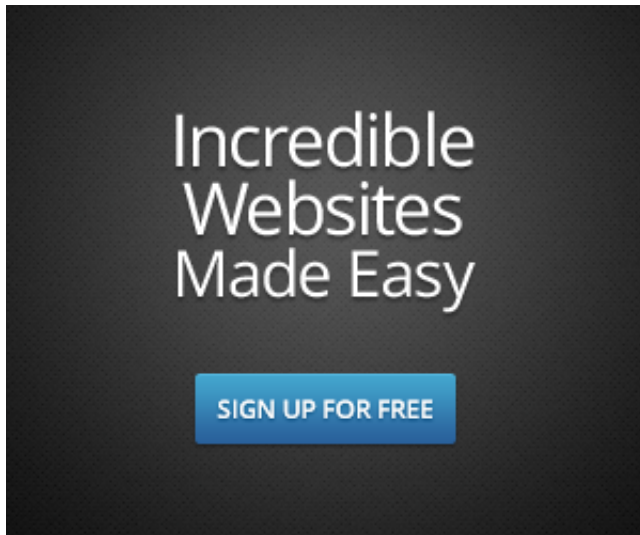




# jQuery Mobile Tutorial: Creating a Restaurant Picker Web App

## Advertisement



With an increase in the number, diversity and complexity of smartphones, more and more companies want to have their own **mobile app**, but creating a native app can be pretty expensive. It requires special skills, as well as special coding tools, and then there is also the need to build an app per platform (Android, iOS, BlackBerry, Windows Phone, etc).

All of this figures in to a higher price tag for the app development. Another solution for developers is then to create something called **web-apps**: HTML CSS apps, **which users can access from their browsers**. They are cross-platform, and cross device.



The jQuery framework has been around the web for a while now, but the jQuery base technology was basically designed for browser apps. **jQuery Mobile** is a framework based on jQuery that enables web designers to create web-apps that are optimized for use on a mobile device (Smartphone and tablets). The framework is touch optimized, uses responsive layout so that elements will automatically adapt on different device sizes, and supports a range of different platforms and devices.

In this jQuery Mobile tutorial, we will create a nice demo app from scratch, to show some of the things that can be easily done using this powerful tool.

Before we start, you might want to [download the files](#), or [see a demo](#), or flash the qrcode for a direct demo on your mobile :



[**Note:** Have you already ordered your copy of our [Printed Smashing Book #3](#)? The book is a professional guide on how to redesign websites and it also introduces a whole new mindset for progressive Web design, written by experts for you.]

## The Concept of the Mini App: Restaurant Picker

We will create an application that will enable the user to choose a restaurant based on what they want to eat tonight, the town where they want to eat and other user's ratings of the restaurants. The jQuery Mobile mini app we're creating will be called "Restaurant Picker".

Please note that this is only the front development, you will of course need a server and a database if you want to create a real app. Also note that jQuery Mobile uses Ajax navigation, so you'll need to put the files either on a local server (xampp, mamp, etc) or on a real server to make the demo work properly.

## Wireframing Our Application.

In order to see where we are going, we will first draw some wireframes to see what each screen of the app will look like.

### Home Screen : Choose a Plate



On the first screen we will put the logo of our application. We will then give the user a choice between different plates. Each choice comes with an image illustrating the plate, and each is a link that will lead to the second page where the user can choose the town.

### Choose a Town

< back

Branding

Choose a town where you want to eat

Q

Filter

Name of town

2

>

Name of town

2

>

Name of town

2

>

Name of town

2

>

Name of town

2

>

Name of town

2

>

Name of town

2

>

Name of town

2

>

On the second screen, the user can choose the town where they want to eat. The towns are displayed in a list of clickable items. Beside each town there is a bubble that gives the user the number of restaurants available for the chosen plate in this town.

Since the list of towns can be pretty long, we will provide a filter so that the user can quickly search for a town at the end of the list. The title bar will use the branding of the application, and a “back” button that user can click to go back to the previous step.

