jQuery to Backbone

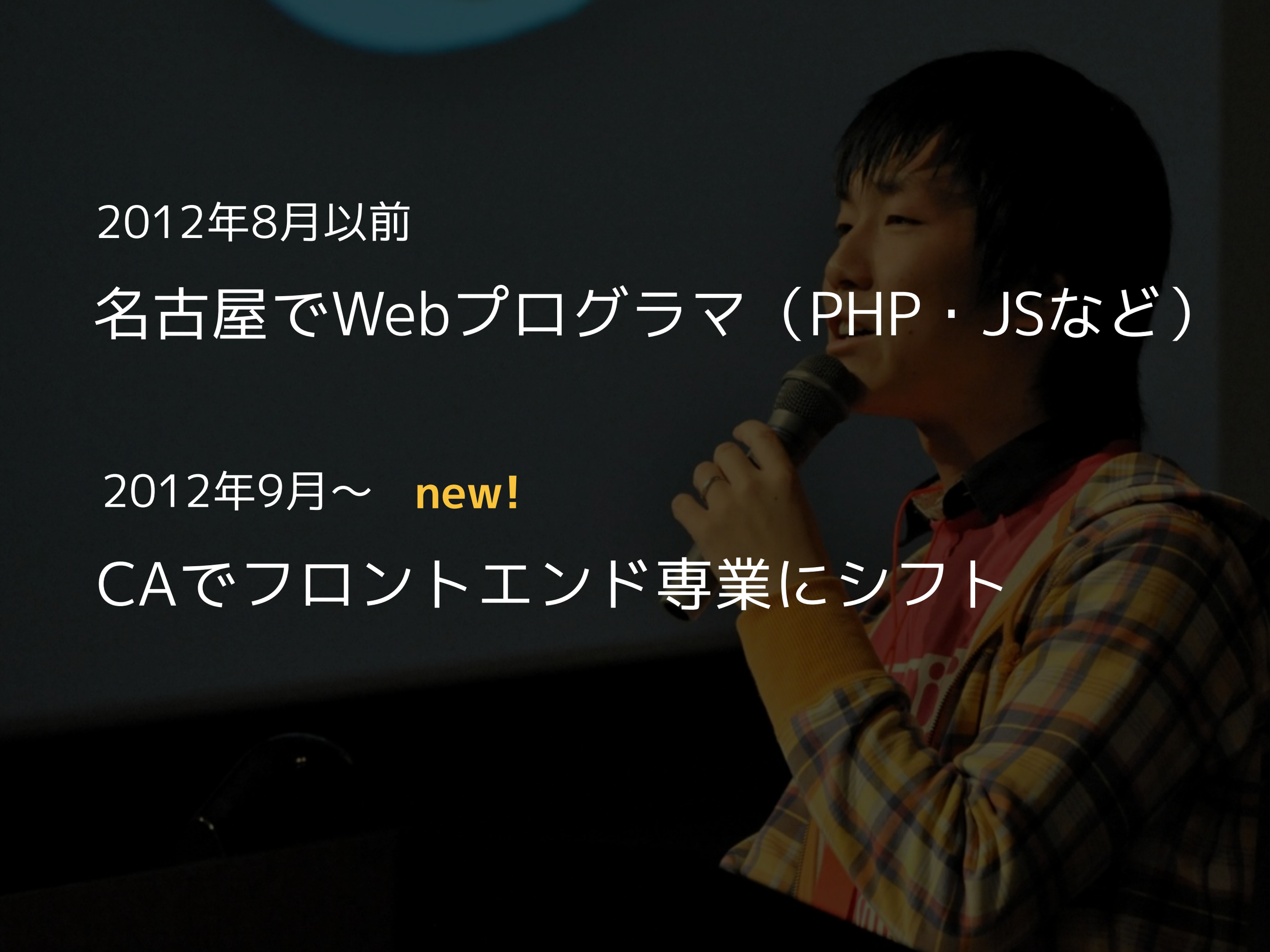
アーキテクチャを意識したJavaScript入門

Frontrend Vol.4

さとう歩 / @ahomu
CyberAgent, Inc.

さとう歩

@ahomu



2012年8月以前

名古屋でWebプログラマ（PHP・JSなど）

2012年9月～　**new!**

CAでフロントエンド専業にシフト

2012年後半

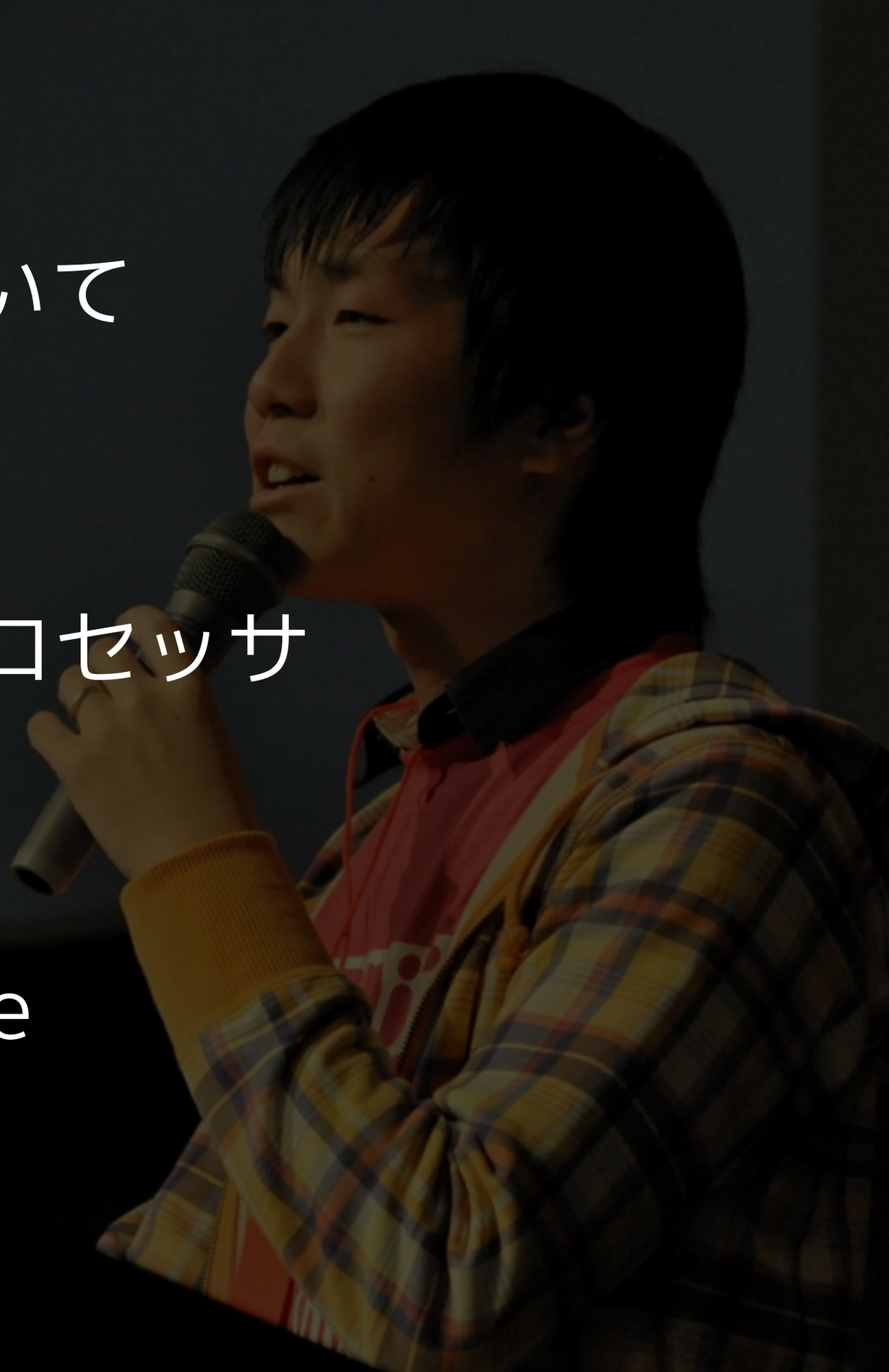
Gruntと自動化について

2013年1月

Stylus・CSSプリプロセッサ

2013年2月　**new!**

jQuery to Backbone



詳しくは

http://aho.mu

流れ

1. はじめに
2. jQueryについて
3. Backboneについて
4. jQuery to Backbone
5. まとめ

はじめに

jQueryは無くなりません

関心のフォーカスを移す機会

易しめに聞ける内容（のつもり）


BACKBONE.JS

BACKBONE.JS

Backbone.js gives structure to web applications by providing **models** with key-value binding and custom events, **collections** with a rich API of enumerable functions, **views** with declarative event handling, and connects it all to your existing API over a RESTful JSON interface.

The project is hosted on GitHub, and the annotated source code is available, as well as an online test suite, an example application, a list of tutorials and a long list of real-world projects that use Backbone. Backbone is available for use under the MIT software license.

You can report bugs and discuss features on the GitHub issues page, on Freenode IRC in the `#documentcloud` channel, post questions to the Google Group, add pages to the wiki or send tweets to @documentcloud.

Backbone is an open-source component of DocumentCloud.

Downloads & Dependencies (Right-click, and use "Save As")

Development Version (0.9.10)	*56kb, Full source, lots of comments*
Production Version (0.9.10)	*6.3kb, Packed and gzipped*
Edge Version (master)	*Unreleased, use at your own risk* build status passing

BACKBONE.JS

http://backbonejs.org/

Initial Release
2010/10/13
Gzipped Size
6.3KB
Latest Version
0.9.10

構造化の手段を提供するモダンライブラリ

View, Model, Collection等を備える

1500行程度の軽量さと
いじりやすい柔軟さ

国内外を問わず利用が広がっている

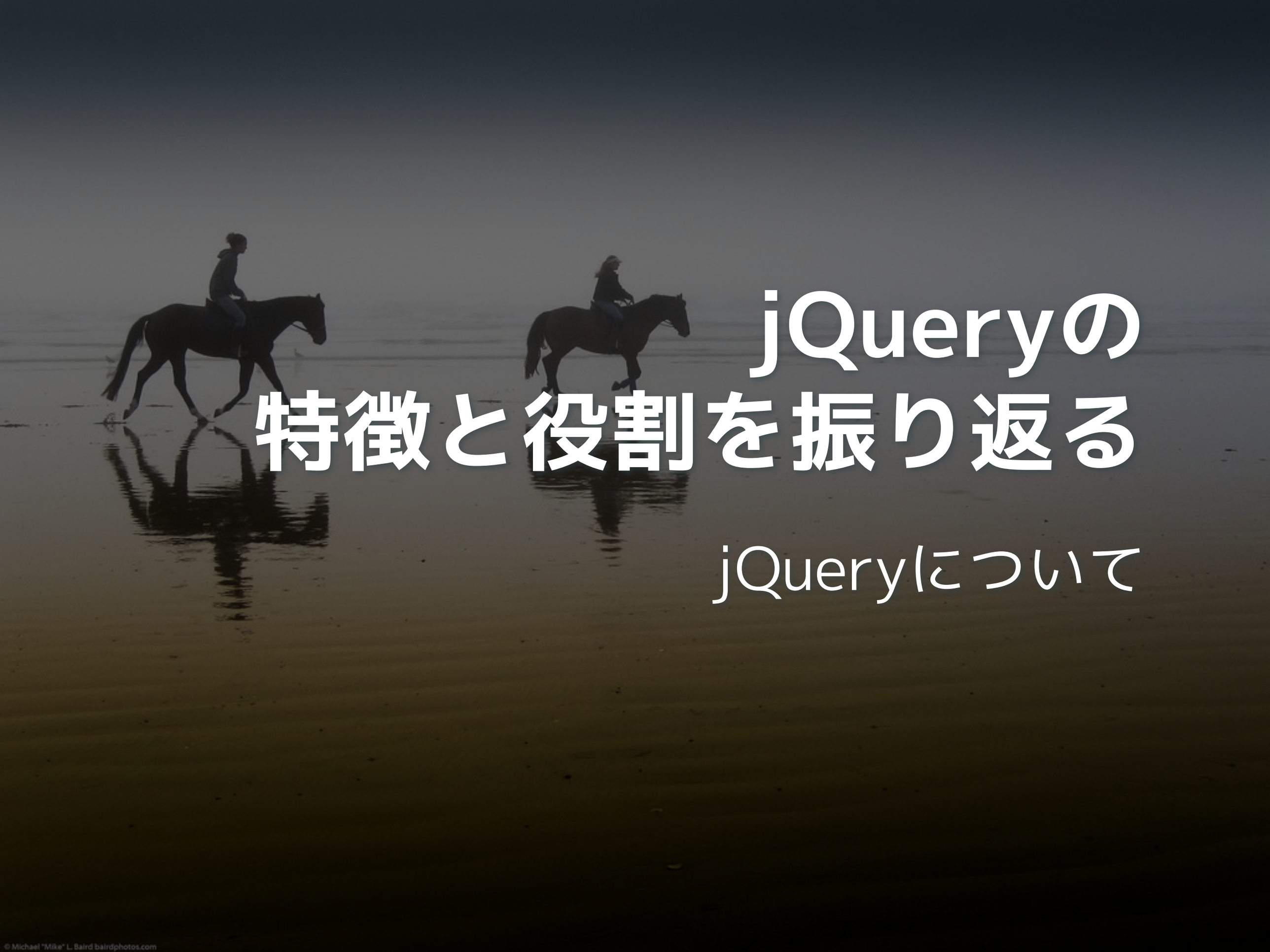
依存するライブラリ

Selector Based Library

- ✓ **jQuery**
- ✓ Zepto.js (lightweight clone)

Utility Belt Library

- ✓ **Underscore.js**
- ✓ Lodash (more faster)



jQueryの特徴と役割を振り返る

jQueryについて

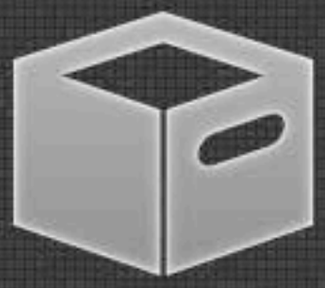
Plugins | Contribute | Events | Support | jQuery Foundation

jQuery

write less, do more.

