



BACKBONE.JS

Radoslav Stankov
init Lab / SofiaJs

23/06/2012

Who am I?

@rstankov

<http://rstankov.com>

<http://github.com/rstankov>



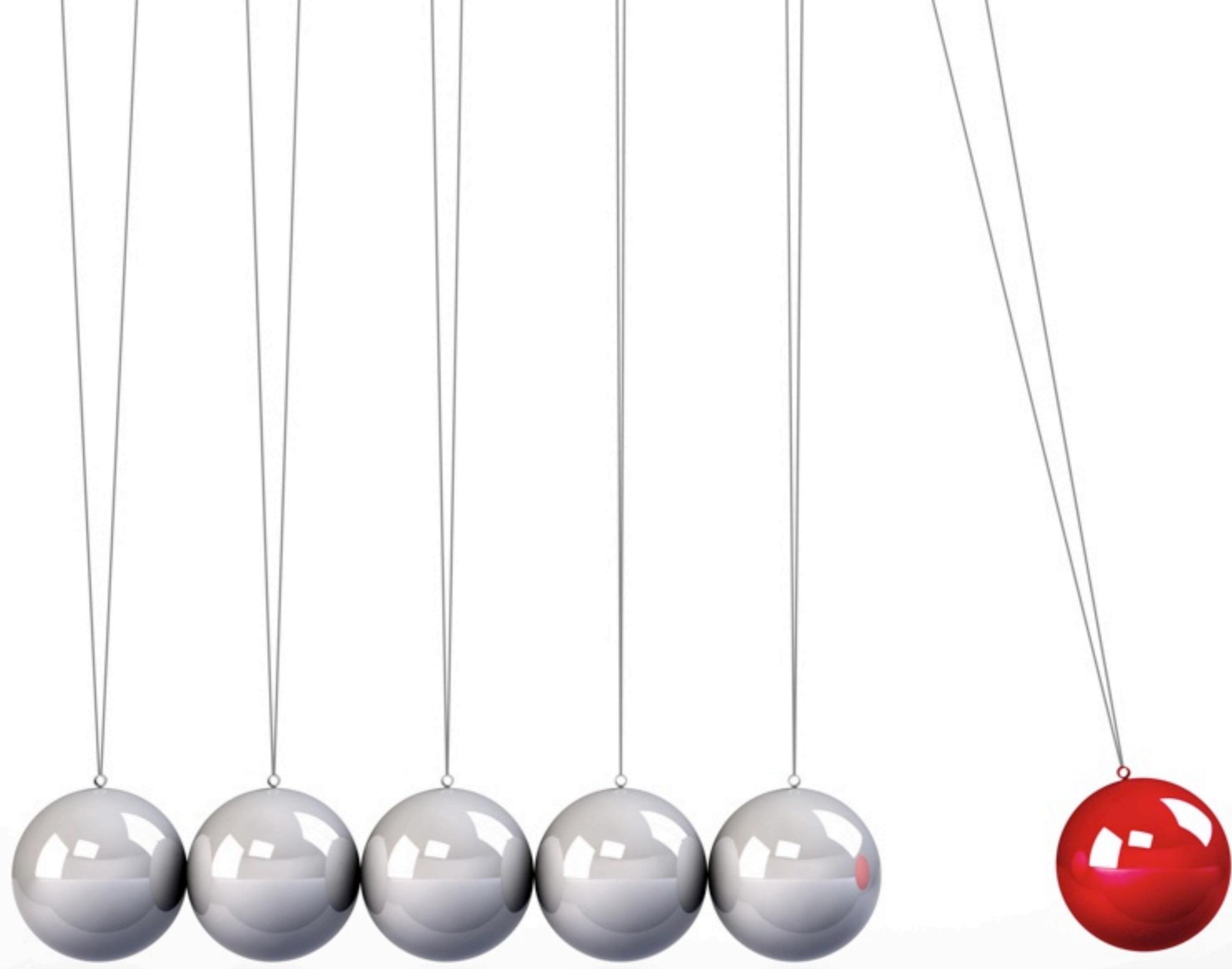
LOOKK







BACKBONE.JS



DocumentCloud

Entities

Search Help

All Documents Your Documents Your Published Documents DocumentCloud's Documents

B.P. Deepwater Horizon Oil Spill Public documents related to the oil spill, submitted by a variety of newsrooms. 33 documents, 1 note, 4 collaborators

Commercial Appeal: Ernest Withers 36 documents, 136 notes

OLC Letters & Memos 19 documents, 0 notes

The Financial Crisis 25 documents, 11 notes

+ New Documents + New Project

Jeremy Ashkenas Manage Accounts / Log Out

Commercial Appeal: Ernest Withers

project: "Commercial Appeal: Ernest Withers"

Open Edit Projects Analyze Publish

Memphis Publishing Company and Marc Penusquia v. FBI 2 notes

Pages From 2009 FOIA Reveal Withers as Informant ME 308 2 notes

Other Informants 4 notes

FBI seeks to tie invaders to Black Panthers 3 notes

FBI targets invaders for prosecution 5 notes

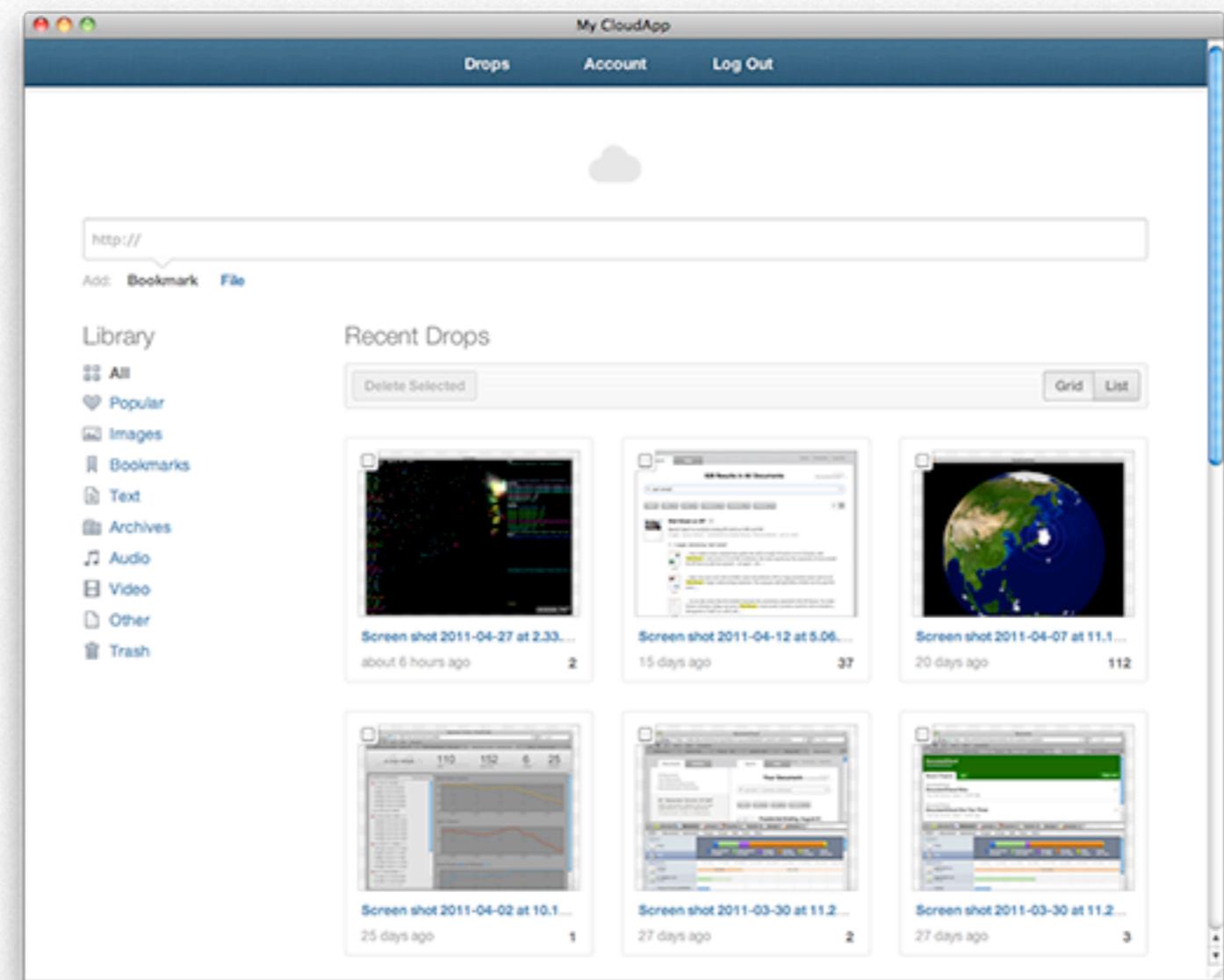
Informant draws links between invaders, others 4 notes

The invaders and COINTELPRO 4 notes

Informant Provides Car Tag Number, Other Details 2 notes

Informant Reports on Political Candidates 4 notes

This screenshot shows the DocumentCloud application interface. The top navigation bar includes tabs for 'Documents' and 'Entities', and buttons for 'Search' and 'Help'. The main content area features a search bar with the query 'project: "Commercial Appeal: Ernest Withers"'. Below the search bar are five buttons: 'Open', 'Edit', 'Projects', 'Analyze', and 'Publish'. The main content area displays a grid of document thumbnails, each with a title and a note count. The titles include 'Memphis Publishing Company and Marc Penusquia v. FBI' (2 notes), 'Pages From 2009 FOIA Reveal Withers as Informant ME 308' (2 notes), 'Other Informants' (4 notes), 'FBI seeks to tie invaders to Black Panthers' (3 notes), 'FBI targets invaders for prosecution' (5 notes), 'Informant draws links between invaders, others' (4 notes), 'The invaders and COINTELPRO' (4 notes), 'Informant Provides Car Tag Number, Other Details' (2 notes), and 'Informant Reports on Political Candidates' (4 notes). On the left side of the interface, there is a sidebar with a list of projects: 'B.P. Deepwater Horizon Oil Spill', 'Commercial Appeal: Ernest Withers' (selected), 'OLC Letters & Memos', and 'The Financial Crisis'. At the bottom left, there are buttons for '+ New Documents' and '+ New Project'. At the bottom right, there is a user profile for 'Jeremy Ashkenas' with options to 'Manage Accounts' or 'Log Out'.



Trajectory — Trajectory

Trajectory Stories Discussions Settings

Velocity: 4 Completed: 404 stories over 48 iterations

0/4 0% complete 1 Unstarted 0 Completed 2 Started All Design Development

As a user I want to leave comments on the iteration to have a discussion about the week so that I can coordinate with the team. Chad Pytel, Design and Development. 2 Deliver Stop

Site Certification for Credit Card Processing Todo Chad Pytel. 4 Deliver Stop

July 25

Disable add comment button when a comment is being added to prevent double clicking. Unassigned, Development. 1

Bug backbone: Hitting "Deliver" doesn't focus the text input anymore. Chad Pytel, Development. 1

Bug Discussion layout breaks with many stories assigned to discussion. See attachment. Unassigned, Development. 2

Todo Put "Accept" button on the left, "Reject" on the right. Unassigned. 3

Bug Closing points estimation popup is not working. Popup should close when clicking outside the popup. Unassigned, Development. 1

Bug As a user, when I delete a story, the iterations should be recalculated. Unassigned. 1

Bug When I click on the completed stories link on the top right (expanding prior iteration's stories), and then click on the completed button on this iteration's toolbar, the window scrolls to the top. It should stay there so that I can easily see stories completed on this iteration. Unassigned. 1

Bug Backbone links aren't styled fancifully. Unassigned, Development. 1

August 01

The screenshot shows the Trajectory application interface. At the top, there's a navigation bar with tabs for 'Trajectory', 'Stories', 'Discussions', and 'Settings'. Below the navigation is a summary section with 'Velocity: 4', a progress bar showing '0/4' tasks completed (0% complete), and counts for 'Unstarted' (1), 'Completed' (0), and 'Started' (2) tasks. There are buttons for 'All', 'Design', and 'Development'. A message from 'Chad Pytel' is displayed: 'As a user I want to leave comments on the iteration to have a discussion about the week so that I can coordinate with the team.' Below this is a 'Todos' section for 'Site Certification for Credit Card Processing' by 'Chad Pytel'. The main area is a list of stories and bugs for the 'July 25' iteration. Stories include: 'Disable add comment button when a comment is being added to prevent double clicking.', 'backbone: Hitting "Deliver" doesn't focus the text input anymore.', 'Discussion layout breaks with many stories assigned to discussion. See attachment.', 'Put "Accept" button on the left, "Reject" on the right.', 'Closing points estimation popup is not working. Popup should close when clicking outside the popup.', 'As a user, when I delete a story, the iterations should be recalculated.', 'When I click on the completed stories link on the top right (expanding prior iteration's stories), and then click on the completed button on this iteration's toolbar, the window scrolls to the top. It should stay there so that I can easily see stories completed on this iteration.', and 'Backbone links aren't styled fancifully.'. Each item has a 'Deliver' and 'Stop' button next to it. The bottom of the screen shows the date 'August 01'.

