



Put some Backbone.js
in your app

Sidney Maestre

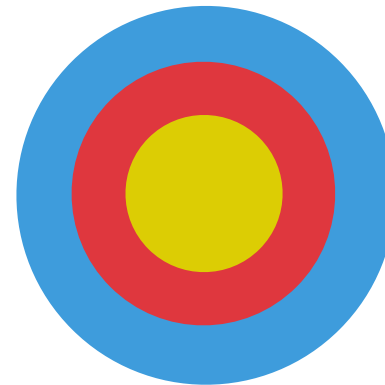
Platform Evangelist



@SidneyAllen









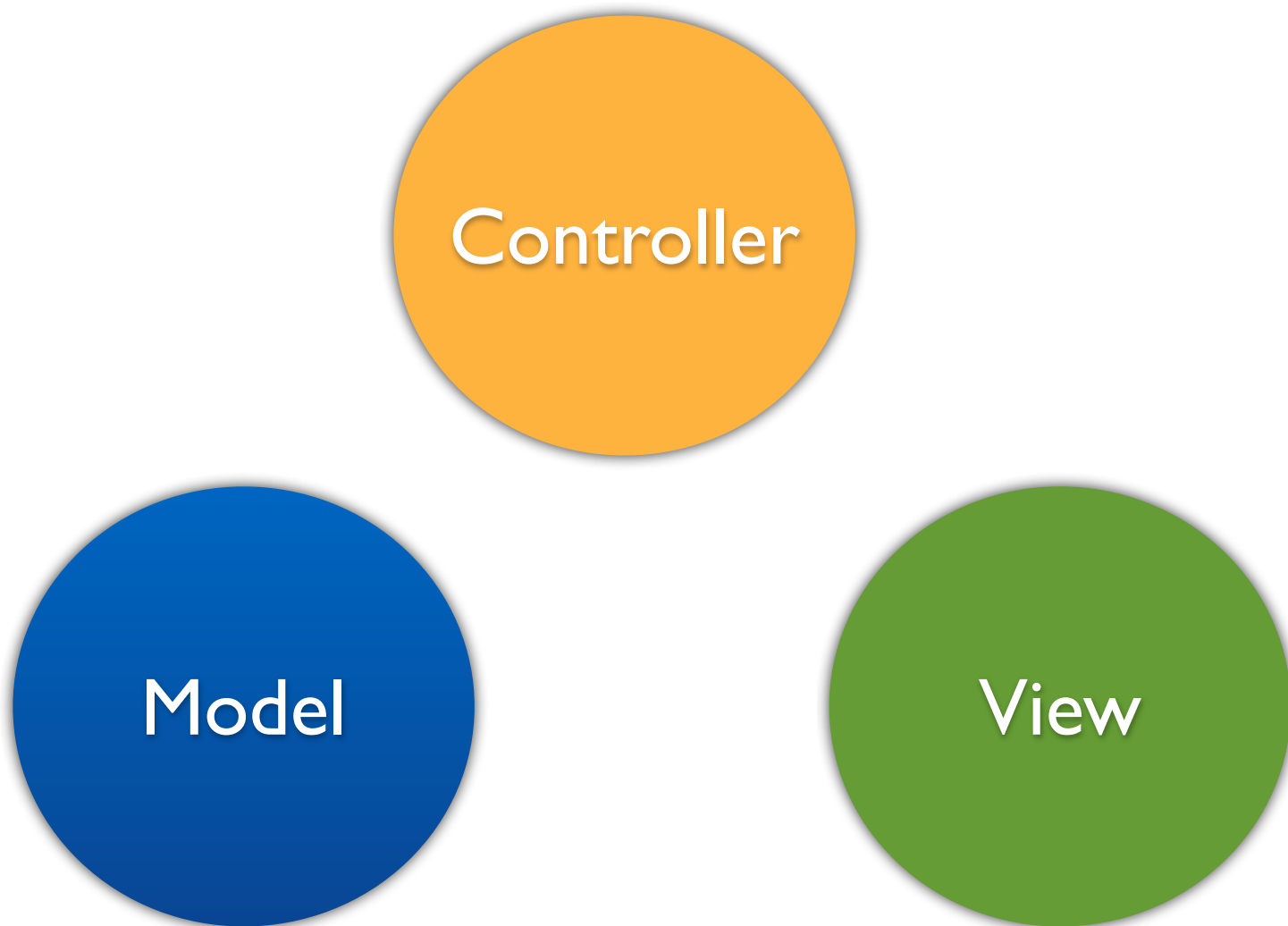
WHAT

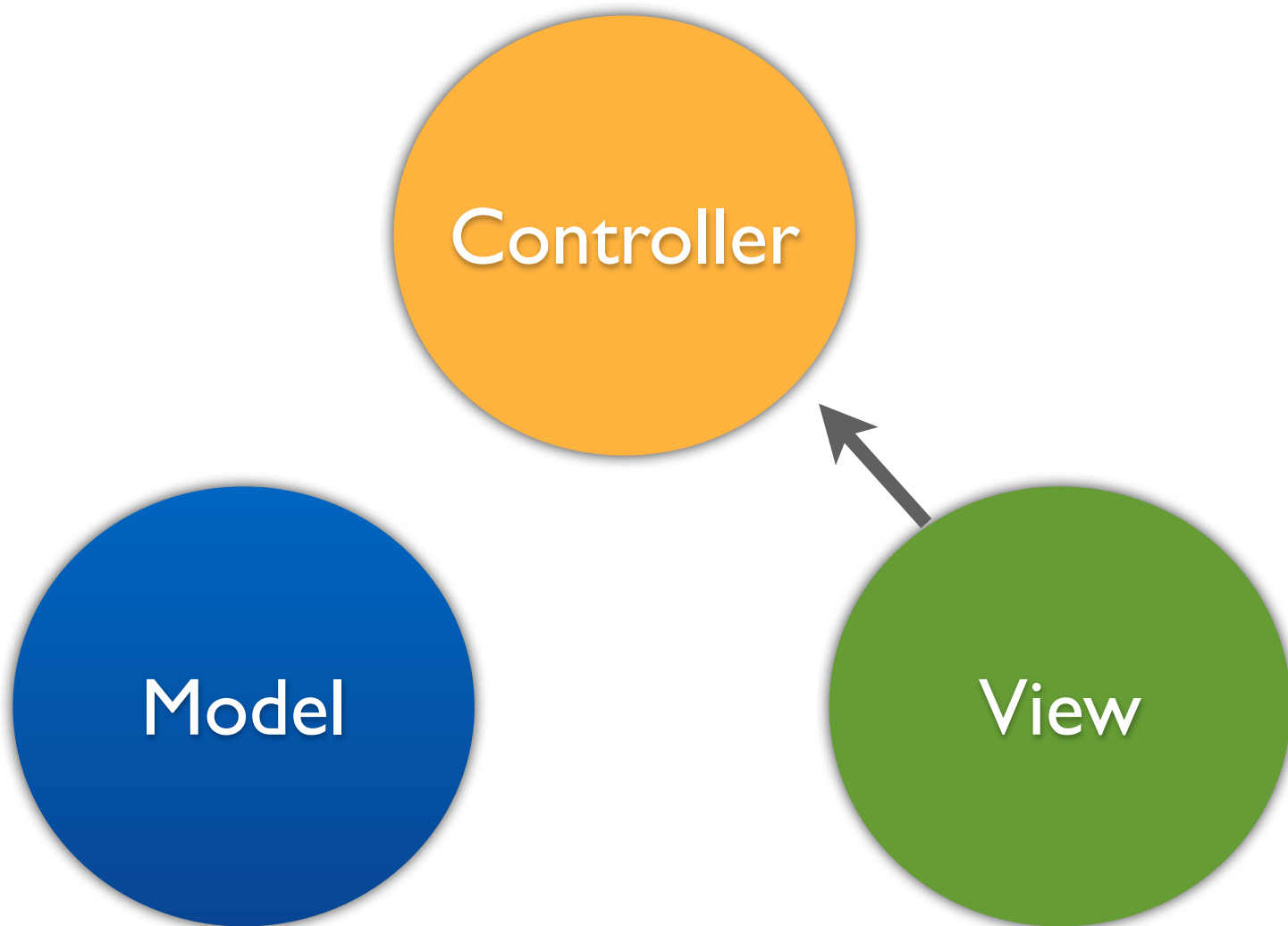


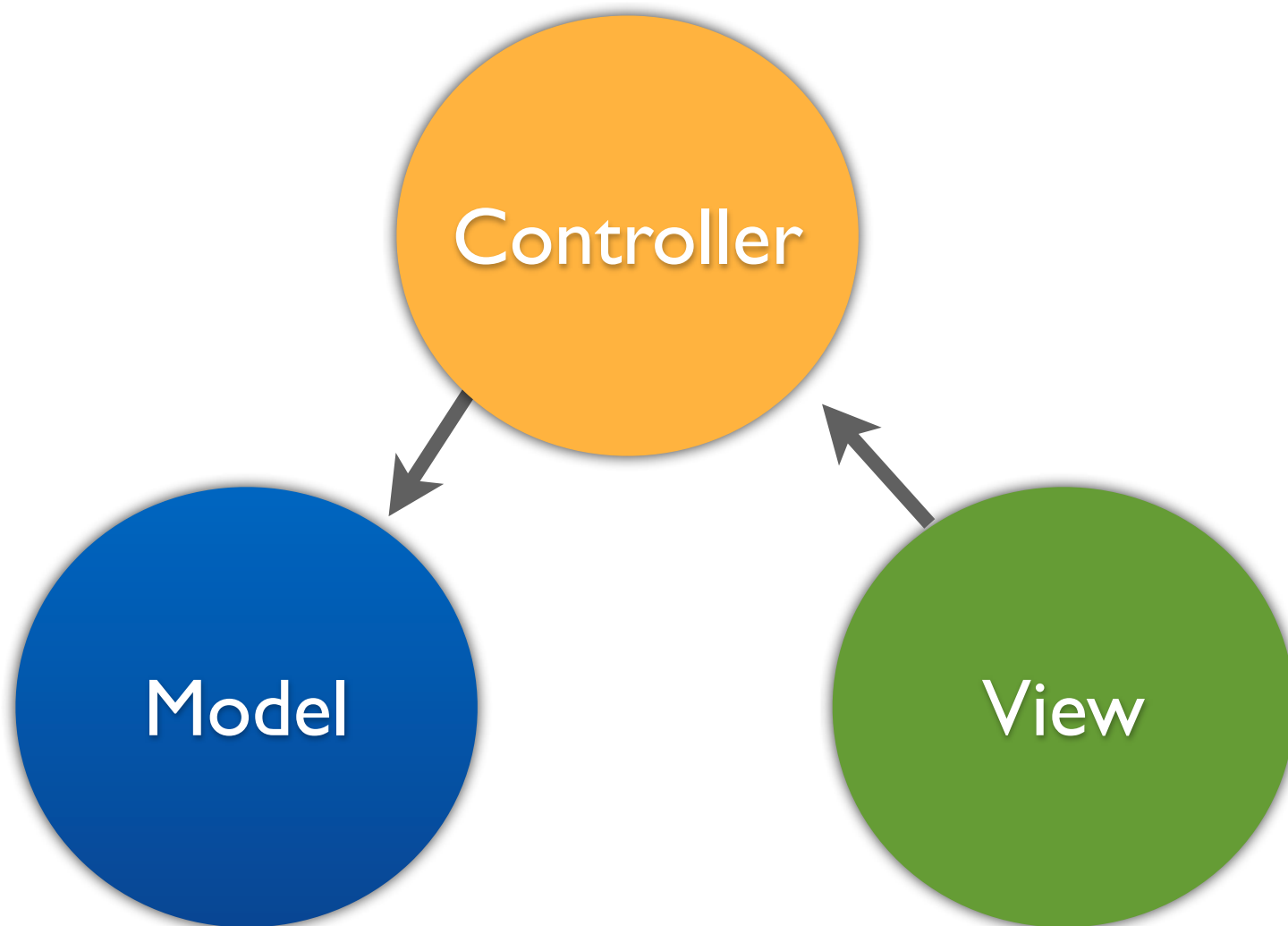
WHY

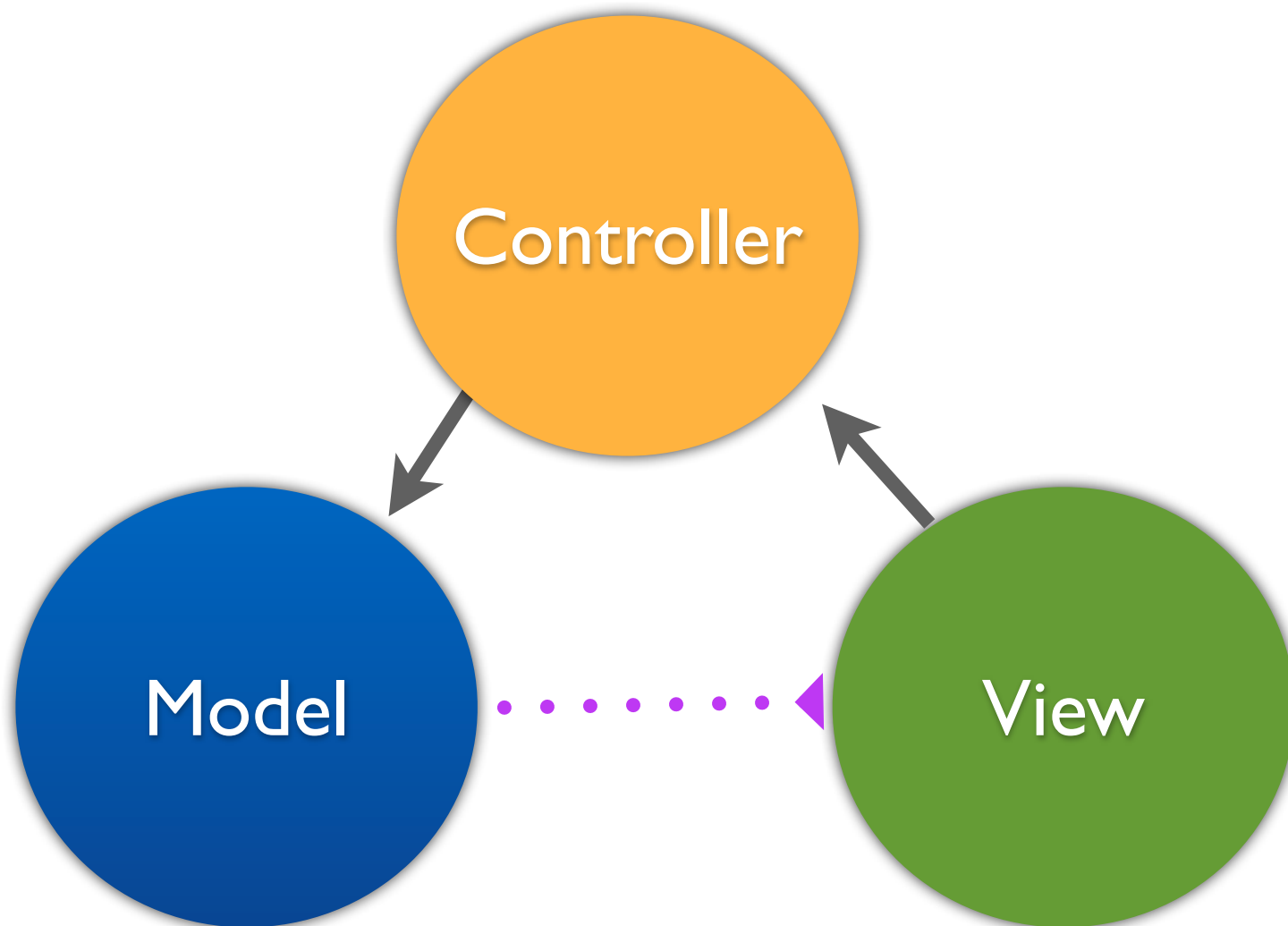
WHY











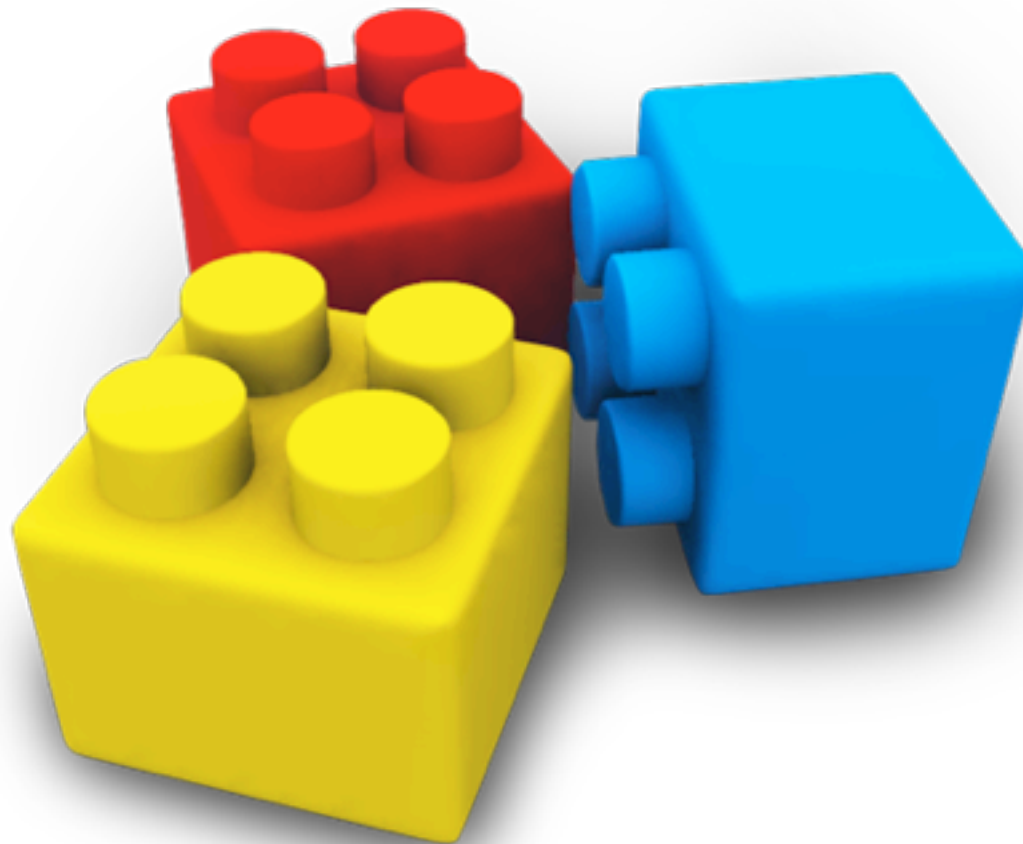
UNDERSCORE.JS



UNDERSCORE.JS



MODULE PATTERN

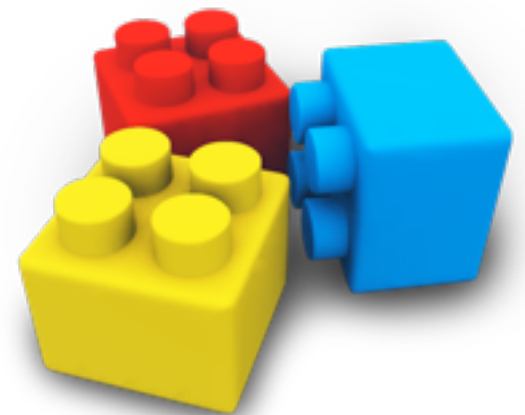


HANDS ON

00-module

MODULE PATTERN

```
var app = (function($){  
    var foo = "bar";  
  
    return {  
        name : 'module pattern',  
        set: function(name) {  
            foo = name;  
        },  
        get: function() {  
            return foo;  
        }  
    };  
})(jQuery));
```



