

STARTING WITH PHP AND TELLDUS LIVE API

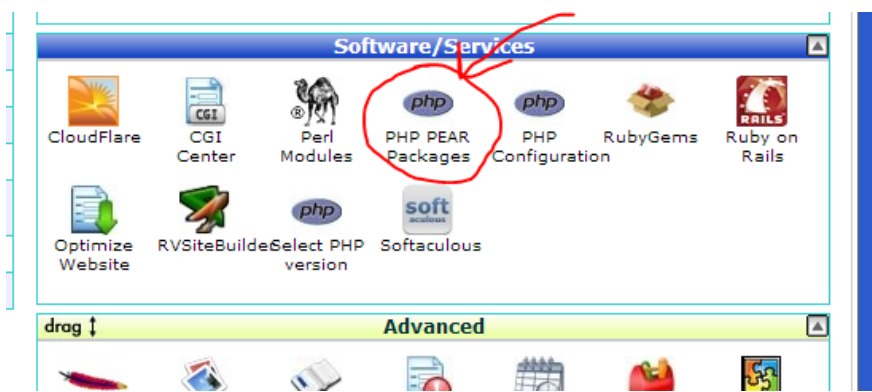
By Robert Andresen
robertan.com

You will need HTTP_OAuth to do API-calls against Telldus Live. I have downloaded this package in cPanel, on my server hosted by ProISP.no.

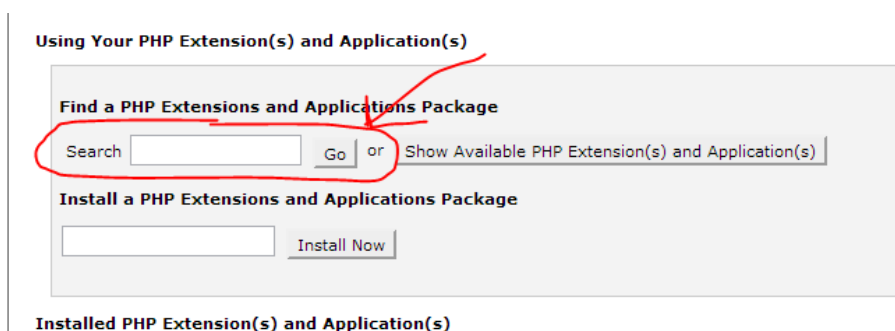
Package can be found here: http://pear.php.net/package/HTTP_OAuth/

Install HTTP_OAuth in cPanel

Click on PHP PEAR Packages



Search for HTTP_OAuth, select it and install.



