PhoneGap & jQuery Mobile

Lesson 4

Thomas Mak

makzan@42games.net

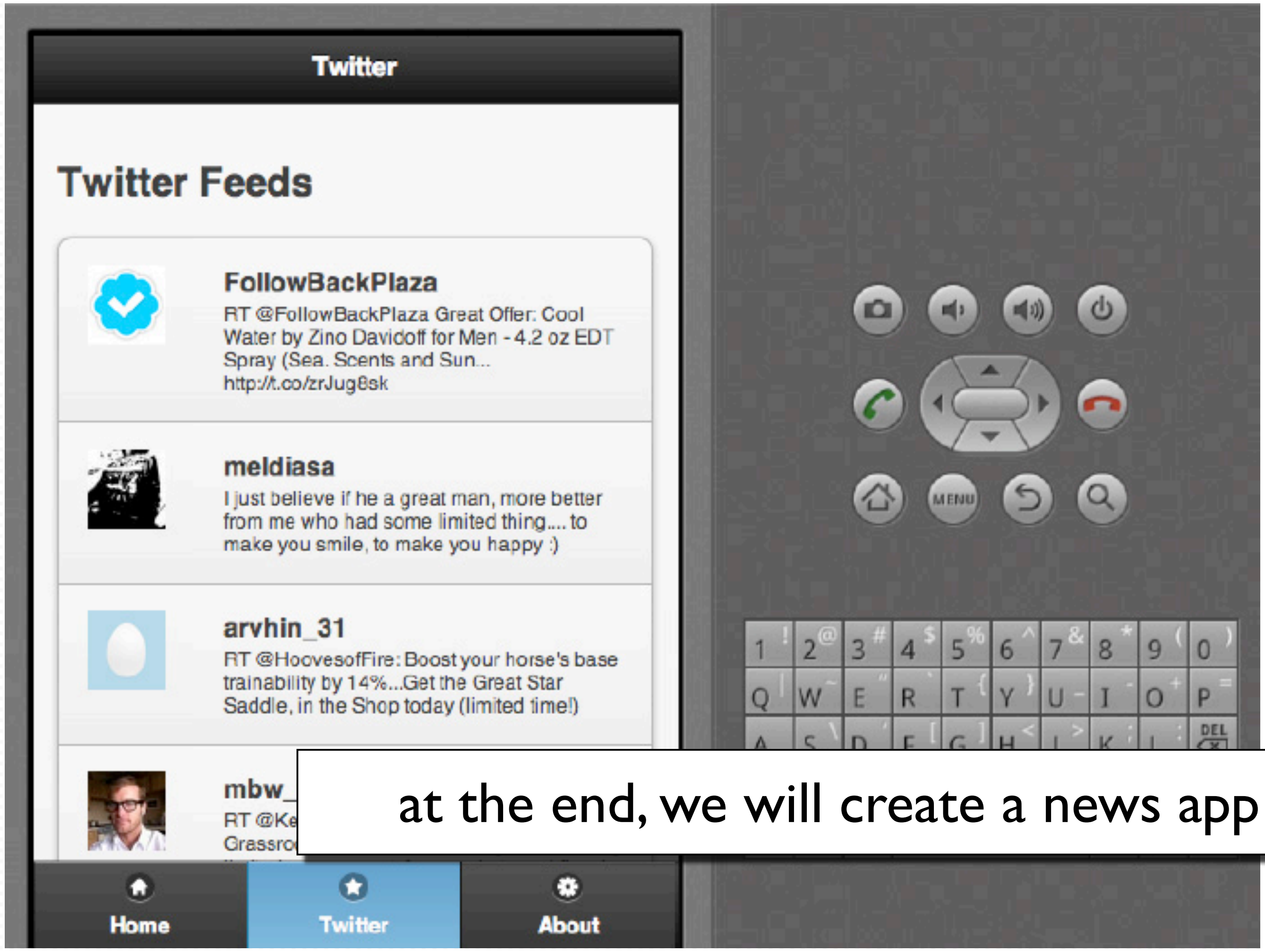
Source Codes

https://github.com/makzan/PhoneGap-Course-Examples

jQuery Mobile

- Nav Bar
- Get Twitter Search JSON Results
- Show Location in Map

HTML5 Cheat Sheet App



FollowBackPlaza
RT @FollowBackPlaza Great Offer: Cool Water by Zino Davidoff for Men - 4.2 oz EDT Spray (Sea. Scents and Sun... http://t.co/zrJug6sk

meldiasa
I just believe if he a great man, more better from me who had some limited thing.... to make you smile, to make you happy :)

arvhin_31
RT @HoovesofFire: Boost your horse's base trainability by 14%...Get the Great Star Saddle, in the Shop today (limited time!)

mbw_
RT @Ke
Grassro

at the end, we will create a news app for company.

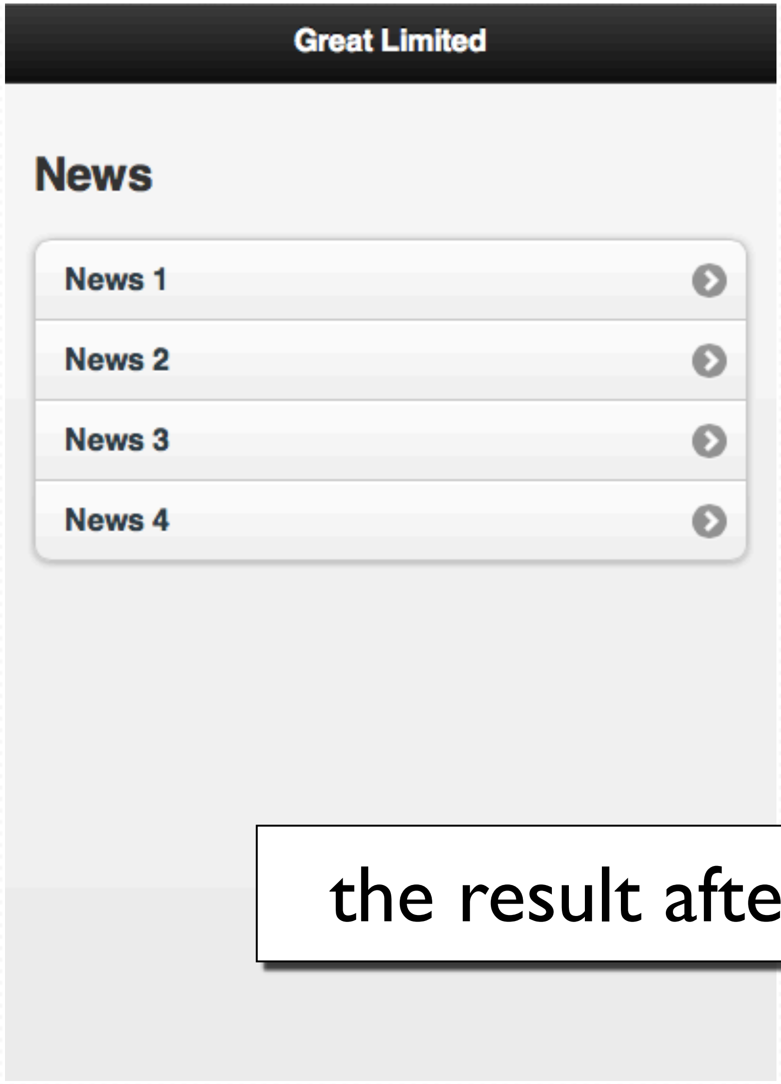
Preparing jQuery Mobile

- as before, we prepare the jQuery Mobile boilerplate.
  - all jQuery Mobile assets.
  - our empty app.js and app.css file

Main Page

In this section, we kick start the project with a main page.

Main Page


Great Limited

News

News 1

News 2

News 3

News 4

the result after this section.

Main Page

```html
<body>
    <!-- ////////// Home -->
    <article data-role='page' id='main'>
      <header data-position='fixed' data-role='header'>
        <h1>Great Limited</h1>
      </header>
      <section data-role='content'>
        <h2>News</h2>
        <ul data-inset='true' data-role='listview' id='news-list'>
          <li>
            <a href='#news-detail'>News 1</a>
          </li>
          <li>
            <a href='#news-detail'>News 2</a>
          </li>
          <li>
            <a href='#news-detail'>News 3</a>
          </li>
          <li>
            <a href='#news-detail'>News 4</a>
          </li>
        </ul>
      </section>
    </article>
  </body>
```

the `index.html` body.

Main Page

```
<body>
    <!-- ////////// Home -->
    <article data-role='page' id='main'>
      <header data-position='fixed' data-role='header'>
        <h1>Great Limited</h1>
      </header>
      <section data-role='content'>
        <h2>News</h2>
        <ul data-inset='true' data-role='listview' id='news-list'>
          <li>
            <a href='#news-detail'>News 1</a>
          </li>
          <li>
            <a href='#news-detail'>News 2</a>
          </li>
          <li>
            <a href='#news-detail'>News 3</a>
          </li>
          <li>
            <a href=
          </li>
        </ul>
      </section>
    </article>
  </body>
```

Note that we used article, header, section, footer to meet the HTML5 spec.

