Mopuan KPI

Integration guide for the KPI services

opsenseapi · Version 1.0 · 9 November 2026

1. What the KPI feature is

KPI gives every employee a score for a period of time. These services let the Mopuan app show those scores and, for employees with the right permission, change the KPI setup.

- **Score:** a number from 0 to 100 per employee, with a tier (`elite`, `gold`, `silver`, `bronze`) and a confidence value.
- **Role:** every employee is scored against the metrics of their OpSense role. The dashboard ranks employees company-wide and inside each role.
- **Metrics:** the things that are measured, for example completion rate or delivered count. Each metric has a weight and a target.
- **Levels:** for each role, a metric is switched on and set to `low`, `med` or `high`. The level decides the metric's target.
- **Manual entries:** weekly values typed in by hand for metrics that the system cannot measure.
- **Score adjustments:** points (-10 to +10) that a manager adds to or removes from one employee's score, with a reason and a date range.
- **Exclusions:** employees or roles that are left out of KPI scoring.

There are ten services. Four read data and six change it.

Service	Use it to	Needs
`SRVGetKpiDashboard`	Get the ranked scores of the employees the actor may see.	view
`SRVGetKpiEmployeeDetail`	Get one employee's scorecard and activity.	view
`SRVGetKpiSettings`	Get the level setup of a role and the exclusions.	view and edit
`SRVGetKpiManualEntries`	Get the manual entries of one week.	view and edit
`SRVSaveKpiRoleLevels`	Set the metric levels of a role.	edit
`SRVApplyKpiPreset`	Apply a ready-made level preset to a role.	edit
`SRVSaveKpiManualEntries`	Save or clear the manual entries of one week.	edit
`SRVSaveKpiExclusions`	Set which employees and roles are excluded.	edit
`SRVSaveKpiScoreAdjustment`	Add a score adjustment to an employee.	edit
`SRVRevokeKpiScoreAdjustment`	Revoke a score adjustment.	edit

2. Authentication and permissions

Every KPI request needs two fields:

- `api_key`: the company's key. It decides which company the call is for.
- `actor_mopuan_user_id`: the Mopuan id of the employee who is using the app.

> **Both fields must come from your server.** Your server fills `actor_mopuan_user_id` with the employee it has authenticated. Never take this value from the phone, and never put the `api_key` in the app. A value sent by the phone can be changed by anyone.

The actor must be an active employee of the company that the `api_key` belongs to. OpSense finds the employee by this id and uses their OpSense role.

What the employee may do follows the permissions of their role in OpSense, for the KPI module (`kpi-dashboard`):

- **Read the dashboard or one employee:** the role can view the KPI module.
- **Read settings or manual entries:** the role can view and edit the KPI module.
- **Change anything:** the role can edit the KPI module.

The employee also sees only the employees that their hierarchy level allows (for example a supervisor sees their own team). Lists and scores are already filtered for you. Get the permissions with `SRVGetUserRoles` (see the Mopuan Roles guide) and hide screens the employee cannot use.

Situation	Response
`api_key` missing	`error_code` 10015
`api_key` invalid or inactive	`error_code` 10016
`actor_mopuan_user_id` missing, not a number, or not an active employee of this company	`error_code` 10030
The employee's role lacks the permission	`error_code` 10031

3. Requests and responses

```
POST https://<api-host>/v1/<service>
Content-Type: application/x-www-form-urlencoded
```

Send the parameters as **form fields**. A JSON request body is not read.

Some fields are nested. Write the nesting in the field name, with square brackets. In curl, use `--data-urlencode` or encode the brackets:

```
metric[work_orders|completion_rate][enabled]=1
metric[work_orders|completion_rate][level]=high
value[12][345]=7.5
note[12][345]=Counted by hand
exclude_user[]=345
```

```
exclude_user[]=346
```

The success body is not wrapped in an extra `result` object. A read returns data:

```
{"result": "success", "data": { ... }}
```

A change returns `result` and a few fields about what was saved:

```
{"result": "success", "user_id": 345}
```

An error returns:

```
{"error_code": "10031", "error_text": "You don't have permission for this action."}
```

`error_code` is a string. Use it in your code. `error_text` is for logs and is in English.

