

Client-side Kung-Fu with Backbone.js

Gur Dotan

@gurdotan

<http://github.com/gurdotan>

Credits



igloolab.com
[@iloveigloo](#)

michele.berto.li
[@MicheleBertoli](#)



Agenda

Preface

Why Backbone.js

Backbone.js Architecture

Backbone.js code

Sample App – if time permits

Questions



Preface

Client side templates

Template

```
<div>
  <h1> Name: {{name}} </h1>
  <h2> Title:  {{title}} </h2>
</div>
```

+

=

Context

```
{
  name : "Shimon",
  title : "President"
}
```

HTML

```
<div>
  <h1> Name: Shimon </h1>
  <h2> Title: President </h2>
</div>
```



Client side templates

```
<script id="person-template" type="text/x-handlebars-template">
  <div>
    <h1> Name: Shimon </h1>
    <h2> Title: President </h2>
  </div>
</script>
```

```
var template = Handlebars.compile($("#person-template"));

var html = template({ name: "Shimon", title: "President" });

$("#body").append(html);
```

Underscore

- Utility tool-belt
- Cross browser
- Ruby inspired syntax

UNDERScore.JS



```
_.each(["Yair", "Zipi", "Shelly"], function(item) {  
  // So something...  
});
```


Why Backbone.js



Why Backbone.js

Browsers used to be weak, stupid creatures

1994



2001



Why Backbone.js

- Browsers became powerful
- Web applications became more complex
- The complexity moved to the client

Why Backbone.js

Yahoo (1998)

YAHOO!

[New](#) [Cool](#) [Today's News](#) [More Yahoos](#)

[1998 Winter Games](#) results, schedules, news

[MegaMarketing BENEFITS EXPOSED!](#)

[Academy Award Nominations](#)

Search [options](#)

[Yahoo! Chat](#) with Wall Street guru **Jim Cramer**, supermodel **Frederique**

[Yellow Pages](#) - [People Search](#) - [Maps](#) - [Classifieds](#) - [Personals](#) - [Chat](#) - [Free Email](#)
[Shopping](#) - [My Yahoo!](#) - [News](#) - [Sports](#) - [Weather](#) - [Stock Quotes](#) - [more...](#)

- Arts and Humanities**
[Architecture](#), [Photography](#), [Literature](#)...
- Business and Economy [Xtra!]**
[Companies](#), [Finance](#), [Employment](#)...
- Computers and Internet [Xtra!]**
[Internet](#), [WWW](#), [Software](#), [Multimedia](#)...
- Education**
[Universities](#), [K-12](#), [College Entrance](#)...
- Entertainment [Xtra!]**
[Cool Links](#), [Movies](#), [Music](#), [Humor](#)...
- Government**
[Military](#), [Politics \[Xtra!\]](#), [Law](#), [Taxes](#)...
- Health [Xtra!]**
[Medicine](#), [Drugs](#), [Diseases](#), [Fitness](#)...
- News and Media [Xtra!]**
[Current Events](#), [Magazines](#), [TV](#), [Newspapers](#)...
- Recreation and Sports [Xtra!]**
[Sports](#), [Games](#), [Travel](#), [Autos](#), [Outdoors](#)...
- Reference**
[Libraries](#), [Dictionaries](#), [Phone Numbers](#)...
- Regional**
[Countries](#), [Regions](#), [U.S. States](#)...
- Science**
[CS](#), [Biology](#), [Astronomy](#), [Engineering](#)...
- Social Science**
[Anthropology](#), [Sociology](#), [Economics](#)...
- Society and Culture**
[People](#), [Environment](#), [Religion](#)...

[What's New](#) - [Weekly Picks](#) - [Today's Web Events](#) - [Yahoo! Internet Life](#)
[Yahoo!igans!](#) for Kids - [Visa Shopping Guide](#) - [Yahoo! Style](#) - [3D Stock Viewer](#)

World Yahoos [Australia & NZ](#) - [Canada](#) - [Denmark](#) - [France](#) - [Germany](#) - [Japan](#) - [Korea](#)
[Norway](#) - [SE Asia](#) - [Sweden](#) - [UK & Ireland](#)

Yahoo! Metros [Atlanta](#) - [Austin](#) - [Boston](#) - [Chicago](#) - [Dallas / Fort Worth](#) - [Los Angeles](#)
[Get Local](#) [Miami](#) - [Minneapolis / St. Paul](#) - [New York](#) - [S.F. Bay](#) - [Seattle](#) - [Wash D.C.](#)

[Smart Shopping with](#)

[How to Suggest a Site](#) - [Company Info](#) - [Openings at Yahoo!](#) - [Contributors](#) - [Yahoo! How-To](#)

Yahoo (2010)

YAHOO!

