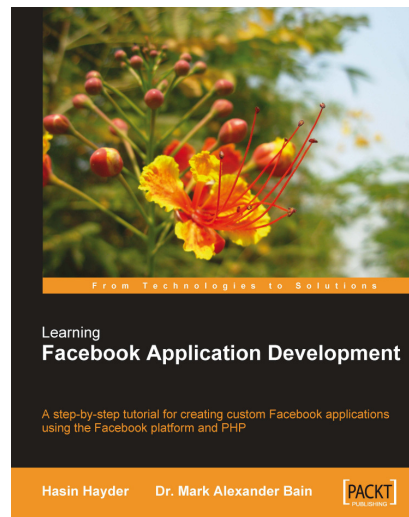




Learning Facebook Application Development

Hasin Hayder
Dr. Mark Alexander Bain



Chapter No. 1

"Getting to Grips with the Facebook Platform"

In this package, you will find:

A Biography of the authors of the book

A preview chapter from the book, Chapter NO.1 "Getting to Grips with the Facebook Platform"

A synopsis of the book's content

Information on where to buy this book

About the Authors

Hasin Hayder, graduated in Civil Engineering from the Rajshahi University of Engineering and Technology (RUET) in Bangladesh. He is a Zend-certified Engineer and expert in developing localized applications. He is currently working as a Technical Director in Trippert Labs and managing the local branch in Bangladesh. Beside his full time job, Hasin writes his blog at <http://hasin.wordpress.com>, writes article in different websites and maintains his open-source framework Orchid at <http://orchid.phpxperts.net>. Hasin lives in Bangladesh with his wife Ayesha and his son, Afif.

Thanks goes to my colleagues at TrippertLabs and phpExperts, for supporting me during this time. Thanks to Abhijeet, Nikhil, and David for their patience while I was late. And definitely, thanks to my family members, especially Ayesha and Afif for their support.

Dr. Mark Alexander Bain first started customizing CRM systems back in the mid '90s when he was team leader for Vodafone's Cascade project—the team took the 'out-of-the-box' Clarify CRM and turned it into a radio base station planning application, complete with a workflow engine for passing jobs between the different departments involved in the planning, building, and implementation of a radio network. Since then, he's lectured at the University of Central Lancashire. Currently Mark writes articles on all things Linux and Open Source for Linux Format, Newsforge.com and Linux Journal. SugarCRM customization, therefore, seems the obvious choice for this, his second book, since it combines Mark's knowledge of working with commercial CRMs and the Open Source philosophy. Mark works from his home on the edge of the Lake District in the UK, where he lives with his wife, two dogs and two cats, and gets the odd visit from his sons—Michael and Simon.

For More Information:

www.packtpub.com/learning-facebook-application-development/book

Learning Facebook Application Development

Using Facebook applications, developers can add custom features to one of the most popular websites in the world. Facebook is the biggest social network among college students, and is gaining ground among professionals too. Facebook applications enable you to add new ways for users to interact with each other using Facebook.

Facebook applications are written in a server-side language, and hosted on the application developer's own server. They interface with Facebook, so that they appear to users to be part of Facebook itself.

This book shows PHP developers how to quickly get started building powerful Facebook applications, how to work with data stored in Facebook, including photos, and how to handle multimedia and other custom data within Facebook applications.

The book covers how to send notifications and invitations from within an application, update a user's profile, create application control panels for users, and much more.

For More Information:

www.packtpub.com/learning-facebook-application-development/book

What This Book Covers

Chapter 1 takes a look at the elements used by your application to interface with Facebook—the Facebook API, FBML, and FQL—that is the Facebook Platform.

Chapter 2 will cover how to extract the Facebook information to be displayed by your application, store information in the Facebook cache so that it can be displayed on your users' Profile pages, and store your own custom data.

Chapter 3 will cover how to manipulate data stored in a database, update your users' profiles automatically, track the users who are using your application, and use the Facebook Dashboard.

Chapter 4 will help you understand Facebook tags to display information in a hassle free and easier way. You will also get an introduction to Facebook Query Language (FQL), and understand why it is sometimes more effective to use FQL rather than the Facebook REST API.

Chapter 5 will see the scripting ability in Facebook, using a subset of standard JavaScript functions developed by Facebook, which is called FBJS.

Chapter 6 will go into the details of managing feeds (both news feed and mini feed) using Facebook REST APIs.

Chapter 7 will focus on creating a successful invitation system, creating a successful notification system, using notifications efficiently, and sending emails.

Chapter 8 will show you how to do the following: creating a photo importer from Facebook, creating a slide show application, and letting viewers upload photos to your album.

Chapter 9 will cover the usage of various tags in Facebook and create an interesting project using them.

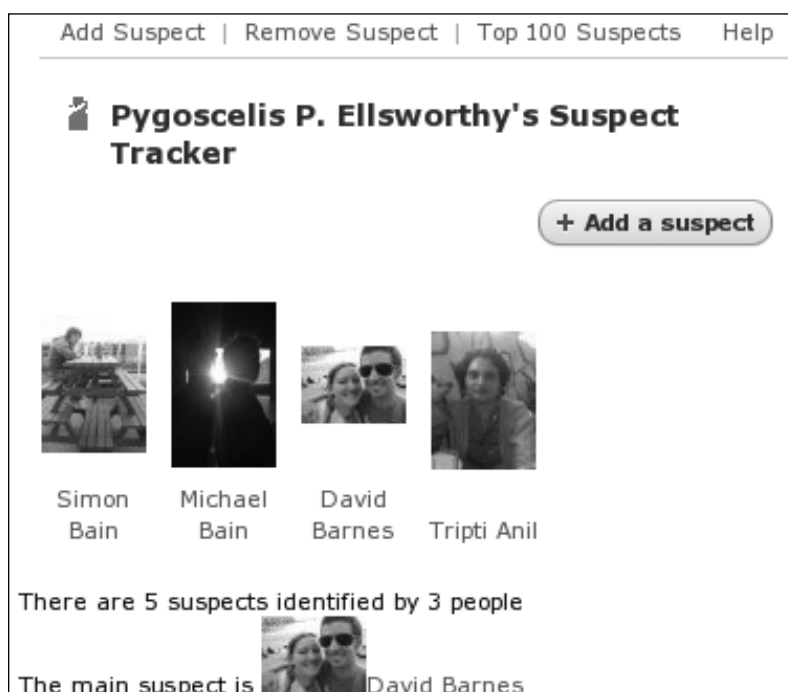
For More Information:

www.packtpub.com/learning-facebook-application-development/book

1

Getting to Grips with the Facebook Platform

If you've read even this far, then it's safe for me to assume that you're already well acquainted with Facebook, and (given the name of this book) it's also fairly safe to assume that you don't want an in-depth discussion on how to use Facebook – you just want to get on with building a Facebook application. So, that's exactly what we're going to do – we're going to build *Pygoscelis P. Ellsworth's Suspect Tracker* as an example:



