

Attendance Module — Business & API Guide

অ্যাটেনড্যান্স মডিউল — ব্যবসা ও API গাইড

বিনামূল্যের বাংলা + ইংরেজি রেফারেন্স | ERPFLOW Backend

Attendance Module Guide — bilingual reference documentation

ENGLISH VERSION

Attendance Module — Business & API Guide

This guide documents every confirmed feature of the **Attendance** module in the ERPFLOW backend. Every statement below is traceable to the reviewed source files listed in Appendix A. Where a detail could not be verified, the guide says exactly: **"Not specified in the reviewed source files."**

বাংলা ভার্সন: [ATTENDANCE_MODULE_GUIDE_BN.md](#) · Bilingual PDF: [ATTENDANCE_MODULE_GUIDE.pdf](#)

1. What is this module?

The Attendance module is the company's **digital time-and-leave engine**. Verified capabilities:

- Record check-in/check-out **punches** (multi-punch per day, multiple sources)
- Maintain master data: **Shifts, Attendance Types, Leave/Holiday Policies**
- Attach shifts and policies to scopes (**Assignments**) — company-wide down to a single employee
- Resolve what an employee effectively holds on a date (**Assignment Resolution**)
- Calculate **daily attendance records** with a frozen policy snapshot
- Run a scheduled or manual **Close Day** calculation job
- Manage **Leave Balances**, accrual, carry-forward and manual adjustments with a full ledger
- Handle the full **Leave Request** lifecycle (preview → apply → approve/reject/cancel)
- Handle **Attendance Correction Requests** (missing/incorrect punch, wrong status)
- Close the month via **Monthly Attendance Approval**, lock records and flag them `ready_for_payroll`

Important: Everything is tied to your **current company**. Every query is scoped through the platform tenancy context; the company arrives on the request as the `X-Company-Id` header.

2. Who uses it?

Access is granted by permission keys enforced by `permission:` middleware on each route group. The registry below is from the module spec (Section 4) cross-checked against route usage:

Permission key	Grants
<code>attendance.menu-view</code>	See the Attendance section; read policies, attendance types, shifts, assignments; list punches
<code>attendance.config-manage</code>	CRUD attendance types, shifts, policies
<code>attendance.policy-group-view</code>	View policy groups and eligibility attributes
<code>attendance.policy-group-manage</code>	Create/edit/status/delete policy groups; eligibility preview; apply/backfill to employees
<code>attendance.assignment-manage</code>	Create/update/end assignments; bulk assign; trigger record recalculation endpoint
<code>attendance.assignment-preview</code>	Assignment resolution diagnostic tool
<code>attendance.punch-create</code>	Punch in/out
<code>attendance.punch-create-others</code>	Punch/list punches on behalf of another employee (enforced inside PunchService)
<code>attendance.record-view-own / -team / -all</code>	Daily record visibility tiers (route accepts any of the three; deeper checks inside controller)
<code>attendance.record-export</code>	Export daily summaries
<code>attendance.record-recalculate</code>	"Recalculate with current policy" override (enforced inside the recalculation service) and close-day job endpoints
<code>attendance.correction-create</code>	Submit/cancel own correction requests; view correction queue
<code>attendance.correction-approve</code>	Approve/reject corrections (direct fallback path)
<code>attendance.correction-override-lock</code>	Approve a correction against a locked month (enforced in AttendanceCorrectionExecutor)
<code>attendance.leave-apply</code>	Apply/preview/cancel own leave
<code>attendance.leave-approve</code>	Approve/reject leave; view any employee's leave
<code>attendance.leave-balance-view</code>	List all balances; view another employee's ledger history
<code>attendance.leave-balance-adjust</code>	Manual balance adjustment; trigger accrual / carry-forward jobs
<code>attendance.monthly-view</code>	Monthly approval grid, detail, breakdown, batch status
<code>attendance.monthly-approve</code>	Build/approve/bulk-approve a month
<code>attendance.monthly-unlock</code>	Unlock an approved month

Per the spec, the self-service keys (`attendance.record-view-own`, `attendance.punch-create`, `attendance.correction-create`, `attendance.leave-apply`) belong to a seeded `employee` role that every user receives by default.

3. Platform conventions that apply to every endpoint

Convention	Fact (verified)
Base URL	<code>/api/v1/attendance</code> (RouteServiceProvider)
Route-name prefix	<code>api.attendance.</code>
Global middleware	<code>auth:api</code> , <code>onboarding.access</code> , <code>onboarding.document</code> wrap every route in the module
Per-route authorization	<code>permission:<key></code> middleware groups (keys in §2)

Convention	Fact (verified)
Tenancy	Company id comes from <code>TenantContext</code> ; controllers throw <code>PermissionDeniedException('company.context')</code> when it is missing. Header: <code>X-Company-Id</code> (spec §0.3)
Employee identity	<code>employee_id</code> references <code>employee_personal_infos.id</code> ; several validations use <code>exists:employee_personal_infos,id</code>
Approval engine	Leave, correction and monthly approval decisions go through the platform ApprovalGateway ; when approval is disabled for the action, <code>onApproved</code> runs synchronously (documented bypass, not an error)
Pagination	List endpoints return paginated resources; default page size is 15 unless stated otherwise

4. Feature catalog

4.1 Module health check

- **Business purpose:** Verify the module is operational and that its assignment-resolution cache can actually be invalidated in this deployment.
- **What it does:** Returns the cache store name, whether it supports tags, an overall `status` of `ok / degraded` , and a human-readable `detail` . Logs activity event `attendance.health_viewed` .
- **Endpoint:** `GET /health` → `AttendanceController@health` (no extra permission beyond the global auth group).
- **Response behavior:** JSON `{module: "attendance", status: "ok"|"degraded", cache:{store, supports_tags, supported, detail}}` . A store without tag support yields `degraded` because resolved assignments could serve stale data for up to 24 hours.
- **Data entities:** None (reads `cache.default` config).

4.2 Attendance Types (master data)

- **Business purpose:** Define the vocabulary of day statuses (Present, Absent, Late, Half Day, Leave, Holiday, Weekend, WFH, Business Trip, Missing Check-In, Missing Check-Out) plus company-specific types.
- **What it does:** CRUD + status toggle. System-seeded rows carry an immutable `system_code` ; HR-created rows have `system_code = NULL` . The calculation engine resolves statuses by `system_code` , never by name/id.
- **Endpoints** (read group under `attendance.menu-view` ; write group under `attendance.config-manage`):

Method	Path	Controller@action
GET	<code>/attendance-types</code>	<code>AttendanceTypeController@index</code>
GET	<code>/attendance-types/{id}</code>	<code>AttendanceTypeController@show</code>
POST	<code>/attendance-types</code>	<code>AttendanceTypeController@store</code> (201)
PUT	<code>/attendance-types/{id}</code>	<code>AttendanceTypeController@update</code>
PATCH	<code>/attendance-types/{id}/status</code>	<code>AttendanceTypeController@updateStatus</code>
DELETE	<code>/attendance-types/{id}</code>	<code>AttendanceTypeController@destroy</code>

- **Validation (verified):** `name` required ≤ 100 unique per company · `code` required ≤ 50 , regex `^[A-Z0-9_]+$`, unique per company · `system_code` **prohibited** on input · `category` required ≤ 50 · `is_paid` , `counts_as_working_day` , `eligible_for_payroll` required booleans · `color` ≤ 30 , `icon` ≤ 50 required · status change accepts only `active|inactive` . Update re-validates uniqueness ignoring the current row.
- **Delete guard (verified):** A system type cannot be deleted (409, "Cannot delete a system-seeded attendance type..."); a type referenced by existing records cannot be deleted (409).

- **Data entity:** table `attendance_types` (fields per model fillable + spec §3.1).

4.3 Shifts

- **Business purpose:** Define working-time templates: start/end time, timezone, break, grace period, minimum hours for present/half-day, working days, overnight flag.
- **Endpoints:**

Method	Path	Permission group	Controller@action
GET	/shifts	menu-view	Shift\ShiftManagementController@index
GET	/shifts/{shiftId}	menu-view	...@show
POST	/shifts	config-manage	...@store (returns 200 "Shift Create Successfully Done")
PUT	/shifts/{shiftId}	config-manage	...@update
PATCH	/shifts/{shiftId}/status	config-manage	...@updateStatus
DELETE	/shifts/{shiftId}	config-manage	...@destroy (message: "Shift archived successfully.")

- **Validation (verified):** `name` ≤ 150 (required on create, nullable on update) · `start_time` / `end_time` required `H:i` · `timezone` nullable valid timezone · `break_minutes` `int` ≥ 0 · `working_hours` numeric $> 0 \leq 24$ · `grace_minutes` `int` ≥ 0 · `min_hours_present` numeric $> 0 \leq 24$ · `min_hours_half_day` numeric $\geq 0 \leq 24$ · `working_days` required array ≥ 1 of distinct integers 1–7 (ISO-8601: 1 = Monday) · `is_overnight` required boolean · `status` nullable in `active|inactive`.
- **Note (code fact):** the list endpoint defaults `per_page` to **1** when not supplied.
- **Data entity:** table `shifts`.

4.4 Attendance Policies (Leave & Holiday)

- **Business purpose:** Author the rules behind leave entitlements and holiday calendars, effective-dated, with a typed JSON `config`.
- **What it does:** Two policy types — `leave` (entitlement, accrual method, carry-forward, encashment, half-day, LWP, advance notice, backdate limit, document threshold) and `holiday` (list of holiday entries with optional `recurring`). Config shape is validated by dedicated rule objects (`HolidayPolicyConfigRule` / `LeavePolicyConfigRule`) selected by policy type; update validates against the stored policy's type.
- **Endpoints:** read under `menu-view` (GET `/policies`, GET `/policies/{id}`); write under `config-manage`: POST `/policies` (201), PUT `/policies/{id}`, PATCH `/policies/{id}/status`, DELETE `/policies/{id}`.
- **Validation (verified):** create requires `policy_type` $\in \{\text{leave, holiday}\}$ (model constants), `name` ≤ 150 , `code` ≤ 50 , `effective_date` date, `config` array, optional `status` $\in \{\text{Active, Inactive, Archived}\}$. Status change uses the same status set.
- **Data entity:** table `attendance_policies` (`policy_type`, `config` json-cast).

