Interface Documentation

Time Tracking API

Contents

Introduction.....2
Getting Started.....2
Authentication.....3
Server Response.....3
Resources.....4

Introduction

TimeTac provides an easy accessible RESTful application programming interface (API) for developers to connect and integrate a TimeTac account into their own apps.

The API as well as every TimeTac company account is decoupled from all the other company accounts. Your API access will be restricted to your account only, and no other API user can ever access your companies' data without your credentials (see chapter "Authentication" for more information).

Getting Started

The TimeTac API is not open to the public. In order to get started with the API, you will need to request a private API key that will enable the API for your TimeTac company account and grant you access to your resources. Please contact us at support@timetac.com and tell us about the project you are planning to use our API for, and we will be happy to assist you with an API key and answers to all the questions you might have.

Once you have your key you will be able to make authenticated requests to our API by calling the URL for a specific resource (see chapter "Resources"). A resource URL will always have the following format:

Generic API URL
`https://go.timetac.com/<company_account>/api/<api_version>/<resource>/<action>/`

- <company_account>
  The name of the requested company account. We will only allow requests to your own company account that will be permanently mapped to your API key.
- <api_version>
  If updates to our API would break backward compatibility, we will release a new API version but keep the previous one available to give every developer enough time to adapt their apps to the new changes.
- <resource>
  The name of the requested resource.
- <action>
  The action you want to perform on the specified resource (get/create/update/delete).

Example
`https://go.timetac.com/TimeTacGmbH/api/v1/tasks/get/?id=31`

Authentication

Every client that wants to initiate requests to the TimeTac API has to implement the HTTP Basic authentication. Because TimeTac has a stateless RESTful API, there is no active session between a client and the server, therefore every single request must have a valid “Authorization” header set. The username will be your company accounts login name and the password your API key. HTTP Basic authentication provides no confidentiality protection for the transmitted user credentials, so please ensure that every request is made only by using a HTTPS connection.

If an error occurs during the authentication, the server will respond with a HTTP status code 401 (not authorized).

Server Response

Every request to the TimeTac API will have a well formed JSON response:

Key	Type	Description
Host	String	The host that processed the request.
Success	Boolean	False if any error occurred while processing the request
NumResults	Integer	The number of results contained in a successful response.
ResourceName	String	The name of the resource that was requested.
Results	Array	If the requested action provides a result set and the set is not empty, this array will contain all resource entities as JSON objects.
Error	Integer	If an error occurred, this will contain the error code so a client can react accordingly (see “Error handling”).
ErrorMessage	String	A human readable error message that will explain the error. These messages are only available in English.

Example

```
{
  "Host": "go.timetac.com",
  "Success": true,
  "NumResults": 1,
  "ResourceName": "User",
  "Results": [
    {
      "id": "1",
      "internal_user_group": "1",
      "active": "1",
      "hr_manager": "1",
      "department_id": "3",
      "username": "manager",
      "lastname": "Wolf",
      "firstname": "David",
      …
    }
  ]
}
```

Error handling

If any error occurred while the request was processed, the “Success” key of the response will be set to false. In that case, two new keys will be included in the response JSON, a error code in “Error” and a human readable description in “ErrorMessage”.

A client should always at least check the error code in “Error” and try to implement a basic form of error correction if possible. The following table should help with identifying a problem:

Error Code	Description
400	Server received a bad request. Most of the time this is due to a missing required parameter.
401	Client tried to access a forbidden resource. This may occur if your account tries to access restricted information or the provided user credentials are wrong.
403	Server received an unauthenticated request. The client did not implement a HTTP Basic authentication header.
422	The request parameters described an unprocessable entity, probably because of a parameter had an invalid value.
500	An internal server error occurred on the backend side.
503	The service is temporary unavailable. The server is experiencing technical difficulties which prevented it from processing the request.

Resources

The following chapters will explain all resources that are available through the TimeTac API and how they should be handled. The API will enable you to create, read, update and delete (CRUD) resources in the TimeTac application, but depending on the resource you are requesting some of these operations may be not available due to security restrictions.