For More Information:

www.packtpub.com/learning-facebook-application-development/book

However, before we jump into building a Facebook application, we need to spend some time looking at the Facebook platform, and by the end of this chapter you will:

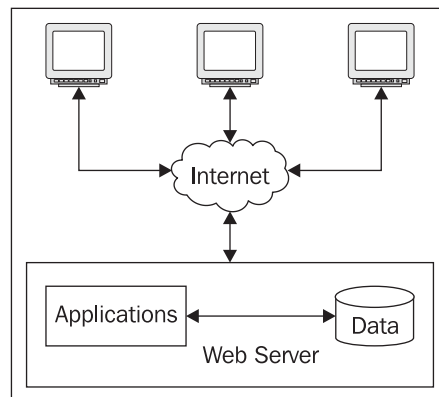
- Understand what the Facebook Platform is, and how it relates to your application
- Know about the elements that make up the Facebook Platform, and how to test them without having to create an application
- Know how to set up the Facebook Platform, ready for your new application

The Purpose of the Facebook Platform

As you develop your Facebook applications, you'll find that the Facebook Platform is essential—in fact you won't really be able to do anything without it. So what does it do? Well, before answering that, let's look at a typical web-based application.

The Standard Web Application Model

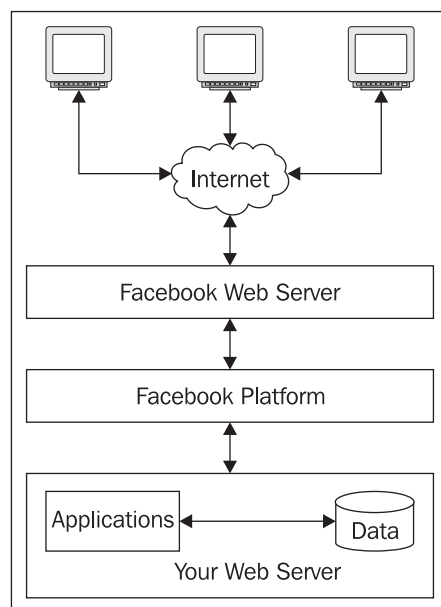
If you've ever designed and built a web application before, then you'd have done it in a fairly standard way. Your application and any associated data would have been placed on a web server, and then your application users will access it from their web browsers via the Internet:



The Facebook model is slightly different.

The Facebook Web Application Model

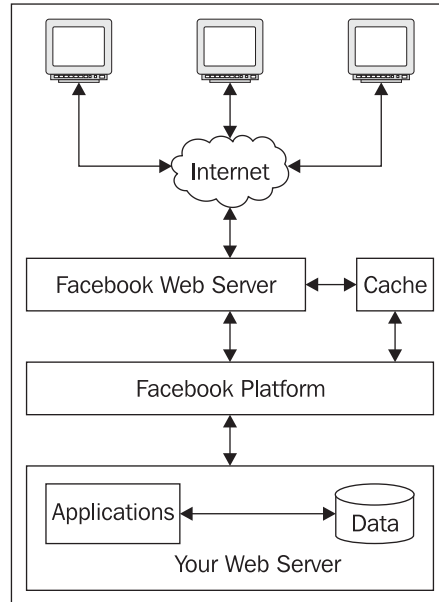
As far as your application users are concerned, they will just access `Facebook.com` and your application, by using a web browser and the Internet. But, that's not where the application lives — *it's actually on your own server*:



Once you've looked at the Facebook web application model and realized that your application actually resides on your own server, it becomes obvious what the purpose of the Facebook Platform is — to provide an interface between your application and itself.

There is an important matter to be considered here. If the application actually resides on your server, and your application becomes very successful (according to Facebook there are currently 25 million active users), then will your server be able to cope with that number of hits?

Don't be *too* alarmed. This doesn't mean that your server will be accessed every time someone looks at his or her profile. Facebook employs a cache to stop that happening:



Of course, at this stage, you're probably more concerned with just getting the application working – so let's continue looking at the Platform, but just bear that point in mind.

The Facebook Platform Elements

There are three elements to the Facebook Platform:

- The Facebook API (Application Programming Interface)
- FBML – Facebook Markup Language
- FQL – Facebook Query Language

We'll now spend some time with each of these elements, and you'll see how you can use them individually, and in conjunction to make powerful yet simple applications. The great thing is that if you haven't got your web server set up yet, don't worry, because Facebook supplies you with all of the tools that you would need in order to do a test run with each of the elements.

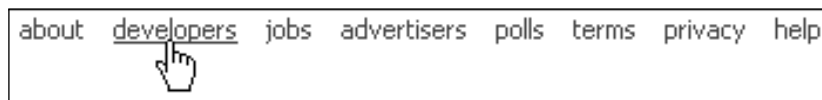
The Facebook API

If you've already done some programming, then you'll probably know what an API (or Application Programming Interface) is. *It's a set of software libraries that enable you to work with an application (in this case, Facebook) without knowing anything about its internal workings.* All you have to do is obtain the libraries, and start making use of them in your own application.

Now, before you start downloading files, you can actually learn more about their functionality by making use of the Facebook API Test Console.

The Facebook API Test Console

If you want to make use of the Facebook Test Console, you'll first need to access the Facebook developers' section—you'll find a link to this at the bottom of every Facebook page:



Alternatively, you can use the URL `http://developers.facebook.com` to go there directly in your browser. When you get to this page, you'll find a link to the **Tools** page:



Or, again, you can go directly to <http://developers.facebook.com/tools.php>, where you'll find the API Test Console:

The screenshot shows the Facebook API Test Console interface. At the top, there's a 'TOOLS' header with two tabs: 'API Test Console' (selected) and 'FBML Test Console'. Below the tabs, the title 'API Test Console' is displayed. A descriptive paragraph states: 'You can experiment with functions and responses, and see what content Facebook Platform makes available. Select the method you wish want to call and the format of the return values.' Below this, a bold message says: 'You need to be logged in to use the console. Log in here.' The form contains several fields: 'User ID' with a text box showing 'Not logged in'; 'Response Format' with a dropdown menu set to 'XML'; 'Callback' with an empty text box; 'Method' with a dropdown menu set to 'friends.get'; and a 'Call Method' button. To the right of the form, a large gray area is labeled 'The results of your method call will show up here.'

