Paging in Joomla

In this tutorial we will explain how to implement paging using Joomla built-in features. Paging is essential in web development since large datasets are not recommended to be sent to client browsers.

Introduction

In this tutorial we will explain how to implement paging using Joomla built-in features. Paging is essential in web development since large datasets are not recommended to be sent to client browsers.

How it works...

mySQL has a nice feature when it comes to pageing. If you add ' LIMIT 10, 20 ' at the end of your query the database engine will return recordset rows from 10 - 30 (we assume that the recordset is larger than 30).

Joomla implements this feature trough database class witch is responsible for handling SQL statements. If you haven't used database class you should read "Introduction to database programming" article.

The code

There are two parameters that are used to pass paging info between pages: limit and limitstart. First is used to define how many rows will be returned per page and the other one defines the page from where the paging starts.

To retrieve this two params you can use the following lines

```
$limit      = intval( mosGetParam( $_REQUEST, 'limit', 0 ) );

$limitstart = intval( mosGetParam( $_REQUEST, 'limitstart', 0 ) );
```

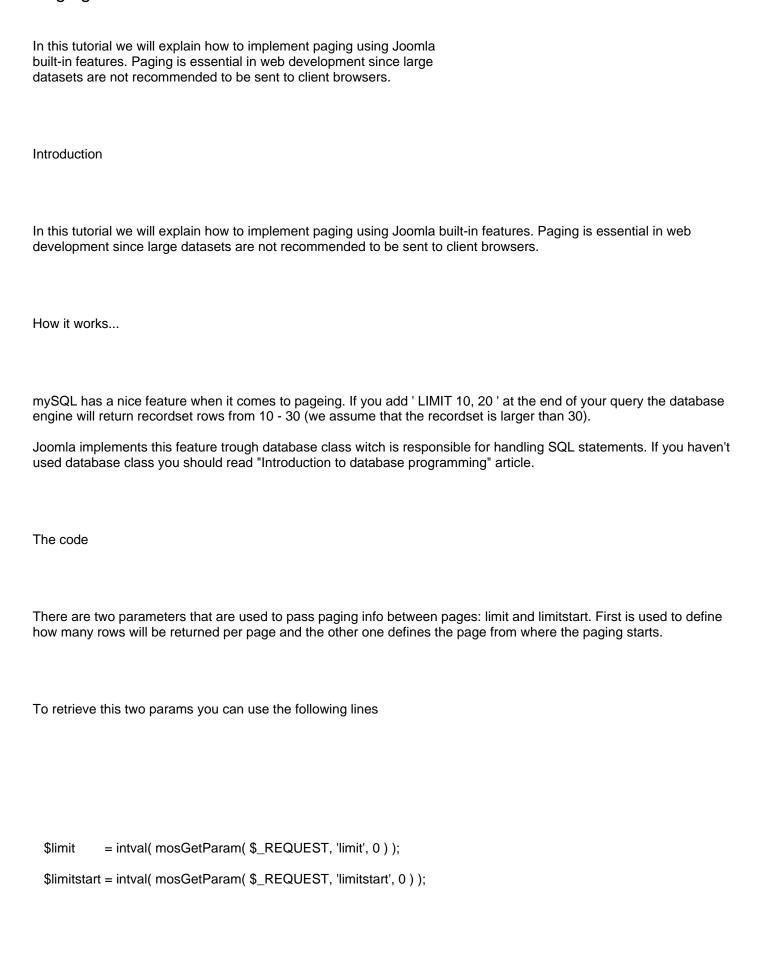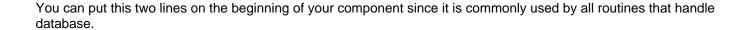You can put this two lines on the beginning of your component since it is commonly used by all routines that handle database.

Next step is to find out how many rows will be returned by the query on which you intend to implement paging.

```
$query = "SELECT COUNT(id) FROM #__users";

$database->setQuery( $query );

$total = $database->loadResult();
```

Now we must make sure that we set $limit and $limitstart variables in case they haven't been set yet.

```
$limit = $limit ? $limit : 15;

if ( $total <= $limit ) {

    $limitstart = 0;

}
```

The first line sets the $limit to 15 if it hasn't been set yet. That means 15 items per page. User can change this number later using a drop-down box.

The second line sets the $limitstart to 0 in case that total number of items is less than $limit variable, insuring that something will be displayed if the $limitstart is greater than max.

Now we must include pageNavigation.php and create mosPageNav object.

```
require_once( $GLOBALS['mosConfig_absolute_path'] . '/includes/pageNavigation.php' );

$pageNav = new mosPageNav( $total, $limitstart, $limit );
```

Then execute a real query with all fields we need to display our items, and pass $limitstart and $limit to $database object.

```
$query = "SELECT id. name, username, email FROM #__users";

$database->setQuery( $query, $limitstart, $limit );

$rows = $database -> loadObjectList();

for( $i=0; $i<count($rows); $i++ )
{
    $row = $rows[$i];
    echo $row->name." / ".$row->username." / ".$row->email."<br>";
}
```

This will write all users on current page to client browser.

To draw paging we use the following code.

```
<div style="position: relative; float: left;">

        <div style="float: right;">Display #

            <?php

                echo $pageNav->getLimitBox( $link );

            ?>

        </div>


        <div style="float: left;">

            <?php

            echo $pageNav->writePagesLinks( $link );

            ?>

        </div>


    </div>


    <div style="position: relative; float: left;">

        <?php

            echo $pageNav->writePagesCounter();

        ?>

</div>
```

This last piece of code can be rearranged, so you can fit it to your component the way you like it.

Summary

This is very basic example of paging implementation in Joomla component, but as the complexity of your component grow the prinicpe stays the same. Joomla and mySQL functionality makes paging implementation easy.