Groupon Now!

Featured Deal All Deals Now! Deals Getaways How It Works Sign In Sign Up

Select category Eat Something Select time Today near Times Square Find Deals

A \$5 for \$10 at Knave in Le Parker Meridien Lobby
0.7 miles away ★★★★ (35)
Redeem today before 4pm

B \$35 for \$70 at 5th Avenue Chocolatiere
0.7 miles away
Redeem today before 7pm

C \$4 for \$7 at Comfort Diner
0.8 miles away ★★★★ (57)
Redeem today before 10:30am

D \$18 for \$24 at Opla
0.9 miles away ★★★★ (52)
Redeem today before 11am

E \$5 for \$9 at 63bites
1 mile away ★★★★ (49)
Redeem today before 1:45pm

F \$6 for \$10 at The Tea Set
1.5 miles away ★★★★ (17)
Redeem today before 11pm

Overview Ratings & Fine Print

\$5 for \$9 at 63bites
44% off Homemade Soups, Salads and More
Tweet 0

Buy Now! Redeem Today Between 6am - 1:45pm or we'll automatically refund you

Map data ©2011 Google - Terms of Use

The screenshot shows the Groupon Now! interface. At the top, there's a navigation bar with links for 'Featured Deal', 'All Deals', 'Now! Deals', 'Getaways', 'How It Works', 'Sign In', and 'Sign Up'. Below this is a search bar with dropdown menus for 'Select category' (set to 'Eat Something'), 'Select time' (set to 'Today'), and 'near' (set to 'Times Square'). A large 'Find Deals' button is to the right of the search bar. On the left, a list of deals is displayed, each with a small icon, a title, a distance from the user, a rating, and a deadline for redemption. To the right of the deals is a map of New York City, specifically the area around Times Square. The map features several green location markers, each containing a letter from A to S. A callout box on the map provides more details about a deal at '63bites', including a photo of the food, the price, the discount percentage, and the option to tweet about it. A large green 'Buy Now!' button is overlaid on the map, along with text indicating the redemption period and a refund guarantee. The bottom of the map includes a copyright notice for Google map data.

Screen Shot 2011-11-17 at 2:15:47 PM.png 1,008x1,099 pixels

foursquare Search people and places...

Activity Lists Mike ▾

Suggestions for Thursday at 2:14pm:

- Saigon Shack
Baratunde has been here.
- Dumpling Man
Jon Stoltzblock, jaka furt and more than 1...
- Pho 32 & Shabu
Jason Lauku, Matt Healy and more than ...

Map data ©2011 Google, Sanborn - Terms of Use

Hi Mike Singleton
Last seen 20 minutes ago at: [Chipotle Mexican Grill \[off the grid\]](#)

Holger Luedorf @ foursquare HQ
10 minutes ago - Comment - East Village, at 5th St
« Ist schön hier!
 Leave a comment...

Baratunde @ Los Angeles International Airport (LAX)
12 minutes ago - Comment - Los Angeles, CA
« Boom. Come to Harvard & Stone after 930p if you wanna kick it w some @thefamousgrouse whisky. #yesknowitsthursday
 Leave a comment...

Mike Singleton @ [off the grid]
20 minutes ago - Comment

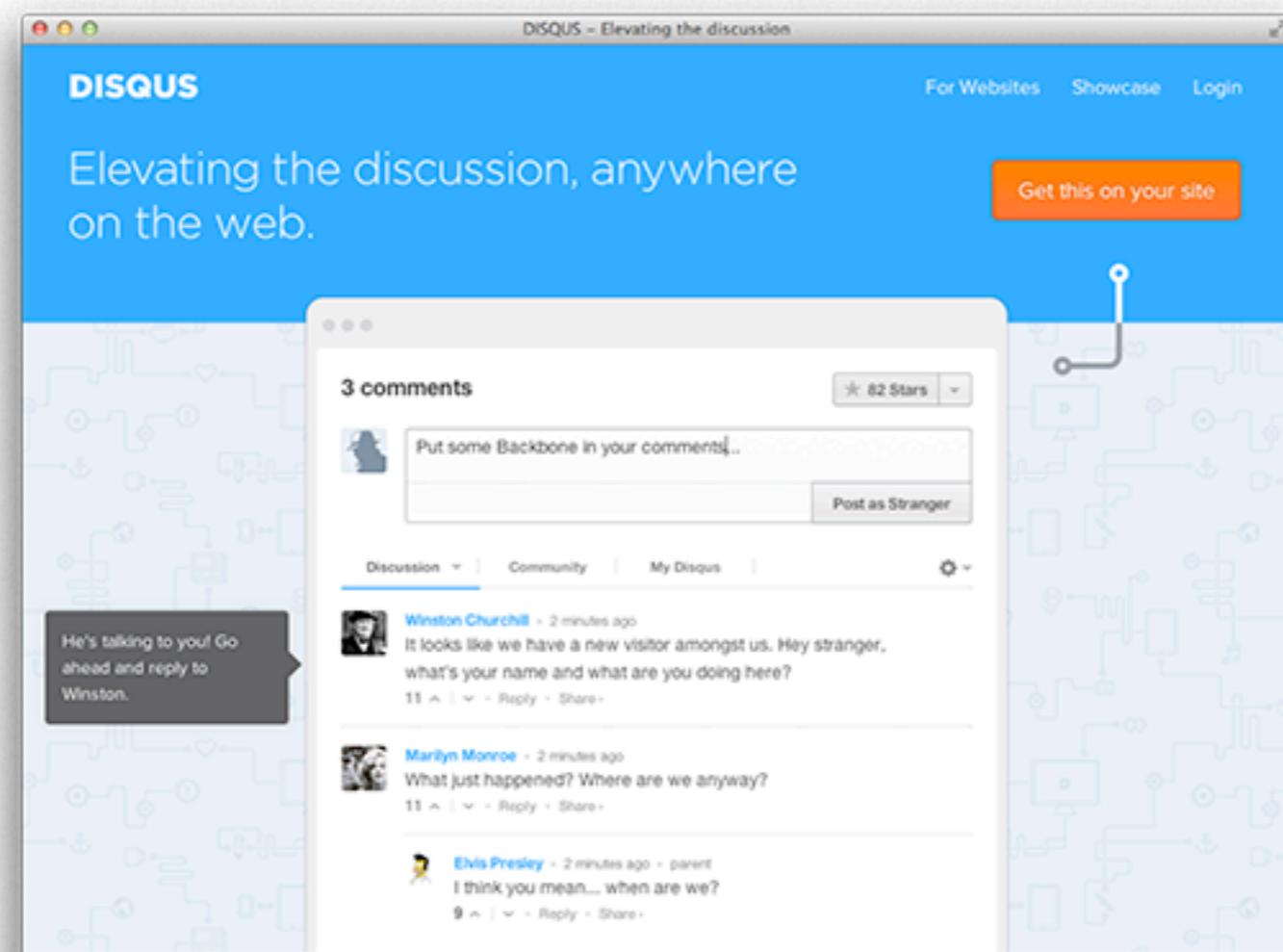
Lists from your friends [See All](#)

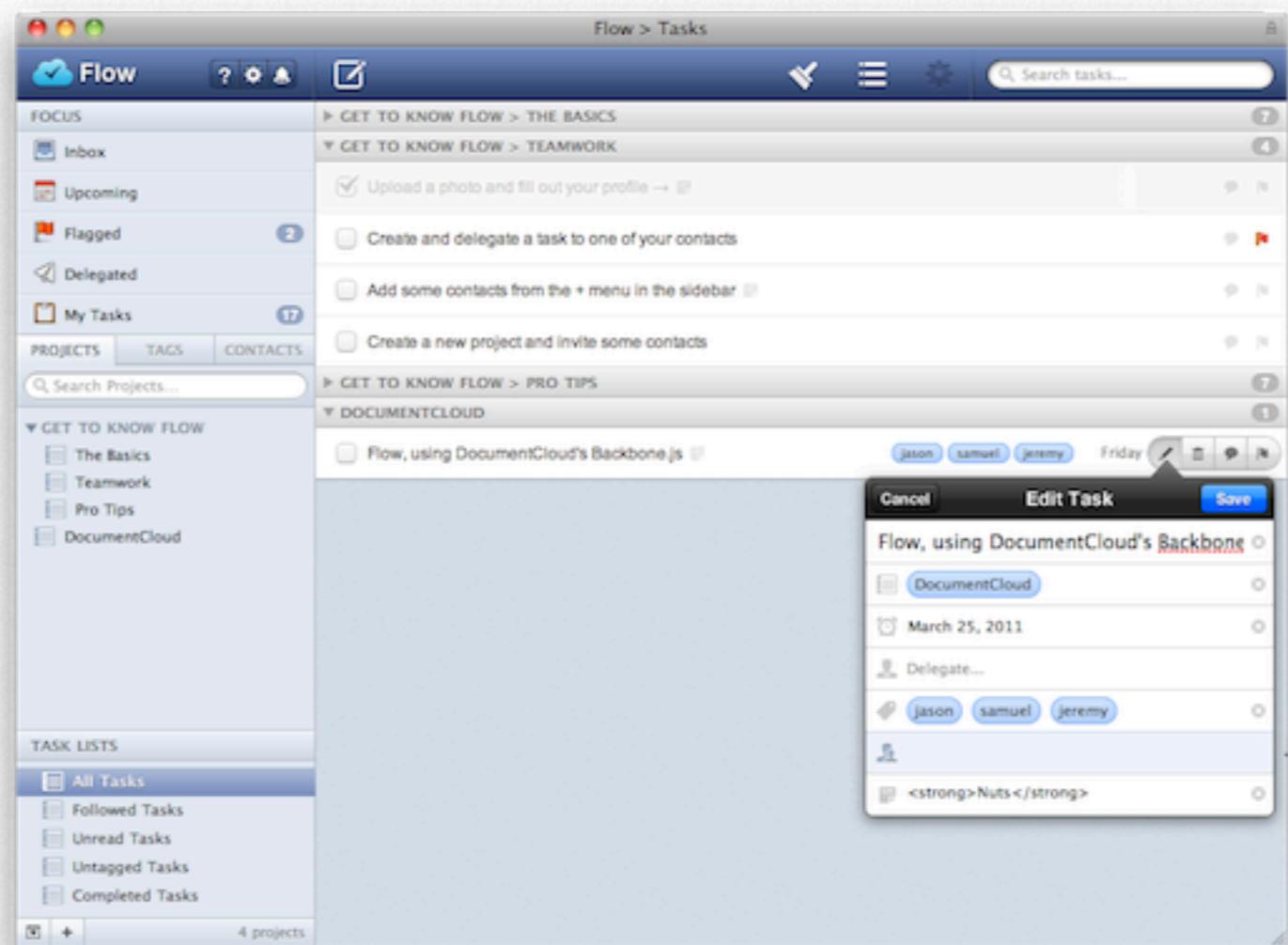
- [Jewlicious Noshes in NYC](#)
Created by jake f.
15 places
10 Followers
- [get your game on](#)
Created by Christen D.
2 places
- [The Layover: Singapore](#)
Created by Travel Channel
22 places
13 Followers