4.5 Policy Groups (eligibility bundles + apply/backfill)

- **Business purpose:** Bundle policies with an **eligibility rule document** so HR can say "who receives what", then push the result onto employees as ordinary assignments.
- **What it does (verified model docblock & services):** Groups are an authoring layer only — nothing at runtime resolves through a group; applying a group creates employee-scoped `assignments` rows. Editing a group never reaches anyone by itself; application is a separate, deliberate act with preview first.
- **Endpoints:**

Method	Path	Permission	Purpose
GET	/policy-groups	policy-group-view	Paginated list

Method	Path	Permission	Purpose
GET	/policy-groups/attributes	policy-group-view	Rule-builder vocabulary: attributes + allowed operators
GET	/policy-groups/{id}	policy-group-view	Detail (includes policies when loaded)
POST	/policy-groups	policy-group-manage	Create (201)
POST	/policy-groups/eligibility/preview	policy-group-manage	Preview how many active employees unsaved rules reach (matched_count , total_active , sample)
PUT	/policy-groups/{id}	policy-group-manage	Update
PATCH	/policy-groups/{id}/status	policy-group-manage	Active/Inactive
DELETE	/policy-groups/{id}	policy-group-manage	Delete
POST	/policy-groups/{id}/apply/preview	policy-group-manage	Backfill preview (eligible / already_assigned / new / conflicts / skipped + sample)
POST	/policy-groups/{id}/apply	policy-group-manage	Queue chunked backfill (202)
GET	/policy-groups/{id}/apply/{batchId}	policy-group-manage	Batch status

- **Eligibility attributes (verified registry):** employment_type_id , grade_id , branch_id , division_id , department_id , section_id , team_id (reference ids from company-scoped tables) and gender , probation_status (normalized strings). Operators: in , not_in , equals , not_equals . Empty rules mean the whole company ("Company General"). match may be all (default) or any .
- **Validation (verified):** name ≤150, code ≤50 (both unique per company — service-level), description ≤1000 nullable, eligibility required array validated by PolicyGroupEligibilityRuleValidator (also checks referenced ids exist in the company), effective_date_strategy ∈ { joining_date , probation_end_date } (default joining_date ; probation strategy falls back to joining date when no probation end recorded), status ∈ { Active , Inactive }, policy_ids array of ints that must be **active** policies of this company. Backfill requests accept optional date and sample_size (0-100 apply-preview; 0-50 eligibility-preview).
- **Data entities:** attendance_policy_groups , pivot attendance_policy_group_policy , produced rows in assignments (with policy_group_id stamped).
- **Auto-assignment:** a PolicyGroupAutoAssignmentService keeps one employee's policy assignments aligned with matching groups (additive only); an EligibilitySweepService announces employees whose organization assignment recently came into force (scheduled command exists).

4.6 Assignments (linking shifts/policies to scopes)

- **Business purpose:** The single source that says which shift, which leave policies and which holiday calendar apply at each organizational level and effective date.
- **Scope model (verified Assignment model):** scope_type ∈ hierarchy company → branch → division → department → section → team → employee ; resolution walks **most-specific first**. assignable_type ∈ { shift , policy }; for policies assignable_subtype ∈ { leave , holiday } and must match the policy's type. Cardinality: shift and holiday are exclusive per scope; leave is multi. Sources: manual , auto , bulk . Statuses: Active , Inactive .

Method	Path	Permission	Notes
GET	/assignments	menu-view	Filters: scope_type, scope_id, assignable_type, assignable_subtype, source, history(bool), per_page ≤100
GET	/assignments/{id}	menu-view	Detail

Method	Path	Permission	Notes
POST	/assignments	assignment-manage	Idempotent create (assignOrGet) — same assignable+scope+effective date returns existing row without a second write/event. Only Active shifts/policies can be assigned. Overlap conflict → AssignmentOverlapException error response
PATCH	/assignments/{id}	assignment-manage	Partial: end_date, status
POST	/assignments/{id}/end	assignment-manage	Requires end_date
POST	/assignments/bulk	assignment-manage	Target by filter{scope_type,scope_id} (company...team) or explicit employee_ids ; sync response or 202 queued {batch_id,status,total,processed,succeeded,failed}
GET	/assignments/bulk/{batchId}	assignment-manage	Batch progress

- **Create validation (verified StoreAssignmentRequest):** as per scope model above; assignable_subtype must be null when type is shift; scope_id prohibited for company, required otherwise; effective_date required; end_date ≥ effective_date. Bulk variant additionally requires either filter or employee_ids .
- **Queued mode threshold:** switching between synchronous and queued is threshold-based; the numeric value is **not specified in the reviewed source files**.
- **Data entity:** table assignments ; scope existence is checked against the matching company-scoped Configuration/Employee tables (verified assertScopeExists).
- **Event:** class AssignmentChanged exists and is imported by AssignmentService .

4.7 Assignment Resolution (diagnostic + engine core)

- **Business purpose:** Answer "what does this employee hold on this date?" — the same answer the punch, calculation and leave engines rely on.
- **Endpoint:** GET /resolve/assignment?employee_id=&date= (attendance.assignment-preview). Validation: employee_id required integer; date required Y-m-d .
- **Response (verified DTO):** {shift, leave_policies, holiday_calendar, timezone, resolved_from} where resolved_from names the winning scope per slot. Timezone falls back shift → company. No active shift ⇒ validation error bag unassigned ("No active shift found for the given date.").
- **Caching:** results cached 24h keyed per company/employee/date with tag-based invalidation when supported.

4.8 Punch In/Out (append-only multi-punch)

- **Business purpose:** Capture raw attendance events from web, mobile, biometric, API or manual entry — never mutated afterwards, only superseded.
- **Endpoints:** POST /punch (attendance.punch-create) · GET /punches?employee_id=&date= (attendance.menu-view).
- **Validation (verified StorePunchRequest):** punch_type ∈ { in , out } · source ∈ { web , mobile , biometric , api , manual } · punch_time nullable Y-m-d H:i:s · employee_id nullable integer · ip_address ≤45 · device_info ≤255 · remarks ≤255.
- **Behavior (verified PunchService):**
- Punches default to the caller's linked employee profile; supplying another employee_id requires attendance.punch-create-others (403 otherwise). Listing follows the same rule.
- "Today" is computed in the company timezone (fallback Asia/Dhaka when unset).
- Assignment resolution runs first; no active shift ⇒ **422** with code: unassigned , message "No shift is assigned for this date."
- Overnight shifts: a local time before start_time is attributed to the previous attendance_date .

- Two consecutive punches of the same type are rejected (422) with the last punch time in the message.
- Rows are append-only (`UPDATED_AT = null`), sequenced per day (`sequence_no`), and can later be superseded via `superseded_by_id` (set by approved corrections — originals are never edited/deleted).
- Event `PunchCreated` fires on creation.
- **Response:** 201 with `PunchResource` (id, company_id, employee_id, attendance_date, punch_type, punch_time ISO-8601, source, ip_address, device_info, remarks, sequence_no, superseded_by_id, correction_request_id, created_at).
- **Data entity:** table `attendance_punches` .

4.9 Daily Attendance Records (view & export)

- **Business purpose:** One calculated row per employee per day — first check-in, last check-out, worked/overtime hours, late/early minutes, punch count, applied-policy snapshot, lock flag.
- **Endpoints:**

Method	Path	Permission group	Notes		
GET	<code>/attendance-records/today</code>	<code>record-view-own \</code>	-team \	- all	Caller needs a linked employee profile (404 otherwise); recalculates today then returns the record
GET	<code>/attendance-records</code>	same	Filters <code>employee_id, start_date, end_date, status</code> ; range span >366 days ⇒ 422; visibility scoped by the logged-in user inside the repository		
GET	<code>/attendance-records/{id}</code>	same	404 if absent; without <code>record-view-all</code> the record must appear in the caller's filtered scope else <code>PermissionDeniedException('attendance.record-view')</code>		
GET	<code>/attendance-records/{id}/punches</code>	same	All punches of that day (including superseded)		
GET	<code>/attendance-records/export</code>	<code>record-export</code>	<code>format</code> query defaults <code>xlsx</code> ; ≤2000 rows (config <code>attendance.exports.export_queue_threshold</code> , default 2000) streams <code>attendance_records_<timestamp>.<format></code> via Excel; larger dispatches <code>QueueAttendanceExportJob</code> and returns 202 {token} (40-char random) for later download		

- **Calculation core (verified `AttendanceRecordService::calculateDaily`):** locked records are returned untouched; otherwise non-superseded punches are paired in time order within the resolved shift/timezone; weekend / holiday (incl. recurring month-day matches) / working-day classification; grace vs late; half-day thresholds from the shift; system attendance types are auto-created on demand keyed by `system_code` ; the applied settings are frozen into `policy_snapshot` (grace minutes, working hours, min-hours present/half-day, working days, timezone, resolved_at). Locked days are never recalculated.
- **Data entities:** `attendance_records` (+ relations employee/shift/attendanceType/punches).

4.10 Record Recalculation (snapshot replay vs live override)

- **Business purpose:** Rebuild a day's numbers after data fixes — normally replaying the **stored** snapshot so history stays faithful; optionally re-resolving **live** policy as an audited override.
- **Endpoint:** POST /records/{recordId}/recalculate . **Route middleware is** attendance.assignment-manage . Body (RecalculateAttendanceRecordRequest): use_current_policy **required boolean**; other documented fields (user_id,date,policy_snapshot,mock_resolved_policy,shift,is_locked,has_permission) nullable.
- **Behavior (verified):** 404 when record missing or belongs to another company. Locked record ⇒ AttendanceRecordLockedException . With use_current_policy=true , the service additionally demands engine permission attendance.record-recalculate else RecalculationUnauthorizedException ; it flushes the resolution cache, resolves live policy and writes a before/after audit log. Without it, the stored policy_snapshot is replayed (older records without a snapshot fall back to live resolve with a warning log).

4.11 Close Day job (manual trigger + status)

- **Business purpose:** Ensure every active employee has a calculated record for a given past day — the same work the nightly scheduler performs (hourly command batches companies whose local time hits hour 02:00, per build sequence card 9.1-BE).
- **Endpoints:** POST /jobs/close-day and GET /jobs/close-day/{batchId} — route permission attendance.record-recalculate , prefix jobs .
- **Validation (verified CloseDayRequest):** date required Y-m-d , before_or_equal:today , after_or_equal:today-90 days ; employee_id nullable exists:employee_personal_infos,id .
- **Behavior (verified):** Skips entirely when the Payroll module's PayrollPeriodGuard reports the period closed → 200 "No employees to process or period is locked." Otherwise chunks active employees **without** a record for that date (chunk size 200) into ProcessCloseDayChunkJob s and returns 202 {batchId} . Status endpoint returns Laravel batch fields (total/pending/failed/processed/progress, status finished|cancelled|pending) or 404.
- **Cross-module dependency:** Modules\Payroll\Support\PayrollPeriodGuard (explicit import).

4.12 Leave Balances (view, adjust, ledger)

- **Business purpose:** Per employee, policy and year: entitled, used, carried-forward and encashed days, with an append-only ledger proving every movement.
- **Available-days formula (verified model accessor):** entitled_days + carried_forward_days - used_days - encashed_days .
- **Endpoints:**

Method	Path	Authorization	Notes
GET	/leave-balances/me	none beyond auth group	Current-year balances for own profile; 403 when login lacks an employee link
GET	/leave-balances/{employeeId}/history	internal check	Own id always; anyone else needs attendance.leave-balance-view else denied. year input defaults to current year; returns paginated ledger rows
GET	/leave-balances	leave-balance-view	Filters employee_id (∃ employee_personal_infos), year 2000–2100, policy_id (∃ attendance_policies)
PATCH	/leave-balances/{id}/adjust	leave-balance-adjust	See below

- **Adjustment (verified):** body days required numeric ≠0, reason required ≤255, allow_negative nullable boolean. Row is locked FOR UPDATE ; adjustment applies to entitled_days ; a negative resulting balance is rejected (DomainException) unless allow_negative ; a manual_adjustment **ledger row** (signed days, reason, actor) is written in the same transaction.

- **Ledger entry types (verified enum):** `accrual`, `carry_forward`, `consumption`, `reversal`, `encashment`, `manual_adjustment`.
- **Data entities:** `leave_balances`, `leave_balance_ledger`.

