

Комментарии к уроку 4.2.

1. Все системные уведомления с сайта отправляются по почтовым шаблонам. В шаблонах есть настройки по умолчанию, которые нужно не забывать исправлять под структуру вашего сайта. Например, чаще всего не совпадают шаблоны ссылок на страницу восстановления пароля, подписку на рассылки и т.д. Подробнее о почтовых шаблонах - http://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=41&LESSON_ID=3523

2. При отображении ошибок в компоненте «Форма авторизации» необходимо установить в параметрах компонента опцию «Показывать ошибки». За вывод ошибки отвечает код:

<?

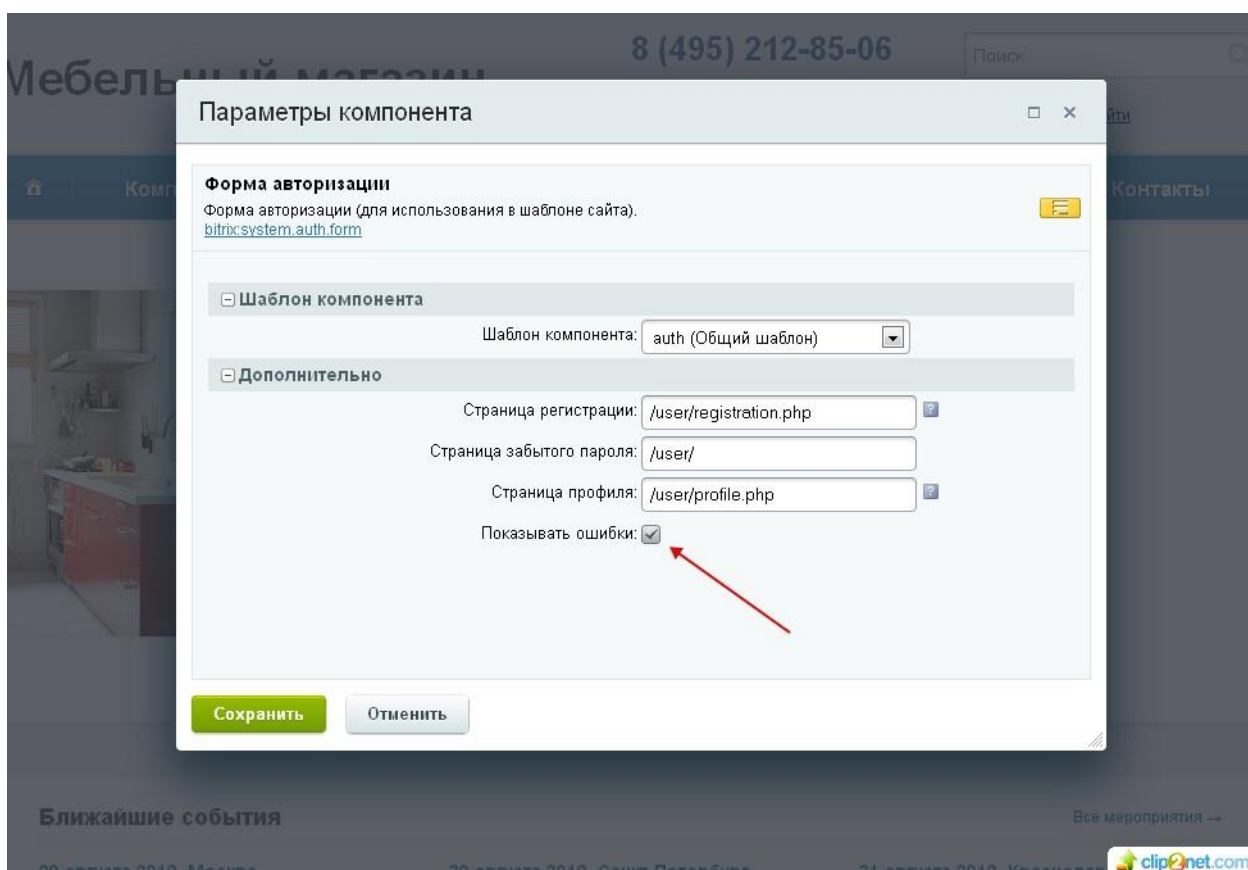
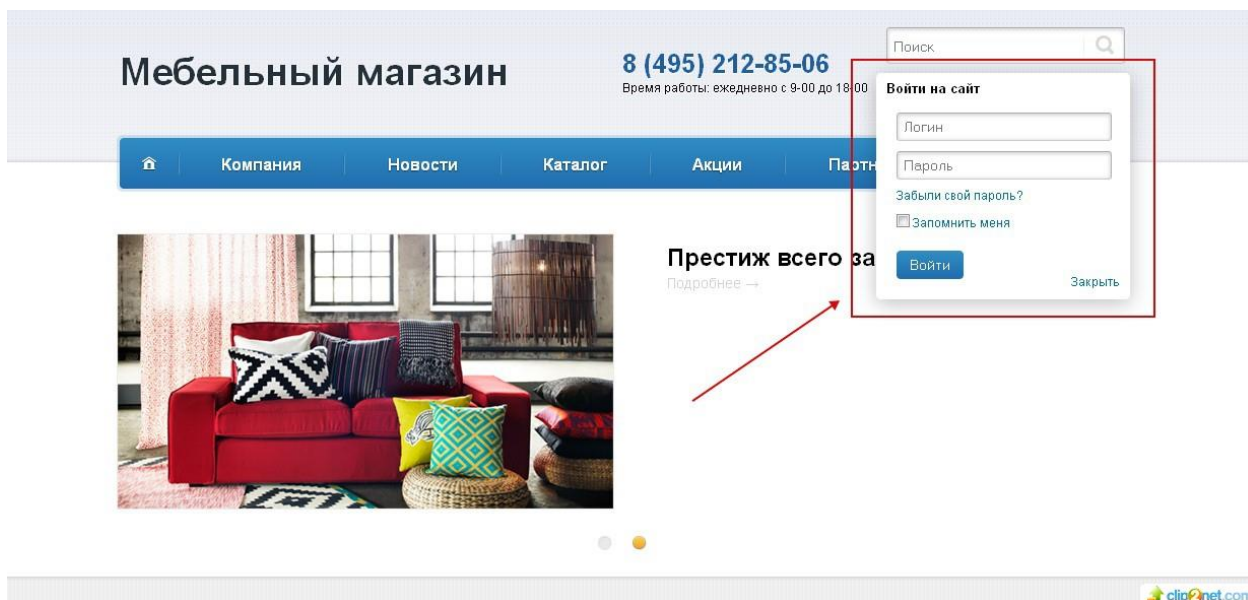
```
if ($arResult['SHOW_ERRORS'] == 'Y' && $arResult['ERROR'])
)ShowMessage($arResult['ERROR_MESSAGE']);
?>
```

