CSC 498: Web Programming

Haidar Harmanani
Department of Computer Science and Mathematics
Lebanese American University
Byblos, 1401 2010 Lebanon

Getting Started

- In order to create a facebook application, you would need the following:
  - A facebook account (obviously!)
  - Install the Developer application
    - http://www.facebook.com/developers/
  - Your own web server
    - Facebook apps run direct from your own server!
  - A language
    - We will use PHP

Facebook Platform

- The platform consists of three core parts:
  - API that defines the various methods through which you can interact with Facebook.
  - FBML or a *Facebook Markup Language*
    - Similar to Coldfusion or ASP.NET's
  - FQL or Facebook Query Language
    - SQL for Facebook

Documentation

- Facebook makes available a set of useful tools
  - http://developers.facebook.com/tools.php
  - http://developers.facebook.com/resources.php
- The resources section contains official libraries for PHP and Java
- Links to unofficial libraries for *ActionScript*, *Cocoa*, *Coldfusion*, *.NET*, *Perl*, *Python* and *Ruby*
- There is a growing community wiki
  - http://wiki.developers.facebook.com

Example Application: Birthdays Book

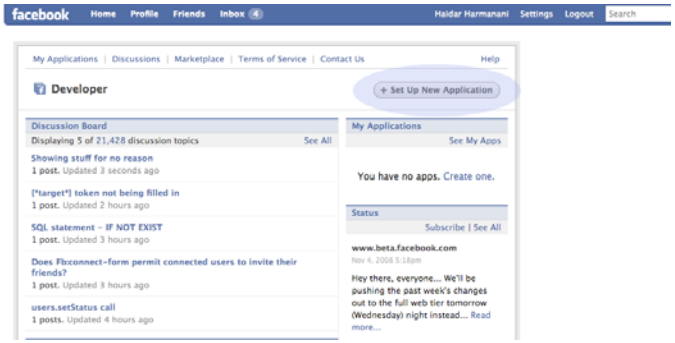
- We will step through the development of a simple application that demonstrates usage of FBML and the API
- The application provides a user with a list of hi/her friends on *Facebook*, ordered by their upcoming birthdays

Set up your Server

- Verify that your server can serve an HTML file.
- Upload the appropriate *Facebook* client library to your server.
  - If you are developing your application with PHP, its client libraries are available at http://developers.facebook.com/clientlibs/facebook-platform.tar.gz.
- If your application stores user or application information in a database, install a database such as MySQL on your server.
  - For more information on MySQL, see the MySQL website: http://dev.mysql.com/doc/mysql/en/index.html.
- Verify that you can write a database program outside of *Facebook* Platform

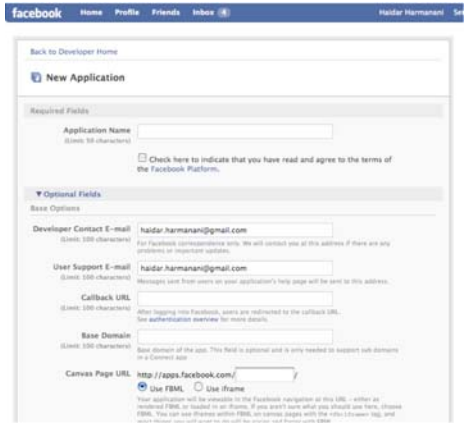
Step 1: Set Up your Application

- log in to the *Facebook* Developer application at:
  - http://www.facebook.com/developer/
- Click on Set up New Application



Step 1: Set Up your Application

- Point your application to the proper site
  - Click on the "Optional Fields" to get started



Step 1: Set Up your Application

Field	Value
Callback URL	http://www.yoursite.com/yourapplication
Canvas Page URL	My Example Application
Can your Applications be added on	Yes
Post-Add URL	http://apps.facebook.com/yourapplication
Developer Mode	Yes
Side Nav URL	http://apps.facebook.com/yourapplication
Private Installation	Yes

Step 1: Set Up your Application

- *Callback URL* serves two functions:
  - After users successfully log in to your app, they will be redirected to this page.
  - The URL serves as the "canvas page"
- *Canvas Page* provides a way for an app to "live inside" Facebook
  - Facebook will fetch the page content (i.e., what will be rendered inside the Facebook frame) from your callback URL
  - Example,
    - If your callback URL is example.com/ and your canvas page URL is apps.facebook.com/pokewall/, then apps.facebook.com/pokewall/poke.php will request example.com/poke.php.