When the user clicks on a town, they are lead to the page where they can choose a restaurant.

### Choose a Restaurant

< back

Branding

Choose a restaurant

image

Name of restaurant  
★

>

image

Name of restaurant  
★ ★

>

image

Name of restaurant  
★ ★ ★

>

image

Name of restaurant  
★ ★ ★ ★

>

image

Name of restaurant  
★

>

The user can now choose in which restaurant he wants to eat. The application displays a list of restaurant with a little image, the name, and the number of rating stars the previous users gave to this restaurant.

The title bar will use the branding of the application, and a “back” button that user can click to go back to previous step. The user can then click on a specific restaurant, to see the details.

## Restaurant Details



The restaurant's details view is composed of three parts: restaurant details, contact details, and the establishment's user rating.

The restaurant details will display a short description of the restaurant, some plates, a picture and a link to their website if they have one.

The contact details will display the address of the restaurant, and an image of a Google map that will locate the restaurant in the town. A link enables the user to open Google maps (either using the browser or the Google map app if available depending on the device) to locate the restaurant on the map. Another link will dial the restaurant phone number directly on the mobile device so that user can place a reservation.

The last part of this view is a select box, where the user can rate the restaurant.

## Building the web-application base

Now that we've see the direction that we are heading, we can begin digging a little bit into the code.

### Some jQuery Mobile Basics

First let's take a look at what the header of our first HTML page will look like :

```

view plain copy to clipboard print ?
01. <!DOCTYPE HTML>
02.
03. <HTML>
04.
05. <head>
06.
07. <meta charset="UTF-8">
08.
09. <title>Restaurant Picker</title>
10.
11. <meta name="viewport" content="width=device-width, initial-scale=1">

```

```

12.
13. <link rel="stylesheet" href="jquery.mobile.structure-1.0.1.CSS" />
14.
15. <link rel="stylesheet" href="jquery.mobile-1.0.1.CSS" />
16.
17. <script src="js/jquery-1.7.1.min.js"></script>
18.
19. <script src="js/jquery.mobile-1.0.1.min.js"></script>
20.
21. </head>

```

[view plain](#) [copy to clipboard](#) [print](#) [?](#)

```

01. <!DOCTYPE HTML>
02.
03. <HTML>
04.
05. <head>
06.
07. <meta charset="UTF-8">
08.
09. <title>Restaurant Picker</title>
10.
11. <meta name="viewport" content="width=device-width, initial-scale=1">
12.
13. <link rel="stylesheet" href="jquery.mobile.structure-1.0.1.CSS" />
14.
15. <link rel="stylesheet" href="jquery.mobile-1.0.1.CSS" />
16.
17. <script src="js/jquery-1.7.1.min.js"></script>
18.
19. <script src="js/jquery.mobile-1.0.1.min.js"></script>
20.
21. </head>

```

The first thing to notice is the “new” HTML5 doctype. jQuery Mobile makes a heavy use of the data- HTML5 attribute to add some elements in the DOM to style the page, so we will need an HTML5 doctype to make it all work nicely.

The second thing to notice is the meta name=viewport tag. We will use this meta tag to gain better control of our viewport. Without this tag, the browser will squeeze our layout in the whole page, and it will look ugly and very tiny. With width=device-width we will use our device width, making the app fit the whole size of the device without being squeezed. The initial-scale property controls the zoom level when the page is first loaded and we will set it to 1, meaning no zoom in or out when page is loaded.

Then we will call our CSS files. In older jQuery Mobile versions the CSS was in one file, but in the newer version they made a distinction between the structure and the actual design (colors, gradients etc) which makes it easier to create custom styles.

We then need to load our jQuery, and jQuery Mobile JavaScript code at the end, since it needs the jQuery library to work.