Main Page

```
<body>
    <!-- ////////// Home -->
    <article data-role='page' id='main'>
      <header data-position='fixed' data-role='header'>
        <h1>Great Limited</h1>
      </header>
      <section data-role='content'>
        <h2>News</h2>
        <ul data-inset='true' data-role='listview' id='news-list'>
          <li>
            <a href='#news-detail'>News 1</a>
          </li>
          <li>
            <a href='#news-detail'>News 2</a>
          </li>
          <li>
            <a href='#news-detail'>News 3</a>
          </li>
          <li>
            <a href
          </li>
        </ul>
      </section>
    </article>
  </body>
```

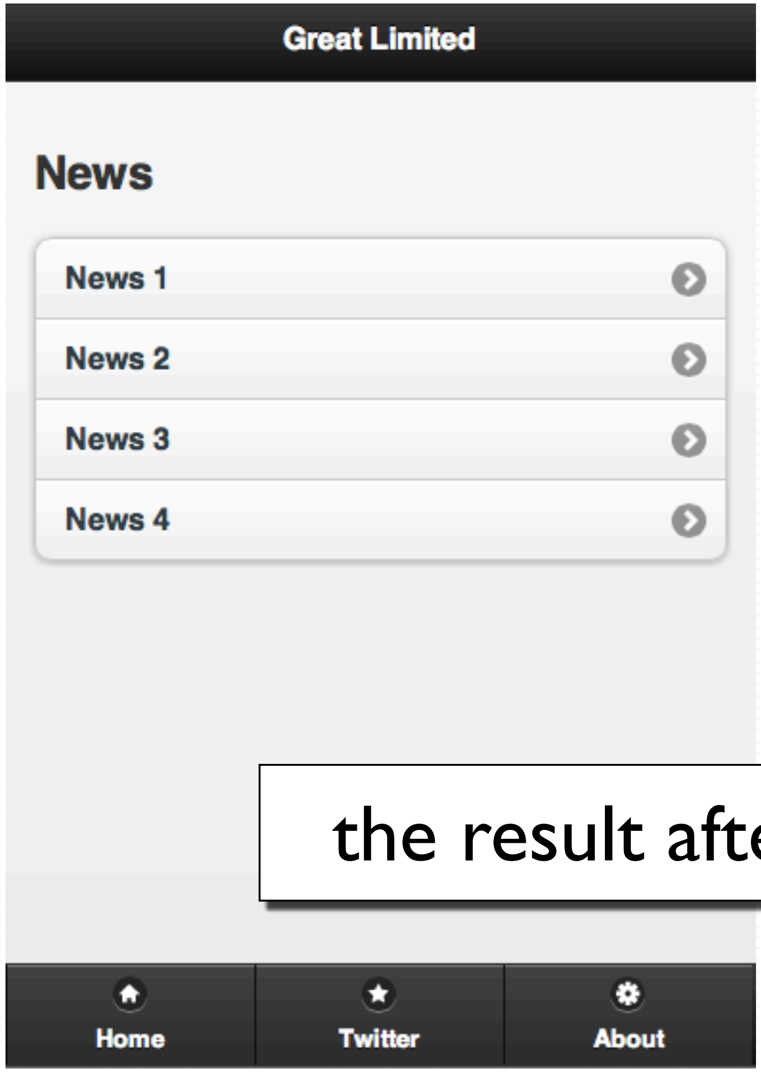
data-inset sets the list to have some margin on left and right edge.

Footer Nav Bar

Nothing special in last section.

In this section, we add a navigation bar at the bottom of the app.

Footer Nav Bar


Great Limited

News

News 1

News 2

News 3

News 4

the result after this section.

Footer Nav Bar

```
<article data-role='page' id='main'>
  <header data-position='fixed' data-role='header'>

  </header>
  <section data-role='content'>

  </section>
  <footer data-id='footer' data-position='fixed' data-role='footer'>
    <div data-role='navbar'>
      <ul>
        <li class='first-page'>
          <a data-icon='home' data-transition='none' href='#main'>Home</a>
        </li>
        <li class='second-page'>
          <a data-icon='star' data-transition='none' href='#twitter'>Twitter</a>
        </li>
        <li class='third-page'>
          <a data-icon='gear' data-transition='none' href='#about'>About</a>
        </li>
      </ul>
    </div>
  </footer>
</article>
```

The same #main element, with footer added.

Footer Nav Bar

```
<article data-role='page' id='main'>
  <header data-position='fixed' data-role='header'>

  </header>
  <section data-role='content'>

  </section>
  <footer data-id='footer' data-position='fixed' data-role='footer'>
    <div data-role='navbar'>
      <ul>
        <li class='first-page'>
          <a data-icon='home' data-transition='none' href='#main'>Home</a>
        </li>
        <li class='second-page'>
          <a data-icon='star' data-transition='none' href='#twitter'>Twitter</a>
        </li>
        <li class='third-page'>
          <a data-icon='gear' data-transition='none' href='#about'>About</a>
        </li>
      </ul>
    </div>
  </footer>
</article>
```

footer role sticks to the bottom.

Footer Nav Bar

```
<article data-role='page' id='main'>
  <header data-position='fixed' data-role='header'>

  </header>
  <section data-role='content'>

  </section>
  <footer data-id='footer' data-position='fixed' data-role='footer'>
    <div data-role='navbar'>
      <ul>
        <li class='first-page'>
          <a data-icon='home' data-transition='none' href='#main'>Home</a>
        </li>
        <li class='second-page'>
          <a data-icon='star' data-transition='none' href='#twitter'>Twitter</a>
        </li>
        <li class='third-page'>
          <a data-icon='gear' data-transition='none' href='#about'>About</a>
        </li>
      </ul>
    </div>
  </footer>
</article>
```

we will share the same data-id across all footers in each page.

Footer Nav Bar

```
<article data-role='page' id='main'>
  <header data-position='fixed' data-role='header'>

  </header>
  <section data-role='content'>

  </section>
  <footer data-id='footer' data-position='fixed' data-role='footer'>
    <div data-role='navbar'>
      <ul>
        <li class='first-page'>
          <a data-icon='home' data-transition='none' href='#main'>Home</a>
        </li>
        <li class='second-page'>
          <a data-icon='star' data-transition='none' href='#twitter'>Twitter</a>
        </li>
        <li class='third-page'>
          <a data-icon='gear' data-transition='none' href='#about'>About</a>
        </li>
      </ul>
    </div>
  </footer>
</article>
```

insider the footer, we have the same navbar as last week.

Footer Nav Bar

```
<article data-role='page' id='main'>
  <header data-position='fixed' data-role='header'>

  </header>
  <section data-role='content'>

  </section>
  <footer data-id='footer' data-position='fixed' data-role='footer'>
    <div data-role='navbar'>
      <ul>
        <li class='first-page'>
          <a data-icon='home' data-transition='none' href='#main'>Home</a>
        </li>
        <li class='second-page'>
          <a data-icon='star' data-transition='none' href='#twitter'>Twitter</a>
        </li>
        <li class='third-page'>
          <a data-icon='gear' data-transition='none' href='#about'>About</a>
        </li>
      </ul>
    </div>
  </footer>
</article>
```

we added data-icon to each link so it looks like a native app's nav bar.

Footer Nav Bar

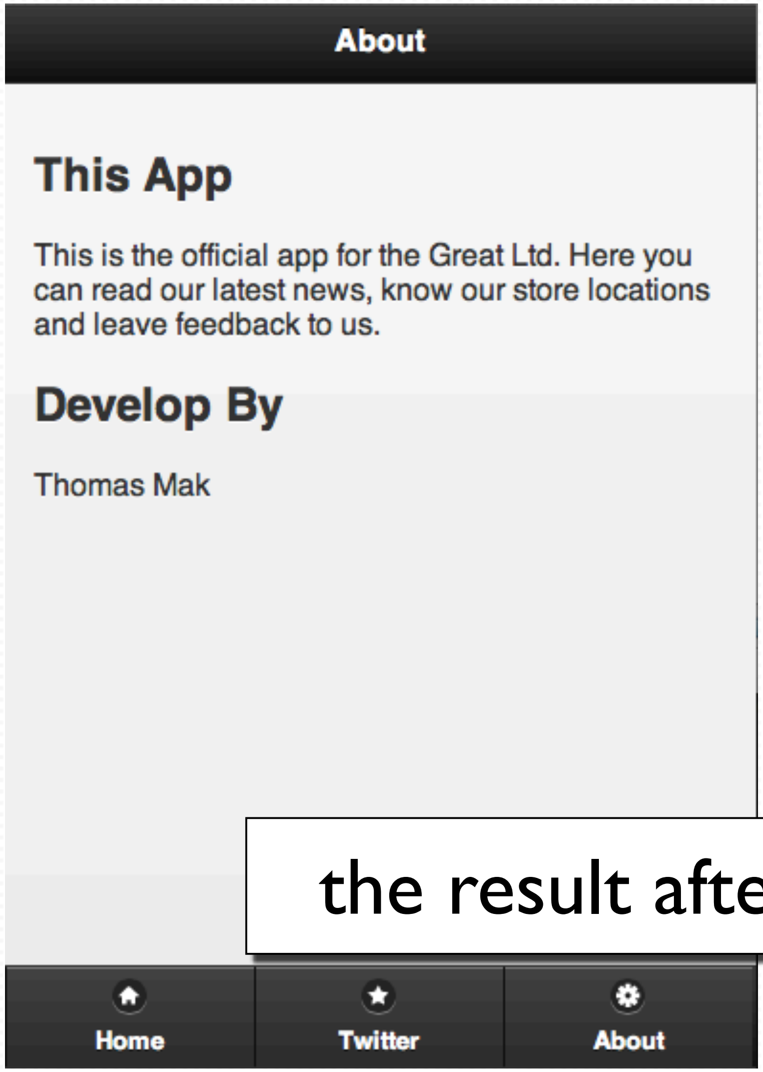
```
<article data-role='page' id='main'>
  <header data-position='fixed' data-role='header'>

  </header>
  <section data-role='content'>

  </section>
  <footer data-id='footer' data-position='fixed' data-role='footer'>
    <div data-role='navbar'>
      <ul>
        <li class='first-page'>
          <a data-icon='home' data-transition='none' href='#main'>Home</a>
        </li>
        <li class='second-page'>
          <a data-icon='star' data-transition='none' href='#twitter'>Twitter</a>
        </li>
        <li class='third-page'>
          <a data-icon='gear' data-transition='none' href='#about'>About</a>
        </li>
      </ul>
    </div>
  </footer>
</article>
```

and normally we do not have page transition when navigating between nav bar.