In almost all cases, you will have to adjust your request method according to the type of action you want to call. The API will respond to the following request methods as defined below:

- **GET**
  Read only request, these Requests will never affect the state of a resource.
  - Parameters are expected to be part of the URI query string
- **POST**
  Creates a new entity of the requested resource. When called multiple times in a row, each request will lead to a completely new entity being created.
  If possible, the created resource will be included in the result of the response.
  - Parameters are expected to be x-www-form-urlencoded
- **PUT**
  Updates an existing resource. These requests are idempotent, so identical subsequent calls won't change the result beyond the initial application.
  If possible, the created resource will be included in the result of the response.
  - Parameters are expected to be in the raw body
- **DELETE**
  Completely removes the specified entity.
  - Parameters are expected to be part of predefined URI segments

For each resource, you will find a table with all the parameters that are supported, alongside with their description, datatype and availability for the different actions:

Name	Type	Description	C(reate)	R(ead)	U(pdate)
Name/key of the parameter	• Integer Whole numbers • Char A single character. • String If not stated otherwise, all unicode characters. • Boolean Either “1” or “0” • Decimal If not stated otherwise, 10 digits to the left, and 2 digits to the right of the decimal point. • Date Values in “YYYY-MM-DD” format. • DateTime Values in “YYYY-MM-DD HH:MM:SS” format	Short explanation	• “-“ Not supported • “O” Optional • “R” Required		

The following parameters are available for all resources:

Name	Type	Description	C	R	U
limit	Integer	Limits the amount entries in the result. Default value: 100	-	O	-
offset	Integer	If there are more results than can be displayed in one response (> limit), offset will skip this many entries and returns the next page of the resultset.	-	O	-
order_by	String	Any of the supported parameters name. This will sort the resultset by the specified field in alphanumerical order.	-	O	-
order_desc	Boolean	If set to “1”, the resultset will be sorted in reversed alphanumerical order.	-	O	-
<fieldname>__operand	String	If specified, the search will use the given operand for its comparison. Supported operands are: • “eq”: Equal (=), Default • “gt”: Greater than (>) • “gteq”: Greater than or equal (>=) • “lt”: Less than (<) • “lteq”: Less than or equal (<=) • “in”: Comma separated list of values to check against. E.g.: username__operand=smith,doe • “like”: Checks if the given value if part of the actual field value.			

Users

Supported actions:

Method	URI	Description
GET	users/get/	Read existing entries
POST	users/create/	Create a new entry
PUT	users/edit/	Update an existing entry
DELETE	users/delete/:id/	Delete an existing entry. ":id" has to be a valid ID of an existing resource.

Description:

TimeTac user accounts related to the company account. Each entry contains both, personal information about the user as well as settings for the account configuration.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset	-	O	R
internal_user_group	Integer	The group this user belongs to: • 1 = Administrator • 2 = Employee	R	O	O
active	Boolean		O	O	O
hr_manager	Boolean		O	O	O
department_id	Integer		R	O	O
start_task_at_login	Boolean		O	O	O
username	String	*Valuerange: [0-9a-zA-Z\.\@_\-]* *Length: 3 - 32*	O	O	O
lastname	String	*Max. length: 64*	R	O	O
firstname	String	*Max. length: 64*	R	O	O
abbrevation	String	*Max. length: 4*	O	O	O
permission_change_all_timers	Boolean		O	O	O
permission_change_assigned_user_timers	Boolean		O	O	O
permission_change_own_timers	Boolean		O	O	O
permission_add_own_timers	Boolean		O	O	O
permission_statistic_for_all_users	Boolean		O	O	O
permission_show_tt_ex_post_one_employees	Boolean		O	O	O
permission_show_tt_ex_post_more_employees	Boolean		O	O	O
permission_show_assign_favourites	Boolean		O	O	O
permission_show_assign_todos	Boolean		O	O	O
permission_edit_tasks_projects	Boolean		O	O	O

permission_show_settings_pm_skills	Boolean		O	O	O
permission_show_settings_client_list	Boolean		O	O	O
internal_cost_per_hour	Decimal		O	O	O
revenue_per_hour	Decimal		O	O	O
Password	String	*Valuerange: [0-9a-zA-ZßäÄöÖüÜ\.\,\#\@_\+\(\)\?\/\'\&\;\:\!\-]* *Length: 6 - 32*	R	-	O
email_address	String		O	O	O
language_id	Int	• 1 = English • 2 = German	O	O	O
phone	String		O	O	O
skype	String		O	O	O
individual_value_1	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
individual_value_2	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
individual_value_3	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
external_id_1	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
external_id_2	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
external_id_3	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
mobile_allowed	Boolean		O	O	O
pin_code	Strin		O	-	O
gender	String		O	O	O
country_id	Integer	The ID of a country. These IDs can be retrieved from “*GET countries/get”.*	O	O	O
allowed_ips	String		O	O	O
payroll_accounting_starts_at	Date	The date at which the accounting for this user should start. If left empty on a create request, the system will default to the current date.	O	O	O
action_log_initial_value_holiday	Decimal	The initial value for the holiday balance	O	O	O
payroll_accounting_initial_value_working_time_total_balance	Decimal	The initial value for the working time balance	O	O	O
public_holiday_template_id	Integer	The ID of a public holiday template. These IDs can be retrieved from “*GET*	R	O	O

