

# Package ‘maposm’

September 17, 2026

**Type** Package

**Title** Get Map Layers from 'OpenStreetMap'

**Version** 0.5.1

**Description** The 'OpenStreetMap' database provides a wide range of highly detailed geographic layers on a global scale. To obtain synthetic information for cartographic purposes, layers must be selected, simplified, merged, or modified. 'maposm' downloads 'OpenStreetMap' extracts and performs these operations to create a set of composite layers of urban areas, buildings, green spaces, main roads, secondary roads, railways, and water bodies.

**URL** <https://codeberg.org/riatelab/maposm>,  
<https://riatelab.r-universe.dev/maposm>

**BugReports** <https://codeberg.org/riatelab/maposm/issues>

**License** GPL-3

**Depends** R (>= 4.2.0)

**Imports** sf, lwgeom, tools, mapsf, graphics, osmextract

**Suggests** tinytest

**Encoding** UTF-8

**Language** en-US

**Config/roxygen2/version** 8.1.0

**NeedsCompilation** no

**Author** Timothée Giraud [cre, aut] (ORCID:  
<<https://orcid.org/0000-0002-1932-3323>>),  
Matthieu Viry [ctb] (ORCID: <<https://orcid.org/0000-0002-0693-8556>>),  
Louis Laurian [ctb] (ORCID: <<https://orcid.org/0009-0002-7275-5536>>)

**Maintainer** Timothée Giraud <[timothee.giraud@cnrs.fr](mailto:timothee.giraud@cnrs.fr)>

**Repository** CRAN

**Date/Publication** 2026-09-17 13:00:02 UTC

## Contents

|                    |   |
|--------------------|---|
| maposm . . . . .   | 2 |
| om_get . . . . .   | 3 |
| om_map . . . . .   | 4 |
| om_read . . . . .  | 4 |
| om_write . . . . . | 5 |

|              |          |
|--------------|----------|
| <b>Index</b> | <b>6</b> |
|--------------|----------|

---

|        |                            |
|--------|----------------------------|
| maposm | <i>Package description</i> |
|--------|----------------------------|

---

## Description

The OpenStreetMap database provides a wide range of highly detailed geographic layers on a global scale. To obtain synthetic information for cartographic purposes, layers must be selected, simplified, merged, or modified. maposm downloads OpenStreetMap extracts and performs these operations to create a set of composite layers of urban areas, buildings, green spaces, main roads, secondary roads, railways, and water bodies.

[om\\_get](#) is used to download, merge and simplify the geographic layers.

[om\\_map](#) is used to map the layers. Several themes are available to map the layers ("light", "dark", "grey" and "pizza").

[om\\_write](#) can be used to write the layers to a geopackage file.

[om\\_read](#) can be used to read these geopackages.

## Author(s)

**Maintainer:** Timothée Giraud <timothee.giraud@cnrs.fr> ([ORCID](#))

Authors:

- Timothée Giraud <timothee.giraud@cnrs.fr> ([ORCID](#))

Other contributors:

- Matthieu Viry ([ORCID](#)) [contributor]
- Louis Laurian ([ORCID](#)) [contributor]

## See Also

Useful links:

- <https://codeberg.org/riatelab/maposm>
- <https://riatelab.r-universe.dev/maposm>
- Report bugs at <https://codeberg.org/riatelab/maposm/issues>

om\_get

*Get map layers***Description**

Download various OpenStreetMap features to create map layers.

**Usage**

```
om_get(
  x,
  r,
  download_directory = tempdir(),
  force_download = FALSE,
  max_file_size = 20,
  force_building = FALSE,
  quiet = FALSE
)
```

**Arguments**

|                                 |                                                                                                       |
|---------------------------------|-------------------------------------------------------------------------------------------------------|
| <code>x</code>                  | an sf or sfc object, or a couple of coordinates (lon/lat, EPSG:4326)                                  |
| <code>r</code>                  | radius of the extraction if <code>x</code> is an sf POINT or a couple of coordinates                  |
| <code>download_directory</code> | directory to store the file containing raw OpenStreetMap data                                         |
| <code>force_download</code>     | if TRUE, the OpenSteetMap file is updated even if it has already been downloaded.                     |
| <code>max_file_size</code>      | the maximum file size to download without asking in interactive mode, in megabytes (default to 20 MB) |
| <code>force_building</code>     | force the extraction of buildings in zones larger than 20 km <sup>2</sup>                             |
| <code>quiet</code>              | if FALSE, the function prints informative messages                                                    |

**Value**

A list of map layers is returned :

- *zone*, the extraction zone;
- *urban*, urban areas;
- *building*, buildings (if the extraction zone area is below 20 km<sup>2</sup> or `force_building` = TRUE);
- *green*, green spaces;
- *road*, main roads;
- *street*, secondary roads;
- *railway*, railways (line);
- *water*, water bodies

If `x` uses an unprojected CRS (lon/lat, EPSG:4326), ouput uses Web Mercator CRS (EPSG:3857), it uses `x` CRS otherwise.

**Examples**

```
res = om_get(c(-61.070, 14.605), r = 600)
om_map(res, title = "Fort-de-France town centre")
```

---

om\_map

*Map layers*

---

**Description**

Display a map of the downloaded layers.

**Usage**

```
om_map(x, title = "Data from OpenStreetMap", theme = "light")
```

**Arguments**

|       |                                                        |
|-------|--------------------------------------------------------|
| x     | a list of map layers                                   |
| title | map title                                              |
| theme | cartographic theme; "light", "dark", "grey" or "pizza" |

**Value**

Nothing is returned. A map is displayed.

**Examples**

```
res = om_get(c(-61.070, 14.605), r = 600)
om_map(res, title = "Fort-de-France town centre", theme = "light")
```

---

om\_read

*Import a list of map layers*

---

**Description**

Read and import a list of layers.

**Usage**

```
om_read(filename)
```

**Arguments**

|          |                   |
|----------|-------------------|
| filename | a geopackage file |
|----------|-------------------|

**Value**

A list of map layers is returned.

**Examples**

```
res = om_get(c(-61.070, 14.605), r = 600)
om_map(res, title = "Fort-de-France town centre")
om_write(res, file.path(tempdir(), "f2f.gpkg"))
r = om_read(file.path(tempdir(), "f2f.gpkg"))
om_map(r, theme = "grey")
```

---

|          |                                  |
|----------|----------------------------------|
| om_write | <i>Save map layers to a file</i> |
|----------|----------------------------------|

---

**Description**

Write a list a map of layers to a geopackage file.

**Usage**

```
om_write(x, filename)
```

**Arguments**

|          |                                          |
|----------|------------------------------------------|
| x        | a list of map layers produce with om_get |
| filename | a geopackage file                        |

**Value**

The layers are saved to a geopackage file. Nothing is returned.

**Examples**

```
res = om_get(c(-61.070, 14.605), r = 600)
om_map(res, title = "Fort-de-France town centre")
(om_write(res, file.path(tempdir(), "f2f.gpkg")))
```

# Index

maposm, [2](#)

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

om\_get, [2](#), [3](#)

om\_map, [2](#), [4](#)

om\_read, [2](#), [4](#)

om\_write, [2](#), [5](#)