HelixTeamHub

Helix TeamHub Developer Guide

2020.1 Enterprise

April 2020

PERFORCE

www.perforce.com

Contents

How to Use this Guide

This manual is intended for developers tasked with developing a Helix TeamHub client for managing version control repositories. It includes a tutorial, migration instructions from older API versions, and a full API reference.

This section provides information on typographical conventions, feedback options, and additional documentation.

Syntax conventions

Helix documentation uses the following syntax conventions to describe command line syntax.

Notation	Meaning
`literal`	Must be used in the command exactly as shown.
italics	A parameter for which you must supply specific information. For example, for a *serverid* parameter, supply the ID of the server.
[`-f`]	The enclosed elements are optional. Omit the brackets when you compose the command.
`...`	Previous argument can be repeated. - `p4 [g-opts] streamlog [ -l -L -t -m max ]` ***`stream1`*** `...` means `1` or more stream arguments separated by a space - See also the use on `...` in Command alias syntax in the *Helix Core P4 Command Reference* **Tip** `...` has a different meaning for directories. See Wildcards in the *Helix Core P4 Command Reference*.
element1 | *element2*	Either *element1* or *element2* is required.

Feedback

How can we improve this manual? Email us at manual@perforce.com.

Other documentation

See https://www.perforce.com/support/self-service-resources/documentation.

What's new in this guide for this release

For a complete list of what's new in this release, see the *Helix TeamHub Release Notes*.

Getting Started

This tutorial will take you through the development of a command line Helix TeamHub client for managing version control repositories. We hope to show you how everything in Helix TeamHub can be accessed through our RESTful APIs, using plain HTTPS calls with JSON payloads.

The tool will be implemented in Ruby. We are working on making tutorials available for other programming languages. In the meantime, we hope you'll be able to follow along even if you're working in some other language.

The Tool

The tool we're building can be used for managing source code repositories: Listing them, creating them, or destroying them.

Listing all repositories of a project

```
repo list my_project
```

Creating a repository in a project:

```
repo create my_project a_repo git
```

Deleting a repository from a project

```
repo delete my_project a_repo
```

Prerequisites

The tutorial is written assuming a Unix based operating system, such as Linux or Mac OS X. However, there is nothing inherently Unix dependent here, so you should be able to apply the tutorial on Windows as well.

You will need to have Ruby 1.9 or later installed. Follow the official installation documentation to get Ruby up and running on your machine.

We'll be using two Ruby gems to help with the lower level details:

- rest-client for making HTTP(S) calls to the API
- json for JSON parsing and serialization

You can get these installed with the `gem` command once you have Ruby installed:

```
gem install rest-client json --no-rdoc --no-ri
```

> **Note**
> Depending on how you installed Ruby you may need to use `sudo` to install Gems.

Set Up The Helix TeamHub Endpoint URL

To make the rest of our work easier, let's put the base URL of the Helix TeamHub API to an environment variable.

If you are using the Helix TeamHub Cloud, this URL is `https://helixteamhub.cloud/api`. If you are using an on-premises installation, replace `helixteamhub.com` with the hostname or IP of your internal Helix TeamHub instance.

Set up an environment variable `HTH_ENDPOINT` with this value. The best way to do this depends on your operating system and shell. For example, on Mac OS X and most Linux installations you can put the following line in `~/.bash_profile`:

```
export HTH_ENDPOINT="https://helixteamhub.cloud/api"
```

Set Up Keys

In order to access Helix TeamHub from tools such as this one, we need two keys. They allow for fine-grained access control, and for making sure whoever's running the tool has the permission to execute the action under question.

We are going to put all keys into environment variables so that the script – as well as any other scripts you want to develop – is easy to run.

The Company And Account Keys

The *Company Key* identifies which company within Helix TeamHub we are dealing with. Most people will only ever have access to a single company's data, but this is not always the case. Because every company in Helix TeamHub has it's own "universe" with unique user and project identifiers, we need to use the Company Key to scope each request to a specific company.

The *Account Key* identifies who exactly is accessing Helix TeamHub (or on whose behalf we're doing it). Each Helix TeamHub user and bot has a unique account key.

A convenient way to acquire both of these keys is to make an authentication request to the Helix TeamHub API. What you need for it is your Helix TeamHub access credentials (company, login (email or user ID) and password). Run the following command, replacing your own company, login and password (as well as the Helix TeamHub URL if you have an on-premises installation):

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/vnd.hth.v1" \
  -d '{"company": "YOUR COMPANY ID", "login": "YOUR LOGIN", "password":
"YOUR PASSWORD" }' \
  https://helixteamhub.cloud/api/account/sessions
```

The command should print out the JSON response returned by Helix TeamHub. Included in it are the keys for the company and the account. Put the values of those to environment variables:

```
export HTH_COMPANY_KEY=value of company_key
export HTH_ACCOUNT_KEY=value of account_key
```

Implement The Script Structure

Now we are ready to start coding our tool. We are going to put everything to a single executable Ruby file. It is a good idea to put it somewhere in your `PATH` so that it's easy to execute. Assuming you have a directory `~/bin` for this purpose, let's create the script there and make it executable:

```
touch ~/bin/repo
chmod +x ~/bin/repo
```

Next, open the newly created file in your favorite Ruby or text editor. At this point we'll add in everything except the actual Helix TeamHub API calls. Set the file's contents as follows (we will be going through them step by step in a moment):

```
#!/usr/bin/env ruby

require 'rest_client'
require 'json'

module HelixTeamHub
  class Client

    def initialize(endpoint, account_key, company_key)
      @endpoint    = endpoint
      @account_key = account_key
      @company_key = company_key
    end

    def list_repos(project_id)
      raise "No project given"         if project_id.nil?

    end
    def create_repo(project_id, repo_id, repo_type)
      raise "No project given"         if project_id.nil?
      raise "No repository given"      if repo_id.nil?
      raise "No repository type given" if repo_type.nil?
      raise "Invalid repository type"  unless %w(git mercurial
subversion).include?(repo_type)

    end

    def delete_repo(project_id, repo_id)
      raise "No project given"         if project_id.nil?
      raise "No repository gien"       if repo_id.nil?

    end

  end
end

command, project_id, repo_id, repo_type = ARGV

client = HelixTeamHub::Client.new(ENV['HTH_ENDPOINT'],
                           ENV['HTH_ACCOUNT_KEY'],
                           ENV['HTH_COMPANY_KEY'])

case command
when 'list'   then client.list_repos(project_id)
when 'create' then client.create_repo(project_id, repo_id, repo_type)
when 'delete' then client.delete_repo(project_id, repo_id)
else               puts "Unknown command: #{command}"
end
```

- On *line 1* we define the shebang, letting the shell know it should use Ruby to run this script.
- On *lines 3-4* we require the Ruby gems we need for the API calls.

- On *lines 5-6* we begin a Ruby class definition for a Helix TeamHub client class we'll be using later in the script: `HelixTeamHub::Client`.
- On *lines 9-14* we define the class constructor, which takes the Helix TeamHub API endpoint and the two keys. It stores everything as instance variables.
- On *lines 16-19* we define a method for listing the repositories of a project. It takes the project identifier as an argument, and validates that it is not `nil`.
- On *lines 21-27* we define a method for creating a repository to a project. It takes the project identifier, the repository identifier, and the reposioty type as arguments. It validates that each of the arguments is present, and that the repository type is one of `git`, `mercurial`, or `subversion`.
- On *lines 29-33* we define a method for deleting a repository from a project. It takes the project identifier and the repository identifier as arguments, and validates that both of them are present.
- On *lines 35-36* we close the `HelixTeamHub::Client` class definition
- On *line 38* we grab the four command line arguments that may have been given to the script: The command to execute, the project identifier in question, the repository identifier, and the repository type. Not all commands need all of these arguments, in which case their value will be `nil`.
- On *lines 40-43* we create an instance of the `HelixTeamHub::Client` class. To its constructor we pass the Helix TeamHub endpoint URL and the two keys, which we get from the environment variables we set up earlier.
- On *lines 45-50* we finally execute the command that was given, by calling the matching method from the `HelixTeamHub::Client` object. An error message is printed if we can't recognize the command.

Implement The List Command

To get a list of repositories, we can call the Helix TeamHub API operation `GET /projects/:project_id/repositories`. Let's do exactly that, using the rest-client library we have installed as a Gem and required. Update the `list_repos` method so that it looks like this:

```
    def list_repos(project_id)
      raise "No project given"         if project_id.nil?

      response = RestClient.get "#{@endpoint}/projects/#{project_
id}/repositories",
                                authorization: auth_header(),
                                accept: :json
      JSON.parse(response)["results"].each do

Table Row Outside Table:
 Table Cell Outside Table: repo
        print_repo(repo)
```

```
  end
end
```

- On *lines 4-6* we make a GET request to the repositories URL, based on the configured Helix TeamHub API endpoint. We include two HTTP headers: `Authorization` for letting Helix TeamHub know who we are, and `Accept` for letting Helix TeamHub know we want back JSON data.
- On *lines 7-9* we parse the returned JSON data to a Ruby data structure, and then iterate over the results (assuming it is an array of repositories). The information of each repository is printed out.

In this code we used two helper methods we haven't defined yet: `auth_header` for defining the contents of the `Authorization` header, and `print_repo` for printing out the information for a repository. Let's define those now.

Within the class definition for `HelixTeamHub::Client`, add two more private methods:

```
def auth_header
  %(hth.company_key="#{@company_key}",account_key="#{@account_key}")
end


def print_repo(repo)
  puts repo["id"]
  puts "=" * 20
  puts "Type:         #{repo["type"]}"
  puts "SSH access:   #{repo["ssh_url"]}"
  puts "HTTP(s) access: #{repo["http_url"]}"
  puts
end
```

- On *lines 1-3* we define the `auth_header` method. It constructs a value for the `Authorization` header that Helix TeamHub recognizes. The value begins with the string `hth`, which is followed by the two API keys.
- On *lines 5-12* we define a method for printing out a repository. It prints the repository identifier, the type of the repository, and the access URLs for the SSH and HTTPS protocols.

Implement The Create Command

When creating a repository, we invoke the same endpoint as for listing repositories, but this time we use the POST method instead of GET: `POST /projects/:project_id/repositories`. This time we also provide a JSON payload with the information for the new repository.

Update the `create_repo` method so that it looks like this:

```
    def create_repo(project_id, repo_id, repo_type)
      raise "No project given"          if project_id.nil?
      raise "No repository given"       if repo_id.nil?
      raise "No repository type given" if repo_type.nil?
      raise "Invalid repository type"  unless %w(git mercurial
subversion).include?(repo_type)

      response = RestClient.post "#{@endpoint}/projects/#{project_
id}/repositories",
                                 {repository: {id: repo_id, type: repo_
type}}.to_json,
                                 authorization: auth_header(),
                                 content_type: :json,
                                 accept: :json
      print_repo(JSON.parse(response))
    end
```

- On *lines 7-11* we make the call to the API. We specify the id and type of the repository in the payload, and indicate that both the POST data and the response should be JSON, by setting the **`Content-Type`** and **`Accept`** headers. We also include the same **`Authorization`** header as before.
- On *line 12* we parse the response, assuming it's a JSON representation of the newly created repository. We print it out using the same method as we used in the listing.

Implement The Delete Command

Finally, let's implement the method for deleting a repository. This will be a DELETE request to the repository's own API resource: **`DELETE /projects/:project_id/repositories/:repository_id`**.

Update the **`delete_repo`** method so that it looks like this:

```
    def delete_repo(project_id, repo_id)
      raise "No project given"          if project_id.nil?
      raise "No repository gien"        if repo_id.nil?

      RestClient.delete "#{@endpoint}/projects/#{project_
id}/repositories/#{repo_id}",
                        authorization: auth_header()
      puts "Deleted"
    end
```

- On *lines 5-6* we make the call to the API. This time there is no content in the request, and we're not really interested in the response body either, so the only header we include is **`Authorization`**.
- On *line 7* we print out a simple message letting the user know what's happening.

Test Drive

We're done with our tool implementation! Let's take it for a test drive.

The tool assumes there's an existing Helix TeamHub Project within which the operations are executed. You may want to create a separate sandbox project just for testing the tool (it's free), or you can use an existing project. In either case, the script will need a *project id*. Here's how you can obtain it:

- For new projects, the value you need is the same as the one you'll put in the *Identifier* field of the *New Project* view.
- For already existing projects, you can see the project identifier, among other places, in the browser address when you have navigated to the project: *https://helixteamhub.cloud/COMPANY_ID/projects/PROJECT_ID/activity.

When you have a project set up, and know it's identifier, you can start playing with the script to list, create, and delete repositories:

```
./repo list my-test-project

./repo create my-test-project a_git_repo git
a_git_repo
====================
Type:          git
SSH access:    hth@helixteamhub.cloud:acme/projects/my-test-
project/repositories/git/a_git_repo
HTTPS access: https://teroparv@helixteamhub.cloud/acme/projects/my-test-
project/repositories/git/a_git_repo

./repo create my-test-project a_hg_repo mercurial
a_hg_repo
====================
Type:          mercurial
SSH access:    ssh://hth@helixteamhub.cloud/acme/projects/my-test-
```

```
project/repositories/mercurial/a_hg_repo
HTTPS access:

./repo list my-test-project
a_hg_repo
====================
Type:         mercurial
SSH access:   ssh://hth@helixteamhub.cloud/acme/projects/my-test-
project/repositories/mercurial/a_hg_repo
HTTPS access:

a_git_repo
====================
Type:         git
SSH access:   hth@helixteamhub.cloud:acme/projects/my-test-
project/repositories/git/a_git_repo
HTTPS access: https://teroparv@helixteamhub.cloud/acme/projects/my-test-
project/repositories/git/a_git_repo

./repo delete my-test-project a_git_repo
Deleted

./repo list my-test-project
a_hg_repo
====================
Type:         mercurial
SSH access:   ssh://hth@helixteamhub.cloud/acme/projects/my-test-
project/repositories/mercurial/a_hg_repo
HTTPS access:
```

Conclusion

In this tutorial you've learned to work with the Helix TeamHub API, and particularly the version control repository management.

All data in Helix TeamHub has similar APIs available, and you should be able to apply what you have learned to everything else. To get an idea of what you can do, see the API Documentation.

Helix TeamHub API Changes

The changelog to help you migrate from v1 to the experimental v2:

- "Changes from v1 to v2" below

Bots migration

To migrate old bot accounts (created with API v1) into new ones (available through the UI and API v2), please run the following script:

```
cd /opt/hth/application/backend/current/

bundle exec rails runner "Helix
TeamHub::Services::BotsMigration.new.migrate"
```

This script is reusable, so it can be run when necessary.

Changes from v1 to v2

General

- API communication follows JSON API specification standards.
- Bot account scheme was revised as a part of Helix TeamHub 3.4.0 release. Since that version, old bot accounts are no longer shown in the UI and can be listed only via API v1. However, there is a reusable migration script available for on-premises installations, which converts old bot accounts into new ones. If you use the cloud installation, contact us.

New endpoints in v2

- "Bot" on page 244
- "Default Reviewer Group" on page 254
- "Project" on page 259
- "Repository Bot" on page 262

Helix TeamHub API v1

This is the home of stable Helix TeamHub RESTful API documentation. There is also another API version documented - the experimental v2.

On these pages you'll find everything you need to know about the Helix TeamHub REST APIs. These give you access to all the data Helix TeamHub stores, and you will be able to build any clients and integrations you need.

This page introduces the general concepts used throughout all Helix TeamHub APIs, from authentication to the calling conventions and the error codes. To get information about specific types of data, use the navigation on the right.

API version

By default, all requests to Helix TeamHub APIs use the latest version, however, we recommend you to explicitly request the wanted version via the Accept header.

```
Accept: application/vnd.hth.v1
```

API stability

Helix TeamHub APIs follow semantic versioning but only the major version number is used. Bug fixes and backwards compatible changes can be added to the same major version. Major Helix TeamHub releases may deprecate older version of APIs and introduce a new version, e.g.

- Helix TeamHub 2.x supports v1.
- Helix TeamHub 3.x may introduce v2 and deprecate v1, or still continue supporting v1.

Authentication

All requests (except login/logout) require the following set of keys to be present:

- An *Account key* - representing the user on whose behalf actions are taken.
- A *Company key* - representing the company into which all actions will be scoped.

If you don't know your Company key and Account key, you can acquire them by authenticating with your company ID, login (email or user ID) and password:

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Accept: application/vnd.hth.v1" \
  -d '{"company": "acme", "login": "chuck@norris.com", "password":
"ChuckN0rr!5" }' \
  https://helixteamhub.cloud/api/account/sessions
```

You will find the **`account_key`**, **`company_key`** and details of the account (that is either User or Collaborator) and company in the response:

```
{
  "api_status": 201,
  "api_timestamp": "2014-12-02T12:26:01Z",
  "account_key": "ab187f75aa6322eff512bfe8ef97b292",
  "company_key": "553ae10f609b0a0d8ac9ebcceefb8d21",
  "account": {
    ...
  },
  "company": {
    ...
  }
}
```

You can use this same mechanism to implement Helix TeamHub authentication into your own tools and scripts: Prompt the user for their company, login (email or user ID) and password, and then execute the authentication request on their behalf. Then you can execute any other requests for them by using their account key and company key.

Finally, to logout the user permanently, you can invalidate the account_key with:

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \

https://helixteamhub.cloud/api/account/sessions/ab187f75aa6322eff512bfe8ef
97b292
```

Authorization

Authorization is always done before you attempt to do anything via our APIs. However, sometimes you end up in a situation where you would like to know beforehand what you or the current user of your application has privileges to do in Helix TeamHub, e.g. to build a better user experience by hiding specific UI components and whatnot.

Luckily, Helix TeamHub has just what you need to accomplish this; we call them privileges. To read the privileges for a collection of objects or for a specific object is trivial and nearly all of our endpoints support the behavior. Basically all you need to do is to append one query parameter, namely **`privileges`**, to your RESTful requests and you're done.

Please note that the response for a collection of objects and for a specific object slightly differs. This is due to the fact that there isn't a point in asking whether or not one could update or delete a collection of objects, as such operations are always done against a specific object. To clear it up, we've provided you with some examples, you'll find them below.

Can I create and/or read projects in my company?

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects?privileges
```

```
{
  "api_status": 200,
  "api_timestamp": "2014-01-10T08:05:03Z",
  "create": true,
  "read": true
}
```

Can I read, update, and/or delete the "platform" project in my company?

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/platform?privileges
```

```
{
  "api_status": 200,
  "api_timestamp": "2014-01-10T08:05:03Z",
  "read": true,
  "update": true,
  "delete": true
}
```

Data objects

Here are the data objects you can access through the Helix TeamHub APIs, and the operations you can execute on them.

Each object, whatever its type, has an *id* attribute, which is its identifier and unique within the object type. That identifier can be used to refer to the object in other contexts.

Object	Create	Show	List	Update	Destroy
Attachment	✓	✓	✓	✓	✓
Branch	✓	✓	✓	✓	✓
Bookmark		✓	✓		
Code Review	✓	✓	✓	✓	✓
Collaborator	✓	✓	✓	✓	✓
Comment	✓	✓	✓	✓	✓
Commit	✓	✓	✓		
Company		✓		✓	✓
Event	✓	✓	✓	✓	
Group	✓	✓	✓	✓	✓

Object	Create	Show	List	Update	Destroy
Group Member	✓	✓	✓	✓	✓
Hook	✓	✓	✓	✓	✓
Hook Service		✓	✓		
License	✓	✓			
Milestone	✓	✓	✓	✓	✓
Password Recovery	✓	✓		✓	
Project	✓	✓	✓	✓	✓
Project Collaborator	✓	✓	✓	✓	✓
Project Group	✓	✓	✓	✓	✓
Project Member			✓		
Project User	✓	✓	✓	✓	✓
Repository	✓	✓	✓	✓	✓
Repository Collaborator	✓	✓	✓	✓	✓
Repository Group	✓	✓	✓	✓	✓
Repository User	✓	✓	✓	✓	✓
SSH Key	✓	✓	✓	✓	✓
Tag		✓	✓		✓
Task	✓	✓	✓	✓	✓
User	✓	✓	✓	✓	✓

HTTP methods

The API endpoints are used with the following HTTP methods:

- GET - Used for retrieving a resource.
- POST - Used for creating a resource.
- PUT - Used for updating or replacing resources.
- DELETE - Used for deleting resources.

Parameter nesting

All API endpoints support both nested and non-nested request parameters for POST and PUT requests. What this means is that you can do

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "norris", "email": "norris@helixteamhub.com" }' \
  https://helixteamhub.cloud/api/users
```

instead of

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "user": { "id": "norris", "email": "norris@helixteamhub.com" } }'
\
  https://helixteamhub.cloud/api/users
```

While the latter example still works, consider the non-nested form the official one for v1.

Attribute types

The data objects may have attributes with the following types:

boolean

```
true
```

string

```
"Åbo Akademi"
```

integer

```
11
```

timestamp

A timestamp in the ISO-8601 format (always UTC). Milliseconds are not included.

```
"2007-11-20-T22:19:17Z"
```

array

A list of other values.

```
[
  "Chuck Norris",
  "Clark Kent"
]
```

object

Another, nested API object.

```
{
  id: "norris",
  first_name: "Chuck",
  ...
}
```

Common attributes

Every API object has the following attributes:

api_status

- Type: **`integer`**

The HTTP status code that describes the response status for this specific object.

When handling only one object, this is always the same as the status code in the HTTP header. However, with multiple objects each one might have a different status code. For instance, there may be a situation where half of a batch of objects could not be created.

```
api_status: 422
```

api_timestamp

- Type: `timestamp`

The point in time when the response was created. As the state of the API objects can change, this timestamps each request so you know how recent the information is.

Note that with heavy requests, the returned objects might have API timestamps that differ from each other.

```
api_timestamp: "2012-01-22T22:11:11Z"
```

Most of the API objects have the following attributes:

created_at

- Type: `timestamp`

The point in time when the object was created.

```
created_at: "2012-01-22T22:11:11Z"
```

updated_at

- Type: `timestamp`

The point in time when the object was last updated.

```
updated_at: "2012-01-22T22:11:11Z"
```

Custom attributes

Some of the API objects allow storing custom attributes within the object using the `properties` attribute.

- Type: `object`

```
"properties": {
  "string": "value",
  "array": [
    1,
    2,
    3
  ],
  "object": {
    "nested": "value"
  }
}
```

Attributes in deactivated objects

When you delete an object in Helix TeamHub, it will usually be a "soft delete", where the object is not actually physically deleted, just deactivated. (Notable exceptions to this rule are project users, project groups, groups, group members, hooks and SSH keys).

When an object has been deactivated, you are still able to access it through the *show* endpoint, but you may not modify or restore it. When an object is deactivated, its identifier is renamed. For instance, from *norris* to *norris-1234567*.

All deactivated objects have two extra attributes:

deleted_at

- Type: `timestamp`

Time of the deletion.

```
deleted_at: "2013-01-18T09:13:55Z"
```

old_id

- Type: `string`

Value of the original identifier before deletion.

```
old_id: "norris"
```

Error responses

Errors caused by API requests are identifiable by the HTTP status codes that are returned along with the responses. In an error situation, the API response includes the following attributes within the (invalid) API object:

api_status

- Type: **`integer`**

HTTP status code.

```
api_status: 404
```

api_message

- Type: **`string`**

HTTP status message.

```
api_message: "Not Found"
```

api_errors

- Type: **`object`**

Related validation errors, grouped by attribute. See the next section for details.

```
{
  email: {
    invalid: true,
    conflicts: ["password"]
  },
  password: {
    conflicts: ["email"]
  }
}
```

Validation errors

When you try to create or update one or more objects and some of them are invalid, the API responds with the HTTP status code **422 Unprocessable Entity**. The objects in the response have an `api_errors` attribute, which includes all the validation errors grouped by attribute.

Different field types may have different kinds of validation errors (see the object-specific references for details):

minimum

- Type: `integer`

The given value is too short (strings), too low (int), or doesn't contain enough objects (arrays). The response contains the minimum value required.

```
minimum: 10
```

maximum

- Type: `integer`

The given value is too long (strings), too high (int), or contains too many objects (arrays). The response contains the maximum value required.

```
maximum: 40
```

conflicts

- Type: `array`

The given value cannot be used with one or more of the other given values. In this case, these other attributes are also returned as incompatible. The array contains the names of the conflicting attributes.

```
conflicts: ["email"]
```

empty

- Type: `boolean`

The value for this attribute cannot be an empty string, 0 or false.

```
empty: true
```

invalid

- Type: **`boolean`**

The given value contains the characters that are not allowed, is of invalid type/format, or is otherwise illegal.

```
invalid: true
```

reserved

- Type: **`boolean`**

The given value is already in use or reserved for something else.

```
reserved: true
```

not_found

- Type: **`boolean`**

The referenced object was not found

```
not_found: true
```

locked

- Type: **`locked`**

The given value can not be changed. Depending on the object type, some other user with different privileges might be able to change the value.