MODULE PATTERN

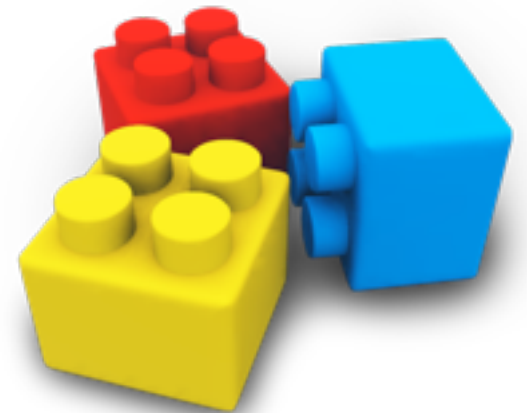
```
app.foo
```

```
app.name
```

```
app.get()
```

```
app.set("StackMob")
```

```
app.get()
```



MODULE PATTERN

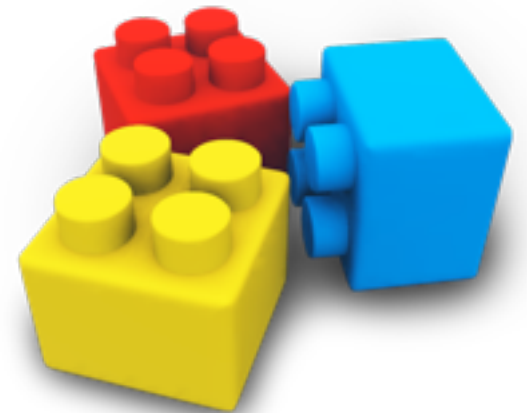
```
app.foo
undefined

app.name
"module pattern"

app.get()
"bar"

app.set("StackMob")
undefined

app.get()
"StackMob"
```



MODEL

MODEL



HANDS ON

01-model

MODEL

```
Wine = Backbone.Model.extend();
```



HANDS ON

```
firstWine = new Wine({name : 'Clos Pegase', year : '2008'})
```

```
firstWine.get('name');
```

```
firstWine.set('name', 'Cakebread');
```

```
firstWine.toJSON()
```



COLLECTION



HANDS ON

02-collection

COLLECTION

```
Wine = Backbone.Model.extend();  
  
Wines = Backbone.Collection.extend({  
  model: Wine  
});
```



HANDS ON

```
wines = new Wines()

wines.toJSON()

w = new Wine({name : 'Clos Pegase'})

wines.add(w)

wines.toJSON()
```



COLLECTION

```
Wine = Backbone.Model.extend();

Wines = Backbone.Collection.extend({
  model: Wine
});

wines = new Wines([
  {winery : "Robert Mondovi"},
  {winery : "CakeBread"}
]);
```



COLLECTION

```
Wine = Backbone.Model.extend();

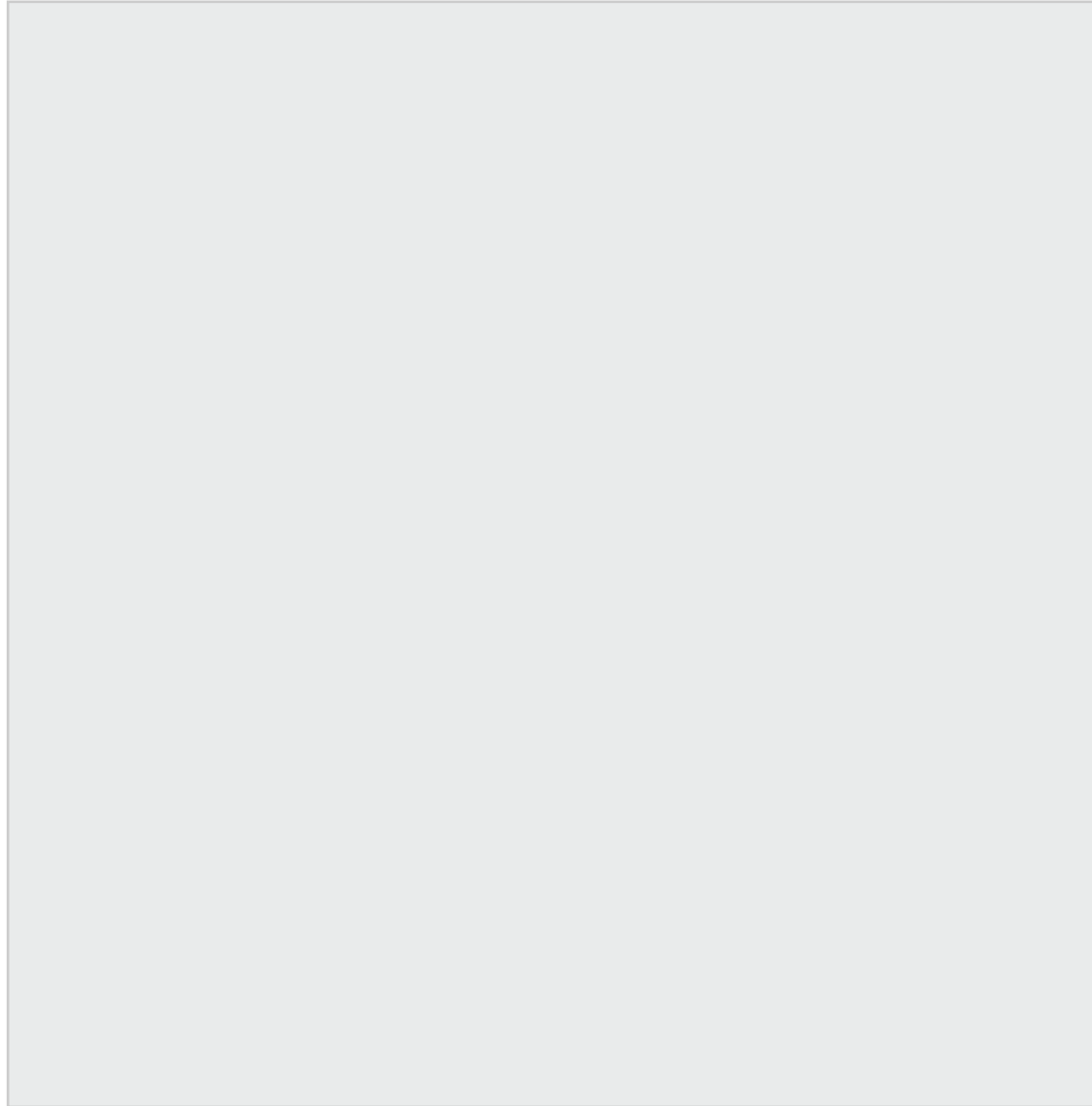
Wines = Backbone.Collection.extend({
  model: Wine,
});

wines = new Wines([
  {winery : "Robert Mondovi"},
  {winery : "CakeBread"}
]);

wines.each( function(model) {
  console.log(model.toJSON());
});
```



