

# Building and packaging mobile apps in Dreamweaver CC

## Requirements

### Prerequisite knowledge

Previous experience with Dreamweaver, jQuery Mobile, and PhoneGap will help you make the most of this tutorial.

### Additional required other products

- PhoneGap Build
- jQuery Mobile

### User level

Intermediate

### Required products

- Dreamweaver ([Download trial](#))

### Sample files

- [meridien\\_mobile\\_final.zip](#)

Using Dreamweaver CC, you can create native mobile applications with jQuery Mobile web pages and then use PhoneGap Build to package the applications for various mobile platforms.

In this tutorial, you'll create a mobile application that display restaurants based on a type of cuisine and identify the location of a restaurant on a map. You'll also learn how to package this application in a format that can be used on multiple mobile platforms.

**Note:** If you are stuck at any point in the tutorial, open the `index.html` page from the sample files, and look at the code.

The main steps in creating and packaging the mobile application are:

- Create a jQuery Mobile web page using the jQuery Mobile option in the Dreamweaver CC Insert panel

The jQuery Mobile framework is a JavaScript framework that enables you to quickly build websites and applications for mobile devices. This framework works on the vast majority of all modern desktop, smartphone, tablet, and e-reader platforms.

- Use jQuery Mobile themes and the jQuery Mobile Swatches panel in Dreamweaver CC to customize the look and feel of your application
- Use PhoneGap APIs to access native features of mobile devices—in this case to point to the location of a restaurant on a map.

PhoneGap exposes APIs to make use of device features such as the accelerometer, contacts, and more.

- Package the application using PhoneGap Build for use on various mobile platforms

PhoneGap Build lets you build packages for platforms you're targeting using the most up-to-date SDK. You can package mobile applications using PhoneGap Build without having to maintain native SDKs.

## Creating a jQuery Mobile page in Dreamweaver CC

You will use the following steps to create a jQuery Mobile page in Dreamweaver CC:

- Create a new site in Dreamweaver CC
- Insert a jQuery Mobile page that acts as a landing page for various restaurant categories
- Create child pages for each of the categories
- Customize the appearance of the site using jQuery Mobile Themes

### Create a new site in Dreamweaver CC

1. In Dreamweaver CC, choose Site > New Site to create a site.
2. In the Site Setup dialog box, type a name for the site; for example, type **Meridien** (see Figure 1).

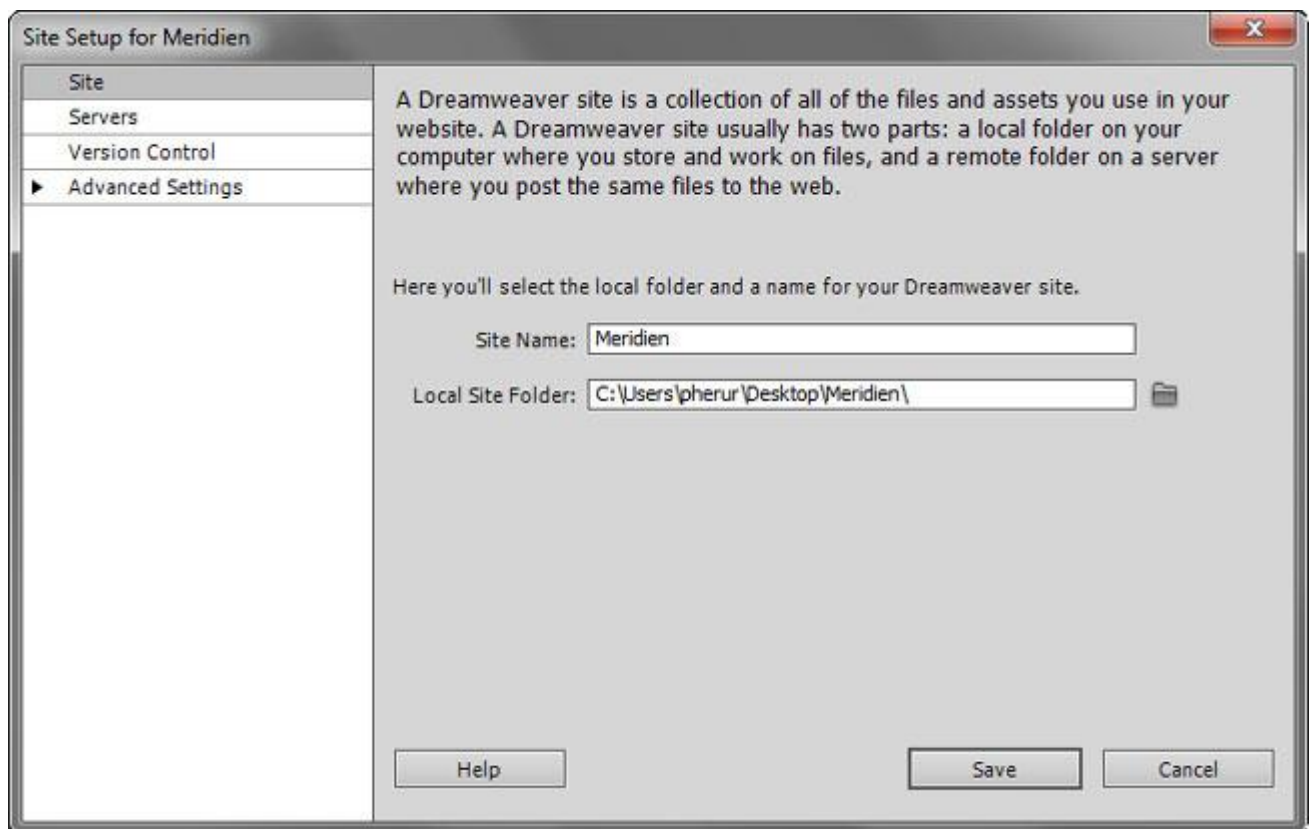


Figure 1. Type a site name along and specify a local site folder.

3. For Local Site Folder, specify a location for the files that you will use in your application.
4. Click Save.
5. In the folder you specified for Local Site Folder, copy the following folders from the sample files: `css`, `img`, `js`. These folders contain the CSS, image, and JavaScript files that you will use when creating pages later in this tutorial.

### Add a page containing the list of cuisines types

Next, you'll create the landing page that will list restaurants by their cuisine type (category).

1. To create a new HTML file, choose File > New.

2. In the New Document dialog box, select Blank Page. For Page Type, select HTML. For Layout, select None. Ensure that HTML5 is selected as the DocType (see Figure 2).

