
FreelanceHunt API Framework

Dmytro Hoi

Apr 11, 2022

CONTENTS:

1	FreelanceHunt API framework	1
1.1	Overview	1
1.2	Installation	2
1.3	Documentation	2
1.4	The Current Version	2
1.5	Licence	2
2	Indices and tables	43
	Python Module Index	45
	Index	47

FREELANCEHUNT API FRAMEWORK

1.1 Overview

> **!! DANGER !!: NOT FOR PRODUCTION USE!**

A framework for working with the [FreelanceHunt API](#).

Requirements:

- Python 3.8
- Requests

In progress:

- [] Threads API: Delete ThreadMessage, Create Support request, Send attachment
- [] Tests
- [x] Thread and ThreadMessage representation

1.2 Installation

To install use:

```
pip install freelancehunt-api
```

1.3 Documentation

Quick usage:

```
from freelancehunt import FreelanceHuntClient

fl = FreelanceHuntClient('YOUR_API_TOKEN')
my_profile = fl.profiles.my_profile
print(my_profile.full_name)
#...
```

The freelancehunt-api documentation [available here](#).

1.4 The Current Version

Now active version is: *0.1.3*

1.5 Licence

The freelancehunt-api is [MIT licenced](#)

1.5.1 freelancehunt package

freelancehunt.client module

Main file of FreelanceHunt API framework.

class freelancehunt.client.**FreelanceHuntClient** (*token*, ***kwargs*)

Bases: freelancehunt.core.FreelancehuntObject

Basic API client for FreelanceHunt.

Parameters *token* (*str*) – Token for access to Freelancehunt API (<https://freelancehunt.com/my/api>)

property *contests*

The Contests part of Freelancehunt API.

Return type *Contests*

property *countries*

The Countries part of Freelancehunt API.

Return type *Countries*

property feed

The Feed part of Freelancehunt API.

Return type *Feed*

property is_token_valid

Check that is token valid.

Return type *bool*

property left_time_limit_update

Second to update remaining API limits.

Return type *int*

property profiles

The Profiles part of Freelancehunt API.

Return type *Profiles*

property projects

The Projects part of Freelancehunt API.

Return type *Projects*

property remaining_limit

Current remaining requests limitation.

Return type *int*

property skills

The Skills part of Freelancehunt API.

Return type *Skills*

property threads

The Threads part of Freelancehunt API.

Return type *Threads*

Module contents

Package initialization and import control.

class `freelancehunt.Cities` (*country_id*, *token=None*, ***kwargs*)

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Cities API part.

Note: This module contains static content. It may be *update()*, but loaded info does not change on the API side.

Warning: For directly usage please set *token* argument.

Parameters

- **token** (*str*) – your API token, optional
- **country_id** (*int*) – API-related country identifier to get cities of it

find (*text*)

Find the names of the cities that contain the desired text.

Parameters **text** (*str*) – the desired text that need to be in an city name.

Return type *List[City]*

Returns list of cities with an text in name.

get (*city_id*)

Get the desired city by city_id.

Parameters **city_id** (*int*) – id of the desired city (<https://apidocs.freelancehunt.com/?version=latest#65a2a9b4-b52d-4706-8c54-990600b58c2b>).

Return type *City*

Returns the desired city.

property list

Get list of all cities.

Return type *List[City]*

Returns list of cities.

update ()

Update static information from API.

Return type *None*

class `freelancehunt.Contests` (*token=None, **kwargs*)
Bases: `freelancehunt.core.FreelancehuntObject`
Provide operations with Contests API part.

Note: NOT IMPLEMENTED YET!

Warning: For directly usage please set *token* argument.

Parameters **token** (*str*) – your API token, optional

class `freelancehunt.Countries` (*token=None, **kwargs*)
Bases: `freelancehunt.core.FreelancehuntObject`
Provide operations with Countries API part.

Note: This module contains static content. It may be *update()*, but loaded info does not change on the API side.

Warning: For directly usage please set *token* argument.

Parameters **token** (*str*) – your API token, optional

property cities

Cities linked to this Country.

Return type *Cities*

Returns object for manipulate with Cities API part

find(text)

Find countries with the desired text.

Parameters **text** (*str*) – text in country name

Raises **ValueError** – No countries found

Return type *List[Country]*

Returns list of countries with an text in name

get(country_id=None, iso_code=None)

Get the filtered country.

Parameters

- **country_id** (*Optional[int]*) – the desired API-related country id, defaults to None
- **iso_code** (*Optional[str]*) – the desired ISO code of country, defaults to None

Raises

- **AttributeError** – No one of parameters filled
- **ValueError** – Contry not found

Return type *Country*

Returns the desired country

property list

Get list of all countries.

Return type *List[Country]*

Returns list of countries

update()

Update static information from API.

class `freelancehunt.Feed(token=None, **kwargs)`

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Feed API part.

Warning: For directly usage please set *token* argument.

Parameters **token** (*str*) – your API token, optional

property contests

Get all messages with linked contest in feed.

Return type *List[FeedMessage]*

get_new()

Get all new notifications in feed.

Return type *List[FeedMessage]*

property list

Get all feed messages.

Return type `List[FeedMessage]`

property projects

Get all messages with linked project in feed.

Return type `List[FeedMessage]`

read()

Mark feed as read.

update()

Get latest feed information.

class `freelancehunt.FreelanceHuntClient` (*token*, ***kwargs*)

Bases: `freelancehunt.core.FreelanceHuntObject`

Basic API client for FreelanceHunt.

Parameters *token* (*str*) – Token for access to Freelancehunt API (<https://freelancehunt.com/my/api>)

property contests

The Contests part of Freelancehunt API.

Return type `Contests`

property countries

The Countries part of Freelancehunt API.

Return type `Countries`

property feed

The Feed part of Freelancehunt API.

Return type `Feed`

property is_token_valid

Check that is token valid.

Return type `bool`

property left_time_limit_update

Second to update remaining API limits.

Return type `int`

property profiles

The Profiles part of Freelancehunt API.

Return type `Profiles`

property projects

The Projects part of Freelancehunt API.

Return type `Projects`

property remaining_limit

Current remaining requests limitation.

Return type `int`

property skills

The Skills part of Freelancehunt API.

Return type *Skills*

property threads

The Threads part of Freelancehunt API.

Return type *Threads*

class `freelancehunt.Profiles` (*token=None, **kwargs*)
 Bases: `freelancehunt.core.FreelancehuntObject`
 Provide operations with Profiles API part.

Warning: For directly usage please set *token* argument.

Parameters *token* (*str*) – your API token, optional

get_employer_details (*profile_id*)

Get information about employer by identifier.

Parameters *profile_id* (*int*) – the desired profile identifier

Return type *Employer*

get_employers_list (*country_id=None, city_id=None, login=None, pages=1*)

Get filtered employer profiles.

Parameters

- **country_id** (*Optional[int]*) – employer from country (API-related Country identifier), defaults to None
- **city_id** (*Optional[int]*) – employer from city (API-related City identifier), defaults to None
- **login** (*Optional[str]*) – with the desired login, defaults to None
- **pages** (*Union[int, Tuple[int], List[int], None]*) – number of pages, defaults to 1

Return type *List[Employer]*

Returns list of filtered employer profiles

get_freelancer_details (*profile_id*)

Get information about freelancer by identifier.

Parameters *profile_id* (*int*) – the desired profile identifier

Return type *Freelancer*

get_freelancers_list (*country_id=None, city_id=None, skill_id=None, login=None, pages=1*)

Get filtered freelancer profiles.

Parameters

- **country_id** (*Optional[int]*) – freelancer from country (API-related Country identifier), defaults to None
- **city_id** (*Optional[int]*) – freelancer from city (API-related City identifier), defaults to None
- **skill_id** (*Optional[int]*) – freelancer skill (API-related Skill identifier), defaults to None

- **login** (Optional[str]) – with the desired login, defaults to None
- **pages** (Union[int, Tuple[int], List[int], None]) – number of pages, defaults to 1

Return type List[Freelancer]

Returns list of filtered freelancer profiles

property my_profile

Get my profile information.

Return type Union[Employer, Freelancer]

Returns information of your account

class freelancehunt.**Projects** (token=None, **kwargs)
 Bases: freelancehunt.core.FreelancehuntObject
 Provide operations with Projects API part.

Warning: For directly usage please set *token* argument.

Parameters **token** (str) – your API token, optional

create_project (information)

Create new project on site.

TBD: Implement a convenient way to create a project.

Parameters **information** (dict) – required and optional params of new project (link: <https://apidocs.freelancehunt.com/?version=latest#ff11ae15-05af-4ee8-ae7c-155cd137506f>)

Return type Project

Returns representation of created project.

get_list (pages=1, only_for_plus=False, skills=None, employer_id=None)

Get projects with filter and from multiple pages.

