

# Payroll documentation

In this app, the time format that we use unix epoch in **milliseconds**. Please make sure that the date you send to API is in milliseconds.

For simplicity, basic authentication is used.

- Create an endpoint where admin can add attendance period start & end date for particular payroll.

Clients can use [POST v1/admin/payrolls](#) to do this. The rule is start & end can't overlap with another payroll range. Because the same attendance, reimbursement and overtime should not be paid more than one.

For example:

Payroll-a -> star = 10, end = 15

Payroll-b -> star = 12, end = 20

Payroll-b will be rejected.

- Create an endpoint where employees can submit their own attendance.

Clients can use [POST v1/client/attendances](#) to record employees attendance.

**“No rules for late or early check-ins or check-outs”** This line lacks clarity.

So I made the client check-in for today and check-out tomorrow. That's why there is a **date** field on the request body.

- Create an endpoint where employees can submit overtime.

Clients can use [POST v1/client/overtimes](#) to propose overtime.

**“Overtime must be proposed after they are done working.”** This line lacks clarity.

What is the definition of **“done working”** here?

I assume that “done working” means they have already performed check-out in [POST v1/client/attendances](#) endpoint.

- Create an endpoint where employees can submit reimbursements

Clients can use [POST v1/client/reimbursements](#) to submit employee's reimbursements.

- Create an endpoint where the admin can run payroll (process payments to employees). Clients can use [POST v1/admin/payrolls/{payrollId}/payments](#), once the payroll finishes the run, the payroll for that period will be locked. Employees who try to submit the attendance, overtime and reimbursement on that payroll period will be rejected.

**Attendance will be paid based** on the number of attendances. Because this line “**No rules for late or early check-ins or check-outs**” lacks clarity. As long as the employee performs check-in even though not performing check-out, this will be counted as a full day (8 hours).

**Overtime will be paid based** on overtime date, for example:

The current payroll-period is 20 until 30. Employee submits overtime for 31, this overtime will not be paid on current payroll since the current payroll end date is 30.

Assuming the salary is monthly, which is 160 hours (assuming a month is 4 weeks).

So calculation for overtime will be:

amount = duration \* 2 \* (salary/160)

**Reimbursement will be paid** based on when the employees submit the reimbursement. The submission date is automatically recorded when reimbursement is successfully made.

- Create an endpoint where employees can generate a payslip. Clients can use [GET v1/client/payroll/{payrollId}](#) to fetch employee payslip.
- Create an endpoint where the admin can generate a summary of all employee payslips. Clients can use [GET v1/admin/payroll/{payrollId}](#) to fetch summary of payments.

# Proposed Database ERD