Download | API Documentation | Blog | Plugins | Browser Support

Search jQuery



Lightweight Footprint

Only 32kB minified and gzipped. Can also be included as an AMD module

CSS3 Compliant

Supports CSS3 selectors to find elements as well as in style property manipulation

Cross-Browser

IE, Firefox, Safari, Opera, Chrome, and more

Download jQuery

v1.9.0 32kB Minified

Download Uncompressed jQuery

View Source on GitHub →

How jQuery Works →

What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensbility, jQuery has changed the way that millions of people write JavaScript.

Who's Using jQuery

Resources

- jQuery Blog
- Contribute to jQuery
- About the jQuery Foundation
- Browse or Submit jQuery Bugs

Other jQuery Foundation Projects

http://jquery.com/

特徴と役割、3つのポイント

①DOM APIを隠蔽して簡潔に記述する

write less, do more.

DOM API は煩雑すぎた

```
elemNode.parentNode.removeChild(elemNode);
```

↓

```
$(elemNode).remove();
```

②クロスブラウザ対応の諸問題を解決する

IE, Firefox, Safari, Opera, Chrome…

かつての立役者の功罪ほか細々

Msxml.XMLHTTP? attachEvent?

↓

$.ajax, $.bind/$.on

③プラグインの充実と
エコシステム形成

Useful and Awesome Plugins!

プラグインがあれば何でもできる

User / Community / Ecosystem

↓

```
$.fn.awesomePlguin('feel good!');
```

jQueryが生まれた頃の問題を解決してくれた

意識することは少なくなってきている

jQueryについての所感

http://havelog.ayumusato.com/develop/javascript/e333-jquery_thiking_misc.html

フロントエンド実装の現状と変化

今、求められるスキルと取り巻く状況

Web
サイト

静的HTML
CMS利用

従来の
Webサービス
Webアプリ

ややリッチな
インターフェース

新しめな
Webサービス
Webアプリ

シングルページ
リッチ＆シームレスなUI
RESTful API

次第に高まってきた フロントエンド実装の比重

静的HTML
CMS利用

サーバーサイドで
画面遷移を設計

シングルページ
リッチインターフェース
RESTful API

フロントエンドに
生まれる新たな問題

Another new problems...

jQueryが解決しない問題

アプリケーションコードの肥大化
スパゲティコードの技術的負債
テスタビリティーの確保
メンテナンスのたびに深まる業

新しい問題のために向けるべき関心？

“

Most Developers realize that structured, maintainable code is important.

ディベロッパーにとって、構造的で
メンテナンスしやすいコードが何たるかを知ることが重要。

via. Digesting JavaScript MVC
https://speakerdeck.com/addyosmani/digesting-javascript-mvc?slide=10

MV*なJavaScriptとアーキテクチャ云々

Backbone.jsについて

アーキテクチャと言えば
猫も杓子もMVC

みんな1度は聞いたことあるはず

From: Trygve Reenskaug

Date: 10 December 1979

MODELS - VIEWS - CONTROLLERS

MODELS

Models represent knowledge. A model could be a single object (rather uninteresting), or could be some structure of objects. ~~The proposed implementation supports knowledge represented in something resembling *semantic nets* (If I understand Laura correctly)~~

There should be a one-to-one correspondence between the model and its parts on the one hand, and the represented world as perceived by the owner of the model on the other hand. The nodes of a model should therefore represent an identifiable part of the problem.

The nodes of a model should all be on the same problem level, it is confusing and considered bad form to mix problem-oriented nodes (e.g. calendar appointments) with implementation details (e.g. paragraphs).

VIEWS

A view is a (visual) representation of its model. It would ordinarily highlight certain attributes of the model and suppress others. It is thus acting as a *presentation filter*.

A view is attached to its model (or model part) and gets the data necessary for the presentation from the model by asking questions. It may also update the model by sending appropriate messages. All these questions and messages have to be in the terminology of the model, the view will therefore have to know the semantics of the attributes of the model it represents. (It may, for example, ask for the model's identifier and expect an instance of Text, it may not assume that the model is of class Text.)

CONTROLLERS

A controller is the link between a user and the system. It provides the user with input by

MVC since 1979

From: Trygve Reenskaug

古典 of Smalltalk

Smalltalk（スモールトーク）は、Simulaのオブジェクト（およびクラス）
Lispの機能、LOGOのエッセンスを組み合わせて作られたクラスベースの純粋
オブジェクト指向プログラミング言語、および、それによって記述構築された
統合化プログラミング環境の呼称。

via. Smalltalk - Wikipedia
http://ja.wikipedia.org/wiki/Smalltalk

JavaScriptにおける MVCまたはMV*の現状

普及してるとは言いがたいが...

TodoMVC

Helping you **select** an MV* framework

Download (1.0.1) | View project on GitHub

Introduction

Developers these days are spoiled with choice when it comes to selecting an **MV* framework** for structuring and organizing their JavaScript web apps.

Backbone, Ember, AngularJS, Spine... the list of new and stable solutions continues to grow, but just how do you decide on which to use in a sea of so many options?

To help solve this problem, we created **TodoMVC** - a project which offers the same Todo application implemented using MV* concepts in most of the popular JavaScript MV* frameworks of today.

Tweet 767 +1

JavaScript Apps

- Backbone.js
- Knockback.js
- AngularJS
- CanJS
- Ember.js
- Maria
- KnockoutJS
- cujo.js
- Dojo
- dermis
- YUI
- Montage
- Agility.js
- Ext.js

Compile To JavaScript

- Spine
- Batman.js

Todo MVC

http://addyosmani.github.com/todomvc/

- Sammy.js
- PlastronJS
- Stapes
- Dijon
- Epitome
- rAppid.js
- soma.js
- Funnyface.js
- DUEL
- Knockout + ClassBinding
- PureMVC
- Olives
- AngularJS (optimized)

- TypeScript + Backbone.js
- Closure

巷に溢れかえる
MV*フレームワーク

TodoMVCだけで30以上が列挙される

MV*なアーキテクチャは身近になってきている

MV*に限らず、色々考え出されている

Home | Download / Install | Tutorials | Live examples | Documentation | Forum | Sou

Knockout.

Simplify dynamic JavaScript UIs by applying the Model-View-View M (MVVM) pattern

Download
v2.2.1 - 14kb min+gz

y concepts

Knockout

http://knockoutjs.com/

Declarative Bindings

Easily associate DOM elements with model data using a concise, readable syntax

Automatic UI Refresh

When your data model's state changes, your UI updates automatically

Dependency Tracking

Implicitly set up chains of relationships between model data, to transform and combine it

Templating

Quickly generate sophisticated, nested UIs as a function of your model data

re features

- Free, open source (MIT license)
- Pure JavaScript — works with any web framework

New: Interactive tutorials

Get started with knockout.js quickly, learning to build *single-page applicatio* *custom bindings* and more with these interactive tutorials.

batman.js

Overview Download Examples Documentation Commun

uild web apps beautifully and quickly

tman.js is a framework for building rich web applications with CoffeeScript
JavaScript. App code is concise and declarative, thanks to a powerful
stem of view bindings and observable properties. The API is designed with
veloper and designer happiness as its first priority.

ant to learn more? Check out **all the features** or some **examples**.

Download batman.js

Latest version: 0.11.2
Updated: July 17th, 2012
Patch release addressing some recent

View the release notes or the code.

batman.js

http://batmanjs.org/

ere's some code:

```
lass Shopify extends Batman.App
 @root 'products#index'
 @resources 'products'

lass Shopify.Product extends Batman.Model
 @persist Batman.RestStorage
```

From the Blog

**Batman.js 0.9.0: The biggest thing since
Killer Croc**

Posted 02 April 2012 by Nick Small

batman.js 0.9.0 is available on github and n

ember

ABOUT GUIDES API COMMUNITY BLOG FORK US

A framework for creating
ambitious web applications.

DOWNLOAD EMBER 1.0.0-PRE.4

47k min+gzip | minified

Ember.js

http://emberjs.com/

MORE PRODUCTIVE OUT OF THE BOX.

Write dramatically less code with

Don't waste time making trivial

Ember.js is built for productivity

AngularJS

http://angularjs.org/

ny AngularJS?

Alternatives

Extensibility

Search or type a command

Explore Gist Blog Help

ahomu

Account settings

Repositories Members

Find a Repository...

All Sources Forks Mirrors

aura

A scalable, event-driven JavaScript architecture for developing widget-based applications. Works wit
Backbone.js and other frameworks.

Last updated 4 days ago

JavaScript ★ 1,921

aura-identity

Visual Resources for the Aura Project

Last updated 23 days ago

★ 1

aura

https://github.com/aurajs/aura

urajs

Joined on Dec 28, 2012

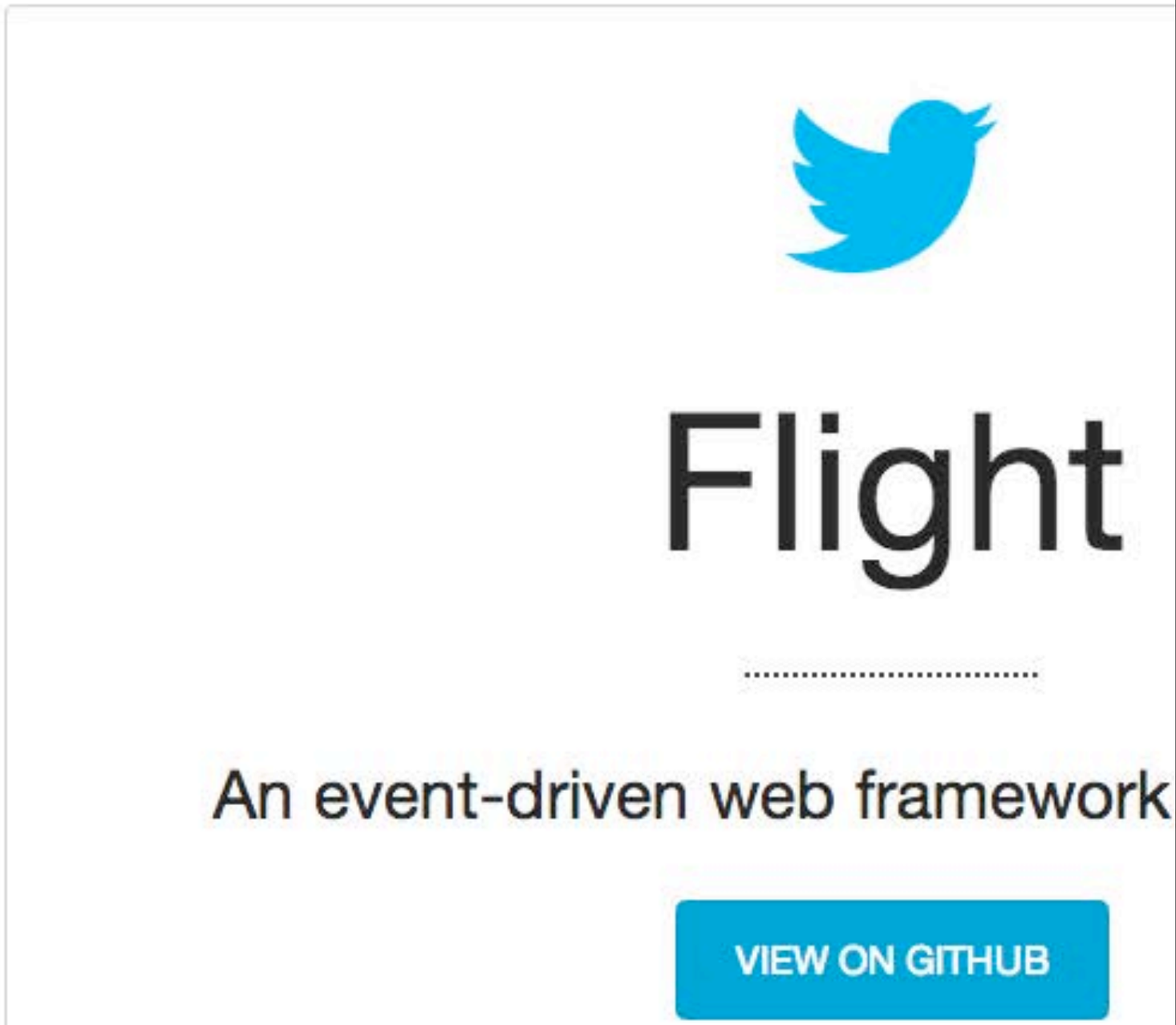
11
members

Applications
GitHub for Mac
GitHub for Windows
GitHub for Eclipse
GitHub mobile apps

