



Node js Modules

#Node JS Notes

Node.js Module:

- Node.js is a light weight framework.
- The core modules include bare minimum functionalities of Node.js.
- These core modules are compiled into its binary distribution and load automatically when Node.js process starts.
- However, you need to import the core module first in order to use it in your application.



Important core modules in Node.js.

| Core Module | Description |
|------------------------------------|---------------------------------------------------------------------------------|
| <u>http</u> | http module includes classes, methods and events to create Node.js http server. |
| <u>url</u> | url module includes methods for URL resolution and parsing. |
| <u>querystring</u> | querystring module includes methods to deal with query string. |
| <u>path</u> | path module includes methods to deal with file paths. |
| <u>fs</u> | fs module includes classes, methods, and events to work with file I/O. |
| <u>util</u> | util module includes utility functions useful for programmers. |



Loading Core Modules:

- In order to use Node.js core or NPM modules, you first need to import it using `require()` function as shown below.
- `var module = require('module_name');`



| OS Module



OS Module

- Node.js OS provides some basic operating-system related utility functions.
- Let's see the list generally used functions or methods.



| Index | Method | Description |
|-------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | os.arch() | This method is used to fetch the operating system CPU architecture. |
| 2. | os.cpus() | This method is used to fetch an array of objects containing information about each cpu/core installed: model, speed (in MHz), and times (an object containing the number of milliseconds the cpu/core spent in: user, nice, sys, idle, and irq). |
| 3. | os.endianness() | This method returns the endianness of the cpu. Its possible values are 'BE' for big endian or 'LE' for little endian. |
| 4. | os.freemem() | This methods returns the amount of free system memory in bytes. |
| 5. | os.homedir() | This method returns the home directory of the current user. |
| 6. | os.hostname() | This method is used to returns the hostname of the operating system. |
| 7. | os.loadavg() | This method returns an array containing the 1, 5, and 15 minute load averages. The load average is a time fraction taken by system activity, calculated by the operating system and expressed as a fractional number. |
| 8. | os.networkinterfaces() | This method returns a list of network interfaces. |
| 9. | os.platform() | This method returns the operating system platform of the running computer i.e.'darwin', 'win32','freebsd', 'linux', 'sunos' etc. |
| 10. | os.release() | This method returns the operating system release. |
| 11. | os.tmpdir() | This method returns the operating system's default directory for temporary files. |
| 12. | os.totalmem() | This method returns the total amount of system memory in bytes. |
| 13. | os.type() | This method returns the operating system name. For example 'linux' on linux, 'darwin' on os x and 'windows_nt' on windows. |
| 14. | os.uptime() | This method returns the system uptime in seconds. |
| 15. | os.userInfo([options]) | This method returns a subset of the password file entry for the current effective user. |



OS Module Example

- `const os=require('os');`
- `console.log("os.freemem(): \n",os.freemem());`
- `console.log("os.homedir(): \n",os.homedir());`
- `console.log("os.hostname(): \n",os.hostname());`
- `console.log("os.endianness(): \n",os.endianness());`
- `console.log("os.loadavg(): \n",os.loadavg());`
- `console.log("os.platform(): \n",os.platform());`
- `console.log("os.release(): \n",os.release());`
- `console.log("os.tmpdir(): \n",os.tmpdir());`
- `console.log("os.totalmem(): \n",os.totalmem());`
- `console.log("os.type(): \n",os.type());`
- `console.log("os.uptime(): \n",os.uptime());`

```
PS D:\lecture\webcode> node .\osdemo.js
os.freemem():
10534465536
os.homedir():
C:\Users\Akash
os.hostname():
DESKTOP-MP4II1D
os.endianness():
LE
os.loadavg():
[ 0, 0, 0 ]
os.platform():
win32
os.release():
10.0.19043
os.tmpdir():
C:\Users\Akash\AppData\Local\Temp
os.totalmem():
17110052864
os.type():
Windows_NT
os.uptime():
264032
PS D:\lecture\webcode> █
```



