wpDataTables

WordPress Table plugin Documentation by [touchmesoft]()

If you notice some mistake, typo, or would like to contribute something to the documentation please contact us through the form on [this page](). Thank you!

Table of contents

1. General information

This chapter will give you some general information about the plugin. Features overview, description of files in the package, installation and configuration guide, quickstart guide.

1.1 Overview

This section is a brief overview of the plugin, it's purposes and main features.

Hi, dear customers. First of all we would like to thank you for the purchase of wpDataTables plugin, hope you will not regret this purchase, and will enjoy using it on your WordPress sites.

First of all I must mention that the front-end part of this plugin is based on a truly excellent DataTables jQuery library, and some plugins to it. I would like to thank its author Allan Jardine for this wonderful piece of code.

Probably you had a situation when you needed to publish a table on your WordPress-based site, which would need to have slightly more features then a simple HTML table: pagination, filters, or something similar: for example a catalog, a price list, a list of business locations, a list of users, and so on, and so on. And you probably discovered that it would require at least some coding, and keeping it regularly updated means that you would need to do all the coding work again and again. wpDataTables is a plugin which would handle all the work, and you would just need to set it up once.

wpDataTables is an independent ready-to-use WordPress table plugin which is designed to make table process of data representation quick, easy and effective. It allows you to quickly build and put on any of your WordPress posts, or pages interactive tables with such features as advanced filtering, sorting, or saving to PDF from almost any possible data source – MySQL query, serialized 2D-array, Excel file, CSV file, JSON and XML input sources.

A short list of wpDataTables features:

- Cute interactive multi-functional front-end jQuery tables with filtering, sorting and pagination features rendered by DataTables library. You can manipulate all features easily from the WordPress administrator panel. The initial javascript is compiled by PHP and then executed in your browser.

- Server-side processing for MySQL-based tables.
- Front-end editing for MySQL-based tables.
- Responsive mode for any tables.
- Advanced front-end table features: print view, save to PDF, save to CSV, save to Excel, Copy to clipboard by TableTools extensions. Enabled by default, can be disabled by just unticking a checkbox.
- Row grouping based on RowGrouping extension. If a lot of rows have same values of some column you can group them based on the value of this column. The sorting will be performed inside of these groups.
- Charts based on Google Chart Tools. Easily render charts from the same dataset as your table.
- Accepted data sources: PHP serialized arrays, MySQL queries, MS Excel XLS and XLSX, OpenOffice Calc ODT, CSV files, JSON objects, XML data source.
- Tweakable columns with different column types – strings, integers, floats, links, dates, images.

1.2 Files and folders in package

This section quickly covers the filestructure of the plugin so you would know where to look for things you might need

Here you can see the list of folders and files which you get in the wpDataTables package (some folders are not given in complete detail since they are not quite important).

ROOT

- **wpdatatables.php** – main plugin entry point file, initialization, configuration, includes, language file loading.

ASSETS

This folder contains the javascript and CSS files required for wpDataTables styling and functioning.

ASSETS/CSS

Cascade stylesheets needed for the wpDataTables plugin.

Separate files are used for some plugins. Main file for the front-end styling is: **wpdatatables.min.css**

Back-end styling: **wpdatatables_admin.css**

ASSETS/JS/JQUERY-DATATABLES

jQuery DataTables plugin and add-ons for it

ASSETS/JS/PHP-DATATABLES

javascript files needed for WPDataTables to work

ASSETS/JS/WPDATATABLES

Javascript needed for the plugin itself to run.

Main front-end JS file is: **wpdatatables.min.js** and its non-minified version **wpdatatables.js** (include it instead of the minified one if you want to change something).

LIB

Folder with 3rd party libraries used by wpDataTables.

SOURCE

Main wpDataTables/WPDataTables source files

- **class.date.wpdatacolumn.php** – The Date column type class file.
- **class.email.wpdatacolumn.php** – The e-mail column type class file.
- **class.float.wpdatacolumn.php** – The float column type class file.
- **class.formatnum.wpdatacolumn.php** – The formatted number column type class file.

- **class.int.wpdatacolumn.php** – The integer type class file.
- **class.link.wpdatacolumn.php** – The URL link column type class file.
- **class.wpdatacolumn.php** – The main column factory class file.
- **class.string.wpdatacolumn.php** – The plain string column type class file.
- **class.sql.php** – The MySQL driver of wpDataTables.
- **class.wpdatatable.php** – The main wpDataTable class file with the main module logic.
- **class.tpl.php** – The lightweight template engine of wpDataTables.
- **class.wdtexception.php** – wpDataTables specific exceptions
- **class.wdttools.php** – class that contains common backend helper funcitons.

SOURCE/LANG

This folder contains the files with languages for the front-end.

LICENSING

Licensing info for the plugin.

LANGUAGES

Folder with the .PO/.MO files for internationalization of the plugin backend (and some front-end parts).

DOCUMENTATION

Folder which contains this PDF file.

CONTROLLERS

- **wdt_functions.php** **–** main plugin functions used in the front-end (non-AJAX).
- **wdt_ajax_functions.php** – handlers for the plugin's front-end AJAX actions (front-end editor, server-side processing for MySQL tables, etc).
- **wdt_admin.php –** back-end (admin panel) controller functions.
- **wdt_admin_ajax_actions.php** **–** handlers for the back-end AJAX actions (saving the table, columns, preview, etc).

CONFIG

This folder contains a config file with several constants used within the plugin.

TEMPLATES

Templates used to render wpDataTables front-end and admin pages.

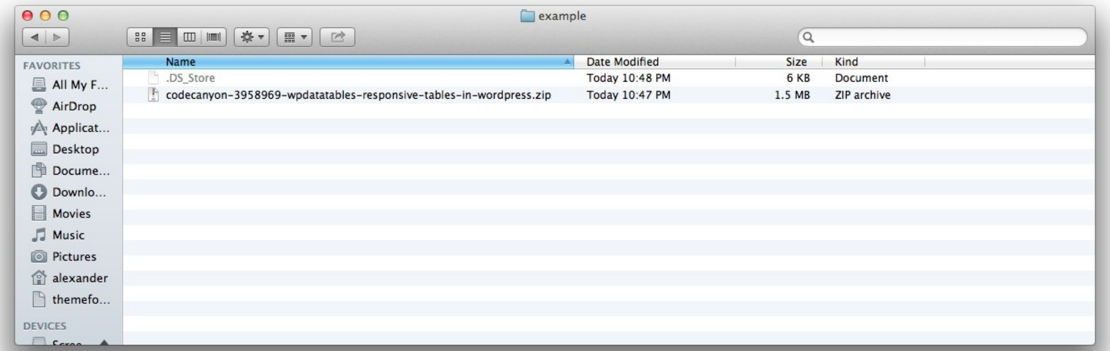
- **browse.inc.php** – template for the wp-admin browser page.
- **chart_js_template.inc.php** – template for Google Chart javascript block.
- **edit_table.inc.php** – template for the wp-admin table editor.
- **generic_table.inc.php** – generic template file combining javascript with rendered table.
- **wpdatatables_table_main.inc.php** – template for the table itself.
- **settings.inc.php** – template for the wp-admin wpDataTables settings page.
- **style_block.inc.php –** template for the style block generated according to the color and font settings from the plugin settings page.
- **filter_widget.inc.php** – template for the filtering widget.

1.3 Installation

This section covers the installation process of the plugin using FTP or ZIP file upload.

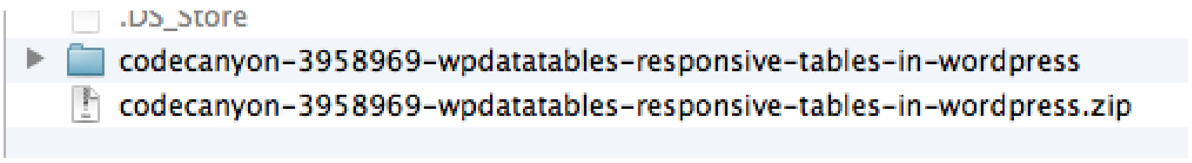
Installation of wpDataTables is pretty straightforward and you will not have any problems if you did it at least once before.

You will receive a file from CodeCanyon, which will be called something like "**codecanyon-3958969-wpdatatables-responsive-tables-in-wordpress.zip**":



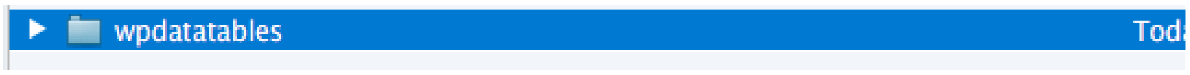
Name	Date Modified	Size	Kind
.DS_Store	Today 10:48 PM	6 KB	Document
codecanyon-3958969-wpdatatables-responsive-tables-in-wordpress.zip	Today 10:47 PM	1.5 MB	ZIP archive

Unzip this file, you will get a folder like this:



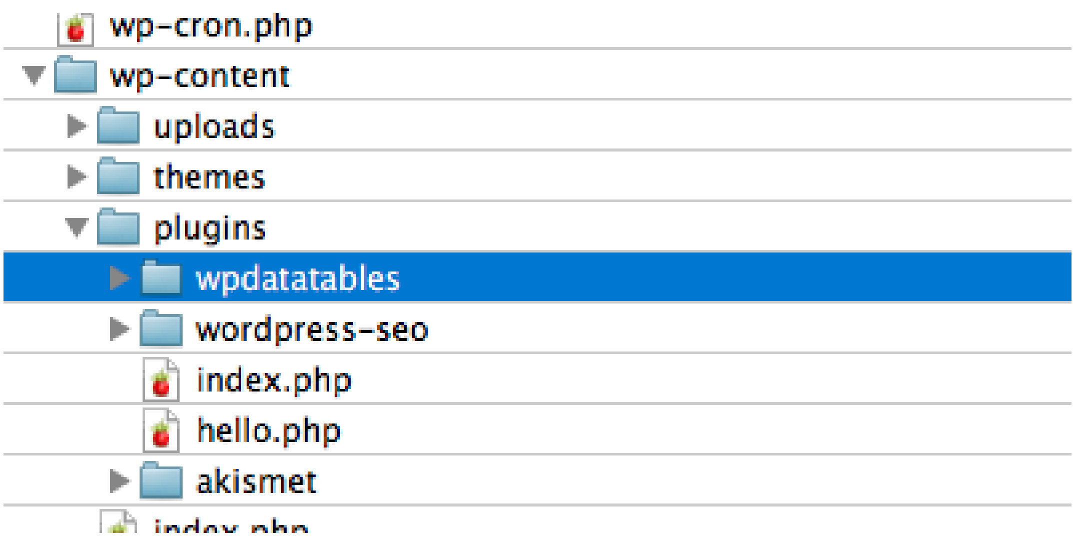
.DS_Store
codecanyon-3958969-wpdatatables-responsive-tables-in-wordpress
codecanyon-3958969-wpdatatables-responsive-tables-in-wordpress.zip