You'll find that the API Test Console has a number of fields:

- **User ID**— A read-only field which (when you're logged on to Facebook) unsurprisingly displays your user ID number.
- **Response Format**— With this, you can select the type of response that you want, and this can be:
 - XML
 - JSON
 - Facebook PHP Client
- **Callback**— If you are using XML or JSON, then you can encapsulate the response in a function.
- **Method**— The actual Facebook method that you want to test.

Once you've logged in, you'll see that your User ID is displayed, and that all the drop-downs are enabled:

The screenshot shows the 'API Test Console' interface. At the top, there's a 'TOOLS' header with two tabs: 'API Test Console' (active) and 'FBML Test Console'. Below the tabs, the title 'API Test Console' is displayed. A descriptive text says: 'You can experiment with functions and responses, and see what content Facebook Platform makes available. Select the method you wish want to call and the format of the return values.' The form includes a 'User ID' field with the value '614902533', a 'Response Format' dropdown menu set to 'XML', an empty 'Callback' text field, and a 'Method (documentation)' dropdown menu set to 'friends.get'. A 'Call Method' button is located below the method dropdown. To the right of the form, a large light gray area contains the text: 'The results of your method call will show up here.'

You will also notice that a new link, **documentation**, appears on the screen, which is very useful. All you have to do is to select a method from the drop-down list, and then click on **documentation**. Once you've done that you'll see:

- A description of the method
- The parameters used by the method
- An example return XML
- A description of the expected response.
- The FQL equivalent (we will discuss this later in the chapter.)
- Error codes

For now, let's just change the **Response Format** to **Facebook PHP Client**, and then click on **Call Method** to see what happens:

The screenshot shows the 'API Test Console' interface. On the left, there are input fields for 'User ID' (614902533), 'Response Format' (Facebook PHP Client), 'Callback' (empty), and 'Method (documentation)' (friends.get). A 'Call Method' button is at the bottom. On the right, the 'Array' response is displayed as a JSON array of 11 user IDs.

```
Array
(
    [0] => 286601116
    [1] => 286603790
    [2] => 500048694
    [3] => 503027213
    [4] => 505543152
    [5] => 511832017
    [6] => 515482742
    [7] => 517311956
    [8] => 521240370
    [9] => 690946142
    [10] => 705175505
)
```

In this case, you can see that the method returns an array of user ids — each one being the ID of one of the friends of the currently logged in user (that is your list of friends because you're the person logged in). You could, of course, go on to use this array in PHP as part of your application, but don't worry about that at the moment. We'll see how to use the methods in a real application, in Chapter 2. For the time being, we'll just concentrate on working with our prototyping in the test console.

However, before we move on, it's worth noting that you can obtain an array of friends only for the currently logged in user. You can't obtain the list of friends for any other user. So, for example, you would not be able to use `friends.get` on id 286601116 or 705175505. In fact, you wouldn't be able to use `friends.get` for 614902533 (as shown in the example) because that's my ID and not yours.

On the other hand, having obtained a list of valid IDs we can now do something more interesting with them. For example, we can use the `users.getInfo` method to obtain the first name and birthday for particular users:

User ID <input type="text" value="614902533"/> Response Format <input type="text" value="Facebook PHP Client"/> Callback <input type="text"/> Method (documentation) <input type="text" value="users.getInfo"/> uids <input type="text" value="614902533,517311956,521240370"/> fields <input type="text" value="first_name,birthday"/> <input type="button" value="Call Method"/>	<pre> Array ([0] => Array ([uid] => 614902533 [first_name] => Mark [birthday] => January 27, 1960) [1] => Array ([uid] => 517311956 [first_name] => David [birthday] => November 3, 1978) [2] => Array ([uid] => 521240370 [first_name] => Tripti [birthday] => June 28, 1977)) </pre>
--	--

As you can see, a multidimensional array is returned to your PHP code (if you were actually using this in an application). Therefore, for example, if you were to load the array into a variable `$birthdays`, then `$birthdays[0][birthday]` would contain January 27, 1960.

Of course, in the above example, the most important piece of information is the first birthday in the array – record that in your diary for future reference. And, if you're thinking that I'm old enough to be your father, well, in some cases this is actually true:

TOOLS

API Test ConsoleFBML Test Console

API Test Console

You can experiment with functions and responses, and see what content Facebook Platform makes available. Select the method you wish want to call and the format of the return values.

User ID

Response Format

Facebook PHP Client

Callback

Method (documentation)

users.getInfo

uids

fields

Call Method

```
Array
(
    [0] => Array
        (
            [uid] => 505543152
            [first_name] => Michael
            [last_name] => Bain
            [birthday] => September 12, 1986
        )
    [1] => Array
        (
            [uid] => 286603790
            [first_name] => Simon
            [last_name] => Bain
            [birthday] =>
        )
)
```

Now that you've come to grips with the API Test console, we can turn our attention to FBML and the FBML Test Console.

[14]

For More Information:

www.packtpub.com/learning-facebook-application-development/book

FBML

FBML, as mentioned previously, is the Facebook Markup Language. If you've used HTML in web pages, then you'll have no problem with FBML. Even if you've not used HTML ever, you'll still have no problems with FBML. It can be used to format your information just like HTML, but it also does a lot more than just formatting as we'll soon see, by looking at the FBML Test Console.

The FBML Test Console

We've already seen that we can learn how to use the Facebook API methods in the API Test Console, and we can do the same with FBML in the FBML Test Console. Just go to the **Tools** page as you did for the API Test Console, but this time click on the **FBML Test Console** tab:

The screenshot shows the 'FBML Test Console' interface. At the top, there's a 'TOOLS' header with two tabs: 'API Test Console' and 'FBML Test Console'. The 'FBML Test Console' tab is selected. Below the tabs, the main area is titled 'FBML Test Console'. It features a large text input field for FBML code. To the right of the input field, there's a 'Preview' section with a 'show outline' checkbox (checked) and a 'Facebook FBML Test Console Sample App' button. Below the preview, there's a 'Profile Actions' section with the text 'No profile actions'. At the bottom right, there's an 'HTML Source' section. At the bottom left, there's an 'Errors' section. The 'user' and 'profile' fields are both set to '314902533'. The 'position' is set to 'wide'. The 'API Key' is '2b84359cad5a9ab45bb801a22ae0ef63'.