VIEW

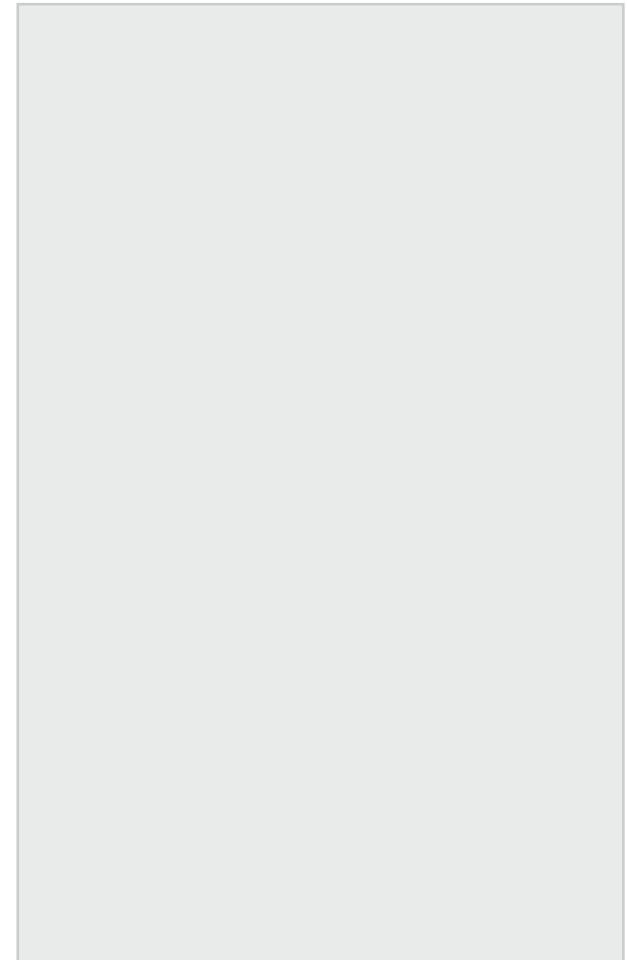


HANDS ON

03-home-view

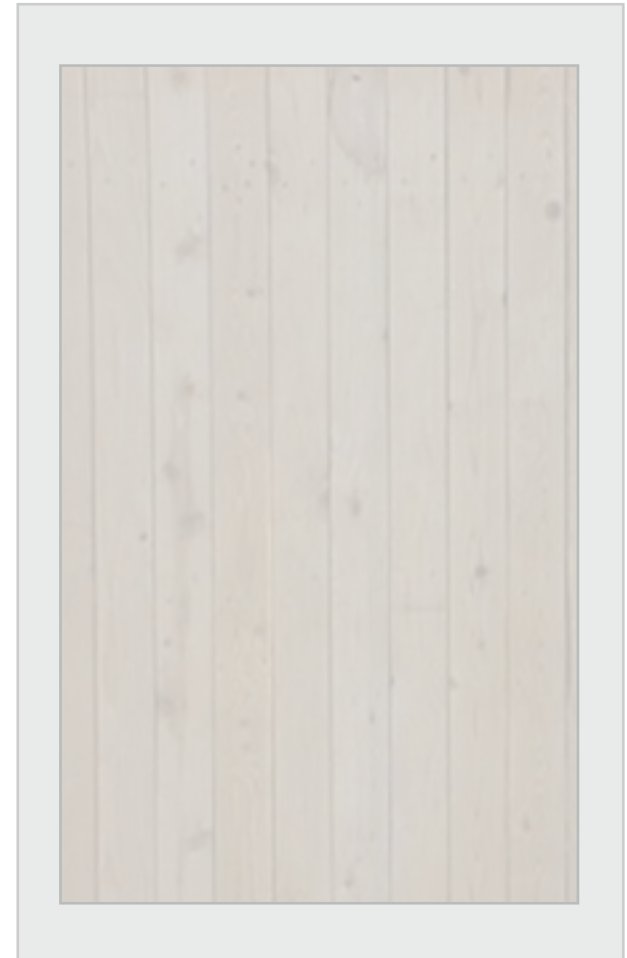
VIEW

```
HomeView = Backbone.View.extend({});
```



VIEW

```
HomeView = Backbone.View.extend({  
  render: function() {  
    this.$el.append("<h1>Wine Cellar</h1>");  
    return this;  
  }  
});
```



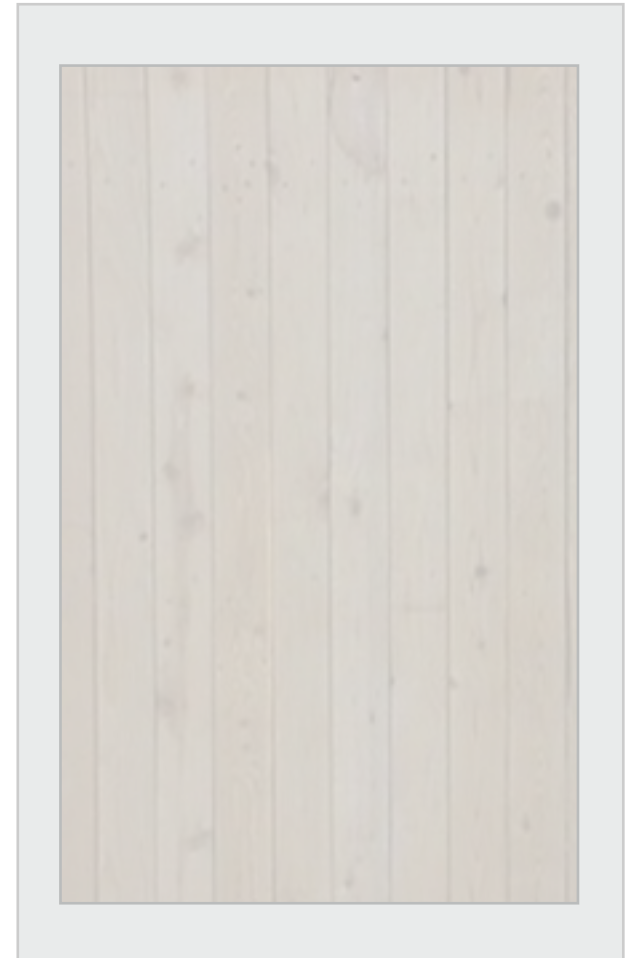
HANDS ON

```
view
```

```
view.render()
```

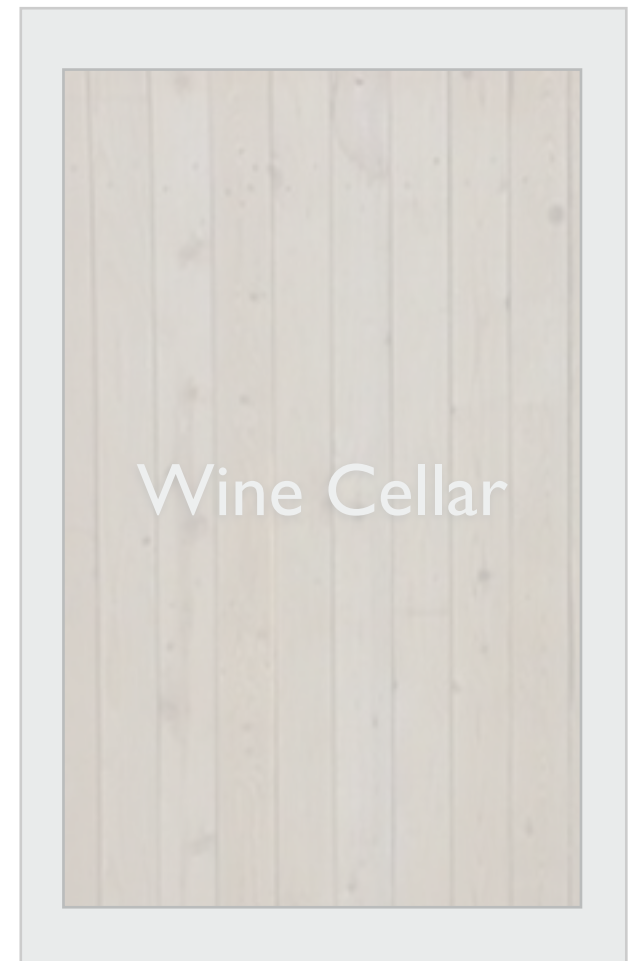
```
view.el
```

```
view.render().el
```



VIEW

```
HomeView = Backbone.View.extend({  
  initialize: function() {  
    this.render();  
  },  
  render: function() {  
    this.$el.append("<h1>Wine Cellar</h1>");  
    return this;  
  }  
});
```



VIEW

```
HomeView = Backbone.View.extend({  
  initialize: function() {  
    this.render();  
  },  
  render: function() {  
    this.$el.append("<h1>Wine Cellar</h1>");  
    return this;  
  }  
});  
  
$(document).ready(function() {  
  view = new HomeView();  
  $('body').append(view.el);  
});
```



VIEW

```
HomeView = Backbone.View.extend({  
  initialize: function() {  
    this.render();  
  },  
  render: function() {  
    this.$el.append("<h1>Wine Cellar</h1>");  
    return this;  
  }  
});
```



Wine Cellar
Wine Cellar
Wine Cellar
Wine Cellar
Wine Cellar

VIEW

```
HomeView = Backbone.View.extend({  
  initialize: function() {  
    this.render();  
  },  
  render: function() {  
    this.$el.empty();  
    this.$el.append("<h1>Wine Cellar</h1>");  
    return this;  
  }  
});
```



Wine Cellar

HANDS ON

04-list-view

VIEW

```
ListView = Backbone.View.extend({  
  tagName: 'ul',  
  
  render: function() {  
    this.$el.empty();  
    this.$el.append("<li>Wine 1</li>");  
    this.$el.append("<li>Wine 2</li>");  
  
    return this;  
  }  
});
```



VIEW

```
HomeView = Backbone.View.extend({  
  initialize: function() {  
    this.render();  
  },  
  render: function() {  
    this.empty();  
    this.$el.append("<h1>Wine Cellar</h1>");  
  
    this.listView = new ListView();  
    this.$el.append(this.listView.render().el);  
  
    return this;  
  }  
});
```



TEMPLATES

HANDS ON

05–basic–template

TEMPLATE

```
<script type="text/template" id="listTemplate">  
  <li><%= winery %></li>  
</script>
```

