CS 5142
Scripting Languages

10/04/2013

PHP

Announcements

- Homework: Try to write a heap
- Make-up Prelim: Tell me by 10/18

Outline

- No HTML5 discussion.
- If interested in HTML and CSS:

  http://cs.nyu.edu/rgrimm/teaching/sp13-web/notes0206.html

- Facebook SDK

Facebook SDK

- Install the Facebook PHP SDK from github:

```
git clone https://github.com/facebook/facebook-php-sdk.git
```

Git Basic Operations

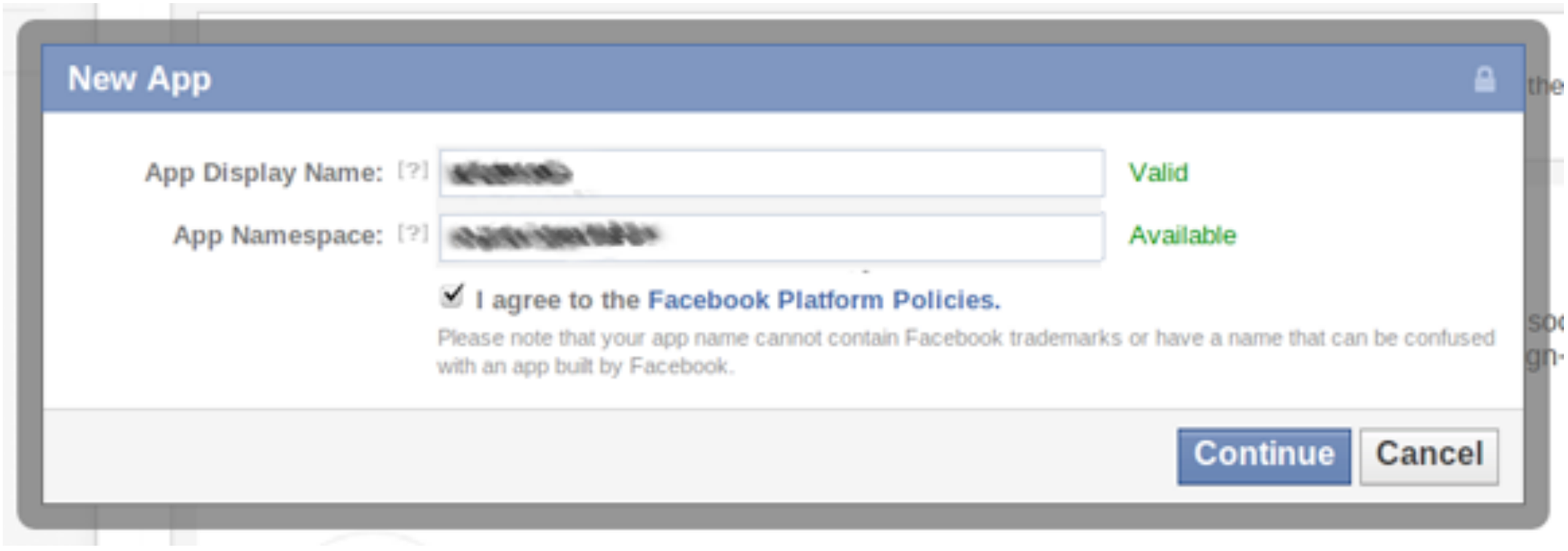
```
# Download from the master branch
$ git clone <repository name>

#  Add a file to the commit
$ git add <fileName>

# Commit files to local repository
$ git commit -a -m "<message>"

# Upload changes to the server
$ git push
```

- Distributed reversion control
- Can work “offline”
- Developed by Linus Torvalds for the Linux kernel

Create New App

Go to: https://developers.facebook.com/apps/

Create new app:



After authorization, Facebook will redirect you to a new URL

Provide a URL for your application (REDIRECT_URI)

