

HybridAuth – Login con redes sociales en Codeigniter


septiembre 18

2016

Integrante: María Fernanda Segovia Chacón

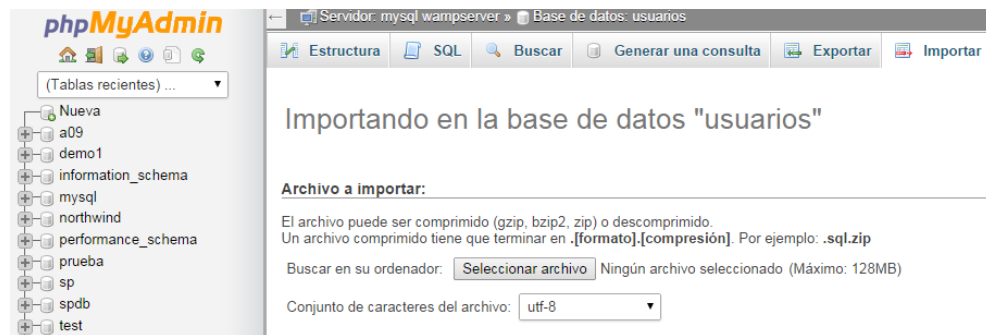


1. Requerimientos:

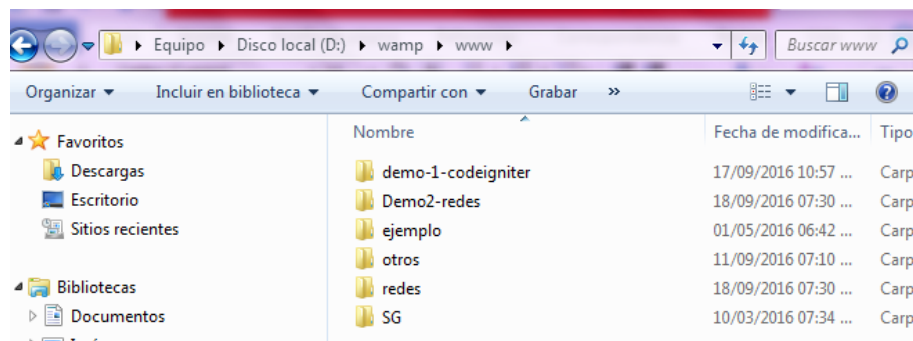
- Contar con un servidor WAMP o XAMP.  Haber descargado el proyecto de “demo-1-codeigniter”
- Descargar HybridAuth Plugins para CodeIgniter en <http://hybridauth.sourceforge.net/plugins.html>

2. Desplegar el Proyecto:

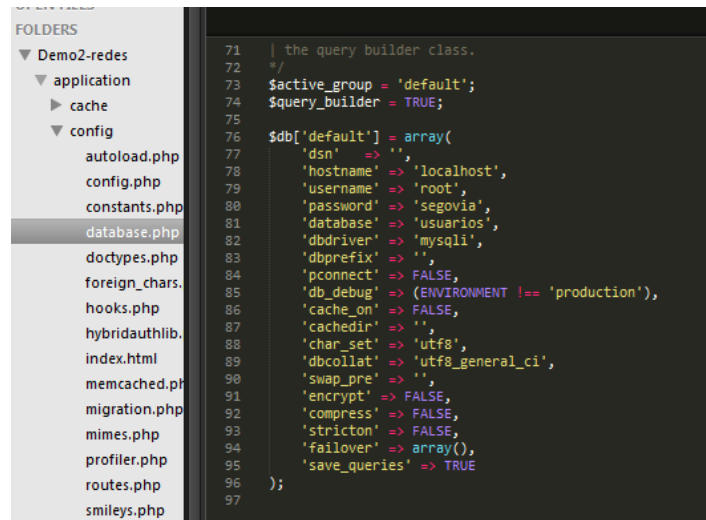
1. En los Archivos descargados de “demo2-redes” existe un archivo llamado demo2.sql, el cual deberán importar a su base de datos Mysql en este caso estoy utilizando phpMyAdmin.



2. El proyecto que se ha descargado se tiene que poner en la carpeta www de wamp server, en mi caso es D:\wamp\www, en xampp la dirección sería c:\xampp\htdocs\.