Step 1: Set Up your Application

- Canvas Page URL
  - The URL at which your app's canvas pages exist.
  - Of the form apps.facebook.com/appname, where appname must be between 7 and 20 characters (inclusive), and can only contain letters, dashes, and underscores.
- Side Nav URL
  - If specified, your application will appear as a link in the Applications menu
  - For users who have interacted with your application.
- Icon
  - A 16 x 16 icon displayed next to your app's name
- Post-Add URL
  - A page to redirect the user after the application is added to a Facebook Page.

Step 1: Set Up your Application

- Post-Authorize Redirect URL
  - When a user authorizes your application, that user is redirected to this URL.
- Post-Remove URL
  - After a user removes your application, this page will receive an automated POST call from Facebook containing the variable fb_sig_user (user ID).
  - This page cannot be a Facebook-framed page, since they are removing the app means they are no longer using the app within Facebook.
- Developer Mode
  - While developing your application, you may want to restrict its visibility.
  - You can (and probably should) check this box to only allow developers of the application to add it.
- Logo
  - A 75x75 logo, used in headers or directory listings where a larger graphic can be displayed.

Step 1: Set Up your Application

- Set up your web server with the relevant Facebook client for your language or framework
  - We will use PHP so we want the official PHP library files
    - http://developers.facebook.com/clientlibs/facebook-platform.tar.gz

Step 2: Build the application

- Start out by including the client library and setting your API key and application "secret" (password)
- You should be able to get these from the 'My Applications' page after you have set up your new

```
<?php
// Include the Facebook client library
```

Step 2: Build the application

- We also set the url for where we are storing our code:

```
$appcallbackurl = 'YOUR_URL';
```

Step 2: Build the application

- The application requires users to be logged in to Facebook
  - Include a snippet from the Facebook code samples to make sure the current user is set up properly
  - Save the code as *config.php*

```
$user = $facebook->require_add();
//catch the exception that gets thrown if the cookie has an invalid session_key in it
try
{
if (!$facebook->api_client->users_isAppAdded())
      {
      $facebook->redirect($facebook->get_add_url());
      }
}
catch (Exception $ex)
{
//this will clear cookies for your application and redirect them to a login prompt
$facebook->set_user(null, null);
$facebook->redirect($appcallbackurl);
}
```

Step 2: Build the application

- Start the main application page
  - *index.php*
- Include the configuration file:

```
<?php
    require_once ('config.php');
```

Step 2: Build the application

- Get a list of all the current user's friends' birthdays, which we will store in the $friends variable:

```
$friends = $facebook->api_client->users_getinfo($facebook->api_client->friends_get(),
  'birthday');
?>
```

Notes on the API

- The code in the previous slides uses two API methods in order to get the information we want
  - Users.getInfo
    - http://wiki.developers.facebook.com/index.php/Users.getInfo

  and
  - Friends.get
    - http://wiki.developers.facebook.com/index.php/Friends.get

Notes on the API

- The API docs do not provide details of the parameters or parameter order required by the library
  - You need to delve into the source code of your chosen library to discover this information.
  - The official libraries are well-documented with comments so you should be able to find your way around
  - Just remember the API method Users.getInfo becomes the users_getinfo library method in the PHP library.

