

Регистрация. E-mail вместо Login

Обновлено 30 августа 2024, 12:34

Алгоритм

- Перенесём компонент `bitrix:main.register` в своё пространство имён и кастомизируем как нам нужно
- Подпишемся на событие чтобы заменить поле LOGIN на EMAIL
- Заменим вывод системного компонента нашим.

bitrix:main.register

Выносим компонент в свое пространство имен `/local/components/kochnev/main.register`

На страницу регистрации добавляем компонент

```
<?APPLICATION->IncludeComponent("kochnev:main.register", "flat", Array(
    "AUTH" => "Y", // Автоматически авторизовать пользователей
    "REQUIRED_FIELDS" => array( // Поля, обязательные для заполнения
        0 => "EMAIL",
        1 => "NAME",
        2 => "LAST_NAME",
        3 => "WORK_COMPANY",
        4 => "PERSONAL_PHONE",
        4 => "PERSONAL_STREET",
    ),
    "SET_TITLE" => "Y", // Устанавливать заголовок страницы
    "SHOW_FIELDS" => array( // Поля, которые показывать в форме
        0 => "EMAIL",
```

```

        1 => "NAME",
        2 => "LAST_NAME",
        3 => "WORK_COMPANY",
        4 => "PERSONAL_PHONE",
        4 => "PERSONAL_STREET",
    ),
    "SUCCESS_PAGE" => "/personal/index.php",          // Страница окончания регистрации
    "USER_PROPERTY" => "", // Показывать доп. свойства
    "USER_PROPERTY_NAME" => "", // Название блока пользовательских свойств
    "USE_BACKURL" => "Y", // Отправлять пользователя по обратной ссылке, если она есть
),
false
);?>

```

В настройках компонента выберем поля, которые надо показывать и укажем поля, обязательные для заполнения.

Кастомизация компонента

Сразу уточню, что для решения этой задачи эффективнее всего именно изменение кода компонента. Обусловлено это тем, что он имеет очень много проверок и условий, обходить которые себе дороже будет, при внесении изменений в шаблон. Плюс мы получаем доступ к порядку вывода полей формы регистрации.

Открываем на редактирование файл `/bitrix/components/demo/main.register/component.php`

Меняем строки:

```

$arDefaultFields = array(
    "LOGIN",
    "PASSWORD",
    "CONFIRM_PASSWORD",
    EMAIL",
);

```

на

```
$arDefaultFields = array(
    "EMAIL",
    "PASSWORD",
    "CONFIRM_PASSWORD",
);
```

Подписываемся на события **OnBeforeUserRegister** и **OnBeforeUserUpdate**

Для этого в папке **local** создаем файл **init.php**, создаем директорию **lib** и в ней создаем файл **Event.php** в который будем складывать все обработчики событий и подключим его в **init.php**

Если вы не работаете с классами, обработчик можно подключать в обычном файле и подключать его в **init.php**. Главное не превращайте свой **init.php** в свалку. В нем должны быть **только** подключения!

Как подключать свои классы описано [здесь](#) и [здесь](#)

local/php_interface/lib/**Events.php**

```
<?php
use Bitrix\Main\EventManager;

$eventManager = EventManager::getInstance();
// User
$eventManager->addEventHandler('main', 'OnBeforeUserRegister', [Helper\UserHelper::class, 'registerByEmailEvent']);
$eventManager->addEventHandler('main', 'OnBeforeUserUpdate', [Helper\UserHelper::class, 'registerByEmailEvent']);
```

local/php_interface/**init.php**

```
<?php
require_once __DIR__ . '/lib/Events.php';
```

Класс **UserHelper**

```
public static function registerByEmailEvent(array &$arFields):array
{
    $arFields["LOGIN"] = $arFields["EMAIL"];
    return $arFields;
}
```

Кастомизация шаблона

local/components/kochnev/main.register/templates/flat/[template.php](#) - здесь представление нашего компонента, верстаем как нам нужно.

