Package 'rebird'

July 23, 2025

Version 1.3.0

Title R Client for the eBird Database of Bird Observations

Description A programmatic client for the eBird database (<https://ebird.org/home>), including functions for searching for bird observations by geographic location (latitude, longitude), eBird hotspots, location identifiers, by notable sightings, by region, and by taxonomic name.

Depends R (>= 2.10)

License MIT + file LICENSE

URL https://docs.ropensci.org/rebird/, https://github.com/ropensci/rebird

BugReports https://github.com/ropensci/rebird/issues

VignetteBuilder knitr

Imports methods, utils, stats, httr (>= 1.0.0), jsonlite, dplyr, assertthat

Suggests knitr, rmarkdown, testthat, covr

RoxygenNote 7.1.1

X-schema.org-applicationCategory Data Access

X-schema.org-keywords birds, birding, ebird, database, data, biology, observations, sightings, ornithology

X-schema.org-isPartOf https://ropensci.org

Encoding UTF-8

NeedsCompilation no

Author Rafael Maia [aut], Scott Chamberlain [aut] (ORCID: <https://orcid.org/0000-0003-1444-9135>), Andy Teucher [aut], Guy Babineau [ctb], Marianna Foos [ctb], David Bradnum [ctb],

Sebastian Pardo [aut, cre] (ORCID:
<https://orcid.org/0000-0002-4147-5796>)

Maintainer Sebastian Pardo <sebpardo@gmail.com>

Repository CRAN

Date/Publication 2021-09-20 07:00:02 UTC

Contents

`ebirdchecklistfeed` *Checklist feed on a date at a region or hotspot*

Description

Returns checklist-level information reported in a given region or hotspot. Note only bird information is species count.

Usage

```
ebirdchecklistfeed(loc, date, max = 10, sleep = 0, key = NULL, ...)
```

Arguments

loc	(required) Region code or locID (if a hotspot). Region code can be country code (e.g. "US"), subnational1 code (states/provinces, e.g. "US-NV"), or subnational2 code (counties, e.g. "US-VA-003").
date	(required) Date of historic observation date formatted according to ISO 8601 (e.g. 'YYYY-MM-DD', or 'YYYY-MM-DD hh:mm'). Hours and minutes are excluded.
max	Maximum number of result rows to return in this request (between 1 and 200, default 10)
sleep	Time (in seconds) before function sends API call. The defaults is zero. Set this to a higher number if you are using this function in a loop with many API calls.
key	eBird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your `.Renviron` file as an environment variable called `EBIRD_KEY`.
...	Curl options passed on to `GET`.

Value

A data.frame containing the collected information:

"locId": unique identifier for the locations

"subId": submission (checklist) identifier

"userDisplayName": first and last name of the observer

"numSpecies": number of species reported

"obsDt": observation date formatted according to ISO 8601 (e.g. 'YYYY-MM-DD', or 'YYYY-MM-DD hh:mm'). Hours and minutes are excluded if the observer did not report an observation time

"obsTime": observation time (24hr)

"subID": deprecated submission identifier

"loc": delimited string of location descriptors

Author(s)

Marianna Foos <marianna.foos@gmail.com>

References

http://ebird.org/

Examples

```
## Not run:
ebirdchecklistfeed(loc = "L207391", date = "2020-03-24", max = 10)

## End(Not run)
```

ebirdfreq *Download historical frequencies of bird observations from eBird*

Description

NOTE: Currently disabled.

Usage

```
ebirdfreq(
  loctype,
  loc,
  startyear = 1900,
  endyear = format(Sys.Date(), "%Y"),
  startmonth = 1,
  endmonth = 12,
  long = TRUE,
  ...
)
```

Arguments

loctype	String with location type. Either "states", "counties", or "hotspots".
loc	String with location identifier. If querying states or provinces, the two letter country code followed by the two letter state code and separated by "-" (e.g. "US-NY"). If querying counties, is as in states/provinces, but appending county identifier after a dash. For counties in the US, the county codes is a 3-digit number specific to each state (e.g. Bronx County: "US-NY-005"). For counties in Canada, county codes are two-letter identifiers (e.g. Metro Vancouver: "CA-BC-GV"). If querying hotspots then the unique identifier is a 6-digit number prepended with an "L" (e.g. "L196159"). All these codes can be found by looking at the URL in each respective location/hotspot webpage (which are accessible through the "Explore Data" tab).
startyear	Starting year for query. Defaults to 1900.
endyear	Ending year for query. Defaults to current year specified by Sys.Date().
startmonth	Starting month for query as an integer (1-12). Defaults to January.
endmonth	Ending month for query as an integer (1-12). Defaults to December.
long	Logical, Should output be in long format? Defaults to TRUE. If FALSE then output will be in wide format.
...	Curl options passed on to GET

Details

This function was the only `rebird` function to not use the API and formulated a url-based query instead. Now you need to be logged into eBird to download the frequency data, but we can't authenticate through R, so this function does not work. This functionality is likely to be added to the API in the future, so we are keeping the function in the meantime, but it throws an informative error, and provides the constructed url to obtain the frequency data manually through your browser.

