Modules, require() and NPM

Paul O'Fallon
@paulofallon

Outline

- Using modules in your application
- Three most common sources of Node modules
- Creating and publishing your own modules

Using modules in your application

```
var foo = require(‘foo’);
var Bar = require(‘bar’);
var justOne = require(‘largeModule’).justOne;

var f = 2 + foo.alpha;
var b = foo.beta() * 3;

var bar = new Bar();

console.log(justOne());
```

← Modules can export variables

← … including functions

← Modules may export objects

Three sources of Node modules

#1: Built-in Modules

- **Come pre-packaged with Node**
- **Are require()'d with a simple string identifier**
  - `var fs = require('fs');`

- **A sample of built-in modules include:**
  - fs
  - http
  - crypto
  - os

http://nodejs.org/api/

pluralsight

hardcore developer training

Three sources of Node modules

#2: Your Project's files

- Each .js file is its own module
- A great way to modularize your application's code
- Each file is require()'d with file system-like semantics:
  - `var data = require('./data');` ← data.js in the same directory
  - `var foo = require('./other/foo');` ← foo.js in the 'other' subdirectory
  - `var bar = require('../lib/bar');`

    ↗ bar.js in the 'lib' directory, "up and over" from this script's directory
- Single variable require() still valid:
  - `var justOne = require('./data').justOne;`

Three sources of Node modules

#2: Your Project’s files

Variables are marked for export via “module.exports”

one.js

```
var count = 2;

var doIt = function(i, callback) {
 // do something, invoke callback
}

module.exports.doIt = doIt;

module.exports.foo = ‘bar’;
```

two.js

```
var one = require(‘./one’);

one.doIt(23, function (err, result) {
  console.log(result);
});

console.log(one.foo);
```

