Introduction

Welcome to the Timing API reference.

If you prefer a more interactive presentation of the API, you can download our Postman Collection and open it in a tool like Postman or Paw.

We also offer three Siri shortcuts that demonstrate starting and stopping timers via the API:

- Start Timing Timer
- Start Timing Timer (letting you select a project for the timer first)
- Start Fixed Timing Timer (letting you enter timer title and project name once)
- Stop Timing Timer

The "Start Fixed Timing Timer" shortcut will ask you once for a combination of timer title and project name. From then on, it will always launch a shortcut with that combination of title and project. You can create several copies of this shortcut (e.g. by duplicating it), each with a different title/project combination.

Use Cases

Using AI Tools

If you prefer using an AI assistant to explore or interact with this API, we've provided a simplified version of the documentation in `LLMs.txt`. This file is designed to be easily parsed by language models and other AI tools, making it easier to surface relevant

information quickly.

You can load the contents of `LLMs.txt` into your preferred AI system to ask questions, generate example code, or better understand how the API works.

Siri Shortcuts and Automation

Using the Timing API, you can quickly create Siri shortcuts to e.g. start and stop timers. During installation, the shortcuts will ask you for an API which you can generate in the 'API Keys' section of the web app. Once installed, simply run the corresponding shortcut to start or stop a timer.

- Start Timing Timer
- Start Timing Timer (letting you select a project for the timer first)
- Start Fixed Timing Timer (letting you enter timer title and project name once)
- Stop Timing Timer

The "Start Fixed Timing Timer" shortcut will ask you once for a combination of timer title and project name. From then on, it will always launch a shortcut with that combination of title and project. You can create several copies of this shortcut (e.g. by duplicating it), each with a different title/project combination.

Feel free to customize these shortcuts to your liking, e.g. by changing the "Start Timer" shortcut to let you select from a couple of preset titles instead. We are also interested in the shortcuts you create, so please let us know what you build with this!

You could also create scripts that start or stop a timer whenever you perform a specific action; see Start a new timer. and Stop the currently running timer. for the corresponding API calls.

Integrating with your billing system

The API also makes it possible to integrate with whatever billing system you are currently using. Simply retrieve your most recent time entries, then send them to your billing system in the desired format. You can also create a script to create a custom report in exactly the format you need, of course.

Make sure to have a look at the `?include_project_data=true` query parameter to include the corresponding project's attributes in the response. This lets you retrieve project titles without having to do a second API call to the "Projects" collection.

GRANDTOTAL INTEGRATION

The GrandTotal plugin for Timing already uses the Timing Web API to import your team members' time entries. To use that functionality, please refer the corresponding section in the documentation.

Zapier integration

We also offer a Zapier integration. This lets you connect the API to thousands of other services with just a few clicks, solving the problems mentioned above without having to write any code. To start using this integration, see the Zapier section in the web app.

Example use cases include:

- Creating projects for your team members whenever a new project is created in your project management system (e.g. Trello, Asana)
- Importing a list of projects for each team member from a Google Spreadsheet

- Automatically sending finished time entries to your billing system
- Automatically exporting your team members' time entries to a Google Spreadsheet
- Sending a message to a Slack channel whenever you start or stop a task
- Emailing you weekly reports of your team members' logged hours
- And much more...

Contact us

We also recommend for you to reach out and let us know your desired use cases! This helps us prioritize which API features to build first.

API Usage

The API root is `https://web.timingapp.com/api/v1`. All endpoint URLs share this prefix. Sample code for each available API call can be found in the right-hand column. Note that query parameters need to be URL-encoded, as shown in the Bash example for the Return a list of time entries. call.

Authentication

The Timing API requires authentication with an API key. You can generate a key in the 'API Keys' section of the web app. Once generated, add an `Authorization` header to each request with value `Bearer {{token}}`, where `{{token}}` is your key.

RATE LIMITING

The API enforces a rate limit of 500 requests per hour. You can retrieve your current quota via the `x-ratelimit-remaining` header attached to every request. In addition, sending more than 200 requests per minute will also trigger a temporary rate limit.

Request and response data

The API expects requests and returns responses in the JSON format. The actual response payload can be found in the `data` field. Additional data might be provided in the `links` and `meta` fields described below. Refer to the descriptions of individual API calls for concrete examples.

REPEATED FIELDS

Query parameters ending with a `[]` can be passed repeatedly. For example, passing `?columns[]=title&columns[]=notes` will show both the "Title" and the "Notes" columns. Optionally, ascending indices can be provided (e.g. `?columns[0]=title&columns[1]=notes`), which makes it easier to build queries in e.g. PHP.

DATE FORMAT

As Timing is a time-tracking application, its API has to work with many dates. Dates returned by the API will always be formatted as an ISO8601 string including microseconds as well as the time zone, for example `2019-01-01T00:00:00.000000+00:00`.

