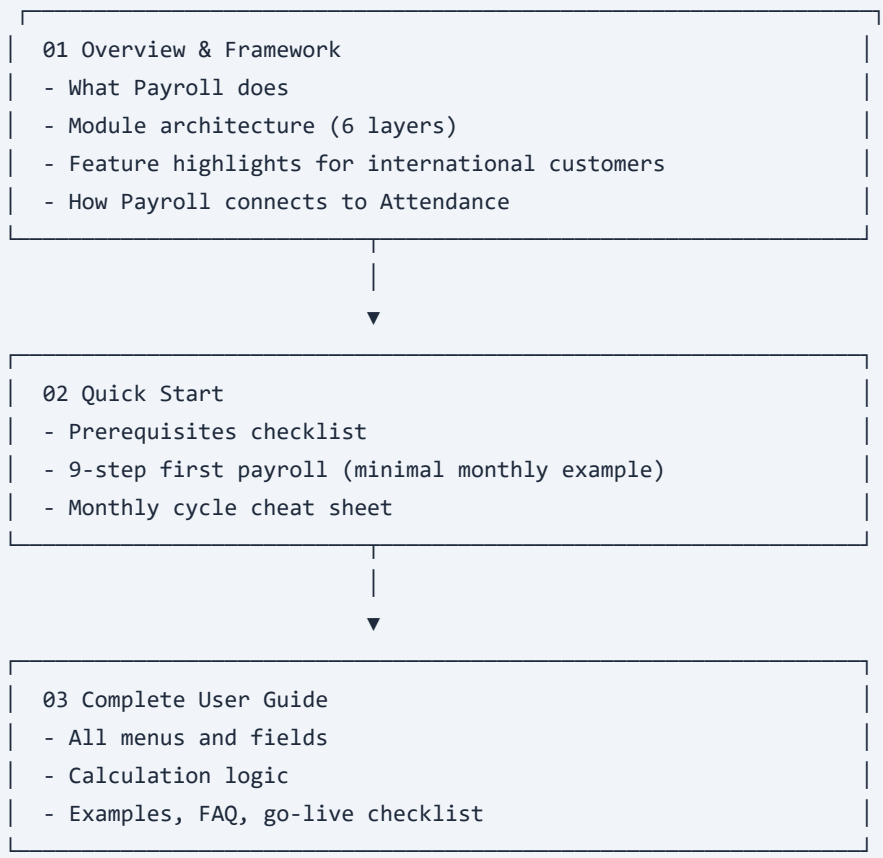
Version: September 2026 | Aligned with current product release

Language: English (primary for international deployment)

How to Use This Documentation

Choose the guide that matches your role and stage:

Documentation Map



Module Location in EPORDO

In the sidebar, open **Payroll** (under Cloud Attendance). Four main areas:

Key Design Principles (International)

1. **Country-agnostic**
   - No hidden default tax or statutory rules. You configure what applies in your jurisdiction.

2. **Configurable without code**
   - Earnings, deductions, formulas, and tax slabs are set in the UI.

3 **Attendance-linked, not hard-wired**
- Time and attendance data is collected per pay period when you need it.

4 **Audit-ready**
- Each pay run stores a configuration snapshot; locked payslips cannot be altered.

5 **Multilingual**
- UI, component names, and PDF payslips support multiple languages.

Support & Technical Reference

For engine-level fields, database schema, and Provider integration details, refer to your implementation partner or internal technical documentation index (`36-payroll-documentation-index.md` if available in your deployment package).

For day-to-day payroll operations, **documents 01-03 above are sufficient**.

EPORDO Payroll - Product Overview & Architecture Framework

> **Audience**: HR leaders, payroll managers, implementation consultants, overseas customers evaluating or deploying EPORDO
>
> **Next step after reading**: 02 - Quick Start Guide

---

1. What Is EPORDO Payroll?

EPORDO Payroll is a **configurable payroll engine** built into **EPORDO Cloud Attendance**. It lets your organization:

6. Define
how pay is calculated
(earnings, deductions, employer costs, tax)

7. Connect to
attendance and HR data
without custom development

8. Run
approved, locked pay cycles
with full audit history

9. Support
monthly, weekly, biweekly, or daily
pay frequencies

10. Operate in
any country
by configuring local rules-not by relying on pre-built country packs

There is **no hidden "default country tax"**. The calculation engine uses exactly what you configure. If a rule is not defined, it does not apply.

