

Installing NodeJS on Centos 6.6.

Node.js is an open-source, cross-platform runtime environment for developing server-side web applications. Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS X, Microsoft Windows, Linux, FreeBSD, IBM AIX, IBM System z and IBM i.

Installing nodejs and npm on centos is very simple.

```
[nodejs-admin@nodejs ~]$ sudo su
[nodejs-admin@nodejs ~]# curl -sL https://rpm.nodesource.com/setup | bash -
[nodejs-admin@nodejs ~]# yum install -y nodejs
```

Installing gcc-c++ and make.

```
[nodejs-admin@nodejs ~]$ sudo yum install gcc-c++ make
[sudo] password for nodejs-admin:
Loaded plugins: fastestmirror, refresh-packagekit, security
Setting up Install Process
Loading mirror speeds from cached hostfile
 * base: mirrors.123host.vn
 * epel: ftp.cuhk.edu.hk
 * extras: centos-hn.viettelidc.com.vn
 * updates: mirrors.vonline.vn
Package 1:make-3.81-20.el6.x86_64 already installed and latest version
Resolving Dependencies
...
```

Complete!

Later on we will need kafka-node lets install that as well.

```
[nodejs-admin@nodejs ~]$ sudo npm install kafka-node
[sudo] password for nodejs-admin:
```

```
> snappy@3.0.6 install /home/nodejs-admin/node_modules/kafka-node/node_modules/snappy
> node-gyp rebuild
```

```
gyp WARN EACCES user "root" does not have permission to access the dev dir "/root/.node-gyp/0.10.36"
gyp WARN EACCES attempting to reinstall using temporary dev dir
```

```
"/home/nodejs-admin/node_modules/kafka-node/node_modules/snappy/.node-gyp"
make: Entering directory `/home/nodejs-admin/node_modules/kafka-node/node_modules/snappy/build'
CXX(target) Release/obj.target/snappy/deps/snappy/snappy-1.1.2/snappy-sinksource.o
CXX(target) Release/obj.target/snappy/deps/snappy/snappy-1.1.2/snappy-stubs-internal.o
CXX(target) Release/obj.target/snappy/deps/snappy/snappy-1.1.2/snappy.o
AR(target) Release/obj.target/deps/snappy/snappy.a
COPY Release/snappy.a
CXX(target) Release/obj.target/binding/src/binding.o
SOLINK_MODULE(target) Release/obj.target/binding.node
SOLINK_MODULE(target) Release/obj.target/binding.node: Finished
COPY Release/binding.node
make: Leaving directory `/home/nodejs-admin/node_modules/kafka-node/node_modules/snappy/build'
```

```
kafka-node@0.2.18 node_modules/kafka-node
  buffer-crc32@0.2.5
  retry@0.6.1
  node-uuid@1.4.1
  async@0.7.0
  lodash@2.2.1
  debug@2.1.1 (ms@0.6.2)
  binary@0.3.0 (buffers@0.1.1, chainsaw@0.1.0)
  node-zookeeper-client@0.2.0 (async@0.2.10, underscore@1.4.4)
  buffermaker@1.2.0 (long@1.1.2)
  snappy@3.0.6 (bindings@1.1.1, nan@1.5.3)
[nodejs-admin@nodejs ~]$ ls
```

Lets do a test.

Create a script called `example.js` with below code.

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');
console.log('Server running at http://127.0.0.1:1337/');
```

Lets start the server on a terminal.

```
[nodejs-admin@nodejs nodejs]$ node example.js
Server running at http://127.0.0.1:1337/
```

Hit the URL from the browser and We can see Hello World. So we are all set.

NodeJS is Ready.

Lets make some simple changes to existings script to handle JSON.

Here is a simple script to handle JSON data.

```
// Getting some 'http' power
var http=require('http');

// Setting where we are expecting the request to arrive.
// http://localhost:8125/upload
var request = {
  hostname: 'localhost',
  port: 8125,
  path: '/upload',
  method: 'GET'
};

// Lets create a server to wait for request.
http.createServer(function(request, response)
{
  // Making sure we are waiting for a JSON.
  response.writeHead(200, {"Content-Type": "application/json"});

  // request.on waiting for data to arrive.
  request.on('data', function (chunk)
  {
    // CHUNK which we recive from the clients
    // For our request we are assuming its going to be a JSON data.
    // We print it here on the console.
    console.log(chunk.toString('utf8'))
  });
});
```

```
    //end of request
    response.end();
// Listen on port 8125
}).listen(8125);
```

Lets fire up the script.

```
[nodejs-admin@nodejs nodejs]$ node node_recv_json.js
```

On a new terminal send some request to our script. Our script is listening on 8125 port.

```
[nodejs-admin@nodejs nodejs]$ curl -H "Content-Type: application/json" \
    -d '{"username":"xyz","password":"xyz"}' http://localhost:8125/upload
```

You will see the message received on the script terminal.

```
[nodejs-admin@nodejs nodejs]$ node node_recv_json.js
{"username":"xyz","password":"xyz"}
```

Now we are all set to do some RND.