When sending dates in your requests, we recommend providing dates in a format appropriate to the type of request:

- For time entries, and other requests where the exact time is important, use an ISO8601 format without microseconds but including the time zone, for example `2019-01-01T00:00:00+00:00` or `2019-01-01 00:00:00+00`.
- For requests where the exact time is not important, such as on a report, use a date-only ISO8601 format, for example `2019-01-01`. In this case, the beginning of a date range (e.g. `start_date_min`) will be interpreted as the start of the day in the user's time zone, and the end of a date range (e.g. `start_date_max`), will be interpreted as the end of the day in the user's time zone.

TIMEZONE

When a timezone is not provided, the default timezone for an unqualified date is assumed to be UTC. For example, `2019-01-01T00:00:00` would be interpreted as `2019-01-01T00:00:00+0000`. This may be subject to change, however, so try to provide a timezone with your request whenever possible.

A default timezone may be provided, using the `X-Time-Zone` header. When this header is set to a valid timezone, any unqualified date is assumed to be in this timezone instead of UTC.

The header only affects input parameters, and all dates are currently returned in UTC. This however may change in the future, so your code should not make any assumptions about this, and always account for the timezone specified in the results.

REFERENCES

The Timing API identifies entities via the `self` field, which contains a link relative to the API root, for example `/projects/1`. This avoids any ambiguity about the type of the linked resource and provides you with a convenient way of looking it up, without having to look up the correct API call in this documentation: Simply append the link's value to the API root, resulting in e.g. `https://web.timingapp.com/api/v1/projects/1`.

References should be treated as opaque strings; your code should not assumptions about their structure.

SHALLOW REFERENCES

For API responses that contain related entities, these entities are usually referenced in a "shallow" fashion. Instead of including the full object, a placeholder containing only the `self` field is provided, e.g. as `"parent": {"self": "/projects/1"}`. For the Return a list of time entries. call, you can append the `?include_project_data=true` query parameter to include the corresponding project's attributes in the response. This lets you retrieve project titles without having to do a second API call to the "Projects" collection.

LINKS

Some responses include links to related queries or entities. This includes pagination and queries for related entries.

PAGINATION

By default, collection responses are paginated to 100 items per page. The links to retrieve the first, last, next and previous pages are part of the response's `links` field. Additional information about the paginated data can be found in the `meta` field.

CUSTOM FIELDS

Some entities (namely projects and time entries) can have custom fields. These are returned as the `custom_fields` collection, with an entry for each custom field on the entity. Custom fields are intended only for scripting purposes. They are exposed only via the API and are not visible anywhere in the Timing app on your Mac nor the web app.

NAMING

Custom field names must be a non-empty string, and may only contain alphanumeric characters, dashes and underscores. Additionally, they must not start with an underscore or contain only digits.

VALUES

Custom field values may only be strings. The only exception to this is providing `null`, which can be provided to remove a custom field.

USAGE

Custom fields may be added when creating or updating an entity. When updating an entity, any custom fields not provided will be left untouched. If you want to remove a custom field, you must explicitly set it to `null`.

Projects

List projects hierarchically.

requires authentication

Return the complete project hierarchy.

See Display the specified project. for the returned attributes.

Example request:

```
curl --request GET \
    --get "https://web.timingapp.com/api/v1/projects/hierarchy" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (200):

Show headers


```
{
    "data": [
        {
            "self": "/projects/1",
            "team_id": null,
            "title": "Project at root level",
            "title_chain": [
                "Project at root level"
            ],
            "color": "#FF0000",
            "productivity_score": 1,
            "is_archived": false,
            "notes": null,
            "children": [
                {
                    "self": "/projects/2",
                    "team_id": null,
                    "title": "Unproductive child project",
                    "title_chain": [
                        "Project at root level",
                        "Unproductive child project"
                    ],
                    "color": "#00FF00",
                    "productivity_score": -1,
                    "is_archived": false,
                    "notes": null,
                    "children": [].
```


REQUEST Try it out ⚡

GET `api/v1/projects/hierarchy`

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

QUERY PARAMETERS

`team_id` string *optional*

The ID of the team to list projects for. Can be omitted to list the user's private projects. See Return a list containing all the teams you are a member of. for obtaining a team ID to provide here.

List projects.

requires authentication

Return a list containing all projects.

See Display the specified project. for the returned attributes.

Example request:

```
curl --request GET \
    --get "https://web.timingapp.com/api/v1/projects?title=root&hide_archived=1" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (200):

