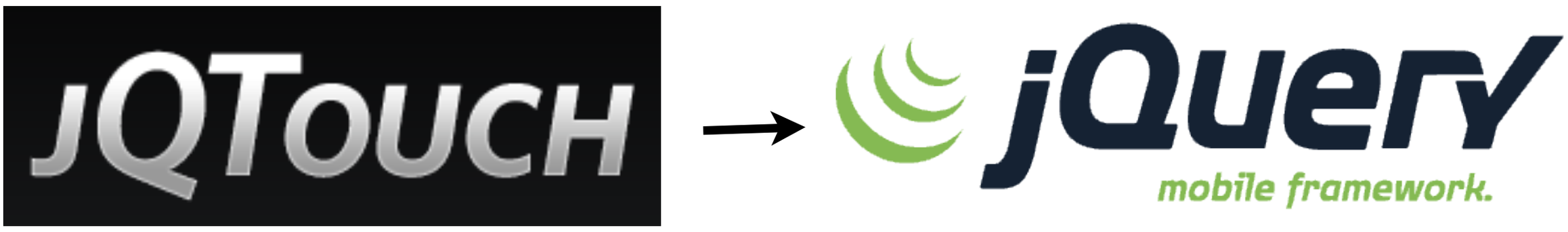
An Introduction to jQuery Mobile

Developing for mobile with web standards

Hello, my name is Tim Rogers and I'm a 17 year old student and developer from outside London. In my spare time, I love developing for mobile platforms using web technologies like HTML, CSS and JavaScript, amongst other things.

So, put your hand up if you own an Android or iOS device.
And who here has previously tried to build their own mobile app?

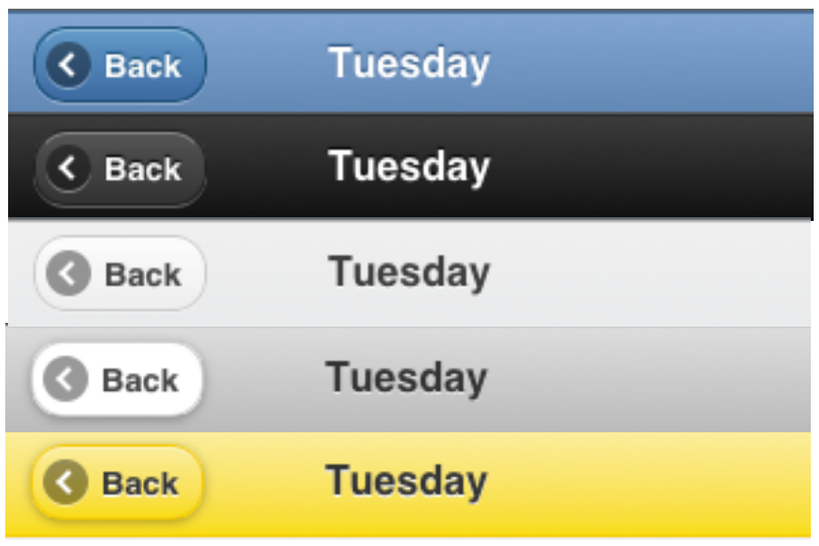
If you're a regular at DevNest, you will have seen me talk about this topic before. However, the business of developing apps with web standards has been greatly enhanced by the release of the alpha of jQuery Mobile, a specific toolkit for doing so which integrates into jQuery.

