**PROIECT CLOUD COMPUTING**

**Aplicatie ce preia de la utilizator un oras si o adresa de email pentru a trimite mail destinatarului cu temperatura din orasul ales.**

**Definitie API:**

API reprezinta acronimul în limba engleza pentru Application Programming Interface, adica interfata de programare a unei aplicatii. O librarie API poate fi privita ca un set de functii puse la dispozitia programatorilor in sensul efectuarii unor anumite operatii sau sarcini.

Un API este o colectie de functii continute in librarii statice sau dinamice, ce pot fi folosite la un moment dat intr-o aplicatie pentru a efectua diverse sarcini.

Librariile API  contin fisierele cu definitiile functiilor utilizate de producatorul respectiv, librariile statice necesare în momentul compilarii, sau librariile dinamice necesare in monentul rularii. API-urile contin de asemenea instructiuni necesare programatorilor in ceea ce priveste apelul functiilor sau compilarea codului sursa.

**Mailgun – Serviciu API pentru mail-uri tranzactionale**

Mailgun API este construit pe baza HTTP. Este REST-ful si are urmatoarele caracteristici:

* Utilizeaza adrese URL resource-oriented si anticipabile.
* Raspunde prin intermediul codurilor de raspuns HTTP standard pentru a indica erori
* Returneaza JSON
* Mailgun are librarii pentru diferite limbaje. Se pot folosi librariile Mailgun, dar si librariile HTTP/REST valabile pentru limbajul de programare utilizat, pentru a face apeluri HTTP catre Mailgun.

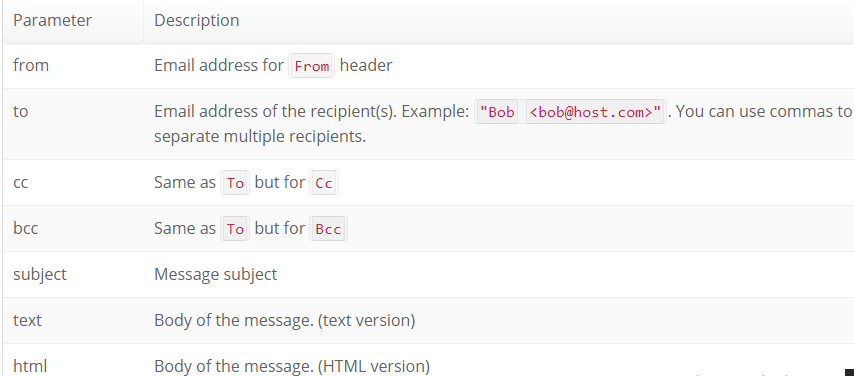
**Autentificare**: la crearea unui account pe site-ul Mailgun, utilizatorul primeste in mod automat o cheie API. Autentificarea la API Mailgun se face prin furnizarea API key-ului in cerere (request). Autentificarea la API se face prin HTTP basic access authentication.

Mailgun returneaza JSON pentru toate apelurile API. JSON nu are un tip de date încorporat, datele sunt transmise ca șiruri codate. Acest format este nativ pentru JavaScript.

**Modul de transmitere a email-urilor folosind Mailgun API:**

Se pot transmite componentele mesajelor, cum a fi „To”, „From”, „Subject”, „HTML”, continutul si atasamente. Mailgun va construi o reprezentare MIME (Multipurpose Internet Mail Extensions) a mesajului si il va trimite.

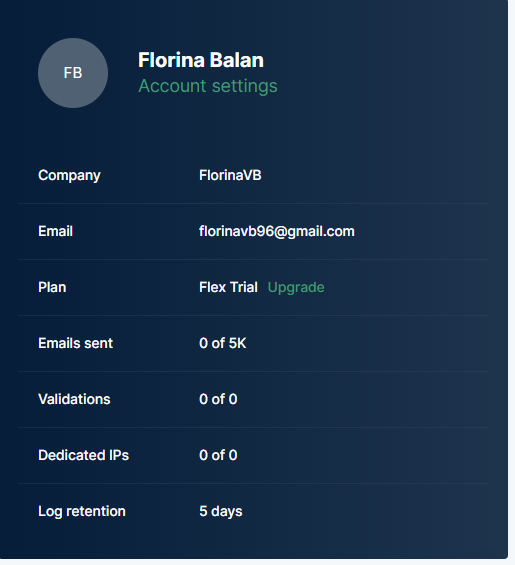
Va trimite un mesaj asamblat din componente. Se pot specifica majoritatea parametrilor de oricate ori se doreste. Acest lucru are sens pentru parametri cum ar fi cc, to sau attachment.