Services
Gauges: Web analytics
Speaker Deck: Presentations
Gist: Code snippets
Job board

Documentation
GitHub Help
Developer API
GitHub Flavored Markdown
GitHub Pages

More
Training
Students & teachers
The Shop
Plans & pricing
The Octodex



Flight

http://twitter.github.com/flight/

MV*に関した議論は
定期的に盛り上がる

そして常に変化しつづけている

gist Search... Discover Gists ahomu

tily / scaling_isomorphic_javascript_code.ja.markdown
Last updated 8 days ago

Star 79 Fork 16

サバクラ両方で動く JavaScript の大規模開発を行うために

Gist Detail
Revisions 9
Stars 79
Forks 16

Download Gist

Clone this gist
https://gist.github.

Embed this gist
<script src="https:/

Link to this gist
https://gist.github.

scaling_isomorphic_javascript_code.ja.markdown Markdown

サバクラ両方で動く JavaScript の大規模開発を行うために

原文：Scaling Isomorphic Javascript Code (This is just for study, please contact me at tily05 atmark gmail.com if any problem.)

考えてみれば Model-View-Controller とか MVC ってよく聞くよね。実際どんなものか知ってる？ 抽象的に言うなら「オブジェクト情報の保持されるグラフィック・システム (つまり、ラスターではないグラフィック、ゲームとか) 上に構築された、表示系を中心としたアプリケーションにおいて、主要な機能どうしの関わりをうまく分離すること」とでも言おうか。もう少し深く考えを押し進めてみれば、これは当然、他のさまざまなアプリケーションにもあてはまる言葉 (bucket term ?) だ。

過去に多くの開発コミュニティが MVC による解決案を提供し、それによってよくあるユースケースにうまく対処し、地位を築くことができた。例をあげるなら、Ruby や Python コミュニティは Rails や Django を作り、MVC アーキテクチャを実現した。

こうした手法は Java, Ruby, Python といった他の言語では容易に受け入れらてきたが、Node.js について十分ではない。理由は単純で、それはただ、**JavaScript が今や isomorphic な言語である**からだ。**isomorphic** というのは「ソースコードのどの行 (もちろん注目すべき例外もあるが) をとってみても、クライアント・サーバーの両方で実行できる」ということを意味している。表面的には無害に見えるが、この特徴のせいで現状の MVC ベースのパターンでは解決できない課題がたくさんある。

この記事では、まず現存するパターンをいくつか取り上げ、いかにそのようなパターンに関する実装や心配事が言

サバクラ両方で動く JavaScript の大規模開発を行うために
https://gist.github.com/tily/1362110

FOR NODE.JS

flatiron

latiron is an adaptable framework for building
nodern web applications. It was built from the
ound up for use with **Javascript** and **Node.js**.

Home | Routing | Templating | Data Management | Middleware | Plugins | Logging

flatiron

http://flatironjs.org/

hat's included?

L Routing
imple tool for **directing traffic** based on
Ls that are recieved both on the browser
d server.

mplating
unobtrusive templating tool that is fast,
L free and simple to use.

ta Management
torage agnostic resource-oriented object
document mapper for building **data**
odels with validation and sanitization.

Philosophy

No one agrees on frameworks. It's difficult to get consensus on how much or how little a framework should do. Flatiron's approach is to package simple to use yet full featured components and let developers subtract or add what they want.

Motivation

Build a collection of decoupled, unobtrusive tools that can operate as well in unison as they do independently. Promote code organization and sustainability by clearly separating development con

Getting started with Flatiron

Full project on G

Since flatiron consists of largely a components, it's simple to start using.

おまえのMVCは
間違っている！

おれは新しい
パターンを作る！

このテの議論に
今回はあまり触れません

Backbone的にも厳密すぎなくてOK

BACKBONE.JS は、

✕ 厳格さと多機能で生産性を
担保する強いフレームワーク

◎ やさしい構造化を
サポートする薄いライブラリ

です。

1500行あまりの軽いコード

View Model Collection Router

Controllerの不在

Not MVCフレームワーク

方法は1つではないし、柔軟に拡張できる

it serves as a foundation for your application, you're meant to **extend and enhance it in the ways you see fit**

via. Backbone.js FAQ
http://backbonejs.org/#FAQ-why-backbone

“

MVCを気にしすぎることはない

と、思います。

デザインパターンは
知っておいて損はない

アーキテクチャと併せて意識したい

Abstract Factory
Builder Factory Method
Prototype Singleton
Adapter Bridge
Composite Decorator
Facade Flyweight Proxy

GoFに代表される様々なデザインパターン

一般に共有できるボキャブラリ

Chain of Responsibility
Command Interpreter
Iterator Mediator
Memento Observer
State Strategy Template
Method Visitor

しかし現実は
独自パターンが跋扈する

Everyone have own pattern

デザインパターンから
一般的なパターンを学ぶ

JSの例で、有名どころを少し紹介

JavaScript Design Patterns: Singleton

This is the first in what should be a pretty long series about JavaScript design patterns. In 1995, Erich Game, Richard Helm, Ralph Johnson and John Vlissides (known as the Gang of Four) published *Design Patterns: Elements of Reusable Object-Oriented Software*, a book cataloging recurring solutions to common dilemmas in software architecture and design. It also started a common vocabulary for referring to these solutions. If you'd like to know more you can find it on Wikipedia.

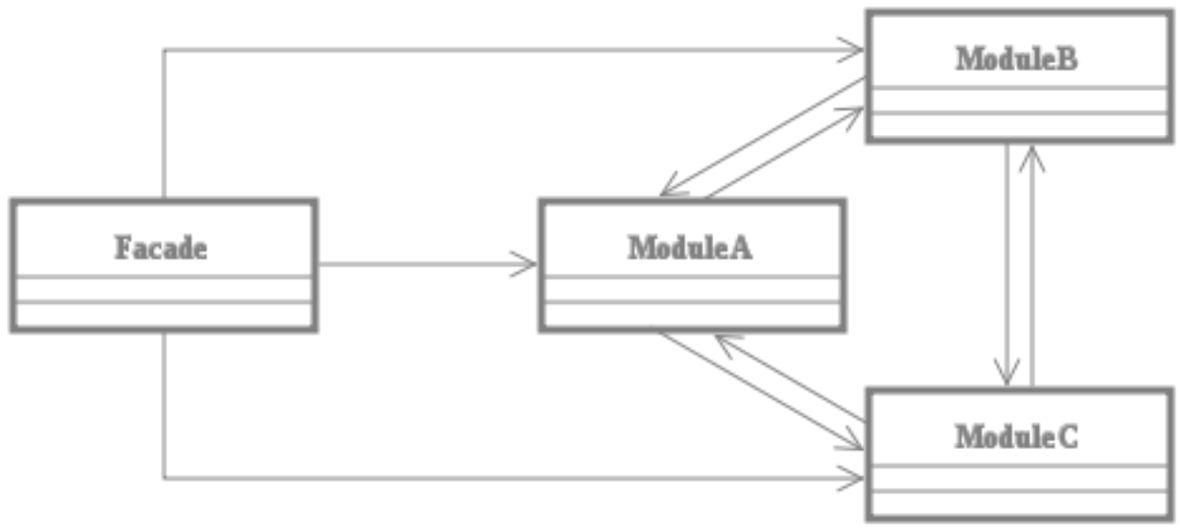
That book's example implementations of each of the solutions were written in C++ and Smalltalk, which are quite different than JavaScript. Another book – *Pro JavaScript Design Patterns* – was written to bring many of those patterns into the context of JavaScript. My hope is to present a lot of the knowledge from that book here, but not so much that I get sued... just enough to keep you interested and possibly get you to buy the book. If you do buy the book, let them know it was because I referred you. Maybe they'll provide me a bit of compensation (probably not, but here's hoping).

JavaScript Design Patterns

http://www.joezimjs.com/javascript/javascript-design-patterns-singleton/

Facade

一番簡単な構造のパターン



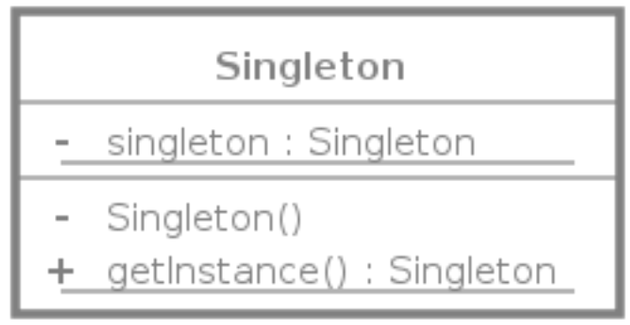
Facade - 複数の処理をまとめる

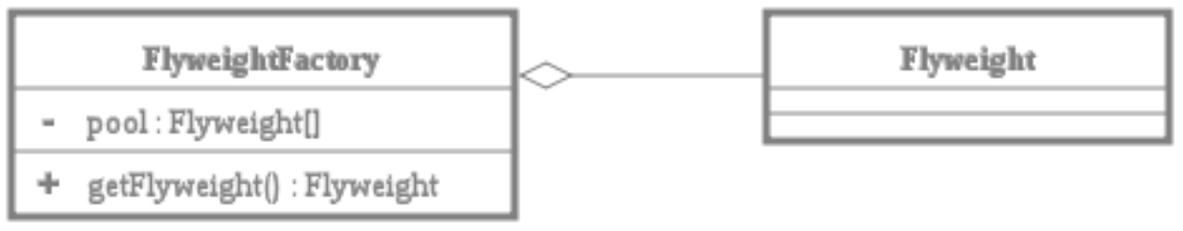
```
// クリックしたときに複数の操作を同時に行う
// 最も広義には誰もがあたりまえに行うパターン
onLikeClick: function() {
  this.likeModel.save();
  this.changeState('liked');
  this.$el.addClass('animate');

  //  ...
  //  ..
  //  .
}
```

Singleton/Flyweight

生成および構造のパターン





Singleton - ただひとつのオブジェクト

```
// 簡易には、不変的オブジェクトになるために
// コンストラクタなしでオブジェクトリテラルで表現
var singletonClass = {
  init: function() {
    // initialize logic
  },
  method: function() {
    // some logic
  }
}
singletonClass.init();
```