Value

This function currently returns an error, but also provides the constructed url to manually obtain the data for the location and dates requested through your browser.

A data frame containing the collected information. If in long format:

"monthQt": month and week (eBird data divides each month by four weeks)

"comName": species common name

"frequency": proportion of times the species was seen in a specified week

"sampleSize" number of complete eBird checklists submitted for specified given week @return If in wide format, then first column is the species list and all other columns are of individual weeks (four in each month). First row contains the number of complete checklists for each week.

Author(s)

Andy Teucher <andy.teucher@gmail.com>, Sebastian Pardo <sebpardo@gmail.com>

References

http://ebird.org/

Examples

```
## Not run:
ebirdfreq("states", "US-NY", 2014, 2014, 1, 12)
ebirdfreq("counties", "CA-BC-GV", 1900, 2015, 1, 3)
ebirdfreq("hotspots", "L196159", long=FALSE)

## End(Not run)
```

`ebirdgeo` *Sightings at location determined by latitude/longitude*

Description

Returns the most recent sighting date and specific location for the requested species of bird reported within the number of days specified and reported in the specified area.

Usage

```
ebirdgeo(
  species = NULL,
  lat = NULL,
  lng = NULL,
  dist = NULL,
  back = NULL,
  max = NULL,
  locale = NULL,
  provisional = FALSE,
  hotspot = FALSE,
  sleep = 0,
  key = NULL,
  ...
)
```

Arguments

species	Species code of the species of interest. Scientific names can be specified if wrapped around the species_code function. Defaults to NULL, so sightings for all species are returned. See eBird taxonomy for more information: https://ebird.org/science/use-ebird-data/the-ebird-taxonomy.
lat	Decimal latitude. value between -90.00 and 90.00, up to two decimal places of precision. Defaults to latitude based on IP.
lng	Decimal longitude. value between -180.00 and 180.00, up to two decimal places of precision. Defaults to longitude based on IP.
dist	Distance defining radius of interest from given lat/lng in kilometers (between 0 and 50, defaults to 25).
back	Number of days back to look for observations (between 1 and 30, defaults to 14).
max	Maximum number of result rows to return in this request (between 1 and 10000, defaults to all).
locale	Language/locale of response (when translations are available). See https://docs.oracle.com/javase/6/docs/a (defaults to en_US).
provisional	Should flagged records that have not been reviewed be included? (defaults to FALSE).
hotspot	Should results be limited to sightings at birding hotspots? (defaults to FALSE).
sleep	Time (in seconds) before function sends API call (defaults to zero. Set to higher number if you are using this function in a loop with many API calls).
key	eBird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your .Renviron file as an environment variable called EBIRD_KEY.
...	Curl options passed on to GET

Value

A data.frame containing the collected information:

"comName": species common name

"howMany": number of individuals observed, NA if only presence was noted

"lat": latitude of the location

"lng": longitude of the location

"locID": unique identifier for the location

"locName": location name

"locationPrivate": TRUE if location is not a birding hotspot

"obsDt": observation date formatted according to ISO 8601 (e.g. 'YYYY-MM-DD', or 'YYYY-MM-DD hh:mm'). Hours and minutes are excluded if the observer did not report an observation time.

"obsReviewed": TRUE if observation has been reviewed, FALSE otherwise

"obsValid": TRUE if observation has been deemed valid by either the automatic filters or a regional viewer, FALSE otherwise

"sciName" species' scientific name

Author(s)

Rafael Maia <rm72@zips.uakron.edu>, Sebastian Pardo <sebpardo@gmail.com>

References

http://ebird.org/

Examples

```
## Not run:
ebirdgeo('amegfi', 42, -76) # American Goldfinch
ebirdgeo(species_code('spinus tristis'), 42, -76) # same as above
ebirdgeo(lat=42, lng=-76, max=10, provisional=TRUE, hotspot=TRUE)
ebirdgeo(species_code('Anas platyrhynchos'), 39, -121, max=5)
library('httr')
ebirdgeo(species_code('Anas platyrhynchos'), 39, -121, max=5, config=verbose())
ebirdgeo(species_code('Anas platyrhynchos'), 39, -121, max=5, config=progress())
# ebirdgeo(species_code('Anas platyrhynchos'), 39, -121, max=5, config=timeout(0.1))

## End(Not run)
```

`ebirdhistorical` *Historic observations on a date at a region or hotspot*

Description

Returns a list of taxa reported in a given region or hotspot on a specific date

Usage

```
ebirdhistorical(
  loc,
  date,
  sortKey = "mrec",
  categories = "all",
  max = 10000,
  fieldSet = "simple",
  provisional = FALSE,
  limitToHotspots = FALSE,
  sleep = 0,
  key = NULL,
  ...
)
```