## What You’d Like to Eat: First Page

Now let’s take a look at the HTML code of our first page, in this exercise we will call this page index.HTML

[view plain](#) [copy to clipboard](#) [print](#) [?](#)

```

01. <body>
02.
03. <div data-role="page" id="home" data-theme="c">
04.
05. <div data-role="content">
06.
07. <div id="branding">

```

```

08.
09. <h1>Restaurant Picker </h1>
10.
11. </div>
12.
13. <div class="choice_list">
14.
15. <h1> What would you'd like to eat? </h1>
16.
17. <ul data-role="listview" data-inset="true" >
18.
19. <li><a href="choose_town.HTML" data-
transition="slidedown">  <h3> Some Sushis</h3></a></li>
20.
21. <li><a href="choose_town.HTML" data-
transition="slidedown">  <h3> A Pizza </h3></a></li>
22.
23. <li><a href="choose_town.HTML" data-
transition="slidedown">  <h3> A Kebab</h3></a></li>
24.
25. <li><a href="choose_town.HTML" data-
transition="slidedown">  <h3> A Burger</h3></a></li>
26.
27. <li><a href="choose_town.HTML" data-
transition="slidedown">  <h3> Some Nems </h3></a></li>
28.
29. <li><a href="choose_town.HTML" data-
transition="slidedown">  <h3> Something more traditional</h3></a></li>
30.
31. </ul>
32.
33. </div>
34.
35. </div>
36.
37. </div>
38.
39. </body>
40.
41. </HTML>

```

view plain copy to clipboard print ?

```

01. <body>
02.
03. <div data-role="page" id="home" data-theme="c">
04.
05. <div data-role="content">
06.
07. <div id="branding">
08.
09. <h1>Restaurant Picker </h1>
10.
11. </div>
12.
13. <div class="choice_list">
14.
15. <h1> What would you'd like to eat? </h1>
16.
17. <ul data-role="listview" data-inset="true" >
18.
19. <li><a href="choose_town.HTML" data-
transition="slidedown">  <h3> Some Sushis</h3></a></li>
20.
21. <li><a href="choose_town.HTML" data-
transition="slidedown">  <h3> A Pizza </h3></a></li>
22.
23. <li><a href="choose_town.HTML" data-

```



```

24. transition="slidedown">  <h3> A Kebap</h3></a></li>
25. <li><a href="choose_town.HTML" data-
26. transition="slidedown">  <h3> A Burger</h3></a></li>
27. <li><a href="choose_town.HTML" data-
28. transition="slidedown">  <h3> Some Nems </h3></a></li>
29. <li><a href="choose_town.HTML" data-
30. transition="slidedown">  <h3> Something more traditional</h3></a></li>
31. </ul>
32.
33. </div>
34.
35. </div>
36.
37. </div>
38.
39. </body>
40.
41. </HTML>

```

jQuery Mobile makes the distinction between HTML documents and pages. A jQuery Mobile app can be composed of one HTML document with multiple pages in it, using the data-role="page" attribute each page is linked using anchors; or of many documents, each one having its own data-role="page" linked using "normal" links. In our case, we will go for creating one HTML document per page.

So first we will open our body, and create our page using <div data-role="page" id="home" data-theme="c">

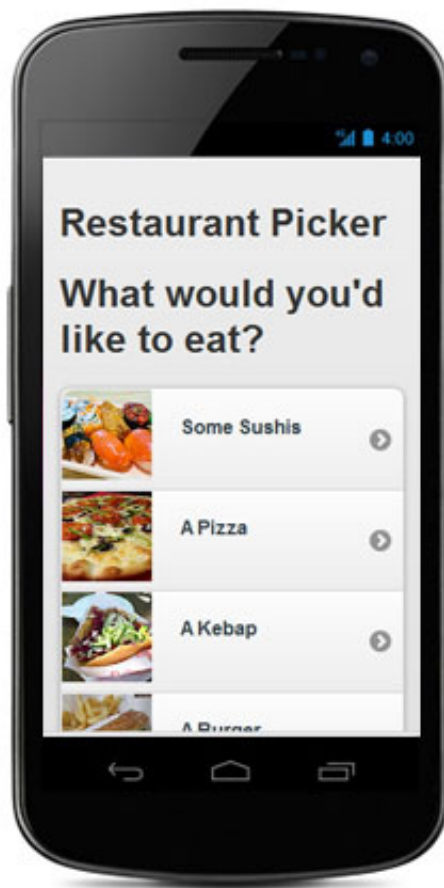
We will then open a content division, in which we put the branding of the app, and our first title followed by the list of different plates.

