

## Login mechanism with codeigniter

Δημιουργήστε έναν login controller. Θα φορτώνει ένα view το οποίο θα εμφανίζει τη φόρμα.

Θα έχει επίσης της εξής μεθόδους:

- **signup** για να εμφανίζει τη φόρμα εγγραφής χρήστη
- **validate\_credentials** για να ελέγχει τα στοιχεία χρήστη και να του επιτρέπει το login
- **create\_member** για να εγγράφει έναν χρήστη στη βάση, αφού πρώτα κάνει validation τα στοιχεία της φόρμας εγγραφής
- **logout** για να καταστραφούν τα session data

### controller/login

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Login extends CI_Controller {

    public function index() {

        $this->load->view('header');

        $this->load->view('login_view');

        $this->load->view('footer');

    }

    public function validate_credentials() {

        $query = $this->member_model->validate();

        if ($query) {

            $data = array(

                'username' => $this->input->post('username'),

                'is_logged_in' => TRUE
```

```

        );

        $this->session->set_userdata($data);

        redirect('site/members');
    } else {

        $this->index();

    }
}

function signup() {

    $this->load->view('header');

    $this->load->view('signup_view');

    $this->load->view('footer');

}

function create_member() {

    $this->form_validation->set_rules('first_name', 'name', 'trim|required');

    $this->form_validation->set_rules('last_name', 'Last name', 'trim|required');

    $this->form_validation->set_rules('email', 'email',
'trim|required|valid_email');

    $this->form_validation->set_rules('username', 'Username',
'trim|required|min_length[4]');

    $this->form_validation->set_rules('password', 'Password',
'trim|required|min_length[4]|max_length[32]');

    $this->form_validation->set_rules('password2', 'Password Confirm',
'trim|required|min_length[4]|max_length[32]|matches[password]');

    if ($this->form_validation->run() == FALSE) { //validation error

        $this->load->view('header');

        $this->load->view('signup_view');

        $this->load->view('footer');

    } else { //store the data

```

```

        if ($query = $this->member_model->create_member()) {

            $this->load->view('header');

            $this->load->view('signup_success');

            $this->load->view('footer');

        } else {

            $this->load->view('header');

            $this->load->view('signup_view');

            $this->load->view('footer');

        }

    }

}

function logout(){

    $this->session->sess_destroy();

    $this->index();

}

}

```

Δημιουργήστε ένα login\_view το οποίο θα έχει τη φόρμα για να κάνει login ο χρήστης

### views/login\_view

```

<?php

echo form_open('login/validate_credentials');

```

```
echo form_input('username', 'Username');  
echo form_password('password', 'Password');  
echo form_submit('submit', 'Login');  
echo anchor('login/signup', 'Create account');  
?>
```

Δημιουργήστε ένα `signup_view` το οποίο θα έχει τη φόρμα για να κάνει εγγραφή ο χρήστης

Προκειμένου να μην συμπληρώνουμε τα πεδία κάθε φορά που κάποιο validation rule δεν τηρείται, χρησιμοποιούμε την [set\\_value\(\)](#)

### [views/signup\\_view](#)

```
<h1> Create account </h1>  
<?php  
echo form_open('login/create_member');  
echo form_input('first_name', set_value('first_name', 'First Name'));  
echo form_input('last_name', set_value('last_name', 'Last Name'));  
echo form_input('email', set_value('email', 'Email'));  
  
echo form_input('username', set_value('username', 'Username'));  
echo form_input('password', set_value('password', 'Password'));  
echo form_input('password2', set_value('password2', 'Password Confirm'));  
  
echo form_submit('submit', 'Create account');  
  
echo validation_errors();  
?>
```

Κάντε τον Login controller default controller στο config/routes.php

```
$route['default_controller'] = "login";
```

Δημιουργήστε τη βάση για τους χρήστες σας

```
CREATE TABLE IF NOT EXISTS `members` (  
  `id` int(5) NOT NULL AUTO_INCREMENT,  
  `username` varchar(65) COLLATE utf8_unicode_ci NOT NULL,  
  `password` varchar(32) COLLATE utf8_unicode_ci NOT NULL,  
  `email` varchar(65) COLLATE utf8_unicode_ci NOT NULL,  
  `first_name` varchar(65) COLLATE utf8_unicode_ci NOT NULL,  
  `last_name` varchar(65) COLLATE utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `username` (`username`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Δημιουργία του member\_model

Θα έχει 2 συναρτήσεις:

- **validate** όπου θα ελέγχει τα στοιχεία στη φόρμα Login
- **create\_member** όπου θα καταχωρεί στη βάση τον χρήστη