---

2. Who Is This For?

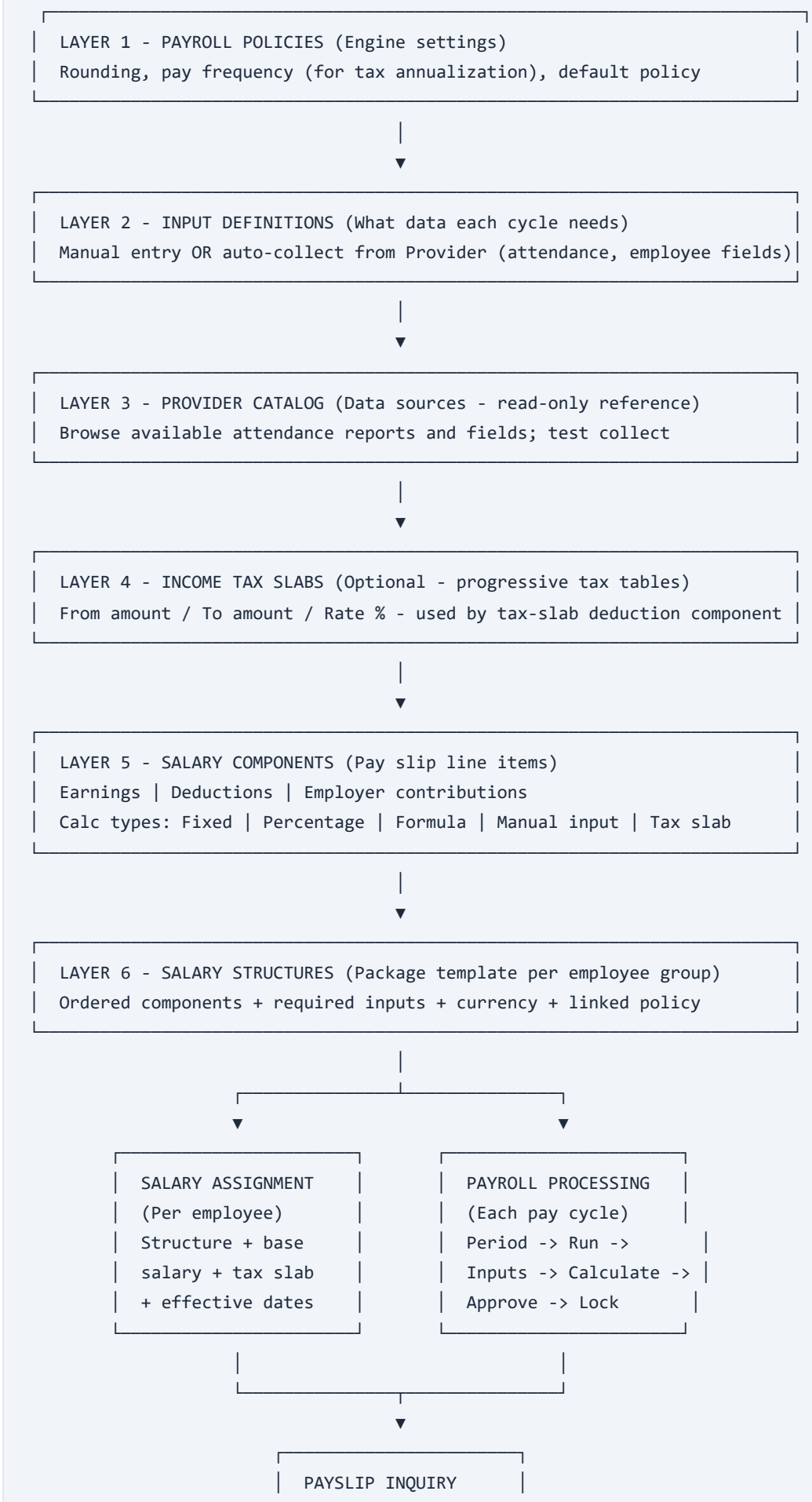
---

3. Core Capabilities at a Glance

---

4. Architecture Framework - Six Configuration Layers

EPORDO Payroll is organized in **six layers**. Configure them **top to bottom** once; then repeat only **Assignment** and **Processing** each pay cycle.