Parameters

- **skills** (Union[int, str, Skill, List[Skill], List[int], Tuple[Skill], Tuple[int], None]) – filter by skills
- **employer_id** (Optional[int]) – projects from employer with id
- **only_for_plus** (bool) – filter only for plus if False, get otherwise, defaults is False
- **pages** (Union[int, Tuple[int], List[int]]) – number of pages to get, defaults - 1

Return type List[Project]

get_project (project_id)

Get specific project by id.

Parameters **project_id** (int) – id of the desired project.

Return type Project

Returns the desired project object.

my_projects (pages=1)

Get my projects list (10 objects).

Parameters `pages` (`Union[int, Tuple[int], List[int]]`) – number of pages, defaults to 1

Raises `BadRequest` – raises when you are not Employer.

Return type `List[Project]`

```
class freelancehunt.Requester (token, language='en', **kwargs)
```

Bases: `object`

Provides requests to API. Singleton.

```
classmethod get_requester (token=None, **kwargs)
```

limit = `None`

```
request (request_type, url, filters=None, payload=None)
```

Make request to API and handle results.

Args: `request_type` (`str`): “POST”, “GET”, “PATCH” or “DEL”.

Return: `dict`: JSON response data in dict

request_date = `None`

token = `None`

```
class freelancehunt.Reviews (token=None, **kwargs)
```

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Reviews API part.

Warning: For directly usage please set `token` argument.

Parameters `token` (`str`) – your API token, optional

```
get_my_reviews ()
```

Get reviews of my profile.

Return type `List[Type[Review]]`

Returns profile reviews

```
get_reviews (profile_type, profile_id)
```

Get reviews of the desired profile.

Parameters

- **profile_type** (`str`) – type of the desired profile, can be “freelancer” or “employer”
- **profile_id** (`int`) – identifier of the desired profile

Return type `List[Type[Review]]`

Returns profile reviews

```
class freelancehunt.Skills (token=None, **kwargs)
```

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Skills API part.

Note: This module contains static content. It may be *update()*, but loaded info does not change on the API side.

Warning: For directly usage please set *token* argument.

Parameters *token* (*str*) – your API token, optional

find (*text*)

Find the names of the skill that contain the desired text.

Parameters *text* (*str*) – the desired text that need to be in an skill name.

Raises **ValueError** – Skill not found.

Return type *List[Skill]*

Returns list of skills with an text in name.

get (*skill_id*)

Get the desired skill by API-related identifier.

Parameters *skill_id* (*int*) – identifier of the desired skill.

Return type *Skill*

Returns the desired skill

property list

Get list of all skills.

Return type *List[Skill]*

Returns list of skills

update ()

Update static information from API.

class *freelancehunt.Threads* (*token=None, **kwargs*)

Bases: *freelancehunt.core.FreelancehuntObject*

Provide operations with Threads API part.

Warning: For directly usage please set *token* argument.

Parameters *token* (*str*) – your API token, optional

create_thread (*to_profile_id, subject, message_html*)

Create new thread.

Parameters

- *to_profile_id* (*int*) – recipient profile id
- *subject* (*str*) – thread subject
- *message_html* (*str*) – the first thread's message

Return type *Thread*

Returns created thread

get_threads (*pages=1*)

Get list of threads.

Parameters `Tuple[int], List[int]` **pages** (*Union[int,)* – count of pages to get, defaults to 1

Return type `List[Thread]`

1.5.2 freelancehunt.models package

freelancehunt.models.bid module

Bid object representation.

Documentation example:

```
{
  "id": 3119168,
  "type": "bid",
  "attributes": {
    "days": 25,
    "safe_type": "employer",
    "budget": {
      "amount": 18000,
      "currency": "UAH"
    }
  },
  "comment": "I can help you with this problem.",
  "status": "active",
  "is_hidden": false,
  "is_winner": false,
  "freelancer": {
    "id": 191397,
    "type": "freelancer",
    "login": "NeoSeo",
    "first_name": "Maxim",
    "last_name": "P.",
    "avatar": {
      "small": {
        "url": "https://content.freelancehunt.com/profile/photo/50/NeoSeo.png",
        "width": 50,
        "height": 50
      },
      "large": {
        "url": "https://content.freelancehunt.com/profile/photo/225/NeoSeo.png",
        "width": 255,
        "height": 255
      }
    }
  },
  "self": "http://api.freelancehunt.com/v2/freelancers/191397"
},
"project": {
  "id": 299172,
  "type": "project",
  "name": "Project with bids",
  "status": {
    "id": 11,
```

(continues on next page)

(continued from previous page)

```

        "name": "Open for proposals"
    },
    "safe_type": "employer",
    "budget": {
        "amount": 2300,
        "currency": "UAH"
    },
    "self": "http://api.freelancehunt.com/v2/projects/299172"
},
"attachment": null,
"published_at": "2018-04-15T23:44:05+03:00"
}
},

```

```

class freelancehunt.models.bid.Bid(id, status, days, safe_type, comment, currency, is_hidden,
                                     is_winner, freelancer, budget, project, **kwargs)
    Bases: freelancehunt.core.FreelancehuntObject

```

Provide operations with Bid API object.

Variables

- **id** (*int*) – bid unique identifier
- **status** (*str*) – bid current status
- **days** (*int*) – days estimated by freelancer for complete this project
- **safe_type** (*str*) – type of payment method (for safe)
- **comment** (*str*) – comment from freelancer about this project
- **currency** (*str*) – freelancer currency for bid
- **is_hidden** (*bool*) – mark hidden from others bid
- **is_winner** (*bool*) – mark winner bid
- **freelancer** (*Freelancer*) – bid creator information
- **budget** (*BudgetInfo*) – estimated budget for this project
- **project** (*Project*) – related project information

choose (*comment*)

Choose this bid.

Note: Only for Employer and your own project.

Parameters **comment** (*str*) – comment for winner to start dialog with freelancer.

Returns status of operation.

Return type bool

classmethod **de_json** (***data*)

Parse json data from API response and make object of this class.

Returns object of this class.

Return type *Bid*

reject ()

Reject this bid.

Note: Only for Employer and your own project.

Returns status of operation.

Return type bool

restore ()

Restore your bid.

Returns status of operation.

Return type bool

revoke ()

Revoke your bid.

Note: Only for Freelancer and your own bid.

Returns status of operation.

Return type bool

freelancehunt.models.budget module

Budget object representation.

Documentation example:

```
"budget": {
  "amount": 2500,
  "currency": "UAH"
},
```

class freelancehunt.models.budget.**BudgetInfo** (*amount, currency, **kwargs*)

Bases: freelancehunt.core.FreelancehuntObject

Provide operations with Budget.

Variables

- **amount** (*int*) – amount of budget
- **currency** (*str*) – current currency

classmethod **de_json** (***data*)

Parse json data from API response and make object of this class.

Returns object of this class.

Return type *BudgetInfo*

property **is_rub**

Check that currency is Russian Ruble.

Returns True if currency is ruble, False otherwise

Return type bool

property is_uah

Check that currency is Ukrainian Hryvnia.

Returns True if currency is hryvnia, False otherwise

Return type bool

freelancehunt.models.city module

City object representation.

Documentation example:

```
{
  "id": 2627,
  "name": ""
},
```

class freelancehunt.models.city.**City** (*id, name, **kwargs*)

Bases: freelancehunt.core.FreelancehuntObject

Provide operations with City.

Variables

- **id** (*int*) – city API identifier
- **name** (*str*) – city name

classmethod **de_json** (***data*)

Parse json data from API response and make object of this class.

Returns object of this class.

Return type *City*

freelancehunt.models.contest module

Contest object representation.

Documentation example:

```
{
  "id": 1390,
  "type": "contest",
  "attributes": {
    "name": "New contest",
    "description": "Transferring party -- The Customers and the Contractors owning_
↪ confidential information and personal data transferred by them to the Website_
↪ Administration during registration and use of the Service.",
    "description_html": "<p>Transferring party -- The Customers and the Contractors_
↪ owning confidential information and personal data transferred by them to the_
↪ Website Administration during registration and use of the Service.</p>",
    "skill": {
      "id": 9,
      "name": "Icon"
    },
  },
  "status": {
```

(continues on next page)

(continued from previous page)

```

        "id": 140,
        "name": "Final"
    },
    "budget": {
        "amount": 1002,
        "currency": "UAH"
    },
    "application_count": 1,
    "published_at": "2019-04-08T17:05:25+03:00",
    "duration_days": 5,
    "final_started_at": "2019-04-13T17:05:25+03:00",
    "freelancer": null,
    "employer": {
        "id": 38444,
        "type": "employer",
        "login": "jeweller",
        "first_name": "Oleg",
        "last_name": "Vinnik",
        "avatar": {
            "small": {
                "url": "https://content.freelancehunt.com/profile/photo/50/jeweller.png",
                "width": 50,
                "height": 50
            },
            "large": {
                "url": "https://content.freelancehunt.com/profile/photo/225/jeweller.png",
                "width": 255,
                "height": 255
            }
        },
        "self": "https://api.freelancehunt.com/v2/employers/38444"
    },
    "updates": []
},
"links": {
    "self": {
        "api": "https://api.freelancehunt.com/v2/contests/1390",
        "web": "https://freelancehunt.com/contest/new-contest/1390.html"
    },
    "comments": "https://api.freelancehunt.com/v2/contests/1390/comments",
    "applications": "https://api.freelancehunt.com/v2/contests/1390/applications"
}
}

```

```

class freelancehunt.models.contest.Contest(id, name=None, budget=None, status=None,
                                           description=None, description_html=None,
                                           skills=None, final_started_at=None, em-
                                           ployer=None, application_count=None,
                                           freelancer=None, updates=None, pub-
                                           lished_at=None, links=None, dura-
                                           tion_days=None, **kwargs)

```

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Contest.