		publicHolidayTemplates /get".			
manual_timetracking	Boolean		O	O	O
automatic_tracker_writing_task_id	Integer		O	O	O
cost_acc_non_working_task_id	Integer		O	O	O
allow_start_task	Boolean		O	O	O
allow_cancel_holiday_request	Boolean		O	O	O
salutation_id	Integer	The ID of the salutation that should be used for this user. These IDs can be retrieved by *"GET salutations/get"*	R	O	O
phone_1	String		O	O	O
phone_2	String		O	O	O
birthday	Date		O	O	O
entry_date	Date		O	O	O
exit_date	Date		O	O	O
master_data_notes	String		O	O	O
social_security_number	String	*Max. length: 32*	O	O	O
address_1	String	*Max. length: 128*	O	O	O
address_2	String	*Max. length: 128*	O	O	O
Zip	String	*Max. length: 8*			
allow_to_see_other_user_leave_type	Boolean		O	O	O
permission_show_edit_user_user_settings	Boolean		O	O	O
terminal_transponder_nr	String	The ID of a RFID transponder used for remote terminal timetracking.	O	O	O
external_id	String	The ID given if a user is managed by an external system. *Max. length: 2083*	O	O	O
company_name	String	*Max. length: 64*	O	O	O

Salutations

Supported actions:

Method	URI	Description
GET	salutation/get/	Read existing entries

Description:

Supported salutations in TimeTac, mainly used by other resources for n:1 relations.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	-
const	String	A constant suitable for programmatic use (e.g. translations).	-	O	-
gender	Char	A single character for m(ale) or f(emale).	-	O	-

Skills

Supported actions:

Method	URI	Description
GET	skills/get/	Read existing entries

Description:

Skills in TimeTac that can be assigned to tasks to define the nature of the work that will be done.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	-
syn	String	The human readable name of the skill.	-	O	-
sort_order	Integer	custom numerical sort order.	-	O	-
internal_cost_per_hour	Decimal		-	O	-
revenue_per_hour	Decimal		-	O	-
rillsoft_id	String	*Max. length: 16*	-	O	-

Public Holiday Templates

Supported actions:

Method	URI	Description
GET	publicHolidayTemplates/get/	Read existing entries

Description:

Public holiday templates define the dates and names of public holidays for different countries, regions and religions.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	-
mother_id	Integer	The ID of the dataset from which it inherits already defined public holidays.	-	O	-
template_name	String	A human readable name for the template.	-	O	-
default_template	Boolean	Whether this should be used as default template for new users.	-	O	-
country_id	Integer	The ID of a country for which that template should be available. These IDs can be retrieved from "*GET countries/get".*	-	O	-

Nodes To Users

Supported actions:

Method	URI	Description
GET	nodesToUsers/get/	Read existing entries
PUT	nodesToUsers/update/	Update an existing entry

Description:

This resource maps relations between users and project/tasks. New combinations will be added automatically, you will just have to edit their properties if you want to.

Parameters:

Name	Type	Description	C	R	U
node	Integer	The ID of the task or project.	-	O	R
user	Integer	The ID of the user.	-	O	R
is_favourite	Boolean		-	O	O
is_todo	Boolean		-	O	O
access	Boolean		-	O	O
target_duration	Integer		-	O	O
revenue_per_hour	Decimal	*Precision: 6* *Scale: 2*	-	O	O

Departments

Supported actions:

Method	URI	Description
GET	departments/get/	Read existing entries

Description:

Your companies departments for organizational structuring.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	-
department_name	String	The human readable name of the department.	-	O	-
supervisor_id	Integer	The ID of the user who is supervising this department.	-	O	-
supervisor_assistant_id	Integer	The ID of the user who is assisting with supervising this department.	-	O	-
mother_id	Integer	The ID of an optional higher level department in a hierarchical structure.	-	O	-

Countries

Supported actions:

Method	URI	Description
GET	countries/get/	Read existing entries

Description:

All countries which are supported in TimeTac.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	-
name	String	The human readable name of the country.	-	O	-

Clients

Supported actions:

Method	URI	Description
GET	clients/get/	Read existing entries
POST	clients/create/	Create a new entry
PUT	clients/edit/	Update an existing entry
DELETE	clients/delete/:id/	Delete an existing entry. ":id" has to be a valid ID of an existing resource.