LAYER 1 - PAYROLL POLICIES (Engine settings)
Rounding, pay frequency (for tax annualization), default policy
LAYER 2 - INPUT DEFINITIONS (What data each cycle needs)
Manual entry OR auto-collect from Provider (attendance, employee fields)
LAYER 3 - PROVIDER CATALOG (Data sources - read-only reference)
Browse available attendance reports and fields; test collect
LAYER 4 - INCOME TAX SLABS (Optional - progressive tax tables)
From amount / To amount / Rate % - used by tax-slab deduction component
LAYER 5 - SALARY COMPONENTS (Pay slip line items)
Earnings | Deductions | Employer contributions
Calc types: Fixed | Percentage | Formula | Manual input | Tax slab
LAYER 6 - SALARY STRUCTURES (Package template per employee group)
Ordered components + required inputs + currency + linked policy
SALARY ASSIGNMENT
(Per employee)
Structure + base
salary + tax slab
+ effective dates
PAYROLL PROCESSING
(Each pay cycle)
Period -> Run ->
Inputs -> Calculate ->
Approve -> Lock
PAYSLIP INQUIRY

```
                        |  View | Export | Print|
                        └──────────────────────┘
```

---

5. Five Core Concepts (Glossary)

Map these to familiar international payroll terms:

Concept relationships

```
 Period          = WHEN you pay (date range)
Structure       = HOW you calculate (components + formulas)
Assignment      = WHO gets WHICH structure and base salary
Run             = ONE execution of payroll for that period
Policy          = ENGINE behavior (rounding, frequency)
```

Important rules:

11. One run can include **many structures** (each employee uses their own assignment).

12. **Mid-period structure changes do not auto-split** pay-align changes to period boundaries.

13. A **Period** does not create attendance data-it only defines the date window.

---

6. Module Structure - Four Menus

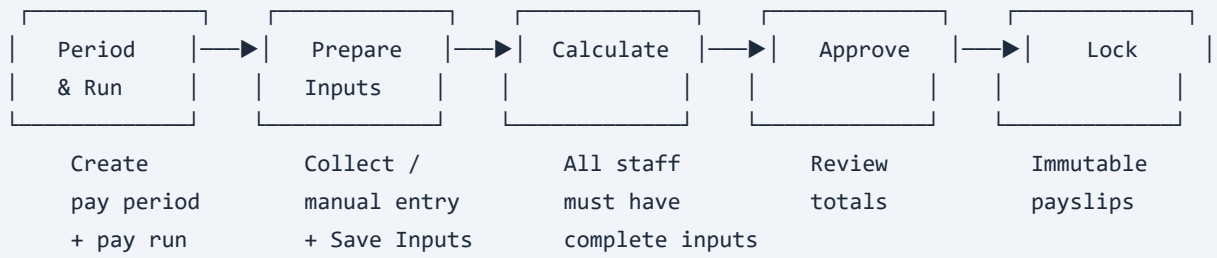
```
EPORDO Cloud Attendance
└── Payroll
    ├── Payroll Settings      <- Configure once (6 tabs)
    ├── Salary Assignment     <- Assign employees
    ├── Payroll Processing    <- Run each cycle (5 steps)
    └── Payslip Inquiry       <- Query and export
```

Payroll Settings - Six tabs (in order)

---

7. Payroll Processing Workflow - Five Steps

Every pay cycle follows the same path:



