

# Package ‘PetfinderR’

May 22, 2018

**Type** Package

**Title** 'Petfinder' API Wrapper

**Version** 1.1.3

**Author** Aaron Schlegel

**Maintainer** Aaron Schlegel <aaron@aaronschlegel.com>

## Description

Wrapper of the 'Petfinder API' <<https://www.petfinder.com/developers/api-docs>> that implements methods for interacting with and extracting data from the 'Petfinder' database. The 'Petfinder REST API' allows access to the 'Petfinder' database, one of the largest online, searchable databases of adoptable animals and animal welfare organizations across North America.

**Encoding** UTF-8

**LazyData** true

**Imports** dplyr, httr, jsonlite, plyr

**RoxygenNote** 6.0.1

**License** MIT + file LICENSE

**URL** <https://github.com/aschleg/PetfinderR>

**VignetteBuilder** knitr

**Suggests** R6, httpptest, testthat, knitr, rmarkdown

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-05-22 12:23:55 UTC

## R topics documented:

breed.list . . . . .	2
pet.find . . . . .	3
pet.get . . . . .	4
pet.getRandom . . . . .	5
Petfinder . . . . .	6
shelter.find . . . . .	6

shelter.get . . . . .	7
shelter.getPets . . . . .	8
shelter.listByBreed . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

breed.list	<i>Method for calling the 'breed.list' method of the Petfinder API. Returns the available breeds for the selected animal.</i>
------------	---

---

### Description

Method for calling the 'breed.list' method of the Petfinder API. Returns the available breeds for the selected animal.

### Usage

```
breed.list(animal, return_df = TRUE)
```

### Arguments

animal	Return breeds of animal. Must be one of 'barnyard', 'bird', 'cat', 'dog', 'horse', 'reptile', or 'smallfurry'.
return_df	If TRUE, the function will coerce the output JSON from the Petfinder API into a data.frame

### Value

List of returned JSON from the Petfinder API. If the parameter return\_df is TRUE, a data.frame is returned instead.

### Examples

```
## Not run:
pf <- Petfinder(key) # Initialize Petfinder class
pf$breed.list('cat')
pf$breed.list('dog')

## End(Not run)
```

---

pet.find *Returns a data.frame of pet records matching input parameters.*

---

### Description

Returns a data.frame of pet records matching input parameters.

### Usage

```
pet.find(location, animal = NULL, breed = NULL, size = NULL, sex = NULL,
         age = NULL, offset = NULL, count = NULL, output = NULL,
         pages = NULL, return_df = FALSE)
```

### Arguments

location	ZIP/postal code, state, or city and state to perform the search.
animal	Return breeds of animal. Must be one of 'barnyard', 'bird', 'cat', 'dog', 'horse', 'reptile', or 'smallfurry'.
breed	Specifies the breed of the animal to search.
size	Specifies the size of the animal/breed to search. Must be one of 'S' (small), 'M' (medium), 'L' (large), 'XL' (extra-large).
sex	Filters the search to the desired gender of the animal. Must be one of 'M' (male) or 'F' (female).
age	Returns animals with specified age. Must be one of 'Baby', 'Young', 'Adult', 'Senior'.
offset	Can be set to the value of lastOffset returned from the previous call to retrieve the next set of results. The pages parameter can also be used to pull a desired number of paged results.
count	The number of records to return. Default is 25.
output	Sets the amount of information returned in each record. 'basic' returns a simple record while 'full' returns a complete record with description. Defaults to 'basic'.
pages	The number of pages of results to return. For example, if pages=4 with the default count parameter (25), 125 results would be returned (25 results from first call and 100 from the next four pages).
return_df	If TRUE, the function will coerce the output JSON from the Petfinder API into a data.frame

### Value

data.frame of pet records matching input parameters.

## Examples

```
## Not run:
pf <- Petfinder(key) # Initialize Petfinder class
pf$pet.find('WA')
pf$pet.find('WA', 'cat', pages = 2)

## End(Not run)
```

---

pet.get

*Returns the available pet record data for the input pet Ids.*

---

## Description

Returns the available pet record data for the input pet Ids.