[Web](#) [Images](#) [Video](#) [Local](#) [Shopping](#) [more](#)

Search: [Web Search](#)

[Yahoo! Home](#) [My Yahoo!](#) [Y! Singapore](#) [Sep 9, 2008](#) [Page Options](#)

Featured [Entertainment](#) [Sports](#) [Video](#)

Pushing memory to the limit
Researchers are astonished after testing the brain in a way no previous experiment ever attempted. [» Story](#)
Find memory improvement exercises
Activity helps seniors' cognition

News [World](#) [Local](#) [Finance](#)

As of 9:45 p.m.
• Benazir Bhutto's widower sworn in as Pakistan's president
• Obama turning up heat on Palin as she energizes GOP
• Gulf Coast prepares for Hurricane Ike [» Category 1 storm](#)
• Fannie, Freddie plan binds taxpayers to troubled housing market
• Auto safety group calls on states to raise the legal driving age
• JetBlue auctions off more than 300 round-trip flights on eBay
• Man with Big Mac obsession says he's eaten 23,000 since 1972
[» More: News | Popular | Election '08](#)

Marketplace
When is the best time to buy?
Expert advice on timing the market & key points to consider before buying. Visit Yahoo! Real Estate now.
Go to Yahoo! Shopping and compare prices on thousands of phones. Find your new cell phone now.
Are you a Celtics fan? Get NBA championship jerseys, tees, hats, and more at the Yahoo! Sports Shop.
Survey says play Family Feud. All the fun of the TV show on your PC. Play Family Feud at Yahoo! Games.

Small Business
• Get a Web Site
• Domain Names
• Sell Online
• Search Ads

Featured Services
• Downloads
• Health
• Kids
• GeoCities
• Mail Plus
• Y! International

Check your mail status: [Sign In](#) [Free mail: Sign Up](#)

[Mail](#) [Messenger](#) [Radio](#)
[Weather](#) [Local](#) [Horoscopes](#)

Yahoo! HotJobs
Find the right one:
Job Searching Made Easy

Key Tip for Career Success!
10 Minutes to a New Job
Keep your Career in Motion

Salary Jackpot!
Max your Salary in a Downturn
The New Power Jobs

Find a new job: [Go](#)

[» Search for jobs outside the U.S.](#)

Yahoo! Autos: Be a Better Car Buyer
Get a look at the 2008 Wagons

[Audi](#) [Ford](#) [BMW](#) [Porsche](#) [Chevrolet](#)

Pulse - What Yahoos Are Into

Most Popular Movie Collections

- Top Rated Sci-Fi Films
- Best Buddy Movies
- Blonde Highlights
- Gold Medal Movies
- Scientifically Inaccurate
- Forbes Best Paid Actors

[» More movie collections](#)

Today's Top Searches

1. Britney Spears
2. Jennifer Aniston
3. Joan Baez
4. Tom Brady
5. Coney Island New York
6. Bonnie Hunt Show
7. Tommy Lee Jones
8. Fannie Mae
9. Keith Oldermann
10. United Airlines

Advertise with Us | [Search Marketing](#) | [Privacy Policy](#) | [Terms of Service](#) | [Suggest a Site](#) | [Yahoo! Telemundo](#) | [Send Feedback](#) | [Help](#)
Copyright © 2008 Yahoo! Inc. All rights reserved. [Copyright/IP Policy](#) | [Company Info](#) | [Participate in Research](#) | [Jobs](#)

Why Backbone.js

From server-side to client-side

We need efficient tools



Why Backbone.js

jQuery is cool but...

Why Backbone.js

jQuery is cool but...

We have to store object informations into the DOM

```
var list = "";
$.each(data, function (index, value) {
    list += "<li id='item-" + value.Id + "'>" + value.Name + "</li>";
});
$("ul").append(list);
```

Why Backbone.js

jQuery is cool but...

We have to store object informations into the DOM

```
var list = "";
$.each(data, function (index, value) {
    list += "<li id='item-" + value.Id + ">" + value.Name + "</li>";
});
$("ul").append(list);
```


Why Backbone.js

jQuery is cool but...

jQuery callback hell

```
$.getJSON("/Items", function (data) {  
  
    var list = "";  
    $.each(data, function (index, value) {  
        list += "<li id=\"item-" + value.Id + "\">" + value.Name + "</li>";  
    });  
    $("ul").append(list);  
  
    $("li").click(function () {  
        var $this = $(this);  
        var id = $this.attr("id").replace("item-", "");  
        $.post("/Items", { id: id }, function () {  
            $this.fadeOut(function () {  
                $this.remove();  
            });  
        });  
    });  
});
```