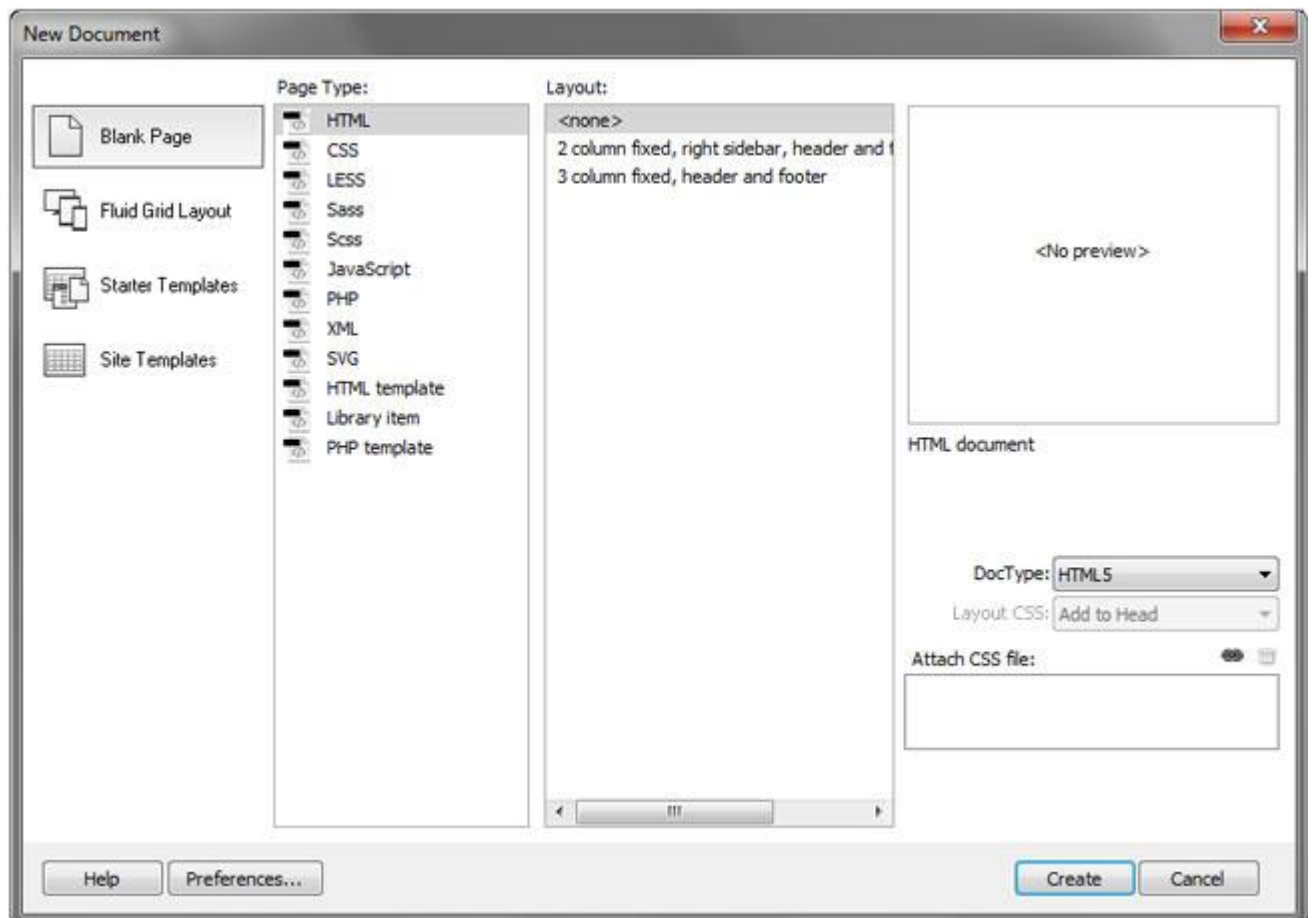


Figure 2. Creating a new HTML page.

### Insert a jQuery Mobile page

1. Choose Insert > jQuery Mobile > Page.
2. In the jQuery Mobile Files dialog box, add CDN links to your jQuery CSS and JS files, or provide the path to local copies. For this example use CDN links and select Combined for CSS Type (see Figure 3).
3. Click OK.



Figure 3. Specifying the location of jQuery Mobile files.

4. You are prompted to add a header and footer for the page. For the first page, include only the header tag by selecting Header (see Figure 4). Click OK.



Figure 4. Setting up the page header.

5. Notice that the jQuery page with a default theme has been added. Click Live to view the Live view rendering (see Figure 5).

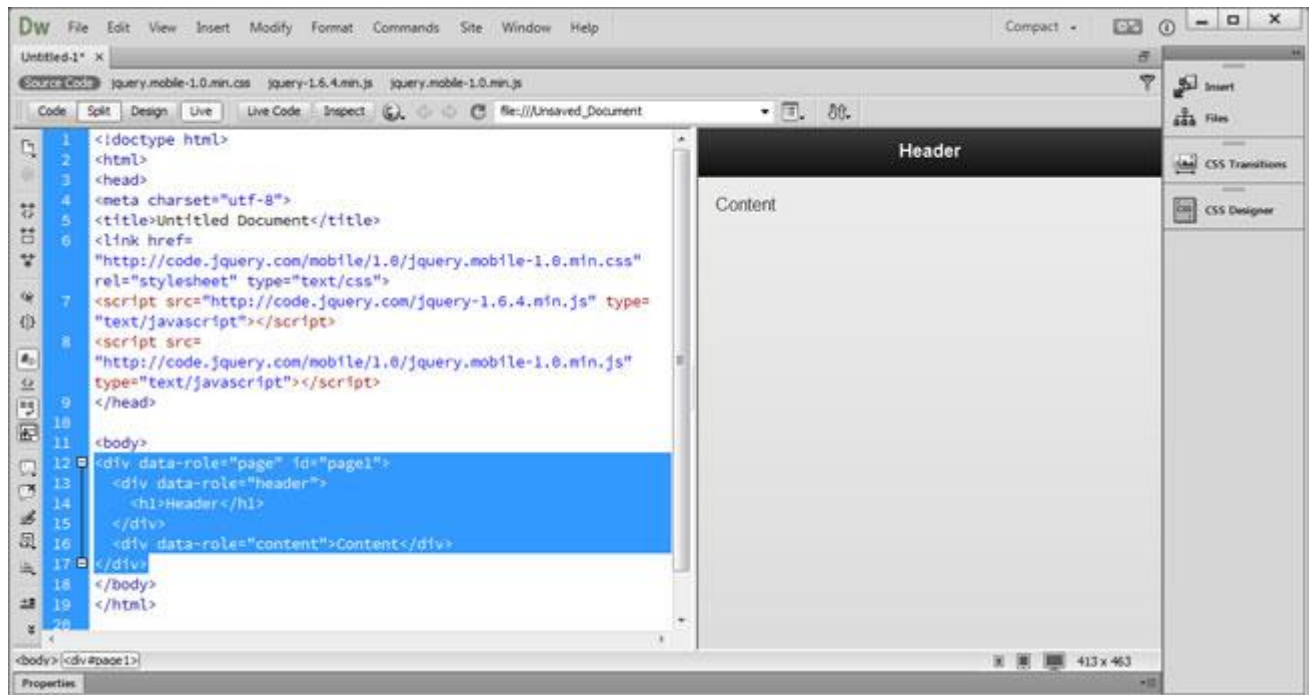


Figure 5. Checking Live view.