Show headers


```
{
    "data": [
        {
            "self": "/projects/1",
            "team_id": null,
            "title": "Project at root level",
            "title_chain": [
                "Project at root level"
            ],
            "color": "#FF0000",
            "productivity_score": 1,
            "is_archived": false,
            "notes": null,
            "children": [
                {
                    "self": "/projects/2"
                }
            ],
            "parent": null,
            "custom_fields": {}
        }
    ]
}
```

REQUEST **Try it out ⚡**

GET `api/v1/projects`

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

QUERY PARAMETERS

`title` string *optional*

Filter for projects whose title contains all words in this parameter. The search is case-insensitive but diacritic-sensitive. Example: `root`

`hide_archived` boolean *optional*

If set to `1`, archived projects and their children will not be included in the result. Example: `1`

`team_id` string *optional*

The ID of the team to list projects for. Can be omitted to list the user's private projects. See Return a list containing all the teams you are a member of. for obtaining a team ID to provide here.

Create project.

requires authentication

Create a new project.

See Display the specified project. for the returned attributes.

☐ If you would like to add rules to a project, please contact us for advice.

Example request:

```
curl --request POST \
    "https://web.timingapp.com/api/v1/projects" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json" \
```

```
    --data "{
    \"title\": \"Acme Inc.\",
    \"parent\": \"\\/projects\\/1\",
    \"color\": \"#FF0000\",
    \"productivity_score\": 1,
    \"is_archived\": false,
    \"notes\": \"Some more detailed notes\",
    \"custom_fields\": {
        \"field_name\": \"field_value\"
    }
}"
```

Example response (201):

Show headers


```
{
    "data": {
        "self": "/projects/2",
        "team_id": null,
        "title": "Acme Inc.",
        "title_chain": [
            "Project at root level",
            "Acme Inc."
        ],
        "color": "#FF0000",
        "productivity_score": 1,
        "is_archived": false,
        "notes": "Some more detailed notes",
        "children": [],
        "parent": {
            "self": "/projects/1"
        },
        "custom_fields": {
            "field_name": "field_value"
        }
    },
    "links": {
        "time-entries": "https://web.timingapp.com/api/v1/time-entries?project[]=/projects/2"
    }
}
```


REQUEST **Try it out ⚡**

POST `api/v1/projects`

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

BODY PARAMETERS

`title` string

The project's title. Example: `Acme Inc.`

`parent` string *optional*

A reference to an existing project. The new project will be appended to the parent's children. Can be a project reference in the form `"/projects/1"`, a project title (e.g. `"Project at root level"`), or an array with the project's entire title chain (e.g. `["Project at root level", "Unproductive child project"]`). Example: `/projects/1`

`color` string *optional*

The project's color, in hexadecimal format (`#RRGGBB`). If omitted, a color with random hue, 70% saturation and 100% value will be used. Example: `#FF0000`

`productivity_score` number *optional*

The project's productivity rating, between -1 (very unproductive) and 1 (very productive). Defaults to 1. Example: `1`

`is_archived` boolean *optional*

Whether the project has been archived. Defaults to false. Example: `false`

`team_id` string *optional*

The ID of the team to add the project to. See Return a list containing all the teams you are a member of. for obtaining a team ID to provide here.

`notes` string *optional*

The project's notes. Example: `Some more detailed notes`

`custom_fields` object *optional*

A list of custom field name/value pairs to store. For more details, see Custom fields.

Show project.

requires authentication

Display the specified project.

The following attributes will be returned:

- `self`: A reference to the entity itself, relative to the API root.
- `title`: The project's title.
- `title_chain`: An array containing the title of the project and all its ancestors. Example: `["Parent", "Child"]`
- `color`: The project's color, in hexadecimal format (`#RRGGBB`). Example: `#FF0000`
- `productivity_score`: The project's productivity rating, between -1 (very unproductive) and 1 (very productive). Example: `1`
- `is_archived`: Whether the project has been archived. Defaults to false. Example: `false`
- `parent`: A reference to the enclosing project.
- `children`: The project's children.
- `team_id`: The ID of the team that this project belongs to, if applicable.

☐ Child projects are provided as references; i.e. they only contain the `self` attribute.

Example request:

```
curl --request GET \
    --get "https://web.timingapp.com/api/v1/projects/1" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (200):

Show headers


```
{
    "data": {
        "self": "/projects/1",
        "team_id": null,
        "title": "Project at root level",
        "title_chain": [
            "Project at root level"
        ],
        "color": "#FF0000",
        "productivity_score": 1,
        "is_archived": false,
        "notes": null,
        "children": [
            {
                "self": "/projects/2"
            }
        ],
        "parent": null,
        "custom_fields": {}
    },
    "links": {
        "time-entries": "https://web.timingapp.com/api/v1/time-entries?project[]=/projects/1"
    }
}
```


REQUEST **Try it out ⚡**

GET **`api/v1/projects/{project_id}`**

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

URL PARAMETERS

`project_id` string

The ID of the project to display. Example: `1`

Update project.

requires authentication

Update the specified project.

See Display the specified project. for the returned attributes.

- ☐ Omitted fields will not be updated, even when using the `PUT` method.

- ☐ Changing a project's parent or children is currently not possible.

- ☐ If you would like to add rules to a project, please contact us for advice.

Example request:

```
curl --request PUT \
    "https://web.timingapp.com/api/v1/projects/1" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json" \
    --data "{
    \"title\": \"Acme Inc.\",
    \"color\": \"#FF0000\",
    \"productivity_score\": 1,
    \"is_archived\": false,
    \"notes\": \"Some more detailed notes\",
    \"custom_fields\": {
        \"field_name\": \"field_value\"
    }
}"
```

Example response (200):

Show headers


```
{
    "data": {
        "self": "/projects/1",
        "team_id": null,
        "title": "Acme Inc.",
        "title_chain": [
            "Acme Inc."
        ],
        "color": "#FF0000",
        "productivity_score": 1,
        "is_archived": false,
        "notes": "Some more detailed notes",
        "children": [
            {
                "self": "/projects/2"
            }
        ],
        "parent": null,
        "custom_fields": {
            "field_name": "field_value"
        }
    },
    "links": {
```

```
        "time-entries": "https://web.timingapp.com/api/v1/time-entries?project[]=/projects/1"
    }
}
```


REQUEST **Try it out ⚡**

PUT `api/v1/projects/{project_id}`

PATCH `api/v1/projects/{project_id}`

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

URL PARAMETERS

`project_id` string

The ID of the project to update. Example: `1`

BODY PARAMETERS

`title` string

The project's title. Example: `Acme Inc.`

`color` string *optional*

The project's color, in hexadecimal format (`#RRGGBB`). Example: `#FF0000`

`productivity_score` number *optional*

The project's productivity rating, between -1 (very unproductive) and 1 (very productive). Example: `1`

`is_archived` boolean *optional*

Whether the project has been archived. Example: `false`

`notes` string *optional*

The project's notes. Example: `Some more detailed notes`

`custom_fields` object *optional*

A list of custom field name/value pairs to update. For more details, see Custom fields.

Delete project.

requires authentication

Delete the specified project and all of its children.

Example request:

```
curl --request DELETE \
    "https://web.timingapp.com/api/v1/projects/1" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (204):

Show headers

```
Empty response
```

REQUEST **Try it out ⚡**

DELETE **`api/v1/projects/{project_id}`**

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

URL PARAMETERS

`project_id` string

The ID of the project to delete. Example: `1`

Reports

Generate report.

requires authentication

Generate a report that can contain both time entries and app usage.

Returns a JSON array with several rows; each row includes the total duration (in seconds) belonging to the corresponding other (configurable) columns.

The `include_app_usage` and `include_team_members` parameters govern whether to include app usage (otherwise, only time entries are returned) as well as data for other team members.
The `start_date_min`, `start_date_max`, `projects`(also see `include_child_projects`) and `search_query` parameters allow filtering the returned data.
The `columns`, `project_grouping_level`, `include_project_data`, `timespan_grouping_mode`, and `sort` parameters govern the presentation of the returned data.

☐ Fetching large amounts of app usage can put a substantial amount of load on our servers, so please be mindful before frequently requesting large amounts of data using this API.

☐ If no date range filter is provided by setting *both* `start_date_min` *and* `start_date_max`, this query returns all time entries between midnight (UTC) 30 days ago and end of day (UTC) today.

Example request:

```
curl --request GET \
    --get "https://web.timingapp.com/api/v1/report?include_app_usage=0&include_team_members=0&team_members[]=%2Fusers%2F1&start_date_min=2019-01-01&start
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (200):

Show headers


