


ReactでGyazzというWikiを自作して運用している話

@shokai  (/Nota/shokai)

Nota, Inc.

Nota, Inc.

<http://gyazo.com> (<http://gyazo.com>)を運営している会社
スクリーンショット共有サービス
アニメGIFでも撮影できる
プルリクにも便利
(https://github.com/nota/gyazz_clone/pull/255)

Gyazo Tシャツ & シール

私に質問や感想を言うとあげます
トーク後でも可
15枚ぐらいある



(<https://gyazo.com/02fb6e1009a6115079f2bb6603971b73>)

Gyazz という名のWiki

2人で作っている  (/Nota/洛西)  (/Nota/shokai)

トーク概要より

WYSIWYG (./WYSIWYG)で画像やリンクも貼れるけど、編集ツールバーやボタンが無いエディタ

- markdownより簡単なシンタックスで編集できる

- Atom (./Atom)を参考にしてReact (./React)で実装した

- contenteditable (./contenteditable)は使っていない

Socket.IO (./Socket.IO)と自作の同期システムでうまいこと同時編集できる

- Gitを参考にsubversionみたいなものを実装した

社内で運用してて、3ヶ月で900 ページくらい作られている

このスライドもGyazzから生成している

「五行の送り火」を「大文字焼き」って書いたら勝手に修正
されてた

Gyazoのヘルプ (<https://gyazz-clone.herokuapp.com/Gyazo>)

大学の研究室などでは5,6年で7000ページ以上

(<http://gyazz.masuilab.org/>増井研/)

YAPC2016 shokaiメモ (./YAPC2016 shokaiメモ)

Wiki (./Wiki)

ワイキキ空港のターミナル連絡バスのこと



(<https://gyazo.com/b245c3244dc646ddc29a8a9cafd15d3b>)

みんなで書けるwebサイト

Wikiのつらみ

メジャーなWikiの方式

独特のシンタックスを手で書く 方式

覚えるのつらい


さっき見ていた場所をちょっと直したいだけなのに、編集モードに入ると全体が生テキストになってる・・・

WYSIWYGエディタ方式

見たまま編集できる

ツールバー、メニューがたくさん

一度作った綺麗な見た目の段落と、同じスタイルの段落をもう一個作るのが難しい

文章書いている時間より装飾整えてる時間の方が長い・・・ ()



(/Nota/shokai)個人の見解です)

Gyazzとは？

 (/Nota/増井) が10年ぐらい前に作った



(<https://gyazo.com/0e7cd8cde4a705f8d5fa46b94053486a>)

ここ数年、いろんな人がGyazzクローンを作っている  (/Nota/洛西)  (/Nota/shokai)

アウトラインエディタのようなWiKi (./WiKi)

編集・閲覧のモード切り替えが無い

カーソルのある行だけがWiKi Syntaxむき出しになる
かんたんなSyntax

がんばって書いたり覚えたりしなくても、適当に見やすく表示する

ページ間リンクを貼っておけば関連検索で発見できる

階層は無い、Web的

アイデアを書いておくと、忘れた頃に関連リンクに現れてピンときたりする

複数人で同時編集できる

Gyazz記法 (./Gyazz記法)

ほぼ 角カッコで囲うだけ

外部リンク

[URL]

内部リンク

[ページタイトル]

#ページタイトル **ハッシュタグ風**

タイトル付き外部リンク

[URL タイトル]

[タイトル URL] **逆順でもいい**

画像埋め込み

[画像のURL]

リンク付き画像

[画像のURL リンク先URL]

[リンク先URL 画像のURL] **逆順でもいい**

アイコン記法

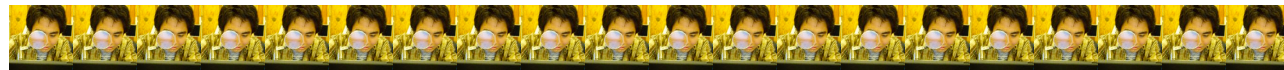
[ページタイトル.icon]