6. In Code view, change the header to **Restaurant Picks** (see Figure 6).

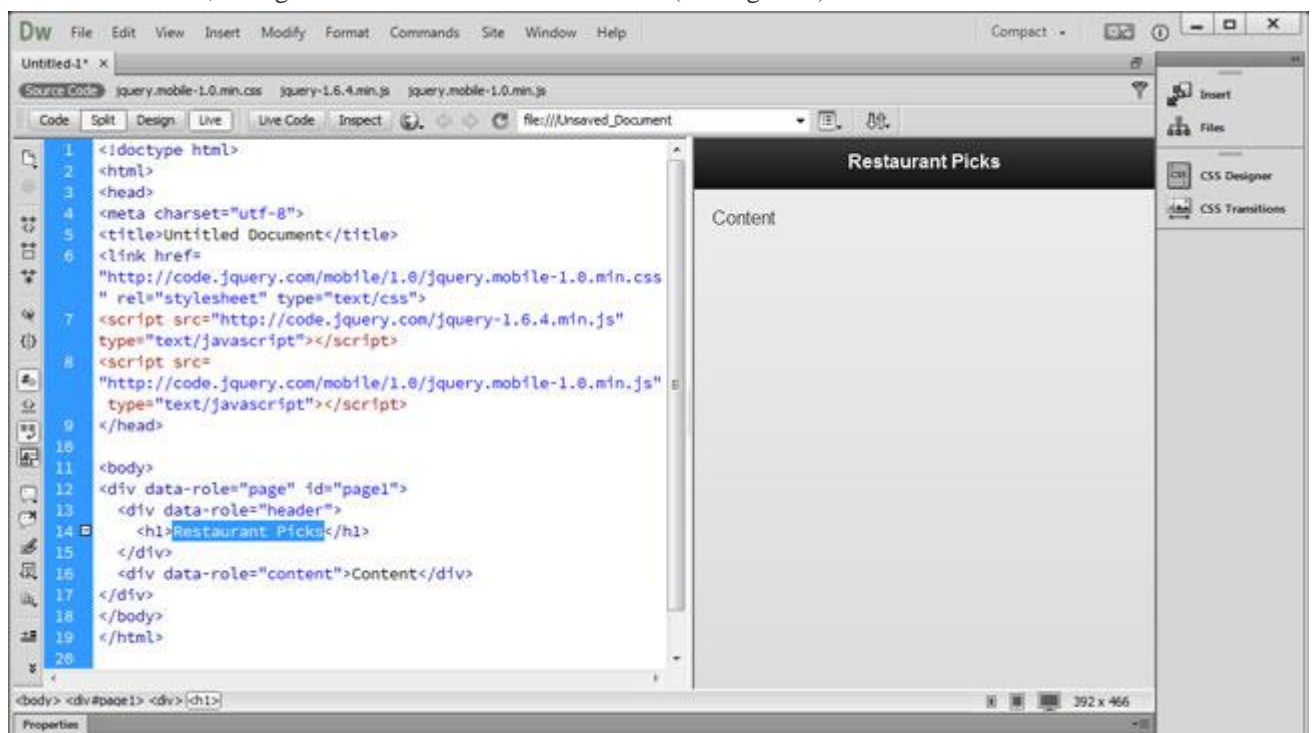


Figure 6. Changing the header.

7. Replace "Content" with an unordered list that will display the categories of restaurants by positioning your cursor in Code view where you want to add the List View widget. In this case, delete the word "Content" and place your cursor at that location.

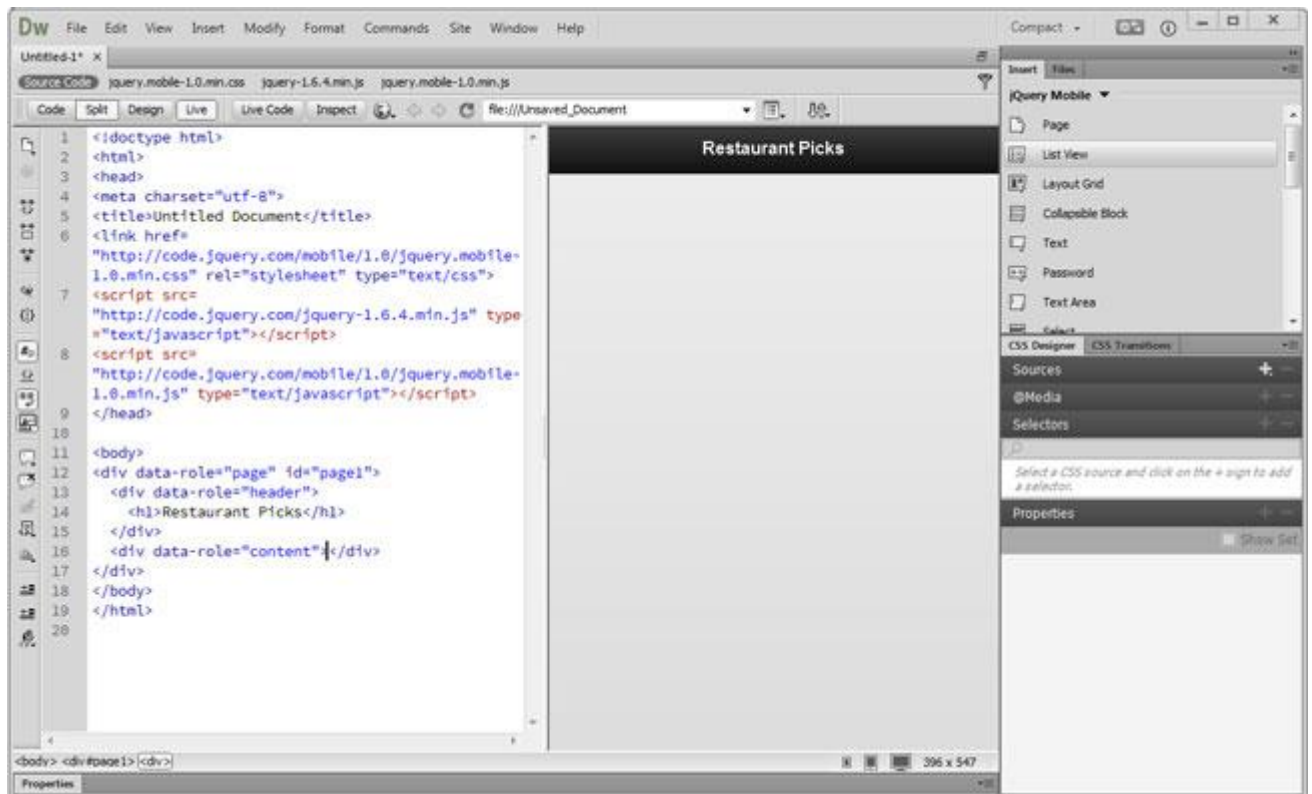


Figure 7. Preparing to add the List View.

8. In the Insert panel, click List View (see Figure 7).
9. For Items select 5. Select Inset and click OK. A list with 5 elements is added to your page (see Figure 8).



Figure 8. Adding the List View.

10. In Code view, replace the word "Page" in each instance of the list with a cuisine of your choice; for example, use Asian, BBQ, Italian, Mexican, and Vegetarian (see Figure 9).



