

Chapter 12 Web Map Service Layers

12.1 What are Web Map Service (WMS) Layers?

A [Web Mapping Service \(WMS\)](#) consists of geospatial data hosted through the internet with standards set by the Open Geospatial Consortium (OGS).

A WMS enables the exchange of spatial information and viewing over the web in the form of a map or image to your browser. The most common formats of Web Mapping Services are Web Map Services (WMS), Web Feature Services (WFS), Web Coverage Services (WCS), Web Processing Services (WPS), Web Map Tile Services (WMTS), and Web Coverage Processing Services (WCPS). However, the Web Map Services (WMS) layer is the most used. It offers basic panning, zooming and quick rendering speeds.

12.2 Loading a WMS server

To load a WMS layer into Leaflet, we use the `L.tileLayer.wms` class. As simple as that. So let's set up our Leaflet map. Create a new JavaScript file named `wms_layers.js` and insert the following code:

```
let map = L.map('myMap').setView([-1.295287148, 36.81984753], 7);

let tileLayer = L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
  maxZoom: 19,
  attribution: '&copy; <a href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>',
});
```

We have added an OSM layer because we want to create a layer control widget that also contains WMS layers so that you can switch between an Open Street Map (OSM) layer and a WMS. For this exercise, we shall add two WMS layers.

```
let wmsLayerTopo = L.tileLayer.wms('https://www.gmrt.org/services/mapserver/wms_merc_mask?
  layers: 'topo',
  format: 'image/png'
})

let wmsLayerTopomask = L.tileLayer.wms('https://www.gmrt.org/services/mapserver/wms_merc_ma
  layers: 'topo-mask',
  format: 'image/png'
})
```

12.3 Adding WMS to layer control

Let's insert the above web map layers into a JavaScript object before parsing it to a layer control.

```
var basemaps = {
  OSM: tileLayer,
  Topo: wmsLayerTopo,
  Topo_mask: wmsLayerTopomask,
}
```

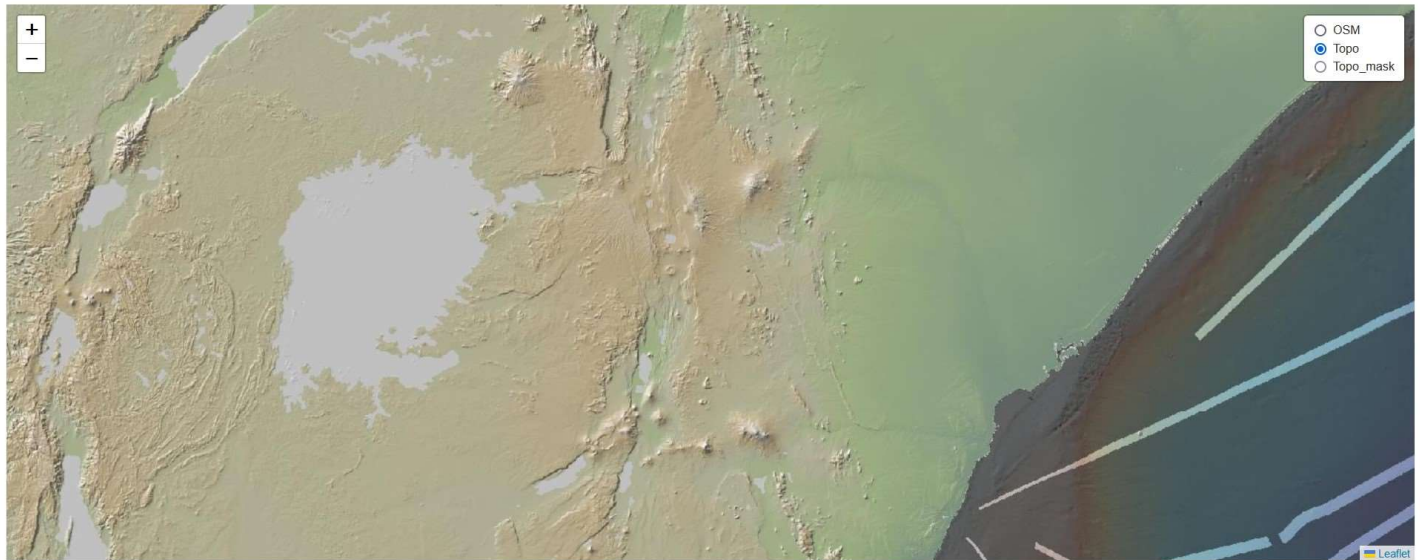
Now let's parse the object into a control. If you did [Chapter 8](#) then this should look familiar.

```
L.control.layers(basemaps).addTo(map);

basemaps.Topo.addTo(map); // To have the Topo as the default map layer
```

The last line `basemaps.Topo.addTo(map)` serves to set up the default layer that will appear when the map is loaded. In this case it is the `Topo` key in `var basemaps`. If this last line is omitted, Leaflet will only show a blank canvas unless one of the radioitems is selected.

```
knitr::include_graphics(rep('D:/gachuhi/my-leaflet/images/wms_layer.jpg'))
```



And that's it!

Even though we wanted to work with African-based WMS layers, of which [Digital Earth Africa](#) provides plenty of them, for some reason this was not possible despite diligently following their documentation. It is quite ironic that they can be easily loaded in Qgis and not Leaflet.

Interested in getting other WMS layers? Go to [Spartineo.com](#).

The files used in this chapter are available [here](#).

12.4 Summary

A Web Mapping Service (WMS) layer in many respects looks and acts like a basemap. Just like Open Street Map and other similar servers, WMS layers host data through the internet. Here are the take homes from this chapter.

- A Web Mapping Service consists of geospatial data hosted through the internet with standards set by the Open Geospatial Consortium (OGS).
- The most common formats of Web Mapping Services are Web Map Services (WMS), Web Feature Services (WFS), Web Coverage Services (WCS), Web Processing Services (WPS), Web Map Tile Services (WMTS), and Web Coverage Processing Services (WCPS).
- The `L.tileLayer.wms` class is used to load WMS layers into Leaflet.

- Just like any other basemap, one can parse WMS layer into Leaflet controls, as seen with `wmsLayerTopo` and `wmsLayerTopomask` .