Path Module



Path Module

- The Node.js path module is used to handle and transform files paths.
- This module can be imported by using the following syntax:
- `var path = require ("path")`



| | | |
|-----|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | <code>path.normalize(p)</code> | It is used to normalize a string path, taking care of '..' and '.' parts. |
| 2. | <code>path.join([path1][, path2][, ...])</code> | It is used to join all arguments together and normalize the resulting path. |
| 3. | <code>path.resolve([from ...], to)</code> | It is used to resolve an absolute path. |
| 4. | <code>path.isabsolute(path)</code> | It determines whether path is an absolute path. an absolute path will always resolve to the same location, regardless of the working directory. |
| 5. | <code>path.relative(from, to)</code> | It is used to solve the relative path from "from" to "to". |
| 6. | <code>path.dirname(p)</code> | It return the directory name of a path. It is similar to the unix dirname command |
| 7. | <code>path.basename(p[, ext])</code> | It returns the last portion of a path. It is similar to the Unix basename command. |
| 8. | <code>path.extname(p)</code> | It returns the extension of the path, from the last '.' to end of string in the last portion of the path. if there is no '.' in the last portion of the path or the first character of it is '.', then it returns an empty string. |
| 9. | <code>path.parse(pathstring)</code> | It returns an object from a path string. |
| 10. | <code>path.format(pathobject)</code> | It returns a path string from an object, the opposite of path.parse above. |



Path Example

- `var path = require("path");`
- `// Normalization`
- `console.log('normalization : ' + path.normalize('/akash/..'));`
- `// Join`
- `console.log('joint path : ' + path.join('/test', 'technolabs', 'node/newfolder', 'tab', '..'));`
- `// Resolve`
- `console.log('resolve : ' + path.resolve('path_example.js'));`
- `// Extension`
- `console.log('ext name: ' + path.extname('path_example.js'));`



Query String Module

- The Node.js Query String provides methods to deal with query string. It can be used to convert query string into JSON object and vice-versa.
- To use query string module, you need to use **require('querystring')**.





| Method | Description |
|-----------------------------------------------------------------|-----------------------------------------|
| <code>querystring.parse(str[, sep][, eq][, options])</code> | converts query string into JSON object. |
| <code>querystring.stringify(obj[, sep][, eq][, options])</code> | converts JSON object into query string. |



parse() Example

- `querystring = require('querystring');`
- `const obj1=querystring.parse('name=akash&company=technolabs');`
- `console.log(obj1);`

```
JS qsdemo.js > [?] obj1
1  querystring = require('querystring');
2  const obj1=querystring.parse('name=akash&company=technolabs');
3  console.log(obj1);
4  [?]
5
6

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\lecture\webcode> node .\qsdemo.js
[Object: null prototype] { name: 'akash', company: 'technolabs' }
PS D:\lecture\webcode> [?]
```



stringify() Example

- `querystring = require('querystring');`
- `const qs1=querystring.stringify({name:'akash',company:'technolabs'});`
- `console.log(qs1);`

```
JS qsdemo.js > ...
1  querystring = require('querystring');
2  const qs1=querystring.stringify({name:'akash',company:'technolabs'});
3  console.log(qs1);
4
5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
D:\lecture\webcode>node qsdemo.js
name=akash&company=technolabs

D:\lecture\webcode>
```