Probably it makes sense to rename this (not necessary, just to be more convenient). You can call the folder just "wpdatatables":

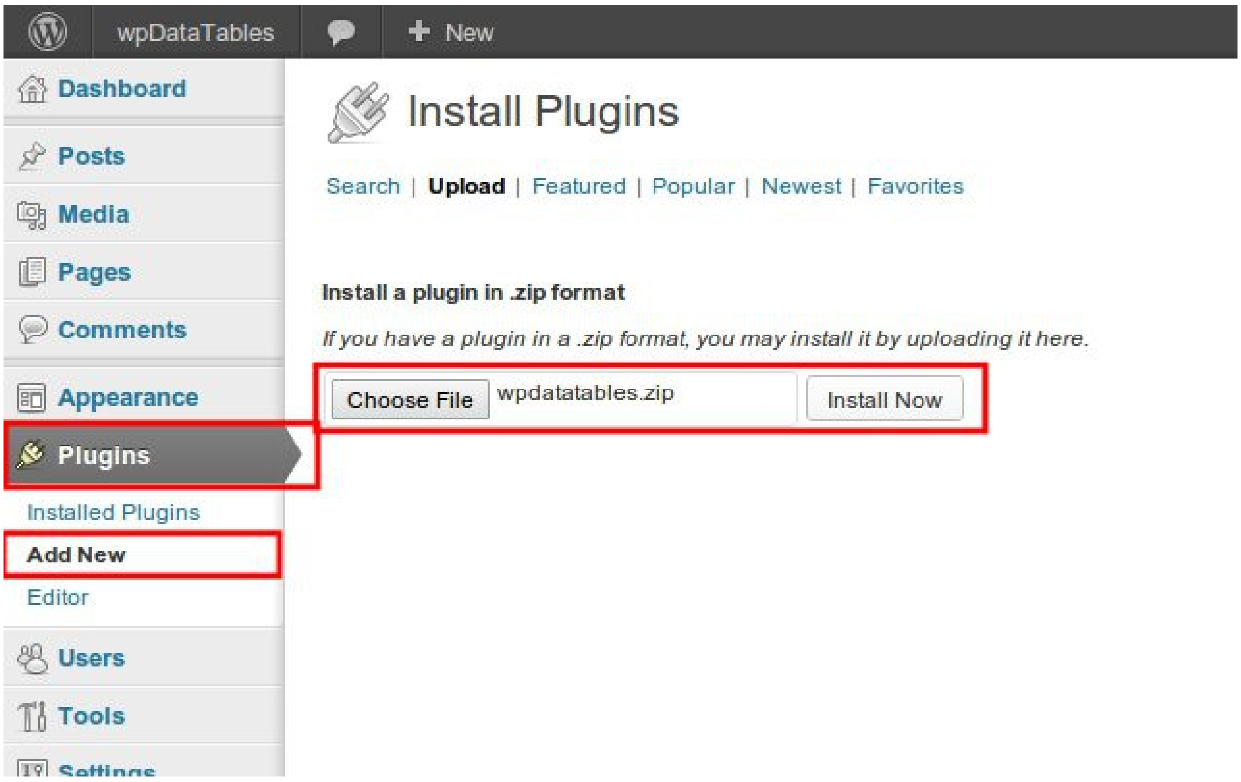


wpdatatables

Then you can upload your WordPress installation in some FTP browser (e.g. CyberDuck for Mac OS X), and upload this folder to your "wp-content/plugins" directory:



Or you can go to the wp-admin panel, to the Plugins tab, pick "**Add new**", "**Upload**", browse for the zip file with plugin, and then press "**Install now**":



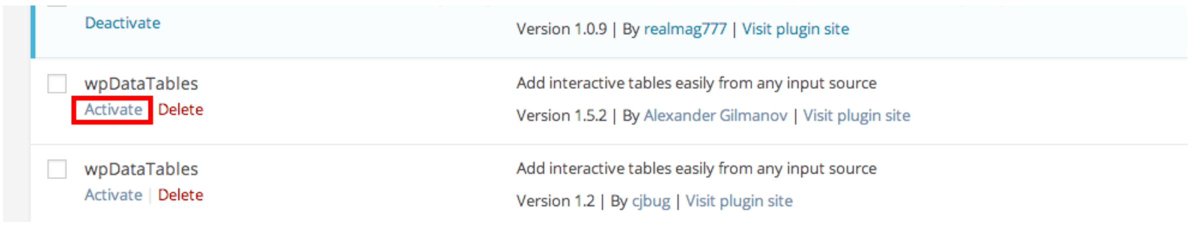
Install a plugin in .zip format

If you have a plugin in a .zip format, you may install it by uploading it here.

wpdatatables.zip

For the second option you'd need FTP account information, so just enter it on the next page, when WordPress requests it.

After that you should have the wpDataTables plugin in your list of plugins on the wp-admin plugin page:

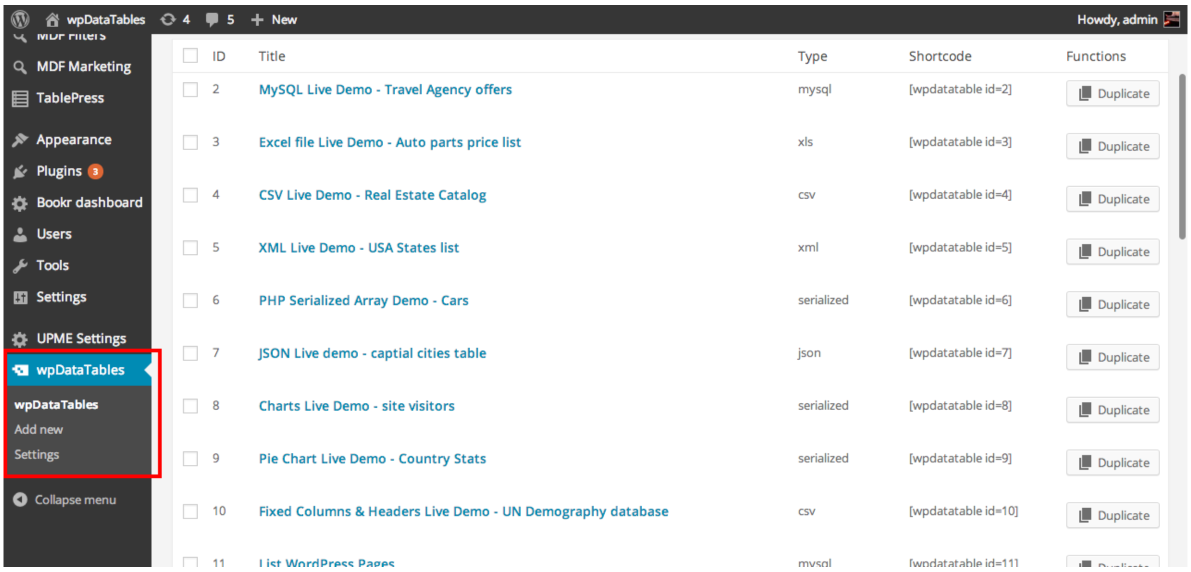


Deactivate | Version 1.0.9 | By realmag777 | Visit plugin site

wpDataTables
Activate | Delete
Add interactive tables easily from any input source
Version 1.5.2 | By Alexander Gilmanov | Visit plugin site

wpDataTables
Activate | Delete
Add interactive tables easily from any input source
Version 1.2 | By cjbug | Visit plugin site

Now, you just need to press “Activate”, after that the plugin will create its options and required tables in the database. After that you should see wpDataTables tab in the admin menu:



ID	Title	Type	Shortcode	Functions
2	MySQL Live Demo - Travel Agency offers	mysql	[wpdatatable id=2]	Duplicate
3	Excel file Live Demo - Auto parts price list	xls	[wpdatatable id=3]	Duplicate
4	CSV Live Demo - Real Estate Catalog	csv	[wpdatatable id=4]	Duplicate
5	XML Live Demo - USA States list	xml	[wpdatatable id=5]	Duplicate
6	PHP Serialized Array Demo - Cars	serialized	[wpdatatable id=6]	Duplicate
7	JSON Live demo - captial cities table	json	[wpdatatable id=7]	Duplicate
8	Charts Live Demo - site visitors	serialized	[wpdatatable id=8]	Duplicate
9	Pie Chart Live Demo - Country Stats	serialized	[wpdatatable id=9]	Duplicate
10	Fixed Columns & Headers Live Demo - UN Demography database	csv	[wpdatatable id=10]	Duplicate
11	List WordPress Pages	mysql	[wpdatatable id=11]	Duplicate

Done! Installation of wpDataTables is finished and now wpDataTables are ready to use. Probably you would like to configure them a little bit before use.

1.4 Updating

This section covers the update process of the plugin.

Note: starting from 1.5.5 an auto-update feature is included, but you still can use the way that is described here.

Updating wpDataTables is a quick and simple process which does not cause any problems.

1. Get the updated zip file from CodeCanyon.
2. Overwrite the contents of wpDataTables plugin folder on your host with the files you received from CodeCanyon. You can also delete the contents of the old directory and upload the contents of the new file.
3. Go to “plugins” page in your WordPress admin panel.
4. Deactivate the plugin
5. Activate the plugin again.

That’s it. All your old tables still exist in the WordPress Database, and you can also use all the new features.

1.5 Configuration

This section covers the settings which you can find on the settings page of the plugin.

wpDataTables plugin has several settings that you can define in the WordPress admin panel.

1. Use separate MySQL connection

Enable this setting if you want to build tables using data from DB or host which is different from the one that you use for your WordPress installation.

Do not check this checkbox if you want to use the WordPress database connection.

