

Eitje open API

Intro

Eitje is scheduling software and its API offers access to our customers' data on various levels.

Data

An API connection is made with an "organisation", which may have one or more "environments": different venues at different locations. Each "environment" has multiple "teams", such as a team for the kitchen and the bar. A user may belong to many teams and many environments.

Setup

In order to start developing an integration, you need:

- your own credentials as a partner, and test credentials, which you may obtain by contacting support@eitje.app

Authentication

The authentication is basic and is done through the header. It relies on your `Partner-Username` and `Partner-Password` and our shared customer's `Api-Username` and `Api-Password`, which they may find on the settings page in Eitje (if they have the "admin" role). You must supply all four keys in the header separately.

Parameters

You can provide parameters as a JSON body or as query params, both are accepted. Authentication must be done through the header though.

Filters

Currently, the only filter that's available is a date range filter on certain resources (Planning shifts, Time registration shifts, Leave requests, Availability shifts and Events). For all these resources, you must pass a `start_date` and an `end_date`, otherwise you'll get an error. There's a maximum length of this date range per resource to prevent slow requests.

We offer two flavors of date filtering: `resource_date` (default) and `updated`. Resource date looks at the date of the specific resource (eg the day on which an event occurs) while updated looks at when the resource was updated. The limit can differ between those two which will be communicated via the API. Below is an example of a filter. If you omit `date_filter_type`, the default (`resource_date`) is used

For `updated`, you can pass a datetime in the ISO8601 standard. For `resource_date` you should always pass a date.

Note: If you pass a simple `date` for `updated` it's parsed as start-of-day for `start_date` and end-of-day for `end_date`. If you don't want this, pass a datetime instead of date.

Limit defaults:

- updated: 3
- resource_date: 7
- If a resource has a different limit, it will be noted.

ruby

```
{
  "filters": {
    "start_date": "2025-03-01",
    "end_date": "2025-03-04",
    "date_filter_type": "updated"
  }
}
```

More info

- test url: `https://open-api-test.eitje.app/open_api`
- production url: `https://open-api.eitje.app/open_api`
- For question you may also reach us at support@eitje.app
- Missing data? Feel free to contact us for new endpoints and attributes
- [Learn more about Eitje](#)

Availability shifts

Availability shifts represent the availability of team members per day, it's organisation wide.

We have three possible statuses: 'available', 'unavailable' and 'partly unavailable'. The first apply for the full date, but for the latter, a start and end datetime are also included to denote the exact time of the unavailability. Note: the `date` represents the start_date of the shift in case of partly unavailable (end_datetime might be on the next day, eg from 21:00-02:00). In short: use `date` for 'available' and 'unavailable' and start/end datetime for 'partly_unavailable'.

This endpoint only returns records actually created by the end user in Eitje. This means you might not get a record back if the user hasn't filled in availability. In Eitje a customer can choose a default for availability: they can choose between default available or unavailable. If no record is present for a user/date combination, that default is used in Eitje.

GET List availability shifts

`https://open-api.eitje.app/open_api/availability_shifts`

Check ins

Users can check in and check out on a central device at their employer's location. We currently only return the pending check ins (check ins that haven't been checked out yet), if you're interested in data after check out, you can use the time registration shifts endpoint.

GET List check ins

https://open-api.eitje.app/open_api/check_ins

Environments

GET List environments

https://open-api.eitje.app/open_api/environments

Retrieve all environments that belong to the organisation, both the active and inactive environments.

Events

Events appear alongside the schedule in Eitje and are useful to let team members know something special is happening on that day.

Events have a 'kind' property that's an enum, with five possible values, due to legacy reasons these names are still in Dutch. Allowed kinds are `['sport', 'meeting', 'training', 'feest', 'overig']`

GET List events

https://open-api.eitje.app/open_api/events

- See filters above for explanation
 - Limits:
 - updated: 14
 - resource_date: 90
-

POST Create event

https://open-api.eitje.app/open_api/events

Required: title, date, environment_id

Optional: remarks, from, till, kind

Optional: remarks, from, till, kind.

PUT Update event

`https://open-api.eitje.app/open_api/events/{{event_id}}`

Update has the same params as create but nothing is required.

DELETE Delete event

`https://open-api.eitje.app/open_api/events/{{event_id}}`

Leave requests

Users can ask for leave and that turns into a leave requests. A request doesn't have restrictions on how long it can be. The status column shows us if the request has been handled by a manager yet.

Property	Type	Explanation
start_datetime	datetime	start datetime of the request
end_datetime	datetime	end datetime of the request
status	enum	a request starts as 'pending' and then a manager can either approve (the status becomes 'approved') or deny (status becomes 'denied'). Only approved requests should be used for determining availability of users
reason	string	reason for the leave give by the requester
process_reason	string	note provided by the handler while handling it (never present on pending requests)

Filters

- See filters above for explanation
 - Limits:
 - updated: 14
 - resource_date: 90
-

GET List leave requests

https://open-api.eitje.app/open_api/leave_requests

Skill Sets

GET List skill sets

https://open-api.eitje.app/open_api/skill_sets

Planning shifts

Planning Shifts are one of the core models in Eitje. They represent a user working in a team, with a specific start and end datetime. They can't be longer than 24 hours. Most shifts have a user, but this isn't required: a planning shift without a user is an 'open' shift that can be offered to your employees.

