

```

Example of save webpage URL in Image file;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Drawing;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            string url = "http://4rapiddev.com/facebook-graph-api/php-load-facebook-albums-and-save-to-mysql-database/";

            Bitmap thumbnail = SaveWebPage2Image(url);

            //specify where image will be save
            thumbnail.Save("d:/web-shot.png", System.Drawing.Imaging.ImageFormat.Png);
        }

        public Bitmap SaveWebPage2Image(string url)
        {
            // Load the webpage into a WebBrowser control
            WebBrowser wb = new WebBrowser();
            wb.ScrollBarsEnabled = false;
            wb.ScriptErrorsSuppressed = true;
            wb.Navigate(url);

            while (wb.ReadyState != WebBrowserReadyState.Complete) { System.Windows.Forms.Application.DoEvents(); }

            // Take Screenshot of the web pages full width
            wb.Width = wb.Document.Body.ScrollRectangle.Width;

            // Take Screenshot of the web pages full height
            wb.Height = wb.Document.Body.ScrollRectangle.Height;

            // Get a Bitmap representation of the webpage as it's rendered in the WebBrowser control
            Bitmap bitmap = new Bitmap(wb.Width, wb.Height);
            wb.DrawToBitmap(bitmap, new Rectangle(0, 0, wb.Width, wb.Height));
            wb.Dispose();

            return bitmap;
        }
    }
}

```

////Example 2

```

private void button1_Click(object sender, EventArgs e)
{
    Form1 f11 = new Form1();
    Image im = new
    Bitmap("C://Users//Mahdi//Documents//BTt.bmp",true);

    f11.button1.BackgroundImage = im;// "~\mnj-
    img.jpg";//'C://Users//Mahdi//Pictures//mnj-img.jpg';
    f11.Activate();
    f11.Show();
}

```

//usefull information

What Are Cookies?

Cookies are small files which are stored on a user's computer. They are designed to hold a modest amount of data specific to a particular client and website, and can be accessed either by the web server or the client computer. This allows the server to deliver a page tailored to a particular user, or the page itself can contain some script which is aware of the data in the cookie and so is able to carry information from one visit to the website (or related site) to the next.

Are Cookies Enabled in my Browser?

To check whether your browser is configured to allow cookies, visit the [Cookie checker](#). This page will attempt to create a cookie and report on whether or not it succeeded.

For information on how to enable or disable cookies, see '[Enabling cookies](#)'.

For information on how to delete cookies, see '[Deleting cookies](#)'.

What's in a Cookie?

Each cookie is effectively a small lookup table containing pairs of (key, data) values - for example (firstname, John) (lastname, Smith). Once the cookie has been read by the code on the server or client computer, the data can be retrieved and used to customise the web page appropriately.

When are Cookies Created?

Writing data to a cookie is usually done when a new webpage is loaded - for example after a 'submit' button is pressed the data handling page would be responsible for storing the values in a cookie. If the user has elected to disable cookies then the write operation will fail, and subsequent sites which rely on the cookie will either have to take a default action, or prompt the user to re-enter the information that would have been stored in the cookie.

Why are Cookies Used?

Cookies are a convenient way to carry information from one session on a website to another, or between sessions on related websites, without having to burden a server machine with massive amounts of data storage. Storing the data on the

server without using cookies would also be problematic because it would be difficult to retrieve a particular user's information without requiring a login on each visit to the website.

If there is a large amount of information to store, then a cookie can simply be used as a means to identify a given user so that further related information can be looked up on a server-side database. For example the first time a user visits a site they may choose a username which is stored in the cookie, and then provide data such as password, name, address, preferred font size, page layout, etc. - this information would all be stored on the database using the username as a key. Subsequently when the site is revisited the server will read the cookie to find the username, and then retrieve all the user's information from the database without it having to be re-entered.

How Long Does a Cookie Last?

The time of expiry of a cookie can be set when the cookie is created. By default the cookie is destroyed when the current browser window is closed, but it can be made to persist for an arbitrary length of time after that.

Who Can Access Cookies?

When a cookie is created it is possible to control its visibility by setting its 'root domain'. It will then be accessible to any URL belonging to that root. For example the root could be set to "whatarecookies.com" and the cookie would then be available to sites in "www.whatarecookies.com" or "xyz.whatarecookies.com" or "whatarecookies.com". This might be used to allow related pages to 'communicate' with each other. It is not possible to set the root domain to 'top level' domains such as '.com' or '.co.uk' since this would allow widespread access to the cookie.

By default cookies are visible to all paths in their domains, but at the time of creation they can be restricted to a given subpath - for example "www.whatarecookies.com/images".

How Secure are Cookies?

