jQuery-Mobile vs. Sencha Touch

May the best win

Qafoo GmbH/crosscan GmbH

October 25, 2012

What comes next?

Welcome

About Me

Jakob Westhoff

- More than 11 years of professional PHP experience
- More than 8 years of professional JavaScript experience
- Open source enthusiast
- Regular speaker at (inter)national conferences
- Consultant, Trainer and Author

Working with

About Me

Jakob Westhoff

- More than 11 years of professional PHP experience
- More than 8 years of professional JavaScript experience
- Open source enthusiast
- Regular speaker at (inter)national conferences
- Consultant, Trainer and Author

Working with

We help people to create high quality web applications.

About Me

Jakob Westhoff

- More than 11 years of professional PHP experience
- More than 8 years of professional JavaScript experience
- Open source enthusiast
- Regular speaker at (inter)national conferences
- Consultant, Trainer and Author

Working with

We help people to create high quality web applications.

`http://qafoo.com`

About Me

Hans-Christian Otto

- More than 8 years of professional PHP experience
- More than 5 years of professional JavaScript experience
- Regular speaker at international conferences
- Director of software development with crosscan GmbH

Working with

Goals of this session

- Getting to know jQuery Mobile
- . . . and Sencha Touch
- Learn about the fudamental differences
- Gather knowledge to decide which one suits your needs

What comes next?

Facts

jQuery Mobile - The Facts

- Mobile Web-Application Framework

jQuery Mobile - The Facts

- Mobile Web-Application Framework
- Support for almost any current mobile platform
  - Apple iOS 3.2-6.0
  - Android 2.x - 4.1
  - Windows Phone 7.x
  - Blackberry 6/7
  - Blackberry Playbook
  - Palm WebOS
  - Firefox Mobile
  - Chrome for Mobile
  - Kindle Fire
  - ...

jQuery Mobile - The Facts

- Quite small
  - jQuery: 33kb (minified+gzipped)
  - jQuery-Mobile: 24kb (minified+gzipped)
  - CSS: 7kb
  - Full: 64kb

jQuery Mobile - The Facts

- Quite small
  - jQuery: 33kb (minified+gzipped)
  - jQuery-Mobile: 24kb (minified+gzipped)
  - CSS: 7kb
  - Full: 64kb
- Don’t use all the features?
  - Use the new Download Builder to create custom packages

jQuery Mobile - The Facts

- Quite small
  - jQuery: 33kb (minified+gzipped)
  - jQuery-Mobile: 24kb (minified+gzipped)
  - CSS: 7kb
  - Full: 64kb
- Don’t use all the features?
  - Use the new Download Builder to create custom packages
- Proper CDN for cachable inclusion exists

jQuery Mobile - The Facts

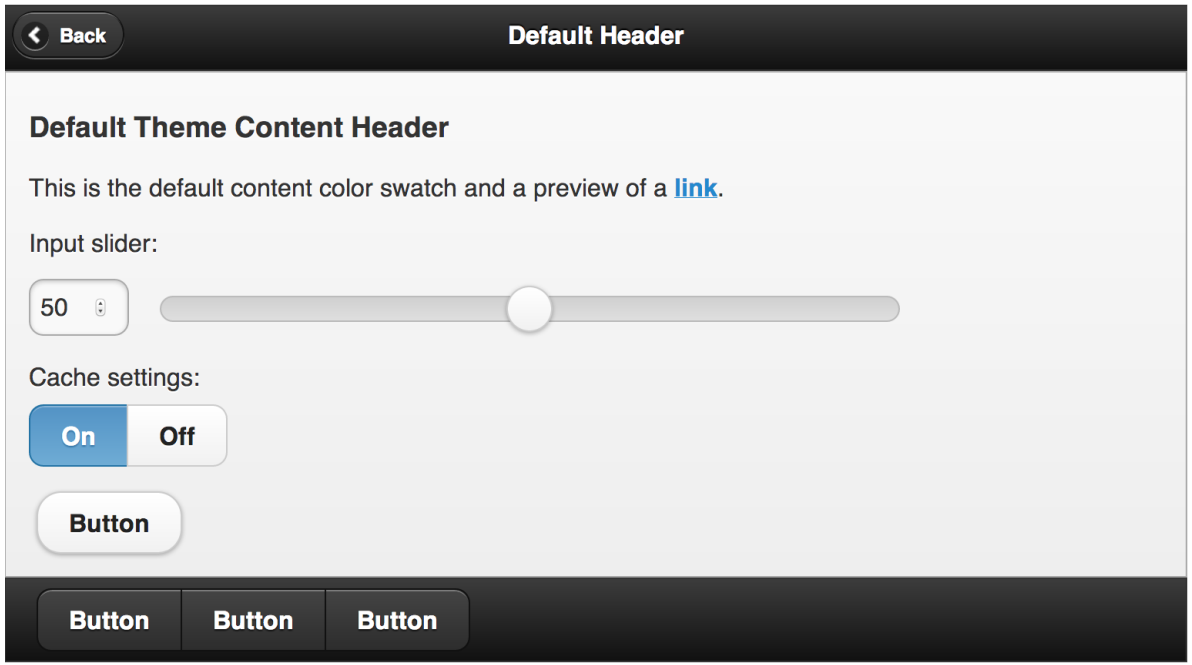
- Fully configurable themes with multiple swatches