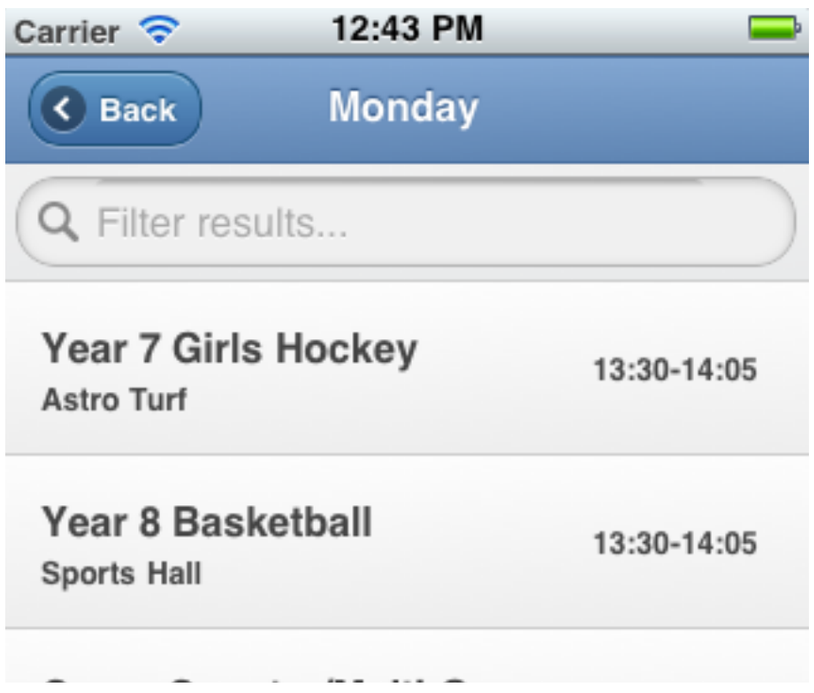


Previously, I have used a library called jQtouch, as in my last presentation. jQTouch was and still is a great jQuery plugin which provides a simple mechanism to create an iPhone style navigation for your web page with very little effort. It was developed by David Kaneda who is now part of Sencha, a company which is making its own interesting framework called Sencha Touch - but I'm not going to go into that today.

More recently, and what I'm going to talk about today, is the specific effort by the jQuery team to create their own framework for developing mobile apps, very simply called jQuery Mobile. This takes the ideas of jQTouch, allowing you to simply make an interface for your web application, but it takes them much further, offering absolutely loads of different interface options and colour schemes and the like for your app, as opposed to jQTouch which offered just two styles. It is compatible with just about every platform you can imagine, including iOS, Android, Blackberry, WebOS, Symbian and even Windows Phone.

Why jQuery Mobile?





So, as you know, you could quite easily just code all the interface and the like for the app yourself - but that would take a pretty long time. You could also use an alternative mobile development framework like jQTouch - but to be honest, all of the frameworks available are either severely lacking in features, or they have lots of features but implement them very poorly. jQuery Mobile has interface patterns for just about everything you could want to do, it makes it very easy, with you simply using standard HTML tags with some special attributes, particularly the data-role attribute.

There are four main features that I really love about jQuery Mobile though. The first is theming. The framework comes with four pre-made colour schemes which allow you to choose between a wide range of looks for your app without changing more than one attribute in your code. Each theme comes with its own buttons and interface elements which gladly means not much work for you. You can see this in action with one of my apps - by changing literally one letter in the code, I get all these different colour schemes including button styles.

Secondly, it includes custom form elements which really fit in well on just about any mobile platform that you can imagine. This isn't just the very basic text inputs, but also custom select menus which look really beautiful, toggle switches, sliders and even traditionally-shaped search fields to make your user feel even more at home, even if they're working from within the browser.

Thirdly, it has a wide range of different list views. The list view is undoubtedly the main part of the standard mobile application these days, after it was popularised with the iPhone at its launch in 2007. Traditionally however, these views have been pretty mundane, with just text that you can click on to move to another point in the navigation. jQuery Mobile, however, brings you a whole range of views which are really very easy to use, including ones with icons, larger thumbnails, counters, list dividers and numbered lists.

Finally, you get the ability to make beautiful dialogs which again fit in with the various

Let’s take a look.

jQuery Mobile documentation and demo

Now, before I get a bit more technical, I just thought it would make sense to quickly show you a bit of how jQuery Mobile works in practice. One of the interesting things is that the documentation on the jQuery mobile website at http://jquerymobile.com is actually made in the form of a demo – you can see everything working as it shows on a device. So now I’m just going to fire up the iPhone simulator and give you a quick look around.

Go to Features page, scroll around a bit, show bullet points, click home button.
Show off the interesting header styles on the main menu
Go onto theme framework to show some of the different designs
Go onto Pages, and show the dialogs and transitions
Go to Toolbars, show fixed positioning, navbars, automatic back button
Show off the form elements in the form elements gallery, interact with them
Show off some of the more interesting list view types (thumbnails, filter bar, count bubble)

As you can see, there are a wide variety of possibilities for your UI when you’re working with jQuery Mobile. Now I’m sure you’re all wondering how you do this yourself, so let’s get started together.

So how do I get going, I hear you ask?

- Insert the code into your <head> tags

```
<link rel="stylesheet" href="http://code.jquery.com/
mobile/1.0a3/jquery.mobile-1.0a3.min.css" />
<script src="http://code.jquery.com/
jquery-1.5.min.js"></script>
<script src="http://code.jquery.com/mobile/1.0a3/
jquery.mobile-1.0a3.min.js"></script>
```

- And now you just get started with making all the “pages” that make your app

So, you’ve seen some of the cool things that jQuery Mobile will let you do, so know I’d imagine you’d like to know how you do these awesome things. Well, it’s pretty easy! Unless you really want to, you won’t even need to download any files and upload them to your web server. All you need to do is stick a little block of code into your HTML file which will load the Javascript and CSS assets dynamically from the jQuery CDN. Alternatively, if you wish to, you can host the static files yourself, and this is especially applicable if you are going to convert your app into a native one using a framework like PhoneGap or Titanium.