There is a lot of concern about privacy and security on the internet. Cookies do not in themselves present a threat to privacy, since they can only be used to store information that the user has volunteered or that the web server already has. Whilst it is possible that this information could be made available to specific third party websites, this is no worse than storing it in a central database. If you are concerned

that the information you provide to a webserver will not be treated as confidential then you should question whether you actually need to provide that information at all.

What are Tracking Cookies?

Some commercial websites include embedded advertising material which is served from a third-party site, and it is possible for such adverts to store a cookie for that third-party site, containing information fed to it from the containing site - such information might include the name of the site, particular products being viewed, pages visited, etc. When the user later visits another site containing a similar embedded advert from the same third-party site, the advertiser will be able to read the cookie and use it to determine some information about the user's browsing history. This enables publishers to serve adverts targetted at a user's interests, so in theory having a greater chance of being relevant to the user. However, many people see such 'tracking cookies' as an invasion of privacy since they allow an advertiser to build up profiles of users without their consent or knowledge.

HTTP COOKIES

Cookies are key/value pairs used by websites to store state informations on the browser. Say you have a website (example.com), when the browser requests a webpage the website can send cookies to store informations on the browser.

Browser request example:

```
GET /index.html HTTP/1.1
Host: www.example.com
```

Example answer from the server:

```
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: foo=10
Set-Cookie: bar=20; Expires=Fri, 30 Sep 2011 11:48:00 GMT
... rest of the response
```

Here two cookies foo=10 and bar=20 are stored on the browser. The second one will expire on 30 September. In each subsequent request the browser will send the cookies back to the server.

```
GET /spec.html HTTP/1.1
Host: www.example.com
Cookie: foo=10; bar=20
Accept: */*
```

SESSIONS: Server side cookies

Server side cookies are known as "sessions". The website in this case stores a single cookie on the browser containing a unique Session Identifier. Status information (foo=10 and bar=20 above) are stored on the server and the Session Identifier is used to match the request with the data stored on the server.

Examples of usage

You can use both sessions and cookies to store: authentication data, user preferences, the content of a chart in an e-commerce website, etc...

Pros and Cons

Below pros and cons of the solutions. These are the first that comes to my mind, there are surely others.

Cookie pros:

- scalability: all the data is stored in the browser so each request can go through a load balancer to different webserver and you have all the informations needed to fulfill the request;
- they can be accessed via javascript on the browser;
- not being on the server they will survive server restarts;
- RESTful: requests don't depend on server state

Cookie cons:

- storage is limited to 1KB if I remember correctly
- secure cookies are not easy to implement: take a look at the paper [A secure cookie protocol](#)

Sessions pros:

- generally easier to use, in php there's probably not much difference.
- unlimited storage

Session cons:

- more difficult to scale
- on web server restarts you can lose all sessions or not depending on the implementation

Writing Generic Applications

For more details visit

http://www.askdaveytaylor.com/how_to_get_started_programming_facebook_app_application.html

This is a guest article written by freelance programmer Ville-Matti Hakanpää

In this brief introductory tutorial I will go through some basic aspects of Facebook application development. The work itself is basic app programming, but it's a whole new world and environment. If you have a wonderful idea, Facebook can work as a great distribution channel for you. Heck, even a bad idea can get some attention fast!

Start out with the most basic of questions, though: what makes a good app?

That's the first and most common question people are asking or they should be asking. The best way to start when you're trying to decide what to develop is to look through all the other applications what are inside Facebook already. Facebook applications are listed and rated by users, so there's a feedback loop there too that tells you the kind of things people find useful and engaging.

Facebook Application(example of generic applications)

How many applications do you get after registering on Facebook. You can check with the following steps:

1. Log in to your Facebook page.
2. In the lower left corner, click the Applications button.
You'll get a menu display as shown in Figure 1.2. The menu displays the applications that you can use right now.

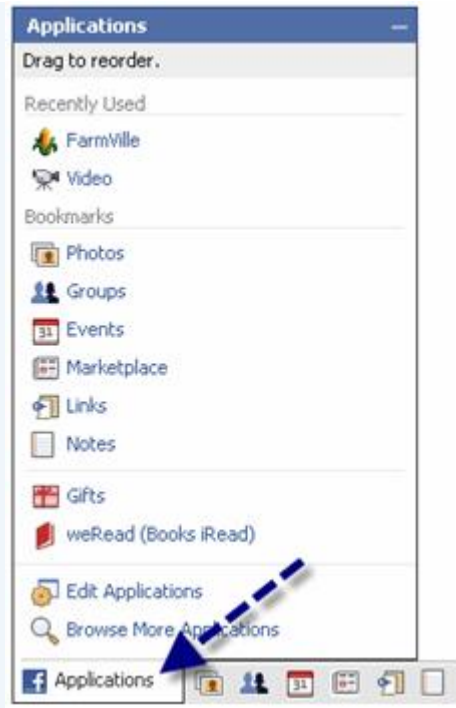


Figure 1.2 Facebook Main Menu

3. To search for other applications, you can click the link Browse More Applications. To edit an existing application, you can click the Edit Applications link.