Arguments

`loc`	(required) Region code or locID (if a hotspot). Region code can be country code (e.g. "US"), subnational1 code (states/provinces, e.g. "US-NV"), or subnational2 code (counties, e.g. "US-VA-003").
`date`	(required) Date of historic observation date formatted according to ISO 8601 (e.g. 'YYYY-MM-DD', or 'YYYY-MM-DD hh:mm'). Hours and minutes are excluded.
`sortKey`	[mrec|create] Whether to order results by latest observation date or by latest creation date. The default is by observation date.
`categories`	[domestic|form|hybrid|intergrade|issf|slash|species|spuh] This is useful for limiting results to certain taxonomic categories. The default is all. Multiple categories may be comma-separated.
`max`	Maximum number of result rows to return in this request. (A number between 1 and 10000. The default is 10000)
`fieldSet`	[simple|full] This is set to restrict results to either all or a subset of sighting fields. The default is simple.
`provisional`	Should flagged records that have not been reviewed be included?
`limitToHotspots`	Should results be limited to sightings at birding hotspots? The default is FALSE.
`sleep`	Time (in seconds) before function sends API call. The defaults is zero. Set this to a higher number if you are using this function in a loop with many API calls.

key	eBird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your .Renviron file as an environment variable called EBIRD_KEY.
...	Curl options passed on to GET.

Value

A data.frame containing the collected information:

"speciesCode": species codes

"comName": species common names

"sciName" species' scientific names

"locId": unique identifier for the locations

"locName": location name

"obsDt": observation date formatted according to ISO 8601 (e.g. 'YYYY-MM-DD', or 'YYYY-MM-DD hh:mm'). Hours and minutes are excluded if the observer did not report an observation time

"howMany": "howMany": number of individuals observed, NA if only presence was noted

"obsValid": TRUE if observation has been deemed valid by either the automatic filters or a regional viewer, FALSE otherwise

"obsReviewed": TRUE if observation has been reviewed, FALSE otherwise

"locationPrivate": TRUE if location is not a birding hotspot

"subID": submission ID

"subnational2Code": county code (returned if simple=FALSE)

"subnational2Name": county name (returned if simple=FALSE)

"subnational1Code": state/province ISO code (returned if simple=FALSE)

"subnational1Name": state/province name (returned if simple=FALSE)

"countryCode": country ISO code (returned if simple=FALSE)

"countryName": country name (returned if simple=FALSE)

"userDisplayName": first and last name of the observer (returned if simple=FALSE)

"obsID": observation ID (returned if simple=FALSE)

"checklistID": checklist ID (returned if simple=FALSE)

"presenceNoted": 'true' if user marked presence but did not count the number of birds. 'false' otherwise (returned if simple=FALSE)

"hasComments": 'true' if comments are included (returned if simple=FALSE)

"hasRichMedia": 'true' if rich media (e.g. photos/sounds) are included (returned if simple=FALSE)

"firstName": observer's first name (returned if simple=FALSE)

"lastName": observer's last name (returned if simple=FALSE)

Author(s)

Guy Babineau <guy.babineau@gmail.com>

References

http://ebird.org/

Examples

```
## Not run:
ebirdhistorical(loc = 'US-VA-003', date='2019-02-14',max=10)
ebirdhistorical(loc = 'L196159', date='2019-02-14', fieldSet='full')

## End(Not run)
```

---

ebirdhotspot *Recent observations at hotspots*

Description

Returns the most recent sighting information reported in a given vector of hotspots.

Usage

```
ebirdhotspot(
  locID,
  species = NULL,
  back = NULL,
  max = NULL,
  locale = NULL,
  provisional = FALSE,
  sleep = 0,
  key = NULL,
  ...
)
```

Arguments

locID	(required) Vector containing code(s) for up to 10 regions of interest; here, regions are the locIDs of hotspots. Values that are not valid or are not hotspots are ignored.
species	Scientific name of the species of interest (not case sensitive). Defaults to NULL, in which case sightings for all species are returned. See eBird taxonomy for more information: http://ebird.org/content/ebird/about/ebird-taxonomy
back	Number of days back to look for observations (between 1 and 30, defaults to 14).
max	Maximum number of result rows to return in this request (between 1 and 10000, defaults to all)
locale	Language/locale of response (when translations are available). See http://java.sun.com/javase/6/docs/api/j (defaults to en_US)

provisional	Should flagged records that have not been reviewed be included? (defaults to FALSE)
sleep	Time (in seconds) before function sends API call (defaults to zero. Set to higher number if you are using this function in a loop with many API calls).
key	eBird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your .Renviron file as an environment variable called EBIRD_KEY.
...	Curl options passed on to GET

Value

A data.frame containing the collected information:

"comName": species common name

"howMany": number of individuals observed, NA if only presence was noted

"lat": latitude of the location

"lng": longitude of the location

"locID": unique identifier for the location

"locName": location name

"locationPrivate": TRUE if location is not a birding hotspot

"obsDt": observation date formatted according to ISO 8601 (e.g. 'YYYY-MM-DD', or 'YYYY-MM-DD hh:mm'). Hours and minutes are excluded if the observer did not report an observation time.