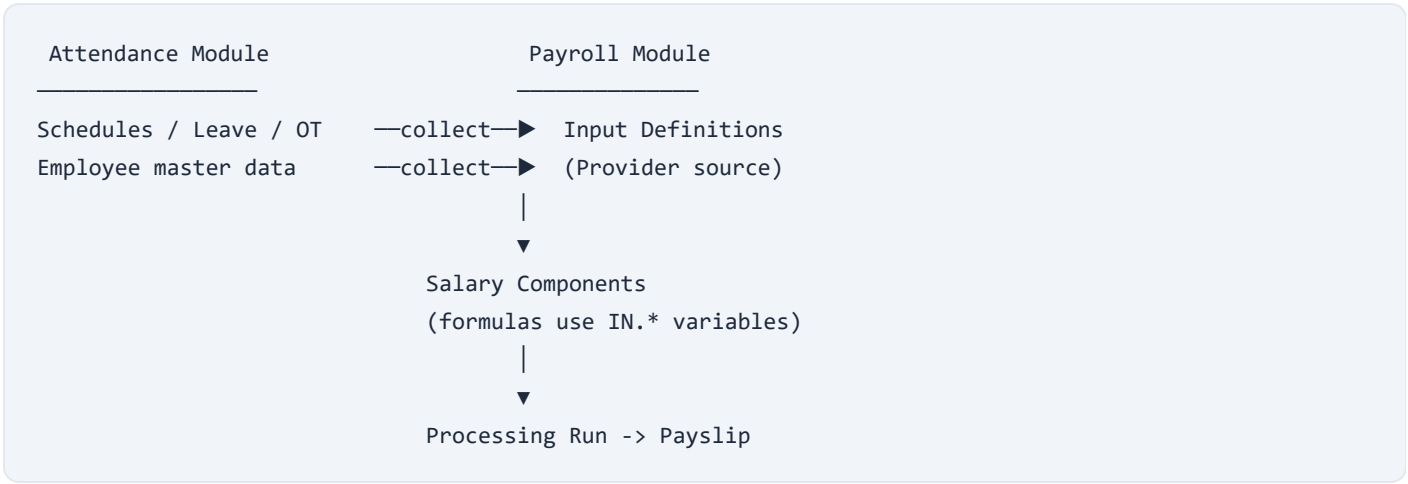
Run status flow: `Draft` -> `Calculated` -> `Approved` -> `Locked` (or `Void` for corrections)

---

8. How Payroll Connects to Attendance

EPORDO keeps **attendance maintenance** and **payroll calculation** separate but linked:

Design benefit: You control exactly which attendance data affects each pay cycle. Payroll does not silently pull data you did not define.



---

9. Component Model - Three Types, Five Calculation Methods

Component types

Calculation methods

Structure ordering rule

Components calculate **in sequence**. Dependencies must come first. **Tax slab deductions must appear after all taxable earnings.**

---

10. International Deployment Highlights

---

11. Feature Comparison - Why EPORDO Payroll Fits Global Operations

---

12. Recommended Implementation Path

```
 Phase 1 - Design (1-2 days)
  - Define pay components and structure for one employee group
  - Decide manual vs auto-collected inputs
  - Confirm tax approach (slab or none)

Phase 2 - Configure (2-3 days)
  - Complete Layers 1-6 in Payroll Settings
  - Assign pilot employees

Phase 3 - Pilot run (1 day)
  - One period, one run, spot-check calculations
  - Approve, lock, export payslips

Phase 4 - Go live
  - Roll out assignments; train payroll staff on monthly cycle
  - Use go-live checklist in Complete User Guide
```

Start here: 02 - Quick Start Guide

Full reference: 03 - Complete User Guide

EPORDO Payroll - Quick Start Guide

> **Goal**: Configure a minimal monthly payroll and produce your **first locked payslip** in one session.
>
> **Time**: 30-45 minutes (assuming employees already exist in EPORDO)
>
> **Prerequisite reading**: 01 - Overview & Framework (optional, 15 min)

Before You Begin - Prerequisites Checklist

What We Will Build (Minimal Example)

A simple **monthly fixed salary** structure:

After setup you will: assign 1+ employees -> open a period -> run payroll -> lock payslips.