```
locked: true
```

Limiting results

The results of all list actions may be limited for purposes of paging or progressive loading. The responses have the following form:

```
{
  "metadata": {
    "more_results": true,
    "next_offset": 2,
    "count": 2
  },
  "results": [
    {
     "key": "value"
    },
    {
     "key": "value"
    },
  ]
}
```

The result is an object with two keys:

- **`metadata`** is an object with information about the result counts. Within it, **`more_results`** defines whether there are more results to fetch or whether the current response has them all. **`count`** defines how many results there are in the current result set, and **`next_offset`** defines what the offset of the next request should be if you are paging or grouping the results.
- **`results`** is the actual array of objects.

To limit the returned results of a request, pass the following arguments as query parameters in the request:

limit

- Type: **`integer`**
- Default: **`100`**
- Maximum: **`10000`**

Limits the amount of results returned. Use this to speed up the request when you know you don't need more than N results.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  "https://helixteamhub.cloud/api/users?limit=3000"
```

offset

- Type: `integer`

Moves the starting point of the returned array from zero to the given number. Use this together with limit to implement pagination or other index-based result grouping.

before

- Type: `timestamp`

Limits returned data to include only objects that were updated before the given point in time.

after

- Type: `timestamp`

Limits returned data to include only objects that were updated after the given point in time.

active

- Type: `boolean`

By default only active objects are returned from the index endpoint. Use this flag to explicitly request deactivated objects.

Searching

search_term

- Type: `string`
- Allowed endpoint: `index`

A search term to look for. Use together with `search_fields`.

search_fields

- Type: Comma separated `string`
- Allowed endpoint: `index`

Optionally limit the attributes to search, by default all the searchable attributes of the object are used. Use together with `search_term`. Refer to the object specific references to see the supported searchable attributes.

Sorting results

To sort results, pass the following arguments as query parameters in your request:

sort

- Type: `string` of attribute name

The results are ordered naturally based on the value of the given attribute.

order

- Type: `string`
- Allowed values: `asc`, `desc`
- Default: `desc`

Sorts the results from low to high or high to low. Use together with `order`.

Child objects

When requesting objects that may have child objects related to them (for example, Projects may have Repositories), you can combine the fetching of those child objects to a single request. This can be done on several levels: You can count the child objects, get their IDs, or get them in full.

Refer to object-specific documentation for the possible child objects.

count

- Type: Comma separated `string`

Count the numbers of child objects the requested object has. Specify the names of the attributes.

- `count=repositories,users` will attach the following attributes to the returned objects:

```
repositories: 32,
users: 15
```

list

- Type: Comma separated `string`

List the IDs of child objects the requested object has. Specify the names of the attributes.

- `list=repositories` will attach the following attributes to the returned objects:

```
repositories: ["amiraali", "luotsi"]
```

include

- Type: Comma separated `string`

Return child objects in full. Specify the names of the attributes.

- `include=repositories` will attach the following attributes to the returned objects:

```
repositories: [
  {
    id: "amiraali",
    type: "git",
    ...
  },
  ...
]
```

expand

- Type: Comma separated `string`

Return linked objects in full. Specify the names of the attributes.

You can use expand in cases where the default behavior of the API is to specify an ID of a linked object, to expand the id to the linked object's full representation.

- `expand=subject` will attach the following attributes to the returned objects:

```
subject: {
  id: username,
  email: email@email.com,
  ...
}
```

Account Settings

Account Settings are user-specific preferences that contain settings for things like e-mail notification frequency.

- Attributes
- Operations
  - `GET /account/settings/notifications`
  - `PUT /account/settings/notifications`

Attributes

email

Interval settings for project-specific e-mail notifications. Possible `interval` values are `instant`, `daily`, `weekly`, and `never`. A valid Project ID must be provided with `project`.

- Type: `array` of `object`
- Required: `true`

```
[
  {
    "project": "luotsi",
    "interval": "instant"
  }
]
```

Operations

GET /account/settings/notifications

Returns all notification settings for the current API user.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/account/settings/notifications
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2014-10-06T14:59:05Z",
  "email": [
    {
      "project": "luotsi",
      "interval": "daily"
    },
    {
      "project": "devrel",
      "interval": "weekly"
    }
  ]
}
```

PUT /account/settings/notifications

Updates User Settings with the provided values.

Example request

```
curl -X PUT \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "email": [{ "project": "luotsi", "interval": "instant" }] }' \
  https://helixteamhub.cloud/api/account/settings/notifications
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2014-10-06T14:59:05Z",
  "email": [
    {
      "project": "luotsi",
      "interval": "instant"
    },
    {
      "project": "devrel",
      "interval": "weekly"
    }
  ]
}
```

Account

Account endpoint provides details of the authenticated account, which can be either User or Collaborator.

- Attributes
- Operations
    - `GET /account`
    - `GET /account/repositories`
    - `GET /account/projects`

For other account-specific endpoints, see:

- Account Settings
- Avatar
- Password Recovery
- SSH Key

Attributes

See attributes of User and Collaborator.

Operations

GET /account

Returns account details, either of User and Collaborator.

GET /account/repositories

Returns all repositories the current account has access to, with optional limit and offset parameters, as a metadata-results object.

GET /account/projects

Returns all projects the current account has access to, with optional limit and offset parameters, as a metadata-results object.

Attachment

Attachments are files related to other objects. Currently only Tasks, Code Reviews, Multi-Repo Code Reviews, and Comments can have attachments.

- Attributes
- Request Parameters
- Operations
  - `GET /attachments`
  - `GET /attachments/:id`
  - `POST /attachments`
  - `PUT /attachments/:id`
  - `DELETE /attachments/:id`

Attributes

id

Universally unique identifier (UUID).

- Type: `string`
- Immutable: `true`
- Unique: `true`

name

Original filename of the attachment.

- Type: `string`

mime

Detected MIME-type of the attachment.

- Type: `string`

size

Size of the attachment in bytes.

- Type: `integer`

path

Full URL for accessing the content of the attachment.

- Type: `string`

versions

List of available versions of the attachment.

- Type: `array`

creator

Creator of the attachment. Can be expanded to get the full User or Collaborator object by including the query parameter `expand=creator` in the request.

- Type: `string` / `object`

Request Parameters

content

The content of the image.

- Type: `multipart/form-data` or Base64 encoded attachment
- Required: `true`

name

The image filename.

- Type: `string`
- Required: `true` with Base64 encoded attachment, `false` with `multipart/form-data`

encoding

The encoding of `content`.

- Type: `string`
- Required: `true` with Base64 encoded attachment, `false` with `multipart/form-data`
- Allowed values: `"base64"`

Operations

GET /attachments

Returns metadata for attachments within the company scope, with optional limit and offset parameters, as a metadata-results object. The scope of the result set can be limited to a specific resource called attachable by adding `attachable_type` and `attachable_id` query params. The attachable can be defined by `attachable_type` (one of: `task`, `code_review`, `multi_code_review`, or `comment`) and identified by the `uuid` attribute (`id` for Comments).

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/attachments
```

Example Response

```
{
   "metadata": {
       "more_results": false,
       "next_offset": 3,
       "total_count": null,
       "count": 3
   },
   "results": [
       {
           "api_status": 200,
           "api_timestamp": "2019-03-08T10:35:27Z",
           "id": "2dbd4ceb-aec7-416b-8db6-b9d84174e7a7",
           "created_at": "2019-02-18T10:09:05Z",
           "updated_at": "2019-02-18T10:09:05Z",
           "name": "Screen Shot 2019-01-31 at 16.17.46.png",
           "mime": "image/png",
            "size": 377776,
            "path": "https://helixteamhub.cloud/api/attachments/2dbd4ceb-
aec7-416b-8db6-b9d84174e7a7",
            "versions": [
                "thumb"
            ],
            "attachable": {
                "id": 1,
                "uuid": "b98d1709-22b0-42af-b8ef-460e5f5712a2"
            },
            "attachable_type": "task",
            "creator": {
                "id": "admin",
                "uuid": "eaa8f420-0538-43ac-8440-10e22f2d2ba5",
                "type": "user"
            }
        },
        {
            "api_status": 200,
            "api_timestamp": "2019-03-08T10:35:27Z",
            "id": "13fc0634-b196-4f09-a205-8f1292daa7f6",
            "created_at": "2019-02-21T19:12:39Z",
            "updated_at": "2019-02-21T19:12:39Z",
            "name": "Screen Shot 2019-02-20 at 14.16.04.png",
            "mime": "image/png",
            "size": 206154,
            "path": "https://helixteamhub.cloud/api/attachments/13fc0634-
b196-4f09-a205-8f1292daa7f6",
            "versions": [
                "thumb"
            ],
            "attachable": null,
            "attachable_type": null,
            "creator": {
                "id": "admin",
                "uuid": "eaa8f420-0538-43ac-8440-10e22f2d2ba5",
                "type": "user"
            }
        },
        {
            "api_status": 200,
            "api_timestamp": "2019-03-08T10:35:27Z",
            "id": "c1a4ff95-add5-4b76-905f-de453059da1d",
            "created_at": "2019-02-26T22:19:02Z",
            "updated_at": "2019-02-27T00:04:12Z",
            "name": "Screen Shot 2019-02-26 at 21.31.25.png",
            "mime": "image/png",
            "size": 168657,
            "path": "https://helixteamhub.cloud/api/attachments/c1a4ff95-
add5-4b76-905f-de453059da1d",
            "versions": [
                "thumb"
            ],
            "attachable": {
                "id": "5c75bb5a222b8044636224a5"
            },
            "attachable_type": "comment",
            "creator": {
                "id": "admin",
                "uuid": "eaa8f420-0538-43ac-8440-10e22f2d2ba5",
                "type": "user"
            }
        }
    ]

}
```

GET /attachments/:id

Returns actual content for the attachment.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/attachments/40d7312d-1ed9-4559-b98a-
60c6bcdb2e5b
```

POST /attachments

Creates a new attachment. Returns metadata of the created attachment.

Content of the attachment can be UTF-8 (default) or Base64 encoded with optional data header. The endpoint also accepts multipart/form-data content-type, which may be needed when dealing with larger files.

Example Requests

JSON

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "content": "Fatal error", "name": "error.log" }' \
  https://helixteamhub.cloud/api/attachments
```

JSON with Base64 encoded content

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "encoding": "base64", "content": "RmF0YWwgZXJyb3I=", "name":
"error.log" }' \
  https://helixteamhub.cloud/api/attachments
```

Multipart/form-data

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -F "content=@screenshot.jpg" \
  https://helixteamhub.cloud/api/attachments
```

Example Response

```
{
   "api_status": 200,
   "api_timestamp": "2019-03-08T10:35:27Z",
   "id": "2dbd4ceb-aec7-416b-8db6-b9d84174e7a7",
   "created_at": "2019-02-18T10:09:05Z",
   "updated_at": "2019-02-18T10:09:05Z",
   "name": "Screen Shot 2019-01-31 at 16.17.46.png",
   "mime": "image/png",
   "size": 377776,
   "path": "https://helixteamhub.cloud/api/attachments/2dbd4ceb-aec7-416b-
8db6-b9d84174e7a7",
   "versions": [
       "thumb"
   ],
   "attachable": {
       "id": 1,
       "uuid": "b98d1709-22b0-42af-b8ef-460e5f5712a2"
   },
   "attachable_type": "task",
   "creator": {
       "id": "admin",
       "uuid": "eaa8f420-0538-43ac-8440-10e22f2d2ba5",
       "type": "user"
   }
}
```

PUT /attachments/:id

Updates the attachment. Returns the updated attachment object.

Example Requests

JSON

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "attachable_type": "code_review", "attachable_id": "4b67646b-7ee8-
4302-a1e1-3ad2acef469a" }' \
  https://helixteamhub.cloud/api/attachments/1f1baf44-abf7-43b7-bd72-
cc546cbfb094
```

Example Response

```
{
   "api_status": 200,
   "api_timestamp": "2019-03-08T11:13:33Z",
   "id": "1f1baf44-abf7-43b7-bd72-cc546cbfb094",
   "created_at": "2019-02-28T02:04:29Z",
   "updated_at": "2019-03-08T11:13:33Z",
   "name": "Screen Shot 2019-02-25 at 10.19.35.png",
   "mime": "image/png",
   "size": 42371,
   "path": "https://helixteamhub.cloud/api/attachments/1f1baf44-abf7-43b7-
bd72-cc546cbfb094",
   "versions": [
       "thumb"
   ],
   "attachable": {
       "id": 1,
       "uuid": "4b67646b-7ee8-4302-a1e1-3ad2acef469a"
   },
   "attachable_type": "code_review",
   "creator": {
       "id": "admin",
       "uuid": "eaa8f420-0538-43ac-8440-10e22f2d2ba5",
       "type": "user"
   }
}
```

DELETE /attachments/:id

Deletes an attachment. Returns metadata about the deleted attachment.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/attachments/40d7312d-1ed9-4559-b98a-
60c6bcdb2e5
```

Avatar

Avatars are user-given profile pictures that are included for Users and Collaborators.

- Attributes
- Request Parameters
- Operations
  - `PUT /account/avatar`

Attributes

large

The large size avatar, sized at 100 by 100 pixels.

- Type: `string`
- Default: `""`

```
"https://helixteamhub.cloud/api/attachments/e9090251-17c6-4b1b-b846-
ee7693041330?version=large"
```

medium

The medium size avatar, sized at 60 by 60 pixels.

- Type: `string`
- Default: `""`

```
"https://helixteamhub.cloud/api/attachments/e9090251-17c6-4b1b-b846-
ee7693041330?version=medium"
```

small

The small size avatar, sized at 30 by 30 pixels.

- Type: `string`
- Default: `""`

```
"https://helixteamhub.cloud/api/attachments/e9090251-17c6-4b1b-b846-
ee7693041330?version=small"
```

Request Parameters

content

The content of the image.

- Type: `multipart/form-data` or Base64 encoded image
- Required: `true`
- Allowed file formats: `png`, `jpg`, `gif`

name

The image filename.

- Type: `string`
- Required: `true` with Base64 encoded image, `false` with `multipart/form-data`

encoding

The encoding of `content`.

- Type: `string`
- Required: `true` with Base64 encoded image, `false` with `multipart/form-data`
- Allowed values: `"base64"`

Operations

PUT /account/avatar

Updates the avatar. Returns the updated avatar.

Example Request with multipart/form-data

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -F "content=@foo.jpg" \
  https://helixteamhub.cloud/api/account/avatar
```

Example Request with Base64 encoded image

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Content-Type: application/json" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -d '{ "encoding": "base64", "content":
"data:image/jpeg;base64,/9j/4AAQSkZJRgABAgA...", "name": "foo.jpg" }' \
  https://helixteamhub.cloud/api/account/avatar
```

Example Response

```
{
   "api_status": 200,
   "api_timestamp": "2019-03-08T12:28:08Z",
   "id": "fbd1761a-765a-439c-b2bd-5951540b3761",
   "created_at": "2019-03-08T12:28:08Z",
   "updated_at": "2019-03-08T12:28:08Z",
   "large": "https://helixteamhub.cloud/api/attachments/fbd1761a-765a-
439c-b2bd-5951540b3761?version=large",
   "medium": "https://helixteamhub.cloud/api/attachments/fbd1761a-765a-
439c-b2bd-5951540b3761?version=medium",
   "small": "https://helixteamhub.cloud/api/attachments/fbd1761a-765a-
439c-b2bd-5951540b3761?version=small"
}
```

Bookmark

A Bookmark represents a bookmark in Mercurial version control Repository.

> **Note**
> Slash characters in `:repository_id` must be encoded with `%2F`

Attributes

id

A name of the bookmark, e.g. `default`.

- Type: `string`

uuid

A universally unique identifier.

- Type: `string`

commit

Currently bookmarked commit

- Type: `string`

Operations

GET /projects/:project_id/repositories/:repository_id/bookmarks

Returns the bookmarks in a repository, with optional limit and offset, as a metadata-results object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/bookmarks
```

Example Response

```
{
  "metadata": {
    "more_results": true,
    "next_offset": 2,
    "count": 2
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2016-06-01T11:06:10Z",
      "id": "default",
      "uuid": "5a801da3-c208-4ccd-876c-4acfb39eda2a-default",
      "commit": "856b72272ad451b86f419d120360b1d8caa4b0c2"
    },
    {
      "api_status": 200,
      "api_timestamp": "2016-06-01T11:06:10Z",
      "id": "feature1",
      "uuid": "5a801da3-c208-4ccd-876c-4acfb39eda2a-feature1",
      "commit": "a810df9f2805d53c4bc0bf886029527e71fe8411"
    }
  ]
}
```

Branch

A Branch represents a branch in a version control Repository.

- Attributes
- Operations
  - `GET /projects/:project_id/repositories/:repository_id/branches`
  - `GET /projects/:project_id/repositories/:repository_id/branches/:id`

- `POST /projects/:project_id/repositories/:repository_id/branches`
- `PUT /projects/:project_id/repositories/:repository_id/branches/:id`
- `DELETE /projects/:project_id/repositories/:repository_id/branches/:id`

> **Note**
> Forward slash characters in `:repository_id` must be encoded with `%2F`

Attributes

id

A name of the branch, e.g. `master`.

- Type: `string`

uuid

A universally unique identifier.

- Type: `string`

commit

The latest commit in the branch

- Type: `string`

protected

An indicator of whether the branch is protected or not. Protected branches cannot be deleted, cannot be pushed with force, and only Project Admin can push to them. Currently branches can be protected only within Git repositories.

- Type: `boolean`

Operations

GET /projects/:project_id/repositories/:repository_id/branches

Returns the branches in a repository, with optional limit and offset, as a metadata-results object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/branches
```

Example Response

```
{
  "metadata": {
    "more_results": false,
    "next_offset": 3,
    "count": 3
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2016-03-18T14:33:33Z",
      "id": "develop",
      "uuid": "46943352-03c3-42f8-807f-e378e98e878c-develop",
      "commit": "ad527b07d4618bad64368ad507d0d193c4387d99",
      "protected": false,
      "repository": {
        "id": "chef",
        "uuid": "46943352-03c3-42f8-807f-e378e98e878c"
      }
    },
    {
      "api_status": 200,
      "api_timestamp": "2016-03-18T14:33:33Z",
      "id": "master",
      "uuid": "46943352-03c3-42f8-807f-e378e98e878c-master",
      "commit": "eadca9ff750e9e504f94e72b702423c399641098",
      "protected": true,
      "repository": {
        "id": "chef",
        "uuid": "46943352-03c3-42f8-807f-e378e98e878c"
      }
    },
    {
      "api_status": 200,
      "api_timestamp": "2016-03-18T14:33:33Z",
      "id": "production",
      "uuid": "46943352-03c3-42f8-807f-e378e98e878c-production",
      "commit": "063f8954ef1a8d997fde9eee93651914ea6b1e75",
      "protected": true,
      "repository": {
        "id": "chef",
        "uuid": "46943352-03c3-42f8-807f-e378e98e878c"
      }
    }
  ]
}
```

GET /projects/:project_id/repositories/:repository_id/branches/:id

Returns a specific branch.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/branches/master
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2016-03-18T14:35:57Z",
  "id": "master",
  "uuid": "46943352-03c3-42f8-807f-e378e98e878c-master",
  "commit": "eadca9ff750e9e504f94e72b702423c399641098",
  "protected": true,
  "repository": {
    "id": "chef",
    "uuid": "46943352-03c3-42f8-807f-e378e98e878c"
  }
}
```

POST /projects/:project_id/repositories/:repository_id/branches

Creates a new branch to the company and returns the created branch object.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "production", "commit":
"063f8954ef1a8d997fde9eee93651914ea6b1e75", "protected": true }' \

https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/branches
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2016-03-18T14:30:50Z",
  "id": "production",
  "uuid": "46943352-03c3-42f8-807f-e378e98e878c-production",
  "commit": "063f8954ef1a8d997fde9eee93651914ea6b1e75",
  "protected": true,
  "repository": {
    "id": "chef",
    "uuid": "46943352-03c3-42f8-807f-e378e98e878c"
  }
}
```

PUT /projects/:project_id/repositories/:repository_id/branches/:id

Updates the branch protection attribute. Only Project Admins are allowed to change branch protection.

Example Request

```
curl -X PUT \
  -H "Content-Type: application/json" \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -d '{ "protected": false }'

https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/branches/
production
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2016-03-18T15:53:05Z",
  "id": "production",
  "uuid": "46943352-03c3-42f8-807f-e378e98e878c-production",
  "commit": "063f8954ef1a8d997fde9eee93651914ea6b1e75",
  "protected": false,
  "repository": {
    "id": "chef",
    "uuid": "46943352-03c3-42f8-807f-e378e98e878c"
  }
}
```

DELETE /projects/:project_id/repositories/:repository_id/branches/:id

Deletes a branch. Returns the deleted branch.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/branches/
production
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2016-03-18T15:53:36Z",
  "id": "production",
  "uuid": "46943352-03c3-42f8-807f-e378e98e878c-production",
  "commit": "063f8954ef1a8d997fde9eee93651914ea6b1e75",
  "protected": false,
  "repository": {
    "id": "chef",
    "uuid": "46943352-03c3-42f8-807f-e378e98e878c"
  }
}
```

Code Review

Functionality allowing people to review and approve the code.

- Attributes
- Searchable attributes
- Operations
  - `GET /projects/:project_id/code_reviews`
  - `GET /projects/:project_id/code_reviews/:id`
  - `POST /projects/:project_id/code_reviews`
  - `PUT /projects/:project_id/code_reviews/:id`
  - `DELETE /projects/:project_id/code_reviews/:id`

- `GET /projects/:project_id/code_reviews/count`
- `PUT /projects/:project_id/code_reviews/:id/actions/merge`
- `PUT /projects/:project_id/code_reviews/:id/actions/close`
- `POST /projects/:project_id/code_reviews/:id/actions/vote`
- `DELETE /projects/:project_id/code_reviews/:id/actions/unvote`

Attributes

id

Automatically generated URL-friendly identifier for the code review.

- Type: `integer`
- Immutable: `true`
- Unique: `true` (in project scope)
- Required: `true`

title

Human-readable title for the code review.

- Type: `string`
- Required: `true`
- Minimum length: `1`
- Maximum length: `100`

description

Free-form description of the code review.

- Type: `string`
- Default: `""`
- Maximum length: `50000`

voting_threshold

Number of required approvals.

- Type: `integer`
- Default: `0`

require_build

Whether or not the code review requires build.

- Type: `boolean`
- Required: `true`

state

Current state of the code review.

- Type: `string`

merge_state

Current merge process state of the code review.

- Type: `string`

head_branch

Name of the source.

- Type: `string`
- Required: `true`

head_branch_type

Type of the source. Allowed values (depends on repository type): `branch`, `bookmark` or `tag`.

- Type: `string`
- Required: `true`
- Default: `branch`

head_branch_id

Recent commit of the source.

- Type: `string`

base_branch

Name of the destination.

- Type: `string`
- Required: `true`

base_branch_type

Type of the destination. Allowed values (depends on repository type): `branch`, `bookmark` or `tag`.

- Type: `string`
- Required: `true`
- Default: `branch`

base_branch_id

Recent commit of the destination.

- Type: `string`

creator

Creator of the code review can be expanded to get the full User or Collaborator object by including the query parameter `expand=creator` in the request.

- Type: `object`

project

Project that the code review belongs to. Can be expanded to get the full Project object by including the query parameter `expand=project` in the request.

- Type: `object`

repository

Repository that the code review belongs to. Can be expanded to get the full Repository object by including the query parameter `expand=repository` in the request.

- Type: `object`
- Required: `true`

Searchable attributes

Code reviews can be searched by their `id`, `title`, and `description` attributes. See the search documentation for instructions how to do this.

Operations

GET /projects/:project_id/code_reviews

Returns all the code reviews in the project, with optional limit and offset parameters, as a metadata-results object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/code_reviews
```

Example Response

```
{
  "metadata": {
    "more_results": false,
    "next_offset": 1,
    "count": 1
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2016-04-01T10:08:59Z",
      "id": 2,
      "created_at": "2016-04-01T10:00:53Z",
      "updated_at": "2016-04-01T10:06:09Z",
      "uuid": "1c5eaeda-312a-40f1-8e7f-24cbcfcecbb7",
      "title": "Garbage collector for Git repositories",
      "description": "Related issue: #333",
      "voting_threshold": 2,
      "state": "open",
      "merge_state": "initial",
      "creator": {
        "id": "norris",
        "uuid": "fdd36536-8a08-4e5b-ab25-6f7c975c9212",
        "type": "user"
      },
      "head_branch": "develop",
      "head_branch_type": "branch",
      "head_commit_id": "746c7d5abbb99bfdb4ee255f313e625919b8a480",
      "base_branch": "master",
      "base_branch_type": "branch",
      "base_commit_id": "06f6e74cc8bd98974fe206daeefa12dd8161d775",
      "project": {
        "id": "luotsi",
        "uuid": "743be99c-d627-4733-8a4f-30472573a1fb"
      },
      "repository": {
        "id": "chef",
        "uuid": "87fe327c-b169-4b8f-84ea-131a59b39c28"
      },
      "require_build": false
    }
  ]
}
```