Demo

```
JS qsdemo.js > ...
1  //Example 1
2  querystring = require('querystring');
3  const obj1=querystring.parse('name=akash&company=technolabs');
4  console.log(obj1);
5  //Example 2
6  querystring = require('querystring');
7  const qs1=querystring.stringify({name:'akash',company:'technolabs'});
8  console.log(qs1);
9
10
11
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

D:\lecture\webcode>node qsdemo.js
[Object: null prototype] { name: 'akash', company: 'technolabs' }
name=akash&company=technolabs

D:\lecture\webcode>

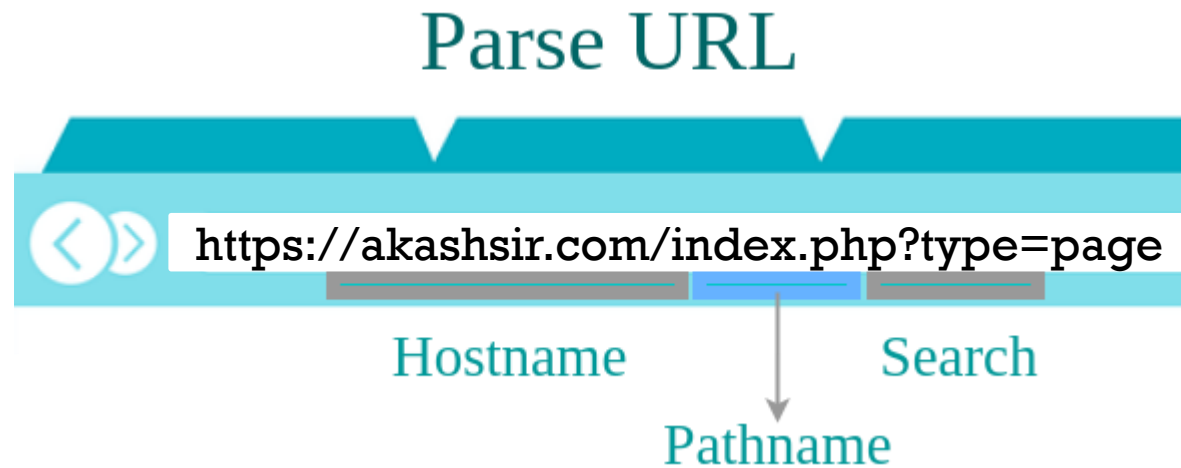


Node.js URL



URL

- use url module, and with the help of parse and query functions, you can extract all the components of URL.



Example

```
// include url module
var url = require('url');
var address = 'http://localhost:8080/index.php?type=page&action=update&id=5221';
var q = url.parse(address,true);

console.log(q.host); //returns 'localhost:8080'
console.log(q.pathname); //returns '/index.php'
console.log(q.search); //returns '?type=page&action=update&id=5221'

var qdata = q.query; // returns an object: { type: page, action: 'update',id='5221' }
console.log(qdata.type); //returns 'page'
console.log(qdata.action); //returns 'update'
console.log(qdata.id); //returns '5221'
```



Example

```
JS demo.js X
JS demo.js > ...
1 // include url module
2 var url = require('url');
3 var address = 'http://localhost:8080/index.php?type=page&action=update&id=5221';
4 var q = url.parse(address,true);
5
6 console.log(q.host); //returns 'localhost:8080'
7 console.log(q.pathname); //returns '/index.php'
8 console.log(q.search); //returns '?type=page&action=update&id=5221'
9
10 var qdata = q.query; // returns an object: { type: page, action: 'update',id='5221' }
11 console.log(qdata.type); //returns 'page'
12 console.log(qdata.action); //returns 'update'
13 console.log(qdata.id); //returns '5221'

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: cmd
localhost:8080
/index.php
?type=page&action=update&id=5221
page
update
5221
```



Get Exclusive Video Tutorials



www.apptutorials.com

<https://www.youtube.com/user/Akashtips>



Connect With Me



Akash Padhiyar
#AkashSir

www.akashsir.com
www.akashtechlabs.com
www.akashpadhiyar.com
www.apptutorials.com

Social Info



Akash.padhiyar



Akashpadhiyar



Akash_padhiyar



+91 99786-21654



#Akashpadhiyar
#apptutorials



Get More Details

www.akashsir.com



If You Liked It !

Rating Us Now



Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4_BZDET



Sulekha

<https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad>

