

B1 - Das Arbeiten mit OSM Daten

Jan-Philipp Kolb

22 Oktober 2018

OpenStreetMap Projekt

OpenStreetMap.org ist ein im Jahre 2004 gegründetes internationales Projekt mit dem Ziel, eine freie Weltkarte zu erschaffen. Dafür sammeln wir weltweit Daten über Straßen, Eisenbahnen, Flüsse, Wälder, Häuser und vieles mehr.

<http://www.openstreetmap.de/>

OpenStreetMap (OSM) ist ein kollaboratives Projekt um eine editierbare Weltkarte zu erzeugen.

Wikipedia - OpenStreetMap

OSM Map Features

Amenity

Used to map facilities used by visitors and residents. For example: toilets, telephones, banks, pharmacies, cafes, parking and schools. See the page [Amenities](#) for an introduction on its usage.

Key	Value	Element	Comment	Rendering	Photo
Sustenance					
amenity	bar		Bar is a purpose-built commercial establishment that sells alcoholic drinks to be consumed on the premises. They are characterised by a noisy and vibrant atmosphere, similar to a party and usually don't sell food. See also the description of the tags amenity=pub ; bar ; restaurant for a distinction between these.		
amenity	bbq		BBQ or Barbecue is a permanently built grill for cooking food, which is most typically used outdoors by the public. For example these may be found in city parks or at beaches. Use the tag fuel=* to specify the source of heating, such as fuel=wood ; electric ; charcoal . For mapping nearby table and chairs, see also the tag tourism=picnic_site . For mapping campfires and firepits , instead use the tag leisure=firepit .		
amenity	biergarten		Biergarten or beer garden is an open-air area where alcoholic beverages along with food is prepared and served. See also the description of the tags amenity=pub ; bar ; restaurant . A biergarten can commonly be found attached to a beer hall, pub, bar, or restaurant. In this case, you can use biergarten=yes additional to amenity=pub ; bar ; restaurant .		
amenity	cafe		Cafe is generally an informal place that offers casual meals and beverages, typically, the focus is on coffee or tea. Also known as a coffeehouse ; shop ; bistro or sidewalk cafe . The kind of food served may be mapped with the tags cuisine=* and diet=* . See also the tags amenity=restaurant ; bar ; fast_food .		
amenity	drinking_water		Drinking water is a place where humans can obtain potable water for consumption. Typically, the water is used for only drinking. Also known as a drinking fountain or water tap .		

Objekttypen in OSM

- Es gibt prinipiell drei verschiedene Objekttypen:
-

Download von OpenStreetMap Daten

- <https://mapzen.com/> - Ausschnitte von OpenStreetMap für einzelne Städte (metro extracts)
- Über Geofabrik lassen sich aktuelle Ausschnitte (auch Shapefiles) herunterladen (<http://download.geofabrik.de/>)
- Kartendaten (**openaprs**)

Bei großen Datenmengen

- Hier geht es nur um das Herunterladen kleiner Ausschnitte.
- Wenn größere Datenmengen benötigt werden, sollte man eine Datenbanklösung finden.
- PostgreSQL hat den Vorteil, dass es Open-Source ist.
- Download PostgreSQL
- Hier ist eine Einführung in PostgreSQL zu finden
- Sehr empfehlenswert: Arbeiten mit pgAdmin III
- Beispiel: um Verknüpfung zu einer Datenbank herzustellen - Doppelklick auf den Server in pgAdmin III

PostGIS für PostgreSQL

- **Installieren** der PostGIS Erweiterung:
`CREATE EXTENSION postgis;`

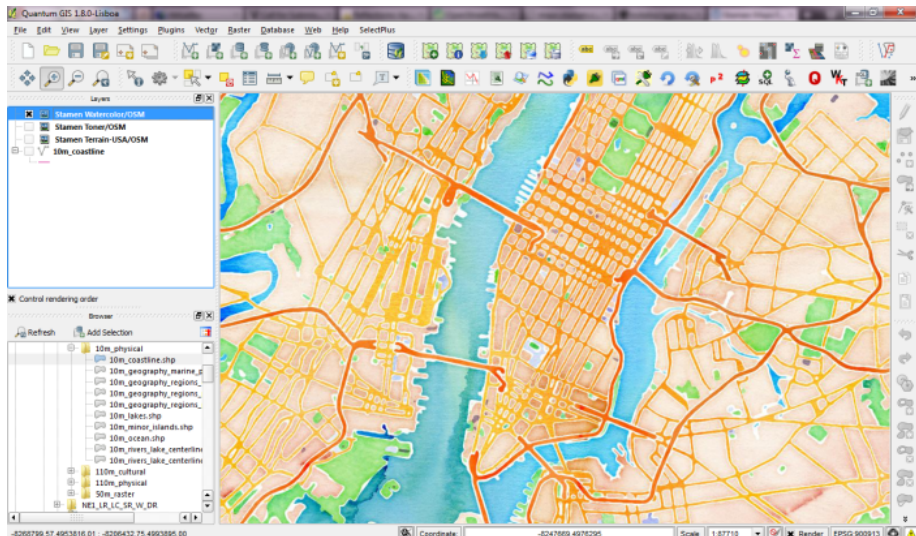
Programm zum Import der OSM Daten in PostgreSQL- osm2pgsql