Other Static Pages

In this section, we will prepare several static pages so later we add logic into them.

Other Static Pages



This App

This is the official app for the Great Ltd. Here you can read our latest news, know our store locations and leave feedback to us.

Develop By

Thomas Mak

the result after this section.

Other Static Pages

```
<!-- ///////// News Detail -->
<article data-role='page' id='news-detail'>
  <header data-position='fixed' data-role='header'>
    <h1>News Title</h1>
    <a data-icon='arrow-l' data-rel='back'>Home</a>
  </header>
  <section data-role='content'>
    <p>News Content here</p>
  </section>
  <footer data-id='footer' data-position='fixed' data-role='footer'>
    <div data-role='navbar'>
      <ul>
        <li class='first-page'>
          <a data-icon='home' data-transition='none' href='#main'>Home</a>
        </li>
        <li class='second-page'>
          <a data-icon='star' data-transition='none' href='#twitter'>Twitter</a>
        </li>
        <li class='third-page'>
          <a data-icon='gear' data-transition='none' href='#about'>About</a>
        </li>
      </ul>
    </div>
  </footer>
</article>
```

the news-detail page

Other Static Pages

```
<!-- /////////// Twitter Page -->
<article data-role='page' id='twitter'>
  <header data-position='fixed' data-role='header'>
    <h1>Twitter</h1>
  </header>
  <section data-role='content'>
    <h2>Twitter Feeds</h2>
    <ul data-inset='true' data-role='listview' id='tweet-list'></ul>
  </section>
  <footer data-id='footer' data-position='fixed' data-role='footer'>
    <div data-role='navbar'>
      <ul>
        <li class='first-page'>
          <a data-icon='home' data-transition='none' href='#main'>Home</a>
        </li>
        <li class='second-page'>
          <a data-icon='star' data-transition='none' href='#twitter'>Twitter</a>
        </li>
        <li class='third-page'>
          <a data-icon='gear' data-transition='none' href='#about'>About</a>
        </li>
      </ul>
    </div>
  </footer>
</article>
```

the twitter page

Other Static Pages

```
<!-- ////////// About Page -->
<article data-role='page' id='about'>
  <header data-position='fixed' data-role='header'>
    <h1>About</h1>
  </header>
  <section data-role='content'>
    <h2>This App</h2>
    <p>
      This is the official app for the Great Ltd. Here you can read our latest news, know ou
store locations and leave feedback to us.
    </p>
    <h2>Develop By</h2>
    <p>Thomas Mak</p>
  </section>
  <footer data-id='footer' data-position='fixed' data-role='footer'>
    <div data-role='navbar'>
      <ul>
        <li class='first-page'>
          <a data-icon='home' data-transition='none' href='#main'>Home</a>
        </li>
        <li class='second-page'>
          <a dat
        </li>
        <li clas
          <a data-icon='gear' data-transition='none' href='#about'>About</a>
```

The About page, the footer part is the same.

Extract the footer

In last section, we created so many duplicated footer.

In this section, we will extract the footer into a template.

Extract the footer

```
<!-- ///////// News Detail -->
<article data-role='page' id='news-detail'>
  <header data-position='fixed' data-role='header'>
    <h1>News Title</h1>
    <a data-icon='arrow-l' data-rel='back'>Home</a>
  </header>
  <section data-role='content'>
    <p>News Content here</p>
  </section>
  <footer data-id='footer' data-position='fixed' data-role='footer'>
    <div data-role='navbar'>
      <ul>
        <li class='first-page'>
          <a data-icon='home' data-transition='none' href='#main'>Home</a>
        </li>
        <li class='second-page'>
          <a data-icon='star' data-transition='none' href='#twitter'>Twitter</a>
        </li>
        <li class='third-page'>
          <a data-icon='gear' data-transition='none' href='#about'>About</a>
        </li>
      </ul>
    </div>
  </footer>
</article>
```

we are going to extract this whole footer tag

Extract the footer

```
<!-- /////////// Twitter Page -->
<article data-role='page' id='twitter'>
  <header data-position='fixed' data-role='header'>
    <h1>Twitter</h1>
  </header>
  <section data-role='content'>
    <h2>Twitter Feeds</h2>
    <ul data-inset='true' data-role='listview' id='tweet-list'></ul>
  </section>
</article>
```

we delete all footer tag in each page.
this is how the twitter page looks like now.

Extract the footer

```
<!-- /////////// Templates -->
<div class='footer-template'>
  <footer data-id='footer' data-position='fixed' data-role='footer'>
    <div data-role='navbar'>
      <ul>
        <li class='first-page'>
          <a data-icon='home' data-transition='none' href='#main'>Home</a>
        </li>
        <li class='second-page'>
          <a data-icon='star' data-transition='none' href='#twitter'>Twitter</a>
        </li>
        <li class='third-page'>
          <a data-icon='gear' data-transition='none' href='#about'>About</a>
        </li>
      </ul>
    </div>
  </footer>
</div>
```

the put the deleted footer tag inside an element named footer-template.

Extract the footer

```
/***** Template *****/
.footer-template {
  display: none;
}
```

the footer-template is for logic use and is not intended to be shown. So we hide it.

Extract the footer

```
(function(){

  // jQuery Ready
  $(function(){
    $("[data-role='page']").each(function() {
      var footer = $(".footer-template").html();
      $(this).append(footer);
    });

    $.mobile.initializePage();
  })

}).call(this);
```

let’s move to the javascript logic.
we append the footer content to each page.

Extract the footer

```
(function(){

  // jQuery Ready
  $(function(){
    $("[data-role='page']").each(function() {
      var footer = $(".footer-template").html();
      $(this).append(footer);
    });

    $.mobile.initializePage();
  })

}).call(this);
```

the [data-role='page'] attribute selector selects all the jQuery Mobile pages.

Extract the footer

```
(function(){

  // jQuery Ready
  $(function(){
    $("[data-role='page']").each(function() {
      var footer = $(".footer-template").html();
      $(this).append(footer);
    });

    $.mobile.initializePage();
  })

}).call(this);
```

and we get the same HTML content of footer.

Extract the footer

```
(function(){

  // jQuery Ready
  $(function(){
    $("[data-role='page']").each(function() {
      var footer = $(".footer-template").html();
      $(this).append(footer);
    });

    $.mobile.initializePage();
  })

}).call(this);
```

finally we tell jQuery Mobile to initialize the page so it renders the footer element into jQuery Mobile style.

Extract the footer

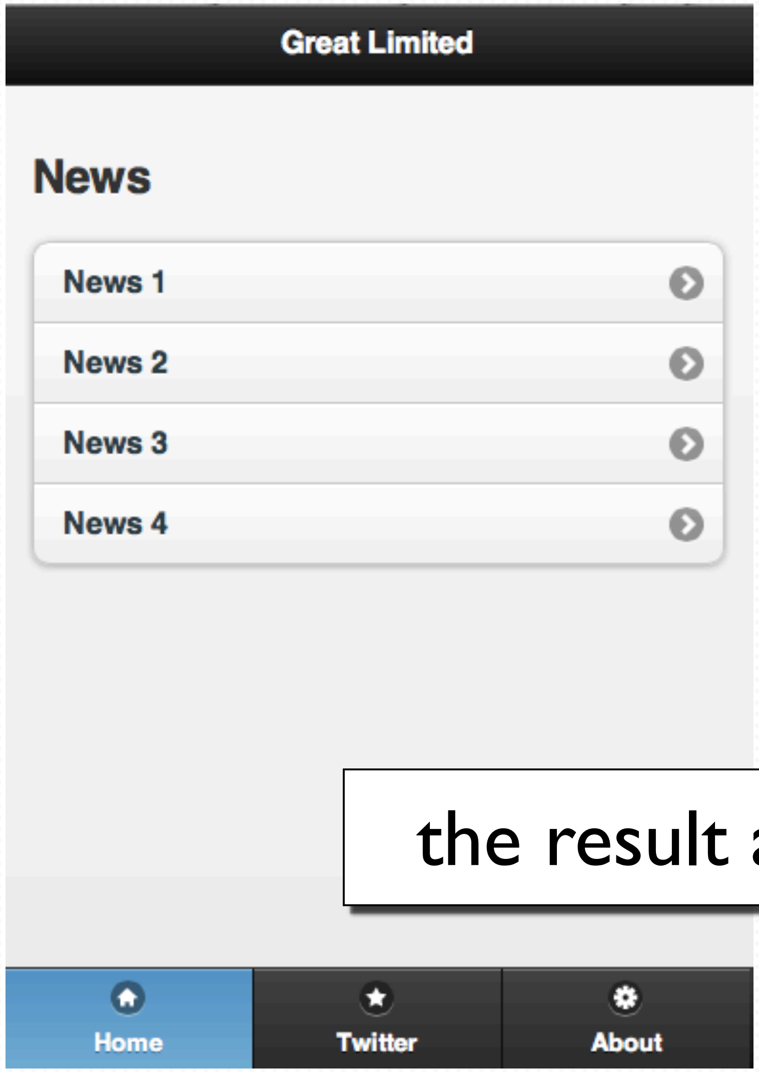
```
<script src='jquery-1.7.2.js'></script>
<script src='jquery.mobile-1.1.0.js'></script>
<script>
  $.mobile.autoInitializePage = false;
</script>
<script src='app.js'></script>
```

one more thing, we tell jQuery mobile that we do not need the auto initialization because we are doing it ourselves.