If you set this checkbox to checked, you will also need to provide the following data in order to be able to query the MySQL database:

- **MySQL host:** The address of MySQL host, domain or IP-address.
- **MySQL database:** The name of the database that you would like to use.
- **MySQL user**: Username which will be used for login.
- **MySQL password**: The password which will be used for login.

2. Interface language

wpDataTables comes with 45 world languages for the main interface features (provided by jQuery DataTables internationalisation). You can have the interface translated to your language by just choosing the language from this selectbox.

Important Only the main table features are covered by this package. It means that table tools ("Copy", "Print", "Excel", ...) and advanced filtering labels aren't translated, but you can translate them yourself if you want.

3. Render charts

This selectbox allows to choose where you prefer to output container with charts, for the tables where you enable them: above or below the table.

4. Base skin

wpDataTables has two built-in skins that you can use. Here you can choose one of them.

5. Render advanced filter

Here you can choose where the column filter will be rendered (if you do not decide to put it in a separate widget for some particular tables): in the table footer, or in the header.

6. Date format

This selectbox allows you to choose from several common date formats which will be used for the date columns.

7. Color and font settings

Here you can find a list of settings which will allow you to change the look of the tables rendered by the plugin. You can change:

- Table font color
- Header background color
- Header border color
- Header font color
- Header active and hover color
- Table inner border color
- Table outer border color
- Even row background color
- Odd row background color
- Hover row color
- Cell color in active (sorted) columns for even rows
- Cell color in active (sorted) columns for odd rows
- Background color for selected rows
- Buttons background color
- Buttons border color
- Buttons font color
- Buttons background hover color
- Buttons hover font color
- Modals (popups used in the plugin) font color
- Modals font color
- Modals background color
- Overlay background color (a background below popups which overlays the site content when they appear)
- Buttons hover border color

- Buttons and inputs border radius (in pixels, if you would like to have them rounded)
- Table font

You will find a short explanation beside each style setting, so you would know what element does it change.

> **Notice for developers:** settings page is using the template: templates/settings.inc.php, main PHP logic is in the file wdt_admin.php, main JS logic is inline and in assets/js/wpdatatables/wpdatatables_admin.js.

1.6 Quickstart guide: your first simple table

This section quickly takes you through creating your first very simple table.

Creating each wpDataTable takes three basic steps:

- Providing a data source
- Setting table parameters
- Pasting table shortcode in your post or page.

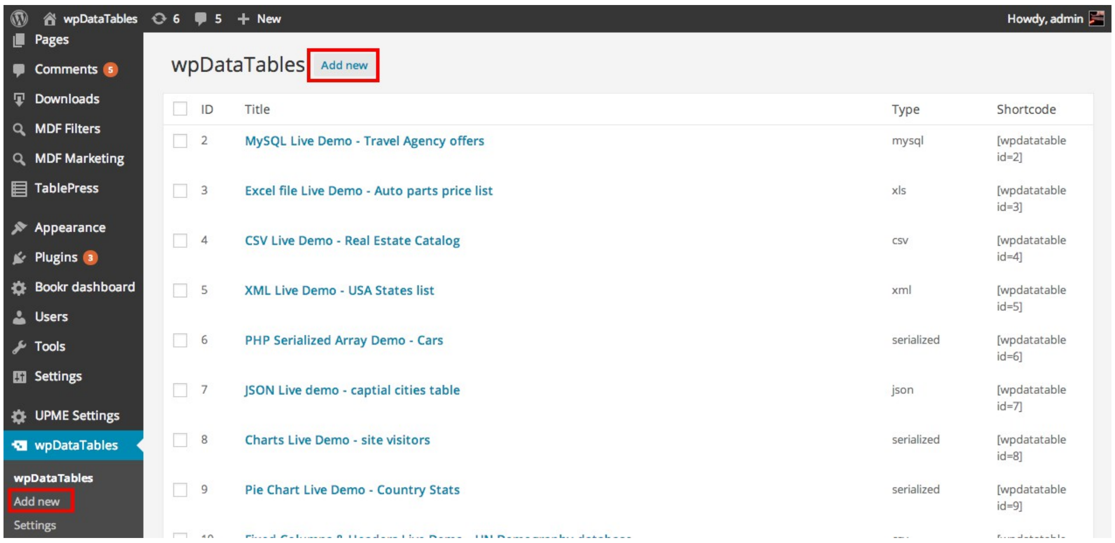
In this section we will go through these very briefly just for a demonstration of the work process. Below in this documentation you can find a detailed explanation of each and every checkbox and input field in the back-end and front-end.

Step 1: Providing a data source.

We will use a simple Excel file with a short list of auto parts for our example You can download it here.

- Go to WordPress Admin panel.

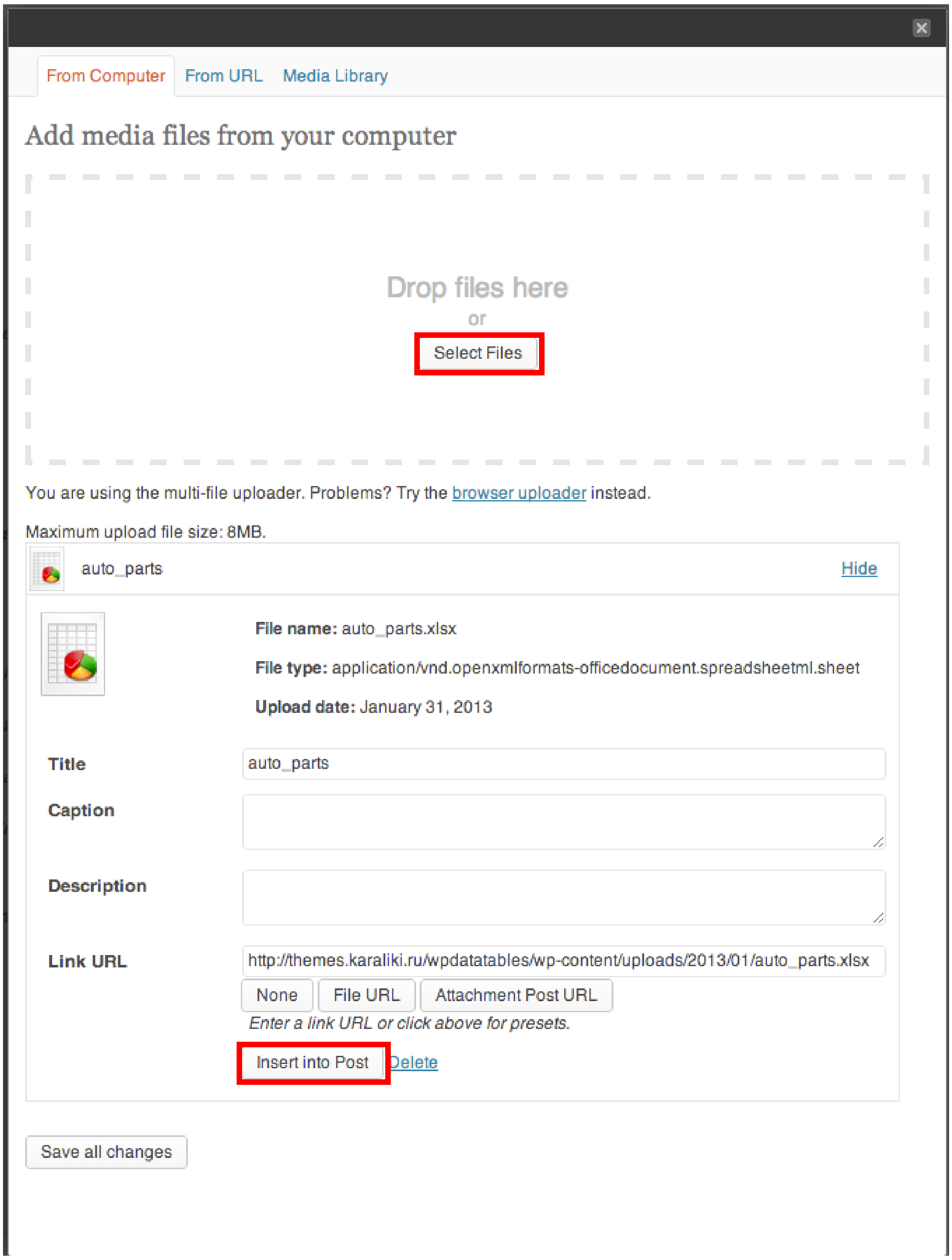
- Open wpDataTables -> Add New.



wpDataTables

ID	Title	Type	Shortcode
2	MySQL Live Demo - Travel Agency offers	mysql	[wpdatatable id=2]
3	Excel file Live Demo - Auto parts price list	xls	[wpdatatable id=3]
4	CSV Live Demo - Real Estate Catalog	csv	[wpdatatable id=4]
5	XML Live Demo - USA States list	xml	[wpdatatable id=5]
6	PHP Serialized Array Demo - Cars	serialized	[wpdatatable id=6]
7	JSON Live demo - captial cities table	json	[wpdatatable id=7]
8	Charts Live Demo - site visitors	serialized	[wpdatatable id=8]
9	Pie Chart Live Demo - Country Stats	serialized	[wpdatatable id=9]
10	Fixed Columns & Headers Live Demo - UN Demography database	csv	[wpdatatable

- Select a table type: Excel file.
- Click “**Upload file**”, upload the Excel file that we prepared, and then “**insert into post**”:



Add media files from your computer

You are using the multi-file uploader. Problems? Try the browser uploader instead.

Maximum upload file size: 8MB.

File name: auto_parts.xlsx

File type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet

Upload date: January 31, 2013

Title	auto_parts
Caption	
Description	
Link URL	http://themes.karaliki.ru/wpdatatables/wp-content/uploads/2013/01/auto_parts.xlsx

Enter a link URL or click above for presets.

Step 2: Setting table parameters.

This step is not mandatory, but usualy you would like to set the parameters, like table title, enable or disable the individual column filter, or just change the order of the columns by dragging them. When you are done click "Save table" and you will get a shortcode in the top of the page.

Step 3: Pasting the shortcode in your post or page.
Simply copy the shortcode that you get in the top of the edit page (something like **[wpdatatable id=1]**) and paste it in a new post or page. Then you can publish it, and open it in the front-end to see the result.

