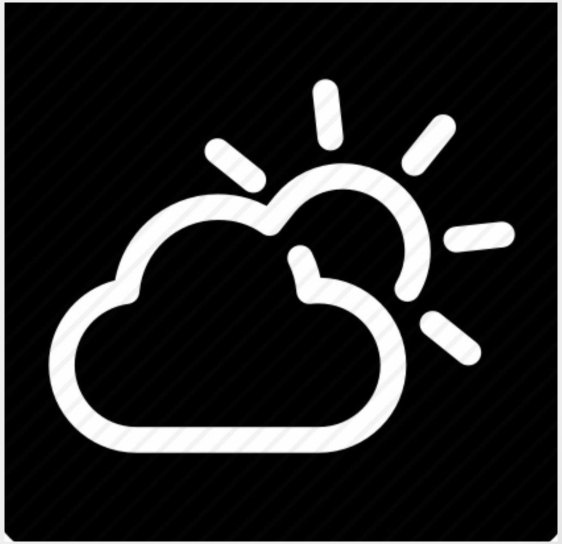
Weather Forecast Application

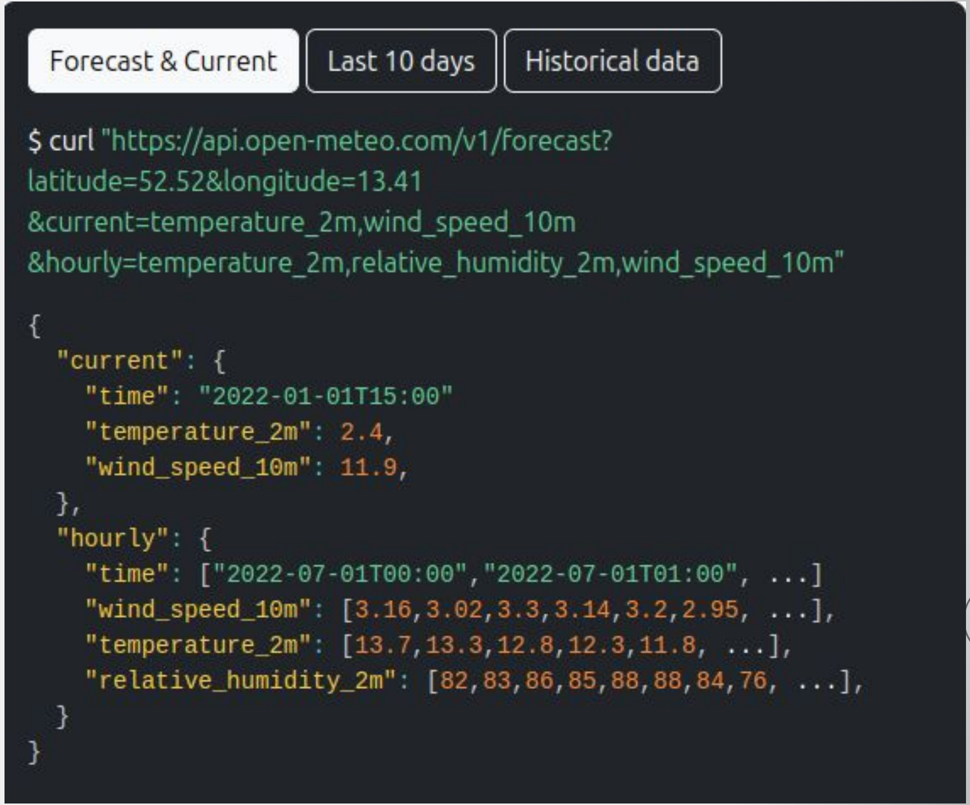
By:
Salma Nasser

Table of contents



Weather API

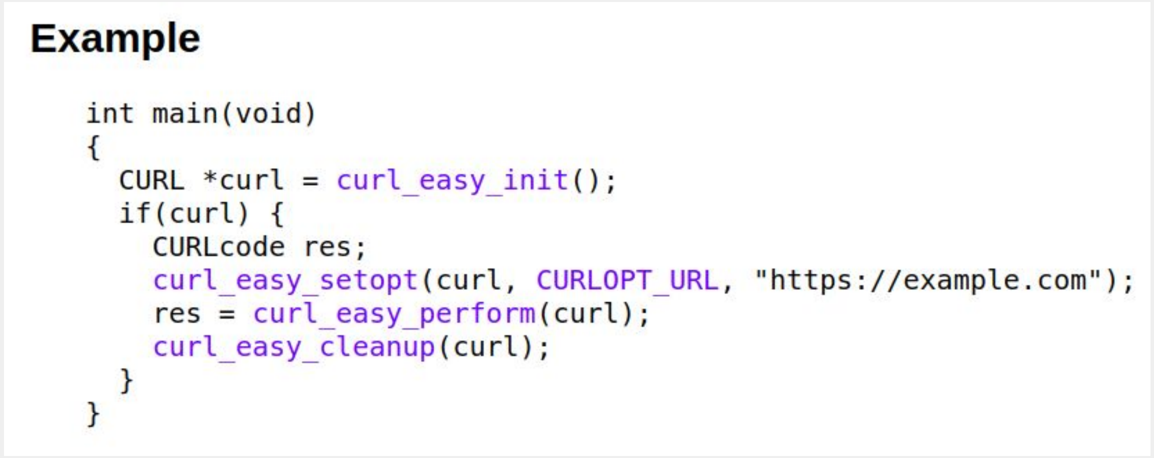
- Open-Meteo API.
- Requesting information through a URL.
- It replies with the information asked in a json format.
- We specify what fields of information we want first.
- To request the weather for a specific location, we have to provide the longitude and latitude of that location.



```
$ curl "https://api.open-meteo.com/v1/forecast?
latitude=52.52&longitude=13.41
&current=temperature_2m,wind_speed_10m
&hourly=temperature_2m,relative_humidity_2m,wind_speed_10m"

{
  "current": {
    "time": "2022-01-01T15:00"
    "temperature_2m": 2.4,
    "wind_speed_10m": 11.9,
  },
  "hourly": {
    "time": ["2022-07-01T00:00","2022-07-01T01:00", ...]
    "wind_speed_10m": [3.16,3.02,3.3,3.14,3.2,2.95, ...],
    "temperature_2m": [13.7,13.3,12.8,12.3,11.8, ...],
    "relative_humidity_2m": [82,83,86,85,88,88,84,76, ...],
  }
}
```