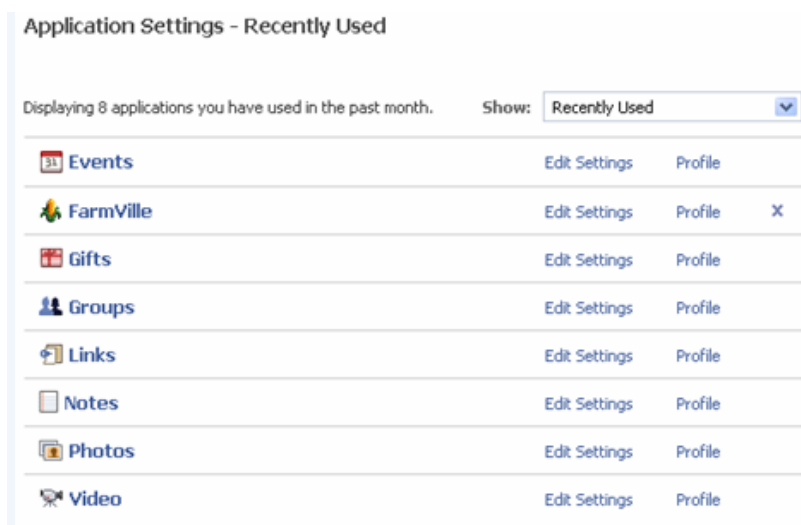


Figure 1.3 List of applications used by a user

From the explanation above, is can you imagine what is meant with Facebook Platform? Yes, Facebook Platform's like a house foundation on which to build the rooms. Fun, home owners can build rooms by knock down model. He can put any part of the building and unload at any time he likes.

Figure 1.3 shows an example list of applications used by a user. He can display a list of applications based on the intensity of usage, which has applications in bookmark it, has been authorized, and so on. Then you can adjust settings on the application. So, this practically is the central of your application control.



Figure 1.4 Setting of application

Third Party Application

Well, fun, Facebook does not only display their application himself. Anyone can create and run applications on Facebook. This is one of the uniqueness of Facebook. this site is not only friendship site that collects just friends, but it also could be developed in accordance with user needs.

You can use Facebook when saturated. Even at the time did not want to meet or greet friends. You can search for game applications on Facebook. One of the popular game is Farmville. In this game, you're not just playing alone, you can also neighbors and help your neighbors. Thus, the concept used in applications is community development.

So, What is Facebook Platform?

By definition, Facebook platform can be interpreted: framework programming that enables application developers create applications that can interact and integrate with Facebook core services. Example, application named Super Wall application.

Super Wall application is not created by Facebook team but by other developers. In the application, you can display wall in a different format. You can display the status of your friends in this application. This means, the application can interact with your data on Facebook service standards.

Mmm ... you should mention that Facebook is a place to integrate all the applications you need. You can build financial applications, appointments, and other business applications. You can use Facebook as a portal of all the data you need. That is the advantage of the Facebook platform.

Facebook Application Architecture

Well, we'll compare how common web application architecture with applications on Facebook. Below is a standard web architecture:

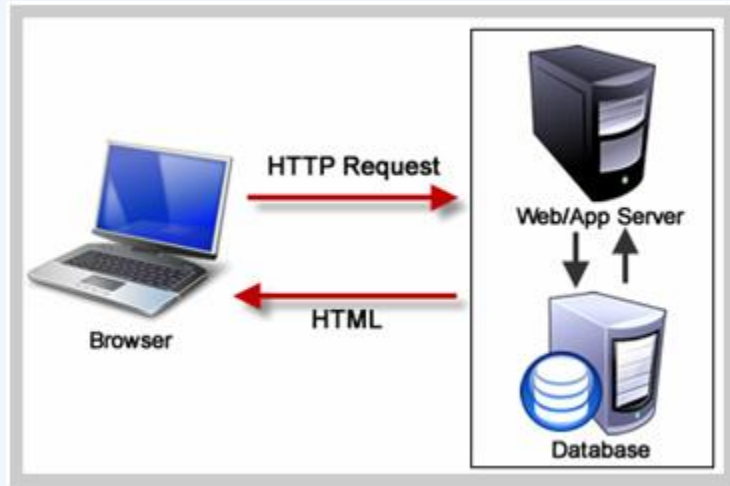


Figure 1.7 Standar web architectur

When you want to open a web page, you send an HTTP Request to the server. On the server there will be a process involving the application and database. The results of this process, an HTML document will be sent to your browser.

