

Package ‘osrm’

November 6, 2015

Type Package

Title Interface Between R and the OpenStreetMap-Based Routing Service
OSRM

Version 1.1

Date 2015-11-06

Description An interface between R and the OSRM API. OSRM is a routing service based on OpenStreetMap data. See <http://project-osrm.org/> for more information. A public API exists but one can run its own instance. This package allows to compute distance (travel time and kilometric distance) between points and travel time matrices.

License GPL-3

LazyData TRUE

Imports jsonlite, sp, RCurl, XML, reshape2, utils

Depends R (>= 2.10)

URL <https://github.com/rCarto/osrm>

BugReports <https://github.com/rCarto/osrm/issues>

Encoding UTF-8

RoxygenNote 5.0.0

NeedsCompilation no

Author Timothée Giraud [cre, aut]

Maintainer Timothée Giraud <timothee.giraud@ums-riate.fr>

Repository CRAN

Date/Publication 2015-11-06 16:36:06

R topics documented:

com	2
osrm	2
osrmTable	3

osrmTableErrors	4
osrmTableOD	5
osrmViaroute	6
osrmViarouteGeom	7

Index	9
--------------	----------

com	<i>Communes Coordinates</i>
-----	-----------------------------

Description

Coordinates of a set of communes in France, Belgium and Luxembourg. Coordinates are in WGS84.

Source

UMS RIATE

osrm	<i>Shortest Paths and Travel Time from OpenStreetMap via an OSRM API</i>
------	--

Description

An interface between R and the OSRM API. OSRM is a routing service based on OpenStreetMap data. A public API exists but one can run its own instance. The package allow to compute distance (travel time and kilometric distance) between points and travel time matrices.

The package relies on the usage of a running OSRM service. By default this service is the OSRM public API (<http://router.project-osrm.org/>). To change the OSRM server, change the `osrm.server` option:

```
options(osrm.server = "http://address.of.the.server/")
```

- `osrmTable` Get travel time matrices between points.
- `osrmTableOD` Get travel time matrices between set of origin points and set of destination points.
- `osrmTableErrors` Detect errors in distance matrices.
- `osrmViaroute` Get travel time and travel distance between two points.
- `osrmViarouteGeom` Get the travel geometry between two points.

`osrmTable`*Get Travel Time Matrices Between Points*

Description

Build and send OSRM API queries to get travel time matrices between points. This function interface the *table* OSRM service.

Usage

```
osrmTable(df, id, x, y)
```

Arguments

<code>df</code>	data frame containing points identifiers, longitudes and latitudes (WGS84).
<code>id</code>	identifier field in df.
<code>x</code>	longitude field in df.
<code>y</code>	latitude field in df.

Value

x A square matrix of time travel (in minutes) is returned.

Note

The public OSRM API does not allow more than 100 locations in distance table query. If you use an other OSRM API service, make sure that more than 100 locations are allowed in table queries (i.e. the "max-table-size" argument, Max. locations supported in distance table query).

See Also

[osrmTableOD](#), [osrmTableErrors](#)

Examples

```
## Not run:  
# Load data  
data("com")  
# Travel time matrix  
distcom <- osrmTable(com[1:50,], id = "comm_id", x = "lon", y = "lat")  
# First 5 rows and columns  
distcom[1:5,1:5]  
  
## End(Not run)
```

osrmTableErrors *Detect Errors in Distance Matrices*

Description

Detect errors in distance matrices.

Usage

```
osrmTableErrors(mat)
```

Arguments

mat A distance matrix.

Value

A list of two vectors of identifier: originErrors and destinationErrors.

Note

An error is detected when at least half of distances are not found for an element.

See Also

[osrmTable](#), [osrmTableOD](#)

Examples

```
## Not run:  
# Load data  
data("com")  
# Travel time matrix  
distcom <- osrmTable(com[1:50,], id = "comm_id", x = "lon", y = "lat")  
# First 5 rows and columns  
distcom[1:5,1:5]  
# Get errors  
osrmTableErrors(distcom)  
  
## End(Not run)
```

`osrmTableOD`*Get Travel Time Matrices Between Origins and Destinations*

Description

Build and send OSRM API queries to get travel time matrices between set of origin points and set of destination points

Usage

```
osrmTableOD(dfo, ido, xo, yo, dfd, idd, xd, yd, limit = 100)
```

Arguments

<code>dfo</code>	data frame containing origin points identifiers, longitudes and latitudes (WGS84).
<code>ido</code>	identifier field in <code>dfo</code> .
<code>xo</code>	longitude field in <code>dfo</code> .
<code>yo</code>	latitude field in <code>dfo</code> .
<code>dfd</code>	data frame containing destination points identifiers, longitudes and latitudes (WGS84).
<code>idd</code>	identifier field in <code>dfd</code> .
<code>xd</code>	longitude field in <code>dfd</code> .
<code>yd</code>	latitude field in <code>dfd</code> .
<code>limit</code>	table query limit of osrm API in use. The default value (100) is the public API limit.