"obsReviewed": TRUE if observation has been reviewed, FALSE otherwise

"obsValid": TRUE if observation has been deemed valid by either the automatic filters or a regional viewer, FALSE otherwise

"sciName" species' scientific name

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

http://ebird.org/

Examples

```
## Not run:
ebirdhotspot(locID=c('L99381','L99382'), species='larus delawarensis')
ebirdhotspot('L99381', max=10, provisional=TRUE)

## End(Not run)
```

`ebirdhotspotlist` *Hotspots in a region or nearby coordinates*

Description

Get the list of hotspots within a region, or within a radius of up to 50 kilometers, from a given set of coordinates.

Usage

```
ebirdhotspotlist(
  regionCode = NULL,
  lat = NULL,
  lng = NULL,
  dist = NULL,
  back = NULL,
  sleep = 0,
  key = NULL,
  ...
)
```

Arguments

`regionCode`	The country, subnational1 or subnational2 code. If 'regionCode' is provided then latitude and longitude are ignored.
`lat`	Decimal latitude. value between -90.00 and 90.00, up to two decimal places of precision. Defaults to latitude based on IP if neither 'regionCode' nor 'lat' and 'lng' are provided.
`lng`	Decimal longitude. value between -180.00 and 180.00, up to two decimal places of precision. Defaults to longitude based on IP if neither 'regionCode' nor 'lat' and 'lng' are provided.
`dist`	The search radius from the given set of coordinates, in kilometers (between 0 and 500, defaults to 25).
`back`	Only fetch hotspots which have been visited up to 'back' days ago (defaults to 'NULL').
`sleep`	Time (in seconds) before function sends API call (defaults to zero. Set to higher number if you are using this function in a loop with many API calls).
`key`	eBird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your `.Renviron` file as an environment variable called `EBIRD_KEY`.
`...`	Curl options passed on to `GET`

Value

A data.frame with ten columns containing:

"locId": unique identifier for the hotspot

"locName": hotspot name

"countryCode": country code

"subnational1Code": subnational1 code (state/province level)

"subnational2Code": subnational2 code (county/municipality level)

"lat": latitude of the hotspot

"lng": longitude of the hotspot

"latestObsDt": Date of latest observation

"numSpeciesAllTime": Total number of species recorded in the hotspot

Author(s)

Sebastian Pardo <sebpardo@gmail.com>, David Bradnum <dbradnum@gmail.com>

References

http://ebird.org/

Examples

```
## Not run:
ebirdhotspotlist("CA-NS-HL")
ebirdhotspotlist("VA")
ebirdhotspotlist(lat = 30, lng = -90, dist = 10)
library(httr)
ebirdhotspotlist("CA-NS-HL", config = verbose())

## End(Not run)
```

---

ebirdloc *Recent observations at a locality*

Description

Returns the most recent sighting information reported in a given vector of locations (including non-hotspots).

Usage

```
ebirdloc(
  locID,
  species = NULL,
  back = NULL,
  max = NULL,
  locale = NULL,
  provisional = FALSE,
  simple = TRUE,
  sleep = 0,
  key = NULL,
  ...
)
```

Arguments

locID	(required) Vector containing code(s) for up to 10 regions of interest; here, values that are not hotspots are returned. Values that are not valid are ignored.
species	Scientific name of the species of interest (not case sensitive). Defaults to NULL, in which case sightings for all species are returned. See eBird taxonomy for more information: http://ebird.org/content/ebird/about/ebird-taxonomy
back	Number of days back to look for observations (between 1 and 30, defaults to 14).
max	Maximum number of result rows to return in this request (between 1 and 10000, defaults to all)
locale	Language/locale of response (when translations are available). See http://java.sun.com/javase/6/docs/api/j (defaults to en_US)
provisional	Should flagged records that have not been reviewed be included? (defaults to FALSE)
simple	Logical. Whether to return a simple (TRUE, default) or detailed (FALSE) set of results fields.
sleep	Time (in seconds) before function sends API call (defaults to zero. Set to higher number if you are using this function in a loop with many API calls).
key	ebird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your `.Renviron` file as an environment variable called `EBIRD_KEY`.
...	Curl options passed on to `GET`

Value

A data.frame containing the collected information:

"comName": species common name

"howMany": number of individuals observed, NA if only presence was noted

"lat": latitude of the location

"lng": longitude of the location

"locID": unique identifier for the location

"locName": location name

"locationPrivate": TRUE if location is not a birding hotspot

"obsDt": observation date formatted according to ISO 8601 (e.g. 'YYYY-MM-DD', or 'YYYY-MM-DD hh:mm'). Hours and minutes are excluded if the observer did not report an observation time.