Now we see how the Facebook platform works. When you open an existing application on Facebook, you send an HTTP Request to the facebook server. Your request will be forwarded to the owner of the server application. There is some information (eg user id) that will be requested from the server to the server application for Facebook. Next, Facebook server will respond and send data to the server application. After the application is processed on the server, it sends the document in a format that is sent to the FBML Facebook servers. By Facebook document is processed and produces an HTML document that is sent to your page.

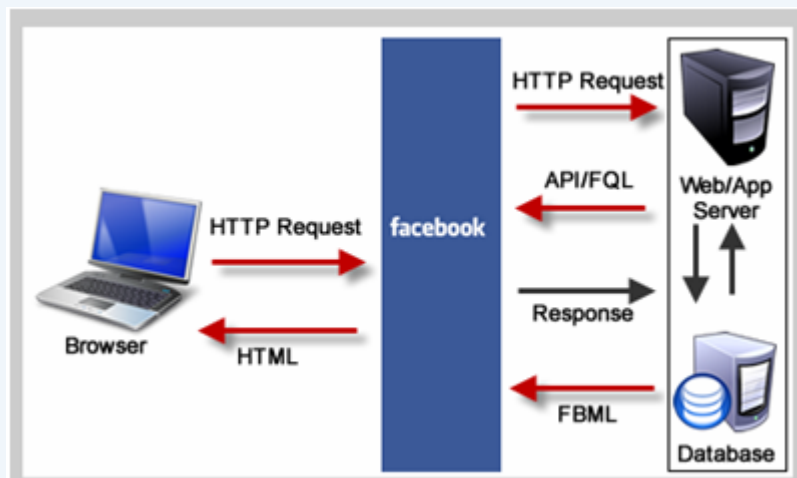


Figure 1.8 Facebook architectur

Facebook Framework Components

In the Facebook platform, there are parts where you will incorporate in your code:

- Facebook API
- FBML (Facebook Markup Language)
- FQL (Facebook Query Language)
- Facebook Javascript

Facebook API

API stands for Application Programming Interface. Facebook API is the heart of your application. Facebook API is a Web services programming interface to access the main Facebook services (profile, friends, photo, event) and the function of Facebook (login, redirect, update the view). API is REST-based architecture (Representational State Transfer).

Facebook Markup Language

Facebook Markup Language is a language like HTML but versions of Facebook for display a canvas page in Facebook. Here are three facts about FBML:

- FBML also contains a subset of HTML elements. Common elements such as p, ul, and h1 is also part of FBML.
- FBML also support for script and style elements. Javascript is not supported in the usual script. However, some javascript code can be supported on the Facebook Javascript. Elements of style can not you use to access the external style sheets. You can only use internal CSS.
- FBML provides some extensions for the user interface (UI) in particular.

Facebook Query Language

Facebook Query Language (FBQL) is a SQL-based interface to the Facebook data. Through FBQL, you can access many tables in the database such as Facebook users, friend, group, group_member, event, event_member, photo, albums, and photo_tag. FBQL is a query language similar to the SQL standard, but there are some limitations:

- SELECT statement must be one table at a time
- Join query is not allowed
- A query must be indexed
- Many implementations FBJS objects with a standard Javascript, but there are many differences.
- When processing the code in the script element, he will relate the function and variable names of the application.



Tag: facebook Category: PHP Framework Post : November 02nd 2009 Read: 126,079

Table of Content

Facebook Programming

First Application

Facebook Platform

[First Facebook App](#)

Facebook API

[Facebook API](#)

[API Friends](#)

[API Events](#)

[API Groups](#)

[API Page](#)

[API Users](#)

Facebook Markup Language

[6 Type of FBML](#)

[FBML Users and Groups](#)

[Content Conditionally](#)

[fb:if-can-see](#)

[fb:if-can-see-photo](#)

[fb:if-is-app-user](#)

[fb:if-is-friend-with-viewer](#)

[Content in Profile Boxes](#)

[fb:if and fb:switch](#)

[UI Element and Widget](#)

[Using UI Element](#)

[Discussion Board](#)

[Creating Wall in Application](#)

[Building Request and Invitations](#)

How to develop generic webpages ?

1-chose the target you want to develop

2-select the targeted fields

3-Design Database

4-enter data to database

5-read the data from database

6-pass data to the createWebPage to create webpage for each database instance