GET /projects/:project_id/code_reviews/:id

Returns specific code review object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/code_reviews/2
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2016-04-01T10:04:07Z",
  "id": 2,
  "created_at": "2016-04-01T10:00:53Z",
  "updated_at": "2016-04-01T10:00:53Z",
  "uuid": "1c5eaeda-312a-40f1-8e7f-24cbcfcecbb7",
  "title": "Garbage collector for Git repositories",
  "description": "Related issue: #123",
  "voting_threshold": 5,
  "state": "open",
  "merge_state": "initial",
  "creator": {
    "id": "norris",
    "uuid": "fdd36536-8a08-4e5b-ab25-6f7c975c9212",
    "type": "user"
  },
  "head_branch": "develop",
  "head_branch_type": "branch",
  "head_commit_id": "746c7d5abbb99bfdb4ee255f313e625919b8a480",
  "base_branch": "master",
  "base_branch_type": "branch",
  "base_commit_id": "06f6e74cc8bd98974fe206daeefa12dd8161d775",
  "project": {
    "id": "luotsi",
    "uuid": "743be99c-d627-4733-8a4f-30472573a1fb"
  },
  "repository": {
    "id": "chef",
    "uuid": "87fe327c-b169-4b8f-84ea-131a59b39c28"
  },
  "require_build": true
}
```

POST /projects/:project_id/code_reviews

Creates a new code review to the project. Returns the created code review object.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "base_branch": "master", "head_branch": "develop", "repository":
"chef", "base_branch_type": "branch", "head_branch_type": "branch",
"voting_threshold": "5", "require_build": true, "title": "Garbage
collector for Git repositories", "description": "Related issue: #123"}' \
  https://helixteamhub.cloud/api/projects/luotsi/code_reviews
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2016-04-01T10:00:53Z",
  "id": 2,
  "created_at": "2016-04-01T10:00:53Z",
  "updated_at": "2016-04-01T10:00:53Z",
  "uuid": "1c5eaeda-312a-40f1-8e7f-24cbcfcecbb7",
  "title": "Garbage collector for Git repositories",
  "description": "Related issue: #123",
  "voting_threshold": 5,
  "state": "open",
  "merge_state": "initial",
  "creator": {
    "id": "norris",
    "uuid": "fdd36536-8a08-4e5b-ab25-6f7c975c9212",
    "type": "user"
  },
  "head_branch": "develop",
  "head_branch_type": "branch",
  "head_commit_id": "746c7d5abbb99bfdb4ee255f313e625919b8a480",
  "base_branch": "master",
  "base_branch_type": "branch",
  "base_commit_id": "06f6e74cc8bd98974fe206daeefa12dd8161d775",
  "project": {
    "id": "luotsi",
    "uuid": "743be99c-d627-4733-8a4f-30472573a1fb"
  },
  "repository": {
    "id": "chef",
    "uuid": "87fe327c-b169-4b8f-84ea-131a59b39c28"
  },
  "require_build": true
}
```

PUT /projects/:project_id/code_reviews/:id

Updates the code review. Returns the updated code review object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "voting_threshold": "2", "require_build": false, "description":
"Related issue: #333"}' \
  https://helixteamhub.cloud/api/projects/luotsi/code_reviews/2
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2016-04-01T10:06:09Z",
  "id": 2,
  "created_at": "2016-04-01T10:00:53Z",
  "updated_at": "2016-04-01T10:06:09Z",
  "uuid": "1c5eaeda-312a-40f1-8e7f-24cbcfcecbb7",
  "title": "Garbage collector for Git repositories",
  "description": "Related issue: #333",
  "voting_threshold": 2,
  "state": "open",
  "merge_state": "initial",
  "creator": {
    "id": "norris",
    "uuid": "fdd36536-8a08-4e5b-ab25-6f7c975c9212",
    "type": "user"
  },
  "head_branch": "develop",
  "head_branch_type": "branch",
  "head_commit_id": "746c7d5abbb99bfdb4ee255f313e625919b8a480",
  "base_branch": "master",
  "base_branch_type": "branch",
  "base_commit_id": "06f6e74cc8bd98974fe206daeefa12dd8161d775",
  "project": {
    "id": "luotsi",
    "uuid": "743be99c-d627-4733-8a4f-30472573a1fb"
  },
  "repository": {
    "id": "chef",
    "uuid": "87fe327c-b169-4b8f-84ea-131a59b39c28"
  },
  "require_build": false
}
```

DELETE /projects/:project_id/code_reviews/:id

Deletes the code review. Returns the deleted code review object.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/code_reviews/2
```

GET /projects/:project_id/code_reviews/count

Returns count of code reviews.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/code_reviews/count
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2016-04-01T10:10:39Z",
  "count": 1
}
```

PUT /projects/:project_id/code_reviews/:id/actions/merge

Starts asynchronous merge of the code review. Returns the code review in processing state.

For git repositories a merge method can be specified by sending the **`merge_method`** parameter. The following options are supported:

- **`merge`** (default)
- **`rebase`**
- **`squash`**

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  https://helixteamhub.cloud/api/projects/luotsi/code_
reviews/2/actions/merge
```

PUT /projects/:project_id/code_reviews/:id/actions/close

Closes the code review. Returns the code review in closed state.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  https://helixteamhub.cloud/api/projects/luotsi/code_
reviews/2/actions/close
```

POST /projects/:project_id/code_reviews/:id/actions/vote

Approves the code review by authorized account.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  https://helixteamhub.cloud/api/projects/luotsi/code_
reviews/2/actions/vote
```

DELETE /projects/:project_id/code_reviews/:id/actions/unvote

Withdraws approval for the code review.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  https://helixteamhub.cloud/api/projects/luotsi/code_
reviews/2/actions/unvote
```

Collaborator

Collaborators are users with limited access to the Company they're created in. Collaborators can only access Projects they have been specifically added to.

- Attributes
- Searchable attributes
- Child objects
- Operations
  - `GET /collaborators`
  - `GET /collaborators/:id`
  - `POST /collaborators`
  - `PUT /collaborators/:id`
  - `DELETE /collaborators/:id`

Attributes

Collaborators share the same attributes with Users.

Searchable attributes

Collaborators can be searched by their `id`, `email`, and `display_name` attributes. See the search documentation for instructions how to do this.

Child objects

The following child objects can be attached to the collaborators returned, to eliminate the need for requesting them separately:

Object	Attribute
Project	`projects`
Repository	`repositories`

For example, to get all of the collaborators's projects in full, attach `include=projects` to your GET request.

See the child object API documentation for a description of the possible operations.

Operations

GET /collaborators

Returns all the collaborators in the company, with optional limit and offset parameters, as a metadata-results object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/collaborators
```

Example Response

```
{
  "metadata": {
    "more_results": false,
    "next_offset": 7,
    "count": 7
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2014-11-25T13:55:11Z",
      "id": "norris",
      "created_at": "2014-11-18T13:19:40Z",
      "updated_at": "2014-11-18T13:19:40Z",
      "type": "collaborator",
      "email": "norris@acme.org",
      "first_name": "Chuck",
      "last_name": "Norris",
      "display_name": "Chuck Norris",
      "company_admin": false,
      "instance_admin": false,
      "description": "",
      "phone": "",
      "title": "",
      "avatar": {
        "large": "",
        "medium": "",
        "small": ""
      },
      "synchronized_fields": [

      ],
      "locale": "en",
      "source": ""
    },
    ...
  ]
}
```

GET /collaborators/:id

Returns a specific collaborator.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/collaborators/norris
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-11-25T13:55:11Z",
  "id": "norris",
  "created_at": "2014-11-18T13:19:40Z",
  "updated_at": "2014-11-18T13:19:40Z",
  "type": "collaborator",
  "email": "norris@acme.org",
  "first_name": "Chuck",
  "last_name": "Norris",
  "display_name": "Chuck Norris",
  "company_admin": false,
  "instance_admin": false,
  "description": "",
  "phone": "",
  "title": "",
  "avatar": {
    "large": "",
    "medium": "",
    "small": ""
  },
  "synchronized_fields": [

  ],
  "locale": "en",
  "source": ""
}
```

POST /collaborators

Creates a new collaborator to the company and returns the created collaborator object. A matching collaborator is first searched from the directory (by **`email`** and **`id`**) when LDAP authentication is enabled for collaborators. The collaborator must be found from the directory when LDAP only authentication is used for collaborators, otherwise a 404 response is returned.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "norris", "email": "norris@acme.org" }' \
  https://helixteamhub.cloud/api/collaborators
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2014-11-25T13:55:11Z",
  "id": "norris",
  "created_at": "2014-11-18T13:19:40Z",
  "updated_at": "2014-11-18T13:19:40Z",
  "type": "collaborator",
  "email": "norris@acme.org",
  "first_name": "",
  "last_name": "",
  "display_name": "norris@acme.org",
  "company_admin": false,
  "instance_admin": false,
  "description": "",
  "phone": "",
  "title": "",
  "avatar": {
    "large": "",
    "medium": "",
    "small": ""
  },
  "synchronized_fields": [

  ],
  "locale": "en",
  "source": ""
}
```

PUT /collaborators/:id

Updates a collaborator. Returns the updated collaborator.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "first_name": "Chuck", "last_name": "Norris" }' \
  https://helixteamhub.cloud/api/collaborators/norris
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-11-25T13:55:11Z",
  "id": "norris",
  "created_at": "2014-11-18T13:19:40Z",
  "updated_at": "2014-11-18T13:19:40Z",
  "type": "collaborator",
  "email": "norris@acme.org",
  "first_name": "Chuck",
  "last_name": "Norris",
  "display_name": "norris@acme.org",
  "company_admin": false,
  "instance_admin": false,
  "description": "",
  "phone": "",
  "title": "",
  "avatar": {
    "large": "",
    "medium": "",
    "small": ""
  },
  "synchronized_fields": [

  ],
  "locale": "en",
  "source": ""
}
```

DELETE /collaborators/:id

Deletes a collaborator. Returns the deleted collaborator.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \
  https://helixteamhub.cloud/api/collaborators/norris
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-11-25T13:55:11Z",
  "id": "norris",
  "created_at": "2014-11-18T13:19:40Z",
  "updated_at": "2014-11-18T13:19:40Z",
  "type": "collaborator",
  "email": "norris@acme.org",
  "first_name": "Chuck",
  "last_name": "Norris",
  "display_name": "norris@acme.org",
  "company_admin": false,
  "instance_admin": false,
  "description": "",
  "phone": "",
  "title": "",
  "avatar": {
    "large": "",
    "medium": "",
    "small": ""
  },
  "synchronized_fields": [

  ],
  "locale": "en",
  "source": ""
}
```

Comment

Comments are snippets of user-created content that are linked to other objects. They are used for discussing and remarking on things that happen in Helix TeamHub.

- Attributes
- Operations
    - `GET /comments`
    - `GET /comments/:id`
    - `POST /comments`
    - `PUT /comments/:id`
    - `DELETE /comments/:id`

Attributes

id

A URL-friendly identifier for the comment.

- Type: `string`
- Unique: `true`

type

The type of the object(s) this comment is targeted.

- Type: `string`
- Required: `true`
- Allowed values: `task`, `comment`

Type defines the target objects for the comment:

Type	Target Object(s)
`comment`	comment
`task`	project, task

author

The author of the comment. Contains the identifier of a User or a Collaborator.

To get the full object representation of the author instead of just their id, expand it in your request by including the query parameter `expand=author`.

- Type: `string` / `object`

content

The body of the comment.

- Type: `string`
- Required: `true`
- Minimum length: `1`
- Maximum length: `50000`

objects

Only used when reading. A hash of object references, showing the target object(s) of the comment. For example: `{"project": "luotsi", "task": 123, ...}`.

- Type: `object`

project

Only used when writing. If the comment is targeted at a Task (comment `type` is `task`), specify the project's identifier.

- Type: `string`

task

Only used when writing. If the comment is targeted at a Task (comment `type` is `task`), specify the task's identifier.

- Type: `integer`

comment

Only used when writing. If the comment is a response to another comment (`type` is `comment`), specify the other comment's identifier. Only one level of nested replies is supported.

- Type: `string`

timestamp

The date and time when the comment was posted, in ISO-8601 format.

- Type: `timestamp`

Operations

GET /comments

Get all comments for the company. Results may be limited with query parameters. Returned as a metadata-results object.

Example Request

Get 2 latest comments for a task.

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \
  https://helixteamhub.cloud/api/comments?project=luotsi&task=1&limit=2
```

Example Response

```
{
  "metadata": {
    "more_results": true,
    "next_offset": 2,
    "total_count": null,
    "count": 2
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2012-10-25T10:19:24Z",
      "id": "5087f00b1229c02952000034",
      "created_at": "2012-10-24T13:41:31Z",
      "updated_at": "2012-10-24T13:41:31Z",
      "type": "task",
      "author": {
        "id": "avesalainen"
      },
      "content": "fiifoo",
      "objects": {
        "task": {
          "id": 1
        },
        "project": {
          "id": "luotsi"
        }
      },
      "timestamp": "2012-10-24T13:41:31Z"
    },
    {
      "api_status": 200,
      "api_timestamp": "2012-10-25T10:19:24Z",
      "id": "5367a96234be9557910001bc",
      "created_at": "2012-09-24T08:42:10Z",
      "updated_at": "2012-09-24T08:42:10Z",
      "type": "task",
      "author": {
        "id": "jp"
      },
      "content": "This has now been fixed.",
      "objects": {
        "task": {
          "id": 1
        },
        "project": {
          "id": "luotsi"
        }
      },
      "timestamp": "2012-09-24T08:42:10Z"
    }
  ]
}
```

GET /comments/:comment_id

Get a specific comment.

POST /comments

Create a new comment.

Example Request

Create a comment for an Task.

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "content": "This has now been fixed.", "project": "luotsi",
"task": "1", "type": "task" }' \
  https://helixteamhub.cloud/api/comments
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2012-09-24T08:42:10Z",
  "id": "5367a96234be9557910001bc",
  "created_at": "2012-09-24T08:42:10Z",
  "updated_at": "2012-09-24T08:42:10Z",
  "type": "task",
  "author": {
    "id": "jp"
  },
  "content": "This has now been fixed.",
  "objects": {
    "task": {
      "id": 1
    },
    "project": {
      "id": "luotsi"
    }
  },
  "timestamp": "2012-09-24T08:42:10Z"
}
```

PUT /comments/:id

Updates a comment. Returns the updated comment object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "content": "hello" }' \
  https://helixteamhub.cloud/api/comments/5367a96234be9557910001bc
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2012-10-25T10:24:55Z",
  "id": "5367a96234be9557910001bc",
  "created_at": "2012-10-25T10:24:55Z",
  "updated_at": "2012-10-25T10:32:21Z",
  "type": "task",
  "author": {
    "id": "jp"
  },
  "content": "hello",
  "objects": {
    "task": {
      "id": 1
    },
    "project": {
      "id": "luotsi"
    }
  },
  "timestamp": "2012-10-25T10:24:55Z"
}
```

DELETE /comments/:id

Deletes a comment. Returns the deleted comment object.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/comments/5367a96234be9557910001bc
```

Commit

A commit is a change in a version control repository (a Git commit, a Subversion revision, or a Mercurial changeset). Please note that as WebDAV, Maven, and Ivy don't support versioning, a dummy commit ID will have to be specified for most of the operations. In Helix TeamHub APIs, commits are the way to access Repository contents. They provide a way to:

- Browse the commits and commit history of a repository.
- Browse the metadata and contents of a commit.
- Access the actual files in a repository.
- Search repository for content and files (git only).

Commits can be created using the respective version control systems. Currently Helix TeamHub API supports creating commits only for git repositories.

- Attributes
- Request Parameters
- Child Objects
- Operations
  - **`GET /projects/:project_id/repositories/:repository_id/commits`**
  - **`POST /projects/:project_id/repositories/:repository_id/commits`**

- `GET /projects/:project_id/repositories/:repository_id/commits/:id`
- `GET /projects/:project_id/repositories/:repository_id/commits/:id/changesets`
- `GET /projects/:project_id/repositories/:repository_id/commits/:id/tree(/:path)`
- `GET /projects/:project_id/repositories/:repository_id/commits/:id/files(/:path)`
- `GET /projects/:project_id/repositories/:repository_id/commits/:id/history(/:path)`
- `GET /projects/:project_id/repositories/:repository_id/commits/:id/future(/:path)`
- `GET /projects/:project_id/repositories/:repository_id/commits/:id/search`
- `GET /projects/:project_id/repositories/:repository_id/compare/commits/:base...:head` *experimental*
- `GET /projects/:project_id/repositories/:repository_id/compare/changesets/:base...:head` *experimental*
- `GET /projects/:project_id/tasks/:task_id/commits`

> **Note**
> Forward slash characters in both `:repository_id` and `:commit_id` must be encoded with `%2F`

Attributes

id

The identifier of the commit (for example, a Git commit hash).

- Type: `string`

timestamp

The time and date of the original commit, in ISO-8601 format (as reported by the version control system). Note that this is derived from the local time on the committer's device.

- Type: `timestamp`

author

The name of the author of the commit (as reported by the version control system).

- Type: `string`

email

The email of the author of the commit (as reported by the version control system).

- Type: `string`

description

The commit message (as reported by the version control system).

- Type: `string`

Request Parameters

parent

The parent commit SHA-1 value or reference, for example `"caed70ef"` or `"master"`.

- Type: `string`
- Required: `true`

modified

An array of objects describing the changes made.

- Type: `array` of `object`
- Required: `true` if `deleted` is not provided, `false` otherwise

```
[
  {
    "path": "README.md",
    "content": "### Hello World"
  },
  {
    "encoding": "base64",
    "path": "norris.png",
    "content": "data:image/png;base64,..."
  }
]
```

deleted

An array of strings describing the removed files and/or directories.

- Type: `array` of `string`
- Required: `true` if `modified` is not provided, `false` otherwise

```
["foo.js", "bar", "baz.html"]
```

Child Objects

A commit's child objects can be accessed as subresources (see Operations), or (partly) expanded while requesting commit objects to minimize the amount of requests.

account

`expand=account`

This requires a valid commit email or author and will return a Helix TeamHub User or Collaborator if a matching one is found, otherwise an empty string. The mapping is primarily done using the commit email and secondarily by the commit author.

changeset

`expand=changeset`

A list of changed files with their respective individual changes.

tree

`expand=tree`

A directory structure (including metadata) of the repository at the time of this commit. Directories, files and Git submodules are represented in the same format (as "tree nodes").

Operations

GET /projects/:project_id/repositories/:repository_id/commits

Lists the newest commits for the given repository, with optional limit and offset parameters, as a metadata-results object.

POST /projects/:project_id/repositories/:repository_id/commits

Creates a new commit. Content of the modified files can be UTF-8 (default) or base64 encoded with optional data header. Description for the commit is optional and a default message is generated when it is omitted. The endpoint also accepts multipart/form-data content-type, which can be needed when dealing with larger files. Currently only git repositories are supported.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{
    "parent": "master",
    "description": "Edited some files",
    "modified": [
      {
        "path": "lib/apps.js",
        "content": "console.log(42);"
      },
      {
        "encoding": "base64",
        "path": "run.js",
        "content": "Zm9vKCk7"
      },
      {
        "encoding": "base64",
        "path": "dot.png",
        "content":
"data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAUAAAAFCAYAAACNbyblAAAAHEl
EQVQI12P4//8/w38GIAXDIBKE0DHxgljNBAAO9TXL0Y4OHwAAAABJRU5ErkJggg=="
      }
    ],
    "deleted": ["foo.js"]
  }' \
  https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/commits
```

Example Request with multipart/form-data

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -F "parent=master" \
  -F "modified[][path]=README.md" \
  -F "modified[][content]=@README.md" \
  https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/commits
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2013-10-21T08:31:15Z",
  "id": "42cc6d9a9bad6758d6b8357e47f662c236a9c59c",
  "author": "Chuck Norris",
  "email": "chuck@norris.com",
  "description": "Edited some files",
  "timestamp": "2013-10-21T08:31:15Z"
}
```

GET /projects/:project_id/repositories/:repository_id/commits/:id

Shows an individual commit object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/luotsi/commits
/c57164ed653ec49665ff91b906ca41101bcc3486
```

Example Response

```
{
    "api_status": 200,
    "api_timestamp": "2012-10-25T13:51:07Z",
    "id": "c57164ed653ec49665ff91b906ca41101bcc3486",
    "author": "Anssi Syrjäsalo",
    "email": "anssi.syrjasalo@eficode.com",
    "description": "User short_name cannot be 'count'",
    "timestamp": "2012-10-24T15:41:55Z"
}
```

GET /projects/:project_id/repositories/:repository_id/commits/:id/changesets

Returns changes made in a commit. The changes are limited to at maximum of 150 files, 50 000 lines, or a total of 1 MB of diff content. Individual file diffs are limited to 50 KB of diff content.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/luotsi/commits
/831c9809309e640a5191a639a7f002012f2a842c/changesets
```

Example Response

```
[
    {
        "api_status": 200,
        "api_timestamp": "2012-10-25T14:03:03Z",
        "path": "spec/models/project_spec.rb",
        "file_type": "text",
        "additions": 1,
        "deletions": 1,
        "action": "modified",
        "truncated": false,
        "diff": [
            {
                "type": "range",
                "content": "@@ -67,7 +67,7 @@ describe Api::V1::Project
do",
                "old_start_line": "67",
                "new_start_line": "67"
            },
            {
                "type": "unchanged",
                "content": " "
            },
            {
                "type": "unchanged",
                "content": "         it \"should be listed\" do"
            },
            {
                "type": "unchanged",
                "content": "           serialized = @api_model.new
(@project, { list: 'users' }).to_api"
            },
            {
                "type": "deleted",
                "content": "-          serialized[:users].should == [
@user.short_name, @user2.short_name ]"
            },
            {
                "type": "added",
                "content": "+          serialized[:users].should =~ [
@user.short_name, @user2.short_name ]"
            },
            {
                "type": "unchanged",
                "content": "         end"
            },
            {
                "type": "unchanged",
                "content": " "
            },
            {
                "type": "unchanged",
                "content": "         it \"should be included\" do"
            }
        ]
    }
]
```

GET /projects/:project_id/repositories/:repository_id/commits/:id/tree(/:path)

Returns one level of file hierarchy and metadata specified by the path. Latest commit for each node can be expanded with `expand=commit`.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/luotsi/commits
/master/tree/lib
```

Example Response

```
{
    "api_status": 200,
    "api_timestamp": "2012-10-25T13:54:56Z",
    "id": "25f79cbd94c83535c6119aab43b32f9b1c79f39c",
    "type": "directory",
    "name": "lib",
    "path": "lib",
    "mime": "",
    "size": 0,
    "file_url": "",
    "children": [
        {
            "api_status": 200,
            "api_timestamp": "2012-10-25T13:54:56Z",
            "id": "baaa70dc339f33191ead4f17a9e82296b4f4d43b",
            "type": "directory",
            "name": "hth",
            "path": "lib/hth",
            "mime": "",
            "size": 0,
            "file_url": "",
            "children": []
        },
        {
            "api_status": 200,
            "api_timestamp": "2012-10-25T13:54:56Z",
            "id": "faa8489b72245fb662772dca55fced6a6e65b6dd",
            "type": "directory",
            "name": "devise",
            "path": "lib/devise",
            "mime": "",
            "size": 0,
            "file_url": "",
            "children": []
        },
        {
            "api_status": 200,
            "api_timestamp": "2012-10-25T13:54:56Z",
            "id": "882d9014f4335261bcf68c4d0596f7983d262d5b",
            "type": "directory",
            "name": "tasks",
            "path": "lib/tasks",
            "mime": "",
            "size": 0,
            "file_url": "",
            "children": []
        }
    ]
}
```

GET /projects/:project_id/repositories/:repository_id/commits/:id/files(/:path)

Returns the contents of a given file, delivered with the correct mime type (and possibly converted to a format as requested by the client e.g. `contentType=html`).

GET /projects/:project_id/repositories/:repository_id/commits/:id/history(/:path)

Returns all the parent commits for a specified commit, with optional limit and offset, as a metadata-results object.

GET /projects/:project_id/repositories/:repository_id/commits/:id/future(/:path)

Returns all the children commits for a specified commit, with optional limit and offset, as a metadata-results object.

GET /projects/:project_id/repositories/:repository_id/commits/:id/search

Searches repository content and paths for the `search_term` parameter. Optional `search_target` parameter can be used to limit the target of the search. Currently only git repositories are supported.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/commits/m
aster/search?search_term=console&search_target=path,content
```

Example Response

```
{
  "content_matches": [
    {
      "path": "javascripts/lib.js",
      "count": 2
    },
    {
      "path": "apps/run.js",
      "count": 1
    }
  ],
  "path_matches": [
    "javascripts/console-fix.js",
    "app/assets/images/console.gif"
  ]
}
```

GET /projects/:project_id/repositories/:repository_id/compare/commits/:base...:head

> **Warning**
> This is an **experimental** API part and might be changed in the future.

Returns a range of commits from `:base` to `:head`. Both identifiers might be commit SHA1 identifiers or branch names. The output is similar to the `/commits` endpoint but narrowed to the given range borders. The endpoint is limited to return at maximum 200 commits.