4. Dates, time zone and ranges

- Send dates as `YYYY-MM-DD`.
- "Today" and the default period follow the **company's time zone**.
- For the dashboard and the employee detail, if `start_date` or `end_date` is missing, is not a valid date, or the end is before the start, the period is the **current month** of the company. The response returns the dates that were used in `start_date` and `end_date`.
- The range may be at most **366 days**. A longer range returns `error_code` 10032.
- A week always starts on **Monday**. If you send any date as `week`, it is moved to the Monday of that week. The response returns the Monday.
- When you save an adjustment or manual entries, the defaults for a missing date ("today", "this week") use the **server's time zone (Europe/London)**, not the company's. Always send the date explicitly.

5. Labels are translation keys

The API does not translate. Names of metrics, sources and tiers come as **translation keys**. The app translates them into the employee's language. If a key is unknown to the app, show the key text as a fallback.

Key family	Meaning
`kpi_source_*`	A source of metrics, for example work orders, deliveries, rota, attendance.
`kpi_metric_*`	A metric name.
`kpi_metric_*_hint`	A short explanation of a metric.
`kpi_scope_*`	A source setting and its hint (`..._hint`).
`kpi_tier_*`	A tier name (`kpi_tier_elite`, `kpi_tier_gold`, `kpi_tier_silver`, `kpi_tier_bronze`). Tier codes in `tier` are `elite`, `gold`, `silver`, `bronze`.

`kpi_role_unassigned`	Shown as `role_name` for an employee with no role.
`wtr_*`	Work order labels, for example `wtr_unassigned`.

Raw labels. Manual metrics are named by the company, in plain text. For them `raw_label` is `true` and `label` is the text to show as it is. The same applies to `raw_hint` and `hint`. When `raw_label` is `false` or missing, translate `label`. Role names (`role_name`, `name` in `roles`) and employee names are plain text and are not translated. The `label` of a preset is also plain text from the company.

6. Services

SRVGetKpiDashboard

Returns the scores of every employee the actor may see, the ranking, the roles and a summary. Needs **view**.

Field	Required	Notes
`api_key`	**Required**	See Authentication.
`actor_mopuan_user_id`	**Required**	Your Mopuan employee id of the signed-in employee. Set by your server. See Authentication.
`start_date`	Optional	Start of the period, `YYYY-MM-DD`. See section 4.
`end_date`	Optional	End of the period, `YYYY-MM-DD`.
`role_id`	Optional	Select one role. The response then adds `targets` and `sources` for that role, and the summary covers that role only. An unknown role id is treated as no role (0).

```
curl -X POST "https://<api-host>/v1/SRVGetKpiDashboard" \
 -d "api_key=<api_key>" \
 -d "actor_mopuan_user_id=345" \
 -d "start_date=2026-11-01" \
 -d "end_date=2026-11-30"
```

```
{
 "result": "success",
 "data": {
   "start_date": "2026-11-01",
   "end_date": "2026-11-30",
   "role_id": 0,
   "can_edit": true,
   "users": {
     "345": {
       "name": "Ayse Demir", "photo": "https://.../uploads/users/ayse.jpg",
       "role_id": 5, "role_name": "Housekeeper",
       "score": 82, "raw_score": 80, "score_adjustment": 2,
       "role_score": 82, "confidence": 74, "has_data": true,
       "scored_by_role": true, "role_rank": 1, "role_size": 6,
       "tier": {"tier": "gold", "tier_label": "kpi_tier_gold"},
       "total": 41, "completed": 38, "rate": 92.7, "avg_minutes": 48,
```

```
      "components": [
        {"source_key": "work_orders", "metric_key": "completion_rate",
         "label": "kpi_metric_completion_rate", "raw_label": false,
         "hint": "kpi_metric_completion_rate_hint", "raw_hint": false,
         "value": 92.7, "target": 95, "achievement": 97.6,
         "credit": 97.6, "weight": 8}
      ]
    }
  },
  "ranked_user_ids": [345, 346],
  "roles": {
    "5": {"id": 5, "name": "Housekeeper", "size": 6, "avg_score": 71,
          "best_user_id": 345, "configured": true, "user_ids": [345, 346]}
  },
  "summary": {"headcount": 8, "with_data": 7, "avg_score": 69,
              "avg_confidence": 70, "total": 210, "completed": 188,
              "completion_rate": 89.5, "avg_minutes": 52, "is_work_orders": true}
 }
}
```


