Оглавление

[Citycrm Замечания в ходе работы 2](#_Toc164953360)

[Русификация валидации. 2](#_Toc164953361)

[Описание механизма Alert с отображением цвета и иконки. 3](#_Toc164953362)

[Как было 3](#_Toc164953363)

[Стало. 3](#_Toc164953364)

[Выводы по результатам работы с PHP разных версий. 4](#_Toc164953365)

[Настройка Kohana 3.3 и 3.4 4](#_Toc164953366)

[Разные PHP 5](#_Toc164953367)

[Порядок параметров в объявлении функии 5](#_Toc164953368)

[ErrorException [ Deprecated ]: iconv(): Passing null to parameter #3 ($string) of type string is deprecated 5](#_Toc164953369)

[Авторизация 6](#_Toc164953370)

[Login 6](#_Toc164953371)

[SQLSTATE[IM001]: Driver does not support this function: driver does not support quoting when using the driver pdo\_odbc on php 8.0.1 6](#_Toc164953372)

[Ошибка pdo нет метода lastInsertId 6](#_Toc164953373)

[Ошибка quote. Редактирование файла Query.php. 15.08.2023 7](#_Toc164953374)

[Сессии и авторизация Одна авторизация на все сайты. 8](#_Toc164953375)

[Путь решения 1: разные имена в session\_key конфигурации \application\config\auth.php 8](#_Toc164953376)

# Citycrm Замечания в ходе работы

|  |  |  |
| --- | --- | --- |
|  | Дата | Прим. |
|  | 6.08.2023  8.08.2023 |  |
|  | 9.08.2023 | Обход ошибки lastinsertId. Перенос PDO.php в app |
|  | 15.08.2023 | Проблема с кавычками при формировании SQL-запроса. |
|  | 11.04.2024 | Добавлено описание механизма Alert и arrAlert. |

# Русификация валидации.

Файл для русификации должен быть указан в методе errors:

$post->errors('validateCard')

Файл русификации должен лежать в папке C:\xampp\htdocs\citycrm\application\messages\validateCard.php

Формат файла русификации:

*<?php defined('SYSPATH') or die('No direct script access.');*

*$messages = array(*

*'name' =>*

*array(*

*'not\_empty' => ':field Не может быть пустым',*

*),*

*'getCardInfo' =>*

*array(*

*'not\_empty'=> 'Номер карты не может быть пустым',*

*'regex'=> 'Недопустимый набор данных. Номер должен состоянить из цифр 0-1, запятой, букв ABCDEF!',*

*),*

*'card' =>*

*array(*

*'t555'=> ':value Карта уже зарегистрирована.',*

*),*

*'id\_pep' => array(*

*'not\_empty'=> 'Пользователь не найден.',*

*),*

*'idPepInfo' => array(*

*'Model\_People::unique\_username'=> 'Пользователь не найден.',*

*'not\_empty'=> 'Номер пользователья не может быть пустым',*

*'digit'=> 'Номер пользователья должен быть числом',*

*),*

*'q' => array(*

*'digit'=> 'Строка поиска ":value" должна состоять из цифр',*

*),*

*);*

*return $messages;*

Указана переменная, затем массив правило – как переводить.

Если указано :field, то будет указано название поля, которое проверяли.

Если указать :value то будет указано значение поля, которое проверяли.

# Описание механизма Alert с отображением цвета и иконки.

## Как было

Если посмотреть исходные коды, то видно, что при обращении к View::factory передается параметр ->bind('alert', $fl)

Во всех представлениях есть строки, выводящие этот alert (если он имеется).

Однако у этой реализации были такие недостатки:

1. Нельзя отобразить результат цветом. Я считаю это существенным недостатком.

## Стало.

Я хотел решить задачи:

1. Цветной фон под сообщение: красный – все плохо, желтый – внимание и т.п.
2. Добавить иконку для разных вариантов.
3. Вывести подробный отчет о результат операции.
4. Сделать так, чтобы alert сам исчезал через некоторое время.

