

PHPD PROFILE

ZHOU Dian 288123 & GUO Zhuyan 288120



catalogue

.....	1
I Environment.....	2
Download the CodeIgniter framework.....	2
II CodeIgniter.....	2
III development.....	3
Design database.....	3
1、 Registered.....	4
2、 login.....	5
3、 The cancellation.....	7
4、 To-do list management function.....	7
5、 Good friend management.....	9
6、 The theme.....	11
IV Testing and validation.....	11

I Environment

To deploy and install the environment in the windows10 environment, we use the phpstudy integrated environment (including PHP, apache, mysql). Please refer to readme.docx for detailed configuration steps.

This time, we wrote a proxy event listing system for registered users, which was developed using codeIgniter framework.

Download the CodeIgniter framework

<https://www.codeigniter.com/> Codeigniter version 3.1.9 is available from codeigniter's official website. Deploy the framework to the project directory. Follow the framework tutorial to ensure that the framework is properly accessed.

【This is the codeigniter3 Chinese document : <https://codeigniter.org.cn/userguide3/> 】

II CodeIgniter

The framework for MVC structure, provides basic functions, to help users to develop quickly. Follow the framework tutorial to get familiar with the framework directory structure and role, understand the use of controllers, views, database calls, routing, etc.

The model is mainly used to manage the data of the application and obtain the data according to the special business rules of the application.

A view is a simple file with little or no logic that is only responsible for presenting the data to the user.

The controller is the glue code that processes and consolidates data back and forth between the view layer and the data store.

Create the.htaccess file in the project root.

RewriteEngine on

RewriteCond \$1 !^(index\.php|assets|images|js|css|uploads|favicon.png)

RewriteCond %(REQUEST_FILENAME) !-f

RewriteCond %(REQUEST_FILENAME) !-d

RewriteRule ^(.*)\$./index.php/\$1 [L]

The initial configuration: **/application/Config/Autoload.php**

Configurable common functions can be loaded,

class libraries: \$autoload['libraries'] = array('email','form_validation','session','pagination');

Validation helper: \$autoload['helper'] = array('url','form','text');

Load model: `$autoload['model'] = array('friend_model','list_model','user_model');`

Basic configuration: **/application/Config/config.php**

Setting baseurl, `$config['base_url'] = 'http://'.$_SERVER['HTTP_HOST'];`

Set session save path。 `$config['sess_save_path'] = BASEPATH..'../session_file';`

Database configuration: **/application/Config/database.php** Set the database name, username, and password.

The most important configuration in/application/Config/routes. The PHP file and set the routing access path, if not set, controllers and views can not be connect, can not be normal access URL.

III development

Design database

用户表

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `username` varchar(255) NOT NULL,  
  `password` varchar(255) NOT NULL,  
  `register_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8;
```

好友申请表

```
CREATE TABLE `apply` (  
  `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '自增 id',  
  `uid` int(11) NOT NULL COMMENT '被申请用户 id userid',  
  `sid` int(11) NOT NULL COMMENT '好友申请人 id',  
  `status` tinyint(1) NOT NULL DEFAULT '0',  
  `text` varchar(255) DEFAULT NULL COMMENT '附言',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

待办管理表

```
CREATE TABLE `todos` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `todo` text COLLATE utf8_unicode_ci,
```

```

`user_id` int(10) unsigned NOT NULL,
`complete` tinyint(1) NOT NULL DEFAULT '0',
`created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
`updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8
COLLATE=utf8_unicode_ci;

```

1、Registered

Public page creation: in/Application /views/ folder, create public template file Templates folder, design registration and login public page header.php and footer.php.

Create registration page: in/Application /views/ folder, create user folder, in this folder create register.php file. Fill in the input fields of user name, email, password and confirm password on the page, and write the page and style.