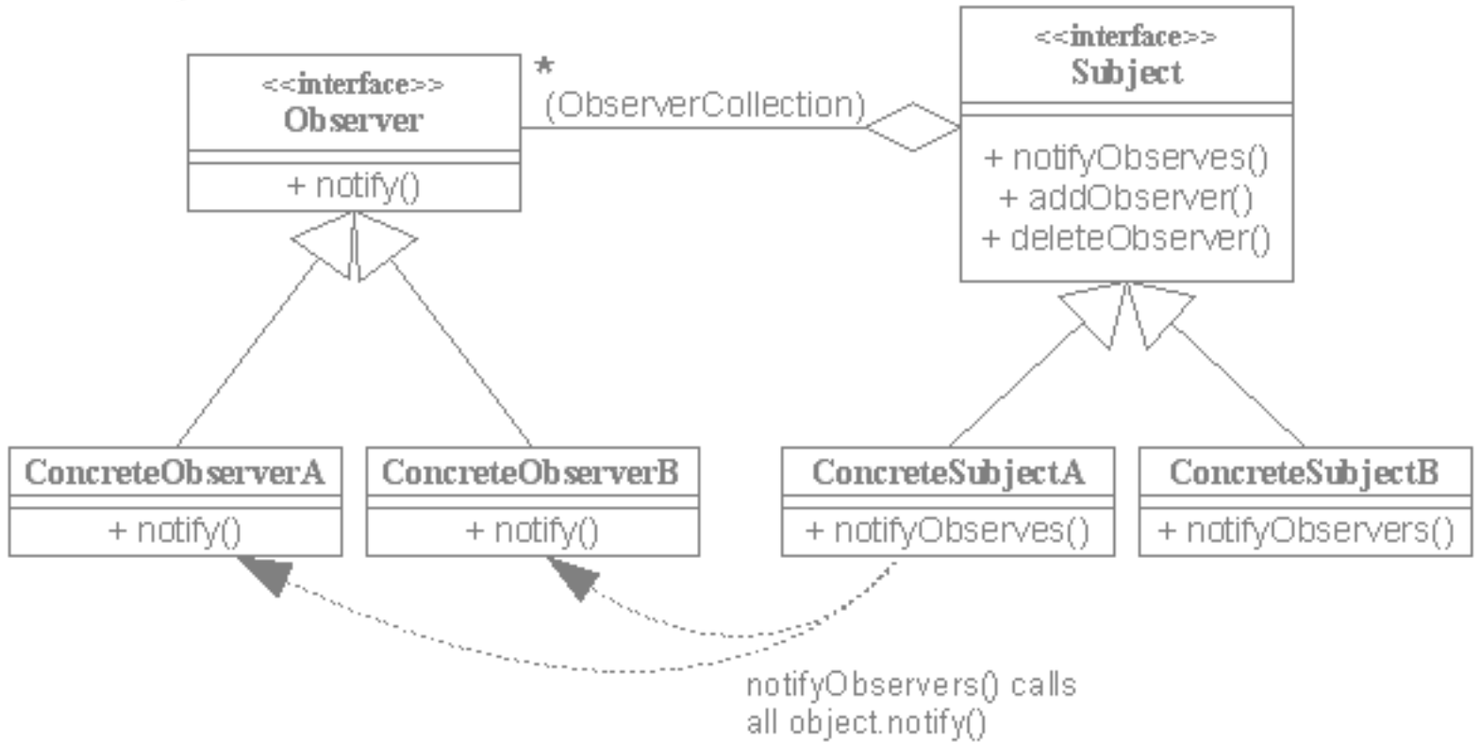
Flyweight - インスタンス生成

```
// 動的にインスタンスを作りたい場合など
// 同じ役割のインスタンスは複数生成させずに使い回す
var factory1 = Flyweight.getFactory('model');
var factory2 = Flyweight.getFactory('view');
var factory3 = Flyweight.getFactory('model');
```

```
// 期待する動作として、modelとして呼んだら同一であること
console.log(factory1 === factory2); //=> false
console.log(factory1 === factory3); //=> true
```

Observer/Mediator

振る舞いのパターン



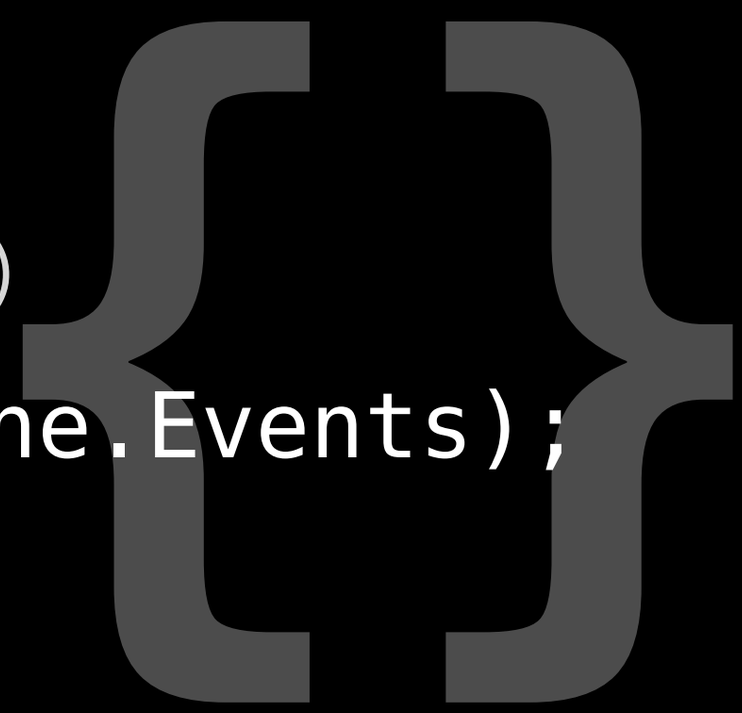
Observer - イベントを監視

```
// 監視される側のModelと、監視する側のView
var model = new FooModel();
var view  = new BarView();

// Modelに直接、Viewのメソッドをリスナーで登録
model.on('change', view.render);

// fetch()によって、changeイベントがトリガー
model.fetch(); // view.render()が実行される
```

Mediator - イベントを仲介



```
// Mediatorを作成（Backboneの例）
var mediator = _.clone(Backbone.Events);

// Mediatorにリスナーを登録
mediator.on('model:change', view.render);
```

```
// Mediatorを通してイベントをトリガー
mediator.trigger('model:change', this);
```

急に当てはめるのは
難しいことがほとんど

何らかの取っかかりが必要、そこで...

やさしい構造化を
サポートするBackboneで
パターンやアーキテクチャの
実践を始めてみると吉

jQuery to Backbone
コードを構造化する

学ぶためのリファクタリング

Backbone.jsにおける コンポーネント

View, Model, Collection, Router

View

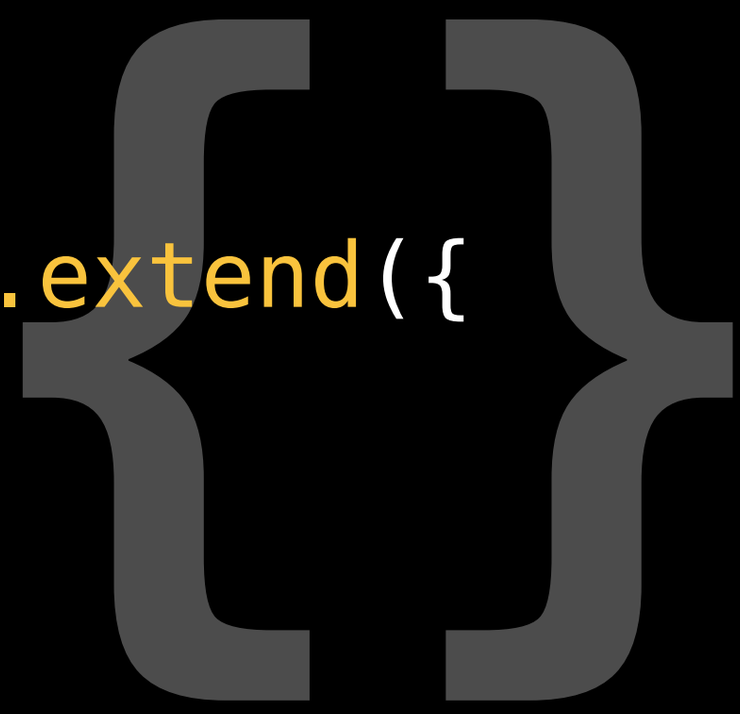
Backbone.View

見た目とUIにおける入出力

DOM要素の管理

ユーザー操作(イベント)制御

典型的なView



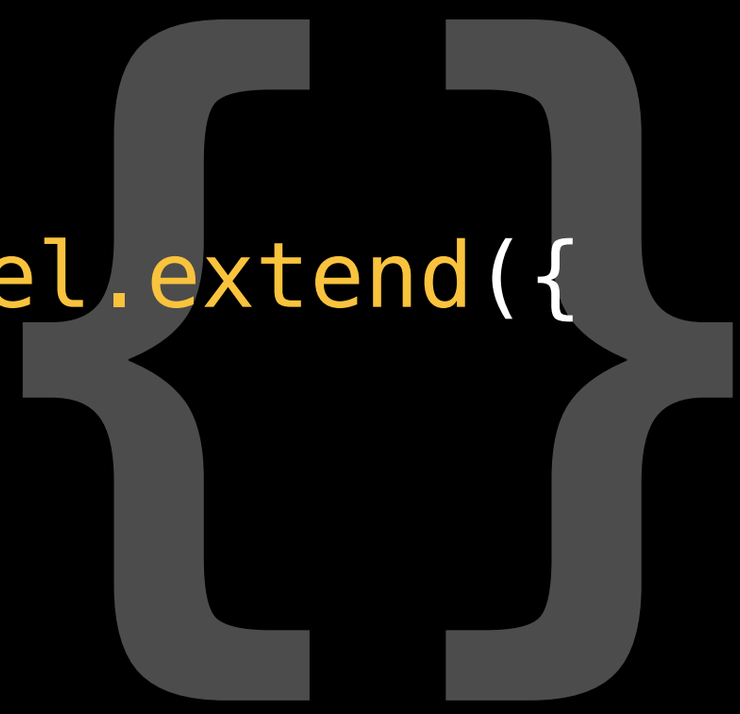
```
var ViewClass = Backbone.View.extend({
  el: ‘#main’,
  initialize: function() {
    // process...
  },
  render: function() {
    this.$el.html(‘rendering html strings’);
  }
});
var view = new ViewClass(); // `initalize`
view.render(); // `render`
```

Model

Backbone.Model

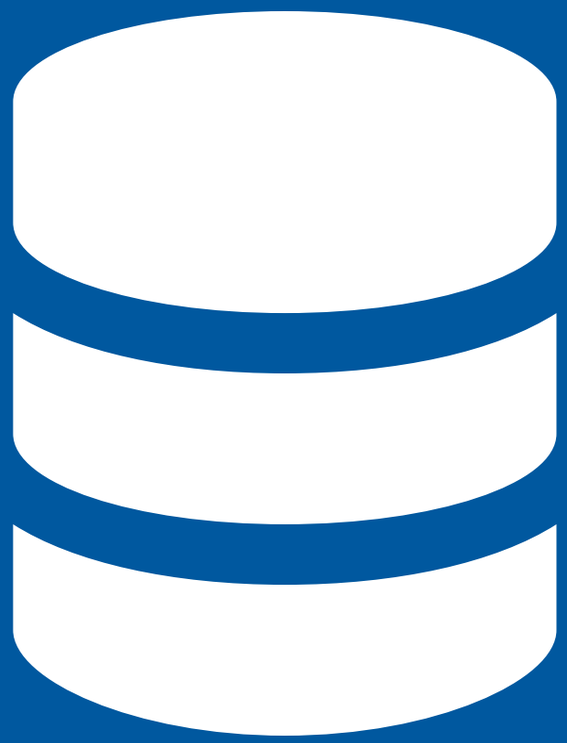
取り扱うデータの一単位
ストレージとの通信・同期
APIや情報のレコードを表現

典型的なModel



```
var ModelClass = Backbone.Model.extend({
  defaults: {},
  url: ‘api/v1/path/to’,
  initialize: function() {
    // process...
  }
});
var model = new ModelClass(); // `initalize`
var view = new ViewClass({model: model});
model.fetch({
  success: view.render
});
```

Collection



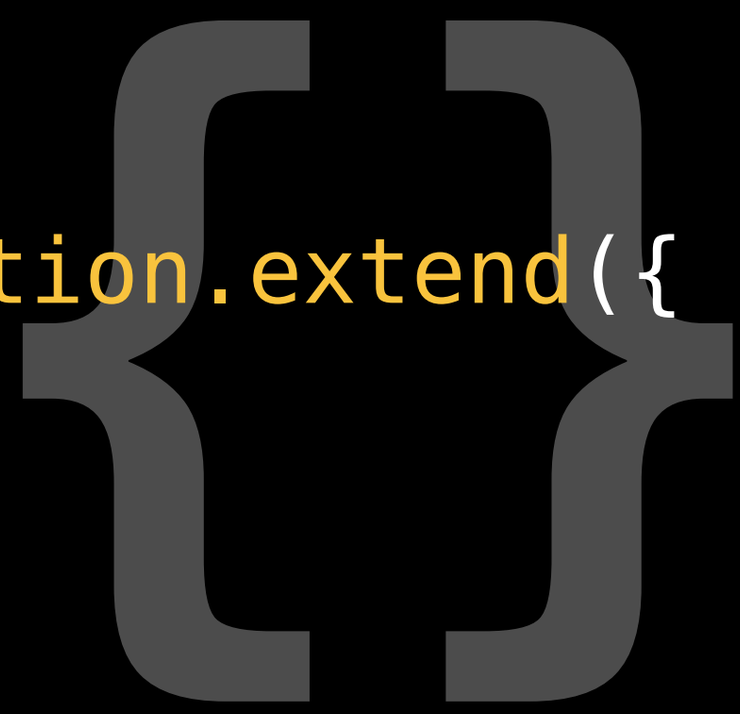
Modelが集合したリスト

リスト操作...where, filterなど

Modelと同様の通信・同期

Backbone.Collection

典型的なCollection



```
var Persons = Backbone.Collection.extend({
   url: ‘api/v1/path/to’,
   model: Person
});
var persons = new Persons();
persons.fetch({
   success: function() {
      this.where({
         name: ‘anonymous’
      })[0].sayName(); // ‘anonymous!’
   }
});
```