Description:

Clients for the company's customer relation management.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	R
company_name	String	*Max. length: 32*	O	O	O
active	Boolean		O	O	O
username	String	*Valuerange: [0-9a-zA-Z\.\@_\-]* *Length: 3 - 32*	O	O	O
address_1	String	*Max. Length: 128*	O	O	O
address_2	String	*Max. Length: 128*	O	O	O
zip	String	*Max. Length: 8*	O	O	O
city	String		O	O	O
country_id	Integer	The ID of a country. These IDs can be retrieved from "*GET countries/get*".	O	O	O
url	String	The URL of the clients company website.	O	O	O
customer_number	String	*Max. length: 32*	O	O	O
contact_person_salutation_id	Integer	He ID of the salutation to be used for the contact person. These IDs can be retrieved from *"GET salutations/get"*.	O	O	O
contact_person_firstname	String	*Max. length: 32*	O	O	O
contact_person_lastname	String	*Max. length: 64*	O	O	O
contact_person_department	String	*Max. length: 64*	O	O	O
phone_1	String	*Max. length: 64*	O	O	O
phone_2	String	*Max. length: 64*	O	O	O
email_address	String		O	O	O

Projects

Supported actions:

Method	URI	Description
GET	projects/get/	Read existing entries
POST	projects/create/	Create a new entry
PUT	projects/edit/	Update an existing entry
DELETE	projects/delete/:id/	Delete an existing entry. ":id" has to be a valid ID of an existing resource.

Description:

Projects help you organize your companies' workload into a hierarchical structure. Each project can be divided into further subprojects/tasks, and augmented with different kind of metadata where necessary.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	R
mother_id	Integer	The ID of the higher level project for a hierarchical structure.	R	O	O
view_id	String	The numerical ordering of the whole hierarchy path.	-	O	-
sort_order	Integer	The numerical order of this project under its mother project.	R	O	O
node_path	String	The hierarchy path from the root up to and including this project. A path will be a comma separated list of all project IDs.	-	O	-
ultimate_mother_id	Integer	The ID of the project at the root of the path hierarchy.	-	O	-
name	String	*Max. length: 128*	R	O	O
is_done	Boolean		O	O	O
view_order	Decimal	*Precision: 40* *Scale: 30*	-	O	-
duration	Integer		-	O	-
target_duration	Integer		O	O	O
begin	Date		O	O	O
deadline	Date		O	O	O
notes	String		O	O	O
client_id	Integer	The ID of the client to which this project belongs. These IDs can be retrieved by *"GET clients/get"*.	O	O	O
t_iv_1	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_2	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_3	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O

t_iv_4	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_5	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_6	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
is_restricted	Boolean		O	O	O
project_leader_id	Integer	The ID of the user that should be the project leader.	O	O	O
budget	Integer		O	O	O

Tasks

Supported actions:

Method	URI	Description
GET	tasks/get/	Read existing entries
POST	tasks/create/	Create a new entry
PUT	tasks/edit/	Update an existing entry
DELETE	tasks/delete/:id/	Delete an existing entry. ":id" has to be a valid ID of an existing resource.

Description:

Tasks help you further organize your projects and timetrackings into smaller chunks that can be managed individually.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	R
mother_id	Integer	The ID of the higher level project for a hierarchical structure.	R	O	O
view_id	String	The numerical ordering of the whole hierarchy path.	-	O	-
sort_order	Integer	The numerical order of this project under its mother project.	R	O	O
node_path	String	The hierarchy path from the root up to and including this project. A path will be a comma separated list of all project IDs.	-	O	-
ultimate_mother_id	Integer	The ID of the project at the root of the path hierarchy.	-	O	-
name	String	*Max. length: 128*	R	O	O
is_done	Boolean		O	O	O
view_order	Decimal	*Precision: 40* *Scale: 30*	-	O	-
duration	Integer		-	O	-
target_duration	Integer		O	O	O
begin	Date		O	O	O
deadline	Date		O	O	O
notes	String		O	O	O
client_id	Integer	The ID of the client to which this project belongs. These IDs can be retrieved by *"GET clients/get"*.	O	O	O
t_iv_1	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_2	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_3	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O