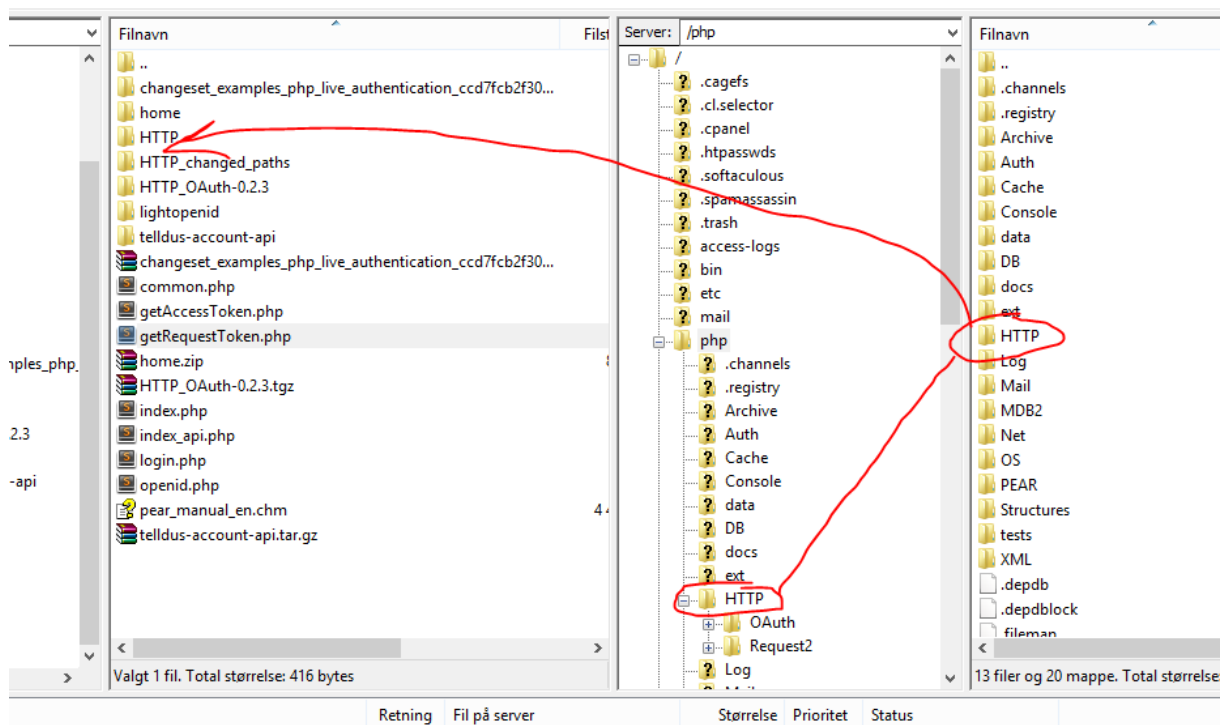
Getting http_OAuth to work

I think this should work right out of the box, but it was a bit tricky on my server. The application couldn't find it's own path and was trying to include the files from the execute root folder.

This could be fixed in three ways:

1. Add the path in php.ini (I didn't have access to this file on my server)
2. Change the path in every file, with the full path from server root (a lot of work).
3. Copy the HTTP folder inside you're application (i used this option ☺).

Copy the HTTP folder into you're folder application



Download telldus template files for login and API-calls

Click «Open source» in the navigation at <http://www.telldus.se/>

This will open the page <http://developer.telldus.se/>

Click «Read more» under the section about Telldus Live for API functions.

Download Telldus Account API files

Examples - PHP - Account

Techniques

- Telldus Account API
- Telldus Account API
- Telldus Live API

Telldus Account Login Service

This example shows how to use Telldus Account Login Service to allow user to login using their Telldus account. We need the LightOpenID library for this example to work.

Walkthrough of the files:

[index.php](#)

First we must check if the user is already logged in. We will do this by checking the session variable:

```
if (isset($_SESSION['loggedin'])) {  
    If the user is logged in, show our user's data:  
  
    echo "<p>You are logged in!</p>";  
    echo "Telldus ID: <b>" . $_SESSION['telldus_id'] . " ";  
    </b><br>";  
    echo "Email: <b>" . $_SESSION['email'] . "</b><br>";  
}
```

Requirements

- PHP >= 5.2.0
- LightOpenID
- [Download](#)
telldus-account-api.tar.gz

Download Telldus Live API

Examples - PHP - Telldus Live!

Techniques

- Telldus Account API
- Telldus Account API
- Telldus Live API

PHP example for Telldus Live!

This example shows the very basics things needed to be able to do API calls, no actual parsing of the response is done.

In order to run this example the server needs to have PEAR::HTTP_OAuth installed. You also need to obtain an API key and secret. These should be set in the file common.php

[getRequestToken.php](#)

To talk to Telldus server the application need to request a request token. We also need to supply a return url so the server knows where to redirect the user after he or she has authorized the application access to his or hers resources.

```
$consumer = new HTTP_OAuth_Consumer(constant('KEY'), constant('SECRET'));  
$consumer->getRequestToken(constant('REQUEST_TOKEN'), constant('BASE_URL').'/getAccessToken.php');  
);  
  
When we have those we store them in the session.  
  
$_SESSION['token'] = $consumer->getToken();  
$_SESSION['tokenSecret'] = $consumer->getTokenSecret();  
  
Redirect the user to the authorization page  
  
$url = $consumer->getAuthorizeUrl();
```

Requirements

- PHP >= 5.2.0
- PEAR::HTTP_OAuth
- API key and secret
- [Code](#)
Telldus Developer
- [Download](#)
telldus-live-api.zip

I had to download each file manually:

[Wiki](#) [Blog](#) [Timeline](#) [Road](#)


source: [examples](#) / [php](#) / [live](#) / [authentication](#) / [common.php @ 7834d9](#)

Revision **7834d9**, 588 bytes checked in by Micke Prag <micke.prag@...>, 14 months ago (diff)
Fixed wrong endpoint in our php example
Property mode set to 100644