Variables

- `id(int)` – contest unique identifier

- **name** (*Optional[str]*) – contest title name, defaults to None
- **budget** (*Optional[BudgetInfo]*) – budget of contest, defaults to None
- **status** (*Optional[str]*) – current contest status, defaults to None
- **description** (*Optional[str]*) – description text, defaults to None
- **description_html** (*Optional[str]*) – description text in html formatting, defaults to None
- **skills** (*Optional[List[Skill]]*) – required skill for freelancer, defaults to None
- **final_started_at** (*Optional[datetime]*) – the closing date, defaults to None
- **employer** (*Optional[Employer]*) – employer information object, defaults to None
- **application_count** (*Optional[int]*) – count of application, defaults to None
- **freelancer** (*Optional[Freelancer]*) – freelancer information object, defaults to None
- **updates** (*Optional[list]*) – list of all contest updates, defaults to None
- **published_at** (*Optional[datetime]*) – the publish date, defaults to None
- **links** (*Optional[dict]*) – linked URL, defaults to None
- **duration_days** (*Optional[int]*) – contest duration in days, defaults to None

classmethod de_json (***data*)

Parse json data from API response and make object of this class.

Return type `Type[ForwardRef]`

Returns object of this class

load_details ()

Load details about current Contest and reload all attributes.

property status

Get status of this contest.

Return type `str`

property status_code

Get status code of this contest.

Return type `int`

freelancehunt.models.country module

Country object representation.

Documentation example:

```
{
    "id": 4,
    "iso2": "AU",
    "name": ""
},
```

class `freelancehunt.models.country.Country` (*id, name, iso2=None, **kwargs*)

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Country.

Variables

- **id** (*int*) – country API identifier
- **iso2** (*str*) – ISO format of country identifier, defaults to None (for Profile “country” var)
- **name** (*str*) – country name

classmethod **de_json** (***data*)

Parse json data from API response and make object of this class.

Returns object of this class.

Return type *Country*

freelancehunt.models.feed module

Feed message object representation.

Documentation example:

```
{
  "id": 1555675871044,
  "type": "feed",
  "attributes": {
    "from": {
      "id": 4,
      "type": "employer",
      "login": "freelancehunt",
      "first_name": "Freelancehunt",
      "last_name": "",
      "avatar": {
        "small": {
          "url": "https://content.freelancehunt.com/profile/photo/50/freelancehunt.png",
          "width": 50,
          "height": 50
        },
        "large": {
          "url": "https://content.freelancehunt.com/profile/photo/225/freelancehunt.png",
          "width": 255,
          "height": 255
        }
      },
      "self": "https://api.freelancehunt.com/v2/employers/4"
    },
    "message": "<img src='https://freelancehunt.com/static/images/fugu/new-text.png' width='16' height='16'/> New message for you.",
    "created_at": "2019-04-19T12:11:11+00:00",
    "is_new": false
  }
}
```

```
class freelancehunt.models.feed.FeedMessage(id, message_from, message, created_at,
                                             is_new, project=None, contest=None,
                                             **kwargs)
```

Bases: freelancehunt.core.FreelancehuntObject

Provide operations with Feed message.

Variables

- **id** (*int*) – feed message unique identifier
- **message_from** (*Union[Employer, Freelancer]*) – message sender user object
- **message** (*str*) – the feed message text
- **created_at** (*datetime*) – of the creation date
- **is_new** (*bool*) – sign that the message is new
- **project** (*Optional[Project]*) – linked project object, defaults to None
- **contest** (*Optional[Contest]*) – linked contest object, defaults to None

property contest

Represent the Contest linked to this Feed message.

Return type *Optional[Contest]*

Returns object of linked Contest or None.

classmethod de_json (data)**

Parse json data from API response and make object of this class.

Returns object of this class.

Return type *FeedMessage*

property is_contest

Check that Feed message linked to some contest.

Return type *bool*

property is_project

Check that Feed message linked to some project.

Return type *bool*

property project

Represent the Project linked to this Feed message.

Return type *Optional[Project]*

Returns object of linked Project or None.

property sender

Load and get sender information.

Return type *Union[Employer, Freelancer]*

property type

Check type of Feed message.

Return type *str*

Returns type of Feed message: “project”, “contest” or “message”.

freelancehunt.models.project module

Project object representation.

Documentation example:

```
{
  "id": 299165,
  "type": "project",
  "attributes": {
    "name": "Looking for Full stack developer",
    "description": "Backend must be PHP like Yii or Laravel.",
    "description_html": "<p>Backend must be PHP like Yii or Laravel.</p>",
    "skills": [
      {
        "id": 56,
        "name": "1C"
      }
    ],
    "status": {
      "id": 11,
      "name": "Open for proposals"
    },
    "budget": {
      "amount": 2300,
      "currency": "UAH"
    },
    "bid_count": 1,
    "is_remote_job": false,
    "is_premium": false,
    "is_only_for_plus": false,
    "location": null,
    "safe_type": "employer",
    "is_personal": null,
    "employer": {
      "id": 23476,
      "type": "employer",
      "login": "hello-world",
      "first_name": "",
      "last_name": ".",
      "avatar": {
        "small": {
          "url": "https://content.freelancehunt.com/profile/photo/50/hello-world.png",
          "width": 50,
          "height": 50
        },
        "large": {
          "url": "https://content.freelancehunt.com/profile/photo/225/hello-world.png",
          "width": 255,
          "height": 255
        }
      }
    },
    "self": "https://api.freelancehunt.com/v2/employers/23476"
  },
  "freelancer": null,
  "updates": [],
  "published_at": "2019-03-25T19:51:53+02:00",
}
```

(continues on next page)

(continued from previous page)

```

    "expired_at": "2019-04-01T16:51:53+03:00"
  },
  "links": {
    "self": {
      "api": "https://api.freelancehunt.com/v2/projects/299165",
      "web": "https://freelancehunt.com/project/looking-for-full-stack-developer/
↪299165.html"
    },
    "comments": "https://api.freelancehunt.com/v2/projects/299165/comments",
    "bids": "https://api.freelancehunt.com/v2/projects/299165/bids"
  }
}

```

```

class freelancehunt.models.project.Project(id, name=None, employer=None, free-
lancer=None, budget=None, safe_type=None,
status=None, description=None, de-
scription_html=None, skills=None,
bid_count=None, is_remote_job=None,
is_premium=None, is_only_for_plus=None,
is_personal=None, updates=None, lo-
cation=None, expired_at=None, pub-
lished_at=None, links=None, **kwargs)

```

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Project.

Variables

- **id** (*int*) – project unique identifier
- **name** (*Optional[str]*) – project title, defaults to None
- **employer** (*Optional[Employer]*) – project creator information, defaults to None
- **freelancer** (*Optional[Freelancer]*) – project executor information, defaults to None
- **budget** (*Optional[BudgetInfo]*) – budget of project, defaults to None
- **safe_type** (*Optional[str]*) – type of safe proposed by the employer, defaults to None
- **status** (*Optional[dict]*) – project status information, defaults to None
- **description** (*Optional[str]*) – description text, defaults to None
- **description_html** (*Optional[str]*) – description text in html formatting, defaults to None
- **skills** (*Optional[List[Skill]]*) – required skills for the executor, defaults to None
- **bid_count** (*Optional[int]*) – count of bids, defaults to None
- **is_remote_job** (*Optional[bool]*) – sign that the project is remote job, defaults to None
- **is_premium** (*Optional[bool]*) – sign that the project only for premium, defaults to None
- **is_only_for_plus** (*Optional[bool]*) – sign that the project only for Plus accounts, defaults to None

- **is_personal** (*Optional[bool]*) – sign that the project is private, defaults to None
- **updates** (*Optional[dict]*) – project updates, defaults to None
- **location** (*Optional[dict]*) – required location for the executor, defaults to None
- **expired_at** (*Optional[datetime]*) – the expire date, defaults to None
- **published_at** (*Optional[datetime]*) – the publish date, defaults to None
- **links** (*Optional[dict]*) – URLs related to project, defaults to None

property active_bids

Get active bids for this project.