Once you’ve done that, everything is ready for you to get playing. Your app is made up of a series of DIV tags, each with a data-role attribute set to “page” and a unique ID to allow you to move swiftly between pages which are stored within the file. Alternatively, you can store the pages in separate files and when you click on a link to them, the jQuery Mobile library will automatically load them for you using AJAX.

Anatomy of a Page

```
<!DOCTYPE html>
<html>
    <head>
    <title>Title</title>
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0a3/
jquery.mobile-1.0a3.min.css" />
    <script type="text/javascript" src="http://code.jquery.com/jquery-1.4.3.min.js"></
script>
    <script type="text/javascript" src="http://code.jquery.com/mobile/1.0a3/
jquery.mobile-1.0a3.min.js"></script>
</head>
<body>

<div data-role="page">

    <div data-role="header">
        <h1>Page Title</h1>
    </div>

    <div data-role="content">
        <p>Page content goes here.</p>
    </div>

    <div data-role="footer">
        <h4>Page Footer</h4>
    </div>
</div>

</body>
</html>
```


Okay, so now we're just going to dive right in and build a page. As you can see, at the top we're using the wonderfully clean new HTML5 doctype, and then doing all the standard bits and bobs with title's and the like, before including the Javascript and CSS files for jQuery Mobile that I showed you on the previous slide.

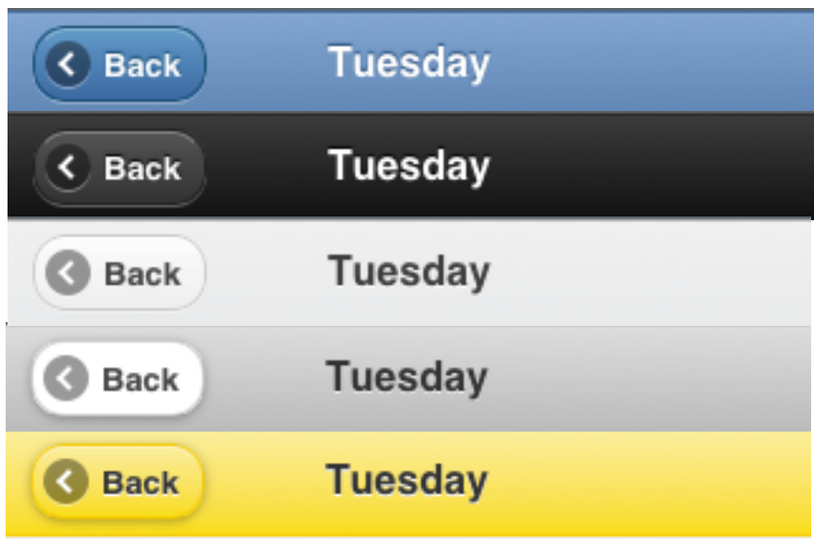
Then we go into the body and get started. As you can see here in various instances, the data-role attribute on a div tag is what really powers jQuery Mobile's layout engine. To make the actual page, we use a div with a data-role attribute of "page". In fact, we can have multiple pages within one HTML file because of this - we just give each DIV a different ID, and we can link between the by creating links with an HREF of hash, followed by the ID.

Show the page running in the simulator at the same time
All the different design elements here simply come from these data roles. We get the little header bar at the top with its classic and beautiful design simply by making a DIV with the right attribute, and the same goes for the main content and the footer. Within each of these tags, we can then add extra things. For example, we can add buttons into the header to do different things and we can add some form controls or a list to the main content area.

Theming

- jQuery Mobile includes a range of different themes which you can make use of in your apps

- They are extremely simple to activate and use, and you can mix together different themes on your page to achieve an eye-catching design.



- Just use the ***data-theme*** attribute:

```
<div data-role="page" data-theme="c">
...
</div>
```


So, I thought it would make sense to perhaps go through the things that I identified as my four favourite features of jQuery Mobile, the first of which is theming. jQuery Mobile comes with a range of colour schemes built in, including all the necessary assets to get going quickly. Of course, you can also make your own themes if you like.