jQuery Mobile - The Facts

- Fully configurable themes with multiple swatches
- Graphical Theme Designer exists (free)
  - ThemeRoller

jQuery Mobile - The Facts

- Fully configurable themes with multiple swatches
- Graphical Theme Designer exists (free)
  - ThemeRoller
- Default theme has 5 different colorations (swatches)



jQuery Mobile - The Facts

- Utilization of `jQuery` and `jQuery-UI` base framework
- Usage of progressive enhancement to realize gui components
- A variety of widgets: (pages, dialogs, toolbars, listviews, buttons, forms, ...)
- Support for Accessibility-features like WAI-ARIA throughout the framework

Sencha Touch - The Facts

- Mobile Web-Application Framework

Sencha Touch - The Facts

- Mobile Web-Application Framework
- Support for only the most important mobile platforms
  - Apple iOS 4-6.0
  - Android 2.3 - 4.1
  - Blackberry 6/7
  - Blackberry Playbook
  - Chrome for Mobile ?
  - ...

Sencha Touch - The Facts

- Not that small
  - JavaScript: 568kb
  - CSS: 178kb
  - Full: 746kb

Sencha Touch - The Facts

- Not that small
  - JavaScript: 568kb
  - CSS: 178kb
  - Full: 746kb
- Don’t use all the features?
  - Use the new Sencha CMD to create custom packages

Sencha Touch - The Facts

- Not that small
  - JavaScript: 568kb
  - CSS: 178kb
  - Full: 746kb
- Don’t use all the features?
  - Use the new Sencha CMD to create custom packages
- Proper CDN for cachable inclusion exists (but this contains the whole framework)

Sencha Touch - The Facts

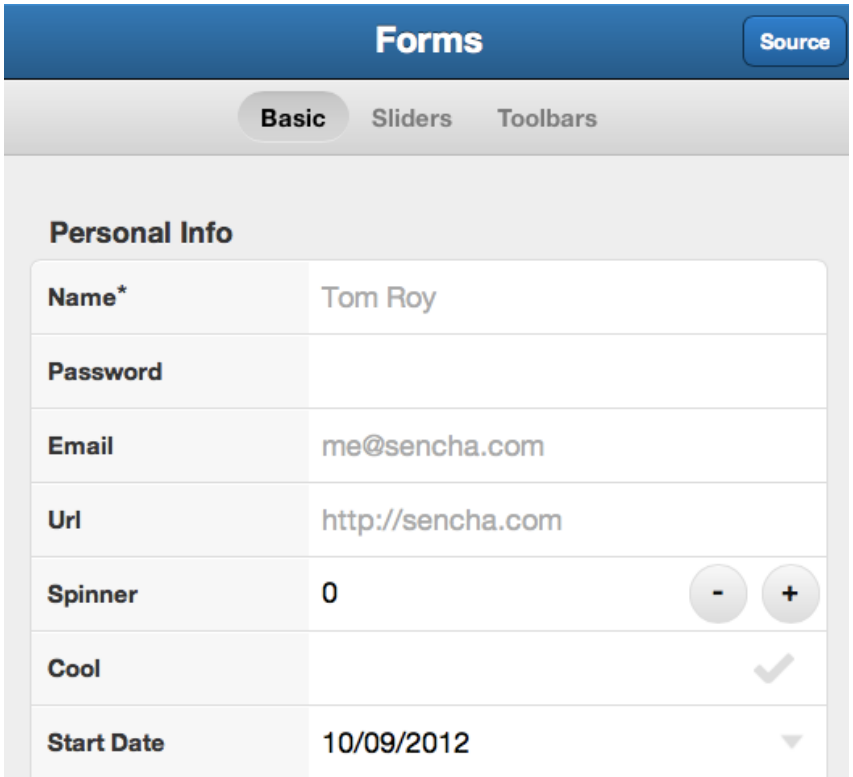
- Fully configurable themes (created using SASS/Compass)

Sencha Touch - The Facts

- Fully configurable themes (created using SASS/Compass)
- Several (not so sophisticated) theme builders exist

Sencha Touch - The Facts

- Fully configurable themes (created using SASS/Compass)
- Several (not so sophisticated) theme builders exist