Nav Bar Active State

In this section, we will set the active state of the footer element.

Nav Bar Active State



Great Limited

News

News 1

News 2

News 3

News 4

the result after this section.

Nav Bar Active State

```
// append footer to each page
$("[data-role='page']").each(function() {
  var footer = $(".footer-template").html();
  $(this).append(footer);


  // footer active state
  var target = 'footer .first-page a';
  if ($(this).attr('id') === 'twitter') {
    target = 'footer .second-page a';
  } else if ($(this).attr('id') === 'about') {
    target = 'footer .third-page a';
  }

  $(this).find(target).addClass('ui-state-persist').addClass('ui-btn-active');

});
```

on each footer, we define the active classes.

Nav Bar Active State

```
// append footer to each page
$("[data-role='page']").each(function() {
  var footer = $(".footer-template").html();
  $(this).append(footer);


  // footer active state
  var target = 'footer .first-page a';
  if ($(this).attr('id') === 'twitter') {
    target = 'footer .second-page a';
  } else if ($(this).attr('id') === 'about') {
    target = 'footer .third-page a';
  }

  $(this).find(target).addClass('ui-state-persist').addClass('ui-btn-active');

});
```

the ui-btn-active class highlight the link as active state.

Nav Bar Active State

```
// append footer to each page
$("[data-role='page']").each(function() {
  var footer = $(".footer-template").html();
  $(this).append(footer);


  // footer active state
  var target = 'footer .first-page a';
  if ($(this).attr('id') === 'twitter') {
    target = 'footer .second-page a';
  } else if ($(this).attr('id') === 'about') {
    target = 'footer .third-page a';
  }

  $(this).find(target).addClass('ui-state-persist').addClass('ui-btn-active');

});
```

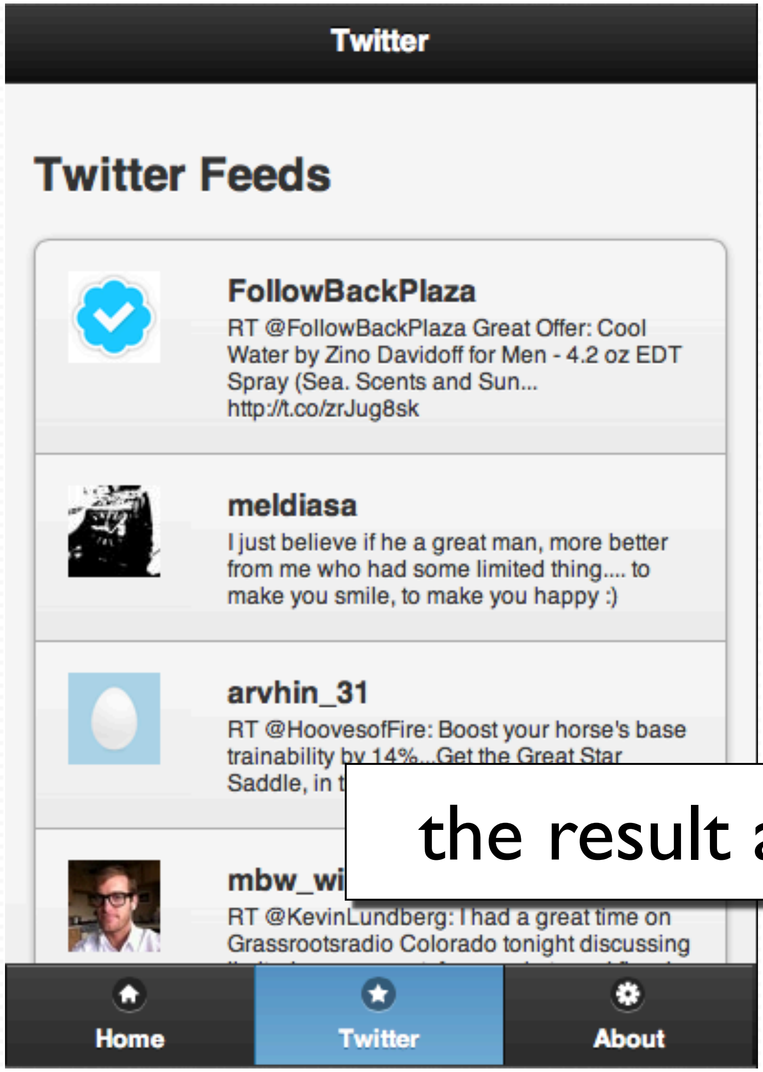
the ui-state-persist makes surethe active state persist.

Twitter Search Query

We are making an app for the company. Most likely we would like to show the company's related tweets.

In this section, we will search the Twitter with the query string "Great Limited" and display the JSON result into listview.

Twitter Search Query



Twitter

Twitter Feeds

FollowBackPlaza
RT @FollowBackPlaza Great Offer: Cool Water by Zino Davidoff for Men - 4.2 oz EDT Spray (Sea. Scents and Sun... http://t.co/zrJug8sk

meldiasa
I just believe if he a great man, more better from me who had some limited thing.... to make you smile, to make you happy :)

arvhin_31
RT @HoovesofFire: Boost your horse's base trainability by 14%...Get the Great Star Saddle, in t

mbw_wi
RT @KevinLundberg: I had a great time on Grassrootsradio Colorado tonight discussing

the result after this section.

Twitter Search Query

```
<!-- /////////// Twitter Page -->
<article data-role='page' id='twitter'>
  <header data-position='fixed' data-role='header'>
    <h1>Twitter</h1>
  </header>
  <section data-role='content'>
    <h2>Twitter Feeds</h2>
    <ul data-inset='true' data-role='listview' id='tweet-list'></ul>
  </section>
</article>
```

the major part is the listview with id #tweet-list

Twitter Search Query

```
var TwitterQuery = (function(){
  function TwitterQuery(query) {
    this.query = query;
    this.twitterURL = "http://search.twitter.com/search.json?q=" + this.query +
"&rpp=10&callback=?";
  }

  TwitterQuery.prototype.fetch = function(element) {
    $.getJSON(this.twitterURL, function(data) {
      $(element).empty();
      for (var i=0, len=data.results.length; i < len; i++) {
        var tweet = data.results[i];
        $(element).append("<li><img src='images/empty.png' style='background-image:url(" +
tweet.profile_image_url + ")'><h1>" + tweet.from_user + "</h1> <p>" + tweet.text + "</p> </
li>");
      }
      $(element).listview('refresh');
    });
  }

  return TwitterQuery;
})();
```

TwitterQuery Class

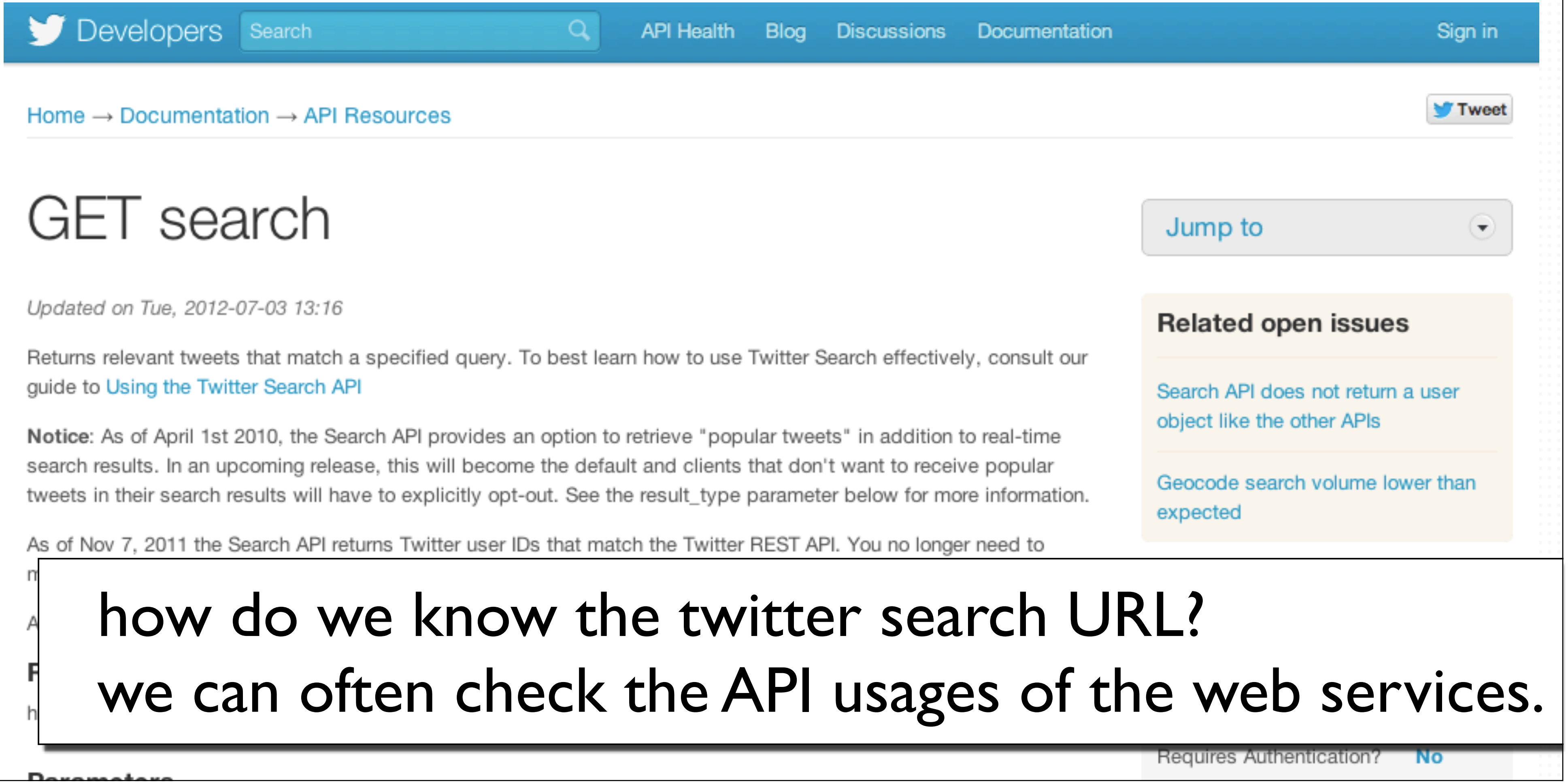
Twitter Search Query

```
var TwitterQuery = (function(){
  function TwitterQuery(query) {
    this.query = query;
    this.twitterURL = "http://search.twitter.com/search.json?q=" + this.query +
"&rpp=10&callback=?";
  }

  TwitterQuery.prototype.fetch = function(element) {
    $.getJSON(this.twitterURL, function(data) {
      $(element).empty();
      for (var i=0, len=data.results.length; i < len; i++) {
        var tweet = data.results[i];
        $(element).append("<li><img src='images/empty.png' style='background-image:url(" +
tweet.profile_image_url + ")'><h1>" + tweet.from_user + "</h1> <p>" + tweet.text + "</p> </
li>");
      }
      $(element).listview('refresh');
    });
  }

  return TwitterQuery;
})();
```