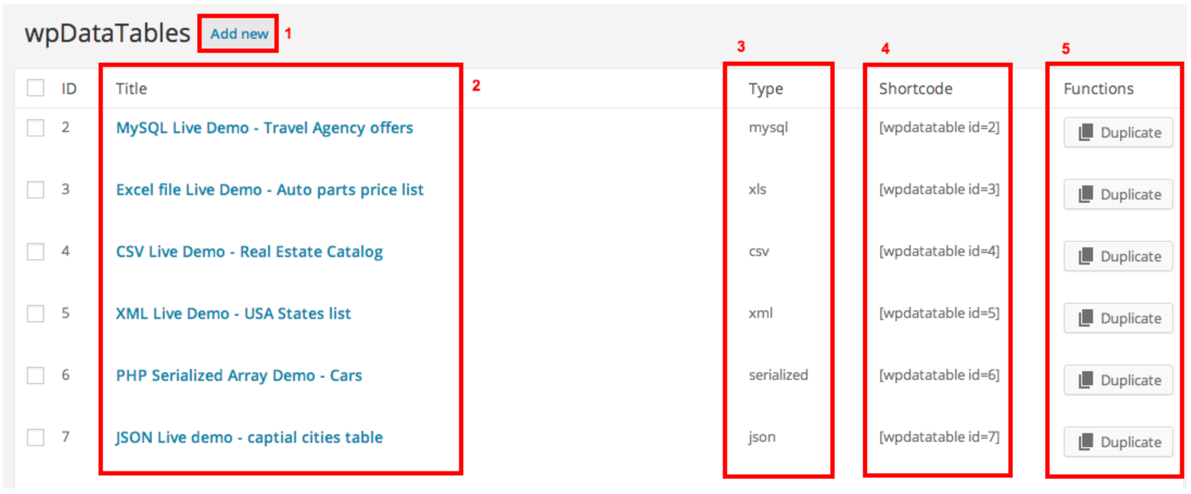
That's it! Your first table is ready to use.

2. Admin panel

This chapter will give an overview on all of plugin's back-end (WP Admin) pages, all the settings, checkboxes, inputs, selectboxes, and everything else you might see there.

2.1 Overview page

This section describes the admin table overview page.



ID	Title	Type	Shortcode	Functions
2	MySQL Live Demo - Travel Agency offers	mysql	[wpdatatable id=2]	Duplicate
3	Excel file Live Demo - Auto parts price list	xls	[wpdatatable id=3]	Duplicate
4	CSV Live Demo - Real Estate Catalog	csv	[wpdatatable id=4]	Duplicate
5	XML Live Demo - USA States list	xml	[wpdatatable id=5]	Duplicate
6	PHP Serialized Array Demo - Cars	serialized	[wpdatatable id=6]	Duplicate
7	JSON Live demo - captial cities table	json	[wpdatatable id=7]	Duplicate

wpDataTables admin Overview page is the default page which is opened when you open wpDataTables in wp-admin panel. It is used to list all the existing tables in your WordPress DB.

Tables overview page is using standard WordPress table admin interface which you can also see in the Pages or Posts admin page.

It has the following elements:

1. **Add new button** – this button opens the page for creating a new wpDataTable.
2. **Table titles column** – this column lists the titles of the tables which you previously created. You will see a "No tables" label when you open it first. By clicking one of the titles you will open the table editor.
3. **Table types column** – in this column you can see which data source was used to create each particular column. Data sources can be: MySQL, XLS, CSV, XML, Serialized PHP array, and JSON.
4. **Shortcodes column** – in this column you can see the shortcodes for each table, so you can easily copy&paste them to your WordPress pages or posts.
5. **Functions column** – in this column you can click on the "Duplicate" button to create a duplicate of the table.

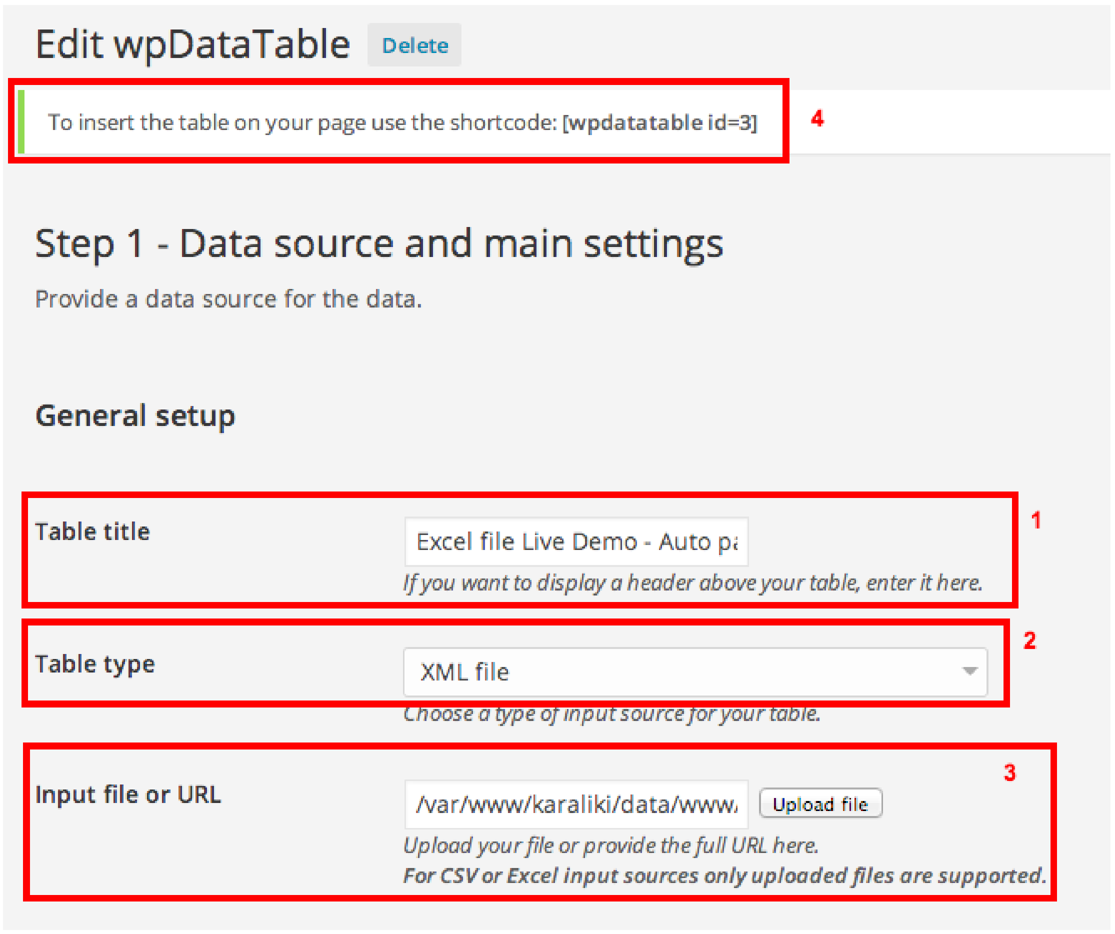
> **Notice for developers:** overview page is using the template: templates/browse.inc.php, logic is in the file controllers/wdt_admin.php

2.2 Table edit page – step 1 -table settings

This section describes all the settings, elements and sections of the table edit page – step 1 (table settings).

Table edit page contains two steps: in first step you provide the data source and configure the settings for the table in general.

General setup (XML, CSV, Excel, JSON, Serialized PHP Array)


Edit wpDataTable

To insert the table on your page use the shortcode: [wpdatatable id=3]

Step 1 - Data source and main settings

Provide a data source for the data.

General setup

Table title: Excel file Live Demo - Auto pa

If you want to display a header above your table, enter it here.

Table type: XML file

Choose a type of input source for your table.

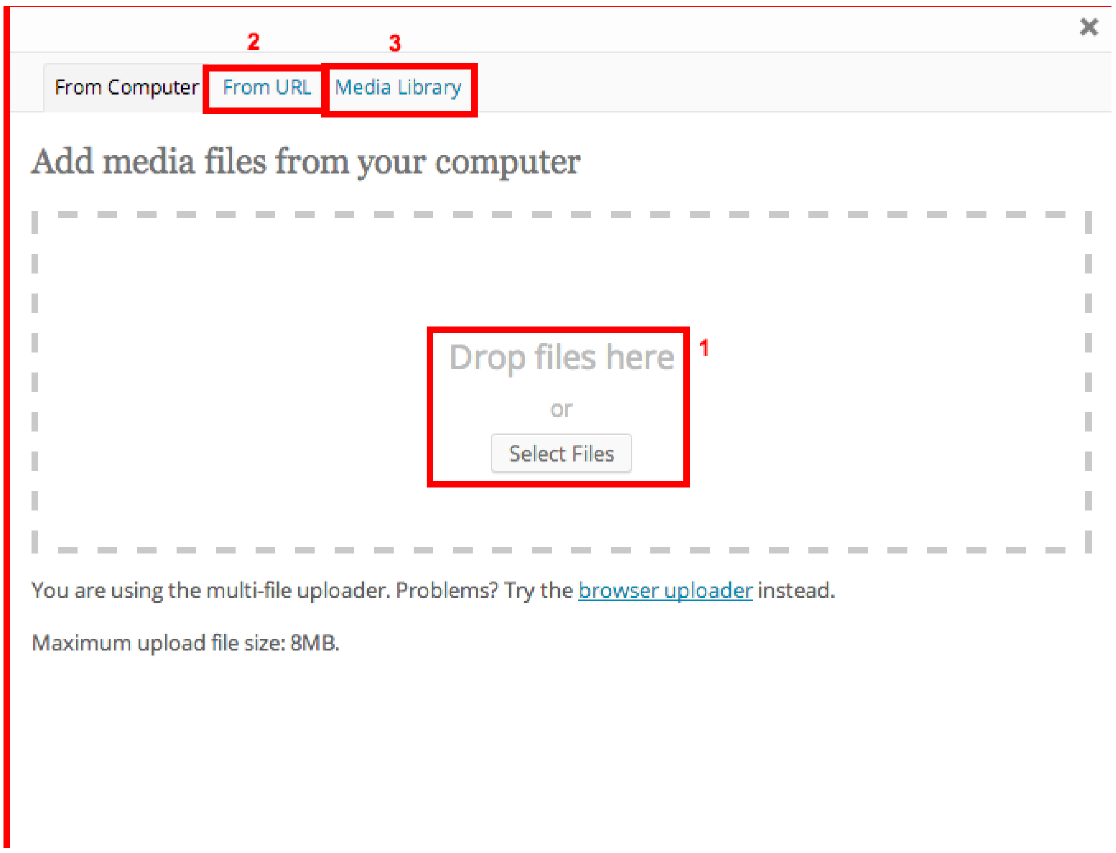
Input file or URL: /var/www/karaliki/data/www.

