

CreativeTable

v1.3

CreativeTable it's a php class that builds a table based on the data retrieved by an SQL query. It has lots of options and it's very, very configurable.

It's perfect to incorporate with your backoffices, because it's very flexible and very easy to build. You can also easily integrate with your sites.

Features

- very easy to configurate
- support almost all sql queries, if not all
- extremely flexible
- super fast
- search and advanced search with multiple search
- items per page
- sort and multi sort
- actions list
- easy to add static rows or to add columns
- integration with any pager system
- multiple pagers in the same table
- multiple tables in the same page
- easy way to add html in your cells
- easy way to configure the attributes of your cells
- various examples (Real, Complex, Ajax, Search, ItemsPerPage, Sort, ExtraCols, Data, 2DArray, Separate, Custom, Display, 2Pagers, 2Tables, LargeDB, 2ItemsPerPage, FormatCols, CellAttributes, Json)
- styles examples

Greetings

I just wanna thank you for buying this class.

If you need any help or have some suggestion to make, please contact me for:

sandro@creativedreams.eu

Very Important

- This class is in constant development
- I need your feedback for improvements
- If you have any problem contact me - sandro@creativedreams.eu
- I will help you immediately (between 10h-20h GMT)
- Good rates implies continuous improvements
- New improvements are available to you for free

So help us both and give me your feedback. ;)

Initial Configuration

In order to view the examples files you must have an external or a local (WAMP, XAMPP) WEB server with PHP installed. You must also have a MYSQL database installed.

If you fill these requisites, then the only thing you need to do is:

- run the DB dump assets/creativedreams_db.sql (it will create the creativdreams DB with the necessary tables)
- optional - to see LargeDB example, run the DB dump assets/creativedreams_large_db.sql
- open configurations.php and change \$db_server, \$db_username, \$db_password, \$db_name to your own values
- run index.html

Documentation

Instructions

To build a table with its component you only need to have these lines of PHP code and the following HTML inclusions.

PHP

You always need to include the Class, gather the necessary information from the database and from the form, and configure the layout of the table. You may check all the available configuration parameters in front.

```
Include_once('creativeTable.php');
```

```
$ct=new CreativeTable();
```

```
// Data Gathering
```

```
$params['sql_query'] = 'SELECT id,rating,title,votes FROM imdbtop250';
```

```
$params['search'] = $search;
```

```
$params['multiple_search'] = $multiple_search;
```

```
$params['items_per_page'] = $items_per_page;
```

```
$params['sort'] = $sort;
```

```
$params['page'] = $page;
```

```
$params['total_items'] = $total_items;
```

```
// Layout Configurations
```

```
$params['header'] = 'Rank,Rating,Title,Votes';
```

```
$params['width'] = '45,60,,70';
```

```
$ct->table($params);
```

```
$ct->display();
```

HTML

You only need to include these 3 lines in the head of the html document

```
<link rel="stylesheet" type="text/css" href="css/creative.css">
```

```
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
```

```
<script type="text/javascript" src="js/creative_table.js"></script>
```


or

```
<script type="text/javascript" src="js/creative_table_ajax.js"></script>
```


Structure

The next image shows the final result of a table with its components, with their html ids in blue.




































search

Search... 

items_per_page

10 

sort

	Rank 	Rating 	Title 	Votes 	Actions
			multi_search		
	1	9.1	The Shawshank Redemption (1994)	489048	 
	2	9.1	The Godfather (1972)	392126	 
	3	9.0	The Godfather: Part II (1974)	232161	 
	4	8.9	Il buono, il brutto, il cattivo. (1966)	148969	 
	5	8.9	Pulp Fiction (1994)	396749	 
	6	8.8	Schindler's List (1993)	262551	 
	7	8.8	12 Angry Men (1957)	109861	 
	8	8.8	One Flew Over the Cuckoo's Nest (1975)	202689	 
	9	8.8	Star Wars: Episode V - The Empire Strikes Back (1980)	265556	 
	10	8.8	The Dark Knight (2008)	436120	 

extra_cols

extra_cols

actions

-- Actions -- 

pager

Page: 1 of 25

1

2

3

4

5

»

...