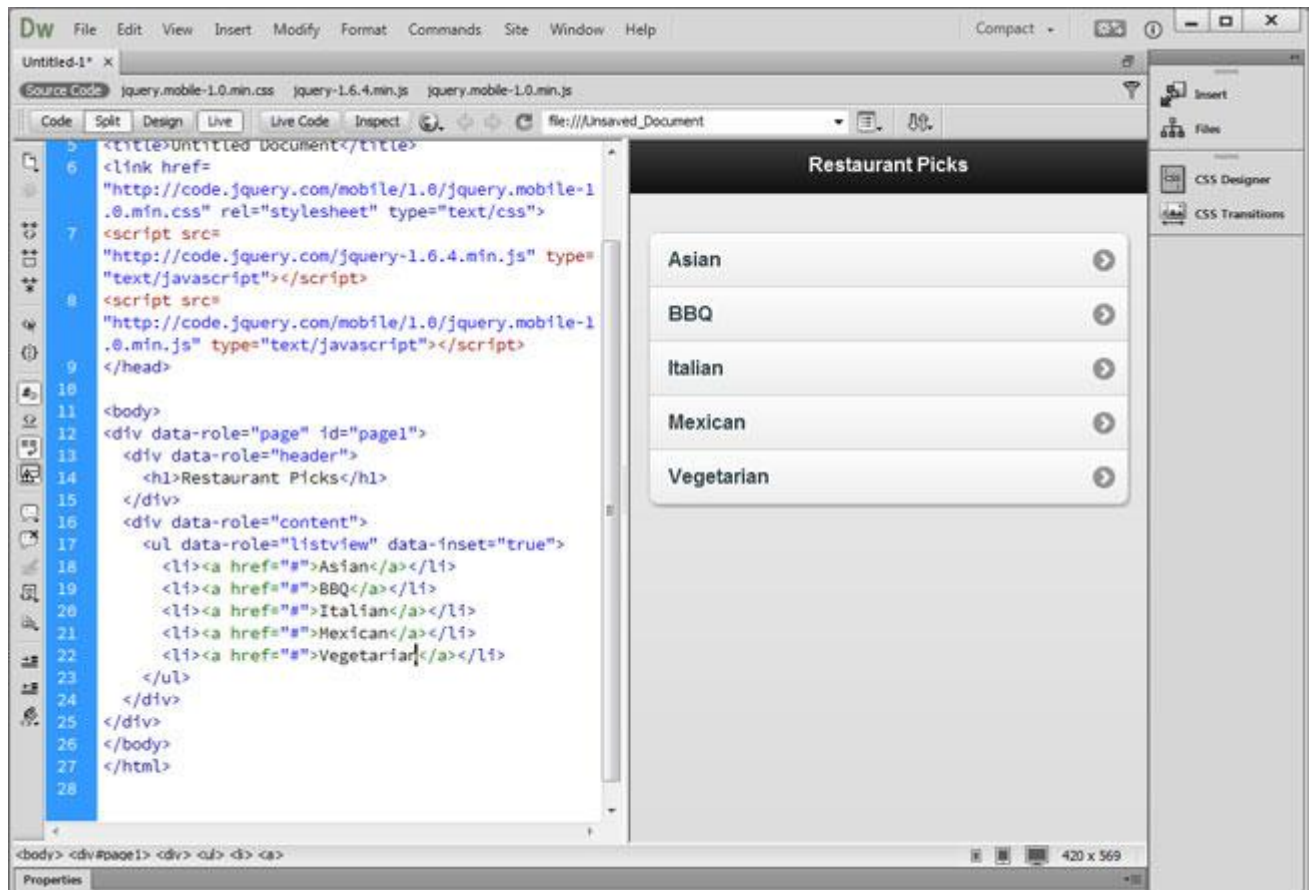


Figure 9. Adding the cuisine options.

11. In Code view, position your cursor above the `<ul>` tag.
12. Choose Insert > Image > Image.
13. Navigate to the img folder and select logo.png (see Figure 10).

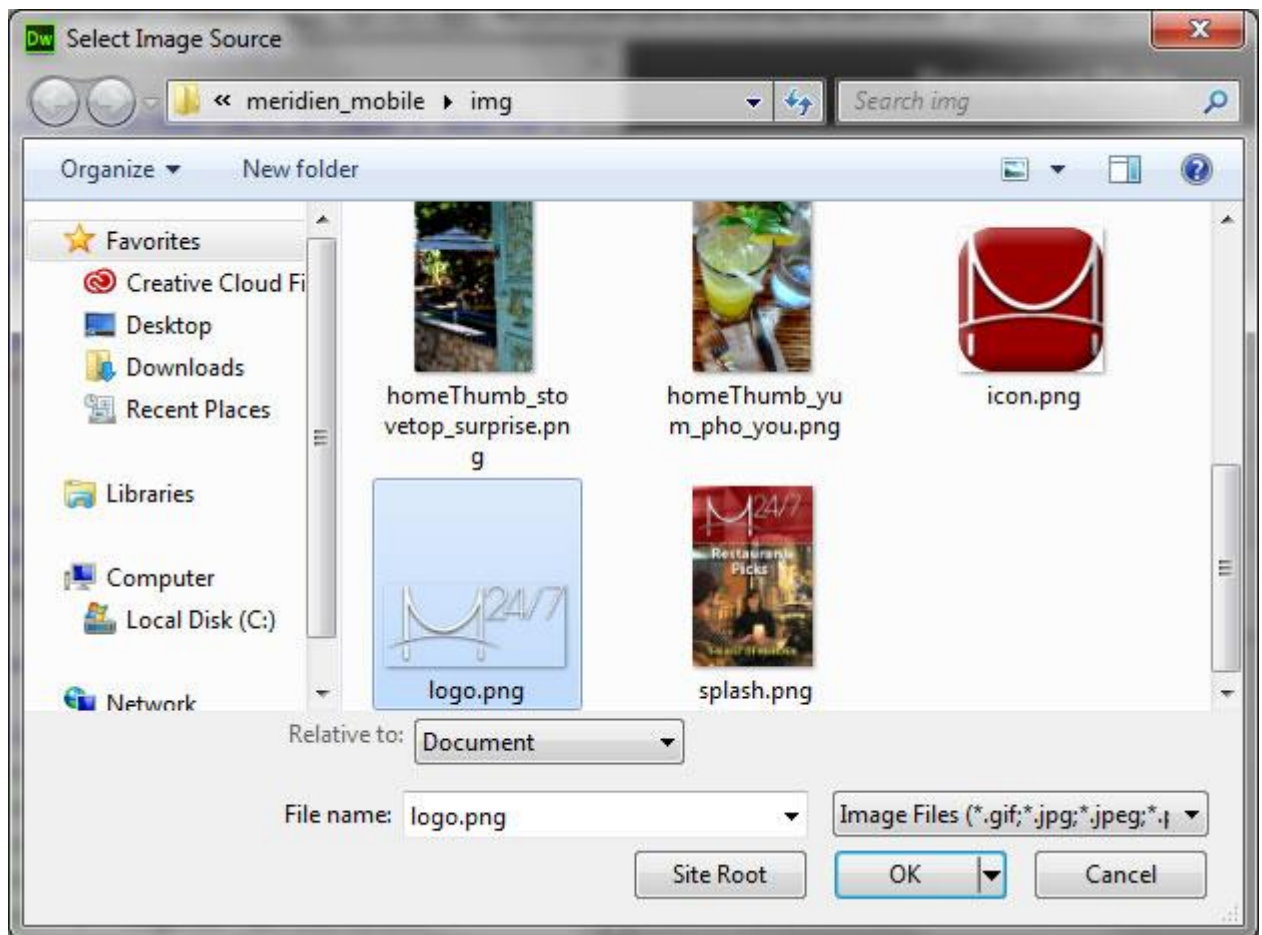


Figure 10. Selecting logo.png.

14. Copy the file into the local `img` folder (see Figure 11).

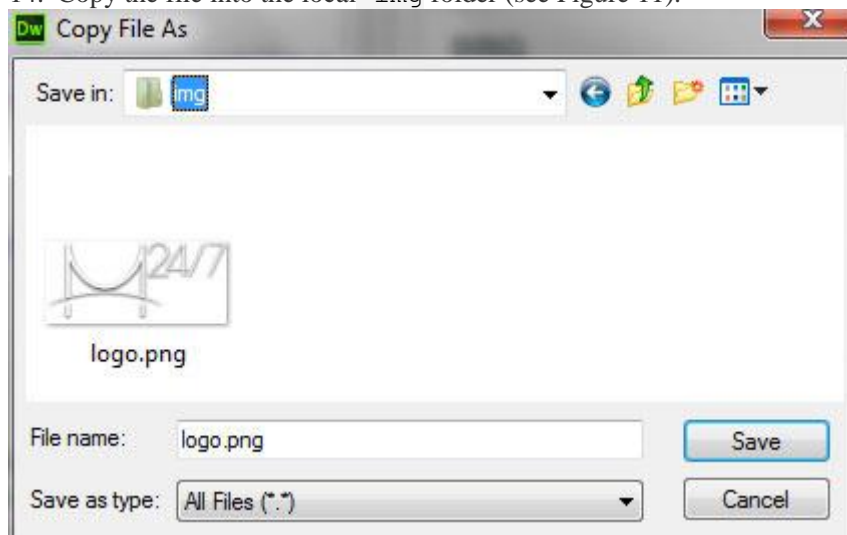


Figure 11. Copying the image file.

The image will appear in Live view (see Figure 12).



