



# BACKBONE.JS

## LESSONS FROM THE BATTLEFIELD

# @ZACHLENDON

- Independent Software Consultant
- Java, Groovy, Grails, Javascript, Mobile
- Used Backbone.js for over a year on previous project

# WHO ARE YOU?

# TL;DR

- *How you approach building your backbone.js application is critical to its ultimate success*
- *Using Backbone.js alone for your client-side SPA development is not something I'd really recommend, but you should understand it and know some strategies for best using it*
- *You can then approach the myriad of Backbone-based options out there with the proper foundation*



A photograph of a yellow envelope being held by a person's hand. The envelope has printed text on it, which is partially obscured by large, bold, black and white text overlaid on the image. The black text reads "Why?" and the white text below it reads "BACKBONE".

ETIKETTÄ KONTAKTAPI

GRAVI

emka.com

Why?

BACKBONE



Inbox (31) - Yahoo! Mail - Microsoft Internet Explorer provided by [REDACTED]

File Edit View Favorites Tools Help

Back Search Favorites Go

Address http://us.mc304.mail.yahoo.com/mc/welcome?.gx=1&tm=1285784687&rand=dppcrtnlldn8#\_pg=showFolder&fid=Inbox&rand=684331820&hash=0cad33b5968a7467a4e2f9718acb321&.jsrand=6808983 Go

Links AlarmPoint® Citrix Clarity CMR Command Lookup Tool F5 ForMyInfo.com MeetingPlace My Courses Nodocs OWA SAP SempraNet Dir SPECTRUM Team Docs (Ver) SnagIt

Hi [REDACTED] | Sign Out | All-New Mail | Help ▾

Preview Mail w/ Toolbar

Yahoo! Mail My Mail AdChoices Search

**YAHOO! MAIL**  
Classic

 **Citibank® lets you link accounts to transfer money easily.**

Learn more >  
visit stories.citi.com

Mail Contacts Calendar Notepad

New Mail Mail Search Try the new Yahoo! Mail

What's New? - Mobile Mail -

**Inbox**

View All | From Contacts | From Connections | Unread | Flagged

Messages 1-25 of 63 First | Previous | Next | Last

	From	Subject	Move...	25 of 63	Size
<input type="checkbox"/>	ATMmarketplace.com	Which ATM is right for your restaurant?	Last	11:36 AM	29KB
<input type="checkbox"/>	[REDACTED]@aol.com	Fwd: Holy Crap.....		10:44 AM	7KB
<input type="checkbox"/>	[REDACTED]@aol.com	Fwd: You know the honeymoon is over when the comedians star...		10:23 AM	63KB
<input type="checkbox"/>	Matt Blizek, Democracy...	This Saturday		9:27 AM	11KB
<input type="checkbox"/>	QuickCert	Get CompTIA A+ Certified Today - Time is running out		8:16 AM	17KB
<input type="checkbox"/>	American Eagle Outfitt...	Last Chance For Fleece On Sale + AE Jeans Under \$30		8:12 AM	24KB
<input type="checkbox"/>	Capital One	[REDACTED] transfer balances with no fees		6:26 AM	22KB
<input type="checkbox"/>	Macy's	Email exclusive! Free Shipping + Lowest Prices of the Season!		6:11 AM	19KB
<input type="checkbox"/>	[REDACTED]	[REDACTED]		6:00 AM	21KB

Folders

- Inbox (31)
- Drafts (7)
- Sent
- Spam (734) [Empty]
- Trash [Empty]

My Photos My Attachments



Gmail Calendar Documents Photos Reader Web more ▾ Hiking Fan ▾ 

**Gmail** by Google  SEARCH MAIL SEARCH THE WEB Show search options Create a filter

Mail  Archive Report spam Delete   Move to ▾ Labels ▾ More ▾  1 - 15 of 15  

**COMPOSE MAIL**

**Inbox (3)**

Starred 

Sent Mail

**Drafts (2)**

**Hiking (3)** 

Urgent! 