Sencha Touch - The Facts

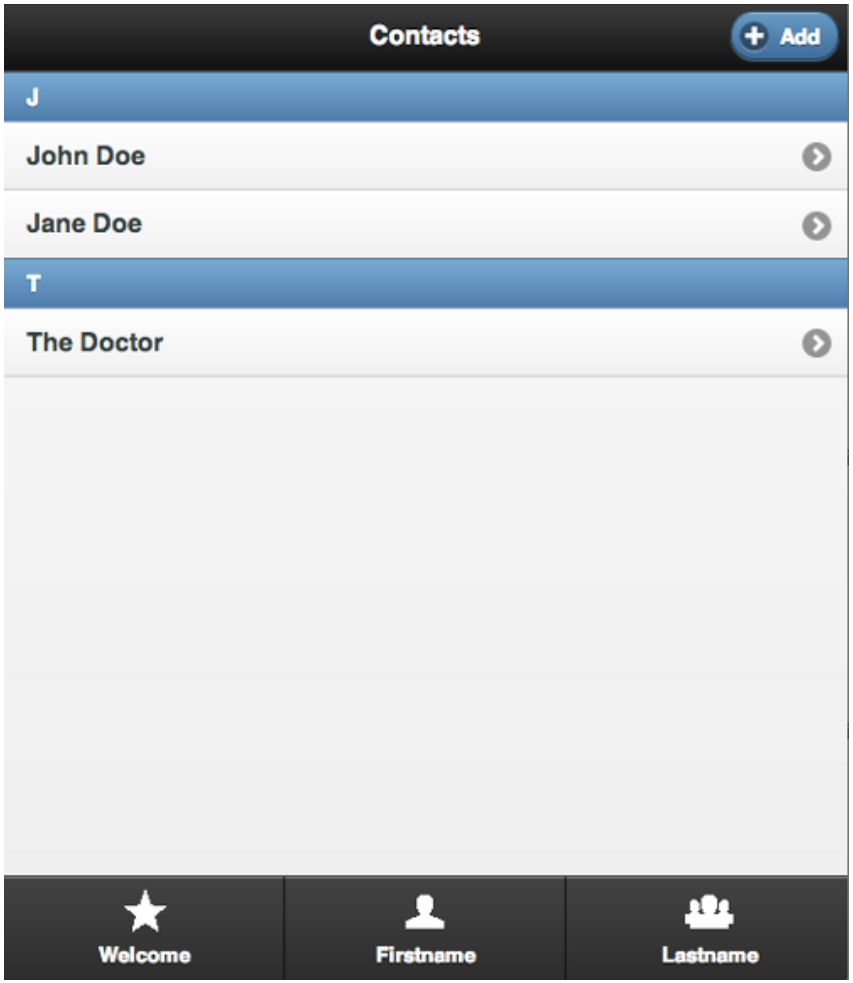
- Similar to Senchas Ext JS framework, but not based on it.
- No progressive enhancement, no seemless degredation, no semantic websites
- A variety of widgets: (dialogs, toolbars, listviews, buttons, forms, sliders ...)
- Object oriented software development (read: Java style)
- Native packaging