constructor with twitter search query.

Twitter Search Query

https://dev.twitter.com/docs/api/1/get/search



Developers Search API Health Blog Discussions Documentation Sign in

Home → Documentation → API Resources

Tweet

GET search

Jump to

Updated on Tue, 2012-07-03 13:16

Returns relevant tweets that match a specified query. To best learn how to use Twitter Search effectively, consult our guide to Using the Twitter Search API

Notice: As of April 1st 2010, the Search API provides an option to retrieve "popular tweets" in addition to real-time search results. In an upcoming release, this will become the default and clients that don't want to receive popular tweets in their search results will have to explicitly opt-out. See the result_type parameter below for more information.

As of Nov 7, 2011 the Search API returns Twitter user IDs that match the Twitter REST API. You no longer need to

Related open issues

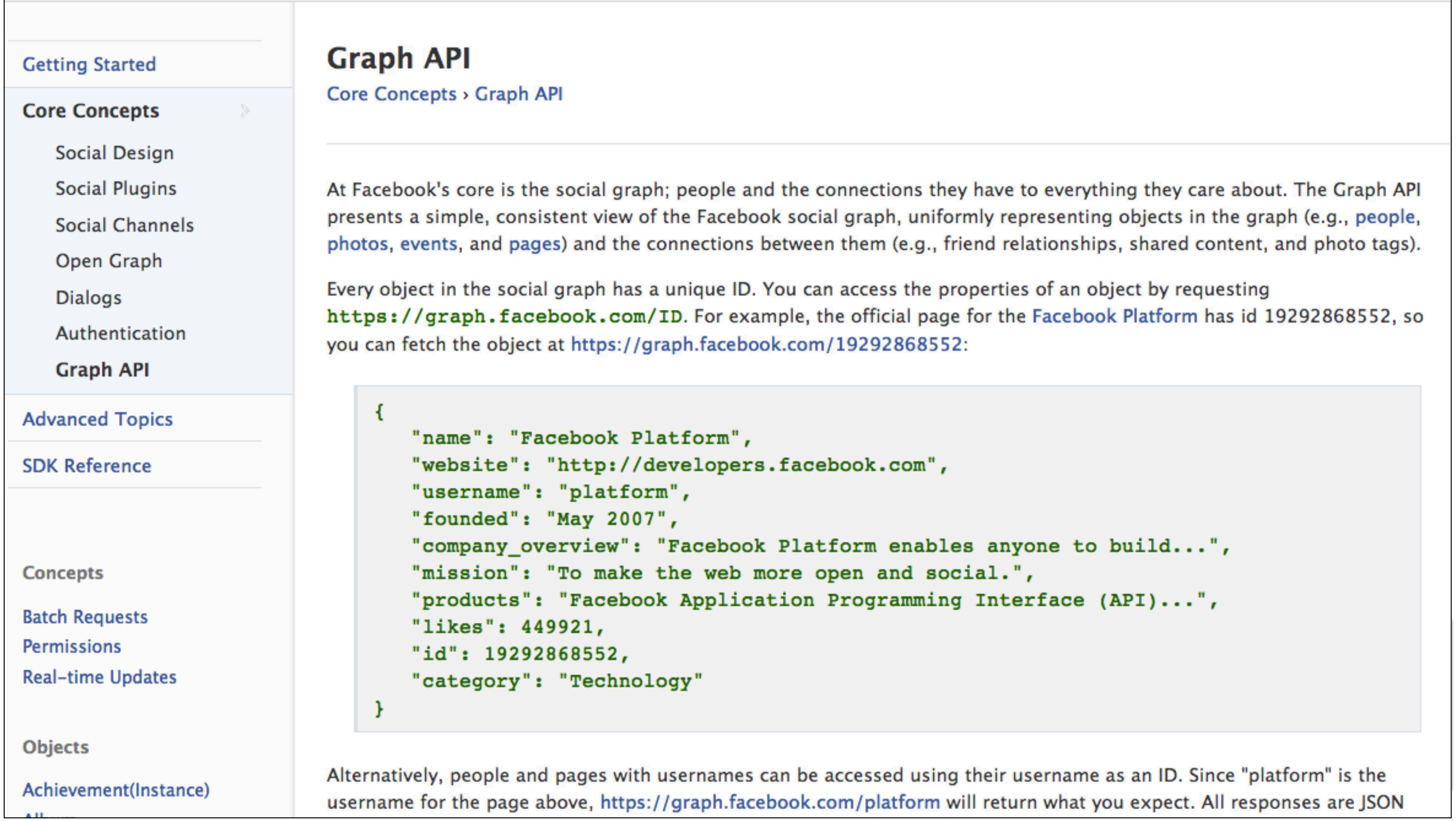
Search API does not return a user object like the other APIs

Geocode search volume lower than expected

Requires Authentication? No

how do we know the twitter search URL?
we can often check the API usages of the web services.

Twitter Search Query



Graph API

Core Concepts › Graph API

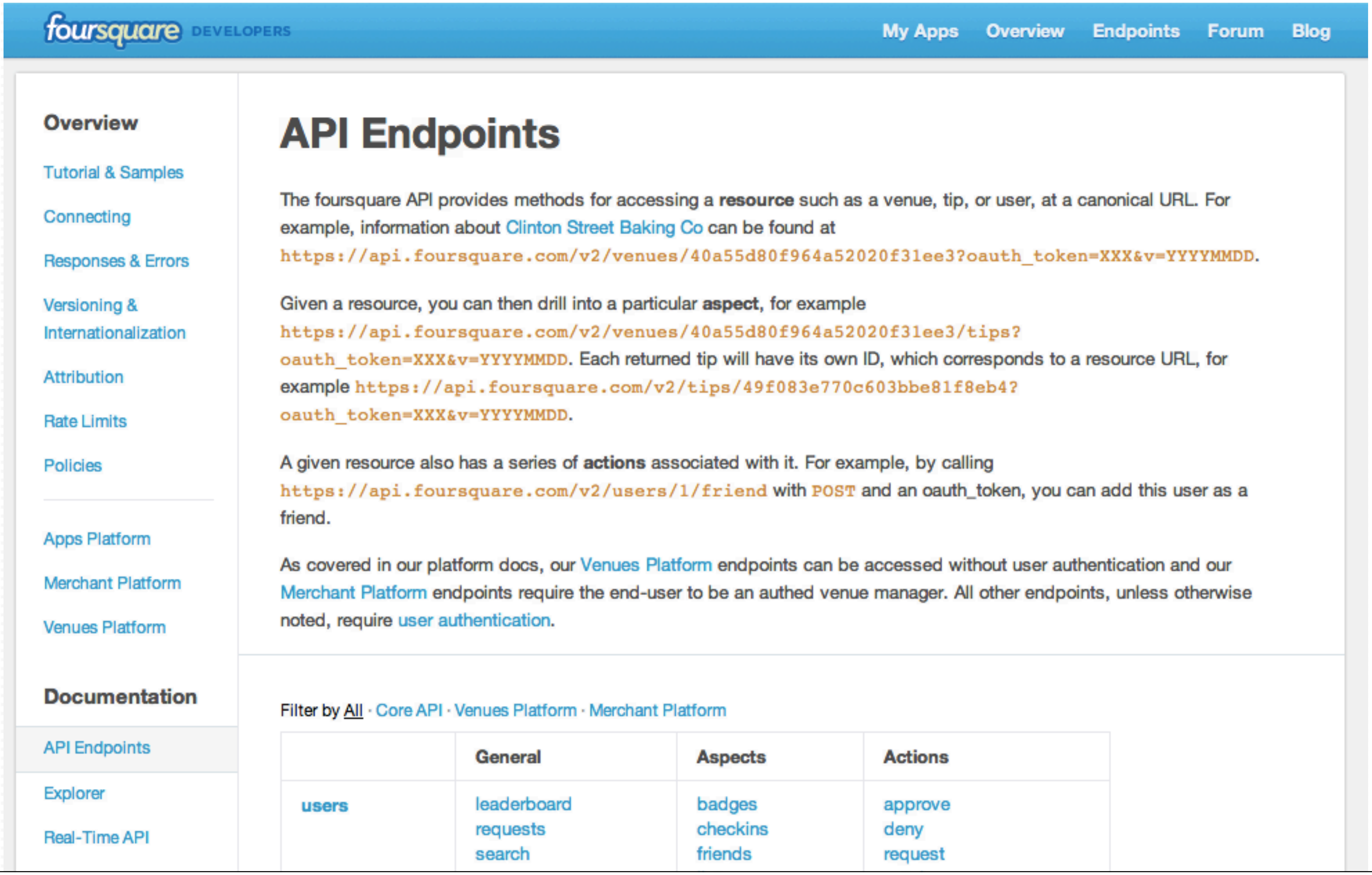
At Facebook's core is the social graph; people and the connections they have to everything they care about. The Graph API presents a simple, consistent view of the Facebook social graph, uniformly representing objects in the graph (e.g., people, photos, events, and pages) and the connections between them (e.g., friend relationships, shared content, and photo tags).

