You need to create 2 files to launch a simulation :
- `hosts` is a file containing a list of address to connect to the hosting computers
- `services.json` is a file containing in JSON a description of the services used in the simulation, that you need to create the dummy Clients
e.g. :

```
[
    {"name":"weighting", "alternatives":["fastest"]},
    {"name":"vehicle", "alternatives":["car", "bike"]},
    {"name":"position", "alternatives":[ "53.315130,-6.238099", "53.303086,-6.287210", "53.379100,-6.212726"]}
]
```

In the folder of the graphhopper-client application is a script folder in which you can find 4 bash scripts to init (create), kill (destroy), start (pause) and stop (unpause) the Platform dockers.
The `init_platform.sh` script will also create the `host_ip_list` file that you need to create the dummy Clients. For now, you will need to manually edit the user (set at « diversify ») in the scripts as it has not yet been parameterized.
- `init_platform.sh` takes 2 parameters : the hosts file and the number of containers to start on each host, e.g. hosts 10
- `kill_platform.sh` takes the hosts file as parameter

The graphhopper-client application has two main classes :
`graphhopper.client.demo.Main` and
`graphhopper.client.demo.DummyClientGenerator`

The DummyClientGenerator class starts the client descriptor generator, which allows to create random fake clients for the simulation. It takes 6 parameters : the file containing the adresses of the Platform hosts, the JSON file describing the services, the minimum number of connected Platforms par Client, the max number of connected Platforms par Client, the number of clients to create, and the output folder.
e.g. : `script/host_ip_list services.json 2 8 150 dummies`

The Main class launches the Client side of the graph, and takes 2 parameters: the folder containing the client descriptors and the port used for the websocket (to send results to a dashboard).
e.g. : `dummies/ 8099`

So, in short :
- create a `hosts` file containing the servers IP addresses
- create a `services.json` file containing the sample code provided earlier
- launch the `init_platform.sh` script with the proper parameters (the `host_ip_list` file is created)
- start the DummyClientGenerator main with the proper parameters (the `clientXX.json` files are created)
- start the Main main with the proper parameters