



Building Ambitious Web Applications

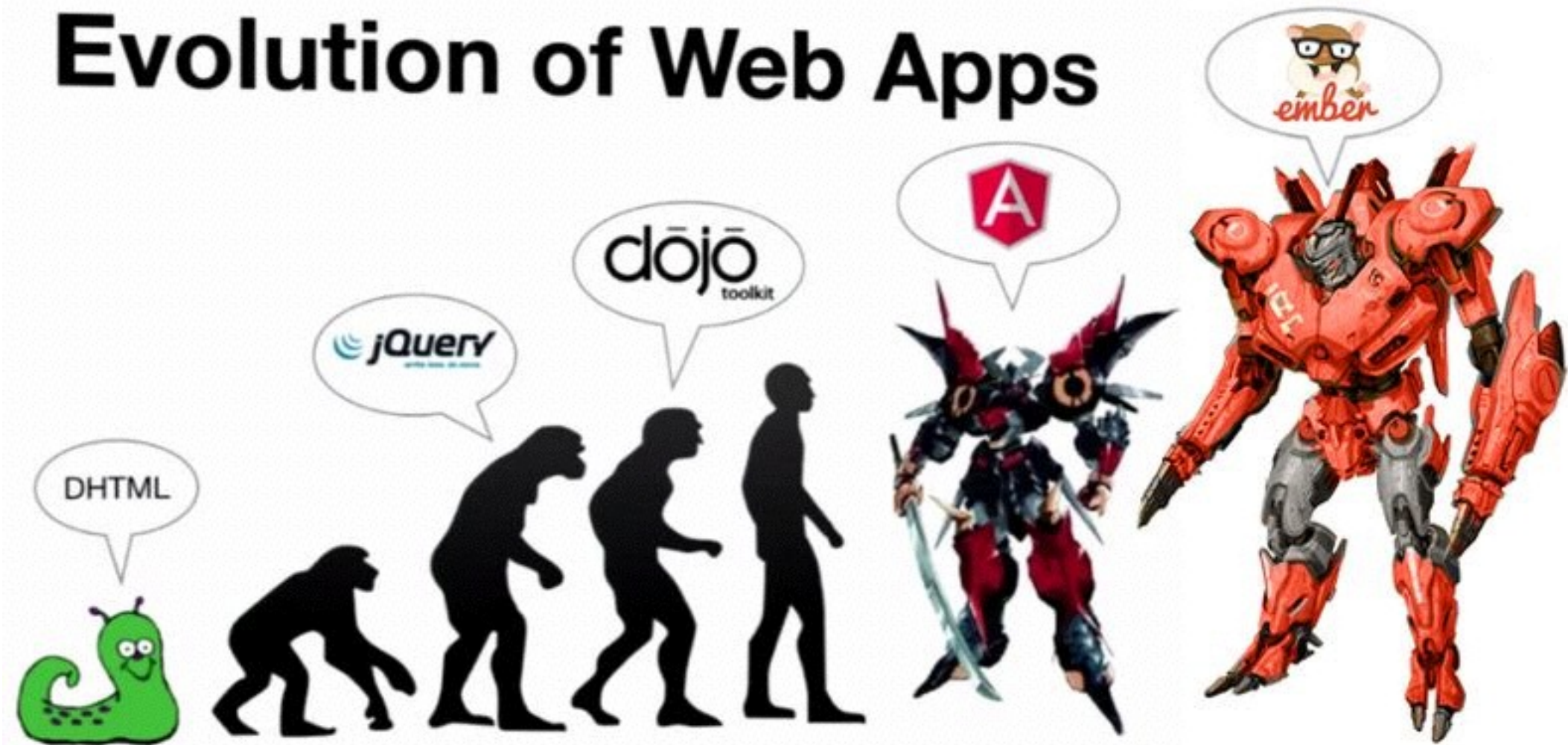
Ryan Anklam



- Senior UI Engineer at Netflix
- Friend of Open Source
- @bittersweetryan

Why Ember?

Evolution of Web Apps



Why Ember

ember

Strong conventions

URL's are first class citizens

Handlebars is the right
abstraction

Ember data

Why Ember





YAHOO!



NETFLIX



nest



Why Ember

ember

Core Concepts

Templates

Routes

Controllers

Components

Core Concepts

ember

Models

Core Concepts

ember

Let's Build
Something

Application & App.Router

Everything will be attached to this object.