## models/member\_model

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Member_model extends CI_Model {

    function validate() {
        $this->db->where('username', $this->input->post('username'));
        $this->db->where('password', md5($this->input->post('password')));
        $query = $this->db->get('members');
        if ($query->num_rows() == 1) {
            return TRUE;
        }
    }

    function create_member() {
        $new_member = array(
            'first_name' => $this->input->post('first_name'),
            'last_name' => $this->input->post('last_name'),
            'email' => $this->input->post('email'),
            'username' => $this->input->post('username'),
            'password' => md5($this->input->post('first_name'))
        );

        $insert = $this->db->insert('members', $new_member);
        return $insert;
    }

}
```

Προσθέτουμε το member\_model στο \$autoload['model'] του config/autoload.php

```
$autoload['model'] = array('user_model', 'member_model');
```

Δημιουργούμε τη συνάρτηση **validate\_credentials()** στον login controller ώστε να ελέγχουμε τα δεδομένα που βάζει ο χρήστης στην login\_form.

Σε περίπτωση που υπάρχει ο χρήστης στη βάση, τότε τον ανακατευθύνουμε σε έναν άλλον controller τον site όπου εκεί θα έχουμε μια συνάρτηση members η οποία θα ανοίγει ένα member view το οποίο θα το βλέπουν μόνο οι διαπιστευμένοι χρήστες.

Δημιουργούμε τον controller με όνομα site ο οποίος θα έχει 2 συναρτήσεις + τον constructor.

- **members** η οποία θα ανοίγει το members\_view
- **is\_logged\_in** η οποία θα ελέγχει αν υπάρχει στο session μια μεταβλητή, η οποία ορίζεται μόλις κάνει Login ο χρήστης μέσω της validate\_credentials() του login controller. Εκ των προτέρων (στον constructor) καλούμε αυτή τη συνάρτηση ώστε να μην επιτραπεί η πρόσβαση σε κάποιον που δεν έχει κάνει login.

## controllers/site

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Site extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->is_logged_in();
    }

    function members() {
        $this->load->view('header');
        $this->load->view('members_view');
        $this->load->view('footer');
    }

    function is_logged_in() {
        $is_logged_in = $this->session->userdata('is_logged_in');
        if (!isset($is_logged_in) || $is_logged_in != TRUE) {
            echo 'You must login first!';
            echo anchor('login', 'Login');
            die();
        }
    }

}
```

Δημιουργούμε τη συνάρτηση signup στον login controller, η οποία θα ανοίγει μια signup φόρμα.

Δημιουργούμε στον login controller μια συνάρτηση create\_member η οποία θα πάρει τα δεδομένα από την signup\_view και θα τα εισάγει στη βάση, αφού πρώτα εφαρμόσει validation rules.

Δημιουργούμε συνάρτηση στο member\_model η οποία θα κάνει την εισαγωγή στη βάση και θα την καλέσει η login/create\_member

Δημιουργούμε το members\_view για να το εμφανίζουμε μόνο όταν κάνει login ένας χρήστης

### views/members\_view

```
<h1> This is members view only! </h1>

<?php
echo '<pre>';
var_dump($this->session->all_userdata());
echo '</pre>';
?>
```

Δημιουργούμε το signup\_success view ώστε όταν επιτυχώς εγγραφεί ένας χρήστης να ενημερωθεί και μπορέσει να κάνει login μετά

### views/signup\_success

```
<h1> successfull signup! </h1>
<?php
echo anchor('login', 'Login Now!');
?>
```



Για να έχουν πρόσβαση στο members\_view μόνο οι χρήστες που έχουν κάνει Login πρέπει πριν καλέσουμε αυτό το View να ελέγξουμε αν υπάρχουν συγκεκριμένα δεδομένα στο session.

## Υλικό



[Login mechanism with Codeigniter](#)

Simple login with codeigniter

<http://www.codefactorycr.com/login-with-codeigniter-php.html>

Αποστολή email



<http://net.tutsplus.com/tutorials/php/codeigniter-from-scratch-day-3/>

## AJAX

Δημιουργήστε ένα view ajax το οποίο να έχει ένα input και ένα textarea στοιχείο.

Σκοπός είναι ότι γράφετε στο input να καλείται μια συνάρτηση ενός controller μέσω ajax (main/search\_user) η οποία να επιστρέφει δεδομένα από τη βάση και να τα εμφανίζει στο textarea. Τα δεδομένα παίρνονται από τον controller καλώντας την getuser1 μέθοδο του user\_model. Προσέξτε ότι η search\_user επιστρέφει δεδομένα σε json μορφή και πρέπει να τα κάνουμε object και να πάρουμε τα πεδία που θέλουμε να εμφανίσουμε στο textarea