Value

A matrix of time distances (in minutes) between origins and destinations is returned.

Note

This function is a wrapper around `osrmTable`. Multiple calls allow to obtain a matrix between different sets of origins and destinations. A default delay is set between calls (1 sec) to be gentle with the public API. You can modify this delay if you use your own server: `options(osrm.delay = your_delay_value)`.

See Also

[osrmTable](#), [osrmTableErrors](#)

Examples

```
## Not run:
# Load data
data("com")
# Travel time matrix
distcom <- osrmTableOD(dfo = com[1:50,], ido = "comm_id", xo = "lon", yo = "lat",
                      dfd = com[51:100,], idd = "comm_id", xd = "lon", yd = "lat")
# First 5 rows and columns
distcom[1:5,1:5]

## End(Not run)
```

osrmViaroute

Get Travel Time and Travel Distance Between Two Points

Description

Build and send an OSRM API query to get travel time and travel distance between two points. This function interface the *viaroute* OSRM service.

Usage

```
osrmViaroute(xo, yo, xd, yd)
```

Arguments

xo	longitude of the origine point.
yo	latitude of the origine point.
xd	longitude of the destination point.
yd	latitude of the destination point.

Value

A named numeric vector is return. It contains travel time (in minutes) and travel distance (in kilometers).

See Also

[osrmViarouteGeom](#)

Examples

```
## Not run:
# Load data
data("com")
# Time and Distance between 2 points
route <- osrmViaroute(xo = com[1,"lon"], yo = com[1,"lat"],
                     xd = com[15,"lon"], yd = com[15,"lat"])
```

```
# Time travel distance (min)
route[1]
# Travel distance (km)
route[2]
# Mean Speed (km/h)
route[2]/(route[1]/60)

## End(Not run)
```

osrmViarouteGeom *Get the Travel Geometry Between Two Points*

Description

Build and send an OSRM API query to get the travel geometry between two points. This function interface the *viaroute* OSRM service.

Usage

```
osrmViarouteGeom(xo, yo, xd, yd, sp = FALSE, ido = "start", idd = "end")
```

Arguments

xo	longitude of the origine point.
yo	latitude of the origine point.
xd	longitude of the destination point.
yd	latitude of the destination point.
sp	if sp is TRUE the function returns a SpatialLinesDataFrame.
ido	identifier of the origin point (use if sp is TRUE).
idd	identifier of the destination point (use if sp is TRUE).

Value

A data frame is return. It contains the longitudes and latitudes of the travel path between the two points. If sp is TRUE a SpatialLinesDataFrame is return. It contains two fields : identifiers of origine and destination.

See Also

[osrmViaroute](#)

Examples

```
## Not run:
# Load data
data("com")
# Travel path between points
routeGeom <- osrmViarouteGeom(xo = com[1,"lon"], yo = com[1,"lat"],
                             xd = com[15,"lon"], yd = com[15,"lat"])
# Display the path
plot(com[c(1,15),3:4], asp =1, col = "red", pch = 20, cex = 1.5)
points(routeGeom[,2:1], type = "l", lty = 2)
text(com[c(1,15),3:4], labels = com[c(1,15),2], pos = 2)

# Travel path between points - output a SpatialLinesDataFrame
routeGeom2 <- osrmViarouteGeom(xo = com[1,"lon"], yo = com[1,"lat"],
                              xd = com[16,"lon"], yd = com[16,"lat"],
                              sp=TRUE,
                              ido = com[1,"comm_id"],
                              idd = com[16,"comm_id"])
class(routeGeom2)
# Display the path
plot(com[c(1,16),3:4], asp =1, col = "red", pch = 20, cex = 1.5)
plot(routeGeom2, lty = 2, add=TRUE)
text(com[c(1,16),3:4], labels = com[c(1,16),2], pos = 2)

## End(Not run)
```


Index

com, [2](#)

osrm, [2](#)

osrm-package (osrm), [2](#)

osrmTable, [2](#), [3](#), [4](#), [5](#)

osrmTableErrors, [2](#), [3](#), [4](#), [5](#)

osrmTableOD, [2-4](#), [5](#)

osrmViaroute, [2](#), [6](#), [7](#)

osrmViarouteGeom, [2](#), [6](#), [7](#)