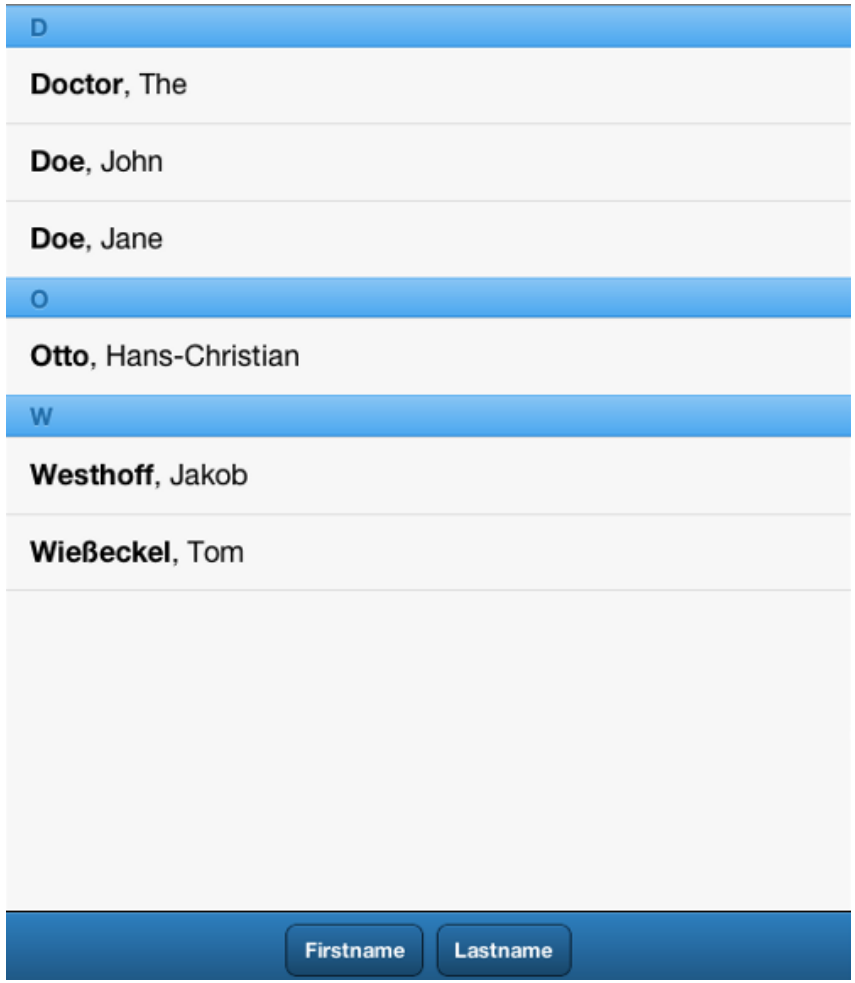
A Demo Application

A simple contacts application

- For demonstration purposes we use a simple application
- Managing contacts
  - List Contacts in different sort orders (firstname, lastname)
  - Add a new contact
  - View contact details

A simple contact application





What comes next?

Bootstrapping

Bootstrapping - jQuery Mobile

- Bootstrapping a jQuery Mobile application
  1. Create HTML document
  2. Load jQuery Mobile CSS
  3. Load jQuery library
  4. Load jQuery Mobile library
  5. Create a jQuery Mobile page

Bootstrapping - jQuery Mobile

1. Create HTML document

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Contact Demo</title>
5      <meta name="viewport" content="width=device-width, initial-
           scale=1">
6  </head>
7  <body>
8
9  </body>
10 </html>
```

Bootstrapping - jQuery Mobile

2. Load jQuery Mobile CSS

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Contact Demo</title>
5      <meta name="viewport" content="width=device-width, initial-
           scale=1">
6      <link rel="stylesheet" href="http://code.jquery.com/mobile
           /1.2.0/jquery.mobile-1.2.0.min.css" />
7  </head>
8  <body>
9
10 </body>
11 </html>
```

Bootstrapping - jQuery Mobile

3. Load jQuery library

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Contact Demo</title>
5      <meta name="viewport" content="width=device-width, initial-
           scale=1">
6      <link rel="stylesheet" href="http://code.jquery.com/mobile
           /1.2.0/jquery.mobile-1.2.0.min.css" />
7      <script src="http://code.jquery.com/jquery-1.8.2.min.js"></
           script>
8  </head>
9  <body>
10
11 </body>
12 </html>
```

Bootstrapping - jQuery Mobile

4. Load jQuery Mobile library

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Contact Demo</title>
5      <meta name="viewport" content="width=device-width, initial-
           scale=1">
6      <link rel="stylesheet" href="http://code.jquery.com/mobile
           /1.2.0/jquery.mobile-1.2.0.min.css" />
7      <script src="http://code.jquery.com/jquery-1.8.2.min.js"></
           script>
8      <script src="http://code.jquery.com/mobile/1.2.0/jquery.mobile
           -1.2.0.min.js"></script>
9  </head>
10 <body>
11
12 </body>
13 </html>
```

Bootstrapping - jQuery Mobile

5. Create a jQuery Mobile page

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Contact Demo</title>
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <link rel="stylesheet" href="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.
           min.css" />
7      <script src="http://code.jquery.com/jquery-1.8.2.min.js"></script>
8      <script src="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.js"></script
           >
9  </head>
10 <body>
11
12 <div data-role="page" id="welcome">
13     <!-- ... -->
14 </div>
15
16 </body>
17 </html>
```

Bootstrapping - jQuery Mobile

- You don’t need to write any JavaScript to create a basic application
- Semantic HTML is enough
- JavaScript is only needed for specialized interaction

Bootstrapping - Sencha Touch

- Bootstrapping a Sencha Touch application
  1. Create HTML document
  2. Load Sencha Touch CSS
  3. Load Sencha Touch JS
  4. Load Application JS
  5. Create Application JS

Bootstrapping - Sencha Touch

1. Create HTML document

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Contact Demo</title>
5      <meta charset="UTF-8">
6  </head>
7  <body></body>
8  </html>
```

Bootstrapping - Sencha Touch

2. Load Sencha Touch CSS

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Getting Started</title>
5      <meta charset="UTF-8">
6      <link rel="stylesheet" href="http://extjs.cachefly.net/touch/
           sencha-touch-2.0.0/resources/css/sencha-touch.css" type="
           text/css">
7  </head>
8  <body></body>
9  </html>
```

Bootstrapping - Sencha Touch

3. Load Sencha Touch JS

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Getting Started</title>
5      <meta charset="UTF-8">
6      <link rel="stylesheet" href="http://extjs.cachefly.net/touch/
           sencha-touch-2.0.0/resources/css/sencha-touch.css" type="
           text/css">
7      <script type="text/javascript" src="http://extjs.cachefly.net/
           touch/sencha-touch-2.0.0/sencha-touch-all-debug.js"></
           script>
8  </head>
9  <body></body>
10 </html>
```

