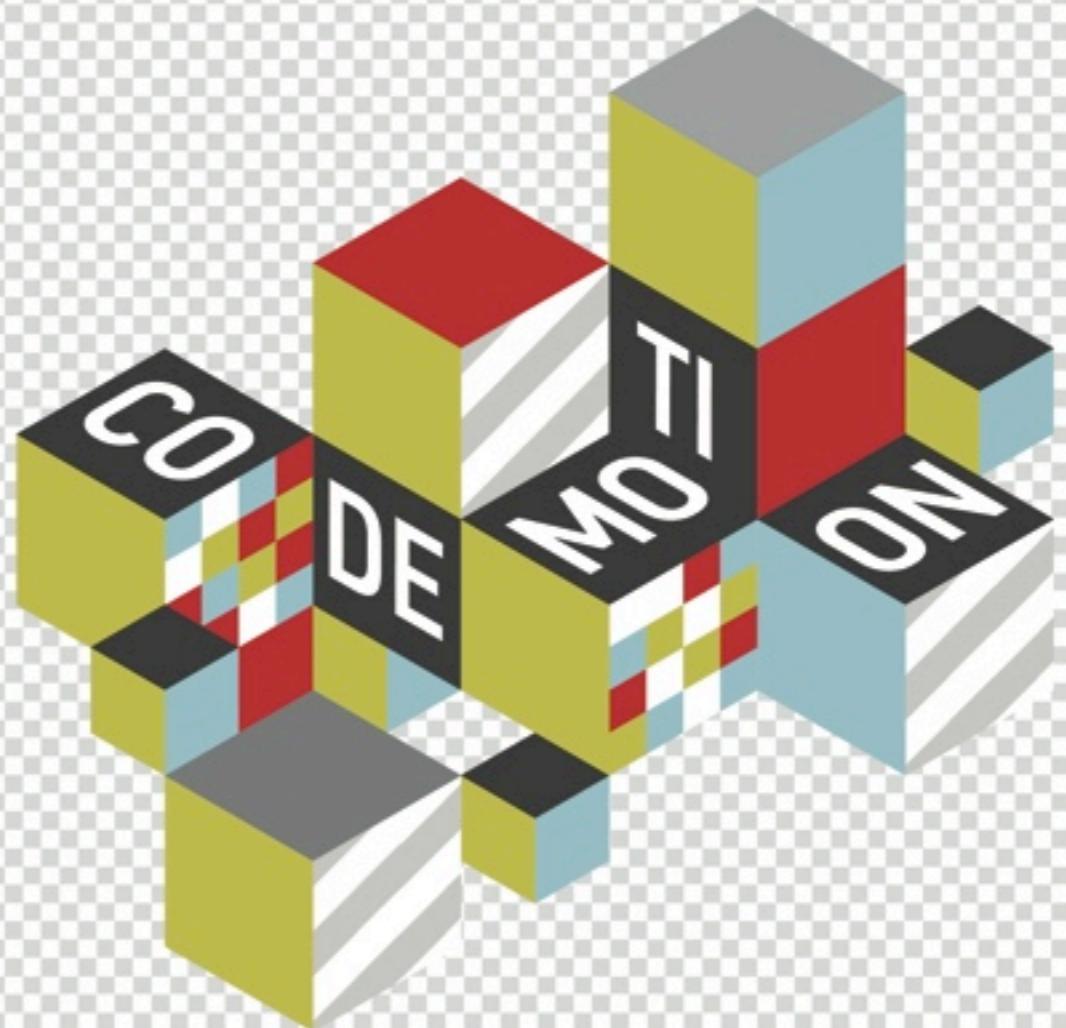




ROMA
23+24.03.2012

www.codemotion.it



EXCEPT WHERE OTHERWISE NOTED, THIS WORK IS LICENSED UNDER:
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Client-side MVC with Backbone.js



igloolab.com
@iloveigloo



michele.berto.li
@MicheleBertoli

Lago di Garda

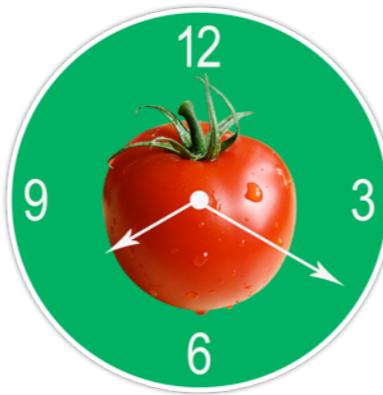




SHARP SQUARE
igloolab.com/sharpsquare



**jQuery
HTML5 UPLOADER**
igloolab.com/jquery-html5-uploader



POMODORO WEBAPP
pomodorowebapp.com



JOBBERONE
jobberone.com



KOALA
koooala.com



Agenda

Why Backbone.js 06:00 min

Architecture 14:00 min

Real World 05:00 min

Tips & Tricks 06:00 min

Extras 04:00 min

Questions 05:00 min

Why Backbone.js



Why Backbone.js

From server-side to client-side

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 do we need?

- Abstraction.
- Decoupling UI from Data.
- No more callbacks.

Why Backbone.js

So, what do we need? (More practically)

- A RESTful service based data layer.
- Events (to keep UI and data up-to-date).
- A template engine.
- A solid routing system.
- All the above things wrapped into a lightweight JavaScript framework.