4.13 Leave Accrual & Carry-Forward jobs

- **Business purpose:** Grow balances monthly per policy rules and roll unused days across year-end — runnable on schedule (dedicated console commands exist) or manually for backfill.
- **Endpoints (permission `attendance.leave-balance-adjust`, prefix `jobs`):**
 - `POST /jobs/accrue-leave` — body: `month` 1–12, `year` `current±5`, optional `policy_id` (`∃ attendance_policies`). Queues chunked accrual; **202** `{batchId}` or **200** `{applied:0}` "No active leave assignments found to accrue or accrual already completed."
 - `POST /jobs/carry-forward` — body: `from_year` `current±5`, optional `policy_id`. **202** `{batchId}` or **200** "No active leave assignments found for carry-forward."
 - `GET /jobs/{batchId}` — Laravel batch progress or 404.
- **Carry-forward rules (verified service):** governed by policy config `carry_forward_allowed` and capped by `max_carry_forward` (`min(available, max)`); idempotent — skips when a `carry_forward` ledger row already exists for the target balance; scopes are resolved **as of the historical date**, so a past-year run uses the organization the employee sat in then.

4.14 Leave Requests (self-service lifecycle)

- **Business purpose:** Employees apply for leave against assigned policies; approvers decide; approval stamps leave days and deducts balance atomically.
- **Endpoints (group permission `attendance.leave-apply` | `attendance.leave-approve`, prefix `/leave-requests`):**

Method	Path	Action
GET	<code>/applicable-policies?date=</code>	Policies the caller may apply against on a date (assignment-driven, not balance-driven so first-time applicants see options). Each row: <code>id</code> , <code>name</code> , <code>code</code> , <code>half_day_allowed</code> , <code>lwp_allowed</code> , <code>available_days</code> , <code>entitled_days</code> , <code>year</code> , <code>balance_provisioned</code> . Requires own employee link (403 with explanatory message otherwise); <code>date</code> validated <code>Y-m-d</code> , defaults today
POST	<code>/preview</code>	Dry-run cost of a candidate range — returns <code>total_days</code> , <code>excluded_dates[]</code> , <code>balance{available_days, after_request_days}</code> , <code>requires_document</code> , <code>document_threshold_days</code> , <code>blockers[]</code> (reported, not thrown), <code>warnings[]</code>
POST	<code>/</code>	Submit (201). Adds <code>reason</code> required, attachment file <code>pdf/jpg/jpeg/png ≤5120KB</code> (stored <code>leave_attachments</code> public disk), optional <code>attachment_path</code> . Blockers become errors here; overlapping approved leave answers 409. Submits via <code>ApprovalGateway</code> (<code>leave-approve</code> , correlation <code>leave_request:{id}</code>)
GET	<code>/</code>	Approver without <code>employee_id</code> sees all (<code>indexOfHr</code>); everyone else is forced to their own — passing a colleague's <code>id</code> without <code>attendance.leave-approve</code> raises that denial. Filters: <code>status</code> , <code>employee_id</code> , <code>policy_id</code> , <code>from</code> , <code>to</code>
GET	<code>/ {id}</code>	Approver sees any; others only their own
PATCH	<code>/ {id}/cancel</code>	Own only; withdraws the linked approval request (stated in response message)
POST	<code>/ {id}/approve</code>	Engine check <code>attendance.leave-approve</code> inside the action; optional <code>comment</code> / <code>reason</code>
POST	<code>/ {id}/reject</code>	Same permission; non-empty <code>reason</code> required else 422

- **Approval effects (verified `LeaveRequestExecutor`):** locks the request row; pending-only (409 "Request already finalized"); rejects overlap with another **approved** request (409); computes per-working-day values excluding weekends/holidays using the resolved shift+holiday calendar; stamps those days as `attendance_records` of the leave type with `policy_snapshot {leave_request_id}` (spec §5.3: executor also deducts balance + writes consumption ledger row); verifies $\Sigma \text{day-values} = \text{total_days}$ (else 422); sets `approved/decided_by/at`; fires `LeaveDecided`.

- **Validation inputs (verified):** preview/apply share `leave_policy_id` int, `start_date / end_date` Y-m-d (end $\geq$ start), `duration_type` $\in \{ \text{full_day}, \text{half_day_first}, \text{half_day_second} \}$.
- **Data entities:** `leave_requests`, `leave_request_days` (per counted working day, `day_value` 1.00/0.50, void support).

4.15 Attendance Correction Requests

- **Business purpose:** Let employees dispute a day — missing check-in/out, incorrect time, wrong status, other — with evidence; HR decides; approval rewrites the day **without destroying history**.
- **Endpoints:**

Method	Path	Permission	Notes	
POST	<code>/correction-requests</code>	correction-create	201; duplicate pending for same date $\Rightarrow$ 409	
POST	<code>/correction-requests/{id}/cancel</code>	correction-create	Policy-authorised cancel	
GET	<code>/correction-requests</code>	correction-create \	correction-approve	Query <code>status</code> validated against allowed set (invalid $\Rightarrow$ 422), <code>employee_id</code> defaults <code>me</code> , <code>from</code> , <code>to</code>
GET	<code>/correction-requests/{id}</code>	correction-create \	correction-approve	Includes <code>approval_request_uuid/status</code> and a <code>current_day</code> snapshot (record metrics + every punch incl. superseded) so approvers need no separate record permission
POST	<code>/correction-requests/{id}/approve</code>	correction-approve	Direct-decision fallback when no platform approval request backs the row (e.g., approval disabled at submission); policy <code>approve check</code>	
POST	<code>/correction-requests/{id}/reject</code>	correction-approve	Inline validation: <code>reason</code> required string $\leq$ 500	

- **Submission validation (verified `StoreCorrectionRequestRequest`):** `attendance_date` required date; `request_type` $\in \{ \text{missing_in}, \text{missing_out}, \text{incorrect_time}, \text{wrong_status}, \text{other} \}$; `requested_check_in` required for `missing_in/incorrect_time` (Y-m-d H:i:s); `requested_check_out` required for `missing_out/incorrect_time`; `attendance_type_override_id` required for `wrong_status` ($\exists$ `attendance_types`); `reason` $\leq$ 5000 required; `attachment` optional file, mimes from config `employee.documents.allowed_mimes` (default pdf,jpg,jpeg,png,doc,docx), size cap `employee.documents.max_file_size_kb` (default 5120).
- **Service guards (verified `CorrectionRequestService`):** own-record only (mismatched `employee_id` $\Rightarrow$ permission denial); correction window (spec: `attendance.correction_window_days`, default 30); no second pending request per date; requested times must fit the resolved shift window; locked-month state recorded on the row (`month_locked`) and overriding it at decision time requires `attendance.correction-override-lock` with prominent audit logging (executor constant + spec).
- **Approval effects (verified `AttendanceCorrectionExecutor`):** inserts replacement punches with `source>manual` and `correction_request_id`, stamps replaced originals `superseded_by_id` (originals never mutated/deleted), validates the resulting sequence (two consecutive identical types $\Rightarrow$ 422), recalculates the day preserving its snapshot; `wrong_status` overrides the day's `attendance_type_id` with an audit log; rejection changes nothing but status/reason. Fires `CorrectionDecided` (imported).
- **Data entity:** `correction_requests` (incl. `attendance_type_override_id` added by migration 2026_08_20).

4.16 Monthly Attendance Approval (close the month → payroll-ready)

- **Business purpose:** Aggregate a person-month into audited totals, surface unresolved issues, approve (optionally overriding), lock the underlying days, and mark the month ready for payroll; unlock reverses this while the month is not yet frozen by Finance.
- **Endpoints:**

Method	Path	Permission	Notes
POST	/monthly-attendance/build	monthly-approve	With <code>employee_id</code> : synchronous single summary (201 resource). Without: bulk build queued (202 batch payload; empty selection ⇒ validation error "No employees found in the specified criteria."). Rebuilding an approved month ⇒ 409
GET	/monthly-attendance	monthly-view	Filters: month, year, status, ready_for_payroll, department_id, search, per_page
GET	/monthly-attendance/{id}	monthly-view	Detail
GET	/monthly-attendance/{id}/breakdown	monthly-view	Day-level records of that month (filters status, per_page; user-scoped)
GET	/monthly-attendance/batch/{batchId}	monthly-view	Cache-backed batch status (TTL 24h; 404 when absent)
POST	/monthly-attendance/{id}/approve	monthly-approve	Body <code>override</code> nullable bool. Already-approved ⇒ 409. Unresolved without override ⇒ 409 listing issues. Override is activity-logged (<code>monthly_attendance.approved_with_override</code>)
POST	/monthly-attendance/bulk-approve	monthly-approve	<code>ids</code> required ∃ <code>monthly_attendance_approvals</code> ; queues chunks of 50; 202 <code>{mode: "queued", batch_id, ...}</code>
POST	/monthly-attendance/{id}/unlock	monthly-unlock	<code>reason</code> required ≤255. Frozen month ⇒ 409 "Cannot unlock a frozen month." Resets <code>is_locked/ready_for_payroll</code> , stores reason, sets status pending, unlocks the days

- **Build totals (verified):** sums present/absent/leave/unpaid-leave/half days, late count, overtime & working hours from the month's records; future months count elapsed days only. `unresolved_flag` becomes true when any of: missing check-out day found, fewer records than elapsed days (unassigned/missing data), **pending correction requests, pending leave requests** — the response-visible issue strings come from `getUnresolvedIssuesList`.
- **Approve effects (verified gateway + executor):** submission through `ApprovalGateway` (`monthly-approve`, entity `monthly_attendance_approvals`, correlation `monthly_attendance:{id}`); executor sets `status=approved`, `approved_by/at`, `is_locked=true`, `ready_for_payroll=true` and locks every `attendance_records` row of that employee-month; event **MonthApproved** fires. Spec §5.3 confirms the executor is the only place these transitions happen.
- **Data entity:** `monthly_attendance_approvals` (unique per company+employee+month+year; carries `frozen_at/frozen_by/unfreeze_reason`).

5. Background jobs & scheduled commands (verified inventory)

Queue jobs: `BuildBulkMonthlyAttendanceJob`, `ProcessBulkMonthlyApproveJob`, `ProcessBulkAssignmentJob`, `ProcessCloseDayChunkJob`, `ProcessLeaveAccrualChunkJob`, `ProcessCarryForwardChunkJob`, `ProcessPolicyGroupBackfillChunkJob`, `QueueAttendanceExportJob`.

Console schedulers: `ScheduleCloseDayCommand` (hourly; fires a company's batch when local hour == 2 — build sequence 9.1-BE), `ScheduleLeaveAccrualCommand`, `ScheduleLeaveCarryForwardCommand` (pair with the manual triggers above — 9.2-BE), `ScheduleEligibilitySweepCommand` (announces recent organization-assignment changes to policy-group auto-assignment).

6. Domain events raised (classes verified)

PunchCreated (dispatched in PunchService) · AssignmentChanged (imported by AssignmentService) · AttendanceCalculated · LeaveDecided (dispatched in LeaveRequestExecutor) · LeaveDayVoided · CorrectionDecided (imported by AttendanceCorrectionExecutor) · MonthApproved (dispatched in MonthlyAttendanceApprovalService).

7. Database entities owned by the module

Migrations present (verified filenames): shifts, attendance_types, attendance_policies, assignments (+ scope-normalization & identity-constraint migration, policy_group_id addition), attendance_punches, leave_balances, leave_balance_ledger, attendance_records, correction_requests (+ attendance_type_override_id migration), leave_requests, leave_request_days, monthly_attendance_approvals, attendance_policy_groups, attendance_policy_group_policy.