Return type `List[Optional[Type[ForwardRef]]]`

property bids

Get all bids for this project.

Return type `List[Optional[Type[ForwardRef]]]`

close()

Close project without winner.

Note: For employer account and your own project.

classmethod de_json (data)**

Parse json data from API response and make object of this class.

Returns object of this class.

Return type *Project*

extend(expired_at)

Extend project end date.

Note: For employer account and your own project.

get_bids(status=None, is_winner=False)

Get filtered bids for this project.

Parameters

- **status** (*Optional[str]*) – status of desired bids
- **is_winner** (*bool*) – get only winner bid

Return type `List[Optional[Type[ForwardRef]]]`

property link

Get direct project link.

load_details()

Load details about current Project and reload all attributes.

reopen()

Reopen project.

Note: For employer account and your own project.

property status

Get current project status.

Return type str

property status_code

Get code of current project status.

Return type int

property winner_bid

Get winner bid for this project.

Return type Optional[Type[ForwardRef]]

freelancehunt.models.review module

Review object representation.

Documentation example:

```
{
  "id": 90500,
  "type": "review",
  "attributes": {
    "published_at": "2019-04-25T15:02:19+03:00",
    "is_pending": false,
    "pending_ends_at": null,
    "comment": "Everything is perfect!",
    "grades": {
      "quality": 10,
      "professionalism": 10,
      "cost": 10,
      "connectivity": 10,
      "schedule": 10,
      "total": 10
    }
  },
  "from": {
    "id": 38444,
    "type": "employer",
    "login": "jeweller",
    "first_name": "Oleg",
    "last_name": "V.",
    "avatar": {
      "small": {
        "url": "https://content.freelancehunt.com/profile/photo/50/jeweller.png",
        "width": 50,
        "height": 50
      },
      "large": {
        "url": "https://content.freelancehunt.com/profile/photo/225/jeweller.png",
        "width": 255,
        "height": 255
      }
    }
  },
  "self": "https://api.freelancehunt.com/v2/employers/38444"
},
{
  "project": {
    "id": 299178,
```

(continues on next page)

(continued from previous page)

```

    "type": "project",
    "name": "Project with response review(good)",
    "status": {
        "id": 21,
        "name": "Project complete"
    },
    "safe_type": "employer",
    "budget": {
        "amount": 1001,
        "currency": "UAH"
    },
    "self": "https://api.freelancehunt.com/v2/projects/299178"
}
}
}

```

```

class freelancehunt.models.review.Review(id, published_at, comment, is_pending, grades,
                                          creator, pending_ends_at=None, project=None,
                                          **kwargs)

```

Bases: `freelancehunt.core.FreelancehuntObject`

Create object to provide operations with Project.

Variables

- **id** (*int*) – review unique identifier
- **published_at** (*datetime*) – the publish date
- **comment** (*str*) – comment from creator
- **grades** (*dict*) – grades of work quality
- **creator** (*Union[Employer, Freelancer]*) – review creator information
- **is_pending** (*bool*) – sign that the project is pending
- **pending_ends_at** (*Optional[datetime]*) – the pending date, defaults to None
- **project** (*Optional[Project]*) – related project object, defaults to None

```

classmethod de_json(**data)

```

Parse json data from API response and make object of this class.

Return type `Type[Review]`

Returns object of this class.

freelancehunt.models.skill module

Skill object representation.

Documentation example:

```

{
    "id": 108,
    "name": "Architectural design"
}

```

```

class freelancehunt.models.skill.Skill(id, name, **kwargs)

```

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Skill.

Variables

- **id** (*int*) – skill unique identifier
- **name** (*str*) – skill name

classmethod de_json (***data*)

Parse json data from API response and make object of this class.

Returns object of this class.

Return type *Skill*

freelancehunt.models.thread module

Thread object representation.

Documentation example:

```
{
  "id": 2237325,
  "type": "thread",
  "attributes": {
    "subject": "Workspace for project High-budget project",
    "first_post_at": "2019-02-21T14:12:35+02:00",
    "last_post_at": "2019-02-21T14:12:35+02:00",
    "messages_count": 1,
    "is_unread": false,
    "has_attachments": false,
    "participants": {
      "from": {
        "id": 340096,
        "type": "employer",
        "login": "ledpodarok",
        "first_name": "Anatoliy",
        "last_name": "S.",
        "avatar": {
          "small": {
            "url": "https://content.freelancehunt.com/profile/photo/50/ledpodarok.png
↪",
            "width": 50,
            "height": 50
          },
          "large": {
            "url": "https://content.freelancehunt.com/profile/photo/225/ledpodarok.png
↪",
            "width": 255,
            "height": 255
          }
        },
        "self": "http://api.freelancehunt.com/v2/employers/340096"
      },
      "to": {
        "id": 3166,
        "type": "freelancer",
        "login": "raznomir",
        "first_name": "Mikhail",
```

(continues on next page)

(continued from previous page)

```

        "last_name": "Tereshchenko",
        "avatar": {
            "small": {
                "url": "https://content.freelancehunt.com/profile/photo/50/raznomir.png",
                "width": 50,
                "height": 50
            },
            "large": {
                "url": "https://content.freelancehunt.com/profile/photo/225/raznomir.png",
                "width": 255,
                "height": 255
            }
        },
        "self": "http://api.freelancehunt.com/v2/freelancers/3166"
    }
},
"links": {
    "self": {
        "api": "http://api.freelancehunt.com/v2/threads/2237325",
        "web": "http://freelancehunt.com/mailbox/read/thread/2237325"
    }
}
}

```

```

class freelancehunt.models.thread.Thread(id, subject, first_post_at, last_post_at, mes-
                                         sages_count, is_unread, has_attachments, sender,
                                         recipient, **kwargs)

```

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Thread.

Parameters

- **id** (*int*) – thread unique identifier
- **subject** (*str*) – thread subject text
- **first_post_at** (*datetime*) – the first message date
- **last_post_at** (*datetime*) – the last message date
- **messages_count** (*int*) – count of messages in thread
- **is_unread** (*bool*) – sign that the thread is unread
- **has_attachments** (*bool*) – sign that the thread has attachments
- **sender** (`Profile`) – thread creator information
- **recipient** (`Profile`) – thread recipient information

answer (*message_html*)

classmethod de_json (***data*)

Parse json data from API response and make object of this class.

Returns object of this class.

Return type *Thread*

get_messages (*pages=1*)

Return type `List[ThreadMessage]`

freelancehunt.models.threadmessage module

ThreadMessage object representation.

Documentation example:

```
{
  "id": 67,
  "type": "message",
  "attributes": {
    "message": "I choose you!",
    "message_html": "I choose you!",
    "posted_at": "2019-04-03T09:33:05+03:00",
    "attachments": [],
    "participants": {
      "from": {
        "id": 38444,
        "type": "employer",
        "login": "jeweller",
        "first_name": "Oleg",
        "last_name": "V.",
        "avatar": {
          "small": {
            "url": "https://content.freelancehunt.com/profile/photo/50/jeweller.png",
            "width": 50,
            "height": 50
          },
          "large": {
            "url": "https://content.freelancehunt.com/profile/photo/225/jeweller.png",
            "width": 255,
            "height": 255
          }
        }
      },
      "self": "https://api.freelancehunt.com/v2/employers/38444"
    },
    "to": {
      "id": 5725,
      "type": "freelancer",
      "login": "alex_yar",
      "first_name": "Andrey",
      "last_name": "Y.",
      "avatar": {
        "small": {
          "url": "https://content.freelancehunt.com/profile/photo/50/alex_yar.png",
          "width": 50,
          "height": 50
        },
        "large": {
          "url": "https://content.freelancehunt.com/profile/photo/225/alex_yar.png",
          "width": 255,
          "height": 255
        }
      },
      "self": "https://api.freelancehunt.com/v2/freelancers/5725"
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

    }
}
}

```

```

class freelancehunt.models.threadmessage.ThreadMessage(id, posted_at, message,
                                                         message_html, sender, re-
                                                         cipient, attachments=None,
                                                         thread=None, **kwargs)

```

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with ThreadMessage.

Parameters

- **id** (*int*) – thread unique identifier
- **message** (*str*) – message text
- **message_html** (*str*) – message text in html
- **posted_at** (*datetime*) – the message post date
- **attachments** (*dict*) – message's attachments
- **sender** (*Profile*) – message creator information
- **recipient** (*Profile*) – message recipient information

answer (*message_html*)

Answer to this message in current thread.

Parameters **message_html** (*str*) – message text to send

Raises

- **ValueError** – Thread not linked to this message!
- **BadRequestError** – Message not sended!