Child objects are available within this result set through the optional `expand` parameter. Currently only git repositories are supported.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/my_project/repositories/my_
repo/compare/commits/f473437413f44ca71d7c6b6ae0f68b32f7bafd7d...master
```

Example Response

```
{
  "metadata": {
    "more_results": false,
    "next_offset": 2,
    "count": 2
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2015-02-16T13:16:30Z",
      "id": "5e251b94869ebcd0029b5e182f0e65010908cfa3",
      "author": "Bruce Wayne",
      "email": "bruce@waynecorp.com",
      "description": "Change to bar",
      "timestamp": "2015-02-06T07:04:13Z"
    },
    {
      "api_status": 200,
      "api_timestamp": "2015-02-16T13:16:30Z",
      "id": "f3d627d76bb76fdd2968f803151d5c5f6fe00de9",
      "author": "Bruce Wayne",
      "email": "bruce@waynecorp.com",
      "description": "Add foo",
      "timestamp": "2015-02-05T19:55:40Z"
    }
  ]
}
```

GET /projects/:project_id/repositories/:repository_id/compare/changesets/:base...:head

WARNING: This is an **experimental** API part and might be changed in the future.

Returns a range of changesets from `:base` to `:head`. Both identifiers might be commit SHA1 identifiers or branch names. The output is similar to the `/changesets` endpoint but narrowed to the given range borders. The same limits are applied as with `/changesets` endpoint. Currently only git repositories are supported.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/my_project/repositories/my_
repo/compare/changesets/f473437413f44ca71d7c6b6ae0f68b32f7bafd7d...master
```

Example Response

```
[
  {
    "api_status": 200,
    "api_timestamp": "2015-02-16T14:24:39Z",
    "path": "foo.txt",
    "file_type": "text",
    "additions": 5,
    "deletions": 3,
    "action": "modified",
    "truncated": false,
    "diff": [
      {
        "type": "range",
        "content": "@@ -1,5 +1,7 @@",
        "old_start_line": "1",
        "new_start_line": "1"
      },
      {
        "type": "unchanged",
        "content": " Lorem ipsum."
      },
      {
        "type": "added",
        "content": "+Dolor sit amet."
      },
      {
        "type": "added",
        "content": "+Lorem ipsumljl."
      },
      {
        "type": "added",
        "content": "+Lorem bla."
      },
      {
        "type": "added",
        "content": "+Lorem ipsumlkj."
      },
      {
        "type": "unchanged",
        "content": " Lorem ipsum."
      },
      {
        "type": "deleted",
        "content": "-Lorem ipsum."
      },
      {
        "type": "deleted",
        "content": "-Lorem ipsum."
      },
      {
        "type": "deleted",
        "content": "-Lorem ipsum."
      },
      {
        "type": "added",
        "content": "+aaabbbccc"
      }
    ],
    "file_url": "https://helixteamhub.cloud/api/projects/my_
project/repositories/my_repo/commits/foobar/files/foo.txt"
  }
```

```
]
```

GET /projects/:project_id/tasks/:task_id/commits

Returns commits that reference the task identifier (e.g. `#123`) in the commit messages.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/tasks/123/commits
```

Example Response

```
{
  "metadata": {
    "more_results": false,
    "next_offset": 0,
    "count": 2
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2016-06-01T12:27:42Z",
      "id": "40e364753c6d116624416df1cb05d3594fc20ea5",
      "author": "Bruce Wayne",
      "email": "bruce@waynecorp.com",
      "description": "Simplify exception handling #123",
      "timestamp": "2016-06-01T12:27:07Z",
      "repository": {
        "id": "chef",
        "uuid": "9ddb4b81-5f7b-42c9-9860-1e519a75b483"
      },
      "project": {
        "id": "luotsi",
        "uuid": "9c54e6ca-402b-4db9-9bc1-fcb6d20f935a"
      }
    },
    {
      "api_status": 200,
      "api_timestamp": "2016-06-01T12:27:42Z",
      "id": "efb518e2e295a8f7b2d08cc5151969f7aeb6d371",
      "author": "Bruce Wayne",
      "email": "bruce@waynecorp.com",
      "description": "Remove deprecated methods. #123 & #124",
      "timestamp": "2016-06-01T12:25:57Z",
      "repository": {
        "id": "chef",
        "uuid": "9ddb4b81-5f7b-42c9-9860-1e519a75b483"
      },
      "project": {
        "id": "luotsi",
        "uuid": "9c54e6ca-402b-4db9-9bc1-fcb6d20f935a"
      }
    }
  ]
}
```

Company

All data accessed through Helix TeamHub APIs is scoped under a company. As a Helix TeamHub user, you only have access to your own company.

- Attributes
- Child objects
- Operations
  - `GET /company`
  - `PUT /company`

Attributes

id

A URL-friendly identifier for the company.

- Type: `string`
- Unique: `true`
- Required: `true`
- Minimum length: `2`
- Maximum length: `100`
- Allowed characters: `a-z`, `A-Z`, `0-9`, `-`, `_`

name

The human-readable name of the company.

- Type: `string`
- Required: `true`
- Minimum length: `2`
- Maximum length: `100`

display_name

Same as the `name` of the company or value of the `id` if the name is not set.

- Type: `string`

Child objects

The following child objects can be attached to the company returned, to eliminate the need for requesting them separately:

Object	Attribute
User	`projects`
Collaborator	`collaborators`
Repository	`repositories`
Project	`projects`

For example, to get all of the company's users in full and the number of repositories it has, attach **`include=users&count=repositories`** to your GET request.

See the child object API documentation for a description of the possible operations.

Operations

GET /company

Returns a company object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/company
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-06-13T12:53:47Z",
  "id": "eficode",
  "created_at": "2011-08-23T21:32:24Z",
  "updated_at": "2013-05-29T10:47:05Z",
  "name": "Eficode",
  "display_name": "Eficode"
}
```

PUT /company

Updates a company. Returns the updated company object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
```

```
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "name": "Eficode Oy" }' \
  https://helixteamhub.cloud/api/company
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-06-13T13:21:32Z",
  "id": "eficode",
  "created_at": "2011-08-23T21:32:24Z",
  "updated_at": "2013-06-13T13:21:32Z",
  "name": "Eficode Oy",
  "display_name": "Eficode Oy"
}
```

Default Reviewer

With default reviewers, you define which project members are automatically added as reviewers to code reviews of a repository.

- "Attributes " below
- "Operations" on the next page
  - **`GET /projects/:project_id/repositories/:repository_id/default_reviewers`**
  - **`POST /projects/:project_id/repositories/:repository_id/default_reviewers`**
  - **`DELETE /projects/:project_id/repositories/:repository_id/default_reviewers/:person_id`**

Attributes

repository

The ID of the repository that the membership is for. When requesting, you can expand it to the full repository object by including the query parameter **`expand=repository`**.

- Type: **`string`**
- Required: **`true`**

person

The ID of the person (**`user`**/**`collaborator`**). When requesting, you can expand it to the full person object by including the query parameter **`expand=person`**.

- Type: **`string`**
- Required: **`true`**

Operations

GET /projects/:project_id/repositories/:repository_id/default_reviewers

Lists all default reviewers for a repository, with optional limit and offset parameters, as a metadata-results object.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/default_
reviewers
```

Example response

```
{
  "metadata": {
    "more_results": false,
    "next_offset": 0,
    "total_count": 1,
    "count": 1
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2018-10-25T09:29:37Z",
      "repository": {
        "id": "acme",
        "uuid": "f90da853-c5f2-4e55-9fa5-773e83b78784"
      },
      "person": {
        "id": "johndoe",
        "uuid": "a942b42f-14d8-4675-b9e4-ce4a4e5f69de",
        "type": "user"
      }
    }
  ]
}
```

POST /projects/:project_id/repositories/:repository_id/default_reviewers

Adds a default reviewer for a repository. The person must have a project or repository role, either directly or through a group.

Example request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "a942b42f-14d8-4675-b9e4-ce4a4e5f69de" }' \

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/default_
reviewers
```

Example response

```
{
  "api_status": 201,
  "api_timestamp": "2018-10-25T09:29:37Z",
  "repository": {
    "id": "acme",
    "uuid": "f90da853-c5f2-4e55-9fa5-773e83b78784"
  },
  "user": {
    "id": "johndoe",
    "uuid": "a942b42f-14d8-4675-b9e4-ce4a4e5f69de",
    type: "user"
  }
}
```

DELETE /projects/:project_id/repositories/:repository_id/default_reviewers/:person_id

Removes a person from default reviewers. Returns the removed default reviewer.

Example request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/default_
reviewers/a942b42f-14d8-4675-b9e4-ce4a4e5f69de
```

Example response

```
"api_status": 200,
  "api_timestamp": "2018-11-28T10:12:28Z",
  "repository": {
    "id": "acme",
    "uuid": "f90da853-c5f2-4e55-9fa5-773e83b78784"
  },
  "user": {
    "id": "johndoe",
    "uuid": "a942b42f-14d8-4675-b9e4-ce4a4e5f69de",
    "type": "user"
  }
}
```

Event

An event is a historical entry of some user or bot action that took place in Helix TeamHub. An event identifies an *action* that has a *subject* (a Helix TeamHub account id), and a specific *point of time*.

Events can also be live. When an event is created, it is usually also marked complete by setting the progress value as 100. However, if an event has a progress value below 100, has a fairly recent timestamp and has also been updated (i.e. the progress value has been incremented) recently, an event is interpreted as a live event. You can expect live updates to these events and show this to the user.

The action-target pair of values is a powerful way of describing all the meaningful things people can do in Helix TeamHub. The author and the hash of linked objects identify the data objects related to the event.

Attributes

id

A URL-friendly identifier for the event.

- Type: `string`
- Unique: `true` (in company scope)

subject

The subject of the event (the identifier of a User or Collaborator). It can be expanded to get the full objects by including the query parameter `expand=subject` in the request.

- Type: `string` / `object`

timestamp

Date and time of the action, in ISO-8601 format.

- Type: `timestamp`

operation

A verb describing what the author has done with the target. One of `added`, `assigned`, `completed`, `created`, `deleted`, `failed`, `reassigned`, `removed`, `unvoted`, `updated`, `voted`.

- Type: `string`

target

A noun describing what the object of the action is. One of `bookmark`, `bot`, `branch`, `build`, `code_review_comment`, `code_review_description`, `code_review_line_comment`, `code_review_require_build`, `code_review_state`, `code_review_threshold`, `code_review_title`, `code_review`, `collaborator`, `comment_reply`, `commit_comment`, `commit_line_comment`, `event_comment`, `milestone`, `project`, `project_short_name`, `project_name`, `project_description`, `project_visibility`, `project_collaborator`, `project_description`, `project_group`, `project_hook`, `project_name`, `project_short_name`, `project_user`, `project_visibility`, `project`, `push`, `repository_hook`, `repository`, `reviewer`, `tag`, `task_branch`, `task_comment`, `task_commit`, `task_description`, `task_member`, `task_milestone`, `task_state`, `task_title`, `task`, `user`.

- Type: `string`

ref

Only applicable to `push`, `branch`, `bookmark` and `tag` targets. A string describing the Branch, Bookmark or Tag affected (e.g. `master`).

- Type: `string`

bookmarks

Only applicable to `push` and `build` targets. A list of bookmarks' names related to the event.

- Type: `array`

commits

Only applicable to `push`, `branch`, `build` and `task_commit` targets. A list of commit identifiers related to the event. The list may contain at maximum 100 commit identifiers. It can be expanded to get the full Commit objects by including the query parameter `expand=objects.commits` in the request.

- Type: `array`

commit_count

Only applicable to `push` and `branch` targets. The total number of commit identifiers related to the event. Note that the count might be larger than the size of `commits` array.

- Type: `int`

last_commit

Only applicable to `push` and `branch` targets. The last commit identifier related to the event. Note that this commit identifier might not be in the `commits` array.

- Type: `string`

files

Only applicable to `push` target. Contains details about changed files.

- Type: `object`

```
"files": {
  "added": [
    "app/model/user.rb",
    "spec/models/user_spec.rb"
  ],
  "modified": [
    "readme.md",
  ],
  "deleted": [
    "user.rb",
  ]
}
```

code_reviews

Only applicable to `push` and `build` targets. A list of Code Reviews related to the event. It can be expanded to get the full Code Review objects by including the query parameter `expand=objects.code_reviews` in the request.

- Type: `array`

role

Only applicable to `project_user`, `project_collaborator`, `project_group` and `bot` targets. The role the User or Collaborator has directly in the project.

short_name

Only applicable to `project` target. The URL-friendly identifier for the Project.

- Type: `string`

name

Only applicable to `build` and `project` targets. The name of the build (e.g. `Development`) or the name of the Project.

- Type: `string`

description

Only applicable to `project_description`, `task_description` and `code_review_description` targets. The free-form description of the Project, Task or Code Review.

- Type: `string`

title

Only applicable to `task_title` and `code_review_title` targets. The state for the Task or Code Review.

- Type: `string`

state

Only applicable to `task_state` and `code_review_state` targets. The human-readable title for the Task or Code Review.

- Type: `string`

visibility

Only applicable to `project` target. The visibility flag of the Project.

- Type: `string`

objects

A hash of linked objects related to the event. The interpretation of this value depends on the action-target pair of the event.

For example: `{"project": { "id": "luotsi" }, "repository": {"id": "chef" }}`.

- Type: `object`

progress

The progress value of an event, as percentage. Must be between 0 and 100. When progress is 100, the event is considered complete.

- Type: `int`

resources

A list of links to external resources related to the event, usually defined as an array of URLs to link to.

Example: `["http://jenkins.eficode.com/project"]`.

- Type: `array`

Operations

GET /events

Get all events within the company. Results may be restricted using query parameters. Returned as a metadata-results object.

Example Request: Get two latest events for a project

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/events?project=luotsi&limit=2
```

Example Response

```
{
  "metadata": {
    "more_results": true,
    "next_offset": 2,
    "count": 2
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2012-10-25T09:47:30Z",
      "id": "50883cfe1229c02952000036",
      "created_at": "2012-10-24T19:09:50Z",
      "updated_at": "2012-10-24T19:09:50Z",
      "subject": {
        "id": "assimov"
      },
      "operation": "created",
      "target": "push",
      "progress": 100,
      "objects": {
          "project": {
            "id": "luotsi"
          },
          "repository": {
            "id": "devrel"
          },
          "commits": [
              {
                "id": "a6e0210cdd2f473104d75c35a7d1052e12b11b46"
              },
              {
                "id": "393fe81b0f6c39f3006f748f5b16cdddcbe01c80"
              }
          ],
          "last_commit": "393fe81b0f6c39f3006f748f5b16cdddcbe01c80",
          "commit_count": 2,
          "ref": "master"
      },
      "timestamp": "2012-10-24T19:09:50Z"
    },
    {
      "api_status": 200,
      "api_timestamp": "2012-10-25T09:47:30Z",
      "id": "50880fe31229c0295a00002d",
      "created_at": "2012-10-24T15:57:23Z",
      "updated_at": "2012-10-24T15:57:23Z",
      "subject": {
        "id": "jlaiho"
      },
      "operation": "created",
      "target": "push",
      "progress": 100,
      "objects": {
          "project": {
            "id": "luotsi"
          },
          "repository": {
            "id": "luotsi"
          },
          "commits": [
              {
                "id": "b63c10447d5c0f1b7f013af7ffe56742b449d100"
              },
              {
                "id": "831c9809309e640a5191a639a7f002012f2a842c"
              }
              ...
          ]
          "last_commit": "a6e0210cdd2f473104d75c35a7d1052e12b11b46",
          "commit_count": 2815
      },
      "timestamp": "2012-10-24T15:57:23Z"
    }
  ]
}
```

Example Request: Get an event for the most recently added project user in a project

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \

https://helixteamhub.cloud/api/events?project=luotsi&limit=1&operation=created&target=project_user
```

Example Response

```
{
  "metadata": {
    "more_results": true,
    "next_offset": 1,
    "count": 1
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2012-10-25T09:58:48Z",
      "id": "50879b531229c06d30000014",
      "created_at": "2012-10-24T07:40:03Z",
      "updated_at": "2012-10-24T07:40:03Z",
      "subject": {
        "id": "thuovinen"
      },
      "operation": "created",
      "target": "project_user",
      "progress": 100,
      "objects": {
          "user": {
            "id": "etuupainen"
          },
          "project": {
            "id": "luotsi"
          },
          "role": "guest"
      },
      "timestamp": "2012-10-24T07:40:03Z"
    }
  ]
}
```

GET /events/:id

Get a specific event.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \
  https://helixteamhub.cloud/api/events/50883cfe1229c02952000036
```

Example Response

```
{
    "api_status": 200,
    "api_timestamp": "2012-10-25T09:47:30Z",
    "id": "50883cfe1229c02952000036",
    "created_at": "2012-10-24T19:09:50Z",
    "updated_at": "2012-10-24T19:09:50Z",
    "subject": {
      "id": "assimov"
    },
    "operation": "created",
    "target": "repository",
    "progress": 100,
    "objects": {
        "project": {
          "id": "luotsi"
        },
        "repository": {
          "id": "devrel"
        }
    },
    "timestamp": "2012-10-24T19:09:50Z"
}
```

POST /events

Create a new event. Only build events can be created through API by using a bot account. Operation can be either "completed" or "failed", name is optional.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "target": "build", "operation": "completed", "project": "luotsi",
"repository": "chef", "name": "Development", "commits":
["d847a7a20f547b8990c31627599aee47441018e6"], "resources":
["http://domain.com/jenkins/job/development/2372"] }' \
  https://helixteamhub.cloud/api/events
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2013-08-15T14:48:02Z",
  "id": "520cea221cf3593b29000004",
  "created_at": "2013-08-15T14:48:02Z",
  "updated_at": "2013-08-15T14:48:02Z",
  "subject": {
    "id": "luotsi-ci-bot"
  },
  "operation": "completed",
  "target": "build",
  "progress": 100,
  "objects": {
    "project": {
      "id": "luotsi"
    },
    "repository": {
      "id": "chef"
    },
    "commits": [
      {
        "id": "d847a7a20f547b8990c31627599aee47441018e6"
      }
    ],
    "resources": [
      "http://domain.com/jenkins/job/development/2372"
    ],
    "name": "Development"
  },
  "timestamp": "2013-08-15T14:48:02Z"
}
```

PUT /events/:id

Only progress and operation of an incomplete build event can be updated.

Example request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "progress": 100, "operation": "completed" }' \
  https://helixteamhub.cloud/api/events/5211cf501cf3592d2100000d
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2013-08-19T07:55:39Z",
  "id": "5211cf501cf3592d2100000d",
  "created_at": "2013-08-19T07:54:56Z",
  "updated_at": "2013-08-19T07:55:39Z",
  "subject": {
    "id": "luotsi-ci-bot"
  },
  "operation": "completed",
  "target": "build",
  "progress": 100,
  "objects": {
    "project": {
      "id": "luotsi"
    },
    "repository": {
      "id": "chef"
    },
    "commits": [
      {
        "id": "d847a7a20f547b8990c31627599aee47441018e6"
      }
    ],
    "resources": [
      "http://domain.com/jenkins/job/development/2372"
    ],
    "name": "Development"
  },
  "timestamp": "2013-08-19T07:54:56Z"
}
```

Group Member

Group member is the link object between Groups and Users.

- Attributes
- Operations
  - **`GET /groups/:group_id/members`**
  - **`GET /groups/:group_id/members/:user_id`**
  - **`POST /groups/:group_id/members`**
  - **`PUT /groups/:group_id/members/:user_id`**
  - **`DELETE /groups/:group_id/members/:user_id`**

Attributes

group

The id of the group the membership is for.

When requesting, can be expanded to the full group object by including the query parameter `expand=group`.

- Type: `string`
- Required: `true`

user

The id of the user the membership is for.

When requesting, can be expanded to the full user object by including the query parameter `expand=user`.

- Type: `string`
- Required: `true`

role

The role the user has in the group. Users with admin role can manage memberships for the group.

- Type: `string`
- Required: `true`
- Allowed values: `admin`, `member`

linked

Tells whether the user is a group member through an LDAP group.

- Type: `boolean`
- Default: false

Operations

GET /groups/:group_id/members

List all the users for a group, with optional limit and offset parameters, as a metadata-results object.

GET /groups/:group_id/member/:user_id

Return a specific group user.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/groups/devs/jlaiho
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2013-11-28T08:36:55Z",
  "group": {
    "id": "devs"
  },
  "user": {
    "id": "jlaiho"
  },
  "role": "admin",
  "linked": false
}
```

POST /groups/:group_id/members

Add a user to a group. Returns the created group user.

Example request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "jlaiho", "role": "member" }' \
  https://helixteamhub.cloud/api/groups/devs/members
```

Example response

```
{
  "api_status": 201,
  "api_timestamp": "2013-11-28T09:07:19Z",
  "group": {
    "id": "devs"
  },
  "user": {
    "id": "jlaiho"
  },
  "role": "member",
  "linked": false
}
```

PUT /groups/:group_id/members/:user_id

Update role for a group user. Returns the updated group user.

Example request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "role": "admin" }' \
  https://helixteamhub.cloud/api/groups/devs/members/jlaiho
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2013-11-28T09:10:16Z",
  "group": {
    "id": "devs"
  },
  "user": {
    "id": "jlaiho"
  },
  "role": "admin",
  "linked": false
}
```

DELETE /groups/:group_id/members/:user_id

Remove a user from a group. Returns the removed group user.

Example request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/groups/devs/members/jlaiho
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2013-11-28T09:11:50Z",
  "group": {
    "id": "devs"
  },
  "user": {
    "id": "jlaiho"
  },
  "role": "admin",
  "linked": false
}
```

Group

Group is a collection of Users. Groups can be linked to Projects by assigning them a project role, which has the same effect as assigning a role for each group member individually.

- Attributes
- Searchable attributes
- Child objects
- Operations
    - `GET /groups`
    - `GET /groups/:id`
    - `POST /groups`
    - `PUT /groups/:id`
    - `DELETE /groups/:id`

Attributes

id

URL-friendly identifier for the group.

- Type: `string`
- Immutable: `true`
- Unique: `true`
- Required: `true`
- Minimum length: `2`
- Maximum length: `100`
- Allowed characters: `a-z`, `A-Z`, `0-9`, `-`, `_`

name

Human-readable name for the group.

- Type: `string`
- Unique: `true`
- Required: `true`
- Minimum length: `2`
- Maximum length: `100`

description

A free-form description of the group.

- Type: `string`
- Default: `""`
- Maximum length: `512`

visibility

Public groups are visible to everyone in the company and anyone can take them into use in their projects. Private groups are only visible to members of the group and only they can manage project roles for the group.

- Type: `string`
- Default: `public`
- Allowed values: `public`, `private`

source

The identifier of the external source where the group is synchronized from.

- Type: `string`
- Default: `""`
- Maximum length: `500`

linked

A group in considered as linked when its source has a value. This field is automatically updated.

- Type: `boolean`
- Default: `false`

Searchable attributes

Groups can be searched by their `id` and `name` attributes. See the search documentation for instructions how to do this.

Child objects

The following child objects can be attached to the groups returned, to eliminate the need for requesting them separately:

Object	Attribute
User	`users`

For example to get all the group's users, attach **`include=users`** to your GET request.

See the child object API documentation for a description of the possible operations.

Operations

GET /groups

Returns all the groups in the company, with optional limit and offset parameters, as a metadata-results object.

GET /groups/:id

Returns specific group object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/groups/designers
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-11-27T15:04:40Z",
  "id": "designers",
  "created_at": "2013-11-18T12:22:15Z",
  "updated_at": "2013-11-27T11:43:04Z",
  "name": "Designers",
  "description": "All designers in our company",
  "visibility": "public",
  "source": "I_DESIGNERS",
  "linked": true
}
```

POST /groups

Creates a new group to the company. Returns the created group object.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "devs", "name": "Developers" }' \
  https://helixteamhub.cloud/api/groups
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2013-11-27T15:24:49Z",
  "id": "devs",
  "created_at": "2013-11-27T15:24:49Z",
  "updated_at": "2013-11-27T15:24:49Z",
  "name": "Developers",
  "description": "",
  "visibility": "public",
  "source": "",
  "linked": false
}
```

PUT /groups/:id

Updates a group. Returns the updated group object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "source": "I_DEVS" }' \
  https://helixteamhub.cloud/api/groups/devs
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-11-27T15:25:51Z",
  "id": "devs",
  "created_at": "2013-11-27T15:24:49Z",
  "updated_at": "2013-11-27T15:25:51Z",
  "name": "Developers",
  "description": "",
  "visibility": "public",
  "source": "I_DEVS",
  "linked": true
}
```

DELETE /groups/:id

Deletes a group. Returns the deleted group object.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/groups/devs
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-11-27T15:27:28Z",
  "id": "devs",
  "created_at": "2013-11-27T15:24:49Z",
  "updated_at": "2013-11-27T15:25:51Z",
  "name": "Developers",
  "description": "",
  "visibility": "public",
  "source": "I_DEVS",
  "linked": true
}
```

Hook Service

A hook service is an external service that can be notified when an event happens in Helix TeamHub.

- Attributes
- Operations
  - `GET /hook_services`
  - `GET /hook_services/:id`

Attributes

id

The identifier of the service.

- Type: `string`

name

The name of the service.

- Type: `string`

readme_raw

The documentation for the service, usually in markdown format.

- Type: `string`

attributes

An array of key value pairs of the possible configurations for the service. Each entry in the array contains type and name keys. The value of type key can be one of the following: `string`, `boolean`, `password`.

- Type: `array`

events

An array of events the service is able to handle. The possible values are `project`, `repository` and `push`.

- Type: `array`

hookables

An array of hookables the service is able to handle. The possible values are `company`, `project` and `repository`.

- Type: `array`

Operations

GET /hook_services

Returns all available hook services, with optional limit and offset parameters, as a metadata-results object.

GET /hook_services/:id

Returns the specific hook service.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/hook_services/flowdock
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-07-10T11:25:04Z",
  "id": "flowdock",
  "name": "Flowdock",
  "readme_raw": "Broadcast commits to Flowdock.",
  "attributes": [
    {
      "type": "string",
      "name": "token"
    }
  ],
  "events": [
    "push"
  ],
  "hookables": [
    "repository"
  ]
}
```

Hook

Hooks can be used to notify external hook services about Helix TeamHub related events.

- Attributes
- Operations
  - `GET /hooks`
  - `GET /hooks/:id`
  - `POST /hooks`
  - `PUT /hooks/:id`
  - `DELETE /hooks/:id`
  - `GET /hooks/count`
  - `GET /projects/:project_id/hooks`
  - `GET /projects/:project_id/hooks/:id`
  - `POST /projects/:project_id/hooks`
  - `PUT /projects/:project_id/hooks/:id`
  - `DELETE /projects/:project_id/hooks/:id`
  - `GET /projects/:project_id/hooks/count`
  - `GET /projects/:project_id/repositories/:repository_id/hooks`

- `GET /projects/:project_id/repositories/:repository_id/hooks/:id`
- `POST /projects/:project_id/repositories/:repository_id/hooks`
- `PUT /projects/:project_id/repositories/:repository_id/hooks/:id`
- `DELETE /projects/:project_id/repositories/:repository_id/hooks/:id`
- `GET /projects/:project_id/repositories/:repository_id/hooks/count`

> **Note**
> Forward slash characters in `:repository_id` must be encoded with `%2F`

Attributes

id

A URL-friendly identifier for the hook.

