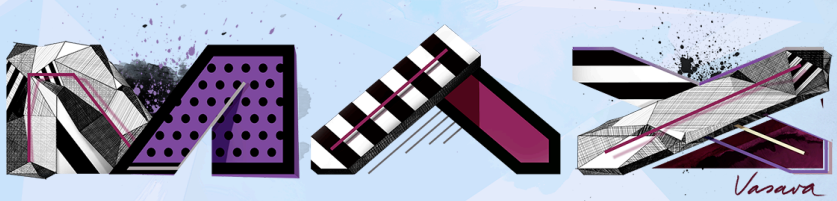
Architecting a PhoneGap Application

Christophe Coenraets @ccoenraets



THE CREATIVITY CONFERENCE

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Christophe Coenraets
@ccoenraets
http://coenraets.org

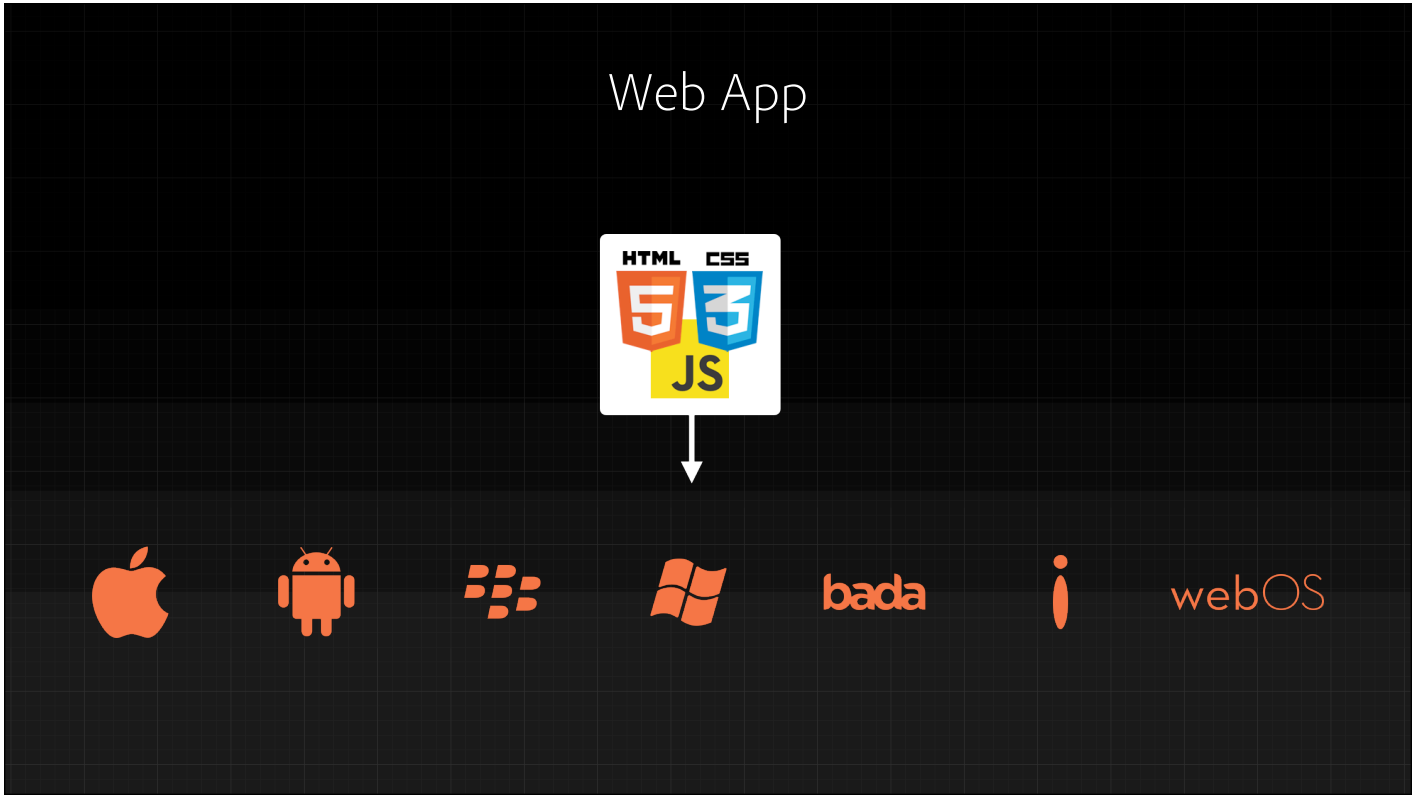
© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