Curl API

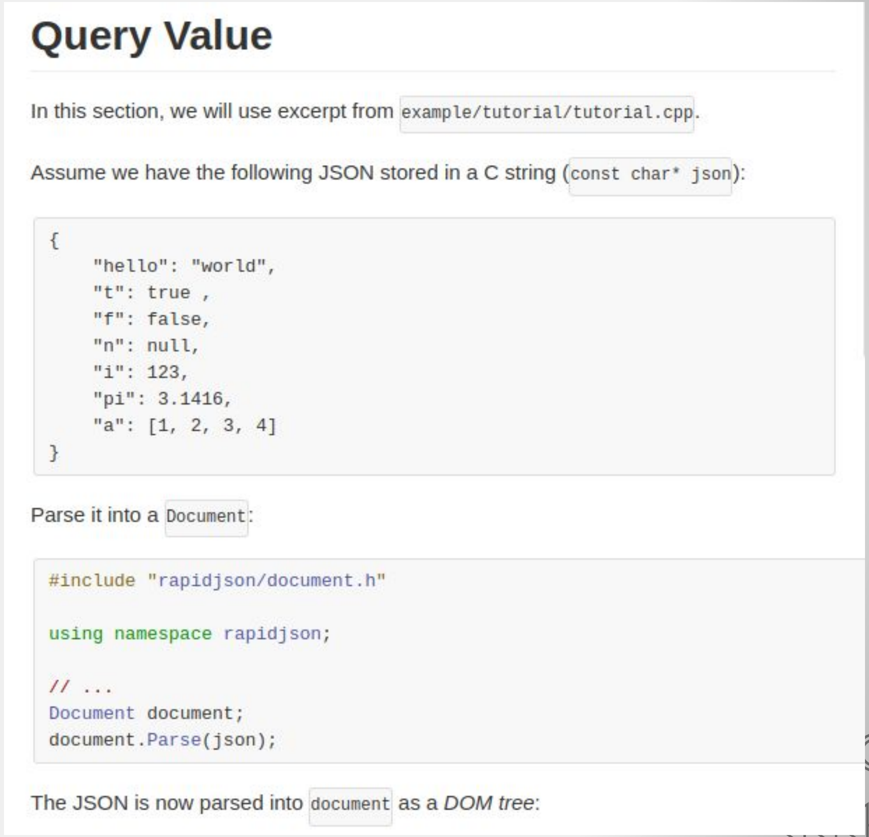

Example

```
int main(void)
{
  CURL *curl = curl_easy_init();
  if(curl) {
    CURLcode res;
    curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");
    res = curl_easy_perform(curl);
    curl_easy_cleanup(curl);
  }
}
```

- The init function used to initialize a transfer through the network and gives back a handler that is used to set options for that single transfer.
- Setopt function takes as an input the transfer handle “curl” and sets the settings for it.
- Perform actually performs the transfer (http request) and by default print the output to the terminal.

RapidJson API

- If I have a character array that have my response from the Curl API in a json format, I can use rapidjson to parse it into a document variable.
- A document variable is like a map but can hold different data types.
- I can check for the datatype before storing the output of indexing.



Query Value

In this section, we will use excerpt from `example/tutorial/tutorial.cpp`.

Assume we have the following JSON stored in a C string (`const char* json`):

```
{
    "hello": "world",
    "t": true ,
    "f": false,
    "n": null,
    "i": 123,
    "pi": 3.1416,
    "a": [1, 2, 3, 4]
}
```

Parse it into a `Document`:

```
#include "rapidjson/document.h"

using namespace rapidjson;

// ...
Document document;
document.Parse(json);
```

The JSON is now parsed into `document` as a *DOM tree*:

QT (Cute)

QComboBox

- I used the ComboBox to display a list of cities the user can choose from.
- Each city have it’s longitude and latitude stored.