Why Backbone.js

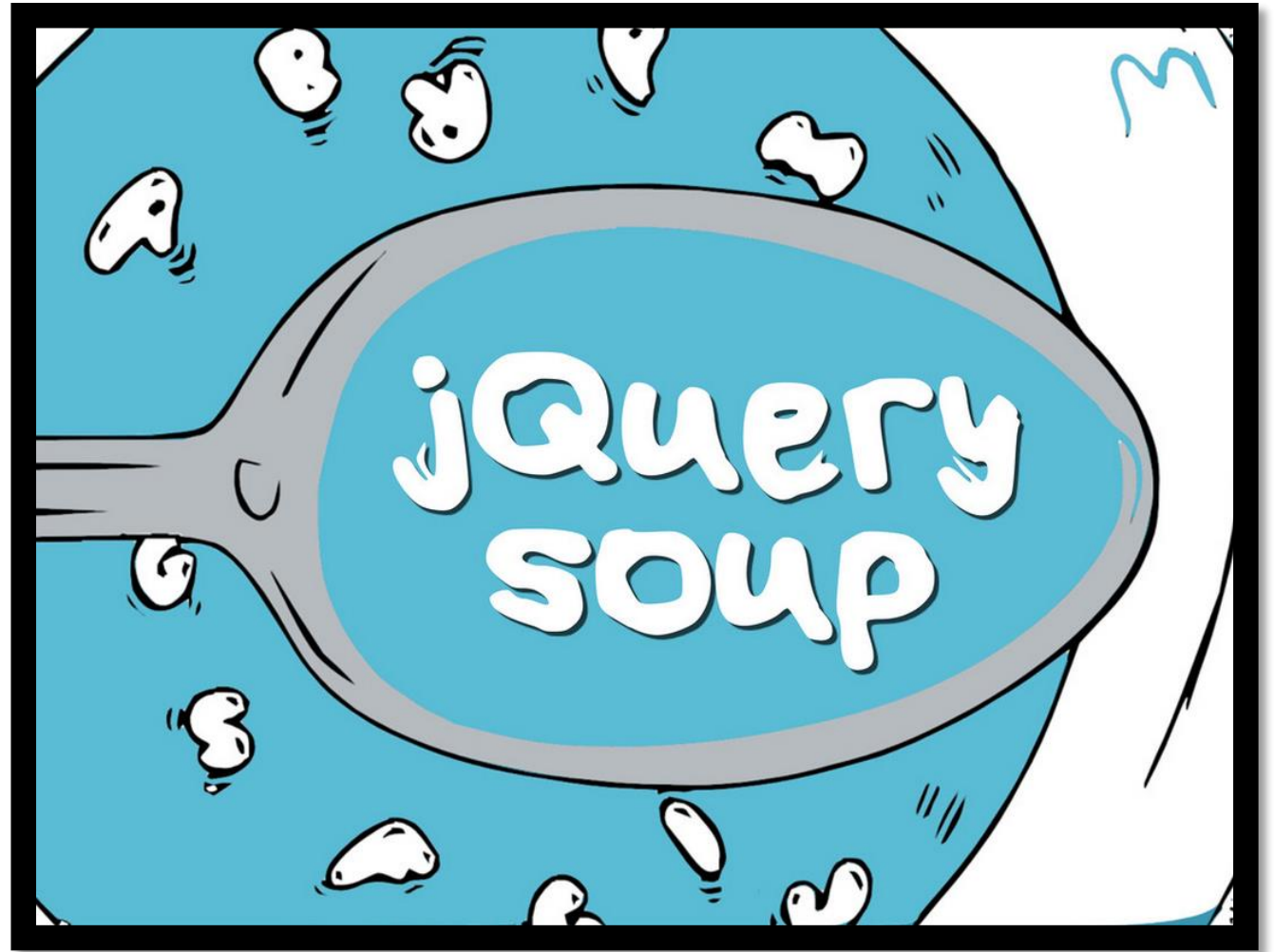
jQuery is cool but...

jQuery callback hell

```
$.getJSON("/Items", function (data) {  
  
    var list = "";  
    $.each(data, function (index, value) {  
        list += "<li id=\"item-" + value.Id + "\">" + value.Name + "</li>";  
    });  
    $("ul").append(list);  
  
    $("li").click(function () {  
        var $this = $(this);  
        var id = $this.attr("id").replace("item-", "");  
        $.post("/Items", { id: id }, function () {  
            $this.fadeOut(function () {  
                $this.remove();  
            });  
        });  
    });  
});
```

Why Backbone.js

“It's all too easy to create JavaScript applications that end up as tangled piles of jQuery selectors and callbacks.”



Why Backbone.js

So, what we need?

- Separation of Concerns.
- Decoupling UI from Data.
- Describe what, not how

Why Backbone.js

Bad Separation of Concerns

```
$.getJSON("/Items", function (data) {  
  var list = "";  
  $.each(data, function (index, value) {  
    list += "<li id='\"item-\" + value.Id + '\">\" + value.Name + "</li>\"";  
  });  
  $("ul").append(list);  
  
  $("li").click(function () {  
    var $this = $(this);  
    var id = $this.attr("id").replace("item-", "");  
    $.post("/Items", { id: id }, function () {  
      $this.fadeOut(function () {  
        $this.remove();  
      });  
    });  
  });  
});
```

Server communication

Markup \ JS spaghetti

DOM event handling

DOM manipulation

Animation

Why Backbone.js

So, what do we really need?

- Models and Views
- Events (so models and views can “talk”).
- A solid routing system.
- Server data layer. (not in lecture scope)
- A lightweight JavaScript framework.

Why Backbone.js

It exists and it's called:
Backbone.js



Who's
using it?