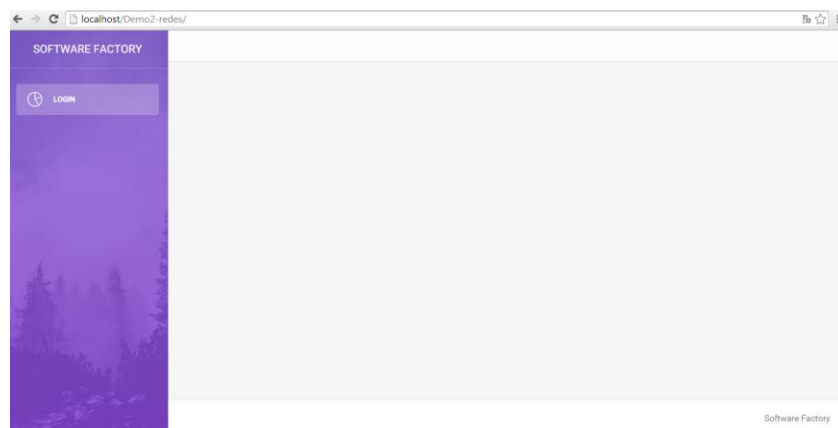


3. Luego de haber incluido la carpeta podemos ingresar al proyecto por medio de sublimetext para cambiar la conexión a la base de datos. Ingresamos a “Demo2-redes/application/config/database.php” y aquí cambiaremos los parámetros de `$db['default']['username'] = 'root', $db['default']['password'] = ''`, `$db['default']['database'] = ''` por el usuario, contraseña y nombre de nuestra base de datos. En mi caso mi conexión a mi base de datos tiene contraseña, de no tenerla dejarla vacía.

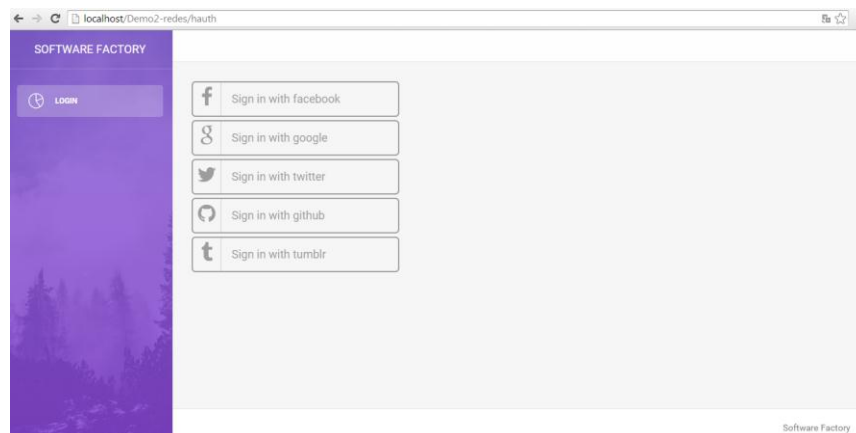


```
71 | the query builder class.
72 */
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => 'segovia',
81     'database' => 'usuarios',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97
```

4. Luego podemos acceder a la página desde nuestro buscador y podremos ver la página web mediante el link “<http://localhost/Demo2-redes/>”.