TEMPLATE

```
<script type="text/template" id="listTemplate">
  <li><%= winery %></li>
</script>
```

```
ListView = Backbone.View.extend({
  tagName: 'ul',

  initialize: function() {
    this.template = _.template($('#listTemplate').html());
    this.render();
  },

  render: function() {
    ...
    this.$el.append(this.template({winery : "Cakebread"}));
    return this;
  }
});
```

HANDS ON

06-collection-template

TEMPLATE

```
ListView = Backbone.View.extend({
  initialize: function() {
    this.template = _.template($('#listTemplate').html());
    this.listenTo(wines, "all", this.render);
    this.render();
  },

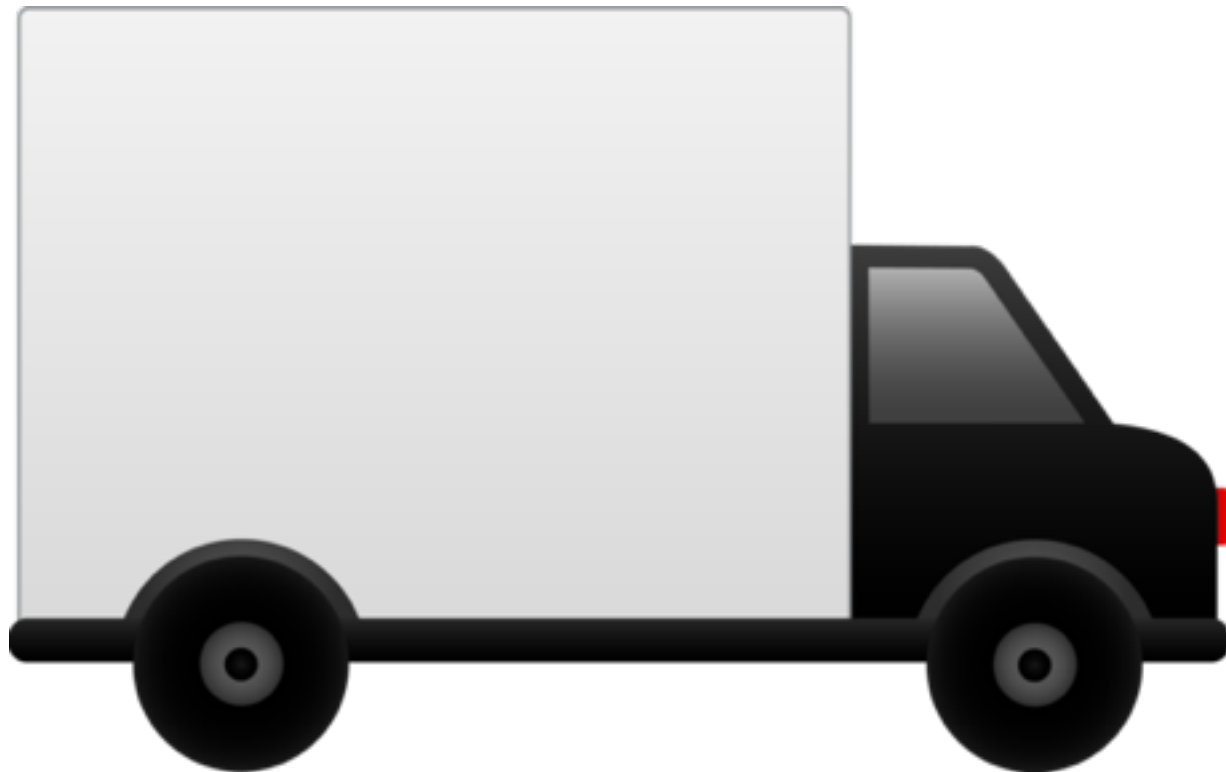
  render: function() {
    var self = this;

    this.$el.empty();

    wines.each(function(model){
      self.$el.append(self.template( model.toJSON() ));
    });

    return this;
  }
});
```


ROUTER



ROUTER



HANDS ON

07-basic-router

ROUTER

```
AppRouter = Backbone.Router.extend({  
  routes:{  
    "":"home",  
    "add":"add"  
  },  
  
  home:function () {  
    new HomeView();  
  },  
  add:function () {  
    new AddView();  
  }  
});
```

ROUTER

```
var app = (function($){  
    ...  
    var initialize = function() {  
        wineApp = new AppRouter();  
        Backbone.history.start();  
    }  
    return { initialize : initialize }  
})(jQuery));  
  
$(document).ready(function () {  
    app.initialize();  
});
```

HANDS ON

10-adv-router

ROUTER

```
var initialize = function() {  
  var wines = new Wines();  
  
  wineApp = new AppRouter({collection : wines});  
  Backbone.history.start();  
}
```

ROUTER

```
AppRouter = Backbone.Router.extend({
  routes:{
    "":"home",
    "add":"add"
  },

  initialize:function(options) {
    this.collection = options.collection;
  },

  home:function () {
    new HomeView({collection:this.collection});
  },
  add:function () {
    new AddView({collection:this.collection});
  }
});
```


LIST VIEW

```
render: function() {  
    var self = this;  
  
    this.$el.empty();  
  
    this.collection.each(function(model) {  
        self.$el.append(self.template(model.toJSON()));  
    });  
  
    return this;  
}
```

HANDS ON

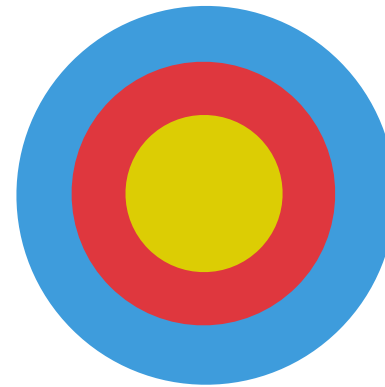
9-events

EVENTS

```
AddView = Backbone.View.extend({  
  
  events: {  
    "click #addBtn": "add",  
  },  
  
  ...  
  
  add: function(e) {  
    // do something...  
  
    return this;  
  }  
});
```

SAVE

```
save: function(e) {  
  e.preventDefault();  
  
  var wine = new Wine({winery:$('#winery').val() });  
  this.collection.add(wine);  
  
  this.router.navigate('#',{trigger: true});  
  
  return this;  
}
```





SaaS



SaaS



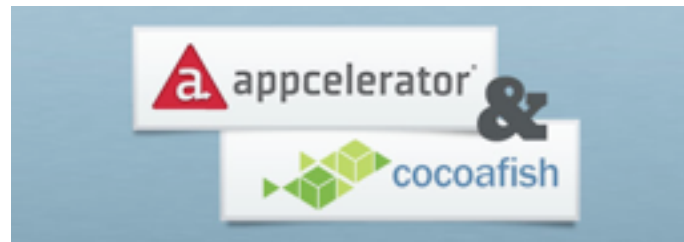
PaaS



SaaS



(m)BaaS



PaaS



API

API RESTful

Custom RESTful API

GET

<http://api.stackmob.com/todo>

PUT

<http://api.stackmob.com/todo/1234>

Easy to use SDKs

Easy to use SDKs

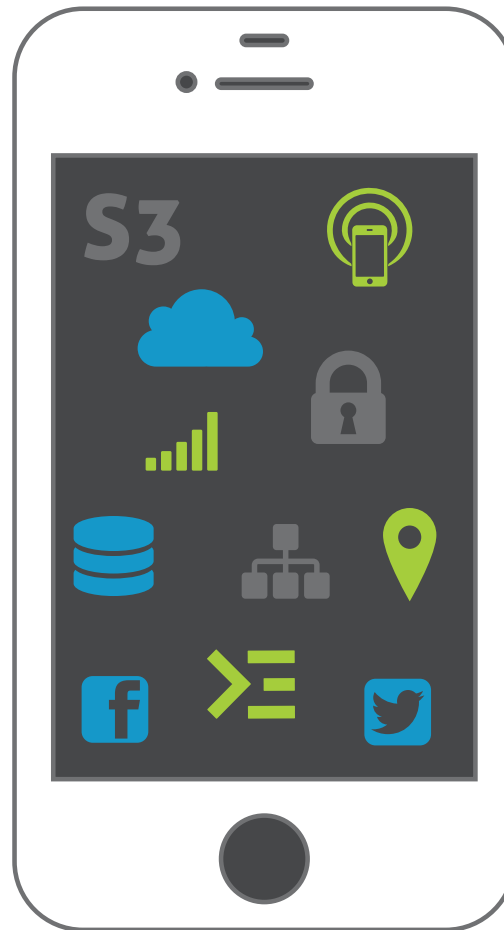


HTML



HTML





Amazon S3

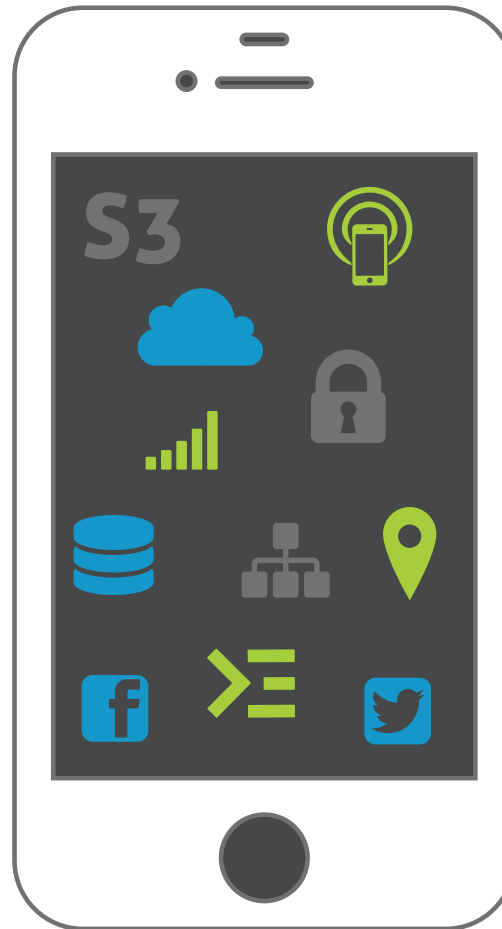
Datastore

Analytics

Load Balancing

Facebook

Access Controls



User
Authentication

Push
Notifications

Versioning

Geo Queries

Collaboration

Twitter

Custom Code





and talk to **any** API



and talk to **any** API

HANDS ON

`stackmob-init.js`

INIT

```
StackMob.init({  
  publicKey: "814004dd-a72a-4425-9d2e-63d21d76588e",  
  apiVersion: 0  
});
```

DASHBOARD

The screenshot shows the StackMob Dashboard interface. At the top, the StackMob logo and 'Dashboard' text are on the left. The navigation bar includes links for Home, Dashboard, Marketplace, Developer, Downloads, and Support. A user profile dropdown for 'Sid' is on the right. Below the navigation bar, a sidebar on the left lists application management options: aptshare (selected), Schema Configuration, Data Management, SDK Downloads, My Modules, Collaboration, Deploy, and Usage Analytics. The main content area is titled 'App Information & Keys' and features a '+ Create New App' button. It is divided into three sections: Notifications, App Info, and Keys/App Settings. The Notifications section contains two announcements from StackMob. The App Info section displays details for the 'aptshare' application, including its organization and team status. The Keys section shows development keys for the application.

StackMob Dashboard

Home Dashboard Marketplace Developer Downloads Support Sid

aptshare App Information & Keys + Create New App

Schema Configuration

Data Management

SDK Downloads

My Modules

Collaboration

Deploy

Usage Analytics

Notifications

New StackMob Developer Center
We're happy to announce the launch of our new StackMob Developer Center! Whether you're a StackMob beginner or a seasoned StackMob pro, the new Developer Center is your go-to source for examples, guides, and downloads to quickly develop your mobile app. The better our docs, the better and faster you can work. [Read more about the new StackMob Developer Center!](#)
Jun 17, 2013, Monday
Erick

StackMob Introduces Smarter Offline Sync
[Learn more from this blogpost.](#)
May 14, 2013, Tuesday

App Info

Application: aptshare

Organization: Sidney Maestre

Teams: No teams. [Manage Teams](#)

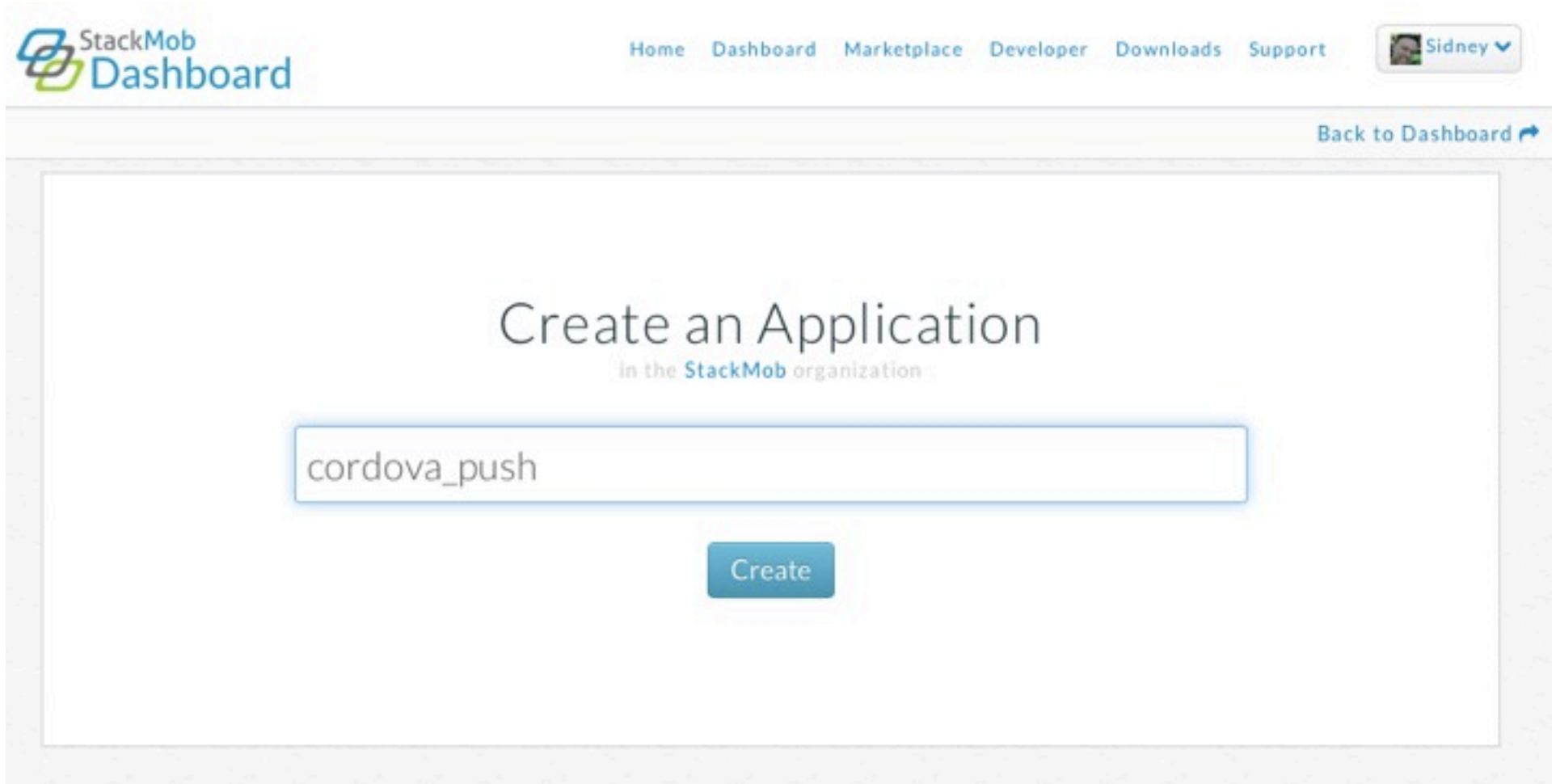
Keys **App Settings**

Development Keys:

Public: 3033af06-aa79-40af-a6cc-57eed020052a

Private: [Show](#)


CREATE APP



The screenshot shows the StackMob Dashboard interface. At the top left is the StackMob logo. To its right is a navigation bar with links: Home, Dashboard, Marketplace, Developer, Downloads, and Support. Further right is a user profile dropdown for 'Sidney'. Below the navigation bar is a 'Back to Dashboard' link with an external icon. The main content area is titled 'Create an Application' with the subtitle 'in the StackMob organization'. Below this is a text input field containing 'cordova_push' and a blue 'Create' button.

StackMob Dashboard

[Home](#) [Dashboard](#) [Marketplace](#) [Developer](#) [Downloads](#) [Support](#)

 Sidney ▾

[Back to Dashboard](#) ↗

Create an Application

in the StackMob organization

Create

COPY



Choose a Project Type

Use our Starter Project

Use Your Own Existing Project

Set up Starter Project

1. [Download JavaScript Starter Project v0.9.2](#)
2. Edit `app.js` and add your public key to the `StackMob.init(..)` block:

```
Showing information for app: cordova_push

1 StackMob.init({
2   publicKey: "f7dbe38c-7400-40cf-a2c1-d51c360333e7",
3   apiVersion: 0
4 });
```

HANDS ON

[10-stackmob.html](#)

STACKMOB SDK

```
stackmob-js-0.9.2-min.js"  
stackmob-init.js
```

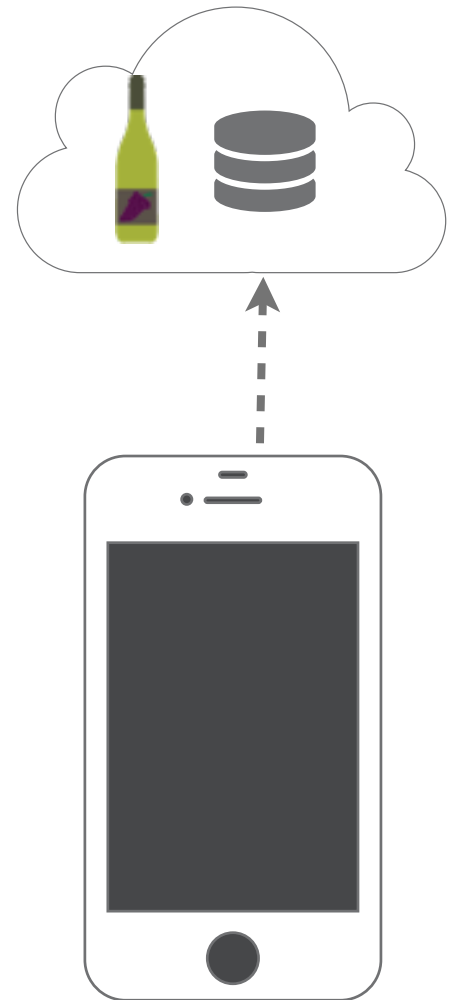
STACKMOB

```
var Wine = StackMob.Model.extend({  
  schemaName: "wine"  
});  
  
var Wines = StackMob.Collection.extend({  
  model: Wine  
});
```