- Type: `string`
- Unique: `true`

hook_service

The identifier of the Hook service. Full object can be expanded with GET requests.

- Type: `string` / `object`

events

An array of events to hook into.

- Type: `array`

config

An object containing the hook service configurations.

- Type: `object`

hookable_type

A type of `hookable` object. Possible values are `company`, `project` and `repository`.

- Type: `string`

hookable

A hookable object.

- Type: `object`

Operations

GET /hooks

Returns all hooks for a company, with optional limit and offset parameters, as a metadata-results object.

GET /hooks/:id

Returns a single hook.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \
  https://helixteamhub.cloud/api/hooks/578f5eea646576644d00000a
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-07-10T12:39:08Z",
  "id": "578f5eea646576644d00000a",
  "created_at": "2013-07-05T12:44:01Z",
  "updated_at": "2013-07-09T07:08:17Z",
  "hook_service": {
    "id": "webhook"
  },
  "events": [
    "project"
  ],
  "config": { "url": "http://requestb.in/foo" },
  "hookable_type": "company",
  "hookable": {
    "id": "hth",
    "uuid": "f38c3579-d43f-4f7e-b74a-b15d1e32ef46"
  }
}
```

POST /hooks

Creates a new hook for a company. Returns the created hook object.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "hook_service": "webhook", "events": ["project"], "config": {
"url": "http://requestb.in/foo" } }' \
  https://helixteamhub.cloud/api/hooks
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2013-07-10T13:17:27Z",
  "id": "578f5eea646576644d00000a",
  "created_at": "2013-07-10T13:17:27Z",
  "updated_at": "2013-07-10T13:17:27Z",
  "hook_service": {
    "id": "webhook"
  },
  "events": [
    "project"
  ],
  "config": { "token": "http://requestb.in/foo" },
  "hookable_type": "company",
  "hookable": {
    "id": "hth",
    "uuid": "f38c3579-d43f-4f7e-b74a-b15d1e32ef46"
  }
}
```

PUT /hooks/:id

Updates a hook for a company. Returns the updated hook object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "hook_service": "webhook", "events": ["project"], "config":
{"token": "http://requestb.in/bar"} }' \
  https://helixteamhub.cloud/api/hooks/578f5eea646576644d00000a
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-07-10T15:28:29Z",
  "id": "578f5eea646576644d00000a",
  "created_at": "2013-07-10T15:26:00Z",
  "updated_at": "2013-07-10T15:28:29Z",
  "hook_service": {
    "id": "webhook"
  },
  "events": [
    "project"
  ],
  "config": { "token": "http://requestb.in/bar" },
  "hookable_type": "company",
  "hookable": {
    "id": "hth",
    "uuid": "f38c3579-d43f-4f7e-b74a-b15d1e32ef46"
  }
}
```

DELETE /hooks/:id

Deletes a hook from a company. Return the deleted hook object.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/hooks/578f5eea646576644d00000a
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-07-10T13:19:48Z",
  "id": "578f5eea646576644d00000a",
  "created_at": "2013-07-10T13:17:27Z",
  "updated_at": "2013-07-10T13:17:27Z",
  "hook_service": {
    "id": "webhook"
  },
  "events": [
    "project"
  ],
  "config": { "token": "http://requestb.in/foo" },
  "hookable_type": "company",
  "hookable": {
    "id": "hth",
    "uuid": "f38c3579-d43f-4f7e-b74a-b15d1e32ef46"
  }
}
```

GET /hooks/count

Returns an object containing the amounts of total, active, and inactive hooks in the company.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/hooks/count
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2016-07-26T11:59:44Z",
  "count": 2,
  "active": 2,
  "inactive": 0
}
```

GET /projects/:project_id/hooks

Returns all hooks for a project, with optional limit and offset parameters, as a metadata-results object.

GET /projects/:project_id/hooks/:id

Returns a single hook.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
https://helixteamhub.cloud/api/projects/luotsi/hooks/51dd5ee71cf35944de000004
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-07-10T12:39:08Z",
  "id": "51dd5ee71cf35944de000004",
  "created_at": "2013-07-05T12:44:01Z",
  "updated_at": "2013-07-09T07:08:17Z",
  "hook_service": {
    "id": "webhook"
  },
  "events": [
    "repository"
  ],
  "config": { "url": "http://requestb.in/foo" },
  "hookable_type": "project",
  "hookable": {
    "id": "luotsi",
    "uuid": "a671a3bb-edb9-45f7-aa71-f3be44a075c2"
  }
}
```

POST /projects/:project_id/hooks

Creates a new hook for a project. Returns the created hook object.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "hook_service": "webhook", "events": ["repository"], "config": {
```

```
"url": "http://requestb.in/foo" } }' \
  https://helixteamhub.cloud/api/projects/luotsi/hooks
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2013-07-10T13:17:27Z",
  "id": "51dd5ee71cf35944de000004",
  "created_at": "2013-07-10T13:17:27Z",
  "updated_at": "2013-07-10T13:17:27Z",
  "hook_service": {
    "id": "webhook"
  },
  "events": [
    "repository"
  ],
  "config": { "token": "http://requestb.in/foo" },
  "hookable_type": "project",
  "hookable": {
    "id": "luotsi",
    "uuid": "a671a3bb-edb9-45f7-aa71-f3be44a075c2"
  }
}
```

PUT /projects/:project_id/hooks/:id

Updates a hook for a project. Returns the updated hook object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "hook_service": "webhook", "events": ["repository"], "config":
{"token": "http://requestb.in/bar"} }' \
https://helixteamhub.cloud/api/projects/luotsi/hooks/51dd5ee71cf35944de000
004
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-07-10T15:28:29Z",
  "id": "51dd5ee71cf35944de000004",
  "created_at": "2013-07-10T15:26:00Z",
  "updated_at": "2013-07-10T15:28:29Z",
  "hook_service": {
    "id": "webhook"
  },
  "events": [
    "repository"
  ],
  "config": { "token": "http://requestb.in/bar" },
  "hookable_type": "project",
  "hookable": {
    "id": "luotsi",
    "uuid": "a671a3bb-edb9-45f7-aa71-f3be44a075c2"
  }
}
```

DELETE /projects/:project_id/hooks/:id

Deletes a hook from a project. Return the deleted hook object.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
https://helixteamhub.cloud/api/projects/luotsi/hooks/51dd5ee71cf35944de000
004
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-07-10T13:19:48Z",
  "id": "51dd5ee71cf35944de000004",
  "created_at": "2013-07-10T13:17:27Z",
  "updated_at": "2013-07-10T13:17:27Z",
  "hook_service": {
    "id": "webhook"
  },
  "events": [
    "repository"
  ],
  "config": { "token": "http://requestb.in/foo" },
  "hookable_type": "project",
  "hookable": {
    "id": "luotsi",
    "uuid": "a671a3bb-edb9-45f7-aa71-f3be44a075c2"
  }
}
```

GET /projects/:project_id/hooks/count

Returns an object containing the amounts of total, active, and inactive hooks in the project.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/hooks/count
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2016-07-26T12:02:42Z",
  "count": 2,
  "active": 2,
  "inactive": 0
}
```

GET /projects/:project_id/repositories/:repository_id/hooks

Returns all hooks for a repository, with optional limit and offset parameters, as a metadata-results object.

GET /projects/:project_id/repositories/:repository_id/hooks/:id

Returns a single hook.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/hooks/51dd5ee71cf35944de000004
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-07-10T12:39:08Z",
  "id": "51dd5ee71cf35944de000004",
  "created_at": "2013-07-05T12:44:01Z",
  "updated_at": "2013-07-09T07:08:17Z",
  "hook_service": {
    "id": "flowdock"
  },
  "events": [
    "push"
  ],
  "config": { "token": "abc" },
  "hookable_type": "repository",
  "hookable": {
    "id": "chef",
    "uuid": "62c5505f-75da-41ed-bc04-de43405d22b1"
  }
}
```

POST /projects/:project_id/repositories/:repository_id/hooks

Creates a new hook for a repository. Returns the created hook object.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "hook_service": "flowdock", "events": ["push"], "config":
{"token": "abc"} }' \
  https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/hooks
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2013-07-10T13:17:27Z",
  "id": "51dd5ee71cf35944de000004",
  "created_at": "2013-07-10T13:17:27Z",
  "updated_at": "2013-07-10T13:17:27Z",
  "hook_service": {
    "id": "flowdock"
  },
  "events": [
    "push"
  ],
  "config": {"token":"abc"},
  "hookable_type": "repository",
  "hookable": {
    "id": "chef",
    "uuid": "62c5505f-75da-41ed-bc04-de43405d22b1"
  }
}
```

PUT /projects/:project_id/repositories/:repository_id/hooks/:id

Updates a hook for a repository. Returns the updated hook object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "hook_service": "flowdock", "events": ["push"], "config":
{"token": "foobar"} }' \

https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/hooks/51d
d5ee71cf35944de000004
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-07-10T15:28:29Z",
  "id": "51dd5ee71cf35944de000004",
  "created_at": "2013-07-10T15:26:00Z",
  "updated_at": "2013-07-10T15:28:29Z",
  "hook_service": {
    "id": "flowdock"
  },
  "events": [
    "push"
  ],
  "config": {"token":"foobar"},
  "hookable_type": "repository",
  "hookable": {
    "id": "chef",
    "uuid": "62c5505f-75da-41ed-bc04-de43405d22b1"
  }
}
```

DELETE /projects/:project_id/repositories/:repository_id/hooks/:id

Deletes a hook from a repository. Return the deleted hook object.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/hooks/51d
d5ee71cf35944de000004
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-07-10T13:19:48Z",
  "id": "51dd5ee71cf35944de000004",
  "created_at": "2013-07-10T13:17:27Z",
  "updated_at": "2013-07-10T13:17:27Z",
  "hook_service": {
    "id": "flowdock"
  },
  "events": [
    "push"
  ],
  "config": {"token":"abc"},
  "hookable_type": "repository",
  "hookable": {
    "id": "chef",
    "uuid": "62c5505f-75da-41ed-bc04-de43405d22b1"
  }
}
```

GET /projects/:project_id/repositories/:repository_id/hooks/count

Returns an object containing the amounts of total, active, and inactive hooks in the repository.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/hooks/cou
nt
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2016-07-26T12:05:14Z",
  "count": 2,
  "active": 2,
  "inactive": 0
}
```

License

Information about the company's current Helix TeamHub license.

Attributes

seats_used

Number of seats used.

- Type: `int`

seats

Total number of seats available.

- Type: `int`

```
academic
```

Whether or not the license is academic.

- Type: `boolean`

Operations

GET /company/license

Returns the company's license information.

Example Request

```
curl -X GET
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'"
  https://helixteamhub.cloud/api/company/license
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-01-25T15:00:03Z",
  "seats_used": 53,
  "seats": 100,
  "academic": false
}
```

Milestone

Milestone is a collection of Tasks in a Project.

- Attributes
- Child objects
- Operations
  - `GET /projects/:project_id/milestones`
  - `GET /projects/:project_id/milestones/:id`
  - `POST /projects/:project_id/milestones`
  - `PUT /projects/:project_id/milestones/:id`
  - `DELETE /projects/:project_id/milestones/:id`

Attributes

Milestone object has support for storing custom attributes.

id

URL-friendly identifier for the milestone.

- Type: `string`
- Immutable: `true`
- Unique: `true`
- Required: `true`
- Minimum length: `2`
- Maximum length: `100`
- Allowed characters: `a-z`, `A-Z`, `0-9`, `-`, `_`

title

Human-readable title for the milestone.

- Type: `string`
- Required: `true`
- Minimum length: `1`
- Maximum length: `100`

description

A free-form description of the milestone.

- Type: `string`
- Default: `""`
- Maximum length: `50000`

default

An indicator of whether the milestone is a default one or not. Only one milestone can be set as a default within the project scope at the same time (setting this flag to `true` for one milestone unsets the same for the other ones).

- Type: `boolean`
- Default: `false`

states

Possible states for the tasks in the milestone. A milestone must define at least one state.

- Type: `array`
- Required: `true`

done_states

States that are considered as final task states in the milestone. Can be used to calculate progress of the milestone.

- Type: `array`

priorities

Possible priorities for the tasks in the milestone. Priorities are considered to be integer based when left as empty.

- Type: `array`

start_date

Optional start date of the milestone.

- Type: `timestamp`

end_date

Optional end date of the milestone.

- Type: `timestamp`

Child objects

The following child objects can be attached to the milestones returned, to eliminate the need for requesting them separately:

Object	Attribute
Task	`tasks`
User	`members`

For example to get all the tasks in the milestone, attach `include=tasks` to your GET request.

See the child object API documentation for a description of the possible operations.

Operations

GET /projects/:project_id/milestones

Returns all the milestones in the project, with optional limit and offset parameters, as a metadata-results object.

GET /projects/:project_id/milestones/:id

Returns specific milestone object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/milestones/default
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-05-12T12:41:39Z",
  "id": "default",
  "created_at": "2014-05-12T11:43:15Z",
  "updated_at": "2014-05-12T11:43:15Z",
  "title": "Default",
  "description": "The default milestone.",
  "default": false,
  "states": [
    "open",
    "wip",
    "blocked",
    "closed"
  ],
  "done_states": [
    "closed"
  ],
  "priorities": [
    "low",
    "normal",
    "high"
  ],
  "start_date": "2014-05-10T09:00:00Z",
  "end_date": null,
  "properties": {

  }
}
```

POST /projects/:project_id/milestones

Creates a new milestone to the project. Returns the created milestone object.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "default", "title": "Default milestone", "states": ["open"]
}' \
  https://helixteamhub.cloud/api/projects/luotsi/milestones
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2014-05-12T15:02:10Z",
  "id": "default",
  "created_at": "2014-05-12T15:02:10Z",
  "updated_at": "2014-05-12T15:02:10Z",
  "title": "Default milestone",
  "description": "",
  "default": false,
  "states": [
    "open"
  ],
  "done_states": [

  ],
  "priorities": [

  ],
  "start_date": null,
  "end_date": null,
  "properties": {

  }
}
```

PUT /projects/:project_id/milestones/:id

Updates a milestone. Returns the updated milestone object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "description": "Sample description" }' \
  https://helixteamhub.cloud/api/projects/luotsi/milestones/default
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-05-12T15:03:58Z",
  "id": "default",
  "created_at": "2014-05-12T15:02:10Z",
  "updated_at": "2014-05-12T15:03:58Z",
  "title": "Default milestone",
  "description": "Sample description",
  "default": false,
  "states": [
    "open"
  ],
  "done_states": [

  ],
  "priorities": [

  ],
  "start_date": null,
  "end_date": null,
  "properties": {

  }
}
```

DELETE /projects/:project_id/milestones/:id

Deletes a milestone. Returns the deleted milestone object.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/milestones/default
```

Password Recovery

Password recoveries can be generated and used when a User or Collaborator has forgotten their Helix TeamHub password and needs a new one.

Please note that operations in Password Recoveries API do not require authentication.

- Attributes
- Operations
  - `POST /account/password_recoveries`
  - `GET /account/password_recoveries/:id`
  - `PUT /account/password_recoveries/:id`

Attributes

id

A unique token used for resetting the password. Generated by Helix TeamHub and delivered to the user in question via email in the form of a password recovery link. Must be provided with GET and PUT requests.

- Type: `string`

email

The email address of the user whose password is being recovered. An email will be sent to the user, instructing them on the password recovery process.

- Type: `string`

password

The new password to set. Required on PUT. Validations depend on instance specific configurations.

- Type: `string`
- Default minimum length: `8`
- Default maximum length: `100`

Operations

POST /account/password_recoveries

Creates a new password recovery token. Attempts to find the user by email and sends a password recovery email if found. The endpoint will always return 201 (even if the email does not really exist).

Example Request

```
curl -X POST
  -H "Accept: application/vnd.hth.v1" \
  -F "email=clark.kent@eficode.com" \
  https://helixteamhub.cloud/api/account/password_recoveries
```

Example Response

```
{
  "api_status": 201
}
```

GET /account/password_recoveries/:id

Returns the user's email and its Company ID when found by the given password recovery token.

Example Request

```
curl -X GET
  -H "Accept: application/vnd.hth.v1" \
  https://helixteamhub.cloud/api/account/password_
recoveries/2eb8ba99d5b4fdbbe26abda420195eb192f6c2ed63f063180f40479eb6865e7
e
```

Example Response

```
{
  "api_status": 200,
  "company": "hth",
  "email": "clark.kent@company.com"
}
```

PUT /account/password_recoveries/:id

Resets the user's password. Returns the user's email and its Company ID on success and validation errors on failure.

Example Request

```
curl -X PUT
  -H "Accept: application/vnd.hth.v1" \
  -F "password=Clar!Ken7" \
  https://helixteamhub.cloud/api/account/password_
recoveries/2eb8ba99d5b4fdbbe26abda420195eb192f6c2ed63f063180f40479eb6865e7
e
```

Example Response

```
{
  "api_status": 200,
  "company": "hth",
  "email": "clark.kent@company.com"
}
```

Project Collaborator

Project Collaborator is the linking object between Projects and Collaborators. With them you define who belongs to which project and what they can do in those projects.

- Attributes
- Operations
  - **`GET /projects/:project_id/project_collaborators`**
  - **`GET /projects/:project_id/project_collaborators/:id`**

- `POST /projects/:project_id/project_collaborators`
- `PUT /projects/:project_id/project_collaborators/:id`
- `DELETE /projects/:project_id/project_collaborators/:id`

Attributes

project

The Project the membership is for. The full representation can be requested with `expand=project`.

- Type: `string`
- Required: `true`

collaborator

The Collaborators the membership is for. The full representation can be requested with `expand=collaborator`.

- Type: `string`
- Required: `true`

role

The role the Collaborators has directly in the Project. Possible values are `master`, `developer` and `guest`.

- Type: `string`
- Required: `true`

highest_role

The highest role the Collaborators has in the Project.

- Type: `string`

Operations

GET /projects/:project_id/project_collaborators

Returns all project collaborators for a Project, with optional limit and offset parameters, as a metadata-results object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \
  https://helixteamhub.cloud/api/projects/platform/project_collaborators
```

Example Response

```
{
  "metadata": {
    "more_results": false,
    "next_offset": 6,
    "count": 6
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2014-11-25T14:18:07Z",
      "project": {
        "id": "platform"
      },
      "collaborator": {
        "id": "norris"
      },
      "role": "guest",
      "highest_role": "guest"
    },
    ...
  ]
}
```

GET /projects/:project_id/project_collaborators/:id

Returns a specific project collaborator in a Project.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/platform/project_
collaborators/norris
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-11-25T14:18:07Z",
  "project": {
    "id": "platform"
  },
  "collaborator": {
    "id": "norris"
  },
  "role": "guest",
  "highest_role": "guest"
}
```

Example request with `collaborator` and `project` fields expanded

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/platform/project_
collaborators/norris?expand=collaborator,project
```

Example response with `collaborator` and `project` fields expanded

```
{
  "api_status": 200,
  "api_timestamp": "2014-11-25T14:18:07Z",
  "project": {
    "api_status": 200,
    "api_timestamp": "2014-11-25T14:30:51Z",
    "id": "platform",
    "created_at": "2014-10-07T15:58:59Z",
    "updated_at": "2014-11-07T09:28:16Z",
    "name": "Platform",
    "description": "",
    "color": "blue",
    "visibility": "company",
    "labels": [

    ],
    "properties": {

    },
    "hook_events": [
      "repository"
    ]
  },
  "collaborator": {
    "api_status": 200,
    "api_timestamp": "2014-11-25T14:30:51Z",
    "id": "norris",
    "created_at": "2014-11-17T07:13:48Z",
    "updated_at": "2014-11-17T16:56:24Z",
    "type": "collaborator",
    "email": "norris@acme.org",
    "first_name": "Chuck",
    "last_name": "Norris",
    "display_name": "Chuck Norris",
    "company_admin": false,
    "instance_admin": false,
    "description": "",
    "phone": "",
    "title": "",
    "avatar": {
      "large": "",
      "medium": "",
      "small": ""
    },
    "synchronized_fields": [

    ],
    "locale": "en",
    "source": ""
  },
  "role": "guest",
  "highest_role": "guest"
}
```

POST /projects/:project_id/project_collaborators

Adds a Collaborator to a project. Returns the added project collaborator.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "norris", "role": "guest" }' \
  https://helixteamhub.cloud/api/projects/platform/project_collaborators
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2014-11-25T14:18:07Z",
  "project": {
    "id": "platform"
  },
  "collaborator": {
    "id": "norris"
  },
  "role": "guest",
  "highest_role": "guest"
}
```

PUT /projects/:project_id/project_collaborators/:id

Updates a project collaborator in a project. Returns the updated project collaborator.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "role": "developer" }' \
  https://helixteamhub.cloud/api/projects/platform/project_
collaborators/norris
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-11-25T14:18:07Z",
  "project": {
    "id": "platform"
  },
  "collaborator": {
    "id": "norris"
  },
  "role": "developer",
  "highest_role": "developer"
}
```

DELETE /projects/:project_id/project_collaborators/:id

Removes a collaborator from a project. Returns the removed project collaborator.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/platform/project_
collaborators/norris
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-11-25T14:18:07Z",
  "project": {
    "id": "platform"
  },
  "collaborator": {
    "id": "norris"
  },
  "role": "developer",
  "highest_role": "developer"
}
```

Project Group

Project group is the link object between Projects and Groups. With them you define which groups belongs to what project, and what they can do in the project.

- Attributes
- Operations
  - `GET /projects/:project_id/project_groups`
  - `GET /projects/:project_id/project_groups/:group_id`
  - `POST /projects/:project_id/project_groups`

- `PUT /projects/:project_id/project_groups/:group_id`
- `DELETE /projects/:project_id/project_groups/:group_id`

Attributes

project

The id of the project the membership is for.

When requesting, can be expanded to the full project object by including the query parameter `expand=project`.

- Type: `string`
- Required: `true`

group

The id of the group the membership is for.

When requesting, can be expanded to the full group object by including the query parameter `expand=group`.

- Type: `string`
- Required: `true`

role

The role the group has in the project. Must be one of `admin`, `manager`, `master`, `developer`, or `guest`.

- Type: `string`
- Required: `true`

Operations

GET /projects/:project_id/project_groups

List all project groups for a project, with optional limit and offset parameters, as a metadata-results object.

GET /projects/:project_id/project_groups/:group_id

Return a specific project group.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/project_groups/devs
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2013-11-28T09:38:42Z",
  "project": {
    "id": "luotsi"
  },
  "group": {
    "id": "devs"
  },
  "role": "developer"
}
```

POST /projects/:project_id/project_groups

Add a group to a project.

Example request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "devs", "role": "developer" }' \
  https://helixteamhub.cloud/api/projects/luotsi/project_groups
```

Example response

```
{
  "api_status": 201,
  "api_timestamp": "2013-11-28T09:41:47Z",
  "project": {
    "id": "luotsi"
  },
  "group": {
    "id": "devs"
  },
  "role": "developer"
}
```

PUT /projects/:project_id/project_groups/:group_id

Update role for a project group. Returns the updated project group.

Example request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "role": "admin" }' \
  https://helixteamhub.cloud/api/projects/luotsi/project_groups/devs
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2013-11-28T09:44:26Z",
  "project": {
    "id": "luotsi"
  },
  "group": {
    "id": "devs"
  },
  "role": "admin"
}
```

DELETE /projects/:project_id/project_groups/:group_id

Remove a group from a project. Returns the removed project group.