Real World

Linkedin



Real World

FourSquare



Real World

Groupon



Real World

Basecamp

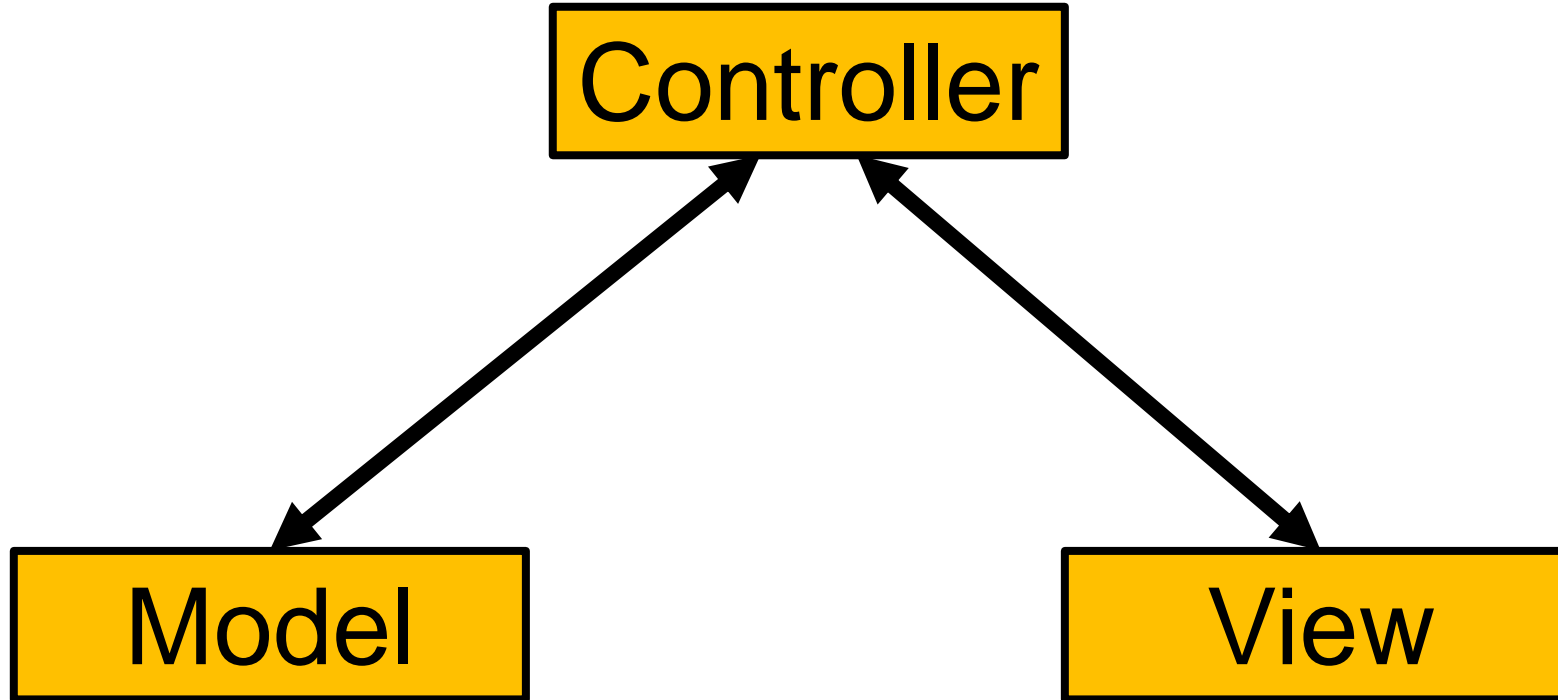


Architecture



Architecture

MVC



Architecture



Jeremy
Ashkenas

DocumentCloud 

Creator of:

 BACKBONE.JS

CoffeeScript

UNDERSCORE.JS

Architecture



Jeremy
Ashkenas

DocumentCloud

Want to know more?
Use the source, Luke



Architecture

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.

Architecture

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.

Architecture

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.

Architecture

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.

Architecture

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.

Architecture

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.

Architecture

Dependencies:

UNDERSCORE.JS

+



```
<script src="//js/libs/underscore-1.4.4.min.js"></script>
<script src="//js/libs/jquery-1.9.1.min.js"></script>
<script src="//js/libs/backbone-1.0.0.min.js"></script>
```

Model

- Represents data
- Throws events.
- Reusable.
- Handling persistence. (not in scope)



Architecture

Model / Collection - View - Template - Router - Utilities

Model – server persistence (not in scope)

• Fetch	→	HTTP GET	/url
• Save (new)	→	HTTP POST	/url
• Save	→	HTTP PUT	/url/id
• Destroy	→	HTTP DELETE	/url/id

Defining a Model

```
var Item = Backbone.Model.extend({  
  idAttribute: "studentId",  
  urlRoot: "/Items"           // not in lecture scope  
});
```

Architecture

Model / Collection - View - Template - Router - Utilities

Models + Events

```
var item = new Item();
```

Models + Events

```
var item = new Item();  
item.on("change:name", function() {  
  alert("Name is: " + this.get("name"));  
});
```

Models + Events

```
var item = new Item();  
item.on("change:name", function() {  
  alert("Name is: " + this.get("name"));  
});  
item.set( { name: "Moshe" } );           // trigger change  
  
// What do we get?
```

Architecture

Model / Collection - View - Template - Router - Utilities

Backbone.Model
+ set \ get

```
var item = new Item();  
item.set({  
  name : "Austin Powers",  
  sex   : "Yes Please"  
});  
  
alert(item.get("name"));
```

VS

Native Javascript
objects + attributes

```
var item = {};  
item.name = "Austin Powers";  
item.sex   = "Yes Please";  
  
alert(item.name);
```

```
var item = new Item();  
item.set( { Name: "Igloo" } ); // trigger change  
item.save(); // trigger sync. Not in lecture scope
```

```
var item = new Item();  
item.set( { Name: "Igloo" } ); // trigger change  
item.save(); // trigger sync. Not in lecture scope
```


Architecture

Model / Collection - View - Template - Router - Utilities

Model methods

- extend
- constructor / initialize
- get
- set
- escape
- has
- unset
- clear
- id
- idAttribute
- cid
- attributes
- defaults
- toJSON
- fetch
- save
- destroy
- validate
- isValid
- url
- urlRoot
- parse
- clone
- isNew
- change
- hasChanged
- changedAttributes
- previous
- previousAttributes

Collection

- A list of models.
- Underscore methods.