Line	
1	<?php
2	
3	session_start();
4	
5	require_once 'HTTP/OAuth/Consumer.php';
6	
7	define('PUBLIC_KEY', '');
8	define('PRIVATE_KEY', '');
9	
10	define('URL', 'http://api.telldus.com'); //https should be used in production!
11	define('REQUEST_TOKEN', constant('URL').'/oauth/requestToken');
12	define('AUTHORIZE_TOKEN', constant('URL').'/oauth/authorize');
13	define('ACCESS_TOKEN', constant('URL').'/oauth/accessToken');
14	define('REQUEST_URI', constant('URL').'/xml');
15	
16	define('BASE_URL', 'http://'.\$_SERVER['SERVER_NAME'].dirname(\$_SERVER['REQUEST_URI']));
17	
18	define('TELLSTICK_TURNON', 1);
19	define('TELLSTICK_TURNOFF', 2);

[View changes...](#)

Download in other formats:
[Plain Text](#) | [Original Format](#)

 Powered by [Trac 0.12.2](#)

Get you're API keys

 Logged in. [Log out >](#)

[Start](#) [Your keys](#) [API explorer](#) [Documentation](#)

API Keys

Your keys

API Keys

In order to access the Telldus Live! API you need an API Key.

For development and testing you can generate a token for your user only

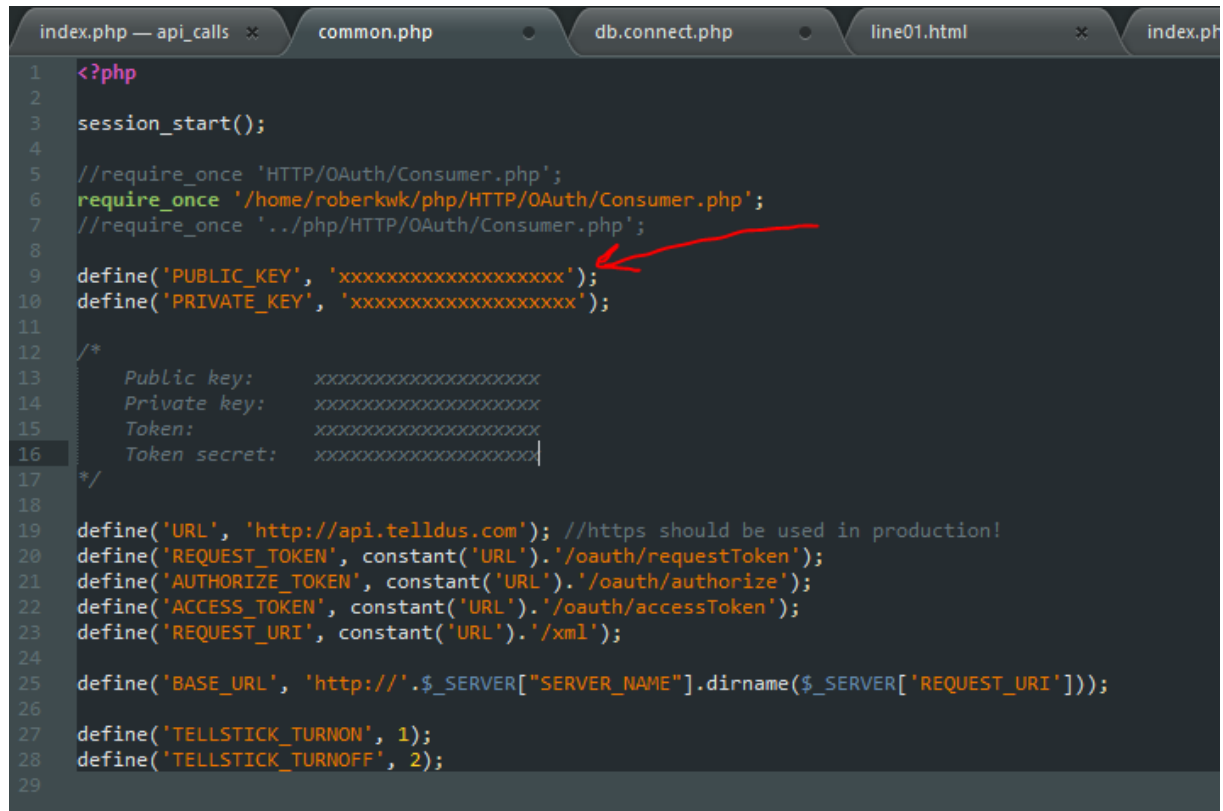
If the intention of your application is to be public, you will need a public API Key. To get one, please contact us.