Step-by-Step Setup (9 Steps)

Step 1 - Create a Payroll Policy (5 min)

Menu: Payroll -> **Payroll Settings** -> tab **Payroll Policies** -> **Add**

Click **Save**.

Step 2 - Review Input Definitions (2 min)

Menu: Payroll Settings -> tab **Input Definitions**

The platform maintains four **system payment-day inputs** (do not delete):

14 `working_days`

15 `payment_days`

16 `employed_payment_days`

17 `period_payment_days`

For this minimal example you do **not** need extra inputs. Add them later for overtime, allowances, etc.

Step 3 - Skip Tax Slabs (for now)

Menu: Payroll Settings -> tab **Income Tax Slabs**

Skip this tab in Quick Start. Return when you need automatic income tax.

Step 4 - Create a Salary Component (3 min)

Menu: Payroll Settings -> tab **Salary Components** -> **Add**

Optional: enable **Depends on payment days** if you prorate by attendance days.

Click **Save**.

Step 5 - Create a Salary Structure (5 min)

Menu: Payroll Settings -> tab **Salary Structures** -> **Add**

Add component row:

Required inputs: For fixed salary only, you may leave minimal requirements. If BASIC uses payment-day proration, include the system payment-day inputs.

Click **Save**. Fix any sequence validation errors if shown.

Step 6 - Assign Employees (5 min)

Menu: Payroll -> **Salary Assignment** -> **Add Assignment**

For each pilot employee:

Tip: Use **Batch Assignment** to assign many employees at once (same structure and settings).

Rule: No overlapping assignments for the same employee.

Step 7 - Create Period and Run (3 min)

Menu: Payroll -> **Payroll Processing**

Step bar: Period & Run

18 Click
Create Period
- enter start and end dates (e.g. `2026-07-01` to `2026-07-31`).

19 Select the period in the list.

20 Click
Create Run
.

If you see *"No employees with valid assignment"*, return to Step 6 and check effective dates and base salary.

Step 8 - Prepare Inputs and Calculate (5 min)

Step bar: Prepare Inputs

21 Open
Calculation preflight
- confirm employee count and no missing required inputs.

22 If you use Provider sources, click
Collect provider inputs
; otherwise skip.

23 For manual inputs, enter values and click
Save Inputs
(required-even if empty for this minimal case, confirm preflight passes).

Step bar: Calculate

24 Click **Calculate** .

Hard rule: Every employee must have all required inputs saved. One missing value **blocks the entire batch**.

On success, status changes to **Calculated**. Review gross and net in the preview.

Step 9 - Approve and Lock (2 min)

Step bar: Approve -> click **Approve**

Step bar: Lock -> click **Lock**

Payslips are now **immutable**. PDF snapshot uses your **current UI language**.

View results: Payroll -> **Payslip Inquiry** -> filter by period/run -> **View** / **Export PDF** / **Export Excel**.

Congratulations - Your First Pay Run Is Complete

You have completed the full lifecycle:

```
Settings -> Assignment -> Period -> Run -> Inputs -> Calculate -> Approve -> Lock -> Payslip
```

Monthly Operations Cheat Sheet

Repeat **Steps 7-9** every pay cycle. Review assignments only when salaries or structures change.

What to Add Next

Details: 03 - Complete User Guide

Common Quick Start Issues

Quick Reference - Menu Paths

```
Payroll
├── Payroll Settings
│   ├── Payroll Policies          <- Step 1
│   ├── Input Definitions         <- Step 2
│   ├── Provider Catalog          <- (browse only)
│   ├── Income Tax Slabs          <- (skip in quick start)
│   ├── Salary Components         <- Step 4
│   └── Salary Structures         <- Step 5
├── Salary Assignment             <- Step 6
├── Payroll Processing            <- Steps 7-9
└── Payslip Inquiry               <- View results
```

Documentation index: 00 - Documentation Index

EPORDO Payroll - Complete User Guide

Audience: HR staff, payroll clerks, implementation consultants

Style: Written against the **current product UI**-what to click, what to fill, what happens next.