Seeders: AttendanceDatabaseSeeder, AttendanceTypeSeeder, ShiftSeeder.

8. Integrations with other modules (explicitly confirmed)

Module / Platform service	Evidence
Employee	EmployeePersonalInfo relations on punches/records/corrections/leave/monthly; exists:employee_personal_infos,id validations; repositories for close-day & bulk-build employee selection; user->employeeProfile self-service links
Configuration	Scope models Branch/Division/Department/Section/Team imported by AssignmentService::scopeModelMap(); departments existence check in monthly build validation
Payroll	Modules\Payroll\Support\PayrollPeriodGuard gates close-day; ready_for_payroll + frozen_at hand-off consumed by Payroll snapshots (§9)
Platform Tenancy	TenantContext company scoping everywhere; X-Company-Id header convention
Platform Permissions	permission: middleware + PermissionEngineContract point-checks (punch-others, record-view-all, leave-approve, leave-balance-view, record-recalculate, override-lock)
Platform Approval Gateway/Engine	leave-approve, correction-approve, monthly-approve submissions with correlationIds; executors registered per entity type (leave_request , attendance_correction , monthly_attendance)
Activity Log	health view, unresolved-month override approval, correction lock-override & wrong-status override
Excel (Maatwebsite)	attendance export streaming + queued export

9. Payroll hand-off facts (from spec + build sequence, code-anchored where possible)

- An approved month sets ready_for_payroll = true; Finance-side **freeze/unfreeze lives in the Payroll module** (payroll.month-freeze , segregation-of-duties with attendance.monthly-approve). Freeze requires approved status; unfreeze requires a reason and, for paid runs, payroll.month-unfreeze-paid .
- attendance_snapshots** (card 8.0-BE, done): immutable, timestamp-less, unique(payroll_run_id, employee_id) ; copied verbatim from frozen monthly approvals — no recalculation; rebuild idempotent.

Four endpoints exist **inside the Payroll module** (build draft-only under `payroll.run-create`, list, show, diagnostic divergence). They are therefore not part of this module's route file.

- Payslips must consume attendance exclusively via

`AttendanceSnapshotServiceInterface::getPayrollAttendanceForRun()`; a guard test proves neither `attendance_records` nor `monthly_attendance_approvals` is queried on that path. Payslip generation itself (8.2a-BE) is **not yet in the codebase** (pending card).

10. Error-behavior cheat-sheet (verified codes/messages)

Situation	Result
Missing company context	<code>PermissionDeniedException('company.context')</code>
Punch on unassigned date	422, code: <code>unassigned</code>
Duplicate consecutive punch type	422 with last-punch message
Date-range >366 days on records list	422 "Date range cannot exceed 366 days."
Invalid correction/leave status filter value	422
Second pending correction same date	409 <code>DuplicateCorrectionRequestException</code>
Approve already-approved month / rebuild approved month / unlock frozen month	409
Unresolved month approve w/o override	409 with comma-joined issue list
Negative leave-balance adjust w/o <code>allow_negative</code>	<code>DomainException</code> "Adjustment rejected: ..."
Overlap conflicts on assignment create/update	<code>AssignmentOverlapException</code> error response
Recalculate locked record / unauthorized live override	Dedicated exceptions (<code>AttendanceRecordLockedException</code> , <code>RecalculationUnauthorizedException</code>)

Anything not enumerated above (exact transport envelopes, rate limits, additional HTTP codes) — **Not specified in the reviewed source files.**

11. What this guide intentionally does not cover

- Front-end screens/UI flows (only backend routes, services, jobs, events reviewed; FE cards referenced in the build sequence were not inspected).
- Role seeding mechanics beyond the spec sentence quoted in §2.
- Exact values for: bulk-assignment sync/queue threshold, pagination envelope internals of the base controller, queue connection names.

For all of the above: **Not specified in the reviewed source files.**

Appendix A — Reviewed sources

1. `backend/Modules/Attendance/routes/api.php` (291 lines, complete)
2. Controllers (17): `AttendanceController`, `PunchController`, `AttendanceRecordController`, `AttendancePolicyController`, `AttendanceTypeController`, `Shift/ShiftManagementController`, `AssignmentController`, `BulkAssignmentController`, `AssignmentResolutionController`, `AttendanceRecalculationController`, `PolicyGroupController`, `LeaveBalanceController`,

LeaveBalanceJobController, CloseDayJobController, LeaveRequest/LeaveRequestManagementController, CorrectionRequestController, MonthlyAttendanceApprovalController

3. Form Requests (45 files — all rules quoted verbatim)

4. Models: Assignment, AttendancePolicy, AttendancePolicyGroup, AttendancePunch, AttendanceRecord, AttendanceType, CorrectionRequest, LeaveBalance, LeaveBalanceLedger, LeaveRequest, LeaveRequestDay, MonthlyAttendanceApproval, Shift

5. Service contracts (22 interfaces) and implementations: PunchService, AttendanceRecordService, AttendanceRecalculationService, AssignmentService, AssignmentResolutionService, CorrectionRequestService, LeaveRequestService, LeaveBalanceService, MonthlyAttendanceApprovalService, CloseDayService, AttendanceTypeService, AttendancePolicyGroupService, PolicyGroupEligibilityEvaluator, BulkAssignmentService contract

6. Approval executors: MonthlyAttendanceExecutor, AttendanceCorrectionExecutor, LeaveRequestExecutor

7. Support: EligibilityAttributeRegistry, GenderNormalizer, BatchProgressStore; DTOs (ResolvedContext, EmployeeEligibilityContext, ResolvedPolicyAssignment, LeaveEvaluation, AutoAssignmentOutcome, PolicySnapshotData)

8. Jobs (8), Events (7), Resources (14 incl. Applicable/Eligibility attribute resources), Exports, Console schedulers (4), Providers (RouteServiceProvider), Migrations (17), Seeders (3)

9. Reference documents: ATTENDANCE_PAYROLL_MODULE_SPEC.md (§0 ground rules, §3 data model, §4 permissions, §5 approvals, §6 contracts, Parts D–F stories), ATTENDANCE_PAYROLL_BUILD_SEQUENCE.md (wave progress incl. 8.0/8.1/8.2 cards, 9.1/9.2 jobs, 6.1/6.2 monthly close/freeze)

10. Style reference: EMPLOYEE_MODULE_GUIDE.md / EMPLOYEE_MODULE_GUIDE_BN.md

Attendance Module — ব্যবসা ও API গাইড

এই গাইডটি ERPFLOW ব্যাকএন্ডের **Attendance** মডিউলের প্রতিটি নিশ্চিত ফিচারকে ডকুমেন্ট করে। নিচের প্রতিটি বাক্য যুক্তিসঙ্গতভাবে যুক্ত উৎস ফাইলের ওপর ভিত্তি করে (আপেন্ডিক্স এ-এ তালিকাভুক্ত)। যদি কোনো বিবরণ যাচাইয়েও পাওয়া না যায়, তবে এই গাইডটি স্পষ্টভাবে লিখবে: **"Not specified in the reviewed source files!"**

English version: [ATTENDANCE MODULE GUIDE EN.md](#) · বাইলিংুয়াল PDF: [ATTENDANCE MODULE GUIDE.pdf](#)

১। এই মডিউল কী?

Attendance মডিউল কর্পোরেটের **ডিজিটাল সময়-ও ছুটি ইঞ্জিন**। যা যাচাই করা গেল:

- ক্যাপচার করা **punch** (একদিনে একাধিক punch, একাধিক সোর্স থেকে)
- মাস্টার ডাটা: **Shifts, Attendance Types, Leave/Holiday Policies**
- shifts এবং policies সংযুক্ত করা স্কোপে (**Assignments**) — কর্পোরেট স্তর থেকে একজন কর্মচারী পর্যন্ত
- ঠিক একটি তারিখে কোন কর্মচারী কী ধরে (এটা **Assignment Resolution**)
- একটি ফ্রিন করা policy snapshot সহ **daily attendance records** হিসাব
- schedules বা ম্যানুয়ালি **Close Day** হিসাব করা জব
- **Leave Balances**, অ্যাকুয়াল, ক্যারি-ফরোড এবং ম্যানুয়াল সমন্বয় একটি পূর্ণ ledger সহ
- **Leave Request** চক্রপূর্ণ ওয়ার্কফ্লো (preview → apply → approve/reject/cancel)
- **Attendance Correction Requests** (missing/incorrect punch, wrong status)
- **Monthly Attendance Approval** দিয়ে মাস বন্ধ করা, রেকর্ড লক করা এবং `ready_for_payroll` ফ্ল্যাগ করা

গুরুত্বপূর্ণ: সবকিছু আপনার **বর্তমান কোম্পানি** সংগে যুক্ত। প্রতিটি কুয়েরীকে platform tenancy context দ্বারা স্কোপ করা হয়; কোম্পানি রিকোয়েস্টে `X-Company-Id` হেডার হিসেবে আসে।

২। কেরা এটা ব্যবহার করবে?

permission: মিডলওয়্যার গঠন আর করা পূর্তি চাবিগুলি চালু করে। নিচের রেজিস্ট্রি স্পেক থেকে (সেকশন ৪) আর রুট ব্যবহার জুড়ে ক্রস-চেক করা হয়েছে:

পারমিশন কী	কি অনুমোদন দেয়
<code>attendance.menu-view</code>	দেখতে Attendance সেকশন; policies, attendance types, shifts, assignments পড়তে; punches তালিকা
<code>attendance.config-manage</code>	attendance types, shifts, policies ক্রিয়াক্ষমক (CRUD)
<code>attendance.policy-group-view</code>	policy groups ও eligibility attributes দেখতে
<code>attendance.policy-group-manage</code>	policy groups তৈরি/এডিট/স্ট্যাটাস/ডিলিট; eligibility preview; কর্মচারীদের প্রয়োগ/ব্যাকফিল
<code>attendance.assignment-manage</code>	assignments তৈরি/আপডেট/শেষ; bulk assign; রেকর্ড পুনরগণনা endpoint ট্রিগার
<code>attendance.assignment-preview</code>	assignment resolution নির্ণয়ীয় টুল
<code>attendance.punch-create</code>	punch in/out
<code>attendance.punch-create-others</code>	অন্য কর্মচারীর পক্ষে punch/তালিকা (PunchService ভেতরে যাচাই করা হয়)

পারমিশন কী	কি অনুমোদন দেয়
attendance.record-view-own / -team / -all	রেকর্ড দৃশ্যমানতা স্তর
attendance.record-export	দৈনিক সারাংশ এক্সপোর্ট
attendance.record-recalculate	"বর্তমান policy দিয়ে পুনর্গণনা" ওভাররাইট (পুনর্গণনা সেবার ভেতরে) এবং close-day জব endpoints
attendance.correction-create	নিজের correction request জমা/বাতিল; correction কিউ দেখা
attendance.correction-approve	correction অনুমোদন/প্রত্যাখ্যান (সরাসরি fallback path)
attendance.correction-override-lock	লকড মাসে correction অনুমোদন (AttendanceCorrectionExecutor ভেতরে যাচাই)
attendance.leave-apply	নিজের leave আবেদন/প্রিভিউ/বাতিল
attendance.leave-approve	leave অনুমোদন/প্রত্যাখ্যান; যেকোনো কর্মচারীর leave দেখা
attendance.leave-balance-view	সকল ব্যালেন্স তালিকা; অন্য কর্মচারীর ledger ইতিহাস
attendance.leave-balance-adjust	ম্যানুয়াল ব্যালেন্স সমন্বয়; accrual / carry-forward জব চালু
attendance.monthly-view	মাসিক অনুমোদন গ্রিড, বিবরণ, ব্রেকডাউন, ব্যাচ স্ট্যাটাস
attendance.monthly-approve	মাস বানানো/অনুমোদন/বান্ধ অনুমোদন
attendance.monthly-unlock	অনুমোদিত মাস আনলক

স্পেক অনুসারে, স্ব-সেবা (self-service) কীগুলি (attendance.record-view-own , attendance.punch-create , attendance.correction-create , attendance.leave-apply) একটি seeded employee ভূমিকার অন্তর্ভুক্ত — যা প্রতিদিন পাওয়া যেন।

৩। প্রতিটি endpoint প্রয়োগ করে কী কনভেনশন?