Every object in the social graph has a unique ID. You can access the properties of an object by requesting `https://graph.facebook.com/ID`. For example, the official page for the Facebook Platform has id 19292868552, so you can fetch the object at https://graph.facebook.com/19292868552:

```
{
   "name": "Facebook Platform",
   "website": "http://developers.facebook.com",
   "username": "platform",
   "founded": "May 2007",
   "company_overview": "Facebook Platform enables anyone to build...",
   "mission": "To make the web more open and social.",
   "products": "Facebook Application Programming Interface (API)...",
   "likes": 449921,
   "id": 19292868552,
   "category": "Technology"
}
```

Alternatively, people and pages with usernames can be accessed using their username as an ID. Since "platform" is the username for the page above, https://graph.facebook.com/platform will return what you expect. All responses are JSON

Twitter Search Query



API Endpoints

The foursquare API provides methods for accessing a **resource** such as a venue, tip, or user, at a canonical URL. For example, information about Clinton Street Baking Co can be found at `https://api.foursquare.com/v2/venues/40a55d80f964a52020f31ee3?oauth_token=XXX&v=YYYYMMDD`.

Given a resource, you can then drill into a particular **aspect**, for example `https://api.foursquare.com/v2/venues/40a55d80f964a52020f31ee3/tips?oauth_token=XXX&v=YYYYMMDD`. Each returned tip will have its own ID, which corresponds to a resource URL, for example `https://api.foursquare.com/v2/tips/49f083e770c603bbe81f8eb4?oauth_token=XXX&v=YYYYMMDD`.

A given resource also has a series of **actions** associated with it. For example, by calling `https://api.foursquare.com/v2/users/1/friend` with `POST` and an oauth_token, you can add this user as a friend.

As covered in our platform docs, our Venues Platform endpoints can be accessed without user authentication and our Merchant Platform endpoints require the end-user to be an authed venue manager. All other endpoints, unless otherwise noted, require user authentication.

Filter by All · Core API · Venues Platform · Merchant Platform

	General	Aspects	Actions
users	leaderboard requests search	badges checkins friends	approve deny request

Twitter Search Query

```
var TwitterQuery = (function(){
  function TwitterQuery(query) {
    this.query = query;
    this.twitterURL = "http://search.twitter.com/search.json?q=" + this.query +
"&rpp=10&callback=?";
  }

  TwitterQuery.prototype.fetch = function(element) {
    $.getJSON(this.twitterURL, function(data) {
      $(element).empty();
      for (var i=0, len=data.results.length; i < len; i++) {
        var tweet = data.results[i];
        $(element).append("<li><img src='images/empty.png' style='background-image:url(" +
tweet.profile_image_url + ")'><h1>" + tweet.from_user + "</h1> <p>" + tweet.text + "</p> </
li>");
      }
      $(element).listview('refresh');
    });
  }

  return TwitterQuery;
})();
```

fetch method fetches the tweets and display it inside the given element.

Twitter Search Query

```
var TwitterQuery = (function(){
  function TwitterQuery(query) {
    this.query = query;
    this.twitterURL = "http://search.twitter.com/search.json?q=" + this.query +
"&rpp=10&callback=?";
  }

  TwitterQuery.prototype.fetch = function(element) {
    $.getJSON(this.twitterURL, function(data) {
      $(element).empty();
      for (var i=0, len=data.results.length; i < len; i++) {
        var tweet = data.results[i];
        $(element).append("<li><img src='images/empty.png' style='background-image:url(" +
tweet.profile_image_url + ")'><h1>" + tweet.from_user + "</h1> <p>" + tweet.text + "</p> </
li>");
      }
      $(element).listview('refresh');
    });
  }

  return TwitterQuery;
})();
```

getJSON or ajax call from jQuery

Twitter Search Query

```
var TwitterQuery = (function(){
  function TwitterQuery(query) {
    this.query = query;
    this.twitterURL = "http://search.twitter.com/search.json?q=" + this.query +
"&rpp=10&callback=?";
  }

  TwitterQuery.prototype.fetch = function(element) {
    $.getJSON(this.twitterURL, function(data) {
      $(element).empty();
      for (var i=0, len=data.results.length; i < len; i++) {
        var tweet = data.results[i];
        $(element).append("<li><img src='images/empty.png' style='background-image:url(" +
tweet.profile_image_url + ")'><h1>" + tweet.from_user + "</h1> <p>" + tweet.text + "</p> </
li>");
      }
      $(element).listview('refresh');
    });
  }

  return TwitterQuery;
})();
```

the tweets are inside data.results array

Twitter Search Query

```
var TwitterQuery = (function(){
  function TwitterQuery(query) {
    this.query = query;
    this.twitterURL = "http://search.twitter.com/search.json?q=" + this.query +
"&rpp=10&callback=?";
  }

  TwitterQuery.prototype.fetch = function(element) {
    $.getJSON(this.twitterURL, function(data) {
      $(element).empty();
      for (var i=0, len=data.results.length; i < len; i++) {
        var tweet = data.results[i];
        $(element).append("<li><img src='images/empty.png' style='background-image:url(" +
tweet.profile_image_url + ")'><h1>" + tweet.from_user + "</h1> <p>" + tweet.text + "</p> </
li>");
      }
      $(element).listview('refresh');
    });
  }

  return TwitterQuery;
})();
```

append the tweet information as listview.

Twitter Search Query

```
var TwitterQuery = (function(){
  function TwitterQuery(query) {
    this.query = query;
    this.twitterURL = "http://search.twitter.com/search.json?q=" + this.query +
"&rpp=10&callback=?";
  }

  TwitterQuery.prototype.fetch = function(element) {
    $.getJSON(this.twitterURL, function(data) {
      $(element).empty();
      for (var i=0, len=data.results.length; i < len; i++) {
        var tweet = data.results[i];
        $(element).append("<li><img src='images/empty.png' style='background-image:url(" +
tweet.profile_image_url + ")'><h1>" + tweet.from_user + "</h1> <p>" + tweet.text + "</p> </
li>");
      }
      $(element).listview('refresh');
    });
  }

  return TwitterQuery;
})();
```

finally we refresh the listview.

Twitter Search Query

```
// Twitter Query Module
(function() {

  var TwitterQuery = (function(){
    // our Twitter Query class implementation.
  })();

  // export the TwitterQuery to global scope
  if (!this.greatLtd) this.greatLtd = {}
  this.greatLtd.TwitterQuery = TwitterQuery;


}).call(this);
```

since this is quite independent to the other logics, we would like to put the class as an individual module.

Twitter Search Query

```
// inside jQuery ready function
// listen to the pageBeforeShow event
$(document).bind('pagebeforeshow', function(event, ui) {
  if ($(event.target).attr('id') === 'twitter') {
    twitter = new greatLtd.TwitterQuery("Great Limited");
    twitter.fetch($("#tweet-list"));
  }
});
```

we need to call the TwitterQuery class.
we call it on every time the twitter page shows.

Twitter Search Query

```
// inside jQuery ready function
// listen to the pageBeforeShow event
$(document).bind('pagebeforeshow', function(event, ui) {
  if ($(event.target).attr('id') === 'twitter') {
    twitter = new greatLtd.TwitterQuery("Great Limited");
    twitter.fetch($("#tweet-list"));
  }
});
```

the pagebeforeshow event is fired before the page is show.