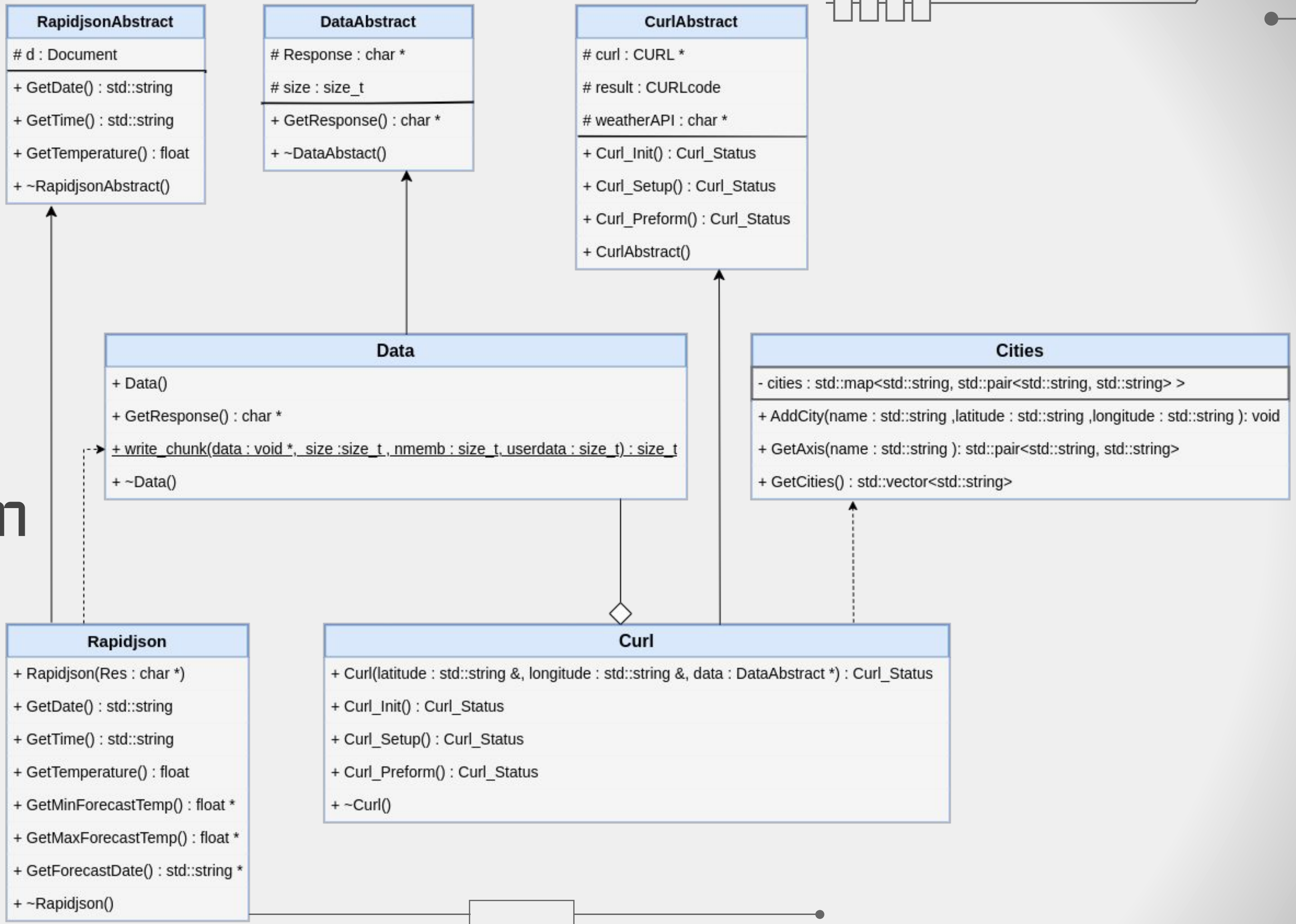
QPushButton

- I used the push button to trigger a refresh for the information displayed based on the chosen governorate