কনভেনশন	তথ্য (verified)
বেস URL	/api/v1/attendance (RouteServiceProvider)
Route-name prefix	api.attendance.
গ্লোবাল মিডলওয়্যার	auth:api , onboarding.access , onboarding.document — মডিউলের প্রতিটি route আবৃত
রুটলেভেল অথরাইজেশন	permission:<key> মিডলওয়্যার গোষ্ঠী (কীগুলি §২ এ)
টেন্যান্সি	কোম্পানি id TenantContext থেকে আসে; কোম্পানি অনুপস্থিতি হলে PermissionDeniedException('company.context') । হেডার: X-Company-Id (spec §0.3)
কর্মচারী পরিচয়	employee_id employee_personal_infos.id -এর রেফারেন্স; কিছু validation exists:employee_personal_infos,id ব্যবহার করে
অ্যাপ্রুভেল ইঞ্জিন	leave, correction এবং monthly অনুমোদন platform ApprovalGateway-এর মধ্য দিয়ে যায়; যখন কোনো action জন্য approval বন্ধ থাকে, onApproved সিঙ্ক্রোনাসভাবে চলে (একটি ক্রটি নয়)
পেজিনেশন	তালিকা endpoint পেজিনেটেড রিসোর্স ফেরত দেয়; ডিফল্ট সাইজ ১৫ (যদি না অন্যথা উল্লেখ করা হয়)

৪। ফিচার ক্যাটালগ

৪.১ মডিউল স্বাস্থ্য যাচাই (Health Check)

- **ব্যবসা উদ্দেশ্য:** যে মডিউলটি স্বাস্থ্য আছে এবং এর assignment-resolution cache বাস্তবে আইনভল্ভ করা যায় কিনা তা যাচাই করা।
- **কী করে:** cache store এর নাম, ট্যাগ সাপোর্ট, সামগ্রিক status (ok / degraded) এবং একটি detail বার্তা ফেরত দেয়। Activity event attendance.health_viewed লগ করে।
- **Endpoint:** GET /health → AttendanceController@health (অতিরিক্ত permission নেই)।
- **Response:** JSON {module: "attendance", status: "ok"|"degraded", cache:{store, supports_tags, supported, detail}}। ট্যাগ সাপোর্ট না থাকলে degraded (কারণ resolved assignments ২৪ ঘণ্টা পর্যন্ত stale থাকতে পারে)।
- **ডেটা এন্টিটি:** নেই (cache.default কনফিগ থেকে পড়ে)।

8.২ Attendance Types (মাস্টার ডাটা)

- **ব্যবসা উদ্দেশ্য:** স্ট্যাটাসের ভাষা সংজ্ঞায়িত করা — Present, Absent, Late, Half Day, Leave, Holiday, Weekend, WFH, Business Trip, Missing Check-In, Missing Check-Out — সহ কোম্পানির নিজস্ব স্ট্যাটাস।
- **কী করে:** CRUD + স্ট্যাটাস টগল। System-seeded রোয়ে system_code আছে; HR-তৈরি রোয়ের system_code = NULL। হিসাব ইঞ্জিন system_code দিয়েই স্ট্যাটাস সলিড করে (নাম/id নয়)।
- **Endpoints** (read attendance.menu-view; write attendance.config-manage):

পদ্ধতি	Path	Controller@action
GET	/attendance-types	AttendanceTypeController@index
GET	/attendance-types/{id}	AttendanceTypeController@show
POST	/attendance-types	AttendanceTypeController@store (201)
PUT	/attendance-types/{id}	AttendanceTypeController@update
PATCH	/attendance-types/{id}/status	AttendanceTypeController@updateStatus
DELETE	/attendance-types/{id}	AttendanceTypeController@destroy

- **Validation (verified):** name required ≤100, company-এ ইউনিক · code required ≤50, regex `^[A-Z0-9_]+$`, company-এ ইউনিক · system_code **prohibited** · category required ≤50 · is_paid, counts_as_working_day, eligible_for_payroll required bool · color ≤30, icon ≤50 required · স্ট্যাটাস পরিবর্তন active|inactive মাত্র। Update বেসিক ৪টিতে uniqueness যাচাই।
- **Delete guard:** সিস্টেম স্ট্যাটাস ডিলিট করা যায় না (409); যদি records আছে তবে ডিলিট করা যায় না (409)।
- **ডেটা এন্টিটি:** attendance_types টেবিল।

8.৩ Shifts

- **ব্যবসা উদ্দেশ্য:** কাজের সময় টেমপ্লেট — সূচনা/শেষ সময়, টাইমজোন, বিরতি, grace period, present/half-day অনুপাত, কর্মদিবস, overnight flag।
- **Endpoints:**

পদ্ধতি	Path	Permission	Controller@action
GET	/shifts	menu-view	Shift\ShiftManagementController@index
GET	/shifts/{shiftId}	menu-view	...@show
POST	/shifts	config-manage	...@store (২০০ "Shift Create Successfully Done")
PUT	/shifts/{shiftId}	config-manage	...@update
PATCH	/shifts/{shiftId}/status	config-manage	...@updateStatus
DELETE	/shifts/{shiftId}	config-manage	...@destroy ("Shift archived successfully.")

- **Validation (verified):** `name` ≤ 150 (create-এ required, update-এ nullable) · `start_time / end_time` required `H:i` · `timezone` nullable valid timezone · `break_minutes` `int` ≥ 0 · `working_hours` `numeric` $> 0 \leq 24$ · `grace_minutes` `int` ≥ 0 · `min_hours_present` `numeric` $> 0 \leq 24$ · `min_hours_half_day` `numeric` $\geq 0 \leq 24$ · `working_days` required array ≥ 1 distinct integers 1–7 (ISO-8601: ১ = সোমবার) · `is_overnight` required `bool` · `status` nullable `active|inactive` |
- **কোড তথ্য:** তালিকা endpoint `per_page` ডিফল্ট ১।
- **ডেটা এন্টিটি:** `shifts` টেবিল।

8.8 Attendance Policies (Leave & Holiday)

- **ব্যবসা উদ্দেশ্য:** Leave সুবিধা ও holiday ক্যালেন্ডার নিয়ম লিখে, effective-ডেটেড, JSON `config` সহ।
- **কী করে:** দুটি ধরণ — `leave` (entitlement, accrual, carry-forward, encashment, half-day, LWP, advance notice, backdate limit, document threshold) এবং `holiday` (holiday এন্ট্রিগুলি + optional `recurring`)। Config shape ভেরিয়েটর (`HolidayPolicyConfigRule` / `LeavePolicyConfigRule`) দ্বারা ধরণ অনুযায়ী যাচাই হয়।
- **Endpoints:** read `menu-view` (GET `/policies`, GET `/policies/{id}`); write `config-manage`: POST `/policies` (201), PUT `/policies/{id}`, PATCH `/policies/{id}/status`, DELETE `/policies/{id}` |
- **Validation (verified):** create-এ `policy_type` $\in \{leave, holiday\}$, `name` ≤ 150 , `code` ≤ 50 , `effective_date` `date`, `config` array, optional `status` $\in \{Active, Inactive, Archived\}$ । স্ট্যাটাস পরিবর্তন একই সেট।
- **ডেটা এন্টিটি:** `attendance_policies` (`policy_type`, `config json-cast`)।

8.৫ Policy Groups (eligibility bundles + apply/backfill)

- **ব্যবসা উদ্দেশ্য:** একটি **eligibility rule document** সহ `policies` বান্ডল করে — HR বলতে পারে "কে কী পায়", তারপর সেটিকে কর্মচারীদের সাধারণ `assignments` হিসেবে প্রয়োগ করে।
- **কী করে (verified model docblock):** Groups কেবলমাত্র একটি লেখার স্তর — runtime-এ কোনো group দিয়ে সমাধান করা হয় না; group প্রয়োগ করলে `assignments` রো তৈরি হয়। Group এডিট করেও কাউকে প্রভাবিত করে না; প্রয়োগ আলাদাভাবে করতে হয় এবং preview আগে নিতে হয়।
- **Endpoints:**

পদ্ধতি	Path	Permission	উদ্দেশ্য
GET	<code>/policy-groups</code>	<code>policy-group-view</code>	পৃষ্ঠাপত্র তালিকা
GET	<code>/policy-groups/attributes</code>	<code>policy-group-view</code>	রিল বিস্তারের ভাষা: <code>attributes + operators</code>
GET	<code>/policy-groups/{id}</code>	<code>policy-group-view</code>	বিস্তারিত
POST	<code>/policy-groups</code>	<code>policy-group-manage</code>	তৈরি (201)
POST	<code>/policy-groups/eligibility/preview</code>	<code>policy-group-manage</code>	প্রিভিউ — কতজন active কর্মচারী (<code>matched_count</code> , <code>total_active</code> , <code>sample</code>)
PUT	<code>/policy-groups/{id}</code>	<code>policy-group-manage</code>	আপডেট
PATCH	<code>/policy-groups/{id}/status</code>	<code>policy-group-manage</code>	Active/Inactive
DELETE	<code>/policy-groups/{id}</code>	<code>policy-group-manage</code>	মুছে ফেলা
POST	<code>/policy-groups/{id}/apply/preview</code>	<code>policy-group-manage</code>	ব্যাকফিল প্রিভিউ (<code>eligible / already_assigned / new / conflicts / skipped + sample</code>)

পদ্ধতি	Path	Permission	উদ্দেশ্য
POST	/policy-groups/{id}/apply	policy-group-manage	চাপার ব্যাকফিল (202)
GET	/policy-groups/{id}/apply/{batchId}	policy-group-manage	ব্যাচ স্ট্যাটাস

- **Eligibility attributes (verified registry):** `employment_type_id`, `grade_id`, `branch_id`, `division_id`, `department_id`, `section_id`, `team_id` (কোম্পানি-স্কোপড টেবিল থেকে) এবং `gender`, `probation_status` (নরমালাইজড স্ট্রিং)। `operators`: `in`, `not_in`, `equals`, `not_equals`। ফাঁকা rule সমগ্র কোম্পানি মানে ("Company General")। `match` হতে পারে `all` (ডিফল্ট) অথবা `any`।
- **Validation (verified):** `name` ≤150, `code` ≤50 (দুটোই company-এ ইউনিক — সেবা-স্তর) · `description` ≤1000 nullable · `eligibility` required array (`PolicyGroupEligibilityRuleValidator` দ্বারা; কোম্পানির আছে এমন আইডি যাচাই করে) · `effective_date_strategy` ∈ { `joining_date`, `probation_end_date` } (ডিফল্ট `joining_date`; `probation fallback joining_date`) · `status` ∈ { `Active`, `Inactive` } · `policy_ids` array of ints যারা সক্রিয় হতে হবে। ব্যাকফিল অনুরোধে optional `date` এবং `sample_size` (0-100 apply-preview; 0-50 eligibility-preview)।
- **ডেটা এন্টিটি:** `attendance_policy_groups`, `pivot attendance_policy_group_policy`, উৎপন্ন `assignments` (সাথে `policy_group_id` stamped)।
- **অটো-অ্যাসাইনমেন্ট:** `PolicyGroupAutoAssignmentService` একজন কর্মচারীর policy assignments স্বয়ংসম্পূর্ণতা করে; `EligibilitySweepService` সাম্প্রত যুক্ত কর্মালয় সংযুক্তি ঘোষণা করে।

