Node.js Exercises

### 1. Write a node.js program for making external http calls.

**var** http = require('http');

http.request({ host: 'www.google.com', method: 'GET', path: "/" }, **function**(response) { response.setEncoding("utf8");

response.on("readable", **function**()

{ console.log(response.read()) }

); }).end();

const http = require("http");

*// notice below, the first parameter to createServer is a callback function!*

const server = http.createServer(**function**(req, res, next) {

*// sending some header info in my response*

res.writeHead(200, { "Content-type": "text/html" });

*// send some data to the client*

res.write("<h1>Hello World!</h1>");

*// end the response*

**return** res.end();

});

*// notice below, another callback function!*

server.listen(3000, **function**() {

console.log("Go to localhost:3000");

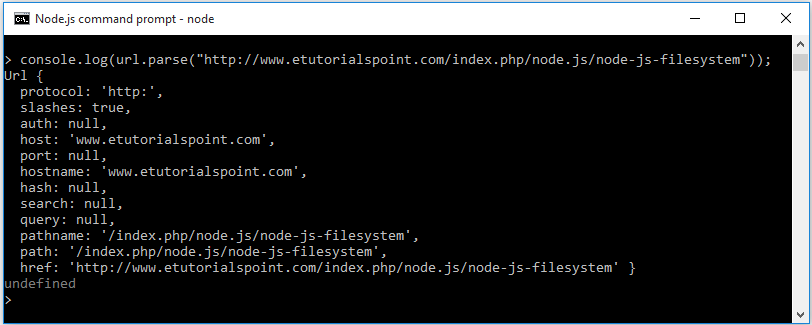
});

### 2. Write a program in node.js to parse the given url.

<http://www.etutorialspoint.com/index.php/nodejs/node-js-filesystem>

console.log(url.parse("http://www.etutorialspoint.com/index.php/nodejs/node-js-filesystem"));

**Output of the above code**



### 3. Write a program to check request header for cookies.

var http = require('http');

var url = require('url');

var server = http.createServer(function(request, response) {

var cookies = request.headers.cookie;

if(!cookies) {

var cookieName = "session";

var cookieValue = "123456";

var expiryDate = new Date();

expiryDate.setDate(expiryDate.getDate() + 1);

var cookieText = cookieName + '=' + cookieValue + ';expires='

+ expiryDate.toUTCString() + ';';

response.setHeader('Set-Cookie', cookieText);

response.writeHead(302, {

'Location': '/'

});

return response.end();

}

cookies.split(';').forEach(function(cookie) {

var m = cookie.match(/(.\*?)=(.\*)$/);

cookies[m[1].trim()] = (m[2] || '').trim();

});

response.end("Cookie set: " + cookies.toString());

}).listen(8080);

### 4. Write a node.js program to replace two or more a's with the letter b on the given string using Regular Expression.

**aaewewedsdewddsxac**

console.log("aaewewedsdewddsxac".replace(new RegExp("[Aa]{2,}"), "b"));

### 5. There is a given object, write node.js program to print the given object's properties, delete the second property and get length of the object.

var user = {

first\_name: "John",

last\_name: "Smith",

age: "38",

department: "Software"

};

### Solution

var user = {

first\_name: "John",

last\_name: "Smith",

age: "38",

department: "Software"

};

console.log(user);

console.log(Object.keys(user).length);

delete user.last\_name;

console.log(user);

console.log(Object.keys(user).length);

**Output of the above code:**

{ **first\_name**: 'John',

last\_name: 'Smith',

age: '38',

department: 'Software' }

4

{ **first\_name**: 'John', age: '38', department: 'Software' }

3

# Node js MySQL Update Data

UPDATE command is used to update data in the table. In this code, the 'SET' clause is used with the UPDATE table to set values for updating columns and the modifying columns are separated by comma. The 'WHERE' clause is used to apply condition to update rows. You can update one or many rows fields together.

### Syntax

**UPDATE** Tablename **SET** column1 = ? [**WHERE** Clause];

### Example

In this given example, we have updated the name of the id #3 of the 'department' table.

**var** mysql = require("mysql");

**var** con = mysql.createConnection({

host: 'localhost',

user: 'root',

password: 'myadmin',

database: 'demo'

});

con.connect(**function**(error){

**if**(error) {

console.log('Error establishing connection to db');

**return**;

}

console.log('Connection Established');

**var** updatedata = "UPDATE department SET name = ? WHERE id = ?";

con.query(updatedata, ["Sales", 3], **function** (error) {

**if**(error) {

**throw**(error);

}

console.log("Data Updated");

});

});