Defining a Collection

```
var Items = Backbone.Collection.extend({  
  model: Item,  
  url: "/Items"      // not in lecture scope  
});
```

Initializing a Collection

```
var items = new Items();  
items.add({name: "Pencil"}); // triggers "add" on collection
```

Initializing a Collection

```
var items = new Items([ {name: "Pencil"}, {name: "Pen"} ]);
```

Fetching a Collection from server

```
var items = new Items();  
items.fetch(); // trigger “reset”. Not in lecture scope
```

Collection methods

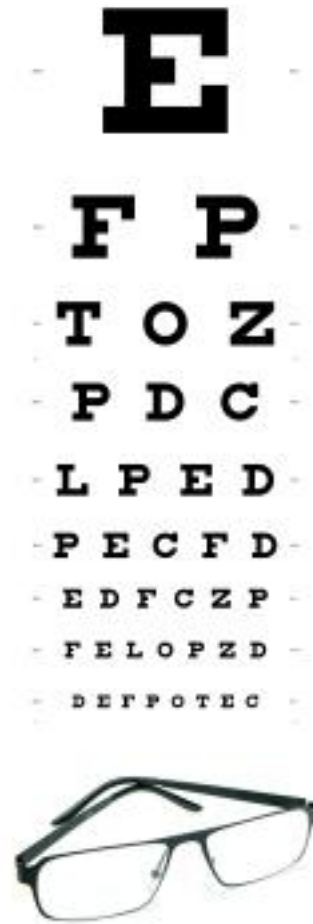
- extend
- model
- constructor / initialize
- models
- toJSON
- add
- remove
- get
- getByCid
- at
- length
- comparator
- sort
- pluck
- url
- parse
- fetch
- reset
- create

Collection – underscore methods

- forEach (each)
- map
- reduce (foldl, inject)
- reduceRight (foldr)
- find (detect)
- filter (select)
- reject
- every (all)
- some (any)
- include
- invoke
- max
- min
- sortBy
- groupBy
- sortedIndex
- shuffle
- toArray
- size
- first
- initial
- rest
- last
- without
- indexOf
- lastIndexOf
- isEmpty
- chain

View

- Manipulates the DOM.
- Delegates DOM events.
- Has a Model / Collection.



Views

listen to Model

events

Architecture

Model / Collection - View - Template - Router - Utilities

Events

Model

View



Trigger
events

Listen to
events



Artist

Album

1

```
var Album = Backbone.Model.extend();  
var album = new Album();
```

2

```
var AlbumView = Backbone.View.extend({  
  initialize: function() {  
    this.model.on("change", this.render)  
  },  
  render: function() {...}  
});  
  
var view = new AlbumView({model: album});
```

Architecture

Model / Collection - View - Template - Router - Utilities

Events

Model



3

```
album.set({ artist: "Pink Floyd"});  
album.set({ album: "The Wall"});
```

```
change:artist  
"Pink Floyd"  
  
change:album  
"The Wall"
```

View



4

```
// view.render  
// view.render
```

Architecture

Model / Collection - View - Template - Router - Utilities

Collection View



Architecture

Model / Collection - **View** - Template - Router - Utilities

```
var ListView = Backbone.View.extend({
  el: $("ul"),
  initialize: function () {
    this.collection.on("reset", this.render, this);
  },
  render: function () {
    this.collection.each(this.addItem, this);
    return this;
  },
  addItem: function (item) {
    var itemView = new ItemView({
      model: item
    });
    this.$el.append(itemView.el);
    itemView.render();
  }
});
```

```
var ItemView = Backbone.View.extend({
  tagName: "li",
  render: function () {
    this.$el.text(this.model.get("Name"));
    return this;
  }
});
```


Architecture

Model / Collection - **View** - Template - Router - Utilities

```
var ListView = Backbone.View.extend({
  el: $("ul"),
  initialize: function () {
    this.collection.bind("reset", this.render, this);
  },
  render: function () {
    this.collection.each(this.addItem, this);
    return this;
  },
  addItem: function (item) {
    var itemView = new ItemView({
      model: item
    });
    this.$el.append(itemView.el);
    itemView.render();
  }
});
```

```
var ItemView = Backbone.View.extend({
  tagName: "li",
  render: function () {
    this.$el.text(this.model.get("Name"));
    return this;
  }
});
```

Architecture

Model / Collection - **View** - Template - Router - Utilities

```
var ListView = Backbone.View.extend({
  el: $("ul"),
  initialize: function () {
    this.collection.on("reset", this.render, this);
  },
  render: function () {
    this.collection.each(this.addItem, this);
    return this;
  },
  addItem: function (item) {
    var itemView = new ItemView({
      model: item
    });
    this.$el.append(itemView.el);
    itemView.render();
  }
});
```

```
var ItemView = Backbone.View.extend({
  tagName: "li",
  render: function () {
    this.$el.text(this.model.get("Name"));
    return this;
  }
});
```