```
{
    "data": [
        {
            "duration": 3600,
            "project": {
                "self": "/projects/1",
                "team_id": null,
                "title": "Project at root level",
                "title_chain": [
                    "Project at root level"
                ],
                "color": "#FF0000",
                "productivity_score": 1,
                "is_archived": false,
                "notes": null,
                "parent": null,
                "custom_fields": {}
            }
        }
    ]
}
```


REQUEST Try it out ⚡

GET `api/v1/report`

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

QUERY PARAMETERS

`include_app_usage` boolean *optional*

Whether to include app usage in the report. If false, only time entries are returned. Default: `0` Example: `0`

`include_team_members` boolean *optional*

If true, the response will also contain time entries that belong to other team members, provided the current user has permission to view them. Default: `0` Example: `0`

`team_members` string[] *optional*

Restricts the query to data associated with the given user. Can be repeated to include time entries from several users.

`start_date_min` string *optional*

Restricts the query to data whose start date is equal to or later than this parameter. Example: `2019-01-01`

`start_date_max` string *optional*

Restricts the query to data whose start date is equal to or earlier than this parameter. Example: `2019-01-01`

`projects` string[] *optional*

Restricts the query to data associated with the given project. Can be repeated to include time entries from several projects. If you would like to include time entries that are not assigned to any project, you can provide an empty string, i.e. `projects[]=`

`include_child_projects` boolean *optional*

If true, the response will also contain time entries that belong to any child projects of the ones provided in `projects[]`. Default: `0` Example: `1`

`search_query` string *optional*

Restricts the query to time entries whose title and/or notes contain all words in this parameter. The search is case-insensitive but diacritic-sensitive. Note: this parameter can not be used when app usage is included. Example: `meeting`

`columns` string[] *optional*

Which columns to show. Can be repeated to provide multiple columns. The `user` column is ignored if `include_team_members` is false. Possible values: `project`, `title`, `notes`, `timespan`, `user`. Default: `user`, `project`, `title`. `start_date` and `end_date` is shown when `timespan` column is sent.

`project_grouping_level` integer *optional*

When this argument is provided, report lines for projects below the given level will be aggregated by their parent project on the given level. For example, when `project_grouping_level` is 0, all times in sub-projects will be counted towards the corresponding project on the "root" (i.e. highest) level in the project tree. Can be a non-negative integer or -1. The default is -1, which indicates no grouping (i.e. all projects will be returned, regardless of how deep they are in the hierarchy). Requires `columns[]` to contain `project`. Example: `0`

`include_project_data` boolean *optional*

If true, the properties of each line's project will be included in the response. Requires `columns[]` to contain `project`. Default: `0` Example: `1`

`timespan_grouping_mode` string *optional*

When this argument is provided, report lines will be aggregated according to the given calendar unit. Possible values: `exact`, `day`, `week`, `month`, `year`. Default: `exact` Example: `day`

`sort` string[] *optional*

Sort the results ascending by the given column; for descending order prefix the column name with a minus sign. Can be repeated to provide multiple sort columns. Default: `-duration`. Examples: `sort[]=-duration` -> Sort descending by duration. `sort[]=user&sort[]=-duration` -> Sort ascending by user, then descending by duration.

Teams

List team members.

requires authentication

Return a list containing all active members of the given team.

Members with pending invitations will be excluded.

The following attributes will be returned:

- `self`: A reference to the entity itself, relative to the API root.
- `email`: The team member's email address.
- `name`: The team member's name. May be null if the team member has not entered a name in their account profile.

Example request:

```
curl --request GET \
    --get "https://web.timingapp.com/api/v1/teams/1/members" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (200):

Show headers

```
{
    "data": [
        {
            "self": "/users/1",
            "email": "johnny@appleseed.net",
            "name": "Johnny Appleseed"
        }
    ]
}
```


REQUEST **Try it out ⚡**

GET **`api/v1/teams/{team_id}/members`**

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

URL PARAMETERS

`team_id` string

The ID of the team to list members for. Example: `1`

List teams.

requires authentication

Return a list containing all the teams you are a member of.

Example request:

```
curl --request GET \
    --get "https://web.timingapp.com/api/v1/teams" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (200):

Show headers


```
{
    "data": [
```

```
        {
            "id": "/teams/1",
            "name": "Demo Team"
        }
    ]
}
```

REQUEST **Try it out ⚡**

GET `api/v1/teams`

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

Time Entries

Start timer.

requires authentication

Start a new timer.

This also stops the currently running timer if there is one.

See Display the specified time entry. for the returned attributes.

☐ The title and project fields can not both be empty.

Example request:

```
curl --request POST \
    "https://web.timingapp.com/api/v1/time-entries/start" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json" \
    --data "{
    \"start_date\": \"2019-01-01T00:00:00+00:00\",
    \"project\": \"Unproductive child project\",
    \"title\": \"Client Meeting\",
    \"notes\": \"Some more detailed notes\",
    \"replace_existing\": false,
    \"custom_fields\": {
        \"field_name\": \"field_value\"
```

