

---

# **petfinder-api Documentation**

***Release 0.1***

**Greg Taylor**

**Jun 01, 2017**



---

## Contents

---

|          |                                                   |           |
|----------|---------------------------------------------------|-----------|
| <b>1</b> | <b>Assorted Info</b>                              | <b>3</b>  |
| <b>2</b> | <b>User Guide</b>                                 | <b>5</b>  |
| 2.1      | Installation . . . . .                            | 5         |
| 2.1.1    | Distribute & Pip . . . . .                        | 5         |
| 2.1.2    | Get the source . . . . .                          | 5         |
| 2.2      | Quickstart . . . . .                              | 6         |
| 2.2.1    | Petfinder.com credentials . . . . .               | 6         |
| 2.2.2    | Instantiate the API client . . . . .              | 6         |
| 2.2.3    | How to determine kwargs for API methods . . . . . | 6         |
| 2.2.4    | Many calls return generators . . . . .            | 7         |
| 2.2.5    | Auto-pagination of results . . . . .              | 7         |
| 2.3      | API Reference . . . . .                           | 7         |
| 2.3.1    | petfinder.PetfinderClient . . . . .               | 7         |
| 2.3.2    | petfinder.exceptions . . . . .                    | 9         |
| <b>3</b> | <b>Indices and tables</b>                         | <b>11</b> |
|          | <b>Python Module Index</b>                        | <b>13</b> |



petfinder-api is a BSD licensed [Python](#) wrapper around the [Petfinder API](#). The module handles preparing and sending API requests, parsing the response, and returning Python objects for usage in your application.

```
import petfinder

# Instantiate the client with your credentials.
api = petfinder.PetFinderClient(api_key='yourkey', api_secret='yoursecret')

# Query away!
for shelter in api.shelter_find(location='30127', count=500):
    print(shelter['name'])

# Search for pets.
for pet in api.pet_find(
    animal="dog", location="29678", output="basic",
    breed="Treeing Walker Coonhound", count=200,
):
    print("%s - %s" % (pet['id'], pet['name']))

# TODO: Find homes for these guys.
```



# CHAPTER 1

---

## Assorted Info

---

- [Issue tracker](#) - Report bugs here.
- [GitHub project](#) - Source code lives here.
- [@gctaylor Twitter](#) - Tweets from the maintainer.



The user guide covers topics such as installation and general use of the module. Content is largely step-by-step instructions for getting started with, and using petfinder-api.

## Installation

This part of the documentation covers the installation process. There are a few different ways to retrieve and install petfinder-api.

### Distribute & Pip

Installing petfinder-api is simple with `pip`:

```
pip install petfinder
```

`easy_install` works, too:

```
easy_install petfinder
```

### Get the source

petfinder-api is developed on [GitHub](#) in the petfinder-api project.

You can either clone the repository:

```
git clone git://github.com/gtaylor/petfinder-api.git
```

Download the tarball:

```
curl -OL https://github.com/gtaylor/petfinder-api/tarball/master
```

Or download the zip:

```
curl -OL https://github.com/gtaylor/petfinder-api/zipball/master
```

## Quickstart

This section goes over how to get up and running quickly. We'll assume that you have already followed the [Installation](#) instructions, and are ready to go.

### Petfinder.com credentials

Before you can make your first query to Petfinder, you'll need to obtain a set of API credentials. You can get them from the [Petfinder API](#) page. You'll need to register first.

Jot down your *API Key* and *API Secret* for use in the next step.

### Instantiate the API client

Next, you'll want to import the module:

```
>>> import petfinder
```

Then instantiate the API client with your Petfinder.com API credentials:

```
>>> api = petfinder.PetFinderClient(api_key='yourkey', api_secret='yoursecret')
```

You are now ready to query against the Petfinder API:

```
>>> breeds = api.breed_list(animal="dog")
```

See [API Reference](#) for a full list of methods. We support all of the query methods on the Petfinder API. Please do continue reading the next few sections in this quickstart for some important tidbits.

### How to determine kwargs for API methods

petfinder-api is a very thin wrapper around the Petfinder API. We point developers to the [Petfinder API docs](#) for all of the required/optional keywords, instead of duplicating it all here.

For example, look at the `pet.find` method on the [Petfinder API docs](#). This maps to our `petfinder.PetFinderClient.pet_find()` method. You can ignore any of the `key` arguments, since petfinder-api sets that for you. The only required parameter for `pet.find` is `location`. Here's a minimal example:

```
for pet in api.pet_find(location="29678"):  
    # This will be a pet record in the form of a dict.  
    print(pet)
```

To outline a process for this, first determine which method in [API Reference](#) you'll want to use:

1. Reference the beginning of the docstring for which Petfinder API call it wraps. For example, `petfinder.PetFinderClient.pet_find()` wraps `pet.find`.
2. Refer to the [Petfinder API docs](#) for the required kwargs.

## Many calls return generators

An extremely important thing to note is that many of the calls return Python generators. This is the most efficient way for us to paginate through multiple pages of results (behind the scenes). Make sure to note the return types for whatever API client method you're calling. More details can be found in [API Reference](#).

## Auto-pagination of results

Another thing to keep in mind is that the petfinder-api client auto-paginates through results. The `count` parameter in the Petfinder API determines how many results are returned per “page”. petfinder-api will happily continue to make additional API requests if you continue to iterate over the generators it returns. This will often continue until you see the infamous `petfinder.exceptions.LimitExceeded` exception, which you will hit on pretty much any `*_find` call, and some `*_list` calls.