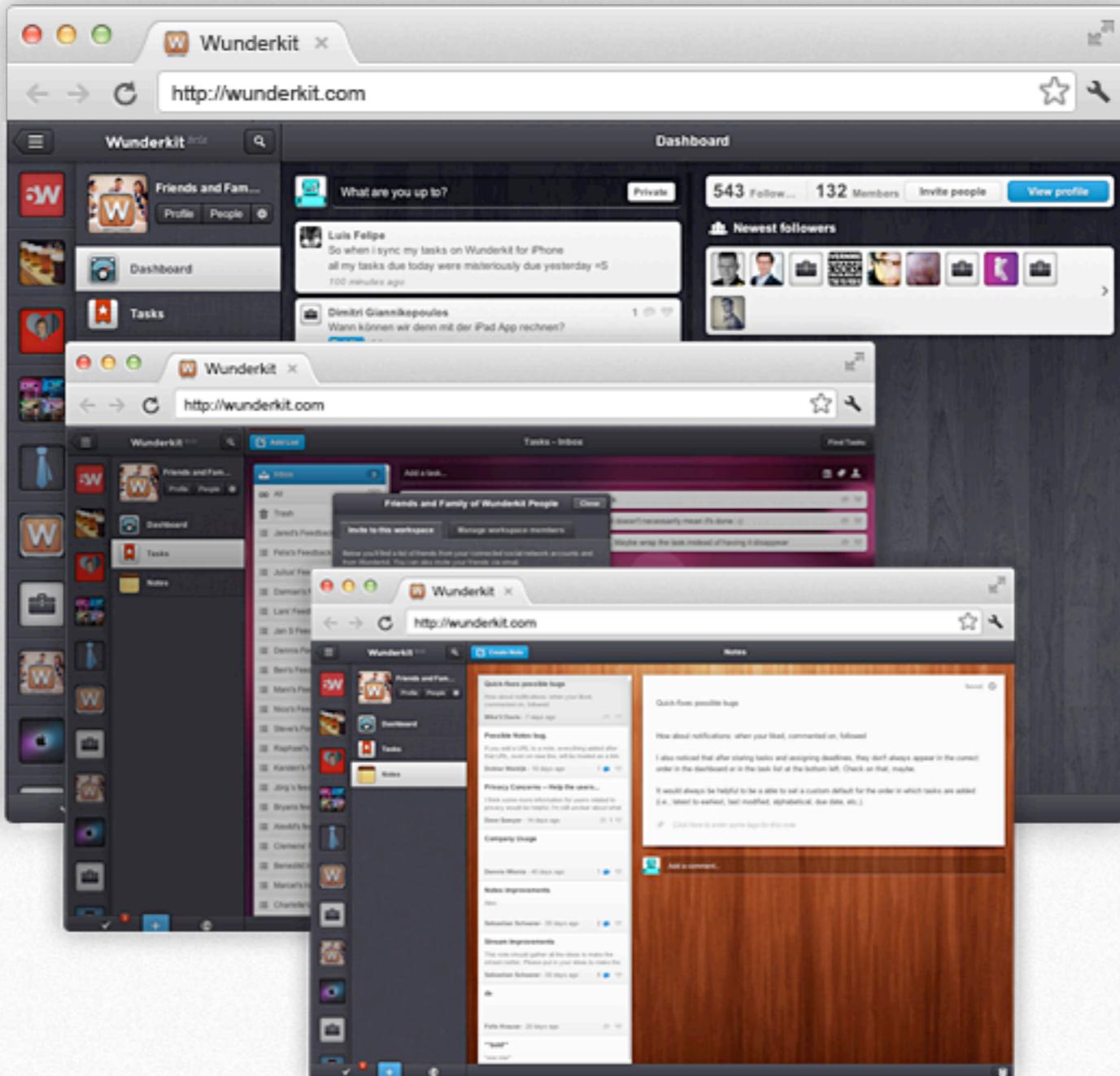
Create and share your own list: [Get started...](#)

Brands on foursquare [See More](#)

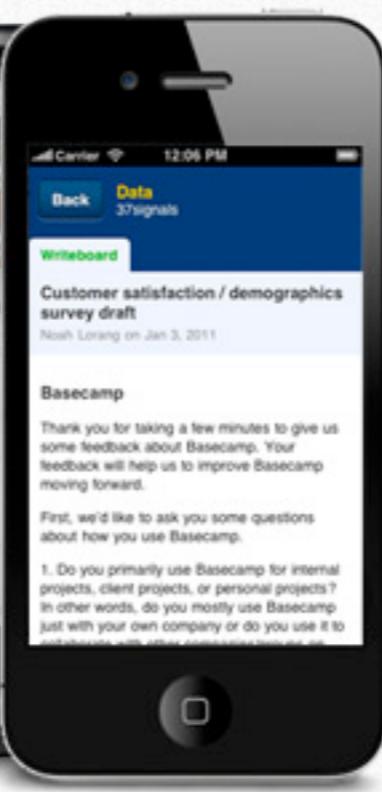
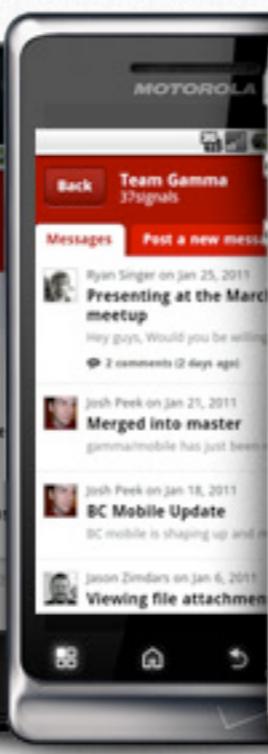
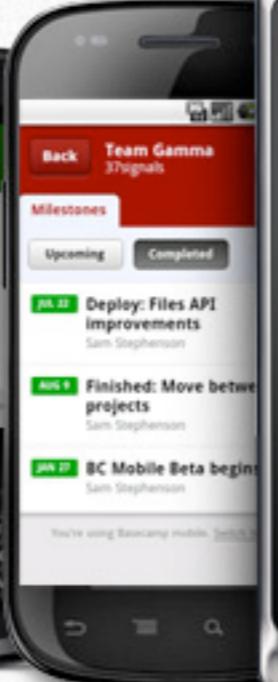
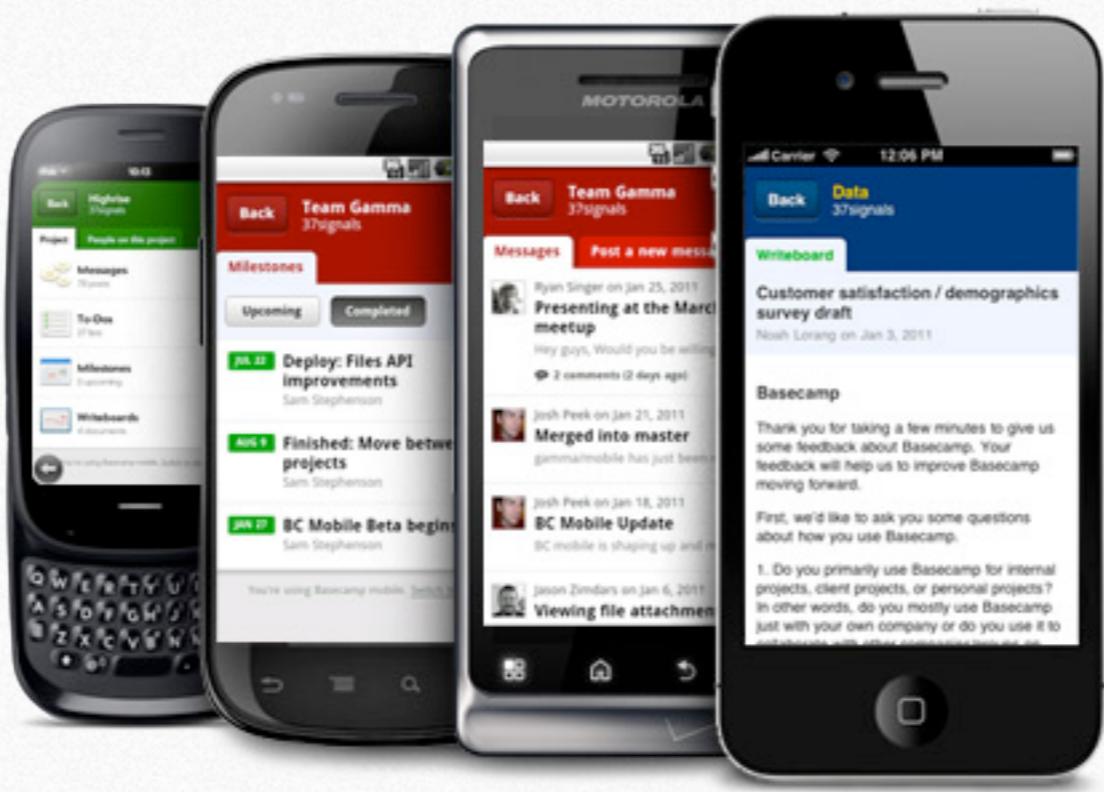
- [BlackBerry](#) 142,439 followers

