**Warning**

Excuse my English. Original instruction written on Russian. I would be pleased, if anybody will translate it to correct English.

**Installation and usage**

* Copy the file Pagination.class.php to project directory;
* Include the file there, when you want to use pagination (require\_ince(‘path-to-class/Pagination.class.php’));
* Initialize class ($pagination = new Pagination($parameters)).

**Parameters of library**

The parameters of library are an array with next indexes:

|  |  |  |
| --- | --- | --- |
| **Index of array** | **Type** | **Description** |
| db (\*) | PDO object | PDO object for database connection. (Class work tested at MySQL and PostgreSQL connections.) **Required, if not indicated parameter db\_config.** |
| db\_config (\*) | array | Array of keys-parameters for database connection. The parameter description is here. **Required, if not indicated parameter db.** |
| table\* | string | Table with needed records  **Required.** |
| url\* | string | URL to page, where displaying records (for example: http://yoursite.com/news, /news/ or /news.php?page**=**).**Required.** |
| page | integer | Number of displaying page. *By default: 1.* |
| limit | integer | Amount of records at page. *By default: 10.* |
| pages\_amount | integer or string | Amount of displaying pages on the right and on the left from current page. *By default: 5*. |
| where | array | Array of criterias for records selection. *No value by default*. |
| with\_pages | boolean | Specifies whether to generate the pagination links. The parameter may be required in case if needed to display random page without possibility move on (for example, as free demonstration materials of the website). *By default: true*. |
| except\_first\_last | boolean | If a parameter “with\_pages” set as true, then this can except link to first and last pages. *By default: false*. |
| except\_groups | boolean | If a parameter “with\_pages” set as true, then this can except link to next and previous groups of pages. Group is set of current page and displaying pages on the left and on the right. *By default: false*. |
| except\_next\_previous | boolean | If a parameter “with\_pages” set as true, then this can except link to next and previous pages. *By default: false*. |
| except\_numeric | boolean | If a parameter “with\_pages” set as true, then this can except link to numeric pages. *By default: false*. |
| label\_first | string | Label for a link to first page. *By default: <<<*. |
| label\_last | string | Label for a link to last page. *By default: >>>*. |
| label\_next\_group | string | Label for a link to next group of pages. *By default: >>*. |
| label\_previous\_group | string | Label for a link to previous group of pages. *By default: <<*. |
| label\_next | string | Label for a link to next page. *By default: >*. |
| label\_previous | string | Label for a link to previous page. *By default: <*. |
| pages\_attributes | array | Array of attributes for navigation links. May include any html-attributes, for example: id, class, data-attributes. The parameter description is here. *No value by default*. |

**The parameter for database connection (db\_config)**

This parameter contains next indexes:

|  |  |
| --- | --- |
| **Индекс массива** | **Описание** |
| host \* | DBMS host. |
| database \* | Database name. |
| user \* | User name. |
| password \* | User password. |
| driver | DBMS driver. *By default: mysql.* |
| port | DBMS port. *No value by default*. |

**The parameter of criterias (where)**

This parameter is array of couples key-value, for example:

// Select records where category\_id =1

array(

‘category\_id’ => 1

);

Key may contains operators: : =, !=, >, <, <>, >=, <=, IS NULL, IS NOT NULL, IN, NOT IN.

In case using operator IS NULL or IS NOT NULL, value must be null, for example:

array(

‘author\_id IS NOT NULL’ => null

);

In case using operator IN or NOT IN, value must be array, for example:

array(

‘category\_id IN => array(2, 3, 8)

);

**The parameter of attributes for navigation links (pages\_attributes)**

This parameter involves indexes of types pages with an array of the form: ‘html-attribute’ => ‘value of attribute’.

Types of pages:

|  |  |
| --- | --- |
| **Index of array** | **Description** |
| first | For a link to first page. |
| previous\_group | For a link to previous group of pages. |
| previous | For a link to previous page. |
| numeric | For a link to numeric pages. |
| current | For a link to current page. For this type of page will be appended parameters for numeric pages, if it exists. |
| next | For a link to next page. |
| next\_group | For a link to next group of pages. |
| last | For a link to last page. |

Example:

$parameters['pages\_attributes'] = array(

'first' => array(

‘id’ => ‘pagination-first’

'data-first' => 1

),

'numeric' => array(

'class’ => 'pagination-numeric'

)

)

**Methods of library**

|  |  |
| --- | --- |
| **Method** | **Description** |
| buildPages() | Generates group of pages. Uses in case if during initialization parameter “with\_pages” did set as false. |
| get() | Returns pagination as an array of data. |
| getCurrentPage() | Returns number of current page. |
| getLimit() | Returns amount of records on page. |
| getPagesAmount() | Returns amount of pages, displaying on the right and on the left from current. |
| getStart() | Returns number of record, from which started show of page (numbering from zero). |
| getTotalRecords() | Returns total amount of records in table, considering criterias, if they did set. |
| getTotalPages() | Returns total amount of pages. |
| getPages() | Returns an array of current group of pages. |
| getStartShow() | Returns number of record, from which started show of page (numbering from one). |
| getEndShow() | Returns number of last displayed record. |

**Array of pagination data**

Array of pagination data contains next indexes:

|  |  |
| --- | --- |
| **Index** | **Contains** |
| current\_page | Contains number of current page. |
| start | Contains number of record, from which started show of page (numbering from zero). |
| limit | Contains amount of records on page. |
| start\_show | Contains number of record, from which started show of page (numbering from one). |
| end\_show | Contains number of last displayed record. |
| total\_records | Contains total amount of records in table, considering criterias, if they did set. |
| total\_pages | Contains total amount of pages. |
| pages | Contains an array of current group of pages. |

**Array of pages group**

Index of pagination array “pages” is array, each element contains next indexes:

|  |  |
| --- | --- |
| **Index** | **Contains** |
| caption | Contains caption for link to page |
| href | Contains link to page. |
| attributes | Contains array of html-attributes |

Also each page contains additional index, defining type of it.

|  |  |
| --- | --- |
| **Index** | **Means** |
| is\_first | This is link to first page. |
| is\_previous\_group | This is link to previous pages group. |
| is\_previous | This is link to previous page. |
| is\_numeric | This is numeric link to page. |
| is\_current | This is link to current page. This type same contains index “is\_numeric”. |
| is\_next | This is link to next page. |
| is\_next\_group | This is link to next pages group. |
| is\_last | This is link to last page. |

**Afterword**

Example displaying pagination, using Twig:

{% if pagination %}

<div>Current page is {{ pagination.current\_page }}</div>

<div>Showed from {{ pagination.start\_show }} to {{ pagination.end\_show }} - {{ pagination.limit }} items</div>

<div>Total {{ pagination.total\_records }} items on {{ pagination.total\_pages }} pages</div>

{% if pagination.pages %}

<ul class="pagination float-group">

{% for page in pagination.pages %}

<li{{ page.is\_current ? ' class="active"' : ''}}>

<a href="{{ page.href }}"{% for key,value in page.attributes %} {{ key }}="{{ value }}"{% endfor %}>{{ page.caption }}</a>

</li>

{% endfor %}

</ul>

{% endif %}

{% endif %}

Here you can learn how to easy make AJAX pagination and see a demo.