To create a jQuery Mobile list, we will put the data-role="listview" attribute on our <ul> element. data-inset="true" is here to style the list as an inset list (with rounded corners and padding around it).

Each list element <li> that contains an <a href> link will be automatically converted in a link list item by jQuery Mobile. To add the image, we simply add an image inside our <a href> link, and jQuery Mobile will do the work for us: it will display it on the left part of the list. Pretty easy.

The data-transition="slidedown" creates the transition between two pages. You can find more transitions in the official documentation.

Here is what our first page looks like:



## In Which Town Do You Want To Eat: Second Page

We will name the second page `choose_town.HTML` . Here is the HTML code, explanation of the tricky parts follows. Note that the header won't change.

[view plain](#) [copy to clipboard](#) [print](#) [?](#)

```

01. <body>
02.
03. <div id="choisir_ville" data-role="page" data-add-back-btn="true">
04.
05. <div data-role="header">
06.
07. <h1> Restaurant Picker</h1>
08.
09. </div>
10.
11. <div data-role="content">
12.
13. <div class="choice_list">
14.
15. <h1> In which town do you want to eat? </h1>
16.
17. <ul data-role="listview" data-inset="true" data-filter="true" >
18.
19. <li><a href="choose_restaurant.HTML" data-
20. transition="slidedown"> Amiens <span > 3 </span></a> </li>
21.
22. <li><a href="choose_restaurant.HTML" data-
23. transition="slidedown"> Bastia <span > 2 </span></a> </li>
24.
25. <li><a href="choose_restaurant.HTML" data-transition="slidedown"> Belfort <span class="ui-li-
    count" > 5 </span></a> </li>
26.
27. <li><a href="choose_restaurant.HTML" data-transition="slidedown"> Bordeaux <span class="ui-
    li-count" > 1</span></a> </li>

```

```

26.
27. ...
28.
29. </ul>
30.
31. </div>
32.
33. </div>
34.
35. </div>
36.
37. </body>

```

view plain copy to clipboard print ?

```

01. <body>
02.
03. <div id="choisir_ville" data-role="page" data-add-back-btn="true">
04.
05. <div data-role="header">
06.
07. <h1> Restaurant Picker</h1>
08.
09. </div>
10.
11. <div data-role="content">
12.
13. <div class="choice_list">
14.
15. <h1> In which town do you want to eat? </h1>
16.
17. <ul data-role="listview" data-inset="true" data-filter="true" >
18.
19. <li><a href="choose_restaurant.HTML" data-
transition="slidedown"> Amiens <span > 3 </span></a> </li>
20.
21. <li><a href="choose_restaurant.HTML" data-
transition="slidedown"> Bastia <span > 2 </span></a> </li>
22.
23. <li><a href="choose_restaurant.HTML" data-transition="slidedown"> Belfort <span class="ui-li-
count" > 5 </span></a> </li>
24.
25. <li><a href="choose_restaurant.HTML" data-transition="slidedown"> Bordeaux <span class="ui-
li-count" > 1</span></a> </li>
26.
27. ...
28.
29. </ul>
30.
31. </div>
32.
33. </div>
34.
35. </div>
36.
37. </body>

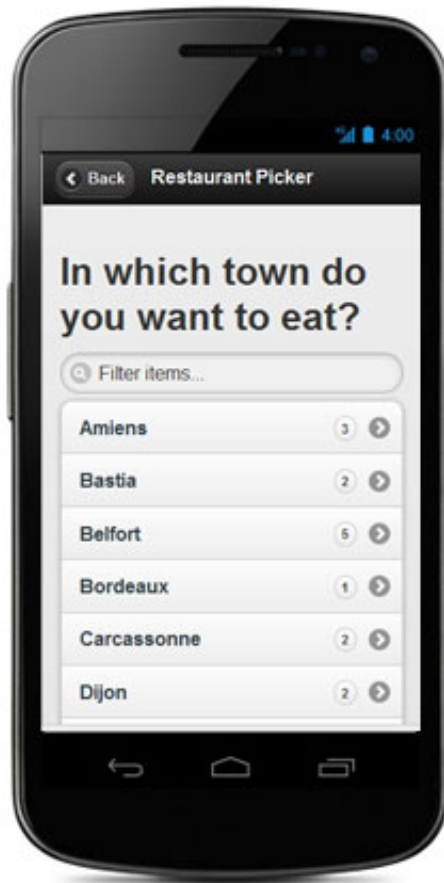
```