"obsReviewed": TRUE if observation has been reviewed, FALSE otherwise

"obsValid": TRUE if observation has been deemed valid by either the automatic filters or a regional viewer, FALSE otherwise

"sciName" species' scientific name

"subnational2Code": county code (returned if simple=FALSE)

"subnational2Name": county name (returned if simple=FALSE)

"subnational1Code": state/province ISO code (returned if simple=FALSE)

"subnational1Name": state/province name (returned if simple=FALSE)

"countryCode": country ISO code (returned if simple=FALSE)

"countryName": country name (returned if simple=FALSE)

"userDisplayName": first and last name of the observer (returned if simple=FALSE)

"firstName": observer's first name (returned if simple=FALSE)

"lastName": observer's last name (returned if simple=FALSE)

"subID": submission ID (returned if simple=FALSE)

"obsID": observation ID (returned if simple=FALSE)

"checklistID": checklist ID (returned if simple=FALSE)

"presenceNoted": 'true' if user marked presence but did not count the number of birds. 'false' otherwise (returned if simple=FALSE)

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

http://ebird.org/

Examples

```
## Not run:
ebirdloc(locID = c('L99381','L99382'))
ebirdloc('L99381', 'Branta canadensis', provisional=TRUE)

## End(Not run)
```

`ebirdnotable` *Recent nearby notable observations*

Description

Returns the most recent notable observations by either latitude/longitude, hotspot or location ID, or particular region.

Usage

```
ebirdnotable(
  lat = NULL,
  lng = NULL,
  dist = NULL,
  locID = NULL,
  region = NULL,
  back = NULL,
  max = NULL,
  provisional = FALSE,
  hotspot = FALSE,
  simple = TRUE,
  sleep = 0,
  key = NULL,
  ...
)
```

Arguments

`lat`	Decimal latitude. value between -90.00 and 90.00, up to two decimal places of precision.
`lng`	Decimal longitude. value between -180.00 and 180.00, up to two decimal places of precision.
`dist`	Distance defining radius of interest from given lat/lng in kilometers (between 0 and 50, defaults to 25)
`locID`	Vector containing code(s) for up to 10 locations of interest.
`region`	Region code corresponding to selected region type. For supported region and coding, see https://confluence.cornell.edu/display/CLOISAPI/eBird-1.1-RegionCodeReference
`back`	Number of days back to look for observations (between 1 and 30, defaults to 14).
`max`	Maximum number of result rows to return in this request (between 1 and 10000, defaults to all).
`provisional`	Should flagged records that have not been reviewed be included? (defaults to FALSE)
`hotspot`	Should results be limited to sightings at birding hotspots? (defaults to FALSE).

simple	Logical. Whether to return a simple (TRUE, default) or detailed (FALSE) set of results fields.
sleep	Time (in seconds) before function sends API call (defaults to zero. Set to higher number if you are using this function in a loop with many API calls).
key	eBird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your .Renviron file as an environment variable called EBIRD_KEY.
...	Curl options passed on to GET

Value

A data.frame containing the collected information:

"speciesCode": species code

"comName": species common name

"sciName" species' scientific name

"locId": unique identifier for the location

"locName": location name

"obsDt": observation date formatted according to ISO 8601 (e.g. 'YYYY-MM-DD', or 'YYYY-MM-DD hh:mm'). Hours and minutes are excluded if the observer did not report an observation time.

"howMany": number of individuals observed, NA if only presence was noted

"lat": latitude of the location

"lng": longitude of the location

"obsValid": TRUE if observation has been deemed valid by either the

"obsReviewed": TRUE if observation has been reviewed, FALSE otherwise

"locationPrivate": TRUE if location is not a birding hotspot automatic filters or a regional viewer, FALSE otherwise

"subId": submission ID

"subnational2Code": county code (returned if simple=FALSE)

"subnational2Name": county name (returned if simple=FALSE)

"subnational1Code": state/province ISO code (returned if simple=FALSE)

"subnational1Name": state/province name (returned if simple=FALSE)

"countryCode": country ISO code (returned if simple=FALSE)

"countryName": country name (returned if simple=FALSE)

"userDisplayName": observer's eBird username (returned if simple=FALSE)

"obsID": observation ID (returned if simple=FALSE)

"checklistID": checklist ID (returned if simple=FALSE)

"presenceNoted": 'true' if user marked presence but did not count the number of birds. 'false' otherwise (returned if simple=FALSE)

"firstName": observer's first name (returned if simple=FALSE)

"lastName": observer's last name (returned if simple=FALSE)

Note

ebirdnotable requires that either latitude/longitude, location ID, or region be passed to the function. Multiple entries will result in the most specific being used. If none is supplied, defaults to lat/lng based on your IP.

Author(s)

Rafael Maia <rm72@zips.uakron.edu>, Sebastian Pardo <sebpardo@gmail.com>

References

http://ebird.org/

Examples

```
## Not run:
ebirdnotable(lat=42, lng=-70)
ebirdnotable(region='US', max=10)
ebirdnotable(region='US-OH')
ebirdnotable(region='CA-NS-HL')
ebirdnotable(locID = c('L275836','L124345'))

## End(Not run)
```

---

ebirdregion *Recent observations at a region or hotspot*

Description

Returns the most recent sighting information reported in a given region or hotspot.