Notes on the API

- FBML is basically HTML
  - The link element isn't permitted and the @import rule isn't supported either, so the simplest solution is to use a server-side include.
  - There are some more useful CSS tips
    - *http://wiki.developers.facebook.com/index.php/CSS_Tips_and_Tricks*

```
<h1>Birthday Books</h1>
<style type="text/css">
<?php include("style.css"); ?>
</style>
```

FBML markup tags

- <fb:name uid="' . $friend[uid] . '"/> is a special FBML markup tags
  - Facebook users are referenced by a unique identifier which is returned by various API methods
  - We pass this *uid* value to the fb:name and fb:profile tags which return the name and profile picture of that user
- A full list of FBML tags can be found at
  - *http://wiki.developers.facebook.com/index.php/Category:FBML_tags*

Step 3: write the Application

- Set-up a couple of empty arrays for storing information about our friends.
- Set a timestamp of the current day and month and store it in the variable $now.
- For a full list of formatting options for the date function see the PHP date page
  - http://uk3.php.net/date

```
<table>
<?php
```

Step 3: write the Application

- Loop through the list of friends in the $friends variable
  - Generate another array of these friends sorted based on upcoming birthday order
- Hold details of each person in a placeholder array, $person
- Place each $person with birthday information into another array called $with_birthday
  - We're going to want to get at their name, their profile picture, and details of their birthday
  - For ease of reference we store these against relevant keys in the array: name, image, month, day, year

```
foreach ($friends as $friend) {
    $person = array();
    $person['name'] = '<fb:name uid="' . $friend[uid] . '"/>';
    $person['image'] = '<fb:profile-pic uid="' . $friend[uid] . '" linked="true"
    size="square" />';
```

Step 3: write the Application

```
$person['day'] = date("jS", strtotime($friend[birthday]));
$person['month'] = date("F", strtotime($friend[birthday]));
$person['absolute_timestamp'] = strtotime($friend[birthday]);
$person['relative_timestamp'] = strtotime($person['month'] . $person['day']);

if ($person['relative_timestamp'] < $now)
{
  // birthday has already happened this year
  $person['year'] = date('Y', strtotime('+1 year'));
}
else
{
  // birthday still to come this year
  $person['year'] = date('Y');
}
```

Step 3: write the Application

- Not everyone enters their birthday details into Facebook.
- Store those people in the $without_birthday array

```
if ($person['absolute_timestamp'])
 {
   $with_birthday[] = $person;
 }
 else
 {
   $without_birthday[] = $person;
 }
}
```

Step 3: write the Application

- Now we have an array of people with birthdays
  - Can loop through, but they're in the wrong order
  - Use the handy PHP array_multisort
    - array_multisort allows us to sort a given dataset ($with_birthdays) by one column ('relative_timestamp_with_year') using different order flags (in this case SORT_ASC, an ascending sort)

```
foreach ($with_birthday as $key => $row)
{
  $relative_timestamp_with_year[$key] = $row['relative_timestamp_with_year'];
}
array_multisort($relative_timestamp_with_year, SORT_ASC, $with_birthday);
?>
```

Step 3: write the Application

- Now loop through the final, sorted array and output a table of our friends together with their birthdays
- We've thrown in some *microformats* to make it possible to export the data later

Step 3: write the Application

```
<table>
<?php
$i = 0;
foreach ($with_birthday as $friend)
{
  echo '<tr class="vevent';
    if ($i == 0) {
      echo ' odd';
      $i = 1;
    } else {
      $i = 0;
    }
  echo '">';
  echo '<td>' . $friend['image'] . '</td>';
  echo '<th scope="row" class="summary vcard"><span class="fn">' . $friend['name'] . '</span></th>';
  echo '<td><abbr class="dtstart" title="' . $friend['year'] . '-'
     . date('m', strtotime($friend['month'])) . '-' . substr($friend['day'],0,-2) . '">'
     . $friend['day'] . ' ' . $friend['month'] . ' ' . $friend['year'] . '</td>';
  echo '</tr>';
}
?>
</table>
```

Step 3: write the Application

- We now have version one of our Birthdays Book application
- As a final touch, we'll track how many people are using our application using Google Analytics
  - We can't include Javascript in our pages, but there is a handy FBML tag, fb:google-analytics—just include your unique identifier from Google Analytics in the uacct parameter

    <fb:google-analytics uacct="YOUR_ANALYTICS_ID" page="Birthdays Book"/>
- The finished application can be found at apps.facebook.com/birthdaysbook and you can download the source files at www.digital-web.com/extras/facebook_application.

