

iranphp articles

توابع کار با فایل xml
Masoud Alinqian
masoud@m2ix.com
.....

عنوان مقاله :
نگارنده :
آدرس پست الکترونیک :
تاریخ نگارش :

توابع کار با فایل xml :

```
<?php
/** {{{ Copyright and License
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public
 * License as published by the Free Software Foundation; either
 * version 2.0 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * General Public License for more details.
 *
 * You should have received a copy of the GNU General Public
 * License along with this library; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 *
 * @copyright 2001-2004 Masoud Alinqian.
 * @author Masoud Alinqian <masoud at m2ix dot com >
 * @author Dave <dave at ovumdesign dot com >
 * @package XML Functions
 * }}}
 */

function xml_get_children($vals, &$i) {

    $children = array();

    if (isset($vals[$i]['value'])) $children[] = $vals[$i]['value'];

    while (++$i < count($vals)) {

        switch ($vals[$i]['type']) {

            case 'cdata':

                $children[] = $vals[$i]['value'];

                break;

            case 'complete':

                $children[] = array(

                    'tag' => $vals[$i]['tag'],

                    'attributes' => isset($vals[$i]['attributes'])?

                        $vals[$i]['attributes'] : null,

                    'value' => $vals[$i]['value'],

                );

                break;

            case 'open':
```

```
        $children[] = array(
            'tag'      => $vals[$i]['tag'],
            'attributes' => isset($vals[$i]['attributes'])?
                $vals[$i]['attributes'] : null,
            'children' => xml_get_children($vals, $i),
        );
        break;
    case 'close':
        return $children;
    }
}

function xml_get_tree($file) {
    $file=realpath($file);

    $data = @implode(NULL,file($file));

    $parser = xml_parser_create();
    $data=ereg_replace("(\\r|\\n)", "", $data);

    xml_parser_set_option($parser, XML_OPTION_CASE_FOLDING, 0);
    xml_parser_set_option($parser, XML_OPTION_SKIP_WHITE, 1);
    xml_parse_into_struct($parser, $data, $vals, $index);
    xml_parser_free($parser);

    return array(
        'tag'      => $vals[0]['tag'],
        'attributes' => isset($vals[0]['attributes'])?
            $vals[0]['attributes'] : null,
        'children' => xml_get_children($vals, $i = 0),
    );
}

function xml_condense_tree($tree) {
```

```
foreach ($tree['children'] as $index => $node) {
    if (isset($node['children'])) {
        $tree['children'][$index] = xml_condense_tree($node);
    } elseif (isset($node['value']) and !$node['attributes']) {
        $tree['values'][$node['tag']] = $node['value'];
        unset($tree['children'][$index]);
    }
}

if (!$tree['children']) unset($tree['children']);
return $tree;
}

function xml_config_to_array($xml_tree,$root=false){
    $array_name=$xml_tree['tag'];
    if (is_array($xml_tree['children']))
    foreach ($xml_tree['children'] as $children){
        $child_id=$children['attributes']['id'];
        $child_name=$children['tag'];
        $child_name.=( $child_id) ? "__".$child_id:'';
        if (is_array($children['children'])) {
            $child_array=xml_config_to_array($children);
            $temp_array[$child_name]=$child_array;
        } else {
            $temp_array[$child_name]=$children['value'];
        }
    }
}

if (!$root)
    $xml_array=$temp_array;
else
    $xml_array[$array_name]=$temp_array;
```

```
return $xml_array;
}

function xml_to_array($xml_tree){
    $array_name=$xml_tree['tag'];
    if ($xml_tree['attributes'])
        foreach ($xml_tree['attributes'] as $attribute_name=>$attribute_value) {
            $temp_name="_$attribute_name";
            $temp_array[$temp_name]=$attribute_value;
        }

    if ($xml_tree['children'])
        foreach ($xml_tree['children'] as $children){
            $temp_name=$children['tag'];
            if (!is_array($temp_array[$temp_name])) $temp_array[$temp_name]=array();
            array_push($temp_array[$temp_name],xml_to_array($children));
        }

    if ($xml_tree['value']) {
        $temp_array['_']=$xml_tree['value'];
    }

    return $temp_array;
}

function recover_xml_array($xml_array) {
    $array_array=$xml_array;
    foreach ($array_array as $array_key=>$array_value) {
        if (is_array($array_value)){
            if(count($array_value)==1) $array[$array_key]=recover_xml_array($array_value[0]);
            else {
                $temp_array=array();
            }
        }
    }
}
```

```
$counter=0;

foreach ($array_value as $child_value){

    $counter++;

    if ($child_value['_name']) {

        $index_name=$child_value['_name'];

    } elseif ($child_value['_id']) {

        $index_name=$child_value['_id'];

    } else {

        $index_name=$counter;

    }

    $temp_array[$counter]=recover_xml_array($child_value);

}

$array[$array_key]=$temp_array;

}

} else {

    $array[$array_key]=$array_value;

}

}

return $array;

}

function array_to_xml($array_tree,$root='root') {

    if (is_array($array_tree)) {

        $xml_temp='';

        $attributes='';

        foreach($array_tree as $xml_key=>$xml_value) {

            $is_array=is_array($xml_value);

            if ($xml_key=='_') {

                $xml_value=string_to_entity($xml_value);

            }

        }

    }

}
```

```
        $xml_temp.=$xml_value;
    } else if (!$is_array && substr($xml_key,0,1)=='_') { # check for is attribute
        $xml_value=string_to_entity($xml_value);
        $attributes.=substr($xml_key,1,strlen($xml_key)-1)."\\"$xml_value" ";
    } else if ($is_array && substr($xml_key,0,1)=='_') { #check for is array
        foreach($xml_value as $child_value) {
            $xml_temp.=array_to_xml($child_value,substr($xml_key,1,strlen($xml_key)-1));
        }
    } else if ($is_array) {
        $xml_temp.=array_to_xml($xml_value,$xml_key);
    } else {
        $xml_value=string_to_entity($xml_value);
        $xml_temp.=$xml_value;
    }
}
}
if ($xml_temp || $xml_temp==='0') {
    $xml_output("<$root $attributes>$xml_temp</$root>");
} else {
    $xml_output("<$root $attributes />");
}
return $xml_output;
}
?>
```