(/Nota/shokai)

アイコンタワー記法

[ページタイトル.icon*数字]



(/Nota/shokai)

太字

[[あいうえお]] 角カッコ2つ

あいうえお

インラインコード

`console.log("hello")` **バッククオートで囲う**

コードブロック記法

test.js

```
function () {  
    alert('hello')  
}
```

code:言語名 もしくは code:ファイル名 の下のブロックに色が
つく

CLI記法

```
$ npm install jquery -save
```

```
% git init
```

行頭がコマンドっぽい行に色がつく

(デモ)

(デモ：ページを作ってみる)

React (./React)のページを書く

hashタグ

関連ページ

外部リンク

画像うめこみ、リンクつける


CLI記法














codeblock記法

アイコン記法

asdf

アイコン記法で投票

 **アニメ2016春**

- 2016年4月 開始のアニメ について語るスレ
 - 前: [アニメ2016新春](#)
 - 次: [アニメ2016夏](#)
 - 良かったら上にageる
- 三者三葉
 - 未確認で進行形の人の初連載の4コマ
 - よい 
 - □今季OP賞 
 - 良作でした。ゆるゆる1期2期とかに近いかな 
 - 鬼畜メガネすざい好き 
- クロムクロ
 - 富山が壊滅しそう
 - けっこう面白そう。1話の時点では一番好きかも 
 - 期待あげ 
- ふらいんぐういっち
 - よいと思う 
 - 非常に良かった 
 - たぶん良い 
- Re:ゼロから始める異世界生活
 - 面白い 
 - 面白かったけど主人公がうざいのでこのくらいで 
 - 面白いけど主人公がウザいので☆4としました 
 - なるう作品 気合入ってる 

(<https://gyazo.com/8afa1ca644575b370cc5518980590b21>)

カーソル行がsyntaxむき出しになる の重要

「これどう書いてるのかな？」がすぐ見れる
HTMLのソース見て勉強したのを思い出す

実装の話

実装環境

Node.js

express

Socket.IO

mongodb

React、jQuery

EventEmitter式Store

全面的にasync-await (./async-await)

Browserify

エディタの実装

カーソルの縦棒はdiv
緑の範囲選択もdiv3つ
隠し <textarea />

IMEウィンドウを見せるためだけに存在する
カーソルの右側に浮いていて、ついてくる


アイコン icon

- twitter定期的にglitchしたアイコンをアップロードしている
 - 約20分ごとに切り替わる
- 日本語入力
- gyazzページにtwitterアイコンが設定されていたのもっと!



test

(<https://gyazo.com/688d80344a2b353762ec9c79e68a637b>)

 (/Nota/Atom)が参考になった

y行目のx文字目の画面上の座標はどこか？

生テキスト こんにちは を
1文字ずつ分割して

editor.html

```
<div class='lines'>  
  <div class='line' id='L1'>  
    <span class='c-1'>こ</span>  
    <span class='c-2'>ん</span>  
    <span class='c-3'>に</span>  
    <span class='c-4'>ち</span>  
    <span class='c-5'>は</span>  
  </div>  
</div>
```

