I found 3 different ways to get the current time from [GitHub](https://gist.github.com/niranjanmalviya/671bb0ca9dcc16093538f0cdf8cbc873).

The first one uses ctime\_s:

#include <chrono>

#include <iostream>

using namespace std;

string CurrentTime\_ctime\_s() {

char curTime[26];

errno\_t error;

chrono::time\_point<chrono::system\_clock> timepoint = chrono::system\_clock::now();

time\_t time\_now\_t = chrono::system\_clock::to\_time\_t(timepoint);

error = ctime\_s(curTime, sizeof(curTime), &time\_now\_t);

if (error != 0) {

return "Error Code : " + error;

}

else {

return curTime;

}

}

int main() {

cout << CurrentTime\_ctime\_s() << endl;

return 0;

}

Second method:

#include <chrono>

#include <iostream>

using namespace std;

string CurrentTime\_strftime() {

tm now\_tm;

errno\_t error;

chrono::time\_point<chrono::system\_clock> timepoint = chrono::system\_clock::now();

time\_t time\_now\_t = chrono::system\_clock::to\_time\_t(timepoint);

error = localtime\_s(&now\_tm, &time\_now\_t);

if (error != 0) {

return "Error Code : " + error;

}

else {

char buf[26];

strftime(buf, 26, "%d-%B-%Y %H:%M:%S", &now\_tm);

return buf;

}

}

int main() {

cout << CurrentTime\_strftime() << endl;

return 0;

}

Third Method:

#include <chrono>

#include <iomanip>

#include <iostream>

#include <sstream>

using namespace std;

string CurrentTime\_put\_time() {

tm now\_tm;

errno\_t error;

chrono::time\_point<chrono::system\_clock> timepoint = chrono::system\_clock::now();

time\_t time\_now\_t = chrono::system\_clock::to\_time\_t(timepoint);

error = localtime\_s(&now\_tm, &time\_now\_t);

if (error != 0) {

return "Error Code : " + error;

}

else {

ostringstream ss;

ss << put\_time(&now\_tm, "%FT%T%z"); // ISO 8601 format

return ss.str();

}

}

int main() {

cout << CurrentTime\_put\_time() << endl;

return 0;

}

The original GitHub code also included <ctime>, but when I tested it, it worked fine without it. I don’t really like the third one as much because I don’t really understand the output format, but besides that these methods seemed good to me.