Note the *App ID and App Secret*

Minimal Application

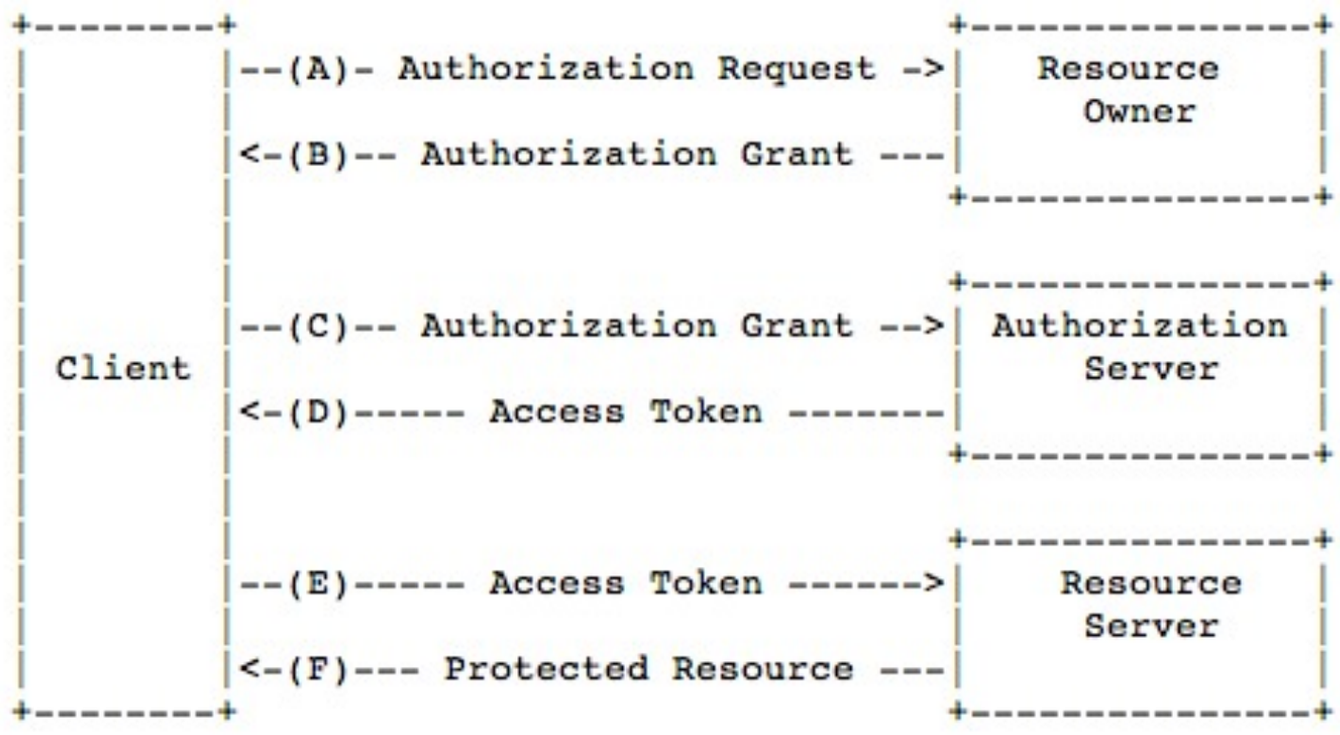
```
// Get the Facebook SDK
require_once("facebook.php");

$config = array();
$config['appId'] = 'YOUR_APP_ID';
$config['secret'] = 'YOUR_APP_SECRET';
$config['fileUpload'] = false; // optional

// Facebook object
$facebook = new Facebook($config);

// Get User ID (if available)
$user = $facebook->getUser();
```

Facebook Object

api	Call Graph API or an FQL query
getAccessToken	Get the access token, give permission to app to use email, profile properties, friend list, etc.
getLoginStatusUrl	Return a URL based on user's login status
getLoginUrl	Return a URL that user click to login, authorize app, redirect back to app
getLogoutUrl	Return a URL that user click to logout of Facebook session, redirect back to app
getUser	Returns userid of the current user

OAuth

- OAuth 2.0 open standard authorization protocol
- Gives 3rd party apps access to an HTTP service for a limited time