12 more ▾

**Chat** 

Search, add, or invite

 Hiking Fan Set status here ▾

 Call phone

 Arielle

 Emily

 Jason

 Michael

 Paul

 	Jason Cornwell	» Please return my stapler - Hi, You seem to have taken my stapler. Please,	1:10 pm
 	Paul McDonald	» Fun Hike Yesterday! - Thanks for the great hike yesterday, it was awesome	1:06 pm
 	Arielle Reinstein	» July 4th weekend - Hi there: I heard you'll be around this weekend and I'd like to go for a hike.	Jun 28
 	JS Bach	» Tonhalle concert Friday - Hey man, there's a great concert this Friday evening.	Jun 22
 	Christine Chiu	» Hi Hiking, Looking for opinion on my diet/fitness app - Hi Hiking, I bumped into you at the trailhead.	Jun 9
 	Yan Tseytlin (2), Draft	» Hey there! - I heard you found a great place to go hiking. Let me know when you're going.	Mar 28
 	Kenneth, me (2)	» Group dinner? - Sushi sounds great! On Fri, Mar 25, 2011 at 10:06 AM, Kenneth wrote:	Mar 25
 	Kenneth, me (2)	» Long time! - Hey Ken! Things have been really good! And lunch sounds great.	Mar 24
 	Michael Bolognino	» This weekend - Hi there. Let's meet up at 8PM tonight for burgers and then hit the trail.	Mar 24
 	Arielle Reinstein	» Dipsea trail - When it stops raining I really want to hike the Dipsea Trail again.	Mar 24
 	Jason Toff	» How are you? - Hey there, We haven't spoken in a while. How are you? Wow, you look great!	Mar 24
 	Jr Wikane	» VW Auction in Tacoma - Hi, I was doing a search on Google for VW's in Tacoma.	Mar 6
 	Google Voice	» New voicemail from (619) 810-5507 at 5:19 AM - Voicemail from (619) 810-5507	Mar 6

# Backbone.js











Tuts+ Premium courses teach you a single skill from top to bottom, inside out.



## Understanding Backbone in .NET

Web Development \ JavaScript

[Tweet](#) 1

[+1](#) 0

[Like](#)

If JavaScript spaghetti code is something that you're all too familiar with, it's time to change things. The solution is [Backbone.js](#), which provides the perfect structure for building modern web applications. In this course, as we work through the process of building a photo app, we'll learn how to use this framework within a .NET application.

# INTRODUCTORY LESSON

What impact does this have on my development world?

# WHY ARE WE DOING ALL THIS DEVELOPMENT ON THE CLIENT ANYWAY?

(Lea Verou: <http://youtu.be/0T57Ivn5-Pw>)

- "*Infinite Scalability*"
- "*Lower Server Cost*"
- "*Responsive UI*"
- *Trend towards everything client but Files, Security and Persisted Data*

# FOR JAVA DEVELOPERS THIS MEANS

- *Expectations of business folks (and how we meet those expectations) are evolving*
- *Spring MVC / !Struts! / etc turns into RESTful front-end to back-end services*
- *Javascript is your new front-end MV\* layer. Become a student of javascript*
- *Learn "Responsive UI" and become better friends with designers*
- *Lightweight javascript templates (and portion of your JS MV\* frameworks) are your new JSPs, GSPs, Velocity/Freemarker/etc server-side template*

# ENTER JAVASCRIPT/TEMPLATING LIBRARY/FRAMEWORK OVERLOAD

- *Infinite Number of Micro Javascript libraries (Microjs.com great resource for perusing)*
- *40+ Javascript MV\* Frameworks*
- *Over 100+ legitimate Mustache.js client-side templating forks + Jade, HAML, etc.*

# KEY FACTORS TO USE IN CHOOSING TECHNOLOGIES TO LEVERAGE IN THIS NEW WORLD

- *Do "relevant" POCs that provide a foundation to what you would build with a few examples*
- *The Three I'd recommend: Backbone.Marionette, Angular, and Ember*

# WHY USE (SOMETHING LIKE) BACKBONE - TRANSLATED

(Based off <http://www.rvl.io/mindelusions/backbonejs>)