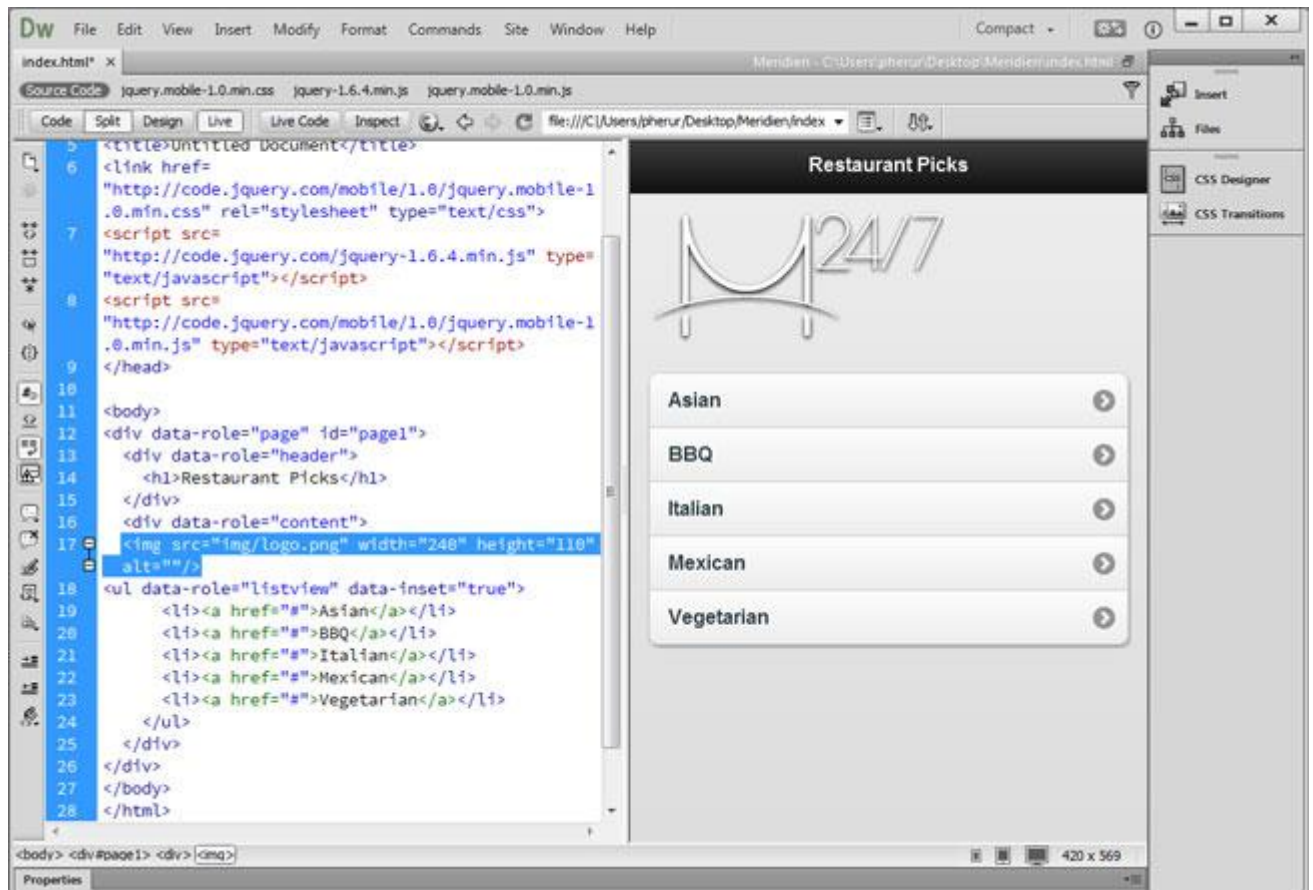


Figure 12. The image in Live view.

### Create child pages for each of the menu items

Now you are ready to link each category (cuisine) to a page that displays a restaurant for that category. For example, the Asian category will display an Asian restaurant (see Figure 13).



Figure 13. Page for Asian cuisine.

1. In Code view, position your cursor at the end of the `</div>` tag that is immediately above the `</body>` tag.
2. Use the jQuery Mobile panel to add another page with the page ID "asian". Ensure that you select the option to add the footer (see Figure 14).

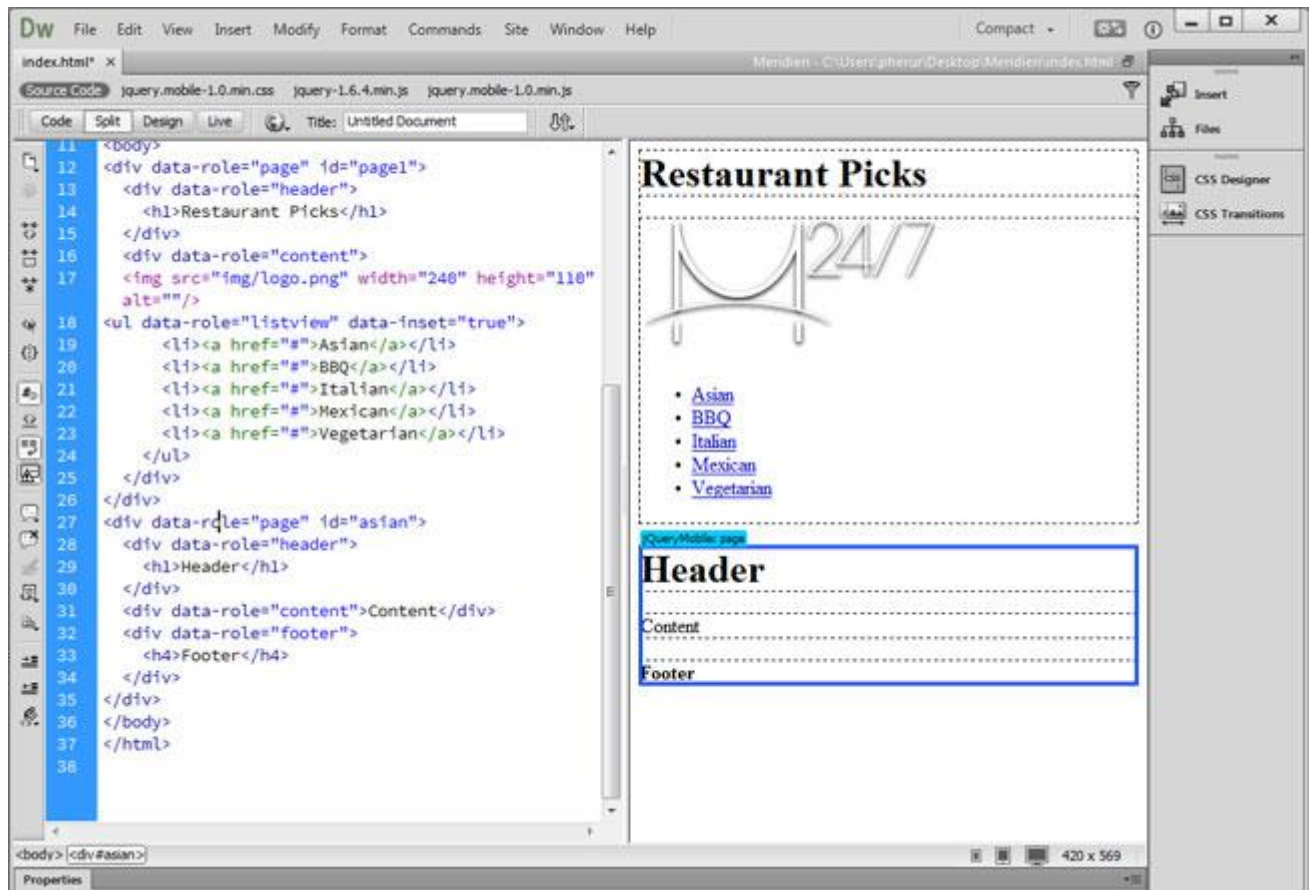


Figure 14. Adding a new child page.