jQueryで位置を取得する

```
let {left, top} = $('.lines #L1 .c-4').position()
```

クリックした位置から何行目の
何文字目なのか？を求める

逆をやればいい

あとはカーソル移動やemacsキーバ
インド等を自前で実装すればok

ReactとjQueryは相性悪い？

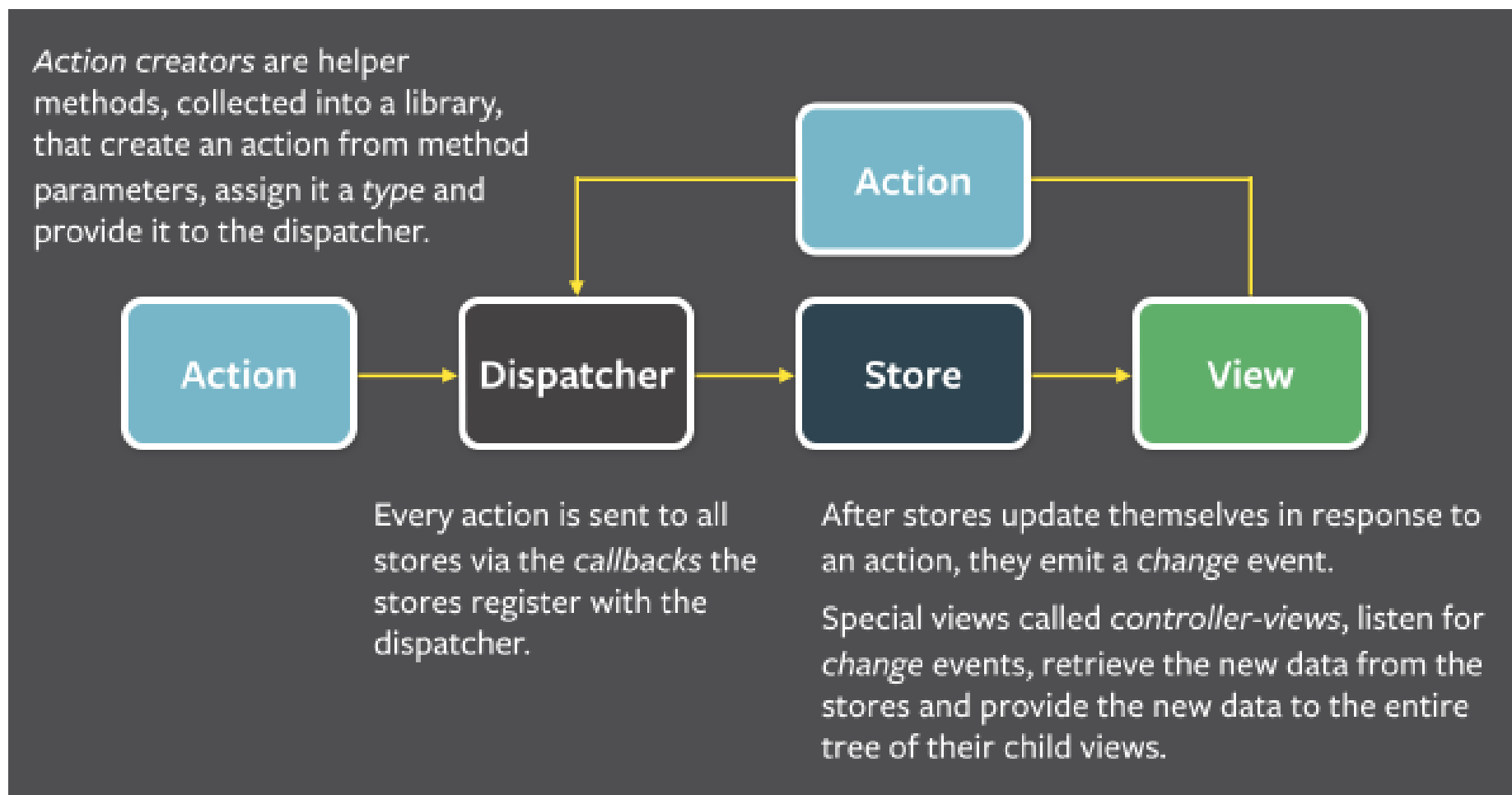
よくインターネットに書いてある

jQueryでDOM書き換えしなければok

jQuery クロスブラウザで座標の扱いが完璧っぽい

ReactとjQueryでflux

flux



(<https://facebook.github.io/flux/docs/overview.html>)

viewはstore (stateとprops)からのみ作られる

viewが他のviewを書き換えない

他のviewの位置を使って描画するviewはどうするの？

他のview componentの位置で位置 補正するcomponent

storeが変更される

render()