Why Backbone.js

It exists and it's called:
Backbone.js



Architecture



Architecture



Jeremy
Ashkenas

 BACKBONE.JS

Oct 13th, 2010

CoffeeScript

UNDERSCORE.JS

DocumentCloud 

Architecture

<http://documentcloud.github.com/backbone>

<https://github.com/documentcloud/backbone>

@documentcloud

#documentcloud on IRC

<https://groups.google.com/forum/#!forum/backbonejs>

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:

- jQuery or Zepto
- Underscore.js
- Json2.js

Architecture

MVC

Architecture

~~MVC~~

Model / Collection

Architecture

~~MVC~~

Model / Collection
Template (View)

Architecture

~~MVC~~

Model / Collection

Template (View)

View (Controller)

Architecture

~~MVC~~

Model / Collection
Template (View)
View (Controller)
Router



Model

- Representing data (auto-generated).
- Handling persistence.
- Throws events.
- Reusable.



Architecture

Model / Collection - View - Template - Router - Utilities

Model

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

Architecture

Model / Collection - View - Template - Router - Utilities

```
var Item = Backbone.Model.extend({  
  idAttribute: "Id",  
  urlRoot: "/Items"  
});
```

Architecture

[Model / Collection](#) - [View](#) - [Template](#) - [Router](#) - [Utilities](#)

```
var item = new Item();
item.set({
  Name: "Igloo"
}); // trigger change
item.save(); // trigger sync
```

Architecture

[Model / Collection](#) - [View](#) - [Template](#) - [Router](#) - [Utilities](#)

Model

- 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.



Architecture

Model / Collection - View - Template - Router - Utilities

```
var Items = Backbone.Collection.extend({  
  model: Item,  
  url: "/Items"  
});
```

Architecture

[Model / Collection](#) - [View](#) - [Template](#) - [Router](#) - [Utilities](#)

```
var items = new Items();
items.fetch() // trigger reset
```

Architecture

Model / Collection - View - Template - Router - Utilities

```
items.comparator = function(item) {  
    return item.get("Name");  
};
```

Architecture

[Model / Collection](#) - [View](#) - [Template](#) - [Router](#) - [Utilities](#)

Collection

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

Architecture

[Model / Collection](#) - [View](#) - [Template](#) - [Router](#) - [Utilities](#)

Collection

- `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.