3. Link the list item in the first page to this newly created page by adding the ID (asian) to the href attribute: `<a href="#asian">Asian</a>`
4. Change the header and footer text to **Restaurant Picks**, and **Meridien 24/7**, respectively.
5. To add a Back button, in Code view, add a data-add-back-btn attribute to the page div tag: `<div data-role="page" id="asian" data-add-back-btn="true">`
6. Delete the word "Content", and ensure that your cursor is at the deleted location.
7. To add the buttons for "Map It" and "Call", choose Insert Panel > jQuery Mobile > Button, and do the following (see Figure 15):
8. For Buttons, select 2.
9. For Button Type menu, select Link.
10. For Layout, select Horizontal.
11. Edit the labels for both the buttons in Code view and change them to "Map It" and "Call":

```
<div data-role="controlgroup" data-type="horizontal"><a href="#" data-role="button">Map It</a><a href="#" data-role="button">Call</a></div>
```

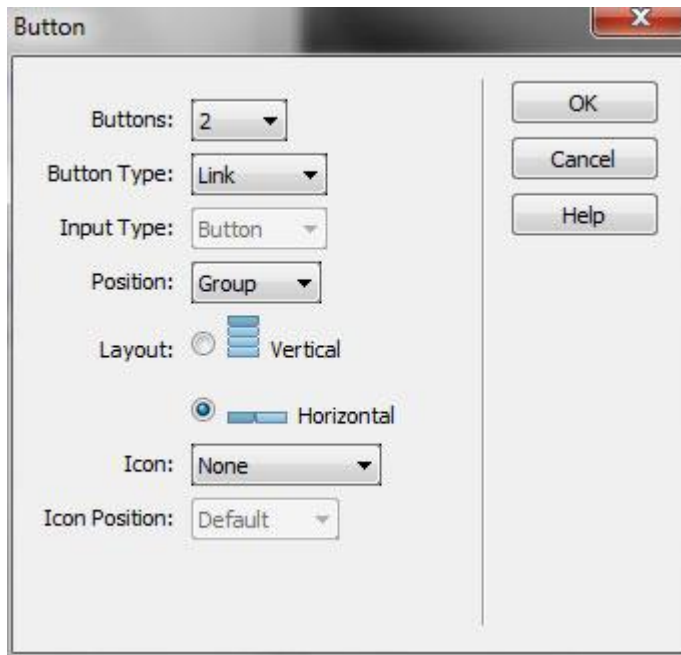


Figure 15. Adding jQuery Mobile buttons.

9. Under `<div data-role="content">`, copy the following code:

```
<div class="addr">
<h2>Sushi Heaven</h2>
<p>5229 Morning Glory Rd <span class="details">Hours: 11am -
10pm</span><span class="details">Price: $$$</span><span
class="details">Web: sushiheaven.mer</span></p>
</div>
```

10. Copy the following code below the `</div>` tag for the buttons you inserted previously.

```
<p class="clearboth">Forget grabbing your sushi from little boats
sailing past your table on a moat. We send our signature rolls your
way in baskets floating through the air, suspended by minature
balloons and blimps. Go ahead, reach up and grab the roll of your
choice, as if you're plucking it out of heaven itself.</p>
```

Repeat this procedure to add pages that are linked to their corresponding categories in the main menu.

## Theming a jQuery Mobile page

Now you can style the pages that you've created.

1. Choose Window > jQuery Mobile Swatches to open the jQuery Mobile Swatches panel (see Figure 16).

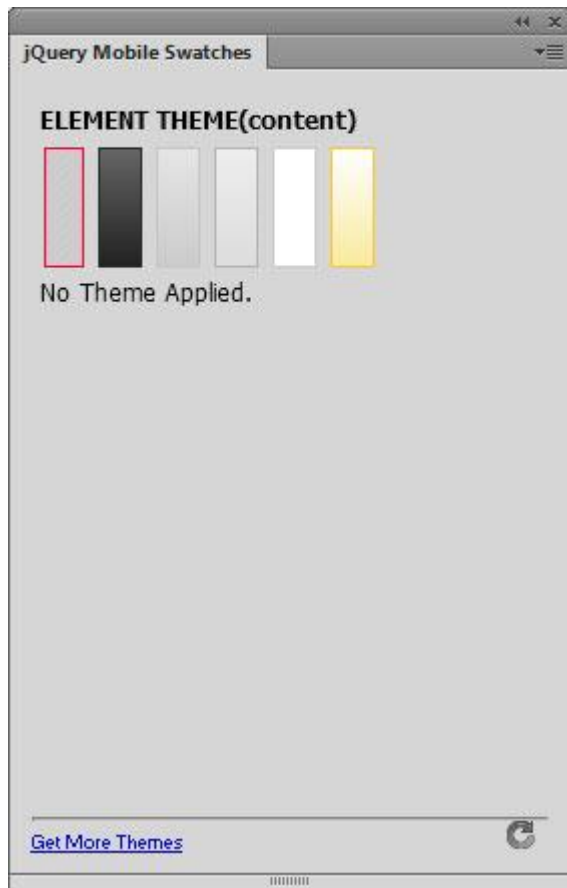


Figure 16. The jQuery Mobile Swatches panel.

2. To theme the content of the first page, select the content in Live view and click a theme present in the panel. In this case, choose the dark theme (see Figure 17).

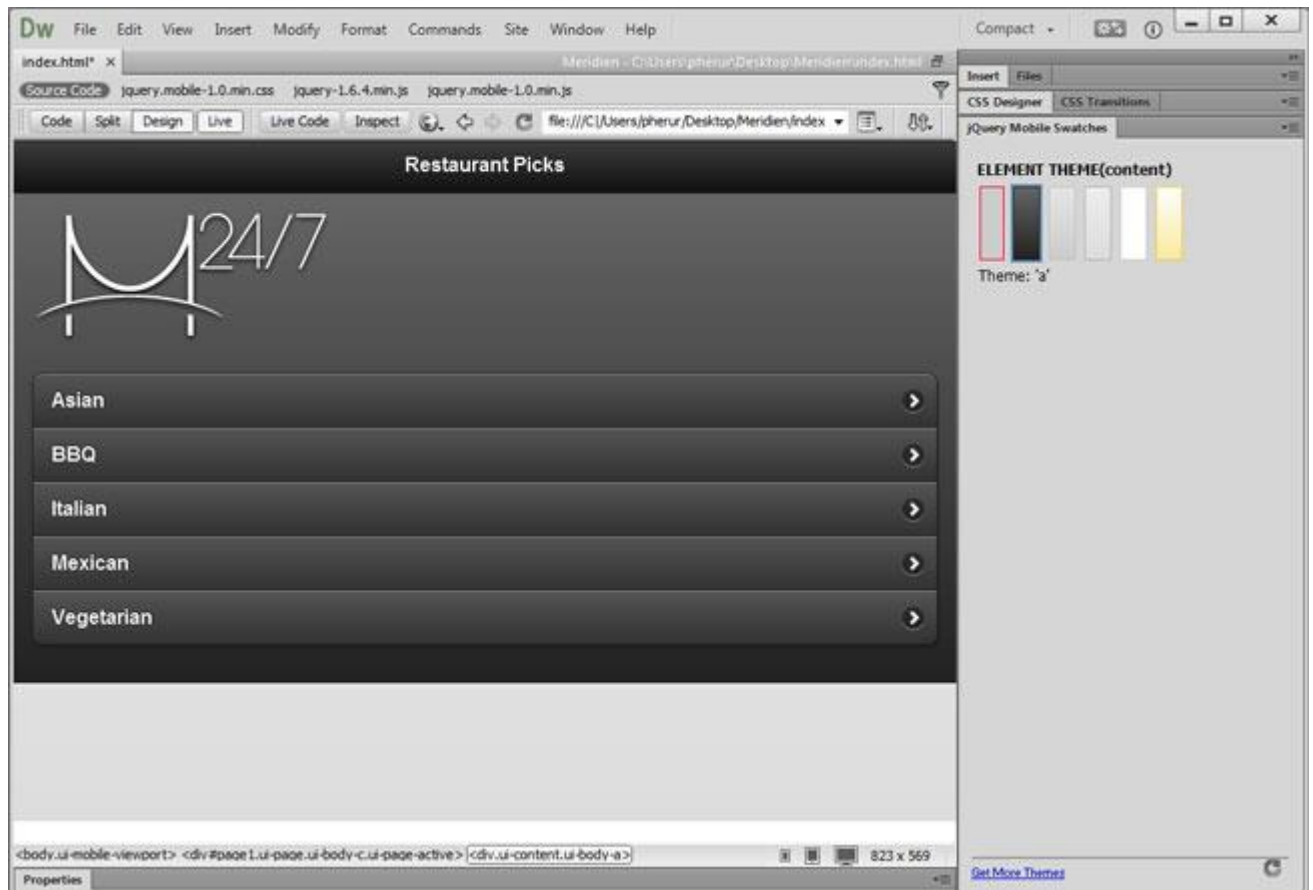


Figure 17. Selecting a theme.