t_iv_4	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_5	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_6	String	A custom field which application can be defined on a per account basis. *Max. length: 32*	O	O	O
is_startable	Boolean		O	O	O
is_billable	Boolean		O	O	O
is_nonworking	Boolean		O	O	O
is_paid_non_working	Boolean		O	O	O
internal_cost_per_hour	Decimal		O	O	O
revenue_per_hour	Decimal		O	O	O
skill_id	Integer	The ID of the skill that is related to this task. These IDs can be retrieved by *"GET skills/get"*	O	O	O
priority	Integer	 *Valuerange: 0 - 4*	O	O	O

Timestamps

Supported actions:

Method	URI	Description
GET	timestamps/get/	Read existing entries
POST	timestamps/create/	Create a new entry This should mainly be used to create a complete timetracking resource, for a shorter and more convenient method to start or stop a timer for a user see chapter “Timer”
PUT	timestamps/edit/	Update an existing entry
DELETE	timestamps/delete/:id/	Delete an existing entry. “:id” has to be a valid ID of an existing resource.

Description:

Timestamps are the core of TimeTac, they track the time spent for every user on every task and project and help you collect additional data about the workload that has been done during a day.

Because timestamps need a lot of backend calculations to be verified and validated, these resources are being processed asynchronously in a background task. As long as the request was syntactically correct the API client will always receive a success response. If the resource could not be inserted, the user account related to the timestamp will be notified inside the TimeTac application, there he will have to review the data for this entry and decide for himself if he wants to correct the error or discard it.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	R
user_id	Integer	The ID of the user this timer belongs to. These IDs can be retrieved by “GET users/get”	R	O	-
task_id	Integer	The ID of the task/project the timetrackings belongs to. These IDs can be retrieved by *“GET projects/get”* or *“GET tasks/get”*	R	O	O
start_time	DateTime	The date and time at which the timer should start. Leave it empty to emulate a leaving booking (type = 2).	R	O	O
end_time	DateTime	The date and time at which the timer should start. Leave it empty to emulate a leaving booking (type = 2).	R	O	O
duration	Integer		-	O	-
status	Integer	A flag mich marks the status of a timetracking: • 1 = still running • 2 = ended	R	O	O
start_ip	String	*Max. length: 16*	O	O	O
end_ip	String	*Max. length: 16*	O	O	O
is_billable	Boolean		O	O	O
notes	String		O	O	O
input_type	Integer	Marks from which input device a timetracking originated: • 1 = web app live • 2 = mobile app	O	O	O

		• 3 = subsequent post • 4 = system generated • 5 = RFID Terminal • 6 = TimeTac API			
t_iv_1	String	A custom field on task/project level which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_2	String	A custom field on task/project level which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_3	String	A custom field on task/project level which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_4	String	A custom field on task/project level which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_5	String	A custom field on task/project level which application can be defined on a per account basis. *Max. length: 32*	O	O	O
t_iv_6	String	A custom field on task/project level which application can be defined on a per account basis. *Max. length: 32*	O	O	O
u_iv_1	String	A custom field on user level which application can be defined on a per account basis. *Max. length: 32*	O	O	O
u_iv_2	String	A custom field on user level which application can be defined on a per account basis. *Max. length: 32*	O	O	O
u_iv_3	String	A custom field on user level which application can be defined on a per account basis. *Max. length: 32*	O	O	O
u_iv_4	String	A custom field on user level which application can be defined on a per account basis. *Max. length: 32*	O	O	O
u_iv_5	String	A custom field on user level which application can be defined on a per account basis. *Max. length: 32*	O	O	O
u_iv_6	String	A custom field on user level which application can be defined on a per account basis. *Max. length: 32*	O	O	O
type	Integer	The type of timetracking action you want to perform: • 0 = status ping • 1 = arriving • 2 = leaving	R	-	-

		• 3 = absence • 4 = complete dataset • 5 = dataset update			
timezone	String	The timezone in which the start or end time was recorded: Timezones are defined in a format of “<Continent>/<City>”, where the city should be the nearest major city that is supported. For a list of supported timezones see http://php.net/manual/en/timezones.php	O	-	-
approved_by_admin	Boolean	Whether the admin has approved time tracking entry (no further changes possible).	-	O	-
status_invoicing	Boolean	The status id of the invoice: • 14: Not billed • 15: Ready for billing • 16: Billed • 19: Paid	-	O	-
invoice_id	Integer	The ID of the associated invoice in TimeTac	-	O	-

Timer

Supported actions:

Method	URI	Description
POST	timer/start/	Start a new timer
POST	timer/stop/	Stop a running timer

Description:

Timer are simplified timestamps for an easier use to start or stop a new timeslot.

