Telerik Academy

Advanced Node.js

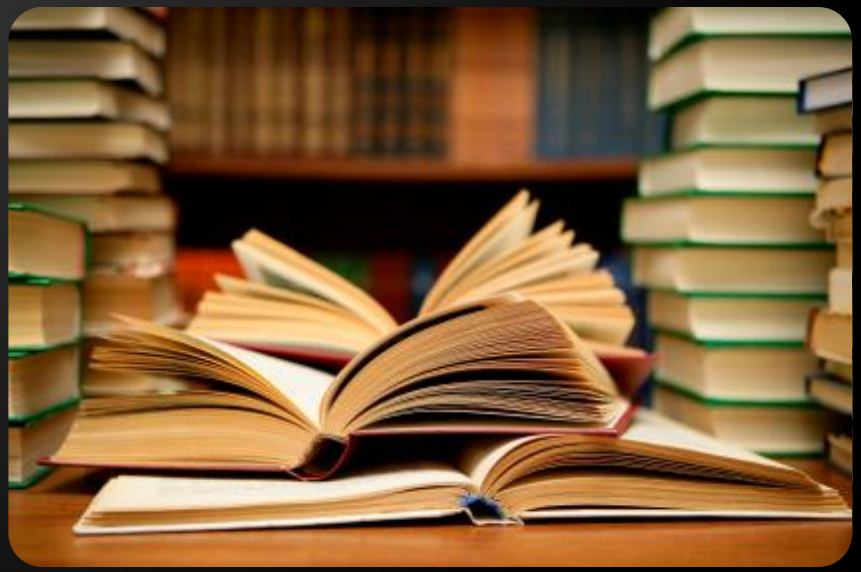
File System, Streams, Events, Debugging

End-to-end JavaScript Applications

Telerik Software Academy
http://academy.telerik.com

Telerik Academy

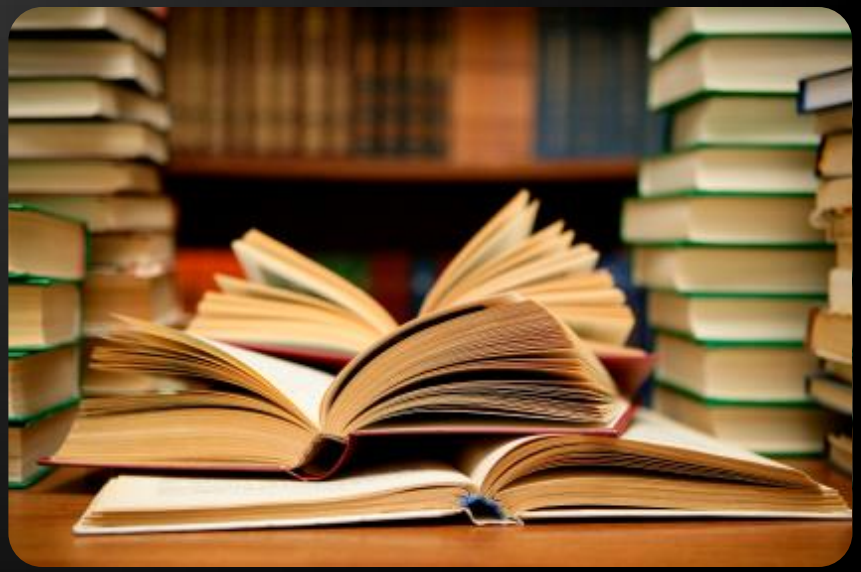
Table of Contents



Telerik Academy

Table of Contents

5. Debugging node.js applications
    1. Tools
    2. Node-inspector
    3. Debug protocol



Telerik Academy

Node.js behind the scenes

Single or multi-threaded?