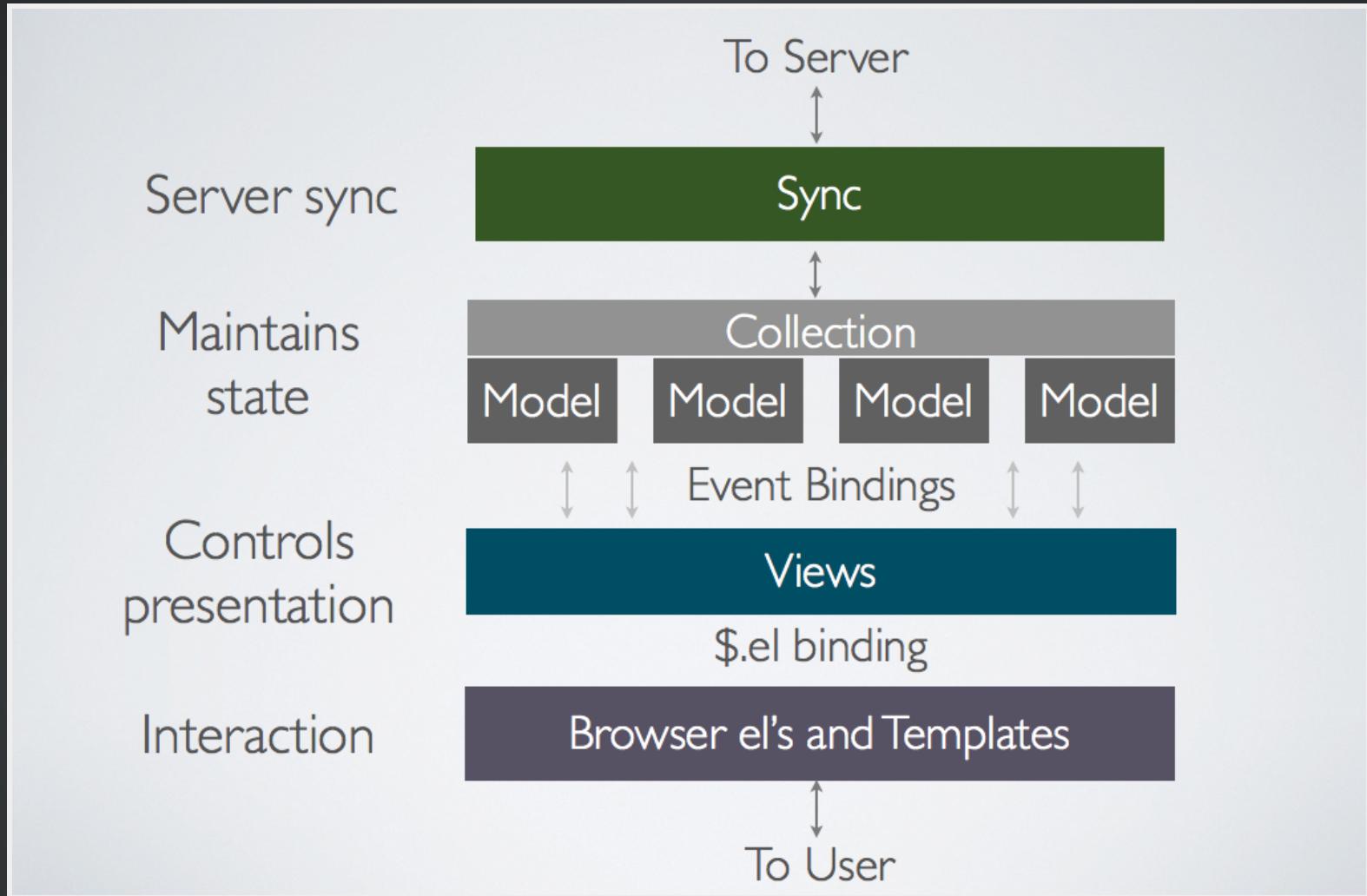
- *We really need MVC for the front end. Traditional methods leave us with code that's too coupled, messy and incredibly hard to maintain.*
- *Storing data and state in the DOM is a bad idea.*
- *Fat models and skinny controllers are the way to go. Workflow is simplified when business logic is taken care of by models.*
- *Templating is an absolute necessity. Putting HTML inside your JavaScript gives you bad karma.*