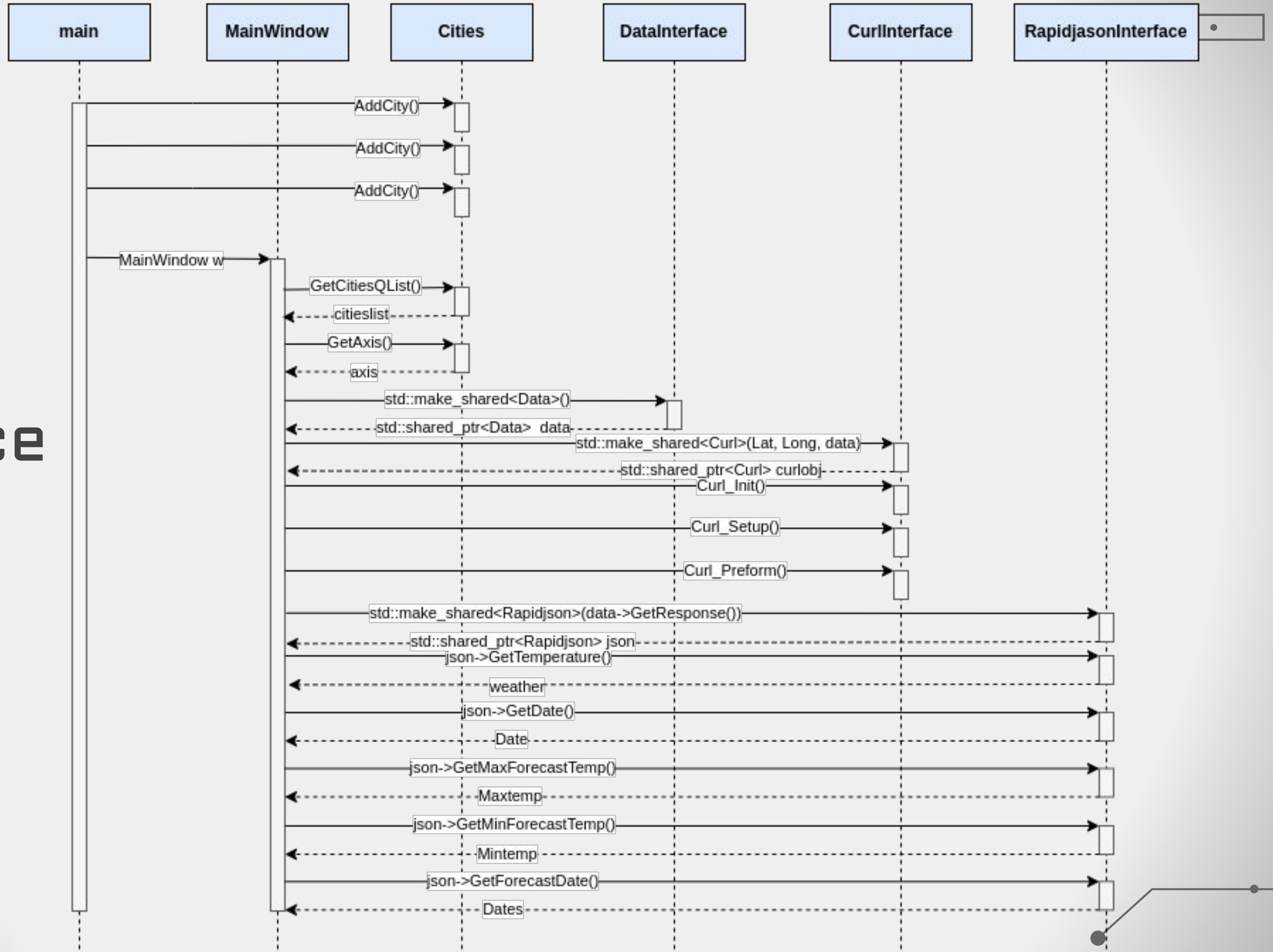
QChart

- QChart is customizable chart for displaying line charts and other types.
- I used a graphicsview to display my QChart as there wasn’t a QChartview.

