JavaScriptの素振りする技術

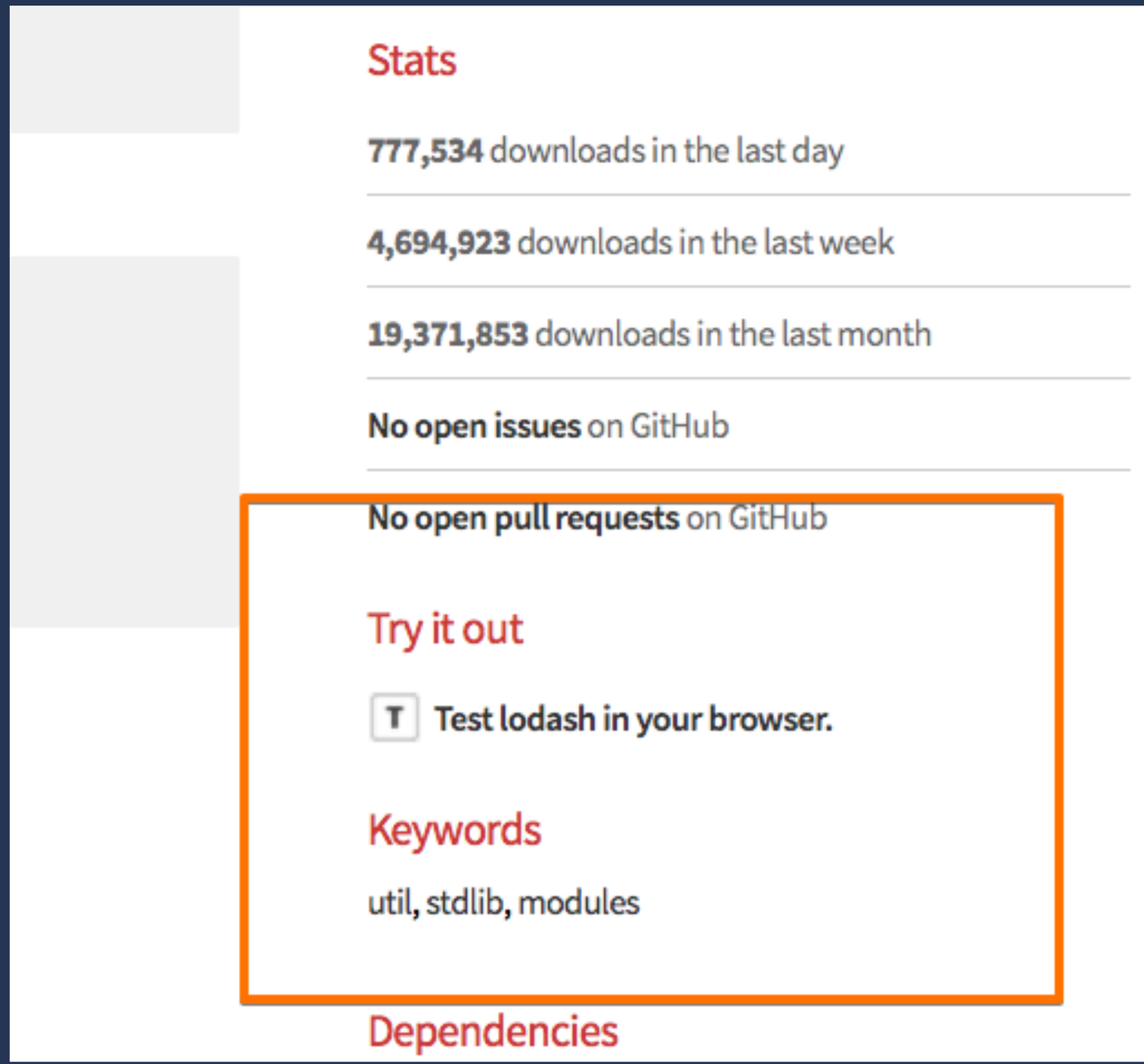
ライブラリをちょっと試す

ライブラリをブラウザで試す

- JavaScriptライブラリはデモを置いてることも多い
- デモがないとローカルで動かすのは面倒くさい
  - わざわざローカルで`npm install`して...
- ちょっとしたこと試す時に使えるツール

npmをその場で試す

- Tonic: a better REPL for node.js
- Browserify + REPLのようなサービス
- npmからリンクが貼られてる!
- 例: Tonic: npm on Tonic



他にも色々

- RequireBin
  - Browserify + JSFiddle的なサービス
- Firefox DevTools
  - URLをページにインジェクト出来る
  - 開発ツールバーの `inject` コマンド
  - ライブラリを読み込ませてConsoleで叩く

JS Envy

For when you just want to mess around in the `console`.

Check out the blog if you're not sure what to do here.

https://raw.githubusercontent.com/jaredreich/notie.js/master/notie.js

Search for a library, powered by cdnjs, or enter the **full pat** and c here.

No results, try using the full path option.

