BACKBONE.ROCKS

Large Scale Single Page Application Done Right

Jeremy Lu // jeremy@pubulous.com // Html5DevConf 2013

HI, MY NAME IS JEREMY LU

FOUNDER, BUILDER, DEVELOPER

What’s in the talk ?

1. Key concepts for successful large scale web development

2. Backbone.rocks feature highlights

Key concepts for successful large scale web development

MV*

Model-driven approach

Modular design

Dependency Injection (DI)

Inter module communication

Automated building process

Testing

House rules for quality control

Deep understanding of Javascript

MV*

MVC, MVP, MVVM...

You've heard all the buzz words

Clear separation of concerns is the key here

Backbone, Angularjs, Emberjs will all do the tricks

If, and only if, done right - which is the **problem**

Pick one, and make sure every team member **totally** understands how it works

Model-driven approach

"Single source of truth" principle

Model is the single source of truth

Views observe model changes and refresh themselves accordingly

Always trigger view changes via model manipulation

App/UI states always maintained in model

Easier for debugging and testing

Modular design

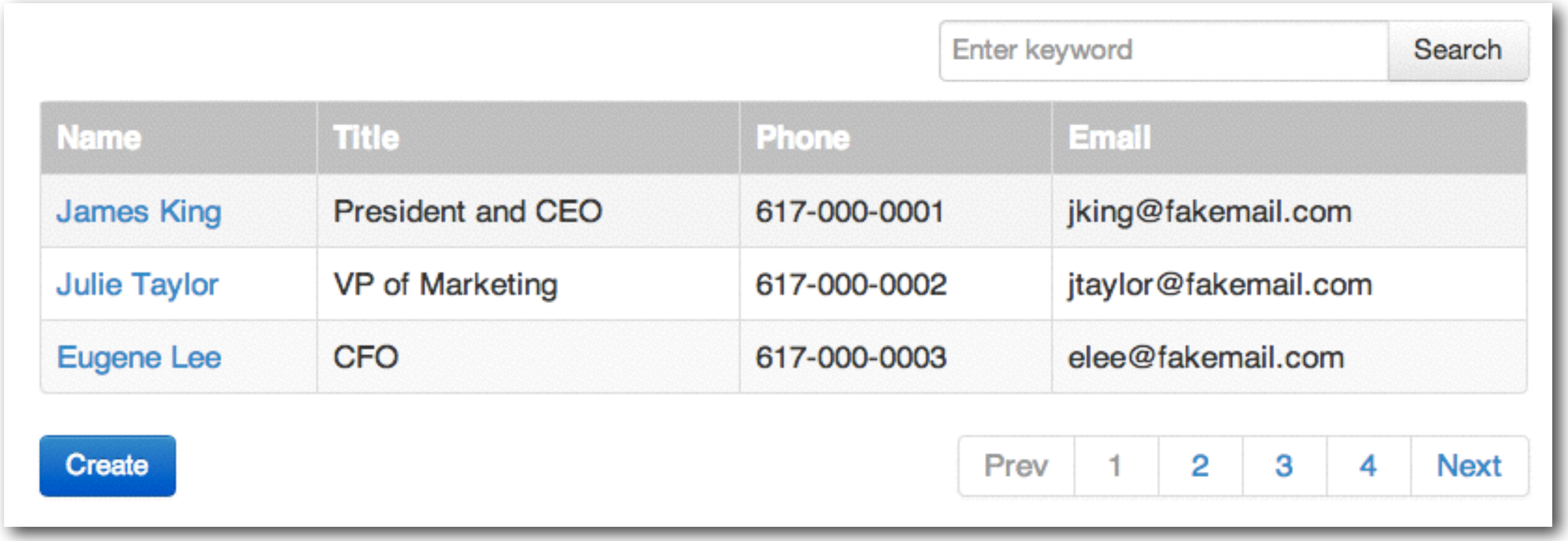
Modular in the sense of
concept and **tooling**