8.৬ Assignments (shifts/policies স্কোপে যুক্ত করা)

- **ব্যবসা উদ্দেশ্য:** ঠিক কোন স্কোপে কোন shift, কোন leave policies ও holiday ক্যালেন্ডার কোনও প্রয়োজনীয় effective তারিখে প্রযোজ্য।
- **স্কোপ মডেল:** `scope_type` ∈ hierarchy `company` → `branch` → `division` → `department` → `section` → `team` → `employee`; resolution সবসময় সবচেয়ে নির্দিষ্ট থেকে। `assignable_type` ∈ { `shift`, `policy` }; policy-এর জন্য `assignable_subtype` ∈ { `leave`, `holiday` } এবং মিলতে হবে policy-এর ধরণের সাথে। কার্ডিনালিটি: `shift` ও `holiday` স্কোপে বিলকুল; `leave` মাল্টি। Sources: `manual`, `auto`, `bulk`। স্ট্যাটাস: `Active`, `Inactive`।

পদ্ধতি	Path	Permission	নোট
GET	/assignments	menu-view	ফিল্টার: <code>scope_type</code> , <code>scope_id</code> , <code>assignable_type</code> , <code>assignable_subtype</code> , <code>source</code> , <code>history</code> , <code>per_page</code> ≤100
GET	/assignments/{id}	menu-view	বিস্তারিত
POST	/assignments	assignment-manage	আইডেম্পোটেন্ট (<code>assignOrGet</code>) — একই assignable+স্কোপ+effective তারিখ আবার না লিখে আগের রো রিটার্ন করে। কেবলমাত্র Active shifts/policies যুক্ত করা যায়। <code>overlap conflict</code> → <code>AssignmentOverlapException</code>
PATCH	/assignments/{id}	assignment-manage	<code>end_date</code> , <code>status</code> (partial)
POST	/assignments/{id}/end	assignment-manage	<code>end_date</code> আবশ্যিক
POST	/assignments/bulk	assignment-manage	<code>filter{scope_type,scope_id}</code> (company...team) অথবা <code>employee_ids</code> ; সিল্ক রেসপন্স অথবা 202 queued { <code>batch_id</code> , <code>status</code> , <code>total</code> , <code>processed</code> , <code>succeeded</code> , <code>failed</code> }
GET	/assignments/bulk/{batchId}	assignment-manage	ব্যাচ অগ্রগতি

- **Create validation:** `shift`-এর জন্য `subtype` অবশ্যই null; `company` স্কোপে `scope_id` নিষিদ্ধ, অন্যথায় required; `effective_date` required; `end_date` ≥ `effective_date`। Bulk-এ `filter` বা `employee_ids` অবশ্যিক।

- **Queued mode শ্রেণীভুক্ত:** সিল্ক vs queued সিদ্ধান্ত শ্রেণীভুক্তের ওপরে নির্ভর করে; সংখ্যাগত মান **Not specified in the reviewed source files**।
- **ডেটা এন্টিটি:** `assignments`; স্কোপ বিদ্যমানতা (`assertScopeExists`) যাচাই করে।
- **ইভেন্ট:** `AssignmentChanged` ক্লাস `AssignmentService` দ্বারা ইম্পোর্টেড।

8.৭ Assignment Resolution (নির্ণায়ী + ইঞ্জিন)

- **ব্যবসা উদ্দেশ্য:** "এই কর্মচারী কী ধরে?" — punch, calculation এবং leave ইঞ্জিনগুলোই একই উত্তর ব্যবহার করে।
- **Endpoint:** `GET /resolve/assignment?employee_id=&date=` (`attendance.assignment-preview`)। **Validation:** `employee_id` required integer; `date` required Y-m-d ।
- **Response (verified DTO):** {`shift`, `leave_policies`, `holiday_calendar`, `timezone`, `resolved_from`} (`resolved_from` জয়তক স্কোপ নাম্বারে)। `timezone` fallback `shift` → `company`। কোন active shift না থাকলে `unassigned` ব্যাগে validation error ("No active shift found for the given date.")।
- **Cache:** ২৪ ঘণ্টা TTL, `company/employee/date` কী, ট্যাগ-ভিত্তিক আইনভঙ্গ্যোগ্য।

8.৮ Punch In/Out (append-only মাল্টি-punch)

- **ব্যবসা উদ্দেশ্য:** Web, mobile, biometric, API বা manual থেকে স্বয়ংক্রিয়ভাবে আঘাত সংগ্রহ — পরের পর্যন্ত পরিবর্তন নয়, শুধুমাত্র সুপারসিড করা যায়।
- **Endpoints:** `POST /punch` (`attendance.punch-create`) • `GET /punches?employee_id=&date=` (`attendance.menu-view`)।
- **Validation (verified `StorePunchRequest`):** `punch_type` ∈ { `in`, `out` } • `source` ∈ { `web`, `mobile`, `biometric`, `api`, `manual` } • `punch_time` nullable Y-m-d H:i:s • `employee_id` nullable int • `ip_address` ≤45 • `device_info` ≤255 • `remarks` ≤255।
- **Behavior (verified `PunchService`):**
 - punch কলার নিজস্ব প্রোফাইলে যায়; অন্য `employee_id` দিলে `attendance.punch-create-others` প্রয়োজন (না হলে 403)। তালিকায়ও একই আবেদন।
 - "আজ" সংগণনা কোম্পানির টাইমজোনে (fallback `Asia/Dhaka`)।
 - প্রথমে assignment resolution; কোন shift না থাকলে **422**, `code: unassigned` ("No shift is assigned for this date.")।
 - Overnight shift: স্থানীয় সময় `start_time` -এর আগে হলে আগের `attendance_date` -এ যায়।
 - একই ধরনের দুটি ক্রমিক punch প্রতিবাদী (422, শেষ punch সময় বার্তায়)।
 - রো read-only (`UPDATED_AT` = null), দৈনিক `sequence_no` সহ; পরে `superseded_by_id` -এ ref (approved correction দ্বারা সেট করা হয়; মূল never mutated/deleted)।
 - `PunchCreated` event ফায়ার হয়।
- **Response:** 201 `PunchResource` (`id`, `company_id`, `employee_id`, `attendance_date`, `punch_type`, `punch_time`, `ISO-8601`, `source`, `ip_address`, `device_info`, `remarks`, `sequence_no`, `superseded_by_id`, `correction_request_id`, `created_at`)।
- **ডেটা এন্টিটি:** `attendance_punches` টেবিল।

8.৯ Daily Attendance Records (দেখা ও এক্সপোর্ট)

- **ব্যবসা উদ্দেশ্য:** প্রতি কর্মচারী/দিন একটি হিসাব রো — প্রথম in, শেষ out, কাজ/ওভারটাইম ঘণ্টা, দেরি/আগে আসা, punch সংখ্যা, প্রয়োগিত policy snapshot, lock ক্ল্যাগ।
- **Endpoints:**

পদ্ধতি	Path	Permission	নোট		
GET	<code>/attendance-records/today</code>	<code>record-view-own</code> \	<code>-team</code> \	- all	কলার employee

পদ্ধতি	Path	Permission	নোট		
					profile লিংক আবশ্যক (না থাকলে 404); today পুনরায় হিসাব করে রেসপন্স
GET	/attendance-records	same	ফিল্টার employee_id, start_date, end_date, status ; range >366 দিন ⇒ 422; ভিজিবিলিটি রেপোজিটরিতে যুক্ত ইউজার ধরে		
GET	/attendance-records/{id}	same	404 যদি না থাকে; record-view-all ছাড়া রেকর্ডটি কলার স্কোপে মিলতে হবে, না মিললে PermissionDeniedException('attendance.record- view')		
GET	/attendance-records/{id}/punches	same	সেই দিনের সব punch (superseded সহ)		
GET	/attendance-records/export	record-export	format query ডিফল্ট xlsx ; ≤2000 সারি (কনফিগ attendance.exports.export_queue_threshold , ডিফল্ট 2000) Excel-এর মাধ্যমে স্ট্রিম attendance_records_<timestamp>.<format> ; তার ওপর QueueAttendanceExportJob ডিসপ্যাচ → 202 {token} (40-অক্ষর random)		

- **Calculation core (verified AttendanceRecordService::calculateDaily)**: লকড রেকর্ড সরাসরি ফেরত; না হলে non-superseded punches সময় ক্রমে জোড়া দেয়া হয় shift/timezone-এ; weekend / holiday (recurring month-day match সহ) / working-day শ্রেণীবিন্যাস; grace vs late; half-day শ্রেণিহোল্ড shift থেকে; সিস্টেম স্ট্যাটাস on-demand system_code দিয়ে; প্রয়োগিত সেটিংগগুলি policy_snapshot -এ ফ্রিন করা হয় (grace_minutes, working_hours, min-hours present/half-day, working_days, timezone, resolved_at)। লকড দিন পুনরায় হিসাব করা যায় না।
- **ডেটা এন্টিটি**: attendance_records (+ relations employee/shift/attendanceType/punches)।

8.১০ Record Recalculation (snapshot replay vs live override)

- **ব্যবসা উদ্দেশ্য**: ডেটা ঠিক করার পর দিন পুনরাং হিসাব — সাধারণত **stored snapshot** replay করে ঐতিহ্য বজায় রাখে; ঐচ্ছিকভাবে **live policy** পুনরায় সমাধান করে audit trail সহ।
- **Endpoint**: POST /records/{recordId}/recalculate | **রুট মিডলওয়্যার** attendance.assignment-manage | Body (RecalculateAttendanceRecordRequest): use_current_policy **required bool**, অন্যান্য nullable (user_id, date, policy_snapshot, mock_resolved_policy, shift, is_locked, has_permission)।
- **Behavior (verified)**: 404 যদি রেকর্ড না থাকে বা অন্য কোম্পানিতে। লকড রেকর্ড ⇒ AttendanceRecordLockedException | use_current_policy=true -এর সাথে ইঞ্জিন পারমিশন attendance.record-recalculate প্রয়োজন (না হলে RecalculationUnauthorizedException); cache ফ্লাশ, live policy সমাধান, before/after audit log। সাধারণত stored policy_snapshot replay হয় (পুরোনো snapshot বিহীন রেকর্ডের জন্য live resolve + warning log)।

8.১১ Close Day job (ম্যানুয়াল ট্রিগার + স্ট্যাটাস)

- **ব্যবসা উদ্দেশ্য**: নিশ্চিত করুন প্রতিটি সক্রিয় কর্মচারী একটি গুঁটে দিবসের রেকর্ড পায় — ঠিক একই কাজ যা রাতের সময়সূচী করে (9.1-BE: ঘণ্টা ০২:০০ স্থানীয় সময়ে কোম্পানির ব্যাচ)।
- **Endpoints**: POST /jobs/close-day এবং GET /jobs/close-day/{batchId} — রুট পারমিশন attendance.record-recalculate, prefix jobs।