We changed the id, so that jQuery Mobile can understand that this is another page. Notice that we used the data-add-back-btn="true" on the page div, this will enable the Ajax back navigation and automatically add a back button to our title bar.

To create our title bar, we will create a div element, with the data-role="header".

To add a filter to our list, we will simply put data-filter="true" on the ul element defining the list. Note that this is a filter that will filter the items of the list, and **is not a search bar**.

The last trick will be creating the little bubbles on the right of list elements, and we will do that by creating a span with the class and the numbers in it. And here is how the second page will look:



## Choose a Restaurant: Third Page

We will name this page choose\_restaurant.HTML and below is what the HTML code will look like.

view plain copy to clipboard print ?

```

01. <body>
02.
03. <div id="choisir_restau" data-role="page" data-add-back-btn="true">
04.
05. <div data-role="header">
06.
07. <h1> Restaurant Picker</h1>
08.
09. </div>
10.
11. <div data-role="content">
12.
13. <div class="choice_list">
14.
15. <h1> Please choose a restaurant.</h1>
16.
17. <ul data-role="listview" data-inset="true" >
18.
19. <li><a href="restaurant.HTML" data-
  transition="slidedown">  <h2> Le Mouffetard</h2> <p> 4 stars
20.
21. <li><a href="restaurant.HTML" data-
  transition="slidedown">  <h2> Chocolate bar </h2> <p> 4 stars
22.
23. <li><a href="restaurant.HTML" data-
  transition="slidedown">  <h2> Restaurant Diona</h2> <p> 1 st
24.
25. <li><a href="restaurant.HTML" data-
  transition="slidedown">  <h2> Tai Shan</h2> <p> 3 stars </p>
26.

```

```

27. <li><a href="restaurant.HTML" data-
    transition="slidedown">  <h2> Arcade</h2> <p> 2 stars </p> <
28.
29. </ul>
30.
31. </div>
32.
33. </div>
34.
35. </div>
36.
37. </body>

```

view plain copy to clipboard print ?