- Generate a private token for my user only
- Contact Telldus to retrieve a public key

Start with PHP

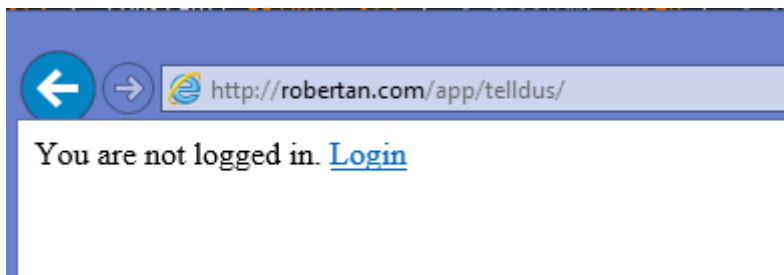
I use sublime text 2 as php-editor with obsidian color scheme.

Insert the generated keys in common.php file

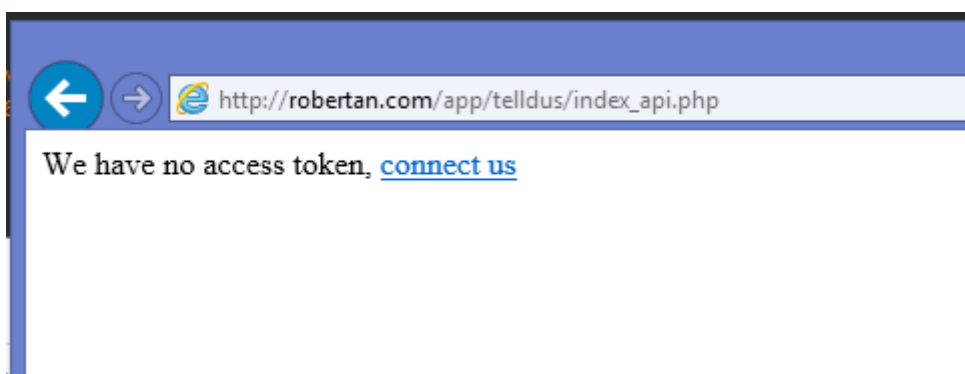


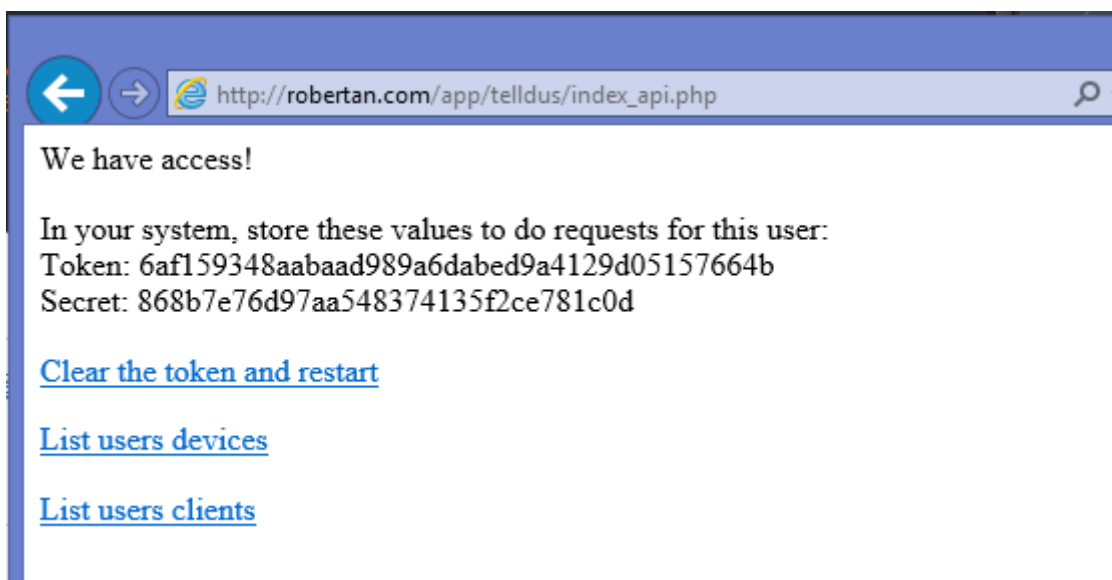
```
1 <?php
2
3 session_start();
4
5 //require_once 'HTTP/OAuth/Consumer.php';
6 require_once '/home/roberkwk/php/HTTP/OAuth/Consumer.php';
7 //require_once '../php/HTTP/OAuth/Consumer.php';
8
9 define('PUBLIC_KEY', 'xxxxxxxxxxxxxxxxxxxx');
10 define('PRIVATE_KEY', 'xxxxxxxxxxxxxxxxxxxx');
11
12 /*
13     Public key:      xxxxxxxxxxxxxxxxxxxxxx
14     Private key:     xxxxxxxxxxxxxxxxxxxxxx
15     Token:           xxxxxxxxxxxxxxxxxxxxxx
16     Token secret:    xxxxxxxxxxxxxxxxxxxxxx
17 */
18
19 define('URL', 'http://api.telldus.com'); //https should be used in production!
20 define('REQUEST_TOKEN', constant('URL').'/oauth/requestToken');
21 define('AUTHORIZE_TOKEN', constant('URL').'/oauth/authorize');
22 define('ACCESS_TOKEN', constant('URL').'/oauth/accessToken');
23 define('REQUEST_URI', constant('URL').'/xml');
24
25 define('BASE_URL', 'http://'.$_SERVER["SERVER_NAME"].dirname($_SERVER['REQUEST_URI']));
26
27 define('TELLSTICK_TURNON', 1);
28 define('TELLSTICK_TURNOFF', 2);
29
```