Usage

```
ebirdregion(
  loc,
  species = NULL,
  back = NULL,
  max = NULL,
  locale = NULL,
  provisional = FALSE,
  hotspot = FALSE,
  simple = TRUE,
  sleep = 0,
  key = NULL,
  ...
)
```

Arguments

loc	(required) Region code or locID (for hotspots). Region code can be country code (e.g. "US"), subnational1 (states/provinces, e.g. "US-NV"), or subnational2 code (counties, e.g. "CA-BC-GV").
species	eBird species code. See ebirdtaxonomy for a full list of scientific names, common names, and species codes. Alternatively, you can wrap the scientific name in the species_code function which will return the eBird species code. Defaults to NULL, in which case sightings for all species are returned. See eBird taxonomy for more information: https://ebird.org/science/use-ebird-data/the-ebird-taxonomy
back	Number of days back to look for observations (between 1 and 30, defaults to 14).
max	Maximum number of result rows to return in this request (between 1 and 10000, defaults to all)
locale	Language/locale of response (when translations are available). See https://docs.oracle.com/javase/6/docs/api/java/util/Locale.html and https://support.ebird.org/en/support/solutions/articles/48000804865-bird-names-in-ebird (defaults to en_US).
provisional	Should flagged records that have not been reviewed be included? (defaults to FALSE)
hotspot	Should results be limited to sightings at birding hotspots? (defaults to FALSE).
simple	Logical. Whether to return a simple (TRUE, default) or detailed (FALSE) set of results fields. Detailed results are only available if loc is a locID.
sleep	Time (in seconds) before function sends API call (defaults to zero. Set to higher number if you are using this function in a loop with many API calls).
key	eBird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your .Renviron file as an environment variable called EBIRD_KEY to avoid having to constantly supply the key, and to avoid accidentally sharing it publicly.
...	Curl options passed on to GET

Value

A data.frame containing the collected information:

"speciesCode": species code

"comName": species common name

"sciName" species' scientific name

"locID": unique identifier for the location

"locName": location name

"obsDt": observation date formatted according to ISO 8601 (e.g. 'YYYY-MM-DD', or 'YYYY-MM-DD hh:mm'). Hours and minutes are excluded if the observer did not report an observation time.

"howMany": number of individuals observed, NA if only presence was noted

"lat": latitude of the location

"lng": longitude of the location

"obsValid": TRUE if observation has been deemed valid by either the automatic filters or a regional viewer, FALSE otherwise

"obsReviewed": TRUE if observation has been reviewed, FALSE otherwise

"locationPrivate": TRUE if location is not a birding hotspot

"subId": submission ID

Author(s)

Rafael Maia <rm72@zips.uakron.edu>

References

http://ebird.org/

Examples

```
## Not run:
ebirdregion(loc = 'US', species = 'btbwar')
ebirdregion(loc = 'US', species = species_code('Setophaga caerulescens')) # same as above
ebirdregion(loc = 'L196159', species = 'bkcchi', back = 30)
ebirdregion('US-OH', max = 10, provisional = TRUE, hotspot = TRUE)

## End(Not run)
```

---

ebirdregioncheck *Check if a region type is valid*

Description

Check if a region type is valid

Usage

```
ebirdregioncheck(loc, key = NULL, ...)
```

Arguments

loc	The location code to be checked.
key	ebird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your .Renviron file as an environment variable called EBIRD_KEY.
...	Curl options passed on to GET

Value

Logical.

Author(s)

Sebastian Pardo <sebpardo@gmail.com>, Andy Teucher <andy.teucher@gmail.com>

References

http://ebird.org/

Examples

```
## Not run:
ebirdregioncheck("US")
ebirdregioncheck("CA-BC")
ebirdregioncheck("CA-BC-GV")

## End(Not run)
```

---

ebirdregioninfo *Region and hotspot info*

Description

Region and hotspot info

Usage

```
ebirdregioninfo(loc, format = "full", key = NULL, ...)
```

Arguments

loc	The location or hotspot code to be checked. A single location only.
format	Different options for displaying hierarchy of the region's name: [nameonly|namequall|detailed|detailednoq defaults to full. Not used for hotspots.
key	eBird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your .Renviron file as an environment variable called EBIRD_KEY.
...	Curl options passed on to GET

Value

When region is a hotspot, a data frame (with some redundant information) containing:

"locId", "locID": hotspot ID

"name", "locName": hotspot name

"latitude", "longitude", "lat", "long": hotspot latitude and longitude (point location)

"countryCode", "countryName": code and name of the country where hotspot is located

"subnational1Code", "subnational1Name": code and name of the subnational1 area (e.g. state or province) where hotspot is located