Чтобы новый механизм работал надо:

1. А модели сформировать список сообщений
   1. Собрать результаты выполнения какой-либо операции

$alert=\_\_('contact.addRfidOk', array(':id\_card'=>$key->id\_card));

* 1. Затем сформивароть массив алертов

$arrAlert[]=array('actionResult'=>0, 'actionDesc'=>$alert);

Массив содержит:

* 'actionResult' – результат выполнения, по которому будет выбран фон сообщения и иконка,
* 'actionDesc' – выводимый текст.
  1. Передать массив в сессию:

Session::instance()->set('arrAlert',$arrAlert);

1. В контроллере сформировать переменную arrAlert

*$arrAlert = $this->session->get('arrAlert'); //извлечь алерт из сессии*

*$this->session->delete('arrAlert');//очистить алерт в сессии*

* 1. Передать переменную алерт во вью:

*->bind('arrAlert', $arrAlert)*

1. в каждую view надо вписать строку

include Kohana::find\_file('views','alert');

содержимое файла можно посмотреть в исподниках.

*C:\xampp\htdocs\citycrm\application\views\Alert.php*

*C:\xampp\htdocs\citycrm\application\views\AlertState.php*

Для работы надо передать переменную

if (isset($arrAlert))

в результате этих действий будет выведено цветное окно с текстовыми сообщениями о результатах работы.

# Выводы по результатам работы с PHP разных версий.

План получается такой:

1. Citycrm сначала отработать на версии PHP 5.6.
2. При переходе на PHP 7 придется аккуратно обрабатывать функции, где параметр может оказаться Null. Очевидно, при работе с PHP 5.6 надо обращать внимание на эти места, и, по возможности, заранее избегать их. Например, использовать Arr::get($array, $name, '') – если нет значения, то использовать пустоту.
3. При переходе от PHP 7 к PHP 8 придется переделать SQL запросы в DB::query с целью избежать ошибок с кавычками. Возможно, все удастся решить штатными методами PHP, но пока это не очевидно как сделать.

В любом случае для citycrm переход на PHP 7 и 8 сопряжено с достаточно большим объемом переделок, и решение этого вопроса я считаю нужным пока отложить.

Для начала надо реализовать CRM на PHP 5.6, отладить его на имеющихся объектах, после чего рутинно переводить на старшие версии PHP.

# Настройка Kohana 3.3 и 3.4

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | что | 3.3 | 3.4 |
| 1 | Bootstrap | 'base\_url' => '/citycrm', | 'base\_url' => '/citycrm/public', |
| 2 | \application\views\template.php | нет | <base href="http://localhost/citycrm/"> |
|  | Путь к js | Моно не указывать домен  <script type="text/javascript" src="js/jquery.js"></script> | Указывать домен  <script type="text/javascript" src="/citycrm/js/jquery.js"></script> |
|  | Сдвиг позиций основного окна |  |  |
|  |  |  |  |

# Разные PHP

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | что | PHP 5.6.5 | PHP 7.X и 8.X |
| 1 | Порядок параметров в объявлении функций |  | См. ниже. |
| 2 | NIULL в различных функциях iconf | Допускается | Не допускается.  См. ниже. |
| 3 | DB::query | Нет особых замечаний | В PHP8 есть проблемы с поддержкой кавычек. См. ниже. |
|  | Работ |  |  |
|  |  |  |  |

# Порядок параметров в объявлении функии

<https://php.watch/versions/8.0/deprecate-required-param-after-optional>

При объявлении функции или метода добавление обязательного параметра после необязательных параметров устарело, начиная с [PHP 8.0](https://php-watch.translate.goog/versions/8.0?_x_tr_sl=en&_x_tr_tl=ru&_x_tr_hl=ru&_x_tr_pto=wapp) .

Что сделал:

В добавил определение для все фукнций т.о.:

Было:

*public function getListF($user, $page = 1, $perpage = 10, $filter)*

стало

*public function getListF($user, $page = 1, $perpage = 10, $filter = null)*

т.е. к последнему параметру добавил *=null*

Ошибка исчезла.

# ErrorException [ Deprecated ]: iconv(): Passing null to parameter #3 ($string) of type string is deprecated

В PHP 8.1 has deprecated passing null as parameters to a lot of core functions. My main problem is with functions like [htmlspecialchars(php)](https://php.net/htmlspecialchars) and [trim(php)](https://php.net/trim), where null no longer is silently converted to the empty string.