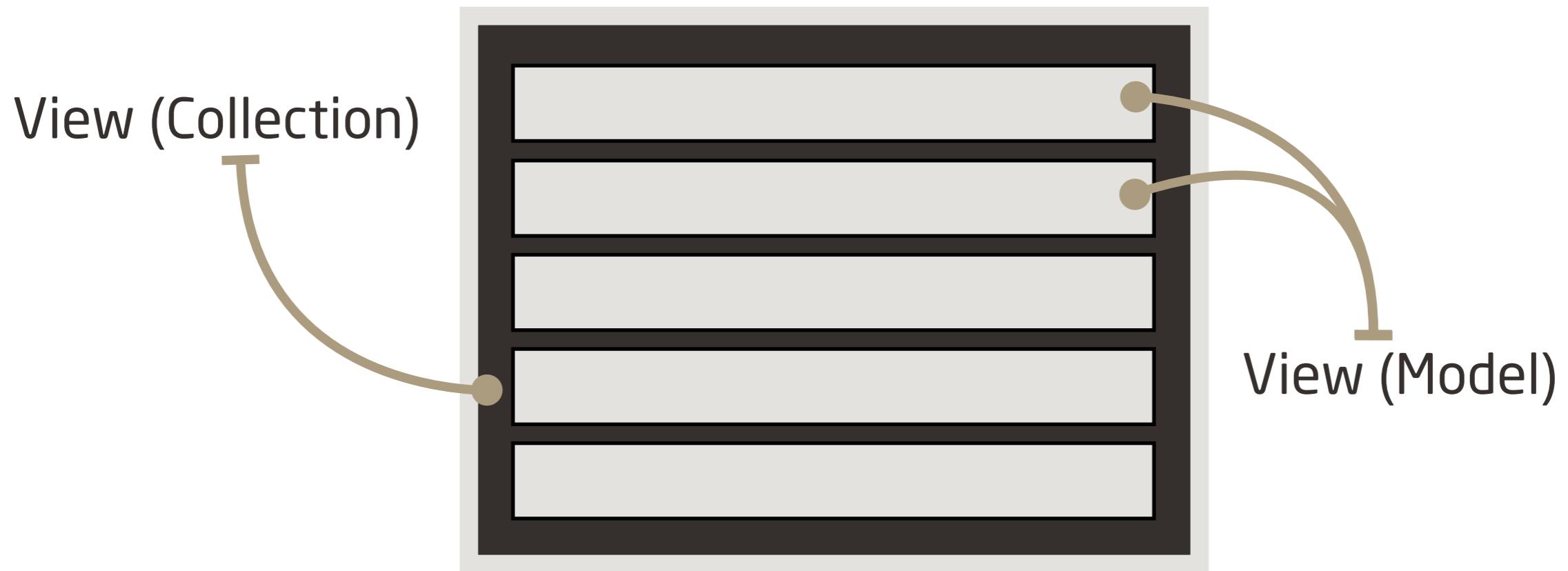
E
F P
T O Z
P D C
L P E D
P E C F D
E D F C Z P
F E L O P Z D
D E F F P O T E C



Architecture

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

View



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: $("ul"),
  initialize: function () {
    this.collection.bind("reset", this.render, this);
  },
  render: function () {
    this.collection.each(this.addItem, this);
    return this;
  },
  addItem: function (item) {
    var itemView = new ItemView({
      model: item
    });
    this.$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: $("ul"),
  initialize: function () {
    this.collection.bind("reset", this.render, this);
  },
  render: function () {
    this.collection.each(this.addItem, this);
    return this;
  },
  addItem: function (item) {
    var itemView = new ItemView({
      model: item
    });
    this.$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: $("ul"),
  initialize: function () {
    this.collection.bind("reset", this.render, this);
  },
  render: function () {
    this.collection.each(this.addItem, this);
    return this;
  },
  addItem: function (item) {
    var itemView = new ItemView({
      model: item
    });
    this.$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: $("ul"),
  initialize: function () {
    this.collection.bind("reset", this.render, this);
  },
  render: function () {
    this.collection.each(this.addItem, this);
    return this;
  },
  addItem: function (item) {
    var itemView = new ItemView({
      model: item
    });
    this.$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: $("ul"),
  initialize: function () {
    this.collection.bind("reset", this.render, this);
  },
  render: function () {
    this.collection.each(this.addItem, this);
    return this;
  },
  addItem: function (item) {
    var itemView = new ItemView({
      model: item
    });
    this.$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 items = new Items();
var listView = new ListView({
  collection: items
});
items.fetch();
```

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

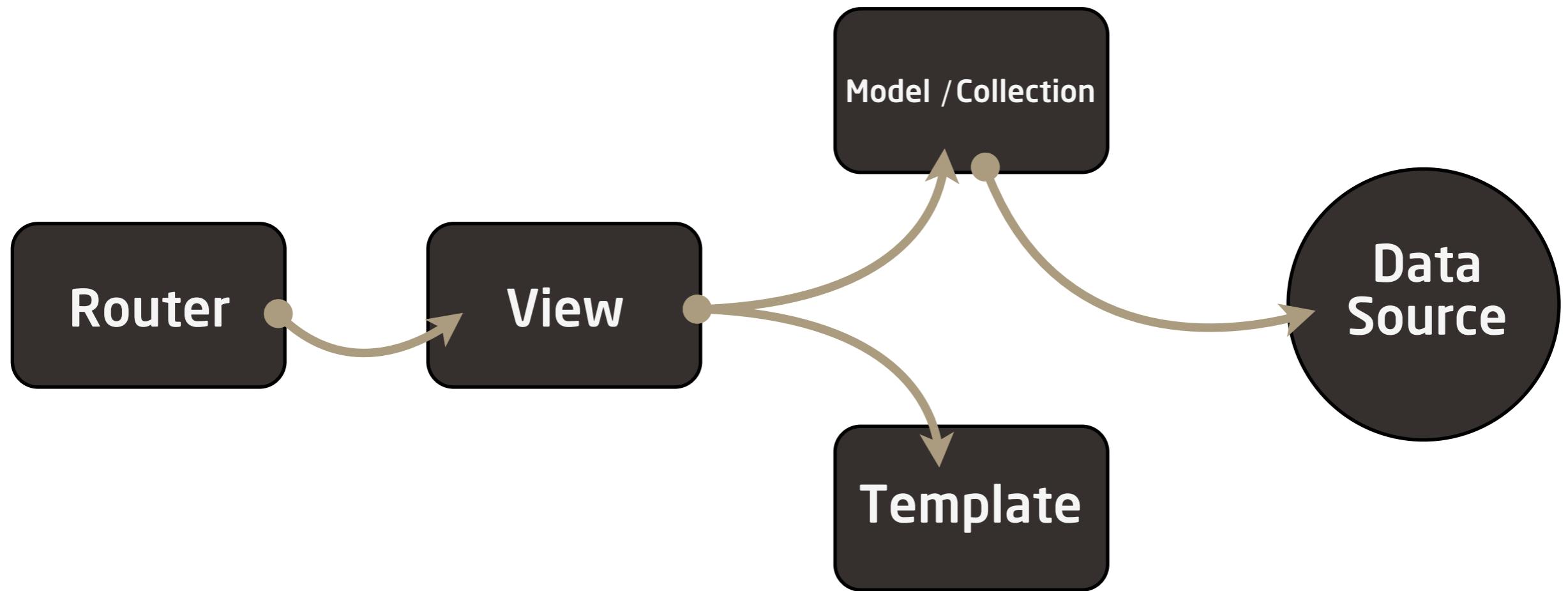
Router

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

Utilities

- History
- Sync
- Utility

Architecture



Real World



Real World

Polar management

- Smart invoicing web application