Field	Notes
`can_edit`	`true` if the actor can edit the KPI module. Use it to show or hide edit buttons.
`users`	Object keyed by employee id (OpSense user id). Only employees the actor may see.
`score`	The score to show: `raw_score` plus `score_adjustment`.
`role_score`	The score from the role's own metrics, or `null` if the role has none.
`confidence`, `has_data`	How much measured data the score rests on, and whether there is any.
`role_rank`, `role_size`	Position inside the role and the number of employees in it.
`photo`	A URL built from the employee's photo file. It is empty if there is no photo. The file itself is not checked, so be ready for a failed image load.
`components`	The metrics behind the score. Empty if the employee is not scored by role.
`ranked_user_ids`	Employee ids in ranking order, best first, for all employees the actor may see.
`roles`	Object keyed by role id. `user_ids` are in ranking order. `configured` is `true` if the role has KPI metrics set up.
`summary`	Totals for the selected role, or for all visible employees if `role_id` is 0. `is_work_orders` tells whether the figures are work order figures.

Errors: 10032 (range longer than 366 days). See section 7 for the shared errors.

SRVGetKpiEmployeeDetail

Returns the scorecard and activity of one employee. Needs **view**.

Field	Required	Notes
`api_key`	**Required**	See Authentication.
`actor_mopuan_user_id`	**Required**	Your Mopuan employee id of the signed-in employee. Set by your server. See Authentication.
`user_id`	Optional	OpSense id of the employee. If missing, the first employee the actor may see is returned.
`start_date`	Optional	See section 4.
`end_date`	Optional	See section 4.
`role_id`	Optional	Echoed back in `role_id`. It lets the app return to the role tab it came from.

```
curl -X POST "https://<api-host>/v1/SRVGetKpiEmployeeDetail" \
 -d "api_key=<api_key>" \
 -d "actor_mopuan_user_id=345" \
 -d "user_id=345" \
 -d "start_date=2026-11-01" \
 -d "end_date=2026-11-30"
```

```
{
 "result": "success",
 "data": {
   "start_date": "2026-11-01", "end_date": "2026-11-30", "role_id": 0,
   "eligible": true,
   "user": { "name": "Ayse Demir", "score": 82, "raw_score": 80,
             "score_adjustment": 2, "rank": 3, "role_rank": 1,
             "tier": {"tier": "gold", "tier_label": "kpi_tier_gold"},
             "components": [ ... ] },
   "scorecard": {
     "earned": 80.4,
     "components": [
       {"source_key": "work_orders", "metric_key": "completion_rate",
        "label": "kpi_metric_completion_rate", "raw_label": false,
        "weight": 8, "point_max": 17.8, "achieved": 17.4,
        "value": "92.7%", "target": "95%", "monthly_target": null}
     ],
     "info_metrics": [
       {"label": "kpi_metric_wo_volume", "raw_label": false, "hint": "...",
        "raw_hint": false, "value": "41", "scored": false, "missing": false}
     ],
     "unavailable_labels": []
   },
   "work_orders": {"stats": {"total": 41, "completed": 38}, "completion_rate": 92.7,
                   "avg_minutes": 48, "recent_orders": [ ... ]},
   "attendance": {"available": true, "onsite_seconds": 518400, "by_date": { ... }},
   "rota": {"days": [ ... ], "worked_days": 18, "total_hours": 144},
   "manual_entries": [ ... ], "manual_available": true,
   "adjustment_history": [
     {"id": 9, "adjustment_points": "2.00", "effective_date": "2026-11-03",
      "effective_end": null, "reason": "Covered extra shifts",
      "status": "active", "added_by_name": "Mehmet Kaya"}
   ]
 }
```

```
}
```

- `eligible` is `false` and `user` is `null` when the employee is not in the list the actor may see, is excluded from KPI, or does not exist in the company. Show a "not available" message. This is not an error.
- `scorecard.components` shows each scored metric as points: `achieved` out of `point_max`. `value`, `target` and `monthly_target` are already formatted text. `info_metrics` are measured metrics. `scored` tells if they count in the score.
- `scorecard.unavailable_labels` lists the sources the role needs but the company cannot use now. Mention them, because the score then rests on less data.
- `adjustment_history` is filled only if the actor can edit the KPI module. For other employees it is an empty list.
- `attendance.available` is `false` when no attendance data could be read. The rest of the page still works.

Errors: 10032 (range longer than 366 days).