3. You'll see that data-theme="a" is added in Code view. Selecting the list will populate the Swatches panel with the relevant theme, which can be modified. You can edit the theme and add icons from the swatches panel (see Figure 18).



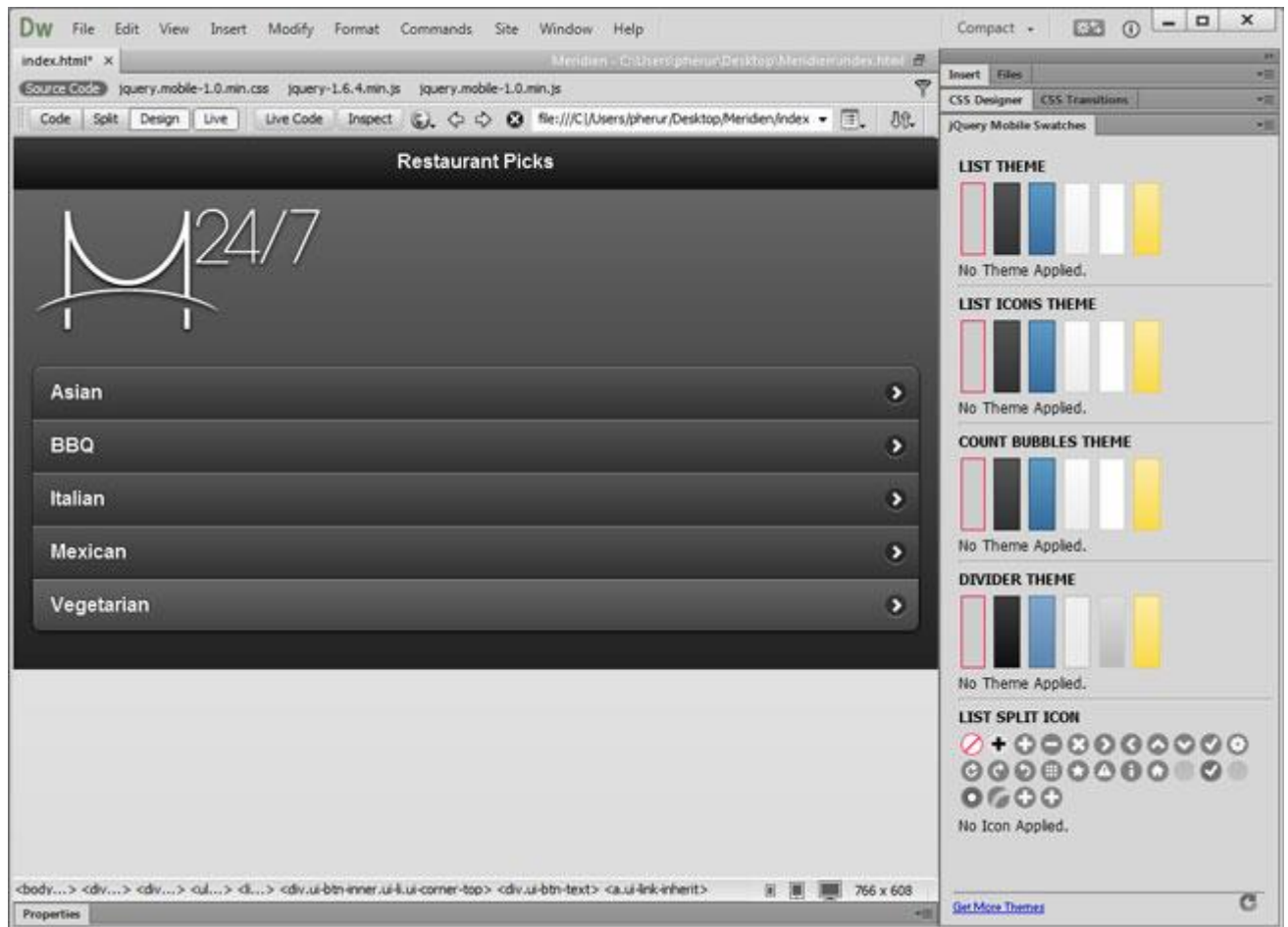


Figure 18. Icons in the jQuery Mobile Swatches panel.

4. To create additional themes, click the Get More Themes link at the bottom of the panel. Use the options in ThemeRoller to create additional themes (see Figure 19).