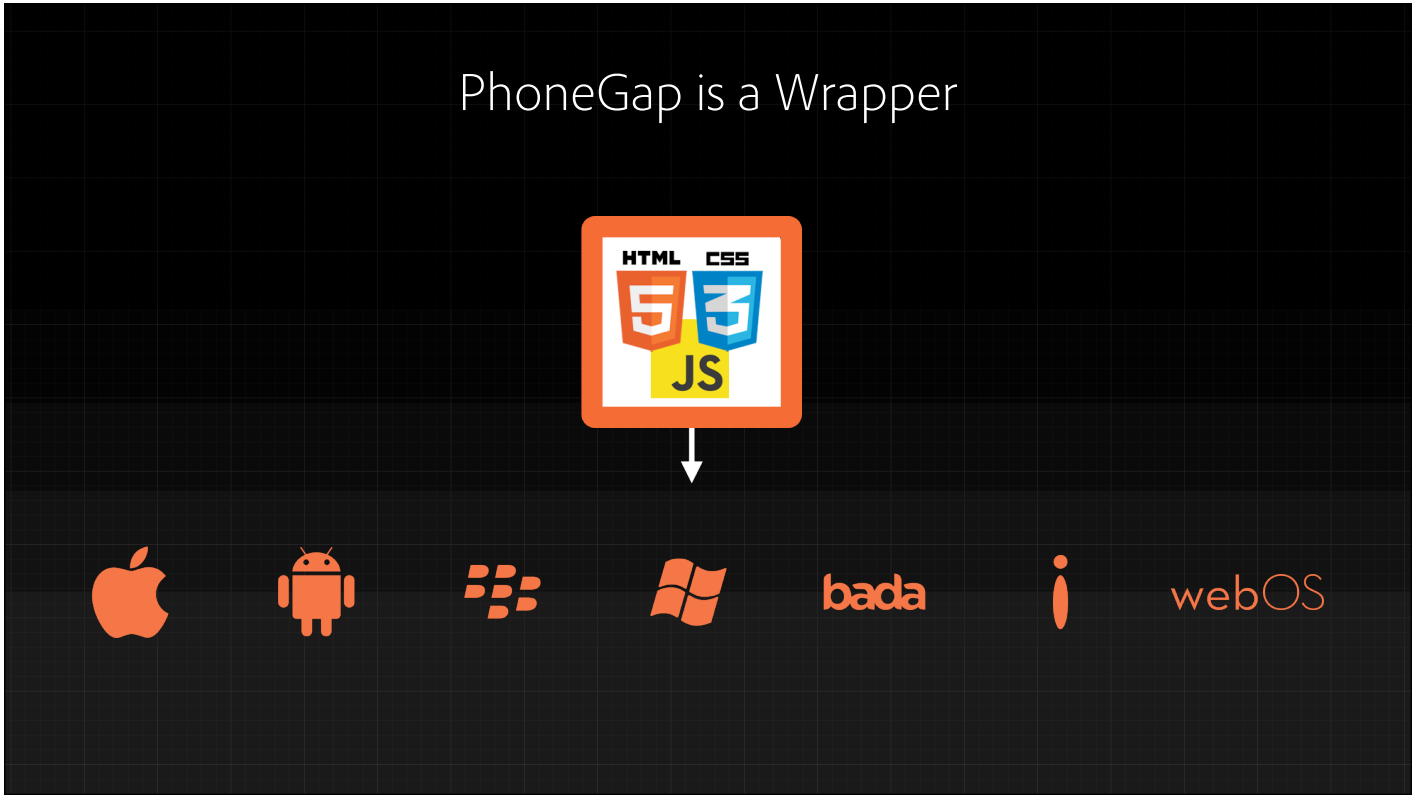
Agenda

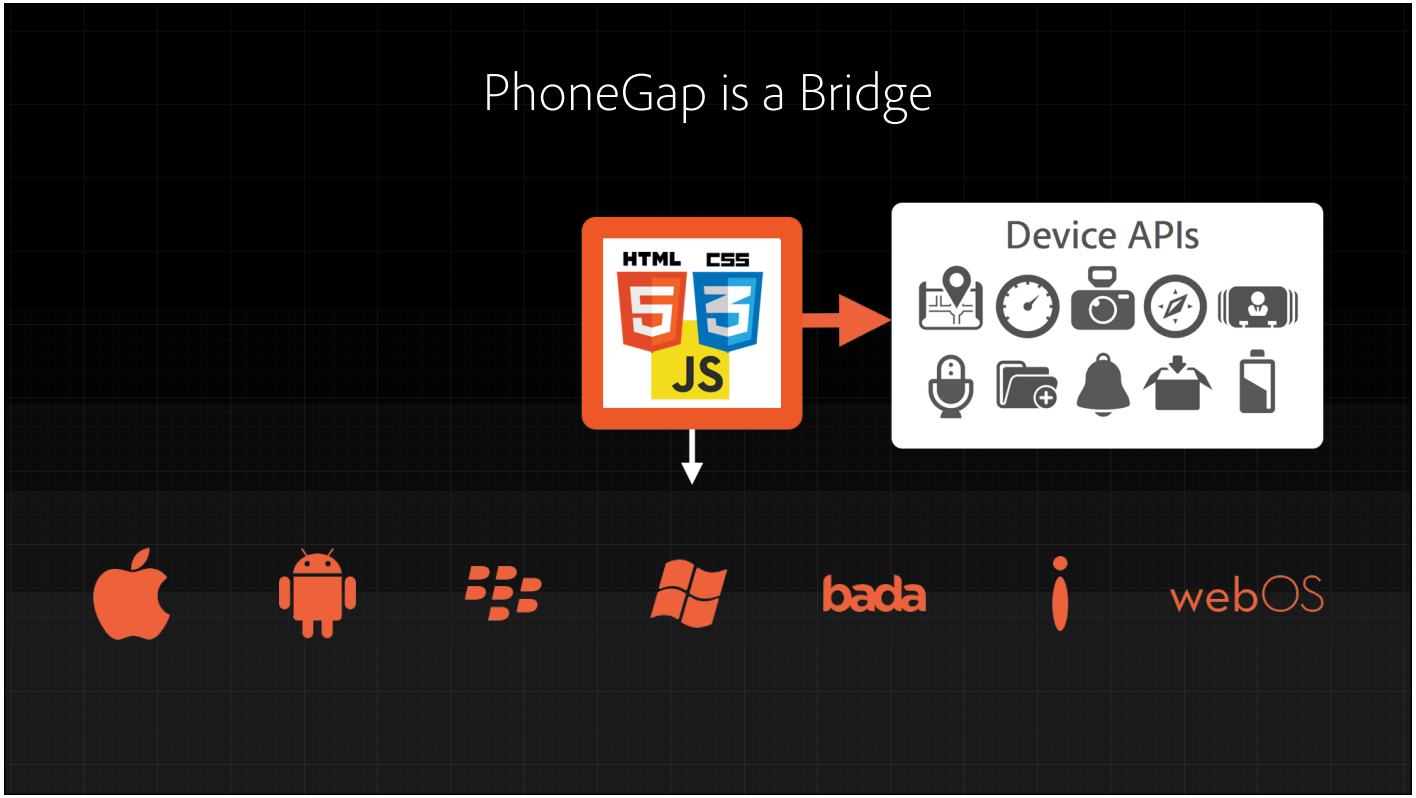
10 architectural principles

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.