Architecture

Model / Collection - **View** - Template - Router - Utilities

```
var ListView = Backbone.View.extend({
  el: $("ul"),
  initialize: function () {
    this.collection.on("reset", this.render, this);
  },
  render: function () {
    this.collection.each(this.addItem, this);
    return this;
  },
  addItem: function (item) {
    var itemView = new ItemView({
      model: item
    });
    this.$el.append(itemView.el);
    itemView.render();
  }
});
```

```
var ItemView = Backbone.View.extend({
  tagName: "li",
  render: function () {
    this.$el.text(this.model.get("Name"));
    return this;
  }
});
```

Architecture

Model / Collection - **View** - Template - Router - Utilities

```
var ListView = Backbone.View.extend({
  el: $("ul"),
  initialize: function () {
    this.collection.on("reset", this.render, this);
  },
  render: function () {
    this.collection.each(this.addItem, this);
    return this;
  },
  addItem: function (item) {
    var itemView = new ItemView({
      model: item
    });
    this.$el.append(itemView.el);
    itemView.render();
  }
});
```

```
var ItemView = Backbone.View.extend({
  tagName: "li",
  render: function () {
    this.$el.text(this.model.get("Name"));
    return this;
  }
});
```

Architecture

Model / Collection - **View** - Template - Router - Utilities

```
var ListView = Backbone.View.extend({
  el: $("ul"),
  initialize: function () {
    this.collection.on("reset", this.render, this);
  },
  render: function () {
    this.collection.each(this.addItem, this);
    return this;
  },
  addItem: function (item) {
    var itemView = new ItemView({
      model: item
    });
    this.$el.append(itemView.el);
    itemView.render();
  }
});
```

```
var ItemView = Backbone.View.extend({
  tagName: "li",
  render: function () {
    this.$el.text(this.model.get("Name"));
    return this;
  }
});
```



Composition

Architecture

Model / Collection - View - Template - Router - Utilities

```
var items = new Items();  
var listView = new ListView({  
    collection: items  
});  
items.fetch();
```

Views

listen to DOM

events

Architecture

Model / Collection - View - Template - Router - Utilities

```
var ItemView = Backbone.View.extend({
  events: {
    "click .btn" : "onClick"
  },
  onClick: function() {
    alert("You just clicked");
  }
})
var view = new ItemView({el: $("#item") });

// User clicks on button. What happens?
```

Architecture

Model / Collection - View - Template - Router - Utilities

```
var ItemView = Backbone.View.extend({
  events: {
    "click .btn" : "onClick"
  },
  onClick: function() {
    alert("You just clicked");
  }
})
var view = new ItemView({el: $("#item") });

// User clicks on button. What happens?
```

Uses jQuery
behind the scenes

Architecture

Model / Collection - View - Template - Router - Utilities

View

- \$ (jQuery or Zepto)
- render
- remove
- make
- delegateEvents
- undelegateEvents
- extend
- constructor / initialize
- el
- \$el
- setElement
- attributes

Template (Underscore.js)

Compiles JavaScript templates into functions that can be evaluated for rendering.

- Mustache
- jQuery-tmpl

Architecture

Model / Collecction - View - **Template** - Router - Utilities

```
<script type="text/template" id="item-template">  
  <li>  
    <%= Name %>  
  </li>  
</script>
```

Architecture

Model / Collection - View - **Template** - Router - Utilities

```
var ItemView = Backbone.View.extend({  
  ...  
  template: _.template($("#item-template").html()),  
  ...  
  render: function () {  
    this.$el.html(this.template(this.model.toJSON()));  
    return this;  
  }  
  ...  
});
```

Router

- Maps urls to function.
- Enable history / bookmarking.



Architecture

Model / Collection - View - Template - Router - Utilities

```
var AppRouter = Backbone.Router.extend({
  routes: {
    '': "initialize"
  },
  initialize: function () {
    ...
  }
});
```


Architecture

Model / Collection - View - Template - Router - Utilities

```
window.AppRouter = Backbone.Router.extend({
  routes: {
    "": "loadInvoices",
    "/add": "addInvoice",
    "/show/:id": "showInvoice",
    "/edit/:id": "editInvoice"
  },
  loadInvoices: function () {
    ...
  },
  addInvoice: function () {
    ...
  },
  showInvoice: function (id) {
    ...
  },
  editInvoice: function (id) {
    ...
  }
});
```