# YOUR DEVELOPMENT PROCESSES SHOULD BE STREAMLINED

- Whether its Webstorm, IntelliJ, or Sublime Text 2, use the best IDE for you that will increase your efficiency. Cost here is so minor in exchange for available efficiency gains
- *Compilers and Watchers*
- *Scaffolding*
- *Asset Management*
- *Dependency Management*
- *Build Processes*

# SO YOU'VE CHOSEN BACKBONE.JS (OR SOMETHING BUILT ON IT)

"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."



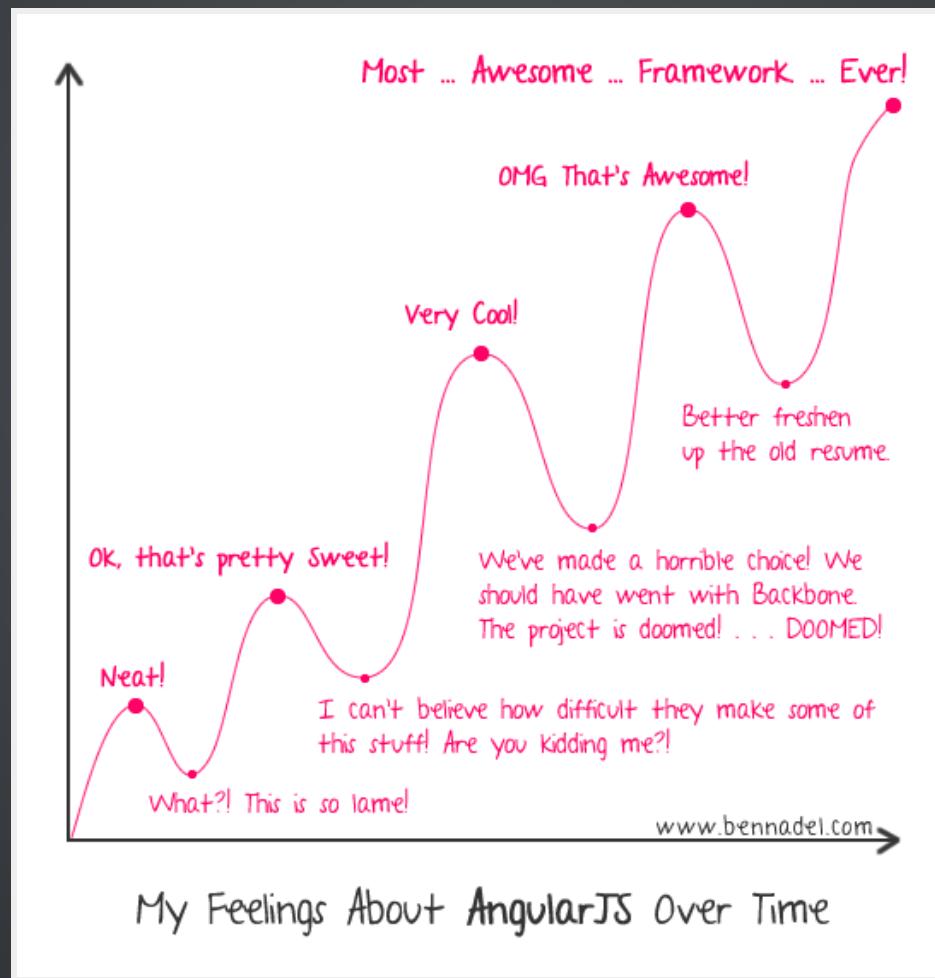
# WHERE YOU'LL WONDER ABOUT YOUR CHOICE OF BACKBONE.JS

- No *2-way data binding*
- No *view management*
- No *module support*
- Other frameworks will (at times) look sexier

"Most Limitations" addressable by micro frameworks  
(Backbone.Stickit, Backbone.LayoutManager, etc.) or more  
full-fledged Backbone frameworks - Thorax,  
Backbone.Marionette, Chaplin, etc.