Authorization

```
$params = array(
    "redirect_uri" => REDIRECT_URI,
    "scope" => "email,read_stream,publish_stream,
                user_photos,user_videos");
echo '<a href="' . $fb->getLoginUrl($params) . '">Login</a>';
```

Authorization

```
// Permission denied
http://example.com/facebook/myapp.php?
error=access_denied&error_reason=user_denied&error_description=The+user
+denied+your+request.

// Permission granted, with code parameter appended
http://example.com/facebook/myapp.php?code=TOKEN_VALUE

// Code value used to get access token
https://graph.facebook.com/oauth/access_token?
client_id=FACEBOOK_APP_ID&redirect_uri=FACEBOOK_REDIRECT_URI&client_secre
t=FACEBOOK_APP_SECRET&code=TOKEN_VALUE
```

SOAP vs. REST

- **A *web API* is a server-side interface for web applications**
- **SOAP = Simple Object Access Protocol**
  - XML on top of HTTP or TCP, describes functions and types of data
  - Successor of XML-RPC
  - Usually associated with a service-oriented architecture
- **REST = Representational State Transfer**
  - Very lightweight
  - Typically uses HTTP PUT/GET/POST/DELETE instead of an XML
  - Usually associated with a resource-oriented architecture
  - Facebook uses REST

Graph API

Read the user's feed of status messages with a GET request:

```
https://graph.facebook.com/me/feed?access_token=ACESS_TOKEN
```

In your script, it looks like this:

```
$data = $facebook->api('/me/feed');
```

Public Profile

GET request:

```
https://graph.facebook.com/soule/
```

In your PHP script:

```
if ($user) {
  try {
    // Proceed knowing you have a logged
    // in user who's authenticated.
    $user_profile = $facebook->api('/me/');
  } catch (FacebookApiException $e) {
    error_log($e);
    $user = null;
  }
}
```

User Information


```
{
   "id": "123456789",
   "name": "Robert Soule",
   "first_name": "Robert",
   "last_name": "Soule",
   "link": "http://www.facebook.com/soule",
   "username": "soule",
   "gender": "male",
   "locale": "en_US"
}
```

