МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ Федеральное государственное автономное образовательное учреждение высшего образования "КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО" ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ Кафедра компьютерной инженерии и моделирования  
  
​

**Отчёт по лабораторной работе № 6  
по дисциплине "Программирование"**

студента 2 курса группы ПИ-б-о-191(1)  
Спичека Игоря Дмитриевича  
направления подготовки 09.03.04 "Программная инженерия"  
​

|  |  |  |
| --- | --- | --- |
| Научный руководитель старший преподаватель кафедры компьютерной инженерии и моделирования | (оценка) | Чабанов В.В. |

​Симферополь, 2020

**Цель:**

1. Закрепить навыки разработки многофайловыx приложений;
2. Изучить способы работы с API web-сервиса;
3. Изучить процесс сериализации/десериализации данных.

**Ход работы:**

Следуя рекомендациям методички выполнил все поставленные задания и написал работающую программу.

Мой ключ API: bde4a344907083d6f10319df852bcc89

Полный код программыы:

#include <iostream>

#include <fstream>

#include <string>

#include <httplib/httplib.h>

#include <nlohmann/json.hpp>

using namespace httplib;

using json = nlohmann::json;

//SIMFEROPOL - 693805

//API key - bde4a344907083d6f10319df852bcc89

void gen\_response(const Request& req, Response& res) {

//openweathermap.org:80/data/2.5/forecast?id=693805&APPID=bde4a344907083d6f10319df852bcc89&units=metric

httplib::Client cli("api.openweathermap.org", 80);

auto resp = cli.Get("/data/2.5/forecast?id=693805&APPID=bde4a344907083d6f10319df852bcc89&units=metric");

//std::cout << resp->body;

if (resp && resp->status == 200) {

//res.set\_content(resp->body, "text/plain");

json result = json::parse(resp->body);

std::cout << result["cod"];

std::ifstream fin("Weather.html");

std::string output = "could not load template :(";

if (!fin.is\_open()) {

std::cout << "Ошибка при открытии макета." << std::endl;

fin.close();

}

else {

output = "";

char ch;

while (fin.get(ch)) {

output += ch;

}

std::string name = "{city.name}";

output.replace(output.find(name), name.length(), result["city"]["name"]);

std::string dt = "{list.dt}";

std::string temp = "{list.main.temp}";

std::string icon = "{list.weather.icon}";

std::string date\_check = "";

std::cout << result["list"].size()<< std::endl;

std::cout << result["list"][1]["main"]["temp"];

std::cout << result["list"][1]["weather"][0]["icon"].dump() << std::endl;

int amt = result["list"].size();

for (int i = 0; i < result["list"].size(); i++) {

if (output.find(dt) != std::string::npos) {

if (result["list"][i]["dt\_txt"].dump().substr(1,11) != date\_check) {

output.replace(output.find(dt), dt.length(), result["list"][i]["dt\_txt"].dump().substr(1,11));

if (output.find(temp) != std::string::npos) {

output.replace(output.find(temp), temp.length()+1, result["list"][i]["main"]["temp"].dump());

}

if (output.find(icon) != std::string::npos) {

std::string iconid = result["list"][1]["weather"][0]["icon"].dump();

std::string url = "http://openweathermap.org/img/wn/"+(iconid.substr(1,iconid.size()-2)+".png");

output.replace(output.find(icon), icon.length(), url);

}

date\_check = result["list"][i]["dt\_txt"].dump().substr(1,11);

std::cout <<" "<< i << std::endl;

}

}

}

}

res.set\_content(output, "text/html");

}

else {

res.set\_content("somethings wrong, i can feel it", "text/text");

}

}

int main(void) {

setlocale(LC\_ALL, "Russian");

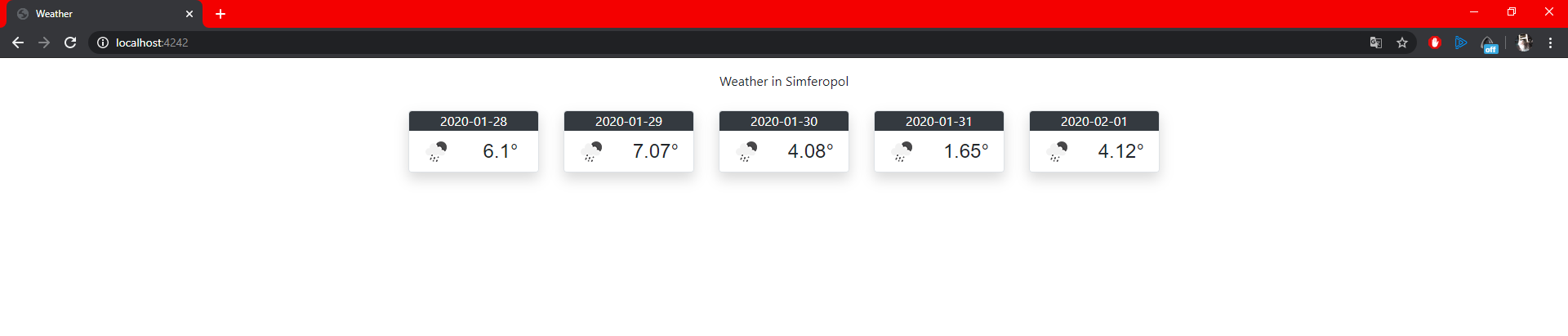
Server svr; // Создаём сервер (пока-что не запущен)

svr.Get("/", gen\_response); // Вызвать функцию gen\_response если кто-то обратиться к корню "сайта"

svr.listen("localhost", 4242); // Запускаем сервер на localhost и порту 1234

}

Скриншот с демонстрацией работы программы:



*Рис.1 Работа программы*

**Вывод**

Я закрепил навыки работы с многофайловыми приложениями, изучил способы работы с API web-сервиса, изучил процесс сериализации/десериализации данных.