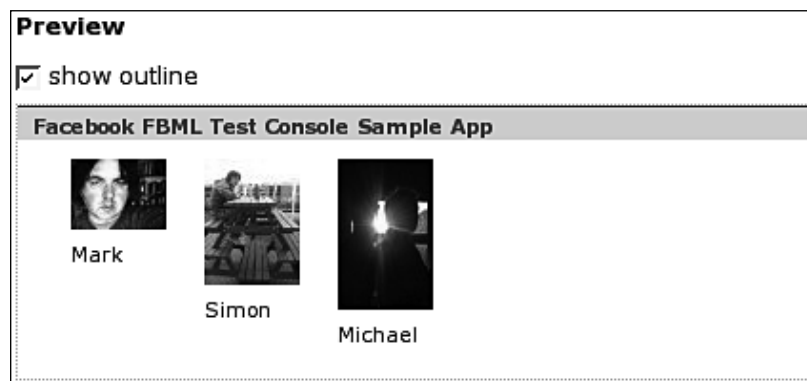
In this console, we're mainly interested in two areas:

- The code entry box — this is the large box on the left-hand-side of the console
- The preview area on the top-right of the console

Next, we need some code to test:

```
<table cellpadding=0 cellspacing=0><tr><td>
  <table cellpadding=0 cellspacing=0>
    <tr><td>
      <fb:profile-pic uid=614902533></fb:profile-pic>
    </td></tr>
    <tr><td>
      <fb:name useyou=false uid=614902533 firstnameonly=true></fb:name>
    </td></tr></table>
  </td><td>
    <table cellpadding=0 cellspacing=0><tr><td>
      <fb:profile-pic uid=286603790></fb:profile-pic>
    </td></tr>
    <tr><td>
      <fb:name useyou=false uid=286603790 firstnameonly=true></fb:name>
    </td></tr></table>
  </td><td>
    <table cellpadding=0 cellspacing=0><tr><td>
      <fb:profile-pic uid=505543152></fb:profile-pic>
    </td></tr>
    <tr><td>
      <fb:name useyou=false uid=505543152 firstnameonly=true></fb:name>
    </td></tr></table>
  </td></tr></table>
```

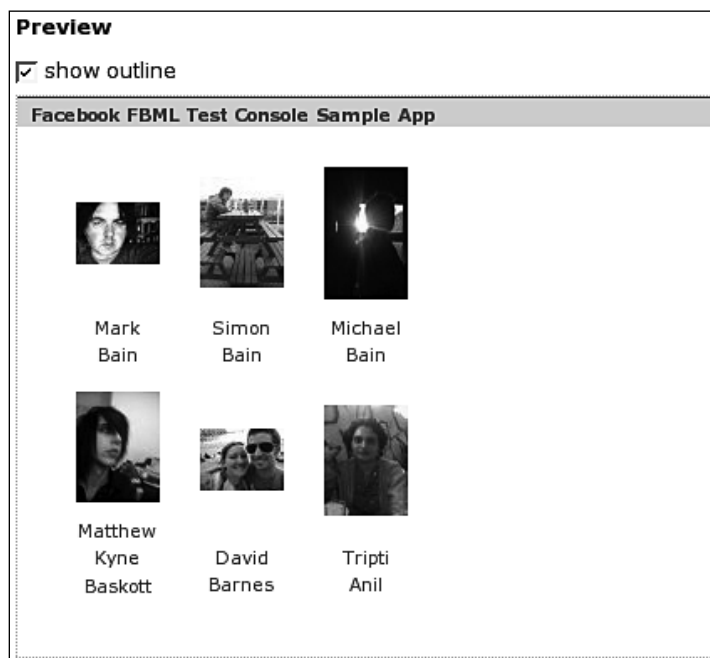
You'll notice from the code that we can mix HTML (we've used `<table>` `</table>`, `<tr>` `</tr>` and `<td>` `</td>` to tabulate the information) with FBML (`<fb:name>` `</fb:name>` and `<fb:profile-pic>` `</fb:profile-pic>` to extract and display the Facebook data). If you put this into the code entry box and click **Preview**, then you'll see how your application would look:



Having seen how we can achieve an output using a mixture of HTML and FBML, it would be interesting to see how you can get much the same results even more easily, by using only FBML:

```
<fb:user-table cols="3">
  <fb:user-item uid="614902533"/>
  <fb:user-item uid="286603790"/>
  <fb:user-item uid="505543152"/>
  <fb:user-item uid="515482742"/>
  <fb:user-item uid="517311956"/>
  <fb:user-item uid="521240370"/>
</fb:user-table>
```

You can see from the code that we're able to use the FBML `<fb:user-table>` tag to create a table, and we've used the `<fb:user-item>` tag to obtain the information that we want from Facebook. The resulting application would look like this:



We've looked at the Facebook API and FBML, and learned how useful they can be. Now, we'll move on to the potentially most powerful element of the Facebook platform—FQL.

FQL

It may interest you to know that the information that you can obtain from Facebook is actually stored in a database, and that's where FQL comes in. So, instead of using the API methods, we can query the tables in the database to obtain the information that we require. For example, if you want the first name, surname, and birthday of each of your friends, then you can use the following query in the API Test Console:

```
SELECT first_name, last_name,birthday
FROM user
WHERE uid IN (  SELECT uid1
                FROM friend
                WHERE uid2=614902533)
```

And don't forget to replace my id (614902533) with your own because you can only use the details of friends of the logged in user, that is you.

Now, we still need to use the API, but for this, we only need one method – `fql.query`:

The screenshot shows the Facebook API Test Console interface. On the left, there are input fields for 'User ID' (614902533), 'Response Format' (Facebook PHP Client), 'Callback' (empty), 'Method (documentation)' (fql.query), and a 'query' text area containing the FQL query. A 'Call Method' button is at the bottom. On the right, the 'Array' response is displayed, showing three friend entries with their first_name, last_name, and birthday.