Telerik Academy

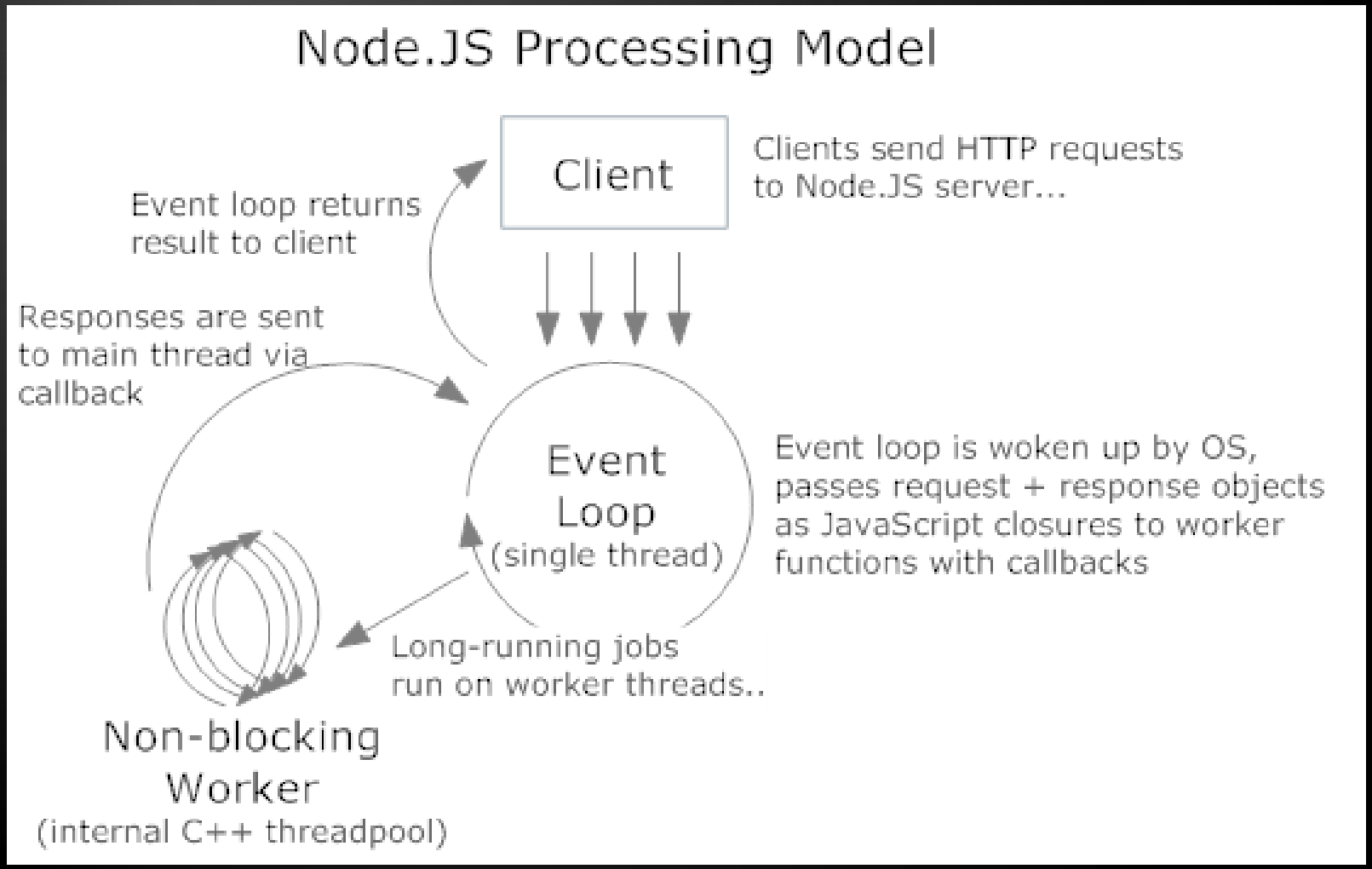
Node.js behind the scenes

- Evented I/O
  - Don't call us, we'll call you
  - Non-blocking
  - Asynchronous
  - Event loop
- Worker threads and the V8 engine
- The event loop is single-threaded, but underneath relies on an internal pool of C++ threads

Telerik Academy

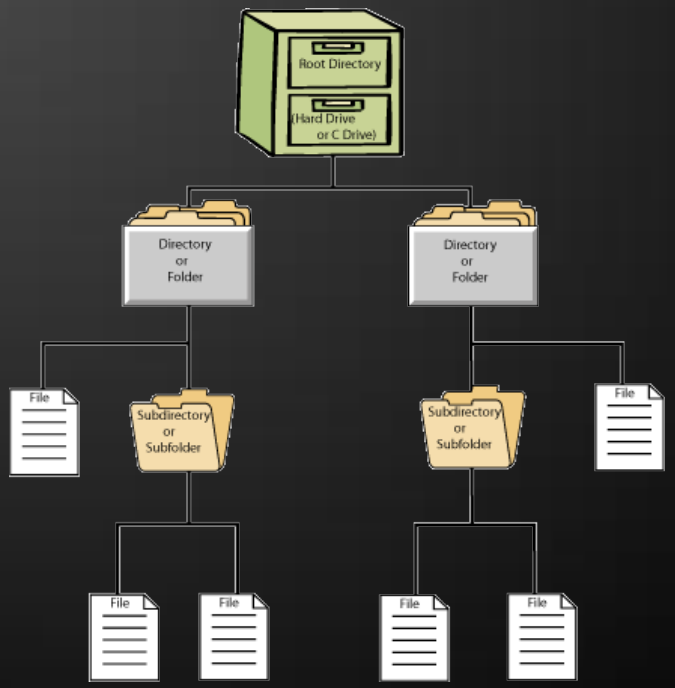
Node.js behind the scenes

- Node.js processing model



Telerik Academy

Working with the file system



Telerik Academy

Working with the file system

- The fs module

```
var fs = require("fs");
```

- Two ways to tackle working with fs:
  - Synchronous
  - Asynchronous (the node.js way)
- Synchronous

```
fs.mkdirSync(path, [mode]);
fs.renameSync(oldPath, newPath);
fs.writeFileSync(filename, data, [options]);
```

Telerik Academy

Working with the file system

- Asynchronous

```
fs.mkdirSync(path, [mode], callback);
fs.renameSync(oldPath, newPath, callback);
fs.writeFileSync(
      filename, data, [options], callback);
```