```
    }
}"
```

Example response (201):

Show headers


```
{
    "data": {
        "self": "/time-entries/2",
        "start_date": "2019-01-01T00:00:00.000000+00:00",
        "end_date": "2019-01-01T00:00:00.000000+00:00",
        "duration": 0,
        "project": {
            "self": "/projects/2"
        },
        "title": "Client Meeting",
        "notes": "Some more detailed notes",
        "is_running": true,
        "creator_id": "/users/1",
        "creator_name": "Johnny Appleseed",
        "custom_fields": {
            "field_name": "field_value"
        }
    },
    "message": "Timer 'Client Meeting' started."
}
```


REQUEST **Try it out ⚡**

POST **`api/v1/time-entries/start`**

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

BODY PARAMETERS

`start_date` string *optional*

The date this timer should have started at. Defaults to "now". Example: `2019-01-01T00:00:00+00:00`

`project` string *optional*

The project this timer is associated with. Can be a project reference in the form `"/projects/1"`, a project title (e.g. `"Project at root level"`), or an array with the project's entire title chain (e.g. `["Project at root level", "Unproductive child project"]`). Example: `Unproductive child project`

`title` string *optional*

The timer's title. Example: `Client Meeting`

`notes` string *optional*

The timer's notes. Example: `Some more detailed notes`

`replace_existing` boolean *optional*

If true, any existing time entries that overlap with the new time entry will be adjusted to avoid overlap, or deleted altogether. Defaults to false. Example: `false`

`custom_fields` object *optional*

A list of custom field name/value pairs to store. For more details, see Custom fields.

Stop timer.

requires authentication

Stop the currently running timer.

Returns the stopped timer's attributes as listed under Display the specified time entry..

Example request:

```
curl --request PUT \
    "https://web.timingapp.com/api/v1/time-entries/stop" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (200):

Show headers


```
{
    "data": {
        "self": "/time-entries/1",
        "start_date": "2019-01-01T00:00:00.000000+00:00",
        "end_date": "2019-01-01T01:00:00.000000+00:00",
        "duration": 3600,
        "project": {
            "self": "/projects/1"
        },
        "title": "Client Meeting",
        "notes": "Some more detailed notes",
        "is_running": false,
        "creator_id": "/users/1",
        "creator_name": "Johnny Appleseed",
        "custom_fields": {}
    },
    "message": "Timer 'Client Meeting' stopped."
}
```


REQUEST **Try it out ⚡**

PUT `api/v1/time-entries/stop`

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

Show latest time entry.

requires authentication

Redirect to the latest time entry.

See Display the specified time entry. for the route the redirect points to.

☐ If no time entry is found, a 404 response will be returned.

☐ This route is only available for `GET` requests. If you'd like to e.g. *edit* the latest time entry, use this route to retrieve a link to the desired entry, then use that link to assemble a new request for the desired action.

Example request:

```
curl --request GET \
    --get "https://web.timingapp.com/api/v1/time-entries/latest" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (302):

Show headers

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8" />
        <meta http-equiv="refresh" content="0;url='https://web.timingapp.com/api/v1/time-entries/1'" />

        <title>Redirecting to https://web.timingapp.com/api/v1/time-entries/1</title>
    </head>
    <body>
        Redirecting to <a href="https://web.timingapp.com/api/v1/time-entries/1">https://web.timingapp.com/api/v1/time-entries/1</a>.
    </body>
</html>
```

REQUEST **Try it out ⚡**

GET `api/v1/time-entries/latest`

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

Show running timer.

Redirect to the currently running timer.

See Display the specified time entry. for the route the redirect points to.

☐ If no timer is currently running, a 404 response will be returned.

☐ This route is only available for `GET` requests. If you'd like to e.g. *edit* the latest time entry, use this route to retrieve a link to the desired entry, then use that link to assemble a new request for the desired action.

Example request:

```
curl --request GET \
    --get "https://web.timingapp.com/api/v1/time-entries/running" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (404):

Show headers

```
{
    "message": "No running timer found."
}
```

REQUEST **Try it out ⚡**

GET `api/v1/time-entries/running`

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

List time entries.

requires authentication

Return a list of time entries.

See Display the specified time entry. for the returned attributes.

Items are ordered descending by their `start_date` field.

☐ If no date range filter is provided by setting *both* `start_date_min` *and* `start_date_max`, this query returns all time entries between midnight (UTC) 30 days ago and end of day (UTC) today.

Example request:

```
curl --request GET \
    --get "https://web.timingapp.com/api/v1/time-entries?start_date_min=2019-01-01&start_date_max=2019-01-01&projects[]=%2Fprojects%2F1&include_child_pro
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (200):

Show headers


```
{
    "data": [
        {
            "self": "/time-entries/1",
            "start_date": "2019-01-01T00:00:00.000000+00:00",
            "end_date": "2019-01-01T01:00:00.000000+00:00",
```

```
            "duration": 3600,
            "project": {
                "self": "/projects/1",
                "team_id": null,
                "title": "Project at root level",
                "title_chain": [
                    "Project at root level"
                ],
                "color": "#FF0000",
                "productivity_score": 1,
                "is_archived": false,
                "notes": null,
                "parent": null,
                "custom_fields": {}
            },
            "title": "Client Meeting",
            "notes": "Some more detailed notes",
            "is_running": false,
            "creator_id": "/users/1",
```