SRVGetKpiSettings

Returns the KPI setup of one role: which metrics are on, their levels, ready-made presets, and the exclusions. Needs **view and edit**.

Field	Required	Notes
`api_key`	**Required**	See Authentication.
`actor_mopuan_user_id`	**Required**	Your Mopuan employee id of the signed-in employee. Set by your server. See Authentication.
`role_id`	Optional	The role to show. If missing or unknown, the first role in the list is used. The response returns the role that was used.
`tab`	Optional	`levels` or `exclusions`. Any other value becomes `levels`. It only selects the tab that is echoed back. All data is returned either way.

```
curl -X POST "https://<api-host>/v1/SRVGetKpiSettings" \
 -d "api_key=<api_key>" \
 -d "actor_mopuan_user_id=345" \
 -d "role_id=5" \
 -d "tab=levels"
```

```
{
 "result": "success",
 "data": {
   "role_id": 5, "tab": "levels",
   "levels_ready": true, "exclusions_ready": true,
   "roles": {"5": "Housekeeper"},
   "available_sources": { ... }, "role_sources": { ... },
   "presets": {"standard": {"preset_key": "standard", "label": "Standard",
                            "is_global": true, "definition": { ... }}},
   "role_rows": [ ... ], "level_defs": [ ... ], "level_values": { ... },
   "enabled_count": 4,
   "candidates_by_role": {"5": {"name": "Housekeeper", "users": { ... }}},
   "exclusions": {"users": { ... }, "roles": { ... }},
   "role_user_counts": {"5": 6}, "excluded_count": 1
```

```
 }
}
```

- `levels_ready` and `exclusions_ready` are `false` if that part of KPI is not set up for the company. Then do not offer saving it: the matching save returns 10034.
- `presets` is keyed by `preset_key`. Use these keys for `SRVApplyKpiPreset`.
- `candidates_by_role` and `exclusions` hold the employees and roles that can be excluded, and who is excluded now. Use them to draw the exclusions tab.
- `role_rows`, `level_defs` and `level_values` describe the metrics and their levels. Metric keys are `<source_key>|<metric_key>`, the same keys that `SRVSaveKpiRoleLevels` takes. Names inside them are translation keys.

SRVGetKpiManualEntries

Returns the manual metrics, the employees, and the values already entered for one week. Needs **view and edit**.

Field	Required	Notes
`api_key`	**Required**	See Authentication.
`actor_mopuan_user_id`	**Required**	Your Mopuan employee id of the signed-in employee. Set by your server. See Authentication.
`week`	Optional	Any date `YYYY-MM-DD`. It is moved to the Monday of that week. If missing, the current week of the company is used.

```
curl -X POST "https://<api-host>/v1/SRVGetKpiManualEntries" \
 -d "api_key=<api_key>" \
 -d "actor_mopuan_user_id=345" \
 -d "week=2026-11-02"
```

```
{
 "result": "success",
 "data": {
   "manual_ready": true,
   "metrics": {"12": {"label": "Room inspection score", "raw_label": true,
                      "unit": "score", "direction": "up", "default_target": 90}},
   "week_start": "2026-11-02", "week_end": "2026-11-08",
   "prev_week": "2026-10-26", "next_week": "2026-11-09",
   "this_week": "2026-11-09", "can_go_next": true,
   "employees": {"345": {"name": "Ayse Demir", "role_id": 5, "role_name": "Housekeeper"}},
   "existing": {"12|345": {"metric_id": 12, "user_id": 345, "value": "88.00",
                           "note": "Counted by hand", "author": "Mehmet Kaya"}},
   "filled_count": 1, "expected_count": 6
 }
}
```

- `metrics` is keyed by metric id. `existing` is keyed by `<metric_id>|<user_id>`.
- `manual_ready` is `false` if manual metrics are not set up. Then `metrics` is empty and saving returns 10034.
- Use `prev_week`, `next_week` and `can_go_next` for paging. A future week cannot be saved.

SRVSaveKpiRoleLevels

Sets the level of each metric of a role. Needs **edit**.

