Package ‘amapGeocode’

April 19, 2021

Type Package

Title An Interface to the ‘AutoNavi Maps’ API Geocoding Services

Version 0.6.0

Description Getting and parsing data of location geocode/reverse-geocode and administrative regions from ‘AutoNavi Maps’<https://lbs.amap.com/api/webservice/summary> API.

License MIT + file LICENSE

Depends R (>= 3.5)

Imports httr, jsonlite, magrittr, sjmisc, stats, xml2, data.table, furrr

Suggests stringr, knitr, rmarkdown, testthat, spelling, covr

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.1.1

Language en-US

URL https://github.com/womeimingzi11/amapGeocode

BugReports https://github.com/womeimingzi11/amapGeocode/issues

NeedsCompilation no

Author Han Chen [aut, cre], Wanyanhan Jiang [ctb]

Maintainer Han Chen <chenhan28@gmail.com>

Repository CRAN

Date/Publication 2021-04-19 07:20:02 UTC

R topics documented:

convertCoord ................................................................. 2
convertCoord.individual .................................................. 3
extractAdmin ................................................................. 4
extractConvertCoord ....................................................... 5
**convertCoord**

Convert coordinate from different coordinate systems to AutoNavi system

**Description**

This function is a wrap of coordinate convert API of AutoNavi Map Service. While how to input the origin coordinate is sill unstable and 95% sure that it will have a breaking change in the future. Please consider carefully if introduced this function in product environment.

**Usage**

```
classic convertCoord(
    locations,
    key = NULL,
    coordsys = NULL,
    sig = NULL,
    output = "data.table",
    keep_bad_request = TRUE,
    ...
)
```

**Arguments**

- **locations**
  - Required.
  - String coordinate point from other coordinate system

- **key**
  - Optional.
  - Amap Key.
  - Applied from AutoNavi Map API official website [https://lbs.amap.com/dev/](https://lbs.amap.com/dev/)

- **coordsys**
  - Optional.
  - Coordinate System.
  - Support: ‘gps’,‘mapbar’, ‘baidu’ and ‘autonavi’-not convert

- **sig**
  - Optional.
  - Digital Signature.
  - How to use this argument? Please check here [https://lbs.amap.com/faq/account/key/72](https://lbs.amap.com/faq/account/key/72)
**convertCoord.individual**

Convert an individual coordinate from different coordinate systems to AutoNavi system

---

### Description

Convert an individual coordinate from different coordinate systems to AutoNavi system

---

#### output

Optional.
Output Data Structure.
Support JSON, XML and data.table. The default value is data.table.

#### keep_bad_request

Optional.
Keep Bad Request to avoid breaking a workflow, especially meaningful in a batch request

... Optional.
For compatibility only

### Value

Returns a JSON, XML or data.table of results containing detailed geocode information. See https://lbs.amap.com/api/webservice/guide/api/convert for more information.

### See Also

convertCoord

### Examples

```r
## Not run:
library(amapGeocode)

# Before the `convertCoord()` is executed,
# the token should be set by `option(amap_key = 'key')`
# or set by key argument in `convertCoord()`

# get result of converted coordinate system as a data.table
convertCoord("116.481499,39.990475", coordsys = "gps")

# get result of converted coordinate system as a XML
convertCoord("116.481499,39.990475", coordsys = "gps", to_table = FALSE)

## End(Not run)
```
Usage

convertCoord.individual(
    locations,
    key = NULL,
    coordsys = NULL,
    sig = NULL,
    output = "data.table",
    keep_bad_request = TRUE,
    ...
)

Arguments

locations  Required.
String coordinate point from other coordinate system

key        Optional.
Amap Key.
Applied from AutoNavi Map API official website https://lbs.amap.com/dev/

coordsys  Optional.
Coordinate System.
Support: 'gps', 'mapbar', 'baidu' and 'autonavi'-not convert

sig        Optional.
Digital Signature.
How to use this argument? Please check here https://lbs.amap.com/faq/account/key/72

output     Optional.
Output Data Structure.
Support JSON, XML and data.table. The default value is data.table.

keep_bad_request
Optional.
Keep Bad Request to avoid breaking a workflow, especially meaningful in a batch request

...
Optional.
For compatibility only

Value

Returns a JSON, XML or data.table of results containing detailed geocode information. See https://lbs.amap.com/api/webservice/guide/api/convert for more information.

extractAdmin  Get Subordinate Administrative Region from getAdmin request Now, it only support extract the first layer of subordinate administrative region information.
Description

Get Subordinate Administrative Region from getAdmin request. Now, it only support extract the first layer of subordinate administrative region information.