Technologies

- .NET RESTful Web Services / MS SQL
- Backbone.js

Real World

Polar management

Polar Management

Benvenuto igloo! [Esci]

Vendita > Fatture COSTI 0 RICAVI 100 TOTALE 100

Aggiungi Genera Flusso Ri.Ba. ←

Vendita Acquisto

Offerte Conferme d'ordine DDT Fatture

1 24/3/2012 Igloo Snc
Via C. Alberto, 31/A Moniga del Garda (BS)

The screenshot shows the Polar Management software interface. At the top, there's a red header bar with the title 'Polar Management' and a user profile icon. Below it, the main area has a dark background. On the left, there's a sidebar with icons for adding new items and generating flowcharts. In the center, the title 'Vendita > Fatture' is displayed above a summary table with columns for COSTI (0), RICAVI (100), and TOTALE (100). Below the summary, there are buttons for 'Aggiungi' and 'Genera Flusso Ri.Ba.' and a back arrow. A large white box contains the word 'Vendita' and 'Acquisto'. Underneath, there are tabs for 'Offerte', 'Conferme d'ordine', 'DDT', and 'Fatture', with 'Fatture' being the active tab. At the bottom of the white box, there's a row of icons and some text: '1' next to a document icon, '24/3/2012', and the company details 'Igloo Snc' and 'Via C. Alberto, 31/A Moniga del Garda (BS)'.

Real World

Polar Management

Benvenuto igloo! [Esci]

Vendita > Fatture

COSTI 0 RICAVI 100 TOTALE 100

✓ Salva ⏪ Annulla ⏪

Codice	Descrizione	U.M.	Q.tà	P.zzo Unità	Importo Netto
1	Soggiorno a Roma		1	100.00	100.00

Ragione Sociale: Igloo Snc

Riferimento numero DDT: 0

Numero: 1

Riferimento data DDT:

Data: 24/3/2012

Aggiungi

Pagamento: Bonifico immediato

Totale Merce: 100.00

Trasporto €: 0

Sconto: 0.00

Totale Imponibile: 100.00

Note:

DataValore

Aggiungi Data Scadenza Pagamento Ri.Ba

Real World

Polar Management

Benvenuto igloo! [Esci]

Vendita > Fatture

COSTI 0 RICAVI 100 TOTALE 100

✓ Salva ⏪ Annulla ⏪

Ragione Sociale:	Igloo Snc	Riferimento numero DDT :	0		
Numero:	1	Riferimento data DDT :			
Data:	24/3/2012				
Codice	Descrizione	U.M.	Q.tà	P.zzo Unità	Importo Netto
1	Soggiorno a Roma		1	100.00	100.00

+ Aggiungi

Pagamento:

Bonifico immediato

Total Merce: 100.00

DataValore

+ Aggiungi Data Scadenza Pagamento Ri.Ba

Trasporto €: 0

Sconto: 0.00

Total Imponibile: 100.00

Note:

Real World

Polar management



localhost:8080/documenti/vendita/fatture#/modifica/1