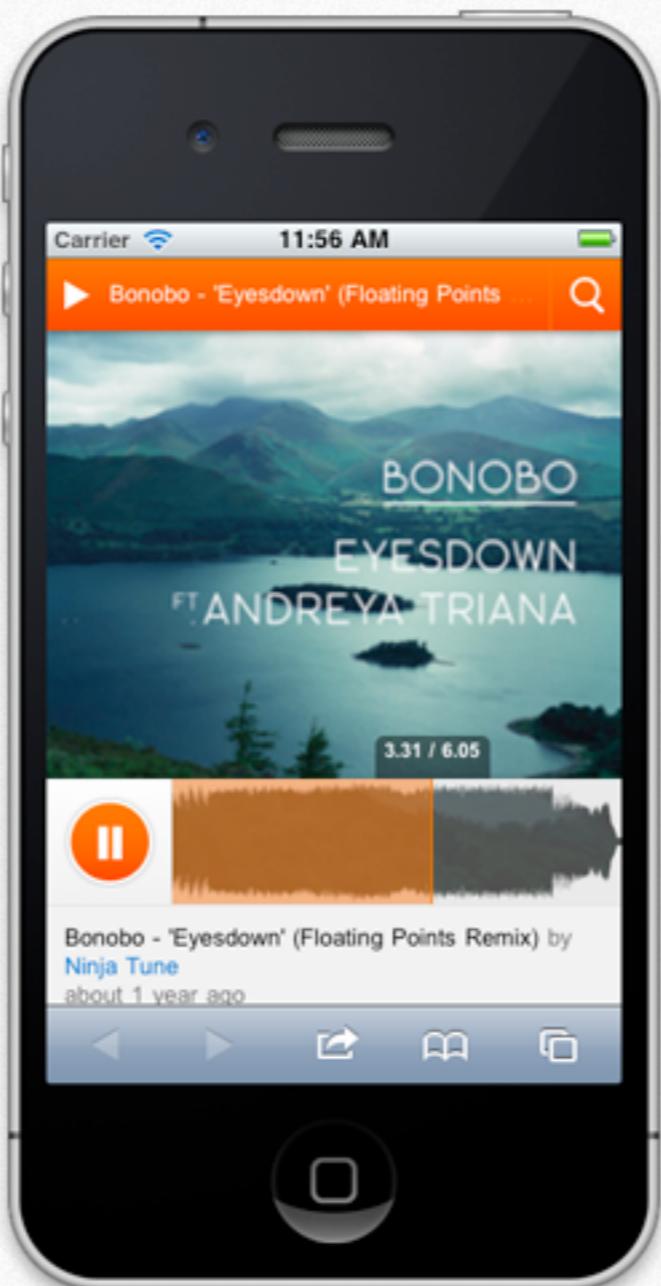


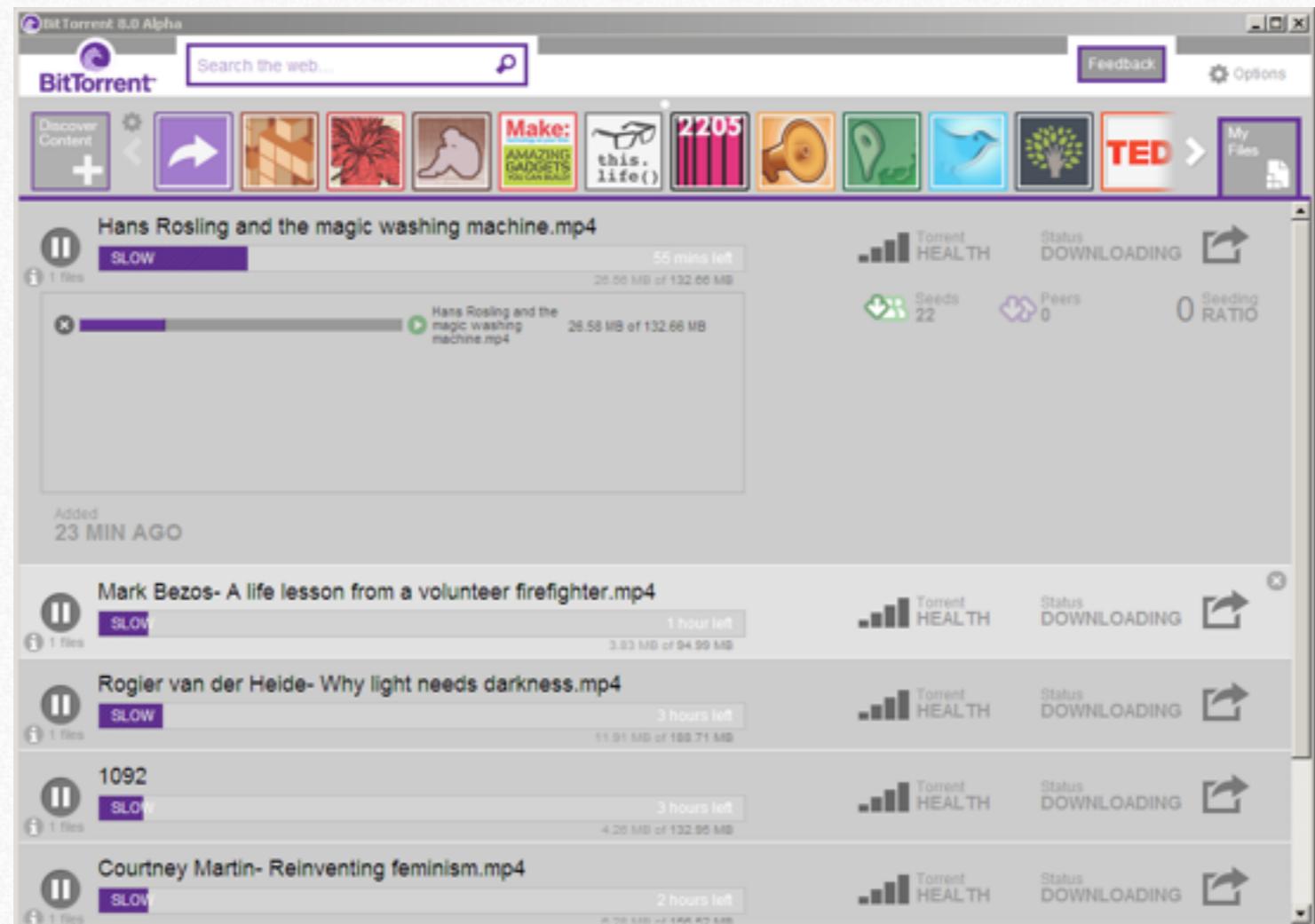














BACKBONE.JS



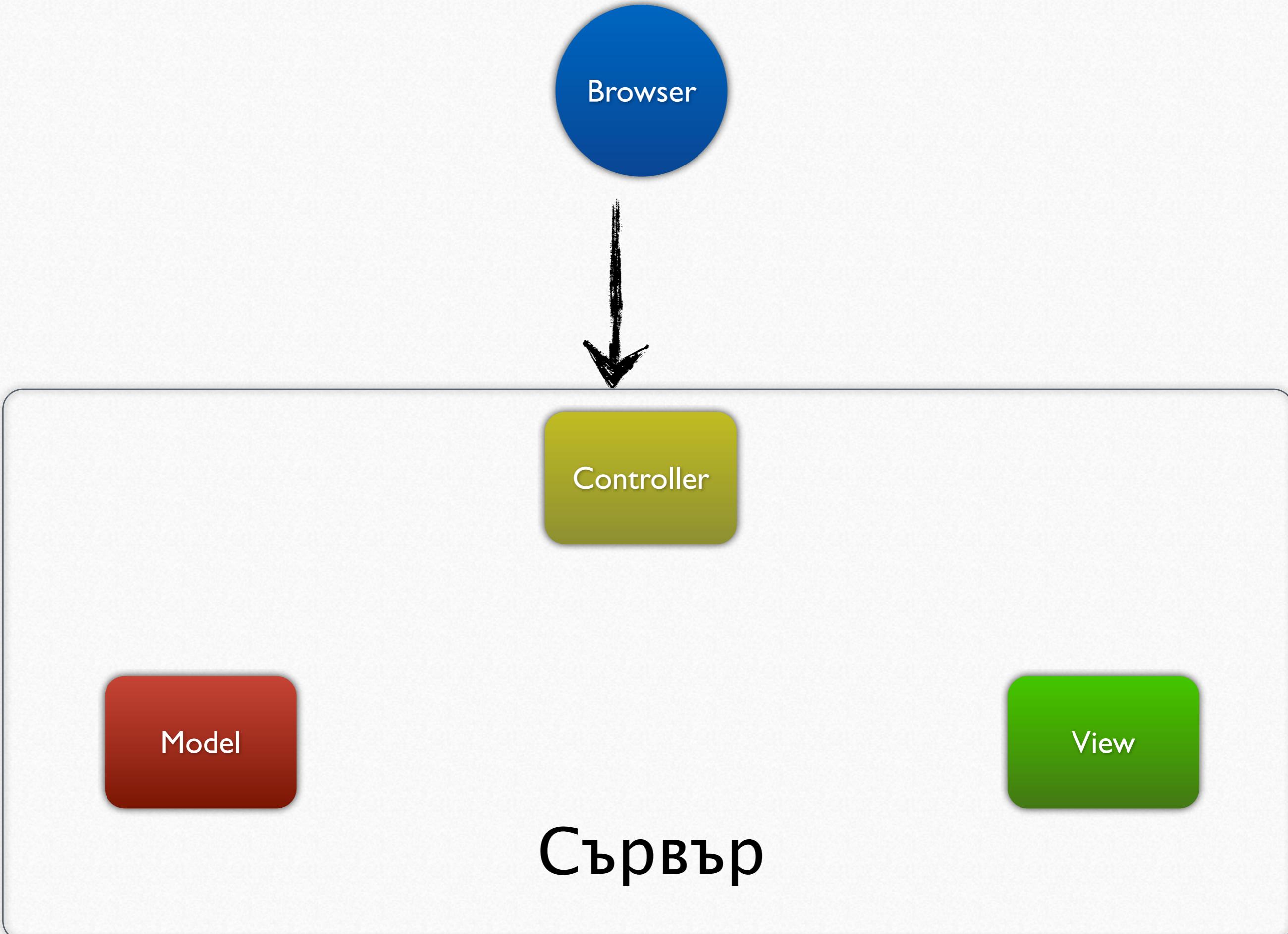
Browser

Controller

Model

View

Сървър



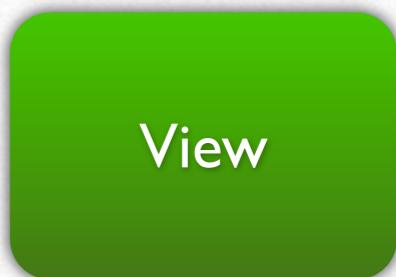
Browser

Controller

Model

View

Сървър



Сървър

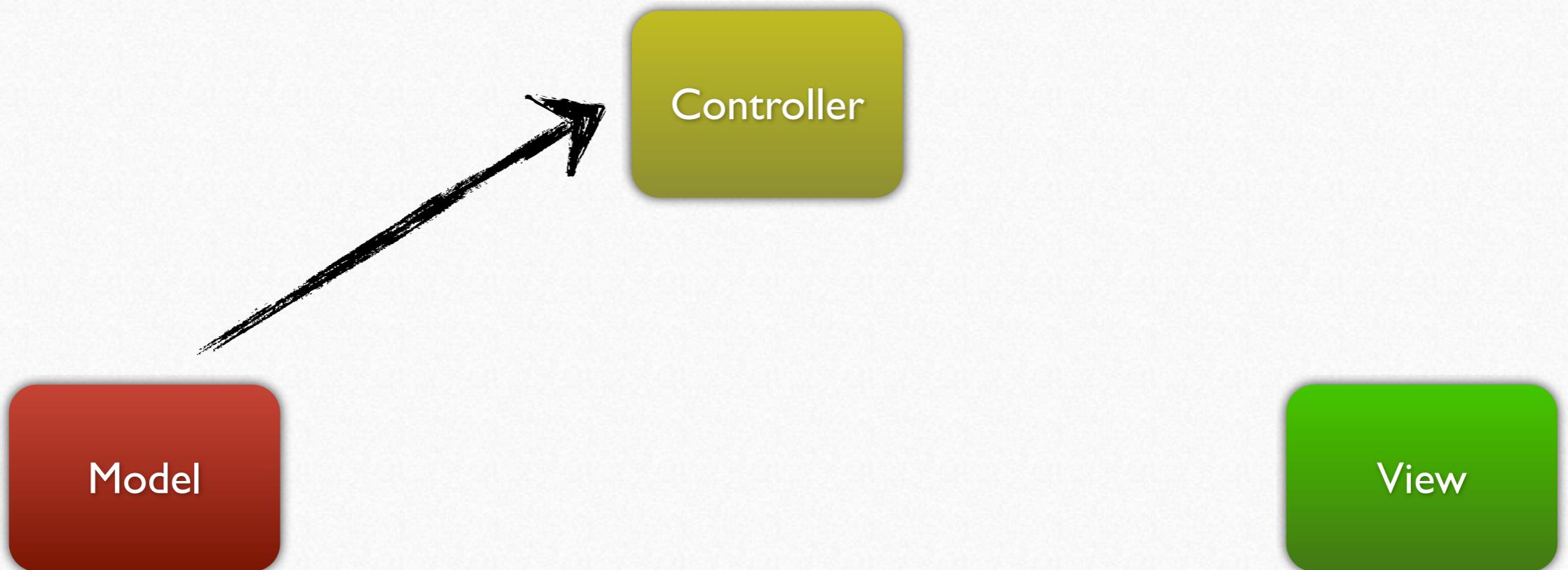
Browser

Controller

Model

View

Сървър



Сървър

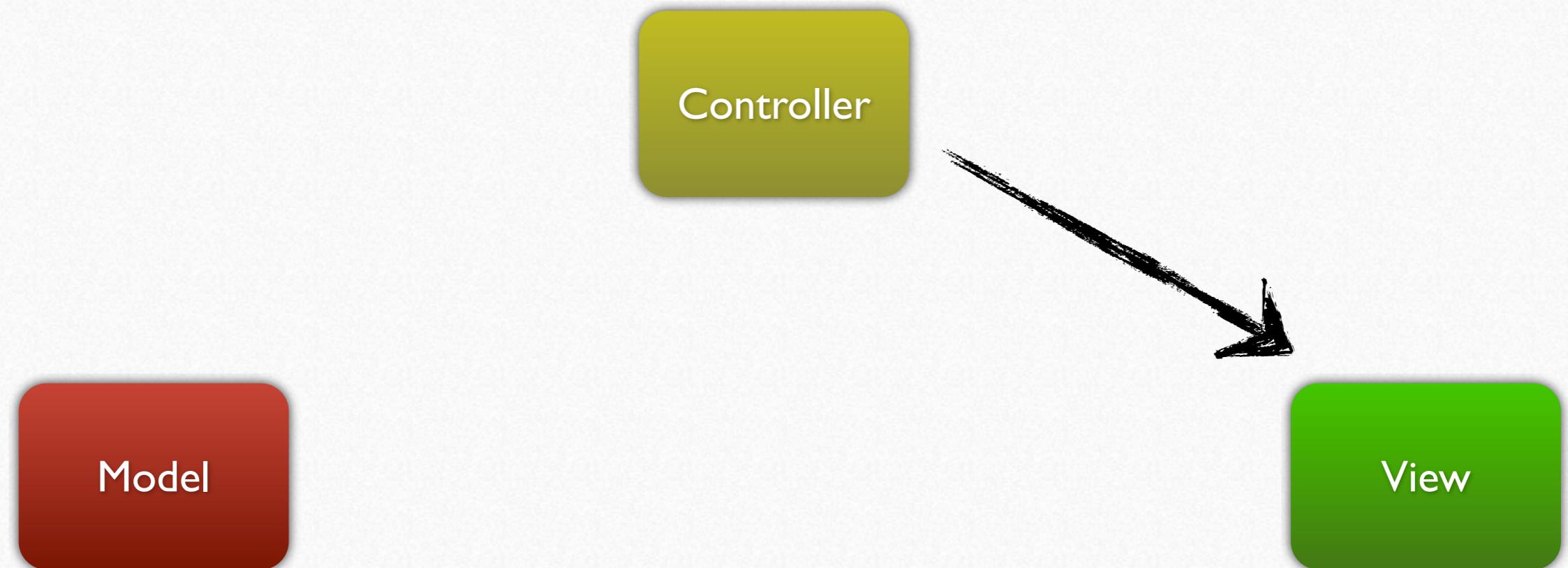
Browser

Controller

Model

View

Сървър



Сървър

Browser

Controller

Model

View

Сървър



Сървър

Browser

Controller

Model

View

Сървър

Browser



Controller

Model

View

Сървър

Browser

Controller

Model

View

Сървър

Browser

Dom

View

Model

Browser



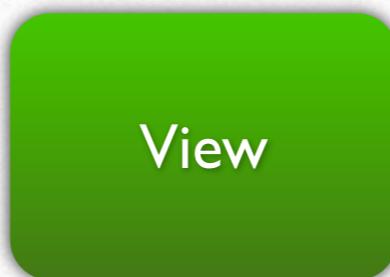
Browser

Dom

View

Model

Browser



Browser

Dom

View

Model

Browser



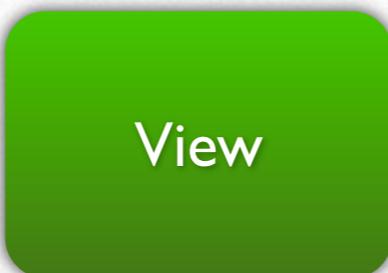
Browser

Dom

View

Model

Browser



Browser

Dom

View

Model



BACKBONE.JS

MV*



BACKBONE.JS

Backbone.Events

```
var object = {};  
  
$.extend(object, Backbone.Events);  
  
object.on('eventName', function() {  
  console.log('1');  
});  
  
object.on('eventName', function() {  
  console.log('2');  
});  
  
object.trigger('eventName');  
  
// prints '1' and '2'
```

```
var object = {};  
$.extend(object, Backbone.Events);
```

```
object.on('eventName', function() {  
  console.log('1');  
});
```

```
object.on('eventName', function() {  
  console.log('2');  
});
```

```
object.trigger('eventName');
```

```
// prints '1' and '2'
```

```
var object = {};  
$.extend(object, Backbone.Events);
```

```
object.on('eventName', function() {  
  console.log('1');  
});
```

```
object.on('eventName', function() {  
  console.log('2');  
});
```

```
object.trigger('eventName');
```

```
// prints '1' and '2'
```

```
var object = {};  
  
$.extend(object, Backbone.Events);  
  
object.on('eventName', function() {  
  console.log('1');  
});  
  
object.on('eventName', function() {  
  console.log('2');  
});  
  
object.trigger('eventName');  
  
// prints '1' and '2'
```

```
var object = {};  
  
$.extend(object, Backbone.Events);  
  
object.on('eventName', function() {  
  console.log('1');  
});  
  
object.on('eventName', function() {  
  console.log('2');  
});  
  
object.trigger('eventName');  
  
// prints '1' and '2'
```



BACKBONE.JS

Backbone.Model

```
var Person = Backbone.Model.extend({  
  initialize: function() {  
    console.log("I'm alive!");  
  }  
});  
  
new Person();
```

```
var Person = Backbone.Model.extend({  
  initialize: function() {  
    console.log("I'm alive!");  
  }  
});
```

```
new Person();
```

```
var Person = Backbone.Model.extend({  
  initialize: function() {  
    console.log("I'm alive!");  
  }  
});
```

```
new Person();
```

```
var Person = Backbone.Model.extend({});  
  
var me = new Person({name: 'Radoslav'});  
  
me.get('name');                                // Radoslav  
me.set({lastName: 'Stankov'});  
me.set(lastName, 'Stankov');  
me.get('lastName');                            // Stankov
```

```
var Person = Backbone.Model.extend({});
```

```
var me = new Person({name: 'Radoslav'});  
  
me.get('name');                                // Radoslav  
me.set({lastName: 'Stankov'});  
me.set(lastName, 'Stankov');  
me.get('lastName');                            // Stankov
```

```
var Person = Backbone.Model.extend({});
```

```
var me = new Person({name: 'Radoslav'});
```

```
me.get('name'); // Radoslav
```

```
me.set({lastName: 'Stankov'});
```

```
me.set(lastName, 'Stankov');
```

```
me.get('lastName'); // Stankov
```

```
var Person = Backbone.Model.extend({});  
  
var me = new Person({name: 'Radoslav'});  
  
me.get('name');                                // Radoslav  
me.set({lastName: 'Stankov'});  
me.set(lastName, 'Stankov');  
me.get('lastName');                            // Stankov
```

```
var Person = Backbone.Model.extend({});
```

```
var me = new Person({name: 'Radoslav'});
```

```
me.get('name'); // Radoslav
```

```
me.set({lastName: 'Stankov'});
```

```
me.set(lastName, 'Stankov');
```

```
me.get('lastName'); // Stankov
```

```
var Person = Backbone.Model.extend({  
  defaults: {  
    name: 'John',  