Usage

extractAdmin(res)

Arguments

res Response from getAdmin.

Value

Returns a data.table which extracts detailed subordinate administrative region information from results of getCoord. See https://lbs.amap.com/api/webservice/guide/api/district for more information.

See Also

getAdmin

Examples

## Not run:
library(dplyr)
library(amapGeocode)

# Before the `getAdmin()` is executed,
# the token should be set by `option(amap_key = `key`)`
# or set by key argument in `getAdmin()`

# Get subordinate administrative regions as a XML
getAdmin("Sichuan Province", output = "XML") %>%
  # extract subordinate administrative regions as a data.table
  extractAdmin()

## End(Not run)

---

**extractConvertCoord** Extract converted coordinate points from convertCoord request

Description

Extract converted coordinate points from convertCoord request
extractCoord

Usage

```r
extractConvertCoord(res)
```

Arguments

- `res` Required.
  Response from convertCoord.

Value

Returns a data.table which extracts converted coordinate points from request of convertCoord. See [https://lbs.amap.com/api/webservice/guide/api/convert](https://lbs.amap.com/api/webservice/guide/api/convert) for more information.

See Also

convertCoord

Examples

```r
## Not run:
library(dplyr)
library(amapGeocode)

# Before the `convertCoord()` is executed,
# the token should be set by `option(amap_key = 'key')`
# or set by key argument in `convertCoord()`

# get result of converted coordinate system as a XML
convertCoord("116.481499,39.990475", coordsys = "gps", to_table = FALSE) %>%
  # extract result of converted coordinate system as a data.table
  extractConvertCoord()

## End(Not run)
```
Value

Returns a data.table which extracts detailed coordinate information from results of getCoord. See https://lbs.amap.com/api/webservice/guide/api/georegeo for more information.

See Also

getCoord

Examples

```r
## Not run:
library(dplyr)
library(amapGeocode)

# Before the `getCoord()` is executed,
# the token should be set by `option(amap_key = 'key')`
# or set by key argument in `getCoord()`

# Get geocode as a XML
getCoord("IFS Chengdu", output = "XML") %>%
  # extract geocode regions as a data.table
  extractCoord()
```

---

### extractLocation

**Extract location from coordinate request**

**Description**

Extract location from coordinate request

**Usage**

```
extractLocation(res)
```

**Arguments**

- `res` Required.
  Response from getLocation.

**Value**

Returns a data.table which extracts detailed location information from results of getLocation. See https://lbs.amap.com/api/webservice/guide/api/georegeo for more information.
getAdmin

See Also

getLocation

Examples

```r
## Not run:
library(dplyr)
library(amapGeocode)

# Before the `getLocation()` is executed,
# the token should be set by `option(amap_key = 'key')`
# or set by key argument in `getLocation()`
# Get reverse-geocode as a XML
getLocation(104.043284, 30.666864, output = "XML") %>%
  # extract reverse-geocode regions as a table
  extractLocation()

## End(Not run)
```

---

getAdmin

Get Subordinate Administrative Regions from location

Description

Get Subordinate Administrative Regions from location

Usage

```r
getAdmin(
  keywords,
  key = NULL,
  subdistrict = NULL,
  page = NULL,
  offset = NULL,
  extensions = NULL,
  filter = NULL,
  callback = NULL,
  output = "data.table",
  keep_bad_request = TRUE,
  ...
)
```

Arguments

- **keywords**
  - Required.
  - Search keywords.
  - Rules: Country/Region, Province/State, City, County/District, Town, Country, Road, Number, Room, Building.
**getAdmin**

- **key**
  - Optional.
  - Amap Key.
  - Applied from 'AutoNavi' Map API official website [https://lbs.amap.com/dev/](https://lbs.amap.com/dev/)

- **subdistrict**
  - Optional.
  - Subordinate Administrative Level.
  - Display level of subordinate administrative regions. Available value: 0,1,2,3.
  - ‘0’ do not return subordinate administrative regions.
  - ‘1’ return first one subordinate administrative regions.
  - ‘2’ return first two subordinate administrative regions.
  - ‘3’ return first three subordinate administrative regions.