ShowMessage(\$arResult['ERROR_MESSAGE']) так же выводит информационные сообщения, не являющиеся ошибками. Чтобы организовать вывод только ошибок, необходимо в конструкцию <?if()?> дописать условие, которое указало бы, что сообщение должно выводиться лишь в том случае, когда \$arResult['ERROR_MESSAGE']['TYPE'] равняется "ERROR".

3. Частой ошибкой является то, что форма авторизации в шапке не выводит сообщения о неверном логине/пароле.

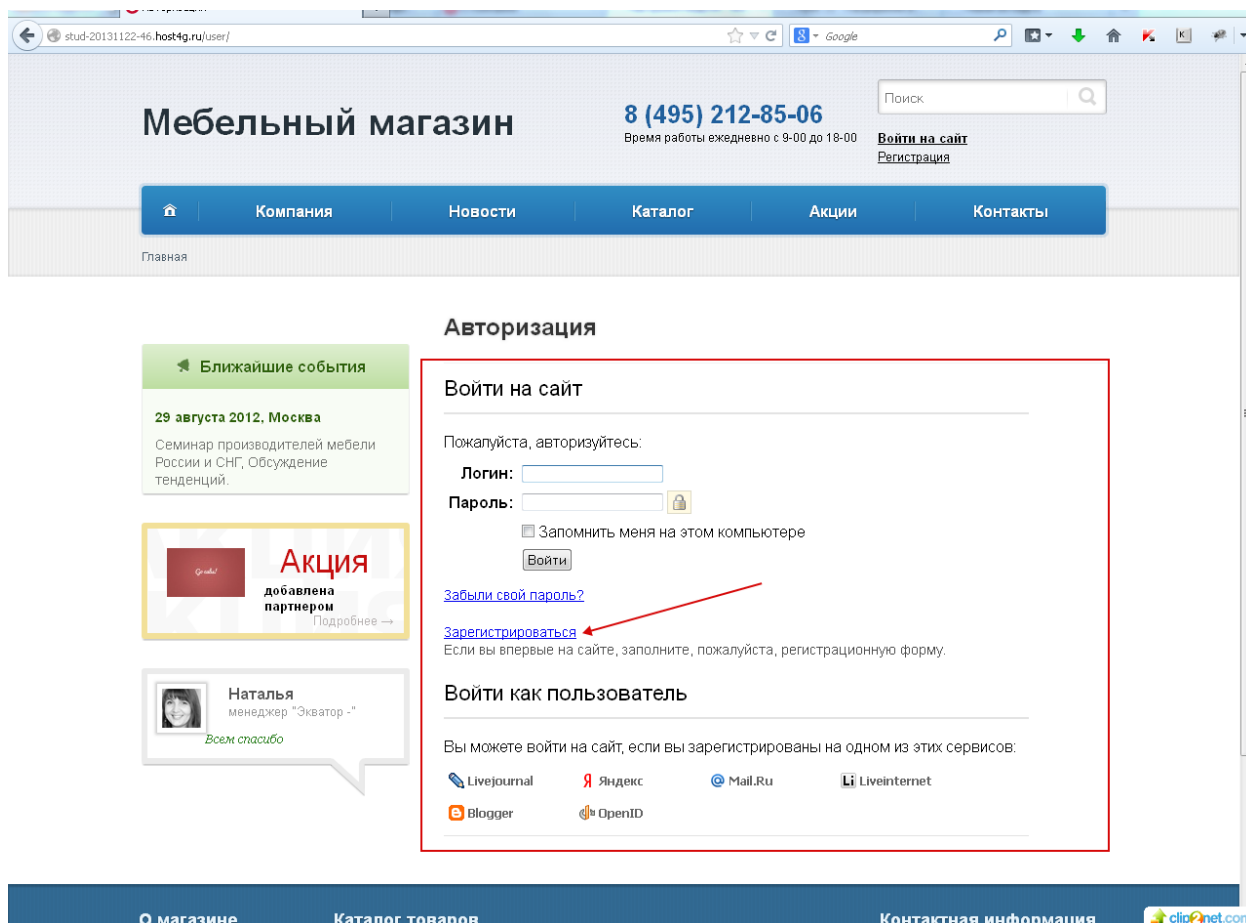
Для того, чтобы выводить сообщение об ошибке, нужно, во-первых, в настройках компонента включить опцию «Показывать ошибки». Во-вторых, в шаблоне компонента нужно сохранить (не удалить при доработке стандартного шаблона) следующие строки:

```
if ($arResult['SHOW_ERRORS'] == 'Y' && $arResult['ERROR'])
    ShowMessage($arResult['ERROR_MESSAGE']);
```