Concept

Organize application into smaller self-contained units

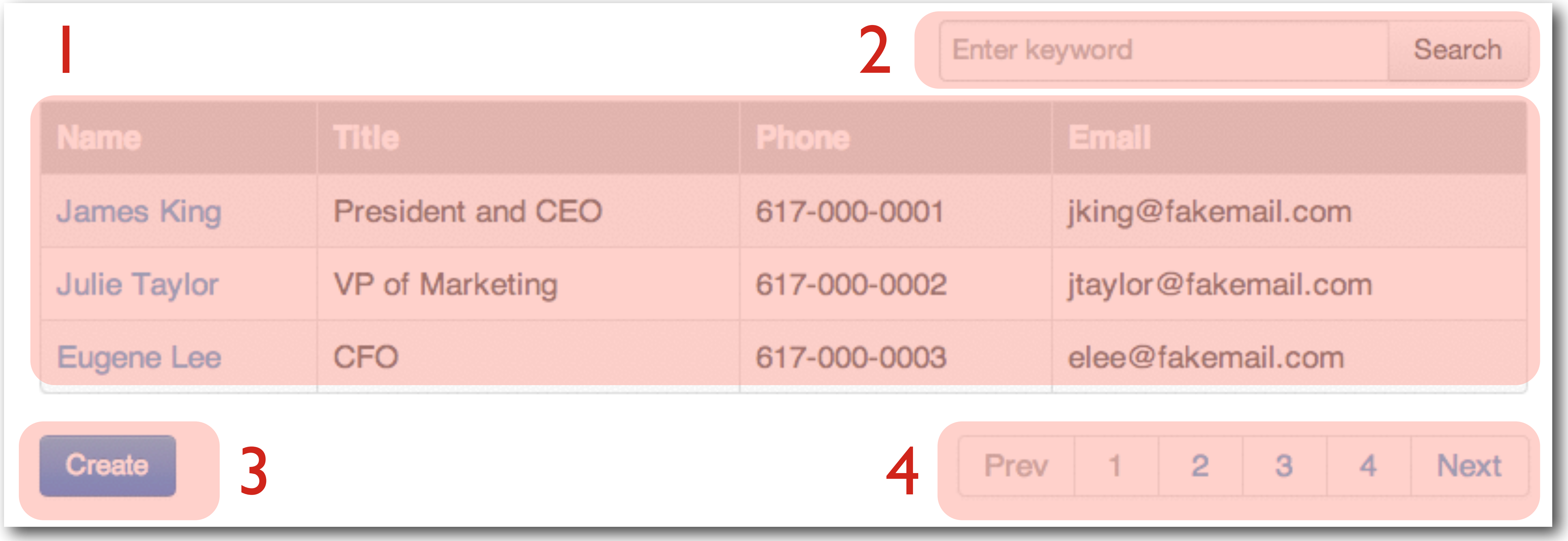
Unit is fully encapsulated and absolutely decoupled from each other and the system

Easier for teams to simultaneously work on multiple parts without interfering or depending on each other



Name	Title	Phone	Email
James King	President and CEO	617-000-0001	jking@fakemail.com
Julie Taylor	VP of Marketing	617-000-0002	jtaylor@fakemail.com
Eugene Lee	CFO	617-000-0003	elee@fakemail.com

A simple page like this



Name	Title	Phone	Email
James King	President and CEO	617-000-0001	jking@fakemail.com
Julie Taylor	VP of Marketing	617-000-0002	jtaylor@fakemail.com
Eugene Lee	CFO	617-000-0003	elee@fakemail.com

Is actually composed of four sub-units

Each unit can be switched or removed and it won’t affect the system at all

Tooling

Module loading and management using “Requirejs”

Requirejs makes module **loading** and **dependency** management much easier for both development and production

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

One line of code gets all dependencies loaded in correct sequence

r.js helps to combine and minify multiple files

Plus, requirejs works well with most **testing** frameworks, like Karma & Testem

Dependency Injection (DI)