# TYPICAL BEST CASE LIFE-CYCLE WITH NEW TECHNOLOGY ADOPTION

(Taken from <http://www.bennadel.com/>)



# LESSONS

# LESSON #1

APIs APIs APIs

# WITHOUT A WELL-DESIGNED RESTFUL JSON INTERFACE YOU WILL GET NOWHERE

- If doesn't exist (it usually doesn't), can be developed concurrently on teams with good communication and processes
- Your new "default" CRUD (a Rails 3-based RESTful API):

```
create → POST /collection  
read → GET /collection[/id]  
update → PUT /collection/id  
delete → DELETE /collection/id
```

- Though can be customized via model url/urlRoot property
- Back-end performance/scalability matters

# YOUR APIs WILL BE THE FUEL TO YOUR FRONT-END FIRE

- What data must I have to present initial view(s)?
- What data can I load on-demand?
- How will users interact with the view(s) they are presented with?
- How will the app respond to those interactions?

developers.helloverb.com/swagger/

**FOR DEVELOPERS**

# Swagger



**Reverb**

WHAT WE MAKE

VORDNIK

OR PUBLISHERS

OR DEVELOPERS

ATMOSPHERE

SCALATRA

**SWAGGER**

WORDNIK API

DEVELOPER BLOG

ABOUT US

Document your API with style

Swagger is a specification and complete framework implementation for describing, producing, consuming, and visualizing RESTful web services. The overarching goal of Swagger is to enable client and documentation systems to update at the same pace as the server. The documentation of methods, parameters, and models are tightly integrated into the server code, allowing APIs to always stay in sync. With Swagger, deploying managing, and using powerful APIs has never been easier.

**Who is responsible for Swagger?**

Both the specification and framework implementation are initiatives from Wordnik. Swagger was developed for Wordnik's own use during the development of [developer.wordnik.com](#) and the underlying system. Swagger development began in early 2010—the framework being released is currently used by Wordnik's APIs, which power both internal and external API clients.

**Why is Swagger useful?**

The Swagger framework simultaneously solves server, client, and documentation/sandbox needs. As a specification, it is language-agnostic. It also provides a long runway into new technologies and protocols beyond REST.

Swagger API Explorer

my-awesome-api.com

**user**

<b>word</b>	ShowHide   List Operations   Expand Operations   Raw	
<b>GET</b> /word.json/{word}/entries	Return entries for a word	
<b>GET</b> /word.json/{word}/examples	Returns examples for a word	
<b>POST</b> /word.json/{word}/examples	Fetches examples for a word	
<b>POST</b> /word.json/{word}/wordForms	Add a Relationship Map to a word	
<b>GET</b> /word.json/{word}/wordForms	Returns other forms of a word	
<b>DELETE</b> /word.json/{word}/wordForms	Deletes a relationship from a word	
<b>GET</b> /word.json/{word}	Given a word as a string, returns the WordObject that represents it	

**Parameters**

Parameter	Value	Description
word	(Required)	String value of WordObject to return
useCanonical		If true will try to return the correct word root ('cats' => 'cat'). If false returns exactly what was requested.
IncludeSuggestions		Return suggestions (for correct spelling, case variants, etc.)
shouldCreate		Create word if not existing

**Try it out!**

<b>GET</b> /word.json/{word}/definitions	Return definitions for a word
<b>GET</b> /word.json/{word}/stats	Returns word statistics
<b>GET</b> /word.json/{word}/topExample	Returns a top example for a word
<b>GET</b> /word.json/{word}/contextualLookup	Returns definitions for a word based on the sentence in which it is found
<b>POST</b> /word.json/{word}/contextualLookup	Returns definitions for a word based on the sentence in which it is found
<b>GET</b> /word.json/{word}/commentCount	Returns the number of comments on a word
<b>GET</b> /word.json/{word}/related	Return related words (thesaurus data) for a word
<b>GET</b> /word.json/{word}/listedIn	Returns WordLists containing a word
<b>GET</b> /word.json/{word}/listedInCount	Returns a count of lists a word appears in