componentDidUpdate が発火

ここでjQueryでDOMを見て座標取得

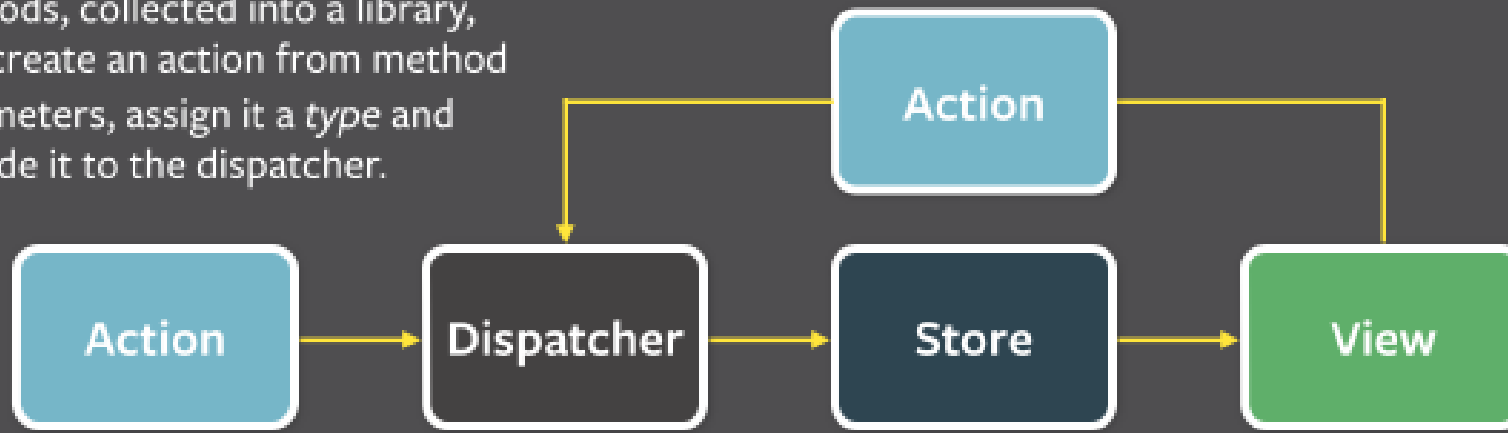
storeを更新、もしくは自分のstateを更新する

```
this.setState({x: 36, y: 250}) // 座標をセット
```

再renderされる

Fluxの円環の理を2回回すイメージ

Action creators are helper methods, collected into a library, that create an action from method parameters, assign it a type and provide it to the dispatcher.



Every action is sent to all stores via the *callbacks* the stores register with the dispatcher.

After stores update themselves in response to an action, they emit a *change* event.

Special views called *controller-views*, listen for *change* events, retrieve the new data from the stores and provide the new data to the entire tree of their child views.

(<https://facebook.github.io/flux/docs/overview.html>)