Inversion of Control
- the key concept behind

Instead of hardcoding dependencies in code,

Inject them at runtime when needed,

For max flexibility

Helps to make each part of the app highly decoupled

DI goes hand in hand with modular design approach

Modules are highly encapsulated

Everything it needs will be passed in or better yet, injected

That's where DI comes into play

```
// hardcoded dependencies
function fooModule( bar, baz ){
 var r = new bar();
 var z = new baz();
}


// dependencies injected at run time
function fooModule(){
 var r = new bar();
 var z = new baz();
}
```

Results in easily maintainable and testable code

Using “Medic-Injector-JS”

A lightweight DI library with easy to use APIs

Inter module communication

A. Global event bus with namespaced event

```
bus.dispatch( “loginView:submit” );
```

Looks good initially, until it grows out of control...

Events transmitted and caught all over the place, who’s doing what ?

Anyone can listen to and dispatch events, extremely hard to control and debug

B. Application model as signal bus

Create an AppModel as global signal bus

Which also persists application states

Trigger events via getter and setter exposed by AppModel

```
AppModel.set( 'login', {name:'john'} );

bus.dispatch( "loginView:submit" );
```

```
AppModel.on( ‘login’, handleLogin );
```

Easier to track who's changing the application state and triggering events

With added benefit of having all app states persisted in one place

Automated build system

Compile, combine, minify, watch for changes and more ...

For Sass, Coffeescript, Javascript...

Huge time saver for development, staging and production

Use Node.js / Grunt

Testing

BDD + TDD

Karma / Mocha / Chai / Sinon

Continuous Integration and Deploy system

Strider CD

Travis CI

Circle CI

Jenkins CI

House rules for quality control

Use Git

github, bitbucket, self-hosted

Everyone works in own feature branch

When feature completed, send "Pull Request" for code review and discussion

Use rebase and merge wisely

Gitflow is a nice to have

Deep understanding of Javascript

The Good Parts

Effective JavaScript

Secrets Of The JavaScript Ninja

Closure

Generic Object Oriented concept

Universal Module Definition

Eventing

Variable scopes

Function as first class citizen

RECAP

key concepts for successful large scale web development

MV*

Model-driven approach

Modular design

Dependency Injection (DI)

Inter module communication

Automated building process

Testing

House rules for quality control

Deep understanding of Javascript

Entering Backbone

It's a MVC **meta**-library

Not a full fledged all-in-one framework

Easy to use, but often times mis-used

Because of it’s minimal design

Lack of explicit guidelines for how to do things certain way

Developers got “creative” and shoot themselves in the foot

Backbone.rocks feature highlight

* Backbone.rocks implemented the gist of aforementioned key concepts on top of backbone