```
window.AppRouter = Backbone.Router.extend({  
  routes: {  
    ...  
    "/modifica/:id": "editInvoice"  
  },  
  editInvoice: function (id) {  
    ...  
  }  
});
```

Real World

Polar management



localhost:8080/documenti/vendita/fatture#/modifica/1

```
window.AppRouter = Backbone.Router.extend({  
  routes: {  
    ...  
    "/modifica/:id": "editInvoice"  
  },  
  editInvoice: function (id) {  
    ...  
  }  
});
```

Real World

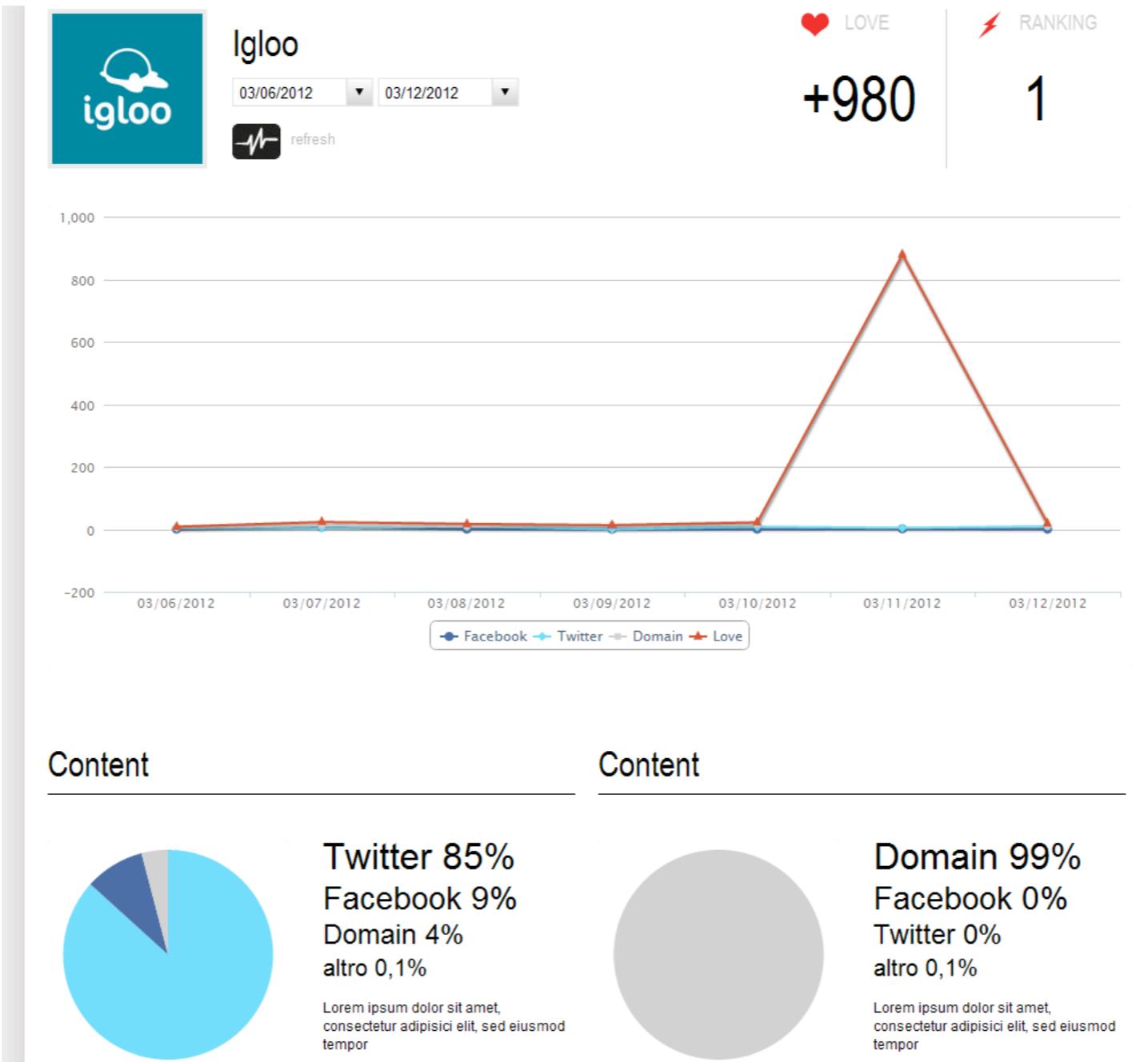
Koala

- Social media analytics

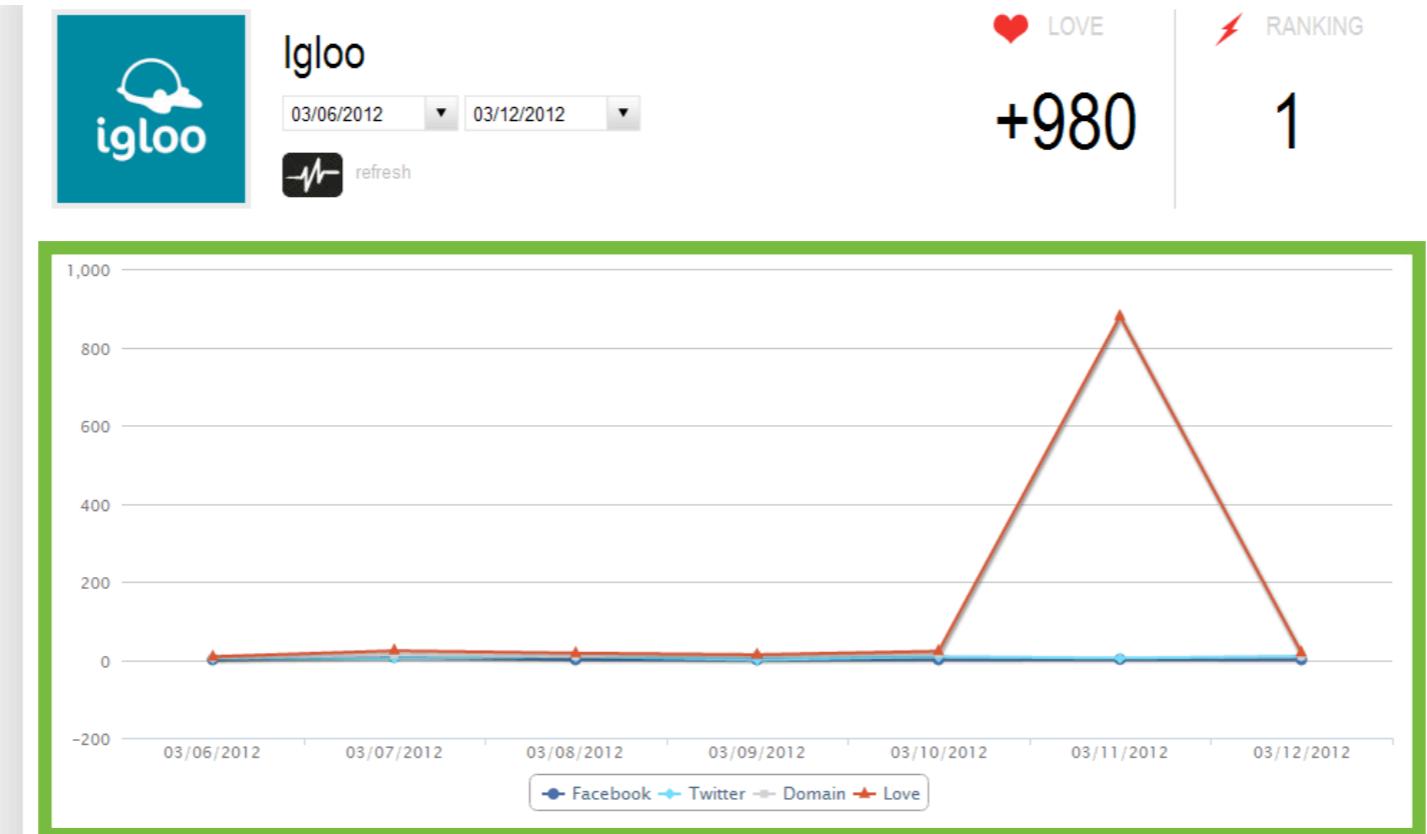
Technologies

- NET RESTful Web Services / RavenDB
- Backbone.js
- Highcharts.js

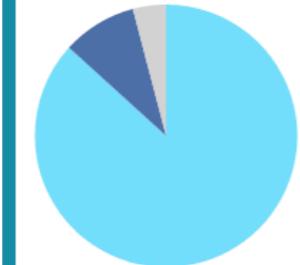
Real World



Real World



Content



Twitter 85%
Facebook 9%
Domain 4%
altro 0,1%

Lorem ipsum dolor sit amet,
consectetur adipisici elit, sed eiusmod
tempor

Content



Domain 99%
Facebook 0%
Twitter 0%
altro 0,1%

Lorem ipsum dolor sit amet,
consectetur adipisici elit, sed eiusmod
tempor

Real World

Koala

```
window.Logs = Backbone.Collection.extend({  
  model: Log,  
  url: "/Data",  
  comparator: function (log) {  
    return log.get("Date");  
  },  
  sum: function (field) {  
    return this.reduce(function (memo, log) {  
      return memo + log.get(field);  
    }, 0);  
  }  
});
```

Real World

Koala

```
window.Logs = Backbone.Collection.extend({  
  model: Log,  
  url: "/Data",  
  comparator: function (log) {  
    return log.get("Date");  
  },  
  sum: function (field) {  
    return this.reduce(function (memo, log) {  
      return memo + log.get(field);  
    }, 0);  
  }  
});
```

Real World

Koala