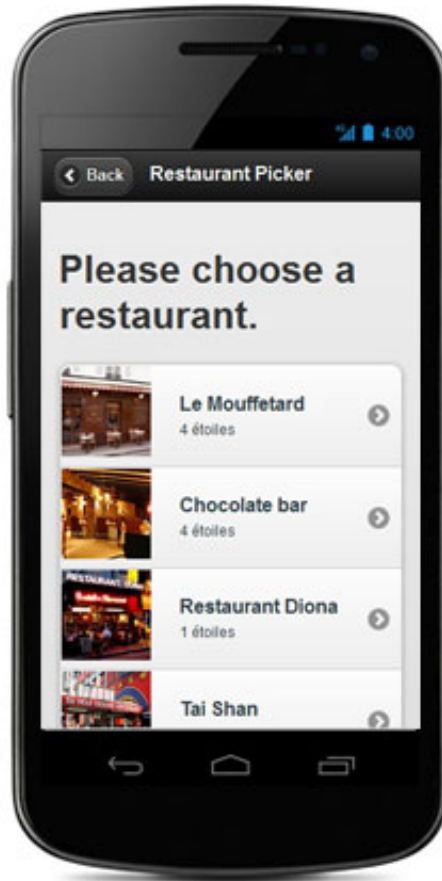
```

01. <body>
02.
03. <div id="choisir_restau" data-role="page" data-add-back-btn="true">
04.
05. <div data-role="header">
06.
07. <h1> Restaurant Picker</h1>
08.
09. </div>
10.
11. <div data-role="content">
12.
13. <div class="choice_list">
14.
15. <h1> Please choose a restaurant.</h1>
16.
17. <ul data-role="listview" data-inset="true" >
18.
19. <li><a href="restaurant.HTML" data-
    transition="slidedown">  <h2> Le Mouffetard</h2> <p> 4 stars </p> <
20.
21. <li><a href="restaurant.HTML" data-
    transition="slidedown">  <h2> Chocolate bar </h2> <p> 4 stars
22.
23. <li><a href="restaurant.HTML" data-
    transition="slidedown">  <h2> Restaurant Diona</h2> <p> 1 st
24.
25. <li><a href="restaurant.HTML" data-
    transition="slidedown">  <h2> Tai Shan</h2> <p> 3 stars </p>
26.
27. <li><a href="restaurant.HTML" data-
    transition="slidedown">  <h2> Arcade</h2> <p> 2 stars </p> <
28.
29. </ul>
30.
31. </div>
32.
33. </div>
34.
35. </div>
36.
37. </body>

```

This page is pretty similar to the first one, except for the title bar and the notation of the customer. We already saw how to create a title bar in previous section. For the customer rating, we add a p element with two classes: .classement is mutual to all the elements and will enable us to style this element with little stars, and then we use the class .one, .two, .three and .four to make the distinction between how many stars the customers gave to the restaurant. We will later in the article see how to style this in a nice way, but for the moment, we'll leave it plain text.

Here is our third page:



## Restaurant Details: Fourth Page

This page called restaurant.HTML is the trickiest of all, mainly because it has lots of elements on it. We will split the code in three parts: the restaurant description, the contact details, and the user rating. Here is our full HTML code.

[view plain](#) [copy to clipboard](#) [print](#) ?

```
01. <body>
02. <div id="restau" data-role="page" data-add-back-btn="true">
03.
04.   <div data-role="header">
05.     <h1> Restaurant Picker</h1>
06.   </div>
07.
08.   <div data-role="content">
09.     <div class="ui-grid-a" id="restau_infos">
10.       <div class="ui-block-a">
11.         <h1> Le Mouffetard</h1>
12.         <p><strong> Restaurant bar in the center of Strasbourg</strong></p>
13.         <p> On the menu: </p>
14.         <ul>
15.           <li> Milkshake with chocolat</li>
16.           <li> Planchettes </li>
17.           <li> Leek pie </li>
18.         </ul>
19.       </div>
20.       <div class="ui-block-b">
21.         <p></p>
22.         <p><a href="http://www.google.fr" rel="external" data-
role="button"> See our website </a></p>
23.       </div>
24.     </div>
25.   </div>
  </div>
```

```

26.
27. <div class="ui-grid-a" id="contact_infos">
28.   <div class="ui-block-a">
29.     <h2> Contact us</h2>
30.     <p>30 Rue des Tonneliers</p>
31.     <p> 67000 Strasbourg </p>
32.   </div>
33.   <div class="ui-block-b">
34.     
35.   </div>
36. </div>
37. <div id="contact_buttons">
38.   <a href="http://maps.google.fr/maps?
q=jeannette+et+les+cycleux&hl=fr&sll=46.75984,1.738281&sspn=10.221882,18.764648&hq=je
role="button" data-icon="maps"> Find us on Google Maps </a>
39.   <a href="tel:0388161072" data-role="button" data-icon="tel"> Call us </a>
40. </div>
41. <hr/>
42.
43. <div id="notation">
44.   <form>
45.     <label for="select-choice-0" class="select"><h2> User rating </h2></label>
46.     <select name="note_utilisateur" id="note_utilisateur" data-native-menu="false" data-
theme="c" >
47.       <option value="one" class="one"> Not good at all </option>
48.       <option value="two" class="two">Average </option>
49.       <option value="three" class="three">Pretty good </option>
50.       <option value="four" class="four"> Excellent </option>
51.     </select>
52.   </form>
53. </div>
54. </div>
55.
56. </div>
57. </body>