- Läuft unter Linux deutlich besser
- so könnte bspw. ein Import in PostgreSQL aussehen:

```
osm2pgsql -c -d osmBerlin --slim -C -k berlin-latest.osm.pb
```


Nutze bspw. QGIS um Shapefiles zu extrahieren

- Plugin OpenLayers



OSM Ausschnitte herunterladen

<www.openstreetmap.org/export>

The screenshot displays the OpenStreetMap web interface. At the top, there are links for 'OpenStreetMap', 'Edit', 'History', and 'Export'. Below these is a search bar and a user profile icon. The main map area shows a city street grid with a green bounding box highlighting a specific area. To the left of the map, the 'Export' section is visible, containing a table of coordinates for the bounding box and a list of export options. The coordinates table shows a bounding box of approximately 39.95159 to 39.94715 latitude and -75.17569 to -75.16558 longitude. The export options include 'Overpass API', 'Planet OSM', 'Download', 'Web Extracts', and 'Other Sources'. The map itself is a detailed street view of a city, likely Philadelphia, with various buildings, streets, and landmarks labeled.

OpenStreetMap Edit History Export

GPS Traces User Diaries Copyright Help About dmrgone

Search Where am I? Go

Export

39.95159
-75.17569 39.94715 -75.16558

License

OpenStreetMap data is licensed under the [Open Data Commons Open Database License \(ODbL\)](#).

Export

If the above export fails, please consider using one of the sources listed below:

- [Overpass API](#)
Download this bounding box from a mirror of the OpenStreetMap database
- [Planet OSM](#)
Regularly-updated copies of the complete OpenStreetMap database
- [Download](#)
Regularly-updated extracts of continents, countries, and selected cities
- [Web Extracts](#)
Extracts for major world cities and their surrounding areas
- [Other Sources](#)
Additional sources listed on the OpenStreetMap wiki

Das R-Paket XML - Gaston Sanchez

```
library("XML")
```

Gaston Sanchez - Dataflow



Getting Data from the Web with R Part 4: Parsing XML/HTML Content

Gaston Sanchez

April-May 2014

Content licensed under [CC BY-NC-SA 4.0](#)

Funktionen im XML Paket

Function	Description
xmlName()	name of the node
xmlSize()	number of subnodes
xmlAttrs()	named character vector of all attributes
xmlGetAttr()	value of a single attribute
xmlValue()	contents of a leaf node
xmlParent()	name of parent node
xmlAncestors()	name of ancestor nodes
getSibling()	siblings to the right or to the left
xmlNamespace()	the namespace (if there's one)

Einzelne Objekte finden

<www.openstreetmap.org/export>

OpenStreetMap [Bearbeiten](#) [Chronik](#) [Export](#) [GPS-Tracks](#) [Benutzer-Blogs](#) [Urheberrecht](#) [Hilfe](#) [Über](#) [Anmelden](#) [Registrieren](#)

Suchen [Wie bei osm?](#) [Los](#) [ip](#)

Relation: Berlin (62422) ✕

Reparatur Admin- und PLZ-Grenze Zehlendorf
Nikolassee

Bearbeitet vor etwa ein Monat von [streckenruder](#)
Version #217 Änderungssatz #44753545

Attribute

ISO3166-2	DE-BE
TMC_cld_58_tabled_1: Class	Area
TMC_cld_58_tabled_1: LCLVersion	12.0
TMC_cld_58_tabled_1: LocationCode	265
admin_level	4
alt_name vi	Béclín
boundary	administrative
capital	yes
contact facebook	http://www.facebook.com/Berlin
contact website	http://www.berlin.de
de:antlicher_gemeindeschlüssel	11000000
de:place	city
de:place:note	Kreisfreie Stadt
de:regionalschlüssel	110000000000
geographical_region	Barrim Berliner Unterfrank Talow Nau ener Platte