Если нужно переопределить порядок полей в форме, то создадим файл [result_modifier.php](#)

```
<?php
if (!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED !== true) {
    die();
}
// меняем порядок следования полей
$arResult['SHOW_FIELDS'] = [
    'EMAIL',
    'NAME',
    'LAST_NAME',
    'PERSONAL_PHONE',
    'WORK_COMPANY',
    'PERSONAL_STREET',
    'PASSWORD',
    'CONFIRM_PASSWORD',
];
```

В [lang](#) файлах можно переопределить

Если зарегистрированный пользователь зайдет на страницу по умолчанию ему отобразится неказистое сообщение о том, что он авторизован. Давайте заменим это сообщение редиректом в персональный раздел.

Меняем этот код:

```
<?php if ($USER->IsAuthorized()): /* если пользователь уже авторизован */ ?>
    <p><?= GetMessage('MAIN_REGISTER_REG_AUTH'); /* Вы зарегистрированы и авторизованы */ ?></p>
    <?php return ?>
<?php endif; ?>
```

на ЭТОТ

```
?php if ($USER->IsAuthorized()): /* если пользователь уже авторизован, ему здесь делать нечего */ ?>
    <?php LocalRedirect('/auth/profile.php'); ?>
<?php endif; ?>
```

Заменим вывод системного компонента нашим

У нас теперь две формы регистрации с разным составом полей:

- первая формируется системным компонентом `system.auth.registration` и доступна по адресу `/auth/?register=yes`
- вторая формируется компонентом `bitrix:main.register` и доступна по адресу `/personal/auth/registration.php`

Давайте это исправим. Удалим весь код из шаблона компонента `system.auth.registration`, и поместим в него вызов компонента `bitrix:main.register` из файла `/personal/auth/register.php`

```
<?
/**
 * Bitrix Framework
 *
 * @package bitrix
 * @subpackage main
 * @copyright 2001-2014 Bitrix
 */

/**
 * Bitrix vars
 *
```

```

* @global CMain $APPLICATION
* @var array $arParams
* @var array $arResult
* @var CBitrixComponentTemplate $this
*/

if (!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED !== true) {
    die();
}

$APPLICATION->IncludeComponent(
    "bannikon:main.register",
    "flat",
    [
        "AUTH" => "Y",
        "REQUIRED_FIELDS" => [
            0 => "EMAIL",
            1 => "NAME",
            2 => "LAST_NAME",
            3 => "WORK_COMPANY",
            4 => "PERSONAL_PHONE",
            4 => "PERSONAL_STREET",
        ],
        "SET_TITLE" => "Y",
        "SHOW_FIELDS" => [
            0 => "EMAIL",
            1 => "NAME",
            2 => "LAST_NAME",
            3 => "WORK_COMPANY",
            4 => "PERSONAL_PHONE",
            4 => "PERSONAL_STREET",
        ],
        "SUCCESS_PAGE" => "/personal/index.php",
        "USER_PROPERTY" => "",
        "USER_PROPERTY_NAME" => "",
        "USE_BACKURL" => "Y",
    ],