Example request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/project_groups/devs
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2013-11-28T09:45:50Z",
  "project": {
    "id": "luotsi"
  },
  "group": {
    "id": "devs"
  },
  "role": "admin"
}
```

Project Member

Project member is the link object between Projects and People (Users or Collaborators).

- Attributes
- Searchable attributes
- Operations
  - `GET /projects/:project_id/project_members`

Attributes

project

Project the membership is for.

When requesting, can be expanded to the full project object by including the query parameter `expand=project`.

- Type: `object`

member

Person the membership is for.

When requesting, can be expanded to the full person object by including the query parameter `expand=member`.

- Type: `object`

Searchable attributes

Project members can be searched by their `id`, `email`, and `display_name` attributes. See the search documentation for instructions how to do this.

Operations

GET /projects/:project_id/project_members

List all project members for the project, with optional limit and offset parameters, as a metadata-results object.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/project_members
```

Example response

```
{
  "metadata": {
    "more_results": false,
    "next_offset": 2,
    "count": 2
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2016-04-01T15:33:04Z",
      "project": {
        "id": "luotsi",
        "uuid": "743be99c-d627-4733-8a4f-30472573a1fb"
      },
      "member": {
        "id": "markowy",
        "uuid": "1c47ff18-d6c5-422c-8eb9-8873897012f0",
        "type": "user"
      }
    },
    {
      "api_status": 200,
      "api_timestamp": "2016-04-01T15:33:04Z",
      "project": {
        "id": "luotsi",
        "uuid": "743be99c-d627-4733-8a4f-30472573a1fb"
      },
      "member": {
        "id": "lfitzpatrick",
        "uuid": "91cbaa07-3985-4858-98db-1ffe54cbba2c",
        "type": "collaborator"
      }
    }
  ]
}
```

Example request with member and project fields expanded

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/tech/project_
members?expand=member,project
```

Example response with member and project fields expanded

```
{
  "metadata": {
    "more_results": false,
    "next_offset": 1,
    "count": 1
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2016-04-01T15:36:30Z",
      "project": {
        "api_status": 200,
        "api_timestamp": "2016-04-01T15:36:30Z",
        "id": "tech",
        "created_at": "2015-09-26T09:54:45Z",
        "updated_at": "2015-09-26T09:54:45Z",
        "uuid": "743be99c-d627-4733-8a4f-30472573a1fb",
        "short_name": "tech",
        "name": "Tech",
        "description": "",
        "color": "orange",
        "visibility": "members",
        "labels": [
          "bug",
          "feature",
          "improvement"
        ],
        "properties": {},
        "hook_events": [
          "repository"
        ]
      },
      "member": {
        "api_status": 200,
        "api_timestamp": "2016-04-01T15:36:30Z",
        "id": "cblanks",
        "created_at": "2015-09-26T09:52:33Z",
        "updated_at": "2015-09-26T09:57:05Z",
        "uuid": "1c47ff18-d6c5-422c-8eb9-8873897012f0",
        "type": "collaborator",
        "email": "cblanks@helixteamhub.com",
        "short_name": "cblanks",
        "first_name": "Claretha",
        "last_name": "Blanks",
        "display_name": "Claretha Blanks",
        "company_admin": false,
        "instance_admin": false,
        "description": "",
        "phone": "",
        "title": "",
        "avatar": {
          "large": "",
          "medium": "",
          "small": ""
        },
        "synchronized_fields": [],
        "locale": "en",
        "source": ""
      }
    }
  ]
```

```
}
```

Project User

Project user is the link object between Projects and Users. With them, you define who belongs to what project, and what they can do in the project.

- Attributes
- Operations
    - `GET /projects/:project_id/project_users`
    - `GET /projects/:project_id/project_users/:user_id`
    - `POST /projects/:project_id/project_users`
    - `PUT /projects/:project_id/project_users/:user_id`
    - `DELETE /projects/:project_id/project_users/:user_id`

Attributes

project

The id of the project the membership is for.

When requesting, can be expanded to the full project object by including the query parameter `expand=project`.

- Type: `string`
- Required: `true`

user

The id of the user the membership is for.

When requesting, can be expanded to the full user object by including the query parameter `expand=user`.

- Type: `string`
- Required: `true`

role

The role the user has directly in the project. Must be one of `admin`, `manager`, `master`, `developer`, or `guest`.

- Type: `string`
- Required: `true`

highest_role

The highest role the user has in the project either directly or through groups. This field is automatically updated.

- Type: `string`

Operations

GET /projects/:project_id/project_users

List all project users for a project, with optional limit and offset parameters, as a metadata-results object.

GET /projects/:project_id/project_users/:user_id

Return a specific project user.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/project_users/vluoto
```

Example response

```
{
    "api_status": 200,
    "api_timestamp": "2012-10-25T08:42:01Z",
    "project": {
      "id": "luotsi"
    },
    "user": {
      "id": "vluoto"
    },
    "role": "developer",
    "highest_role": "admin"
}
```

Example request with user and project fields expanded

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/project_
users/vluoto?expand=user,project
```

Example response with user and project fields expanded

```
{
    "api_status": 200,
    "api_timestamp": "2012-10-25T09:29:37Z",
    "project": {
        "api_status": 200,
        "api_timestamp": "2012-10-25T09:29:37Z",
        "id": "luotsi",
        "created_at": "2012-10-25T09:29:37Z",
        "updated_at": "2012-10-25T09:29:37Z",
        "name": "Helix TeamHub",
        "description": "Helix TeamHub is Eficode's new product. It's
called Helix TeamHub.",
        "color": "green",
        "avatar": {
            "large": "/projects/luotsi/avatars/luotsi_l.png",
            "medium": "/projects/luotsi/avatars/luotsi_m.png",
            "small": "/projects/luotsi/avatars/luotsi_s.png"
        },
        "visibility": "company"
    },
    "user": {
        "api_status": 200,
        "api_timestamp": "2012-10-25T09:29:37Z",
        "id": "vluoto",
        "created_at": "2012-10-25T09:29:37Z",
        "updated_at": "2012-10-25T09:29:37Z",
        "email": "veli-matti.luoto@eficode.com",
        "first_name": "Veli-Matti",
        "last_name": "Luoto",
        "display_name": "Veli-Matti Luoto",
        "company_admin": true,
        "instance_admin": true,
        "description": "*^-D",
        "phone": "+358 50 599 0681",
        "title": "Software Engineer",
        "avatar": {
            "large": "",
            "medium": "",
            "small": "",
        },
        "synchronized_fields": [],
        "locale": "en",
        "source": "",
    },
    "role": "developer",
    "highest_role": "admin"
}
```

POST /projects/:project_id/project_users

Add a project user for a project.

Example request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "asyrjasalo", "role": "guest" }' \
  https://helixteamhub.cloud/api/projects/luotsi/project_users
```

Example response

```
{
  "api_status": 201,
  "api_timestamp": "2012-10-25T09:29:37Z",
  "project": {
    "id": "luotsi"
  },
  "user": {
    "id": "asyrjasalo"
  },
  "role": "guest",
  "highest_role": "guest"
}
```

PUT /projects/:project_id/project_users/:user_id

Update role for a project user. Returns the updated project user.

Example request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "role": "admin" }' \
  https://helixteamhub.cloud/api/projects/luotsi/project_users/asyrjasalo
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2013-11-28T10:12:20Z",
  "project": {
    "id": "luotsi"
  },
  "user": {
    "id": "asyrjasalo"
  },
  "role": "admin",
  "highest_role": "admin"
}
```

DELETE /projects/:project_id/project_users/:user_id

Remove a user from a project. Returns the removed project user.

Example request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/project_users/asyrjasalo
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2013-11-28T10:12:28Z",
  "project": {
    "id": "luotsi"
  },
  "user": {
    "id": "asyrjasalo"
  },
  "role": "admin",
  "highest_role": "admin"
}
```

Project

A project is where most of the collaboration happens in Helix TeamHub. A company can have 0 to n projects, and each project belongs to exactly one company. Most of the other objects are handled in the scope of projects.

- Attributes
- Searchable attributes
- Child objects
- Operations
  - `GET /projects`
  - `GET /projects/:id`
  - `POST /projects`
  - `PUT /projects/:id`
  - `DELETE /projects/:id`

Attributes

Project object has support for storing custom attributes.

id

A URL-friendly identifier for the project.

- Type: `string`
- Unique: `true`
- Required: `true`
- Minimum length: `2`
- Maximum length: `100`
- Allowed characters: `a-z`, `A-Z`, `0-9`, `-`, `_`

name

The project's human-readable name.

- Type: `string`
- Unique: `true`
- Required: `true`

- Minimum length: `2`
- Maximum length: `100`

visibility

The project's visibility flag: Can be visible to only its members or to all people in the company.

- Type: `string`
- Default: `company`
- Allowed values: `company`, `members`

color

The background color of the default project avatar.

- Type: `string`
- Default: `blue`
- Allowed values: `blue`, `green`, `red`, `orange`, `turquoise`, `purple`

labels

Possible labels for the tasks in the project.

- Type: `array`

description

A free-form description of the project.

- Type: `string`
- Default: `""`
- Maximum length: `5000`

hook_events

All hook events the project supports.

- Type: `array`

The hook_events array contains string identifiers of all the supported hook events.

```
"hook_events": ["repository"]
```

automatic_depot_creation

If set to `true`, a new Graph depot named `//<project_short_name>/` is created in the Helix4Git server. Any Git repositories added to the project will be automatically created in this depot

- Type: `boolean`
- Default: `false`

properties

See custom attributes for more information.

Searchable attributes

Projects can be searched by their `id`, `name`, and `description` attributes. See the search documentation for instructions how to do this.

Child objects

The following child objects can be attached to the projects returned, to eliminate the need for requesting them separately:

Object	Attribute
User	`users`
Collaborator	`collaborators`
Groups	`groups`
Repository	`repositories`
Hook	`hooks`
Milestone	`milestones`
Task	`tasks`
Subscriber	`subscribers`

For example, to get all of the project's users in full and the number of repositories it has, attach `include=users&count=repositories` to your GET request.

See the child object API documentation for a description of the possible operations.

Operations

GET /projects

Returns all the projects in the company, with optional limit and offset parameters, as a metadata-results object.

GET /projects/:id

Returns one project as a project object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-06-14T12:06:57Z",
  "id": "luotsi",
  "created_at": "2011-08-23T21:43:20Z",
  "updated_at": "2013-05-11T09:53:09Z",
  "name": "Helix TeamHub",
  "description": "The next generation software development and
collaboration hub for the world.",
  "color": "red",
  "labels": [
    "bug",
    "feature",
    "improvement"
  ],
  "visibility": "members",
  "properties": {
  }
}
```

POST /projects

Creates a new project to the company. Returns the created project object.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "platform", "name": "Platform" }' \
  https://helixteamhub.cloud/api/projects
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2013-06-14T12:06:57Z",
  "created_at": "2011-08-23T21:43:20Z",
  "updated_at": "2013-05-11T09:53:09Z",
  "id": "platform",
  "name": "Platform",
  "description": "",
  "color": "red",
  "labels": [

  ],
  "visibility": "members",
  "properties": {

  }
}
```

PUT /projects/:id

Updates a project. Returns the updated project object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "visibility": "company" }' \
  https://helixteamhub.cloud/api/projects/platform
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2013-06-14T12:06:57Z",
  "created_at": "2011-08-23T21:43:20Z",
  "updated_at": "2013-05-11T09:53:09Z",
  "id": "platform",
  "name": "Platform",
  "description": "",
  "color": "red",
  "labels": [

  ],
  "visibility": "company",
  "properties": {

  }
}
```

DELETE /projects/:id

Deletes a project. Returns the deleted project object.

Repository

A repository is where the source code is stored. Repositories belong to a project. A project may have multiple repositories.

There are a number of types of repositories in Helix TeamHub: Git, Helix Git, Mercurial, Subversion, WebDAV, Maven, and Ivy.

- Attributes
- Searchable attributes
- Child Objects
- Operations
  - `GET /projects/:project_id/repositories`
  - `GET /projects/:project_id/repositories/:id`
  - `POST /projects/:project_id/repositories`
  - `PUT /projects/:project_id/repositories/:id`
  - `DELETE /projects/:project_id/repositories/:id`

> **Note**
> Forward slash characters in `:id` must be encoded with `%2F`

Attributes

id

The repository identifier/short name.

- Type: `string`
- Unique: `true` (within the project)
- Required: `true`
- Minimum length: `2`
- Maximum length: `100`
- Allowed characters: `a-z`, `A-Z`, `0-9`, `-`, `_` (and `/` only for Git)

type

One of Helix TeamHub's version control systems: `git`, `helix_git`, `subversion`, `mercurial`, `webdav`, `maven`, or `ivy`.

- Type: `string`
- Required: `true`

> **Note**
> When the `helix_git` type is used you must include the `helix_path` in the request, see `helix_path` for details.

helix_path

When the `helix_git` repo `type` is used you must include the `helix_path` in the request: `"type":"helix_git", "helix_path":"//<project_name>/<repo_name>"`

http_url

The HTTP(s) URL for version control access.

https_url

Deprecated, use `http_url`.

- Type: `string`

ssh_url

The SSH URL for version control access. Not supported for WebDAV, Maven, or Ivy repositories.

- Type: `string`

creator

Creator of the repository can be expanded to get the full User object by including the query parameter `expand=creator` in the request.

- Type: `object`

hook_events

All hook events the repository supports.

- Type: `array`

The hook_events array contains string identifiers of all the supported hook events.

```
"hook_events": ["push"]
```

default_identifier

The identifier used as a default for repository operations where the reference name is not provided.

- Type: `string`
- Default: `master` for Git, `default` for Mercurial, `HEAD` for Subversion, `webdav` for WebDAV, `maven` for Maven, and `ivy` for Ivy repositories.

default_base_branch

The default target reference for code reviews created inside the repository.

- Type: `string`
- Default: `master` for Git, `default` for Mercurial, `HEAD` for Subversion, `webdav` for WebDAV, `maven` for Maven, and `ivy` for Ivy repositories.

default_voting_threshold

The default number of required approvals for code reviews created inside the repository.

- Type: `integer`
- Default: `0`

default_require_build

Whether or not build is required for code reviews created inside the repository.

- Type: `boolean`
- Default: `false`

enforce_voting

Enforces voting for code reviews created inside the repository.

- Type: `boolean`
- Default: `false`

enforce_build

Enforces build requirement for code reviews created inside the repository.

- Type: `boolean`
- Default: `false`

properties

See custom attributes for more information.

Searchable attributes

Repositories can be searched by their `id` attribute. See the search documentation for instructions how to do this.

Child objects

The following child objects can be attached to the repositories returned, to eliminate the need for requesting them separately:

Object	Attribute
Branch	`branches`
Tag	`tags`
Hook	`hooks`

For example, to get all of the repository branches and tags in full, attach `include=branches,tags` to your GET request.

See child object API documentation for a description of the possible operations.

Operations

GET /projects/:project_id/repositories

Returns all repositories for a project, with optional limit and offset parameters, as a metadata-results object.

GET /projects/:project_id/repositories/:id

Returns a single repository.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/repositories/luotsi
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2012-10-24T14:03:48Z",
  "id": "luotsi",
  "created_at": "2012-10-24T14:03:48Z",
  "updated_at": "2012-10-24T14:03:48Z",
  "type": "git",
  "ssh_url":
"hth@helixteamhub.cloud:eficode/projects/luotsi/repositories/git/luotsi",
  "https_url":
"https://helixteamhub.cloud/eficode/projects/luotsi/repositories/git/luots
i",
  "http_url":
"https://helixteamhub.cloud/eficode/projects/luotsi/repositories/git/luots
i",
  "creator": {
    "id": "jp"
  },
  "hook_events": [
    "push"
  ],
  "properties": {

  }
}
```

POST /projects/:project_id/repositories

Creates a new repository for a project. Returns the created repository.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "foo", "type": "subversion" }' \
  https://helixteamhub.cloud/api/projects/luotsi/repositories
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2012-10-24T14:03:48Z",
  "id": "foo",
  "created_at": "2012-10-24T14:03:48Z",
  "updated_at": "2012-10-24T14:03:48Z",
  "type": "subversion",
  "ssh_url":
"svn+ssh://hth@helixteamhub.cloud/acme/projects/luotsi/repositories/subversion/foo",
  "https_url":
"https://helixteamhub.cloud/eficode/projects/luotsi/repositories/subversion/foo",
  "http_url":
"https://helixteamhub.cloud/eficode/projects/luotsi/repositories/subversion/foo",
  "creator": {
    "id": "jp"
  },
  "hook_events": [
    "push"
  ],
  "properties": {

  }
}
```

PUT /projects/:project_id/repositories/:id

Updates a repository. Returns the updated repository.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "id": "bar" }' \
  https://helixteamhub.cloud/api/projects/luotsi/repositories/foo
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2012-10-24T14:03:48Z",
  "id": "foo",
  "created_at": "2012-10-24T14:03:48Z",
  "updated_at": "2012-10-24T14:03:48Z",
  "type": "subversion",
  "ssh_url": "svn+ssh://hth@helixteamhub.cloud/acme/projects/luotsi/repositories/subversion/foo",
  "https_url": "https://helixteamhub.cloud/eficode/projects/luotsi/repositories/subversion/foo",
  "http_url": "https://helixteamhub.cloud/eficode/projects/luotsi/repositories/subversion/foo",
  "creator": {
    "id": "jp"
  },
  "hook_events": [
    "push"
  ],
  "properties": {
  }
}
```

DELETE /projects/:project_id/repositories/:id

Deletes a repository. Returns the deleted repository.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/repositories/foo
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2012-10-24T14:03:48Z",
  "id": "foo",
  "created_at": "2012-10-24T14:03:48Z",
  "updated_at": "2012-10-24T14:03:48Z",
  "type": "subversion",
  "ssh_url":
"svn+ssh://hth@helixteamhub.cloud/acme/projects/luotsi/repositories/subver
sion/foo",
  "https_url":
"https://helixteamhub.cloud/eficode/projects/luotsi/repositories/subversio
n/foo",
  "http_url":
"https://helixteamhub.cloud/eficode/projects/luotsi/repositories/subversio
n/foo",
  "creator": {
    "id": "jp"
  },
  "hook_events": [
    "push"
  ],
  "properties": {

  }
}
```

Repository Collaborator

Repository Collaborator is the link object between Repositories and Collaborators. With them, you define who belongs to what repository, and what they can do in the repository.

- "Attributes" below
- "Operations" on the facing page
  - `GET /projects/:project_id/repositories/:repository_id/repository_collaborators`
  - `GET /projects/:project_id/repositories/:repository_id/repository_collaborators/:collaborator_id`
  - `POST /projects/:project_id/repositories/:repository_id/repository_collaborators`
  - `PUT /projects/:project_id/repositories/:repository_id/repository_collaborators/:collaborator_id`
  - `DELETE /projects/:project_id/repositories/:repository_id/repository_collaborators/:collaborator_id`

Attributes

repository

The ID of the repository that the membership is meant for. When requesting, can be expanded to the full repository object by including the query parameter `expand=repository`.

- Type: `string`
- Required: `true`

collaborator

The ID of the collaborator that the membership is meant for. When requesting, can be expanded to the full collaborator object by including the query parameter `expand=collaborator`.

- Type: `string`
- Required: `true`

role

The role that the collaborator has directly in the repository. Must be one of **`admin`**, **`master`**, **`developer`**, or **`guest`**.

- Type: `string`
- Required: `true`

highest_role

The highest role the collaborator has in the repository. This field is automatically updated.

- Type: `string`

Operations

GET /projects/:project_id/repositories/:repository_id/repository_collaborators

Lists all repository collaborators for a repository, with optional limit and offset parameters, as a metadata-results object.

GET /projects/:project_id/repositories/:repository_id/repository_collaborators/:collaborator_id

Returns a specific repository collaborator.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/repositor
y_collaborators/janedoe
```

Example response

```
{
   "api_status": 200,
   "api_timestamp": "2018-10-25T09:29:37Z",
   "repository": {
       "id": "acme"
   },
   "collaborator": {
       "id": "janedoe"
   },
   "role": "developer",
   "highest_role": "admin"
}
```

POST /projects/:project_id/repositories/:repository_id/repository_collaborators

Adds a repository collaborator for a repository. The collaborator must have a project role before a repository role can be defined.

Example request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "repository_collaborator": { "id": "janedoe", "role": "guest" } }'
\

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/repositor
y_collaborators
```

Example response

```
{
   "api_status": 201,
   "api_timestamp": "2018-10-25T09:29:37Z",
   "repository": {
       "id": "acme"
   },
   "collaborator": {
      "id": "janedoe"
   },
   "role": "guest",
   "highest_role": "guest"
}
```

PUT /projects/:project_id/repositories/:repository_id/repository_collaborators/:collaborator_id

Updates role for a repository collaborator. Returns the updated repository collaborator.

Example request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "repository_collaborator": { "role": "developer" } }' \

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/repositor
y_collaborators/janedoe
```

Example response

```
{
   "api_status": 200,
   "api_timestamp": "2018-11-28T10:12:20Z",
   "repository": {
       "id": "acme"
   },
   "collaborator": {
       "id": "janedoe"
   },
   "role": "developer",
   "highest_role": "developer"
}
```

DELETE /projects/:project_id/repositories/:repository_id/repository_collaborators/:collaborator_id

Removes a collaborator from a repository. Returns the removed repository collaborator.

Example request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/repositor
y_collaborators/janedoe
```

Example response

```
{
   "api_status": 200,
   "api_timestamp": "2018-11-28T10:12:28Z",
   "repository": {
       "id": "acme"
   },
   "collaborator": {
       "id": "janedoe"
   },
   "role": "developer",
   "highest_role": "developer"
}
```

Repository Group

Repository Group is the link object between Repositories and Groups. With them, you define which groups belong to what repository, and what the group members can do in the repository.

- "Attributes" below
- "Operations" on the next page
  - `GET /projects/:project_id/repositories/:repository_id/repository_groups`
  - `GET /projects/:project_id/repositories/:repository_id/repository_groups/:group_id`
  - `POST /projects/:project_id/repositories/:repository_id/repository_groups`
  - `PUT /projects/:project_id/repositories/:repository_id/repository_groups/:group_id`
  - `DELETE /projects/:project_id/repositories/:repository_id/repository_groups/:group_id`

Attributes

repository

The Id of the repository that the membership is meant for. When requesting, can be expanded to the full repository object by including the query parameter `expand=repository`.

- Type: `string`
- Required: `true`

group

The ID of the group that the membership is meant for. When requesting, can be expanded to the full group object by including the query parameter `expand=group`.

- Type: `string`
- Required: `true`

role

The role that the group members have directly in the repository. Must be one of `admin`, `master`, `developer`, or `guest`.

- Type: `string`
- Required: `true`

Operations

GET /projects/:project_id/repositories/:repository_id/repository_groups

Lists all repository groups for a repository, with optional limit and offset parameters, as a metadata-results object.

GET /projects/:project_id/repositories/:repository_id/repository_groups/:group_id

Returns a specific repository group.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/repositor
y_groups/devs
```

Example response

```
{
   "api_status": 200,
   "api_timestamp": "2018-10-25T09:29:37Z",
   "repository": {
       "id": "acme"
   },
   "group": {
       "id": "devs"
   },
   "role": "developer"
}
```

POST /projects/:project_id/repositories/:repository_id/repository_groups

Adds a repository group for a repository. The group must have a project role before a repository role can be defined.

Example request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "repository_group": { "id": "devs", "role": "developer" } }' \

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/repositor
y_groups
```

Example response

```
{
   "api_status": 201,
   "api_timestamp": "2018-10-25T09:29:37Z",
   "repository": {
       "id": "acme"
   },
   "group": {
       "id": "devs"
   },
   "role": "guest"
}
```

PUT /projects/:project_id/repositories/:repository_id/repository_groups/:group_id

Updates role for a repository group. Returns the updated repository group.

Example request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "repository_group": { "role": "guest" } }' \

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/repositor
y_groups/devs
```

Example response

```
{
   "api_status": 200,
   "api_timestamp": "2018-11-28T10:12:20Z",
   "repository": {
       "id": "acme"
   },
   "group": {
       "id": "devs"
   },
   "role": "guest"
}
```

DELETE /projects/:project_id/repositories/:repository_id/repository_groups/:group_id

Remove a group from a repository. Returns the removed repository group.

Example request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/repositor
y_groups/devs
```

Example response

```
{
   "api_status": 200,
   "api_timestamp": "2018-11-28T10:12:28Z",
   "repository": {
       "id": "acme"
   },
   "group": {
       "id": "devs"
   },
   "role": "guest"
}
```

Repository User

Repository User is the link object between Repositories and Users. With them, you define who belongs to what repository, and what they can do in the repository.

- "Attributes" on the next page
- "Operations" on the next page
  - **`GET /projects/:project_id/repositories/:repository_id/repository_users`**
  - **`GET /projects/:project_id/repositories/:repository_id/repository_users/:user_id`**
  - **`POST /projects/:project_id/repositories/:repository_id/repository_users`**
  - **`PUT /projects/:project_id/repositories/:repository_id/repository_users/:user_id`**
  - **`DELETE /projects/:project_id/repositories/:repository_id/repository_users/:user_id`**

Attributes

repository

The ID of the repository that the membership is meant for. When requesting, can be expanded to the full repository object by including the query parameter `expand=repository`.

- Type: `string`
- Required: `true`

user

The ID of the user that the membership is meant for. When requesting, can be expanded to the full user object by including the query parameter `expand=user`.

- Type: `string`
- Required: `true`

role

The role that the user has directly in the repository. Must be one of `admin`, `master`, `developer`, or `guest`.

- Type: `string`
- Required: `true`

highest_role

The highest role that the user has in the repository. This field is automatically updated.

- Type: `string`

Operations

GET /projects/:project_id/repositories/:repository_id/repository_users

Lists all repository users for a repository, with optional limit and offset parameters, as a metadata-results object.

GET /projects/:project_id/repositories/:repository_id/repository_users/:user_id

Returns a specific repository user.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/repositor
y_users/johndoe
```

Example response

```
{
   "api_status": 200,
   "api_timestamp": "2018-10-25T09:29:37Z",
   "repository": {
       "id": "acme"
   },
   "user": {
       "id": "johndoe"
   },
   "role": "developer",
   "highest_role": "admin"
}
```

POST /projects/:project_id/repositories/:repository_id/repository_users

Adds a repository user for a repository. The user must have a project role either directly or through a group before a repository role can be defined.

Example request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "repository_user": { "id": "johndoe", "role": "guest" } }' \

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/repositor
y_users
```

Example response

```
{
   "api_status": 201,
   "api_timestamp": "2018-10-25T09:29:37Z",
   "repository": {
       "id": "acme"
   },
   "user": {
       "id": "johndoe"
   },
   "role": "guest",
   "highest_role": "guest"
}
```

PUT /projects/:project_id/repositories/:repository_id/repository_users/:user_id

Updates the role for a repository user. Returns the updated repository user.

Example request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "repository_user": { "role": "admin" } }' \

https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/repositor
y_users/johndoe
```

Example response

```
{
   "api_status": 200,
   "api_timestamp": "2018-11-28T10:12:20Z",
   "repository": {
       "id": "acme"
   },
   "user": {
       "id": "johndoe"
   },
   "role": "admin",
   "highest_role": "admin"
}
```

DELETE /projects/:project_id/repositories/:repository_id/repository_users/:user_id

Removes a user from a repository. Returns the removed repository user.