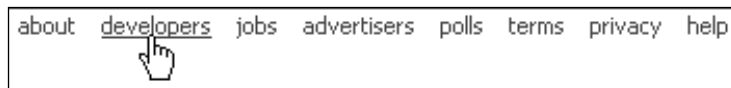
Index	first_name	last_name	birthday
[0]	Tim	Charlton	June 24, 1987
[1]	Simon	Bain	
[2]	Jon	Phillips	May 28, 1979

You'll realize of course, that this means that you're not limited to the information supplied by the API methods and the FBML tags. You're actually able to extract the exact data that you want.

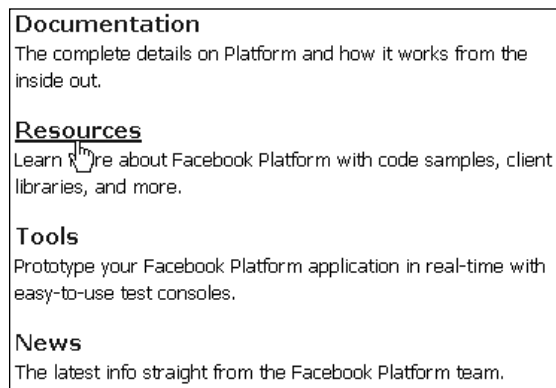
Now that you're conversant with the Facebook Platform, you will want to create your first Facebook application. However, before you do that, you'll need to obtain the Facebook Client Libraries.

Obtaining the Facebook Client Libraries

If you're going to build a Facebook application, then you're going to need access to the Facebook API, and to do that, you'll have to download the client libraries onto your server. In order to do this, you'll need to go to the developers' page—remember that you can use the URL `http://developers.facebook.com`, or use the link given at the bottom of each Facebook page:




Once you're there, find the link to the **Resources** page:




Or, go directly to `http://developers.facebook.com/resources.php`.




When you're in the **Resources** page, you can click on the library that you want to use. In this instance, we're going to be using the **PHP (4 and 5) Client Library**:




Download a Client Library



 **PHP (4 and 5) Client Library** | Description of contents
Updated June 12, 2007.

 **Java Client Library** | Description of contents
Updated May 26, 2007.

Unofficial Client Libraries | Disclaimer


 Actionscript  Cocoa  ColdFusion

 Perl  PHP4  Python

 Ruby  VB.NET

Of course, if you want to know more about a library, then just click on **Description of contents**:

Download a Client Library

 **PHP (4 and 5) Client Library** | [Description of contents](#)
Updated June 12, 2007.

The package contains two directories and a README file.

The **client** directory contains these core components:

facebookapi_php5_restlib.php contains the implementation of the methods that constitute the client library.

facebook.php contains a class that can be used by the web client application.

facebook_desktop.php contains a class that can be used by the desktop client application.

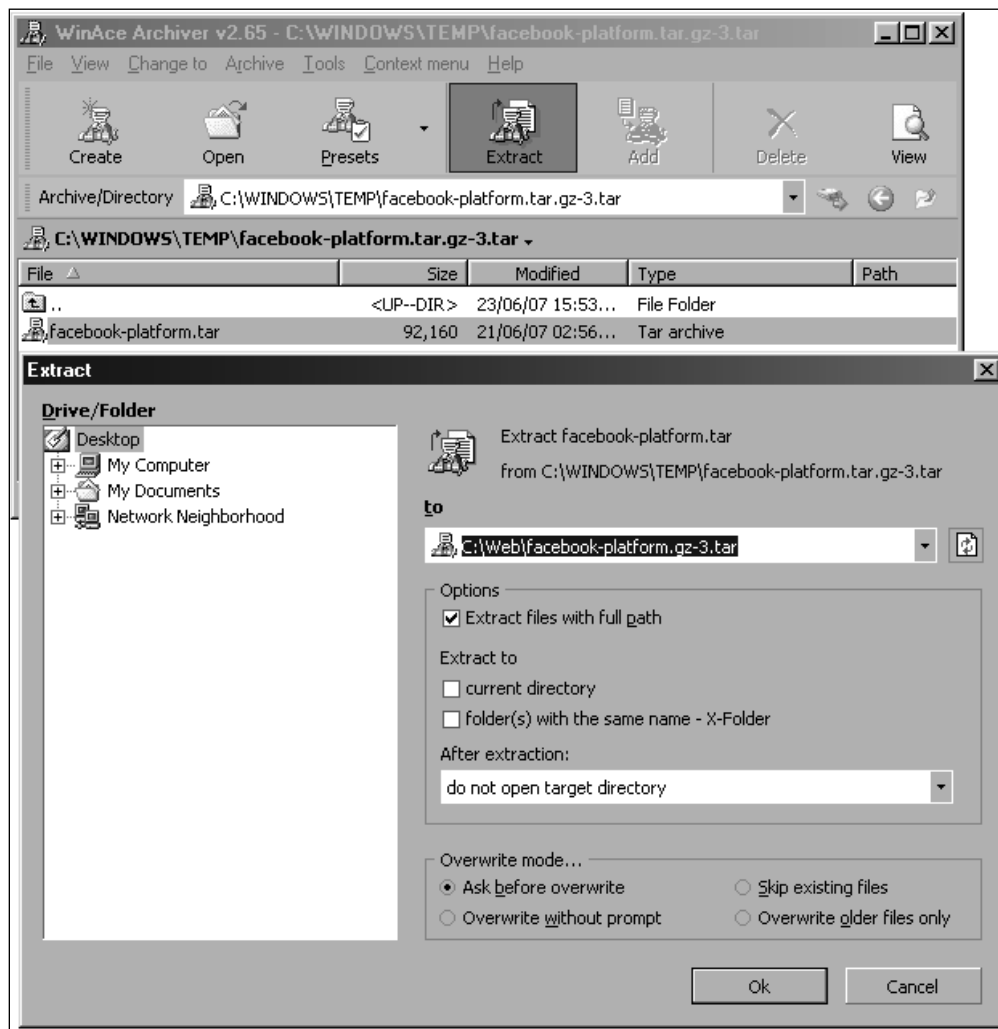
The **php4client** directory contains similar components for PHP4.

The **footprints** directory contains code for a sample application called "footprints".