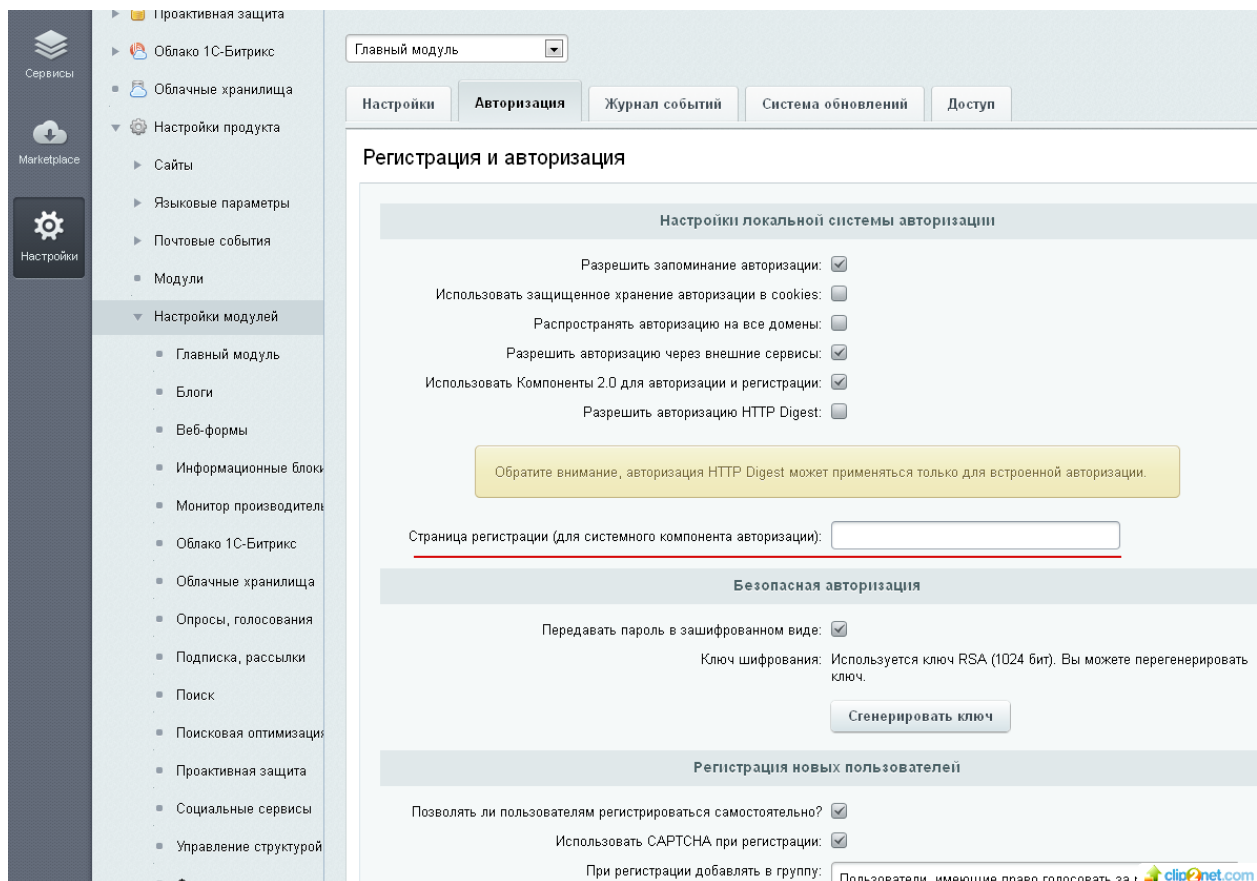
4. Следующей распространённой ошибкой является некорректно настроенная ссылка «Регистрация» в стандартной форме авторизации.

При попытке доступа неавторизованного пользователя к закрытому разделу, Битрикс вызывает форму авторизации, компонент `system.auth.authorize` с шаблоном `.default`.



В отличие от обычного вызова компонентов на страницах или в шаблоне сайта, данный компонент подключается системой автоматически «при необходимости», и настройки компонента система указывает самостоятельно.

В шаблоне данной формы авторизации есть ссылка «Зарегистрироваться». URL данной ссылки задаётся в админке, в настройках главного модуля:



5. Опыт показывает, что иногда слушатели начинают знакомство с php почти одновременно с освоением системы Битрикс. При этом, к сожалению, не знают про средства отладки кода в php. Это важный навык, он сильно облегчает процесс создания сайта.

