

Multi-package projects with Lerna

Philip Nuzhnyy // Lisbon.js // Dec 12, 2018



I Am Devloper

@iamdevloper

Follow



- hey how did your hackathon go?
- not too bad, we got Babel set up
- yep...
- yep.

3:53 AM - 17 Oct 2016

1,150 Retweets 2,146 Likes



23



1.2K



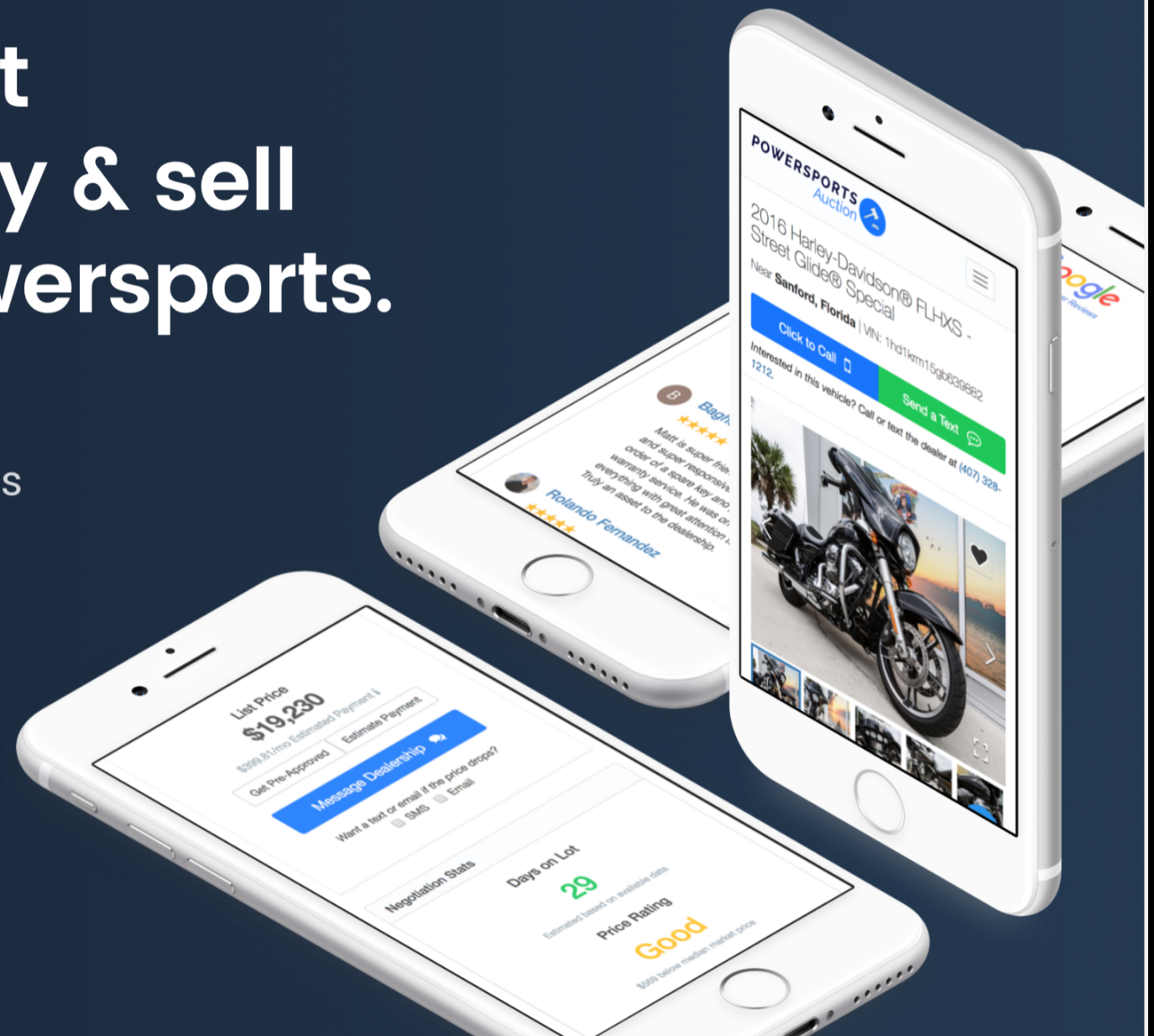
2.1K



How did we get there?