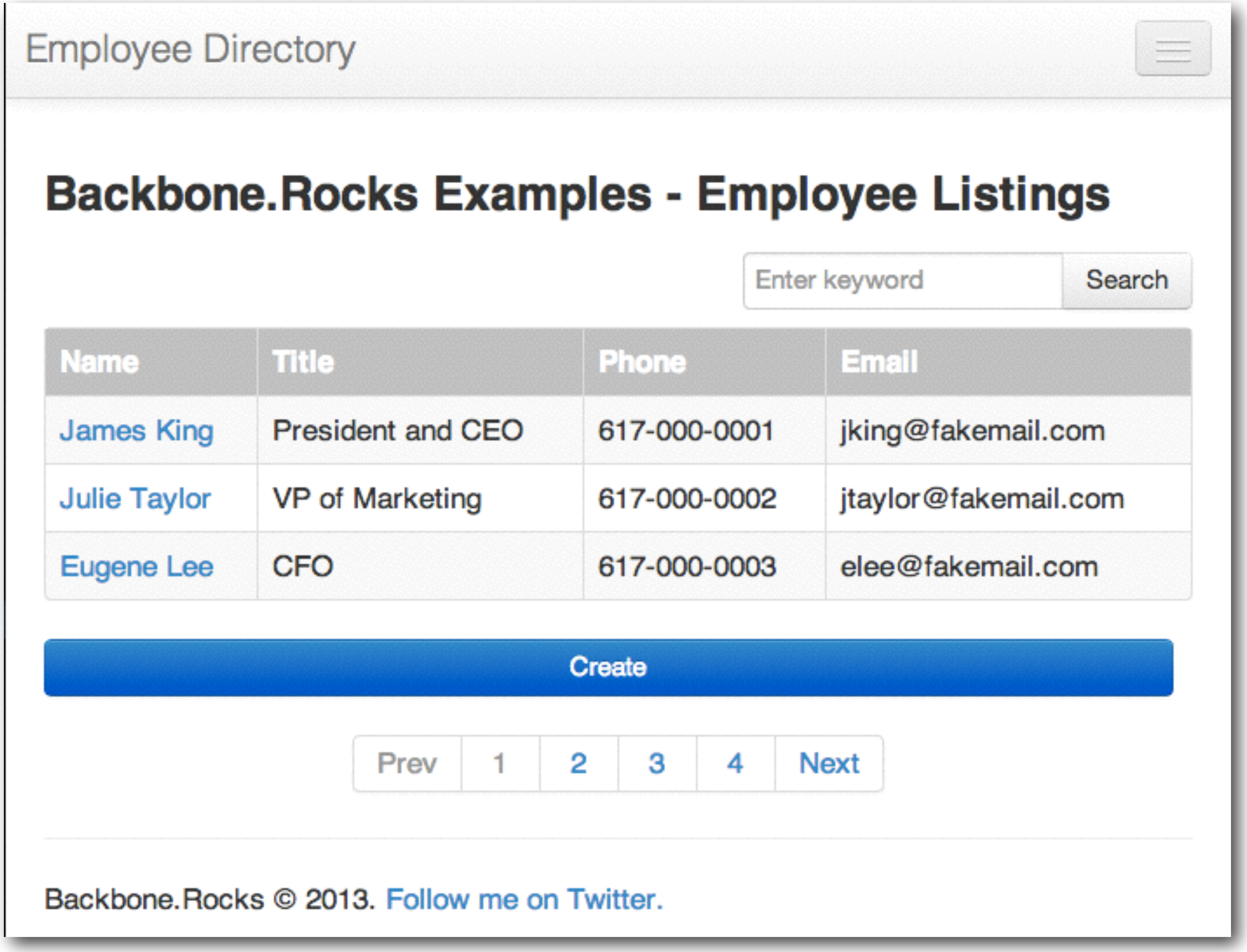
* It’s an add-on to Backbone

* It’s a development showcase for large scale web application

* It’s also a free ebook available on github

Demo

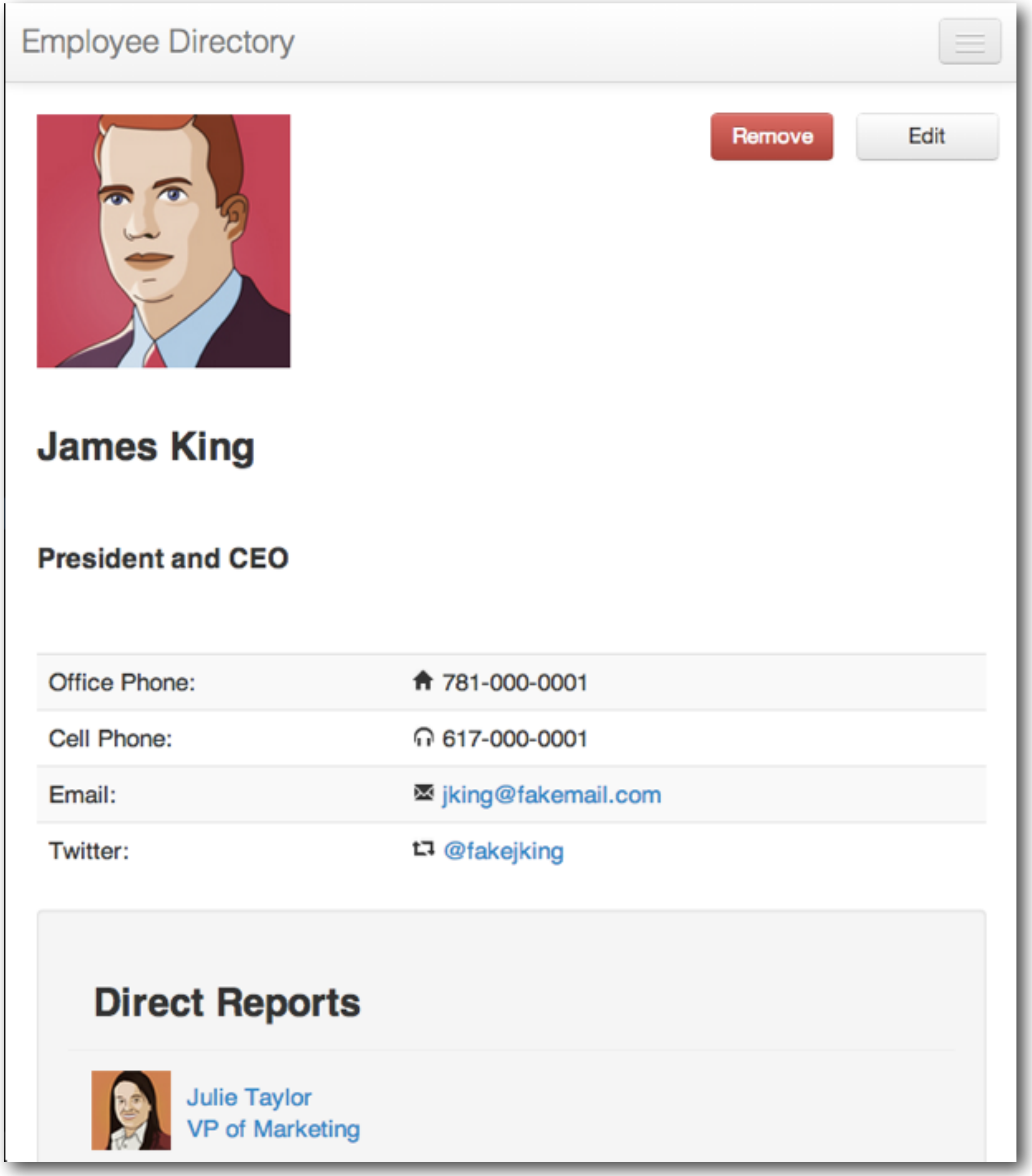
https://github.com/devmatters/backbone.rocks


Employee Directory

Backbone.Rocks Examples - Employee Listings

Name	Title	Phone	Email
James King	President and CEO	617-000-0001	jking@fakemail.com
Julie Taylor	VP of Marketing	617-000-0002	jtaylor@fakemail.com
Eugene Lee	CFO	617-000-0003	elee@fakemail.com

Backbone.Rocks © 2013. Follow me on Twitter.