複数人で同時編集

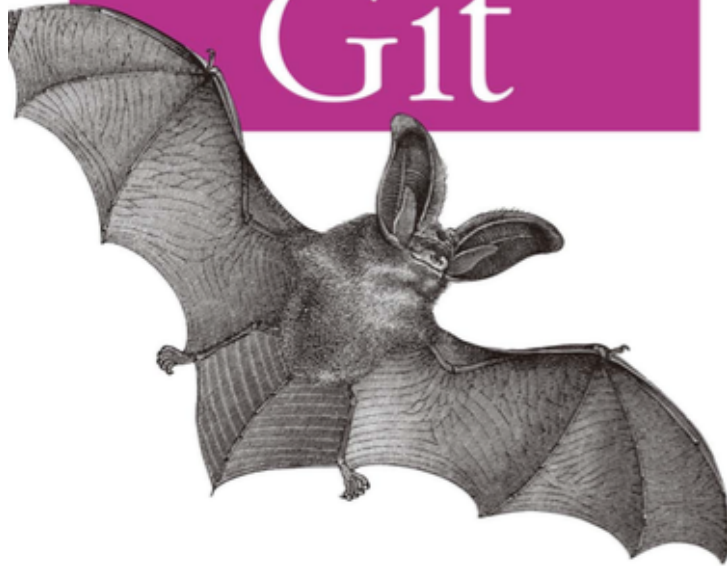
同時編集

実用Git (./実用Git)を読む

*Powerful Techniques for Centralized and
Distributed Project Management*

実用

Git



O'REILLY®
オライリージャパン

Jon Loeliger 著
吉藤 英明 監訳

本間 雅洋、渡邊 健太郎、浜本 隼生 訳

(<https://gyazo.com/20b4fe90c13dafd705d1f75d02adb4de>)

こういう「コミット」をやりとりすればなんとかなるのでは！？

commit.json

```
{  
  "id": hash,  
  "parent_id": parent_hash, // 1つ前への参照  
  "changes": [ 変更内容, 変更内容, 変更内容 ]  
}
```

編集の命令は3つだけ

insert

隣の行のidを指定して新しい行を追加

javascript

```
{
  _insert: positionId, // 文字を挿入する一つ下の位置にある行
  のId
  lines: {
    text: text,
    id: lineId // 新しく生成された行のid
  }
}
```

新規に行を挿入する

最後に挿入するときは、特殊IDの `_end` を指定する。

複数行を挿入するときは、insertを複数個作る

update

idを指定して更新

javascript

```
{  
  _update: lineId, // 変更する行のId  
  lines: { text: text }  
}
```

行を変更する

delete

idを指定して削除

javascript

```
{  
  _delete: lineId, // 削除する行のId  
  lines: -1  
}
```

行を削除する

コンフリクトしたら？

ユーザー側でmerge画面にするわけにはいかない
コンフリクト判定はサーバー側
clientはとりあえずpushする
pushがrejectされたらpull
git rebase のような操作をする

気合マージ

mergeしよう→コンフリクト→pullしてみる

→insertが参照していた行が削除されてる、どこにinsertしていいのかわからない！

→commitをさかのぼって、「削除された行を参照したinsertをした行」を参照してinsertする

Slack通知

差分だけ通知する

commitの中のinsert/update系をmergeして並べ替える



Gyazz BOT 2:26 AM

New lines on [Nota](#)

Toshiyuki Masui

[ページを綺麗に公開する](#)

こういう方針でいいのか?

- GyazzテキストをすっぴんなHTMLに変換
- カッコいいCSSをお気に召すまま付加



Gyazz BOT 2:39 AM ☆

New lines on [Nota](#)

Sho Hashimoto

[ページを綺麗に公開する](#)

ログインせずアクセスすると、同じ見目で編集できないページが表示されるようにする **shokai**

- ページ毎にメニューから「publish」する
- URLは同じ
- とりあえず単一のページの公開からやる
- wiki全体を公開設定も（後で）
- 関連ページやリンク先は・・・publishされてなければ表示しない方が良い？

undo操作

自分がやった操作のみをundoしたい
同時編集集中の他人の操作はundoしたくない

revertコミットを作って貯めておく

pushしてコンフリクトしなかったcommit

insert

同じ行idをdelete

update

元のテキストにupdate

delete

元の位置にinsert

ctrl-Z で逆順に適用する

おわり・まとめ

WYSIWYGで画像やリンクも貼れるけど、編集ツールバーやボタンが無いエディタ

Socket.IOと自作の同期システムでうまいこと同時編集できる



(<https://gyazo.com/0e7cd8cde4a705f8d5fa46b94053486a>)

We are hiring

<https://www.wantedly.com/projects/52220>
(<https://www.wantedly.com/projects/52220>)

京都研究開発拠点



NOTA

(<https://gyazo.com/66e38989c110a0c6386e2f0c20dd6b75>)
五山の送り火で有名な大文字山が見える
フルリモート可



(<https://gyazo.com/715507b411a50c80e5a6ac9f9fc2cd35.png>)