Beispiel: administrative Grenzen Berlin

Administrative Grenzen für Deutschland

```
url <- "https://api.openstreetmap.org/api/0.6/relation/62422"
```

```
BE <- xmlParse(url)
```

```
BE <- xmlParse("../data/62422.xml")
```

```
-<osm version="0.6" generator="CGImap 0.4.0 (19884 thorn-03.openstreetmap.org)" copyright="OpenStreetMap and contributors" attribution="http://www.openstreetmap.org/copyright"
  license="http://opendatacommons.org/licenses/odbl/1-0/">
  -<relation id="62422" visible="true" version="209" changeset="36072269" timestamp="2015-12-20T19:49:52Z" user="tbicr" uid="278800">
    <member type="node" ref="240109189" role="admin_centre"/>
    <member type="way" ref="50291800" role="outer"/>
    <member type="way" ref="77913336" role="outer"/>
    <member type="way" ref="315222039" role="outer"/>
    <member type="way" ref="77487568" role="outer"/>
    <member type="way" ref="315222038" role="outer"/>
    <member type="way" ref="98035898" role="outer"/>
    <member type="way" ref="77501737" role="outer"/>
```

Das XML analysieren

- Tobi Bosede - Working with XML Data in R

```
xmltop = xmlRoot(BE)
class(xmltop)
```

```
## [1] "XMLInternalElementNode" "XMLInternalNode"
## [3] "XMLAbstractNode"
```

```
xmlSize(xmltop)
```

```
## [1] 1
```

```
xmlSize(xmltop[[1]])
```

```
## [1] 337
```


Nutzung von Xpath

Xpath, the XML Path Language, is a query language for selecting nodes from an XML document.

```
xpathApply(BE,"//tag[@k = 'population']")
```

```
## [[1]]
```

```
## <tag k="population" v="3440441"/>
```

```
##
```

```
## attr(,"class")
```

```
## [1] "XMLNodeSet"
```

Quelle für die Bevölkerungsgröße

```
xpathApply(BE,"//tag[@k = 'source:population']")
```

```
## [[1]]
```

```
## <tag k="source:population" v="http://www.statistik-berlin-b"
```

```
##
```

```
## attr(,"class")
```

```
## [1] "XMLNodeSet"
```

-Statistik Berlin Brandenburg

Etwas überraschend:

```
xpathApply(BE,"//tag[@k = 'name:ta']")
```

```
## [[1]]
```

```
## <tag k="name:ta" v="<U+0BAA><U+0BC6><U+0BB0><U+0BCD><U+0BB2
```

```
##
```

```
## attr(,"class")
```

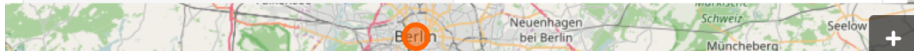
```
## [1] "XMLNodeSet"
```



OpenStreetMap



name:sw	Berlin
name:szl	Berlin
name:ta	பெர்லின்
name:te	बर्लिन
name:tet	Berlin



Geographische Region

```
region <- xpathApply(BE,  
  "//tag[@k = 'geographical_region']")  
# regular expressions  
region[[1]]
```

```
## <tag k="geographical_region" v="Barnim;Berliner Urstromtal;  
  
<tag k="geographical_region"  
  v="Barnim;Berliner Urstromtal;  
  Teltow;Nauener Platte"/>
```

Landkreis



Weiteres Beispiel

```
url2<-"http://api.openstreetmap.org/api/0.6/node/25113879"  
obj2<-xmlParse(url2)  
obj_amenity<-xpathApply(obj2,"//tag[@k = 'amenity']")[[1]]  
obj_amenity
```

```
## <tag k="amenity" v="university"/>
```

Wikipedia Artikel

```
xpathApply(obj2,"//tag[@k = 'wikipedia']")[[1]]
```

```
## <tag k="wikipedia" v="de:Universität Mannheim"/>
```

```
xpathApply(obj2,"//tag[@k = 'wheelchair']")[[1]]
```

```
xpathApply(obj2,"//tag[@k = 'name']")[[1]]
```

Das C und das A

```
url3<-"http://api.openstreetmap.org/api/0.6/node/303550876"  
obj3 <- xmlParse(url3)  
xpathApply(obj3,"//tag[@k = 'opening_hours']")[[1]]  
  
## <tag k="opening_hours" v="Mo-Sa 09:00-20:00; Su,PH off"/>
```


Hin und weg

```
url4<-"http://api.openstreetmap.org/api/0.6/node/25439439"  
obj4 <- xmlParse(url4)  
xpathApply(obj4,"//tag[@k = 'railway:station_category']")[[1]]
```

```
## <tag k="railway:station_category" v="2"/>
```