Controller creation: /application/controllers/ Register.php ,

```

public function index()
{
    $data['title'] = '注册';

    $this->form_validation->set_rules('username', 'Username', 'required');
    //      $this->form_validation->set_rules('username',      'Username',
'required|callback_check_username_exists');
    $this->form_validation->set_rules('email', 'Email', 'required');
    $this->form_validation->set_rules('password', 'Password', 'required');
    $this->form_validation->set_rules('password_confirmation',      'Confirm
Password', 'matches[password]');
    // print_r($this->input->post('password'));exit;
    if($this->form_validation->run() === FALSE){
        $this->load->view('templates/header');
        $this->load->view('user/register', $data);
        $this->load->view('templates/footer');
    } else {
        // Encrypt password
        $enc_password = md5($this->input->post('password'));
    }
}

```

```

        $this->user_model->register($enc_password);

        // Set message
        $this->session->set_flashdata('user_registered', '注册成功请登录');

        redirect('home');
    }
}

```

Create model: `/application/models/ User_model.php` ,

```

public function register($enc_password){
    // User data array
    $data = array(
        'name' => $this->input->post('username'),
        'username' => $this->input->post('username'),
        'email' => $this->input->post('email'),
        'username' => $this->input->post('username'),
        'password' => $enc_password
    );

    // Insert user
    return $this->db->insert('users', $data);
}

```

Finally, configure the routing information in the Train.php file.

2、login

Create a registration page: in `/application/views/`, Create a folder named User, Create the register.php file in this folder. Fill in the input fields of user name, email, password and confirm password on the page, and write the page and style.

Controller creation: `/application/controllers/ Login.php` ,

Write login controller

```

public function index()
{
    $data['title'] = '登录';

    $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('templates/header');
        $this->load->view('user/login', $data);
    }
}

```

```

        $this->load->view('templates/footer');
    } else {

        // Get username
        $username = $this->input->post('username');
        // Get and encrypt the password
        $password = md5($this->input->post('password'));

        // Login user
        $user_id = $this->user_model->login($username, $password);

        if($user_id){
            // Create session
            $user_data = array(
                'user_id' => $user_id,
                'username' => $username,
                'logged_in' => true
            );

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

            // Set message
            $this->session->set_flashdata('user_loggedin', '登录成功');

            redirect('home');
        } else {
            // Set message
            $this->session->set_flashdata('login_failed', '登录失败');

            redirect('login');
        }
    }
}

```

In user_model.php, write the login database model.

```

// Log user in
public function login($username, $password){
    // Validate
    $this->db->where('username', $username);
    $this->db->where('password', $password);

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

    if($result->num_rows() == 1){

```

```

        return $result->row(0)->id;
    } else {
        return false;
    }
}

```

Finally, configure the routing information in the Train.php file.

3、 The cancellation

Controller creation: **/application/controllers/ User.php** ,

Write user controller and logout method

// Log user out

```

public function logout(){
    // Unset user data
    $this->session->unset_userdata('logged_in');
    $this->session->unset_userdata('user_id');
    $this->session->unset_userdata('username');

    // Set message
    $this->session->set_flashdata('user_loggedout', '你已登出');

    redirect('login');
}

```

Finally, configure the routing information in the Train.php file.

4、 To-do list management function

Public page creation: in/Application /views/ folder, create the common folder of public template files, design the registration and login public page menu.php and bottom.php.

Todo page creation: in the /application/views/ folder, create lists folder, in this folder create home.php file. And write the page and style of to-do list.

Controller creation: **/application/controllers/ Lists.php** ,

Write a backlog controller in which state modification USES Ajax methods to switch states.

```

public function home()
{
    // Check login

```



```

        if(!$this->session->userdata('logged_in')){
            redirect('login');
        }
        $offset = 0;
        // Pagination Config
        $config['base_url'] = base_url() . 'home';
        $config['total_rows'] = $this->db->count_all('todos');
        $config['per_page'] = 20;
        $config['uri_segment'] = 3;
        $config['attributes'] = array('class' => 'pagination-link');

        // Init Pagination
        $this->pagination->initialize($config);

        $data['title'] = '待办列表';

        $data['todos'] = $this->list_model->get_lists(FALSE, $config['per_page'],
$offset);

        $this->load->view('common/menu');
        $this->load->view('lists/home', $data);
        $this->load->view('common/bottom');
    }