Router

Backbone.Router

URLによるルーティング

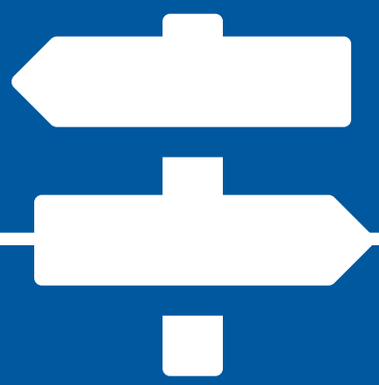
hashchange, pushstate

遷移処理のnavigate

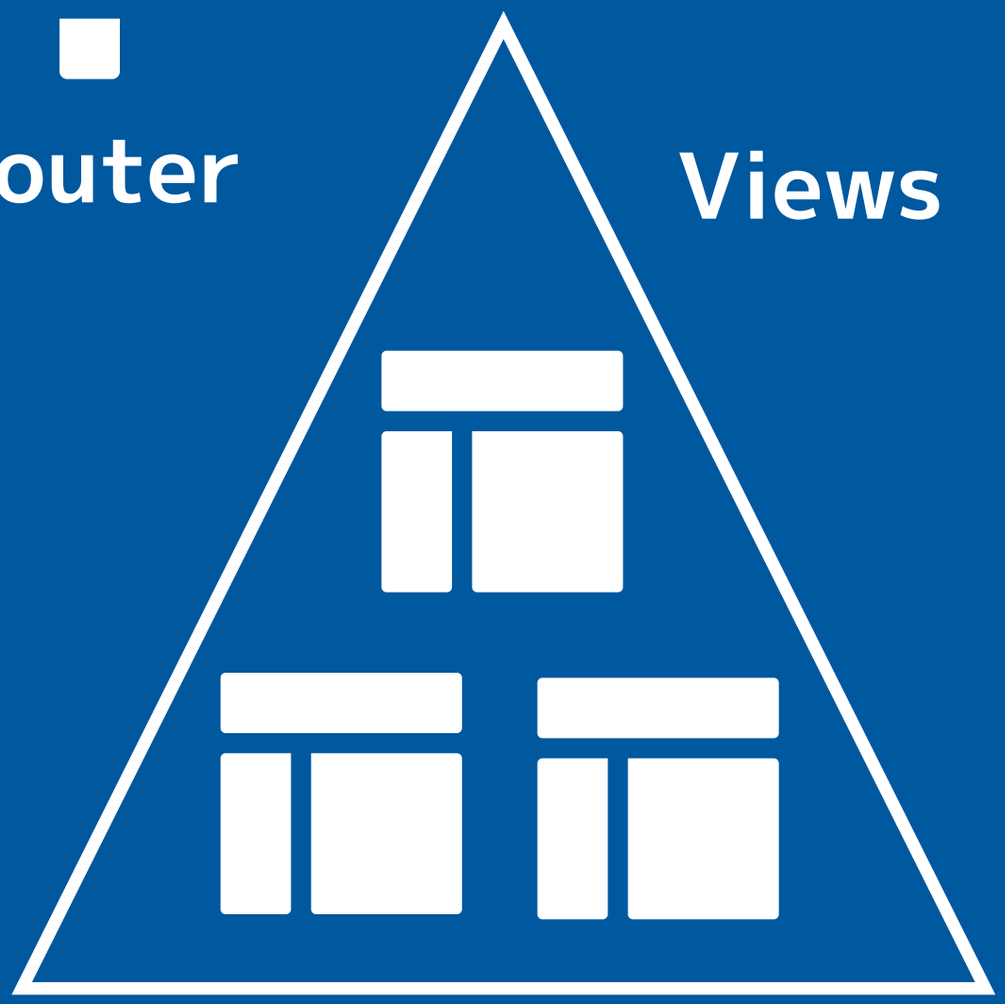
典型的なRouter

```
var Router = Backbone.Router.extend({
  routes: {
    'store/:storeId': 'gotoStore'
  },
  gotoStore: function(storeId) {
    new StoreView({
      model: new Store(storeId);
    });
  }
});
var app = new Router();
Backbone.history.start();
```

Backbone



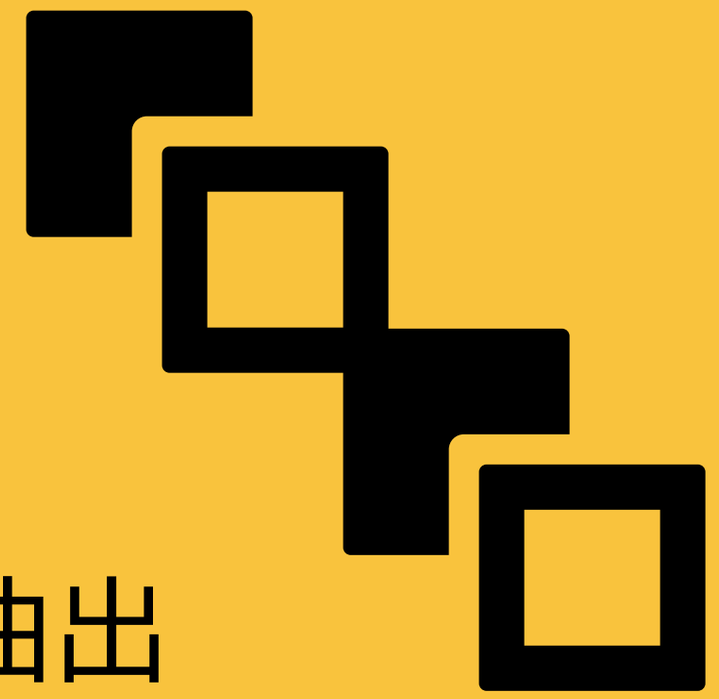
Router



Views

via. Backbonification - Migrating NewsBlur From DOM Spaghetti to Backbone.js
https://speakerdeck.com/samuelclay/backbonification-migrating-newsblur-from-dom-spaghetti-to-backbone-dot-js?slide=12

Backbone.jsを実際に使ってみる



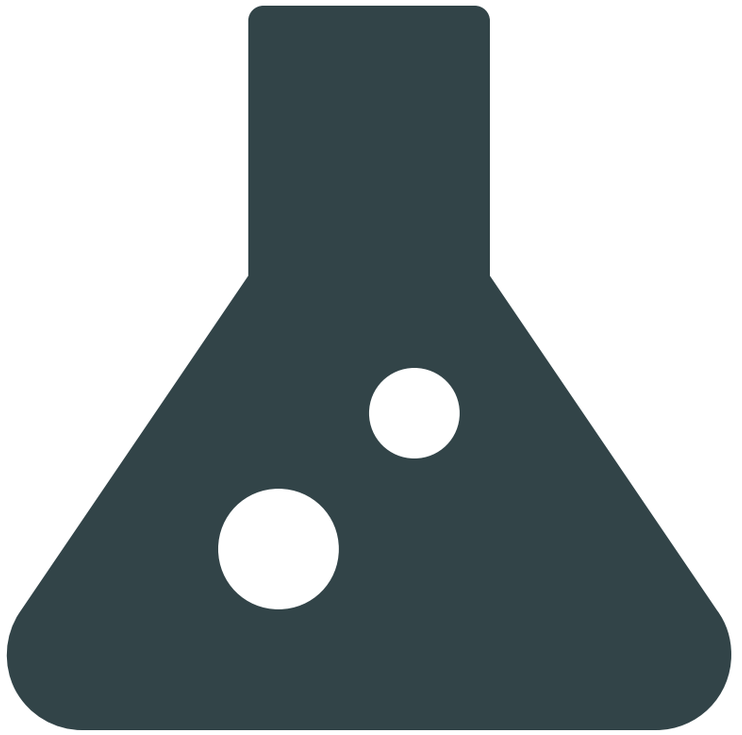
Viewの分離とメソッドの抽出

GitHub APIを使った Gistビューワー

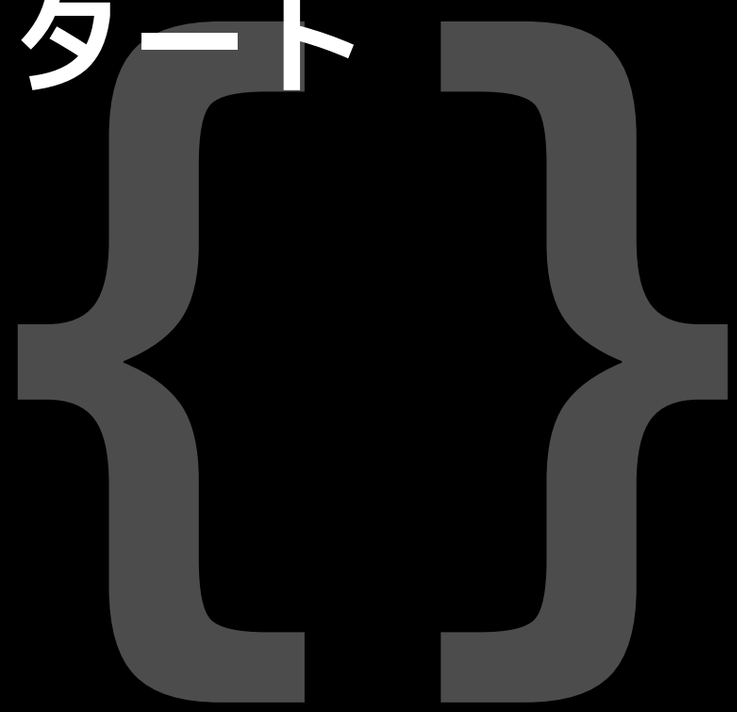
実用性はさておき、あくまでサンプル

DEMO

1. Backbone.Viewを作成
2. renderメソッドを抽出
3. テンプレートの分離
4. イベントの定義

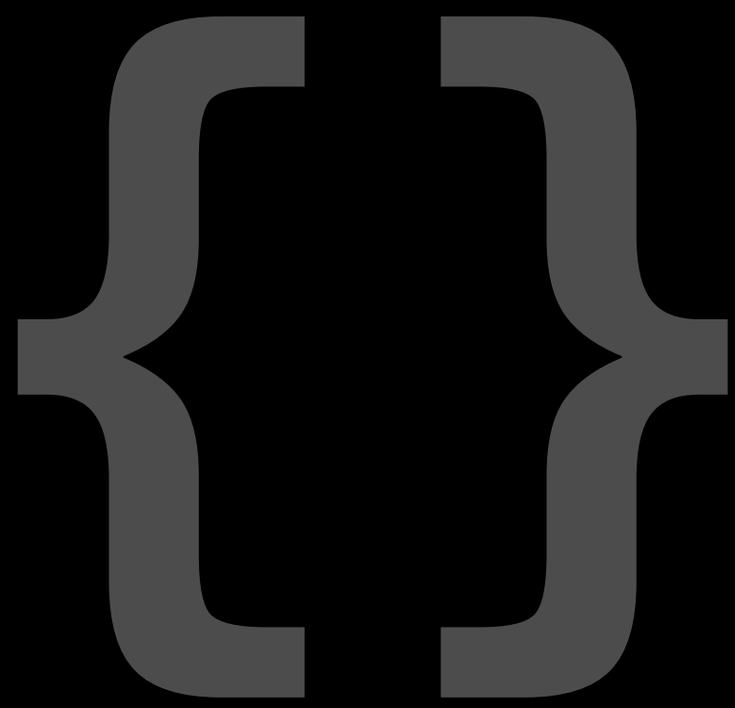


ピュアなjQueryコードからスタート



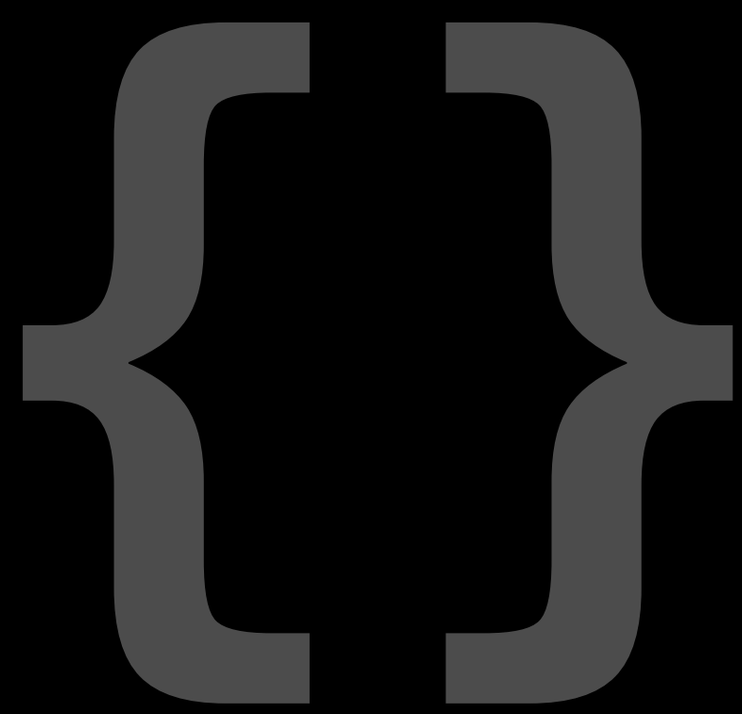
```
var $list    = $('#js-gists');
$.ajax({
  method: 'GET',
  url: 'https://api.github.com/gists',
  data: oauthData,
  dataType: 'json'
}).done(function(data) {
  var i = 0,
      html = '',
      item;
  while (item = data[i++]) {
    html += '<li>'+
      '<a data-src="'+item.url+'" href="#">'+item.description+'</a>'+
      '<a href="'+item.html_url+'">Show in gists</a>'+
      '</li>';
  }
  $list.html(html);
});
$list.on('click', '[data-src]', previewGist);
```

