UNIVERSITY of STIRLING

Mobile Applications

Cordova & jQuery Lab 5

This lab contains a checkpoint at the end. Please contact any lecturing staff when you reach this point to receive your credit.

The basic aim of this lab is work on designing your own first cordova project and link this to jQuery.

Develop your first own page

Firstly, create a new cordova project. Remember, you want to do this in a directory you have write access too and which is not local to the machine you are using. Alongside the directory for last weeks project would be a good idea.

```
cordova create cordova2.
```

This will create a project called cordova2. Check that the new directory has been created, then change into it. The www folder should have an index.html file and directories for Javascript and CSS (amongst others). Add the platforms browser (and Android if you wish) by executing

```
cordova platform add browser.
```

Let's simplify the app somewhat. Change the index.html, taking out the large comments. Also comment out the stylesheet link in the header (comments in html are between the tags <!-- and --> Finally, edit your file to include:

```
<body>
<div id="deviceready">
      <h1>Hello World!</h1>
      <button id="hellobutton">Say Hello</button>
    <p id="hellop"></p>
</div>
…
```

Then let's simplify the Javascript file too. Change its content to be:

```
var app = {
 initialize: function() {
    document.addEventListener('deviceready',this.onDeviceReady.bind(this), false);
    },

 onDeviceReady: function() {
    console.log("received deviceready");
    document.getElementById("hellobutton").addEventListener("click",
this.printhello);
    },

 printhello: function() {
    console.log("in printhello");
    document.getElementById("hellop").innerHTML = "Hello World!";
    },
};
app.initialize();
```

This Javascript code defines an object (app) which has three methods. The method initialize is run as the file is executed (on page load) and registers an event handler for deviceready. This is a key Cordova event which is raised once the app is fully loaded on the device. There should be no user (i.e. code) initiated actions until after this event is received.
Once this has happened an event handler for 'click' of the hellobutton is registered (note: there will be no action taken if the button is clicked before deviceready is received).
The printhello function is executed if a user 'clicks' the button.

You should be able to execute the app in a browser by executing the command
`cordova run browser`.

Verify the behaviour.

Adding jQuery to the project

Before we continue developing this app, let's add jQuery look and feel to it. For this you'll need to download three jQuery files:

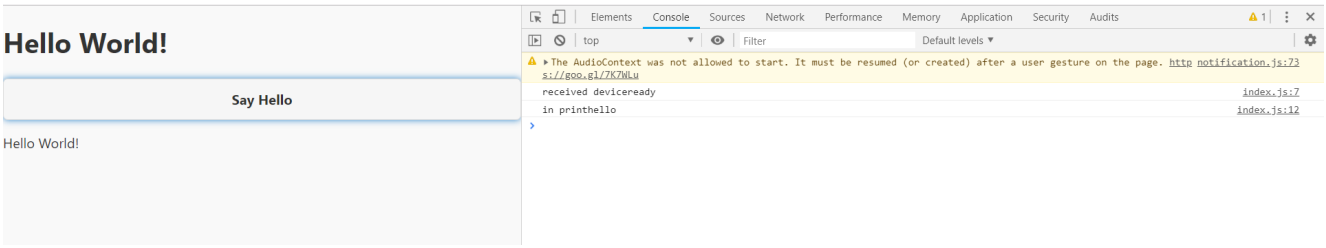
Download https://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css
Download https://code.jquery.com/jquery-1.11.3.min.js
Download https://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js

Save them to your project folder (say inside the js subfolder).

Finally, you will need to link these files in your project (in the index.html): As part of the <head> tag add (or simply edit the tag you commented out before and remove the comment tags):

```
<link rel="stylesheet" type="text/css" href="js/jquery.mobile-1.4.5.min.css" />
<script type="text/javascript" src="js/jquery-1.11.3.min.js" id="cordova-jquery"></script>
<script type="text/javascript" src="js/jquery.mobile-1.4.5.min.js"></script>
```

New try this out and see if your page look has changed. You might want to check the Chrome Developer Tools (Ctrl-Shift I) for error messages. Note the log messages being displayed in the Console part of the Chrome debugging interface. Verify these against your code. Make some change and check that the code behaves as you predict.



