WordPress & Backbone.js

The Dawn of Web Apps
@JesperBylund
User Experience Developer
Odd Alice
WordPress & BackBone.js

The Dawn of Web Apps
In the beginning there was static content
Suddenly, mobile emerged
Everything became function and really cool form
Erik Eng  →  Öppen ateljé
3 mins · Stockholm

http://instagram.com/p/XeomnJprNb/
3 Comments

JesperBylund.com

Join the discussion...

Jake Tracey  •  4 months ago
I think you're on the right track except for the bit of technology already exists, it's iCloud.

To pull off actual seamless user switching the Apple TV would need much better hardware + a camera. For now $100 for a decent Netflix client + AirPlay streaming is a steal, if only it was Jailbroken.

Jesper Bylund  Mod  •  Jake Tracey  •  4 months ago
Thank you!

I'm not worried about the infrastructure solution, Apple will pull it off, and as you said; probably in the back end. But I am worried about the UI.
There's simply no easy way to manage the number of users that might be involved on a couch.

Jake Tracey  •  Jesper Bylund  •  4 months ago
I've only been using it for a couple of weeks, but I think the Xbox One has this covered from a UX perspective.

The combination of a camera + facial recognition + voice commands works amazingly well.

Come home from work, Say "Xbox on" and it turns on everything and says "Hey Jake" once it recognises my face.
JavaScript eats the world
WordPress + Backbone.js
BackBone.js
JavaScript framework for RESTful JSON APIs
Model + View + Router
Model
+
View
Model: JSON object
View: functions and representations for the model
WordPress
(JSON API)
What can WordPress serve?
Everything.
<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/sites/$site/categories/slug:$category</td>
</tr>
<tr>
<td>POST</td>
<td>/sites/$site/categories/slug:$category</td>
</tr>
<tr>
<td>GET</td>
<td>/sites/$site/categories</td>
</tr>
<tr>
<td>GET</td>
<td>/sites/$site/tags</td>
</tr>
<tr>
<td>GET</td>
<td>/sites/$site/tags/slug:$tag</td>
</tr>
<tr>
<td>POST</td>
<td>/sites/$site/tags/slug:$tag</td>
</tr>
<tr>
<td>POST</td>
<td>/sites/$site/categories/new</td>
</tr>
<tr>
<td>POST</td>
<td>/sites/$site/tags/new</td>
</tr>
<tr>
<td>POST</td>
<td>/sites/$site/categories/slug:$category/delete</td>
</tr>
<tr>
<td>POST</td>
<td>/sites/$site/tags/slug:$tag/delete</td>
</tr>
</tbody>
</table>

WP API delivers models to Backbone.js
BackBone.js renders templates from Models
WordPress templates as backwards compatibility?
LoL!
Nothing works without JS.
So how can we **use** it?
Demo Time
Collection (model)
// Post model
var Post = Backbone.Model.extend({
    defaults: {},
    idAttribute: 'ID',
});
// collection or loop of posts

var Posts = Backbone.Collection.extend({
    model: Post,
    initialize: function () {
        this.fetch({
            success: this.fetchSuccess,
            error: this.fetchError
        });
        this.deferred = new $.Deferred();
    },
    deferred: Function.prototype.constructor.prototype,
    fetchSuccess: function (collection, response) {
        collection.deferred.resolve();
    },
    fetchError: function (collection, response) {
        throw new Error("Fetch did not get collection from API");
    },
    parse: function( response ) {
        return response.posts;
    }
});
// Posts view
var PostsView = Backbone.View.extend({
  el: '#posts',

  events: {
    'click span.read': 'showStory',
  },

  initialize: function(){
    _.bindAll(this, 'createPost', 'showMore');
    this.$el.html('<h3>Loading posts</h3>');
    this.collection = new Posts();
    this.collection.bind('add', this.createPost);
    this.render();
  },

  render: function(){
    _(this.collection.models).each(function(post){ // in case collection
      this.createPost(post);
    }, this);
    return this;
  },

  createPost: function(post){
    var postView = new PostView({
      model: post
    });
    $('h3', this.$el).hide();
    this.$el.append( postView.render().el );
  },

  showMore: function(){
    console.log('showMore');
  }
});
But what can we build with this?
Anything WordPress couldn’t do
Questions?

@jesperbylund