```
window.Logs = Backbone.Collection.extend({  
  model: Log,  
  url: "/Data",  
  comparator: function (log) {  
    return log.get("Date");  
  },  
  sum: function (field) {  
    return this.reduce(function (memo, log) {  
      return memo + log.get(field);  
    }, 0);  
  }  
});
```

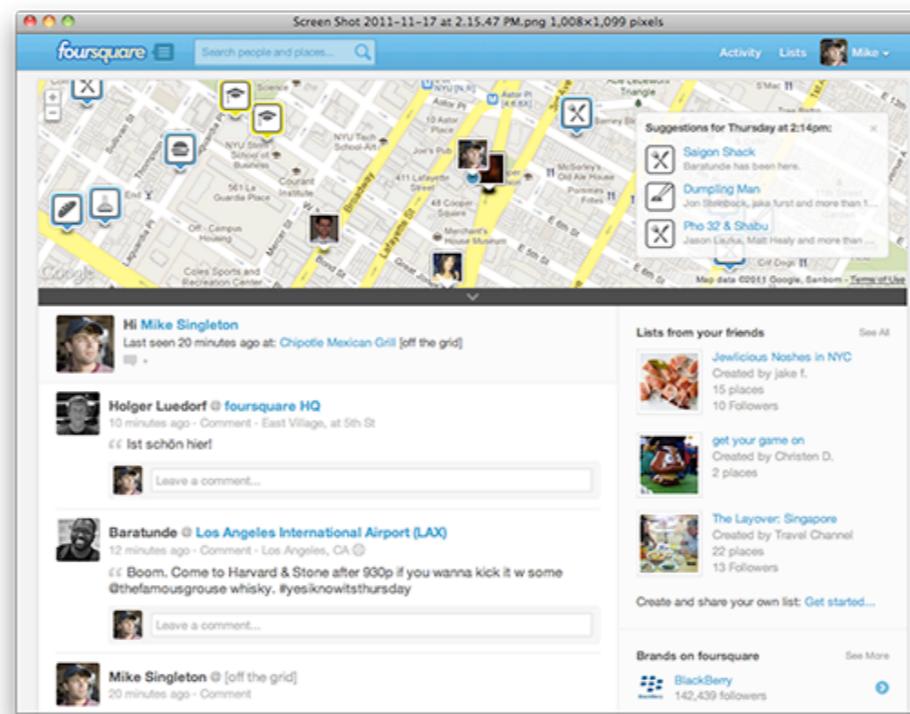
Real World