Because timers need a lot of backend calculations to be verified and validated, these resources are being processed asynchronously in a background task. An API client will always receive a success response, as long as the request was syntactically correct. If the timer could not be inserted, the user account related to the timer will be notified inside the TimeTac application, there he will have to review your timer data and decide for himself if he wants to correct the error or discard it.

Parameters:

Name	Type	Description	C	R	U
user_id	Integer	The ID of the user this timer belongs to. These IDs can be retrieved by “GET users/get”	R	-	-
task_id	Integer	The ID of the task/project the timetrackings belongs to. These IDs can be retrieved by *“GET projects/get”* or *“GET tasks/get”*	R	-	-
timestamp	DateTime	The date and time at which the timer should start.	R	-	-
notes	String		O	-	-
timezone	String	The timezone in which the timestamp was recorded: Timezones are defined in a format of “<Continent>/<City>”, where the city should be the nearest major city that is supported. For a list of supported timezones see http://php.net/manual/en/timezones.php	O	-	-

Messages

Supported actions:

Method	URI	Description
GET	messages/get/	Read existing entries
POST	messages/create/	Create a new entry This should mainly be used to create a complete timetracking resource, for a shorter and more convenient method to start or stop a timer for a user see chapter “Timer”
PUT	messages/edit/	Update an existing entry
DELETE	messages/delete/:id/	Delete an existing entry. “:id” has to be a valid ID of an existing resource.

Description:

Messages from individual Users, visible for every member of the company account in the Dashboard.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	R
sender_id	Integer	The ID of the user this message belongs to. These IDs can be retrieved by “GET users/get”	R	O	O
message	Integer	The text of the message that should be displayed.	R	O	O
timestamp	DateTime	The date and time at which the message was sent. If left empty on an create call, the current Date and Time will be automatically set.	O	O	O

Timesheetaccounting

Supported actions:

Method	URI	Description
GET	timesheetaccounting/get/	Read existing entries

Description:

Timesheet accounting, every user and day has its own entry with all the key figures summarized for that date.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	-
user_id	Integer	The ID of the user this timesheet belongs to. These IDs can be retrieved by “GET users/get”.	-	O	-
date	Date	The date of the day the entry belongs to.	-	O	-
Various other key figures available for read access. Please contact our support if you have questions regarding the meaning of specific fields for your account.			-	O	-

Absence Days

Supported actions:

Method	URI	Description
GET	absencedays/get/	Read existing entries

Description:

Absence days for every user in the system.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	-
user_id	Integer	The ID of the user this absence belongs to. These IDs can be retrieved by “GET users/get”.	-	O	-
date	Date	The date of the day the entry belongs to.	-	O	-
type	String	The type of absence	-	O	-
subtype	String	The subtype of the absence	-	O	-
other_paid_leave_id	Integer	The id of the leave type	-	O	-
value	Decimal	The length of the absence in <value_unit>	-	O	-
value_unit	String	The unit the value is measured in, d (days) or hs (hours)	-	O	-
request_id	Integer	The ID of the holiday request this absence belongs to. These IDs can be retrieved by “GET holidayrequests/get”.	-	O	-
Status	Integer	The status of the absence. • 0 = pending • 1 = approved	-	O	-
updated	DateTime	The timestamp this entry was edited	-	O	-
username	String	The username of the user this absence belongs to	-	O	-

Holiday Requests

Supported actions:

Method	URI	Description
GET	holidayrequests/get/	Read existing entries
POST	holidayrequests/create/	Create a new entry

Description:

Holiday requests. Most absences are requested by users in TimeTac and later on approved/declined/cancelled. This resources stores all requests in the system.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	-
user_id	Integer	The ID of the user this request belongs to. These IDs can be retrieved by “GET users/get”.	R	O	-
date	Date	The date of the day the entry belongs to.	-	O	-
type	Integer	The type of request: • 1 = holiday • 2 = overtime_reduction • 3 = other paid leaves	R	O	-
other_paid_leave_id	Integer	The id of the leave type	O	O	-
approved_by_user	Integer	The ID of the user this request was approved by. These IDs can be retrieved by “GET users/get”.	-	O	-
Created	DateTime	The timestamp this request was issued at.	-	O	-
request_comment	String	An optional comment added by the requesting user	O	O	-
Status	Integer	The status of the absence. • 0 = pending • 1 = approved	O	O	-
granted_user_id	Integer	The ID of the user this request was approved by. These IDs can be retrieved by “GET users/get”.	-	O	-
granted_timestamp	DateTime	The timestamp the request was granted at.	-	O	-
granted_comment	String	An optjonal comment added by the user who approved/declined the request.	-	O	-
from_date	Date	The start date of the requested timeframe	R	O	-
to_date	Date	The end date of the requested timeframe	R	O	-
duration	Decimal	The requested duration of workdays/time	O	O	-
duration_unit	String	The unit the duration was measured in	-	O	-
Updated	DateTime	The timestamp the request was edited.	-	O	-

Timestamps Changelog

Supported actions:

Method	URI	Description
GET	timestamps_changelog/get/	Read existing entries

Description:

Changelog for the timestamps resource.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	-
change_timestamp	DateTime	The timestamp this entry was edited.	-	O	-
change_type	String	The type of change that was made.	-	O	-
change_reason_id	Integer	The reason why the timer was changed: • 0: Default • 1: Automatic break calculations • 2: Automatic timer rounding	-	O	-
change_user_id	Integer	The ID of the user that changed this timestamp. These IDs can be retrieved by “GET users/get”.	-	O	-
tracker_id	Integer	The ID of the timestam belongs to. These IDs can be retrieved by “GET timestamps/get”.	-	O	-
user_id	Integer	The ID of the user this timestam belongs to. These IDs can be retrieved by “GET users/get”.	-	O	-
task_id	Integer	The ID of the task this timestam belongs to. These IDs can be retrieved by “GET tasks/get”.	-	O	-
start_time	DateTime	The timestamp the timer started on before the change.	-	O	-
new_start_time	DateTime	The timestamp the timer started on after the change.	-	O	-
end_time	DateTime	The timestamp the timer ended on before the change.	-	O	-
new_end_time	DateTime	The timestamp the timer ended on after the change.	-	O	-
duration	Integer	The duration of the timer	-	O	-

Teams

Supported actions:

Method	URI	Description
GET	teams/get/	Read existing entries
POST	teams/create/	Create a new entry
PUT	teams/edit/	Update an existing entry
DELETE	teams/delete/:id/	Delete an existing entry. “:id” has to be a valid ID of an existing resource.

Description:

Teams group different employees together. In contrast to departments are teams and employees in a many-to-many relation, so that each employee can be part of multiple teams.

Parameters:

Name	Type	Description	C	R	U
id	Integer	The ID of the existing dataset.	-	O	R
name	String	The name of the team that is shown to the users.	R	O	O
team_leader_id	Integer	The ID of the user that will act as the team leader. These IDs can be retrieved by “GET users/get”.	R	O	O
project_id	Integer	The ID of the project this team belongs to, if it is a project team.	O	O	O
data_changed	DateTime	The timestamp this entry was modified.	-	O	-
current_team_members	String	A concatenated list of all user IDs that belong to this team. Generated automatically for performance reasons.	-	O	-
use_pm_user_with_teamplaner_access_table	Integer	Automatic flag if there are users in the team using the team planner feature.	-	O	-
current_user_with_teamplaner_access	String	Concatenated list of users with the teamplanner feature enabled.	-	O	-

Teammembers

Supported actions:

Method	URI	Description
GET	teammembers/get/	Read existing entries
POST	teammembers/create/	Create a new entry
DELETE	teammembers/delete/:team_id/:user_id/	Delete an existing entry. ":parameter" have to be valid IDs of existing resources.

Description:

Many-to-many relation mapping for users and teams.

This resource does not support the "update" action, you will have to delete the old mapping and add a new one.

Parameters:

Name	Type	Description	C	R	U
team_id	Integer	The ID of the team the user belongs to. These IDs can be retrieved by "GET teams/get".	R	O	-
user_id	Integer	The ID of the user that should be mapped to a team. These IDs can be retrieved by "GET users/get".	R	O	-