The themes are really simple to activate, requiring only one tiny attribute on your page which will then be inherited on all your other elements, just as you would expect. You are also free to mix-and-match themes by specifying styles for individual elements. The magical attribute which you can use is data-role (say with dash) - simply apply this to the element which has the data-role of page, and you'll be off. The styles will be passed down to its child elements as well. For the attribute's value, you can choose between 5 themes named with letters which you can find in the documentation, although you can manually make your own ones and apparently there will soon by a tool available to quickly roll your own themes. This same attribute can be applied to just about anything to the same effect for finer tuning.

Forms

- Naturally, forms form the backbone of many applications that you might like the build.

- Just create a form as normal:

```
<form action="submit.php" method="POST">
...
</form>
```

- ...and make your elements!

```
<div data-role="fieldcontain">
    <label for="username">Username:</label>
    <input type="text" name="username" id="username" value="" />
</div>
```

```
<div data-role="fieldcontain">
    <label for="targeting">Missile targeting:</label>
    <select name="targeting" id="targeting" data-role="slider">
        <option value="off">Off</option>
        <option value="on">On</option>
    </select>
</div>
```


jQuery Mobile also has great support for forms, with a wide variety of well-designed elements which will fit right into your application, of course working with the themes that I've just spoken to you about. This even includes less common elements like the slider, and supposedly support is soon coming for a date picker element! Forms are very important to just about any application because they provide for interaction and the provision of information by the user, whether that is a username and password or simply a check box!

And yes, if anyone is thinking it, I did notice the little tongue twister on my slide there with "forms form"! Now, I'm sure you're sensing a common thread here, but getting your forms going in jQuery Mobile is also really simple. You just create a normal form tag with the standard post method and action and it will work as you'd expect. Once you've posted it, jQuery Mobile will display the correctly formatted page which is returned by the server using its own interface libraries, so you can just code another quick screen as I demonstrated on the "Anatomy of a Page" slide.

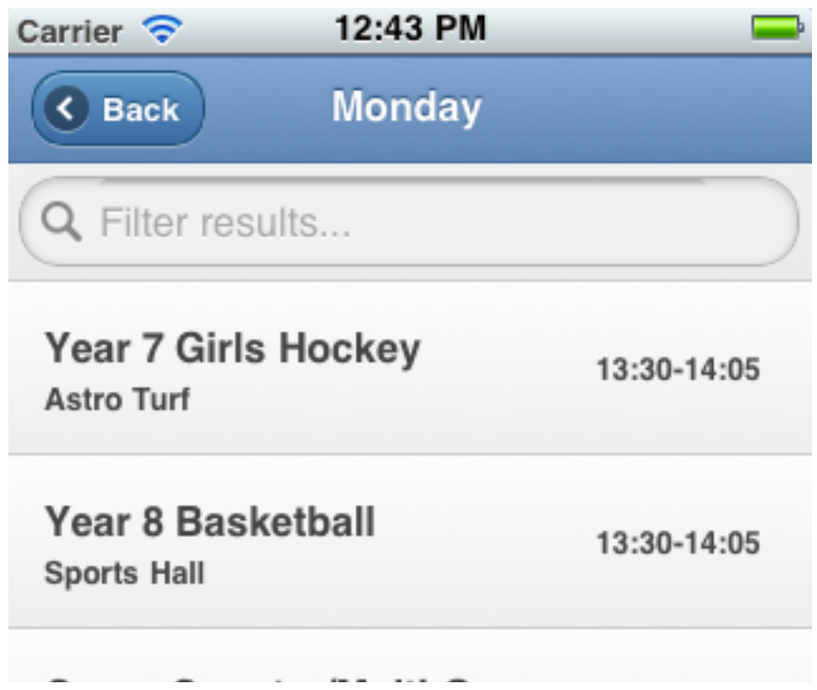
And now you just create your elements. It's really simple, and you basically just create an element as you would on any other page and it will be correctly styled and initialised for you. You can surpress this functionality if you wish however. It is also important that you are careful with the id attributes of your fields - they must be unique to each HTML document, since multiple "pages" may be held within one document which will screw things up if you duplicate your IDs.

The library of course includes support for labels. You should group your fields together in a div of data-role "fieldcontain" and then just make your label and element as usual and they will be styled for you. You can see above the code to make a standard text input element.