- **Validation (verified CloseDayRequest):** date required Y-m-d , before_or_equal:today , after_or_equal: today-90 days ; employee_id nullable exists:employee_personal_infos,id ।
- **Behavior (verified):** Payroll মডিউলের PayrollPeriodGuard যদি সময়সীমা বন্ধ ঘোষণা করে → **200** "No employees to process or period is locked." নাহয়, রেকর্ড না থাকা সক্রিয় কর্মচারীকে chunk size 200-এ ProcessCloseDayChunkJob এ ভাগ করে **202** {batchId} ফেরত। স্ট্যাটাস endpoint Laravel batch ফিল্ড বা 404 ফেরত দেয়।
- **Cross-module dependency:** Modules\Payroll\Support\PayrollPeriodGuard (এক্সপ্লিসিট ইম্পোর্ট)।

8.১২ Leave Balances (দেখা, সমন্বয়, ledger)

- **ব্যবসা উদ্দেশ্য:** কর্মচারী/policy/বছর অনুযায়ী entitled, used, carried-forward, encashed — সহ একটি পূর্ণ ledger।
- **Available-days সূত্র (verified model accessor):** entitled_days + carried_forward_days - used_days - encashed_days ।
- **Endpoints:**

পদ্ধতি	Path	Authorization	নোট
GET	/leave-balances/me	auth group ছাড়া কিছু নেই	বর্তমান বছর; employee লিংক না থাকলে 403
GET	/leave-balances/{employeeId}/history	internal check	নিজের id সর্বদা; অন্যের জন্য attendance.leave-balance-view আবশ্যিক। year ডিফল্ট বর্তমান বছর; paginated ledger
GET	/leave-balances	leave-balance-view	ফিল্টার employee_id (∃ employee_personal_infos), year 2000-2100, policy_id (∃ attendance_policies)
PATCH	/leave-balances/{id}/adjust	leave-balance-adjust	নিচে দেখুন

- **Adjustment (verified):** body days required numeric ≠0, reason required ≤255, allow_negative nullable bool। FOR UPDATE লক; entitled_days -এ প্রয়োগ; negative ফলে DomainException (allow_negative ছাড়া) এবং সেই transaction-এ manual_adjustment ledger রো (signed days, reason, actor)।
- **Ledger entry types (verified enum):** accrual , carry_forward , consumption , reversal , encashment , manual_adjustment ।
- **ডেটা এন্টিটি:** leave_balances , leave_balance_ledger ।

8.১৩ Leave Accrual & Carry-Forward jobs

- **ব্যবসা উদ্দেশ্য:** মাসিকভাবে policy নিয়ম অনুযায়ী ব্যালেন্স বাড়ান এবং year-end-এ অব্যবহৃত দিন স্থানান্তর করুন — schedule (কলোয়াল কমান্ড আছে) বা ম্যানুয়ালি backlog হিসাবে।
- **Endpoints (পারমিশন attendance.leave-balance-adjust , prefix jobs):**
- POST /jobs/accrue-leave — body: month 1-12, year current±5, optional policy_id (∃ attendance_policies)। chunked accrual; **202** {batchId} অথবা **200** {applied:0} "No active leave assignments found to accrue or accrual already completed."
- POST /jobs/carry-forward — body: from_year current±5, optional policy_id । **202** {batchId} অথবা **200** "No active leave assignments found for carry-forward."
- GET /jobs/{batchId} — Laravel batch progress অথবা 404।
- **Carry-forward নিয়ম (verified service):** policy config carry_forward_allowed ও max_carry_forward (min(available, max)) দ্বারা নিয়ন্ত্রিত; idempotent — লক্ষ্য balance এর জন্য carry_forward ledger রো আছে কিনা চেক করে; স্কোপ historical date অনুযায়ী সমাধান করা হয় (অতীতের বছরের রানের জন্য সেই সংস্থাগত সংযুক্তি ব্যবহার করে)।

8.১৪ Leave Requests (স্ব-সেবা জীবনচক্র)

- **ব্যবসা উদ্দেশ্য:** কর্মচারীরা assigned policies-এর বিরুদ্ধে ছুটি আবেদন করে; অনুমোদক সিদ্ধান্ত নেয়; অনুমোদন leave দিন stamp করে এবং ব্যালেন্স হ্রাস করে।
- **Endpoints (group `attendance.leave-apply` | `attendance.leave-approve` , prefix `/leave-requests`):**

পদ্ধতি	Path	কাজ
GET	<code>/applicable-policies?date=</code>	কলার জন্য প্রযোজ্য policies (assignment-driven, balance নয়, ফলে প্রথমবারও দেখা যায়)। ফিল্ড: <code>id</code> , <code>name</code> , <code>code</code> , <code>half_day_allowed</code> , <code>lwp_allowed</code> , <code>available_days</code> , <code>entitled_days</code> , <code>year</code> , <code>balance_provisioned</code> । employee লিংক আবশ্যিক (না থাকলে 403); <code>date</code> <code>Y-m-d</code> , ডিফল্ট <code>today</code>
POST	<code>/preview</code>	ড্রাই-রান — <code>total_days</code> , <code>excluded_dates[]</code> , <code>balance{available_days, after_request_days}</code> , <code>requires_document</code> , <code>document_threshold_days</code> , <code>blockers[]</code> (রিপোর্টেড, throw করে না), <code>warnings[]</code>
POST	<code>/</code>	জমা (201)। <code>reason required</code> , attachment ফাইল <code>pdf/jpg/jpeg/png ≤5120KB</code> (<code>leave_attachments public disk</code>)। ব্লকার <code>error</code> -এ রূপান্তরিত; অন্য <code>approved leave</code> -এর সাথে <code>overlap ⇒ 409</code> । <code>ApprovalGateway</code> (<code>leave-approve</code> , <code>correlation leave_request:{id}</code>)
GET	<code>/</code>	<code>approver + employee_id</code> ছাড়া <code>indexForHr</code> (সব); অন্যরা নিজেরাই — <code>colleague id</code> বিনা <code>attendance.leave-approve ⇒ denial</code> । ফিল্টার: <code>status</code> , <code>employee_id</code> , <code>policy_id</code> , <code>from</code> , <code>to</code>
GET	<code>/ {id}</code>	<code>approver</code> সব দেখে; অন্যরা শুধুমাত্র নিজস্ব
PATCH	<code>/ {id} /cancel</code>	নিজস্ব only; linked approval request বাতিল করে (রেসপন্সে উল্লেখ)
POST	<code>/ {id} /approve</code>	অ্যাকশনের ভেতরের চেক <code>attendance.leave-approve</code> ; optional <code>comment / reason</code>
POST	<code>/ {id} /reject</code>	একই পারমিশন; non-empty <code>reason</code> ছাড়া 422

- **Approve effects (verified `LeaveRequestExecutor`):** row লক; pending-only (409 "Request already finalized"); অন্য approved request-এর সাথে `overlap ⇒ 409`; weekends/holidays বাদে working day-এ day value গণনা (resolved shift+holiday calendar); সেই দিনগুলিকে leave type-এর `attendance_records` হিসেবে `policy_snapshot {leave_request_id}` সহ stamp করে (spec §5.3: executor সেবামতে balance হ্রাস + consumption ledger রো লিখে) ; `Σday-values == total_days` যাচাই (না হলে 422) ; `approved/decided_by/at` সেট; `LeaveDecided` ইভেন্ট ফায়ার°
- **Validation inputs:** preview/apply শেয়ার করে `leave_policy_id int`, `start_date / end_date Y-m-d` (`end ≥ start`), `duration_type ∈ { full_day , half_day_first , half_day_second }`।
- **Data entities:** `leave_requests` , `leave_request_days` (প্রতি working day, `day_value 1.00/0.50`, void সহায়তা)।

8.১৫ Attendance Correction Requests

- **ব্যবসা উদ্দেশ্য:** কর্মচারী একদিন প্রতিবাদ করতে পারে — missing check-in/out, incorrect time, wrong status, other — evidence সহ; HR সিদ্ধান্ত; অনুমোদন ঐতিহ্য ধ্বংস না করে দিনটি পুনর্লিখন করে।
- **Endpoints:**

পদ্ধতি	Path	Permission	নোট	
POST	<code>/correction-requests</code>	correction-create	201; duplicate pending same date ⇒ 409	
POST	<code>/correction-requests/{id}/cancel</code>	correction-create	Policy-authorised cancel	
GET	<code>/correction-requests</code>	correction-create \	correction-approve	query <code>status validated</code> (invalid ⇒ 422), <code>employee_id</code> ডিফল্ট <code>me</code> , <code>from</code> , <code>to</code>
GET	<code>/correction-requests/{id}</code>	correction-create \	correction-approve	<code>approval_request_uuid/status</code> + <code>current_day snapshot</code> (record metrics + সব

পদ্ধতি	Path	Permission	নোট
			punch, superseded সহ) — approver-এর আলাদা record permission লাগে না
POST	/correction-requests/{id}/approve	correction-approve	সরাসরি fallback যখন কোনো প্ল্যাটফর্ম approval request না পেছনে (যেমন submission-এ approval বন্ধ)
POST	/correction-requests/{id}/reject	correction-approve	Inline validation: reason required string ≤500

- **Submission validation (verified StoreCorrectionRequestRequest):** attendance_date required date; request_type ∈ {missing_in, missing_out, incorrect_time, wrong_status, other}; requested_check_in required (missing_in/incorrect_time, Y-m-d H:i:s); requested_check_out required (missing_out/incorrect_time); attendance_type_override_id required (wrong_status, ∃ attendance_types); reason ≤5000 required; attachment optional file, mimes config থেকে employee.documents.allowed_mimes (ডিফল্ট pdf,jpg,jpeg,png,doc,docx), size cap employee.documents.max_file_size_kb (ডিফল্ট 5120)।
- **Service guards (verified CorrectionRequestService):** নিজস্ব row only (employee_id মিলয়ন না হলে denial); correction window (spec: attendance.correction_window_days, ডিফল্ট 30); একই তারিখে ২য় pending নেই; requested times shift window-এর মধ্যে আস্তে হয়; month_locked রোয়ে রেকর্ড করা হয় এবং এটি ওভাররাইট করতে attendance.correction-override-lock দরকার (executor constant + spec)।
- **Approve effects (verified AttendanceCorrectionExecutor):** source=manual এবং correction_request_id - যুক্ত replacement punch যোগ করে, superseded করা originals-কে superseded_by_id Stamp করে (originals কখনো mutate/delete হয় না), ফলাফল sequence যাচাই (একই ধরনের দুটি ক্রমিক ⇒ 422), snapshot বজায়ে day-এর পুনরায় হিসাব; wrong_status ওয় attendance_type_id -কে ওভাররাইট করে audit log; reject কোনো thing বদলায় না, কেবল status/reason। CorrectionDecided ইভেন্ট (ইম্পোর্টেড)।
- **Data entity:** correction_requests (অন্তর্ভুক্ত attendance_type_override_id, migration 2026_08_20)।

8.১৬ Monthly Attendance Approval (মাস বন্ধ → payroll-ready)

- **ব্যবসা উদ্দেশ্য:** একজন/মাস ভিতরে audited সংযোজন সংখ্যা বের করে, অসমস্যা দেখায়, অনুমোদন (optional override), ভিতরের দিনগুলো লক করে এবং ready_for_payroll -এ চিহ্নিত করে। unlock Finance দ্বারা ফ্রিজ না হওয়া পর্যন্ত বিপরীত কাজ করে।
- **Endpoints:**