Last »

Total: 250

Corresponding to this abstract HTML structure with the default id ("ct"):

```
1 <form id="ct_form">
2   <input type="hidden" id="ct_items_per_page">
3   <input type="hidden" id="ct_sort">
4   <input type="hidden" id="ct_page">
5   <input type="hidden" id="ct_total_items">
6
7   <div id="search_container">
8     <input id="ct_search" />
9   </div>
10
11  <div id="ct_items_per_page_container">
12    <select id="ct_items_per_page_change"></select>
13  </div>
14
15  <table id="ct" class="">
16    <thead>
17      <tr id="ct_sort">
18        <th></th>
19      </tr>
20      <tr id="ct_multiple_search">
21        <th><input id="ct_multiple_search1" /></th>
22      </tr>
23    </thead>
24    <tbody>
25      <tr class="odd">
26        <td></td>
27      </tr>
28      <tr class="even">
29        <td></td>
30      </tr>
31    </tbody>
32  </table>
33
34  <div id="ct_actions_container">
35    <select id="ct_actions"></select>
36  </div>
37
38  <div id="ct_pager_container">
39    ...
40  </div>
41
42 </form>
```

Parameters

Parameters	Default	Description
sql_query	null	sql query to obtain the data. You must discriminate all fields i.e. SELECT id,rating,title,votes FROM imdbtop250 and not use SELECT * FROM imdbtop250 <code>sql query</code>
table_key	null	primary key of the table. Boost table performance. <code>text</code>
data	null	data in array format <code>bidimensional array - \$arr[rows][columns]</code>
search	null	search word <code>text</code>
multiple_search	null	multiple search by column <code>an array with the search words by column</code>
items_per_page	first value of items_per_page_init	selected items per page <code>integer</code>
sort	true	sort string <code>true, false, 1a_2d_t_f</code>
page	1	number of the actual page <code>integer</code>
total_items	null	total items of the data <code>integer</code>
id	table	id of the table. Usefull for configuring the CSS when using multiples tables on same page <code>text</code>
class	null	class of the table. Usefull for configuring the CSS when using multiples tables on same page <code>text</code>
form_init	true	includes the html form tag or not <code>true; false;</code>
form_url	null	specifies the form action url <code>text</code>
header	true	text for the header fields separated by commas i.e. 'ID,Movie Title,Votes,...'

		text
table_width	null	table width with in % or px i.e. 100%, 700px, ...
width	null	columns width integer separated by commas i.e. 50,100,,20,,...
search_init	true	show or not show the search option. You may also choose wich fields of the BD to search by building a string with t (true) anf f (false) that represent each filed i.e. 'tftf' search 1st and 3rd column only true, false, ttftt
search_type	like	to search words that either looks like, it's equal, the beginning looks like or the end beginning looks like the keyword like, =, beginning_like, end_like
search_html	...	html to customize the search box html
multiple_search_init	hide	hide, show all, not show, or show only some columns of multiple search hide, true, false, ttftt, ttftthide
multiple_search_type	like	to search words that either looks like, it's equal, the beginning looks like or the end beginning looks like the keyword on the multiple search like, =, beginning_like, end_like
items_per_page_init	50*\$i	initialization of select box with the items per page. Values separated by columns or a formula using \$i for the iteration. false; 10,20,50,100; (\$i+1)*10
items_per_page_all	#TOTAL_ITEMS#	show the all option in items per page and define the text to show text; false
items_per_page_url	null	javascript code when items per page is changed text
items_per_page_ids	null	when using multiple items per page in one table you need to tell the ids to the table ids separated by commas i.e. ct1,ct2,...
sort_init	true	show or not to show the sort options. You may also choose wich columns to sort.

		true, false ttfft
sort_order	adt	order of the sort. A string that represents (ascending, descending, true) i.e 'adt' = ascending -> descending -> true normal state adt, ad, da, dat
sort_url	null	javascript code when sort column is clicked text
extra_cols	null	array containing the information about extra columns array(array(col,header,width,html),array(...),...)
odd_even	true	create class odd and even in rows false, true
no_results	No results found.	html for the row when no results are returned html
actions	null	array containing the value and the text of the actions select box array(array(\$value,\$text),...)
actions_url	ctActions(#ID#)	javascript code when actions select box is changed text
pager	null	html for the pager html
pager_ids	null	when using multiple pagers in one table you need to tell ids of the pagers are assign to the table ids separated by commas i.e. ct1,ct2,...
ajax_url	\$_SERVER['PHP_SELF']	url to be called by the ajax request url
display_cols	null	show or not show some columns, by building a string with t (true) anf f (false) that represent each column i.e. 'tfft' hide the second column string composed by "t" or "f", i.e., ttfft or fttt
debug	true	show the errors in the table or not true, false

Possible values

In search_html, items_per_page_url, items_per_page_all, extra_col it is possible to use the following tags:

- #ID#
- #ROW#
- #PAGE#
- #ITEMS_PER_PAGE#
- #TOTAL_ITEMS#

Also, in extra_col it is possible to get the value of a column by using #COL + COLUMN NUMBER + #, ie:

- #COL2#
- #COL5#

Methods

In majority of cases you only need to call from the CreativeTable class the **table**, **add_col** and **display** methods, but if you need something from the class you have also the following available methods:

Method	Arguments	Description
table	null	builds the table
add_row	\$arr_html,\$row	adds a static row
add_col	\$col,\$header,\$width, \$html,\$op='init_data'	adds a new column
get_sql	\$check_table_key=false	return the final composed sql query considering or not the table_key improvement
get_sql_select	null	return the sql query corresponding to selecting fields and tables parameters
get_sql_where	\$check_table_key=true	return the sql query corresponding to conditions parameters considering or not the table_key improvement
get_sql_order	null	return the sql query corresponding to order parameters
get_sql_limit	null	return the sql query corresponding to limit parameters
change_tags	\$str	change the tags for their value #ID#, #PAGE, #ITEMS_PER_PAGE#, #TOTAL_ITEMS#
change_tag_col	\$str,\$arr_cols	change the column tags for their value #COL1#, #COL2#, ...
set_cell_attribute	\$value,\$row,\$column	set an html attribute to a specific row, column or cell
draw_items_per_page	\$id_items_per_page	displays the html fot a custom items per page
display	\$op="", \$ajax=false	displays the creative table with its components \$op - string separated by commas to recieve the components of the table separately i.e: 'form_init, search, items_per_page, table, thead, tbody, actions, pager, form_final, javascript' or 'form_ini,table,form_final' or 'thead,tbody', ... if

		\$op=" all shown. \$ajax - if true, then prepare the data for an ajax transfer (utf8_encode,etc.)
--	--	------------------------------------------------------------------------------------------------------

Javascript

The Class needs the ***creative_table-1.3.js*** or the ***creative_table_ajax-1.3.js*** to work properly. A minified version is also available.

The functions in those files are pre prepared to handle the table requisites, but you may alter those functions to satisfy your needs. The only thing you need to know is that those functions are called when the corresponding states are changed.

Method	Arguments	Description
ctSearchFocus	table_id	handle the opacity of the search string
ctSearchBlur	table_id	handle the opacity of the search string
ctSearchKeypress	table_id	handle the opacity of the search string
ctSearch	table_id	does the search
ctMultiSearch	table_id	does the advanced search
ctShowAdvancedSearch	table_id	Show or hides the advanced input boxes
ctItemsPerPage	table_id	called when item per page are changed and saves the last items per page
ctSort	table_id,sort_column	builds the sort string to pass to the table. sort_column - integer corresponding to the column number
ctActions	table_id	called when actions are changed
ctPager	table_id,page	the pager should call this method to indicate wich page is to show page - number of the page to show
ctSubmitForm post version	table_id,sort,page, pass_total_items	submit the form with all the information about the table and its components table_id - id of the table to update sort - string built by ctSort of type t_1a_t_2d_t page - number of the page to show pass_total_items - boolean to indicate if is to pass the total_items to the class

ctSubmitForm ajax version	table_id,page, pass_total_items, reload_option	submit the form with all the information about the table and its components table_id - id of the table to update page - number of the page to show pass_total_items - boolean to indicate if is to pass the total_items to the class reload_option - string to indicate which components are to reload 'form_init, search, items_per_page, table, thead, tbody, actions, pager, form_final, javascript'
onInit	null	called on initialization of the ctSubmitForm
onComplete	null	called on finalization of the ctSubmitForm

Examples

You have 19 examples with different configurations:

- Real – a basic configuration with a simple query
- Complex – a basic configuration with a more complex query
- Ajax – an ajax example
- Search – an example with suggestions of the search parameter to see the result and learn more easily
- Items Per Page – an example with suggestions of the items_per_page parameter to see the result and learn more easily
- Sort – an example with suggestions of the items_per_page parameter to see the result and learn more easily
- ExtraCols – an example of how to add more columns
- Data – a data manipulation example. Adding links to the cells
- 2DArray – how to fill the table with a 2D Array
- Separate – building each components of the table separatly
- Custom – building your own custumize components of the table
- Display – how to display only some columns
- 2Pagers – how to had 2 Pagers in the same table
- 2Tables – display 2 tables in the same page
- LargeDB – an example showing the performance of CreativeTable using a large DB
- 2ItemsPerPage – how to had 2 Items Per Page in the same table
- FormatCols – an easy way to add html to your cells
- CellAttributes – how to add html attributes to your cells
- Json – an example for those who uses older browsers and PHP version lower than 5.2.0

Styles

You have 6 pre-configured CSS:

- Creative
- Creative2Tables
- Blue
- Green
- Gray
- Black

The CSS structure is:

- `/* RESET */`
- `/* TOP CONTAINER : SEARCH + ITEMS PER PAGE */`
- `/* SEARCH */`
- `/* ITEMS_PER_PAGE */`
- `/* TABLE */`
- `/* BOTTOM CONTAINER : ACTIONS + PAGER */`
- `/* ACTIONS */`
- `/* PAGER */ /* YOU MAY DELETE THIS IF YOU DONT USE THE CREATIVE PAGER */`

Greetings

Once again, thank you so much for purchasing this theme.

If you need any help or have some suggestion to make, please contact me for:

sandro@creativedreams.eu

I will be glad to help you.

Cheers,

Sandro