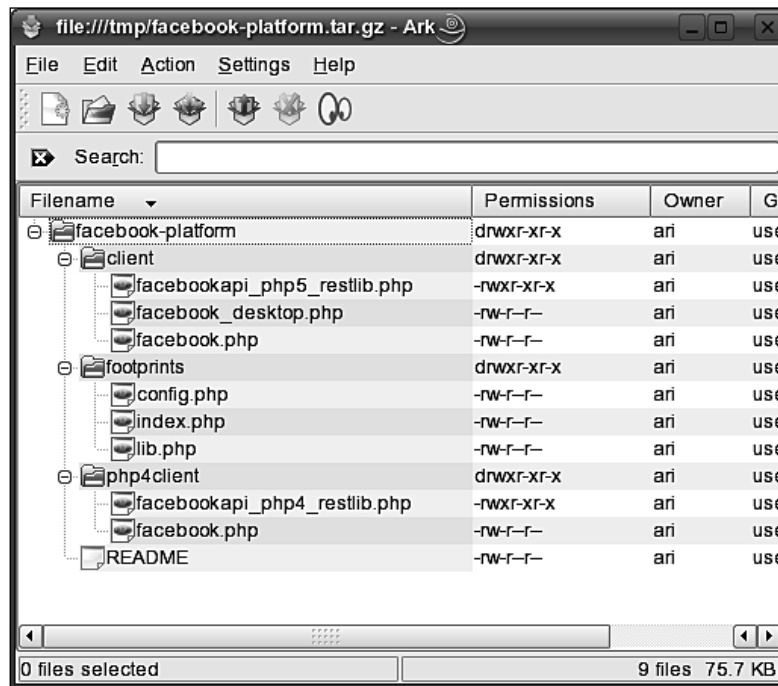
See the **README** file for additional information.

So far, everything we've looked at will be the same whether you are using Windows or Linux on your server, and that's true for most of what we'll be doing. However, this is one instance when what you do will depend on your server's operating system.

The file that you're going to download is compressed, but not in a form normally used in Windows (you'll notice that they have a `tar.gz` suffix). So, you'll need to obtain software that can uncompress the files for you. A quick Google search will find a lot of them, but a suitable one is available from <http://www.winace.com>, although you'll find that it does display some annoying pop-ups, so you may want to delete it once you're finished. However, once you've installed WinAce, you'll be able to uncompress the libraries:



If, on the other hand, you're using Linux, then you'll be able to uncompress the files quite readily. If you prefer to use a GUI type interface, then there is a software, which you can use, called Ark:



However, if you prefer to use the command line (as I do), then you can both download and uncompress the files very easily:

```
cd /srv/www
wget http://developers.facebook.com/clientlibs/facebook-platform.tar.
gz
tar -xvzf facebook-platform.tar.gz
```

Of course, you'll need to change `/srv/www` to a suitable directory on your own server, one that you can write to. And in case you're not used to the commands, `cd` moves you to the directory, `wget` downloads the library, and `tar` uncompresses it. If you type the commands on the command line, then you'll see something like this:



```
bainm@aeneas:/srv/www - Shell - Konsole
Session Edit View Bookmarks Settings Help

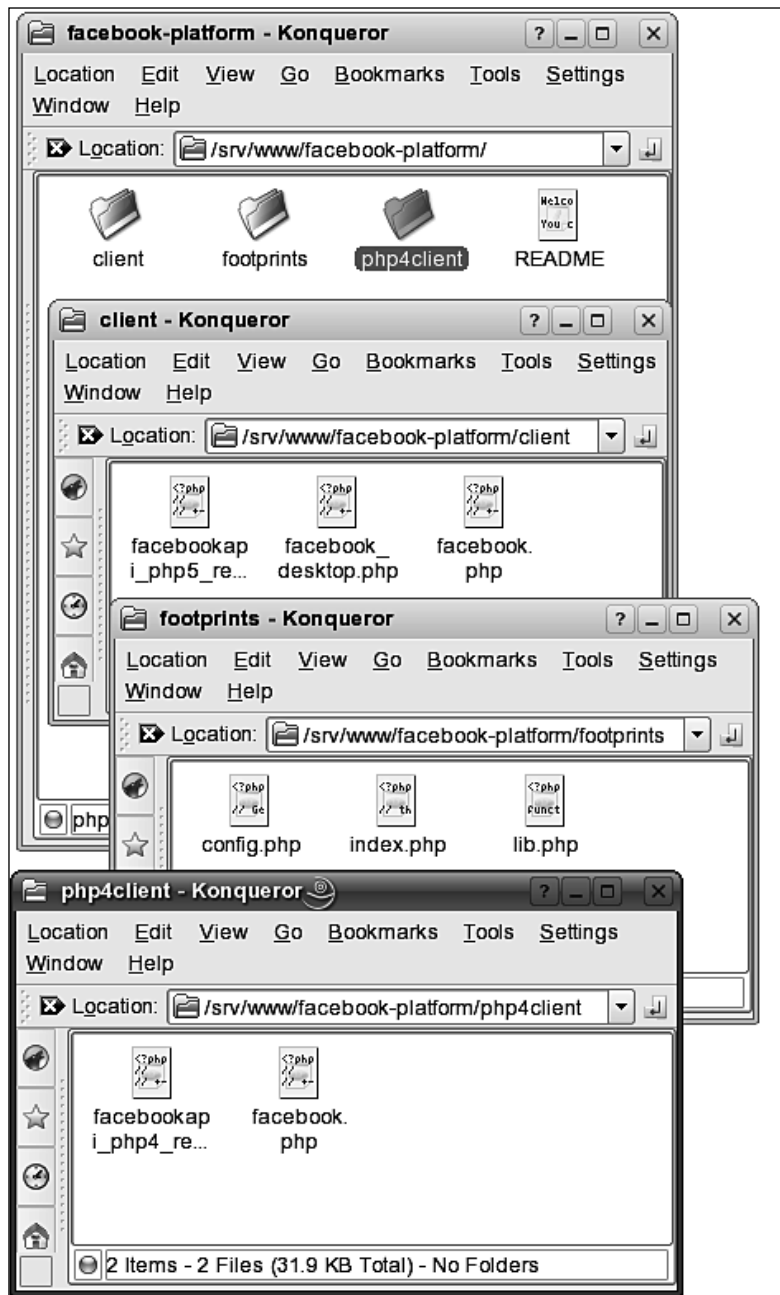
bainm@aeneas:~> cd /srv/www
bainm@aeneas:/srv/www> wget http://developers.facebook.com/clientlibs/facebook-platform.tar.gz
--19:12:53-- http://developers.facebook.com/clientlibs/facebook-platform.tar.gz
=> `facebook-platform.tar.gz.3'
Resolving developers.facebook.com... 204.15.20.44
Connecting to developers.facebook.com|204.15.20.44|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 19,359 (19K) [application/x-tar]

100%[=====] 19,359      43.40K/s

19:12:54 (43.34 KB/s) - `facebook-platform.tar.gz.3'
saved [19359/19359]

bainm@aeneas:/srv/www> tar -xvzf facebook-platform.tar.gz
facebook-platform/
facebook-platform/client/
facebook-platform/client/facebook.php
facebook-platform/client/facebook_desktop.php
facebook-platform/client/facebookapi_php5_restlib.php
facebook-platform/php4client/
facebook-platform/php4client/facebook.php
facebook-platform/php4client/facebookapi_php4_restlib.php
facebook-platform/footprints/
facebook-platform/footprints/config.php
facebook-platform/footprints/index.php
facebook-platform/footprints/lib.php
facebook-platform/README
bainm@aeneas:/srv/www>
```