# LESSON #2

TDD Javascript

# THE WORKFLOW IS GREAT AND SHOULDN'T BE HARD

- Tests clarify risk and thought, and document functionality
- If the test is too hard to write odds are the function is too complicated
- As for tools, I personally really like Jasmine. And Jasmine-JQuery. And Testem (or Karma/Testacular).
- Mocha appears great for Coffeescript
- Use Jasmine Bootstrap Reporter for pretty reports, Jasmine JUnit reporter for JUnit XML output
- For dependencies, Jasmine has nice 'spies' for mocking/spying/faking behavior. Use Sinon.js for better spy feedback. Jasmine has fixtures for mocking the DOM and jasmine-ajax library is great for mocking ajax requests

```
it('should sync correctly', function() {
  //mockup data
  var event = new App.Models.Event({ title: "TekSystems JUG"});

  //fake server
  this.server = sinon.fakeServer.create();

  //fake response
  this.server.respondWith("POST", "/events",
    [ 200, {"Content-Type": "application/json", '>{"id": 1, "title": "TekSystems JUG"}' }]);

  var spy = sinon.spy()
  event.on("sync", spy);

  event.save();

  this.server.respond();

  expect(spy).toHaveBeenCalledOnce();
  expect(spy).toHaveBeenCalledWith(event);
  expect(event.get("title")).toEqual("TekSystems JUG");

  this.server.restore
});
```

# LESSON #3

Models Rule

# MODELS ARE THE HEART OF A JAVASCRIPT APPLICATION

In Backbone they, generically speaking, interact with data

- Perform Validation
- Define default values
- Perform business logic on encapsulated data
- Manage changes to data

In Backbone collections manage sets of models and as such their importance in function is very similar to that of models

Underscore.js and Lodash.js are key libraries to leverage to interact with collections

```
define(['underscore', 'backbone'], function(_, Backbone) {
  var TodoModel = Backbone.Model.extend({
    localStorage: new Backbone.LocalStorage('todos-backbone'),
    defaults: {
      content: 'empty todo...',
      done: false
    },
    initialize: function() {
    },
    // Toggle the `done` state of this todo item.
    toggle: function() {
      this.save({done: !this.get('done')});
    },
    // Remove this Todo from *localStorage* and delete its view.
    clear: function() {
      this.destroy();
      this.view.remove();
    }
  });
  return TodoModel;
});
```

What is wrong with this above example from the Todos app?...

## ADDITIONAL MODEL RECOMMENDATIONS

- They should not know about views
- They don't all have to be persisted
- Override Backbone sync to change persistence strategies
- For example, Persist to localStorage when possible  
(backbone-localstorage.js good option to use)

# NESTED MODELS

- Common, no official way to handle
- **Backbone.Relational helps**
- Override parse and toJSON method pre/post AJAX requests when dealing with nested models

# AND ONE FINAL MODEL RECOMMENDATION

- Drive form population and handle non-view form changes  
...By using Backbone.stickit

Which to me is Backbone's missing 2-way binding framework

# FOR SIMPLE RADIO BUTTONS IN YOUR BACKBONE VIEW

```
bindings: {  
  'input[name=color]': {  
    observe: 'color',  
    onSet: 'setSelectedValue'  
  }  
,
```

Then typically at end of "render" function...

```
this.stickit();  
//or optionally this.stickit(this.pathToModel, this.bindings)
```

