

Navigating the Front End Ocean

Sam Farmer

Me

Senior Applications Architect at Markel FirstComp

Developer since 1997

Live in Washington, DC with wife and two daughters

Play soccer weekly and big Arsenal fan

Agenda

- Will cover:
 - What is Responsive Design, Grids, CSS Compliers and Templating?
 - Differences between AJAX and Websockets
 - What type of application to create, light or heavy client, and when
 - When to use particular JavaScript frameworks and libraries
 - Creating iOS and Android applications
- Other good sessions at this time

You

- How many have:
 - Used jQuery?
 - Sprinkled JavaScript on to a page?
 - Built a "single-page" app?

3 questions...

What are you building?

What devices?

What skills do you have?

Three types today

HTML with JavaScript

JavaScript on HTML

Just JavaScript

Techniques

HTML5 & CSS

Grids

Divide page into grid

```
<div class="row">
  <div class="span4">The left</div>
  <div class="span4">The middle</div>
  <div class="span4">The right</div>
</div>
```

```
.span5 { float: left; width: 380px; }
.span4 { float: left; width: 300px; }
.span3 { float: left; width: 220px; }
```

Media Queries

Rules in CSS

Responsive Design: Adjust page based on device size

Works well with Grid based systems

```
@media (max-width: 767px) {  
    .row [class*="span"] {  
        display: block;  
    }  
}
```

CSS Processors

Variables, functions in CSS!

Compile in browser or on server

```
@font-size: 1em;  
@dark-color: Black;  
@light-color: Yellow;
```

```
html, body { font-family: Arial; }
```

```
h1 { font-size: (@font-size*2); color: @dark-color; }
```

```
.headerBar { font-size: (@font-size*1.3); color: @dark-color; }
```

Technique	HTML with JavaScript	JavaScript on HTML	Just JavaScript
Media Queries	Bootstrap Foundation Less Framework		
Grid	Bootstrap Foundation Less Framework		
CSS Processors	LessCSS SASS		

Writing HTML

Writing with JS / jQuery

- This quickly becomes a pain!

Templating

mustache.js

```
<script id="template" type="text/html">
  {{#data}}
    <tr>
      <td>{{name}}</td>
      <td>{{city}}</td>
      <td>{{rate}}</td>
      <td>{{rooms}}</td>
    </tr>
  {{/data}}
</script>
```

```
$.getJSON("some.json", function( data ) {
  var template = $("#template").html()
  var html = Mustache.to_html(template, { data: data })
  $(".output").html( html )
});
```

PURE

```
<tbody>
  <tr>
    <td class="name"></td>
    <td class="city"></td>
    <td class="rate"></td>
    <td class="rooms"></td>
  </tr>
</tbody>
```

```
var directive = {
  'tbody tr' : {
    'row<-context': {
      'td.name' : 'row.name',
      'td.city' : 'row.city',
      'td.rate' : 'row.rate',
      'td.rooms': 'row.rooms'
    }
  };
  $(".output").render( data, directive )
```

Just JavaScript

- No HTML
- Define everything in JavaScript

Technique	HTML with JavaScript	JavaScript on HTML	Just JavaScript
Templating		mustache PURE handlebars underscore EJS	
Components			ExtJS Sencha Touch YUI dojo SproutCore Cappuccino

Navigating

Event Listeners

- Bind callback to when user performs an action

```
<button id="buttonOne">Vanilla JavaScript Example</button>
```

```
document.getElementById("buttonOne").addEventListener("click", function()
  console.log( "Vanilla Ice Cream is better")
});
```

```
<button id="buttonTwo">jQuery Example</button>
```

```
$("#buttonTwo").on("click", function() {
  console.log("jQuery simplifies")
});
```

Routing / Deep Linking

- # and on stays in the browser
- Listen to changes and react
- REST Inspired
- Built into lots of frameworks

<http://www.site.com/#/user/200>

Technique

HTML with
JavaScript

JavaScript on HTML

Just JavaScript

Routing

sammy
backbone
Ember
...

ExtJS
Sencha Touch
YUI
dojo
SproutCore
Cappuccino

Event
Listeners

jQuery
Angular

ExtJS
Sencha Touch
YUI
dojo
SproutCore
Cappuccino

Server Communication

AJAX

- Advantages
 - Works over port 80 and 443 (SSL)
 - Cross domain possible
 - Integrated in plenty of frameworks
- Disadvantages
 - New connection each time
 - Can only poll not receive a request

Websockets

- Advantages
 - Immediacy - operates over TCP
 - Smaller data packets
 - Push and pull
- Disadvantages
 - Average browser and port support

Technique	HTML with JavaScript	JavaScript on HTML	Just JavaScript
AJAX	jQuery zepto xui	jQuery zepto xui	ExtJS Sencha Touch YUI dojo SproutCore Cappuccino
Websockets	ColdFusion, Java, PHP, .NET, Ruby, Python, etc	ColdFusion, Java, PHP, .NET, Ruby, Python, etc	ColdFusion, Java, PHP, .NET, Ruby, Python, etc

Data Storage

Form

- Useful when have no other need for data

Recordsets

- Useful when:
 - no other need for data
 - huge amounts of data and better for pagination

Models

- Define model and relationship in JavaScript
 - Frameworks keep in sync with server
 - Problem keeping in sync when data is volatile
- Very performant when have multiple needs for data

Browser Storage

- Useful for user preference data
- Useful when no back end
 - key/value or mini-database
- Mobile apps / games

Technique	HTML with JavaScript	JavaScript on HTML	Just JavaScript
Form	jQuery zepto xui	jQuery zepto xui	
Recordset	jQuery zepto xui	jQuery zepto xui	
Model		Backbone Ember Angular dojo	ExtJS Sencha Touch YUI dojo SproutCore Cappuccino

Framework Patterns

MV*

- Back end favorite is MVC
- Front end MV*
- <http://todomvc.com/>

todos

How does this get added?