Upload your file or provide the full URL here.

For CSV or Excel input sources only uploaded files are supported.

The general setup section in case for Excel, CSV, XML, JSON or Serialized PHP array contains the following elements:

1. **Table title** – a text input field where you can provide a title for your table. This title will be later displayed in the overview page (see docs section 2.1), and above the table itself in the front-end.
2. **Table type** – a dropdown selectbox where you can choose one of the supported data sources: Excel file, CSV file, XML, JSON, or serialized PHP array.
3. **Input file or URL.** – a text input box used to specify the input file location. For serialized PHP array you would need to provide a direct URL to the file which returns the serialized array; for Excel, CSV, XML or JSON you would need to press the “Upload file” button, which would open the media uploader:

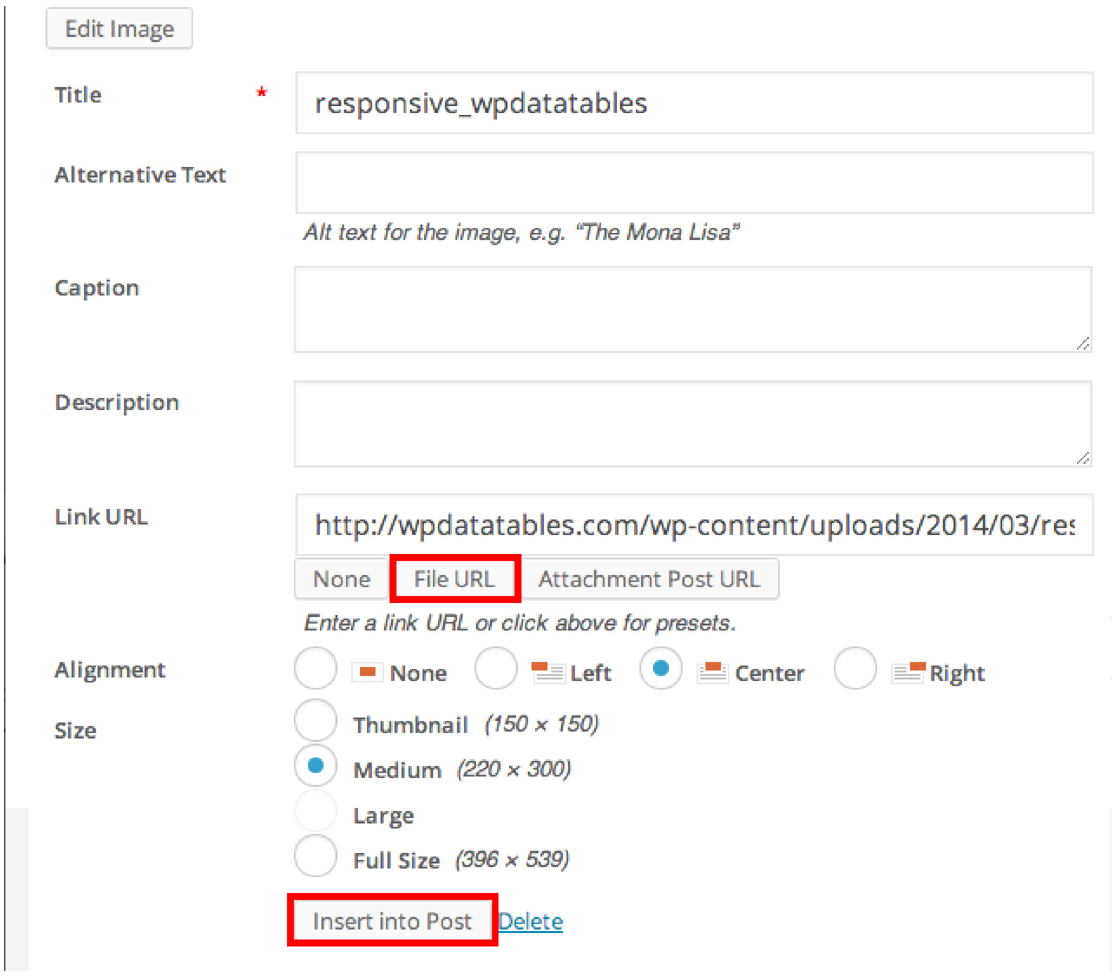


Add media files from your computer

You are using the multi-file uploader. Problems? Try the browser uploader instead.

Maximum upload file size: 8MB.

In the media uploader you can either upload a file (by pressing **button 1** – or just drag&drop a file to this section), provide an URL to the file that you would like to use (in **tab 2**), or if the file is already in your Media library (probably you uploaded it before) you can locate it in the **tab 3**. When you upload, or locate the file, you would then need to make sure that a “File URL” option is selected (so that a direct file URL would be in the text field, not an attachment URL with “?attachment_id=xxx” in the end).


Edit Image

Title *	responsive_wpdatatables
Alternative Text	
	Alt text for the image, e.g. "The Mona Lisa"
Caption	
Description	
Link URL	http://wpdatatables.com/wp-content/uploads/2014/03/res
	None / File URL / Attachment Post URL
	Enter a link URL or click above for presets.
Alignment	None / Left / [✓ Center] / Right
Size	Thumbnail (150 × 150)
	[✓ Medium (220 × 300)]
	Large
	Full Size (396 × 539)

When all is done you would need to press the "Insert into Post" button, so that the media uploader would be closed, and the file URL would be returned to the "Input file or URL" input field.

4. **Shortcode** – this block is invisible for new tables. When the table is saved and general info is set, the table gets a new ID, and a shortcode is generated. You can copy this shortcode and paste it to your post or page to render the table in the WordPress frontend.

General setup (MySQL table type)

General setup section looks slightly different in case of MySQL table type.

