**Упражнение №2(callback)**

Refactor (Рефакторизируйте?) скрипт (#1) в функцию getWeather(city, callback) с обработчиком callback(error, data).

Напишите скрипт для получения погоды для всех трех

городов параллельно:

nodejs\_learning/serv(new)/script2\_2.js

<http://jsbin.com/wegeye/2/edit>

function getWeather(city, callback) {

var xhr = new XMLHttpRequest();

xhr.open("GET", "http://api.openweathermap.org/data/2.5/weather?q="+String(city)+"&appid=c0c6030143a04eff2a48acd37bd81dbb");

xhr.send();

xhr.onreadystatechange = function() {

if (xhr.readyState != 4) return;

return callback(xhr.status, xhr.responseText);

}

}

getWeather("Kiev", function (error, data) {

if (error != 200) {

alert( "error message" + ': ' + error);

} else {

var weather = JSON.parse(data);

console.log("Country code: " + weather.sys.country + "\n"

+ "City: " + weather.name + "\n" +

"Temperature: " + (weather.main.temp - 273).toFixed(1) + " celsius\n" +

"Humidity: " + weather.main.humidity + "%\n\n");

}

});

getWeather("London", function (error, data) {

if (error != 200) {

alert( "error message" + ': ' + error);

} else {

var weather = JSON.parse(data);

console.log("Country code: " + weather.sys.country + "\n"

+ "City: " + weather.name + "\n" +

"Temperature: " + (weather.main.temp - 273).toFixed(1) + " celsius\n" +

"Humidity: " + weather.main.humidity + "%\n\n");

}

});

getWeather("New York", function (error, data) {

if (error != 200) {

alert( "error message" + ': ' + error);

} else {

var weather = JSON.parse(data);

console.log("Country code: " + weather.sys.country + "\n"

+ "City: " + weather.name + "\n" +

"Temperature: " + (weather.main.temp - 273).toFixed(1) + " celsius\n" +

"Humidity: " + weather.main.humidity + "%\n\n");

}

});

Напишите скрипт для получения погоды для всех трех городов последовательно:

nodejs\_learning/serv(new)/script2\_3.js

<http://jsbin.com/julizum/1/edit>

function getWeather(city, callback) {  
 var xhr = new XMLHttpRequest();  
 xhr.open("GET", "http://api.openweathermap.org/data/2.5/weather?q="+String(city)+"&appid=c0c6030143a04eff2a48acd37bd81dbb");   
 xhr.send();  
   
 xhr.onreadystatechange = function() {  
 if (xhr.readyState != 4) return;  
 return callback(xhr.status, xhr.responseText);  
 }  
  
}  
  
getWeather("Kiev", function callback(error, data) {  
 if (error != 200) {  
 alert( "error message" + ': ' + error);  
 } else {  
 var weather = JSON.parse(data);  
 console.log("Country code: " + weather.sys.country + "\n"  
 + "City: " + weather.name + "\n" +  
 "Temperature: " + (weather.main.temp - 273).toFixed(1) + " celsius\n" +  
 "Humidity: " + weather.main.humidity + "%\n\n");  
 }  
 getWeather("London", function callback(error, data) {  
 if (error != 200) {  
 alert( "error message" + ': ' + error);  
 } else {  
 var weather = JSON.parse(data);  
 console.log("Country code: " + weather.sys.country + "\n"  
 + "City: " + weather.name + "\n" +  
 "Temperature: " + (weather.main.temp - 273).toFixed(1) + " celsius\n" +  
 "Humidity: " + weather.main.humidity + "%\n\n");  
 }  
 getWeather("New York", function callback(error, data) {  
 if (error != 200) {  
 alert( "error message" + ': ' + error);  
 } else {  
 var weather = JSON.parse(data);  
 console.log("Country code: " + weather.sys.country + "\n"  
 + "City: " + weather.name + "\n" +  
 "Temperature: " + (weather.main.temp - 273).toFixed(1) + " celsius\n" +  
 "Humidity: " + weather.main.humidity + "%\n\n");  
 }  
 });  
 });  
});

Вывод в консоль, в обоих случаях идентичен, отличается только последовательность вызова. В script2\_2.j, произойдет параллельный вызов погоды всех трех городов. В script2\_3.js -последовательный.

**Country code: UA**

**City: Kiev**

**Temperature: 14.1 celsius**

**Humidity: 71%**

**Country code: GB**

**City: London**

**Temperature: 22.1 celsius**

**Humidity: 61%**

**Country code: US**

**City: New York**

**Temperature: 29.1 celsius**

**Humidity: 58%**

**Упражнение №3(promise)**

Refactor (Рефакторизируйте?) скрипт (#1) в функцию getWeather(city), которая возвращает Promise

Напишите скрипт для получения погоды для всех трех городов последовательно:

nodejs\_learning/serv(new)/script3\_2.js

<http://jsbin.com/bafumu/1/edit>

var cb;

function getWeather(city) {

return new Promise(function(resolve, reject) {

var xhr = new XMLHttpRequest();

xhr.open("GET", "http://api.openweathermap.org/data/2.5/weather?q="+String(city)+"&appid=c0c6030143a04eff2a48acd37bd81dbb");

xhr.onload = function() {

if (this.status == 200) {

resolve(this.responseText);

let weather = JSON.parse(this.responseText);

console.log("Country code: " + weather.sys.country + "\n"