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



```
App.Router.map(function() {
  // A resource
  this.resource('sessions', { path: 'sessions' }, function() {
    this.route('add');
  });

  // http://<server>/#/sessions
  // http://<server>/#/sessions/add
  // http://<server>/#/session/1
  this.route('session');

  // http://<server>/#/speakers
  // http://<server>/#/speakers/Ryan%20Anklam
  this.resource('speakers', { path: 'speakers' }, function() {
    this.route('speaker', { path:('/:name') });
  });
});
```

Templates

Handlebars + Ember

===

Powerful Templating

```
<script type="text/x-handlebars">
```

```
<header>
```

```
  <h1> Welcome to cf.Objective (2014) </h1>
```

```
</header>
```

```
<nav>
```

```
  <ul>
```

```
    <li class="nav-item">{{#link-to "/"}}Home</li>
```

```
    <li class="nav-item">{{#link-to 'sessions'}}Sessions{{/link-to}}</li>
```

```
    <li class="nav-item">{{#link-to 'speakers'}}Speakers{{/link-to}}</li>
```

```
  </ul>
```

```
</nav>
```

```
<div class="content">
```

```
  {{outlet}}
```

```
</div>
```

```
</script>
```

http://<server>/

http://<server>/#/sessions

http://<server>/#/speakers

Routes will output here

Loads in {{outlet}} when index route is active

```
<script type="text/x-handlebars" data-template-name="index">  
  <div class="welcome">  
    <p class="welcome-message">  
      This is an app to show off some of what is  
      cool in Ember.js  
    </p>  
  </div>  
</script>
```


Will load into `{{outlet}}` when a data promise is in flight

```
<script type="text/x-handlebars" data-template-name="loading">  
  <div class="loading">Loading...</div>  
</script>
```

Will load into {{outlet}} when a data promise is rejected

```
<script type="text/x-handlebars" data-template-name="error">  
  <div class="error">  
    There was an error loading this page.  
  </div>  
</script>
```

Loads in {{outlet}} when session route is active



```
<script type="text/x-handlebars" data-template-name="session">
  <h2 {{bind-attr class=goingClass}}>
    {{name}}{{#if going}}&nbsp;(Going){{/if}}
  </h2>
  <p>{{description}}</p>
  <button {{action 'going' this}} class="button">
    I'm {{#if going}}Not{{/if}} Going
  </button>
</script>
```

Routes

```
this.route( 'session', { path : 'session/:id' } );
```

```
App.SessionRoute = Ember.Route.extend({  
  model : function( params ){  
    return this.store.find( 'session', params.session_id );  
  }  
});
```



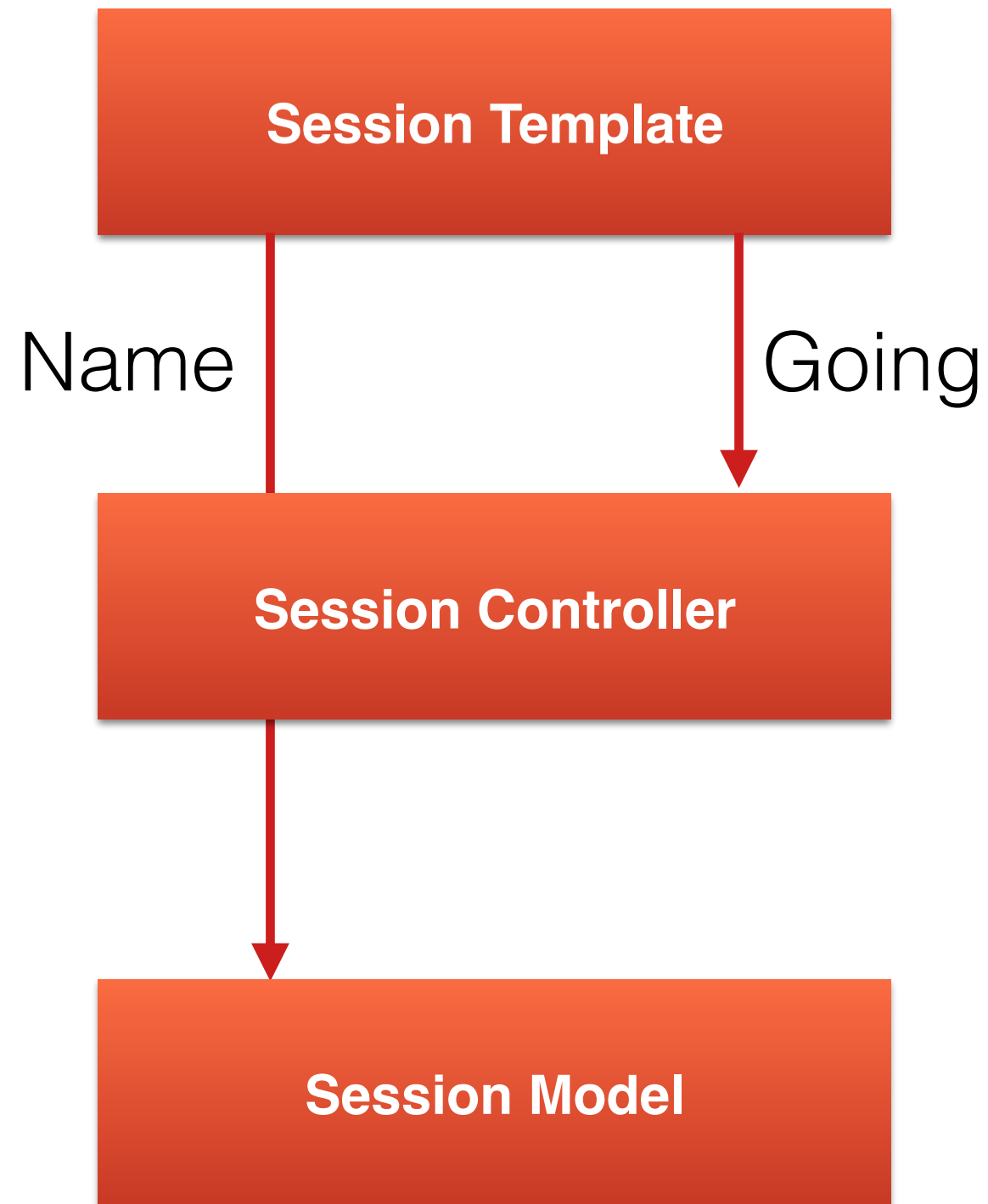
```
this.resource( 'sessions', { path : 'sessions' }, function(){  
  this.route( 'add' );  
} );
```

```
App.SessionsIndexRoute = Ember.Route.extend({  
  model: function() {  
    return this.store.find( 'session' );  
  }  
});
```

```
this.resource( 'sessions', { path : 'sessions' }, function(){  
  this.route( 'add' );  
} );
```

```
App.SessionsAddRoute = Ember.Route.extend({  
  model : function(){  
    return {  
      speakers : this.store.find( 'speaker' ),  
      tracks : ['JS', 'Architecture']  
    };  
  }  
});
```