Other Facebook Issues

FBML

- Facebook Markup Language (FBML)
  - An evolved subset of HTML with some elements removed, and others which have been added that are specific to Facebook
- Enables developers to build full Facebook Platform applications that integrate into a user's Facebook experience.
- You can hook into several Facebook integration points, including the profile, profile actions, Facebook canvas, News Feed and Mini-Feed.
- You set the FBML for a profile box by calling profile.setFBML through the API.
- The FBML is cached on Facebook's server until profile.setFBML is called again through a canvas page.

FBML Examples

- Fb:name
  - Renders the name of the user specified, optionally linked to his or her profile
  - <fb:name uid="$tagger" capitalize="true" /> tagged a photo of <fb:name subjectid="$tagger" uid="$tagee" />
- Fb:user-status
  - Returns the status of the user specified by uid
  - <fb:user-status uid="12345" linked="true"></fb:user-status>

FQL

- Facebook Query Language
  - Allows developers to use a SQL-style
- A way to query the same *Facebook* data you can access through the other API functions, but with a SQL-style interface.
  - Many of the normal API calls are simple wrappers for FQL queries
- advantages of using FQL over our more traditional API methods are as follows:
  - Condensed XML reduces bandwidth and parsing costs.
  - More complex requests can reduce the number of requests necessary.
  - Provides a single consistent, unified interface for all of your data.

FQL Examples

- Examples
  - SELECT name, pic FROM user WHERE uid=211031 OR uid=4801660
  - SELECT education_history, current_location
  - SELECT education_history.name, current_location.zip WHERE "Stanford" IN education_history.name OR current_location.zip = 07079

FBJS

- FBJS is *Facebook's* solution for developers who want to use JavaScript in their *Facebook* applications
- Typically, providers who allow developers to embed JavaScript within their domain force developers to use *iframes* to sandbox their code.
- Facebook has taken a different approach to this problem.
  - JavaScript gets parsed, and any identifiers get *prepended* with your application ID.

FBJS: The Basics

- The JavaScript syntax is exactly the same.
  - Can create objects, use anonymous functions, create timeouts and almost any other thing you can think of
  - Modifying the DOM tree is slightly different, however.

FBJS: Example

```
<a href="#" onclick="hello_world(this); return false;">Hello World!</
  a>
<script>
<!-- function random_int(lo, hi)
{
return Math.floor((Math.random() * (hi - lo)) + lo);
}

function hello_world(obj)
{
   var r = random_int(0, 255), b = random_int(0, 255), g =
   random_int(0, 255);
   var color = r+', '+g+', '+b; obj.setStyle('color', 'rgb('+color+')'); }
//-->
```

FBJS DOM Objects

- Retrieving Objects
  - A handle to an FBJS DOM object can be retrieved by either calling document.getElementById, or document.createElement.
  - Additionally, the "this" pointer in DOM events also points to the target of the event.
- Manipulating Objects
  - FBJS DOM objects implement most of the same methods regular JavaScript objects implement including: appendChild, insertBefore, removeChild, and cloneNode.
  - Properties like parentNode, nextSibling, src, href (and many many others) have been redefined as a couplet of getters and setters.
  - Instead of obj.parentNode just call obj.getParentNode(), and so on

FBJS DOM Objects

JavaScript	FBJS getter	FBJS setter
parentNode	getParentNode	
nextSibling	getNextSibling	
previousSibling	getPreviousSibling	
firstChild	getFirstChild	
lastChild	getLastChild	
childNodes	getChildNodes	
innerHTML	n/a	setInnerFBML
innerHTML	n/a	setInnerXHTML
action	getAction	setAction
Value	getValue	setValue
…	…	…