

# Package ‘rinat’

October 14, 2020

**Type** Package

**Title** Access 'iNaturalist' Data Through APIs

**Version** 0.1.6

**Date** 2020-10-13

**Description** A programmatic interface to the API provided by the 'iNaturalist' website <<https://www.inaturalist.org/>> to download species occurrence data submitted by citizen scientists.

**License** MIT + file LICENSE

**Imports** httr, plyr, jsonlite, ggplot2, maps, sp

**RoxygenNote** 7.1.1

**URL** <https://docs.ropensci.org/rinat/> (website)

<https://github.com/ropensci/rinat> (devel)

**Suggests** knitr, rmarkdown, testthat, sf

**VignetteBuilder** knitr

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Vijay Barve [aut] (<<https://orcid.org/0000-0002-4852-2567>>),  
Edmund Hart [aut],  
Stéphane Guillou [cre] (<<https://orcid.org/0000-0001-8992-0951>>)

**Maintainer** Stéphane Guillou <[stephane.guillou@member.fsf.org](mailto:stephane.guillou@member.fsf.org)>

**Repository** CRAN

**Date/Publication** 2020-10-14 09:10:03 UTC

## R topics documented:

get_inat_obs . . . . .	2
get_inat_obs_id . . . . .	3
get_inat_obs_project . . . . .	4
get_inat_obs_user . . . . .	5
get_inat_taxon_stats . . . . .	5
get_inat_user_stats . . . . .	6
inat_map . . . . .	7

---

get_inat_obs	<i>Download iNaturalist data</i>
--------------	----------------------------------

---

### Description

Primary function to retrieve observations from iNaturalist, allows users to search for data, or just filter results by a subset of what is offered by the API.

### Usage

```
get_inat_obs(
  query = NULL,
  taxon_name = NULL,
  taxon_id = NULL,
  place_id = NULL,
  quality = NULL,
  geo = NULL,
  year = NULL,
  month = NULL,
  day = NULL,
  bounds = NULL,
  maxresults = 100,
  meta = FALSE
)
```

### Arguments

query	Query string for a general search.
taxon_name	Filter by iNat taxon name. Note that this will also select observations of descendant taxa. Note that names are not unique, so if the name matches multiple taxa, no observations may be returned.
taxon_id	Filter by iNat taxon ID. Note that this will also select observations of descendant taxa.
place_id	Filter by iNat place ID.
quality	The quality grade to be used. Must be either "casual" or "research". If left blank both will be returned.
geo	Flag for returning only results that are georeferenced, TRUE will exclude non-georeferenced results, but they cannot be excluded.
year	Return observations only in that year (can only be one year, not a range of years).
month	Return observations only by month, must be numeric, 1...12
day	Return observations only on a given day of the month, 1...31

bounds	A bounding box of longitude (-180 to 180) and latitude (-90 to 90) to search within. It is a vector in the form of southern latitude, western longitude, northern latitude, and eastern longitude. Alternatively supply an sf or sp object from which the bounding box will be derived.
maxresults	The maximum number of results to return. Should not be a number higher than 10000.
meta	(logical) If TRUE, the output of this function is a list with metadata on the output and a data.frame of the data. If FALSE (default), just the data.frame.

**Value**

A dataframe of the number of observations requested.

**Note**

Filtering doesn't always work with the query parameter for some reason (a problem on the API end). If you want to filter by time, it's best to use the scientific name and put it in the 'taxa' field, and not in the query field. Another issue is that the query parameter will search the entire entry, so it is possible to get unintended results. Depending on your use case it may be advisable to use the "taxon" field instead of the query field.

**Examples**

```
## Not run:
### Make a standard query
get_inat_obs(query = "Monarch Butterfly")

##Filter by a bounding box of Northern California
bounds <- c(38.44047, -125, 40.86652, -121.837)
get_inat_obs(query = "Mule Deer", bounds = bounds)

## Filter with by just taxon, allows higher order filtering,
## Here we can search for just stone flies (order Plecoptera)
get_inat_obs(taxon_name = "Plecoptera")

## get metadata (the number of results found on the server)
out <- get_inat_obs(query = "Monarch Butterfly", meta = TRUE)
out$meta

## End(Not run)
```

---

get\_inat\_obs\_id

*Get information on a specific observation*


---

**Description**

Get information on a specific observation

**Usage**

```
get_inat_obs_id(id)
```

**Arguments**

`id` a single id for an iNaturalist observation record

**Value**

a list with full details on a given record

**Examples**

```
## Not run:
m_obs <- get_inat_obs(query="Monarch Butterfly")
get_inat_obs_id(m_obs$id[1])

## End(Not run)
```

---

`get_inat_obs_project` *Download observations or info from a project*

---

**Description**

retrieve observations from a particular iNaturalist project. This function can be used to get either observations or information from a project by project name or ID

**Usage**

```
get_inat_obs_project(grpid, type = c("observations", "info"), raw = FALSE)
```

**Arguments**

`grpid` Name of the group as an iNaturalist slug or group id

`type` character Either "observations" or "info" Observations returns all observations, and "info" returns project details similar to what you can find on a project webpage.

