УПРАЖНЕНИЕ #1 (POSTGRES)

CREATE SCHEMA IF NOT EXISTS dntbvl04;

CREATE TABLE dntbvl04.weather (

id bigserial not null primary key,

date timestamp,

city varchar(100),

data json

);

INSERT INTO dntbvl04.weather(date, city, data) VALUES (current\_timestamp, 'Kiev', '{"coord":{"lon":30.52,"lat":50.43},"weather":[{"id":800,"main":"Clear","description":"clear sky","icon":"01d"}],"base":"stations","main":{"temp":295.49,"pressure":1011,"humidity":43,"temp\_min":295.15,"temp\_max":296.15},"visibility":10000,"wind":{"speed":5,"deg":310},"clouds":{"all":0},"dt":1499092200,"sys":{"type":1,"id":7358,"message":0.0028,"country":"UA","sunrise":1499046757,"sunset":1499105494},"id":703448,"name":"Kiev","cod":200}' );

УПРАЖНЕНИЕ #2 (REQUEST REMOTE DATA)

В этом упражнении я вынес функцию getWeather в отдельный модуль, а запись в БД выполнил с помощью коллбэка чтобы не нагромождать код.

обозначаем роуты в app.js{

var myAPI = require('./routes/api')(app);

app.use('/api/getWeather', myAPI);

}

var express = require('express');

var router = express.Router();

var getWeather = require('../customModules/getWeather');

module.exports = function(app) {

// GET /

router.get('/:city', function(req, res, next) {

var city = req.params.city;

getWeather.getWeather(city,function callback(data){

if (data) {

var date = new Date();

const db = app.get('db');

db.dntbvl04.weather.insert({date: date, city: city, data: data})

.then(function(result) {

res.send("successful insert to DB");

});

} else {

console.log("error");

}

});

});

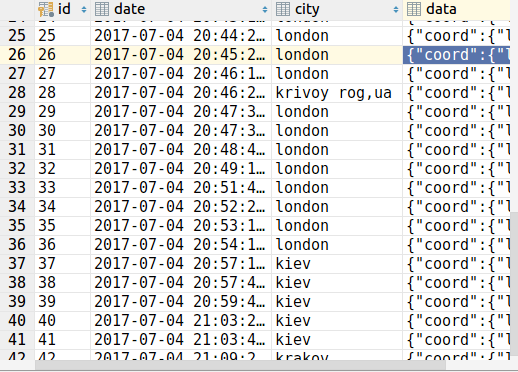
return router;

};

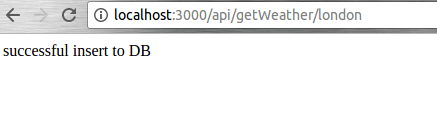
./customModules/getWeather

var rp = require('request-promise');  
var express = require('express');  
var massive = require('massive');  
  
function getWeather(city, callback) {  
 if (!city) {  
 console.log("error: no data");  
 } else {  
  
 if(city) {  
 rp("http://api.openweathermap.org/data/2.5/weather?q="  
 + String(city)  
 + "&appid=c0c6030143a04eff2a48acd37bd81dbb")  
 .then(function (response) {  
 console.log(response);  
 return callback(response);  
 })  
 .catch(function (err) {  
 console.error(err);  
 return;  
 });  
 }else {  
 res.send("Error:" + res);  
 }  
 }  
}  
  
module.exports = {  
 getWeather: getWeather  
};

как результат в БД:



в браузере:



УПРАЖНЕНИЕ #3 (GET DATA)

обозначаем роуты в app.js{

var index = require('./routes/index')(app);

app.use('/', index);

}

index.js:

var express = require('express');

var router = express.Router();

module.exports = function(app) {

/\* GET home page. \*/

app.get('/latest\_data/:id', function(req, res, next) {

const db = app.get('db');

var id = Number(req.params.id);

// Get the data from DB

db.dntbvl04.weather.findOne(id)

.then(function(result) {

console.log(result);

if (result) {

res.render('index', {

title: 'Our Express App with a database access example',

subtitle: 'our example of using EJS and DB',

entry: result

})

} else {

res.send('Error');

}

})

.catch(function(err){

res.send('Error');

})

});

return router;

};

index.ejs:

<html>

<head>

<title><%= title %></title>

<link rel="stylesheet" href="/stylesheets/style.css">

</head>

<body>

<h1><%= title %></h1>

<p>Welcome to <%= subtitle %></p>

<p>Entry ID: <b><%= entry.id %></b></p>

<p>date: <b>[<%= entry.date %>]</b></p>

<p>city: <b>[<%= entry.city %>]</b></p>

<p>data: <b>[<%= JSON.stringify(entry.data, null, '\t') %>]</b></p>

<p>current weather: <ul>

<li><%- JSON.stringify(entry.data.weather[0].description, null, '\t') %></li>

<li>temp: <%- JSON.stringify((entry.data.main.temp - 273).toFixed(1), null, '\t') %> celsius</li>

<li>pressure: <%- JSON.stringify(entry.data.main.pressure, null, '\t') %></li>

<li>humidity: <%- JSON.stringify(entry.data.main.humidity, null, '\t') %> %</li>

</ul>

</p>

</body>

</html>

Результат:

