

WebApp(s) Mobile(s)

avec Backbone & co



Sainté **MobileDays**

22-23 novembre 2013 | Saint-Etienne

@k33g_org

“Avant-vendeur”

Steria Lyon

Nous allons voir :

Ma problématique

Des frameworks web mobile

Mon choix > JQM

Backbone

Backbone + JQM & co

Ma problématique “web mobile”

Produire

vite (& facile) > **avv + démo**

“beau ou pas trop moche”

> **“faire rêver” le client**

bien > **réutilisable & exploitable**

Les frameworks “web mobile(s)”

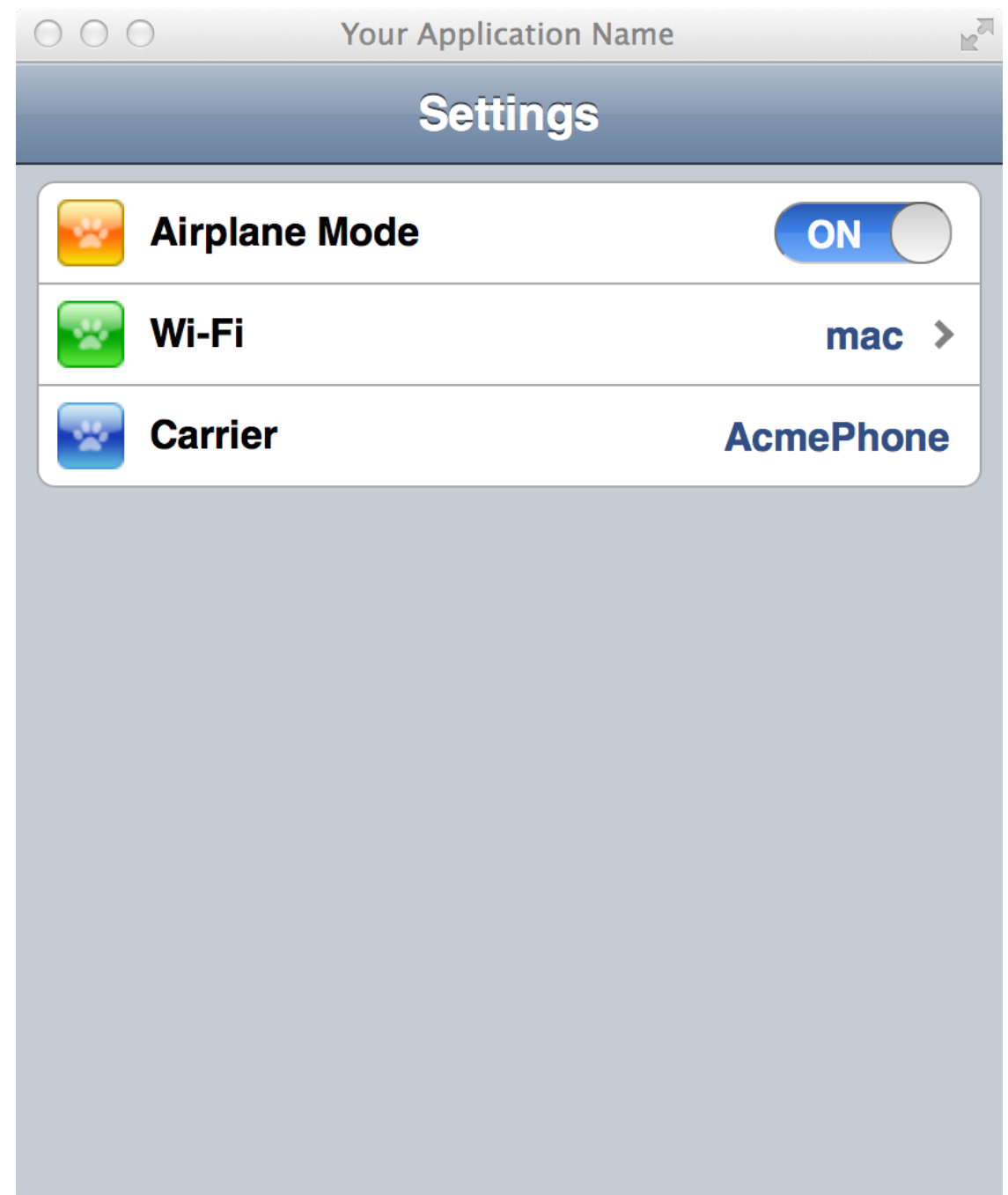
Pourquoi les utiliser ?

Ergonomie mobile
Design mobile
Spécificités mobiles

ils sont nombreux

Quelques exemples

Dojo-Mobile
à base de tags html
+ javascript
skins iOS, Android, BlackBerry
“fait tout”
mais ...



C'est du web!

EVITER le "OS like à tout prix"

c'est souvent "moyen"

>> restez simple

>> personnalisez

SenchaTouch

Composants Javascript

Très puissant

Reconnu

forte communauté

“fait tout”

mais ...

The screenshot shows the Sencha Touch Kitchen Sink application. The left sidebar contains a menu with items: Buttons, Forms (highlighted), List, Nested List, Icons, Toolbars, Carousel, Tabs, Bottom Tabs, and Overlays. The main content area is titled 'Forms' and has sub-tabs for 'Basic', 'Sliders', and 'Toolbars'. The 'Basic' tab is active, displaying a 'Personal Info' form. The form contains the following fields:

Personal Info	
Name*	Tom Roy
Password	
Email	me@sencha.com
Url	http://sencha.com
Spinner	0 - +
Cool	<input checked="" type="checkbox"/>
Start Date	11/13/2013 ▼
Rank	Master ▼
Bio	

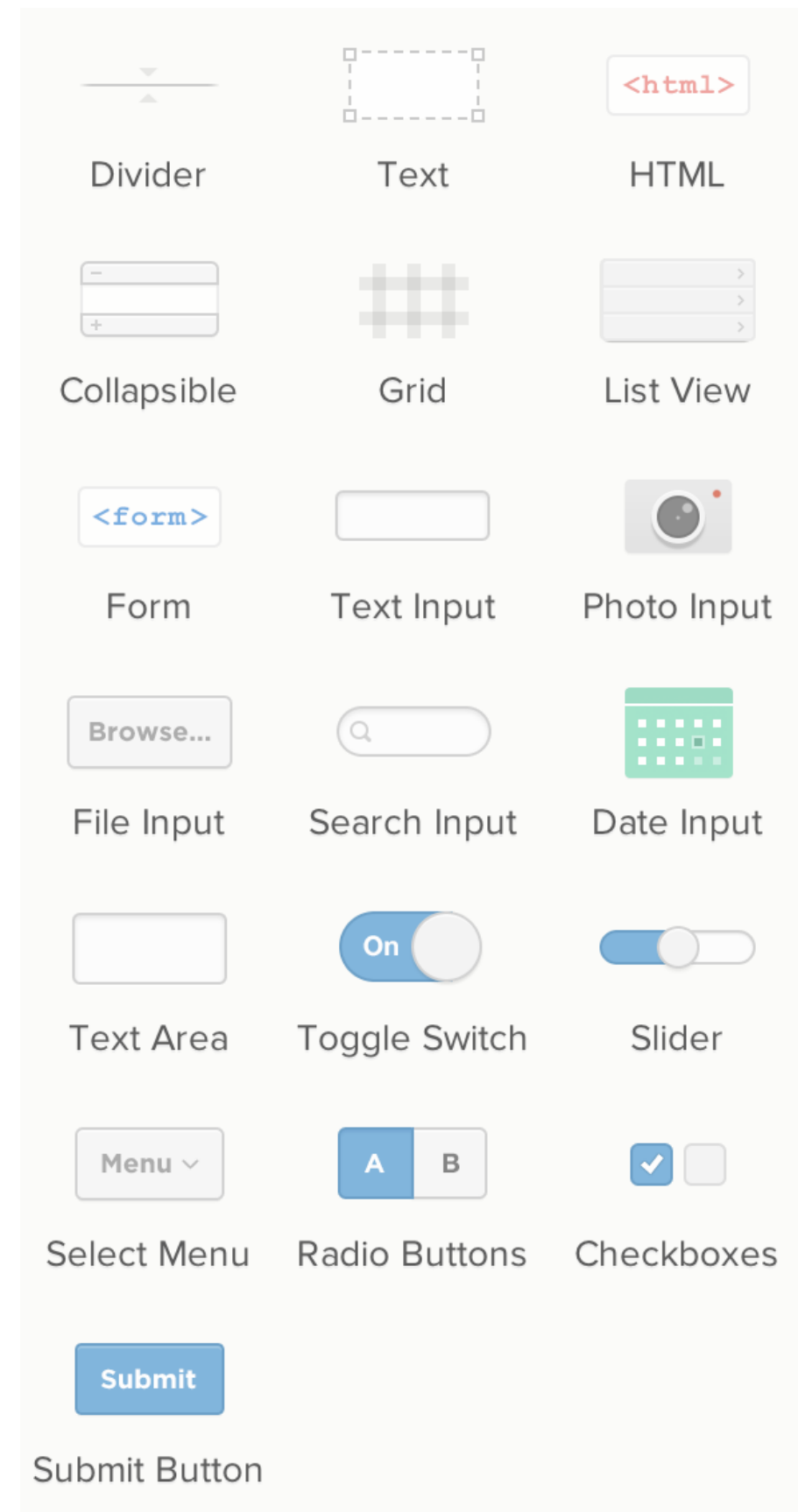
Beau, mais ...
difficile à “skinner”
apprentissage long

```
FormPanel.js
7 Ext.define('KitchenSink.view.FormPanel', {
8     extend: 'Ext.form.Panel',
9
10    requires: [
11        'Ext.form.FieldSet',
12        'Ext.field.Number',
13        'Ext.field.Spinner',
14        'Ext.field.Password',
15        'Ext.field.Email',
16        'Ext.field.Url',
17        'Ext.field.DatePicker',
18        'Ext.field.Select',
19        'Ext.field.Hidden',
20        'Ext.field.Radio'
21    ],
22    id: 'basicform',
23    config: {
24        items: [
25            {
26                xtype: 'fieldset',
27                id: 'fieldset1',
28                title: 'Personal Info',
29                instructions: 'Please enter the information above.',
30                defaults: {
31                    labelWidth: '35%'
32                },
33                items: [
34                    {
35                        xtype: 'textfield',
36                        name: 'name',
37                        label: 'Name',
38                        placeholder: 'Tom Roy',
39                        autoCapitalize: true,
40                        required: true,
41                        clearIcon: true
42                    },
43                    {
44                        xtype: 'passwordfield',
45                        name: 'password',
46                        label: 'Password'
47                    },
48                    {
49                        xtype: 'emailfield',
50                        name: 'email',
51                        label: 'Email',
52                        placeholder: 'me@sencha.com'
```

Mon choix >
jQuery Mobile

simple
à base de tags
composants “html”
transitions
“skinnable” facilement

...



Fonctionnement de JQM

Sélection des <elements **data-???** = “something”>

Ajout de balises, classes css, gestion des évènements

Votre HTML devient une IHM Mobile

un peu de HTML >> **démo**

00-jqm & 01-jqm_skin

... mais ...

JQM ne “fait pas tout”

Backbone

1 “librairie” MVC
très légère
avec un modèle objet
(& facile à lire)

Classes are back! \o/


```
var Klass = function() {};  
Klass.extend = Backbone.Model.extend;  
  
var TinyToon = Klass.extend({  
    constructor : function (firstName, lastName){  
        this.firstName = firstName;  
        this.lastName = lastName;  
  
        TinyToon.counter +=1;  
    },  
    toString : function() {  
        return this.firstName + " " +this.lastName  
    }  
}, { /* --- static --- */  
    counter : 0,  
    getCounter : function(){ return TinyToon.counter; }  
});  
  
var babs = new TinyToon("Babs", "Bunny")  
var buster = new TinyToon("Buster", "Bunny")
```

```
var TinyRabbit = TinyToon.extend({  
  constructor : function (firstName, lastName){  
    TinyRabbit.__super__.constructor  
      .call(this, firstName, lastName);  
    TinyRabbit.counter +=1;  
  },  
  whoAmI : function () { return "I'm a RABBIT!"; }  
},{  
  counter : 0,  
  getCounter : function(){ return TinyRabbit.counter; }  
});
```

BackBone

= 4 composants importants

Model
Collection
View (+ Template)
Router

un peu de JS >> **démo**

02-bb

Backbone & REST

`$.ajax()`

```
var TinyToon = Backbone.Model.extend({  
  urlRoot : "/tinytoons"  
});
```

```
var TinyToons = Backbone.Collection.extend({  
  model : TinyToon,  
  url : "/tinytoons"  
});
```

buster.**save**()

POST : <http://domain/tinytoons>

var tinyToon = **new** TinyToon({id:"001"})

tinyToon.**fetch**()

GET : <http://domain/tinytoons/001>

buster.set("twitter", "@buster")

bob.**save**()

PUT : <http://domain/tinytoons/001>

buster.**destroy**()