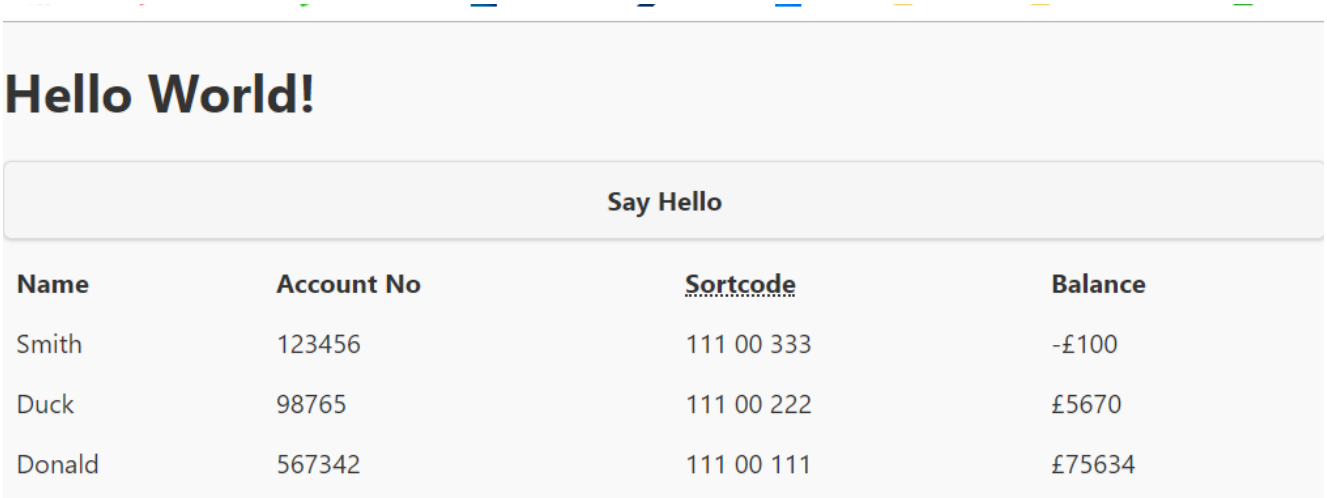
We will explore jQuery a little more in a lecture.

Adding a Table to the app

Add HTML tags to your page's HTML file displaying a HTML table to represent some bank account information. Make sure you have a header row and at least four further rows with some data, and columns for account holder name, account number, sort code, and balance.

Verify the results.

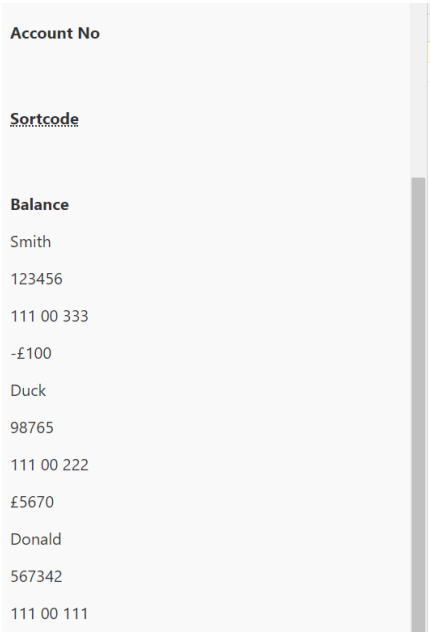
Next, let's give the table the jQuery treatment.
Change the table opening tag to include the following attributes:
data-role="table"
data-mode="reflow"
id="account-table"
class="ui-responsive"