Field	Required	Notes
`api_key`	**Required**	See Authentication.
`actor_mopuan_user_id`	**Required**	Your Mopuan employee id of the signed-in employee. Set by your server. See Authentication.
`role_id`	**Required**	The role to change.
`metric[<source_key>|<metric_key>][enabled]`	Optional	Send any non-empty value (for example `1`) to switch the metric on. Metrics you do not send as enabled are not switched on.
`metric[<source_key>|<metric_key>][level]`	Optional	`low`, `med` or `high`. Required for every metric that is on.

```
curl -X POST "https://<api-host>/v1/SRVSaveKpiRoleLevels" \
 -d "api_key=<api_key>" \
 -d "actor_mopuan_user_id=345" \
 -d "role_id=5" \
 -d "metric[work_orders|completion_rate][enabled]=1" \
 -d "metric[work_orders|completion_rate][level]=high"
```

```
{"result": "success", "role_id": 5}
```

Errors: 10032 (role not found), 10033 (a metric that is on has a level other than `low`, `med`, `high`), 10034 (levels not set up), 10035 (another KPI change is being saved: try again in a moment), 10036 (a metric that is on has no target at the chosen level; nothing is saved).

SRVApplyKpiPreset

Applies a ready-made preset to a role. It sets which metrics are on and their levels. Needs **edit**.

Field	Required	Notes
`api_key`	**Required**	See Authentication.
`actor_mopuan_user_id`	**Required**	Your Mopuan employee id of the signed-in employee. Set by your server. See Authentication.
`role_id`	**Required**	The role to change.
`preset_key`	**Required**	A key from `presets` in `SRVGetKpiSettings`.

```
curl -X POST "https://<api-host>/v1/SRVApplyKpiPreset" \
 -d "api_key=<api_key>" \
 -d "actor_mopuan_user_id=345" \
 -d "role_id=5" \
 -d "preset_key=standard"
```

```
{"result": "success", "role_id": 5}
```

Errors: 10032 (role not found), 10034 (levels not set up), 10035 (busy, try again), 10037 (preset not found). Reload `SRVGetKpiSettings` after a success.

SRVSaveKpiManualEntries

Saves or clears manual values for one week. Needs **edit**.

Field	Required	Notes
`api_key`	**Required**	See Authentication.
`actor_mopuan_user_id`	**Required**	Your Mopuan employee id of the signed-in employee. Set by your server. See Authentication.
`week`	Optional	Any date in the week, `YYYY-MM-DD`. Moved to Monday. A week after the current week is refused.
`value[<metric_id>][<user_id>]`	Optional	A number. An empty or non-numeric value **clears** the saved entry.
`note[<metric_id>][<user_id>]`	Optional	A note, up to 255 characters.

```
curl -X POST "https://<api-host>/v1/SRVSaveKpiManualEntries" \
 -d "api_key=<api_key>" \
 -d "actor_mopuan_user_id=345" \
 -d "week=2026-11-02" \
 -d "value[12][345]=88" \
 -d "note[12][345]=Counted by hand"
```

```
{"result": "success", "week": "2026-11-02", "written": 1, "cleared": 0}
```

- Metrics and employees that are unknown, or that the actor may not see, are skipped without an error.
- `written` is the number of values saved. `cleared` is the number of entries removed.

Errors: 10034 (manual entries not set up), 10038 (week is in the future).

SRVSaveKpiExclusions

Sets which employees and roles are left out of KPI. Needs **edit**.

Field	Required	Notes
`api_key`	**Required**	See Authentication.
`actor_mopuan_user_id`	**Required**	Your Mopuan employee id of the signed-in employee. Set by your server. See Authentication.
`exclude_user[]`	Optional	An employee id to exclude. Repeat the field for more employees.
`exclude_role[]`	Optional	A role id to exclude. Repeat the field for more roles.

```
curl -X POST "https://<api-host>/v1/SRVSaveKpiExclusions" \
 -d "api_key=<api_key>" \
 -d "actor_mopuan_user_id=345" \
 -d "exclude_user[]=345" \
 -d "exclude_user[]=346" \
 -d "exclude_role[]=8"
```

```
{"result": "success"}
```

Send the complete list. The call replaces the exclusions for all employees and roles the actor may see. Anything you do not send is no longer excluded. Send no `exclude_*` fields to clear them all. Ids the actor may not see are ignored.

Errors: 10034 (exclusions not set up).

SRVSaveKpiScoreAdjustment

Adds points to or removes points from one employee's score for a date range. Needs **edit**.

