

Use OSM API overpass

Jan-Philipp Kolb

12/06/2015

Introduction

- ▶ The package osmar provides good possibilities to download information about small areas from OpenStreetMap.
- ▶ With the this link it is possible to export data directly from OpenStreetMap.

But Sometimes you might want to have information only on one specific topic and you want that information for a bigger area. In this case osmar might not be the perfect solution.

Download specific information

There are several ways to download information from OpenStreetMap. One of these possibilities is the mapquest Xapi Service:

`http://open.mapquestapi.com/xapi/`

To get information, you have to specify a bounding box:

`[left,bottom,right,top]`

- ▶ Defining an adequate bounding box

Alternatives/Example to define bounding box

[left,bottom,right,top]

or:

[east,south,west,north]

or:

[lon.min,lat.min,lon.max,lat.max]

for example Mannheim Seckenheim:

[8.5382,49.4573,8.5808,49.4759]

Download Link

Links to download information from this section
(pubs in this case):

```
"http://open.mapquestapi.com/xapi/api/0.6/node%5B  
amenity=pub%5D%5Bbbox=8.5382,49.4573,8.5808,49.4759%5D"
```

Using R to download information:

The library XML is essential to perform the following steps:

```
install.packages("XML")  
library(XML)
```

Now the command xmlParse can be used to download information:

```
ab <- xmlParse("http://open.mapquestapi.com/xapi/api/0.6/n  
amenity=pub%5D%5Bbbox=-77.041579,38.885851,  
-77.007247,38.900881%5D")  
  
bbox <- "-77.041579,38.885851,-77.007247,38.900881"  
abc <- xmlParse(paste("http://open.mapquestapi.com/xapi/ap  
amenity=pub%5D%5Bbbox=",  
bbox,"%5D",sep=""))
```

Finding a specific place

```
library(ggmap)
# lon - east west
# lat - south north
MAcode1 <- geocode("Mannheim Sandhofen")
MAcode2 <- geocode("Mannheim Rheinau")

MAcode3 <- geocode("Mannheim Schloss")
MAcode4 <- geocode("Mannheim Wallstadt")

# [left,bottom,right,top]
bbox2 <- paste(MAcode3$lon,MAcode2$lat,MAcode4$lon,MAcode1$lon)
abcd <- xmlParse(paste("http://open.mapquestapi.com/xapi/api/v1/json?
                        amenity=pub%5D%5Bbbox=",
                        bbox2,"%5D",sep=""))
```

Getting a bounding box from administrative areas.

- ▶ Administrative Areas from GADM.

```
library(raster)
LUX3 <- getData('GADM', country='LUX', level=3)

LUXT <- LUX3[LUX3@data$NAME_3=="Troisvierges",]
```

```
library(sp)
bLUXT <- bbox(LUXT)
bbox3 <- paste(bLUXT[1,1],bLUXT[2,1],bLUXT[1,2],bLUXT[2,2])
abct <- xmlParse(paste("http://open.mapquestapi.com/xapi/api/amenity=pub%5D%5Bbbox=",
                        bbox3,"%5D",sep=""))
```








Available Map Features

An overview of the different map features is available on the OSM wiki.

Primary features [\[edit\]](#)

Aerialway

Aerialways cover forms of transport for people or goods using aerial wires including cable-cars, chair-lifts and drag-lifts. See [Aerialway](#) for an introduction.

Key	Value	↕	Element	Comment	Rendering	Photo
aerialway	cable_car			cablecar or tramway . Just one or two large cars. The cable forms a loop, but the cars do not loop around, they just move up and down on their own side.		
aerialway	chair_lift			chairlift . Looped cable with a series of single chairs (typically seating two or four people, but can be more). Exposed to the open air. This implies oneway=yes . Any two-way chairlifts should be tagged oneway=no .		

Usage of overpass API

- ▶ An very good alternative for xapi is the overpass API. Here are some use cases available.
- ▶ An example on how to use the overpass API with R.
- ▶ Overpass-turbo is an alternative, but no API.
- ▶ OpenAlfaPlot on how to query Openstreetmap using the Overpass API.
- ▶ About overpass API in German

The overpass API

You can get all streets in Troisdorf with the following command:

```
Troisdorf <- xmlParse("http://www.overpass-api.de/api/int  
area[name=\"Troisdorf\"];  
way(area)[highway][name];out;")
```

Information on how to organise the command line.

Example: Find all cinemas in Bonn which are at most 100m away from bus stops

```
CinemasBonn <- xmlParse("http://www.overpass-api.de/api/int  
area[name=\"Bonn\"];node(area)[high  
node(around:100)[amenity=cinema];ou
```

All restaurants in Ilvesheim:

```
rest_IH <- xmlParse("http://www.overpass-api.de/api/interpr  
area[name=\"Ilvesheim\"];no  
[amenity=restaurant];out;")
```

Find restaurants in Feudenheim (admin_level=9)

```
rest_FD <- xmlParse("http://www.overpass-api.de/api/interpol  
                    area[name=\"Feudenheim\"] ;  
                    [amenity=restaurant];out;")
```

Get the administrative boundaries for the city of Mannheim:

```
boundaryMA <- xmlParse("http://www.overpass-api.de/api/intel  
                        area[name=\"Mannheim\"];way(area)  
                        [boundary=administrative];out;")
```

More than one condition:

```
boundMA <- xmlParse("http://www.overpass-api.de/api/interpol  
area[name=\"Mannheim\"];way(area)  
[boundary=administrative][admin_level=9]
```

Output options

More about the Query language used to get information via the overpass API.

```
geomMA <- xmlParse("http://www.overpass-api.de/api/interpreter?data=area[name=\"Mannheim\"];way(area)[boundary=administrative];out geom;")
```

Five different output formats are possible, a description is available [here](#).

Download csv

```
"http://overpass-api.de/api/interpreter?data=[out:csv(:,:, 'type', :: 'id')];area[name='Feudenheim'];node(area)[amenity=restaurant];out;)"
```

This can be also used to load data directly into the workspace of R:

```
info <- read.csv("http://overpass-api.de/api/interpreter?data=[out:csv(:,:, \"type\", :: \"id\")];area[name=\"Feudenheim\"];node(area)[amenity=restaurant];out;")
```

Download information about Bayern

```
infoBayern <- xmlParse("http://www.overpass-api.de/api/intel  
area[name=\"Bayern\"];way(area)  
[amenity=charging_station];out;")
```

Processment of XML data

One way to process information from a XML is to use the command `xmlToList`

```
root <- xmlRoot(boundMA)
infoAttrs <- xmlSApply(root, xmlAttrs)
```

xpath

- ▶ Xpath Introduction
- ▶ xpath axes
- ▶ Gaston Sanchez - how to work with web data

Resources

- ▶ All streets project
- ▶ Clipping spatial data in R:
- ▶ Other information

More information on getting data from the web with R

- ▶ First off all the following website
`http://gastonsanchez.com/work/webdata/`
- ▶ There is a CRAN task view on Web Technologies and Services
- ▶ Further examples - Streets of ...
- ▶ Converting osm file to shapefile in R