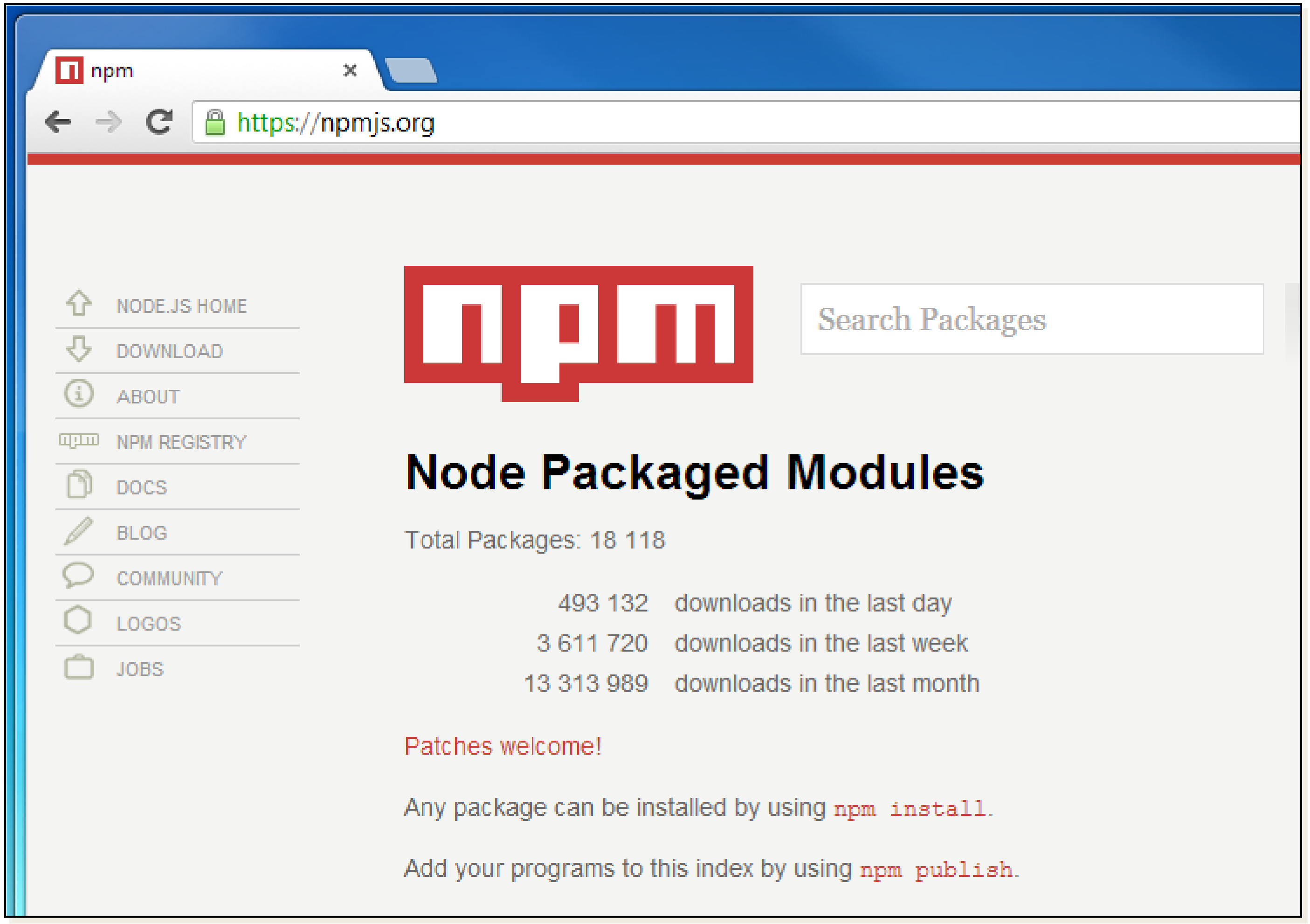
⊗ ~~console.log(one.count);~~

pluralsight

hardcore developer training

Three sources of Node modules

#3: Third Party Modules via Node Package Manager (NPM) registry



Node Packaged Modules

Total Packages: 18 118

```
   493 132   downloads in the last day
 3 611 720   downloads in the last week
13 313 989   downloads in the last month
```

Patches welcome!

Any package can be installed by using `npm install`.

Add your programs to this index by using `npm publish`.

Three sources of Node modules

#3: Third Party Modules via Node Package Manager (NPM) registry

- **Installed via “npm install *module_name*” into “node_modules” folder**
- **Are require()’d via simple string identifiers, similar to built-ins**
  - var request = require(‘request’);
- **Can require() individual files from within a module, but be careful!**
  - var BlobResult = require(‘azure/lib/services/blob/models/blobresult’);

- **Some modules provide command line utilities as well**
- **Install these modules with “npm install –g *module_name*”**
  - Examples include: express, mocha, azure-cli

pluralsight

hardcore developer training

Publishing your own module

- **package.json (located in your project root)**

```
{
  “name” : “coolstuff”,   // required
  “version” : “0.0.1”,    // required
  “author” : “Paul O’Fallon”,
  “description” : “A cool module!”,
  “keywords” : [“cool”, “awesome”],
  “repository”: {
    “type” : “git”,
    “url” : “https://github.com/pofallon/coolstuff.git”
  },
  “dependencies” : {
    “underscore” : “1.4.x”,
    “request” : “>=2.1.0”,
  },
  “main” : “lib/cool.js”
}
```

- **“npm publish .” (from within project root)**
- **“npm install *module_name*” (from an empty directory) – verify it!**

Conclusion

- How to include modules in your project
- Three common sources of Node modules
- Define and publish your own module

pluralsight

hardcore developer training

References

- Felix’s Node.js Style Guide: http://nodeguide.com/style.html
- npm cheatsheet: http://blog.nodejitsu.com/npm-cheatsheet
- package.json, An interactive guide: http://package.json.jit.su/
- What is the file ‘package.json’? http://docs.nodejitsu.com/articles/getting-started/npm/what-is-the-file-package-json
- npm’s semantic versioning: https://github.com/isaacs/node-semver
- Node.js documentation http://nodejs.org/api/
- NPM homepage https://npmjs.org/