

FACEBOOK PHP SDK

Version 2.0.3 - 11-5-2010

<http://github.com/facebook/php-sdk/>

class FacebookApiException

Thrown when an API call returns an exception.

```
protected $result;
// The result from the API server that represents the exception
// information.

public function __construct($result)
// Make a new API Exception with the given result.

public function getResult()
// Return the associated result object returned by the API server.

public function getType()
// Returns the associated type for the error. This will default to
// 'Exception' when a type is not available.

public function __toString()
// To make debugging easier.
```

example.php

```
<?php
require './src/facebook.php';

// Create our Application instance.
$facebook = new Facebook(array(
    'appId' => '254732073152',
    'secret' => '904270b68a2c3d54485323652da4d14',
    'cookie' => true,
));

Sme = null;
// Session based API call.
if ($session) {
    try {
        $uid = $facebook->getUser();
        $sme = $facebook->api('/me');
    } catch (FacebookApiException $e) {
        error_log($e);
    }
}

// login or logout url will be needed depending on
// current user state.
if ($sme) {
    $logoutUrl = $facebook->getLogoutUrl();
} else {
    $loginUrl = $facebook->getLoginUrl();
}

// This call will always work since we are fetching
// public data.
$snaitik = $facebook->api('/naitik');
?>
```

```
<!doctype html>
<html
  xmlns:fb="http://www.facebook.com/2008/fbml">
<head>
<title>php-sdk</title>
<style>
  font-family: Lucida Grande, Verdana, Arial, sans-
  serif;
</style>
  h1 a {
    text-decoration: none;
    color: #3b5998;
  }
  h1 a:
    text-decoration: underline;
  </style>
</head>
<body>
  <!--
  We use the JS SDK to provide a richer user
  experience. For more info,
  look here: http://github.com/facebook/connect.js
  -->
  <div id="fb-root"></div>
  <script>
  window.fbAsyncInit = function() {
  FB.init({
  appId: '
  status: true, // check login status
  cookie: true, // enable cookies to allow the server to
  access the session
  xfbml: true // parse XFBML
  });
  // whenever the user logs in, we refresh the page
  FB.Event.subscribe('auth.login', function() {
  window.location.reload();
  });
  </script>
  <script>
  <h1><a href="example.php">php-sdk</a></h1>
  <?php if ($sme): ?>
  <a href="
  
  </a>
  <?php else: ?>
  Using JavaScript &amp; XFBML: <b>login-
  button</b>
  </div>
  Without using JavaScript &amp; XFBML:
  <a href="
  
  </a>
  <?php endif ?>
  <h3>Sessions/h3>
  <?php if ($sme): ?>
  <pre><?php print_r($session); ?></pre>
  <h3>You/h3>
  
  <h3>Your User Object/h3>
  <pre><?php print_r($sme); ?></pre>
  <?php else: ?>
  <strong><em>You are not
  Connected.</em></strong>
  <?php endif ?>
  <h3>Naitik/h3>
  
  </a>
  <?php echo $snaitik['name']; ?>
  </body>
  </html>
```

class Facebook

Provides access to the Facebook Platform.

```
public function __construct($config)
// Initialize a Facebook Application.

public function setAppId($appId)
// Set the Application ID.

public function getAppId()
// Get the API Key.

public function setApiSecret($apiSecret)
// Set the API Secret.

public function getApiSecret()
// Get the API Secret.

public function setCookieSupport($cookieSupport)
// Set the Cookie Support status.

public function useCookieSupport()
// Get the Cookie Support status.

public function setBaseDomain($domain)
// Set the base domain for the Cookie.

public function getBaseDomain()
// Get the base domain for the Cookie.

public function setSession($session=null, ,
    $write_cookie=true)
// Set the Session.

public function getSession()
// Get the session object. This will automatically look for a signed
// session sent via the Cookie or Query Parameters if needed.

public function getUser()
// Get the UID from the session.

public function getLoginUrl($params=array())
// Get a Login URL for use with redirects. By default, full page
// redirect is assumed. If you are using the generated URL with a
// window.open() call in JavaScript, you can pass in display=popup as
// part of the $params.

public function getLogoutUrl($params=array())
// Get a Logout URL suitable for use with redirects.

public function getLoginStatusUrl($params=array())
// Get a login status URL to fetch the status from facebook.

public function api(/* polymorphic */)
// Make an API call.

protected function _restserver($params)
// Invoke the old restserver.php endpoint.

protected function _graph($path,
    $method='GET', $params=array())
// Invoke the Graph API.

protected function _oauthRequest($url,
    $params)
// Make a OAuth Request

protected function makeRequest($url, $params,
    $ch=null)
// Makes an HTTP request. This method can be overridden by
// subclasses if developers want to do fancier things or use
// something other than curl to make the request.

protected function getSessionCookieName()
// The name of the Cookie that contains the session.

protected function setCookieFromSession($session=null)
// Set a JS Cookie based on the _passed_in_ session. It does not use
// the currently stored session -- you need to explicitly pass it in.

protected function validateSessionObject($session)
// Validates a session _version=3 style session object.

protected function getApiUrl($method)
// Build the URL for api given parameters.

protected function getUrl($name, $path="",
    $params=array())
// Build the URL for given domain alias, path and parameters.

protected function getCurrentUrl()
// Returns the Current URL, stripping it of known FB parameters that
// should not persist.

protected static function generateSignature($params, $secret)
// Generate a signature for the given params and secret.
```

Cheat Sheet available at
www.princessdesign.net

Princessdesign
web studio with wireless imagination