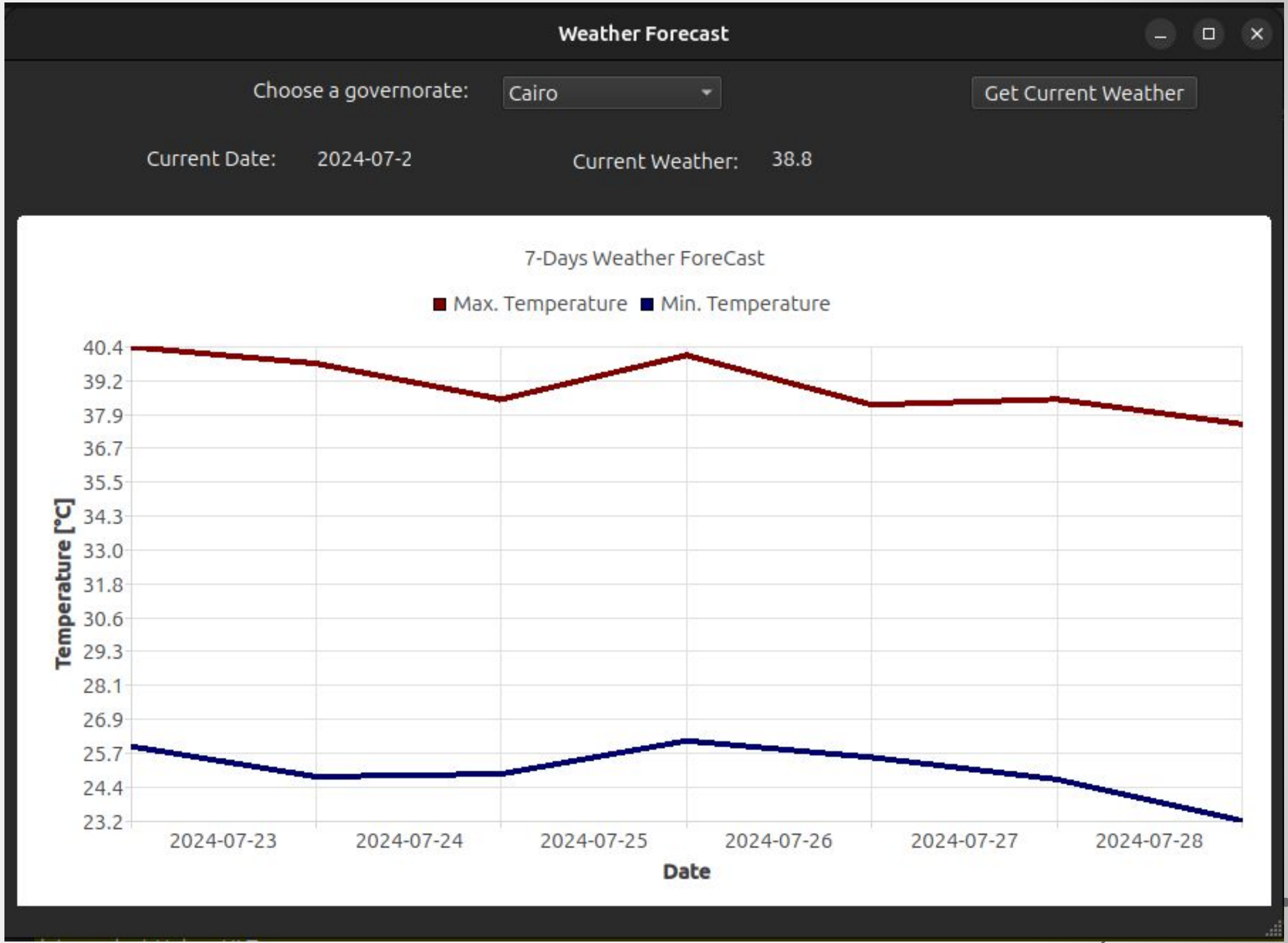
Class Diagram


RapidjsonAbstract
# d : Document
+ GetDate() : std::string
+ GetTime() : std::string
+ GetTemperature() : float
+ ~RapidjsonAbstract()
DataAbstract
# Response : char *
# size : size_t
+ GetResponse() : char *
+ ~DataAbstact()
CurlAbstract
# curl : CURL *
# result : CURLcode
# weatherAPI : char *
+ Curl_Init() : Curl_Status
+ Curl_Setup() : Curl_Status
+ Curl_Preform() : Curl_Status
+ CurlAbstract()
Data
+ Data()
+ GetResponse() : char *
+ write_chunk(data : void *, size :size_t , nmemb : size_t, userdata : size_t) : size_t
+ ~Data()
Cities
- cities : std::map<std::string, std::pair<std::string, std::string> >
+ AddCity(name : std::string ,latitude : std::string ,longitude : std::string): void
+ GetAxis(name : std::string): std::pair<std::string, std::string>
+ GetCities() : std::vector<std::string>
Rapidjson
+ Rapidjson(Res : char *)
+ GetDate() : std::string
+ GetTime() : std::string
+ GetTemperature() : float
+ GetMinForecastTemp() : float *