`raw` logical TRUE or FALSE. If TRUE and searching for project info, returns the raw output of parsed JSON for that project. Otherwise just some basic information is returned as a list

**Details**

An iNaturalist slug is usually the project as single string with words separated by hyphens. For instance, the project "State Flowers of the United States" has a slug of "state-flowers-of-the-united-states-eol-collection". This can be extracted from the URL for the project usually. The state flowers project has the following URL <http://www.inaturalist.org/projects/state-flowers-of-the-united-states-eol-collection>

**Examples**

```
## Not run:
  get_inat_obs_project(354, type = "observations")
  get_inat_obs_project("crows-in-vermont", type="info",raw=FALSE)

## End(Not run)
```

---

get\_inat\_obs\_user      *Download observations for a user*

---

**Description**

Get all the observations of a specific iNaturalist user.

**Usage**

```
get_inat_obs_user(username, maxresults = 100)
```

**Arguments**

username	username of the iNaturalist user to fetch records
maxresults	the maximum number of results to return

**Value**

a list with full details on a given record

**Examples**

```
## Not run:
  m_obs <- get_inat_obs(query="Monarch Butterfly")
  get_inat_obs_user(as.character(m_obs$user_login[1]))

## End(Not run)
```

---

get\_inat\_taxon\_stats      *Get stats on taxon counts*

---

**Description**

Get stats on taxa within a constrained range. This range can be by user, place, project, day or date range. Output will be a count of the total number of taxa observed at each taxonomic level.

**Usage**

```
get_inat_taxon_stats(
  date = NULL,
  date_range = NULL,
  place = NULL,
  project = NULL,
  uid = NULL
)
```

**Arguments**

date	retrieve observations on a specific date, must be a string in the form YYYY-MM-DD
date_range	a vector of dates, in the form YYYY-MM-DD
place	get taxon stats by place, you can find place id's on the iNaturalist page: <a href="https://www.inaturalist.org/places">https://www.inaturalist.org/places</a> , must be a numeric ID
project	get taxon stats by project id (numeric ID)
uid	get taxon stats by user id (string)

**Value**

a vector listing counts of observations at each level of identification possible (species, genus, etc.)

**Examples**

```
## Not run:
counts <- get_inat_taxon_stats(date = "2010-06-14")

## End(Not run)
```

---

```
get_inat_user_stats Get stats on users
```

---

**Description**

Get stats on which users reported the most species or had the most observations within a given range. This range can be by user, place, project, day or date range. Output will be a count of the total number of taxa observed at each taxonomic level.

**Usage**

```
get_inat_user_stats(
  date = NULL,
  date_range = NULL,
  place = NULL,
  project = NULL,
  uid = NULL
)
```

**Arguments**

date	retrieve observations on a specific date, must be a string in the form YYYY-MM-DD
date_range	a vector of dates, in the form YYYY-MM-DD
place	get taxon stats by place, you can find place id's on the iNaturalist page: <a href="https://www.inaturalist.org/places">https://www.inaturalist.org/places</a> , must be a numeric ID
project	get taxon stats by project id
uid	get taxon stats by user id

**Value**

a list with two data frames with of the 5 users with the most observations and the most species

**Examples**

```
## Not run:
counts <- get_inat_user_stats(date = "2010-06-14")

## End(Not run)
```

---

inat_map	<i>Plot iNaturalist observations</i>
----------	--------------------------------------

---

**Description**

Plot observations from iNaturalist. You have the option of automatically plotting, or returning a ggplot map object that you can add layers onto.

**Usage**

```
inat_map(data, map = "usa", subregion = ".", plot = TRUE)
```

**Arguments**

data	data frame of iNaturalist observations.
map	the map region to plot, you can find full documentation in the <a href="#">map</a> package, default is USA.
subregion	the name of the subregion to plot, see full documentation in the <a href="#">map</a> package.
plot	a logical value. TRUE plots the map object and returns it, and FALSE returns a ggplot object that you can modify and plot later.

**Value**

A ggplot map object.

**Examples**

```
## Not run:
  m_obs <- get_inat_obs(taxon_name = "Ambystoma maculatum")
  salamander_map <- inat_map(m_obs, plot = FALSE)
  ### Now we can modify the returned map
  salamander_map + borders("state") + theme_bw()

## End(Not run)
```



# Index

`get_inat_obs`, [2](#)  
`get_inat_obs_id`, [3](#)  
`get_inat_obs_project`, [4](#)  
`get_inat_obs_user`, [5](#)  
`get_inat_taxon_stats`, [5](#)  
`get_inat_user_stats`, [6](#)

`inat_map`, [7](#)

`map`, [7](#)