Controllers



```
this.route( 'session', { path : 'session/:id' } );
```

```
<button {{action 'going' this}} class="button">
```

```
App.SessionController = Ember.ObjectController.extend({
```

```
  actions : {
```

```
    going : function( session ){
```

```
      session.toggleProperty( 'going' );
```

```
    }
```

```
  },
```

```
  goingClass : function(){
```

```
    return (this.get( 'going' ))? 'attending' : '';
```

```
  }.property( 'going' )
```

```
});
```

```
<h2 {{bind-attr class=goingClass}}>
```

```
App.SessionsIndexController.attending : function() {  
  return this.reduce( function( prev, curr ) {  
    if( curr.get( 'going' ) ) {  
      return prev + 1;  
    }  
    else {  
      return prev;  
    }  
  }, 0 );  
}.property( '@each.going' ),  
});
```

this is the collection of objects returned by the route

Tells the controller to update when any going property changes

```
App.SessionsAddController = Ember.ObjectController.extend({
  actions : {
    addSession : function(){
      var session = this.store.createRecord( 'session', {
        track : this.get( 'track' ),
        name : this.get( 'name' ),
        description : this.get( 'description' ),
        speaker : this.get( 'speaker' )
      } );
      this.transitionToRoute( 'session', session);
    }
  }
});
```

Redirect to the session route

Ember Data

- Swappable Adapters
- Relationships
- Fully embraces promises
- Data caching
- Fixtures speed up development


```
App = Ember.Application.create();  
App.ApplicationAdapter = DS.FixtureAdapter.extend();
```






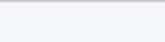

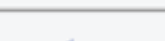
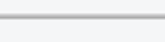
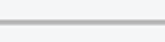
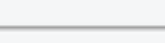
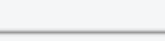
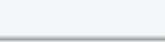
```
this.route( 'session', { path : 'session/:id' } );
```

```
App.Session = DS.Model.extend({  
  track : DS.attr( 'string' ),  
  name   : DS.attr( 'string' ),  
  description: DS.attr( 'string' ),  
  going: DS.attr( 'boolean', {defaultValue : false} ),  
  speaker : DS.belongsTo( 'speaker' )  
});
```

```
App.Speaker = DS.Model.extend({
  firstName  : DS.attr( 'string' ),
  lastName   : DS.attr('string'),
  fullName   : function(){
    return
      this.get( 'firstName' ) + ' ' + this.get( 'lastName' );
  }.property( 'firstName', 'lastName' ),
  bio        : DS.attr( 'string' ),
  sessions: DS.hasMany('session', { async : true } )
});
```

Components

No, Seriously.
Components.