Architecture

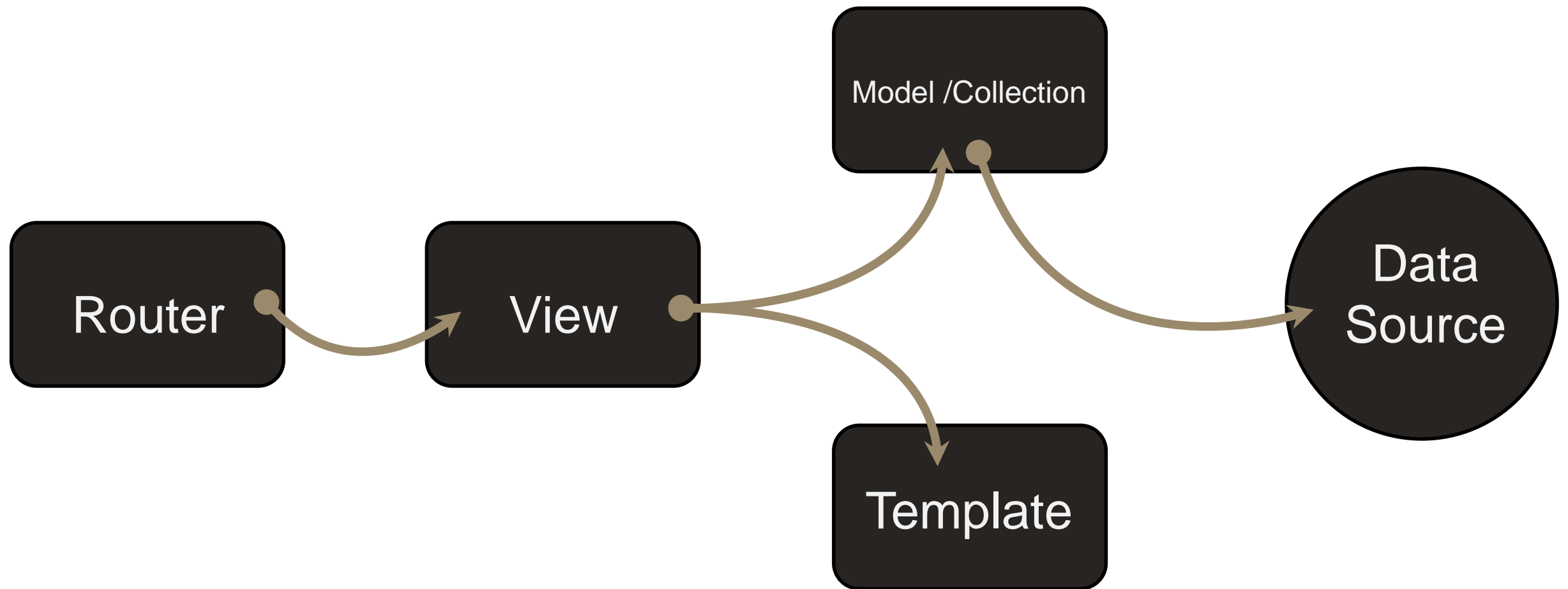
Model / Collecction - View - Template - Router - Utilities

Router

- extend
- routes
- constructor / initialize
- route
- navigate

Architecture

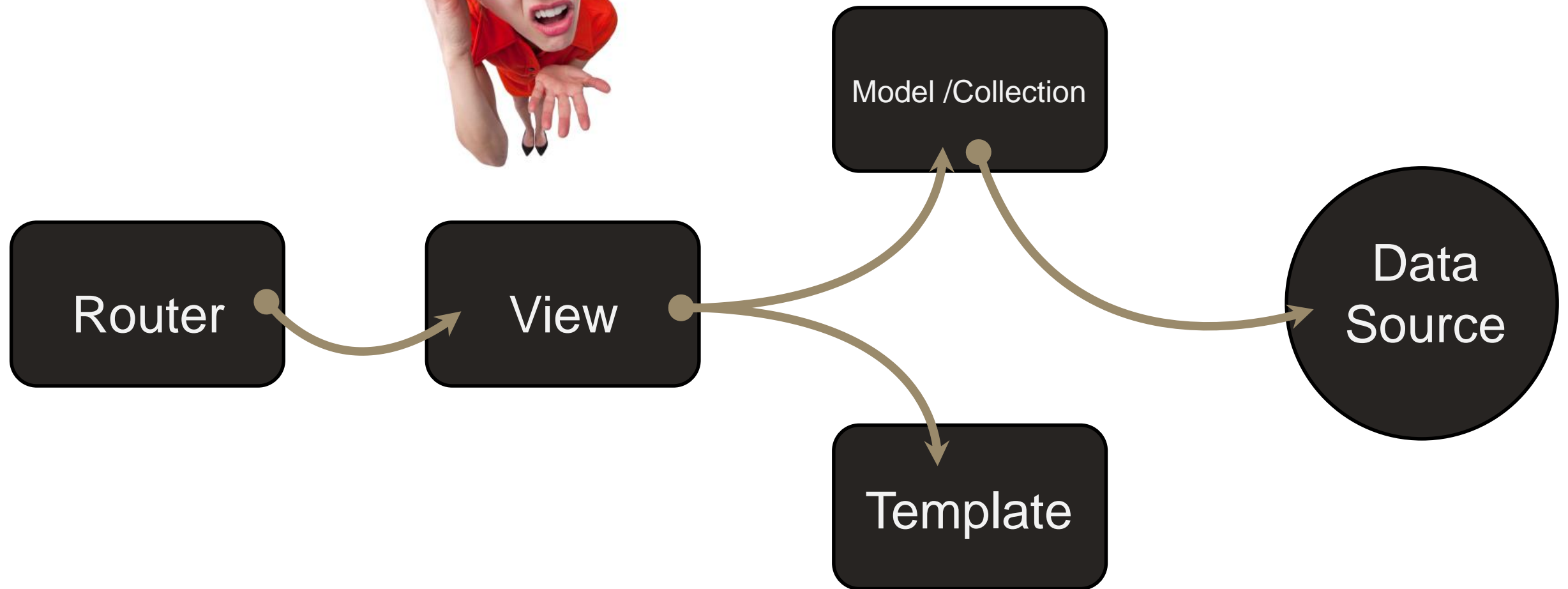
Model / Collection - View - Template - Router - Utilities