# OR A MORE COMPLEX BINDING FOR SELECT DROP-DOWNS

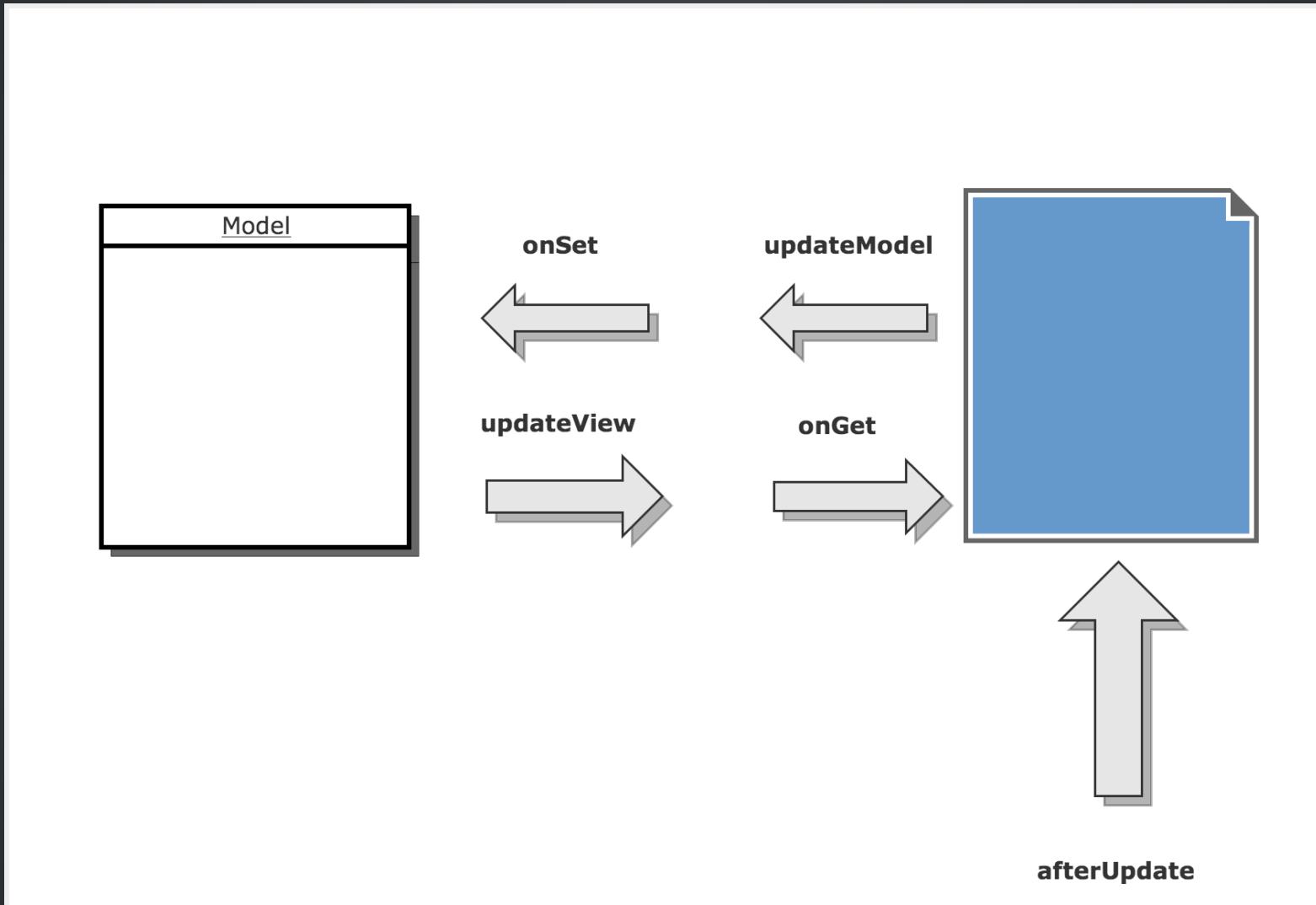
```
bindings: {  
    'select': {  
        observe: 'state',  
        onSet: 'setSelectedValue',  
        selectOptions: {  
            collection: 'this.collection',  
            labelPath: 'text',  
            valuePath: 'id'  
        }  
    }  
,
```

# WHICH CAN SIMPLIFY TEMPLATES TO LOOK LIKE

```
<li>
  <select name="states" id="{{id}}" class="form-select required">
  />
</li>
```