Return type `Type[ThreadMessage]`

Returns new message object

classmethod **de_json** (***data*)

Parse json data from API response and make object of this class.

Returns object of this class.

Return type `ThreadMessage`

freelancehunt.models.user module

Profile object representation.

Documentation example:

```

{
  "id": 77278,
  "type": "freelancer",
  "attributes": {
    "login": "Artemkins",
    "first_name": "",

```

(continues on next page)

(continued from previous page)

```
"last_name": "Yacuk",
"avatar": {
  "small": {
    "url": "https://content.freelancehunt.com/profile/photo/50/Artemkins.png",
    "width": 50,
    "height": 50
  },
  "large": {
    "url": "https://content.freelancehunt.com/profile/photo/225/Artemkins.png",
    "width": 255,
    "height": 255
  }
},
"birth_date": null,
"created_at": "2014-09-08T15:21:45+03:00",
"cv": null,
"cv_html": null,
"rating": 31597,
"rating_position": 1,
"arbitrages": 0,
"positive_reviews": 295,
"negative_reviews": 0,
"plus_ends_at": null,
"is_plus_active": true,
"is_online": false,
"visited_at": null,
"location": {
  "country": {
    "id": 1,
    "name": "Ukraine"
  },
  "city": {
    "id": 3184,
    "name": "Kharkiv"
  }
},
"verification": {
  "identity": true,
  "birth_date": false,
  "phone": true,
  "website": false,
  "wmid": false,
  "email": false
},
"contacts": null,
"status": {
  "id": 10,
  "name": "Available for hire"
},
"skills": [
  {
    "id": 14,
    "name": "Search engine optimization",
    "rating_position": 0
  },
  {
    "id": 127,
```

(continues on next page)

(continued from previous page)

```

        "name": "Contextual advertising",
        "rating_position": 0
    },
    {
        "id": 131,
        "name": "Social media marketing",
        "rating_position": 0
    }
]
},
"links": {
    "self": {
        "api": "https://api.freelancehunt.com/v2/freelancers/77278",
        "web": "https://freelancehunt.com/freelancer/Artemkins.html"
    },
    "reviews": "https://api.freelancehunt.com/v2/freelancers/77278/reviews"
}
}

```

```

class freelancehunt.models.user.Employer(id, login, type, first_name, last_name,
                                           avatar=None, birth_date=None, created_at=None,
                                           cv=None, cv_html=None, rating=None, rating_position=None,
                                           arbitrages=None, positive_reviews=None, negative_reviews=None,
                                           plus_ends_at=None, is_plus_active=None, is_online=None,
                                           visited_at=None, location=None, verification=None,
                                           contacts=None, status=None, skills=None, links=None, **kwargs)

```

Bases: `freelancehunt.models.user.Profile`

Provide operations with Employer profile.

```

class freelancehunt.models.user.Freelancer(id, login, type, first_name, last_name,
                                             avatar=None, birth_date=None, created_at=None,
                                             cv=None, cv_html=None, rating=None, rating_position=None,
                                             arbitrages=None, positive_reviews=None, negative_reviews=None,
                                             plus_ends_at=None, is_plus_active=None, is_online=None,
                                             visited_at=None, location=None, verification=None,
                                             contacts=None, status=None, skills=None, links=None, **kwargs)

```

Bases: `freelancehunt.models.user.Profile`

Provide operations with Freelancer profile.

```

class freelancehunt.models.user.Profile(id, login, type, first_name, last_name,
                                         avatar=None, birth_date=None, created_at=None,
                                         cv=None, cv_html=None, rating=None, rating_position=None,
                                         arbitrages=None, positive_reviews=None, negative_reviews=None,
                                         plus_ends_at=None, is_plus_active=None, is_online=None,
                                         visited_at=None, location=None, verification=None,
                                         contacts=None, status=None, skills=None, links=None, **kwargs)

```

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Profile.

Note: Information will be loaded by `load_details()` if attribute is *None*

Variables

- **id** (*int*) – unique profile identifier
- **login** (*str*) – profile login name
- **type** (*str*) – type of account
- **first_name** (*str*) – user first name
- **last_name** (*str*) – user last name
- **avatar** (*dict*) – avatars information, defaults to None
- **birth_date** (*datetime*) – string representation of the birth date, defaults to None
- **created_at** (*str*) – string representation of the profile create date, defaults to None
- **cv** (*str*) – profile summary information, defaults to None
- **cv_html** (*str*) – profile summary information in html formatting, defaults to None
- **rating** (*int*) – user rating points, defaults to None
- **rating_position** (*int*) – user rating position, defaults to None
- **arbitrages** (*int*) – count of arbitrages, defaults to None
- **positive_reviews** (*int*) – count of positive reviews, defaults to None
- **negative_reviews** (*int*) – count of negative reviews, defaults to None
- **plus_ends_at** (*Optional[datetime]*) – string representation of the “Plus” status end date, defaults to None
- **is_plus_active** (*bool*) – sign that the user has active a “Plus” status, defaults to None
- **is_online** (*bool*) – sign that the user is online, defaults to None
- **visited_at** (*datetime*) – string representation of the last visited date, defaults to None
- **verification** (*dict*) – verification statuses, defaults to None
- **contacts** (*dict*) – user contacts (links), defaults to None
- **skills** (*List[Skill]*) – user skills, defaults to None
- **links** (*dict*) – URLs related to profile, defaults to None

property city

Get user city.

Return type *City*

property country

Get user country.

Return type *Country*

```

classmethod de_json (**data)
    Parse json data from API response and make object of this class.

    Return type Type[Profile]

    Returns object of this class.

property full_name
    Get the full name for the current profile.

    Return type str

    Returns full name of the user

load_details ()
    Load details about current User and reload all attributes.

property profile_url
    Get public profile URL.

    Return type str

    Returns public URL for the user

property reviews
    Get reviews of this profile.

    Return type List[Type[Review]]

property status
    Get current profile status.

    Return type str

property status_code
    Get code of profile status.

    Return type int

```

1.5.3 freelancehunt.packages package

freelancehunt.packages.bids module

[Freelancehunt Documentation - Bids API.](#)

```

class freelancehunt.packages.bids.Bids (token=None, **kwargs)
    Bases: freelancehunt.core.FreelancehuntObject

    Provide operations with Bids API part.

```

Warning: For directly usage please set *token* argument.

Parameters **token** (*str*) – your API token, optional

```

choose (project_id, bid_id, comment)
    Choose this bid.

```

Note: Only for Employer and your own project.

Parameters

- **project_id** (int) – get bid for the desired project
- **bid_id** (int) – bid identifier
- **comment** (str) – comment for winner to start dialog with freelancer

Return type bool

Returns status of operation

get_my_bids (*project_id=None, status=None*)

Get my filtered bids.

Parameters

- **project_id** (Optional[int]) – get bid for the desired project or all (None), defaults to None
- **status** (Optional[str]) – get bids with the desired status, defaults to None

Return type List[Bid]

Returns list of my filtered bids

get_project_bids (*project_id, status=None, is_winner=False*)

Get filtered projects bid.

Parameters

- **project_id** (int) – project where find bids
- **status** (Optional[str]) – get bids with the desired status or all (None), defaults to None
- **is_winner** (bool) – get winner bid or all (False), defaults to False

Return type List[Bid]

Returns list of filtered bids

property my_active_bids

Get my active bids.

Return type List[Bid]

Returns list of my active bids

reject (*project_id, bid_id*)

Reject this bid.

Note: Only for Employer and your own project.

Parameters

- **project_id** (int) – get bid for the desired project
- **bid_id** (int) – bid identifier

Return type bool

Returns status of operation

restore_bid (*project_id*, *bid_id*)

Restore your bid.

Parameters

- **project_id** (*int*) – get bid for the desired project
- **bid_id** (*int*) – bid identifier

Return type `bool`

Returns status of operation

revoke_bid (*project_id*, *bid_id*)

Revoke your bid.

Note: Only for Freelancer and your own bid.

Parameters

- **project_id** (*int*) – get bid for the desired project
- **bid_id** (*int*) – bid identifier

Return type `bool`

Returns status of operation

freelancehunt.packages.cities module

Freelancehunt Documentation - Cities API.

class `freelancehunt.packages.cities.Cities` (*country_id*, *token=None*, ***kwargs*)

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Cities API part.

Note: This module contains static content. It may be *update()*, but loaded info does not change on the API side.

Warning: For directly usage please set *token* argument.

Parameters

- **token** (*str*) – your API token, optional
- **country_id** (*int*) – API-related country identifier to get cities of it

find (*text*)

Find the names of the cities that contain the desired text.

Parameters **text** (*str*) – the desired text that need to be in an city name.

Return type `List[City]`

Returns list of cities with an text in name.

get (*city_id*)

Get the desired city by city_id.

Parameters **city_id** (int) – id of the desired city (<https://apidocs.freelancehunt.com/?version=latest#65a2a9b4-b52d-4706-8c54-990600b58c2b>).