Web App
HTML
CSS
5
3
JS
bada
webOS

PhoneGap is a Wrapper




PhoneGap is a Bridge
HTML
CSS
JS
Device APIs
bada
webOS

	Hybrid	Native
Skills	HTML, JS, CSS	Obj C, Java, C/C++
Cross Platform	Yes	No
Device APIs	Yes	Yes
Distribution	App Store	App Store
Updates	App Store + Instant	App Store
Performance	Fast	Faster

Data Access

```
$.ajax({url: "/api/employee/3"}).done(function(employee) {
   //Do something with employee
});
```

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

#1

Abstract Data Access

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

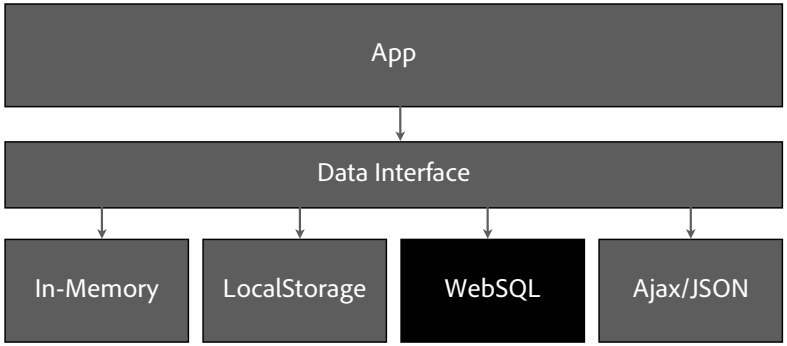
Abstract Data Access

```
dataAdapter.findById(3).done(function(employee) {
   //Do something with employee
});
```

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Pluggable Data Adapters

- Common API
- Asynchronous



© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Benefits

- Experience first!
- Fast iterative prototyping
- Testable

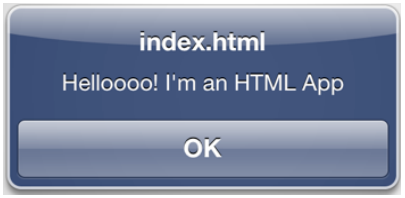
© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

#2

Keep your application "browser runnable"

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Keep Your Application Browser Runnable



- Default JavaScript alert gives away the fact that your application is not native
- PhoneGap API:

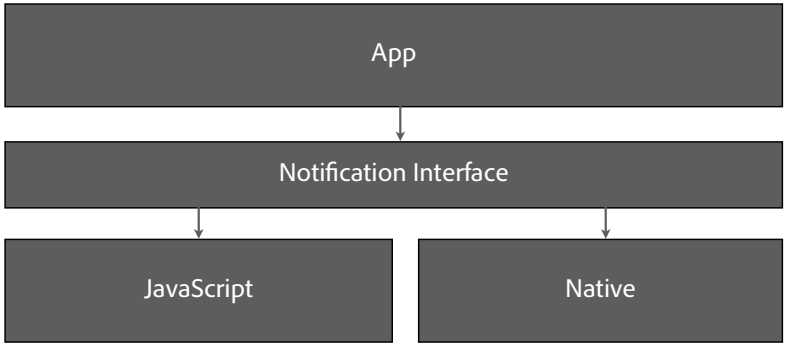
```
navigator.notification.alert(message, alertCallback, [title], [buttonName])
```

... but that doesn't work in your browser

- Solution: Build an abstraction layer
  - Display JavaScript alert when running in the browser
  - Display Native alert when running on device

