Инструкция по эксплуатации программного обеспечения "Модуль ID"

Оглавление

1. Описание и стек технологий

Сервис представляет собой скачиваемый архив с необходимым набором файлов.

Версии ПО:

- Nginx 1.10.3
- Apache 2.4
- PHP 7.0.33
- Percona Server MySQL 5.7.32
- Конфигурации подключения к БД и специфичные секреты сервиса хранятся в глобальных инклудах по пути /etc/php/7.0/include/

2. Запуск ПО Модуль ID

Для запуска достаточно перенести файлы из архива “module.zip” в заранее настроенное окружение и создать необходимые таблицы в БД.
Для работы потребуется создать таблицы в БД MySQL из файла DDL_tables.md, который находится в том же архиве. Для вашей системы измените настройки в файле settings.mysql.php согласно требуемым вам настройкам. Для вынесения из кода проекта константы соли паролей следует изменить файл privatesalt.php и вынести в переменные окружения. После этого следует указать путь до этого файла в файле \inc\userprint.class.inc.

3. Функции модуля

3.1. Регистрация

Параметры:

- скрипт: "ajax/register.php"

Параметры POST:

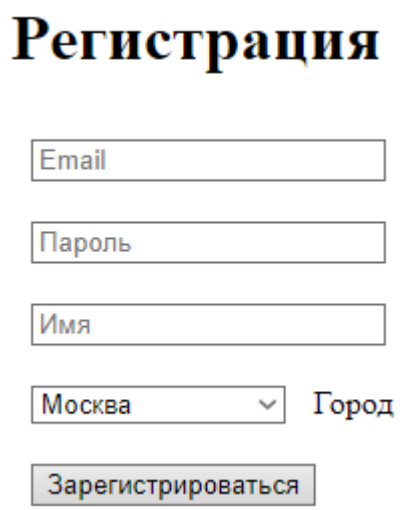
- email - почта
- password - пароль
- login - логин (имя)

Пример обращения с формы на HTML-странице:

```
<form class="auth-modal__form js-auth-modal__form-register"
action="ajax/register.php" method="post">
     <div class="auth-modal__row ui-widget">
          <input id="auth-email" class="auth-modal__input
auth-modal__input--email js-auth-modal__input
js-register-w-email-email" type="email" name="email"
maxlength="40" autocomplete="off" placeholder="Email" required>
     </div>
     <div class="auth-modal__row">
          <input id="auth-register-password"
class="auth-modal__input auth-modal__input--password
js-auth-modal__input js-register-w-email-password" type="password"
name="password" maxlength="30" autocomplete="off"
placeholder="Пароль" required>
     </div>
     <div class="auth-modal__row">
          <input id="auth-name" class="auth-modal__input
js-auth-modal__input js-register-w-email-name" type="text"
name="login" maxlength="35" placeholder="Имя" required>
     </div>
     <div class="auth-modal__row">
          <select id="loc_id3" name="loc_id3">
               <option value="4400">Москва</option>
               <option value="4962">Санкт-Петербург</option>
          </select>
          <label class="auth-modal__input-label"
for="loc_id3">Город</label>
     </div>
     <div class="auth-modal__inline">
          <div class="btns btns--center">
               <?php echo '<input type="hidden" name="csrf"
value="' . $token . '" >' ?>
```

```
                <button class="btn btn--small auth-modal__btn
auth-modal__btn--registration
js-register-w-email-submit">Зарегистрироваться</button>
            </div>
      </div>
</form>
```

Пример реализации:



3.2. Авторизация

Параметры:

- скрипт: "auth.controller.inc/login"

Параметры POST:

- login - логин
- password - пароль
- remember - “Запомнить меня” опционально.
- return - адрес переадресации после авторизации

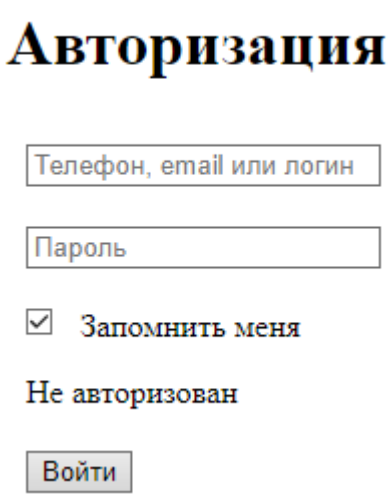
Группа тестовой среды: Первая

Пример обращения с формы на HTML-странице:

```
<form class="auth-login__form js-page-auth-modal__form-login"
action="auth/login" method="post">
      <input type="hidden" name="return" value="">
      <div class="auth-modal__row">
            <input id="auth-login-page"
```

```
                    class="auth-modal__input
auth-modal__input--email js-page-auth-modal__input
js-page-login-login"
                    name="login"
                    maxlength="40"
                    placeholder="Телефон, email или логин"
                    title=""
                    required
                    >
          <div class="alert js-page-alert-login
js-page-login-error-login"></div>
     </div>
     <div class="auth-modal__row">
          <input id="auth-password-page"
                    class="auth-modal__input
auth-modal__input--password js-page-auth-modal__input
js-page-login-password"
                    name="password"
                    type="password"
                    maxlength="30"
                    placeholder="Пароль"
                    autocomplete="off"
                    required
                    >
          <div class="auth-modal__fade"></div>
          <div class="alert js-page-alert-password
js-page-login-error-password"></div>
     </div>
     <div class="auth-login__remember">
          <label>
               <input class="js-page-login-remember
auth-login__input-remember hide" type="checkbox" name="remember"
checked>
               <span class="auth-login__icon-remember"></span>
               <span style="display: inline-block;
vertical-align: middle;">Запомнить меня</span>
          </label>
     </div>
     <?php echo $User instanceof User ? "<p>Авторизован, user_id =
{$User->user_id}, имя = {$User->name}, email = {$User->email}</p>"
: '<p>Не авторизован</p>' ?>
</form>
```