FETCH

```
var Wine = StackMob.Model.extend({  
  schemaName: "wine"  
});  
  
var Wines = StackMob.Collection.extend({  
  model: Wine  
});  
  
var wines = new Wines();  
  
wines.fetch({async: true});
```



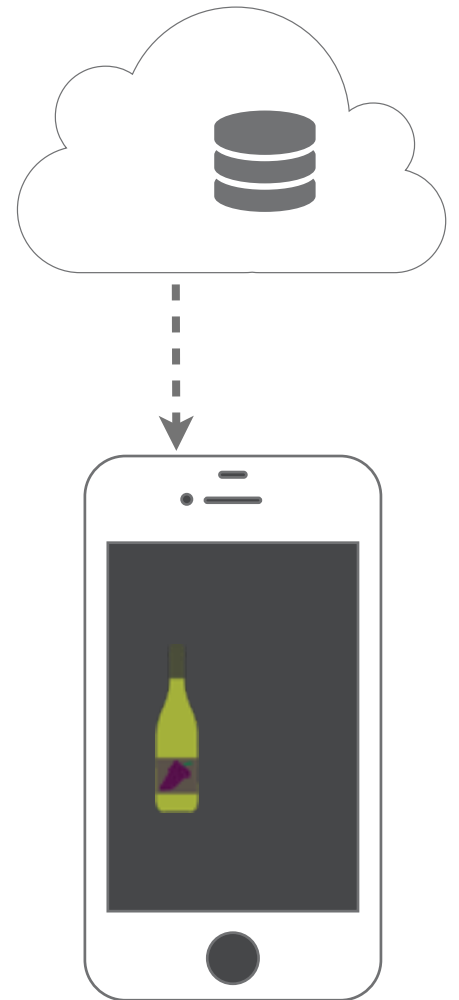
FETCH

```
var Wine = StackMob.Model.extend({
  schemaName: "wine"
});

var Wines = StackMob.Collection.extend({
  model: Wine
});

var wines = new Wines();

wines.fetch({async: true});
```



CREATE

```
var Wine = StackMob.Model.extend({
  schemaName: "wine"
});

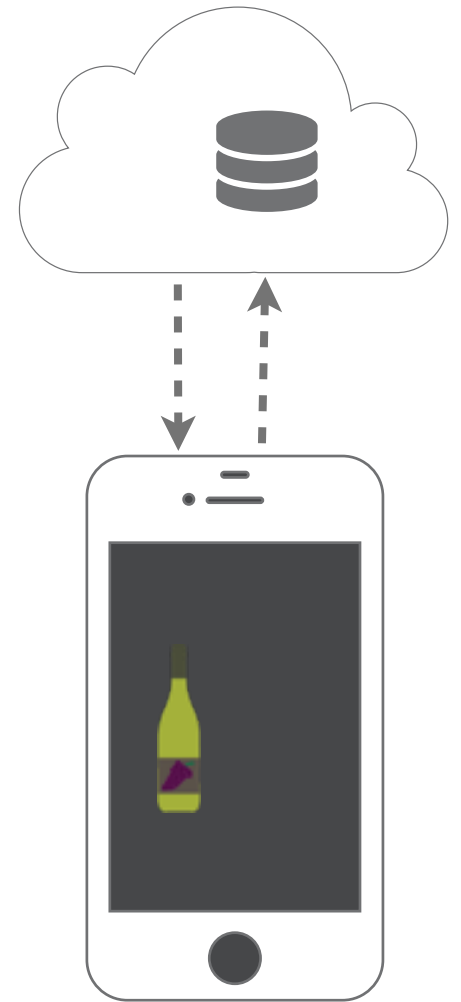
var Wines = StackMob.Collection.extend({
  model: Wine
});

var wines = new Wines();

wines.fetch({async: true});

var wine = new Wine({name: 'Robert Mondavi'});

wine.create({
  success: function(model){
    wines.add(model);
  }
});
```



HANDS ON

11-update

TEMPLATE

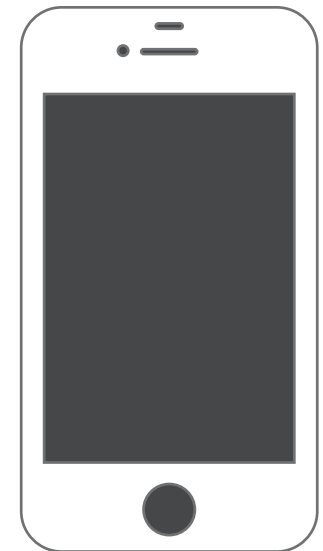
```
<script type="text/template" id="listTemplate">  
  <li><a href="#update/<%= wine_id %>"><%= winery %></a></li>  
</script>
```

ROUTER

```
AppRouter = Backbone.Router.extend({  
  routes:{  
    "":"home",  
    "add":"add",  
    "update/:id":"update"  
  },  
  
  ...  
  
  update:function(e) {  
    model = this.collection.get(e);  
    new UpdateView({model: model});  
  }  
});
```

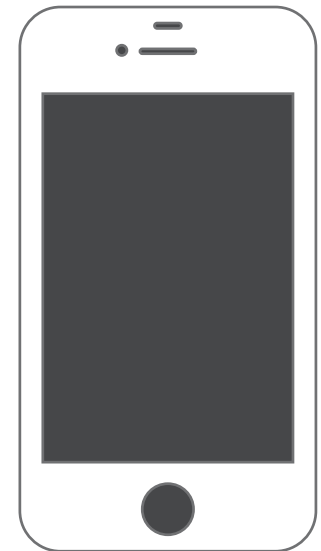
VIEW

```
var UpdateView = Backbone.View.extend({  
  ...  
  initialize: function() {  
    this.model = this.options.model;  
  },  
  
  save: function(e) {  
    this.model.save({winery:$('#winery').val()}, {  
      success: function(model) {  
        }  
    });  
  
    return this;  
  }  
});
```



VIEW

```
var UpdateView = Backbone.View.extend({  
  ...  
  initialize: function() {  
    this.model = this.options.model;  
  },  
  
  save: function(e) {  
    this.model.save({winery:$('#winery').val()}, {  
      success: function(model) {  
        }  
    });  
  
    return this;  
  }  
});
```



HANDS ON

12-user



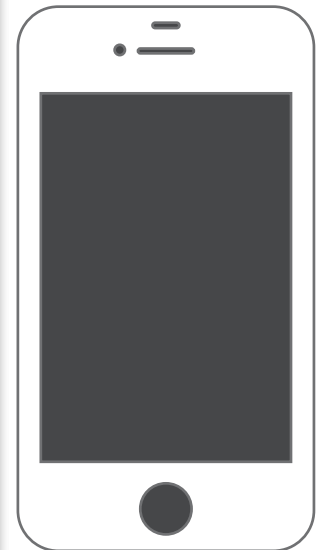
SIGNUP

```
signup:function (e) {  
  e.preventDefault();  
  
  var username = $("#signUpForm input.username").val();  
  var pw = $("#signUpForm input.password").val();  
  
  var user = new StackMob.User({ username: username,  
                                  password: pw });  
  
  return this;  
}
```



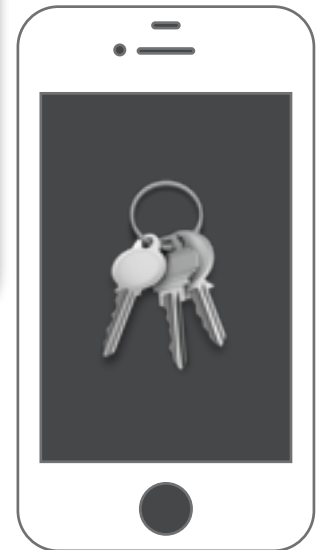
SIGNUP

```
signup:function (e) {  
  e.preventDefault();  
  
  var username = $("#signUpForm input.username").val();  
  var pw = $("#signUpForm input.password").val();  
  
  var user = new StackMob.User({ username: username,  
                                  password: pw });  
  
  user.create({  
    success: function(data) {  
      console.log(data.toJSON());  
    },  
    error: function(error) {  
      console.log(error);  
    }  
  });  
  
  return this;  
}
```



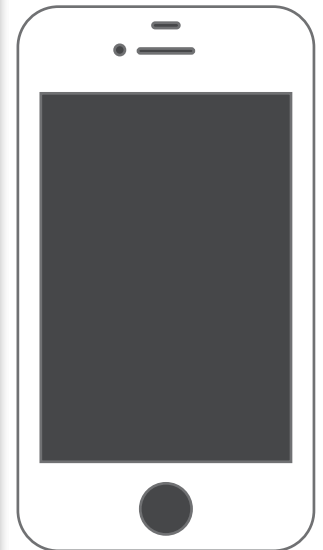
LOGIN

```
login:function (e) {  
  e.preventDefault();  
  
  var username = $("#signUpForm input.username").val();  
  var pw = $("#signUpForm input.password").val();  
  
  var user = new StackMob.User({ username: username,  
                                  password: pw });  
  
  return this;  
}
```