UI language: Set the interface to **English** (or your deployment language) before following this guide.

Version: September 2026 | Aligned with current product release

Related docs: Documentation Index | Overview & Framework | Quick Start

---

Contents

25 What you can do with it

26 Concepts you must know first

27 Menus and recommended setup order

28 Payroll Settings - the six tabs

29 Salary Assignment

30 Payroll Processing - period to lock

31 Payslip Inquiry

32 How amounts are calculated

33 Typical setup examples

34 FAQ and common mistakes

35 Go-live checklist

---

1. What you can do with it

First principle: There is no hidden "default country tax". What you configure is what the engine uses; what you omit does not exist for the engine.

Start from a blank tenant and configure in order. The product **does not** offer a one-click "Import demo template" button-you define rules that match your jurisdiction.

---

2. Concepts you must know first

```
 Period      = Date range for this pay cycle (e.g. 2026-07-01 ~ 07-31)
Structure    = How to calculate: components, formulas, required inputs
Assignment   = Who uses which structure, base salary, and tax slab
Run          = One calculation job under a period (with a config snapshot)
Policy       = Engine-level parameters (rounding, pay frequency for tax annualization)
```

One run may include many structures (each employee uses their own assignment).

Mid-period structure changes do not auto-split one employee into two segments-align changes to period boundaries.

---

3. Menus and recommended setup order

Under **Payroll**:

Recommended setup order

The Settings page onboarding alert matches this sequence:

Policy -> Input defs -> Provider catalog -> Income Tax Slabs -> Components -> Structure, then Assignment and Processing.

```
 (1) Payroll Policies
(2) Input Definitions
(3) Provider Catalog (browse sources; usually read-only)
(4) Income Tax Slabs (skip if no automatic income tax)
(5) Salary Components
(6) Salary Structures
(7) Salary Assignment
(8) Payroll Processing (every cycle)
(9) Payslip Inquiry / export
```

> **New to EPORDO Payroll?** Follow the hands-on path in Quick Start Guide first.

---

4. Payroll Settings - the six tabs

Open **Payroll Settings**. Tab order matches the product.

4.1 Payroll Policies

Engine-level parameters-not "how basic salary is defined".

What affects calculation today: rounding fields, default flag, and pay frequency (mainly for tax annualization). If the UI notes that LWP/OT JSON or working-days mode are not wired into the engine yet, implement attendance logic via **Input Definitions + component formulas + Provider collect**-do not rely on unused policy JSON.

Practical note: For monthly pay with join/leave proration by calendar days, policy join/leave mode often uses "by payment days". For weekly/daily pay where `payment_days` already reflects actual days worked, avoid double proration.

When creating a policy, enter code and country yourself (no prefilled demo country template).

4.2 Input Definitions

Answer: **What quantities does this cycle need, and where do they come from?**

System payment-day inputs (platform-maintained; UI locks delete/deactivate):

`working_days`, `payment_days`, `employed_payment_days`, `period_payment_days`

Keep them if components use "depends on payment days" or join/leave proration. Types must remain numeric.

Principle: If it is not defined, the engine will not use it-e.g. if you do not define holiday days, no holiday deduction will occur.

4.3 Provider Catalog

Read-only catalog of built-in sources (attendance reports, employee fields, ...). Use **Test collect** with Employee ID + date range to verify outputs.

You do not create providers here; binding happens on **Input Definitions** when Source = Provider.

Collection behavior (overview):

36 Full calendar months prefer monthly reports

37 Partial months aggregate daily data

38 Cross-month ranges split then merge

39 Missing reports do not invent data

4.4 Income Tax Slabs

Maintain progressive rows: **From amount / To amount (empty = no upper limit) / Rate %**.