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

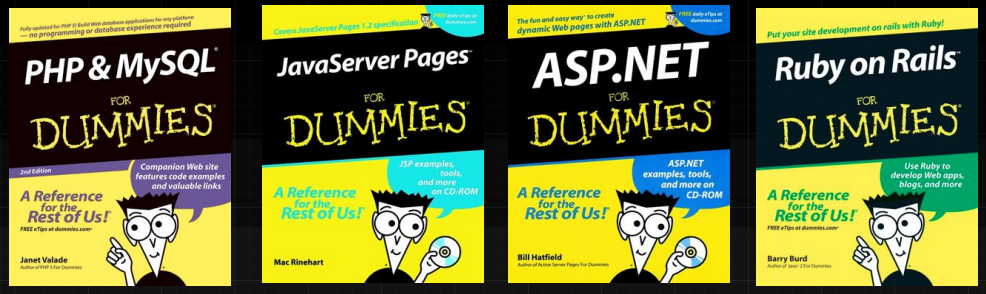
Notification Interface



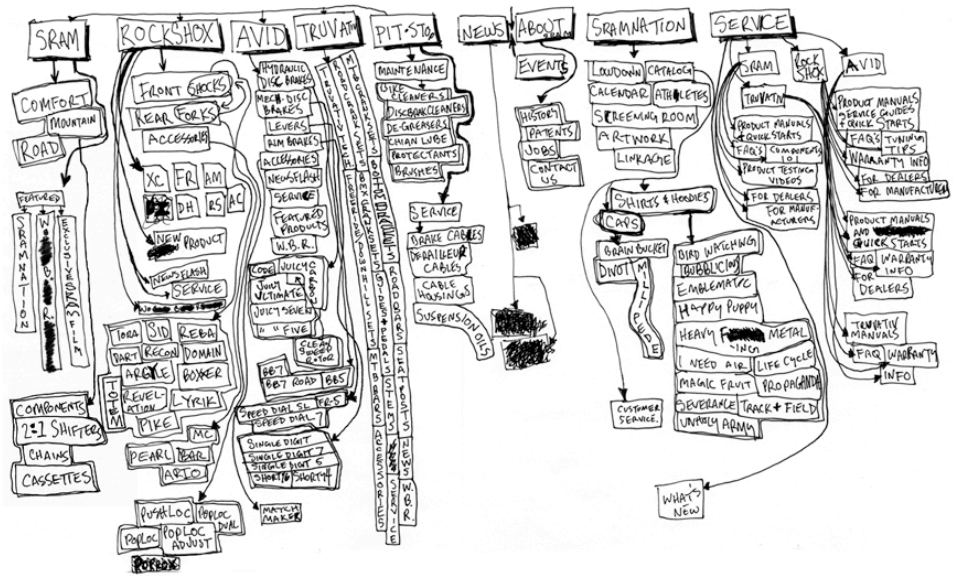
© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

Architecting Web Apps



Old School



© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Adobe

New School

```
<html>
<head>
    <title>Huge App</title>
    <script src="my-app.js"></script>
</head>
<body></body>
</html>
```

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

	Multi-Page	Single Page
# Pages	Many	One
UI Generation Tier	Server	Client
Languages	Java, .NET, PHP, RoR, ...	JavaScript
Offline Support	Limited	Yes
Page Transitions	Browser	Developer
Performance	Laggy	Fast
App Assets Loaded	Many Times	One Time

#3

Use Single Page Architecture

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Use Single Page Architecture

Benefits

- Fast
- Works offline
- Control over experience

Caveats

- More Complex
- Memory management
- Modular Strategy

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Building HTML with JavaScript

```
var html =
  '<div class="header">' +
      '<a href="#" class="button-left">List</a>' +
      '<h1>Employee</h1>' +
  '</div>' +
  '<div class="details">' +
      '<img src="pics/' + e.firstName + '_' + e.lastName + '.jpg"/>' +
      '<h1>' + e.firstName + ' ' + e.lastName + '</h1>' +
      '<h2>' + e.title + '</h2>' +
      '<ul class="list">' +
          '<li><a href="tel:' + e.officePhone + '">Call Office<br/>'
             + e.officePhone + '</a></li>' +
          '<li><a href="tel:' + e.cellPhone + '">Call Cell<br/>'
             + e.cellPhone + '</a></li>' +
          '<li><a href="sms:' + e.cellPhone + '">SMS<br/>'
             + e.cellPhone + '</a></li>' +
      '</ul>' +
  '</div>';
```

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Templates

```
<div class="header">
    <a href="#" class="button-left">List</a>
    <h1>Employee</h1>
</div>
<div class="details">
    <img src="pics/{{firstName}}_{{lastName}}.jpg"/>
    <h1>{{firstName}} {{lastName}}</h1>
    <h2>{{title}}</h2>
    <ul class="list">
        <li><a href="tel:{{officePhone}}">Call Office<br/>{{officePhone}}</a></li>
        <li><a href="tel:{{cellPhone}}">Call Cell<br/>{{cellPhone}}</a></li>
        <li><a href="sms:{{cellPhone}}">SMS<br/>{{cellPhone}}</a></li>
    </ul>
</div>
```