Double-click to edit a todo

Credits: [Christoph Burgdorf](#), [Eric Bidelman](#), [Jacob Mumm](#) and [Igor Minar](#)

Part of [TodoMVC](#)

Backbone

html

```
<header id="header">
  <h1>todos</h1>
  <input id="new-todo" placeholder="What needs to
be done?" autofocus>
</header>
```

Backbone

js/views/app.js

```
app.AppView = Backbone.View.extend({
  events: {
    'keypress #new-todo': 'createOnEnter',
    'click #clear-completed': 'clearCompleted',
    'click #toggle-all': 'toggleAllComplete'
  },
  createOnEnter: function (e) {
    if (e.which !== ENTER_KEY || !this.$input.val().trim()) {
      return;
    }
    app.Todos.create(this.newAttributes());
    this.$input.val('');
  }
});
```

Angular

html

```
<header id="header">
  <h1>todos</h1>
  <form id="todo-form" ng-submit="addTodo()">
    <input id="new-todo" placeholder="What needs to
be done?" ng-model="newTodo" autofocus>
  </form>
</header>
```

Angular

js/controllers/todoCtrl.js

```
todomvc.controller('TodoCtrl', function TodoCtrl($scope, $location,  
todoStorage, filterFilter) {  
  var todos = $scope.todos = todoStorage.get();  
  $scope.addTodo = function () {  
    if (!$scope.newTodo.length) {  
      return;  
    }  
    todos.push({  
      title: $scope.newTodo,  
      completed: false  
    });  
    $scope.newTodo = '';  
  };  
});
```

Ember

html

```
<header id="header">
  <h1>todos</h1>
  {{view Ember.TextField id="new-todo"
    placeholder="What needs to be done?"
    valueBinding="newTitle" action="createTodo"}}
</header>
```

Ember js/controllers/todos_controllers.js

```
Todos.TodosController = Ember.ArrayController.extend({
  createTodo: function () {
    var title = this.get('newTitle');
    if (!title.trim()) {
      return;
    }
    Todos.Todo.createRecord({
      title: title,
      isCompleted: false
    });
    this.set('newTitle', '');
    // Save the new model
    this.get('store').commit();
  }
})
```

YUI html

```
<header id="header">
  <h1>todos</h1>
  <input id="new-todo" placeholder="What needs to be
done?" autofocus>
</header>
```

YUI

js/app.js

```
TodoApp = Y.Base.create('todoApp', Y.App, [], {
  events: { '#new-todo': { keypress: 'enterCreate' } }
  enterCreate: function (e) {
    var ENTER_KEY = 13;
    var todoList = this.get('todoList');
    var inputNode = this.get('inputNode');
    var value = inputNode.get('value');
    if (e.keyCode !== ENTER_KEY || !value) {
      return;
    }
    todoList.create({
      title: value
    });
    inputNode.set('value', '');
  }
})
```

Framework Make Up

	Backbone	Angular	Ember	YUI
jQuery?	Depends On	Can use	Depends On	Nope
Other	underscore		handlebars	

Which and when?

HTML with JavaScript

JavaScript on HTML

Just JavaScript

Websites
Single form

Applications with framework

HTML Feel

Desktop Feel

Free Open Source

Cost

Complexity

Beyond the Browser

Webview

- Take HTML / JS
- Styling options from native-like to own
- Access native features via JavaScript API
- Package in a shell using PhoneGap or Sencha Touch

Conversion to Native

- Converts JavaScript to a native app

```
var addBtn = Ti.UI.createButton( { title:'+' } );
addBtn.addEventListener('click', function() {
    new AddWindow().open();
});
```

```
var addButton = Ti.UI.createButton({
    title: 'Add', width: '300dp', height: '40dp', top: '80dp'
});
addButton.addEventListener('click', function() {
    addTask(itemField.value, self);
});
```

Technique

JavaScript on HTML

Just JavaScript

Webview

Phonegap
Appcelerator

Phonegap
Appcelerator

Native

Appcelerator

Appcelerator

Style like Native

jQuery Mobile UI

Sencha Touch

Testing

Unit Testing

47+

qUnit

```
test( "The first test", function() {
    var first = 1;
    ok( first == "1", "Passed!" );
});
```

```
test( "The second test", function() {
    var first = 1;
    strictEqual( first, "1", "Passed!" );
});
```

```
test( "The third test", function() {
    var first = 1;
    strictEqual( first, 1, "Passed!" );
});
```

Selenium

- Browser tests
- Options for automation

Teenagers

and Backward Compatibility

1

Sammy	0.7
Cappuccino	0.9
mustache	0.7
SproutCore	1.10
dojo	1.9
zepto	1
Angular	1

2

jQuery	2
PURE	2.7
Sencha Touch	2.2
xui	2.3

4

backbone	1
LessCSS	1.3
jQuery	1.9
handlebars	1
EJS	1
Ember	1
PhoneGap	1.7

Avg

1.9

Bootstrap

YUI

Titanium

4

Foundation

4

Less Framework

Thank you!

Q&A

@sam_farmer
sam@samfarmer.com
www.samfarmer.com