Cluster	RPS Trend (3 min)	RPS		Error %		Per Instance RPS		Load Average		Latency	Instances
vhs-stateless vhs-us-east-1 vhs-stateless-v466		1	-14.1%	100	+0%	0.1	-6.3%	0	-11.6%	50% 36 ms 99% 43 ms	Up: 4 Out of Service: 0 Starting: 0 Down: 0 N/A: 0
vhs-int vhs-us-east-1 vhs-int-v036		1	-16.5%	100	+0%	0.4	-0.6%	0.1	-4.3%	50% 13 ms 99% 60 ms	Up: 2 Out of Service: 0 Starting: 0 Down: 0 N/A: 0
vhs vhs-us-east-1 vhs-v374		1.1	-8.6%	100	+0%	0.2	-13.1%	0.1	-6.7%	50% 9 ms 99% 17 ms	Up: 2 Out of Service: 0 Starting: 0 Down: 0 N/A: 0
api-testdata api-us-east-1		2.1	-13.6%	100	+0%	0.3	-6.4%	0	-13%	50% 3 ms 99% 6 ms	Up: 1 Out of Service: 0 Starting: 0 Down: 0 N/A: 0
api-ci-dependency-smoke api-us-east-1		2.2	-7.4%	100	+0%	0.1	-12.3%	0.1	-10.6%	50% 2 ms 99% 2 ms	Up: 1 Out of Service: 0 Starting: 0 Down: 0 N/A: 0
apiproxy-dev apiproxy-us-east-1		4	-13.9%	100	+0%	0.1	-14%	0.3	+0%	50% 7 ms 99% 16 ms	Up: 1 Out of Service: 0 Starting: 0 Down: 0 N/A: 0
api-perflab api-us-east-1		11.4	-14.3%	100	+0%	0.2	-4.4%	0.5	-14.6%	50% 1 ms 99% 2 ms	Up: 2 Out of Service: 1 Starting: 0 Down: 0 N/A: 0
api-test api-us-east-1		3.6	-3.1%	25.6	-16.3%	1.2	-11.5%	0.2	-9.9%	50% 84 ms 99% 3440 ms	Up: 3 Out of Service: 0 Starting: 1 Down: 0 N/A: 0
api-int api-us-east-1		1.8	-7.4%	8.5	-10.5%	0.6	-4.4%	0	+0%	50% 175 ms 99% 1182 ms	Up: 3 Out of Service: 0 Starting: 3 Down: 0 N/A: 0
apiproxy-website apiproxy-us-east-1		3.6	-11%	11.2	-13.9%	3.6	-16.2%	0.2	-8.3%	50% 80 ms 99% 1296 ms	Up: 1 Out of Service: 0 Starting: 0 Down: 0 N/A: 0
apiproxy-int apiproxy-us-east-1		1.1	-1.5%	0.6	-11.3%	1.1	-2%	0.1	-2.4%	50% 30 ms 99% 198 ms	Up: 1 Out of Service: 0 Starting: 0 Down: 0 N/A: 0
apiproxy-test apiproxy-us-east-1		11.3	-9.5%	1.3	-5.5%	3.8	-13.9%	0.1	-12.2%	50% 13 ms 99% 281 ms	Up: 3 Out of Service: 0 Starting: 0 Down: 0 N/A: 0
vhs-stateless.int vhs-us-east-1 vhs-stateless.int-v037		96.3	-0.9%	0.1	-1.7%	16.1	-0.4%	0.7	-14.5%	50% 80 ms 99% 987 ms	Up: 6 Out of Service: 0 Starting: 0 Down: 0 N/A: 0

Built with components

ember

```
<script type="text/x-handlebars" data-template-name="components/speaker-bio">
```

Becomes component name

Properties are encapsulated

```
<article class="speaker-bio">
  <h2 class="speaker-name">{{name}}</h2>
  <section class="speaker-description">
    {{bio}}
  </section>
  {{#if sessions}}
    <h3>Sessions</h3>
    <ul>
      {{#each session in sessions}}
        <li>{{session.name}}</li>
      {{/each}}
    </ul>
  {{/if}}
</article>
</script>
```

```
<script type="text/x-handlebars" data-template-name="speakers">
  <h1>Speakers</h1>
  {{#each
    Add the component
    Pass properties into component
    {{#with item}}
      {{speaker-bio name=fullName bio=bio sessions=sessions}}
    {{/with}}
  }}
  {{/each}}
</script>
```

```
App.SpeakerBioComponent = Ember.Component.extend({  
  actions: {  
    toggleBody: function() {  
      this.toggleProperty('isShowingBody');  
    }  
  }  
});
```

Getting Help

Ember Inspector

Why Ember



<http://discuss.emberjs.com/>

IRC: [#emberjs](https://freenode.net)

Looking Ahead

Ember CLI

(currently in beta)

Looking Ahead

ember

HTMLBars

(currently in alpha)

Ember-Data 1.0

(currently in beta)

Why Ember

The Ember logo, featuring the word "ember" in a white, lowercase, cursive script font, positioned in the bottom right corner of the slide.