- **page**
  - Optional.
  - Which page to return.
  - Each time the outmost layer will return a maximum of 20 records. If the limit is exceeded, please request the next page of records with the page argument.

- **offset**
  - Optional.
  - Maximum records per page.
  - Maximum value is 20.

- **extensions**
  - Optional.
  - Return results controller.
  - ‘base’: does not return the coordinates of the administrative district boundary.
  - ‘all’: returns only the boundary value of the current query district, not the boundary value of the child node.

- **filter**
  - Optional.
  - Filter administrative regions.
  - Filtering by designated administrative divisions, which returns information only for the province/municipality.
  - It is strongly recommended to fill in this parameter in order to ensure the correct records.

- **callback**
  - Optional.
  - Callback Function.
  - The value of callback is the customized function. Only available with JSON output. If you don’t understand, it means you don’t need it, just like me.

- **output**
  - Optional.
  - Output Data Structure.
  - Support JSON, XML and data.table. The default value is data.table.

- **keep_bad_request**
  - Optional.
  - Keep Bad Request to avoid breaking a workflow, especially meaningful in a batch request

**Value**

Returns a JSON or XML of results containing detailed subordinate administrative region information. See [https://lbs.amap.com/api/webservice/guide/api/district](https://lbs.amap.com/api/webservice/guide/api/district) for more information.
getAdmin.individual

Get an individual data.table of Subordinate Administrative Regions from location

Description

Get an individual data.table of Subordinate Administrative Regions from location

Usage

getAdmin.individual(
    keywords,
    key = NULL,
    subdistrict = NULL,
    page = NULL,
    offset = NULL,
    extensions = NULL,
    filter = NULL,
    callback = NULL,
    output = "data.table",
    keep_bad_request = TRUE,
    ...
)
Arguments

keywords
Required.
Search keywords.
Rules: Country/Region, Province/State, City, County/District, Town, Country,
Road, Number, Room, Building.

key
Optional.
Amap Key.
Applied from 'AutoNavi’ Map API official website https://lbs.amap.com/dev/

subdistrict
Optional.
Subordinate Administrative Level.
Display level of subordinate administrative regions. Available value: 0,1,2,3.
‘0’ do not return subordinate administrative regions.
‘1’ return first one subordinate administrative regions.
‘2’ return first two subordinate administrative regions.
‘3’ return first three subordinate administrative regions.

page
Optional.
Which page to return.
Each time the outmost layer will return a maximum of 20 records. If the limit is exceeded, please request the next page of records with the page argument.

offset
Optional.
Maximum records per page.
Maximum value is 20.

extensions
Optional.
Return results controller.
‘base’: does not return the coordinates of the administrative district boundary.
‘all’: returns only the boundary value of the current query district, not the boundary value of the child node.

filter
Optional.
Filter administrative regions.
Filtering by designated administrative divisions, which returns information only for the province/municipality.
It is strongly recommended to fill in this parameter in order to ensure the correct records.

callback
Optional.
Callback Function.
The value of callback is the customized function. Only available with JSON output. If you don’t understand, it means you don’t need it, just like me.

output
Optional.
Output Data Structure.
Support JSON, XML and data.table. The default value is data.table.

keep_bad_request
Optional.
Keep Bad Request to avoid breaking a workflow, especially meaningful in a batch request.
... Optional.
For compatibility only

Value
Returns a JSON or XML of results containing detailed subordinate administrative region information. See https://lbs.amap.com/api/webservice/guide/api/district for more information.

getCoord
Get coordinate from location

Description
Get coordinate from location

Usage
getCoord(
    address,
    key = NULL,
    city = NULL,
    sig = NULL,
    output = "data.table",
    callback = NULL,
    keep_bad_request = TRUE,
    ...
)