"subnational2Code", "subnational2Name": code and name of the subnational2 area (e.g. county) where hotspot is located

"isHotspot": logical, whether region is a hotspot (should always be TRUE)

"hierarchicalName": full hotspot name including subnational1, subnational2, and country info

When region is a subnational1, subnational2, or country code, a data frame containing:

"region": name of the region, varies depending on value of "format" provided

"minX", "maxX", "minY", "maxY": lat/long bounds of the region

Author(s)

Sebastian Pardo <sebpardo@gmail.com>, Andy Teucher <andy.teucher@gmail.com>, Guy Babineau <guy.babineau@gmail.com>

References

http://ebird.org/

Examples

```
## Not run:
ebirdregioninfo("US")
ebirdregioninfo("CA-BC-GV")
ebirdregioninfo("CA-BC-GV", format = "revdetailed") # reverse order of region name
ebirdregioninfo("L196159")

## End(Not run)
```

`ebirdregionspecies` *Get a list of species codes ever seen in a location.*

Description

Returns the eBird codes for all species-level taxa recorded in a particular region or location. Codes are returned in taxonomic order.

Usage

```
ebirdregionspecies(location, key = NULL, ...)
```

Arguments

location	Any valid location, USFWS region, subnational2, subnational1, country, or custom region code. (Location can be a hotspot or personal location).
key	eBird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your .Renviron file as an environment variable called EBIRD_KEY.
...	Curl options passed on to GET

Value

A single column data.frame containing the collected information:

"speciesCode": eBird species code, suitable for joining to the ebirdtaxonomy

Author(s)

David Bradnum <david.bradnum@gmail.com>

References

http://ebird.org/

Examples

```
## Not run:
ebirdregionspecies("GB") # all in Great Britain
ebirdregionspecies("GB-ENG") # all in England
ebirdregionspecies("GB-ENG-LND") # all in London

library(dplyr)
taxonomy <- ebirdtaxonomy()
localSpecies <- ebirdregionspecies("L5803024") # specific hotspot
inner_join(localSpecies, taxonomy)

## End(Not run)
```

---

ebirdsubregionlist *List sub-regions within a specified region.*

Description

List sub-regions within a specified region.

Usage

```
ebirdsubregionlist(
  regionType = c("country", "subnational1", "subnational2"),
  parentRegionCode,
  key = NULL,
  ...
)
```

Arguments

regionType	The type of region to search for. Must be one of 'country', 'subnational1' or 'subnational2'.
parentRegionCode	The region to search within. Must be a valid country or subnational1 code. If 'regionType' is 'country' then this parameter is ignored (since the search will automatically be world-wide).
key	eBird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your .Renviron file as an environment variable called EBIRD_KEY.
...	Curl options passed on to GET

Value

A data.frame containing:

"code": eBird code for the subregion

"name": full name for the subregion

Author(s)

David Bradnum <dbradnum@gmail.com>

References

http://ebird.org/

Examples

```
## Not run:
ebirdsubregionlist("country")
ebirdsubregionlist("subnational1", "US")
ebirdsubregionlist("subnational2", "US-NY")

## End(Not run)
```

`ebirdtaxonomy` *eBird Taxonomy*

Description

Returns a data.frame of all taxa in the eBird taxonomy for the given combination of categories. Defaults to all categories. Any taxon with the category of 'species' may be used as a parameter in service calls that take a species code. Any taxon not in this category will be rejected by these services at this time.

Usage

```
ebirdtaxonomy(cat = NULL, locale = NULL, key = NULL, ...)
```

Arguments

`cat`	Species category. String or character vector with one of more of: "domestic", "form", "hybrid", "intergrade", "issf", "slash", "species", "spuh". If not specified, defaults to all. For more info about the meaning of species categories, see https://ebird.org/science/use-ebird-data/the-ebird-taxonomy.
`locale`	Language/locale of response (when translations are available). See https://docs.oracle.com/javase/6/docs/api/java/util/Locale.html and https://support.ebird.org/en/support/solutions/articles/48000804865-bird-names-in-ebird (defaults to en_US).
`key`	eBird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your `.Renviron` file as an environment variable called `EBIRD_KEY` to avoid having to constantly supply the key, and to avoid accidentally sharing it publicly.
`...`	Curl options passed on to `GET`

Value

A data.frame containing the collected information:

"sciName": Taxon's scientific name.

"comName": Taxon's common name.

"speciesCode": Unique species code.

"category": Taxon's species category.

"taxonOrder": Numeric value determining the order in which taxonomic lists are presented.

"bandingCodes": Taxon's ABA banding code(s).

"comNameCodes": Taxon's common name code(s).

"sciNameCodes": Taxon's scientific name code(s).

"order": Taxon's order.

"familyComName": Family's common name.

"familySciName": Family's scientific name.

"reportAs": Species code to report taxon as.

"extinct": Logical, whether the taxon is considered extinct.

"extinctYear": Year taxon became extinct. Currently unavailable.