おもむろにViewを作成



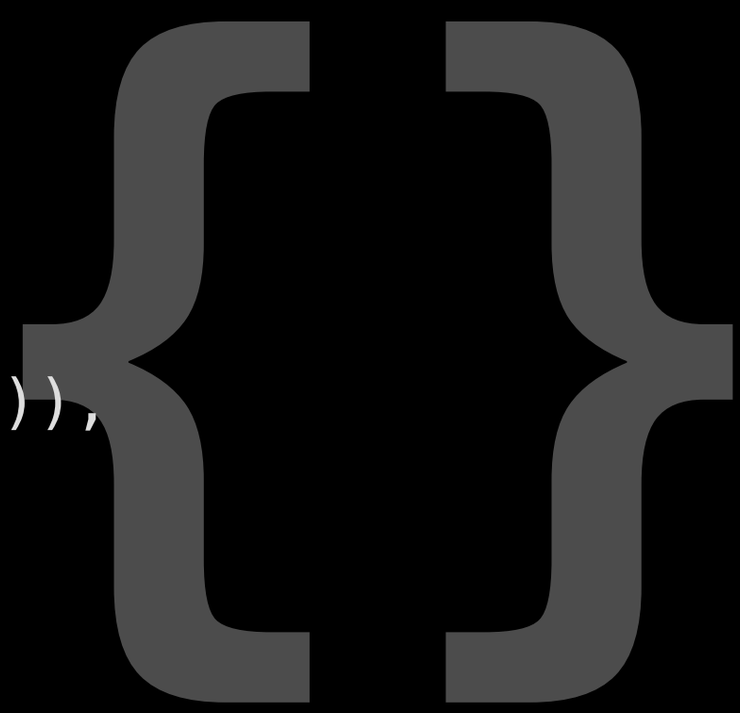
```
var GistsListView = Backbone.View.extend({
  el: '#js-gists',
  initialize: function() {
    var $list = this.$el;
    $.ajax({
      method: 'GET',
      url: 'https://api.github.com/gists',
      data: oauthData,
      dataType: 'json'
    }).done(function(data) {
      var i = 0, html = '', item;
      while (item = data[i++]) {
        html += '<li>'+
          '<a data-src="'+item.url+'" href="#">'+item.description+'</a>'+
          '<a href="'+item.html_url+'">Show in gists</a>'+
          '</li>';
      }
      $list.html(html);
    });
    $list.on('click', '[data-src]', previewGist);
  }
});
var gistsList = new GistsListView();
```

renderメソッドを抽出



```
var GistsListView = Backbone.View.extend({
  el: '#js-gists',
  initialize: function() {
    _.bindAll(this);
    $.ajax({
      method: 'GET',
      url: 'https://api.github.com/gists',
      data: oauthData,
      dataType: 'json'
    }).done(this.render);
    this.$el.on('click', '[data-src]', previewGist);
  },
  render: function(data) {
    var i = 0, html = '', item;
    while (item = data[i++]) {
      html += '<li>'+
        '<a data-src="'+item.url+'" href="#">'+item.description+'</a>'+
        '<a href="'+item.html_url+'">Show in gists</a>'+
        '</li>';
    }
    this.$el.html(html);
    return this;
  }
}); var gistsList = new GistsListView();
```

テンプレートの分離

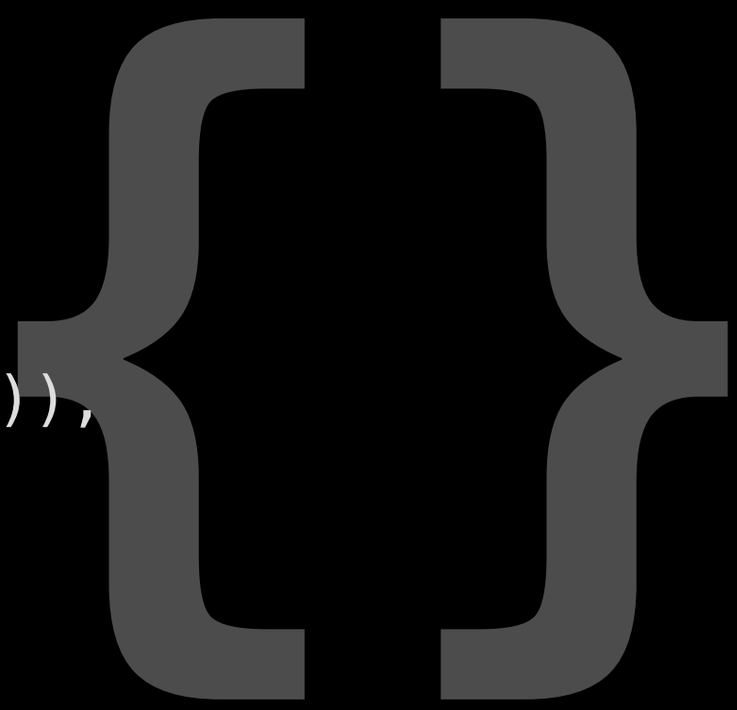


```
var GistsListView = Backbone.View.extend({
  el: '#js-gists',
  tmpl: _.template($('#tmpl-js-gists').html()),
  initialize: function() {
    _.bindAll(this);
    $.ajax({
      method: 'GET',
      url: 'https://api.github.com/gists',
      data: oauthData,
      dataType: 'json'
    }).done(this.render);
    this.$el.on('click', '[data-src]', previewGist);
  },
  render: function(data) {
    this.$el.html(this.tmpl({items: data}));
    return this;
  }
});
var gistsList = new GistsListView();
```

Underscoreテンプレート

```
<script id="tmpl-js-gists" type="tmpl/text">
<% _.each(items, function(item) { %>
<li>
  <a data-id="<%= item.id %>" data-src="<%= item.url %>">
    <%= item.description %>
  </a>
  <a href="<%= item.html_url %>">Show in gists</a>
</li>
<% }); %>
</script>
```

イベントの定義

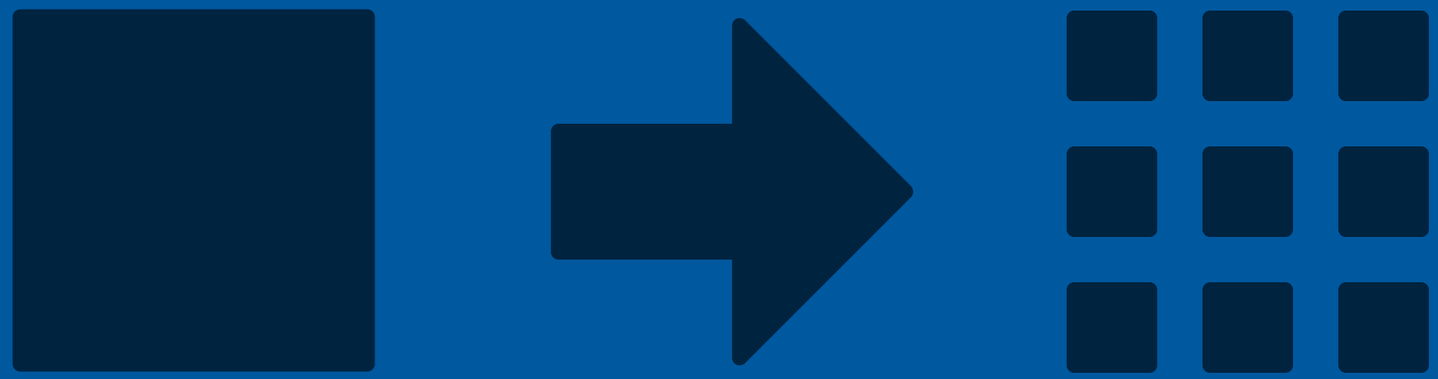


```
var GistsListView = Backbone.View.extend({
  el: '#js-gists',
  tmpl: _.template($('#tmpl-js-gists').html()),
  events: {
    'click [data-src]': previewGist
  },
  initialize: function() {
    _.bindAll(this);
    $.ajax({
      method: 'GET',
      url: 'https://api.github.com/gists',
      data: oauthData,
      dataType: 'json'
    }).done(this.render);
  },
  render: function(data) {
    this.$el.html(this.tmpl({items: data}));
    return this;
  }
});
var gistsList = new GistsListView();
```