Arguments
address Required.
Structured address information.
Rules: Country/Region, Province/State, City, County/District, Town, Country, Road, Number, Room, Building.
key Optional.
Amap Key.
Applied from ’AutoNavi’ Map API official websitehttps://lbs.amap.com/dev/
city Optional.
Specify the City.
The default value is NULL which will search country-wide. The default value is NULL.
sig Optional.
Digital Signature.
How to use this argument? Please check herehttps://lbs.amap.com/faq/account/key/72
getCoord.individual

Description

Get an individual coordinate from location

Value

Returns a JSON, XML or data.table of results containing detailed geocode information. See https://lbs.amap.com/api/webservice/guide/api/georegeo for more information.

See Also

extractCoord

Examples

## Not run:
library(amapGeocode)

# Before the `getCoord()` is executed, 
# the token should be set by `option(amap_key = 'key')` 
# or set by key argument in `getCoord()`

# Get geocode as a data.table
getCoord("IFS Chengdu")
# Get geocode as a XML
getCoord("IFS Chengdu", output = "XML")

## End(Not run)
Usage

getCoord.individual(
    address,
    key = NULL,
    city = NULL,
    sig = NULL,
    output = "data.table",
    callback = NULL,
    keep_bad_request = TRUE,
    ...
)

Arguments

address  Required.
Structured address information.
Rules: Country/Region, Province/State, City, County/District, Town, Country,
Road, Number, Room, Building.

key    Optional.
Amap Key.
Applied from 'AutoNavi' Map API official website https://lbs.amap.com/dev/

city    Optional.
Specify the City.
The default value is NULL which will search country-wide. The default value is NULL.

sig    Optional.
Digital Signature.
How to use this argument? Please check here https://lbs.amap.com/faq/account/key/72

output    Optional.
Output Data Structure.
Support JSON, XML and data.table. The default value is data.table.

callback    Optional.
Callback Function.
The value of callback is the customized function. Only available with JSON output. If you don’t understand, it means you don’t need it, just like me.

keep_bad_request    Optional.
Keep Bad Request to avoid breaking a workflow, especially meaningful in a batch request

...    Optional.
For compatibility only
Value

Returns a JSON, XML or data.table of results containing detailed geocode information. See [https://lbs.amap.com/api/webservice/guide/api/georegeo](https://lbs.amap.com/api/webservice/guide/api/georegeo) for more information.

---

**getLocation**

*Get location from coordinate*

---

**Description**

Get location from coordinate

**Usage**

```r
getLocation(
  lng, lat, key = NULL, poitype = NULL, radius = NULL, extensions = NULL, roadlevel = NULL, sig = NULL, output = "data.table", callback = NULL, homeorcorp = 0, keep_bad_request = TRUE, ...
)
```

**Arguments**

- **lng**: Required.
  Longitude in decimal
- **lat**: Required.
  Latitude in decimal
- **key**: Optional.
  Amap Key.
  Applied from 'AutoNavi' Map API official website [https://lbs.amap.com/dev/](https://lbs.amap.com/dev/)
- **poitype**: Optional.
  Return nearby POI types.
  When 'extensions = all', this argument makes sense. For detailed poitype type, please refer [https://lbs.amap.com/api/webservice/download](https://lbs.amap.com/api/webservice/download)
- **radius**: Optional.
  Searching radius.
  radius ranges from 0 to 3000, the default value is 1000, unit: meter.
getLocation

extensions

Optional. Return results controller.
- ‘base’: the default value, it only return base information about coordinate.
- ‘all’: it will return nearby POI, road information and cross information.

roadlevel

Optional. Road levels.
When ‘extensions = all’, this argument makes sense.
- ‘roadlevel=0’, return all roads.
- ‘roadlevel=1’, only return main roads.

sig

Optional. Digital Signature. How to use this argument? Please check here https://lbs.amap.com/faq/account/key/72

output

Optional. Output Data Structure. Support JSON, XML and data.table. The default value is data.table.

callback

Optional. Callback Function. The value of callback is the customized function. Only available with JSON output. If you don’t understand, it means you don’t need it, just like me.