    lastName: 'Doe',  
  }  
});
```

```
var me = new Person();  
me.get('name'); // John  
me.get('lastName'); // Doe
```

```
var Person = Backbone.Model.extend({  
  defaults: {  
    name: 'John',  
    lastName: 'Doe',  
  }  
});
```

```
var me = new Person();  
me.get('name'); // John  
me.get('lastName'); // Doe
```

```
var Person = Backbone.Model.extend({  
  defaults: {  
    name: 'John',  
    lastName: 'Doe',  
  }  
});
```

```
var me = new Person();  
me.get('name'); // John  
me.get('lastName'); // Doe
```

```
var Person = Backbone.Model.extend({  
  defaults: {  
    name: 'John',  
    lastName: 'Doe',  
  }  
});
```

```
var me = new Person();  
me.get('name'); // John  
me.get('lastName'); // Doe
```

```
var Calculator = Backbone.Model.extend({  
  defaults: {  
    value: 0  
  },  
  value: function(){  
    return this.get('value');  
  },  
  sum: function(value){  
    this.set(value, value + this.get('value'));  
  },  
  reset: function(){  
    this.set(value, 0);  
  }  
});
```

```
var Calculator = Backbone.Model.extend({  
  defaults: {  
    value: 0  
  },  
  value: function(){  
    return this.get('value');  
  },  
  sum: function(value){  
    this.set(value, value + this.get('value'));  
  },  
  reset: function(){  
    this.set(value, 0);  
  }  
});
```

```
var Calculator = Backbone.Model.extend({  
  defaults: {  
    value: 0  
  },  
  value: function(){  
    return this.get('value');  
  },  
  sum: function(value){  
    this.set(value, value + this.get('value'));  
  },  
  reset: function(){  
    this.set(value, 0);  
  }  
});
```

```
var Calculator = Backbone.Model.extend({  
  defaults: {  
    value: 0  
  },  
  value: function(){  
    return this.get('value');  
  },  
  sum: function(value){  
    this.set(value, value + this.get('value'));  
  },  
  reset: function(){  
    this.set(value, 0);  
  }  
});
```

```
var Calculator = Backbone.Model.extend({  
  defaults: {  
    value: 0  
  },  
  value: function(){  
    return this.get('value');  
  },  
  sum: function(value){  
    this.set(value, value + this.get('value'));  
  },  
  reset: function(){  
    this.set(value, 0);  
  }  
});
```

```
var cal = new Calculator();

cal.on('change:value', function(model, value){
  console.log(value);
});

cal.on('change', function(model){
  console.log(model.get('value'));
});

cal.on('all', function(eventName) {
  console.log('I see everything!', eventName);
});
```

```
var cal = new Calculator();
```

```
cal.on('change:value', function(model, value){  
  console.log(value);  
});
```

```
cal.on('change', function(model){  
  console.log(model.get('value'));  
});
```

```
cal.on('all', function(eventName) {  
  console.log('I see everything!', eventName);  
});
```

```
var cal = new Calculator();

cal.on('change:value', function(model, value){
  console.log(value);
});
```

```
cal.on('change', function(model){
  console.log(model.get('value'));
});
```

```
cal.on('all', function(eventName) {
  console.log('I see everything!', eventName);
});
```

```
var cal = new Calculator();

cal.on('change:value', function(model, value){
  console.log(value);
});

cal.on('change', function(model){
  console.log(model.get('value'));
});

cal.on('all', function(eventName) {
  console.log('I see everything!', eventName);
});
```

```
var cal = new Calculator();

cal.on('myEvent', function(){
  console.log('KaBoom....');
});

cal.trigger('myEvent');
```

```
var Product = Backbone.Model.extend({  
  urlRoot: '/products'  
});
```

```
var chair = new Product({  
  name: 'chair',  
  price: 10  
});
```

```
chair.save();
```

```
// POST /products
```

```
var Product = Backbone.Model.extend({
  url: function(){
    return '/products/' + (this.isNew() ? '' : this.id);
  }
});

var chair = new Product({
  id: 5,
  name: 'chair',
  price: 10
});

chair.save();

// PUT /products/1
```

<http://backbonejs.org/#Model-save>

И още...

- validate
- escape
- has
- unset
- clear
- hasChanged
- changedAttributes
- previousAttributes
- fetch
- toJSON
- clone



BACKBONE.JS

Backbone.View

Todos

What needs to be done?

□ Model

□ Collection

□ View

□ Router

4 items left.