• Wikipedia Artikel Bahnhofskategorien

Stufe	Bahnsteigkanten	Bahnsteiglänge	Reisende/Tag	Zughalte/Tag
6	1	bis 90 m	bis 49	bis 10
5	2	> 90 bis 140 m	50 bis 299	11 bis 50
4	3 bis 4	> 140 bis 170 m	300 bis 999	51 bis 100
3	5 bis 9	> 170 bis 210 m	1000 bis 9999	101 bis 500
2	10 bis 14	> 210 bis 280 m	10.000 bis 49.999	501 bis 1000
1	ab 15	> 280 m	ab 50.000	ab 1001

Prozent	Kategorie
> 90 %	1
> 80 bis 90 %	2
> 60 bis 80 %	3
> 50 bis 60 %	4
> 40 bis 50 %	5
> 25 bis 40 %	6
bis 25 %	7

Exkurs: Bahnhofskategorien

- **rvest: Easily Harvest (Scrape) Web Pages**

```
library(rvest)
bhfkat<-read_html(
  "https://de.wikipedia.org/wiki/Bahnhofskategorie")
df_html_bhfkat<-html_table(
  html_nodes(bhfkat, "table")[[2]],fill = TRUE)
```

Bahnhofskategorien Übersicht

Stufe	Bahnsteigkanten	Bahnsteiglänge[Anm 1]	Reisende/Tag
(0)	—	—	—
1	01	> 000 bis 090 m	00.000 bis 00.049
2	02	> 090 bis 140 m	00.050 bis 00.299
3	03 bis 04	> 140 bis 170 m	00.300 bis 0.0999
4	05 bis 09	> 170 bis 210 m	01.000 bis 09.999
5	10 bis 14	> 210 bis 280 m	10.000 bis 49.999
6	00i ab 15	> 280 m bis 000	000000 ab 50.000
Gewichtung	20 %	20 %	20 %

Nur fliegen ist schöner

```
url5<-"http://api.openstreetmap.org/api/0.6/way/162149882"  
obj5<-xmlParse(url5)  
xpathApply(obj5,"//tag[@k = 'name']")[[1]]
```

```
## <tag k="name" v="City-Airport Mannheim"/>
```

```
xpathApply(obj5,"//tag[@k = 'website']")[[1]]
```

```
## <tag k="website" v="http://www.flugplatz-mannheim.de/">
```

```
xpathApply(obj5,"//tag[@k = 'iata']")[[1]]
```

```
## <tag k="iata" v="MHG"/>
```

Mehr Beispiele, wie man mit XML Daten umgeht:

- Deborah Nolan - **Extracting data from XML**
- Duncan Temple Lang - **A Short Introduction to the XML package for R**

Noch mehr Informationen

- Web Daten manipulieren
- Tutorial zu xquery
- R und das Web (für Anfänger), Teil II: XML und R
- Gaston Sanchez - **String Manipulation**
- Nutzung, Vor- und Nachteile OSM
- Forschungsprojekte im Zusammenhang mit OpenStreetMap

Referenzen

```
citation("XML")
```

```
##
```

```
## To cite package 'XML' in publications use:
```

```
##
```

```
## Duncan Temple Lang and the CRAN Team (2018). XML: Tools for
```

```
## Parsing and Generating XML Within R and S-Plus. R package
```

```
## version 3.98-1.12. https://CRAN.R-project.org/package=XML
```

```
##
```

```
## A BibTeX entry for LaTeX users is
```

```
##
```

```
## @Manual{,
```

```
## title = {XML: Tools for Parsing and Generating XML Within
```

```
## author = {Duncan Temple Lang and the CRAN Team},
```

```
## year = {2018},
```

```
## note = {R package version 3.98-1.12},
```

```
## url = {https://CRAN.R-project.org/package=XML}
```

Das neuere Paket

```
citation("xml2")
```

```
##
```

```
## To cite package 'xml2' in publications use:
```

```
##
```

```
## Hadley Wickham, James Hester and Jeroen Ooms (2018). xml2
```

```
## XML. R package version 1.2.0.
```

```
## https://CRAN.R-project.org/package=xml2
```

```
##
```

```
## A BibTeX entry for LaTeX users is
```

```
##
```

```
## @Manual{,
```

```
## title = {xml2: Parse XML},
```

```
## author = {Hadley Wickham and James Hester and Jeroen Ooms
```

```
## year = {2018},
```

```
## note = {R package version 1.2.0},
```

```
## url = {https://CRAN.R-project.org/package=xml2}
```

- **Wiki zum Downlaod** von Openstreetmap Daten
- **Openstreetmap Blog**
- Liste möglicher Datenquellen für räumliche Analysen (weltweit und in **Deutschland**)
- **SALB** - Administrative Grenzen

<http://wiki.openstreetmap.org/wiki/SALB>