Employee Directory

Remove
Edit

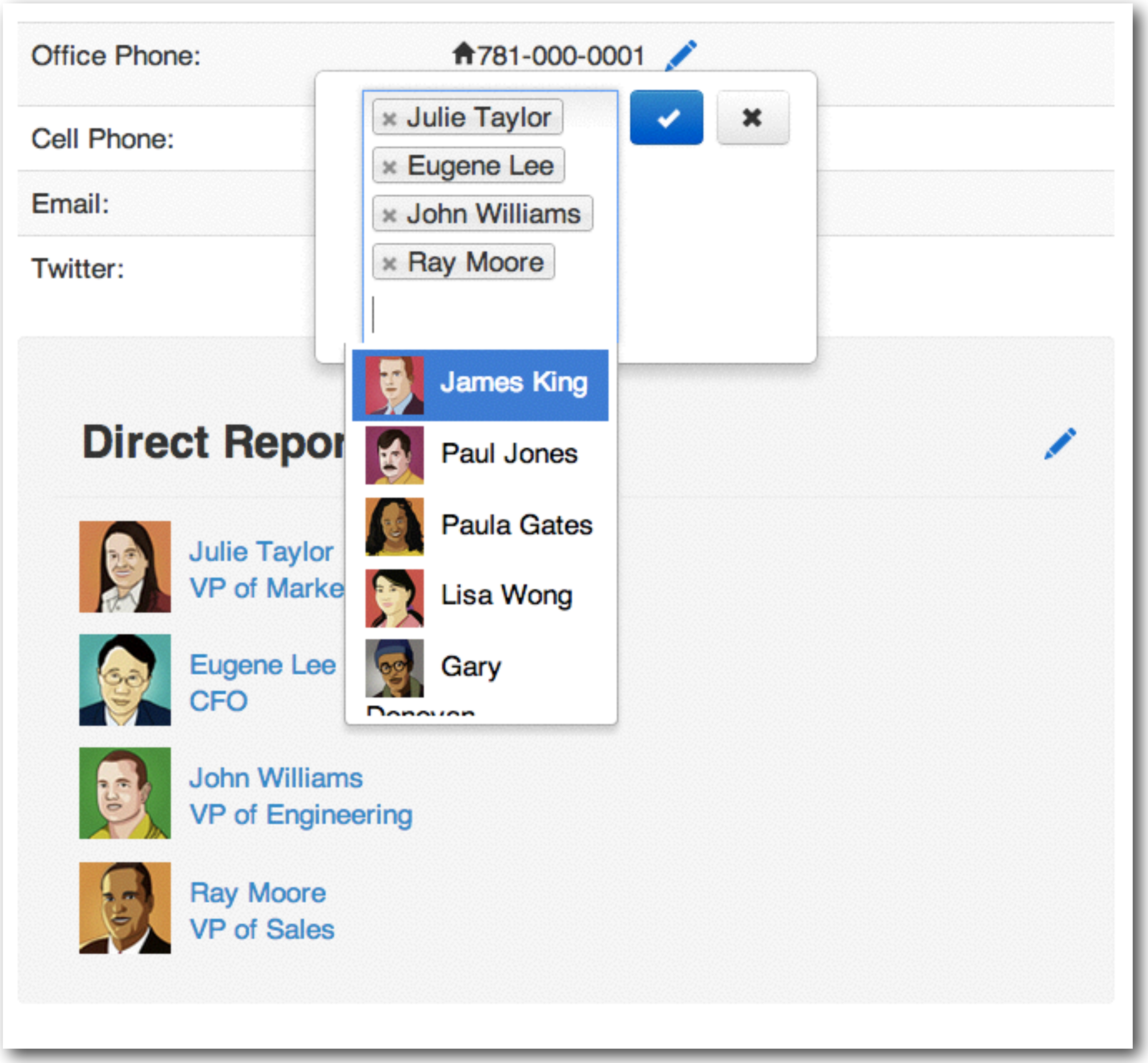
James King

President and CEO

Office Phone:	781-000-0001
Cell Phone:	617-000-0001
Email:	jking@fakemail.com
Twitter:	@fakejking

Direct Reports

Julie Taylor
VP of Marketing



Office Phone: 781-000-0001

Cell Phone:

Email:

Twitter:

× Julie Taylor
× Eugene Lee
× John Williams
× Ray Moore

James King
Paul Jones
Paula Gates
Lisa Wong
Gary Donovan

Direct Repor

Julie Taylor
VP of Marke

Eugene Lee
CFO

John Williams
VP of Engineering

Ray Moore
VP of Sales

Model

- Declarative **one-to-many** property and handler
- Declarative **validators** for each property
- Declarative **default value** setting and re-setting

```
oneToMany: {
    reports: {type: 'js/models/ReportCollection', lazy: true}
},
validators: {
    firstName: 'validateName',
    lastName: 'validateName'
},
validateName: function( value, field, options ){
    if( value === “” )
        return {field: field, msg: 'can not be empty'};
},
defaults:{
    blog: "",
    phone: "",
    city: "",
    department: "",
    email: "",
    firstName: "",
    id: null
},
```