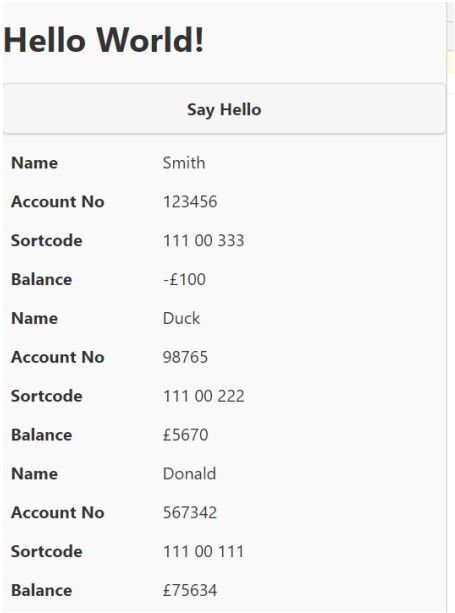


jQuery Tables use a responsive design, i.e. the layout of the table adjusts to the width of the screen/window available. The jQuery table feature initialises when the data-role="table" attribute is added to the <table> tag. The data-mode attribute specifies how the table design is adjusted. With the value "reflow" (default), the table columns at narrow widths are presented as a formatted block of label/data pairs. This is suitable for tables where no information can be omitted and with complex

or lengthy data formatting that doesn't need comparison across rows of data. Try it out by narrowing the browser window.

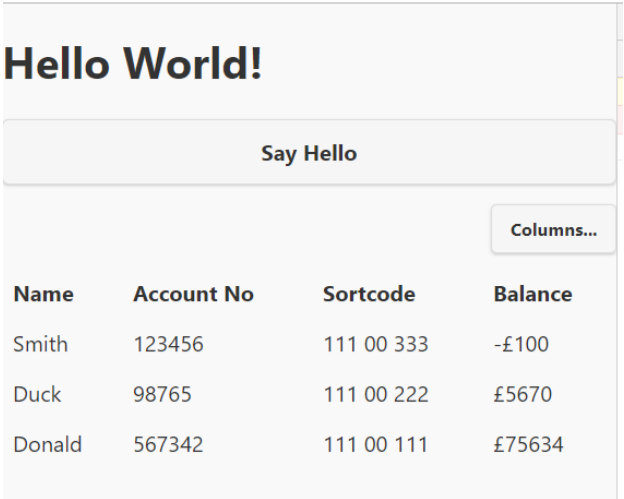
It looks a bid odd, doesn't it (see screenshot on the right)? We will need to introduce <thead> and <tbody> tags. The former should surround the header row, and the latter the rest of the rows in the table.





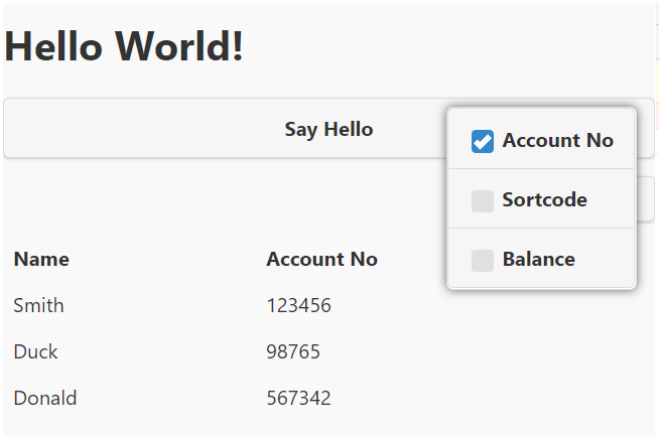
Further, the id attribute simply assigns and id to the table, which can be used to identify it in associated Javascript code. The class attribute specifies a CSS class to be applied to the table.

Next let's change the formatting if space is tight in such a way that certain columns are compacted and/or simply omitted. For this change set data-mode to "columntoggle". Does it work?



Note that there is also a new button appeared at the top of the table allowing users to select which columns they want to see. At the moment you cannot select anything when you push it. Actually, no columns are actually removed neither. Let's fix that.