+ GetMaxForecastTemp() : float *
+ GetForecastDate() : std::string *
+ ~Rapidjson()
Curl
+ Curl(latitude : std::string &, longitude : std::string &, data : DataAbstract *) : Curl_Status
+ Curl_Init() : Curl_Status
+ Curl_Setup() : Curl_Status
+ Curl_Preform() : Curl_Status
+ ~Curl()

Sequence Diagram



Application Look



Current Date: 2024-07-2

Current Weather: 38.8

References

- Weather API: https://open-meteo.com/
- Curl API: https://curl.se/libcurl/c/
- Curl Tutorial 1: https://www.youtube.com/watch?v=mJVchgjkgL8
- Curl Tutorial 2: https://www.youtube.com/watch?v=KSc4zf5t6I4&t=307s
- RapidJson: https://rapidjson.org/
- RapidJson Tutorial: https://www.geeksforgeeks.org/how-to-read-and-parse-json-file-with-rapidjson/
- QT: https://doc.qt.io/qtcreator/index.html
- QT Tutorial 1: https://www.youtube.com/watch?v=cXojtB8vS2E&t=1150s
- QT Tutorial 2: https://www.youtube.com/watch?v=b7_KRIusLP4&list=PLh0cogPqXcJOrXhV2f6rrxcyBx48oQoYs
- QT Tutorial 3: https://youtu.be/MHn3ZTWcyXk?t=811

Do you have any questions?

Thanks!