homeorcorp

Optional. Optimize the order of returned POI or not.
When ‘extensions = all’, this argument makes sense.
- ‘homeorcorp=0’, do not optimize, by default.
- ‘homeorcorp=1’, home related POIs are first, by default.
- ‘homeorcorp=2’, corporation related POIs are first, by default.

keep_bad_request

Optional. Keep Bad Request to avoid breaking a workflow, especially meaningful in a batch request

... Optional. For compatibility only

Value

Returns a JSON, XML or data.table of results containing detailed reverse geocode information. See https://lbs.amap.com/api/webservice/guide/api/georegeo for more information.

See Also

extractCoord

Examples

```r
## Not run:
library(amapGeocode)

# Before the `getLocation()` is executed,
```
getLocation.individual

Get an individual location from coordinate

Description
Get an individual location from coordinate

Usage

getLocation.individual(
  lng,
  lat,
  key = NULL,
  poitype = NULL,
  radius = NULL,
  extensions = NULL,
  roadlevel = NULL,
  sig = NULL,
  output = "data.table",
  callback = NULL,
  homeorcorp = 0,
  keep_bad_request = TRUE,
  ...
)

Arguments

lng  Required.
    Longitude in decimal
lat  Required.
    Latitude in decimal
key  Optional.
    Amap Key.
    Applied from 'AutoNavi' Map API official website https://lbs.amap.com/dev/
getLocation.individual

poitype  Optional.
Return nearby POI types.
When ‘extensions = all’, this argument makes sense. For detailed poitype type, please refer https://lbs.amap.com/api/webservice/download

radius  Optional.
Searching radius.
radius ranges from 0 to 3000, the default value is 1000, unit: meter.

extensions  Optional.
Return results controller.
‘base’: the default value, it only return base information about coordinate.
‘all’: it will return nearby POI, road information and cross information.

roadlevel  Optional.
Road levels.
When ‘extensions = all’, this argument makes sense.
‘roadlevel=0’, return all roads.
‘roadlevel=1’, only return main roads.

sig  Optional.
Digital Signature.
How to use this argument? Please check here: https://lbs.amap.com/faq/account/key/72

output  Optional.
Output Data Structure.
Support JSON, XML and data.table. The default value is data.table.

callback  Optional.
Callback Function.
The value of callback is the customized function. Only available with JSON output. If you don’t understand, it means you don’t need it, just like me.

homeorcorp  Optional.
Optimize the order of returned POI or not.
When ‘extensions = all’, this argument makes sense.
‘homeorcorp=0’, do not optimize, by default.
‘homeorcorp=1’, home related POIs are first, by default.
‘homeorcorp=2’, corporation related POIs are first, by default.

keep_bad_request  Optional.
Keep Bad Request to avoid breaking a workflow, especially meaningful in a batch request

...  Optional.
For compatibility only

Value

Returns a JSON, XML or data.table of results containing detailed reverse geocode information. See https://lbs.amap.com/api/webservice/guide/api/georegeo for more information.
**num_coord_to_str_loc**

Take longitude and latitude from location string out.

**Description**

Take longitude and latitude from location string out.

**Usage**

```python
num_coord_to_str_loc(lng, lat)
```

**Arguments**

- **lng**
  - Required.
  - Longitude in decimal
- **lat**
  - Required.
  - Latitude in decimal

**Value**

Comma binded coordinate string

---

**str_loc_to_num_coord**

Take longitude and latitude from location string out.

**Description**

Take longitude and latitude from location string out.

**Usage**

```python
str_loc_to_num_coord(str_location)
```

**Arguments**

- **str_location**
  - Required.
  - Location string from response

**Value**

vector contains Longitude and Latitude in numeric
Index

convertCoord, 2, 3, 6
convertCoord.individual, 3

extractAdmin, 4, 10
extractConvertCoord, 5
extractCoord, 6, 13, 16
extractLocation, 7

getAdmin, 5, 8
getAdmin.individual, 10
getCoord, 7, 12
getCoord.individual, 13
getLocation, 8, 15
getLocation.individual, 17

num_coord_to_str_loc, 19

str_loc_to_num_coord, 19