Bootstrapping - Sencha Touch

4. Load Application JS

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Getting Started</title>
5      <meta charset="UTF-8">
6      <link rel="stylesheet" href="http://extjs.cachefly.net/touch/
           sencha-touch-2.0.0/resources/css/sencha-touch.css" type="
           text/css">
7      <script type="text/javascript" src="http://extjs.cachefly.net/
           touch/sencha-touch-2.0.0/sencha-touch-all-debug.js"></
           script>
8      <script type="text/javascript" src="app.js"></script>
9  </head>
10 <body></body>
11 </html>
```

Bootstrapping - Sencha Touch

- That's your only .html file
- You won't change it anymore
- The rest is plain JS

Bootstrapping - Sencha Touch

4. Create Application JS

```
1  Ext.application({
2      launch: function()
3      {
4
5
6
7
8      }
9  });
```

Bootstrapping - Sencha Touch

4. Create Application JS

```
1  Ext.application({
2      launch: function()
3      {
4          var view = Ext.create('Ext.Panel', {
5              fullscreen: true,
6              html: 'Foo'
7          });
8      }
9  });
```

Bootstrapping - Conclusion

- jQuery Mobile
  - No fancy generators/tools needed
  - Plain semantic HTML
  - Easy to bootstrap
- Sencha Touch
  - No fancy generators/tools needed
  - Very simple and static html
  - Writing JavaScript is required

What comes next?

UI Design

UI Design - jQuery Mobile

- jQuery Mobile has a special approach to UI design
  - Semantic HTML5
  - Progressive Enhancement

UI Design - jQuery Mobile

- jQuery Mobile has a special approach to UI design
  - Semantic HTML5
  - Progressive Enhancement
- Declarative creation of the user interface using known HTML elements
  - Each ui element has a role (Page, Listview, Button, ...)
  - Heavy usage of HTML5 `data` attributes (`data-role`, `data-theme`, `...`)

Concept of pages

- Each jQuery Mobile application consists of an arbitrary number of pages

Concept of pages

- Each jQuery Mobile application consists of an arbitrary number of pages
- Multiple pages may reside in one HTML document

Concept of pages

- Each jQuery Mobile application consists of an arbitrary number of pages
- Multiple pages may reside in one HTML document
- Pages may be split up into different HTML documents
  - jQuery Mobile will automatically load them on demand (asynchronously)

Concept of pages

- A welcome page for the application

```
1  <div data-role="page" id="welcome">
2    <div data-role="content">
3      <h2>Welcome to this session about jQuery-Mobile and Sencha
           Touch</h2>
4      <p>
5        This simple application demonstrates the different concepts
           used in jQuery-Mobile to realize a mobile web application
           .
6      </p>
7    </div>
8  </div>
```

Concept of pages

- A welcome page for the application

```
<div data-role="page" id="welcome">
  <div data-role="content">
    <h2>Welcome to this session about jQuery-Mobile and Sencha
        Touch</h2>
    <p>
      This simple application demonstrates the different concepts
          used in jQuery-Mobile to realize a mobile web application
          .
    </p>
  </div>
</div>
```

- Every page is contained by a `<div>`

Concept of pages

- A welcome page for the application

```
<div data-role="page" id="welcome">
  <div data-role="content">
    <h2>Welcome to this session about jQuery-Mobile and Sencha
        Touch</h2>
    <p>
      This simple application demonstrates the different concepts
          used in jQuery-Mobile to realize a mobile web application
          .
    </p>
  </div>
</div>
```

- Every page is contained by a `<div>`
- A page is annotated with the role `page`

Concept of pages

- A welcome page for the application

```
<div data-role="page" id="welcome">
  <div data-role="content">
    <h2>Welcome to this session about jQuery-Mobile and Sencha
        Touch</h2>
    <p>
      This simple application demonstrates the different concepts
          used in jQuery-Mobile to realize a mobile web application
          .
    </p>
  </div>
</div>
```

- Every page is contained by a `<div>`
- A page is annotated with the role `page`
- Each page needs to provide a unique id for identification/navigation

Header and Footer

- Most mobile web applications want to display a header and a footer for each page

```
<div data-role="page" id="welcome">
  <div data-role="header">
    <h1>Welcome</h1>
  </div>

  <div data-role="content">
    <h2>Welcome to this session about jQuery-Mobile and Sencha
        Touch</h2>
    <!-- ... -->
  </div>

  <div data-role="footer"></div>
</div>
```

Header and Footer

- Most mobile web applications want to display a header and a footer for each page

```
<div data-role="page" id="welcome">
  <div data-role="header">
    <h1>Welcome</h1>
  </div>

  <div data-role="content">
    <h2>Welcome to this session about jQuery-Mobile and Sencha
        Touch</h2>
    <!-- ... -->
  </div>

  <div data-role="footer"></div>
</div>
```

- Header and footer may be fixed: `data-position="fixed"`

Header and Footer

- Header and Footer may contain other widgets/elements