Architecture

Model / Collection - View - Template - Router - Utilities

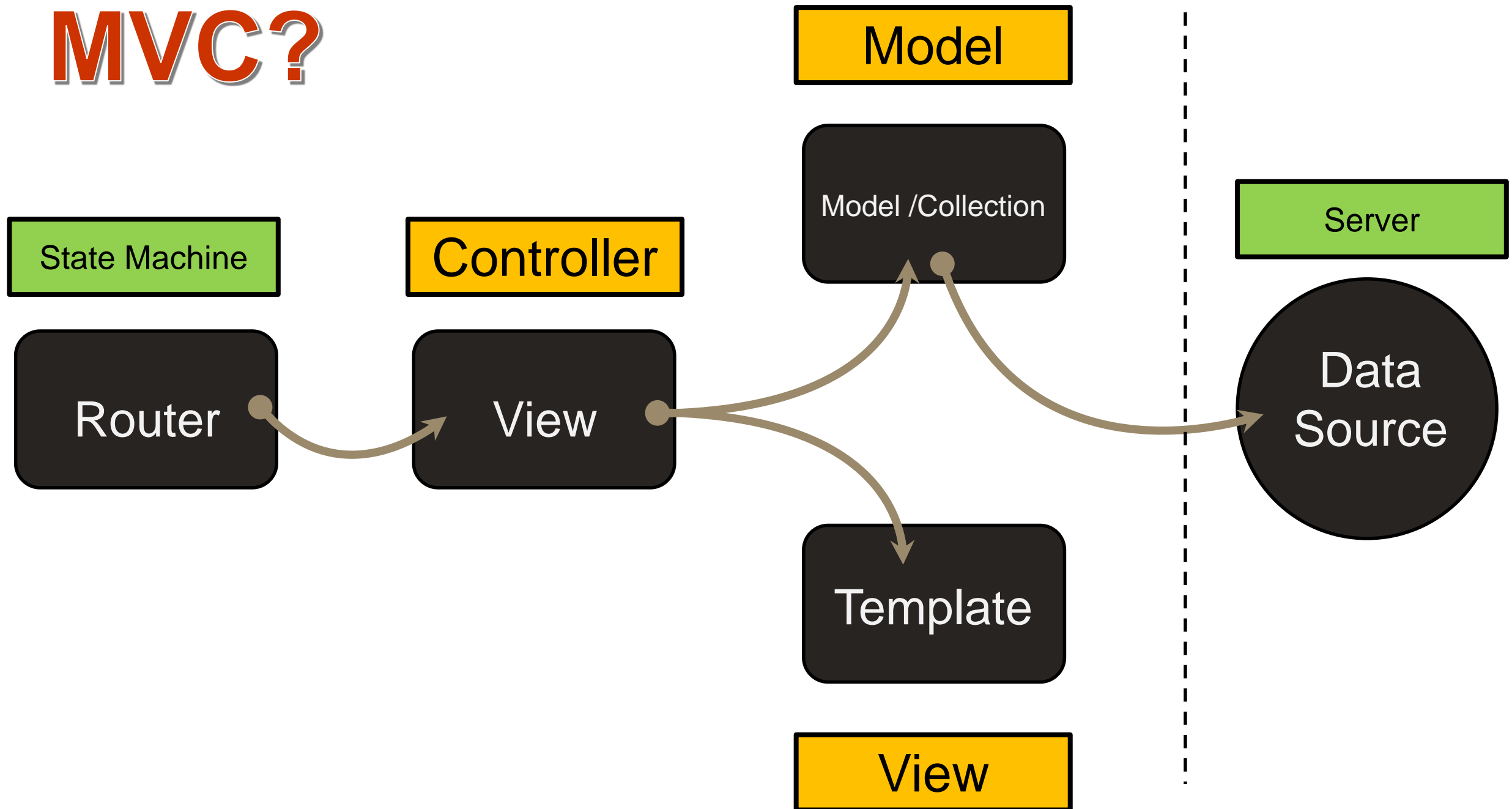
MVC?



Architecture

Model / Collection - View - Template - Router - Utilities

MVC?



Extras



Extras

Plugins

- Backbone-Nested
- Backbone.Memento
- Backbone.Validations
- Backbone.localStorage
- ...

<https://github.com/documentcloud/backbone/wiki/Extensions%2C-Plugins%2C-Resources>

Extras

Cheat Sheet

- <http://www.igloolab.com/downloads/backbone-cheatsheet.pdf>

Backbone.js

+

- Lightweight
- Powerful
- Code is clean (and maintainable)

-

- Too verbose (for small applications)

Questions ?



Credits:

www.igloolab.com

@iloveigloo