REQUEST **Try it out ⚡**

GET **`api/v1/time-entries`**

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

QUERY PARAMETERS

`start_date_min` string *optional*

Restricts the query to time entries whose start date is equal to or later than this parameter. Example: `2019-01-01`

`start_date_max` string *optional*

Restricts the query to time entries whose start date is equal to or earlier than this parameter. Example: `2019-01-01`

`projects` string[] *optional*

Restricts the query to time entries associated with the given project. Can be repeated to include time entries from several projects. If you would like to include time entries that are not assigned to any project, you can provide an empty string, i.e. `projects[]=`

`include_child_projects` boolean *optional*

If true, the response will also contain time entries that belong to any child projects of the ones provided in `projects[]`. Default: `0` Example: `1`

`search_query` string *optional*

Restricts the query to time entries whose title and/or notes contain all words in this parameter. The search is case-insensitive but diacritic-sensitive. Example: `meeting`

`is_running` boolean *optional*

If provided, returns only time entries that are either running or not running. Default: `0` Example: `0`

`include_project_data` boolean *optional*

If true, the properties of the time entry's project will be included in the response. Default: `0` Example: `1`

`include_team_members` boolean *optional*

If true, the response will also contain time entries that belong to other team members, provided the current user has permission to view them. Default: `0` Example: `0`

`team_members` string[] *optional*

Restricts the query to data associated with the given user. Can be repeated to include time entries from several users.

Create time entry.

requires authentication

Create a new time entry.

See Display the specified time entry. for the returned attributes.

- ☐ The title and project fields can not both be empty.

Example request:

```
curl --request POST \
    "https://web.timingapp.com/api/v1/time-entries" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json" \
    --data "{
    \"start_date\": \"2019-01-01T00:00:00+00:00\",
    \"end_date\": \"2019-01-01T01:00:00+00:00\",
    \"project\": \"Unproductive child project\",
    \"title\": \"Client Meeting\",
    \"notes\": \"Some more detailed notes\",
    \"replace_existing\": false,
    \"custom_fields\": {
        \"field_name\": \"field_value\"
    }
}"
```

Example response (201):

Show headers

```
{
    "data": {
        "self": "/time-entries/2",
```

```
            "start_date": "2019-01-01T00:00:00.000000+00:00",
            "end_date": "2019-01-01T01:00:00.000000+00:00",
            "duration": 3600,
            "project": {
                "self": "/projects/2"
            },
            "title": "Client Meeting",
            "notes": "Some more detailed notes",
            "is_running": false,
            "creator_id": "/users/1",
            "creator_name": "Johnny Appleseed",
            "custom_fields": {
                "field_name": "field_value"
            }
        }
    }
```


REQUEST **Try it out ⚡**

POST `api/v1/time-entries`

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

BODY PARAMETERS

`start_date` string

The time entry's start date and time. Example: `2019-01-01T00:00:00+00:00`

`end_date` string

The time entry's end date and time. Example: `2019-01-01T01:00:00+00:00`

`project` string *optional*

The project this time entry is associated with. Can be a project reference in the form `"/projects/1"`, a project title (e.g. `"Project at root level"`), or an array with the project's entire title chain (e.g. `["Project at root level", "Unproductive child project"]`). Example: `Unproductive child project`

`title` string *optional*

The time entry's title. Example: `Client Meeting`

`notes` string *optional*

The time entry's notes. Example: `Some more detailed notes`

`replace_existing` boolean *optional*

If true, any existing time entries that overlap with the new time entry will be adjusted to avoid overlap, or deleted altogether. Defaults to false. Example: `false`

`custom_fields` object *optional*

A list of custom field name/value pairs to store. For more details, see Custom fields.

Show time entry.

requires authentication

Display the specified time entry.

The following attributes will be returned:

- `self`: A link to the entity itself, relative to the API root.
- `start_date`: The time entry's start date and time.
- `end_date`: The time entry's end date and time.
- `duration`: The time entry's total duration, in seconds.
- `project`: The project this time entry is associated with.
- `title`: The time entry's title.
- `notes`: The time entry's notes.
- `is_running`: Whether the time entry is currently running. Only one time entry can be running at any given time.