A slab alone does nothing. Tax appears only when **all** of the following are true:

40 A
Deduction
component uses calc type
Tax slab

41 That deduction is sequenced
after
all taxable earnings in the structure

42 The employee's assignment selects an
Income tax slab

43 At least one earning is marked **Taxable earning** and has a calculated amount

If no slab is selected on assignment, tax is typically **0** (the engine does not guess).

4.5 Salary Components

A component is one calculated line on the payslip.

Type

Calc type

For **Fixed**, amount source is often assignment base salary / fixed amount / formula.

Optional flags: depends on payment days, join/leave proration, condition expression, taxable earning (earnings only).

Grid status: **Active / Inactive**.

Use codes with letters, digits, and underscore only-avoid spaces and operators so formulas remain reliable.

4.6 Salary Structures

A structure = **ordered component rows** + **required inputs** + optional structure overrides.

Steps:

44 Select a **Policy**, set **Currency** (you must enter this-no country default currency).

45 **Add component row**

; sequence controls calculation order (save often renumbers 10, 20, 30...).

46 Mark
required inputs
that must exist before calculate.

47 Optionally override amount/formula/condition per structure;
Tax slab deductions cannot be overridden
at structure level.

Ordering rules:

48 Referenced components must calculate first

49 **Tax slab deduction must follow all taxable earnings**

50 Fix sequence errors shown on save

Usually keep **one** "assignment base salary" Fixed component per structure; other fixed lines use fixed amount or formula.

---

5. Salary Assignment

Menu: **Salary Assignment**.

5.1 Single assignment

Use **Add Assignment** (or equivalent):

5.2 Batch assignment

Use **Batch Assignment**:

51 **By employees**
: multi-select people; shared structure/salary/slab

52 **By departments**
: same assignment for employees in selected departments (
not
sub-departments, per UI hint)

53 Option to
skip employees with overlapping assignment

54 Review success / skipped / total after
Apply Batch

Department scope follows the user's department permissions.

---

6. Payroll Processing - period to lock

Menu: **Payroll Processing**. Step bar:

Period & Run -> Prepare Inputs -> Calculate -> Approve -> Lock

6.1 Step 1: Period & Run

Payroll Period

Period scope filter: **Active** / **Include closed**. You cannot **Create Run** on a closed period until it is reopened.

Processing Run

Run status: **Draft -> Calculated -> Approved -> Locked**, plus **Void**.

If you see *"No employees with valid assignment"*, finish **Salary Assignment** first (base salary > 0, effective dates covering the period).

6.2 Step 2: Prepare Inputs

Guided order: **Preflight -> Auto collect -> Manual entry -> Saved overview**.

55 **Calculation preflight**

Review employee count, saved inputs, and gaps for Provider vs manual sources (sample missing rows).

56 **Auto collect (Provider)**

- Click **Collect provider inputs**

- Options: allow partial missing / fail on first error; **Skip existing values**

- Review Written / Missing result table

57 **Manual entry**

- Shows only Manual inputs referenced by the structure

- Use **Batch Fill** -> **Apply to Filtered** for bulk entry

- **Must click Save Inputs**-values are not saved until you do

58 **Saved inputs overview**

Filter by employee code, input item, source (Manual / Provider).

6.3 Step 3: Calculate

Click **Calculate** or **Recalculate**.

Hard rule: **Every** employee in the run must have **all** required inputs saved. Any missing item **blocks the whole batch**-partial success is not supported. The UI lists missing rows so you can return to Step 2.

On success, status becomes **Calculated**; review gross / deduction / net in the preview (full list also in Payslip Inquiry).

6.4 Step 4: Approve

Click **Approve**. Inputs are generally frozen after approval; you can still **Lock** or **Revert to Draft** (follow confirmation dialogs).

6.5 Step 5: Lock

Click **Lock**. Payslip snapshots are saved in the **current UI language** and become immutable.

To correct a locked run: **Void Run**, then create a new run and recalculate.

---

7. Payslip Inquiry

Menu: **Payslip Inquiry**.

Filters

59 Payroll Period, Processing Run

60 Status: All / Draft / Calculated / Approved / Locked / Void

61 Department, employee code / name keyword

The list shows **created** time.