Многие вопросы можно решить самостоятельно, выполнив отладку. Посмотрев, какие значения имеют переменные, входные аргументы функций и т.д. Часто требуется посмотреть, какие имеются поля в массиве `$arResult` или `$arParams` в шаблоне компонента...

В простейшем случае для просмотра значения какой-либо переменной можно вывести её отладочное представление на экран или записать в лог. В более сложных случаях часто используются такие отладчики, как XDEBUG в связке со средой разработки, позволяющие посмотреть выполнение кода по шагам.

Здесь мы рассмотрим простейшие варианты отладки, которые обычно достаточны для начинающих разработчиков.

Во-первых, для вывода отладочного значения переменной на экран, можно использовать стандартные функции php «в чистом виде». Например, `var_dump()` или `print_r()`.

<http://www.php.net/manual/ru/function.var-dump.php>

<http://www.php.net/manual/ru/function.print-r.php>

Чтобы вывести значение какой-либо переменной или массива `$a` на экран, разместите в коде:

```
echo '<pre>'; var_dump($a); echo '</pre>';
```

Обрамление тегом `<pre>` нужно для того, чтобы браузер выводил все переносы строк и отступы «как есть», для удобочитаемости.

Иногда для удобства программисты пишут свои функции-обёртки над стандартными функциями php. Например, функция ниже выводит отладочное представление лишь для пользователя, заходящего с определённого IP:

```
function dump($var){  
    if($_SERVER['REMOTE_ADDR'] == '97.81.135.226'){  
        echo '<pre>'; var_dump($var); echo '</pre>';  
    }  
}
```

Такое бывает нужно, когда требуется внести правки в код прямо на рабочем сайте, и требуется вывести отладочное представление переменной так, чтобы оно было видно для программиста, но не видно для обычных посетителей сайта. В этом случае требуется определить функцию для отладки в `init.php`, чтобы она была доступна везде на сайте. И тогда в любом месте кода можно вывести значение переменной, вызвав

```
dump($a);
```

При выполнении одного из предыдущих заданий вы определяли в `init.php` функцию `test_dump()`, выводящую отладочную информацию о переменной лишь для админа, и ничего не выводящую для остальных:

```
function test_dump($var, $die = false, $all = false)  
{  
    global $USER;  
    if( ($USER->GetID() == 1) || ($all == true))  
    {  
        ?>  
        <font style="text-align: left; font-size: 10px"><pre><?var_dump($var)?></pre></font><br>  
        <?  
    }  
    if($die)  
    {  
        die;  
    }  
}
```

Иногда вывод отладочного представления переменной на экран затруднён. Например, если нужно посмотреть значение переменной в коде, обрабатывающем ajax-запрос. Или если после выполнения кода происходит обновление или редирект страницы. В этом случае можно вывести отладочное представление переменной в лог или отправить на почту.

Для записи отладочного представления в лог можно использовать функцию AddMessage2Log(), имеющуюся в Битрикс.

http://dev.1c-bitrix.ru/api_help/main/functions/debug/addmessage2log.php

Для её использования достаточно в init.php определить константу, содержащую путь к лог-файлу:

```
define("LOG_FILENAME", $_SERVER["DOCUMENT_ROOT"]."/имя_файла.txt");
```

Если файл отсутствует, система его создаст при первой записи. Далее в любом месте кода для записи отладочного представления переменной в лог можно использовать конструкцию:

```
AddMessage2Log(print_r($var, true));
```

Для того, чтобы отослать отладочное представление переменной на почту, в Битрикс определена функция SendError():

http://dev.1c-bitrix.ru/api_help/main/functions/debug/senderror.php

Для её использования нужно в init.php определить константу с почтой, на которую должны отсылаться письма:

```
define("ERROR_EMAIL", "ваша@почта.ru");
```

После этого в любом месте кода можно вызвать функцию:

```
SendError(print_r($var, true) . "\n");
```

Естественно, для этих функций также можно написать свои, более удобные обёртки.