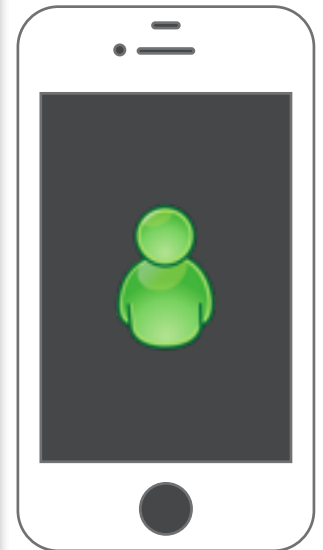
LOGIN

```
login:function (e) {  
  e.preventDefault();  
  
  var username = $("#signUpForm input.username").val();  
  var pw = $("#signUpForm input.password").val();  
  
  var user = new StackMob.User({ username: username,  
                                  password: pw });  
  
  user.login(true,{  
    success: function(data) {  
      console.log(data);  
    },  
    error: function(error) {  
      console.log(error);  
    }  
  });  
  
  return this;  
}
```



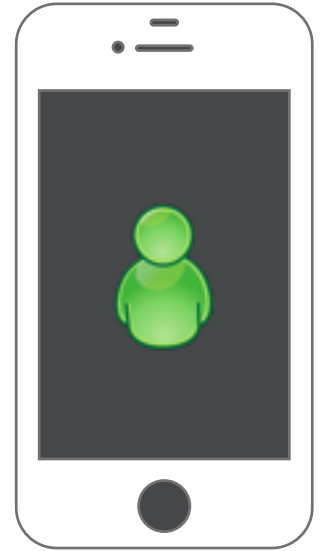
LOGIN

```
login:function (e) {  
  e.preventDefault();  
  
  var username = $("#signUpForm input.username").val();  
  var pw = $("#signUpForm input.password").val();  
  
  var user = new StackMob.User({ username: username,  
                                  password: pw });  
  
  user.login(true,{  
    success: function(data) {  
      console.log(data);  
    },  
    error: function(error) {  
      console.log(error);  
    }  
  });  
  
  return this;  
}
```



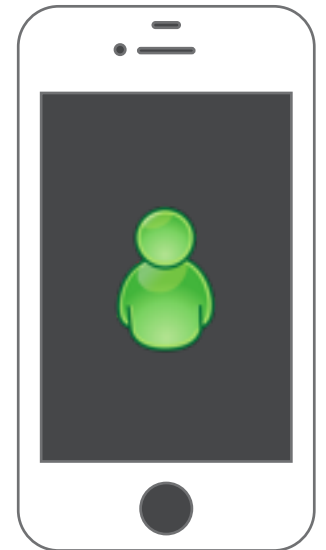
STATUS

```
status:function (e) {  
  e.preventDefault();  
  
  console.log(StackMob.isLoggedIn());  
  
  return this;  
}
```



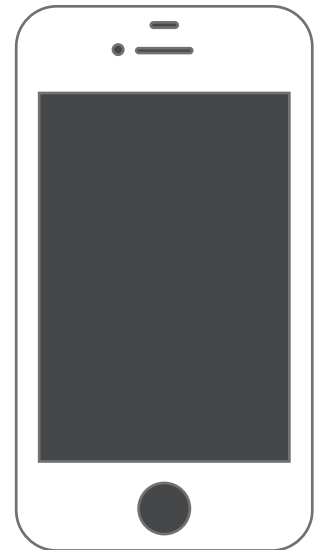
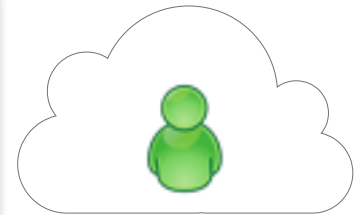
LOGOUT

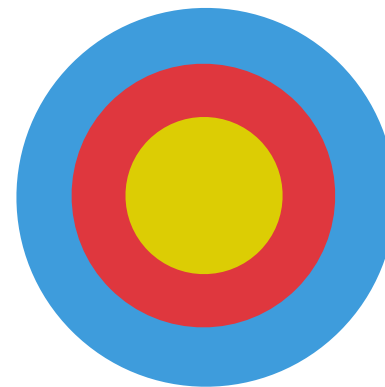
```
logout:function (e) {  
  e.preventDefault();  
  
  var user = new StackMob.User();  
  
  return this;  
}
```



LOGOUT

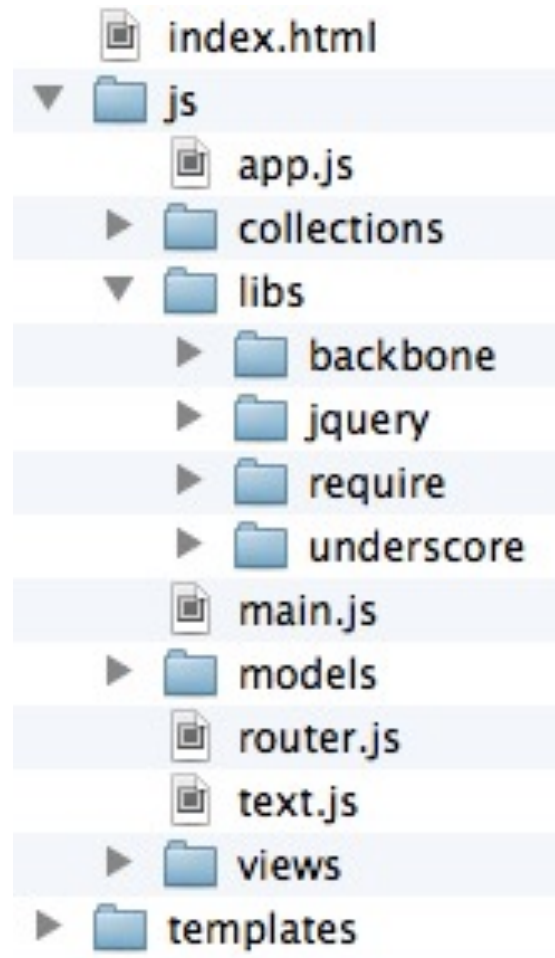
```
logout:function (e) {  
  e.preventDefault();  
  
  var user = new StackMob.User();  
  user.logout({  
    success: function() {  
      console.log("logout success");  
    },  
    error: function(error) {  
      console.log(error);  
    }  
  });  
  
  return this;  
}
```







STRUCTURE



MODEL

```
Wine = Backbone.Model.extend();
```



MODEL

```
define([  
  'backbone'  
], function(Backbone) {  
  var WineModel = Backbone.Model.extend();  
  return WineModel;  
});
```



COLLECTION

```
Wine = Backbone.Model.extend();  
  
Wines = Backbone.Collection.extend({  
  model: Wine  
});
```



COLLECTION

```
define([
  'backbone',
  'models/wine/Model'
], function(Backbone, Model){

  var WineCollection = Backbone.Collection.extend({
    model: Model
  });

  return WineCollection;
});
```



VIEW

```
ListView = Backbone.View.extend({  
  ...  
  render: function() {  
    var      el = this.$el,  
            template = this.template;  
  
    el.empty();  
  
    wines.each(function(wine){  
      el.append(template( wine.toJSON() ));  
    });  
  
    return this;  
  }  
});
```



VIEW

```
define([
  'jquery',
  'underscore',
  'backbone',
  'text!templates/wine/WineListTemplate.html'
], function($, _, Backbone, WineListTemplate){

  var WineListView = Backbone.View.extend({

    ...

  });

  return WineListView;
});
```



HANDS ON

require

main.js

```
<script data-main="js/main" src="js/libs/require/require.js"></script>
```

BASEURL

```
require.config({  
  baseUrl: "/js/"  
});
```

PATHS

```
require.config({
  baseUrl: "/js/",
  paths: {
    jquery: 'libs/jquery/jquery-1.8.2',
    underscore: 'libs/underscore/underscore-1.4.4',
    backbone: 'libs/backbone/backbone-1.0.0',
    templates: '../templates',
    app: 'app'
  }
});
```

SHIM

```
require.config({
  baseUrl: "/js/",
  paths: {
    jquery: 'libs/jquery/jquery-1.8.2',
    underscore: 'libs/underscore/underscore-1.4.4',
    backbone: 'libs/backbone/backbone-1.0.0',
    templates: '../templates',
    app: 'app'
  },

  shim: {
    underscore: {
      exports: '_'
    },
    backbone: {
      deps: ["underscore", "jquery"],
      exports: "Backbone"
    }
  }
});
```

START APP

```
require(['jquery','app'], function( $, App ){  
    $(function(){  
        App.initialize();  
    });  
});
```


APP.JS

```
define([
  'router'
], function(Router){

  var initialize = function() {
    Router.initialize();
  };

  return {
    initialize: initialize
  };
});
```

RJS



BUILD.JS

```
({  
  appDir: "../",  
  baseUrl: "js",  
  dir: "../..../appdirectory-build",  
  paths: {  
    ...  
  },  
  
  shim: {  
    ...  
  },  
  modules: [  
    {  
      name: "main"  
    }  
  ]  
})
```

BUILD

```
node r.js -o build.js optimize=none
```

347k

OPTIMIZE

```
node r.js -o build.js
```

134k

Resources

bit.ly/freebackbonejs

github.com/SidneyAllen

developer.stackmob.com

Q&A