Example request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
https://helixteamhub.cloud/api/projects/luotsi/repositories/acme/repository_users/johndoe
```

Example response

```
{
   "api_status": 200,
   "api_timestamp": "2018-11-28T10:12:28Z",
   "repository": {
       "id": "acme"
   },
   "user": {
      "id": "johndoe"
   },
   "role": "admin",
   "highest_role": "admin"
}
```

SSH Key

An SSH key is used for authenticating an account (a user or a bot) when accessing repositories over SSH.

- Attributes
- Operations
    - For current user
        - **`GET /account/ssh_keys`**
        - **`GET /account/ssh_keys/:id`**
        - **`POST /account/ssh_keys`**
        - **`PUT /account/ssh_keys/:id`**
        - **`DELETE /account/ssh_keys/:id`**

Attributes

id

A URL-friendly identifier of the key.

- Type: `string`
- Unique: `true`
- Immutable: `true`

title

A human-readable name of the SSH key (for example, "Macbook @ work")

- Type: `string`
- Required: `true`
- Maximum length: `50`

content

The actual contents of the public key. Must be in OpenSSH format. Must be either an RSA key (`ssh rsa...`) or a DSA key (`ssh-dss...`).

- Type: `string`
- Required: `true`
- Minimum length: `380`
- Maximum length: `1500`

Operations

For current user

This endpoint supports both Users and Collaborators.

GET /account/ssh_keys

Returns all the SSH keys of a user, with optional limit and offset parameters, as a metadata-results object.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/account/ssh_keys
```

Example response

```
{
  "metadata": {
    "more_results": false,
    "next_offset": 2,
    "count": 2
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2012-10-24T14:03:48Z",
      "id": "51b9d9d21cf35908d0000134",
      "created_at": "2012-10-24T14:03:48Z",
      "updated_at": "2012-10-24T14:03:48Z",
      "title": "My work laptop",
      "content": "ssh-rsa ... norris@NorrisMacBookPro"
    },
    {
      "api_status": 200,
      "api_timestamp": "2012-10-24T14:03:48Z",
      "id": "512786801229c077f6002480",
      "created_at": "2012-10-24T14:03:48Z",
      "updated_at": "2012-10-24T14:03:48Z",
      "title": "My desktop computer at home",
      "content": "ssh-dss ... chuck@homecomp"
    }
  ]
}
```

GET /account/ssh_keys/:id

Returns one SSH key, based on its id.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/account/ssh_keys/51b9d9d21cf35908d0000134
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2012-10-24T14:03:48Z",
  "id": "51b9d9d21cf35908d0000134",
  "created_at": "2012-10-24T14:03:48Z",
  "updated_at": "2012-10-24T14:03:48Z",
  "title": "My work laptop",
  "content": "ssh-rsa ... norris@NorrisMacBookPro"
}
```

POST /account/ssh_keys

Creates a new SSH key for the user. Returns the created SSH key object.

Example request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "title": "My IPad2", "content": "ssh-rsa ... norris@NorrisIPad"}'
\
  https://helixteamhub.cloud/api/account/ssh_keys
```

Example response

```
{
  "api_status": 201,
  "api_timestamp": "2012-10-24T14:03:48Z",
  "id": "51b9d9d21cf35908d0000134",
  "created_at": "2012-10-24T14:03:48Z",
  "updated_at": "2012-10-24T14:03:48Z",
  "title": "My IPad2 Pro",
  "content": "ssh-rsa ... norris@NorrisIPad"
}
```

PUT /account/ssh_keys/:id

Updates an SSH key. Returns the updated SSH key object.

Example request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "title": "My old Macbook Pro" }'
  https://helixteamhub.cloud/api/account/ssh_keys/51b9d9d21cf35908d0000134
```

Example response

```
{
  "api_status": 201,
  "api_timestamp": "2012-10-24T14:03:48Z",
  "id": "51b9d9d21cf35908d0000134",
  "created_at": "2012-10-24T14:03:48Z",
  "updated_at": "2012-10-24T14:03:48Z",
  "title": "My old Macbook Pro",
  "content": "ssh-rsa ... norris@NorrisMacBookPro"
}
```

DELETE /account/ssh_keys/:id

Deletes an SSH key. Returns the deleted SSH key object.

Example request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_KEY'" \
  https://helixteamhub.cloud/api/account/ssh_keys/51b9d9d21cf35908d0000134
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2012-10-24T14:03:48Z",
  "id": "51b9d9d21cf35908d0000134",
  "created_at": "2012-10-24T14:03:48Z",
  "updated_at": "2012-10-24T14:03:48Z",
  "title": "My work laptop",
  "content": "ssh-rsa ... norris@NorrisMacBookPro"
}
```

Subscriber

A subscriber is a User or a Collaborator who receives notifications from a subscribable. The currently supported subscribables are Project and Task.

- Operations
  - For Projects
    - `GET /projects/:project_id/subscribers`
    - `GET /projects/:project_id/subscribers/:id`
    - `POST /projects/:project_id/actions/subscribe`
    - `DELETE /projects/:project_id/actions/unsubscribe`
  - For Tasks
    - `GET /projects/:project_id/tasks/:task_id/subscribers`
    - `GET /projects/:project_id/tasks/:task_id/subscribers/:id`
    - `POST /projects/:project_id/tasks/:task_id/actions/subscribe`
    - `DELETE /projects/:project_id/tasks/:task_id/actions/unsubscribe`

Operations

GET /projects/:project_id/subscribers

Get all subscribers for a Project, with optional limit and offset parameters, as a metadata-results object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/platform/subscribers
```

Example Response

See User or Collaborator docs for an example response.

GET /projects/:project_id/subscribers/:id

Get a specific Project subscriber.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/platform/subscribers/norris
```

Example Response

See User or Collaborator docs for an example response.

POST /projects/:project_id/actions/subscribe

Subscribe to a Project.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/platform/actions/subscribe
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2014-11-27T11:11:25Z"
}
```

DELETE /projects/:project_id/actions/unsubscribe

Unsubscribe from a Project.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/platform/actions/unsubscribe
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-11-27T11:11:25Z"
}
```

GET /projects/:project_id/tasks/:task_id/subscribers

Get all subscribers for a Task, with optional limit and offset parameters, as a metadata-results object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
```

```
KEY'" \
  https://helixteamhub.cloud/api/projects/platform/tasks/1/subscribers
```

Example Response

See User or Collaborator docs for an example response.

GET /projects/:project_id/tasks/:task_id/subscribers/:id

Get a specific Task subscriber.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/platform/tasks/1/subscribers/norri
s
```

Example Response

See User or Collaborator docs for an example response.

POST /projects/:project_id/tasks/:task_id/actions/subscribe

Subscribe to a Task.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/platform/tasks/1/actions/subscribe
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2014-11-27T11:11:25Z"
}
```

DELETE /projects/:project_id/tasks/:task_id/actions/unsubscribe

Unsubscribe from a Task.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/platform/tasks/1/actions/unsubscri
be
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-11-27T11:11:25Z"
}
```

Tag

A Tag represents a tag in a version control Repository.

- Attributes
- Operations
    - `GET /projects/:project_id/repositories/:repository_id/tags`

Note
Forward slash characters in `:repository_id` must be encoded with `%2F`

Attributes

id

The name of the tag, e.g. `release-1.0`.

- Type: `string`

commit

The latest commit in the tag.

- Type: `string`

Operations

GET /projects/:project_id/repositories/:repository_id/tags

Returns the tags in a repository, with optional limit and offset, as a metadata-results object.

Request Example

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/repositories/chef/tags
```

Response Example

```
{
  "metadata": {
    "more_results": false,
    "next_offset": 3,
    "count": 3
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2013-01-24T15:26:46Z",
      "id": "release-1.0",
      "commit": "e88aed270f0b88769b4e94af50982ac04235e11a"
    },
    {
      "api_status": 200,
      "api_timestamp": "2013-01-24T15:26:46Z",
      "id": "release-2.0",
      "commit": "ba50a8a9c645403d5387972da964be81f5ff02d2"
    },
    {
      "api_status": 200,
      "api_timestamp": "2013-01-24T15:26:46Z",
      "id": "release-2.1",
      "commit": "0bcefe2bb2bc356928e92aba5df99176383845b5"
    }
  ]
}
```

Task

Tasks can be used to keep track of things that need to be done in a Project. They are grouped into Milestones.

- Attributes
- Searchable attributes
- Child objects
- Operations
  - `GET /projects/:project_id/tasks`
  - `GET /projects/:project_id/tasks/:id`
  - `POST /projects/:project_id/tasks`
  - `PUT /projects/:project_id/tasks/:id`
  - `DELETE /projects/:project_id/tasks/:id`
  - `GET /projects/:project_id/tasks/count`

Attributes

Task object has support for storing custom attributes.

id

Automatically generated URL-friendly identifier for the task.

- Type: `integer`
- Immutable: `true`
- Unique: `true` (in project scope)
- Required: `true`

title

Human-readable title for the task.

- Type: `string`
- Required: `true`
- Minimum length: `1`
- Maximum length: `100`

description

A free-form description of the task.

- Type: `string`
- Default: `""`
- Maximum length: `50000`

due_date

Optional due date of the task.

- Type: `timestamp`

state

State must be one of states defined in the milestone it belongs to.

- Type: `string`
- Required: `true`

priority

Priority must be either one of the priorities defined in the milestone, or integer based when milestone does not define priorities.

- Type: `string` / `integer`
- Required: `true`

labels

Tasks can have optional labels, which are defined in the project it belongs to.

- Type: `array`

creator

Creator of the task can be expanded to get the full User or Collaborator object by including the query parameter `expand=creator` in the request.

- Type: `string` / `object`

members

Project users assigned to the task. Can be expanded to get the full User and/or Collaborator objects by including the query parameter `expand=members` in the request. Please note that you can only assign users from the project the issue was created to.

- Type: `array`

milestone

Milestone of the task can be expanded to get the full Milestone object by including the query parameter `expand=milestone` in the request.

- Type: `string` / `object`
- Required: `true`

Searchable attributes

Tasks can be searched by their `id`, `title`, and `description` attributes. See the search documentation for instructions how to do this.

Child objects

The following child objects can be attached to the tasks returned, to eliminate the need for requesting them separately:

Object	Attribute
Comment	`comments`

Object	Attribute
Attachment	`attachments`
Subscriber	`subscribers`

For example to get all the comments and attachments for the task, attach `include=comments,attachments` to your GET request.

See the child object API documentation for a description of the possible operations.

Operations

GET /projects/:project_id/tasks

Returns all the tasks in the project, with optional limit and offset parameters, as a metadata-results object.

GET /projects/:project_id/tasks/:id

Returns specific task object.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/tasks/1
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-05-13T07:48:47Z",
  "id": 1,
  "created_at": "2014-04-23T12:20:17Z",
  "updated_at": "2014-05-09T14:31:01Z",
  "title": "Feature request",
  "description": "This would be an awesome feature!",
  "due_date": "2014-05-20T12:00:00Z",
  "state": "open",
  "priority": 3,
  "labels": [
    "bug"
  ],
  "creator": {
    "id": "jp"
  },
  "members": [
    {
      "id": "tair"
    },
    {
      "id": "vellu"
    }
  ],
  "milestone": {
    "id": "default"
  },
  "properties": {

  }
}
```

POST /projects/:project_id/tasks

Creates a new task to the project. Returns the created task object.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "title": "Found a bug", "description": "Fix all the bugs!",
"milestone": "current", "state": "next", "priority": "normal" }'
  https://helixteamhub.cloud/api/projects/luotsi/tasks
```

Example Response

```
{
  "api_status": 201,
  "api_timestamp": "2014-05-13T07:55:26Z",
  "id": 2,
  "created_at": "2014-05-13T07:55:26Z",
  "updated_at": "2014-05-13T07:55:26Z",
  "title": "Found a bug",
  "description": "Fix all the bugs!",
  "due_date": null,
  "state": "next",
  "priority": "normal",
  "labels": [

  ],
  "creator": {
    "id": "jp"
  },
  "members": [

  ],
  "milestone": {
    "id": "current"
  },
  "properties": {

  }
}
```

PUT /projects/:project_id/tasks/:id

Updates a task. Returns the updated task object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "members": ["tair", "vellu"] }' \
  https://helixteamhub.cloud/api/projects/luotsi/tasks/2
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-05-13T07:58:00Z",
  "id": 2,
  "created_at": "2014-05-13T07:55:26Z",
  "updated_at": "2014-05-13T07:58:00Z",
  "title": "Found a bug",
  "description": "Fix all the bugs!",
  "due_date": null,
  "state": "next",
  "priority": "normal",
  "labels": [

  ],
  "creator": {
    "id": "jp"
  },
  "members": [
    {
      "id": "tair"
    },
    {
      "id": "vellu"
    }
  ],
  "milestone": {
    "id": "current"
  },
  "properties": {

  }
}
```

DELETE /projects/:project_id/tasks/:id

Deletes a task. Returns the deleted task object.

Example Request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/projects/luotsi/tasks/2
```

GET /projects/:project_id/tasks/count

Returns an object containing the total count of tasks in the project. Accepts optional query parameters for filtering the result set.

Example Request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \

https://helixteamhub.cloud/api/projects/luotsi/tasks/count?milestone=curre
nt
```

Example Response

```
{
  "api_status": 200,
  "api_timestamp": "2014-05-15T13:45:54Z",
  "count": 142
}
```

User

A user is a human being with a Helix TeamHub account. A user can log into Helix TeamHub.

- Attributes
- Searchable attributes
- Child objects
- Operations
    - `GET /users`
    - `GET /users/:id`
    - `POST /users`
    - `PUT /users/:id`
    - `DELETE /users/:id`
    - `GET /users/count`

Attributes

id

A URL-friendly identifier (the username). Must start with a word character ([a-zA-Z0-9_]).

- Type: `string`
- Unique: `true`
- Required: `true`
- Minimum length: `1`
- Maximum length: `100`
- Allowed characters: `a-z`, `A-Z`, `0-9`, `-`, `_`, `+`, `.`

email

The user's email address.

- Type: `string`
- Unique: `true`
- Required: `true`

registration_email

You may use the optional `registration_email` address when creating a new user whose `email` address does not yet exist. The registration email will be sent to this address instead of the user's `email` address.

- Type: `string`

first_name

The user's first name.

- Type: `string`
- Maximum length: `32`

last_name

The user's last name.

- Type: `string`
- Maximum length: `32`

display_name

The display_name is automatically generated from user's attributes. Primarily both first_name and last_ name are used. In case either one of them is absent, will the present one be used by itself. As a last resort, the email attribute is used.

- Type: `string`

password

A new password for the user (can be in plaintext or already hashed). You can omit the password attribute from the request if you do not want to set or update it. Users created without password attribute will get a link to set the initial password within the registration email, unless LDAP only authentication method is used in the Helix TeamHub instance. Validations depend on instance specific configurations.

- Type: `string`
- Default minimum length: `8`
- Default maximum length: `100`

pre_hash

The hash algorithm if the value of the password attribute has already been hashed externally. This is mainly used for users that are synchronized from external systems. For example, passwords stored in LDAP directories are already digested.

- Type: `string`
- Default: `null`
- Allowed algorithms: `ldap_sha1`, `ldap_md5`, `null`

password_changed_at

Timestamp of the last password change when password expiration is enabled.

- Type: `timestamp`

password_expires_at

Timestamp when the password will expire when password expiration is enabled.

- Type: `timestamp`

password_expired

Current password expiration status when password expiration is enabled.

- Type: `boolean`

password_never_expires

Whether or not the password will never expire for an account. Allows disabling password expiration for an account when password expiration is enabled. Only company admins can update the value.

- Type: `boolean`
- Default: `false`

synchronized_fields

An array of attributes in User API, which are actively being updated from external source, such as LDAP or AD.

- Type: `string`
- Default: `[]`

company_admin

Whether or not the user has admin privileges for the company.

- Type: `boolean`
- Default: `false`

instance_admin

Whether or not the user has admin privileges for the instance.

- Type: `boolean`
- Default: `false`

description

A free form profile description of the user.

- Type: `string`
- Default: `""`
- Maximum length: `512`

phone

The user's phone number.

- Type: `string`
- Default: `""`
- Maximum length: `32`
- Allowed characters: `0-9`, `+`, `-`

title

The user's title in the company, for example "Software Engineer".

- Type: `string`
- Default: `""`
- Maximum length: `60`

locale

The user's locale.

- Type: `string`
- Default: `en`
- Allowed values: `en`, `zh`

source

The identifier of the external source where the user is synchronized from.

- Type: `string`
- Default: `""`
- Maximum length: `500`

last_login_at

Timestamp of the last time the user logged in to Helix TeamHub. If the user has never logged in, the response is `null`.

- Type: `timestamp`

login_count

The number of times the user has logged in to Helix TeamHub.

- Type: `integer`

helix_auth_method

Helix authentication method used by the user. Valid responses when the user has Helix authentication, `"perforce"` or `"ldap"`. If the user is not using Helix authentication, an empty string is returned.

- Type: `string`
- Default: `"perforce"` if Helix authentication is configured

avatar

The avatar of the user, as a child object.

- Type: `object`
- Default: `{}`

The avatar objects contains URLs to the avatar image in several sizes:

```
"avatar": {
  "large": "system/uploads/4f85e344-6156-418f-baf0-d4b65c9df148-
100x100.jpg",
  "medium": "system/uploads/4f85e344-6156-418f-baf0-d4b65c9df148-
60x60.jpg",
  "small": "system/uploads/4f85e344-6156-418f-baf0-d4b65c9df148-30x30.jpg"
}
```

Searchable attributes

Users can be searched by their `id`, `email`, and `display_name` attributes. See the search documentation for instructions how to do this.

Child objects

The following child objects can be attached to the users returned, to eliminate the need for requesting them separately:

Object	Attribute
Project	`projects`
Repository	`repositories`

For example, to get all of the user's projects in full, attach `include=projects` to your GET request.

See the child object API documentation for a description of the possible operations.

Operations

GET /users

Returns all the users in the company, with optional limit and offset parameters, as a metadata-results object.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/users
```

Example response

```
{
  "metadata": {
    "more_results": false,
    "next_offset": 2,
    "count": 2
  },
  "results": [
    {
      "api_status": 200,
      "api_timestamp": "2019-10-24T14:03:48Z",
      "id": "norris",
      "created_at": "2019-10-24T14:03:48Z",
      "updated_at": "2019-10-24T14:03:48Z",
      "uuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
      "type": "user",
      "email": "chuck.norris@company.com",
      "short_name": "cnorris",
      "first_name": "Chuck",
      "last_name": "Norris",
      "display_name": "Chuck Norris",
      "company_admin": true,
      "instance_admin": true,
      "description": "Walker Texas Ranger",
      "phone": "",
      "title": "Executive Executor",
      "avatar": {
        "api_status": 200,
        "api_timestamp": "2019-10-24T14:03:48Z",
        "id": "YYYYYY-YYYY-YYYY-YYYY-YYYYYYYY",
        "created_at": "2019-10-24T14:03:48Z",
        "updated_at": "2019-10-24T14:03:48Z",
        "large": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-
100x100.jpg",
        "medium": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-
60x60.jpg",
        "small": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-
30x30.jpg"
      },
      "synchronized_fields": [
        "short_name"
      ],
      "locale": "en",
      "source": ""
      "last_login_at": "2019-10-24T14:03:53Z",
      "login_count": 1,
      "password_given": true,
      "password_changed_at": null,
      "password_expires_at": null,
      "password_expired": false,
      "password_never_expires": false,
      "helix_auth_method": ""
    },
    {
      "api_status": 200,
      "api_timestamp": "2019-10-24T14:03:48Z",
      "id": "ckent",
      "created_at": "2019-10-24T14:03:48Z",
      "updated_at": "2019-10-24T14:03:48Z",
      "uuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
      "type": "user",
      "email": "clark.kent@company.com",
      "short_name": "ckent",
      "first_name": "Clark",
      "last_name": "Kent",
      "display_name": "Clark Kent",
      "company_admin": false,
      "instance_admin": false,
      "description": "",
      "phone": "+358 1234 567",
      "title": "Software engineer",
      "avatar":{
        "api_status": 200,
        "api_timestamp": "2019-10-24T14:03:48Z",
        "id": "YYYYYY-YYYY-YYYY-YYYY-YYYYYYYY",
        "created_at": "2019-10-24T14:03:48Z",
        "updated_at": "2019-10-24T14:03:48Z",
        "large": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-
100x100.jpg",
        "medium": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-
60x60.jpg",
        "small": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-
30x30.jpg"
      },
      "synchronized_fields": [
        "short_name"
      ],
      "locale": "en",
      "source": "external"
      "last_login_at": "2019-10-24T14:03:54Z",
      "login_count": 2,
      "password_given": true,
      "password_changed_at": null,
      "password_expires_at": null,
      "password_expired": false,
      "password_never_expires": false,
      "helix_auth_method": ""
    }
  ]
}
```

GET /users/:id

Returns a single user.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/users/norris
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2019-10-24T14:03:48Z",
  "id": "norris",
  "created_at": "2019-10-24T14:03:48Z",
  "updated_at": "2019-10-24T14:03:48Z",
  "uuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "type": "user",
  "email": "chuck.norris@company.com",
  "short_name": "cnorris",
  "first_name": "Chuck",
  "last_name": "Norris",
  "display_name": "Chuck Norris",
  "company_admin": true,
  "instance_admin": true,
  "description": "Walker Texas Ranger",
  "phone": "",
  "title": "Executive Executor",
  "avatar": {
    "api_status": 200,
    "api_timestamp": "2019-10-24T14:03:48Z",
    "id": "YYYYYY-YYYY-YYYY-YYYY-YYYYYYYY",
    "created_at": "2019-10-24T14:03:48Z",
    "updated_at": "2019-10-24T14:03:48Z",
    "large": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-100x100.jpg",
    "medium": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-60x60.jpg",
    "small": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-30x30.jpg"
  },
  "synchronized_fields": [
    "short_name"
  ],
  "locale": "en",
  "source": ""
  "last_login_at": "2019-10-24T14:03:53Z",
  "login_count": 1,
  "password_given": true,
  "password_changed_at": null,
  "password_expires_at": null,
  "password_expired": false,
  "password_never_expires": false,
  "helix_auth_method": ""
}
```

POST /users

Creates a new user to the company and returns the created user object. A matching user is first searched from the directory (by `email` and `id`) when LDAP authentication is enabled. The user must be found from the directory when LDAP only authentication is used, otherwise a 404 response is returned.

Example request

```
curl -X POST \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "first_name": "Clark", "last_name": "Kent", "id":"ckent", "email":
"clark.kent@company.com", "password": "Clar!Ken7" }' \
  https://helixteamhub.cloud/api/users
```

Example response

```
{
  "api_status": 201,
  "api_timestamp": "2019-10-24T14:03:48Z",
  "id": "ckent",
  "created_at": "2019-10-24T14:03:48Z",
  "updated_at": "2019-10-24T14:03:48Z",
  "uuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "type": "user",
  "email": "clark.kent@company.com",
  "short_name": "ckent",
  "first_name": "Clark",
  "last_name": "Kent",
  "display_name": "Clark Kent",
  "company_admin": false,
  "instance_admin": false,
  "description": "",
  "phone": "",
  "title": "",
  "avatar":{
    "api_status": 201,
    "api_timestamp": "2019-10-24T14:03:48Z",
    "id": "",
    "created_at": "",
    "updated_at": "",
    "large": "",
    "medium": "",
    "small": ""
  },
  "synchronized_fields": [],
  "locale": "en",
  "source": "external"
  "last_login_at": null,
  "login_count": 0,
  "password_given": true,
  "password_changed_at": null,
  "password_expires_at": null,
  "password_expired": false,
  "password_never_expires": false,
  "helix_auth_method": ""
}
```

PUT /users/:id

Updates a the information of a single user. Returns the updated user object.

Example request

```
curl -X PUT \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  -H "Content-Type: application/json" \
  -d '{ "description": "Retired" }' \
  https://helixteamhub.cloud/api/users/norris
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2019-10-24T14:03:48Z",
  "id": "norris",
  "created_at": "2019-10-24T14:03:48Z",
  "updated_at": "2019-10-24T14:03:48Z",
  "uuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "type": "user",
  "email": "chuck.norris@company.com",
  "short_name": "cnorris",
  "first_name": "Chuck",
  "last_name": "Norris",
  "display_name": "Chuck Norris",
  "company_admin": true,
  "instance_admin": true,
  "description": "Retired",
  "phone": "",
  "title": "Executive Executor",
  "avatar": {
    "api_status": 200,
    "api_timestamp": "2019-10-24T14:03:48Z",
    "id": "YYYYYY-YYYY-YYYY-YYYY-YYYYYYYY",
    "created_at": "2019-10-24T14:03:48Z",
    "updated_at": "2019-10-24T14:03:48Z",
    "large": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-
100x100.jpg",
    "medium": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-
60x60.jpg",
    "small": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-
30x30.jpg"
  },
  "synchronized_fields": [
    "short_name"
  ],
  "locale": "en",
  "source": ""
  "last_login_at": "2019-10-24T14:03:53Z",
  "login_count": 1,
  "password_given": true,
  "password_changed_at": null,
  "password_expires_at": null,
  "password_expired": false,
  "password_never_expires": false,
  "helix_auth_method": ""
}
```

DELETE /users/:id

Deletes a user. Returns the deleted user object.