View

- Declarative **model event** handler
- Cascaded **sub-views** handling
- Automated **view disposal** to prevent memory leakage

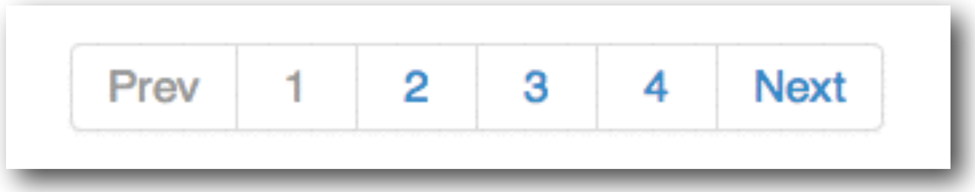
```
modelEvents: {
    "reset reportCollection" : "resetHandler"
},


resetHandler: function(){
 this.redraw();
},
```

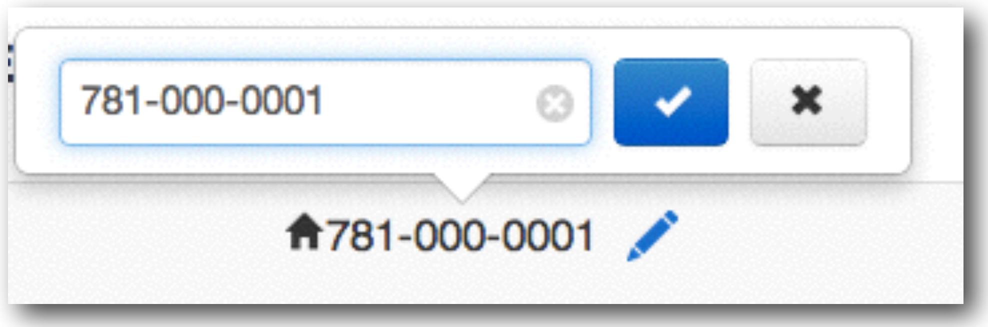
- Form validation and error handler
- Works with any client-side template technology
- Full RWD interface using Bootstrap

Examples on using various rich UI components

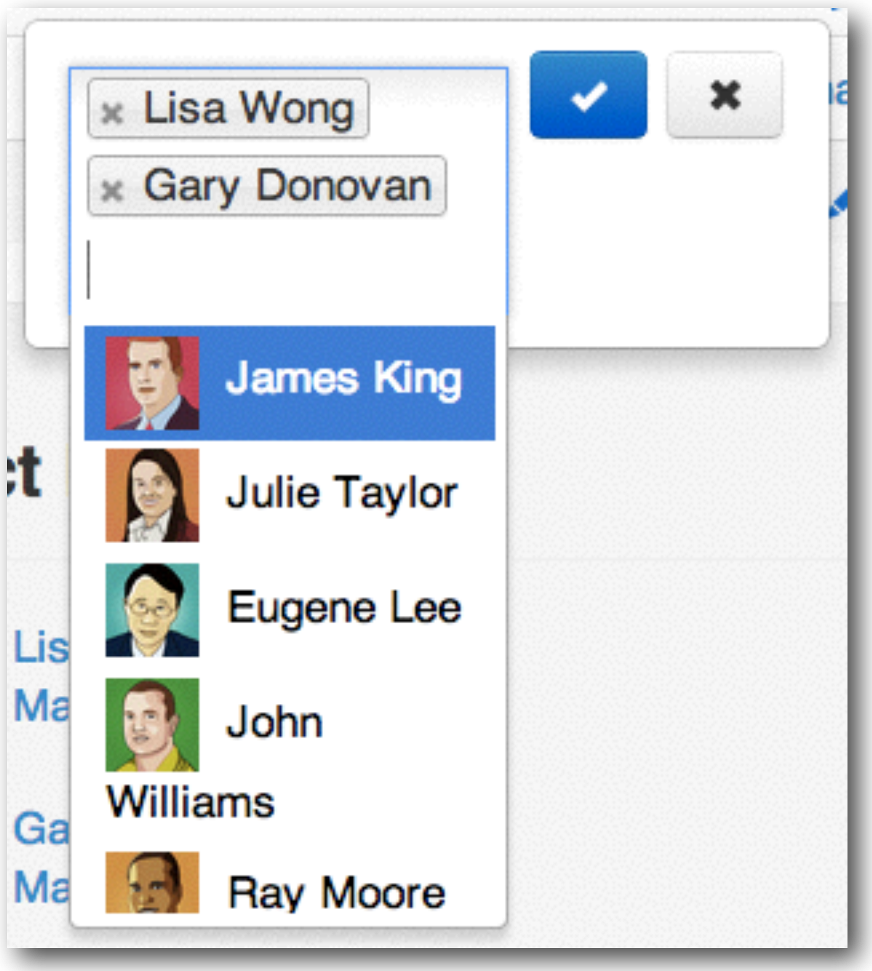
Pagination of large collection using "backbone.paginator" with bootstrap ui control