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

#4

Use Templates

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Templates

Benefits

- Maintainable
- Toolable
- Separation of concerns

Examples

- Mustache.js
- Handlebars.js
- Underscore.js

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Loading Templates

1. <script type="text/template"></script>

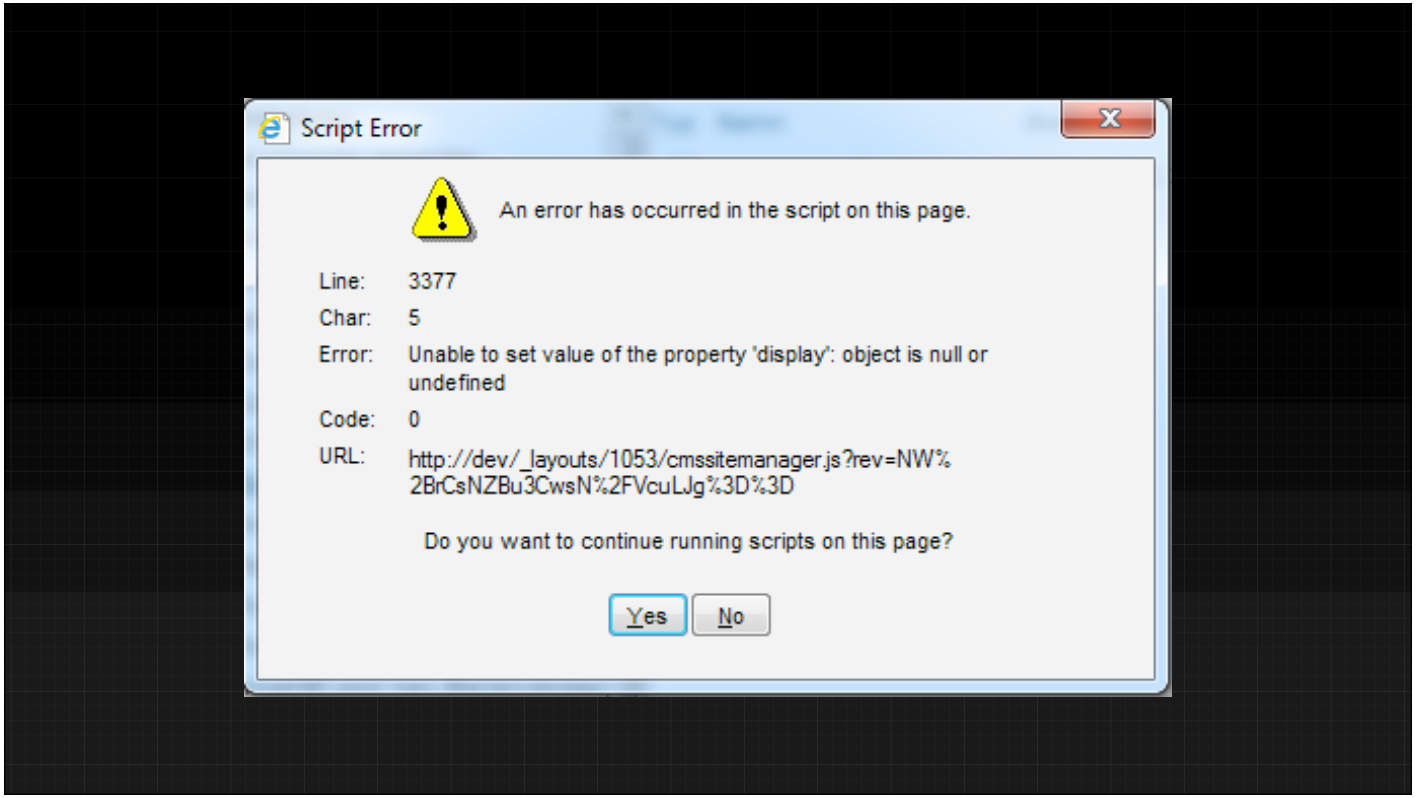
```
<script id="home-tpl" type="text/x-handlebars-template">
    <span id="firstName">{{firstName}}</span>
    <span id="lastName">{{lastName}}</span>
    <span id="phone">{{phone}}</span>
</script>
```

2. Template loader

3. Require.js text module

4. Build (Grunt.js)

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.


