**URL:**

[**https://phabi.ch/2020/05/06/run-osrm-in-docker-on-windows/**](https://phabi.ch/2020/05/06/run-osrm-in-docker-on-windows/)

**Run OSRM in Docker on Windows**

Posted on [6. May 2020](https://phabi.ch/2020/05/06/run-osrm-in-docker-on-windows/) by [Fabian Leuthold](https://phabi.ch/author/phabi/)



After the OSRM routing service is no longer available on Windows we have a good reason to try to run it in a Docker container. However, there are some hurdles that have to be overcome in order to get your daemon up and running. SInce I’ve gone through this, I write down the necessary steps so others don’t have to google and try & error around with all the settings.

**Install Docker Desktop for Windows**

First, you’ve got to install Docker Desktop for windows. Installation is straight forward and you can get it [here](https://hub.docker.com/editions/community/docker-ce-desktop-windows). Follow the installation steps and start Docker Desktop. After succeeding, you should see the Docker-whale icon in the task tray, which allows you to edit the docker settings.

https://phabi.ch/wp-content/uploads/2020/05/dockertray.pngTask tray showing Docker icon

**Get your OSRM Docker Image**

Now, open the Windows shell (hit Windows-key, type ‘cmd’ and hit ENTER) issue the following command in order to get your desired docker image of the osrm backend:

C:\>docker pull osrm/osrm-backend:latest

This command will fetch the most recent tag of the osrm-backend. If you want to fetch a specific version, use the following command instead, indicating the version of the tag desired:

C:\>docker pull osrm/osrm-backend:v5.22.0

It makes sense to go for a specific version, if you want to be able to reproduce an identical configuration later on. In order to see the available tags/versions of the osrm-backend Docker images, please refer to [this website](https://hub.docker.com/r/osrm/osrm-backend/tags).

With the following command you can display the docker images locally available:

C:\>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

osrm/osrm-backend latest 4ca46ac970dc 4 months ago 100MB

osrm/osrm-backend v5.22.0 daceec677b86 15 months ago 99.5MB

**Hint:**For this tutorial, I’ll use the latest version of the osrm-backend, so I don’t provide any tag-declaration in the subsequent docker commands. If you want to use a specific tag, remember to replace the occurrences of ‘osrm/osrm-backend’ by ‘osrm/osrm-backend:vX.Y.Z’, where X.Y.Z represent the version of your desired tag.

**Get your desired Open Street Map material**

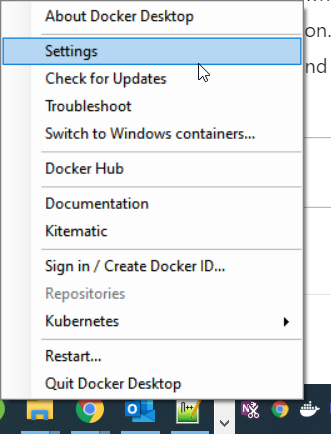
Since routing is not much fun without maps, you now have to download the maps for the country, where you want to do routing calculations for. There are different sources. You might have your own map material source, I always use Open Street Maps. One can download the files here:  
  
<http://download.geofabrik.de/>

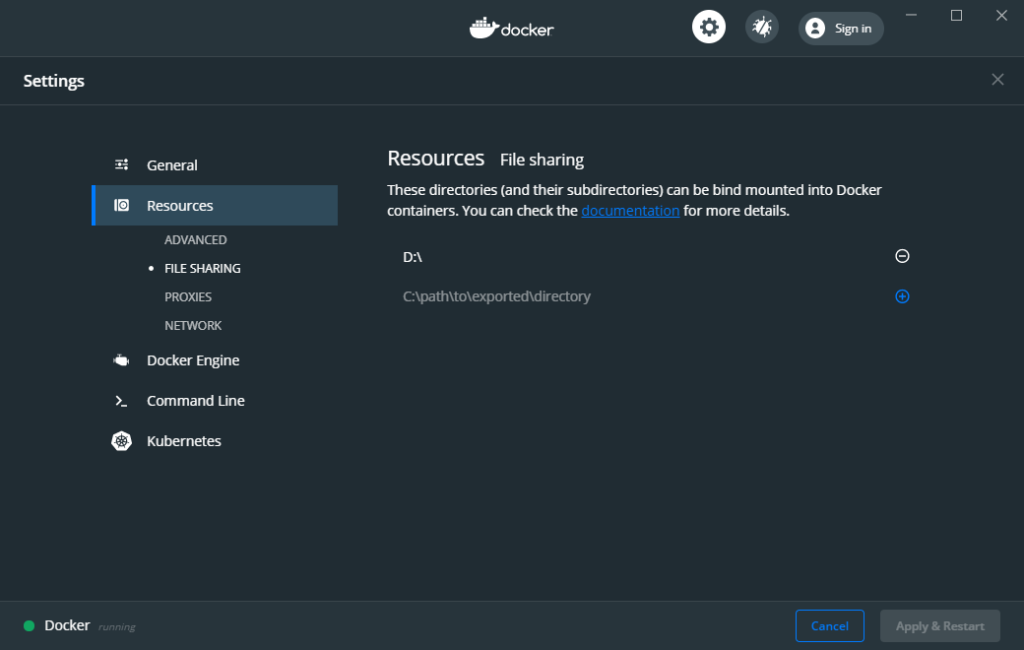
You can either download map-material for a whole continent or for specific countries. Look for an archive file i.e. for your country with the extension ‘\*osm.pbf’. Let’s assume you want to do routing in switzerland, then look for a file named something like ‘switzerland-latest.osm.pbf’. **Notice: Remember the location, where you download the map-files**. On windows, depending on your browser, they are usually stored in the Download-folder.

