Mobile Application Development

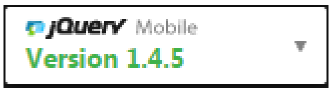
Session 8 – Activities

The final product(s) for this session can be found at:
http://www.dcs.bbk.ac.uk/lo/mad/madexamples/session8/classactivities/zedlandhotels/zedland-hotels.html

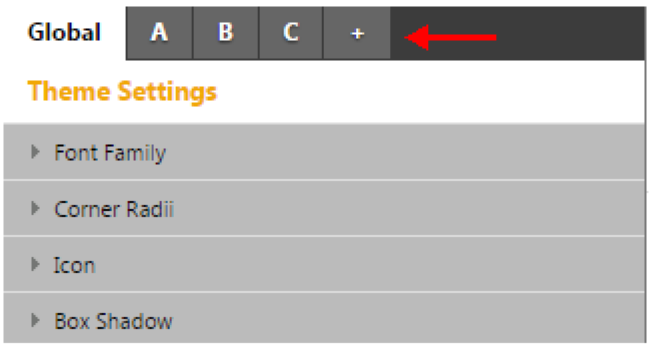
1. Create and Implement a ThemeRoller Swatch

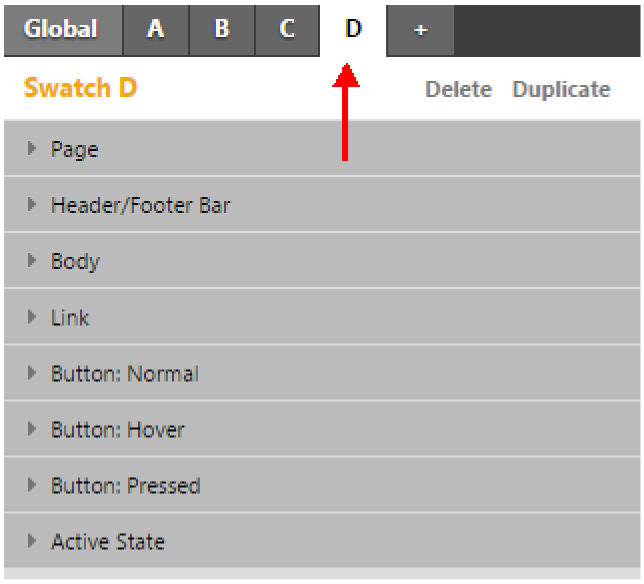
Creating a Swatch

Go to http://themeroller.jquerymobile.com/ and open the ThemeRoller application. Make sure the jQuery Mobile option is set to the latest version of jQuery Mobile.



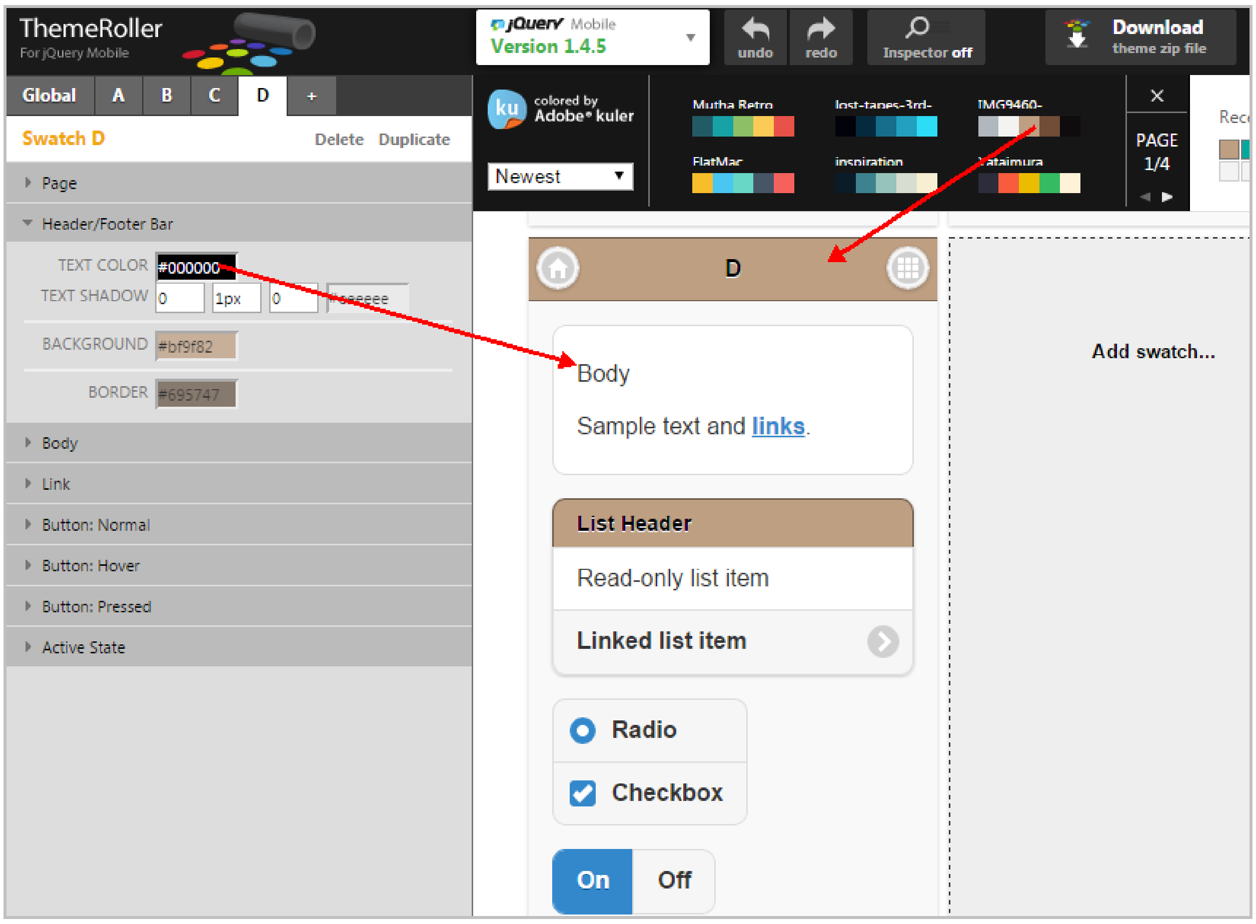
Next, click the + in the left column to create a new theme. This will be swatch *D* the first time you create a new swatch.