There are some validations in Planning Shifts:

- A user can never have two overlapping planning shifts
 - A shift's breaktime can't be longer than its total duration
 - A planning shift can't be edited when it's linked to a time registration shift
 - A shift must always have a team.
-

GET List planning shifts

https://open-api.eitje.app/open_api/planning_shifts

See filters above for info about the necessary date filter

- If the shift is not published, the shift is only visible for managers, not team members
 - You can pass `wage_cost: true` if you also want to receive the forecasted wage cost for these planning shifts.
-

POST Create planning shift

https://open-api.eitje.app/open_api/planning_shifts

Required: start, end, team_id

Optional: user_id, skill_set_id, remarks, break_minutes (integer), published (boolean).

PUT Update

`https://open-api.eitje.app/open_api/planning_shifts/{{planning_shift_id}}`

DELETE Delete

`https://open-api.eitje.app/open_api/planning_shifts/{{planning_shift_id}}`

POST Clear planning shifts

`https://open-api.eitje.app/open_api/planning_shifts/clear`

This endpoint removes all planning shifts in a team on a day, useful for integration partners if they want to reset what they've created. The system only allows this if none of the shifts have been updated in Eitje. Once one shift has been updated, no other shift from that day/team can be deleted anymore as a safeguard for the user. You'll receive an error explaining this.

In the future, we might allow this under a flag. You'll get the deleted records as a response so you can see what got deleted

Revenue groups

Revenue groups are arbitrary categories into which customers can divide revenue used for analytical purposes. A common pattern in restaurants is creating groups for 'kitchen' and for 'drinks'. Combined with revenue teams (see below), this enables the customer to compare revenue from certain products with labour costs from certain teams (eg comparing kitchen revenue with kitchen costs) to give them insight in how a part of their business is performing.

Property	Type	Explanation
growth_pct	integer	Used for revenue forecasting if you choose for the '1 year in past' forecast model. Eitje will take the revenue of the past year (if it's currently the third monday of March, it will pick the third monday of March last year) and increase it with this percentage to come to a forecast
target_pct	integer	Used to determine if your labour costs% of your revenue is healthy. If the labour costs% gets above this threshold, the UI will warn the user

GET List revenue groups

https://open-api.eitje.app/open_api/revenue_groups

Revenue days

All revenue in Eitje is stored daily inside a 'revenue day'. A revenue day is always connected to a revenue group, and tells you 'how much revenue was there on date x for group y'. Revenue is always reported in cents and we both have the actual revenue ('amt_in_cents') and the forecasted revenue ('forecast_amt_in_cents'). The forecast is only returned if the customer explicitly uploaded forecast data into Eitje.

Property	Type	Explanation
date	date	the date on which the revenue occurs
revenue_group	resource	group this revenue day belongs to
amt_in_cents	integer	actual (realized) revenue of that day
forecast_amt_in_cents	integer	forecasted revenue of that day (uploaded by user)

GET List revenue days

https://open-api.eitje.app/open_api/revenue_days

- See filters above for explanation
- Limits:
 - updated: 14
 - resource_date: 90

Body raw (json)

json

```
{
  "filters": {
    "start_date": "2025-03-01",
    "end_date": "2025-04-30"
  }
}
```

Revenue teams

With revenue teams a customer can assign a certain % of labour costs of a team to a revenue group. They can either choose to assign the whole team (100%) or a part of the team. Sometimes, certain teams should be 'split' among the revenue groups, often due to the team not generating any revenue. Think of a 'management' team that works across all divisions, or an 'office' team that's responsible for HR.

Property	Type	Explanation
fract	integer	percentage of the labour costs of this team that is assigned to the revenue group
team	resource	team this revenue team belongs to
revenue_group	resource	group this revenue team belongs to

GET List revenue teams

https://open-api.eitje.app/open_api/revenue_teams

Shift types

Shift types are the types of planning shifts (time registration shifts have fixed types). A customer can create as many types as they want to further categorize their planning, apart from teams. The `show_call` property adds a phone button to the employee's shift in the app. The `hide_end` property hides the end time for the employee (but not for the manager) and is often used if a manager is unsure when the shift will end.

Validations:

- A shift type's name must be unique within the organisation

Editable columns (for create & update):

- name (string, required)
- active (boolean, default true)
- hide_end (boolean)
- show_call (boolean)

GET List shift types

`https://open-api.eitje.app/open_api/shift_types`

POST Create shift type

`https://open-api.eitje.app/open_api/shift_types`

PUT Update shift type

`https://open-api.eitje.app/open_api/shift_types/{{shift_type_id}}`

DELETE Delete shift type

`https://open-api.eitje.app/open_api/shift_types/{{shift_type_id}}`

A shift type can only be deleted if it's not associated with any shift.

Teams

Teams are part of an environment and is the main way of categorizing information in Eitje. A team always has a single

environment, while an environment may have one or more teams. Almost everything is connected to a team. For shifts, this is required. It's up to the user how to use teams, but most businesses use it to represent their business units, eg 'bar' and 'kitchen' for a restaurant or 'front of house' and 'cleaning' for a hotel.

Editable columns:

- name (string, required)
- environment (required, only editable on create)
- scheduling_disabled (boolean) - team won't appear on schedule
- revenue_active (boolean, default true) - if the team should count for revenue calculations.

GET List teams

`https://open-api.eitje.app/open_api/teams`

POST Create team

`https://open-api.eitje.app/open_api/teams`

PUT Update team

`https://open-api.eitje.app/open_api/teams/{{team_id}}`

DELETE Delete team

`https://open-api.eitje.app/open_api/teams/{{team_id}}`

A team can only be deleted if it's not associated with any shift

Time registration shifts

GET List time registration shifts

`https://open-api.eitje.app/open_api/time_registration_shifts`

See info about necessary date filter above. You can pass `wage_cost: true` if you also want to receive the wage cost for these shifts.

Users

GET List users

`https://open-api.eitje.app/open_api/users`

Retrieve all users that belong to the organisation.

- This includes both active and inactive users.