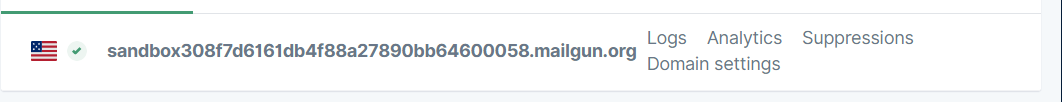


**Pasi urmati:**

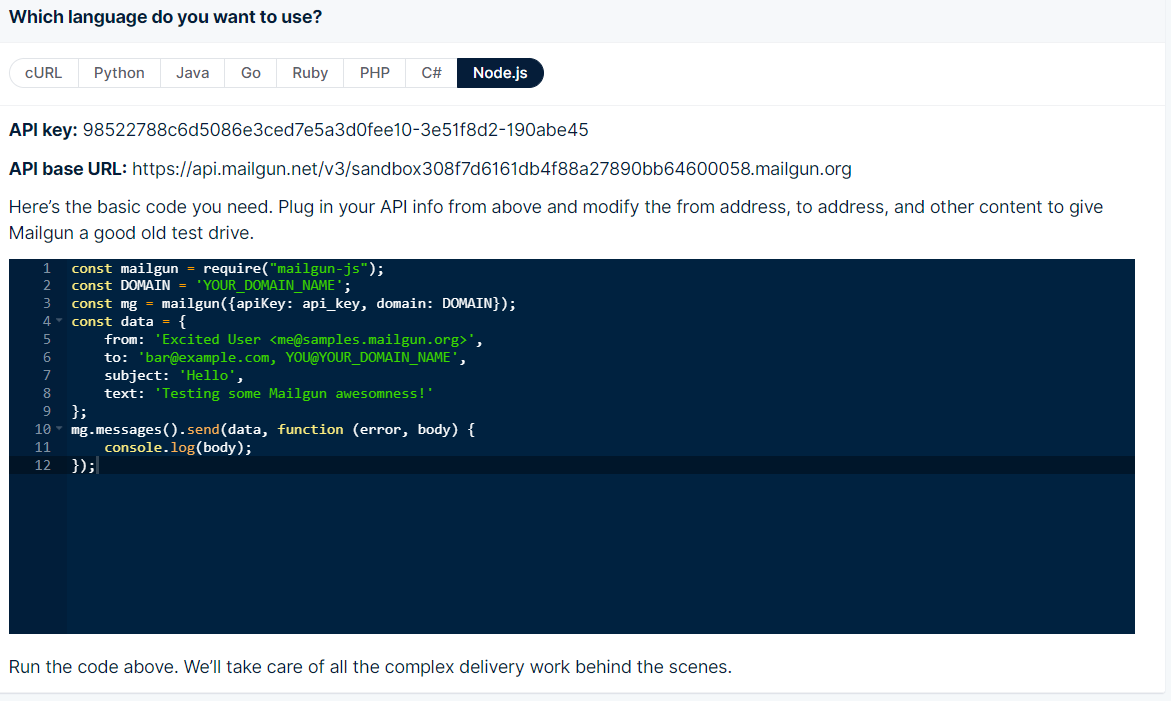
1. Sign Up (creare Account) pe [https://app.mailgun.com](https://app.mailgun.com/)