Return type *City*

Returns the desired city.

property list

Get list of all cities.

Return type *List[City]*

Returns list of cities.

update ()

Update static information from API.

Return type *None*

freelancehunt.packages.contests module

[Freelancehunt Documentation - Contests API.](#)

class `freelancehunt.packages.contests.Contests` (*token=None, **kwargs*)

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Contests API part.

Note: NOT IMPLEMENTED YET!

Warning: For directly usage please set *token* argument.

Parameters **token** (*str*) – your API token, optional

freelancehunt.packages.countries module

[Freelancehunt Documentation - Countries API.](#)

class `freelancehunt.packages.countries.Countries` (*token=None, **kwargs*)

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Countries API part.

Note: This module contains static content. It may be *update()*, but loaded info does not change on the API side.

Warning: For directly usage please set *token* argument.

Parameters **token** (*str*) – your API token, optional

property cities

Cities linked to this Country.

Return type *Cities*

Returns object for manipulate with Cities API part

find(text)

Find countries with the desired text.

Parameters **text** (*str*) – text in country name

Raises **ValueError** – No countries found

Return type *List[Country]*

Returns list of countries with an text in name

get(country_id=None, iso_code=None)

Get the filtered country.

Parameters

- **country_id** (*Optional[int]*) – the desired API-related country id, defaults to None
- **iso_code** (*Optional[str]*) – the desired ISO code of country, defaults to None

Raises

- **AttributeError** – No one of parameters filled
- **ValueError** – Contry not found

Return type *Country*

Returns the desired country

property list

Get list of all countries.

Return type *List[Country]*

Returns list of countries

update()

Update static information from API.

freelancehunt.packages.feed module

Freelancehunt Documentation - Feed API.

class `freelancehunt.packages.feed.Feed(token=None, **kwargs)`

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Feed API part.

Warning: For directly usage please set *token* argument.

Parameters **token** (*str*) – your API token, optional

property contests

Get all messages with linked contest in feed.

Return type `List[FeedMessage]`

get_new()
Get all new notifications in feed.

Return type `List[FeedMessage]`

property list
Get all feed messages.

Return type `List[FeedMessage]`

property projects
Get all messages with linked project in feed.

Return type `List[FeedMessage]`

read()
Mark feed as read.

update()
Get latest feed information.

freelancehunt.packages.profiles module

Freelancehunt Documentation - Profiles API.

class `freelancehunt.packages.profiles.Profiles` (*token=None, **kwargs*)
Bases: `freelancehunt.core.FreelancehuntObject`
Provide operations with Profiles API part.

Warning: For directly usage please set *token* argument.

Parameters *token* (*str*) – your API token, optional

get_employer_details (*profile_id*)
Get information about employer by identifier.

Parameters *profile_id* (*int*) – the desired profile identifier

Return type *Employer*

get_employers_list (*country_id=None, city_id=None, login=None, pages=1*)
Get filtered employer profiles.

Parameters

- **country_id** (*Optional[int]*) – employer from country (API-related Country identifier), defaults to *None*
- **city_id** (*Optional[int]*) – employer from city (API-related City identifier), defaults to *None*
- **login** (*Optional[str]*) – with the desired login, defaults to *None*
- **pages** (*Union[int, Tuple[int], List[int], None]*) – number of pages, defaults to *1*

Return type `List[Employer]`

Returns list of filtered employer profiles

get_freelancer_details (*profile_id*)

Get information about freelancer by identifier.

Parameters **profile_id** (*int*) – the desired profile identifier

Return type *Freelancer*

get_freelancers_list (*country_id=None, city_id=None, skill_id=None, login=None, pages=1*)

Get filtered freelancer profiles.

Parameters

- **country_id** (*Optional[int]*) – freelancer from country (API-related Country identifier), defaults to None
- **city_id** (*Optional[int]*) – freelancer from city (API-related City identifier), defaults to None
- **skill_id** (*Optional[int]*) – freelancer skill (API-related Skill identifier), defaults to None
- **login** (*Optional[str]*) – with the desired login, defaults to None
- **pages** (*Union[int, Tuple[int], List[int], None]*) – number of pages, defaults to 1

Return type *List[Freelancer]*

Returns list of filtered freelancer profiles

property my_profile

Get my profile information.

Return type *Union[Employer, Freelancer]*

Returns information of your account

freelancehunt.packages.projects module

Freelancehunt Documentation - Projects API.

class `freelancehunt.packages.projects.Projects` (*token=None, **kwargs*)

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Projects API part.

Warning: For directly usage please set *token* argument.

Parameters **token** (*str*) – your API token, optional

create_project (*information*)

Create new project on site.

TBD: Implement a convenient way to create a project.

Parameters **information** (*dict*) – required and optional params of new project (link: <https://apidocs.freelancehunt.com/?version=latest#ff11ae15-05af-4ee8-ae7c-155cd137506f>)

Return type *Project*

Returns representation of created project.

get_list (*pages=1, only_for_plus=False, skills=None, employer_id=None*)

Get projects with filter and from multiple pages.

Parameters

- **skills** (Union[int, str, *Skill*, List[*Skill*], List[int], Tuple[*Skill*], Tuple[int], None]) – filter by skills
- **employer_id** (Optional[int]) – projects from employer with id
- **only_for_plus** (bool) – filter only for plus if False, get otherwise, defaults is False
- **pages** (Union[int, Tuple[int], List[int]]) – number of pages to get, defaults - 1

Return type List[*Project*]

get_project (*project_id*)

Get specific project by id.

Parameters **project_id** (int) – id of the desired project.

Return type *Project*

Returns the desired project object.

my_projects (*pages=1*)

Get my projects list (10 objects).

Parameters **pages** (Union[int, Tuple[int], List[int]]) – number of pages, defaults to 1

Raises **BadRequest** – raises when you are not Employer.

Return type List[*Project*]

freelancehunt.packages.reviews module

Freelancehunt Documentation - Reviews API.

class freelancehunt.packages.reviews.**Reviews** (*token=None, **kwargs*)

Bases: freelancehunt.core.FreelancehuntObject

Provide operations with Reviews API part.

Warning: For directly usage please set *token* argument.

Parameters **token** (*str*) – your API token, optional

get_my_reviews ()

Get reviews of my profile.

Return type List[Type[*Review*]]

Returns profile reviews

get_reviews (*profile_type, profile_id*)

Get reviews of the desired profile.

Parameters

- **profile_type** (*str*) – type of the desired profile, can be “freelancer” or “employer”
- **profile_id** (*int*) – identifier of the desired profile

Return type `List[Type[Review]]`

Returns profile reviews

freelancehunt.packages.skills module

Freelancehunt Documentation - Skills API.

class `freelancehunt.packages.skills.Skills` (*token=None, **kwargs*)

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Skills API part.

Note: This module contains static content. It may be *update()*, but loaded info does not change on the API side.

Warning: For directly usage please set *token* argument.

Parameters `token` (*str*) – your API token, optional

find (*text*)

Find the names of the skill that contain the desired text.

Parameters `text` (*str*) – the desired text that need to be in an skill name.

Raises `ValueError` – Skill not found.

Return type `List[Skill]`

Returns list of skills with an text in name.

get (*skill_id*)

Get the desired skill by API-related identifier.

Parameters `skill_id` (*int*) – identifier of the desired skill.

Return type `Skill`

Returns the desired skill

property list

Get list of all skills.

Return type `List[Skill]`

Returns list of skills

update ()

Update static information from API.

freelancehunt.packages.threads module

Freelancehunt Documentation - Threads API.

class `freelancehunt.packages.threads.Threads` (*token=None, **kwargs*)

Bases: `freelancehunt.core.FreelancehuntObject`

Provide operations with Threads API part.

Warning: For directly usage please set *token* argument.

Parameters `token` (*str*) – your API token, optional

create_thread (*to_profile_id, subject, message_html*)

Create new thread.

Parameters

- **to_profile_id** (*int*) – recipient profile id
- **subject** (*str*) – thread subject
- **message_html** (*str*) – the first thread's message

Return type `Thread`

Returns created thread

get_threads (*pages=1*)

Get list of threads.