Twitter Search Query

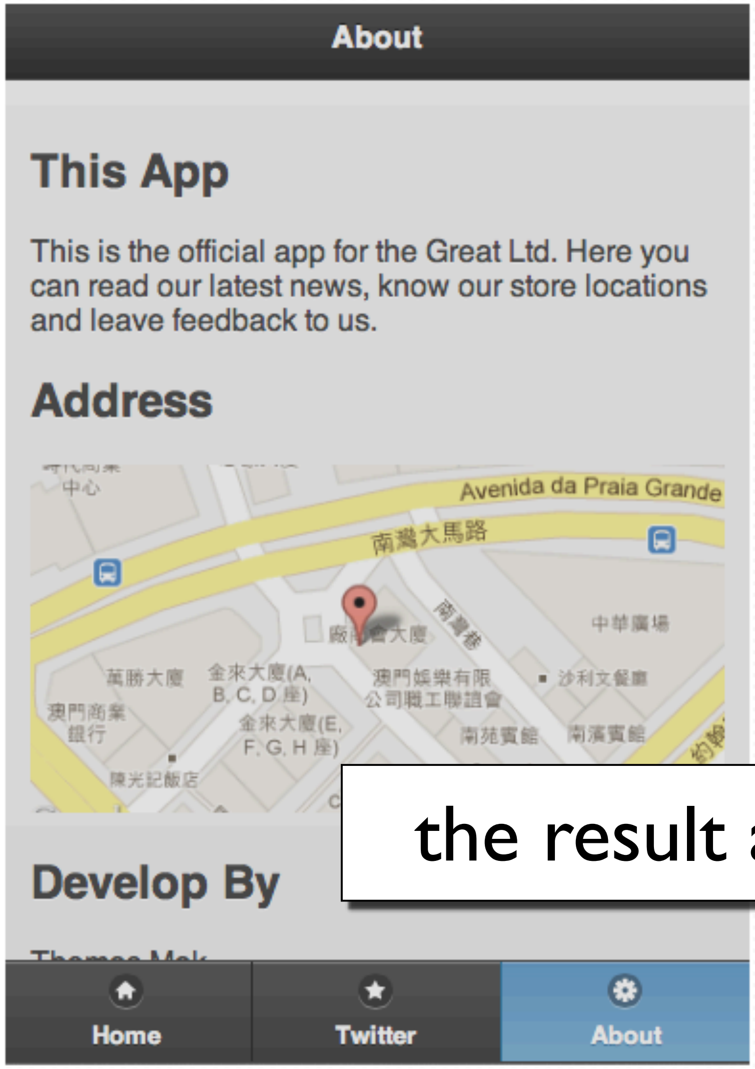
```
pagebeforeshow - called before the next page shows

pageshow - called after the next page finished in transitions and shows.

pagebeforehide - called before the current page hides

pagehide - called after the current page finished out transitions and hided.
```

we have several more events.

Showing Location in Map

In this section, we will how the company location in a map inside the about page.

Showing Location in Map



About

This App

This is the official app for the Great Ltd. Here you can read our latest news, know our store locations and leave feedback to us.

Address

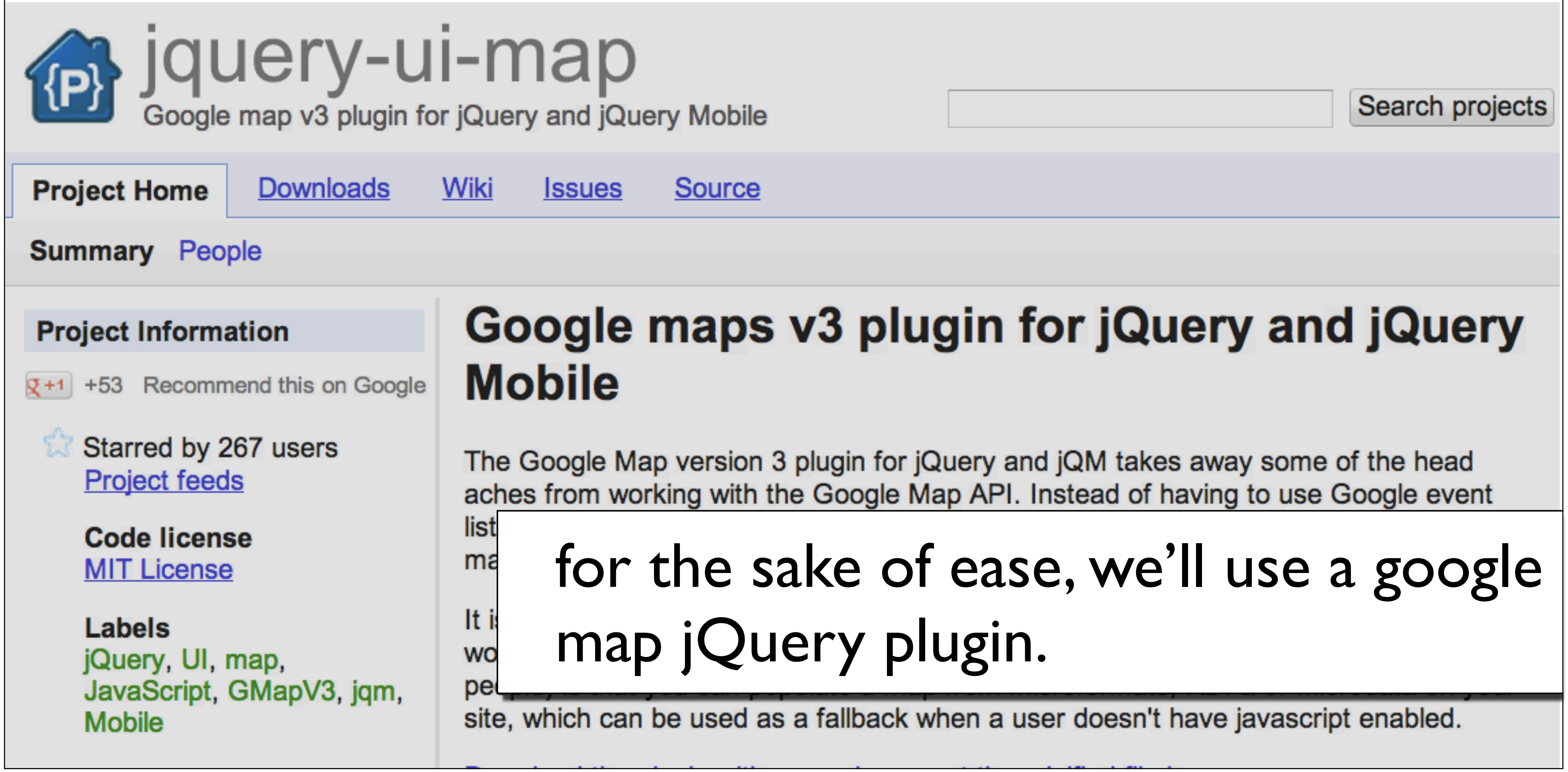
Develop By

Thomas Mak

the result after this section.

Showing Location in Map

http://code.google.com/p/jquery-ui-map/



Project Information

+53 Recommend this on Google

Starred by 267 users
Project feeds

Code license
MIT License

Labels
jQuery, UI, map, JavaScript, GMapV3, jqm, Mobile

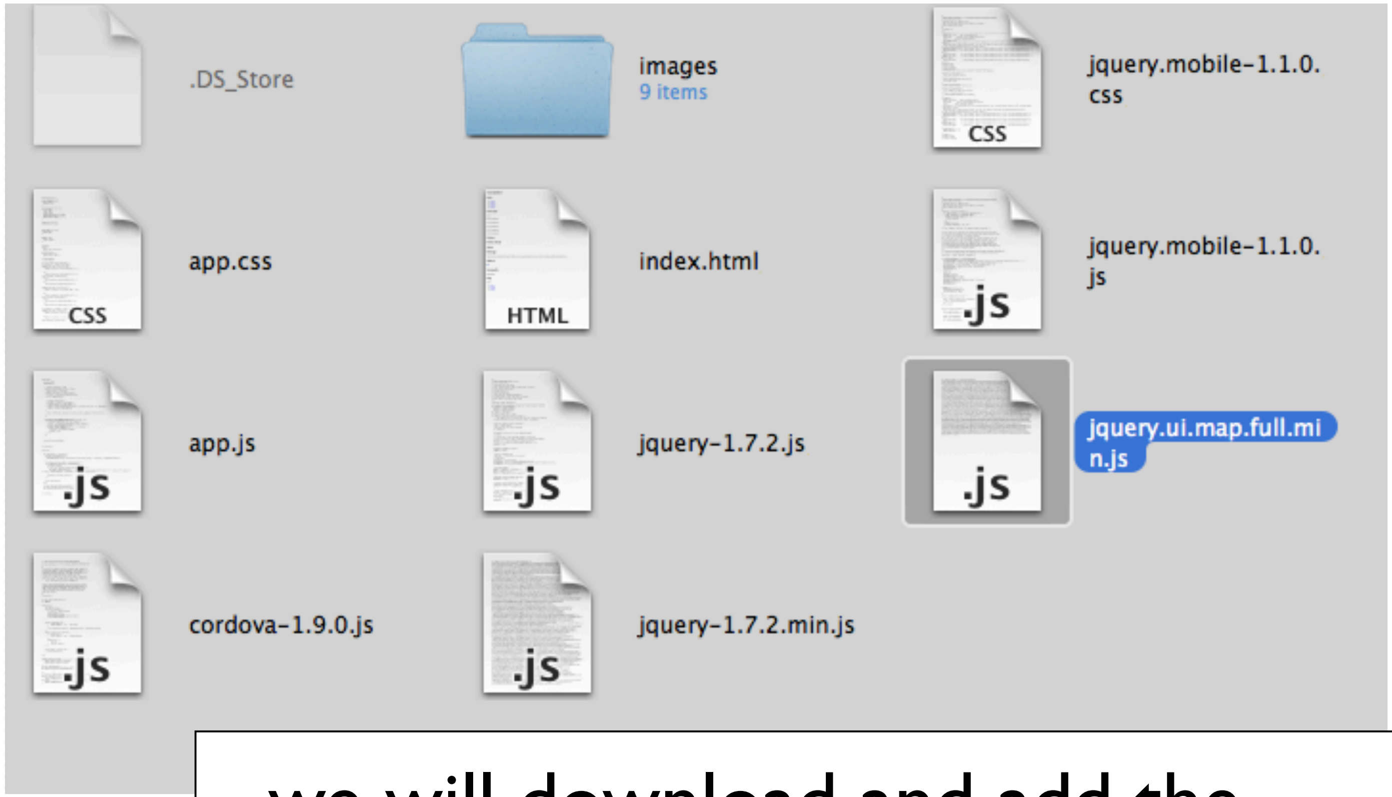
Google maps v3 plugin for jQuery and jQuery Mobile

The Google Map version 3 plugin for jQuery and jQM takes away some of the head aches from working with the Google Map API. Instead of having to use Google event list
ma

It i
wo
pe
site, which can be used as a fallback when a user doesn't have javascript enabled.

for the sake of ease, we'll use a google map jQuery plugin.