Upload the files to the server and login



After login, open the index.php file that are included in the Live API files
(This API files should be in a own directory)





If you get something like this error:

Warning: Cannot modify header information - headers already sent by (output started at /home/roberkwk/public_html/app/telldus/HTTP/Request2.php:2101)
in /home/roberkwk/public_html/app/telldus/getAccessToken.php on line 13

Add following to the files that have the error-message (color marked above):

`ob_start();`

Like this:



```
1 <?php
2 ob_start();
3
4 require_once 'common.php';
5
6 $consumer = new HTTP_OAuth_Consumer(constant('PUBLIC_KEY'), constant('PRIVATE_KEY'));
7
8 $consumer->getRequestToken(constant('REQUEST_TOKEN'), constant('BASE_URL').'/getAccessToken.php');
9
10 $_SESSION['token'] = $consumer->getToken();
11 $_SESSION['tokenSecret'] = $consumer->getTokenSecret();
12
13 $url = $consumer->getAuthorizeUrl(constant('AUTHORIZE_TOKEN'));
14 header('Location:'.$url);
15
16 ?>
```

Recieve sensor data without login

To get temp-data without login, look at this forum-thread:

<http://www.telldus.com/forum/viewtopic.php?f=8&t=2174#p14244>

Demo-code by kea in forum:

```
<?php

session_start();
require_once 'HTTP/OAuth/Consumer.php';

// Keys for demo user from http://api.telldus.com/keys/showToken
define('PUBLIC_KEY', 'FEHUVW84RAFR5SP22RABURUPHAFRUNU');
define('PRIVATE_KEY', 'ZUXEVEGA9USTAZEWRETHAQUBUR69U6EF');
define('TOKEN', '0af6933e26366c308117c191784fb48e04eab090c');
define('TOKEN_SECRET', '6f206a2e194af3d4c3dc36e749edbd52');

define('URL', 'http://api.telldus.com'); //https should be used in
production!
define('REQUEST_TOKEN', constant('URL').'/oauth/requestToken');
define('AUTHORIZE_TOKEN', constant('URL').'/oauth/authorize');
define('ACCESS_TOKEN', constant('URL').'/oauth/accessToken');
define('REQUEST_URI', constant('URL').'/xml');

define('BASE_URL',
'http://'.$_SERVER['SERVER_NAME'].dirname($_SERVER['REQUEST_URI']));

$consumer = new HTTP_OAuth_Consumer(constant('PUBLIC_KEY'),
constant('PRIVATE_KEY'),
constant('TOKEN'), constant('TOKEN_SECRET'));

// Get temperature for sensor with ID 105473
// OBS: Demo sensor 105473 är en dummy sensor, därför ger den alltid samma
// värden och last updated någon gång år 2013!!!
$params = array('id'=> 105473);
$response = $consumer->sendRequest(constant('REQUEST_URI').'/sensor/info',
$params, 'GET');
echo( htmlentities($response->getBody()));

?>
```