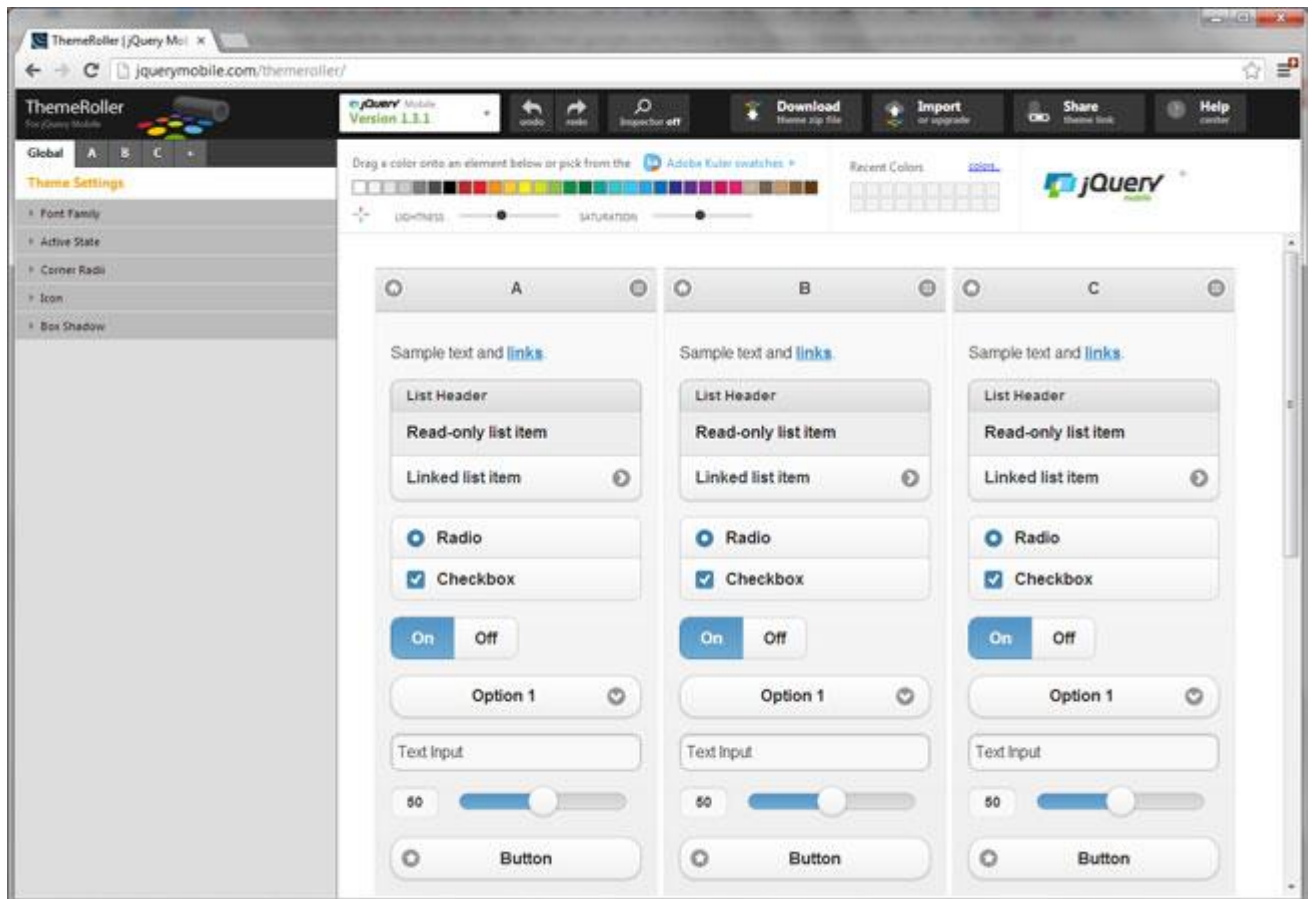


Figure 19. Creating a new theme with ThemeRoller.

5. After creating the theme, download it, and link the CSS file in Dreamweaver CC. This will populate the Swatches panel with the newly created theme and you can use it at relevant locations.

## Linking the Map It button to Google maps

Next you'll use PhoneGap APIs to locate the restaurant on a map when the user clicks the Map It button. If the location of the user cannot be determined, a fake map with the message, "Your location could not be determined" is displayed.

1. In Code view, replace the content between the <head></head> tags with the following code:

```
<meta name="viewport" content="width=deviceWidth; initial-
scale=1.0; maximum-scale=1.0; minimum-scale=1.0; user-
scalable=0;" />
<title>Meridien 24/7</title>
<link href="css/styles.css" rel="stylesheet" type="text/css">
<link href="http://code.jquery.com/mobile/1.0/jquery.mobile-
1.0.min.css" rel="stylesheet" type="text/css">
<script src="http://code.jquery.com/jquery-1.6.4.min.js"
type="text/javascript"></script>
<script src="http://code.jquery.com/mobile/1.0/jquery.mobile-
1.0.min.js" type="text/javascript"></script>
<script src="js/phonegap.js"></script>
```

```

<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false"></script>
<script>
$(function () {
var map = null;
var destMarker = null;
var lat = 37.771829;
var lng = -122.401681;
var latlng;
var cPosition =
navigator.geolocation.getCurrentPosition(geoSuccess, geoFail,
{timeout:3000, enableHighAccuracy:true});

function geoSuccess (pos) {
lat = pos.coords.latitude;
lng = pos.coords.longitude;
latlng = new google.maps.LatLng(lat,lng);
}

function geoFail () {
latlng = new google.maps.LatLng(lat,lng); // Default location is
San Francisco
$('.errorMsg').css('display', 'block');
$('#gMap').css('background-image', 'url(img/fakeMap.jpg)');
}

function makeMap () {
if (map == null) {
latlng = new google.maps.LatLng(lat,lng);
var options = { zoom: 16, center: latlng, disableDefaultUI: true,
mapTypeId: google.maps.MapTypeId.ROADMAP };
map = new google.maps.Map(document.getElementById('gMap'),
options);
uRHere = new google.maps.Marker({ position: latlng, map: map,
title: 'You are here' });
}
}

$('.mapIt').click(function () {
makeMap();
var dest = null;
var category = $(this).closest('div[data-
role="page"][id]').attr('id');
var restName = $(this).parent('div').siblings('h2').text();
switch (category) {
case 'asian':
dest = new google.maps.LatLng(lat + 0.002, lng + 0.002);
break;
case 'bbq':

```

```

dest = new google.maps.LatLng(lat + 0.001, lng - 0.002);
break;
case 'italian':
dest = new google.maps.LatLng(lat + 0.003, lng + 0.0015);
break;
case 'mexican':
dest = new google.maps.LatLng(lat - 0.002, lng - 0.002);
break;
case 'veggie':
dest = new google.maps.LatLng(lat - 0.001, lng - 0.003);
break;
}
if (destMarker == null) {
destMarker = new google.maps.Marker({ position: dest, map: map,
icon: 'img/fork.png', title: restName });
} else {
destMarker.setPosition(dest);
destMarker.setTitle(restName);
}
$('#map h1').text(restName);
});

$('body > *').css({minHeight: window.innerHeight +
'px !important'});

document.addEventListener('deviceready', doDeviceReady, false);

function doDeviceReady () {
// The following tells the app to fade #page1 in after the splash
screen
$('#page1').fadeIn(5000);
}
});
</script>

```

2. In each child page that you created corresponding to various categories, replace the content within the <div></div> tags for the Map It and Call buttons with the following code:

```

<div data-role="controlgroup" data-type="horizontal" ><a
href="#map" data-role="button" data-inline="false"
class="mapIt">Map It</a><a href="tel://4907774116" data-
role="button" data-inline="false">Call</a></div>

```

3. Add this code at the end of all the pages, just above the </body> tag in the code view.

```

<div data-role="page" id="map" data-theme="a">
<div data-role="header" data-position="fixed">
<h1>Page Header</h1>
</div>
<div data-role="content">
<p class="errorMsg">Your location could not be determined.</p>

```

```

<div id="gMaps"></div>
</div>
<div data-role="footer" data-position="fixed">
<h4>Meridien 24/7</h4>
</div>
</div>
<div data-role="page" id="loader" data-theme="a">
<div data-role="content">
<div id="loading"></div>

```

4. Press F12 to preview the pages in Google Chrome. After navigating to a child page, click Map It to display the location of the restaurant.

## Using the PhoneGap Build service through Dreamweaver CC

Now that your application is done, you can use PhoneGap Build to create packages for various mobile platforms.

1. Choose Site > PhoneGap Build Service > PhoneGap Build Service in Dreamweaver CC.
2. Login using your Adobe ID credentials.
3. After logging in, click Continue to create a new project (see Figure 20).



Figure 20. Creating a new PhoneGap Build project.

4. Choose the platforms for which you want to build. If you have authentication details for a particular platform, provide them here. To start the build process for all platforms click Continue (see Figure 21).

**Note:** It is not mandatory to provide any authentication keys here.

A screenshot of the PhoneGap Build Service window. The window has a title bar with 'PhoneGap Build Service' and standard window controls. Below the title bar, there is a text prompt: 'Enter key and password info corresponding to the mobile platform'. The window is divided into three sections for different mobile platforms: iOS, Android, and BlackBerry. Each section contains a logo, a 'Key' dropdown menu, and one or more password input fields. The 'Continue' button is at the bottom.

PhoneGap Build Service

Enter key and password info corresponding to the mobile platform

**iOS**

Key

Password

**Android**

Key

Password

KSPassword

**BlackBerry**

Key

Password

Figure 21. Preparing to build packages.

5. PhoneGap Build will create the selected packages created for you. After you download a package (see Figure 22) to the appropriate mobile device, it is ready to use.





Figure 22. Downloading the application.

## Where to go from here

For more information on PhoneGap Build and Dreamweaver CC, see [Packaging web applications as native mobile applications with PhoneGap Build](#) or watch the [Packaging an App with PhoneGap Build](#) video. If you don't already have Dreamweaver CC, [download it now](#).