div.content 480px×291px

```
<!DOCTYPE html>
▼<html>
  ▶<head>...</head>
  ▶<style type="text/css">...</style>
  ▶<style type="text/css" style="display: none !important;">...</style>
  ▼<body>
    <!-- Todo App Interface -->
    ▼<div id="todoapp">
      ▶<div class="title">...</div>
      ▼<div class="content">
        ▼<div id="create-todo">
          <input id="new-todo" placeholder="What needs to be done?" type="text">
          <span class="ui-tooltip-top" style="display: none; ">Press Enter to save this task</span>
        </div>
        ▶<div id="todos">...</div>
        ▶<div id="todo-stats">...</div>
      </div>
    </div>
    <!-- Templates -->
    ▶<script type="text/template" id="item-template">...</script>
    ▶<script type="text/template" id="stats-template">...</script>
  </body>
```

```
var UserNameView = Backbone.View.extend({  
  tagName: 'input',  
  className: 'string optional',  
  id: 'user-name',  
  attributes: {  
    type: 'string',  
    name: 'user[name]'  
  }  
});
```

```
var userName = new UserNameView();
```

```
console.log(userName.el);
```

```
<input type="string" name="user[name]" id="user-name" class="string optional">
```

```
var UserNameView = Backbone.View.extend({  
  tagName: 'input',  
  className: 'string optional',  
  id: 'user-name',  
  attributes: {  
    type: 'string',  
    name: 'user[name]'  
  }  
});
```

```
var userName = new UserNameView();
```

```
console.log(userName.el);
```

```
<input type="string" name="user[name]" id="user-name" class="string optional">
```

```
var UserNameView = Backbone.View.extend({  
  tagName: 'input',  
  className: 'string optional',  
  id: 'user-name',  
  attributes: {  
    type: 'string',  
    name: 'user[name]'  
  }  
});
```

```
var userName = new UserNameView();
```

```
console.log(userName.el);
```

```
<input type="string" name="user[name]" id="user-name" class="string optional">
```

```
var UserNameView = Backbone.View.extend({  
  tagName: 'input',  
  className: 'string optional',  
  id: 'user-name',  
  attributes: {  
    type: 'string',  
    name: 'user[name]'  
  }  
});
```

```
var userName = new UserNameView();  
  
console.log(userName.el);
```

```
<input type="string" name="user[name]" id="user-name" class="string optional">
```

```
var UserNameView = Backbone.View.extend({  
  tagName: 'input',  
  className: 'string optional',  
  id: 'user-name',  
  attributes: {  
    type: 'string',  
    name: 'user[name]'  
  }  
});
```

```
var userName = new UserNameView();
```

```
console.log(userName.el);
```

```
<input type="string" name="user[name]" id="user-name" class="string optional">
```

```
var UserNameView = Backbone.View.extend({  
  tagName: 'input',  
  className: 'string optional',  
  id: 'user-name',  
  attributes: {  
    type: 'string',  
    name: 'user[name]'  
  }  
});
```

```
var userName = new UserNameView();
```

```
console.log(userName.el);
```

```
<input type="string" name="user[name]" id="user-name" class="string optional">
```

```
var UserNameView = Backbone.View.extend({  
  tagName: 'input',  
  className: 'string optional',  
  id: 'user-name',  
  attributes: {  
    type: 'string',  
    name: 'user[name]'  
  }  
});
```

```
var userName = new UserNameView();
```

```
console.log(userName.el);
```

```
<input type="string" name="user[name]" id="user-name" class="string optional">
```

```
var UsersListView = Backbone.View.extend({  
  el: '#users-list'  
});  
  
var userList = new UsersListView();  
  
console.log(userList.el);
```

```
var EditBoxView = Backbone.View.extend({});  
  
var element = $('#edit-box-view').get(0),  
    editBox = new EditBoxView({el: element});  
  
console.log(editBox.el === element);
```

```
var DocumentView = Backbone.View.extend({  
  events: {  
    'dblclick': 'open',  
    'click .grid .doc': 'select',  
    'customEvent .title': 'custom'  
  },  
  open: function() {},  
  select: function() {},  
  custom: function() {}  
});
```

```
var DocumentView = Backbone.View.extend({  
  events: {  
    'dblclick': 'open',  
    'click .grid .doc': 'select',  
    'customEvent .title': 'custom'  
  },  
  open: function() {},  
  select: function() {},  
  custom: function() {}  
});
```

```
var DocumentView = Backbone.View.extend({  
  events: {  
    'dblclick': 'open',  
    'click .grid .doc': 'select',  
    'customEvent .title': 'custom'  
  },  
  open: function() {},  
  select: function() {},  
  custom: function() {}  
});
```

```
var DocumentView = Backbone.View.extend({  
  events: {  
    'dblclick': 'open',  
    'click .grid .doc': 'select',  
    'customEvent .title': 'custom'  
  },  
  open: function() {},  
  select: function() {},  
  custom: function() {}  
});
```

```
var DocumentView = Backbone.View.extend({  
  events: {  
    'dblclick': 'open',  
    'click .grid .doc': 'select',  
    'customEvent .title': 'custom'  
  },  
  open: function() {},  
  select: function() {},  
  custom: function() {}  
});
```

```
var DocumentView = Backbone.View.extend({  
  events: {  
    'dblclick': 'open',  
    'click .grid .doc': 'select',  
    'customEvent .title': 'custom'  
  },  
  open: function() {},  
  select: function() {},  
  custom: Function()  
});
```

```
var DocumentView = Backbone.View.extend({  
  events: {  
    'dblclick': 'open',  
    'click .grid .doc': 'select',  
    'customEvent .title': 'custom'  
  },  
  open: function() {},  
  select: function() {},  
  custom: function() {}  
});
```

```
<script type="text/template" id="news">
  <h1><%= title %></h1>
  <time><%= created_at %></time>
  <p><%= text %></p>
</script>
```

```
var NewsView = Backbone.View.extend({  
  template: _.template($('#news').html()),  
  render: function() {  
    this.$el.html(this.template(this.model));  
    return this;  
  }  
});
```

```
var NewsView = Backbone.View.extend({  
  template: _.template($('#news').html()),  
  render: function() {  
    this.$el.html(this.template(this.model));  
    return this;  
  }  
});
```

```
var NewsView = Backbone.View.extend({  
  template: _.template($('#news').html()),  
  render: function() {  
    this.$el.html(this.template(this.model));  
    return this;  
  }  
});
```

```
var view = new NewsView({  
  model: {  
    title: "News Title",  
    created_at: "Today",  
    text: "Long text"  
  }  
});  
  
$('body').html(view.render().el);
```

```
<script type="text/template" id="news">
  <h1><%= title %></h1>
  <time><%= created_at %></time>
  <p><%= text %></p>
</script>
```

```
<div>
  <h1>News Title</h1>
  <time>Today</time>
  <p>Long text</p>
</div>
```