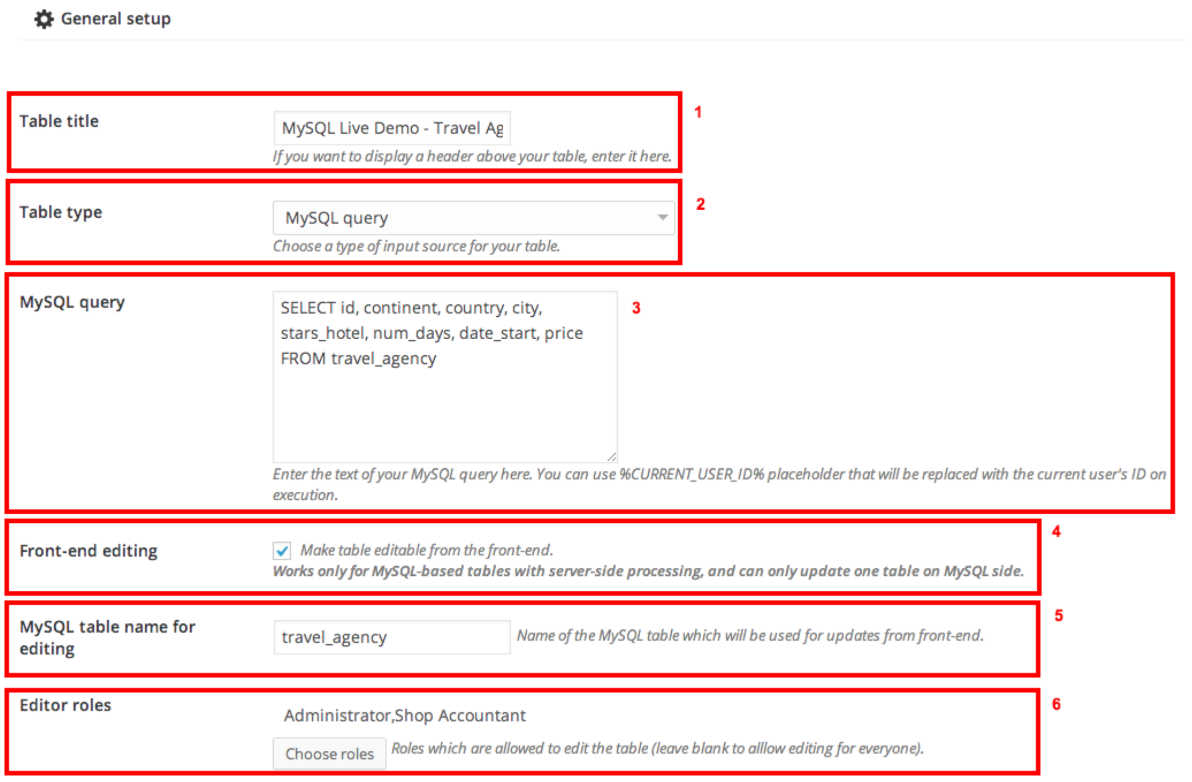

General setup

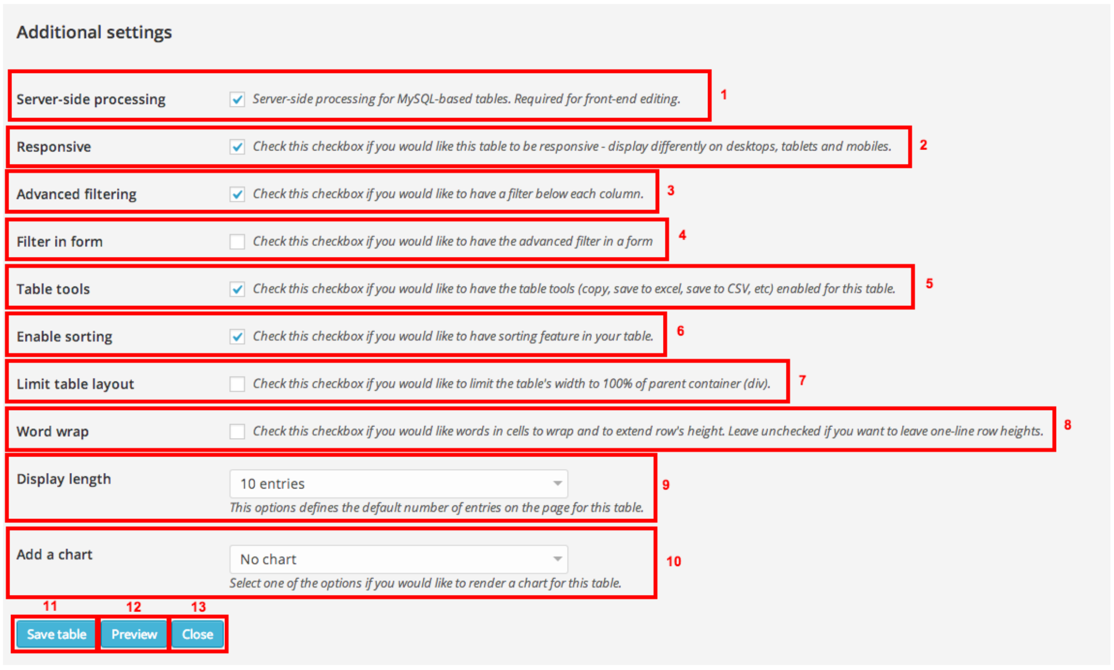
Table title	MySQL Live Demo - Travel Ag *If you want to display a header above your table, enter it here.*	1
Table type	MySQL query *Choose a type of input source for your table.*	2
MySQL query	SELECT id, continent, country, city, stars_hotel, num_days, date_start, price FROM travel_agency *Enter the text of your MySQL query here. You can use %CURRENT_USER_ID% placeholder that will be replaced with the current user's ID on execution.*	3
Front-end editing	✓ *Make table editable from the front-end.* ***Works only for MySQL-based tables with server-side processing, and can only update one table on MySQL side.***	4
MySQL table name for editing	travel_agency *Name of the MySQL table which will be used for updates from front-end.*	5
Editor roles	Administrator,Shop Accountant Choose roles *Roles which are allowed to edit the table (leave blank to alllow editing for everyone).*	6

1. **Table title** – a text input field where you can provide a title for your table. This title will be later displayed in the overview page (see docs section 2.1), and above the table itself in the front-end.
2. **Table type** – a dropdown selectbox where you can choose one of the supported data sources: Excel file, CSV file, XML, JSON, or serialized PHP array.
3. **MySQL query** – a textarea input field, where you can provide the SQL query text that is supposed to return the dataset for the table. Please keep in mind the following tips for using MySQL in wpDataTables:
   - It is better not to use quotes in the column names.
   - It is better not to put the semicolumn in the end of the query
   - If you have a large datasource, and would like to use the server-side processing feature of wpDataTables, please keep in mind that server-side processing allows using only simple “SELECT * FROM table” queries, so if you need to return a result from some

    complicated query (e.g. joining two or more tables, or having a "WHERE" part, etc) please prepare a MySQL view (stored query) on the database side, so later you would build a wpDataTable based on a simple SELECT query to this view
    - Do not use "SORT BY" section in the query, since wpDataTables has its own sorting mechanism.
4. **Front-end editing** – this check-box would set the table as editable. Please keep in mind the following limitations for the editable tables:
    - **Only one MySQL table can be editable at a time**. Which means if you have a query with joins from several tables you won't be able to make it editable.
    - **For editing you would need to have a unique ID field**. Usually autoincrement integer, set as primary key.
5. **MySQL table name for editing:** in this input field (available only for editable tables) you would need to provide the name of MySQL table once again. This is needed for the update/insert queries.
6. **Editor roles:** you can use this block (available only for editable tables) to choose the user roles that will be able to edit this table.

Additional settings

The additional settings block allows you to configure all the features for the table in general.



Additional settings

Server-side processing	✓ Server-side processing for MySQL-based tables. Required for front-end editing.	1
Responsive	✓ Check this checkbox if you would like this table to be responsive - display differently on desktops, tablets and mobiles.	2
Advanced filtering	✓ Check this checkbox if you would like to have a filter below each column.	3
Filter in form	☐ Check this checkbox if you would like to have the advanced filter in a form	4
Table tools	✓ Check this checkbox if you would like to have the table tools (copy, save to excel, save to CSV, etc) enabled for this table.	5
Enable sorting	✓ Check this checkbox if you would like to have sorting feature in your table.	6
Limit table layout	☐ Check this checkbox if you would like to limit the table's width to 100% of parent container (div).	7
Word wrap	☐ Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave unchecked if you want to leave one-line row heights.	8
Display length	10 entries This options defines the default number of entries on the page for this table.	9
Add a chart	No chart Select one of the options if you would like to render a chart for this table.	10

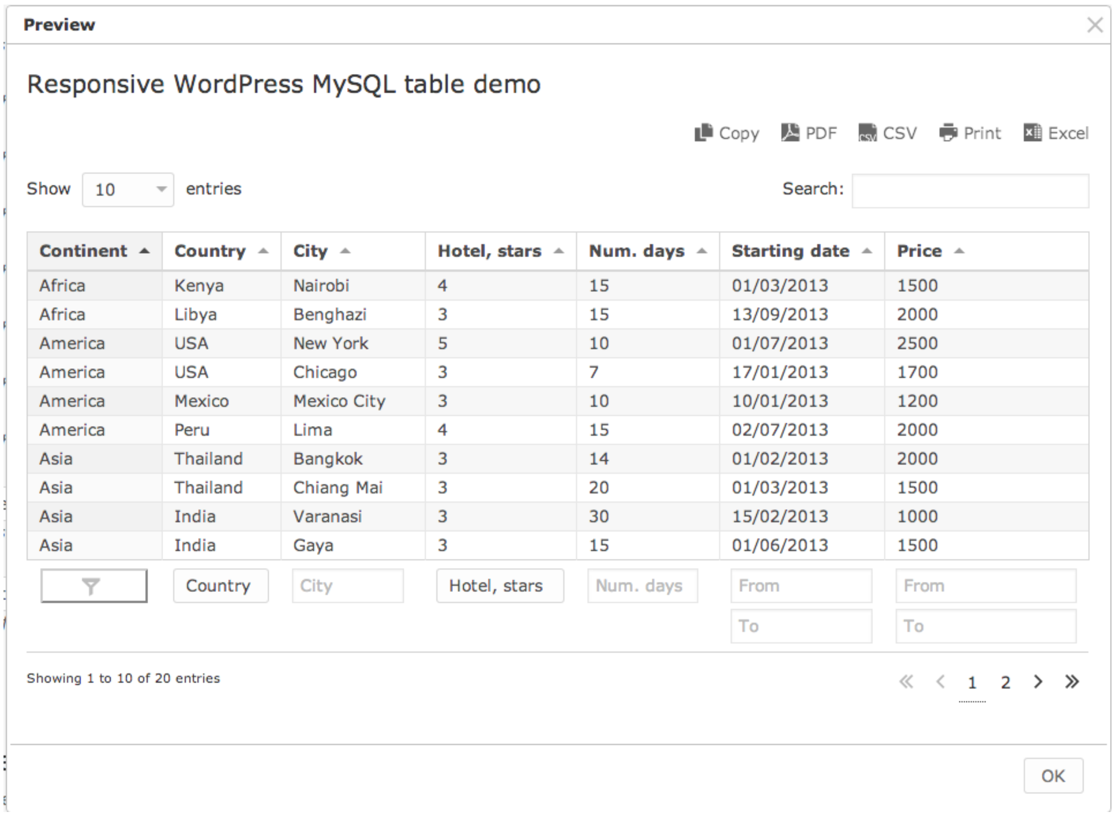
1. **Server-side processing.** This checkbox is enabled only for MySQL-based tables; it enables the server-side processing feature. This feature means that only several rows (10 by default) will be fetched from the server at a time, and each time you switch to the next page, or sort by some column, or search, or filter the table, it will send an AJAX request to the server, return and render the same amount of rows again. It makes sense for large datasets, smaller ones would work even slower. See documentation section 5.1 for more details
2. **Responsive.** This feature enables responsiveness for the table, which means that it will be displayed differently on desktops and mobile devices. If your site supports responsiveness (adapts the interface to mobile devices) it makes sense to enable this feature. See documentation section 5.2 for more details.
3. **Advanced filtering.** This feature enables a possibility to have an individual filter for each column. The filter can be rendered below the column, in the header of the column, in a form above the table,

wpDataTables

or in the widget which you can put whereever you need. See documentation section 5.3 for more details.

4. **Filter in form.** Enabling checkbox will put the filter in a form that is separated from table, or in a widget, which you can put in a sidebar, footer, or wherever your table allows. See documentation section 5.3 for more details.
5. **Table tools.** Checking this checkbox will enable table tools block for the table frontend (export to CSV, export to Excel, export to PDF, print view). See documentation section 5.4 for more details.
6. **Enable sorting.** Checking this checkbox would allow sorting the data in the tables by clicking on the header of each column in ascending and descending order. See documentation section 5.5 for more details.
7. **Limit table layout.** By default the table in the front-end is as wide as it is needed by the content, and may exceed the parent container width. This checkbox forces the table to fit in the parent container. This checkbox must be checked if you want to set the widths for the columns. See documentation section 5.6 for more details.
8. **Word Wrap.** Checking this checkbox will make the words wrap and extend the cells height if the content does not fit in one line.
9. **Display length.** In this dropdown you can choose whether you want to display 10, 25, 50, 100 or all table entries in the front-end by default.
10. **Add a chart.** This dropdown allows to select one of the options for the chart: No Chart, Area Chart, Bar Chart, Column Chart, Line Chart, Pie chart. By default charts aren't enabled. See documentation chapter 8 for info on charts.
11. **Save table.** This button saves the table settings and enables the preview and Step 2 if this is a new table. Please note that if you changed the data source or MySQL query for MySQL-based tables, you will also need to configure settings in Step 2 once again.

12. **Preview.** This button enables preview mode: a popup with the table will appear when you click on it, so you could see how the table will look in the front-end.