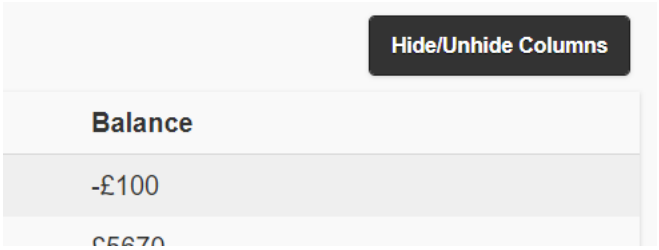
Add an attribute data-priority to the <th> tags and give it a priority number as value, e.g. data-priority=”1”. Make sure you assign each column a different value. If you leave out this attribute from a column that means that that column cannot be removed. Note that all columns with the data-priority attribute will now be available for (de)selection via the Columns button.



Finally, can you change the behavior of your button in such a way that pushing it, will change the name of one of the account holders in the table?

Improve the look of your table

In the previous steps we used the jQuery data mode ‘columntoggle’. Add the class ‘table-stripe’ to your table. Is it easier to read?
Next, change the text of the button to select/deselect columns to “Hide/Unhide Columns”. The button's default text is "Columns..." but can be set by adding the data-column-btn-text attribute to the table to the text string you want in the button.



Add a header and footer to the page

Next, redesign your page to include a header and a footer part (as well as a main body element). You will find an example of this in the jQuery lecture. Try setting data-position="fixed" for the footer. Can you spot the difference?

Once you have a single page with this structure, create a second such page. Allow for navigation between the two pages as shown in the lecture.
Change the simple link (as shown in the lecture) to be a button. Add an icon to the button to indicate going forward (page 2) or going back (to page 1). You can find a list of available symbols at http://demos.jquerymobile.com/1.4.5/icons/

Next let's try and revamp the two page solution into a single page, but with navigation tabs. JQuery offers tabs and a navbar to achieve this.

The basic structure of your page needs to be:

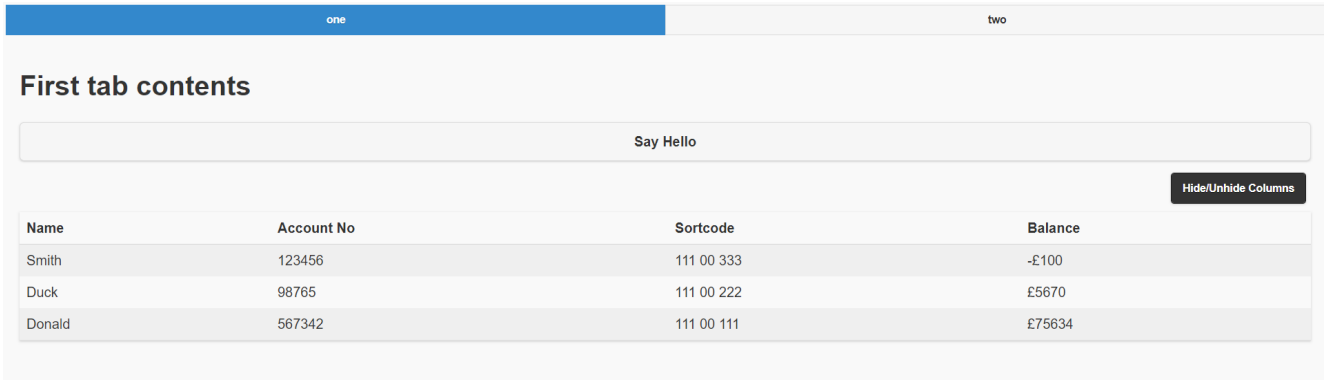
```
<div data-role="tabs">

     <div data-role="navbar">
          <ul>
               <li><a href="#one" data-theme="a">one</a></li>
               <li><a href="#two" data-theme="a">two</a></li>
          </ul>
     </div>

     <div id="one" class="ui-content">
          <h1>First tab contents</h1>
     </div>

     <div id="two" class="ui-content">
          <h1>Second tab contents</h1>
     </div>
</div>
```

Redesign your page in such a way that the table is shown on one tab, and some other content on the second tab.



Checkpoint