```
<div data-role="page" id="list">
  <div data-role="header">
    <h1>Contacts</h1>
    <a href="#" class="ui-btn-right">Add</a>
  </div>

  <div data-role="content"></div>

  <div data-role="footer">
    <div data-role="navbar">
      <ul>
        <li><a href="#">Welcome</a></li>
        <li><a href="#">Firstname</a></li>
        <li><a href="#">Lastname</a></li>
      </ul>
    </div>
  </div>
</div>
```

Navigation between pages

- For navigation between pages use their id as document fragment

Navigation between pages

- For navigation between pages use their id as document fragment

```
<div data-role="page" id="list">
  <div data-role="header">
    <h1>Contacts</h1>
    <a href="#add" class="ui-btn-right">Add</a>
  </div>

  <div data-role="content"></div>

  <div data-role="footer">
    <div data-role="navbar">
      <ul>
        <li><a href="#welcome">Welcome</a></li>
        <li><a href="#list">Firstname</a></li>
        <li><a href="#list">Lastname</a></li>
      </ul>
    </div>
  </div>
</div>
```

Navigation between pages

- jQuery Mobile does have a AJAX navigation enginge, which kicks in automatically

Navigation between pages

- jQuery Mobile does have a AJAX navigation enginge, which kicks in automatically
- The engine automatically navigates to the referenenced page
  - Without leaving the current page context

Navigation between pages

- jQuery Mobile does have a AJAX navigation enginge, which kicks in automatically
- The engine automatically navigates to the referenenced page
  - Without leaving the current page context
- If hasn't been loaded yet an XHR request will be fired automatically

Creating Listviews

- Listviews are often used in mobile applications
- Let's take a look at a listview for the stored contacts

```
<div data-role="page" id="list">
  <div data-role="header"><!-- ... --></div>

  <div data-role="content">
    <ul data-role="listview" data-autodividers="true">
      <li><a href="#">John Doe</a></li>
      <li><a href="#">Jane Doe</a></li>
      <li><a href="#">The Doctor</a></li>
    </ul>
  </div>

  <div data-role="footer"><!-- ... --></div>
</div>
```

Dialogs and Popups

- Sometimes you want to display a dialog or popup inside your application

```
<div data-role="page" id="list">
  <div data-role="header"><!-- ... --></div>

  <div data-role="content">
    <ul data-role="listview" data-autodividers="true">
      <li><a href="#view" data-rel="dialog">John Doe</a></li>
      <li><a href="#view" data-rel="dialog">Jane Doe</a></li>
      <li><a href="#view" data-rel="dialog">The Doctor</a></li>
    </ul>
  </div>

  <div data-role="footer"><!-- ... --></div>
</div>
```

UI Design - Sencha Touch

- Sencha Touch uses object oriented code for UI design
- Classes that represent UI elements are instantiated
  - Classes are created using `Ext.create`, not using `new`
  - Constructors receive a configuration object

UI Design - Sencha Touch

- Sencha Touch uses object oriented code for UI design
- Classes that represent UI elements are instantiated
  - Classes are created using `Ext.create`, not using `new`
  - Constructors receive a configuration object
- MVC
  - UI Classes (”components”) are the View
  - Models represent data
  - Controllers are the glue code

UI Design - Sencha Touch

- Sencha Touch uses object oriented code for UI design
- Classes that represent UI elements are instantiated
  - Classes are created using `Ext.create`, not using `new`
  - Constructors receive a configuration object
- MVC
  - UI Classes (”components”) are the View
  - Models represent data
  - Controllers are the glue code
    - Not shown in this talk

Sencha Touch Components

- Each Sencha Touch application consists of an arbitrary number of components

Sencha Touch Components

- Each Sencha Touch application consists of an arbitrary number of components
- Some components are containers

Sencha Touch Components

- Each Sencha Touch application consists of an arbitrary number of components
- Some components are containers
- A Container has a layout
  - Fit
  - Card
  - HBox/VBox

Sencha Touch Components

- Each Sencha Touch application consists of an arbitrary number of components
- Some components are containers
- A Container has a layout
  - Fit
  - Card
  - HBox/VBox
- There is a special full screen container called `viewport`

Sencha Touch Components

- Each Sencha Touch application consists of an arbitrary number of components
- Some components are containers
- A Container has a layout
  - Fit
  - Card
  - HBox/VBox
- There is a special full screen container called `viewport`
- Components only visualize data from models

Docked Items

- Header and Footer are created by `docking` components to the edges of containers.

```
1  Ext.create('Ext.Panel', {
2      items: [
3          Ext.create('Ext.TitleBar', {
4              docked: 'top',
5              title: 'Foo'
6          }),
7          Ext.create('Ext.TitleBar', {
8              docked: 'bottom',
9              title: 'Baz'
10         })
11     ],
12     fullscreen: true,
13     html: 'Bar'
14 });
```

Navigation between components

- Switching views is done with js code.

Navigation between components

- Switching views is done with js code.