## Usage

```
pet.get(petId, return_df = FALSE)
```

## Arguments

petId	ID of the pet record to return. Accepts a character for a single record or a list or vector of petIds.
return_df	If TRUE, the function will coerce the output JSON from the Petfinder API into a data.frame

## Value

data.frame of input petId(s).

## Examples

```
## Not run:
pf <- Petfinder(key) # Initialize Petfinder class
pf$pet.get(petId)
petIds <- c(petId1, petId2, petId3)
pf$pet.get(petIds)

## End(Not run)
```

---

pet.getRandom	Returns a randomly selected pet record. The possible result can be filtered with input parameters.
---------------	--

---

### Description

Returns a randomly selected pet record. The possible result can be filtered with input parameters.

### Usage

```
pet.getRandom(records = NULL, animal = NULL, breed = NULL, size = NULL,  
sex = NULL, location = NULL, shelterId = NULL, output = NULL,  
return_df = FALSE)
```

### Arguments

records	Selects the amount of desired random results to be returned. Each returned record is counted as one call to the Petfinder API.
animal	Return breeds of animal. Must be one of 'barnyard', 'bird', 'cat', 'dog', 'horse', 'reptile', or 'smallfurry'.
breed	Specifies the breed of the animal to search.
size	Specifies the size of the animal/breed to search. Must be one of 'S' (small), 'M' (medium), 'L' (large), 'XL' (extra-large).
sex	Filters the search to the desired gender of the animal. Must be one of 'M' (male) or 'F' (female).
location	ZIP/postal code, state, or city and state to perform the search.
shelterId	Filters randomly returned results down to a specific shelter.
output	Sets the amount of information returned in each record. 'basic' returns a simple record while 'full' returns a complete record with description. Defaults to 'basic'.
return_df	If TRUE, the function will coerce the output JSON from the Petfinder API into a data.frame

### Value

data.frame of randomly selected pet record(s).

### Examples

```
## Not run:  
#' pf <- Petfinder(key) # Initialize Petfinder class  
pf$pet.getRandom(animal = 'cat')  
pf$pet.getRandom(10, 'dog')  
  
## End(Not run)
```

---

Petfinder	<i>Creates an authenticated connection with the Petfinder API. The stored authentication is then used to call the Petfinder API methods. An API key can be obtained from Petfinder by creating an account on their developer page (<a href="https://www.petfinder.com/developers/api-key">https://www.petfinder.com/developers/api-key</a>). The function wraps the <code>.Petfinder.class R6</code> class.</i>
-----------	---

---

### Description

Creates an authenticated connection with the Petfinder API. The stored authentication is then used to call the Petfinder API methods. An API key can be obtained from Petfinder by creating an account on their developer page (<https://www.petfinder.com/developers/api-key>). The function wraps the `.Petfinder.class R6` class.

### Usage

```
Petfinder(key, secret = NULL)
```

### Arguments

key	The API key received from Petfinder
secret	The secret key received from Petfinder along with the API key. Only used for methods that require additional authentication (not currently used).

### Value

Intialized Petfinder object that is then used to access the API.

### Examples

```
## Not run:
pf <- Petfinder(key) # Creates the connection with the Petfinder API.
pf$breed.list('cat') # The connection can now be used to access the Petfinder API methods.

## End(Not run)
```

---

shelter.find	<i>Returns a <code>data.frame</code> of shelter records matching input parameters.</i>
--------------	--

---

### Description

Returns a `data.frame` of shelter records matching input parameters.

### Usage

```
shelter.find(location, name = NULL, offset = NULL, count = NULL,
             pages = NULL, return_df = FALSE)
```

**Arguments**

location	ZIP/postal code, state, or city and state to perform the search.
name	Full or partial shelter name
offset	Can be set to the value of lastOffset returned from the previous call to retrieve the next set of results. The pages parameter can also be used to pull a desired number of paged results.
count	The number of records to return. Default is 25.
pages	The number of pages of results to return. For example, if pages=4 with the default count parameter (25), 125 results would be returned (25 results from first call and 100 from the next four pages).
return_df	If TRUE, the function will coerce the output JSON from the Petfinder API into a data.frame

**Value**

List of returned JSON from the Petfinder API. If the parameter return\_df is TRUE, a data.frame is returned instead.

