Node JS*

Hannes Hirzel

Abstract

A quick start guide to server side JavaScript programming with Node JS. Links to more information are provided.

1 Introduction

nodejs.org[1] is an event-driven environment for server-side JavaScript programming.

What is Node.js?[2] (O'Reilly)

There are many frameworks which make use of node.js, for example express[3] (more[4]).

Stackoverflow wiki page with links on How do I get started with node[5] .

2 Installation

The installation on MSWindows is as simple as placing the file **node.exe** in the path. There is as well a regular installer file which is probably the easiest for most people. See nodejs.org[6] and https://nodejs.org/en/download/package-manager/[7] (Linux).

Node package manager

With the installation a package manager called **npm** is also installed. With it additional modules may be installed. The documentation is here https://docs.npmjs.com/[8] . There are thousands of node modules. You find them in the repository https://www.npmjs.com/[9] . The global installation of a module is as follows:

```
npm install -g theModuleName
```

As you might have different projects with different versions of the same module you might want to install a module local to a project. See below.

*Version 1.32: Apr 2, 2016 1:55 am -0500

†http://creativecommons.org/licenses/by/4.0/

[1] http://nodejs.org

[2] http://radar.oreilly.com/2011/07/what-is-node.html

[3] http://expressjs.com

[4] http://nodeframework.com/.

[5] http://stackoverflow.com/questions/2353818/how-do-i-get-started-with-node-js?rq=1

[6] http://www.nodejs.org/

[7] https://nodejs.org/en/download/package-manager/

[8] https://docs.npmjs.com/

[9] https://www.npmjs.com/

3 Examples

Assuming the "hello world" code below is in a file called 'example.js' you run it with `node example.js` from the command line.

"Hello world" web server

```
    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/');
```

(source)[10]

This module[11] shows how a Node project is set up which uses a more comprehensive static web server contained in a package.

Writing a file

```
    var fs = require("fs");
fs.writeFile('message.txt', 'Hello Node', function (err) {
  if (err) throw err;
  console.log('It\'s saved!');
});
```

(source)[12]

4 Examples of node packages

Below are some more examples of modules/utility programs you can install with the **npm** package manager. It gives you a set which may be used to construct html files with everything included. This for example useful to create platform independent emails.

```
    npm install -g grunt-cli
```

This installs the **grunt** build tool (command line version). For more on this tool, see section below.

`npm install -g jade`

Jade is a tool which translates the small Jade template language to XHTML. Read more in this[13] cnx module.

`npm install -g optimist`

optimist is a package to evaluate command line parameters.

`npm install -g inliner`

The utility **inliner**[14] turns your web page to a single HTML file with everything inlined. This is useful to create appcache manifests[15] for mobile devices when you want to reduce the number of http requests.

ShellJS[16] is a portable (Windows/Linux/OS X) implementation of Unix shell commands on top of the Node.js API. You can use it to eliminate your shell script's dependency on Unix while still keeping its familiar and powerful commands.

[10] http://nodejs.org/
[11] http://legacy.cnx.org/content/m59684/latest/
[12] http://www.nodejs.org/api/fs.html#fs_fs_writefile_filename_data_options_callback
[13] http://cnx.org/content/m48109/latest/
[14] https://www.npmjs.com/package/inliner
[15] http://legacy.cnx.org/content/m41696/latest/
[16] http://documentup.com/arturadib/shelljs

Local installation
In the previous examples the `-g` option which means "global" installation was used. This means all node projects installed use the same version. Often it is required that a project is self-contained and comes with its own particular versions of the node packages used.

For this to happen a `package.json` file is generated with the

```
npm init
```

command. To generate a package description with **npm** is useful even if you do not plan to publish your project (Ref[17]). Then the modules are installed locally with the option '–save-dev' as shown below.

First installation of modules

```
npm install jade --save-dev
npm install rho --save-dev
npm install juice --save-dev
npm install inliner --save-dev
npm install shelljs --save-dev
```

Meta data of the project
package.json

```
{
  "name": "myemailpubpackage",
  "version": "0.1.0",
  "description": "email publishing",
  "main": "index.js",
  "keywords": [
    "html",
    "publishing",
    "email"
  ],
  "author": "N. N.",
  "license": "MIT",
  "devDependencies": {
    "jade": "~1.11.0",
    "rho": "^0.2.3",
    "inliner": "~1.3.0",
    "juice": "~1.4.2",
    "shelljs": "^0.5.3"
  }
}
```

After deleting the `node_modules` subfolder it may be brought back with the command below.

```
npm install
```

Normally if you develop with version control storing the `node_modules` is excluded.

Scripts for npm
The `package.json` file may have a `scripts` section where you can define scripts to be executed.

Node js task runners are **grunt**, **gulp** or just **npm** (more[18]).

