

1. Record Nr.	UNINA9910959871203321
Autore	Pasquali Sandro
Titolo	Mastering Node.js // Sandro Pasquali
Pubbl/distr/stampa	Birmingham : , : Packt Publishing, , 2013
ISBN	9781782166337 1782166335
Edizione	[1st ed.]
Descrizione fisica	1 online resource (346 p.)
Disciplina	005.2762
Soggetti	Internet programming JavaScript (Computer program language) Web site development
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Cover -- Copyright -- Credits -- About the Author -- Acknowledgments -- About the Reviewers -- www.PacktPub.com -- Table of Contents -- Preface -- Chapter 1: Understanding the Node Environment -- Extending JavaScript -- Events -- Modularity -- The Network -- V8 -- Memory and other limits -- Harmony -- The process object -- The Read-Eval-Print Loop and executing a Node program -- Summary -- Chapter 2: Understanding Asynchronous Event-Driven Programming -- Broadcasting events -- Collaboration -- Queueing -- Listening for events -- Signals -- Forks -- File events -- Deferred execution -- process.nextTick -- Timers -- setTimeout -- setInterval -- unref and ref -- Understanding the event loop -- Four sources of truth -- Callbacks and errors -- Conventions -- Know your errors -- Building pyramids -- Considerations -- Listening for file changes -- Summary -- Chapter 3: Streaming Data Across Nodes and Clients -- Exploring streams -- Implementing readable streams -- Pushing and pulling -- Writable streams -- Duplex streams -- Transforming streams -- Using PassThrough streams -- Creating an HTTP server -- Making HTTP requests -- Proxying and tunneling -- HTTPS, TLS(SSL), and securing your server -- Creating a self-signed certificate for development -- Installing a real SSL certificate -- The request object -- The URL module -- The Querystring module -- Working with headers

-- Using cookies -- Understanding content types -- Handling favicon requests -- Handling POST data -- Creating and streaming images with Node -- Creating, caching, and sending a PNG representation -- Summary -- Chapter 4: Using Node to Access the Filesystem -- Directories, and iterating over files and folders -- Types of files -- File paths -- File attributes -- Opening and closing files -- fs.open(path, flags, [mode], callback) -- fs.close(fd, callback) -- File operations. fs.rename(oldName, newName, callback) -- fs.truncate(path, len, callback) -- fs.ftruncate(fd, len, callback) -- fs.chown(path, uid, gid, callback) -- fs.fchown(fd, uid, gid, callback) -- fs.lchown(path, uid, gid, callback) -- fs.chmod(path, mode, callback) -- fs.fchmod(fd, mode, callback) -- fs.lchmod(path, mode, callback) -- fs.link(srcPath, dstPath, callback) -- fs.symlink(srcPath, dstPath, [type], callback) -- fs.readlink(path, callback) -- fs.realpath(path, [cache], callback) -- fs.unlink(path, callback) -- fs.rmdir(path, callback) -- fs.mkdir(path, [mode], callback) -- fs.exists(path, callback) -- fs.fsync(fd, callback) -- Synchronicity -- Moving through directories -- Reading from a file -- Reading byte by byte -- fs.read(fd, buffer, offset, length, position, callback) -- Fetching an entire file at once -- fs.readFile(path, [options], callback) -- Creating a readable stream -- fs.createReadStream(path, [options]) -- Reading a file line by line -- The Readline module -- Writing to a file -- Writing byte by byte -- fs.write(fd, buffer, offset, length, position, callback) -- Writing large chunks of data -- fs.writeFile(path, data, [options], callback) -- fs.appendFile(path, data, [options], callback) -- Creating a writable stream -- fs.createWriteStream(path, [options]) -- Caveats -- Serving static files -- Redirecting requests -- Location -- Implementing resource caching -- Handling file uploads -- Putting it all together -- Summary -- Chapter 5: Managing Many Simultaneous Client Connections -- Understanding concurrency -- Concurrency is not parallelism -- Routing requests -- Understanding routes -- Using Express to route requests -- Using Redis for tracking client state -- Storing user data -- Handling sessions -- Cookies and client state -- A simple poll -- Centralizing states -- Authenticating connections -- Basic authentication -- Handshaking. Summary -- Further reading -- Chapter 6: Creating Real-time Applications -- Introducing AJAX -- Responding to calls -- Creating a stock ticker -- Bidirectional communication with Socket.IO -- Using the WebSocket API -- Socket.IO -- Drawing collaboratively -- Listening for Server Sent Events -- Using the EventSource API -- The EventSource stream protocol -- Asking questions and getting answers -- Building a collaborative document editing application -- Summary -- Chapter 7: Utilizing Multiple Processes -- Node's single-threaded model -- The benefits of single-threaded programming -- Multithreading is already native and transparent -- Creating child processes -- Spawning processes -- Forking processes -- Buffering process output -- Communicating with your child -- Sending messages to children -- Parsing a file using multiple processes -- Using the cluster module -- Cluster events -- Worker object properties -- Worker events -- Real-time activity updates of multiple worker results -- Summary -- Chapter 8: Scaling Your Application -- When to scale? -- Network latency -- Hot CPUs -- Socket usage -- Many file descriptors -- Data creep -- Tools for monitoring servers -- Running multiple Node servers -- Forward and reverse proxies -- Nginx as a proxy -- Using HTTP Proxy -- Message queues - RabbitMQ -- Types of exchanges -- Using Node's UDP module -- UDP multicasting with Node -- Using Amazon Web Services in your application -- Authenticating -- Errors -- Using S3 to store files -- Working with buckets -- Working with objects -- Using AWS with a Node server -- Getting and setting data with DynamoDB --

Searching the database -- Sending mail via SES -- Authenticating with Facebook Connect -- Summary -- Chapter 9: Testing your Application -- Why testing is important -- Unit tests -- Functional tests -- Integration tests.  
Native Node testing and debugging tools -- Writing to the console -- Formatting console output -- The Node debugger -- The assert module -- Sandboxing -- Distinguishing between local scope and execution context -- Using compiled contexts -- Errors and exceptions -- The domain module -- Headless website testing with ZombieJS and Mocha -- Mocha -- Headless web testing -- Using Grunt, Mocha, and PhantomJS to test and deploy projects -- Working with Grunt -- Summary -- Appendix A: Organizing Your Work -- Loading and using modules -- Understanding the module object -- Resolving module paths -- Using npm -- Initializing a package file -- Using scripts -- Declaring dependencies -- Publishing packages -- Globally installing packages and binaries -- Sharing repositories -- Appendix B: Introducing the Path Framework -- Managing state -- Bridging the client/server divide -- Sending and receiving -- Achieving a modular architecture -- Appendix C: Creating your own C++ Add-ons -- Hello World -- Creating a calculator -- Implementing callbacks -- Closing thoughts -- Links and resources -- Index.

---

### Sommario/riassunto

This book contains an extensive set of practical examples and an easy-to-follow approach to creating 3D objects. This book is great for anyone who already knows JavaScript and who wants to start creating 3D graphics that run in any browser. You don't need to know anything about advanced math or WebGL; all that is needed is a general knowledge of JavaScript and HTML. The required materials and examples can be freely downloaded and all tools used in this book are open source.

---