Linkedin



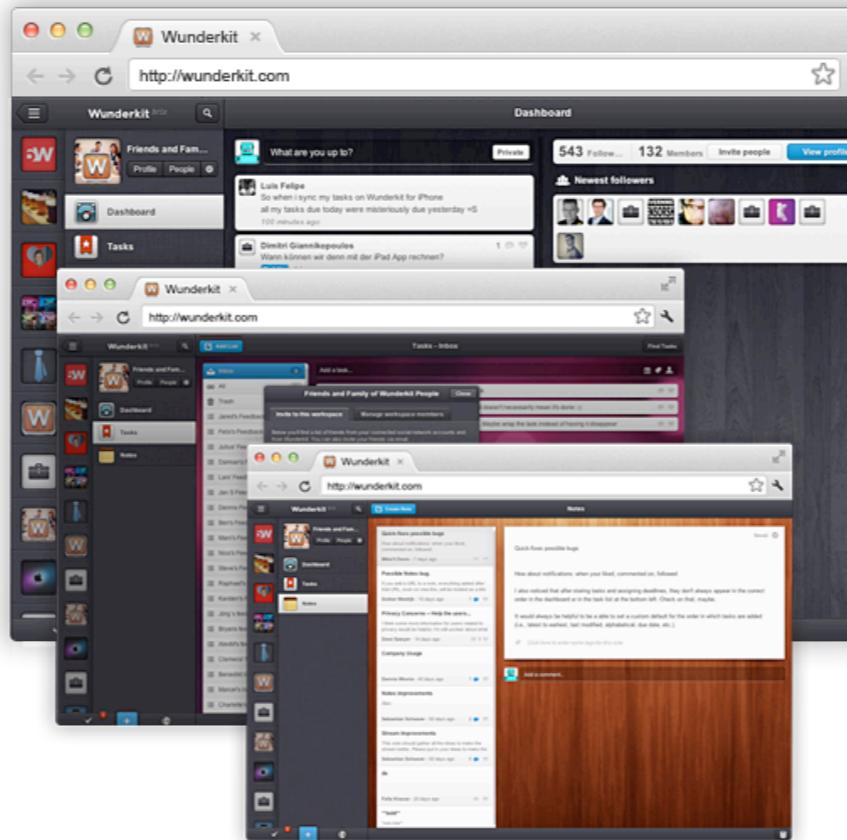
Real World

FourSquare



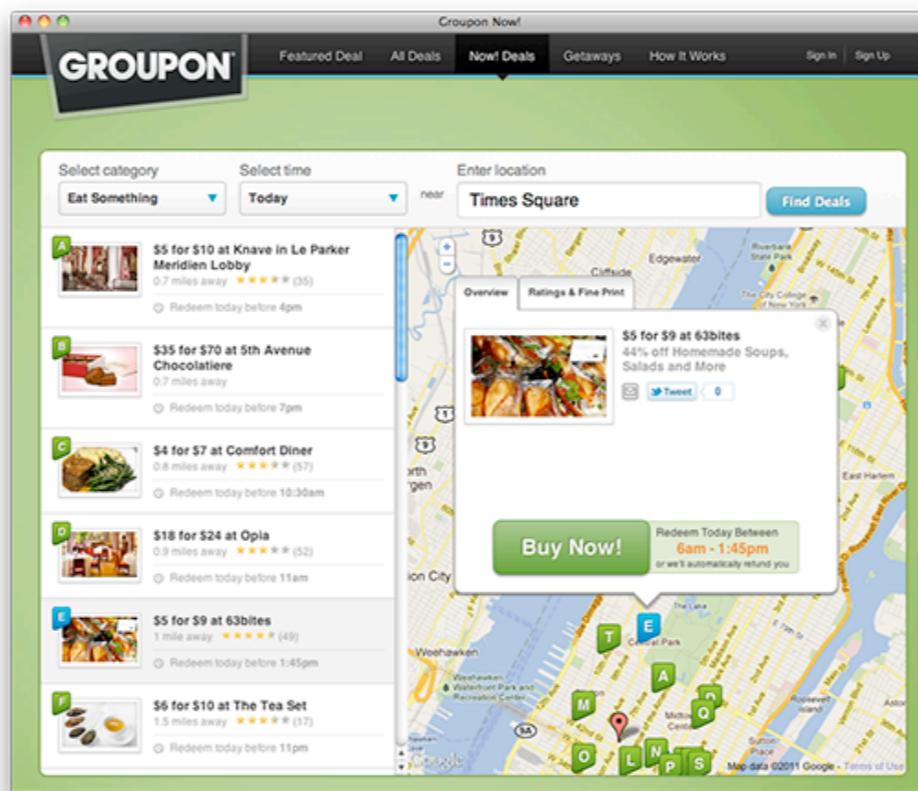
Real World

WunderKit



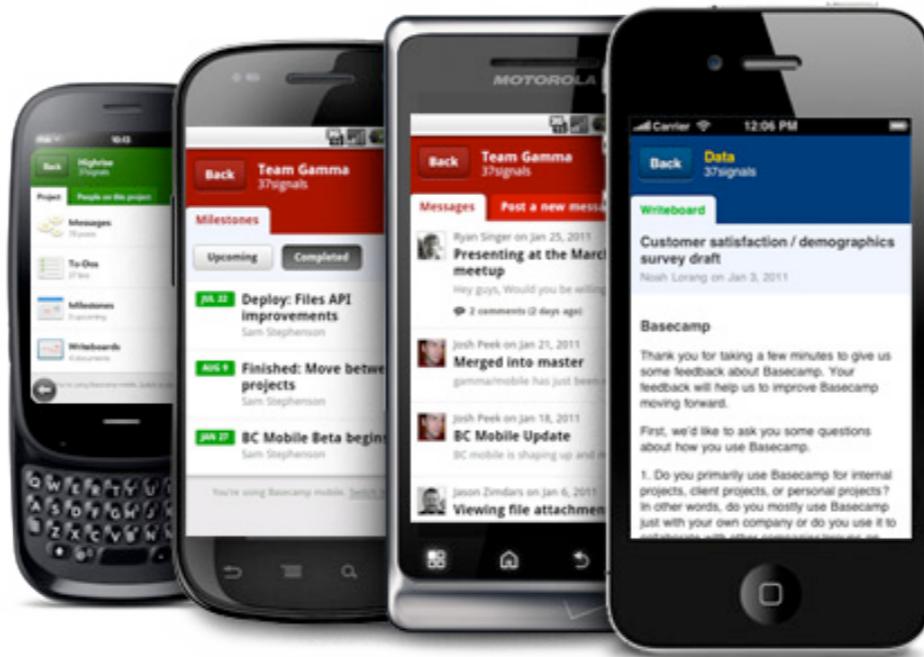
Real World

Groupon



Real World

Basecamp



Tips & Tricks



Tips & Tricks

idAttribute

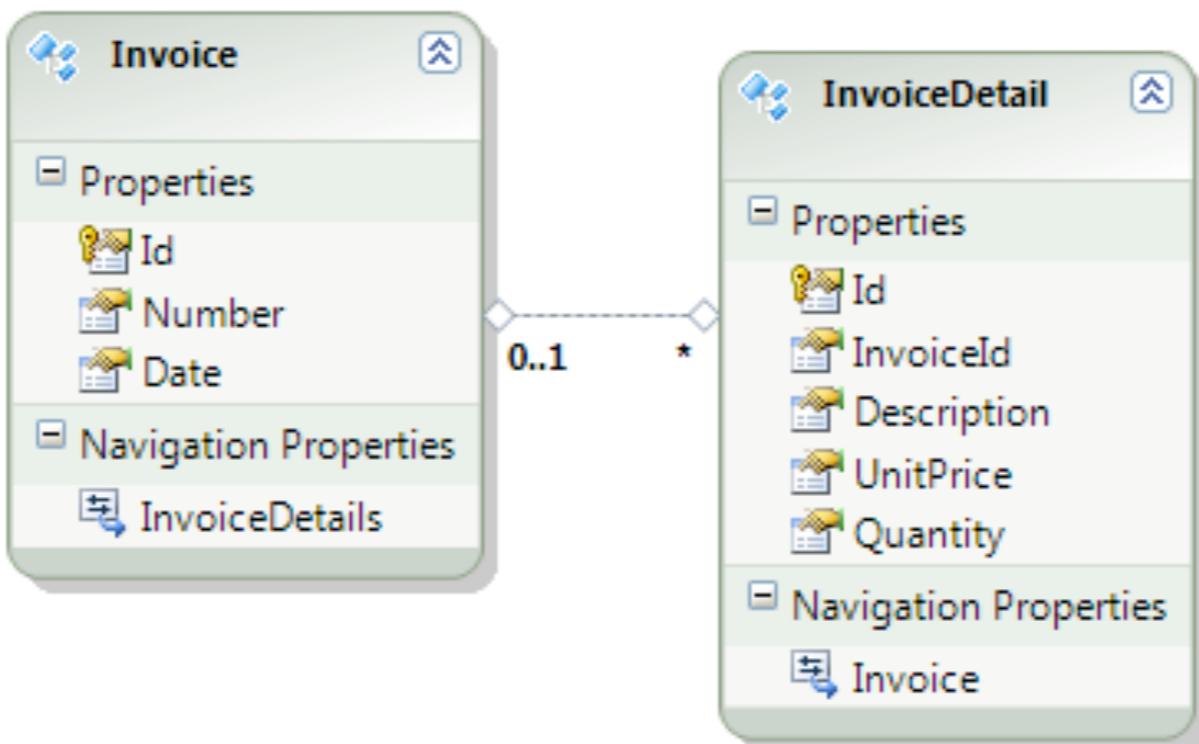
```
idAttribute: "id"
```