1. Log in pe site si identificare Domain furnizat de Mailgun pentru testing:



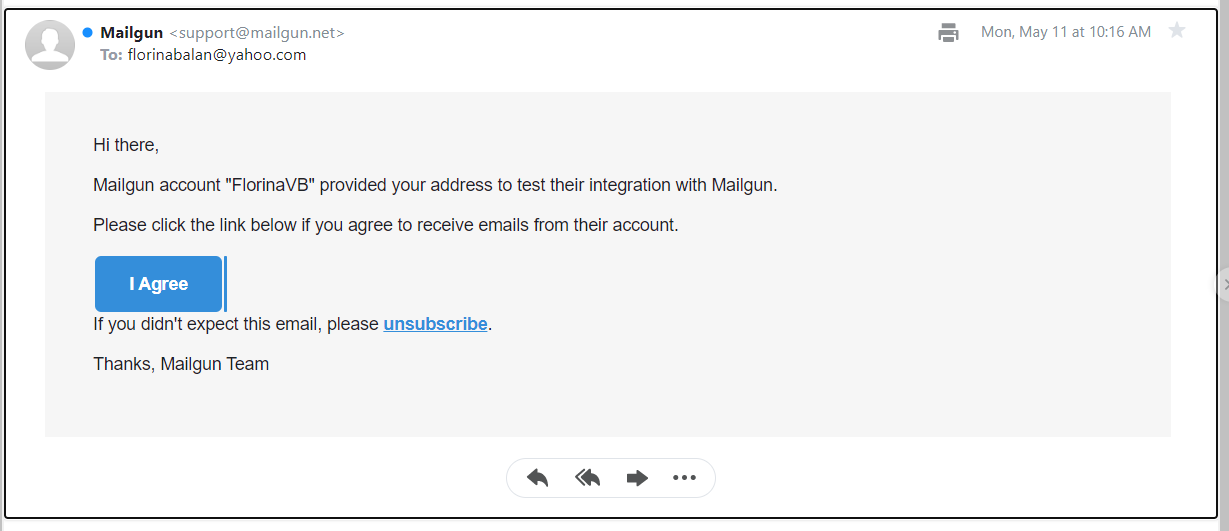
1. Obtinere API key:

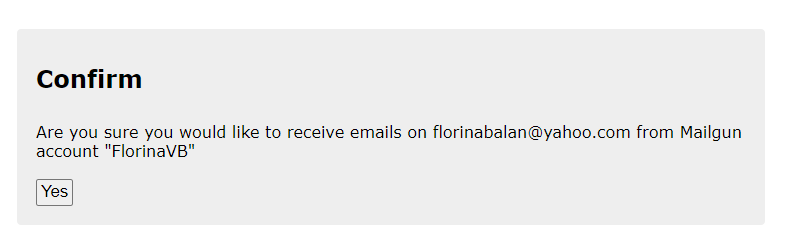


1. Adaugare destinatari autorizati si verificare e-mail

Domeniile Sandbox sunt restrictionate.

Pentru a putea trimite mail cuiva, utilizand domeniul furnizat de Mailgun, destinatarul trebuie adaugat in lista cu destinatari autorizati, urmand sa primeasca un mail pe care trebuie sa il verifice:

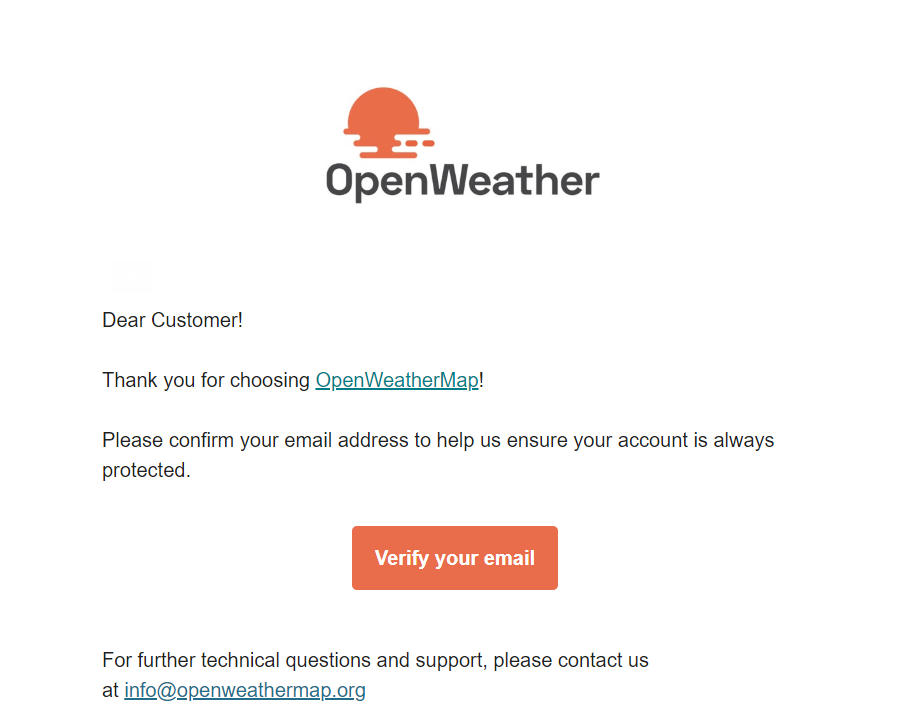


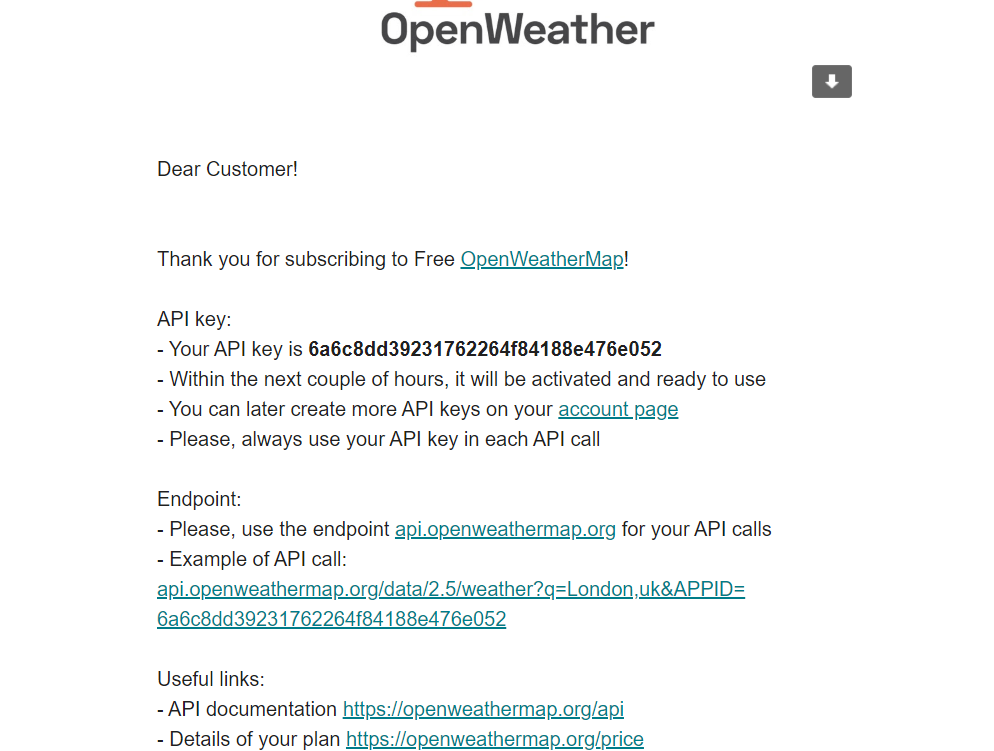


**OpenWeatherMap**

Este unul dintre cei mai importanti furnizori de informatie meteo digitala.

Pasi urmati pentru obtinerea API key-ului:

1. Sign up (creare Account) 
2. Verificare Mail pentru a obtine API key-ul:



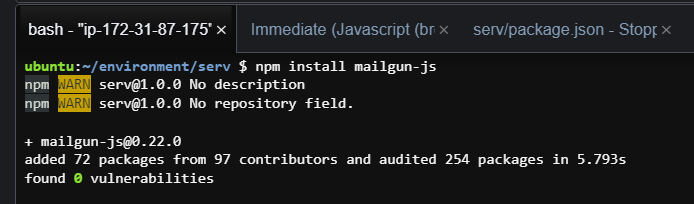
Pentru a apela API key-ul in aplicatia noastra, ne vom folosi de numele orasului (City name).