**Mount a host-directory into your Docker Container**

In order for OSRM to work, you need to provide a folder, where the routing server can extract the map files and preprocess them for operation. Say, you want to map the directory “D:\data\” into your docker container and have it available at “/data”, then perform the following steps:

* Choose Settings from the Docker Task Tray menu

  
Docker Desktop Tasktray menue

* Within the settings-Window choose Resources / ‘FILE SHARING’:
* 

Choose the drive, where you want to provide the ‘data’-directory to be mounted into the docker container. In my situation I would check**‘D’** in order to **map**the **local host’s** (that is my Windows’s) directory**‘D:\data\’ to ‘/data/’ within my docker container.**

Make sure to press the “Apply” button to tell the app, you really mean it.  Now you should get asked for your Windows credentials. If not, choos ‘Reset credentials’, make some fake changes, restore the desired settings and click “Apply” once more (after this step, be aware that the docker desktop has now your credentials and can do, whatever it wants to. You might now want to be sure to run docker desktop only when you really need it and make sure, it is up to date, check for security issues etc.).

* Now copy the map file(s) to your shared folder

In my case, this means moving the file ‘switzerland-latest.osm.pbf’ from ‘Downloads’ to ‘D:\data\’.

Now it’s time to initialize your OSRM routing server!

**Initialize your OSRM Backend Service**

Now that you have configured a shared-folder on your host, you can startup the OSRM Backend to decompress the map-material. In my case for the switzerland-maps, I would have to issue the following three commands:

C:\>docker run -t -v d:/data:/data osrm/osrm-backend osrm-extract -p /opt/car.lua /data/switzerland-latest.osm.pbf

C:\>docker run -t -v d:/data:/data osrm/osrm-backend osrm-contract /data/switzerland-latest.osrm

These commands cause the server to deflate the map files and to prepare Contraction Hierarchies, which are needed for the server to find the routes between given locations. Both commands issue a lot of log-output, telling you about what’s going on behind the scenes. I’ll refrain from dumping the whole output here. As long as you don’t see any error-message and you can see something like “[info] finished preprocessing”, everything should be fine.

Hint: If you should want to provide a different vehicle-profile (\*.lua-file), you can do so by placing it in your /data directory and referencing it accordingly.

**Launch the OSRM Backend Service**

After the Initializsation steps have succcessfully been executed, you can start your OSRM backend service issueing the following command:

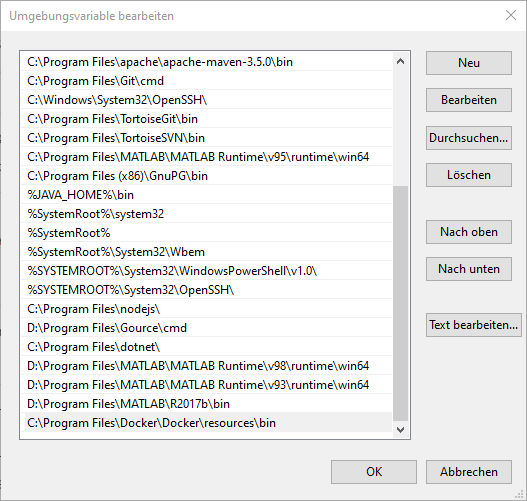
C:\>docker run -t -i -p 5000:5000 -v d:/data:/data osrm/osrm-backend osrm-routed /data/switzerland-latest.osrm

If you haven’t got the docker binary on your path, you might see some message like this:

docker' is not recognized as an internal or external command,

operable program or batch file.

In this case just add the path to the docker binary to your system environment’s path variable:

Add Docker bin-Path to your systems’ ‘Path’ Environment Variable

In case your docker command from above succeeded, this launches the OSRM Backend Service and you should get an output telling you the following:

C:\>docker run -t -i -p 5000:5000 -v d:/data:/data osrm/osrm-backend osrm-routed /data/switzerland-latest.osrm

[info] starting up engines, v5.22.0

[info] Threads: 2

[info] IP address: 0.0.0.0

[info] IP port: 5000

[info] http 1.1 compression handled by zlib version 1.2.8

[info] Listening on: 0.0.0.0:5000

[info] running and waiting for requests

If you see this: Congrats! Your OSRM server is now up and running! You can now try to send a routing request to your server. Open a browser and paset the following string into the adress-bar:  
  
<http://localhost:5000/route/v1/driving/8.724283,47.507603;8.421834,47.316917?steps=true>  
  
Now you should get a description for the route to get from the start location, described by the first lng/lat-pair to the end location, given by the second lng/lat-pair. You might want to change the coordinates to match locations of your country. You can easily determine the geolocations in longitude/latitude (or vice versa) by consulting Google Maps and right-clicking on a location, then choosing ‘what’s here?` from the context menue.