Loaded Libraries

ubusercontent.com/jaredreich/notie.js/master/…

JS Envy

Window Changes

New Properties

notie

New Methods

Console

ライブラリをちょっと試すまとめ

- ライブラリをちょっと試すだけならブラウザだけでイケる
- 説明文をそのまま鵜呑みよりは一行でも実行する
- 実行するためのツールは色々充実してきている
- Node.js向けでもBrowserifyで動くレベルならブラウザでREPLができる

ライブラリの新しい機能を試す

- ライブラリで新しい機能追加された
- リリースノートに細かいことが書かれてない
- 関連: われわれは、いかにして変更点を追うか
- ちょっと探しても見つからなかったら実際に試す

例) Jasmineのランダムテスト

- 2015-12-07のJS: Jasmine 2.4.0、Redux入門、Firefox Platform Status - JSer.info
  - Jasmine 2.4.0で追加されたランダムテストの紹介
- Release Notesには細かいことが書かれない
- 実際に試さないと正確なことが書けなかった

動かす前の認識

Run jasmine's specs in random order

デフォルトでランダム実行になった?

Add support for returning run details for reporting randomness

どういう意味?

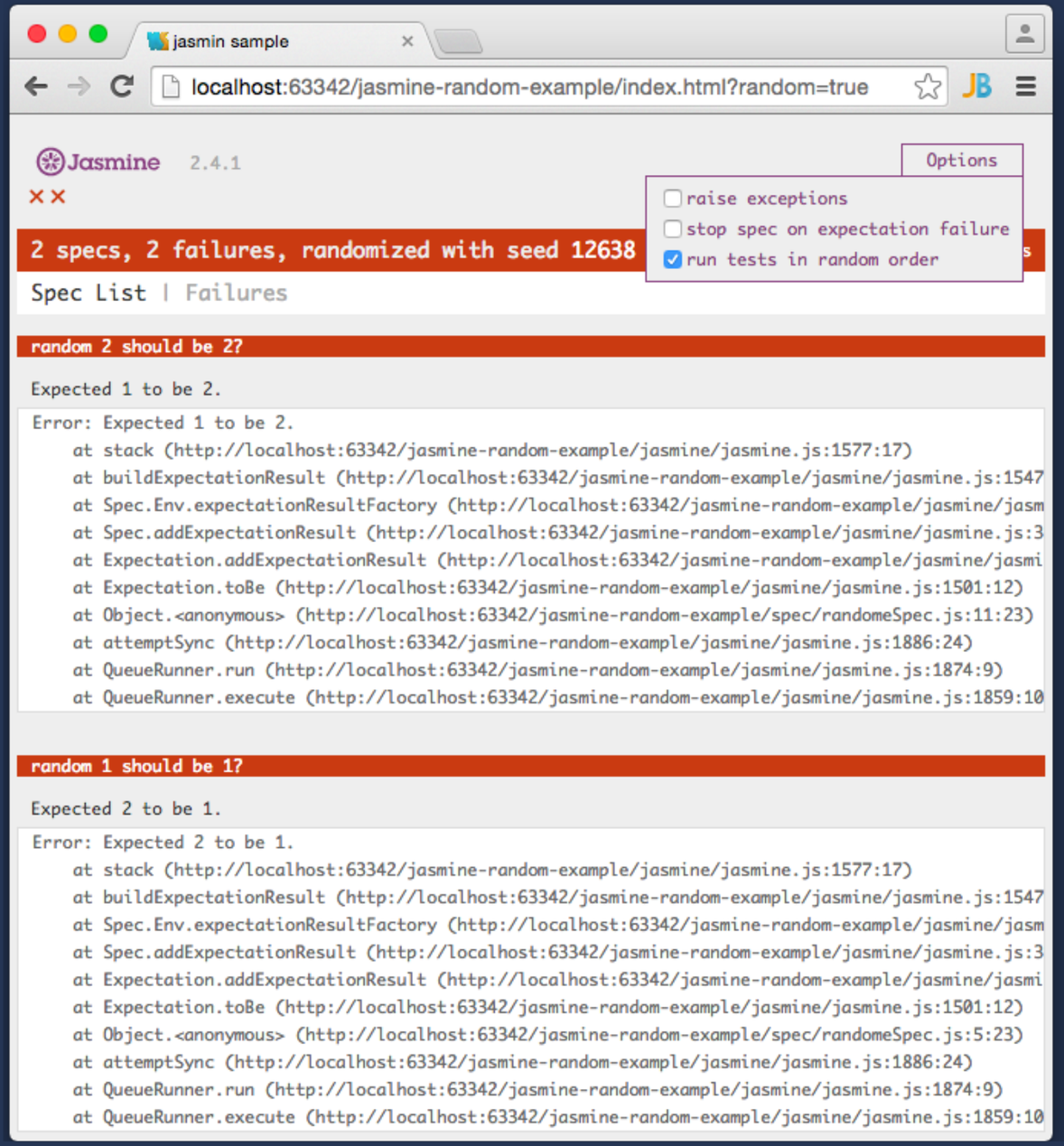
動かす

```
$ mkdev jasmine-random-example
$ npm install -g jasmine
$ jasmine init
$ jasmine examples
$ jasmine # run
```