You can also get more exotic elements like the toggle buttons which have been popularised by the iOS platform in a similar way. To create a toggle switch, you make a field container element as above, and then just create a simple select element with two options, but give it

List views

- A classic foundation to your mobile application - a way of getting around
- In their most simple form:




```
<ul data-role="listview">
   <li><a href="#about">About</a></li>
   ...
</ul>
```

- But there are a whole variety of more complex list views...

```
<ul data-role="listview">
   <li><a href="#inbox">Inbox</a> <span class="ui-li-count">8
   </span></li>
</ul>
```

```
<ul data-role="listview" data-filter="true">
   <li><a href="#apple">Apples</a></li>
   <li><a href="#banana">Bananas</a></li>
   ...
</ul>
```


The foundation of most applications that we will build as developers is the list view. They were popularised among mobile apps by the iPhone, with the most basic navigation controller because very important and iconic as a way of building your app. If you look at your Android or iOS device, you will see these just about everywhere, ranging from the simple to the complex. You see simply ones which are just lists, but then there is a whole range of more adapted versions including those with headers, counters, sections, thumbnails and even searching systems.

In their most simple form, creating a list view is as simple as creating an unordered list as you can see above using the <ul> element, giving it a special data-role attribute of “listview” and then adding some <li> elements for the list items, often with these linking to other pages or parts within the application.

There are a range of other more complex types. For instance, a popular one is where you have small numbers representing numbers of items to the side of a list - for example, on an email app, you might have the number of unread messages. With jQuery Mobile, this affect can be achieved quickly and easily - to do this, we simply create an unordered list as shown above, but we then add a element within the unordered list item with a special class of “ui-li-count” which represents the counter. This will then be rendered by jQuery Mobile in the expected way.

Another popular thing which you might like to do is creating a list which you can search through, as shown in the screenshot on the screen. Without a framework, this would require a lot of effort, forcing you to create a search control and build you own searching behaviour, but with jQuery Mobile, it is incredibly easy to create a search control which works perfectly as users would expect. Within jQuery Mobile, literally all you have to do to access this behaviour is to add an attribute of “data-filter” with value “true” to the unordered list elements and it will just work. You can see this code example on screen as well.

Dialogs

- The dialog is another important UI element which has been used since the beginnings of GUI interfaces
- jQuery Mobile has built-in support for very nice looking dialogs.

- You create a dialog just like any other jQuery Mobile page...

```
<div data-role="page" id="about">
    <div data-role="header"><h1>About</h1></div>
    <div data-role="content">
        ...
    </div>
</div>
```

- And then you open it **almost** like any other...

```
<a href="#about" data-rel="dialog">About</a>
```


My final favourite feature to show you from the jQuery Mobile library is the dialog. Dialogs were completely missing from the jQTouch library that I used to use and have showed you in previous presentations, forcing you to either use a basic page or to use ugly and clunky Javascript popups. Rather, the popup dialogs offered in jQuery Mobile are rather nice looking and bit really well into the rest of the framework. Dialogs are used a lot in applications in a variety of ways - as confirmations, about screens, and even just for showing more details about something. They are really easy to make in jQuery Mobile.

Dialogs are really good in jQuery Mobile because they can literally contain anything which a standard page does and they are loaded in much the same way. As such, they support all the features of the library like theming and the different transitions and interface styles. To create a dialog, you simply create a page with a unique ID as shown on the anatomy of a page slide earlier - for example, on screen you can see the code for a page I've made called "About". You then open it like any other page by creating a link with an HREF attribute of a hash followed by the page's ID, or you can even link to an external page. The only difference is that you had a "data-rel attribute" with the value "dialog" to make it appear with a dialog as in the screenshot, with a dark background and rounded corners. Once the dialog is opened, any links in it will appear as any other page would, and you can even simply create a close button for the dialog using a link with a "data-rel" attribute of "back".

Thank you.

- You can grab this presentation and all the materials from GitHub at http://bit.ly/jqm-devnest

- If you have any questions, feel free to grab me here or contact me however you like.

Email: tim.rogers@carnochanrogers.co.uk
Twitter: @timrogers
Phone: 01708 873928 *(leave a message)*

Thanks for taking the time to watch my presentation, and I hope you enjoyed it and found it informative. I've shown you some of the great things that you can use to make a really excellent mobile app using jQuery Mobile - with this library, it really is simple to make a good-looking and functional app. If you'd like to grab this presentation and take another look, you can get it on GitHub at bit.ly/jqm-devnest. Once you're there, click the Downloads button and grab the Basic package if you just want the Keynote file that I'm using and the code samples, but if you want a more fully featured package with various presentation types like PDF, QuickTime movie and PPT file for PowerPoint, grab the full package.

I'm quite happy to take any questions now if you have any, so fire away. Otherwise, come and speak to me later or contact me via email, Twitter or phone as detailed on the screen - I won't bite!

Thanks again for listening!