

support.php

12/9/14, 12:00 PM

```
<?php
class Support extends MY_Controller {
    public $message_limit = null;

    public function __construct() {
        parent::__construct();
        if ($this->userid != $this->admin && !in_array($this->userid, $this->
            support_admin) && $this->uri->segment(2) != 'landing' && $this->uri->
            segment(2) != 'login') {
            die('no access');
        }
        $this->message_limit = 70;
    }

    function all_images() {

        $items = $this->db
            ->select('items.image_id, items.id')
            ->from('items')
            ->join('users', 'users.id = items.user_id')
            ->where('items.image_id !=', '')
            ->where('items.quantity >', 0)
            ->where('items.stealth', 0)
            ->where('users.active', 1)
            ->where('users.stealth_mode', 0)
            ->where('users.role', 1)
            ->group_by('image_id')
            ->get()->result();

        foreach ($items as $item) {

            echo $this->extras->image($item->image_id, 2).br();
            echo $item->id.br(2);

        }

    }

    function landing() {

        if ($this->user_info->id) exit('already logged in');

        echo form_open('support/login');
        echo form_input('user');
        echo form_password('pass');
        echo form_submit('submit', 'go');
        echo form_close();

    }

    function login() {

        if ($this->user_info->id) exit('already logged in');
```



```

$this->load->library('hasher');

$pass = $this->input->post('pass');
$username = $this->input->post('user');
$user_id = $this->market_model->single_cell('users', 'id', 'user',
    $username);

if ($user_id != $this->admin && !in_array($user_id, $this->support_admin)
    ) exit('no access');

$user = $this->db->from('users')->where('id', $user_id)->get()->row();
$db_pass = $user->pass;
$usalt = $user->usalt;
$hash_pass = $this->hasher->hash_pass($pass, $usalt);

if ($db_pass == $hash_pass) {

    $this->ma_session->set_data('user_id', $user_id);
    $this->ma_session->set_data('token', rand());

    if ($user_id == $this->admin) redirect('mastermind');

    redirect('support');

} else {

    exit('invalid login');

}

}

function index() {

    $now = time();
    $one_month_ago = $now - 30*24*3600;
    $two_days_ago = $now - 2*24*3600;
    $three_days_ago = $now - 3*24*3600;
    $one_day_ago = $now - 24*3600;

    # customer support data
    $data['customer_message_count'] = $this->db
        ->from('messages')
        ->where('read', 0)
        ->where('to', $this->support_id)
        ->get()->num_rows();
    $customer_message_freshness = $this->db
        ->from('messages')
        ->where('read', 0)
        ->where('to', $this->support_id)
        ->order_by('created', 'asc')
        ->limit(1)
        ->get()->row()->created;
    $data['customer_message_freshness'] = $this->extras->display_freshness
        ($customer_message_freshness, 'short', 2);
    $data['cs_in_3d'] = $this->db

```

```
->from('messages')
->where('to', $this->support_id)
->where('created >', $three_days_ago)
->get()->num_rows();

$data['cs_in_1d'] = $this->db
->select('from')
->from('messages')
->where('to', $this->support_id)
->where('created >', $one_day_ago)
->distinct()
->get()->num_rows();

# vendor support data
$data['vendor_message_count'] = $this->db
->select('from')
->from('messages')
->where('read', 0)
->where('to', $this->vendor_support_id)
->distinct()
->get()->num_rows();

$vendor_message_freshness = $this->db
->from('messages')
->where('read', 0)
->where('to', $this->vendor_support_id)
->order_by('created', 'asc')
->get()->row()->created;

$data['vendor_message_freshness'] = $this->extras->display_freshness
($vendor_message_freshness, 'short', 2);

$data['vs_in_3d'] = $this->db
->select('from')
->from('messages')
->where('to', $this->vendor_support_id)
->where('created >', $three_days_ago)
->distinct()
->get()->num_rows();

$data['vs_in_1d'] = $this->db
->select('from')
->from('messages')
->where('to', $this->vendor_support_id)
->where('created >', $one_day_ago)
->distinct()
->get()->num_rows();

# resolution data
$resolutions = $this->db
->from('transactions')
->where(
    array(
        'action' => 5,
        'finalized' => 0,
        'canceled' => 0
    )
)
->order_by('created', 'desc')
```

```
->get()->result();

$count = 0;
$resolution_freshness = $now;
foreach ($resolutions as $resolution) {
    $due = $resolution->dispute_opened + $resolution->dispute_duration;
    if ($now > $due) {
        $count++;
        if ($due < $resolution_freshness) $resolution_freshness = $due;
    }
}
$data['resolution_count'] = $count;
if ($resolution_freshness < $now) $data['resolution_freshness'] = $this->
    extras->display_freshness($resolution_freshness, 'short', 2);
$data['reso_out_3d'] = $this->db
    ->from('resolutions')
    ->where('proposer', "0")
    ->where('created >', $three_days_ago)
    ->count_all_results();
$data['reso_out_1d'] = $this->db
    ->from('resolutions')
    ->where('proposer', "0")
    ->where('created >', $one_day_ago)
    ->count_all_results();

$data['user_flags'] = ceil($this->db->query("SELECT SUM(weight) as w FROM
    flags WHERE type = 'user' GROUP BY type_id ORDER BY w DESC")->row()->
    w);
$data['item_flags'] = ceil($this->db->query("SELECT SUM(weight) as w FROM
    flags WHERE type = 'item' GROUP BY type_id ORDER BY w DESC")->row()->
    w);

$data['withdrawal_switch_state'] = ($this->extras->get_var
    ('withdrawal_switch') ? 'on' : 'off');

$data['view'] = 'support/support';
$this->load->view('support/template', $data);

}

}

# end
```