```

Create the model: `/application/models/ List_model.php` ,

Write the backlog model.

```

public function get_lists($slug = FALSE, $limit = FALSE, $offset = FALSE){
    if($limit){
        $this->db->limit($limit, $offset);
    }

    $user_id = $this->session->userdata('user_id');

    if($slug === FALSE){
        $this->db->order_by('id', 'ASC');
        $query = $this->db->get_where('todos', array('user_id' => $user_id));
        return $query->result_array();
    }
    $query = $this->db->get_where('todos', array('user_id' => $user_id));
    return $query->row_array();
}

public function create_list(){

```

```

        $data = array(
            'name' => $this->input->post('name'),
            'todo' => $this->input->post('todo'),
            'user_id' => $this->session->userdata('user_id'),
            'complete' => '0'
        );

        return $this->db->insert('todos', $data);
    }

    public function delete_list($id){
        $this->db->where('id', $id);
        $this->db->delete('todos');
        return true;
    }
}

```

Finally, configure the routing information in the Train.php file.

5、 Good friend management

Friend page creation: in the /application/views/ folder, create the friend folder, in this folder create the index.php file. And write the friend list page and style.

Controller: /application/controllers/ Friend.php

Write a friend management controller

```

public function index(){
    // Check login
    if(!$this->session->userdata('logged_in')){
        redirect('login');
    }
    $offset = 0;
    // Pagination Config
    $config['base_url'] = base_url() . 'home';
    $config['total_rows'] = $this->db->count_all('apply');
    $config['per_page'] = 20;
    $config['uri_segment'] = 3;
    $config['attributes'] = array('class' => 'pagination-link');

    // Init Pagination
}

```

```

$this->pagination->initialize($config);

$data['title'] = '好友列表';

$data['friends'] = $this->friend_model->get_lists(FALSE, $config['per_page'],
$offset);

$this->load->view('common/menu');
$this->load->view('friend/index', $data);
$this->load->view('common/bottom');
}

public function find(){
    $data['title'] = '好友列表';
    // Check login
    if(!$this->session->userdata('logged_in')){
        redirect('login');
    }
    $status = intval($this->input->post('status'));
    $data['friends'] = $this->friend_model->get_list_by_status($status);

    $this->load->view('common/menu');
    $this->load->view('friend/index', $data);
    $this->load->view('common/bottom');
}

public function add (){
    // POST data
    $id = $this->input->post('id');
    $result = $this->friend_model->create_list($id);

    if($result){
        $data = array('code'=>'1', 'message' => '添加成功');
    }else{
        $data = array('code'=>'0', 'message' => '添加失败');
    }
    echo json_encode($data);
}

```

Create the model: `/application/models/ Friend_model.php` ,

```

public function get_lists($slug = FALSE, $limit = FALSE, $offset = FALSE){
    if($limit){
        $this->db->limit($limit, $offset);
    }
}

```

```

    }

    $user_id = $this->session->userdata('user_id');

    if($slug === FALSE){
        $sql = "select  a.id as id, sid as uid, name, email, status from apply a LEFT
JOIN users u on a.sid = u.id where a.uid='$user_id'";
        $query = $this->db->query($sql);
        return $query->result_array();
    }
    $query = $this->db->get_where('friend', array('user_id' => $user_id));
    return $query->row_array();
}

public function create_list(){
    $data = array(
        'sid' => $this->input->post('id'),
        'status' => '1',
        'uid' => $this->session->userdata('user_id')
    );

    return $this->db->insert('apply', $data);
}

public function delete_list($id){
    $this->db->where('id', $id);
    $this->db->delete('apply');
    return true;
}

```

Finally, configure the routing information in the Train.php file.

6、The theme

Click on a different theme on the page, use Ajax events, change the page style and color, and store the theme information through localStorage.

IV Testing and validation

You can use Chrome browser by clicking on the website you have created from the PhpStudy "Websites" TAB, right-clicking on the page, or typing `localhost` in your browser. Check the console Network and console for page execution.