コンポーネントで
分割すれば構造化される

自然とTestableなコードにもなる

モジュラーなJavaScript



依存性の低さによる変更のしやすさ
個々が独立していることによる交換のしやすさ
継続的なメンテナンスのしやすさ
リファクタリングとテスト

大きな現実
小さな実装

細かい実装を構造化して現実の要件へ

ModelとCollectionの利用

GitHub APIの関連処理を抽出

Gist

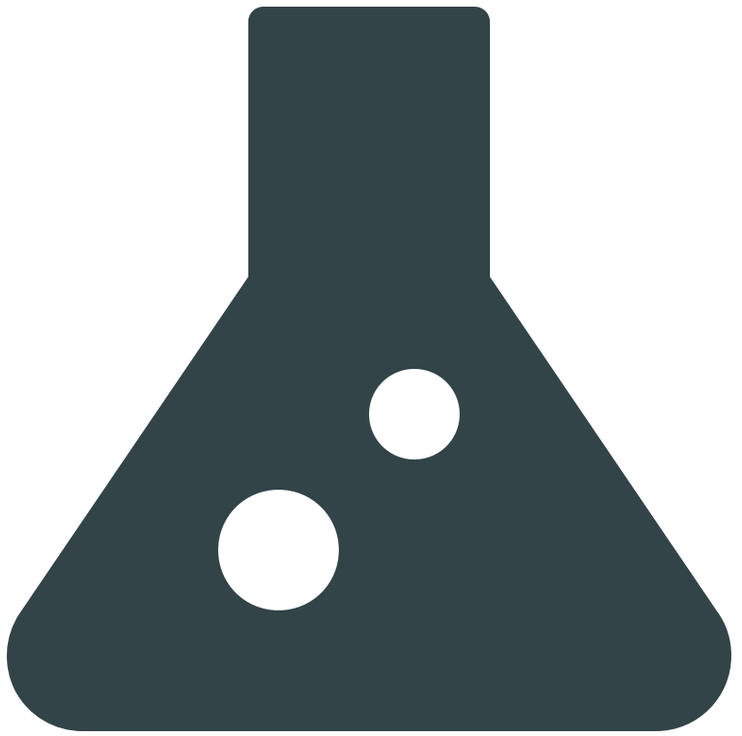
Gists

Model

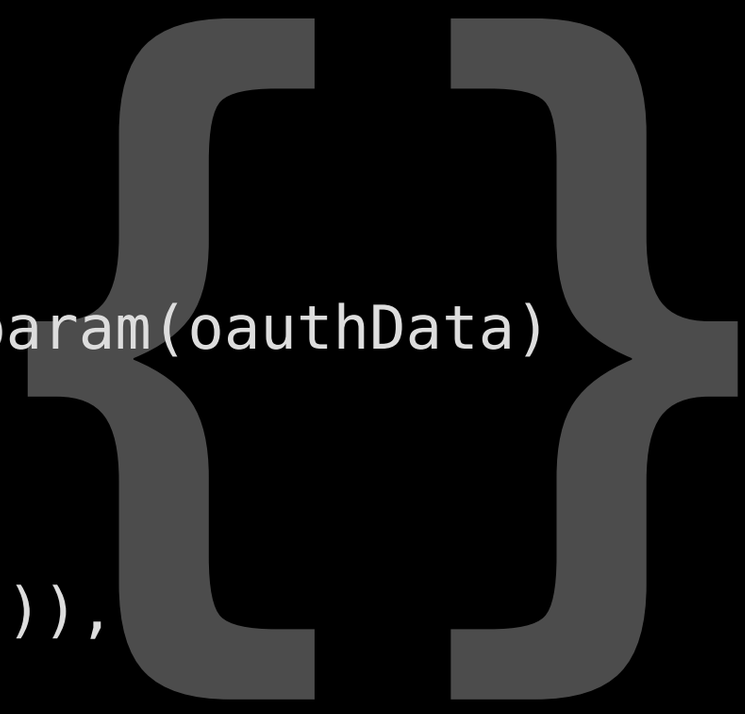
Collection

DEMO

ModelとCollection

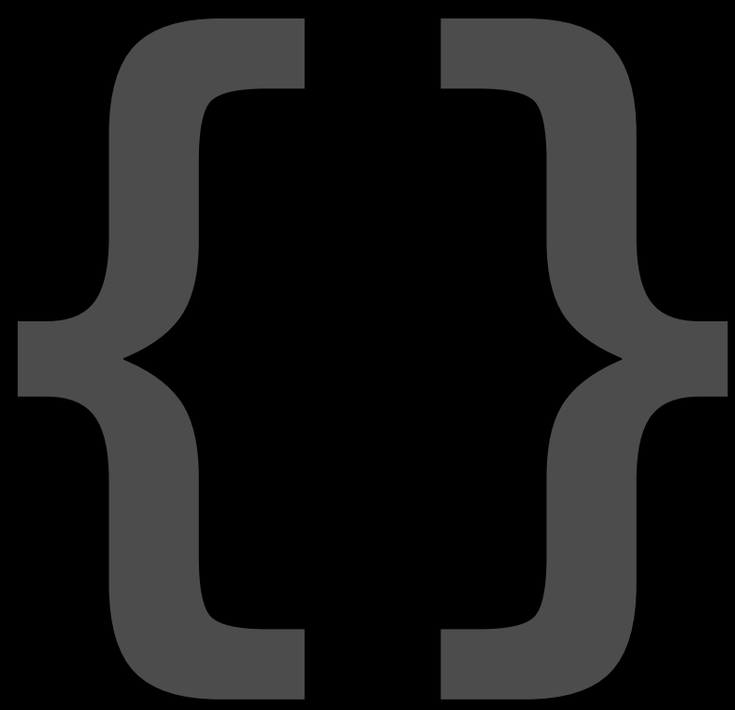


Collectionを作成



```
var Gists = Backbone.Collection.extend({
  url: 'https://api.github.com/gists?' + $.param(oauthData)
});
var GistsListView = Backbone.View.extend({
  el: '#js-gists',
  tmpl: _.template($('#tmpl-js-gists').html()),
  events: {
    'click [data-src]': 'preview'
  },
  initialize: function() {
    _.bindAll(this);
    this.collection.fetch({ success: this.render });
  },
  render: function() {
    this.$el.html(this.tmpl({items: this.collection.toJSON()}));
    return this;
  }
});
var gistsList = new GistsListView({ collection: new Gists() });
```

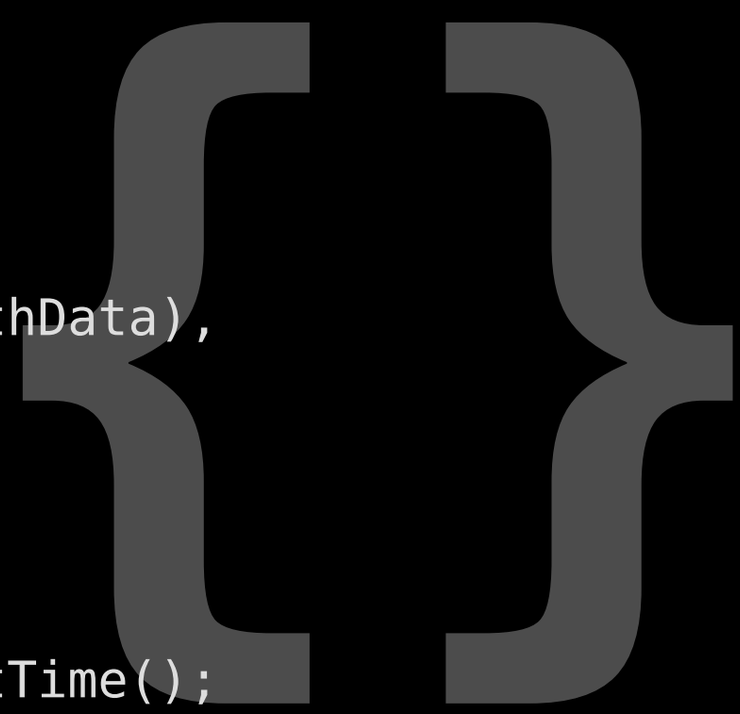
Modelを作成



```
// Model & Collection
var Gist = Backbone.Model.extend({
  url: function() {
    return this.get('url');
  }
});
var Gists = Backbone.Collection.extend({
  url: 'https://api.github.com/gists?' + $.param(oauthData),
  model: Gist
});

// GistListView#preview
preview: function(event) {
  gistPreview.model = this.collection.where({
    id: $(event.currentTarget).attr('data-id')
  })[0];
  gistPreview.show();
  return false;
},
```

Collectionのソート



```
var Gists = Backbone.Collection.extend({
  url: 'https://api.github.com/gists?' + $.param(oauthData),
  model: Gist,
  order_by: 'updated_at',
  comparator: function(gist) {
    switch(this.order_by) {
      case 'updated_at':
        return - new Date(gist.get('updated_at')).getTime();
    }
  }
});
// GistListView
events: {
  'click #js-sort-updated': 'sortByUpdatedAt',
},
initialize: function() {
  _.bindAll(this);
  this.collection.fetch({success: this.render});
  this.collection.on('sort', this.render);
},
sortByUpdatedAt: function() {
  this.collection.order_by = 'updated_at';
  this.collection.sort();
},
```

ライブラリの利用強度とパフォーマンス

ぶっちゃけパフォーマンスに影響は？

jQueryにも言えるが
使い方によっては簡単に重くなる

```
// ループ内でappendすんな、毎回セレクタ走らせんな!!
$.each(persons, function(person) {
  $(‘ul’).append(‘<li>’+person.name+’</li>’);
});
```

パフォーマンスを考えて
慎重に使うことが重要

ライブラリの努力を無駄にしないため

まとめ

より良い学びのために、その他

アーキテクチャを考えるためのBackbone

学習手段としてのライブラリ

Backbone has made me a better programmer

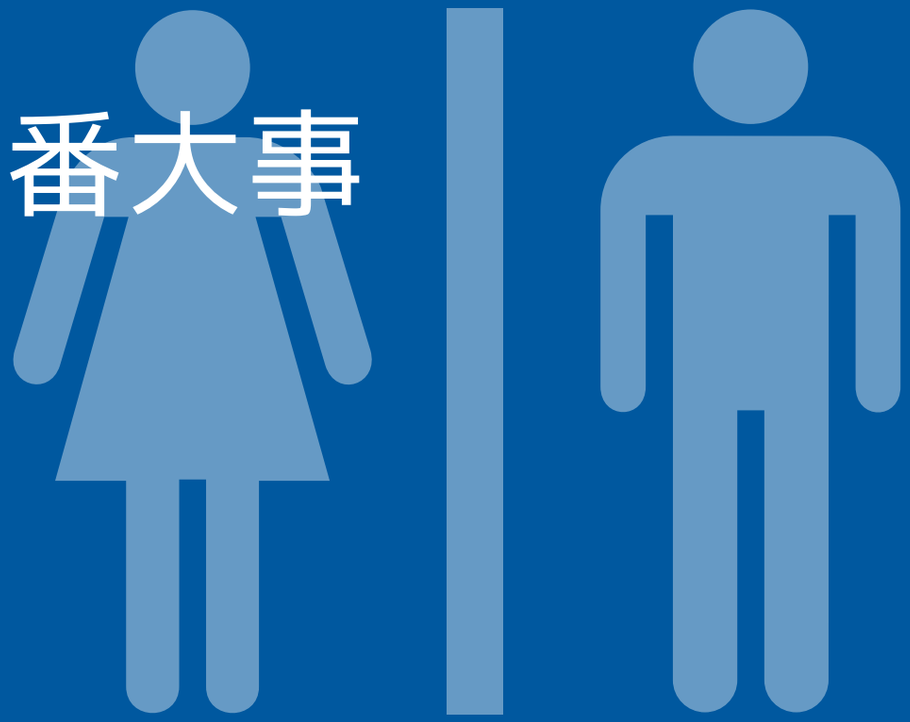
via. Backbone has made me a better programmer | Float Left
http://floatleft.com/notebook/backbone-has-made-me-a-better-programmer

フルスタックでないので学習コストは低い

もちろんデフォルトの機能は限られる

良い習慣のために
とりあえず分けてみる

「ものはためし」が一番大事



ライブラリの利用が学習につながる

フレームワークで言語に入る例もある

A JAVASCRIPT
MODULE LOADER


```
/* ---

RequireJS is a JavaScript file and module
loader. It is optimized for in-browser use, but
it can be used in other JavaScript environments,
like Rhino and Node. Using a modular script
loader like RequireJS will improve the speed and
quality of your code.

IE 6+ .......... compatible ✔
Firefox 2+ ..... compatible ✔
Safari 3.2+ .... compatible ✔
Chrome 3+ ...... compatible ✔
Opera 10+ ...... compatible ✔

Get started then check out the API.

--- */
```

Latest Release: 2.1.4
Open source: new BSD or MIT licensed
web design by Andy Chung © 2011

RequireJS

http://requirejs.org/

Dependencies? AMD?

勉強会資料シェア Getting Started with RequireJS
http://havelog.ayumusato.com/develop/javascript/e525-into_requirejs.html

to から with へ
Backbone with jQuery

あらためて次の関心へ

アーキテクチャや
デザインパターンは
手を動かしてみるのが一番
自分で良い方法を選んで
書けるようになるのが
これから大事になる(と思う)

参考リソース

手を動かすときのお供に

ackbone / 194 commi

Merge pull request #41 from maepon/master ···

studiomohawk authored 6 days ago latest commit fddd40f3fb

docs	6 days ago	Merge pull request #41 from maepon/master [studiomohawk]
.gitmodules	2 months ago	change directory name [Layzie]
README.md	16 days ago	#39 Update README.md [ahomu]
backbone.js	2 months ago	Fix by supervisor's feedback [ahomu]
original @ c36df02	2 months ago	change directory name [Layzie]

README.md

Backboneドキュメント日本語訳

部分的な翻訳済みドキュメントを集めていって、翻訳カバー率100%を目指します。

Pull Request・Issue・ご寄稿・ご指摘、いずれも歓迎しています。お気軽にどうぞ。このREADME文書自体の内容についても、ご指摘・ご提案があれば随時お知らせください。

en.ja-ossへの参加について

READMEレポジトリはenja-ossのトップページです。このレポジトリのREADMEやIssuesにて 本グループに参加いただく際の注意点などを記載しています。グループワークを円滑に行うために一読してください。