Below are links with explanations how to use npm as a task runner. building-a-simple-command-line-tool-with-npm[19]

[17] https://docs.npmjs.com/cli/install
[18] http://www.slant.co/topics/1276/viewpoints/9/~what-are-the-best-node-js-build-systems-task-runners~npm
[19] http://blog.npmjs.org/post/118810260230/building-a-simple-command-line-tool-with-npm

Below are some links with explanations how to use this feature. building-a-simple-command-line-tool-with-npm[20]

https://docs.npmjs.com/misc/scripts[21]

how-to-use-npm-as-a-build-tool/[22]

a-facade-for-tooling-with-npm-scripts/[23]

Demo how to implement commands with npm

The code below shows how to define commands with npm by using the scripts section of the file package.json. A 'start' command has been defined which calls a node program called 'index.js'.

package.json

```
    {
  "name": "mypackage",
  "version": "0.1.0",
  "description": "Simple node js example",
  "main": "index.js",
  "scripts": {
    "start": "node index.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "nodejs"
  ],
  "author": "N.N.",
  "license": "MIT",
  "dependencies": {
    "shelljs": "~0.5.3"
  }
}
```

index.js

```
    #! /usr/bin/env node

var shell = require("shelljs");

console.log('start of index.js');
shell.exec("echo shell.exec works");
```

Running the code

```
    npm start
```

Result

```
    > mypackage@0.1.0 start /media/user/data/cnx/node-example
> node index.js

start of index.js
shell.exec works
```

[20] http://blog.npmjs.org/post/118810260230/building-a-simple-command-line-tool-with-npm
[21] https://docs.npmjs.com/misc/scripts
[22] http://blog.keithcirkel.co.uk/how-to-use-npm-as-a-build-tool/
[23] https://bocoup.com/weblog/a-facade-for-tooling-with-npm-scripts/

Installing and running the example

- Download[24] files
- Unpack
- npm install
- npm start

5 Example of npm calling an XSLT processor

This example implements a call to an external tool the **Saxon XSLT processor**.
package.json

```
    {
  "name": "mypackage",
  "version": "0.1.0",
  "description": "Simple node js example",
  "main": "index.js",
  "scripts": {
    "start": "node index.js",
    "test": "echo \"Error: no test specified\" && exit 1",
    "xslt": "java -jar tool\saxon9he.jar -?"
  },
  "keywords": [
    "nodejs"
  ],
  "author": "N.N.",
  "license": "MIT",
  "dependencies": {
    "shelljs": "~0.5.3"
  }
}
```

A command **xslt** has been defined in the scripts section.
running the xslt command

```
    npm run xslt
```

Result

```
    E:\node-ex2>npm run xslt

> mypackage@0.1.0 xslt E:\node-ex2
> java -jar tool\saxon9he.jar -?

Saxon-HE 9.3.0.5J from Saxonica
Usage: see http://www.saxonica.com/documentation/using-xsl/commandline.xml
Format: net.sf.saxon.Transform options params
Options available: -? -a -c -config -cr -dtd -expand -explain -ext -im -init -it -l -m -now -o -opt -or
-TP -traceout -tree -u -val -versionmsg -warnings -x -xi -xmlversion -xsd -xsdversion -xsiloc -xsl -xsl
Use -XYZ:? for details of option XYZ
Params:
```

[24] http://cnx.org/content/m48107/latest/node-ex1.zip

```
  param=value           Set stylesheet string parameter
  +param=filename       Set stylesheet document parameter
  ?param=expression     Set stylesheet parameter using XPath
  !param=value          Set serialization parameter

E:\node-ex2>
```

Download of project file
Download[25] (2MB, as it contains the XSLT processor). Proceed as above.

6 Using node to construct a build script

It is as well possible to bypass npm for the construction of a simple build script and use node for it instead directly.

Node as a build script[26]

commander
The node module commander[27] may be used to implement command line programs with node. It is similar to **optimist** mentioned above.

7 Task runner and build tools

There are specialised task runners and build tools for node/npm. Prominent ones are grunt[28] and gulp[29] .

The **grunt** task runner / builder uses a file **Gruntfile.js** where the tasks are configured and defined. The file **package.json** contains information about the project and the npm modules needed.

Prerequisite for grunt examples
All the examples below need a global installation of the grunt command line interface.

grunt-cli installation

```
npm install -g grunt-cli
```

(0) The typical structure of a Gruntfile.js

```
module.exports = function(grunt) {

  // add configuration of tasks here


  // load plugins


  // define and register tasks to be available
  // for use on the command line

};
```

Gruntfile.js examples in this[30] module.

[25] http://cnx.org/content/m48107/latest/node-ex2.zip
[26] http://blog.millermedeiros.com/node-js-as-a-build-script/
[27] https://github.com/tj/commander.js
[28] http://gruntjs.com/
[29] http://gulpjs.com/
[30] https://legacy.cnx.org/content/m59593/latest/