**Aplicatia propriu-zisa**

Acum ca am obtinut cele doua API keys, putem trece la scrierea script-ului in node.js:

1. Vom deschide Cloud9 IDE si vom crea un script in node.js, denumit „index.js”.

Instalare mailgun-js package in bash:

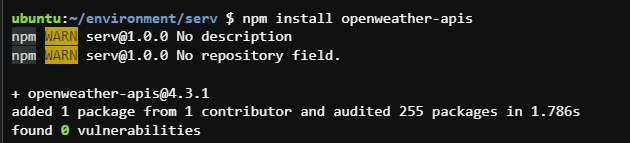


Este un modul Node.js simplu, utilizat pentru interactiunea cu API-ul Mailgun. Este destinat utilizarii in mediul Node.js si nu in Browser.

Acest modul functioneaza furnizand obiecte proxy pentru interactiunea cu diferite resurse prin API-ul Mailgun. Cele mai multe metode primesc un parametru de tip data, care este un obiect Javascript ce contine argumentele pentru Mailgun API. Toate metodele primesc parameteru callback (apel) final cu doi parametri: error si body. Se incearca parsarea body-ului intr-un obiect Javascript si returnarea sa catre apel, pentru o utilizare mai usoara de catre client.

Daca apare o eroare, un obiect Error va fi pasat parametrului error din apel (callback).

Instalare openweather-apis package in node.js:



Cream 3 variabile: doua pentru api keys si una pentru domeniu:

// API keys

var API\_KEY = '98522788c6d5086e3ced7e5a3d0fee10-3e51f8d2-190abe45';

var DOMAIN = 'sandbox308f7d6161db4f88a27890bb64600058.mailgun.org';

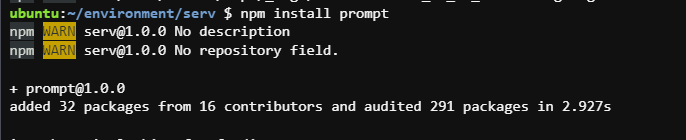
var wAPPID = '6a6c8dd39231762264f84188e476e052';

O cheie API (API key) este un cod transmis de program care apeleaza o interfata de programare a aplicatiilor (API) pentru a identifica programul apelant, dezvoltatorul sau utilizatorul acestuia pe site-ul Web. Cheile API sunt utilizate pentru a urmari si a controla modul in care se utilizeaza API.

//Dependente - Prompt, Mailgun si OpenWeatherMap

//Setam si API keys

//Prompt este folosit pentru introducerea date de la tastatura

var prompt = require('prompt');

var mailgun = require('mailgun-js')({apiKey: API\_KEY, domain: DOMAIN});

var weather = require('openweather-apis');

weather.setAPPID(wAPPID);

In continuare, vom folosi prompt.start() pentru a prelua orasul ai adresa de email introduse de la tastatura de catre utilizatorul aplicatiei:

prompt.start();

prompt.get(['city', 'email'], function (err, result) {

console.log('City: ' + result.city);

console.log('Email: ' + result.email);

var city = result.city;

var emailTo = result.email;

//Console.log scrie in consola

//Setam orasul in OpenWeatherMap

weather.setCity(city);

In final, preluam temperatura pentru orasul selectat din OpenWeatherMap, iar in corpul aceleiasi functii vom seta datele pentru trimiterea email-ului prin Mailgun.

weather.getTemperature(function(err, temp){

//Adaugam temperatura intr-un string ce va fi trimis pe mail

var mailBody = temp + ' Grade celsius in ' + city;

console.log('Temperature: ' + temp);

//Setam datele pentru Mailgun (trimitere email)

const data = {

from: 'Informatii vreme <weather@info.com>',

to: emailTo,

subject: 'Temperatura actuala in orasul ' + city,

text: mailBody

};