Responsive WordPress MySQL table demo

Show 10 entries

Search:

Continent	Country	City	Hotel, stars	Num. days	Starting date	Price
Africa	Kenya	Nairobi	4	15	01/03/2013	1500
Africa	Libya	Benghazi	3	15	13/09/2013	2000
America	USA	New York	5	10	01/07/2013	2500
America	USA	Chicago	3	7	17/01/2013	1700
America	Mexico	Mexico City	3	10	10/01/2013	1200
America	Peru	Lima	4	15	02/07/2013	2000
Asia	Thailand	Bangkok	3	14	01/02/2013	2000
Asia	Thailand	Chiang Mai	3	20	01/03/2013	1500
Asia	India	Varanasi	3	30	15/02/2013	1000
Asia	India	Gaya	3	15	01/06/2013	1500
	Country	City	Hotel, stars	Num. days	From / To	From / To

Showing 1 to 10 of 20 entries

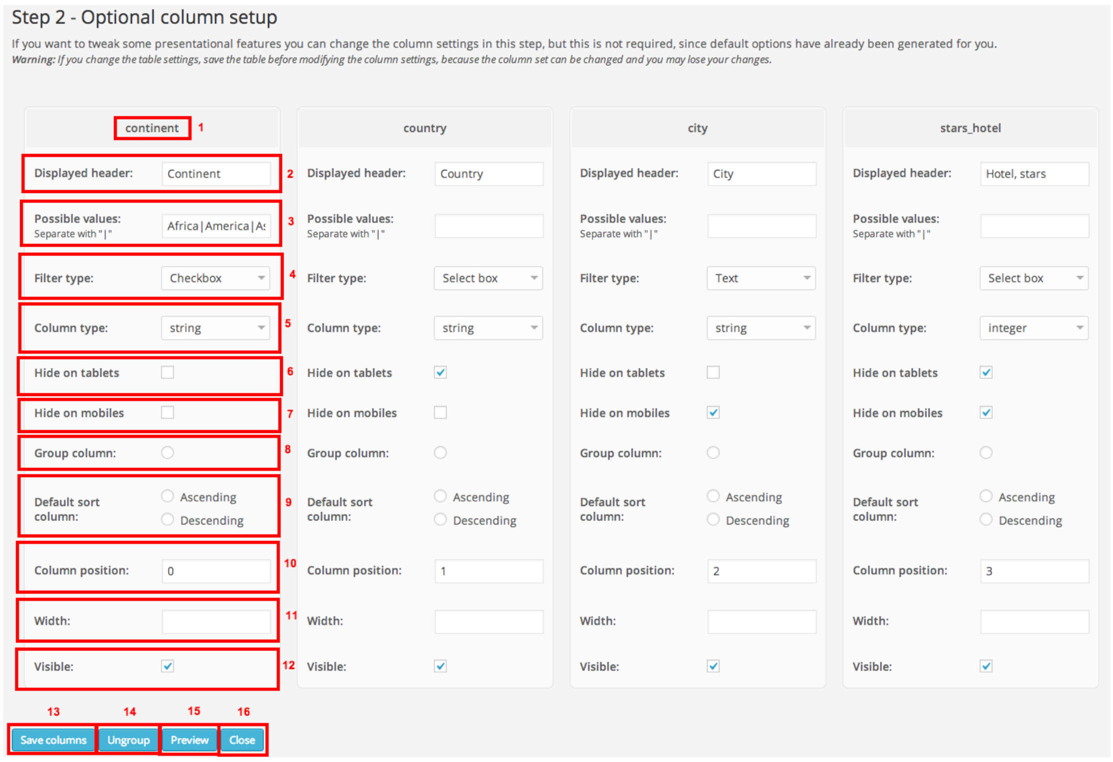
« ‹ 1 2 › »

13. **Close.** This button cancels all the changes that you did (after a confirmation) and opens the overview page.

2.3 Table edit page – step 2 -column settings

This section describes all the settings, elements and sections of the table edit page – step 2 (column settings).

Step 2 of the table edit page becomes visible when you save the general settings in the new table. In this step you can specify the detailed settings per each column. Below you can see the description of each input element of this step:



Step 2 - Optional column setup

If you want to tweak some presentational features you can change the column settings in this step, but this is not required, since default options have already been generated for you.
Warning: *If you change the table settings, save the table before modifying the column settings, because the column set can be changed and you may lose your changes.*

	continent	country	city	stars_hotel
Displayed header:	Continent	Country	City	Hotel, stars
Possible values: Separate with "|"	Africa|America|As			
Filter type:	Checkbox	Select box	Text	Select box
Column type:	string	string	string	integer
Hide on tablets	☐	✓	☐	✓
Hide on mobiles	☐	☐	✓	✓
Group column:	○	○	○	○
Default sort column:	○ Ascending ○ Descending	○ Ascending ○ Descending	○ Ascending ○ Descending	○ Ascending ○ Descending
Column position:	0	1	2	3
Width:				
Visible:	✓	✓	✓	✓

1. **Original column header.** In this part you will see the original column heading from the input data source – MySQL table column name, or heading from Excel or CSV file. This label cannot be changed within wpDataTables, it is used to identify the original source column.
2. **Displayed header.** This is the header that will be shown in the frontend. E.g. if your original MySQL column name is "usr1_from" you can put here "User one from" and this name will be shown in the header.
3. **Possible values.** This is an optional input field. Here you can provide all the possible values for the column. These values will be used in dropdown filter (see documentation section 5.3 for more details), and also can be used in front-end editor for the dropdown input type (see documentation section 7.2 for more details).
4. **Filter type.** This dropdown is only visible if you set the advanced filtering enabled in Step 1. Here you can choose how to display the filter for this particular column: none, text, number, number range,

date range, select box, or checkbox. See documentation section 5.3 for more details on filtering.

5. **Column type.** This dropdown defines the data type in the column; which reflects in rendering and the sorting rules for the column. In most of the cases the column type will be defined correctly on the table create step, but you can change them if you like. See documentation chapter 6 for more info on different column types.
6. **Hide on tablets.** This checkbox is only visible if you set the responsive feature enabled for this table. If this checkbox is checked this column will not be visible on tablets. See documentation section 5.2 for more information about responsive tables.
7. **Hide on phones.** This checkbox is only visible if you set the responsive feature enabled for this table. If this checkbox is checked this column will not be visible on mobile devices. See documentation section 5.2 for more information about responsive tables.
8. **Group column.** This radio button is used to select a column which will be used as a column for row grouping. Only one column can be chosen for grouping; for ungrouping please choose the button (14). Please note that grouping does not work correctly with TableTools so you would need to decide which extensions you prefer. See the demo on row grouping here ; check documentation section 6.8 for more info on column grouping.
9. **Default sort column.** These two checkboxes are used to define which column will be used for initial sorting, in other words, when the table opens in the front-end it will be sorted in ascending or descending order by the column that you define. By default it will be sorted by first column ascendingly.
10. **Column position.** This input field displays the current index of the column (zero-based). It can be also used to define the column

order manually, but it's easer to just drag& drop the columns around.

11. **Width.** This input can be used to specify the column width in eiter percents or pixels: but please keep in mind that you need to set the **Limit table layout** checkbox as enabled if you decide to do so, or these settings will be ignored.
12. **Visible.** This checkbox defines if this column will be visible in the front-end. Please keep in mind, that invisible column still are present in the page, so if e.g. you are querying a database, and choose to show 5 columns out of 15, other 10 are still present in browser's memory. So it is better not to query only columns you might need (e.g. "SELECT column1, column2, column3 FROM table" instead of "SELECT * FROM table").
13. **Save columns.** This buttons sends the save request to the server, and persists the current column settings in the DB.
14. **Ungroup.** This button is used to untick the "Group column" radio (8) if you accidentally ticked it. Or if you decide not to use this feature.
15. **Preview.** Toggles table preview (see description above in 2.2)
16. **Close.** This button cancels last unsaved changes (after a confirmation) and returns you to the table overview page.

> **Notice for developers:** table edit (and new table) page is using the template: templates/edit_table.inc.php, main PHP logic is in the file controllers/wdt_admin.php, main JS logic is inline.

3. Frontend

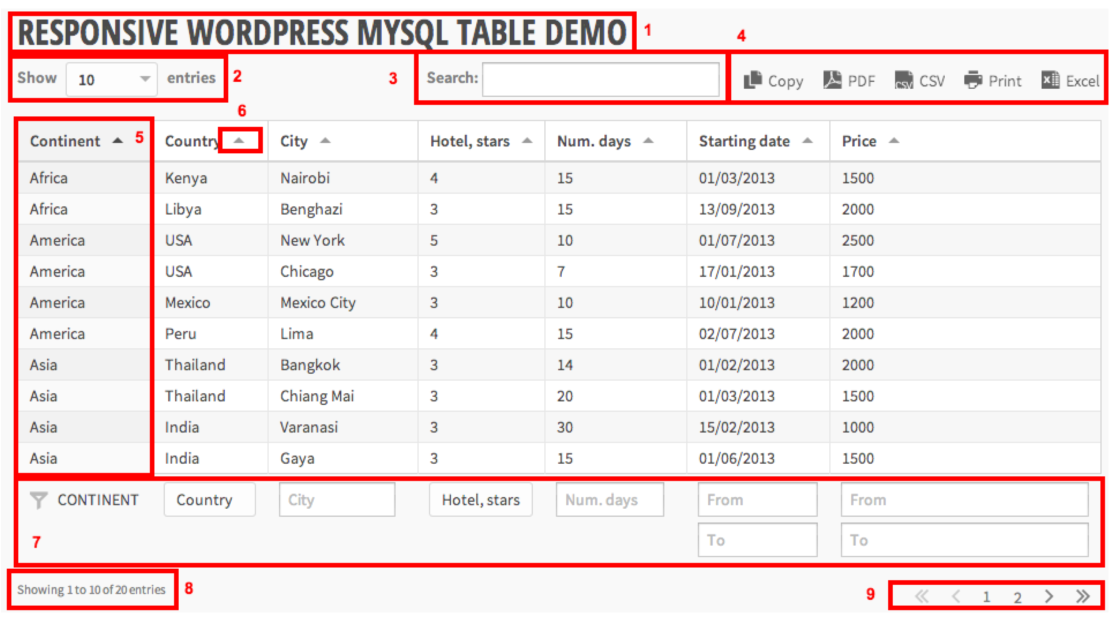
This chapter will give an overview of the front-end look of the plugin: the table itself, additional input and filtering elements.

3.1 Table

This section gives on overview of the frontend look of the table generated by the plugin, it's main elements and blocks.

You can see the generated table frontend view when you open a post or page to which you inserted the shortcode in WordPress frontend. This is the main plugin view.

Column headings and order are taken from the settings you provide in the back-end (see documentation section 2.3 for more info on configuring the columns).