The important thing to consider is that a higher `count` will result in larger, but fewer HTTP requests to the Petfinder API, whereas smaller values will result in more frequent, but smaller requests from the API.

## API Reference

This page serves as a complete reference to all public classes and exceptions. This is probably only useful after you have read [Quickstart](#).

### petfinder.PetfinderClient

`class petfinder.PetFinderClient (api_key, api_secret, endpoint='http://api.petfinder.com/')`

Simple client for the Petfinder API. You'll want to pull your API details from <http://www.petfinder.com/developers/api-key> and instantiate this class with said credentials.

Refer to <http://www.petfinder.com/developers/api-docs> for the required kwargs for each method. It is safe to ignore the `key` argument, as this client handles setting that for you.

#### breed\_list

`PetFinderClient.breed_list (**kwargs)`

breed.list wrapper. Returns a list of breed name strings.

**Return type** list

**Returns** A list of breed names.

#### pet\_get

`PetFinderClient.pet_get (**kwargs)`

pet.get wrapper. Returns a record dict for the requested pet.

**Return type** dict

**Returns** The pet's record dict.

## pet\_getrandom

`PetFinderClient.pet_getrandom(**kwargs)`

`pet.getRandom` wrapper. Returns a record dict or Petfinder ID for a random pet.

**Return type** dict or str

**Returns** A dict of pet data if output is 'basic' or 'full', and a string if output is 'id'.

## pet\_find

`PetFinderClient.pet_find(**kwargs)`

`pet.find` wrapper. Returns a generator of pet record dicts matching your search criteria.

**Return type** generator

**Returns** A generator of pet record dicts.

**Raises** `petfinder.exceptions.LimitExceeded` once you have reached the maximum number of records your credentials allow you to receive.

## shelter\_find

`PetFinderClient.shelter_find(**kwargs)`

`shelter.find` wrapper. Returns a generator of shelter record dicts matching your search criteria.

**Return type** generator

**Returns** A generator of shelter record dicts.

**Raises** `petfinder.exceptions.LimitExceeded` once you have reached the maximum number of records your credentials allow you to receive.

## shelter\_get

`PetFinderClient.shelter_get(**kwargs)`

`shelter.get` wrapper. Given a shelter ID, retrieve its details in dict form.

**Return type** dict

**Returns** The shelter's details.

## shelter\_getpets

`PetFinderClient.shelter_getpets(**kwargs)`

`shelter.getPets` wrapper. Given a shelter ID, retrieve either a list of pet IDs (if output is 'id'), or a generator of pet record dicts (if output is 'full' or 'basic').

**Return type** generator

**Returns** Either a generator of pet ID strings or pet record dicts, depending on the value of the output keyword.

**Raises** `petfinder.exceptions.LimitExceeded` once you have reached the maximum number of records your credentials allow you to receive.

## shelter\_listbybreed

`PetFinderClient.shelter_listbybreed(**kwargs)`

shelter.listByBreed wrapper. Given a breed and an animal type, list the shelter IDs with pets of said breed.

**Return type** generator

**Returns** A generator of shelter IDs that have breed matches.

## petfinder.exceptions

Petfinder API exceptions.

**exception** `petfinder.exceptions.AuthenticationFailure`

Raised when an authentication failure occurs.

**exception** `petfinder.exceptions.GenericInternalError`

Your guess is as good as mine.

**exception** `petfinder.exceptions.InvalidGeographicalLocationError`

Raised when an invalid geographical location is specified.

**exception** `petfinder.exceptions.InvalidRequestError`

Raised when an invalid or mal-formed request is sent.

**exception** `petfinder.exceptions.LimitExceeded`

Raised when usage limits are exceeded.

**exception** `petfinder.exceptions.PetfinderAPIError`

Base class for all Petfinder API exceptions. Mostly here to allow end users to catch all Petfinder exceptions.

**exception** `petfinder.exceptions.RecordDoesNotExistError`

Raised when querying for a record that does not exist.

**exception** `petfinder.exceptions.RequestIsUnauthorizedError`

Raised when attempting to call a method that the user is unauthorized for.

**exception** `petfinder.exceptions.UnrecognizedError`

This is raised if the API returns a status code that we don't have a matching exception for. This is more for future-proofing, in case they add extra status codes.



## CHAPTER 3

---

### Indices and tables

---

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



**p**

`petfinder.exceptions`, [9](#)



## A

AuthenticationFailure, 9

## B

breed\_list() (petfinder.PetFinderClient method), 7

## G

GenericInternalError, 9

## I

InvalidGeographicalLocationError, 9

InvalidRequestError, 9

## L

LimitExceeded, 9

## P

pet\_find() (petfinder.PetFinderClient method), 8

pet\_get() (petfinder.PetFinderClient method), 7

pet\_getrandom() (petfinder.PetFinderClient method), 8

petfinder.exceptions (module), 9

PetfinderAPIError, 9

PetFinderClient (class in petfinder), 7

## R

RecordDoesNotExistError, 9

RequestIsUnauthorizedError, 9

## S

shelter\_find() (petfinder.PetFinderClient method), 8

shelter\_get() (petfinder.PetFinderClient method), 8

shelter\_getpets() (petfinder.PetFinderClient method), 8

shelter\_listbybreed() (petfinder.PetFinderClient method),  
9

## U

UnrecognizedError, 9