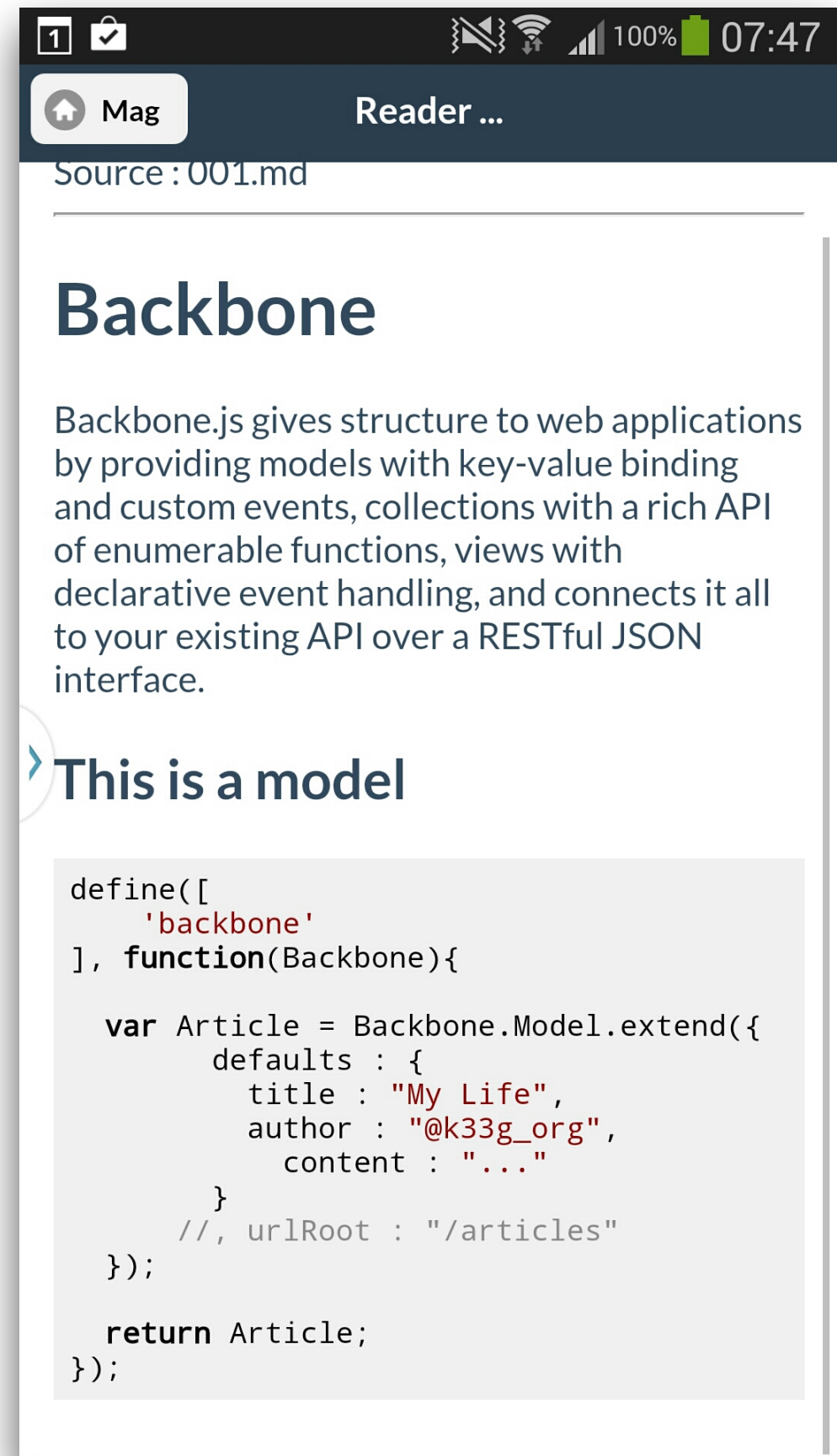
DELETE : <http://domain/tinytoons/001>

var tinyToons = **new** TinyToons() /*collection*/

tinyToons.**fetch**()

GET : <http://domain/tinytoons>

Backbone
+ JQM (& co)
c'est simple



Backbone (+ dépendances)

jQueryMobile

Require.js (+ plugin “text”)

showdown.js

highlight.js

Explication de code >> **démo**

03-bbjqm

>> 1 template de projet tout prêt

[https://github.com/k33g/
saintemobiledays](https://github.com/k33g/saintemobiledays)

>> amusez vous avec (fork it)

Si vous souhaitez aller plus loin :

Grunt

Grunt-contrib-jst

Grunt-contrib-copy

Grunt-contrib-uglify

Questions ?

ping me @k33g_org