//Apelam Mailgun pentru a trimite emailul

mailgun.messages().send(data, (error, body) => {

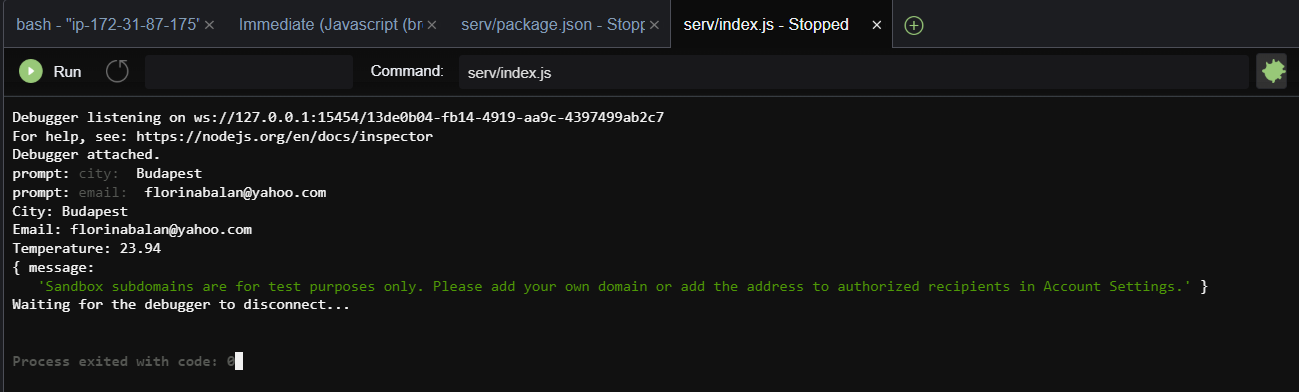
console.log(body);

});

});

});

Rulare cod:



**Fluxul de date:**

Aplicatia noastra trimite request de tip GET temperature catre OpenWeatherOrg API care returneaza raspunsul (temperatura pentru orasul selectat), care este stocat in variabila mailBody (var mailBody = temp + ' Celsius grades in ' + city;).

Apoi, este trimisa o cerere catre Mailgun, care drept raspuns returneaza data (mailgun.messages().send(data, (error, body) => {console.log(body);), si astfel este trimis email-ul.

**Testare aplicatie:**

In cazul de fata, orasul selectat este Budapetsa, iar mail-ul va fi trimis catre [florinabalan@yahoo.com](mailto:florinabalan@yahoo.com) si va returna temperatura.

In momentul trimiterii unei cereri de mesaj, Mailgun plaseaza mesajul intr-o coada de asteptare:

{ id:

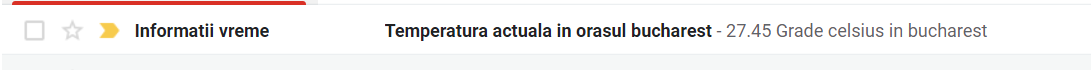
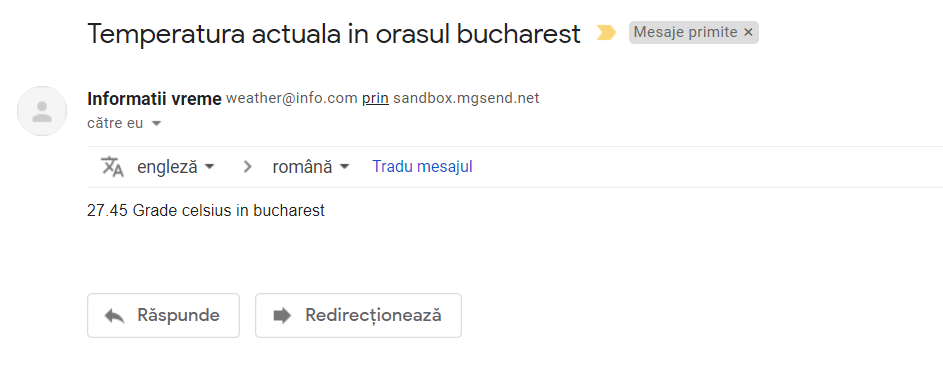
'<20200511100712.1.48CD38578783ADAF@sandbox308f7d6161db4f88a27890bb64600058.mailgun.org>',

message: 'Queued. Thank you.' }





Am testat si pentru adresa de gmail, cu orasul selectat Bucuresti(Bucharest) si am primit urmatorul email:

**Bibliografie si link-uri utile:**

<https://www.npmjs.com/package/mailgun-js>

<https://openweathermap.org>

<https://aws.amazon.com/>

<https://despretot.info/api-application-programming-interface/>

<https://app.mailgun.com/>