Example request:

```
curl --request GET \
    --get "https://web.timingapp.com/api/v1/time-entries/1?other_user_id=%2Fusers%2F1" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (200):

Show headers


```
{
    "data": {
        "self": "/time-entries/1",
        "start_date": "2019-01-01T00:00:00.000000+00:00",
        "end_date": "2019-01-01T01:00:00.000000+00:00",
        "duration": 3600,
        "project": {
            "self": "/projects/1"
        },
        "title": "Client Meeting",
        "notes": "Some more detailed notes",
        "is_running": false,
        "creator_id": "/users/1",
        "creator_name": "Johnny Appleseed",
        "custom_fields": {}
    }
}
```


REQUEST Try it out ⚡

GET `api/v1/time-entries/{time_entry_id}`

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

URL PARAMETERS

`time_entry_id` string

The ID of the time entry to display. Example: `1`

QUERY PARAMETERS

`other_user_id` string *optional*

The ID of the other user must be provided when making changes to the time entries of a colleague. This, along with the user having the required permissions, confirms that changes to the time entries of a colleague are allowed, and that the correct user is being targetted. Without this field, operations for other users will be rejected to prevent accidental changes to other users' time entries. Example: `/users/1`

Update time entry.

requires authentication

Update the specified time entry.

See Display the specified time entry. for the returned attributes.

- ☐ Omitted fields will not be updated, even when using the `PUT` method.

- ☐ A time entry's title and project fields can not both be empty.

Example request:

```
curl --request PUT \
    "https://web.timingapp.com/api/v1/time-entries/1?other_user_id=%2Fusers%2F1" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json" \
    --data "{
    \"start_date\": \"2019-01-01T00:00:00+00:00\",
    \"end_date\": \"2019-01-01T01:00:00+00:00\",
    \"project\": \"Unproductive child project\",
    \"title\": \"Client Meeting\",
```

```
    \"notes\": \"Some more detailed notes\",
    \"replace_existing\": false,
    \"custom_fields\": {
        \"field_name\": \"field_value\"
    }
}"
```

Example response (200):

Show headers


```
{
    "data": {
        "self": "/time-entries/1",
        "start_date": "2019-01-01T00:00:00.000000+00:00",
        "end_date": "2019-01-01T01:00:00.000000+00:00",
        "duration": 3600,
        "project": {
            "self": "/projects/2"
        },
        "title": "Client Meeting",
        "notes": "Some more detailed notes",
        "is_running": false,
        "creator_id": "/users/1",
        "creator_name": "Johnny Appleseed",
        "custom_fields": {
            "field_name": "field_value"
        }
    }
}
```


REQUEST **Try it out ⚡**

PUT **`api/v1/time-entries/{time_entry_id}`**

PATCH **`api/v1/time-entries/{time_entry_id}`**

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

URL PARAMETERS

`time_entry_id` string

The ID of the time entry to update. Example: `1`

QUERY PARAMETERS

`other_user_id` string *optional*

The ID of the other user must be provided when making changes to the time entries of a colleague. This, along with the user having the required permissions, confirms that changes to the time entries of a colleague are allowed, and that the correct

user is being targetted. Without this field, operations for other users will be rejected to prevent accidental changes to other users' time entries. Example: `/users/1`

BODY PARAMETERS

`start_date` string *optional*

The time entry's start date and time. Example: `2019-01-01T00:00:00+00:00`

`end_date` string *optional*

The time entry's start date and time. Example: `2019-01-01T01:00:00+00:00`

`project` string *optional*

The project this time entry is associated with. Can be a project reference in the form `"/projects/1"`, a project title (e.g. `"Project at root level"`), or an array with the project's entire title chain (e.g. `["Project at root level", "Unproductive child project"]`). Example: `Unproductive child project`

`title` string *optional*

The time entry's title. Example: `Client Meeting`

`notes` string *optional*

The time entry's notes. Example: `Some more detailed notes`

`replace_existing` boolean *optional*

If true and the entry's start or end date has changed, any existing time entries that overlap with the updated time entry will be adjusted to avoid overlap, or deleted altogether. Defaults to false. Example: `false`

`custom_fields` object *optional*

A list of custom field name/value pairs to update. For more details, see Custom fields.

Delete time entry.

requires authentication

Delete the specified time entry.

Example request:

```
curl --request DELETE \
    "https://web.timingapp.com/api/v1/time-entries/1?other_user_id=%2Fusers%2F1" \
    --header "Authorization: Bearer {{token}}" \
    --header "Content-Type: application/json" \
    --header "Accept: application/json"
```

Example response (204):

Show headers

```
Empty response
```

REQUEST **Try it out ⚡**

DELETE `api/v1/time-entries/{time_entry_id}`

HEADERS

`Authorization`

Example: `Bearer {{token}}`

`Content-Type`

Example: `application/json`

`Accept`

Example: `application/json`

URL PARAMETERS

`time_entry_id` string

The ID of the time entry to delete. Example: `1`

QUERY PARAMETERS

`other_user_id` string *optional*

The ID of the other user must be provided when making changes to the time entries of a colleague. This, along with the user having the required permissions, confirms that changes to the time entries of a colleague are allowed, and that the correct user is being targetted. Without this field, operations for other users will be rejected to prevent accidental changes to other users' time entries. Example: `/users/1`