**Examples**

```
## Not run:
pf <- Petfinder(key) # Initialize Petfinder class
pf$shelter.find('WA', count = 5)
pf$shelter.find('WA', pages = 3)

## End(Not run)
```

---

shelter.get	<i>Returns shelter records of input shelterIds.</i>
-------------	---

---

**Description**

Returns shelter records of input shelterIds.

**Usage**

```
shelter.get(shelterId)
```

**Arguments**

shelterId	ID of the shelter record to return. Accepts a character for a single record or a list or vector of shelterIds.
-----------	--

**Value**

data.frame of input shelterId(s).

**Examples**

```
## Not run:
#' pf <- Petfinder(key)
pf$shelter.get(shelterId)
shelterIds <- c(shelterId1, shelterId2, shelterId3)
pf$shelter.get(shelterIds)

## End(Not run)
```

---

shelter.getPets	<i>Returns a collection of pet records for an individual shelter.</i>
-----------------	---

---

**Description**

Returns a collection of pet records for an individual shelter.

**Usage**

```
shelter.getPets(shelterId, status = NULL, offset = NULL, count = NULL,
  output = NULL, pages = NULL, return_df = FALSE)
```

**Arguments**

shelterId	Desired shelter's ID
status	Filters returned collection of pet records by the pet's status. Must be one of 'A' (adoptable, default), 'H' (hold), 'P' (pending), 'X' (adopted/removed).
offset	Can be set to the value of lastOffset returned from the previous call to retrieve the next set of results. The pages parameter can also be used to pull a desired number of paged results.
count	The number of records to return. Default is 25.
output	Sets the amount of information returned in each record. 'basic' returns a simple record while 'full' returns a complete record with description. Defaults to 'basic'.
pages	The number of pages of results to return. For example, if pages=4 with the default count parameter (25), 125 results would be returned (25 results from first call and 100 from the next four pages).
return_df	If TRUE, the function will coerce the output JSON from the Petfinder API into a data.frame

**Value**

data.frame of pet records associated with specified shelterId.



## Examples

```
## Not run:
pf <- Petfinder(key) # Initialize Petfinder class
pf$shelter.getPets('WA40') # Seattle Area Feline Rescue.

## End(Not run)
```

---

`shelter.listByBreed` *Returns a data.frame of shelter IDs listing animals matching the input animal breed.*

---

## Description

Returns a data.frame of shelter IDs listing animals matching the input animal breed.

## Usage

```
shelter.listByBreed(animal, breed, offset = NULL, count = NULL,
  pages = NULL)
```

## Arguments

<code>animal</code>	Return breeds of animal. Must be one of 'barnyard', 'bird', 'cat', 'dog', 'horse', 'reptile', or 'smallfurry'.
<code>breed</code>	Specifies the breed of the animal to search.
<code>offset</code>	Can be set to the value of lastOffset returned from the previous call to retrieve the next set of results. The pages parameter can also be used to pull a desired number of paged results.
<code>count</code>	The number of records to return. Default is 25.
<code>pages</code>	The number of pages of results to return. For example, if pages=4 with the default count parameter (25), 125 results would be returned (25 results from first call and 100 from the next four pages).

## Value

data.frame of pet records associated with specified shelterId.

## Examples

```
## Not run:
pf <- Petfinder(key) # Initialize Petfinder class
pf$shelter.listByBreed('cat', 'Abyssinian')
pf$shelter.listByBreed('dog', 'Golden Retriever')

## End(Not run)
```

# Index

`breed.list`, 2

`pet.find`, 3

`pet.get`, 4

`pet.getRandom`, 5

`Petfinder`, 6

`shelter.find`, 6

`shelter.get`, 7

`shelter.getPets`, 8

`shelter.listByBreed`, 9