At the moment, it doesn't actually matter where you place the libraries, but regardless of whether you're using Windows or Linux, you should end up with a set of PHP files somewhere on your server:



And now, you are ready to move on to telling Facebook about the application that you're going to be creating.

So far, we've spent a bit of time looking at the Facebook Platform—you have learned where it fits in with any application that you build, and you have learned how to experiment with the Platform's functionality, without even having an application. In Chapter 2, you'll learn how to create that first application.

However, before you create your own application, you're going to have to add another application—Facebook's Developer.

Adding the Developer Application

You may be wondering why you need to add another application in order to create your own. Quite simply—you don't use developer to create your application—you use the developer application to *add* yours into Facebook.

So, start by going to the developer's page (remember that you'll find a link at the bottom of every Facebook page, or you can go directly to <http://www.facebook.com/developers>). Next, you'll have to click on **Get Started**, and then **Add Facebook Developer Application**:

Add Developer to your Facebook account?



Allow this application to...

☒ Know who I am and access my information

☒ Put a box in my profile

☒ Place a link in my left-hand navigation

☒ Publish stories in my News Feed and Mini-Feed

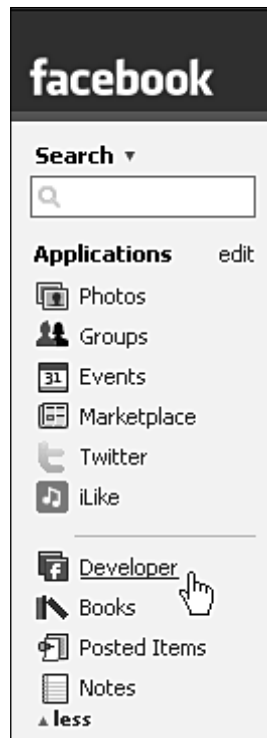
☒ Place a link below the profile picture on any profile.

Developer was created by Facebook and is pretty sweet. By clicking 'add', you agree to the Platform Application Terms of Use.

➔ Add Developer or Cancel

You can change the settings if you want, but you might as well keep the default settings, and press **Add Developer**.

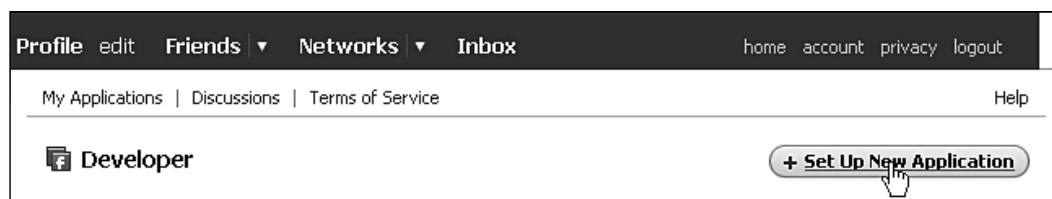
Once you've added the Developer application, you'll be able to access it from your left-hand side navigation:



At this point, you don't actually need an application. What we're going to do is set up our new application in Facebook, and *then* create it.

Setting Up Your Application


If you click on **Developer** in your left-hand side navigation pane, you'll see a button to **Set Up New Application**:



You'll now find that you need to type in details about your application (obviously!). Some fields are required, while others are optional. We'll deal with the required fields first.

Required Fields

Don't worry, you won't have masses of fields to fill in before you can get started with your application—in fact there is only one required field, plus a tick box:

Facebook Developer	Back to Developer Home
 New Application	
Required Fields	
Application Name (Limit: 50 characters)	<input type="text"/>
<input type="checkbox"/> Check here to indicate that you have read and agree to the terms of the Facebook Platform.	
► Optional Fields	
<input type="button" value="Submit"/> or <input type="button" value="Cancel"/>	

So, all you have to do, at this stage, is choose a suitable name for your application. As this will be displayed on Facebook, decide on a name that describes your application best (avoiding anything rude or insulting). In this example, I'm developing an application for an imaginary private detective, Pygoscelis P. Ellsworthy, so I'm going to name the application Pygoscelis P. Ellsworthy's Suspect Tracker:

Required Fields	
Application Name (Limit: 50 characters)	<input type="text" value="Pygoscelis P. Ellsworthy's Suspect Tracker"/>
<input checked="" type="checkbox"/> Check here to indicate that you have read and agree to the terms of the Facebook Platform.	

You'll need to tick the box to say that you've read the Facebook Platform terms and conditions (and we *always* read those, don't we?). However, don't click on **Submit** yet, as the remainder of the fields are defined as optional, although some are still essential (as we're about to see).

Optional Fields

By default, the optional fields are hidden, so to see them, click on **Optional Fields** (no surprise there then).

Base Options

As soon as you click on **Optional Fields**, the **Base options** will become visible – these are the options required by any application that you build, although others will be made available to you depending on what you enter into the **Base options**.

▼ Optional Fields

Base options

Support E-mail
(Limit: 100 characters) We will contact you at this address if there are any problems or important updates.

Callback Url
(Limit: 100 characters) After logging into Facebook, users are redirected to the callback URL. See authentication overview for more details.

Canvas Page URL /
☒ Use FBML ☒ Use iframe
Your application will be viewable in the Facebook navigation at this URL - either as rendered FBML or loaded in an iframe.

Application Type ☒ Website ☐ Desktop

IP Addresses of Servers Making Requests
(comma-separated)
If you supply this information (e.g. 10.1.20.1, 101.20.3), requests from addresses other than those listed will be rejected.

Can your application be added on Facebook? ☒ Yes ☐ No
Select Yes if your application can be added to a user's Facebook account.

