Battle Ready! - Hardening Node.js for the Enterprise

Jeff Barczewski

codewinds.com
NOW!!!
OUR FRAMEWORK
MUST BE REALLY
REALLY GOOD

SIGH

SIGH

SIGH

"LEAKY ABSTRACTIONS"

See "The Law of Leaky Abstractions" from Joel Spolsky
(http://www.joelospolsky.com/articles/leaky-abstractions.html)
Over 10 years old but still true.
Why build scalable, durable apps?
It is interesting to see who has scalable web architectures and who doesn’t. Kmart’s site melted down for me today. Try #nodejs
See plans & prices for 2015!
Starting November 15, you can enroll in an affordable health plan that works for you

SEE PLANS & PRICES   GET READY TO ENROLL

You may still be able to get coverage for the rest of 2014. Learn more.
Hiring pro tip: Hippies are not the same as hipsters

I have made this mistake and we now have a frontend MVC framework written in Fortran.
Who am I?

- 25 yrs professional developer
- 3 yrs full time with Node.js
- MasterCard, Elsevier, RGA, USAF
- CodeWinds Podcast and Training
Elsevier Education Platform

- Rapid Development Apps
- Node.js
- Isomorphic Javascript
MasterCard CaaS

- Crypto as a Service
- Node.js plus addon HSM
- REST API
- Highly scalable and available
- Reporting, Administration
MasterCard / Apple Pay Launch

- Complex deployment, many third parties
- Field testing
- US public launch - "It just worked"
Other Enterprise Use

- Walmart
- Pay Pal, Linkedin
- Yahoo, ebay, Uber, Netflix
- [http://nodejs.org/industry/](http://nodejs.org/industry/)
Legacy Code & Stagnation
"Inspire with your actions"
How to Introduce Node.js

Simple beginnings

- CLI tools
- Batch processing
- Admin applications
- Green field projects, REST, SPA
Smart API reverse proxy

- SSO, batching
- Orchestration, caching
- Device specific content
- Migration, AB Testing

API Proxy by Yury Baranovsky
Inspired
F15 Eagle

F-15E Strike Eagle by Gerry Metzler - IMG_214 Licensed CC BY-SA 2.0

Afghanistan, F-15E 391st" by Staff Sgt. Aaron Allmon (USAF) - Src. Public domain
F15 Video

- Full F15 video from 67FS - I shared an edited portion of this video in my talk
Life happens
Traits that inspire me

- Performant
- Adaptability
- Durability
- Trusted
Design and Foundations

What frameworks and tools?

- Hapi, Express, Restify
- Is it battle tested? Quality? Community? License?
- Prototype with API
- Joi, Wreck
var express = require('express');
var compression = require('compression');
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var bodyParser = require('body-parser');

var app = express();
app.use(compression());
app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(cookieParser());
app.get('/', function (req, res) {
    res.send({ hello: 'world' });
});

// catch 404 and forward to error handler
app.use(function(req, res, next) {
    var err = new Error('Not Found');
    err.status = 404;
    next(err);
});

app.listen(3000, function()
    console.log('Express server listening on port ' + 3000); });
Hapi Example

```javascript
var Hapi = require('hapi');

var server = new Hapi.Server(3000);

server.route({
    method: 'GET', path: '/',
    handler: function (request, reply) {
        reply({ hello: 'world' });
    }
});

server.start(function () {
    console.log('Server running at: ', server.info.uri);
});
```
Design and Foundations

Where to store data?

- Redis - caching, sessions, queues
- REST services - SQL (postgres), noSQL (riak, dynamo, leveldb)
- async libraries, uses, verify
- Sharding plan
Design and Foundations

Reducing risk

- RDD, Test API, CI
- Git pull requests, shrinkwrap
- jshint, markdown, jsdoc
- Node.js / JS training
How do we scale up?

- Node.js evented architecture
- Cluster, Child processes, Containers, Zones
- Load balancer, nginx, HA proxy, apache
How do we know it will scale?

- Load Test, stress test, vary workload
- bench-rest, wrk, ab
- load runner, blitz.io
- benchmark frequently
bench-rest

// unique PUT and then a GET for each iteration

var flow = {
  main: [
    {
      put: 'http://localhost:8000/foo_#{INDEX}',
      json: 'mydata_#{INDEX}'
    },
    {
      get: 'http://localhost:8000/foo_#{INDEX}'
    }
  ]
};

module.exports = flow;
NEVER RELY JUST ON HOPE

ALWAYS DO MORE!
Resilience

How do we make this durable?

- child processes, cluster_master_ext
- Detect process exit, restart
- DB failures, reconnect generic-pool, backoff
- Monit, upstart, forever, nscale
```javascript
var clusterMaster = require("cluster-master-ext");

clusterMaster({
  exec: "worker.js", // script to run
  size: 5, // number of workers
  env: { SOME: "environment_vars" },
  args: [ "--deep", "doop" ],
  silent: true,
  signals: true
});
```
How to make this hot deployable?

- Rolling restart
- Handle shutdown, exit cleanly, fail safe timeout
- Signals: TERM, INT - shutdown, HUP - rolling restart
How do we monitor?

- Logging? Events?
- Bunyan, jeffbski/hapi-bunyan-lite
- Monit, upstart, forever, nscale
- Notifying, dtrace, vasync
- Charting - Rickshaw, zag, statsd/graphite, influxdb
Demonstration
True Friend
Resilient Solutions

- Strong
- Reliable
- Trusted
- Adaptable

Properties of Resilient Systems by DoD Licensed CC BY-NC-ND 3.0
To learn more

- http://codewinds.com/nodevember
- CodeWinds - Resilient Systems
- jeff@codewinds.com
- @codewinds @jeffbski