With Stickit rendering the option tags for you

# STICKIT FLOW



# LESSON #4

Raise and bind to events to ensure maximum simplicity

- Great for helping simplify your code
- Backbone.Model, Backbone.View, Backbone.Collection, Backbone.Router and Backbone.History prototypes all inherit from Backbone.Events
- Gives you access to on(), off(), listenTo(), stopListening() and trigger() methods to manage your events
- Model change events:  
create,change,destroy,sync,error,add,remove
- Model change events propagated up through collections
- Easy to create your own custom events
- Easy to create global event listener if event needs bubbled to multiple views

```
var globalEventListener = _.extend({}, Backbone.Events)
```

## Events property of view defines a hash of declarative callbacks for DOM events

```
events:{  
    "click a.continue" : "next",  
    "swiperight" : "back",  
    "pagebeforechange" : "render",  
    "pagechange" : "postRender"  
},
```

While the generic syntax for binding callbacks to Backbone events looks like the following and ensure "this" is not lost

```
initialize:function () {  
    this.collection.on('add', this.load, this);  
},
```

# LESSON #5

Break down your Views and Templates as much as possible

- Use Backbone.LayoutManager to help you define different layouts for your app and assemble views (this is also what Marionette uses, for example)

```
define([
  "./Header",
  "./Footer",
  "./Content",
  "collections/Events"
], function(Header, Footer, Content, Events){
  if(!window.layout) {
    //layout for the app
    window.layout = new Backbone.Layout({
      template: "#main",
      views: {
        "header": new Header(),
        ".content": new Content({
          collection: new Events(window.events)
        }),
        "footer": new Footer()
      }
    });
  }
  return window.layout;
});
```

- Which might be invoked in an app.js via:

```
require([
  "views/layout"
], function(layout){
  // Attach layout to the main container.
  layout.$el.appendTo("body");

  // Render the Layout
  layout.render();
});
```

- Templates should be "business"-logicless and as simple as possible

```
<li>
  <label for="radioButton-{id}">{text}</label>
  <input type="radioButton" name="option" id="radioButton-{id}" class="radio required" value="{id}" />
</li>
```

# LESSON #6

Control the URL

- You should be able to reload the app at any point and get a state that is at least close to the user's current view state
- Do this through initializing Backbone.History at start of processing and using Backbone navigate function

```
Backbone.history.navigate(this.getPageUrl(), {trigger:true});
```

- options:
- trigger:true/false - invoke router action (or not)
- replace:true - change url but don't add history

# BONUS LESSON #1

Mobile - specifically JQuery Mobile (JQM)

- Good baseline summary of routing issue in Addy Osmani's Backbone Fundamental book
- Typically best to override JQM routing handling
- Place to do what Addy suggests how to handle is in on mobileinit event

```
$(document).bind("mobileinit", function(){  
    $.mobile.defaultPageTransition = 'slide';  
    $.event.special.swipe.horizontalDistanceThreshold = 180;  
    $.mobile.hashListeningEnabled = false;  
    $.mobile.pushStateEnabled = false;  
    $.ajaxSetup({  
        cache: false  
    });  
});
```

## JQM CHANGEPAGE() AND ITS ASSOCIATED PAGEBEFORECHANGE/PAGECHANGE EVENTS

- Need to pay attention to changePage direction ("reverse")
- If JQM page changing to is not static, need to create page, add to DOM, then change to it using its selector.
- Can only hook into page events in a view that has an \$el that is a parent of the \$el corresponding with the view being moved via the changePage

# BONUS LESSON #2 - AVOID THE ZOMBIE APOCALYPSE



©Waters03

- Remove views from the DOM AND release events
- Pre 0.99

```
this.remove();

//remove all models bindings
this.model.off(null, null, this);

//unbind events set on this view
this.off();
```

- 0.99 and greater

```
//will call stopListening()
//so use .listenTo() instead of .on() for events to have events released automatically
this.remove();
```

- Alternative: frameworks like Backbone.Marionette has built-in handling of this for you

# FINAL THOUGHTS

- Further Reading:  
**Backbone Fundamentals by Addy Osmani**
- Questions/Comments/etc?
- @zachlendon / [zach.lendon@gmail.com](mailto:zach.lendon@gmail.com)

**THANK YOU!**