Geocoding (Finding Coordinates):

In [16]:
```
from geopy.geocoders import Nominatim   #OpenStreetMap data to find the geographic coordinates
```

In [22]:
```
geolocator = Nominatim(user_agent="my_geocoder")
location1 = geolocator.geocode("Bahnhofstr. (Hauptbahnhof), 59555 Lippstadt")
location2 = geolocator.geocode("Dr.-Arnold-Hueck-Straße 3, 59557 Lippstadt")
```

In [23]:
```
print("Latitude:", location1.latitude)
print("Longitude:", location1.longitude)
```

```
Latitude: 51.6708806
Longitude: 8.3487947
```

In [24]:
```
print("Latitude:", location2.latitude)
print("Longitude:", location2.longitude)
```

```
Latitude: 51.67384885
Longitude: 8.364541368286515
```

Routing (Shortest Path-Duration & Distance):

In [109…
```
import requests

url = "https://api.openrouteservice.org/v2/directions/driving-car"   #Directions API
params = {                                                           #This dictionary contains the parameters for the API request
    "api_key": "5b3ce3597851110001cf6248f15a51c93d7940b3a89085ac795757b1",  #API key from OpenRouteService
    "start": "8.3487947,51.6708806",          # Point A coordinates
    "end": "8.364541368286515,51.67384885",   # Point B coordinates
}

response = requests.get(url, params=params)   #Sends a GET request
data = response.json()                        #JSON response from the API

Duration = data['features'][0]['properties']['summary']['duration']
Distance = data['features'][0]['properties']['summary']['distance']
```

```
print(duration) #travel time
print(distance) #distance in meters
```

```
200.2
1401.1
```

Map Visualization:

In [70]:
```
import polyline
import folium
from IPython.display import display
```

In [88]:
```
# Get and decode route geometry
geometry = data['features'][0]['geometry']['coordinates']
route_geometry = [(coord[1], coord[0]) for coord in geometry]  # Reverse coordinates for Folium
```

In [110…
```
print(geometry)
```

```
[[8.348457, 51.671043], [8.348572, 51.671163], [8.348645, 51.671203], [8.348662, 51.671232], [8.348962, 51.671208], [8.349338, 5
1.671191], [8.349531, 51.671229], [8.349968, 51.671203], [8.350276, 51.671203], [8.350344, 51.671204], [8.35048, 51.671209], [8.
350927, 51.671241], [8.351199, 51.67128], [8.352488, 51.671529], [8.353087, 51.671639], [8.353728, 51.671751], [8.35429, 51.6718
55], [8.354326, 51.671862], [8.354418, 51.671879], [8.355004, 51.671982], [8.355559, 51.67207], [8.356056, 51.672096], [8.35680
1, 51.672075], [8.357147, 51.672065], [8.357702, 51.672053], [8.35808, 51.672049], [8.358691, 51.672053], [8.359299, 51.672071],
[8.359312, 51.672072], [8.35988, 51.67211], [8.360396, 51.672156], [8.361518, 51.672273], [8.362143, 51.672314], [8.362194, 51.6
72317], [8.362831, 51.672331], [8.363495, 51.672312], [8.363765, 51.672297], [8.364149, 51.672252], [8.364752, 51.672146], [8.36
4817, 51.672092], [8.364969, 51.672061], [8.365116, 51.672099], [8.365169, 51.672146], [8.365187, 51.672203], [8.365162, 51.6722
65], [8.365097, 51.672313], [8.364952, 51.672341], [8.364876, 51.672381], [8.364786, 51.672475], [8.364649, 51.672599], [8.3645
9, 51.672759], [8.364584, 51.672955], [8.364593, 51.673008], [8.364657, 51.673259], [8.364769, 51.673331], [8.364706, 51.67366],
[8.364669, 51.673858]]
```

In [121…
```
# Create the map centered on the route by averaging..
map_center = [(location1.latitude + location2.latitude) / 2,
              (location1.longitude + location2.longitude) / 2]
my_map = folium.Map(location=map_center, zoom_start=16)
```

In [122…
```
# Add markers for start and end
folium.Marker([location1.latitude, location1.longitude], popup="Start").add_to(my_map)
folium.Marker([location2.latitude, location2.longitude], popup="End").add_to(my_map)

# Draw the route
folium.PolyLine(route_geometry, color="blue", weight=2.5, opacity=1).add_to(my_map)
```

```
print(f"Distance: {distance} meters\nDuration: {duration} seconds")
display(my_map)
```

```
Distance: 1401.1 meters
Duration: 200.2 seconds
```