Add colours to your new swatch. This is done by either dragging colours from the colour palette or the Adobe Kuler colour palette, or by entering colours for the element you are styling as hexadecimal values.

Note that your colours need to be based on a valid colour scheme that is derived from colour theory, the same as for any webpage. To create a schema, you will need to use a colour picker tool such as Adobe Kuler. For usability, it is recommended that you keep the colour scheme for any mobile app as simple as possible. In the small screen, there is just not enough room for the interplay of several colours at once.



TEXT COLOR #000000
TEXT SHADOW 0 1px 0 #eeeeee
BACKGROUND #bf9f82
BORDER #695747

Applying a Theme

Once you have created your theme, click the download tab at the top of the screen. You will then be presented with a dialogue asking you to name the theme, and instructions for how to include the theme in your application. Follow these instructions, making sure you include the links to the provided CSS files before the main jQuery Mobile CSS file.

```
<head>
<title>Zedland Hotel Finder</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="zedland-custom.min.css" />
<link rel="stylesheet" href="jquery.mobile.icons.min.css" />
<link rel="stylesheet" href="//ajax.googleapis.com/ajax/libs/ →
jquerymobile/1.4.3/jquery.mobile.min.css" />
```

Like all other linked files or scripts, this declaration needs to be made in the head of every page in the application, not just the landing page.

Save your pages. The theme is now ready for use.

We could apply the theme at page level. However, this would time-consuming, as we would have to make changes to each page in our application.

```
<div data-role="page" id="home" data-theme="d">
```

It would be much better if we could apply our theme at the application level, to all pages. To apply the theme across all pages we need to make changes to the jQuery Mobile default configurations.

Applying a Theme Globally
Create a new JavaScript file. Name the file *config.js*. Create a script tag to link to this file in the head of each page in your application.

Because we are going to make changes at framework level, we will need to use *mobileinit*. The *mobileinit* event fires before the framework has had a chance to fully load, so allowing us to make changes to the framework.

In *config.js*, add the following event handler.

```
$(document).bind('mobileinit', function () {
 , , ,
});
```

Once we have this event handler, we can instruct jQuery Mobile what changes we want to make to the default configuration. In our case this is to change the theme from the default to our new theme, *d*. We make this change by accessing the *options* of the *prototype* of the *page* widget.

```
$(document).bind('mobileinit', function () {
  $.mobile.page.prototype.options.theme  = "d";
});
```

Save *config.js*. Upload your amended and new files to the DCSIS server. Test your application in your mobile device.

Applying Custom CSS

ThemeRoller essentially allows us to make changes to colour associated features of an application. It does not allow us to make significant changes to layout features such as margins, padding, width, height, etc. To make changes here, we need to be able to access jQuery Mobile's CSS classes, and to tweak their dimensional, and positional properties. In the following example, we will do just that in order to manipulate the width of a particular subset of listview elements that are positioned in the content section of our application.

Open *styles.css*. Create the following markup. This will change the margin property in our primary and secondary navigation listviews.

```
.ui-content .ui-listview{
    margin: 5% 10% 0 10%;
}
```

Save *styles.css*. Upload the file to the DCSIS server, and test your application in your mobile device.

.