Parameters `Tuple[int], List[int]` **pages** (*Union[int,]*) – count of pages to get, defaults to 1

Return type `List[Thread]`

1.5.4 freelancehunt.utils package

freelancehunt.utils.errors module

All custom errors.

exception `freelancehunt.utils.errors.APIRespondingError`

Bases: `freelancehunt.utils.errors.FreelancehuntError`

FreelanceHunt API is not responding now.

exception `freelancehunt.utils.errors.AuthenticationError`

Bases: `freelancehunt.utils.errors.FreelancehuntError`

Authentication error raised by API server.

exception `freelancehunt.utils.errors.NotEmployerError`

Bases: `freelancehunt.utils.errors.FreelancehuntError`

Client are not employer.

exception `freelancehunt.utils.errors.UnexpectedError`

Bases: `freelancehunt.utils.errors.FreelancehuntError`

Error not recognized by framework.

exception `freelancehunt.utils.errors.ValidationError`
Bases: `freelancehunt.utils.errors.FreelancehuntError`
Some fields in POST data is not validated by server.

freelancehunt.utils.requester module

Requests singleton.

class `freelancehunt.utils.requester.Requester` (*token, language='en', **kwargs*)
Bases: `object`
Provides requests to API. Singleton.

classmethod `get_requester` (*token=None, **kwargs*)

limit = `None`

request (*request_type, url, filters=None, payload=None*)
Make request to API and handle results.
Args: `request_type` (str): "POST", "GET", "PATCH" or "DEL".
Return: dict: JSON response data in dict

request_date = `None`

token = `None`

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

f

- `freelancehunt`, 3
- `freelancehunt.client`, 2
- `freelancehunt.models.bid`, 11
- `freelancehunt.models.budget`, 13
- `freelancehunt.models.city`, 14
- `freelancehunt.models.contest`, 14
- `freelancehunt.models.country`, 16
- `freelancehunt.models.feed`, 17
- `freelancehunt.models.project`, 19
- `freelancehunt.models.review`, 22
- `freelancehunt.models.skill`, 23
- `freelancehunt.models.thread`, 24
- `freelancehunt.models.threadmessage`, 26
- `freelancehunt.models.user`, 27
- `freelancehunt.packages.bids`, 31
- `freelancehunt.packages.cities`, 33
- `freelancehunt.packages.contests`, 34
- `freelancehunt.packages.countries`, 34
- `freelancehunt.packages.feed`, 35
- `freelancehunt.packages.profiles`, 36
- `freelancehunt.packages.projects`, 37
- `freelancehunt.packages.reviews`, 38
- `freelancehunt.packages.skills`, 39
- `freelancehunt.packages.threads`, 40
- `freelancehunt.utils.errors`, 40
- `freelancehunt.utils.requester`, 41

A

`active_bids()` (*freelancehunt.models.project.Project* property), 21
`answer()` (*freelancehunt.models.thread.Thread* method), 25
`answer()` (*freelancehunt.models.threadmessage.ThreadMessage* method), 27
`APIRespondingError`, 40
`AuthenticationError`, 40

B

`Bid` (class in *freelancehunt.models.bid*), 12
`Bids` (class in *freelancehunt.packages.bids*), 31
`bids()` (*freelancehunt.models.project.Project* property), 21
`BudgetInfo` (class in *freelancehunt.models.budget*), 13

C

`choose()` (*freelancehunt.models.bid.Bid* method), 12
`choose()` (*freelancehunt.packages.bids.Bids* method), 31
`Cities` (class in *freelancehunt*), 3
`Cities` (class in *freelancehunt.packages.cities*), 33
`cities()` (*freelancehunt.Countries* property), 4
`cities()` (*freelancehunt.packages.countries.Countries* property), 34
`City` (class in *freelancehunt.models.city*), 14
`city()` (*freelancehunt.models.user.Profile* property), 30
`close()` (*freelancehunt.models.project.Project* method), 21
`Contest` (class in *freelancehunt.models.contest*), 15
`contest()` (*freelancehunt.models.feed.FeedMessage* property), 18
`Contests` (class in *freelancehunt*), 4
`Contests` (class in *freelancehunt.packages.contests*), 34
`contests()` (*freelancehunt.client.FreelanceHuntClient* property), 2
`contests()` (*freelancehunt.Feed* property), 5

`contests()` (*freelancehunt.FreelanceHuntClient* property), 6
`contests()` (*freelancehunt.packages.feed.Feed* property), 35
`Countries` (class in *freelancehunt*), 4
`Countries` (class in *freelancehunt.packages.countries*), 34
`countries()` (*freelancehunt.client.FreelanceHuntClient* property), 2
`countries()` (*freelancehunt.FreelanceHuntClient* property), 6
`Country` (class in *freelancehunt.models.country*), 16
`country()` (*freelancehunt.models.user.Profile* property), 30
`create_project()` (*freelancehunt.packages.projects.Projects* method), 37
`create_project()` (*freelancehunt.Projects* method), 8
`create_thread()` (*freelancehunt.packages.threads.Threads* method), 40
`create_thread()` (*freelancehunt.Threads* method), 10

D

`de_json()` (*freelancehunt.models.bid.Bid* class method), 12
`de_json()` (*freelancehunt.models.budget.BudgetInfo* class method), 13
`de_json()` (*freelancehunt.models.city.City* class method), 14
`de_json()` (*freelancehunt.models.contest.Contest* class method), 16
`de_json()` (*freelancehunt.models.country.Country* class method), 17
`de_json()` (*freelancehunt.models.feed.FeedMessage* class method), 18
`de_json()` (*freelancehunt.models.project.Project* class method), 21
`de_json()` (*freelancehunt.models.review.Review* class

method), 23
 de_json() (freelancehunt.models.skill.Skill class method), 24
 de_json() (freelancehunt.models.thread.Thread class method), 25
 de_json() (freelancehunt.models.threadmessage.ThreadMessage class method), 27
 de_json() (freelancehunt.models.user.Profile class method), 30

E

Employer (class in freelancehunt.models.user), 29
 extend() (freelancehunt.models.project.Project method), 21

F

Feed (class in freelancehunt), 5
 Feed (class in freelancehunt.packages.feed), 35
 feed() (freelancehunt.client.FreelanceHuntClient property), 2
 feed() (freelancehunt.FreelanceHuntClient property), 6
 FeedMessage (class in freelancehunt.models.feed), 17
 find() (freelancehunt.Cities method), 3
 find() (freelancehunt.Countries method), 5
 find() (freelancehunt.packages.cities.Cities method), 33
 find() (freelancehunt.packages.countries.Countries method), 35
 find() (freelancehunt.packages.skills.Skills method), 39
 find() (freelancehunt.Skills method), 10
 freelancehunt module, 3
 freelancehunt.client module, 2
 freelancehunt.models.bid module, 11
 freelancehunt.models.budget module, 13
 freelancehunt.models.city module, 14
 freelancehunt.models.contest module, 14
 freelancehunt.models.country module, 16
 freelancehunt.models.feed module, 17
 freelancehunt.models.project module, 19
 freelancehunt.models.review module, 22
 freelancehunt.models.skill

module, 23
 freelancehunt.models.thread module, 24
 freelancehunt.models.threadmessage module, 26
 freelancehunt.models.user module, 27
 freelancehunt.packages.bids module, 31
 freelancehunt.packages.cities module, 33
 freelancehunt.packages.contests module, 34
 freelancehunt.packages.countries module, 34
 freelancehunt.packages.feed module, 35
 freelancehunt.packages.profiles module, 36
 freelancehunt.packages.projects module, 37
 freelancehunt.packages.reviews module, 38
 freelancehunt.packages.skills module, 39
 freelancehunt.packages.threads module, 40
 freelancehunt.utils.errors module, 40
 freelancehunt.utils.requester module, 41
 FreelanceHuntClient (class in freelancehunt), 6
 FreelanceHuntClient (class in freelancehunt.client), 2
 Freelancer (class in freelancehunt.models.user), 29
 full_name() (freelancehunt.models.user.Profile property), 31

G

get() (freelancehunt.Cities method), 4
 get() (freelancehunt.Countries method), 5
 get() (freelancehunt.packages.cities.Cities method), 33
 get() (freelancehunt.packages.countries.Countries method), 35
 get() (freelancehunt.packages.skills.Skills method), 39
 get() (freelancehunt.Skills method), 10
 get_bids() (freelancehunt.models.project.Project method), 21
 get_employer_details() (freelancehunt.packages.profiles.Profiles method), 36
 get_employer_details() (freelancehunt.Profiles method), 7