Для строки

*$q = $contacts->getCountAdmin(iconv('UTF-8', 'CP1251', $filter));*

*Стало*

*$q = $contacts->getCountAdmin(iconv('UTF-8', 'CP1251', $filter ?? '1'));*

# Авторизация

## Login

Авторизация происходит по цепочке Auth->login далее \_login, а реализация \_login сделана в файле City.php.

Там формируется запрос к БД СКУД, и все полученные данные сохраняются как массив с именем auth\_user (а это имя настраивается в C:\xampp\htdocs\citycrm\application\config\auth.php)ю

Т.о., при вызове метода Auth::instance()->get\_user() я получаю набор данных, указанных в запросе в City.php.

# SQLSTATE[IM001]: Driver does not support this function: driver does not support quoting when using the driver pdo\_odbc on php 8.0.1

<https://github.com/yiisoft/yii/issues/4343>

<https://www.php.net/manual/ru/pdo.quote.php>

PDO::quote — Заключает строку в кавычки для использования в запросе.

Проблему можно обойти, если :

1. SQL-запрос DB::query сформировать заранее,
2. Либо в DB::query вместо -> param() использовать конструкцию \_\_('xxx', array()).

В любом случае для citycrm это достаточно большой объем переделок, и решение этого вопроса я считаю нужным пока отложить.

# Ошибка pdo нет метода lastInsertId

Для устранения ошибки я сделал копию файла в C:\xampp\htdocs\citycrm\application\classes\Kohana\Database\PDO.php (чтобы вызывался именно этот файл), и в нем закомментировал строку

*elseif ($type === Database::INSERT)*

*{*

*// Return a list of insert id and rows created*

*return array(*

*//$this->\_connection->lastInsertId(),//Бухаров*

*$result->rowCount(),*

*);*

*}*

Я предполагаю, что теперь, при реализации метода Database::INSERT система не будет пытаться вызвать метод lastinsertId().

Дальнейшая работа подтвердила, что ошибок при вставке данных более не возникает.

# Ошибка quote. Редактирование файла Query.php. 15.08.2023

При подключении к БД СКУД через драйвер все работало штатно.

При подключении к БД СКУД через ODBC появилась ошибка: SQL запрос не содержал в себе параметры.

Важно: при работе через драйвер все ОК, а при работе через ODBC – параметры sql запроса пусты.

Анализ показал, что ошибка происходит тут:

*public function compile($db = NULL)*

*{*

*if ( ! is\_object($db))*

*{*

*// Get the database instance*

*$db = Database::instance($db);*

*}*

*// Import the SQL locally*

*$sql = $this->\_sql;*

*if ( ! empty($this->\_parameters))*

*{*

*// Quote all of the values*

*--------🡪>>>> $values = array\_map(array($db, 'quote'), $this->\_parameters);*

*// Replace the values in the SQL*

*$sql = strtr($sql, $values);*

*}*

*return $sql;*

*}*

Я переписал эту функцию т.о.:

public function compile($db = NULL)

{

if ( ! is\_object($db))

{

// Get the database instance

$db = Database::instance($db);

}

// Import the SQL locally

$sql = $this->\_sql;

if ( ! empty($this->\_parameters))

{

// Quote all of the values

//$values = array\_map(array($db, 'quote'), $this->\_parameters);

foreach ($this->\_parameters as $key=>$value)// обход проблемы с кавычками 15.08.2023

{

$res[$key]='\''.$value.'\'';

}

$values=$res;

// Replace the values in the SQL

$sql = strtr($sql, $values);

}

return $sql;

}

Т.е. вместо

//$values = array\_map(array($db, 'quote'), $this->\_parameters);

Я сделал перебор данных в цикле и добавление одинарных кавычек.

Измененый файл Query.php положил в C:\xampp\htdocs\citycrm\application\classes\Kohana\Database\Query.php.

Проблемы с формированием SQL-запроса исчезли

# Сессии и авторизация Одна авторизация на все сайты.

Выглядит так: я авторизовался на сайте citycrm, затем захожу на city, и вижу результаты авторизации на citycrm. Но, т.к. структура данных для авторизации на citycrm другая, то на city авторизация выдает ошибку

## Путь решения 1: разные имена в session\_key конфигурации \application\config\auth.php

Это позволило быстро решить проблему: каждый сайт показывает свои параметры авторизации