```
// CouchDB  
idAttribute: "_id"
```

```
// .NET  
idAttribute: "Id"
```

Tips & Tricks

Related Models



Tips & Tricks

Related Models

```
var Invoice = Backbone.Model.extend({  
  idAttribute: "Id"  
});  
  
var Invoices = Backbone.Collection.extend({  
  model: Invoice,  
  url: "/Invoices"  
});
```

Tips & Tricks

Related Models

```
▼ child
  ► _byCid: Object
  ► _byId: Object
  length: 1
  ▼ models: Array[1]
    ▼ 0: child
      ► _callbacks: Object
      ► _escapedAttributes: Object
      ► _previousAttributes: Object
      _setting: false
      ▼ attributes: Object
        Date: "/Date(1325372400000)/"
        Id: 47
        ► InvoiceDetails: Array[1]
        Number: 1
        ► __proto__: Object
        cid: "c0"
      ► collection: child
      id: 47
      ► __proto__: ctor
      length: 1
      ► __proto__: Array[0]
    ► __proto__: ctor
```

Tips & Tricks

Related Models

```
▼ child
  ► _byCid: Object
  ► _byId: Object
  length: 1
  ▼ models: Array[1]
    ▼ 0: child
      ► _callbacks: Object
      ► _escapedAttributes: Object
      ► _previousAttributes: Object
      _setting: false
      ▼ attributes: Object
        Date: "/Date(1325372400000)/"
        Id: 47
        ► InvoiceDetails: Array[1]
          Number: 1
        ► __proto__: Object
        cid: "c0"
      ► collection: child
        id: 47
      ► __proto__: ctor
      length: 1
      ► __proto__: Array[0]
    ► __proto__: ctor
```

Tips & Tricks

Related Models

```
var InvoiceDetail = Backbone.Model.extend({  
  idAttribute: "Id"  
});
```

```
var InvoiceDetails = Backbone.Collection.extend({  
  model: InvoiceDetail,  
  url: "/InvoiceDetails"  
});
```

Tips & Tricks

Related Models

```
var Invoice = Backbone.Model.extend({
  idAttribute: "Id",
  initialize: function () {
    this.setInvoiceDetails();
  },
  setInvoiceDetails: function () {
    this.set({
      InvoiceDetails: new InvoiceDetails(this.get("InvoiceDetails"))
    });
  }
});
```

Tips & Tricks

Related Models

```
▼ child
  ► _byCid: Object
  ► _byId: Object
  length: 1
  ▼ models: Array[1]
    ▼ 0: child
      ► _callbacks: Object
      _changing: false
      ► _escapedAttributes: Object
      _moreChanges: false
      ► _previousAttributes: Object
      _setting: false
      ▼ attributes: Object
        Date: "/Date(1325372400000)/"
        Id: 47
      ► InvoiceDetails: child
        Number: 1
      ► __proto__: Object
      cid: "c0"
      ► collection: child
      id: 47
      ► __proto__: ctor
      length: 1
      ► __proto__: Array[0]
      ► __proto__: ctor
```

Tips & Tricks

Related Models

```
▼ child
  ► _byCid: Object
  ► _byId: Object
  length: 1
  ▼ models: Array[1]
    ▼ 0: child
      ► _callbacks: Object
      _changing: false
      ► _escapedAttributes: Object
      _moreChanges: false
      ► _previousAttributes: Object
      _setting: false
      ▼ attributes: Object
        Date: "/Date(1325372400000)/"
        Id: 47
        ► InvoiceDetails: child
          Number: 1
        ► __proto__: Object
        cid: "c0"
        ► collection: child
        id: 47
        ► __proto__: ctor
        length: 1
        ► __proto__: Array[0]
        ► __proto__: ctor
```

Tips & Tricks

Related Models

```
var Invoice = Backbone.Model.extend({
  idAttribute: "Id",
  initialize: function () {
    this.setInvoiceDetails();
    this.bind("sync", this.setInvoiceDetails);
  },
  setInvoiceDetails: function () {
    this.set({
      InvoiceDetails: new InvoiceDetails(this.get("InvoiceDetails"))
    });
  }
});
```

Extras



Extras

Plugins

- Backbone-Nested
- Backbone.Memento
- Backbone.Validations
- Backbone.localStorage
- backbone.couchdb.js
- ...

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

Extras

- <http://sproutcore.com> (Apple/iCloud)
- <http://knockoutjs.com>
- <http://emberjs.com>
- <http://batmanjs.org> (Shopify)

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 ?



Grazie
www.igloolab.com
@iloveigloo