TOS URL
The URL pointing to your application's Terms of Service, which the user must accept.

or

You'll see that **Support E-Mail** has already been filled in for you, using your Facebook email address. Obviously, you'll need to change that, if that's not the address that you want to use. Apart from **Support E-Mail**, there are only two fields that we're interested in at the moment:

- **Callback URL** – The URL of your application (that is on your server).
- **Canvas Page URL** – The URL for your application within Facebook (obviously this needs to be unique).

So, in the case of Pygoscelis P. Ellsworth's Suspect Tracker, the optional fields would look like this:

▼ Optional Fields

Base options

Support E-mail

mark.bain@linuxtalk.co.uk

(Limit: 100 characters)

We will contact you at this address if there are any problems or important updates.

Callback Url

http://213.123.183.16/f8/penguin_pi/

(Limit: 100 characters)

After logging into Facebook, users are redirected to the callback URL. See authentication overview for more details.

Canvas Page URL

http://apps.facebook.com/penguin_pi/

☒ Use FBML
☒ Use iframe

Your application will be viewable in the Facebook navigation at this URL - either as rendered FBML or loaded in an iframe.

Application Type

☒ Website
☐ Desktop

IP Addresses of Servers Making Requests

(comma-separated)

If you supply this information (e.g. 10.1.20.1, 101.20.3), requests from addresses other than those listed will be rejected.

Can your application be added on Facebook?

☒ Yes
☐ No

Select Yes if your application can be added to a user's Facebook account.

TOS URL

The URL pointing to your application's Terms of Service, which the user must accept.

And a big tip here – make sure that you end the **Callback URL** with a / (in your code, NOT in the Installation options box), and you'll save yourself many hours of heartache wondering why your application isn't working correctly.

You can ignore all the other fields in this section at the moment, apart from one question that you must answer, **Can your application be added on Facebook?** Obviously, the answer will be **Yes**. But don't worry about your application being accessed by thousands of people before you're ready – in a moment we'll see how to stop that.

When you set **Can your application be added on Facebook?** to **Yes**, you'll find that the **Installation Options** are displayed.

Installation Options

Since we want our application to be added on to Facebook, we'll need to fill in some **Installation Options**. As with the **Base options**, we can leave most of them blank, just filling in a few essential ones:

Installation Options	
Post-Add URL (Limit: 100 characters)	<input type="text" value="http://apps.facebook.com/penguin_pi/"/> <small>The URL to redirect to after a user adds your application.</small>
Application Description (Limit: 250 characters)	<div><div></div><div>A short description of your application, displayed on your "add application" page</div></div>
Post-Remove URL (Limit 100 characters)	<input type="text"/> <small>URL at which you want to be notified when a user removes your application from their Facebook account. Cannot be a Facebook-framed page.</small>
Default FBML	<div><div>Welcome to Pygoscelis P. Ellsworth's Suspect Tracker</div><div>This FBML will be rendered in a user's profile if setFBML hasn't been called for that user.</div></div>
Default Profile Box Column	<div><div><input checked="" type="radio"/> Wide <input type="radio"/> Narrow</div><div>The column in which your application's profile box should be displayed when first added by a user.</div></div>
Developer Mode	<input checked="" type="checkbox"/> Check this box to only allow developers of the application to install the application.

As you can see, the **Post-Add URL** needs to be added (this should be the same as your **Canvas Page URL**), and you can enter some arbitrary text into the **Default FBML** box.

Now, I did mention a little while ago that you can prevent your application being used before you're ready, and this is the section where you do that. Just tick the **Developer Mode** box.

Integration Points

The only field that you need to fill in at this point is the **Side Nav URL** (and this should be set to the same as your application's **Canvas Page URL**, in the base options):

Integration Points

Side Nav URL
(Limit: 100 characters)

URL for your app if you want a link in the side nav. Must be a Facebook canvas page.

Edit URL
(Limit: 100 characters)

Optional "edit" link displayed on your application's profile box for a user who has added your application and is looking at their own profile.

Privacy URL
(Limit: 100 characters)

Link to a privacy configuration page for your application.

Help URL
(Limit: 100 characters)

Link to a help page for your application.

Private Installation

☐ **Check this box to disable News Feed and Mini-Feed installation stories for your application.**

Message Attachment

Attachment action:

The action in the attachment dropdown menu for your application. (Limit: 20 characters)

Callback URL:

The URL from which you can fetch content for message attachments. (Limit: 100 characters)

All you have to do now is click **Submit**, and Facebook will create an API key for you (you'll need that and the secret key when you write the code for your application):

API key created.

You have 1 key | [Apply for another key](#)

☐ **Pygoscells P. Ellsworthy's Suspect Tracker** | [Edit Settings](#) | [Delete App](#)

About Page	View About Page Edit About Page	Submit Application »
API Key	322d68147c78d2621079317b778cfe10	Once you have completed your application you may submit it to our product directory.
Secret	0a53919566eeb272d7b96a76369ed90c	
Support Email	mark.bain@linuxtalk.co.uk	
Callback URL	http://213.123.183.16/f8/penguin_pi	

And with that done, we can move on to actually building the application itself.

Summary

In this chapter, we've looked at the Facebook Platform and learned that it consists of three elements that must be used by your application to interface with Facebook:

- The Facebook API—A library of PHP methods that will enable your application to extract data from Facebook.
- FBML—The Facebook Markup Language—used to obtain and format Facebook data.
- FQL—The Facebook Query Language—used to extract information directly from the Facebook database.

Each of the platform elements can be used in the development of your application, but you don't need to develop an application in order to test them out. For that you can make use of:

- The API Test Console
- The FBML Test Console

Before you can start developing your own application, you will need to download the Facebook Client Library, and if you're using Windows, then you'll need to obtain suitable software for uncompressing it.

You have to add the Facebook Developer application. Once you've done that, you can start adding your own applications.

When setting up your application, you must give it a sensible name, and a suitable location (the Callback URL). The Callback URL must be to your own server, and don't forget to add a / at the end of the URL.

When you submit your application, you'll be given an API key, and a secret key. You'll need to use these in your application. Now, you are ready to create your application.

Where to buy this book

You can buy Learning Facebook Application Development from the Packt Publishing website: <http://www.packtpub.com/learning-facebook-application-development/book>

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our [shipping policy](#).

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.



www.PacktPub.com

For More Information:

www.packtpub.com/learning-facebook-application-development/book