```

view plain copy to clipboard print ?

```

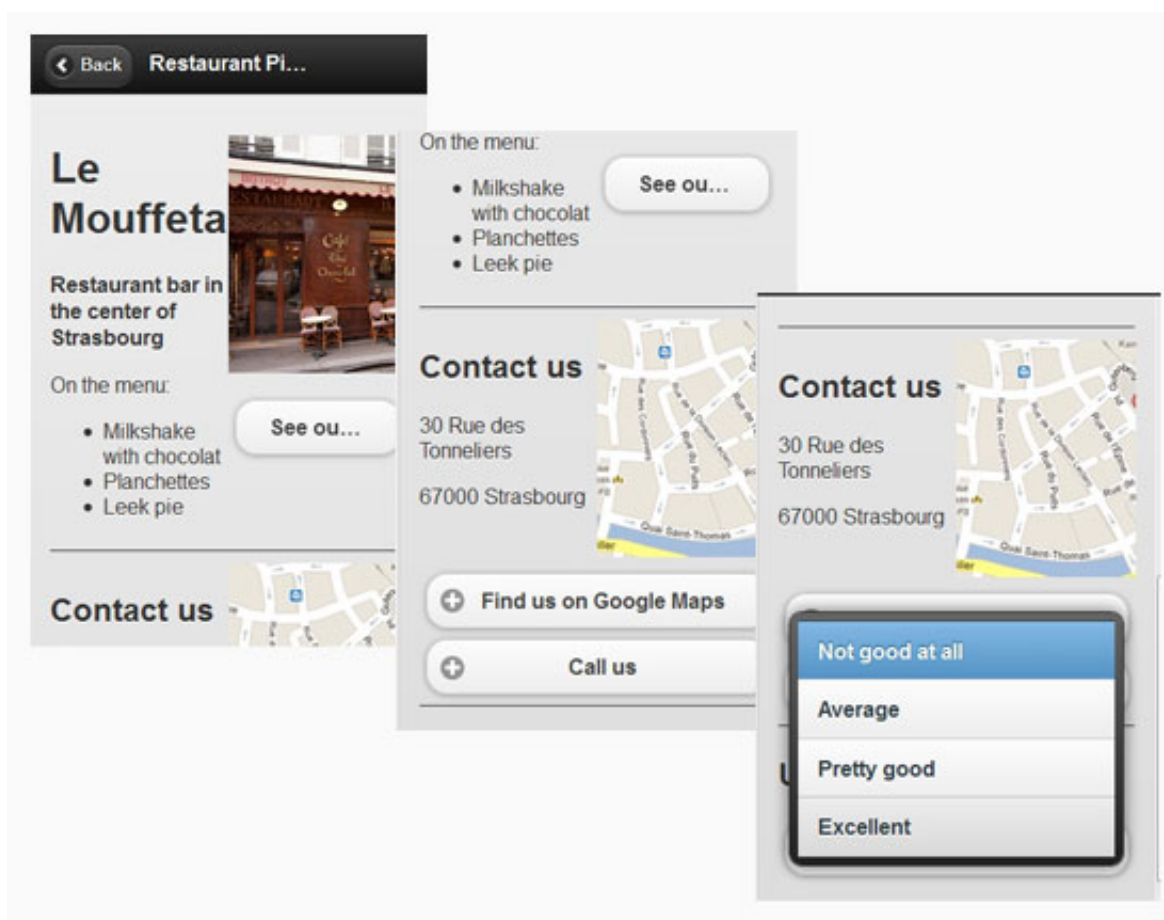
01. <body>
02. <div id="restau" data-role="page" data-add-back-btn="true">
03.
04.   <div data-role="header">
05.     <h1> Restaurant Picker</h1>
06.   </div>
07.
08.   <div data-role="content">
09.     <div class="ui-grid-a" id="restau_infos">
10.       <div class="ui-block-a">
11.         <h1> Le Mouffetard</h1>
12.         <p><strong> Restaurant bar in the center of Strasbourg</strong></p>
13.         <p> On the menu: </p>
14.         <ul>
15.           <li> Milkshake with chocolat</li>
16.           <li> Planchettes </li>
17.           <li> Leek pie </li>
18.         </ul>
19.       </div>
20.       <div class="ui-block-b">
21.         <p></p>
22.         <p><a href="http://www.google.fr" rel="external" data-
role="button"> See our website </a></p>
23.       </div>
24.     </div>
25.   <hr/>
26.
27.   <div class="ui-grid-a" id="contact_infos">
28.     <div class="ui-block-a">