The smartest, most efficient way to buy & sell motorcycles & powersports.

Search \$250MM worth of used motorcycles & powersports for sale from 800+ dealers and growing.



Problem

- web app choking on background tasks
- increasing inventory
- keep our dev and deployment flow simple

Solution

- split a monolith web app into 2 parts: app and workers
- extract common dependencies and reuse them
- monorepo

Monorepos

Monorepo is a software development strategy where code for many projects are stored in the same repository

source: [Wikipedia](#)

Monorepos

As of 2017 this software engineering practice was over a decade old, but had only recently been named

source: [Wikipedia](#)

Monorepos: Pros

- Single lint, build, test and release process.
- Easy to coordinate changes across modules.
- Single place to report issues.
- Easier to setup a development environment.
- Tests across modules are run together which finds bugs that touch multiple modules easier.

source: [Why is Babel a monorepo](#)

Monorepos: Cons

- Codebase looks more intimidating.
- Repo is bigger in size.
- Can't npm install modules directly from GitHub

source: [Why is Babel a monorepo](#)

Branch: master ▾

react / packages /

Create new file

Upload files

Find file

History



mrkev and acdlite Implement pauseExecution, continueExecution, dumpQueue for Scheduler (#...

Latest commit 8df4d59 5 days ago

..

create-subscription	Publish a local release (canary or stable) to NPM (#14260)	19 days ago
eslint-plugin-react-hooks	Publish a local release (canary or stable) to NPM (#14260)	19 days ago
events	Remove unused simulated flag parameter (#14127)	a month ago
jest-mock-scheduler	Fixed scheduler setTimeout fallback (#14358)	10 days ago
jest-react	Publish a local release (canary or stable) to NPM (#14260)	19 days ago
react-art	Publish a local release (canary or stable) to NPM (#14260)	19 days ago
react-cache	Publish a local release (canary or stable) to NPM (#14260)	19 days ago
react-debug-tools	Remove useMutationEffect (#14336)	14 days ago
react-dom	Added ErrorBoundary tests for useEffect and useLayoutEffect (#14401)	5 days ago
react-is	Adding isMemo check to react-is package (#14313)	14 days ago
react-native-renderer	Remove `import * as` pattern from the codebase (#14282)	22 days ago
react-noop-renderer	[Fizz] New Server Rendering Infra (#14144)	12 days ago
react-reconciler	TestRenderer toJSON should not expose the Array wrapper Suspense uses...	7 days ago
react-stream	[Fizz] New Server Rendering Infra (#14144)	12 days ago
react-test-renderer	TestRenderer toJSON should not expose the Array wrapper Suspense uses...	7 days ago
react	Implement pauseExecution, continueExecution, dumpQueue for Scheduler (#...	5 days ago
scheduler	Implement pauseExecution, continueExecution, dumpQueue for Scheduler (#...	5 days ago
shared	Implement pauseExecution, continueExecution, dumpQueue for Scheduler (#...	5 days ago

Back to Lerna

Lerna is a tool that optimizes the workflow around managing multi-package repositories with git and npm.

Lerna: how it works

```
my-lerna-repo/  
  - package.json  
  - packages/  
    - app/  
      - package.json  
    - workers/  
      - package.json  
  - shared/  
    - package.json
```

repository structure

Lerna: lerna bootstrap

- npm install all external dependencies of each package
- Symlink together all Lerna packages that are dependencies of each other
- npm run prepublish in all bootstrapped packages
- npm run prepare in all bootstrapped packages

Lerna bootstrap: example


```
{  
  "name": "babel-core",  
  "dependencies": {  
    "babel-generator": "^6.9.0",  
    "source-map": "^0.5.0"  
  }  
}
```

package.json for babel-core module

Lerna: run

Run an npm script in each package that contains that script

```
lerna run dev
```

**Top level package.json for all the
chores**

root package.json

CI integration

- push to dev/master trigger staging/prod deployment
- lint and test all the packages using top level test and lint
- deploy the web app (docker image)
- deploy workers (heroku based tiny nodejs services)

How it worked out

- pretty painful at first
- an ongoing effort
- 9 packages at the moment
- all new major features/experiments get a package

Conclusion

- don't do this at home (unless you really need this)
- treat your sub packages as if they were 3rd party npm modules
- move all house keeping to the root `package.json`

Thank you 🙏