5. Al ingresar a la opción de login, podemos ver las opciones para identificarnos mediante algunas redes sociales.

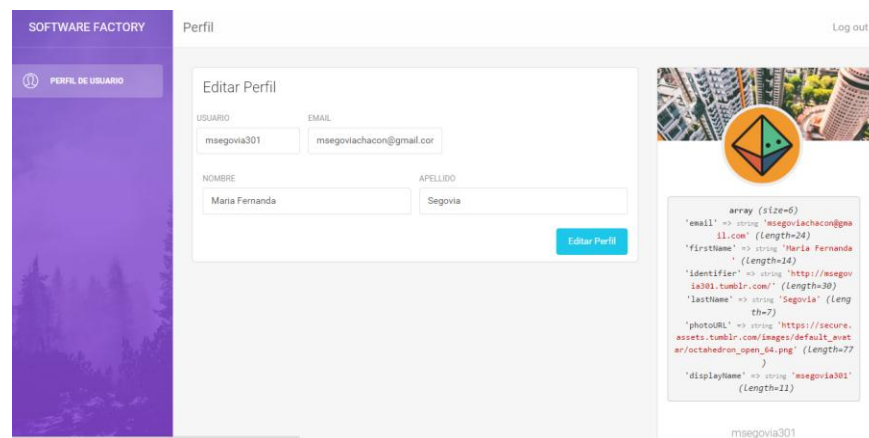




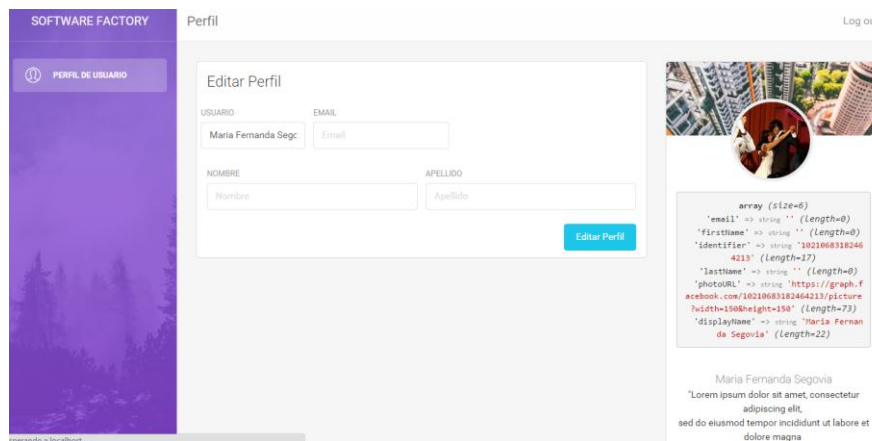
6. Por ejemplo al autenticarnos nos pide los permisos para la aplicación (Ejemplo: tumblr)



7. Se guardan los datos en la base de datos y se muestra la información del usuario.



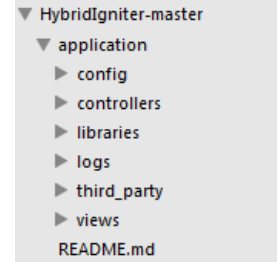
8. Igual con las demás redes Sociales





3. MVC CodeIgniter – HybridAuth

De la carpeta HybridAuth que descargamos los archivos que están dentro se copian según la estructura. Los que están en controllers se copian en controllers de nuestro proyecto.



De todos los archivos el hybridauthlib.php que se encuentra en application/config/ contiene los arrays de las claves de accesos de las redes sociales. Cuando se cree una aplicación en una red social las clases se ingresan aquí.

```
"Facebook" => array (
    "enabled" => true,
    "keys" => array ( "id" => "1026594830788708", "secret" => "c8f6b988b0f638ec44dd3bb146c62f44" ),
    "scope" => "email, public_profile"
),

"Twitter" => array (
    "enabled" => true,
    "keys" => array ( "key" => "BmkJcP6eCrikAnBZVviTKOI3G", "secret" => "34N2u7r1HZAjfb10hf1PFECmuqfzVwhMOGifOWZCHznUJ4U7am" )
),

// windows live
"Live" => array (
),

"MySpace" => array (
),

"LinkedIn" => array (
),

"Foursquare" => array (
),

"Github" => array (
    "enabled" => true,
    "keys" => array ( "id" => "6d9f79f7dea578d0d2bf", "secret" => "47d9d6d20a73de2a7bb45e2568240ddb77bf1dc2" )
),

"Tumblr" => array (
    "enabled" => true,
    "keys" => array ( "key" => "AFKU48GkbP3cgUSP33tXCqThH4298YQS1TKkXTnrOwXP8J1uz", "secret" => "ZTViizD0c1sq5IX6wQaruU8ddhMTDTv885NF6z19Q2hTQ" )
)
```

En las vistas para ingresar un botón para autenticar a un usuario se llama de la siguiente manera.

```
<div class="content">
    <div class="container-fluid">
        <div class="row">
            <div class="col-md-4">
                <a class="btn btn-block btn-social btn-lg btn-facebook" href="php echo base_url() ?&gt;hauth/login/Facebook"&gt;
                    &lt;span class="fa fa-facebook"&gt;&lt;/span&gt; Sign in with facebook
                &lt;/a&gt;
                &lt;a class="btn btn-block btn-social btn-lg btn-google" href="<?php echo base_url() ?&gt;hauth/login/Google"&gt;
                    &lt;span class="fa fa-google"&gt;&lt;/span&gt; Sign in with google
                &lt;/a&gt;
                &lt;a class="btn btn-block btn-social btn-lg btn-twitter" href="<?php echo base_url() ?&gt;hauth/login/Twitter"&gt;
                    &lt;span class="fa fa-twitter"&gt;&lt;/span&gt; Sign in with twitter
                &lt;/a&gt;
                &lt;a class="btn btn-block btn-social btn-lg btn-github" href="<?php echo base_url() ?&gt;hauth/login/Github"&gt;
                    &lt;span class="fa fa-github"&gt;&lt;/span&gt; Sign in with github
                &lt;/a&gt;
                &lt;a class="btn btn-block btn-social btn-lg btn-tumblr" href="<?php echo base_url() ?&gt;hauth/login/Tumblr"&gt;
                    &lt;span class="fa fa-tumblr"&gt;&lt;/span&gt; Sign in with tumblr
                &lt;/a&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre

```

Los usuarios se guardan en la base de datos mediante el controlador hauth.php

```
public function login($provider)
{
    log_message('debug', "controllers.HAuth.login($provider) called");

    try
    {
        log_message('debug', 'controllers.HAuth.login: loading HybridAuthLib');
        $this->load->library('HybridAuthLib');

        if ($this->hybridauthlib->providerEnabled($provider))
        {
            log_message('debug', "controllers.HAuth.login: service $provider enabled, trying to authenticate.");
            $service = $this->hybridauthlib->authenticate($provider);

            if ($service->isUserConnected())
            {
                log_message('debug', 'controller.HAuth.login: user authenticated.');
                $user_profile = $service->getUserProfile();

                log_message('info', 'controllers.HAuth.login: user profile:'.PHP_EOL.print_r($user_profile, TRUE));
                $data['user_profile'] = $user_profile;

                $datos=$this->usuario->agregar($user_profile->displayName, $user_profile->email,$user_profile->firstName,$user_profile->identifier,$user_

                $data['user_profile']=$datos[0];

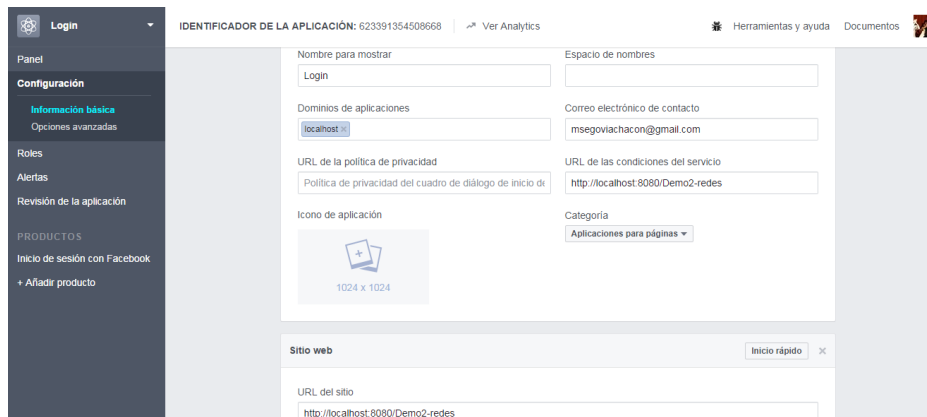
                $this->load->view('hauth/done',$data);
            }
            else // Cannot authenticate user
            {
                show_error('cannot authenticate user');
            }
        }
        else // This service is not enabled.
        {
            log_message('error', 'controllers.HAuth.login: This provider is not enabled ('.$provider.'');
            show_404($SERVER['REQUEST_URI']);
        }
    }
}
```

4. Crear aplicación en Facebook

1. Se ingresa <https://developers.facebook.com/>



2. Se ingresa la dirección de tu proyecto nombre, etc.





3. En la página principal estarán las claves para ingresar en el proyecto.

The screenshot shows the main page of an application. On the left is a dark sidebar with a 'Login' menu item. The main content area is titled 'Página principal' and contains a 'Login' section. This section includes a version number 'v2.7', an application identifier '623391354508668', and a secret key field with a 'Mostrar' button. The top of the page displays the application identifier and a 'Ver Analytics' link.

4. En otras redes sociales es igual por ejemplo en Tumblr

The screenshot shows the 'Registra tu aplicación' (Register your application) form on the Tumblr website. The form is located on the right side of a dark blue header. It contains several input fields for registration details: application name, website URL, App Store URL, Google Play Store URL, and a description (limited to 400 characters).