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

Jesper Bylund
Edit Profile

News Feed
Messages
Events 5
Close Friends 20+
Family 7
Adding the Fun
Jesper Bylund
Projekt Svensexa

PAGES
Shrtnws 3
Blankpage.io
Odd Alice 3
#wpbar 5
#wpsthlm 5
Pages Feed 20+
Like Pages 20+
Create Ad

GROUPS
Skösex Alumni 3
Swedish Growth H... 6
Krunch i Stockholm 8
Bubblan 5
30 FESTER FRA... 20+
Nextstartups

Update Status | **Add Photos/Video**

What's on your mind?

Viewing most recent stories · Back to top stories

Erik Eng ▸ **Öppen ateljé**
3 mins · Stockholm ·

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

Jesper Wendel's birthday is today

Mys Långfreda on April 18

SPONSORED

Vinn en resa till Forks
Platsen där Twillight utspelar...

Se hela Twilight sag
TV6! Alla filmer und
kvällar, 15-19 april 2
http://bi...

SC 2012 R2 - Geek Week
labcenter.se

Lär dig deploya,
managera & hantera
Windows klient/Serv
System Center. 19-2
Sthlm

Vi söker drivna Campaign...
keybroker.se

Vi söker fler Campa
Analysts. Skicka din
ansökan till
jobs@keybroker.com

Resan mot drömjobbet...
mdh.se

Läs mer om våra
utbildningar och var
våra studenter är att
på arbetsmarknaden

Hemavan Tärnaby®

3 Comments JesperBylund.com Login

Sort by Oldest Share Favorite

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.

· Reply · Share ›

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.

· Reply · Share ›

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

Single Page **Apps**

Model

+

View

+

Router

Model

+

View

Model: JSON object

View: **functions** and **representations** for the model

WordPress (JSON API)

What can WordPress serve?

Everything.

GET /sites/$site/posts/

GET /sites/$site/posts/$post_ID

POST /sites/$site/posts/$post_ID

GET /sites/$site/posts/slug:$post_slug

POST /sites/$site/posts/new

POST /sites/$site/posts/$post_ID/delete

GET /sites/$site/posts/$post_ID/likes/

POST /sites/$site/posts/$post_ID/likes/new

POST /sites/$site/posts/$post_ID/likes/mine/delete

GET /sites/$site/posts/$post_ID/likes/mine/

GET /sites/$site/posts/$post_ID/reblogs/mine

POST /sites/$site/posts/$post_ID/reblogs/new

POST /sites/$site/posts/$post/related

What can WordPress serve?

Everything.

GET /sites/$site/comments/

GET /sites/$site/posts/$post_ID/replies/

GET /sites/$site/comments/$comment_ID

POST /sites/$site/comments/$comment_ID

POST /sites/$site/posts/$post_ID/replies/new

POST /sites/$site/comments/$comment_ID/replies/new

POST /sites/$site/comments/$comment_ID/delete

What can WordPress serve?

Everything.

GET /sites/$site/categories/slug:$category

POST /sites/$site/categories/slug:$category

GET /sites/$site/categories

GET /sites/$site/tags

GET /sites/$site/tags/slug:$tag

POST /sites/$site/tags/slug:$tag

POST /sites/$site/categories/new

POST /sites/$site/tags/new

POST /sites/$site/categories/slug:$category/delete

POST /sites/$site/tags/slug:$tag/delete

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,
    url: 'https://public-api.wordpress.com/rest/v1/sites/jesperbylund.
        com/posts/?number=10&callback=?',
    initialize: function () {
        this.fetch({
            success: this.fetchSuccess,
            error: this.fetchError
        });
        this.deferred = new $.Deferred();
    },
    deferred: Function.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;
    }
});
```

View

```
// 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
        is not empty
            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');
    }

});
```

```
// post view
var PostView = Backbone.View.extend({
    tagName: 'div',
    className: 'post',
    events: {
        'click span.delete': 'unrender',
        'click h2': 'showPost',
    },

    initialize: function(){
        _.bindAll(this, 'remove', 'render','unrender', 'showPost');
        this.model.bind('change', this.render);
    },

    render: function(){
        this.$el.html('<div><img class="featured-image" src="'+this.model.get('
            featured_image')+'" />
                        <h2>'+this.model.get('title')+'</h2></div>');
        return this;
    },

    unrender: function(){
        $(this.el).remove();
    },

    showPost: function(){

        $('div.content').removeClass('active');
        if( $('div.content', this.$el).length > 0 ){
            $('div.content', this.$el).addClass('active');
        } else {
            $('div', this.$el).append('<div class="content active">'+this.model
                .get('content')+'</div>' );
        }
        $('html, body').animate({
            scrollTop: $('h2', this.$el).offset().top
        }, 500);

    }
});
```

But what can we **build**
with this?

Anything WordPress couldn’t do

Questions?

@jesperbylund