- Official documentation

  http://nodejs.org/api/fs.html

Telerik Academy

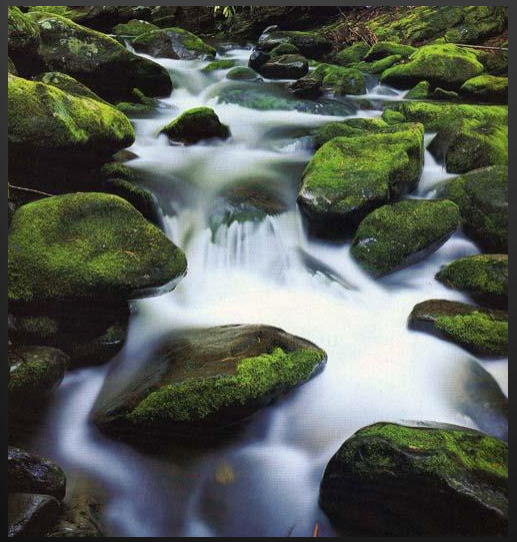
Working with fs

Live demo

Telerik Academy

Streams

Streams, buffers and chunks



Streams

- Types
  - Readable – can only be read (process.stdin)
  - Writeable – can only be written to (process.stdout)
  - Duplex – both Readable and Writeable (tcp sockets)
  - Transform – Duplex streams where the output is computer from the input (zlib, crypto)

Streams

- Buffers
  - JS is unicode friendly, but not the best when it comes to binary data.
  - Handles octet streams
  - Can be easily transformed into JS object by setting an encoding in the buffer's toString() method

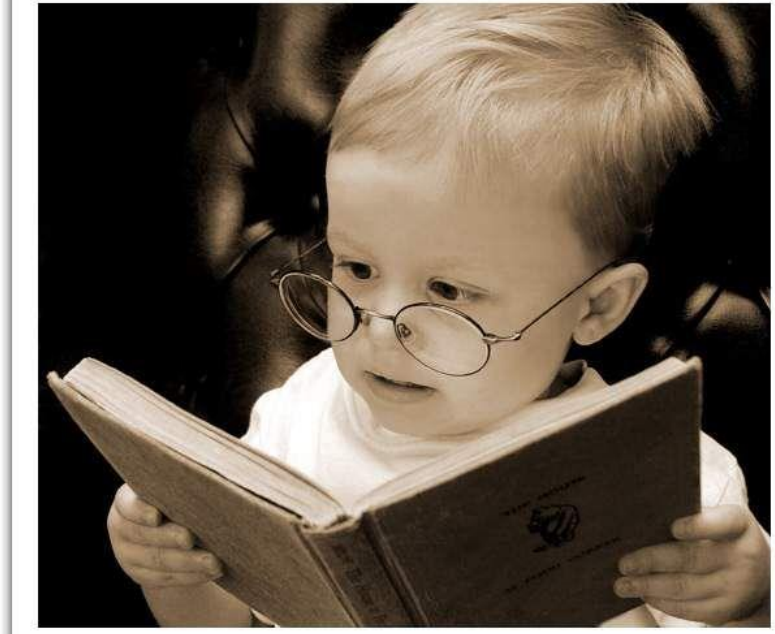
Streams

1. Readable stream
   - Events
     - readable, data, end, close, error
   - Methods
     - read, pause, resume, pipe, unpipe, unshift
   - Implementation interface
     - _read(size)
     - push(chunk, [encoding])

Telerik Academy

Readable streams

Live demo



Streams

1. Writeable stream
   - Events
     - Drain, finish, pipe, unpipe, error
   - Methods
     - write, end
   - Implementation interface:
     - _write(chunk, encoding, callback)

Telerik Academy

Writeable streams

Live demo

Streams

- Duplex stream
  - Implements both the Readable and Writeable interfaces
  - Example - a TCP socket
- Transform stream
  - A special kind of duplex stream where the output is a transformed version of the input
  - _transform(chunk, encoding, callback)
  - http://codewinds.com/blog/2013-08-20-nodejs-transform-streams.html

Telerik Academy

Duplex and Transform

Live demo

Telerik Academy

Events

Events

- Huge application in node.js applications
- Many objects in node.js emit events, i.e. are EventEmitters. Examples are fs, net, zlib, etc.
- Basically an implementation of the publish/subscribe pattern

Telerik Academy

EventEmitter

Live demo

Telerik Academy

Debugging

Telerik Academy

Debugging in node.js

- Debugging in node.js
  - The V8 debug protocol
    - JSON based protocol
  - IDEs with a debugger
    - Webstorm
    - Visual Studio
    - Sublime Text (+node.js plugin)
    - Node-inspector (awesomeness!)

Telerik Academy

Debugging

Live demo

Telerik Academy

http://algoacademy.telerik.com

Free Trainings @ Telerik Academy

- "C# Programming @ Telerik Academy
  - csharpfundamentals.telerik.com

- Telerik Software Academy
  - academy.telerik.com

- Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy

- Telerik Software Academy Forums
  - forums.academy.telerik.com