Author(s)

Andy Teucher <andy.teucher@gmail.com>, Sebastian Pardo <sebpardo@gmail.com>

References

http://ebird.org/

Examples

```
## Not run:
ebirdtaxonomy()
ebirdtaxonomy(cat = c("spuh", "slash"))

## End(Not run)
```

---

getlatlng *get latitude and longitude from ip address*

Description

Returns the most recent and nearest reported sighting information with observations of a species.

Usage

```
getlatlng()
```

Value

a vector of length 2 with lat, lng in that order

Author(s)

Andy Teucher <andy.teucher@gmail.com>

References

http://ipinfo.io

Examples

```
## Not run:
getlatlng()

## End(Not run)
```

`nearestobs` *Recent nearby observations of a species*

Description

Returns the most recent and nearest reported sighting information with observations of a species.

Usage

```
nearestobs(
  speciesCode,
  lat = NULL,
  lng = NULL,
  dist = NULL,
  back = NULL,
  max = NULL,
  locale = NULL,
  provisional = FALSE,
  hotspot = FALSE,
  sleep = 0,
  key = NULL,
  ...
)
```

Arguments

`speciesCode`	(required) Species code of the species of interest. Scientific names can be specified if wrapped around the `species_code` function. Defaults to NULL, so sightings for all species are returned. See eBird taxonomy for more information: https://ebird.org/science/use-ebird-data/the-ebird-taxonomy.
`lat`	Decimal latitude. value between -90.00 and 90.00, up to two decimal places of precision. Defaults to latitude based on IP.
`lng`	Decimal longitude. value between -180.00 and 180.00, up to two decimal places of precision. Defaults to longitude based on IP.
`dist`	Distance defining radius of interest from given lat/lng in kilometers (between 0 and 50, defaults to 25)
`back`	Number of days back to look for observations (between 1 and 30, defaults to 14).
`max`	Maximum number of result rows to return in this request (between 1 and 10000, defaults to all).
`locale`	Language/locale of response (when translations are available). See https://docs.oracle.com/javase/6/docs/api/java/util/Locale.html (defaults to en_US).
`provisional`	Should flagged records that have not been reviewed be included? (defaults to FALSE).

hotspot	Should results be limited to sightings at birding hotspots? (defaults to FALSE).
sleep	Time (in seconds) before function sends API call (defaults to zero. Set to higher number if you are using this function in a loop with many API calls).
key	eBird API key. You can obtain one from https://ebird.org/api/keygen. We strongly recommend storing it in your .Renviron file as an environment variable called EBIRD_KEY.
...	Curl options passed on to GET

Value

A data.frame containing the collected information:

"speciesCode": species code

"comName": species common name

"sciName" species' scientific name

"locId": unique identifier for the location

"locName": location name

"obsDt": observation date formatted according to ISO 8601 (e.g. 'YYYY-MM-DD', or 'YYYY-MM-DD hh:mm'). Hours and minutes are excluded if the observer did not report an observation time.

"howMany": number of individuals observed, NA if only presence was noted

"lat": latitude of the location.

"lng": longitude of the location.

"obsValid": TRUE if observation has been deemed valid by either the automatic filters or a regional viewer, FALSE otherwise

"obsReviewed": TRUE if observation has been reviewed, FALSE otherwise

"locationPrivate": TRUE if location is not a birding hotspot

"subId": submission ID

Author(s)

Rafael Maia <rm72@zips.uakron.edu>, Sebastian Pardo <sebpardo@gmail.com>

References

http://ebird.org/

Examples

```
## Not run:
nearestobs('cangoo', 42, -76) # Canada Goose
nearestobs(species_code('branta canadensis'), 42, -76) # Same as above
nearestobs(species_code('branta canadensis'), 42, -76, max=10, provisional=TRUE, hotspot=TRUE)

## End(Not run)
```

rebird-deprecated *Deprecated functions in rebird*

Description

These functions still work but will be removed (defunct) in the next version.

Details

- ebirdregioncheck: Deprecated: 'ebirdregioncheck' will be removed in the next version of rebird. Use 'ebirdregioninfo' instead.
- ebirdloc: Deprecated: 'ebirdloc' will be removed in the next version of rebird as it might not be supported in the new eBird API. Use 'ebirdregion' instead.
- ebirdhotspot: Deprecated: 'ebirdhotspot' will be removed in the next version of rebird as it might not be supported in the new eBird API. Use 'ebirdregion' instead.

species_code *Return species code*

Description

Returns the species code for a given scientific name. Uses an internally-stored version of the taxonomy. Also provides a message with the common name, scientific name, and species code of the species.

Usage

```
species_code(sciname = NULL)
```

Arguments

sciname	(required) Character string of length 1 with the scientific name to look for. Case insensitive.

Value

A character string with the eBird species code.

Author(s)

Sebastian Pardo <sebpardo@gmail.com>

References

http://ebird.org/

Examples

```
species_code("Anhinga anhinga")
```

Index