

How it works...

First, we got the total models count and initialized the new pagination component instance with it by passing the totalcount variable to our Pagination instance. Then, we used the $pages->pageSize field to set up the page size for our pagination. After that, we created a sorter instance for the model, specifying model attributes we wanted to sort by and applying order conditions to the query by calling orderBy and passing $sort->orders as a parameter. Then, we called all() to get records from the DB.

At this point, we have the models list, pages, and data used for the link pager, and the sorter that we use to generate sorting links.

In a view, we use the data we have gathered. First, we generate links with the Sort: :link method. Then, we list the models. Finally, using the LinkPager widgets, we render the pagination control.

See also

Visit the following links to get more information about pagination and sorting:

• <http://www.yiiframework.com/doc-2.0/yii-data-pagination.html>

• [http://www.yiiframework.com/doc-2.0/yii-data-sort.html](http://www.yiiframework.com/doc-2.0/yii-data-pagination.html)

• [http://www.yiiframework.com/doc-2.0/guide-output-pagi](http://www.yiiframework.com/doc-2.0/yii-data-sort.html)nation.html [http://www.yiiframework.com/doc-2.0/guide-output-sorting.html](http://www.yiiframework.com/doc-2.0/guide-output-pagination.html)