Package 'ip2location.io'

March 25, 2025

Title Batch IP Data Retrieval and Storage Using 'IP2Location.io'

Version 0.0.0-2

Description A system for submitting multiple IP information queries to 'IP2Location.io''s IP Geolocation API and storing the resulting data in a dataframe. You provide a vector of IP addresses and your 'IP2Location.io' API key. The package returns a dataframe with one row per IP address and a column for each available data field (data fields not included in your API plan will contain NAs). This is the second submission of the package to CRAN.

License MIT + file LICENSE

Imports dplyr, httr, jsonlite, tidyselect

Encoding UTF-8

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0), mockery

Config/testthat/edition 3

NeedsCompilation no

Author Oriane Georgeac [aut, cre] (<https://orcid.org/0000-0001-6531-0075>)

Maintainer Oriane Georgeac <oriane.georgeac@gmail.com>

Repository CRAN

Date/Publication 2025-03-25 09:10:02 UTC

Contents

`get_ip_data` *Retrieve and Save Data for Multiple IP Addresses*

Description

This function retrieves information for a list of IP addresses using the IP2Location.io API, processes the data, and returns the data as a dataframe. The function also handles missing values by assigning NAs for any data field not provided by the user's API plan.

Usage

```
get_ip_data(ip_addresses, api_key)
```

Arguments

`ip_addresses`	A vector of IP addresses.
`api_key`	Your IP2Location.io API key in quotation marks.

Details

The function extracts the following fields (some of which may contain NAs depending on the user's API plan) for each IP address:

Field	Description
ip	IP address
country_code	Country code
country_name	Country name
region_name	Region name
district	District
city_name	City name
latitude	Latitude
longitude	Longitude
zip_code	Zip code
time_zone	Time zone
asn	Autonomous system number
as	Autonomous system
isp	Internet service provider
domain	Domain
net_speed	Network speed
idd_code	International dialing code
area_code	Area code
weather_station_code	Weather station code
weather_station_name	Weather station name
mcc	Mobile country code
mnc	Mobile network code
mobile_brand	Mobile brand
elevation	Elevation

usage_type	Usage type
address_type	Address type
ads_category	Ads category
ads_category_name	Ads category name
continent_name	Continent name
continent_hemisphere	Continent hemisphere
country_capital	Country capital
country_language	Country language
region_code	Region code
time_zone_olson	Time zone (Olson format)
time_zone_current_time	Current time in the time zone
is_proxy	Whether the IP is a proxy (limited to public proxies in the Free and Starter plans)
fraud_score	Fraud score
proxy_last_seen	Last seen time of the proxy
proxy_type	Proxy type
proxy_threat	Proxy threat
proxy_provider	Proxy provider
proxy_is_vpn	Whether the proxy is a VPN
proxy_is_tor	Whether the proxy is Tor
proxy_is_data_center	Whether the proxy is a data center
proxy_is_public_proxy	Whether the proxy is a public proxy
proxy_is_web_proxy	Whether the proxy is a web proxy
proxy_is_web_crawler	Whether the proxy is a web crawler
proxy_is_residential_proxy	Whether the proxy is a residential proxy
proxy_is_consumer_privacy_network	Whether the proxy is a consumer privacy network
proxy_is_enterprise_private_network	Whether the proxy is an enterprise private network
proxy_is_spammer	Whether the proxy is a spammer
proxy_is_scanner	Whether the proxy is a scanner
proxy_is_botnet	Whether the proxy is a botnet

Value

A data frame with the extracted data for each IP.

Note

This function uses the IP2Location.io API. Make sure you have a valid API key. https://www.ip2location.io/pricing

Examples

```
## Not run:
# Example usage
ip_addresses <- c("8.8.8.8", "1.1.1.1")         # Example IP addresses
api_key <- "your_api_key_here"                  # Replace with your API key
ip_data <- get_ip_data(ip_addresses, api_key)   # Returns a dataframe
# If the user wants to save the dataframe as a CSV, they can do so:
write.csv(ip_data, "IP2location.csv", row.names = FALSE)
```

```
## End(Not run)
```

`safe_extract` *Safe Data Extraction from API Response*

Description

This function extracts the value of a given field from the parsed API response. It handles missing or empty data gracefully by returning NA if the field is missing.

Usage

```
safe_extract(x, field)
```

Arguments

`x`	The parsed JSON response (usually a list).
`field`	The name of the field to extract (as a string).

Value

The extracted field value, or NA if missing or empty.

Index