Пример реализации:



3.3. Проверка авторизации

Параметры:

- скрипт: "user.class.inc/getUserId"

В коде проекта вызывается метод:

```
/**
 * Получение user_id пользователя из сессий
 *
 * @return int
 */
public static function getUserId() {
      $userid = 0;
      if ($_SESSION['user_id'] > 0) {
            $userid = $_SESSION['user_id'];
      }
      return (int) $userid;
}
```

Пример реализации:

$isAuthorize = User::getUserId();

3.4. Разавторизация

Параметры:

- скрипт: "auth.controller.inc/logout"

Пример обращения с формы на HTML-странице:

```
<div class="auth-login__inline">
     <div class="btns btns--center">
          <?php echo $User instanceof User ? '<a
     href="auth/logout">Выйти</a>' : '<button class="btn
     btn--small auth-login__btn
     js-page-login-submit">Войти</button>' ?>
     </div>
</div>
```

Авторизован, user_id = 2

Аккаунт подтвержден

Выйти

3.5. Скрипты

3.5.1. Скрипт отображения страницы авторизации (auth.php)

Листинг:

```
<?php
require_once 'inc/lmautoloader.class.inc';
LmAutoloader::register();

require_once getenv('DOCUMENT_ROOT') . '/config.php';
require_once 'inc/header.inc';

$lmMysql = lmMysql::getInstance();

function makeUsersHtml($usersData) {
    $usersHtml = '<table
border="1px"><tr><td>user_id</td><td>name</td><td>email</td><
td>print</td><td>salt</td><td>safe_print</td></tr>';
    foreach ($usersData as $user) {
      $usersHtml .= '<tr>' .
           '<td>' . $user['user_id'] . '</td>' .
           '<td>' . $user['name'] . '</td>' .
           '<td>' . $user['email'] . '</td>' .
           '<td>' . $user['print'] . '</td>' .
           '<td>' . $user['salt'] . '</td>' .
           '<td>' . $user['safe_print'] . '</td>' .
           '</tr>';
    }
```

```
    $usersHtml .= '</table>';
    return $usersHtml;
}

$usersData = $lmMysql->fetchAllByKey('user_id',
                'SELECT u.user_id, ui.`name`, e.email,
p.print, p.salt, p.safe_print FROM users u
INNER JOIN users_i18n ui ON u.user_id = ui.user_id
INNER JOIN emails e ON e.user_id = u.user_id
INNER JOIN prints p ON p.user_id = u.user_id');

$usersHtml = makeUsersHtml($usersData);

include('auth.html');
```

3.5.2. Скрипт регистрации (ajax/register.php)

Параметры POST:

- email - почта
- password - пароль
- login - Имя пользователя

Листинг:

```
<?php
require_once __DIR__ . '/../inc/lmautoloader.class.inc';
LmAutoloader::register();

require_once getenv('DOCUMENT_ROOT') . '/config.php';

header('Content-type: text/html; charset=utf-8');

Origin::getHeadersForOptionsSubdomain();
$allheaders = getallheaders();

$DataFilter = new DataFilter([
    DataFilter::FROM_POST => [
       'type' => new DataTypeString(),
       'action' => new DataTypeString(),
       'email' => new DataTypeString(),
       'password' => new DataTypeString(),
       'lastname' => new DataTypeString(),
       'login' => new DataTypeString(),
       'loc_id1' => new DataTypeString(),
       'loc_id2' => new DataTypeString(),
       'loc_id3' => new DataTypeString(),
       'loc_id4' => new DataTypeString(),
```

```
            'cityName' => new DataTypeString(),
            'regionId' => new DataTypeString(),
            'data' => new DataTypeString(),
            'kaptchakey' => new DataTypeString(),
            'kaptchakey2' => new DataTypeString(),
            'fingerprint' => new DataTypeString(),
            'clientData:json' => [
                 'user_agent' => new DataTypeString(),
                 'timezone_offset' => new DataTypeInteger(),
            ],
            'csrf' => new DataTypeString(),
        ],
        DataFilter::FROM_GET => [
            'track' => new DataTypeString(),
        ],
            ]);

    $Reg = new Registrate();

    $registerResult = $Reg->proceedRegistrate(
           $DataFilter->fromPost(),
           $Ecosystem,
           false,
           Auth::SOURCE_AUTH_WEB,
           Auth::TYPE_OF_AUTH_INTERACTIVE,
           null,
           Registrate::REG_FROM_MODAL_WINDOW,
           User::REG_METHOD_EMAIL
    );

    if (isset($registerResult['ok'], $registerResult['id']) &&
       $registerResult['ok'] === 'ok' &&
       $registerResult['id'] > 0) {
        $userId = $registerResult['id'];
        RedirectHelper::fastRedirect('/auth.php?user_id=' .
$userId);
    }
    RedirectHelper::fastRedirect('/auth.php');
```

3.6. Последовательность выполнения операций

3.6.1. Регистрация

- Зайти на страницу auth.php
- Ввести данные в форму блока “Регистрация”
- Нажать Enter или кнопку “Зарегистрироваться”

3.6.2. Авторизация

- Зайти на страницу auth.php
- Ввести данные в форму блока “Авторизация”
- Нажать Enter или кнопку “Войти”

3.6.3. Подтверждение аккаунта

- Зайти на страницу auth.php
- Авторизоваться
- Если email не подтвержден, в блоке “Авторизация” присутствует ссылка “Подтвердить email” и код подтверждения
- Переход по ссылке запускает подтверждение аккаунта, после подтверждения происходит переход на auth.php