## views/ajax

```
<div class="something">
  <input name="search_data" id="search_data" type="text" width="60px"
onkeyup="ajaxSearch();">
</div>

<div class="somethingelse">
  <br/>
  <textarea id="area" name="area" cols="80" rows="7">
  </textarea>
</div>

<script type="text/javascript">
  function ajaxSearch() {
    var input_data = $('#search_data').val();
    document.getElementById('area').innerHTML = "";
    if (input_data.length === 0) {
    } else {

      var post_data = {
        'search_data': input_data
      };

      $.ajax({
        type: "POST",
        url: "<?php echo base_url(); ?>index.php/main/search_user/",
        data: post_data,
        success: function(data) {
          // return success
          if (data.length > 0) {
            var json = JSON.parse(data);
            console.log(data);
            var items = Object.keys(json);
            items.forEach(function(item) {
              console.log(item);
              if (item == 'msg') {
                $('#area').html('No data!');
                return;
              }
              $('#area').append(item + '=' + ' username: ' +
json[item].username + ' password : ' + json[item].email + ' id : ' + json[item].id +
'\n');
            });
          }
        }
      });
    }
  }
});
```

```
    }  
}  
  
</script>
```

## controllers/main

```
<?php  
  
if (!defined('BASEPATH'))  
    exit('No direct script access allowed');  
  
class Main extends CI_Controller {  
  
    public function index() {  
        $this->load->view('header');  
        $this->load->view('login_form');  
        $this->load->view('footer');  
    }  
  
    public function check() {  
  
        $this->form_validation->set_rules('username', 'username', 'required');  
        $this->form_validation->set_rules('password', 'password', 'required');  
  
        if ($this->form_validation->run() == FALSE) {  
            $this->load->view('header');  
            $this->load->view('login_form');  
            $this->load->view('footer');  
        } else {  
            $this->load->view('header');  
            $this->load->view('form_success');  
            $this->load->view('footer');  
        }  
        // echo $this->input->post('username');  
    }  
  
    public function get_users() {  
        $data['users'] = $this->user_model->getusersinfo();  
        $this->load->view('header');  
        $this->load->view('users', $data);  
        $this->load->view('footer');  
    }  
  
    public function get_user() {
```

```

$this->form_validation->set_rules('name', 'name', 'required');

if ($this->form_validation->run() == FALSE) {
    $this->load->view('header');
    $this->load->view('user');
    $this->load->view('footer');
} else {
    $username = $this->input->post('name');
    $this->session->set_userdata('name', $username);

    $data['user'] = $this->user_model->getuser($username);
    $this->load->view('header');
    $this->load->view('user', $data);
    $this->load->view('footer');
}
}

public function search_user() {
    $search_data = $this->input->post('search_data');
    $res = $this->user_model->getuser1($search_data);
    if ($res) {
        echo json_encode($res);
    } else {
        $error = array('msg' => "No data found");
        echo json_encode($error);
    }
}

public function ajaxsearch() {
    $this->load->view('header');
    $this->load->view('ajax');
    $this->load->view('footer');
}

public function login() {

    $res = $this->user_model->check_login('Akis', 'dsds');
    var_dump($res);
    if ($res) {
        $this->load->view('header');
        $this->load->view('user');
        $this->load->view('footer');
    }
    else {
        $this->load->view('header');
        $this->load->view('ajax');
        $this->load->view('footer');
    }
}
}

```

## models/user\_model

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class User_model extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    public function getusersinfo() {

        $query = $this->db->get('users');

        if ($query->num_rows() > 0) {
            $data = $query->result_array();
            return $data;
        } else {
            return FALSE;
        }
    }

    public function getuser($username) {
        $query = $this->db->get_where('users', array('username' => $username));
        // log_message('info', 'SQL '. $this->db->last_query());

        if ($query->num_rows() > 0) {
            $data = $query->result_array();
            return $data;
        } else {
            return FALSE;
        }
    }

    public function getuser1($username) {
        $this->db->like('username', $username);
        $query = $this->db->get('users');
        if ($query->num_rows() > 0) {
            $data = $query->result_array();
            return $data;
        } else {
```

```
        return FALSE;
    }
}
}
```

## Email

Δημιουργήστε έναν email controller και καλέστε τον. Υποθέτουμε ότι χρησιμοποιούμε το gmail για να στείλει το mail.

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Email extends CI_Controller {

    public function __construct() {
        parent::__construct();
    }

    function index() {

        $config = array(
            'protocol' => 'smtp',
            'smtp_host' => 'ssl://smtp.gmail.com',
            'smtp_port' => 465,
            'smtp_user' => '-----USERNAME----',
            'smtp_pass' => '--PASSWORD--',
            'smtp_timeout' => '60',
            'starttls' => 'TRUE'
        );

        $this->load->library('email', $config);
        $this->email->set_newline("\r\n");
        $this->email->from('user@gmail.com');
        $this->email->to('user@hua.gr');
```

```
$this->email->subject('test email from codeigniter');  
$this->email->message('this is the message!');  
  
if ($this->email->send()){  
    echo $this->email->print_debugger();  
    echo 'Email sent!';  
}  
else{  
    show_error($this->email->print_debugger());  
    echo $this->email->print_debugger();  
}  
}  
}
```

Ο κώδικας για να τον κατεβάσετε [εδώ](#)