Model

View

DOM

Model

View

DOM



Model

View

DOM

Model

View

DOM



Model

View

DOM

Model

View



DOM

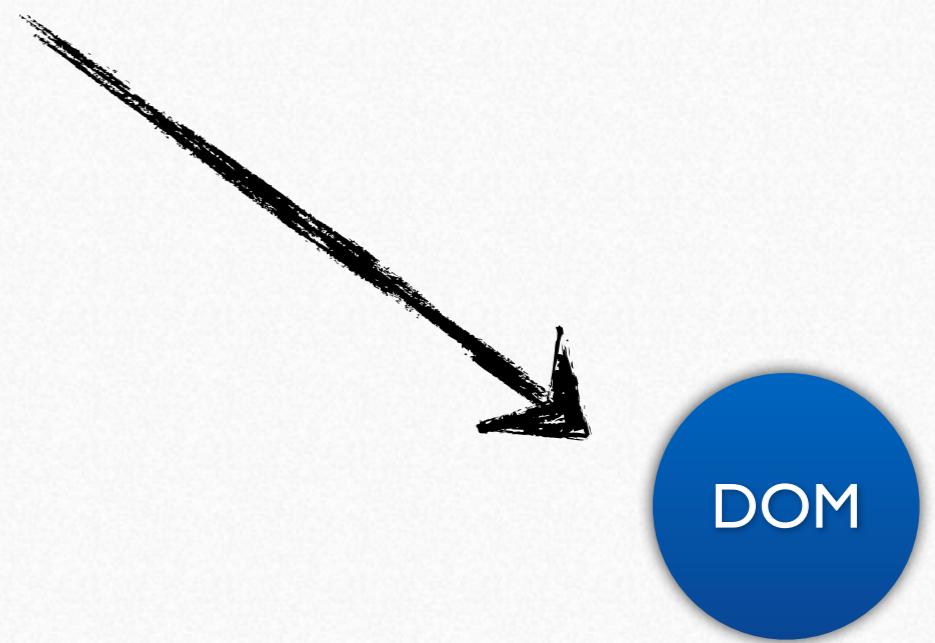
Model

View

DOM

Model

View



Model

View

DOM

Model

View

DOM

Model

View

DOM

View 2

Model

View

View 2

DOM



Model

View

DOM

View 2

Model

View



DOM

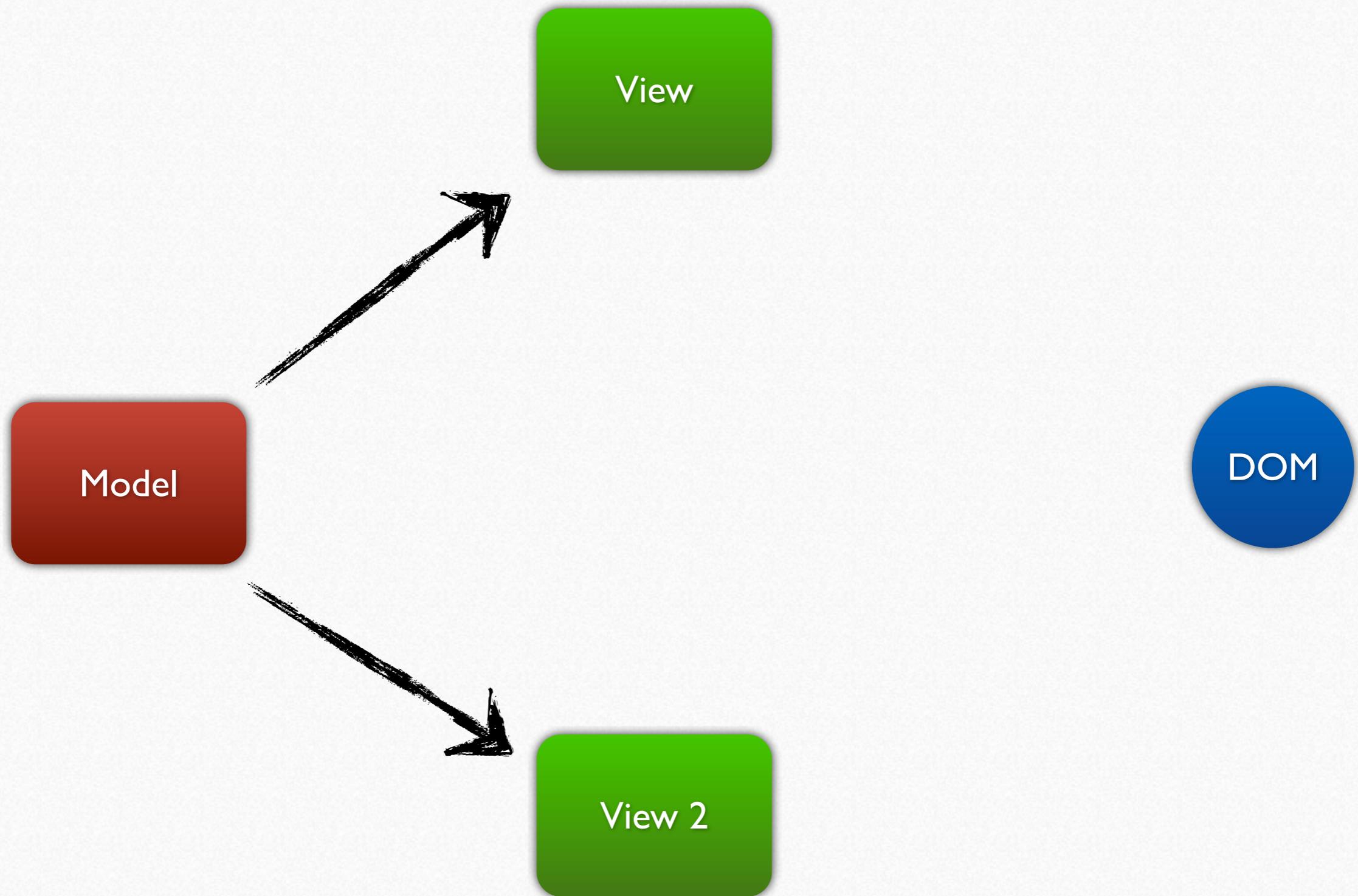
View 2

Model

View

DOM

View 2



Model

View

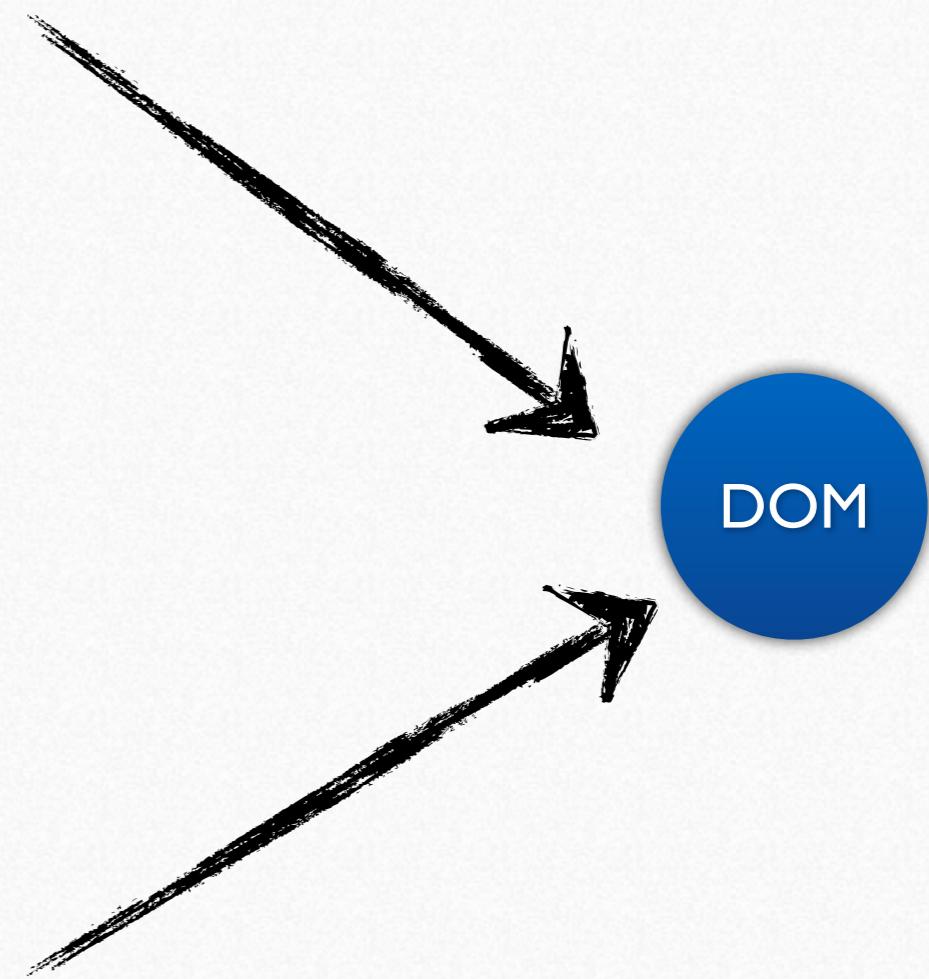
DOM

View 2

Model

View

View 2



DOM

Model

View

DOM

View 2

Model

View

DOM

View 2

Model

View

View 2

DOM

Model

View

View 2

View 3

DOM

Model

View

View 2

View 3

View 4

DOM

Model

View

View 2

View 3

View 4

View .. N

DOM

Model

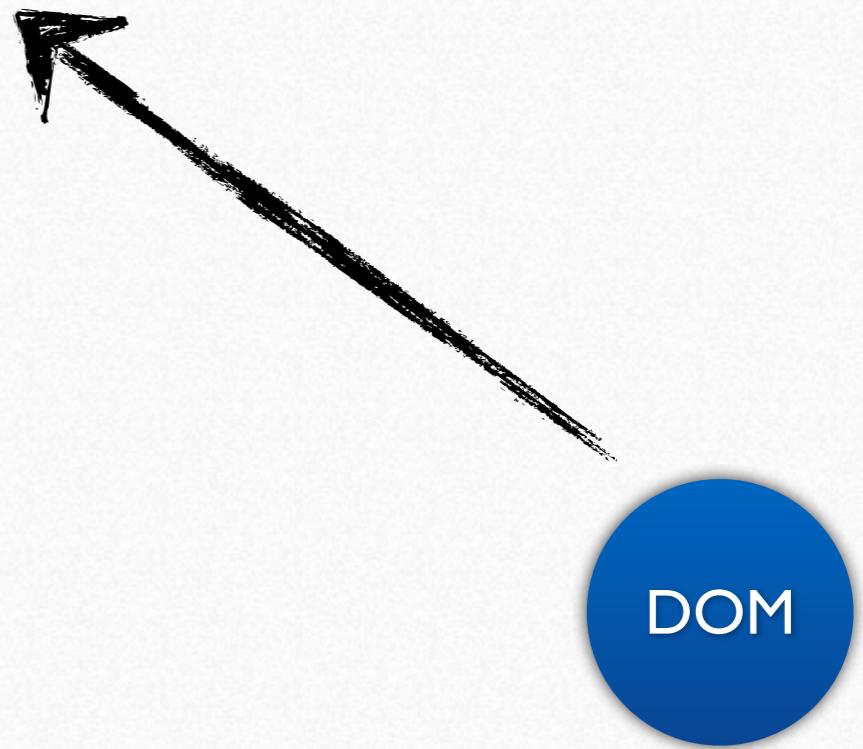
View

View 2

View 3

View 4

View .. N



Model

View

View 2

View 3

View 4

View .. N

DOM

Model

View

View 2

View 3

View 4

View .. N

DOM



Model

View

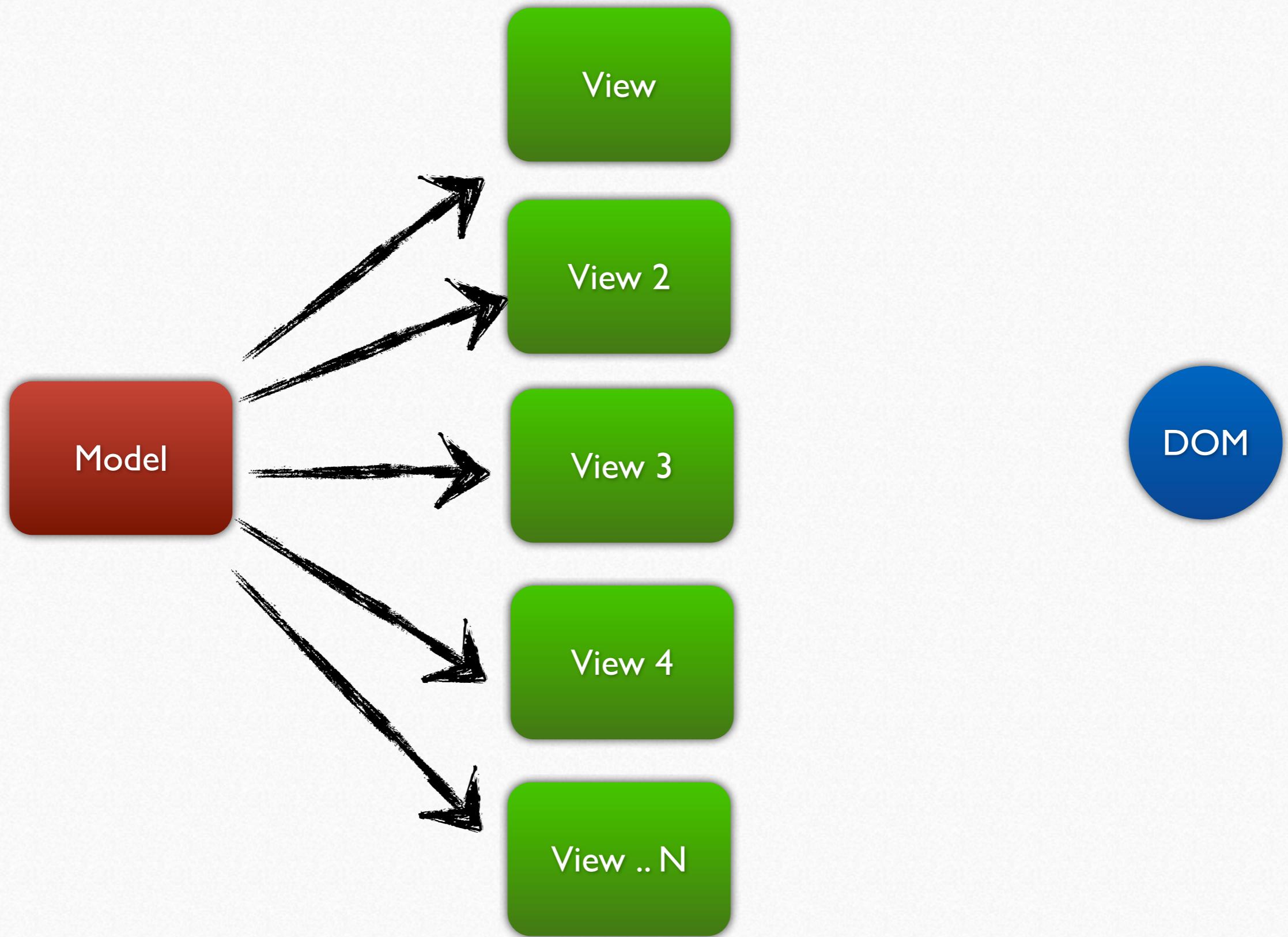
View 2

View 3

View 4

View .. N

DOM



Model

View

View 2

View 3

View 4

View .. N

DOM

Model

View

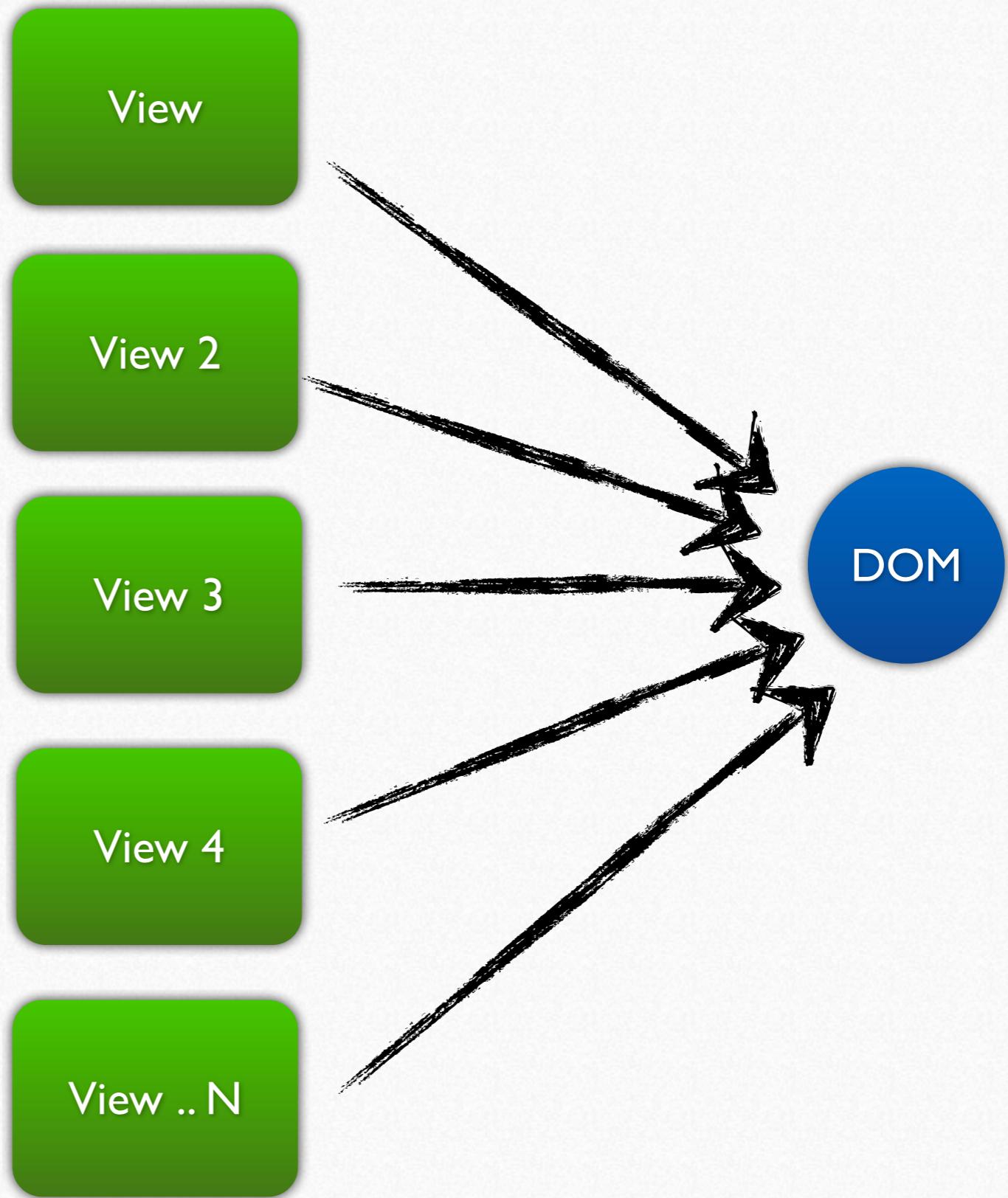
View 2

View 3

View 4

View .. N

DOM



Model

View

View 2

View 3

View 4

View .. N

DOM

Demo 1



```
var Calculator = Backbone.Model.extend({  
  defaults: {  
    value: 0  
  },  
  increment: function() {  
    this.set('value', this.get('value') + 1);  
  },  
  decrement: function() {  
    this.set('value', this.get('value') - 1);  
  },  
  getValue: function() {  
    return this.get('value');  
  }  
});
```

```
var ButtonsView = Backbone.View.extend({
  events: {
    'click .plus': 'plus',
    'click .minus': 'minus'
  },
  plus: function() {
    this.model.increment();
  },
  minus: function() {
    this.model.decrement();
  }
});
```

```
var DisplayView = Backbone.View.extend({
  initialize: function() {
    this.model.on('change:value', this.render, this);
    this.render();
  },
  render: function() {
    this.$el.html(this.model.getValue());
    return this;
  }
});
```

```
var cal = new Calculator();

new ButtonsView({model: cal, el: '.buttons'});
new DisplayView({model: cal, el: '.display'});
```



BACKBONE.JS

Backbone.Collection

```
var ProductsCollection = Backbone.Collection.extend({  
  model: Product  
});  
  
var products = new ProductsCollection();  
  
products.fetch();  
  
products.on('reset', function(list) {  
  console.log('Loaded', list.length, 'records');  
});
```

```
products.on('add', function(model) {  
  console.log('new product added');  
});
```

```
products.on('remove', function(model) {  
  console.log('item product removed');  
});
```

И още...