- jasmine.jsonに"random": falseというのが増えていた
  - => デフォルトはfalseだった

動かす

- Node.jsのjasmineだとtrueにしても何故かランダムじゃない
- jasmineはHTML上で動かせる事を思い出した
- HTMLで動かしたら設定を見つけた！
- *run details for reporting randomness* はseed値のパーマネントを作るという意味



動かした後の認識

- デフォルトではランダム順序の実行ではなかった

Add support for returning run details for reporting randomness

これはseed値が失敗した時にでるという意味だと分かった

動かすことで得たもの

- 10分ぐらいで適当に動かせて認識を正すことができた
- ついでにそのままGitHubにpushして動くサンプルを作れた
- https://github.com/azu/jasmine-random-example/
- http://azu.github.io/jasmine-random-example/?random=true

壁

実際に動かすまでには壁がある

- ライブラリを読み込んで実行するまでに色々手順が必要
- 自分なり手順をテンプレ化しておく = 素振り
- npm install -> write code -> git push が大まかな流れ

@azu のケース

```
# ghqディレクトリにhogeを作ってhogeへ移動
mkdev hoge
# git, npm, license init
init-node.sh
### Development... ###
# Githubリポジトリを作成
hub create -d "description"
# git push -uのスクリプト
git pushup
```

使用してるスクリプト

- https://gist.github.com/azu/09dd6f27f52e2e8d9978
- 基本的に覚えられるコマンドしか使わない
- `init-node.sh`はpecoで色々なパターンを選択して使う
- 大体どの言語でも同じパターンで作って公開してる

ライブラリの使い勝手を試す

- 使い勝手を把握するのは実際に何かを書かないと分かりにくい
- コストが高いのであんまり多用できない
- でも、書かないと使い勝手を見るのは難しい

ダイナモS+
@mizchi

フォロー

素振り専用のリポジトリ作ってめっちゃ素振りしているので結局コード書く速さも動く速さも書いた分量によると思うよ
github.com/mizchi-sandbox

4
リツイート

いいね
8

7:14 - 2014年11月13日

何を持って使い勝手を試すか

- 結局素振り
- 書くものがないならElectronやNW.js
  - 環境が固定されてる、新しい機能が使いやすい
  - メンテを考えないならコンテキストが混ざってるNW.jsだと楽

とりあえず作る

- 作ってGitHubにあげる
- 完成しなくてもGitHubにあげる
- そのままローカルのゴミ箱に捨てるよりはGitHubに捨てる
- ゴミ箱に捨ててしまうと記憶からも無くなってしまう

適当に作ったもの

- JSer.info Pull Request Form
  - Angularを試したくなり作った
  - JSer.infoに紹介してもらいたい記事のPull Requestが出来るようになりました - JSer.info
- JSer.info contributing item preview
  - Vue.js 1.0を試したくなり作った

- azu/hatebu-mydata-search
  - Flux Utilsを試したくなり作った
  - はてなブックマーク検索を作りながらFlux Utilsについて学ぶ | Web Scratch
- azu/bookmarkletter
  - benjamn/ast-typesを使いたくて作った
  - ブックマークレットを作るコマンドラインツール | Web Scratch

- azu/video-prefetcher
- azu/video-shortcut-controller
- azu/video-transcript-note
- azu/video-transcript-tracker
  - `<video>`と`<track>`に触りたくて作った
  - 動画とルビ翻訳された字幕をみながらMarkdownメモできるアプリを書いた | Web Scratch

- jser/stat-js
  - naturalを使った自然言語解析がやりたくなった
- azu/audio-node-flow
  - Web Audio APIに触りたくて書いた
  - => Web Audio APIの標準に同様のものが追加されてた
    Web Audio Method Chaining Sample
  - JavaScriptとWeb Audio事始め

Issueを出す

- 問題がある時に一番いいのは再現可能なサンプル
  - 合わせてスクリーンショットなど
- 再現可能なサンプルを作って公開するのは面倒

Issueのサンプル

- JSFiddleみたいなパーマネントリンクだけ済むならそれを出す
- コマンドだったり、ファイルサイズみたいな問題だと実際にリポジトリを作る
- 素振りで慣れておけばサンプルをあげるのも5分かからない

Issueのサンプル : deku

- Reduce build file size by azu · Pull Request #297 · dekujs/deku
- Add "browser" field for browserify by azu
- Browserifyで使うとファイルサイズが50KB増える問題
- 実際にファイルサイズが50KBになるリポジトリへリンク
  - azu/component-type-with-browserify-issue

まとめ

- 実際にローカル環境を作らなくてもJavaScriptは動かせる
- ローカル環境でもパターン化してスグ動かせるように素振り
- バグ報告には再現可能なサンプルを一緒に出そう
- ゴミはゴミ箱ではなくGitHubへ
- ライブラリ書く側はドキュメントを分かりやすく書こう、デモを作ろう