```
1  var firstView = Ext.create('Ext.Panel', {
2      html: 'Foo'
3  });
4
5  var secondView = Ext.create('Ext.Panel', {
6      html: 'Bar'
7  });
8
9  Ext.Viewport.add(firstView);
10 Ext.Viewport.add(secondView);
11
12 Ext.Viewport.setActiveItem(secondView);
```

Creating Listviews

```
1  Ext.create('Ext.List', {
2      fullscreen: true,
3      itemTpl: '{title}',
4      data: [
5          { name: 'John Doe' },
6          { name: 'Jane Doe' },
7          { name: 'The Doctor' }
8      ]
9  });
```

Dialogs and Popups

```
1  Ext.Msg.prompt('Name', 'Please enter your name:', function(button,
       text) {
2      Ext.Msg.alert('Welcome!', 'Hi, ' + text + '!', Ext.emptyFn);
3  });
```

UI Design - Conclusion

- jQuery Mobile
  - Usage of HTML5 + Progressive Enhancement
  - Semantic Markup
  - Easily create user interfaces for static content
  - Page navigation for free
  - Automatic XHR lazy fetching of pages
- Sencha Touch
  - HTML 5 - but you don't write it
  - Declarative programming using JavaScript
  - "Page" navigation has to be done programmatically
  - Autoloading of application components

What comes next?

Dynamic Data

Dynamic Data - jQuery Mobile

- Using static content with jQuery Mobile is easy
- How about dynamic creation of content
  - XML, JSON, Localstorage, ...

An event based approach

- jQuery Mobile fires all kind of different events during processing
  - `pagebeforeload`
  - `pageload`
  - `pageloadfailed`
  - `pagebeforechange`
  - `pagechange`
  - `pagechangefailed`
  - `pagebeforeshow`
  - `pageshow`
  - `...`

An event based approach

- jQuery Mobile fires all kind of different events during processing
  - `pagebeforeload`
  - `pageload`
  - `pageloadfailed`
  - `pagebeforechange`
  - `pagechange`
  - `pagechangefailed`
  - `pagebeforeshow`
  - `pageshow`
  - `...`

`pagebeforechange` - dynamic interaction

- The `pagebeforechange` event is fired before jQuery Mobile changes to another page view
- It's the perfect place for dynamic manipulation
  - The page is not visible yet
  - All information about the requested page is available

Sorting order injection - Contact App

- The contact application needs two versions of the contact list
  - Sorted by firstname
  - Sorted by lastname

Sorting order injection - Contact App

- The contact application needs two versions of the contact list
  - Sorted by firstname
  - Sorted by lastname
- Solution
  - Add a parameter to the page links indicating the sort order
  - Hijacking the page change to that page in order to modify it dynamically

Adding a page link parameter

- Add a parameter to the page links indicating the sort order

```
<div data-role="footer">
  <div data-role="navbar">
    <ul>
      <li><a href="#welcome">Welcome</a></li>
      <li><a href="#list?firstname">Firstname</a></li>
      <li><a href="#list?lastname">Lastname</a></li>
    </ul>
  </div>
</div>
```

Hijacking the page change event

- Hijacking the page change to that page in order to modify it dynamically

```
1  $(document).bind('pagebeforechange', function(e, data) {
2      if (typeof data.toPage !== "string") {
3          return;
4      }
5      var url = $.mobile.path.parseUrl(data.toPage);
6      if (url.hash.indexOf("#list?") !== 0 ){
7          return;
8      }
9      var order = url.hash.replace( /^#list\?/, "" );
10
11     // Render listview with appropriate sorting in #list page
12
13     $(":jqmRole(listview)", page).listview("refresh");
14
15     e.preventDefault();
16     $.mobile.changePage("#list");
17 });
```

Other manipulation options

- The same technique can be used for the detailed view dialog

Other manipulation options

- The same technique can be used for the detailed view dialog
- Changes can be applied on the fly, by simply binding to events on html elements
- All event registration should be done inside the `pageinit` event of the page to be manipulated
  - Sort of the `domReady` equivalent inside the jQuery Mobile world

-

Sencha Touch data package

- Models

Sencha Touch data package

- Models
- Proxies to load models

Sencha Touch data package

- Models
- Proxies to load models
- Stores to group models of one kind

Sencha Touch data package

- Models
- Proxies to load models
- Stores to group models of one kind
- Lists can be bound to a store

Stores

- Will be filled using a proxy
  - A `MemoryProxy` is available
- Sorting of models
- Filtering of models
- Querying of models
- Grouping of models

Using Stores

```
1  var contactsStore = Ext.create('Ext.data.Store', {
2      fields: ['firstName', 'lastName'],
3      data: [
4          {firstName: 'John', lastName: 'Doe' },
5          {lastName: 'Doctor', firstName: 'The'},
6          {firstName: 'Jane', lastName: 'Doe'}
7      ],
8  });
9
10 var contactsList = Ext.create('Ext.dataview.List', {
11     itemTpl: '<b>{lastName}</b>, {firstName}',
12     store: contactsStore
13 });
```

User Interactions with list items