Actions

Detail layout: Employee Earnings & Deductions and **Employer Contributions (CTC)** are shown in separate sections. Employer contribution lines are not paid to the employee and are excluded from employee gross/net.

---

8. How amounts are calculated

Calculation pipeline

62 Load the run's **configuration snapshot** (structures, components, policies)

63 Load each employee's **saved inputs** and assignment (base salary, tax slab)

64 Evaluate components in structure sequence (Fixed / Percentage / Formula / Manual input / Tax slab)

65 Components with a false **condition** evaluate to 0

66 Apply policy rounding to components and/or net

67 Sum gross, deductions, net; employer contributions separately

8.1 Payment-day proration (Fixed-style)

When a component depends on payment days, a common pattern is:

`amount × (payment_days / working_days)`

With join/leave proration enabled, multiply by `SYS.proration_factor` (derived from employed payment days vs full period days).

If `working_days = 0`, related proration yields **0** (no divide-by-zero).

8.2 Income tax (Tax slab)

68 Sum all **Taxable earning** amounts for the period -> period taxable base

69 Annualize using pay frequency (e.g. monthly × 12)

70 Apply progressive slab to annual amount -> annual tax

71 Divide by periods per year -> period tax

72 Write result to the tax-slab deduction component

Pay frequency is taken from **Structure** first, then **Policy**-do not rely on Period settings alone for tax.

8.3 Formula variables (for configuration)

Use **Validate Expression** in the formula editor before save.

---

9. Typical setup examples

Example A: Monthly pay + base + housing % + OT formula + income tax

73 **Policy** : Monthly; round component and net to `0.01`; half up; both.

74 **Input defs** : four payment-day inputs (Manual or Provider); overtime hours; nationality if needed for conditions.

75 **Tax slab**

: progressive rows per local rules.

76 **Components**

:

- BASIC - Fixed, assignment base, optional proration

- HRA - Percentage of BASIC

- OT - Formula referencing overtime input

- TAX - Deduction, Tax slab

- Mark taxable earnings per local law (often BASIC; HRA may or may not be taxable)

77 **Structure**

: BASIC -> HRA -> OT -> ... -> TAX; mark required inputs.

78 **Assignment**

: structure, base salary, tax slab per employee.

79 **Processing**

: Create Period -> Create Run -> collect/manual + Save Inputs -> Calculate -> Approve -> Lock.

Example B: Fixed salary only, no income tax

Skip Income Tax Slabs and tax-slab deduction components. Structure contains earnings and ordinary deductions only. Assignment needs no tax slab.

Example C: Weekly pay with attendance-based OT

80 **Policy**

: Weekly pay frequency.

81 **Input defs**

: OT hours (Provider from attendance daily report).

82 **Components**

: BASIC (Fixed), OT (Formula: `IN.ot_hours * hourly_rate`).

83 Ensure policy/structure frequency aligns-mismatch with period labels can cause incorrect tax annualization if tax is added later.

---

10. FAQ and common mistakes

---

11. Go-live checklist

Configuration

84 Policy frequency and rounding match company rules

85 Input definitions cover all required structure inputs; Manual vs Provider sources correct

86 Provider outputs match variable mappings (spot-check with Test collect)

87 Component types, calc types, taxable flags correct

88 Structure order validates on save; tax slab after all taxable earnings

89 Assignments non-overlapping; tax slab assigned where needed; base salary > 0

90 Currency set correctly on each structure

One pilot cycle

91 Active period -> Create Run -> preflight -> collect/manual -> **Save Inputs**

92 Calculate succeeds; spot-check gross/deductions/net vs manual calculation

93 Approve -> Lock; Payslip Inquiry Excel/PDF export OK

94 Switch UI language and confirm labels (status, source, calc type) display correctly

Access control

95 Payroll menus and action buttons match role design

96 Department data scope correct for batch assignment and inquiry

---

Appendix A - UI term reference

Appendix B - Related documentation

For engine fields, DDL, and Provider ADR, refer to your technical documentation index- not required for day-to-day operation.