- Underscore Methods
- add / remove / at
- sort / comparator
- reset
- create
- url
- toJSON



BACKBONE.JS

Backbone.Router



<http://example.com/#/page/backbone-demo>

```
var AppRouter = Backbone.Router.extend({
  routes: {
    'pages' : 'index',
    'pages/search/:q': 'search',
    'pages/:id': 'show'
  },
  initialize: function() {
    console.log('initialize');
  },
  index: function() { /* code */ },
  search: function(query) { /* code */ },
  show: function(id) { /* code */ }
});
```

```
var app = new AppRouter();
```

```
Backbone.history.start();
```

```
var AppRouter = Backbone.Router.extend({
  routes: {
    'pages' : 'index',
    'pages/search/:q': 'search',
    'pages/:id': 'show'
  },
  initialize: function() {
    console.log('initialize');
  },
  index: function() { /* code */ },
  search: function(query) { /* code */ },
  show: function(id) { /* code */ }
});
```

```
var app = new AppRouter();
```

```
Backbone.history.start();
```

```
var AppRouter = Backbone.Router.extend({
  routes: {
    'pages' : 'index',
    'pages/search/:q': 'search',
    'pages/:id': 'show'
  },
  initialize: function() {
    console.log('initialize');
  },
  index: function() { /* code */ },
  search: function(query) { /* code */ },
  show: function(id) { /* code */ }
});
```

```
var app = new AppRouter();
```

```
Backbone.history.start();
```

```
var AppRouter = Backbone.Router.extend({  
  routes: {  
    'pages': 'index',  
    'pages/search/:q': 'search',  
    'pages/:id': 'show'  
  },  
  initialize: function() {  
    console.log('initialize');  
  },  
  index: function() { /* code */ },  
  search: function(query) { /* code */ },  
  show: function(id) { /* code */ }  
});
```

```
var app = new AppRouter();
```

```
Backbone.history.start();
```

```
var AppRouter = Backbone.Router.extend({
  routes: {
    'pages' : 'index',
    'pages/search/:q': 'search',
    'pages/:id': 'show'
  },
  initialize: function() {
    console.log('initialize');
  },
  index: function() { /* code */ },
  search: function(query) { /* code */ },
  show: function(id) { /* code */ }
});
```

```
var app = new AppRouter();
```

```
Backbone.history.start();
```

```
var AppRouter = Backbone.Router.extend({  
  routes: {  
    'pages': 'index',  
    'pages/search/:q': 'search',  
    'pages/:id': 'show'  
  },  
  initialize: function() {  
    console.log('initialize');  
  },  
  index: function() { /* code */ },  
  search: function(query) { /* code */ },  
  show: function(id) { /* code */ }  
});
```

```
var app = new AppRouter();
```

```
Backbone.history.start();
```

```
var AppRouter = Backbone.Router.extend({
  routes: {
    'pages' : 'index',
    'pages/search/:q': 'search',
    'pages/:id': 'show'
  },
  initialize: function() {
    console.log('initialize');
  },
  index: function() { /* code */ },
  search: function(query) { /* code */ },
  show: function(id) { /* code */ }
});
```

```
var app = new AppRouter();
```

```
Backbone.history.start();
```

```
var AppRouter = Backbone.Router.extend({  
  routes: {  
    'pages' : 'index',  
    'pages/search/:q': 'search',  
    'pages/:id' : 'show'  
  },  
  initialize: function() {  
    console.log('Initialize');  
  },  
  index: function() { /* code */ },  
  search: function(query) { /* code */ },  
  show: function(id) { /* code */ }  
});
```

```
var app = new AppRouter();
```

```
Backbone.history.start();
```

```
var AppRouter = Backbone.Router.extend({
  routes: {
    'pages' : 'index',
    'pages/search/:q': 'search',
    'pages/:id': 'show'
  },
  initialize: function() {
    console.log('initialize');
  },
  index: function() { /* code */ },
  search: function(query) { /* code */ },
  show: function(id) { /* code */ }
});
```

```
var app = new AppRouter();
```

```
Backbone.history.start();
```

```
var AppRouter = Backbone.Router.extend({
  routes: {
    'pages' : 'index',
    'pages/search/:q': 'search',
    'pages/:id': 'show'
  },
  initialize: function() {
    console.log('initialize');
  },
  index: function() { /* code */ },
  search: function(query) { /* code */ },
  show: function(id) { /* code */ }
});
```

```
var app = new AppRouter();
```

```
Backbone.history.start();
```

```
var AppRouter = Backbone.Router.extend({
  routes: {
    'pages' : 'index',
    'pages/search/:q': 'search',
    'pages/:id': 'show'
  },
  initialize: function() {
    console.log('initialize');
  },
  index: function() { /* code */ },
  search: function(query) { /* code */ },
  show: function(id) { /* code */ }
});
```

```
var app = new AppRouter();
```

```
Backbone.history.start();
```

```
app.navigate('pages', {trigger: true});  
app.navigate('pages/search/title', {trigger: true});  
app.navigate('pages/3', {trigger: true});
```

```
app.navigate('pages', {trigger: true});  
app.navigate('pages/search/title', {trigger: true});  
app.navigate('pages/3', {trigger: true});
```

```
app.navigate('pages', {trigger: true});  
app.navigate('pages/search/title', {trigger: true});  
app.navigate('pages/3', {trigger: true});
```

site.com/path#pages

site.com/path#pages/search/title

site.com/path#pages/3

HTML



```
Backbone.history.start({  
  pushState: true,  
  root: '/path/'  
});
```

site.com/path#pages

site.com/path#pages/search/title

site.com/path#pages/3

site.com/path/pages

site.com/path/pages/search/title

site.com/path/pages/3

Demo 2





BACKBONE.JS



@jimweirich

Jim Weirich

OH: "It is half-assed, but it the right half
of the ass."

25 Jan via [Twitter for iPad](#)



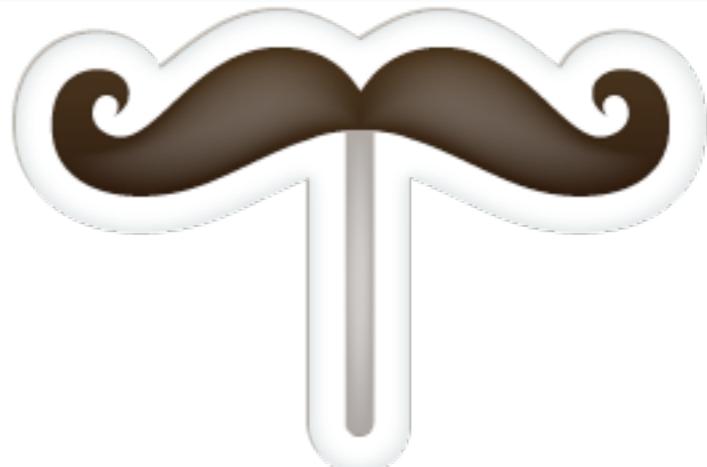


UNDERSCORE.JS



UNDERSCORE.JS

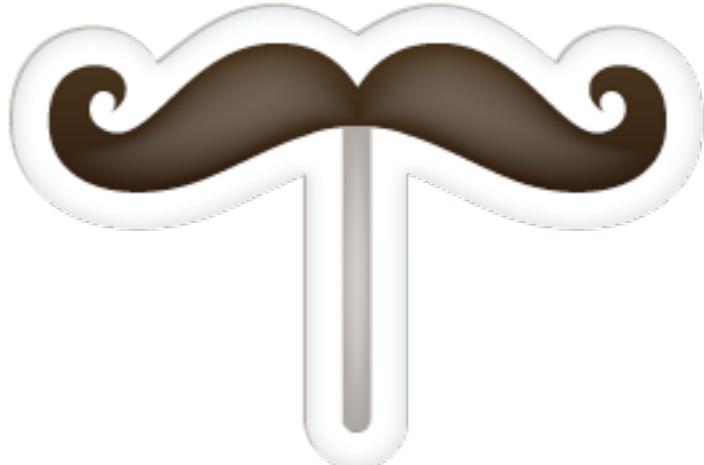
handlebars





underscore.js

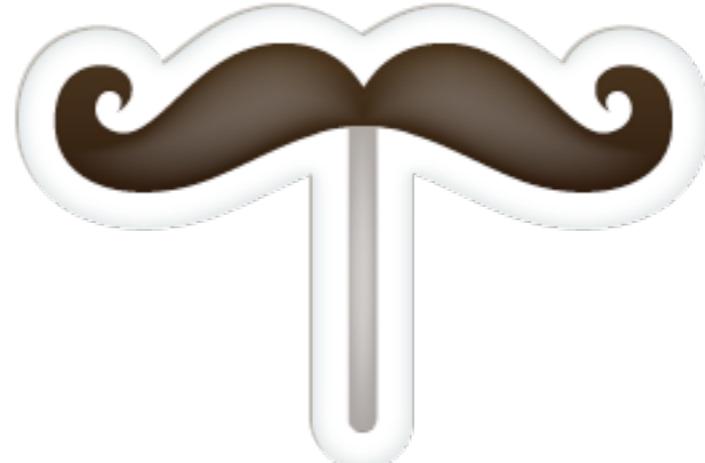
handlebars





underscore.js

handlebars

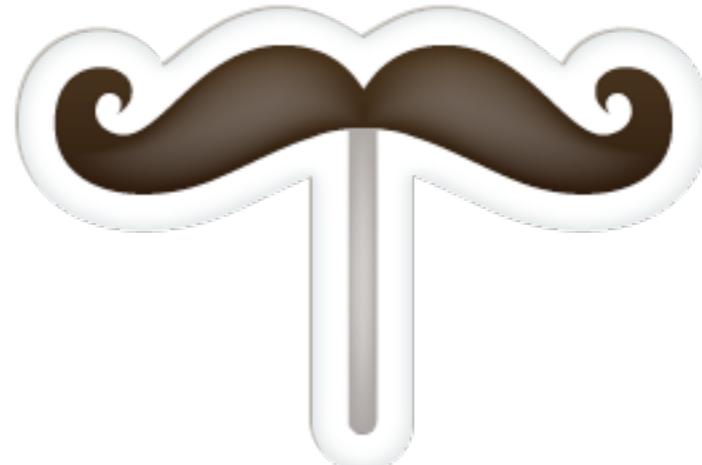


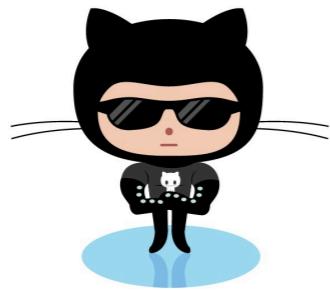


underscore.js



handlebars





Plugins

<https://github.com/RStankov/backbone-handlebars>

<https://github.com/RStankov/backbone-bind-to>

<https://github.com/n-time/backbone.validations>

<https://github.com/PaulUithol/Backbone-relational>

<https://github.com/thoughtbot/backbone-support>

Resources

Resources

<http://backbonejs.org/>

Resources

<http://backbonejs.org/>

<https://workshops.thoughtbot.com/products/l-backbone-js-on-rails>

Resources

<http://backbonejs.org/>

<https://workshops.thoughtbot.com/products/I-backbone-js-on-rails>

<http://recipeswithbackbone.com/>

Resources

<http://backbonejs.org/>

<https://workshops.thoughtbot.com/products/l-backbone-js-on-rails>

<http://recipeswithbackbone.com/>

<https://github.com/addyosmani/backbone-fundamentals>

Problems



Problems

Problems

Ghost views

Problems

Ghost views

Plugin system

Problems

Ghost views

Plugin system

Template system

Problems

Ghost views

Plugin system

Template system

Subviews

Problems

Ghost views

Plugin system

Template system

Subviews

Weak conventions

Problems

Ghost views

Plugin system

Template system

Subviews

Weak conventions

Backbone.Application

Alternatives

Alternatives

The Knockout logo is displayed on a red rectangular background with a black double-line border. The word "Knockout." is written in a bold, white, italicized sans-serif font. A horizontal white line is positioned below the letter "t".

Knockout.

Alternatives

Spine



Alternatives

ember

Spine

Knockout.



Code & Slides:

<https://github.com/RStankov/backbone-presentation>

Questions?

Thanks for listening :)

@rstankov