また参加いただく際にできればREADMEレポジトリをStarではなく、Watchいただければ。Google Groupなどを使った連絡についても

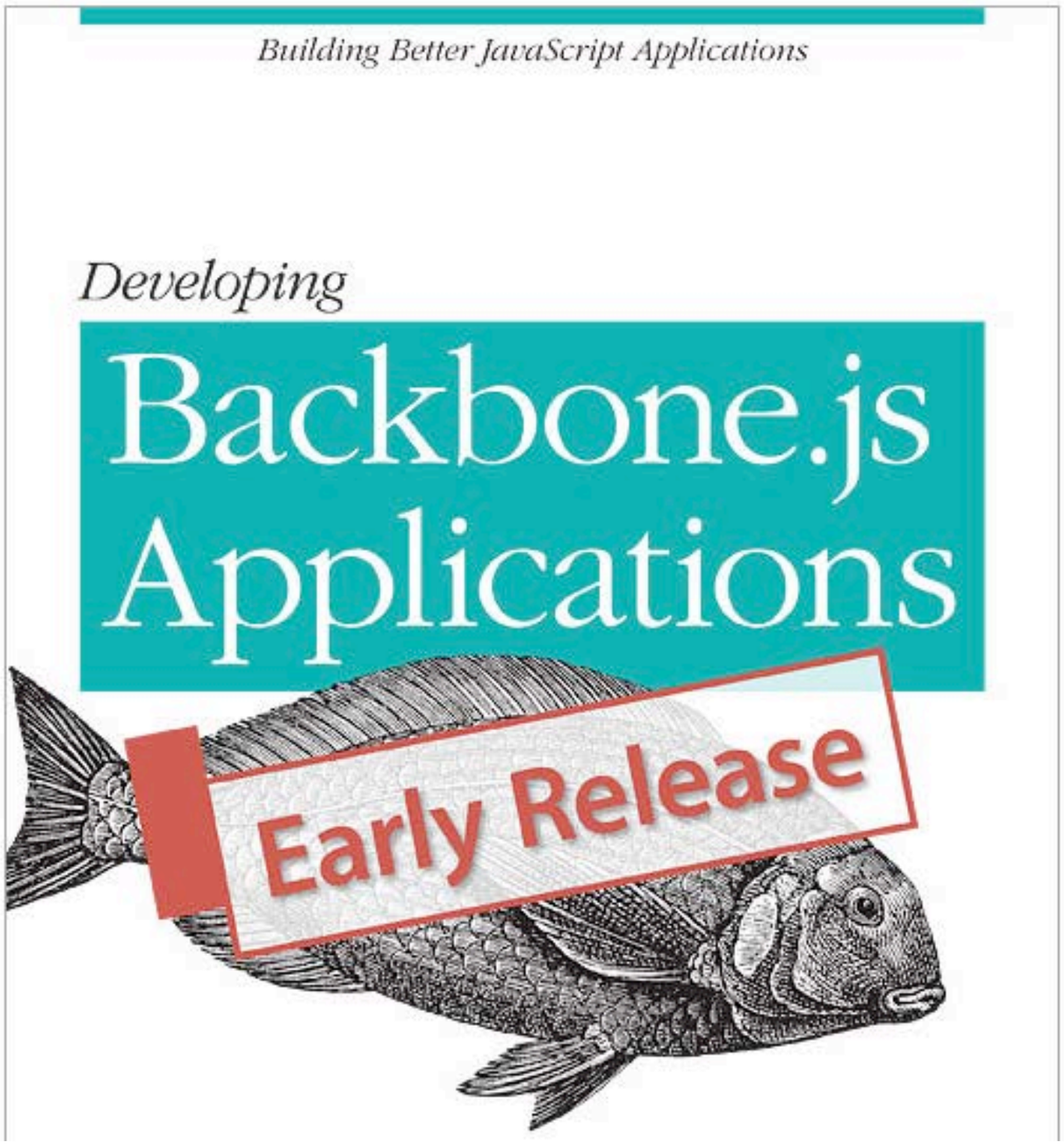
en.ja OSS Backbone 日本語訳

https://github.com/enja-oss/Backbone/

Developing Backbone.js Applications

By Addy Osmani @addyosmani

Star 5,562 Tweet 1,166



Was this helpful?

Developing Backbone.js Applications

http://addyosmani.github.com/backbone-fundamentals/

Adventar

Googleアカウントでログイン

Backbone.js Advent Calendar 2012

作成者: ahomu

登録状況: 25/25人

BackboneのTIPSとかパターンとか、他のライブラリとの連携などなど。

フルスタック感がない分、各自でいろいろ編み出しながら使いこなしていると思うので、そのあたりの共有がてら書きましょう！

※ 登録した日に自分のブログなどにエントリーを書いてください。ブログがない人はgistとかに書くといいと思います。
※ 登録するにはログインが必要です。

2012-12-01（土） ahomu

Backbone.jsで今つくっている構成について

http://havelog.ayumusato.com/develop/javascript/e531-backbone_webapp_case_intro.html

2012-12-02（日） mitsuruog

backbone.localstorage.jsとBackbone.Syncのお話

Advent Calendar

http://www.adventar.org/calendars/15

havelog

Backbone.jsを使うときに参考になるリソース 2012
年末版 (Backbone Advent Calendar 2012 25th day)

2012-12-26

Backboneを使うときの参考情報まとめ

Advent Calendarがネタ切れの折、最終日が冴えない小ネタで終わるよりはマシかという
ことでリストアップしてみた。

日本語リソース

では早速、日本語のリソースから。古い情報はリストから外しているので、いくらか偏り

Follow ahomu

その他の参考情報

http://havelog.ayumusato.com/develop/javascript/e544-backbone_learning_resources.html

最後に派生ライブラリ
There's More Than One Way To Do It

用途に合わせて拡張された具体例

Download
1.0.0-rc4

Marionette.js on GitHub

Marionette

http://marionettejs.com/

Marionette.js

Make your Backbone applications dance!

Chaplin

Web application framework on top of Backbone.js.

Star 1,670

Follow @chaplinjs 315

Web application framework

Chaplin is an architecture for JavaScript applications using the Backbone.js library. Chaplin addresses Backbone's limitations by providing a lightweight and flexible structure that features well-proven design patterns and best practices.

Some of Chaplin's features:

- CoffeeScript **class hierarchies** as well as object composition
- **Module** encapsulation and lazy-loading using AMD modules
- Cross-module communication using the **Mediator** and Publish/Subscribe patterns
- **Controllers** for managing individual UI views
- Rails-style **routes** which map URLs to controller actions
- A route dispatcher and a **top-level view** manager
- Extended **model, view and collection** classes to avoid repetition and enforce conventions
- Strict **memory management** and object disposal
- A **collection view** for easy and intelligent list rendering

Downloads

Chaplin is available in two builds: AMD and Common.js. Pick AMD if you want to build your app manually with require.js or common.js if

Chaplin

http://chaplinjs.org/

Thorax

An opinionated, battle tested Backbone + Handlebars framework to build large scale web applications.

Standalone

Open the `index.html` file from the downloaded project in your browser.

Download 2.0.0b5

Node + Lumbar

Run `npm start` from the downloaded project.

Download 2.0.0b5

Mobile

Run `npm start` from the downloaded project.

Download 2.0.0b5

Rails

Run `rails server` from the downloaded project.

Download 2.0.0b5

Thorax 2 is presently in beta, the stable source and documentation are still available.

Thorax can be used standalone in any JavaScript environment in addition the boilerplate projects provided above.

```
var view = new Thorax.View({
  template: "Hello world!"
});
view.render();
$("body").append(view.el);
```

Editable Examples

All of the examples use the same sample data.

- Simple Todos
- $.model
- Context
- view & template helpers
- empty helper
- freeze
- LayoutView

Resources

- Thorax TodoMVC
  - Standalone Source
  - Thorax + Lumbar Source
  - Thorax + Require.js Source

Thorax

http://thoraxjs.org/

HAUTELOOK Engineering Blog

Vertebrae front-end framework built with Backbone.js and RequireJS using AMD

May 24, 2012 by Bill Heaton in User Interface with 8 Comments

Vertebrae provides *AMD* structure and additional objects for extending *Backbone.js* as an application framework.

- Github source : https://github.com/hautelook/vertebrae

The plan: build an application, mostly with open source libraries

Following a review of many MV* front-end frameworks/libraries, such as Ember, JavaScriptMVC, Spine, Knockout, and Backbone.js our team decided to begin building a framework with Backbone.js, RequireJS, Underscore.js, jQuery, Mustache.js with code organized in packages of (AMD) modules, see previous post: Optimize and Build a Backbone.js JavaScript application with Require.JS using Packages...

Vertebrae

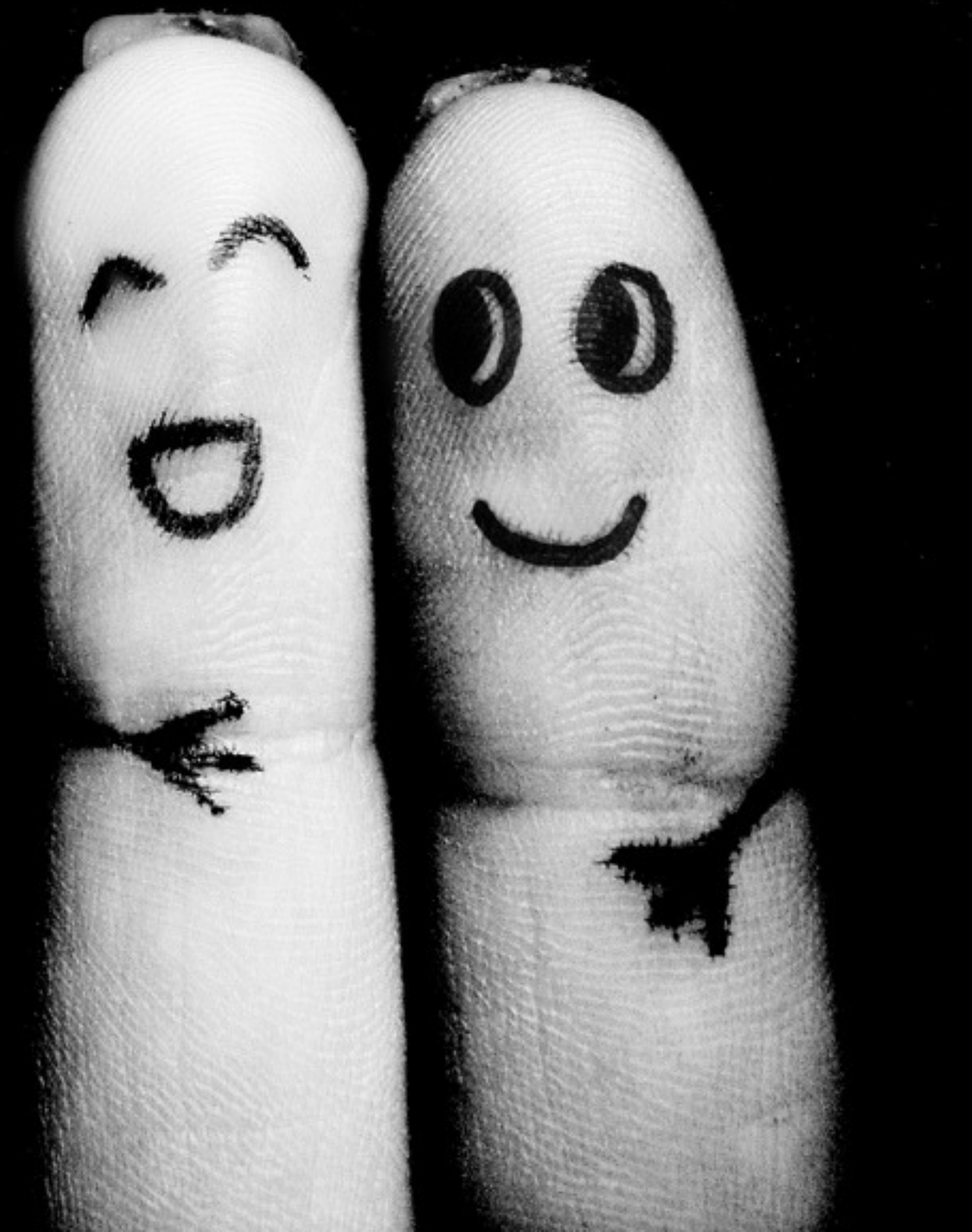
https://github.com/pixelhandler/vertebrae

ぼくのかんがえた
さいきょうの…

凝ったロジックと新たな複雑性の罠

悪いこと？

だれもが考える



Juliana Coutinho

学習目的であれば
とても良い手段

他と比較しながら、作って理解する

Become Frontend ROCKSTAR

Deciding good approach by yourself

JavaScript Development Tools

jQuery Performance Tips

Testable Javascript

JavaScript Architecture

キャッチアップはつづく！

Architecture / Modular
Dependency / AMD
Promises / FlowControl
Build / Package Management
Testing / Refactoring

etc...

FRONTREND

Thank you! おしまい

http://aho.mu

@ahomu

github.com/ahomu

Photo Credits...thx♡

1. Two equestrian riders, girls on horseback, in low tide reflections on serene Morro Strand State Beach http://www.flickr.com/photos/mikebaird/2985066755
2. Energy Drinks - Monster, Red Bull and Rockstar http://www.flickr.com/photos/aukirk/8170825503
3. - Good Friends http://www.flickr.com/photos/ngmmemuda/4166182931
4. Rhino relaxation http://www.flickr.com/photos/macinate/2810203599
5. Whale backbone http://www.flickr.com/photos/vagawi/2257918524/
6. Sleeping 猫 http://www.flickr.com/photos/hansel5569/7687221498/
7. Alien vs Predator http://www.flickr.com/photos/steampirate/1056958115/