Showing Location in Map



we will download and add the jquery.ui.map.full.min.js file to our project.

Showing Location in Map

```
<!-- ////////// Map Page -->
<article data-role='page' id='map-page'>
  <header data-position='fixed' data-role='header'>
    <h1>Map</h1>
    <a data-icon='arrow-l' data-rel='back'>About</a>
  </header>
  <section data-role='content'>
    <div id='map'></div>
  </section>
</article>
```

let’s add a map page to the index.html

Showing Location in Map

```
<!-- ////////// Map Page -->
<article data-role='page' id='map-page'>
  <header data-position='fixed' data-role='header'>
    <h1>Map</h1>
    <a data-icon='arrow-l' data-rel='back'>About</a>
  </header>
  <section data-role='content'>
    <div id='map'></div>
  </section>
</article>
```

the empty #map element is the hook that we are going to apply the map there.

Showing Location in Map

```
$('#map').gmap('addMarker', {'position': '57.7973333,12.0502107', 'bounds': true})
```

we are going to use the above code to apply the google map to the #map element.

Showing Location in Map

```
<h2>Address</h2>
<a data-transition='pop' href='#map-page'>
  <img src='images/map.png'>
</a>
```

we need a link to the map page, let’s add it into our About page.

Showing Location in Map

```
<h2>Address</h2>
<a data-transition='pop' href='#map-page'>
  <img src='images/map.png'>
</a>
```

Avenida da Praia Grande
南灣大馬路
中心
中華廣場
萬勝大廈
金來大廈(A, B, C, D 座)
澳門娛樂有限公司職工聯誼會
沙利文餐廳
澳門商業銀行
金來大廈(E, F, G, H 座)
南苑賓館
南濱賓館
陳光記飯店
Caravela Pastcladia
Cafe e Nata

the map.png is a static image file of the location snapshot.

Showing Location in Map

```
<script src='http://maps.google.com/maps/api/js?sensor=true'></script>
<script src='jquery.ui.map.full.min.js'></script>
```

let's also include the map related javascript files before our app.js

Showing Location in Map

```
<script src='http://maps.google.com/maps/api/js?sensor=true'></script>
<script src='jquery.ui.map.full.min.js'></script>
```

note that we included the google map official javascript API.

Showing Location in Map

```
// listen to the pageBeforeShow event
$(document).bind('pagebeforeshow', function(event, ui) {
  if ($(event.target).attr('id') === 'twitter') {
    twitter = new greatLtd.TwitterQuery("Great Limited");
    twitter.fetch($("#tweet-list"));
  } else if ($(event.target).attr('id') === 'map-page') {
    $('#map').gmap('addMarker', {
      'position': '22.192362,113.54206',
      'bounds': true
    });
  }
});
```

now we apply the google map to the #ma
element when the map-page shows.

Showing Location in Map

```
// listen to the pageBeforeShow event
$(document).bind('pagebeforeshow', function(event, ui) {
  if ($(event.target).attr('id') === 'twitter') {
    twitter = new greatLtd.TwitterQuery("Great Limited");
    twitter.fetch($("#tweet-list"));
  } else if ($(event.target).attr('id') === 'map-page') {
    $('#map').gmap('addMarker', {
      'position': '22.192362,113.54206',
      'bounds': true
    });
  }
});
```

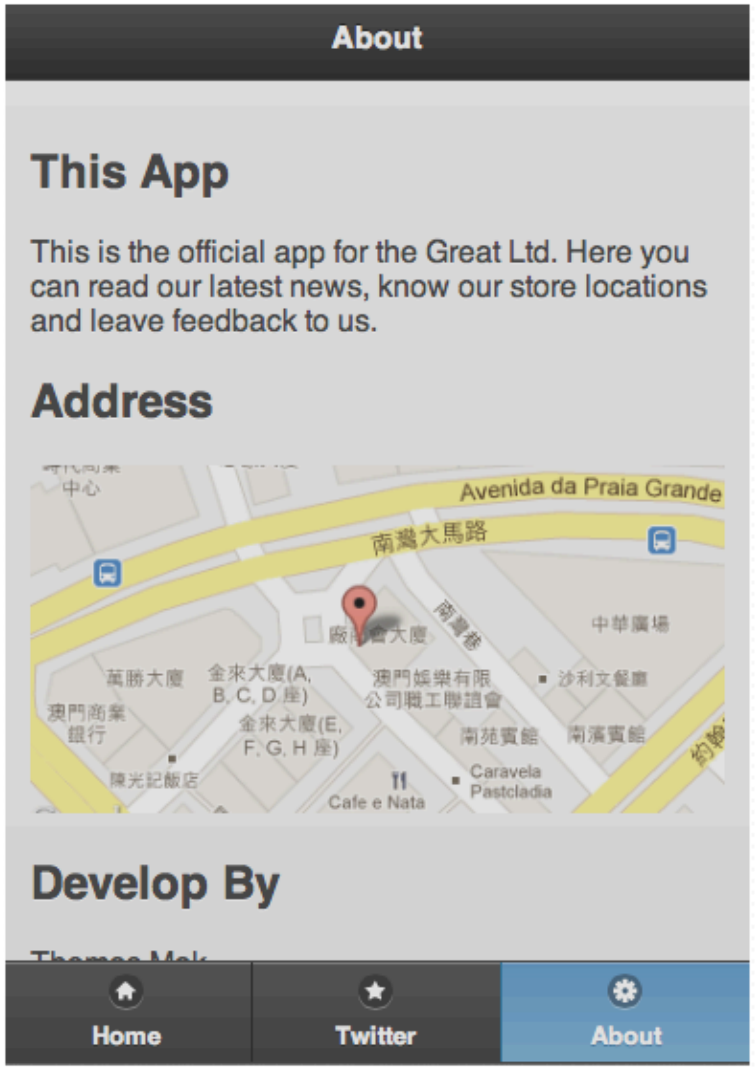
the position is where we are now.
the bounds tells the google map to display the area of current position.

Showing Location in Map

```
/***** Map View *****/
#about img {
  width: 100%;
}

#map {
  width: 100%;
  height: 400px;
}
```

before we test the app, we need some styles to define the dimension of the map.

Showing Location in Map



About

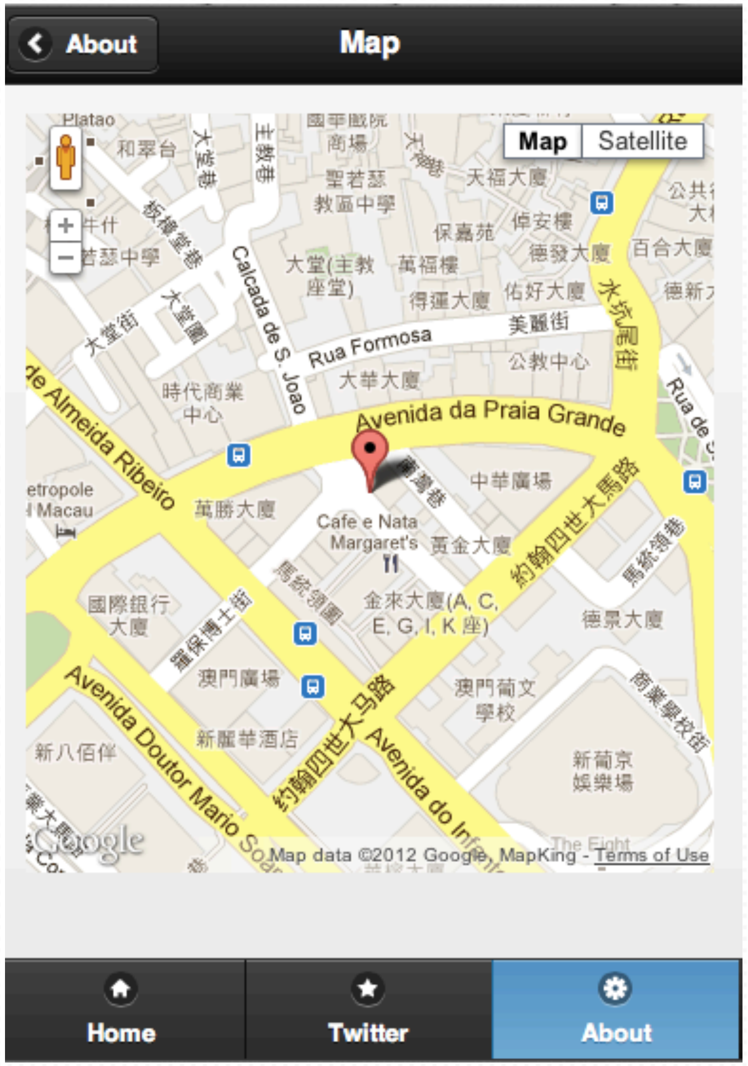
This App

This is the official app for the Great Ltd. Here you can read our latest news, know our store locations and leave feedback to us.

Address

Develop By

Thomas Mak



About
Map
Map
Satellite
Avenida da Praia Grande
Rua Formosa
Calcada de S. Joao
Avenida Doutor Mario Soares
Cafe e Nata Margaret's
Map data ©2012 Google, MapKing - Terms of Use
Home
Twitter
About