Example request

```
curl -X DELETE \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/users/norris
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2019-10-24T14:03:48Z",
  "id": "norris",
  "created_at": "2019-10-24T14:03:48Z",
  "updated_at": "2019-10-24T14:03:48Z",
  "uuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "type": "user",
  "email": "chuck.norris@company.com",
  "short_name": "cnorris",
  "first_name": "Chuck",
  "last_name": "Norris",
  "display_name": "Chuck Norris",
  "company_admin": true,
  "instance_admin": true,
  "description": "Walker Texas Ranger",
  "phone": "",
  "title": "Executive Executor",
  "avatar": {
    "api_status": 200,
    "api_timestamp": "2019-10-24T14:03:48Z",
    "id": "YYYYYY-YYYY-YYYY-YYYY-YYYYYYYY",
    "created_at": "2019-10-24T14:03:48Z",
    "updated_at": "2019-10-24T14:03:48Z",
    "large": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-
100x100.jpg",
    "medium": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-
60x60.jpg",
    "small": "system/uploads/d2641ea3-ba02-43a6-8566-bff3493c6bbc-
30x30.jpg"
  },
  "synchronized_fields": [
    "short_name"
  ],
  "locale": "en",
  "source": ""
  "last_login_at": "2019-10-24T14:03:53Z",
  "login_count": 1,
  "password_given": true,
  "password_changed_at": null,
  "password_expires_at": null,
  "password_expired": false,
  "password_never_expires": false,
  "helix_auth_method": ""
}
```

GET /users/count

Returns an object containing the amounts of total, active, and inactive users in the company.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.hth.v1" \
  -H "Authorization: hth.company_key='$COMPANY_KEY',account_key='$ACCOUNT_
KEY'" \
  https://helixteamhub.cloud/api/users/count
```

Example response

```
{
  "api_status": 200,
  "api_timestamp": "2020-01-24T15:38:00Z",
  "count": 175,
  "active": 119,
  "inactive": 56
}
```

Helix TeamHub API v2

Please note that this is an experimental version. Documentation for a stable one can be found here.

On these pages, you'll find everything you need to know about the Helix TeamHub REST APIs. These give you access to all data TeamHub stores and let you build any clients and integrations you need.

This page introduces the general concepts used throughout all TeamHub APIs, from authentication to the calling conventions and the error codes. To get information about specific types of data, use the navigation pane on the left.

- API version
- API stability
- Headers
- Authentication

API version

Starting with API v2, there is no need to include any additional headers describing the API version. Versioning is based on part of the URL, for example `/api/v2/bots`.

API stability

TeamHub APIs follow semantic versioning but only the major version number is used. Bug fixes and backward-compatible changes can be added to the same major version. Major TeamHub releases may deprecate an older version of APIs and introduce a new version. For example:

- Helix TeamHub 2.x supports v1.
- Helix TeamHub 3.x may introduce v2 and deprecate v1, or still continue supporting v1.

Headers

All requests require `Accept` and `Content-Type` headers to be set to `application/vnd.api+json`.

Authentication

All requests (except login/logout) require only `account_key` parameter to be present. It represents the user on whose behalf actions are taken.

Bot

A bot account can be taken into use in multiple projects throughout a company. A bot cannot log into TeamHub, but it can access the TeamHub APIs and source code repositories.

Anyone with access to TeamHub may create bots. The creator automatically becomes the owner of the bot and can assign other users either as members or owners to the bot.

Bots can be created as private or as public. Private bots are only visible to their members and owners, while public bots are visible to everyone.

Bot credentials are only visible to its members and owners and, only bot owners can manage the bot credentials.

Company admin bots can access all projects by default, as well as manage users and groups.

Attributes

short-name

A URL-friendly identifier. Also used as the bot's version control username.

- Type: `string`
- Required: `true`
- Unique: `true` (within company)
- Minimum length: `2`
- Maximum length: `40`
- Allowed characters: `a-z`, `A-Z`, `0-9`, `-`, `_`

password

The bot's password. Used for version control access. When creating a bot, TeamHub automatically generates a password if you do not provide one. When updating a bot, leave empty if you do not want to change it. Validations depend on instance-specific configurations.

- Type: `string`
- Default minimum length: `8`
- Default maximum length: `100`
- Required: `false`

visibility

- Type: `string`
- Required: `false`
- Default: `private`
- Allowed values: `private`, `public`

> **Note**
> Company admin bots are always public.

company-admin

- Type: `boolean`
- Required: `false`
- Default: `false`

Related Resources

The following related resources can be attached to the bots returned, to eliminate the need for requesting them separately:

Object	Attribute
SSH Keys	`ssh-keys`
Users>	`bot-users`

For example, to get all of the bot's SSH keys in full, attach `include=ssh-keys,bot-users` to your GET request.

Operations

GET /bots

Returns all bots in the company.

GET /bots/:id

Returns the bot as a bot object.

POST /bots

Creates a new bot. Returns the bot object.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -d '{ "data": { "attributes": { "short-name": "hth-bot" } } }' \
  http://dev.helixteamhub.com/api/v2/bots?account_key=$ACCOUNT_KEY
```

Example Response

```
{
  "data": {
    "id": "61daa045-2ce7-42bb-8d30-c89b736090c1",
    "type": "bots",
    "attributes": {
      "short-name": "hth-bot",
      "password": "iE%1UzWOS-k_jjIa_w78",
      "company-admin": false,
      "account-key": "4aed6ec3e0b1010c427659099fb0b6bc",
      "visibility": "private",
      "created-at": "2016-02-04T16:04:41.283Z",
      "updated-at": "2016-02-04T16:04:41.283Z"
    },
    "relationships": {
      "creator": {
        "data": {
          "id": "fdd36536-8a08-4e5b-ab25-6f7c975c9212",
          "type": "users"
        }
      },
      "company": {
        "data": {
          "id": "ba64aeb7-d9ba-4a8e-a3aa-f9352d224f6a",
          "type": "companies"
        }
      },
      "ssh-keys": {
        "data": []
      },
      "bot-users": {
        "data": [{
          "id": "61daa045-2ce7-42bb-8d30-c89b736090c1_fdd36536-8a08-4e5b-
ab25-6f7c975c9212",
          "type": "bot-users"
        }]
      }
    }
  }
}
```

PUT/PATCH /bots/:id

Updates the existing bot. Returns the updated bot object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -d '{ "data": { "attributes": { "visibility": "public", "short-name":
"public-hth-bot" } } }' \
  http://dev.helixteamhub.com/api/v2/bots/61daa045-2ce7-42bb-8d30-
c89b736090c1?account_key=$ACCOUNT_KEY
```

Example Response

```
{
  "data": {
    "id": "61daa045-2ce7-42bb-8d30-c89b736090c1",
    "type": "bots",
    "attributes": {
      "short-name": "public-hth-bot",
      "password": "iE%1UzWOS-k_jjIa_w78",
      "company-admin": false,
      "account-key": "4aed6ec3e0b1010c427659099fb0b6bc",
      "visibility": "public",
      "created-at": "2016-02-04T16:04:41.283Z",
      "updated-at": "2016-02-04T16:10:36.532Z"
    },
    "relationships": {
      "creator": {
        "data": {
          "id": "fdd36536-8a08-4e5b-ab25-6f7c975c9212",
          "type": "users"
        }
      },
      "company": {
        "data": {
          "id": "ba64aeb7-d9ba-4a8e-a3aa-f9352d224f6a",
          "type": "companies"
        }
      },
      "ssh-keys": {
        "data": []
      },
      "bot-users": {
        "data": [{
          "id": "61daa045-2ce7-42bb-8d30-c89b736090c1_fdd36536-8a08-4e5b-
ab25-6f7c975c9212",
          "type": "bot-users"
        }]
      }
    }
  }
}
```

DELETE /bots/:id

Deletes the existing bot. Returns a 204 status code and blank response.

GET /bots/:bot-id/ssh-keys

Returns all SSH keys added to the bot.

GET /bots/:bot-id/ssh-keys/:id

Returns the SSH key added to the bot as an SSH key object.

POST /bots/:bot-id/ssh-keys

Creates and adds an SSH key to the bot. Returns the SSH key object.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -d '{ "data": { "attributes": { "title": "hth-bot-ssh-key", "content":
"ssh-rsa ... user@machine" } } }' \
  http://dev.helixteamhub.com/api/v2/bots/61daa045-2ce7-42bb-8d30-
c89b736090c1/ssh-keys?account_key=$ACCOUNT_KEY
```

Example Response

```
{
  "data": {
    "id": "73ca16c5-627c-4373-9e10-8e435cbc62eb",
    "type": "ssh-keys",
    "attributes": {
      "title": "hth-bot-ssh-key",
      "body": "ssh-rsa ... user@machine",
      "created-at": "2016-02-04T16:29:33.303Z",
      "updated-at": "2016-02-04T16:29:33.303Z"
    },
    "relationships": {
      "account": {
        "data": {
          "id": "61daa045-2ce7-42bb-8d30-c89b736090c1",
          "type": "bots"
        }
      }
    }
  }
}
```

PUT/PATCH /bots/:bot-id/ssh-keys/:id

Updates the existing SSH key. Returns the updated SSH key object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -d '{ "data": { "attributes": { "title": "hth-bot-ssh-key-updated",
"content": "ssh-rsa ... user@machine" } } }' \
  http://dev.helixteamhub.com/api/v2/bots/61daa045-2ce7-42bb-8d30-
c89b736090c1/ssh-keys/73ca16c5-627c-4373-9e10-8e435cbc62eb?account_
key=$ACCOUNT_KEY
```

Example Response

```
{
  "data": {
    "id": "73ca16c5-627c-4373-9e10-8e435cbc62eb",
    "type": "ssh-keys",
    "attributes": {
      "title": "hth-bot-ssh-key-updated",
      "body": "ssh-rsa ... user@machine",
      "created-at": "2016-02-04T16:29:33.303Z",
      "updated-at": "2016-02-04T16:39:00.143Z"
    },
    "relationships": {
      "account": {
        "data": {
          "id": "61daa045-2ce7-42bb-8d30-c89b736090c1",
          "type": "bots"
        }
      }
    }
  }
}
```

DELETE /bots/:bot-id/ssh-keys/:id

Removes the SSH key. Returns 204 status code and blank response.

GET /bots/:bot-id/bot-users

Returns all users added to the bot as bot-users objects.

GET /bots/:bot-id/bot-users/:id

Returns the user added to the bot as a bot-user object.

POST /bots/:bot-id/bot-users

Adds a user to the bot. Does not create a new user. Returns the bot-user object.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -d '{ "data": { "attributes": { "role": "member" }, "relationships": {
"user": { "data": { "id": "ef5e59d0-990a-4a43-83af-e2db0bbf080a" } } } }
}' \
  http://dev.helixteamhub.com/api/v2/bots/61daa045-2ce7-42bb-8d30-
c89b736090c1/bot-users?account_key=$ACCOUNT_KEY
```

Example Response

```
{
  "data": {
    "id": "61daa045-2ce7-42bb-8d30-c89b736090c1_ef5e59d0-990a-4a43-83af-
e2db0bbf080a",
    "type": "bot-users",
    "attributes": {
      "role": "member"
    },
    "relationships": {
      "bot": {
        "data": {
          "id": "61daa045-2ce7-42bb-8d30-c89b736090c1",
          "type": "bots"
        }
      },
      "user": {
        "data": {
          "id": "ef5e59d0-990a-4a43-83af-e2db0bbf080a",
          "type": "users"
        }
      }
    }
  }
}
```

PUT/PATCH /bots/:bot-id/bot-users/:id

Updates the existing bot-user. Returns the updated bot-user object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -d '{ "data": { "attributes": { "role": "owner" } } }' \
  http://dev.helixteamhub.com/api/v2/bots/61daa045-2ce7-42bb-8d30-
c89b736090c1/bot-users/61daa045-2ce7-42bb-8d30-c89b736090c1_ef5e59d0-990a-
4a43-83af-e2db0bbf080a?account_key=$ACCOUNT_KEY
```

Example Response

```
{
  "data": {
    "id": "61daa045-2ce7-42bb-8d30-c89b736090c1_ef5e59d0-990a-4a43-83af-
e2db0bbf080a",
    "type": "bot-users",
    "attributes": {
      "role": "owner"
    },
    "relationships": {
      "bot": {
        "data": {
          "id": "61daa045-2ce7-42bb-8d30-c89b736090c1",
          "type": "bots"
        }
      },
      "user": {
        "data": {
          "id": "ef5e59d0-990a-4a43-83af-e2db0bbf080a",
          "type": "users"
        }
      }
    }
  }
}
```

DELETE /bots/:bot-id/bot-users/:id

Removes the user from the bot. Returns a 204 status code and blank response.

Default Reviewer Group

With default reviewer groups, you define which project groups are automatically added as reviewers to code reviews of a repository.

- "Operations" below
  - **GET /projects/:project_id/repositories/:repository_id/default-reviewer-groups**
  - **GET /projects/:project_id/repositories/:repository_id/default-reviewer-groups/:default_reviewer_group_id**
  - **POST /projects/:project_id/repositories/:repository_id/default-reviewer-groups**
  - **DELETE /projects/:project_id/repositories/:repository_id/default-reviewer-groups/:default_reviewer_group_id**d

Operations

GET /projects/:project_id/repositories/:repository_id/default-reviewer-groups

Lists all default reviewer groups for a repository.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -H "Authorization: Token $ACCOUNT_KEY" \

https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/default-reviewer-groups
```

Example response

```
{
  "data": [
    {
      "id": "f90da853-c5f2-4e55-9fa5-773e83b78784_2dc25c4d-abd8-4235-824a-
6dce895c82cb",
      "type": "default-reviewer-groups",
      "relationships": {
        "repository": {
          "data": {
            "id": "f90da853-c5f2-4e55-9fa5-773e83b78784",
            "type": "repositories"
          }
        },
        "group": {
          "data": {
            "id": "2dc25c4d-abd8-4235-824a-6dce895c82cb","type": "groups"
          }
        }
      }
    }
  ],
  "links": {
    "self":
"https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/defau
lt-reviewer-groups?page%5Bnumber%5D=1&page%5Bsize%5D=25",
    "first":
"https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/defau
lt-reviewer-groups?page%5Bnumber%5D=1&page%5Bsize%5D=25",
    "prev": null,
    "next": null,
    "last":
"https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/defau
lt-reviewer-groups?page%5Bnumber%5D=1&page%5Bsize%5D=25"
  },
  "meta": {
    "total": 1
  }
}
```

GET /projects/:project_id/repositories/:repository_id/default-reviewer-groups/:default_reviewer_group_id

Returns a specific default reviewer group. You can construct **`default_reviewer_group_id`** by joining **`repository_id`** and **`group_id`** using an underscore (`_`).

Example request

```
curl -X GET \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -H "Authorization: Token $ACCOUNT_KEY" \

https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/default-reviewer-groups/f90da853-c5f2-4e55-9fa5-773e83b78784_2dc25c4d-abd8-4235-824a-6dce895c82cb
```

Example response

```
{
  "data": {
    "id": "f90da853-c5f2-4e55-9fa5-773e83b78784_2dc25c4d-abd8-4235-824a-
6dce895c82cb",
    "type": "default-reviewer-groups",
    "relationships": {
      "repository": {
        "data": {
          "id": "f90da853-c5f2-4e55-9fa5-773e83b78784",
          "type": "repositories"
        }
      },
      "group": {
        "data": {
          "id": "2dc25c4d-abd8-4235-824a-6dce895c82cb",
          "type": "groups"
        }
      }
    }
  }
}
```

POST /projects/:project_id/repositories/:repository_id/default-reviewer-groups

Adds a default reviewer group for a repository. The group must have a project or repository role before you can add it.

Example request

```
curl -X POST \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -H "Authorization: Token $ACCOUNT_KEY" \
  -d '{ "data": { "relationships": { "group": { "data": { "type":
"groups", "id": "2dc25c4d-abd8-4235-824a-6dce895c82cb" } } } }' \
https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/default-reviewer-groups
```

Example response

```
{
  "data": {
    "id": "f90da853-c5f2-4e55-9fa5-773e83b78784_2dc25c4d-abd8-4235-824a-6dce895c82cb",
    "type": "default-reviewer-groups",
    "relationships": {
      "repository": {
        "data": {
          "id": "f90da853-c5f2-4e55-9fa5-773e83b78784",
          "type": "repositories"
        }
      },
      "group": {
        "data": {
          "id": "2dc25c4d-abd8-4235-824a-6dce895c82cb",
          "type": "groups"
        }
      }
    }
  }
}
```

DELETE /projects/:project_id/repositories/:repository_id/default-reviewer-groups/:default_reviewer_group_id

Removes the default reviewer group from the repository. Returns a 204 status code and a blank response. You can construct `default_reviewer_group_id` by joining `repository_id` and `group_id` using an underscore (_).

Example request

```
curl -X DELETE \
  -H "Accept: application/vnd.api+json" \
  -H "Authorization: Token $ACCOUNT_KEY" \

https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/defaul
t-reviewer-groups/f90da853-c5f2-4e55-9fa5-773e83b78784_2dc25c4d-abd8-4235-
824a-6dce895c82cb
```

Project

A project is where most of the collaboration happens in Helix TeamHub. A company can have `0` to `n` projects, and each project belongs to exactly one company. Most of the other objects are handled in the scope of projects.

- Operations
  - `GET /projects/:project-id/project-bots`
  - `GET /projects/:project-id/project-bots/:id`
  - `POST /projects/:project-id/project-bots`
  - `PUT/PATCH /projects/:project-id/project-bots/:id`
  - `DELETE /projects/:project-id/project-bots/:id`

Operations

GET /projects/:project-id/project-bots

Returns all bots added to the project as project-bot objects.

GET /projects/:project-id/project-bots/:id

Returns the bot added to the project as a project-bot object.

POST /projects/:project-id/project-bots

Adds a bot to the project. Does not create a new bot. Returns the project-bot object.

Supported roles are `guest`, `developer`, and `master`.

Example Request

```
curl -X POST \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -d '{ "data": { "attributes": { "role": "developer" }, "relationships":
{ "bot": { "data": { "id": "public-hth-bot" } } } } }' \
  http://dev.helixteamhub.com/api/v2/projects/bungie-studios/project-
bots?account_key=$ACCOUNT_KEY
```

Example Response

```
{
  "data": {
    "id": "f5ea54ee-46d0-43f4-8f93-9f1948d261db_61daa045-2ce7-42bb-8d30-
c89b736090c1",
    "type": "project-bots",
    "attributes": {
      "role": "developer"
    },
    "relationships": {
      "project": {
        "data": {
          "id": "f5ea54ee-46d0-43f4-8f93-9f1948d261db",
          "type": "projects"
        }
      },
      "bot": {
        "data": {
          "id": "61daa045-2ce7-42bb-8d30-c89b736090c1",
          "type": "bots"
        }
      }
    }
  }
}
```

PUT/PATCH /projects/:project-id/project-bots/:id

Updates the existing project-bot. Returns the updated project-bot object.

Example Request

```
curl -X PUT \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -d '{ "data": { "attributes": { "role": "master" } } }' \
  http://dev.helixteamhub.com/api/v2/projects/bungie-studios/project-
bots/f5ea54ee-46d0-43f4-8f93-9f1948d261db_61daa045-2ce7-42bb-8d30-
c89b736090c1?account_key=$ACCOUNT_KEY
```

Example Response

```
{
  "data": {
    "id": "f5ea54ee-46d0-43f4-8f93-9f1948d261db_61daa045-2ce7-42bb-8d30-
c89b736090c1",
    "type": "project-bots",
    "attributes": {
      "role": "master"
    },
    "relationships": {
      "project": {
        "data": {
          "id": "f5ea54ee-46d0-43f4-8f93-9f1948d261db",
          "type": "projects"
        }
      },
      "bot": {
        "data": {
          "id": "61daa045-2ce7-42bb-8d30-c89b736090c1",
          "type": "bots"
        }
      }
    }
    }
}
```

DELETE /projects/:project-id/project-bots/:id

Removes the bot from the project. Returns a 204 status code and blank response.

Repository Bot

Repository Bot is the link object between Repositories and Bots. With them, you define which bots belong to what repository, and what they can do in the repository.

- "Attributes" below
- "Operations" below
  - `GET /projects/:project_id/repositories/:repository_id/repository-bots`
  - `GET /projects/:project_id/repositories/:repository_id/repository-bots/:repository_bot_id`
  - `POST /projects/:project_id/repositories/:repository_id/repository-bots`
  - `PUT/PATCH /projects/:project_id/repositories/:repository_id/repository-bots/:repository_bot_id`
  - `DELETE /projects/:project_id/repositories/:repository_id/repository-bots/:repository_bot_id`

Attributes

role

The role the bots have directly in the repository. Must be one of `master`, `developer`, or `guest`.

- Type: `string`
- Required: `true`

Operations

GET /projects/:project_id/repositories/:repository_id/repository-bots

List all repository bots for a repository.

Example request

```
curl -X GET \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -H "Authorization: Token $ACCOUNT_KEY" \

https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/reposi
tory-bots
```

Example response

```
{
  "data": [
    {
      "id": "f90da853-c5f2-4e55-9fa5-773e83b78784_85b7af71-3eb5-4526-8aa9
-f9219391c506",
      "type": "repository-bots",
      "attributes": {
        "role": "guest"
      },
      "relationships": {
        "repository": {
          "data": {
            "id": "f90da853-c5f2-4e55-9fa5-773e83b78784",
            "type": "repositories"
          }
        },
        "bot": {
          "data": {
            "id": "85b7af71-3eb5-4526-8aa9-f9219391c506",
            "type": "bots"
          }
        }
      }
    },
    {
      "id": "f90da853-c5f2-4e55-9fa5-773e83b78784_4fe239d7-6913-4ed3-b2a4-
c92cfa8f68af",
      "type": "repository-bots",
      "attributes": {
        "role": "master"
      },
      "relationships": {
        "repository": {
          "data": {
            "id": "f90da853-c5f2-4e55-9fa5-773e83b78784",
            "type": "repositories"
          }
        },
        "bot": {
          "data": {
            "id": "4fe239d7-6913-4ed3-b2a4-c92cfa8f68af",
            "type": "bots"
          }
        }
      }
    }
  ],
  "links": {
    "self":
"https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/repos
itory-bots?page%5Bnumber%5D=1&page%5Bsize%5D=25",
    "first":
"https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/repos
itory-bots?page%5Bnumber%5D=1&page%5Bsize%5D=25",
    "prev": null,
    "next": null,
    "last":
"https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/repos
itory-bots?page%5Bnumber%5D=1&page%5Bsize%5D=25"
  },
  "meta": {
    "total": 2
  }
}
```

GET /projects/:project_id/repositories/:repository_id/repository-bots/:repository_bot_id

Return a specific repository bot. The `repository_bot_id` can be constructed by joining the `repository_id` `and bot_id` (or `short-name` attributes) using an underscore (`_`).

Example request

```
curl -X GET \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -H "Authorization: Token $ACCOUNT_KEY" \

https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/reposi
tory-bots/acme_my-bot
```

Example response

```
{
  "data": {
    "id": "f90da853-c5f2-4e55-9fa5-773e83b78784_85b7af71-3eb5-4526-8aa9-
f9219391c506",
    "type": "repository-bots",
    "attributes": {
    "role": "master"
  },
    "relationships": {
      "repository": {
        "data": {
          "id": "f90da853-c5f2-4e55-9fa5-773e83b78784",
          "type": "repositories"
        }
      },
      "bot": {
        "data": {
          "id": "85b7af71-3eb5-4526-8aa9-f9219391c506",
          "type": "bots"
        }
      }
    }
  }
}
```

POST /projects/:project_id/repositories/:repository_id/repository-bots

Add a repository bot for a repository. The bot must have a project role before a repository role can be defined.

Example request

```
curl -X POST \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -H "Authorization: Token $ACCOUNT_KEY" \
  -d '{ "data": { "attributes": { "role": "developer" }, "relationships":
{ "bot": { "data": { "type": "bots", "id": "4fe239d7-6913-4ed3-b2a4-
c92cfa8f68af" } } }, "type": "repository-bots" } }' \

https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/reposi
tory-bots
```

Example response

```
{
  "data": {
    "attributes": {
      "role": "developer"
    },
    "id": "f90da853-c5f2-4e55-9fa5-773e83b78784_85b7af71-3eb5-4526-8aa9-f9219391c506",
    "relationships": {
      "bot": {
        "data": {
          "id": "85b7af71-3eb5-4526-8aa9-f9219391c506",
          "type": "bots"
        }
      },
      "repository": {
        "data": {
          "id": "f90da853-c5f2-4e55-9fa5-773e83b78784",
          "type": "repositories"
        }
      }
    },
    "type": "repository-bots"
  }
}
```

PUT/PATCH /projects/:project_id/repositories/:repository_id/repository-bots/:repository_bot_id

Update role for a repository bot. Returns the updated repository bot.

Example request

```
curl -X PATCH \
  -H "Accept: application/vnd.api+json" \
  -H "Content-Type: application/vnd.api+json" \
  -H "Authorization: Token $ACCOUNT_KEY" \
  -d '{ "data": { "attributes": { "role": "master" } } }' \

https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/repository-bots/acme_my-bot
```

Example response

```
{
  "data": {
    "attributes": {
      "role": "master"
    },
    "id": "f90da853-c5f2-4e55-9fa5-773e83b78784_85b7af71-3eb5-4526-8aa9-f9219391c506",
    "relationships": {
      "bot": {
        "data": {
          "id": "85b7af71-3eb5-4526-8aa9-f9219391c506",
          "type": "bots"
        }
      },
      "repository": {
        "data": {
          "id": "f90da853-c5f2-4e55-9fa5-773e83b78784",
          "type": "repositories"
        }
      }
    },
    "type": "repository-bots"
  }
}
```

DELETE /projects/:project_id/repositories/:repository_id/repository-bots/:repository_bot_id

Remove the bot from the repository. Returns 204 status code and a blank response.

Example request

```
curl -X DELETE \
  -H "Accept: application/vnd.api+json" \
  -H "Authorization: Token $ACCOUNT_KEY" \

https://helixteamhub.cloud/api/v2/projects/luotsi/repositories/acme/repository-bots/acme-my_bot
```