পদ্ধতি	Path	Permission	নোট
POST	/monthly-attendance/build	monthly-approve	employee_id -সহ: সিক্স (201)। ছাড়া: bulk queued (202); ফাঁকা নির্বাচন ⇒ "No employees found in the specified criteria."। approved মাস পুনর্গঠন ⇒ 409
GET	/monthly-attendance	monthly-view	ফিল্টার: month, year, status, ready_for_payroll, department_id, search, per_page
GET	/monthly-attendance/{id}	monthly-view	বিস্তারিত
GET	/monthly-attendance/{id}/breakdown	monthly-view	দৈনিক রেকর্ড (ফিল্টার status, per_page; user-scoped)
GET	/monthly-attendance/batch/{batchId}	monthly-view	cache ভিত্তিক ব্যাচ স্ট্যাটাস (TTL 24h; 404)

পদ্ধতি	Path	Permission	নোট
POST	/monthly-attendance/{id}/approve	monthly-approve	body override nullable bool already-approved ⇒ 409 unresolved without override ⇒ 409 with issue list override activity-log-এ (monthly_attendance.approved_with_override)
POST	/monthly-attendance/bulk-approve	monthly-approve	ids required (∃ monthly_attendance_approvals); chunks of 50 queued; 202 { mode: "queued" , batch_id , ... }
POST	/monthly-attendance/{id}/unlock	monthly-unlock	reason required ≤255 frozen month ⇒ 409 "Cannot unlock a frozen month." is_locked/ready_for_payroll রিসেট; reason সংরক্ষণ; status pending; দিনগুলো আনলক

- **Build totals (verified):** present/absent/leave/unpaid-leave/half days, late count, overtime & working hours সমষ্ট। unresolved_flag হয় যদি: missing check-out দিন পাওয়া যায়, elapsed days < records (unassigned/missing data), **pending correction requests, pending leave requests** — স্ট্রিং getUnresolvedIssuesList -থেকে।
- **Approve effects (verified gateway + executor):** ApprovalGateway (monthly-approve , entity monthly_attendance_approvals , correlation monthly_attendance:{id}) ; executor সেট status=approved , approved_by/at , is_locked=true , ready_for_payroll=true এবং সেই employee-month-এর সব attendance_records লক করে ; event **MonthApproved** ফায়া🔥। spec §5.3 নিশ্চিত করে এই transitions executor-এরই মাধ্যমে।
- **Data entity:** monthly_attendance_approvals (unique per company+employee+month+year; frozen_at/frozen_by/unfreeze_reason) °

৫। ব্যাকগ্রাউন্ড জব ও schedul কনসোল কমান্ড

Queue jobs: BuildBulkMonthlyAttendanceJob , ProcessBulkMonthlyApproveJob , ProcessBulkAssignmentJob , ProcessCloseDayChunkJob , ProcessLeaveAccrualChunkJob , ProcessCarryForwardChunkJob , ProcessPolicyGroupBackfillChunkJob , QueueAttendanceExportJob ।

Console schedulers: ScheduleCloseDayCommand (hourly; 9.1-BE — local hour == 2-এ কোম্পানির ব্যাট চালু) , ScheduleLeaveAccrualCommand , ScheduleLeaveCarryForwardCommand (ম্যানুয়াল ট্রিগারের সাথে জোড়ি দেয় — 9.2-BE) , ScheduleEligibilitySweepCommand (policy-group auto-assignment-এর জন্য সাম্প্রত সংযুক্তি ঘোষণা)।

৬। ডোমেইন ইভেন্ট

PunchCreated (PunchService-এ ডিসপ্যাচ) • AssignmentChanged (AssignmentService-এ ইম্পোর্টেড) • AttendanceCalculated • LeaveDecided (LeaveRequestExecutor-এ ডিসপ্যাচ) • LeaveDayVoided • CorrectionDecided (AttendanceCorrectionExecutor-এ ইম্পোর্টেড) • MonthApproved (MonthlyAttendanceApprovalService-এ ডিসপ্যাচ)।

৭। মডিউলের ডেটাবেস এন্টিটি

Migrations (verified filenames): shifts , attendance_types , attendance_policies , assignments (+ scope-normalization & identity-constraint, policy_group_id addition), attendance_punches , leave_balances , leave_balance_ledger , attendance_records , correction_requests (+ attendance_type_override_id migration), leave_requests , leave_request_days , monthly_attendance_approvals , attendance_policy_groups , attendance_policy_group_policy ।

Seeders: AttendanceDatabaseSeeder , AttendanceTypeSeeder , ShiftSeeder ।

৮। অন্য মডিউলের সাথে ইন্টিগ্রেশন

মডিউল / প্ল্যাটফর্ম সেবা	প্রমাণ
Employee	punches/records/corrections/leave/monthly-এ EmployeePersonalInfo রিলেশন; exists:employee_personal_infos,id validation; close-day & bulk-build employee সিলেকশন রিপোজিটরি; user->employeeProfile স্ব-সেবা
Configuration	AssignmentService::scopeModelMap() -এ Branch/Division/Department/Section/Team; monthly build validation-এ departments existence check
Payroll	Modules\Payroll\Support\PayrollPeriodGuard close-day গেট; ready_for_payroll + frozen_at hand-off (\$৯)
Platform Tenancy	TenantContext স্কোপিং; X-Company-Id হেডার
Platform Permissions	permission: middleware + PermissionEngineContract point-checks (punch-others, record-view-all, leave-approve, leave-balance-view, record-recalculate, override-lock)
Platform Approval Gateway/Engine	leave-approve, correction-approve, monthly-approve সাবমিশন correlationId -সহ; executors entity type ভিত্তিক (leave_request , attendance_correction , monthly_attendance)
Activity Log	health view, unresolved-month override, correction lock-override & wrong-status override
Excel (Maatwebsite)	attendance এক্সপোর্ট স্ট্রিমিং + queued export

৯। পে-রোল হ্যান্ড-অফ (spec + build sequence থেকে, কোড-ভিত্তিক যেখানে সম্ভব)

- অনুমোদিত মাস ready_for_payroll = true সেট করে; Finance-এর **freeze/unfreeze Payroll** মডিউলে থাকে (payroll.month-freeze , attendance.monthly-approve -এর সাথে segregation-of-duties)। Freeze requires approved; unfreeze requires কারণ এবং paid run-এর জন্য payroll.month-unfreeze-paid ।
- **attendance_snapshots** (কার্ড 8.0-BE, সম্পন্ন): immutable, timestamp-less, unique(payroll_run_id, employee_id) ; frozen monthly approval থেকে verbatim copy (কোনো recalculation নেই); idempotent rebuild একটি **Payroll** মডিউলের মধ্যে ৪টি endpoint আছে (build draft-only payroll.run-create , list, show, diagnostic divergence) — অর্থাৎ Attendance রুট ফাইলের অংশ নয়।
- Payslip AttendanceSnapshotServiceInterface::getPayrollAttendanceForRun() দিয়েই consumption করা আবশ্যিক; guard test প্রমাণ করে attendance_records ও monthly_attendance_approvals payslip path-এ কোয়েরি হয় না। Payslip generation (8.2a-BE) এখনও কোডে নেই (pending card)।

১০। এরর আচরণ চিপ-শিট (verified codes/messages)

পরিস্থিতি	ফলাফল
কোম্পানি কনটেক্সট অনুপস্থিত	PermissionDeniedException('company.context')
unassigned তারিখে punch	422, code: unassigned
একই ধরনের দ্বিতীয় ক্রমিক punch	422 (শেষ punch সময় বার্তায়)
Records list-এর range >366 দিন	422 "Date range cannot exceed 366 days."
অবৈধ correction/leave status filter	422

পরিস্থিতি	ফলাফল
একই তারিখে দ্বিতীয় pending correction	409 DuplicateCorrectionRequestException
approved month approve/rebuild/unlock frozen	409
unresolved month override ছাড়া approve	409 with issue list
allow_negative ছাড়া negative ব্যালেন্স সমন্বয়	DomainException "Adjustment rejected: ..."
assignment create/update-এ overlap	AssignmentOverlapException
locked record recalculation / unauthorized live override	AttendanceRecordLockedException , RecalculationUnauthorizedException

এর বাইরের যে কোনো thing (exact transport envelope, rate limits, অতিরিক্ত HTTP codes) — **Not specified in the reviewed source files**।

১১। এই গাইড ইন্টেনশনালি কভার করে না

- Front-end স্ক্রিন/UI flow (কেবলমাত্র backend routes, services, jobs, events রিভিউ করা হয়েছে)
- Role seeding mechanics (৯২ উল্লেখের বাদে)
- Exact মান: bulk-assignment sync/queue threshold, base controller-এর pagination envelope internals, queue connection names

এদের জন্য: **Not specified in the reviewed source files**।

আপেন্ডিক্স এ — রিভিউড সূত্র

1. backend/Modules/Attendance/routes/api.php (২৯৯১ লাইন, সম্পূর্ণ)
2. Controllers (১৭): AttendanceController, PunchController, AttendanceRecordController, AttendancePolicyController, AttendanceTypeController, Shift/ShiftManagementController, AssignmentController, BulkAssignmentController, AssignmentResolutionController, AttendanceRecalculationController, PolicyGroupController, LeaveBalanceController, LeaveBalanceJobController, CloseDayJobController, LeaveRequest/LeaveRequestManagementController, CorrectionRequestController, MonthlyAttendanceApprovalController
3. Form Requests (৪৫টি ফাইল — সব validation আলাদাভাবে উল্লিখিত)
4. Models: Assignment, AttendancePolicy, AttendancePolicyGroup, AttendancePunch, AttendanceRecord, AttendanceType, CorrectionRequest, LeaveBalance, LeaveBalanceLedger, LeaveRequest, LeaveRequestDay, MonthlyAttendanceApproval, Shift
5. Service contracts (২২টি) এবং implementations: PunchService, AttendanceRecordService, AttendanceRecalculationService, AssignmentService, AssignmentResolutionService, CorrectionRequestService, LeaveRequestService, LeaveBalanceService, MonthlyAttendanceApprovalService, CloseDayService, AttendanceTypeService, AttendancePolicyGroupService, PolicyGroupEligibilityEvaluator, BulkAssignmentService
6. Approval executors: MonthlyAttendanceExecutor, AttendanceCorrectionExecutor, LeaveRequestExecutor
7. Support: EligibilityAttributeRegistry, GenderNormalizer, BatchProgressStore; DTOs (ResolvedContext, EmployeeEligibilityContext, ResolvedPolicyAssignment, LeaveEvaluation, AutoAssignmentOutcome, PolicySnapshotData)

8. Jobs (৮), Events (৭), Resources (১৪), Exports, Console schedulers (৪), Providers (RouteServiceProvider), Migrations (১৭), Seeders (৩)

9. রেফারেন্স ডক: ATTENDANCE_PAYROLL_MODULE_SPEC.md (\$0, \$3, \$4, \$5, \$6, Parts D-F) এবং ATTENDANCE_PAYROLL_BUILD_SEQUENCE.md

10. স্টাইল রেফারেন্স: EMPLOYEE_MODULE_GUIDE.md / EMPLOYEE_MODULE_GUIDE_BN.md