JSON

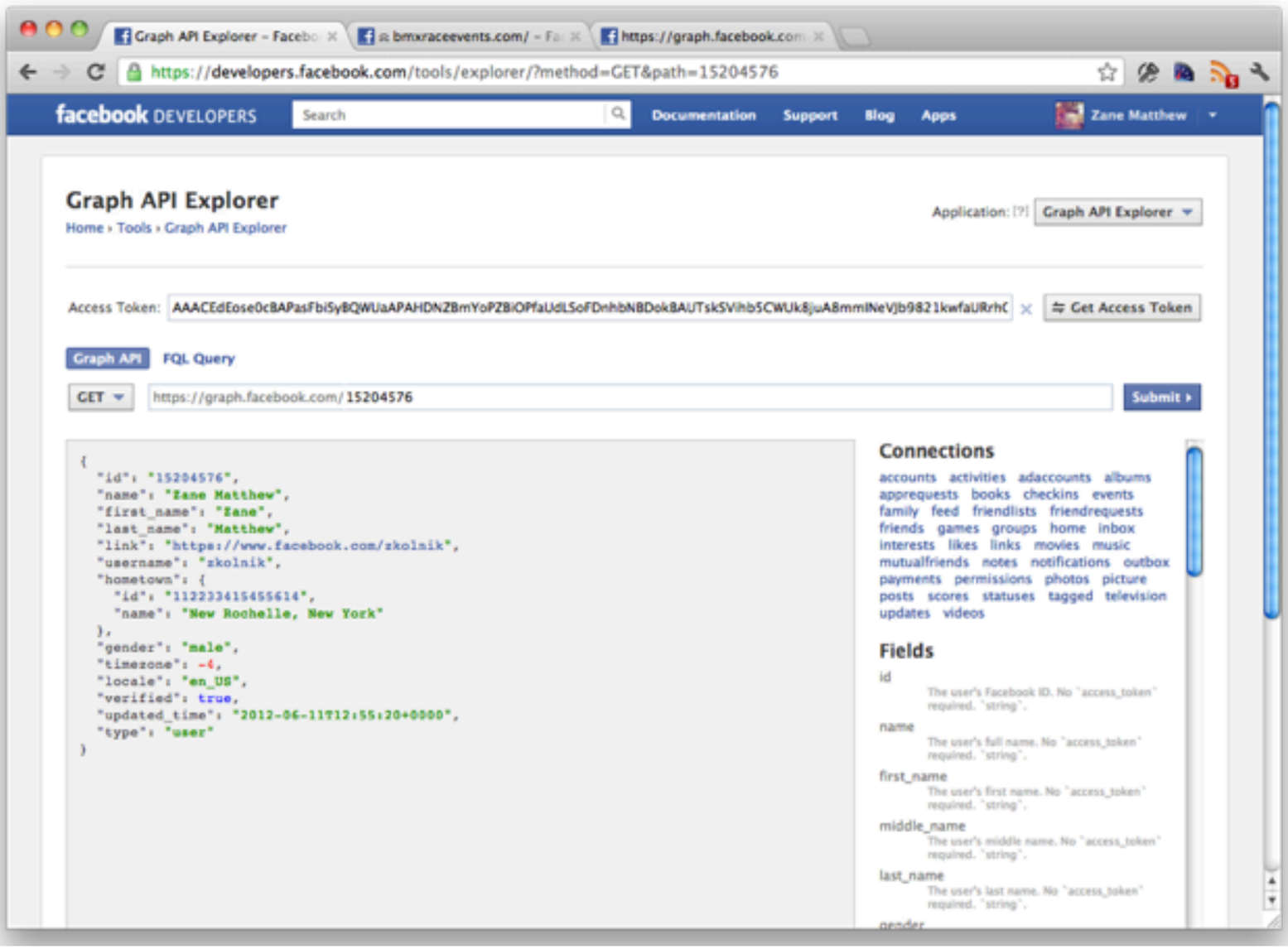
- Javascript Object Notation
- Derived from JavaScript
- Human-readable data format
- Alternative to XML
- Basic data types: Number, String, Boolean, Array, Object, null
- Structure characters {} [] : ,

JSON Example

```
<!DOCTYPE html>
<html>
<body>
<h2>JSON Object Creation in JavaScript</h2>
<p>
Name: <span id="jname"></span><br />
Age: <span id="jage"></span><br />
Address: <span id="jstreet"></span><br />
Phone: <span id="jphone"></span><br />
</p>
<script>
var JSONObject= {
"name":"John Johnson",
"street":"Oslo West 555",
"age":33,
"phone":"555 1234567"};
document.getElementById("jname").innerHTML=JSONObject.name
document.getElementById("jage").innerHTML=JSONObject.age
document.getElementById("jstreet").innerHTML=JSONObject.street
document.getElementById("jphone").innerHTML=JSONObject.phone
</script>

</body>
</html>
```

Graph Explorer

https://developers.facebook.com/tools/explorer



Update a User

Post to status feed:

```
<?php
$data = array("message" => "Hello World!");
$status = $facebook->api("/me/feed", "POST", $data);
```

Upload a photo:

```
<?php
$facebook->setFileUploadSupport(true);
$data = array(
    "name" => "a vacation photo",
    "image" => "@/home/soule/vacation/img42.jpg");
$status = $facebook->api("/me/photos", "POST", $data);
```

PHP:The Bad

Reference:

http://me.veekun.com/blog/2012/04/09/php-a-fractal-of-bad-design/

Examples:

```
“foo” == TRUE, “foo” == 0, but 0 != TRUE
```

```
NULL < -1 and NULL == 0
```

Soap-box

Evaluating PHP

Strenghts

- Simplicity
- Portability
- Large libraries
- Many database bindings
- Popularity

Weaknesses

- Error handling
- Lack of scalability
  - Compared to Java
- Low-level
  - Compared to Ruby on Rails or Google Web Toolkit

Last Slide

- Next week: client-side JavaScript