Field	Required	Notes
`api_key`	**Required**	See Authentication.
`actor_mopuan_user_id`	**Required**	Your Mopuan employee id of the signed-in employee. Set by your server. See Authentication.
`user_id`	**Required**	OpSense id of the employee. It must be an employee the actor may see.
`adjustment_points`	**Required**	A number from -10 to +10, not zero. Two decimals are kept. A comma is accepted as the decimal mark.
`reason`	**Required**	Why the adjustment is made. Up to 500 characters.
`effective_date`	Optional	First day, `YYYY-MM-DD`. If missing or not valid, today (server time zone) is used.
`effective_end`	Optional	Last day, `YYYY-MM-DD`. If missing or not valid, the adjustment has no end date.

```
curl -X POST "https://<api-host>/v1/SRVSaveKpiScoreAdjustment" \
 -d "api_key=<api_key>" \
 -d "actor_mopuan_user_id=345" \
 -d "user_id=345" \
 -d "adjustment_points=2" \
 -d "reason=Covered extra shifts" \
 -d "effective_date=2026-11-03" \
 -d "effective_end=2026-11-30"
```

```
{"result": "success", "user_id": 345}
```

Errors: 10032 (employee not found), 10034 (adjustments not set up), 10039 (points are zero or outside -10 to +10), 10040 (reason is empty), 10041 (`effective_end` is before `effective_date`). The new adjustment appears in `adjustment_history` of `SRVGetKpiEmployeeDetail`.

SRVRevokeKpiScoreAdjustment

Revokes an active adjustment. The adjustment stays in the history with the status `revoked`, but no longer counts. Needs **edit**.

Field	Required	Notes
`api_key`	**Required**	See Authentication.
`actor_mopuan_user_id`	**Required**	Your Mopuan employee id of the signed-in employee. Set by your server. See Authentication.
`id`	**Required**	The adjustment id (`id` in `adjustment_history`).
`user_id`	**Required**	The employee the adjustment belongs to.

```
curl -X POST "https://<api-host>/v1/SRVRevokeKpiScoreAdjustment" \
 -d "api_key=<api_key>" \
 -d "actor_mopuan_user_id=345" \
 -d "id=9" \
 -d "user_id=345"
```

```
{"result": "success", "user_id": 345}
```

Errors: 10032 (`id` or `user_id` is missing or the employee is not found), 10034 (adjustments not set up). If the `id` does not match an active adjustment of that employee, nothing changes and the call still returns success. Reload the detail to confirm.

7. Error codes

error_code	Meaning	Retry?
10015	`api_key` missing	No. Fix the request.
10016	`api_key` invalid or inactive	No. Check the key.
10030	Actor not found: `actor_mopuan_user_id` missing, not a number, or not an active employee of the company	No. Fix the id.
10031	The employee's role does not have the KPI permission this service needs	No. Hide the feature.
10032	Role or employee not found or not allowed, or the date range is longer than 366 days	No. Fix the request.
10033	A metric that is on has no valid level (`low`, `med`, `high`)	No.
10034	The needed KPI part (levels, manual entries, exclusions, adjustments) is not set up for the company	No.

10035	Another KPI change is being saved	Yes, after a moment.
10036	A metric that is on has no target at the chosen level; nothing was saved	No. Choose another level.
10037	Preset not found	No.
10038	The manual entry week is in the future	No.
10039	Adjustment points are zero or outside -10 to +10	No.
10040	Adjustment reason is missing	No.
10041	Adjustment end date is before the start date	No.
10099	The request could not be completed on our side	Yes

```
{"error_code": "10039", "error_text": "Adjustment must be between -10 and +10, and cannot be zero."}
```

8. Recommendations

- **Cache dashboards briefly.** Scores are calculated on each call and can take a moment. Keep a dashboard response for one to a few minutes for the same employee, period and role. Clear it after any change you make.
- **Show only what the employee may see.** Read the permissions with `SRVGetUserRoles`. Hide KPI for roles without view permission. Show edit screens only if `can_edit` is `true` or the role can edit. The API still checks every call.
- **Always send the actor from your server.** Check the employee's session on your side first.
- **Send explicit dates.** Always send `start_date`, `end_date`, `week` and `effective_date`, so the time zone cannot change the result.
- **Translate labels in the app,** and print `raw_label` texts as they are.
- **Handle empty data.** An employee with `has_data` `false` has no score yet. Show "no data", not zero.
- **Do not retry changes blindly.** Only retry 10035 and 10099, and reload the data first, because a change may already have been saved.