```

```

29. <h2> Contact us</h2>
30. <p>30 Rue des Tonneliers</p>
31. <p>67000 Strasbourg </p>
32. </div>
33. <div class="ui-block-b">
34. 
35. </div>
36. </div>
37. <div id="contact_buttons">
38. <a href="http://maps.google.fr/maps?
q=jeannette+et+les+cycleux&hl=fr&sll=46.75984,1.738281&sspn=10.221882,18.764648&hq=je
role="button" data-icon="maps"> Find us on Google Maps </a>
39. <a href="tel:0388161072" data-role="button" data-icon="tel"> Call us </a>
40. </div>
41. <hr/>
42.
43. <div id="notation">
44. <form>
45. <label for="select-choice-0" class="select"><h2> User rating </h2></label>
46. <select name="note_utilisateur" id="note_utilisateur" data-native-menu="false" data-
theme="c" >
47. <option value="one" class="one"> Not good at all </option>
48. <option value="two" class="two">Average </option>
49. <option value="three" class="three">Pretty good </option>
50. <option value="four" class="four"> Excellent </option>
51. </select>
52. </form>
53. </div>
54. </div>
55.
56. </div>
57. </body>

```



For the restaurant details, we used the build in two column elements of jQuery Mobile. To create a two column block, we create a block with two child blocks. To create a nice button for the website, we added the data-role="button" to the a href element, and a rel="external" so that jQuery Mobile knows that Ajax



should not used to open this link, but that this is an external link.

For the contact details, we once again used the two column trick. What's to emphasize here are the buttons. The data-icon="maps" and data-icon="tel" will enable us to add custom icons to those buttons.

The interesting part about adding a GoogleMap link, is that some mobiles will be able to detect it, and will ask the user if they want to open them using the browser; if the Google maps app is installed on the device. The other interesting part is the href=tel:0388161072 protocol. This won't work on a classic browser, but Smartphones will be able to open this link in the phone dial box, to directly call the number.

The last part is the user rating. For this we will use a simple select menu. The data-native-menu="false" enables us to style the select using jQuery Mobile, so that we will be able to add some nice stars here too. Here again you will notice that we added the one, two, three, four class for further styling.

And here we have a fully functional jQuery Mobile webapp. But this app looks very "jQueryMobile-like", so we now have to give it a little more styling to make it look nicer.

## Next up on Page Two: Adding Some Style

The tutorial is not quite finished yet! There is more to learn over on page two.



Incredible Websites  
Made Easy

SIGN UP FOR FREE



