```
1  contactsList.on(
2      'itemtap',
3      function(list, index, element, model) {
4          Ext.Msg.alert(
5              'Contact Details',
6              model.get('firstName') + model.get('lastName')
7          );
8      }
9  );
```

Sorting of stores

```
1  var toolbar = Ext.create('Ext.Toolbar', {
2    items: [{
3      text: 'Firstname',
4      handler: function() {
5        contactsList.setItemTpl('{firstName} <b>{lastName}</b>')
6        contactsStore.sort('firstName', 'ASC');
7      }
8    },{
9      text: 'Lastname',
10     handler: function() {
11       contactsList.setItemTpl('<b>{lastName}</b>, {firstName}')
12       contactsStore.sort('lastName', 'ASC');
13     }
14   }],
15   docked: 'bottom'
16 })
17
18 Ext.Viewport.add(toolbar);
```

Real-Life-Stores

- Stores can load data via proxies
- Readers convert proxy response to model
- AJAX, JSON, XML, JSONP, localstorage, Ext.Direct, ...

Real-Life-Stores

- Stores can load data via proxies
- Readers convert proxy response to model
- AJAX, JSON, XML, JSONP, localstorage, Ext.Direct, ...

```
var contactsStore = Ext.create('Ext.data.Store', {
    fields: ['firstName', 'lastName'],
    proxy: {
        type: 'ajax',
        url : '/contacts.json',
        reader: {
            type: 'json',
            rootProperty: 'contacts'
        }
    },
    autoLoad: true
});
```

JsonStore expected data

```
{
    'contacts': [
        {'firstName': 'John', 'lastName': 'Doe' },
        {'lastName': 'Doctor', 'firstName': 'The'},
        {'firstName': 'Jane', 'lastName': 'Doe'}
    ]
}
```

Dynamic Data - Conclusion

- jQuery Mobile
  - jQuery Mobile is really good with mostly static content
  - Dynamic data can be inserted, but it's not really elegant
  - Hard to maintain if the application is mostly dynamic
  - Of course a proper "framework" could be written for those cases
- Sencha Touch
  - Sencha Touch is overhead for apps with mostly static content
  - Changing from static to dynamic content is straightforward
  - Framework is highly optimized for dynamic content

What comes next?

Conclusion

Conclusion

jQuery Mobile

Sencha Touch

Conclusion

jQuery Mobile

- Focused on mostly all mobile browsers

Sencha Touch

Conclusion

jQuery Mobile

- Focused on mostly all mobile browsers

Sencha Touch

- Focused on webkit mobile browsers

Conclusion

jQuery Mobile

- Focused on mostly all mobile browsers
- Semantic HTML for UI

Sencha Touch

- Focused on webkit mobile browsers

Conclusion

jQuery Mobile

- Focused on mostly all mobile browsers
- Semantic HTML for UI

Sencha Touch

- Focused on webkit mobile browsers
- JavaScript OOP for UI

Conclusion

jQuery Mobile

- Focused on mostly all mobile browsers
- Semantic HTML for UI
- Small library filesize

Sencha Touch

- Focused on webkit mobile browsers
- JavaScript OOP for UI

Conclusion

jQuery Mobile

- Focused on mostly all mobile browsers
- Semantic HTML for UI
- Small library filesize

Sencha Touch

- Focused on webkit mobile browsers
- JavaScript OOP for UI
- Quite big files for mobile

Conclusion

jQuery Mobile

- Focused on mostly all mobile browsers
- Semantic HTML for UI
- Small library filesize
- Perfect fit for applications focused on static data

Sencha Touch

- Focused on webkit mobile browsers
- JavaScript OOP for UI
- Quite big files for mobile

Conclusion

jQuery Mobile

- Focused on mostly all mobile browsers
- Semantic HTML for UI
- Small library filesize
- Perfect fit for applications focused on static data

Sencha Touch

- Focused on webkit mobile browsers
- JavaScript OOP for UI
- Quite big files for mobile
- Perfect fit for applications focused on dynamic data

Conclusion

jQuery Mobile

- Focused on mostly all mobile browsers
- Semantic HTML for UI
- Small library filesize
- Perfect fit for applications focused on static data

Sencha Touch

- Focused on webkit mobile browsers
- JavaScript OOP for UI
- Quite big files for mobile
- Perfect fit for applications focused on dynamic data

If no special requirements kick in: Choose what you like and know best!

Thanks for listening

Questions, comments or annotations?

Rate us: http://joind.in/7333

Slides: http://talks.qafoo.com

Jakob Westhoff
Contact: <jakob@qafoo.com>
Follow Me: @jakobwesthoff
Hire Qafoo: http://qafoo.com

Hans-Christian Otto
Contact: <c.otto@lab9.de>
Follow Me: @muhdiekuh