



# Presented by: Rob Roach

CTO: [www.taggo.me](http://www.taggo.me)

@rjayroach

<https://github.com/rjayroach>

<http://rjayroach.wordpress.com>

# Why Would I Use It?

- Building Desktop-like experiences

# Conventional Web App Paradigm

- Client makes request
- Server replies – probably many times
- Result: relatively slow
- The further client is from server the slower it gets :-)

# Global App – Simple and Quick

- Imagine your app 80% is client side JS (SPA)
- With assets and script on a CDN your app is globally distributed with ease
- Local assets = happy users

# How to Accomplish this?

- JQuery – ouch!
- JS MVC Framework – cool!
- Which one? - scratch head

# How can Ember.js help me?

- Small choices are made for you, just like Rails
- Lets you solve more interesting problems
- Manages the complexity of a large application
- Stuff you didn't know you needed is already there
- Built by people scratching their own itch

# Hello World!

- Remember your first Rails app?
- Simple
- Quick
- Conventions



# gem 'ember-rails'

```
rails g ember:bootstrap
app/assets/javascripts/
|-- controllers
|   |-- application_controller.js.coffee
|   `-- entries_controller.js.coffee
|-- helpers
|-- models
|   `-- entries.js.coffee
|-- my_ember_app
|   `-- application.js
|-- my_ember_app.js
|-- router.js.coffee
|-- routes
|   `-- application_route.js.coffee
|-- store.js.coffee
|-- templates
|   |-- application.hbs
|   |-- entries.hbs
|   `-- new_entry.hbs
`-- views
    `-- new_entry_view.js.coffee
```

# Routing

```
window.App = Ember.Application.create();

// Router
App.Router.map(function() {
  this.resource('tables', function() {
    this.resource('table', { path: ':table_id' });
  });
});
```

# Models

```
App.Store = DS.Store.extend({  
  revision: 11,  
  adapter: 'DS.FixtureAdapter'  
  //adapter: 'DS.RESTAdapter'  
});
```

```
App.Table = DS.Model.extend({  
  tab: DS.belongsTo('App.Tab')  
});
```

# Controller

```
App.FoodController = Ember.ArrayController.extend({
  addFood: function(food) {
    var table = this.controllerFor('table').get('model'),
        tabItems = table.get('tab.tabItems');

    tabItems.createRecord({
      food: food,
      cents: food.get('cents')
    });
  }
});
```

# View Template – 2 way data binding

```
<div id="container">  
  <h1>Raffler</h1>  
  {{view Ember.TextField valueBinding="newEntryName"}}  
  {{newEntryName}}  
</div>
```

```
<ul>  
  {{#each entries}}  
    <li>{{name}}</li>  
  {{/each}}  
</ul>
```

# Result

- Processing on client
- Server becomes an API server
  - There is a project for that

# Caveats

- Ember-data – 0.13
- validations framework – no official one yet
- SEO – need workarounds

# Concerns

- Pace of Development
  - Ember 1.0 RC – Feb 15, 2013
  - Ember 1.0 RC5 – Jun 01, 2013
- Relevance - Stackoverflow
  - 1,500 as of November 08, 2012
  - 4,052 as of June 7, 2013
- Support - Team
  - Yehuda Katz – Committer to Rails, jQuery, Ember lead



# Concerns 2

- What about my Ruby / Rails skilz?
- JS MVC is a separation of concerns from server to client – inevitable?
- You augment, rather than obviate your Rails skilz – still need both

# ADOPTION

GROUPON

LivingSocial

Square

Zendesk

Discourse

Travis CI

# Resources

- Railscasts:
  - 2 episodes and 1 on `ActiveModel::Resources`
- Peepcode
- Emberwatch.com
- Emberjs.com