```

```
        false  
    );
```

Не забываем заменить файлы других системных компонентов:

```
<?php  
/*  
 * Файл local/templates/.default/components/bitrix/system.auth.authorize/.default/lang/ru/template.php  
 */  
$MESS['SYS_AUTH_AUTHORIZE_TITLE']      = 'Авторизация';  
// теперь e-mail используется как логин  
$MESS['SYS_AUTH_AUTHORIZE_LOGIN']      = 'E-mail';  
$MESS['SYS_AUTH_AUTHORIZE_PASSWORD']   = 'Пароль';  
$MESS['SYS_AUTH_AUTHORIZE_REMEMBER']    = 'Запомнить меня';  
$MESS['SYS_AUTH_AUTHORIZE_CAPTCHA_TITLE'] = 'Защита от автоматической регистрации';  
$MESS['SYS_AUTH_AUTHORIZE_CAPTCHA_TEXT'] = 'Введите символы с картинки';  
$MESS['SYS_AUTH_AUTHORIZE_SUBMIT']     = 'Войти';  
$MESS['SYS_AUTH_AUTHORIZE_REG_USER']   = 'Зарегистрироваться';  
$MESS['SYS_AUTH_AUTHORIZE_FORGOT']     = 'Забыли пароль?';
```

```
<?php  
/*  
 * Файл local/templates/.default/components/bitrix/system.auth.forgotpasswd/.default/lang/ru/template.php  
 */  
$MESS['SYS_AUTH_FORGOT_TITLE']         = 'Восстановление пароля';  
// теперь восстановление пароля только по e-mail  
$MESS['SYS_AUTH_FORGOT_HELP']          = 'Введите Ваш E-mail: инструкция по смене пароля будет отправлена на этот почтовый адрес.';  
$MESS['SYS_AUTH_FORGOT_LOGIN']         = 'Логин';  
$MESS['SYS_AUTH_FORGOT_EMAIL']         = 'E-mail';  
$MESS['SYS_AUTH_FORGOT_CAPTCHA_TITLE'] = 'Защита от автоматической регистрации';  
$MESS['SYS_AUTH_FORGOT_CAPTCHA_TEXT'] = 'Введите символы с картинки';  
$MESS['SYS_AUTH_FORGOT_SUBMIT']        = 'Отправить';  
$MESS['SYS_AUTH_FORGOT_AUTH']          = 'Авторизация';
```

```

<?php
/*
 * Файл local/templates/.default/components/bitrix/system.auth.forgotpasswd/.default/template.php
 */
if (!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED!==true) die();
?>

<!-- логин больше не нужен, удаляем или комментируем код ниже -->
<!--
<div>
    <span>
        <?= GetMessage('SYS_AUTH_FORGOT_LOGIN'); /* логин пользователя */ ?>
    </span>
    <span>
        <input type="text" name="USER_LOGIN" maxlength="50" value="<?= $arResult["LAST_LOGIN"]; ?>" />
    </span>
</div>
-->

```

```

<?php
/*
 * Файл local/templates/.default/components/bitrix/system.auth.changepasswd/.default/lang/ru/template.php
 */
$MESS['SYS_AUTH_CHANGE_TITLE']      = 'Смена пароля';
// теперь e-mail используется как логин
$MESS['SYS_AUTH_CHANGE_LOGIN']      = 'E-mail';
$MESS['SYS_AUTH_CHANGE_CHECK']      = 'Контрольная строка';
$MESS['SYS_AUTH_CHANGE_PASSWORD']   = 'Новый пароль';
$MESS['SYS_AUTH_CHANGE_CONFIRM']     = 'Подтверждение пароля';
$MESS['SYS_AUTH_CHANGE_CAPTCHA_TITLE'] = 'Защита от автоматической регистрации';
$MESS['SYS_AUTH_CHANGE_CAPTCHA_TEXT'] = 'Введите символы с картинки';
$MESS['SYS_AUTH_CHANGE_SUBMIT']      = 'Изменить пароль';
$MESS['SYS_AUTH_CHANGE_AUTH']        = 'Авторизация';
$MESS['SYS_AUTH_CHANGE_REQUIRED']    = 'Эти поля обязательны для заполнения';

```


Есть еще один момент, о котором есть смысл упомянуть. Если при изменении пароля не заполнить поле «E-mail», появляется сообщение об ошибке *«Логин должен быть не менее 3 символов»*.

Меняется это в недрах ядра Битрикс. Поэтому небольшой хак:

```
<?php
/*
 * Файл local/templates/.default/components/bitrix/system.auth.changepasswd/.default/template.php
 */
if (!defined("B_PROLOG_INCLUDED") || B_PROLOG_INCLUDED!==true) die();
?>

<div class="system-auth-changepasswd">

<h2><?= GetMessage('SYS_AUTH_CHANGE_TITLE'); /* заголовок формы */ ?></h2>

<?php
// небольшой хак, который убирает сообщение о длине логина
if (isset($arParams['AUTH_RESULT']['TYPE'])
    && $arParams['AUTH_RESULT']['TYPE'] == 'ERROR'
    && false !== iconv_strpos($arParams['AUTH_RESULT']['MESSAGE'], 'Логин должен быть')) {
    $arParams['~AUTH_RESULT']['MESSAGE'] = 'Недопустимое значение поля «E-mail»';
}
?>

<p><?php ShowMessage($arParams["~AUTH_RESULT"]); /* сообщение о результате смены пароля */ ?></p>
```