Script Error
An error has occurred in the script on this page.
Line: 3377
Char: 5
Error: Unable to set value of the property 'display': object is null or undefined
Code: 0
URL: http://dev/_layouts/1053/cmssitemanager.js?rev=NW%2BrCsNZBu3CwsN%2FVcuLJg%3D%3D
Do you want to continue running scripts on this page?
Yes
No

#5

Provide Structure to Your Application
Using a MV* Architecture

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Model

```
var Employee = function() {

    this.url = "/employee";

    this.validate = function() {

    };

});
```

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

View

```
var EmployeeView = function () {

    this.initialize = function () {

    };

    this.render = function() {

    };

}
```

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Controller

```
var JSONPAdapter = function(url) {

    this.findById = function(id) {
        return $.ajax({url: url + "/" + id, dataType: "jsonp"});
    }

    this.findByName = function(searchKey) {
        return $.ajax({url: url + "?name=" + searchKey, dataType: "jsonp"});
    }

}
```

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

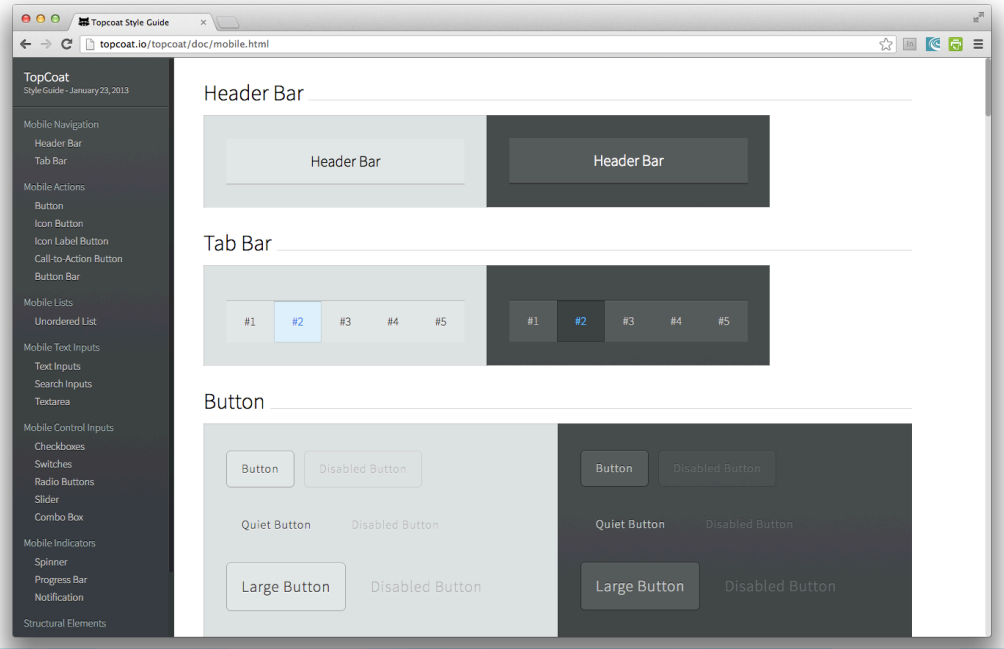
#6

Consider Frameworks

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Framework Landscape

Topcoat



© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Topcoat Resources

http://topcoat.io

https://github.com/topcoat/topcoat

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

#7

Routing

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

#8

Abstract Device Features

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Interaction	Mouse	Touch
Notification	JavaScript	Native
Storage	online	offline
Sensors	unavailable	available

#9

Hide HTMLish Behaviors

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Hide HTMLish Behaviors

- alert()
- -webkit-touch-callout: none;
- -webkit-tap-highlight-color:rgba(0, 0, 0, 0);

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

#10

Architect for Performance

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Architect for Performance

- Don't generate UI on the server
- Don't wait for data to display the UI
- Cache everything (data, selectors, precompiled templates, ...)
- Use Hardware acceleration
- Avoid click event's 300ms delay
- Use CSS sprite sheets
- Limit shadows and gradients
- Avoid reflows
- Do you need that framework?
- Test

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.

Summary

1. Abstract data access
2. Keep Application browser runnable
3. Use single page application
4. Use templates
5. Use MV* architecture
6. Consider a framework
7. Implement routing
8. Abstract device features
9. Hide HTMLish behaviors
10. Architect for performance

© 2013 Adobe Systems Incorporated. All Rights Reserved. Adobe Confidential.


Adobe