<code>get_employers_list()</code>	(<i>freelancehunt.packages.profiles.Profiles</i> method), 36	<code>is_project()</code>	(<i>freelancehunt.models.feed.FeedMessage</i> property), 18
<code>get_employers_list()</code>	(<i>freelancehunt.Profiles</i> method), 7	<code>is_rub()</code>	(<i>freelancehunt.models.budget.BudgetInfo</i> property), 13
<code>get_freelancer_datails()</code>	(<i>freelancehunt.packages.profiles.Profiles</i> method), 36	<code>is_token_valid()</code>	(<i>freelancehunt.client.FreelanceHuntClient</i> property), 3
<code>get_freelancer_datails()</code>	(<i>freelancehunt.Profiles</i> method), 7	<code>is_token_valid()</code>	(<i>freelancehunt.FreelanceHuntClient</i> property), 6
<code>get_freelancers_list()</code>	(<i>freelancehunt.packages.profiles.Profiles</i> method), 37	<code>is_uah()</code>	(<i>freelancehunt.models.budget.BudgetInfo</i> property), 14
<code>get_freelancers_list()</code>	(<i>freelancehunt.Profiles</i> method), 7	L	
<code>get_list()</code>	(<i>freelancehunt.packages.projects.Projects</i> method), 37	<code>left_time_limit_update()</code>	(<i>freelancehunt.client.FreelanceHuntClient</i> property), 3
<code>get_list()</code>	(<i>freelancehunt.Projects</i> method), 8	<code>left_time_limit_update()</code>	(<i>freelancehunt.FreelanceHuntClient</i> property), 6
<code>get_messages()</code>	(<i>freelancehunt.models.thread.Thread</i> method), 25	<code>limit</code>	(<i>freelancehunt.Requester</i> attribute), 9
<code>get_my_bids()</code>	(<i>freelancehunt.packages.bids.Bids</i> method), 32	<code>limit</code>	(<i>freelancehunt.utils.requester.Requester</i> attribute), 41
<code>get_my_reviews()</code>	(<i>freelancehunt.packages.reviews.Reviews</i> method), 38	<code>link()</code>	(<i>freelancehunt.models.project.Project</i> property), 21
<code>get_my_reviews()</code>	(<i>freelancehunt.Reviews</i> method), 9	<code>list()</code>	(<i>freelancehunt.Cities</i> property), 4
<code>get_new()</code>	(<i>freelancehunt.Feed</i> method), 5	<code>list()</code>	(<i>freelancehunt.Countries</i> property), 5
<code>get_new()</code>	(<i>freelancehunt.packages.feed.Feed</i> method), 36	<code>list()</code>	(<i>freelancehunt.packages.cities.Cities</i> property), 34
<code>get_project()</code>	(<i>freelancehunt.packages.projects.Projects</i> method), 38	<code>list()</code>	(<i>freelancehunt.packages.countries.Countries</i> property), 35
<code>get_project()</code>	(<i>freelancehunt.Projects</i> method), 8	<code>list()</code>	(<i>freelancehunt.packages.feed.Feed</i> property), 36
<code>get_project_bids()</code>	(<i>freelancehunt.packages.bids.Bids</i> method), 32	<code>list()</code>	(<i>freelancehunt.packages.skills.Skills</i> property), 39
<code>get_requester()</code>	(<i>freelancehunt.Requester</i> class method), 9	<code>list()</code>	(<i>freelancehunt.Skills</i> property), 10
<code>get_requester()</code>	(<i>freelancehunt.utils.requester.Requester</i> class method), 41	<code>load_details()</code>	(<i>freelancehunt.models.contest.Contest</i> method), 16
<code>get_reviews()</code>	(<i>freelancehunt.packages.reviews.Reviews</i> method), 38	<code>load_details()</code>	(<i>freelancehunt.models.project.Project</i> method), 21
<code>get_reviews()</code>	(<i>freelancehunt.Reviews</i> method), 9	<code>load_details()</code>	(<i>freelancehunt.models.user.Profile</i> method), 31
<code>get_threads()</code>	(<i>freelancehunt.packages.threads.Threads</i> method), 40	M	
<code>get_threads()</code>	(<i>freelancehunt.Threads</i> method), 11	module	
I		<code>freelancehunt</code> , 3	
<code>is_contest()</code>	(<i>freelancehunt.models.feed.FeedMessage</i> property), 18	<code>freelancehunt.client</code> , 2	
		<code>freelancehunt.models.bid</code> , 11	
		<code>freelancehunt.models.budget</code> , 13	
		<code>freelancehunt.models.city</code> , 14	
		<code>freelancehunt.models.contest</code> , 14	
		<code>freelancehunt.models.country</code> , 16	
		<code>freelancehunt.models.feed</code> , 17	
		<code>freelancehunt.models.project</code> , 19	
		<code>freelancehunt.models.review</code> , 22	

freelancehunt.models.skill, 23
 freelancehunt.models.thread, 24
 freelancehunt.models.threadmessage, 26
 freelancehunt.models.user, 27
 freelancehunt.packages.bids, 31
 freelancehunt.packages.cities, 33
 freelancehunt.packages.contests, 34
 freelancehunt.packages.countries, 34
 freelancehunt.packages.feed, 35
 freelancehunt.packages.profiles, 36
 freelancehunt.packages.projects, 37
 freelancehunt.packages.reviews, 38
 freelancehunt.packages.skills, 39
 freelancehunt.packages.threads, 40
 freelancehunt.utils.errors, 40
 freelancehunt.utils.requester, 41
 my_active_bids() (freelancehunt.packages.bids.Bids property), 32
 my_profile() (freelancehunt.packages.profiles.Profiles property), 37
 my_profile() (freelancehunt.Profiles property), 8
 my_projects() (freelancehunt.packages.projects.Projects method), 38
 my_projects() (freelancehunt.Projects method), 8

N

NotEmployerError, 40

P

Profile (class in freelancehunt.models.user), 29
 profile_url() (freelancehunt.models.user.Profile property), 31
 Profiles (class in freelancehunt), 7
 Profiles (class in freelancehunt.packages.profiles), 36
 profiles() (freelancehunt.client.FreelanceHuntClient property), 3
 profiles() (freelancehunt.FreelanceHuntClient property), 6
 Project (class in freelancehunt.models.project), 20
 project() (freelancehunt.models.feed.FeedMessage property), 18
 Projects (class in freelancehunt), 8
 Projects (class in freelancehunt.packages.projects), 37
 projects() (freelancehunt.client.FreelanceHuntClient property), 3
 projects() (freelancehunt.Feed property), 6
 projects() (freelancehunt.FreelanceHuntClient property), 6

projects() (freelancehunt.packages.feed.Feed property), 36

R

read() (freelancehunt.Feed method), 6
 read() (freelancehunt.packages.feed.Feed method), 36
 reject() (freelancehunt.models.bid.Bid method), 12
 reject() (freelancehunt.packages.bids.Bids method), 32
 remaining_limit() (freelancehunt.client.FreelanceHuntClient property), 3
 remaining_limit() (freelancehunt.FreelanceHuntClient property), 6
 reopen() (freelancehunt.models.project.Project method), 21
 request() (freelancehunt.Requester method), 9
 request() (freelancehunt.utils.requester.Requester method), 41
 request_date (freelancehunt.Requester attribute), 9
 request_date (freelancehunt.utils.requester.Requester attribute), 41
 Requester (class in freelancehunt), 9
 Requester (class in freelancehunt.utils.requester), 41
 restore() (freelancehunt.models.bid.Bid method), 13
 restore_bid() (freelancehunt.packages.bids.Bids method), 32
 Review (class in freelancehunt.models.review), 23
 Reviews (class in freelancehunt), 9
 Reviews (class in freelancehunt.packages.reviews), 38
 reviews() (freelancehunt.models.user.Profile property), 31
 revoke() (freelancehunt.models.bid.Bid method), 13
 revoke_bid() (freelancehunt.packages.bids.Bids method), 33

S

sender() (freelancehunt.models.feed.FeedMessage property), 18
 Skill (class in freelancehunt.models.skill), 23
 Skills (class in freelancehunt), 9
 Skills (class in freelancehunt.packages.skills), 39
 skills() (freelancehunt.client.FreelanceHuntClient property), 3
 skills() (freelancehunt.FreelanceHuntClient property), 6
 status() (freelancehunt.models.contest.Contest property), 16
 status() (freelancehunt.models.project.Project property), 21
 status() (freelancehunt.models.user.Profile property), 31

status_code() (freelance-hunt.models.contest.Contest property), 16
 status_code() (freelance-hunt.models.project.Project property), 22
 status_code() (freelancehunt.models.user.Profile property), 31

T

Thread (class in freelancehunt.models.thread), 25
 ThreadMessage (class in freelance-hunt.models.threadmessage), 27
 Threads (class in freelancehunt), 10
 Threads (class in freelancehunt.packages.threads), 40
 threads() (freelancehunt.client.FreelanceHuntClient property), 3
 threads() (freelancehunt.FreelanceHuntClient property), 7
 token (freelancehunt.Requester attribute), 9
 token (freelancehunt.utils.requester.Requester attribute), 41
 type() (freelancehunt.models.feed.FeedMessage property), 18

U

UnexpectedError, 40
 update() (freelancehunt.Cities method), 4
 update() (freelancehunt.Countries method), 5
 update() (freelancehunt.Feed method), 6
 update() (freelancehunt.packages.cities.Cities method), 34
 update() (freelancehunt.packages.countries.Countries method), 35
 update() (freelancehunt.packages.feed.Feed method), 36
 update() (freelancehunt.packages.skills.Skills method), 39
 update() (freelancehunt.Skills method), 10

V

ValidationError, 40

W

winner_bid() (freelancehunt.models.project.Project property), 22