+ "City: " + weather.name + "\n" +

"Temperature: " + (weather.main.temp - 273).toFixed(1) + " celsius\n" +

"Humidity: " + weather.main.humidity + "%\n\n");

} else {

var error = new Error(this.statusText);

error.code = this.status;

reject(error);

}

};

xhr.onerror = function() {

reject(new Error("Network Error"));

};

xhr.send();

});

}

getWeather("kiev")

.then(function(response) {

return getWeather("london");

})

.then(function(response) {

return getWeather("new york");

})

Напишите скрипт для получения погоды для всех трех городов параллельно используя Promise.all:

nodejs\_learning/serv(new)/script3\_3.js

<http://jsbin.com/mugageh/1/edit>

var cb;

function getWeather(city) {

return new Promise(function(resolve, reject) {

var xhr = new XMLHttpRequest();

var method = "GET";

var url = "http://api.openweathermap.org/data/2.5/weather?q="+String(city)+"&appid=c0c6030143a04eff2a48acd37bd81dbb";

xhr.open(method, url, true); //true for async | false for sync

xhr.onload = function() {

if (this.status == 200) {

resolve(this.responseText);

} else {

var error = new Error(this.statusText);

error.code = this.status;

reject(error);

}

};

xhr.onerror = function() {

reject(new Error("Network Error"));

};

xhr.send();

cb = function(weather) {

console.log("Country code: " + weather.sys.country + "\n"

+ "City: " + weather.name + "\n" +

"Temperature: " + (weather.main.temp - 273).toFixed(1) + " celsius\n" +

"Humidity: " + weather.main.humidity + "%\n\n");

}

});

}

let cities = [

"kiev",

"london",

"new york"

];

Promise.all(cities.map(getWeather))

.then(results => {

for (var i = 0; i < cities.length; i++){

//console.log(results[i]);

let weather = JSON.parse(results[i]);

cb(weather);

}

});

Вывод в консоль аналогичен 2 заданию, за исключением того что в script3\_2.js вызов последовательный, а в script3\_3.js параллельный.

**Country code: UA**

**City: Kiev**

**Temperature: 14.1 celsius**

**Humidity: 71%**

**Country code: GB**

**City: London**

**Temperature: 22.1 celsius**

**Humidity: 61%**

**Country code: US**

**City: New York**

**Temperature: 29.1 celsius**

**Humidity: 58%**

**Упражнение №4(async)**

Используя getWeather(city,callback), напишите скрипт для получения погоды для всех трех городов последовательно используя async.series:

nodejs\_learning/serv(new)/script4\_1.js

<http://jsbin.com/didaxu/2/edit>

function getWeather(callback, city){

var xhr = new XMLHttpRequest();

xhr.open('GET', 'http://api.openweathermap.org/data/2.5/weather?q=' + city + '&appid=a164f3ec180b85f5a0bbbcba58ed2406');

xhr.send();

xhr.onload = function() {

if (this.status == 200) {

var weather = JSON.parse(xhr.responseText);

console.log("Country code: " + weather.sys.country + "\n"

+ "City: " + weather.name + "\n" +

"Temperature: " + (weather.main.temp - 273).toFixed(1) + " celsius\n" +

"Humidity: " + weather.main.humidity + "%\n\n");

callback(null, city);

} else {

callback("Error" + this.status + "for city " + city);

}

};

xhr.onerror = function() {

console.log("Нет доступа в интернет");

process.exit(-1);

};

}

async.series([

function(callback) {

getWeather(callback, "kiev");

},

function(callback) {

getWeather(callback, "london");

},

function(callback) {

getWeather(callback, "new York");

}

], function(error, results) {

if(error)

console.log(error);

else

console.log(results);

});

Напишите скрипт для получения погоды для всех трех городов параллельно используя async.queue(concurrency=3):

nodejs\_learning/serv(new)/script4\_2.js

<http://jsbin.com/tujogap/1/edit>

function getWeather(city, callback){

var xhr = new XMLHttpRequest();

var method = "GET";

var url = 'http://api.openweathermap.org/data/2.5/weather?q=' + city + '&appid=a164f3ec180b85f5a0bbbcba58ed2406';

xhr.open(method, url, true);

xhr.send();

xhr.onload = function() {

if (xhr.readyState == 4) {

callback(xhr.status, xhr.responseText);

} else{

return callback("Error" + this.status + "for city " + city);

}

}

}

function weatherOutput(error, data) {

if(error == 200) {

//console.info(error);

var weather = JSON.parse(data);

console.log("Country code: " + weather.sys.country + "\n"

+ "City: " + weather.name + "\n" +

"Temperature: " + (weather.main.temp - 273).toFixed(1) + " celsius\n" +

"Humidity: " + weather.main.humidity + "%\n\n");

} else{

console.log('error: ' + error);

}

};

var q = async.queue(function (task, callback) {

getWeather(task.city, weatherOutput);

}, 3);

q.push({city: 'kiev'});

q.push({city: 'london'});

q.push({city: 'new york'});

q.drain = function() {

console.log('all items have been processed');

};

Вывод в консоль аналогичен 2 и 3 заданию, в script4\_1.js вызов последовательный, а в script4\_2.js параллельный.

**Country code: UA**

**City: Kiev**

**Temperature: 14.1 celsius**

**Humidity: 71%**

**Country code: GB**

**City: London**

**Temperature: 22.1 celsius**

**Humidity: 61%**

**Country code: US**

**City: New York**

**Temperature: 29.1 celsius**

**Humidity: 58%**