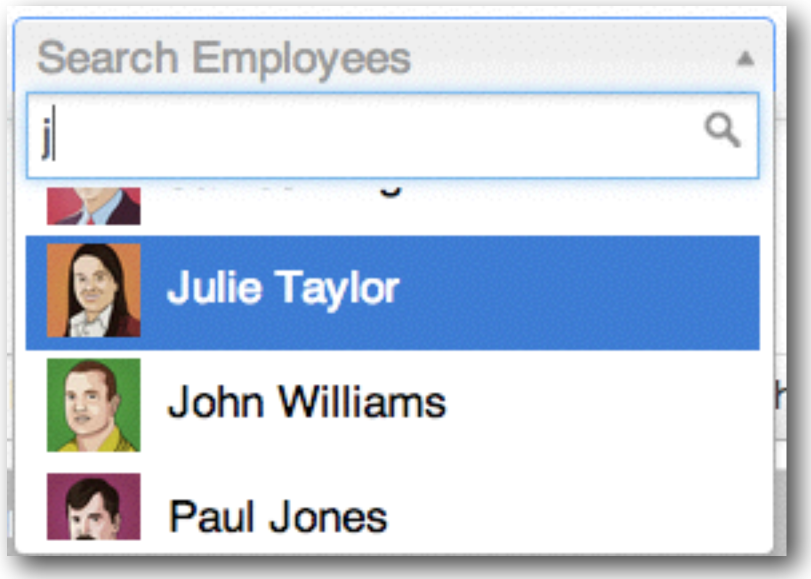
Form editing using "X-Editables" component



Better dropdown menus and lists using "Select2" component



Infinite scroll with paginated collection



Misc.

- Full OO approach (class inheritance using prototype)
- Examples on how to override model/collection Sync
- Controller example

- Customized Routing
- i18n/L10n support with LangManager using “polyglot” from Airbnb
- Drop-in library which provides many convenient utilities to make development easier

A few words on Angularjs

- It has most of the niceties of backbone.rocks built in
- Highly efficient and actively developed
- Comes with full testing support, including UI test based on Selenium/WebDriver
- **Very** steep learning curve, lack of training materials

- For new project, especially CRUD kind, go with angularjs
- For existing project, or DOM/UI heavy project, go with backbone.rocks
- We love angularjs, currently working on "Angularjs.rocks" book, will port the app too

About Us

- Jeremy Lu, founder of

  visual-marks.com
  pubulous.com
  lovelyreader.com

- Riaworks Consulting - professional training and consulting services
- 中文, 日本語 OK
- jeremy@pubulous.com

Q&A