RESPONSIVE WORDPRESS MYSQL TABLE DEMO

Show 10 entries

Search:

Continent	Country	City	Hotel, stars	Num. days	Starting date	Price
Africa	Kenya	Nairobi	4	15	01/03/2013	1500
Africa	Libya	Benghazi	3	15	13/09/2013	2000
America	USA	New York	5	10	01/07/2013	2500
America	USA	Chicago	3	7	17/01/2013	1700
America	Mexico	Mexico City	3	10	10/01/2013	1200
America	Peru	Lima	4	15	02/07/2013	2000
Asia	Thailand	Bangkok	3	14	01/02/2013	2000
Asia	Thailand	Chiang Mai	3	20	01/03/2013	1500
Asia	India	Varanasi	3	30	15/02/2013	1000
Asia	India	Gaya	3	15	01/06/2013	1500
CONTINENT	Country	City	Hotel, stars	Num. days	From / To	From / To

Showing 1 to 10 of 20 entries

1. **Table title.** If you have set a title on the table edit page, you will see it here in a <h3> HTML element.
2. **Selector for number of entries.** This is a dropdown selectbox, where the front-end user can choose how many rows does he want to have in the table. If you specify a number of rows in the table edit page in the plugin back-end, this value will be pre-selected by default.
3. **Global table search field.** This input field allows front-end users to filter the table contents by some value; the algorithm searches

within all cells, and shows a row if at least one of the cells contains the value entered in the search field. If the table does not use server-side processing feature, it will also search by multiple parts, e.g. input “Asia 12” will search by both values in all cells. In tables with server-side processing this is not implemented because of performance reasons.

4. **TableTools block.** This block contains buttons for CSV, Excel, PDF export, also copy to clipboard function and print view. See documentation section 3.4 for more details about TableTools extension.

   If the table is editable, the Add new, Edit and Delete buttons are also inserted in this block.
5. **Active (sorted) column.** The column that is slightly highlighted is the current active column, which means that the table data is sorted by the values from this column in ascending or descending order. You can see the direction of the small triangle near the column heading: up means ascending order, down means descending. Sorting rules depend on the column type that you have provided, see documentation section 6 for more info on column types and features.
6. **Sorting trigger button.** The same triangle can be used to activate the sorting on a different column, or change the sorting direction. If you press control when clicking on the sorting trigger button you can sort by two columns.
7. **Advanced filtering block.** The advanced filtering block is visible if you enabled advanced filtering for the table in the back-end (see documentation section 2.2 for more details on the table setup). Each column gets a separate filtering input, which allows users to filter

the table data by the values in this column. See documentation section 3.3 for detailed overview of the advanced filtering block.

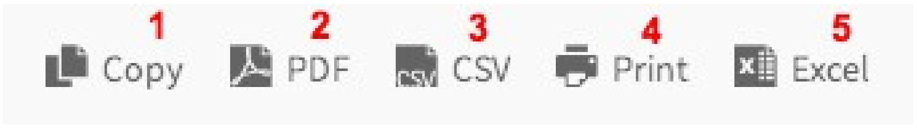
8. **Info section.** In this section you can see how many rows does the table have, and which rows are currently shown.
9. **Pagination controls.** Using this controls user scan navigate through pages. Each page contains as many rows as the user defines per each table.

3.2 Table tools

This section gives on overview of the table tools block which is rendered in the top right corner of the table.

Table tools is a third-party library, based on a Flash engine, which allows users to have some extra features on the table frontend: copy the table content to the clipboard, export the table content to a PDF, CSV or an Excel file, or just to hide everything except for the table on the page, to get a print view. See section 5.4 for more info about TableTools library.

When you create an editable table, the editing controls are also rendered in the TableTools block. See documentation sections 3.4 and 7 for more info on editable tables.



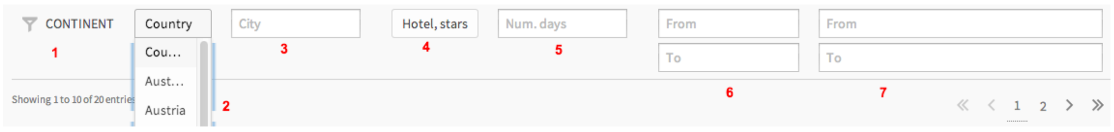
1. **Copy to clipboard** – this button allows front-end users to copy the data from the table to the clipboard. An important limitation: for tables with server-side processing feature enabled only the visible rows will be copied. So if you need to copy all rows you would first

need to select “Show: all rows” feature in the number of entries selector.

2. **Export to PDF** – this button creates a PDF file with the data from the table. There’s a same limitation here: only visible rows will be exported from a table with server-side processing feature enabled. Also please keep in mind that Row Grouping does not combine with Export to PDF feature, so if you have grouping columns, they will still be rendered as usual columns in PDF.
3. **Export to CSV –** this button creates a CSV file with data from the table. Only visible rows will be exported if the table has server-side processing feature enabled. Row grouping does not reflect in the resulting CSV file.
4. **Print view –** this button hides everything on the page except for the table so that the table can be printed.
5. **Export to Excel –** this button creates an Excel file with data from the table Only visible rows will be exported if the table has server-side processing feature enabled. Row grouping does not reflect in the resulting Excel file. Please note: the generated file is not a “real” XLS, it’s a CSV with XLS extension, but most Excel installations will handle it correctly.

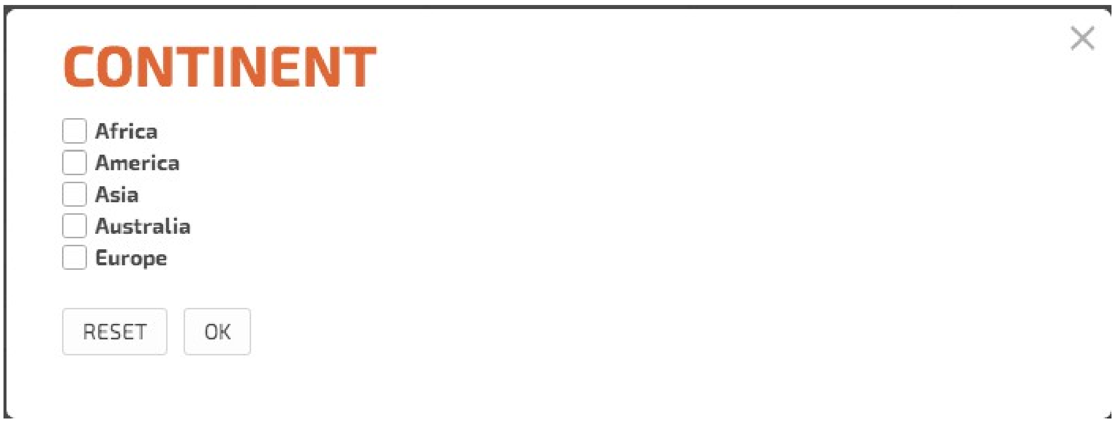
3.3 Filter block

This section gives an overview of the advanced filtering block, which is rendered below the table, in a form above the table, or a widget according to the setting that you define on the table edit page.



Advanced filtering block is a block where filter inputs per each individual column are rendered. You can define a filter type per each column (see documentation section 2.3). Different filter input types produce different filtering behavior. Please see documentation section 5.3 for detailed info on the advanced filtering feature.

1. **Checkbox filter** – this filter type outputs a button on the frontend. When user clicks this button he gets a popup window, where he can choose one ore more values for the column. Only the rows that contain these values in the filtered column will be visible in the table.



2. **Dropdown filter** – this filter type outputs a dropdown selectbox. Users can choose one value per column, only the rows that contain these values in the filtered column will be visible in the table.
3. **Text filter** – this filter type renders a simple text input, where front-end users can type whatever they want. Only the rows, where the cells of the filtered column contain the text which user typed in will be then shown in the table.
4. **Number filer** – this filter type renders an input element where users can type in different numbers. Only the rows, where the cells of the filtered table contain this exact number will be shown in the table.
5. **Number filter** (see previous one)
6. **Date range filter** – this filter type renders two datepicker inputs. Users can choose a starting date and (or) ending date. This filter works correctly only on column date types. The rows where the date

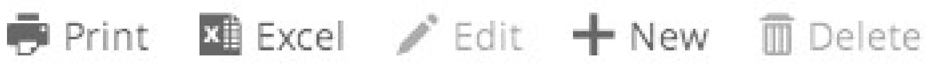
value of the filtered column is between the starting and ending date will be then shown on the table.

7. **Number range filter** – this filte type renders two number input elemetns. There user can specify the minimum and (or) maximum number values for the column. The rows where the number value in this column lies between these two numbers will be displayed in the table.

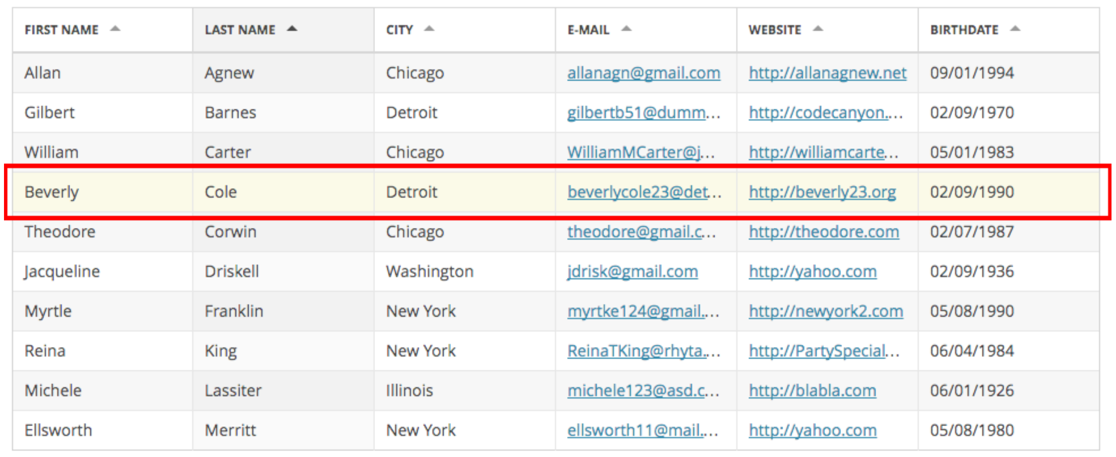
3.4 Frontend editor

This section gives a brief overview of the front-end editor popup.

Front-end editor is available only for MySQL-based tables with front-end editing and server-side processing feature enabled. If you configure the front-end editing for the table you get a different look of the TableTools block: instead of PDF, CSV and Copy buttons you will get a Edit, New and Delete buttons:

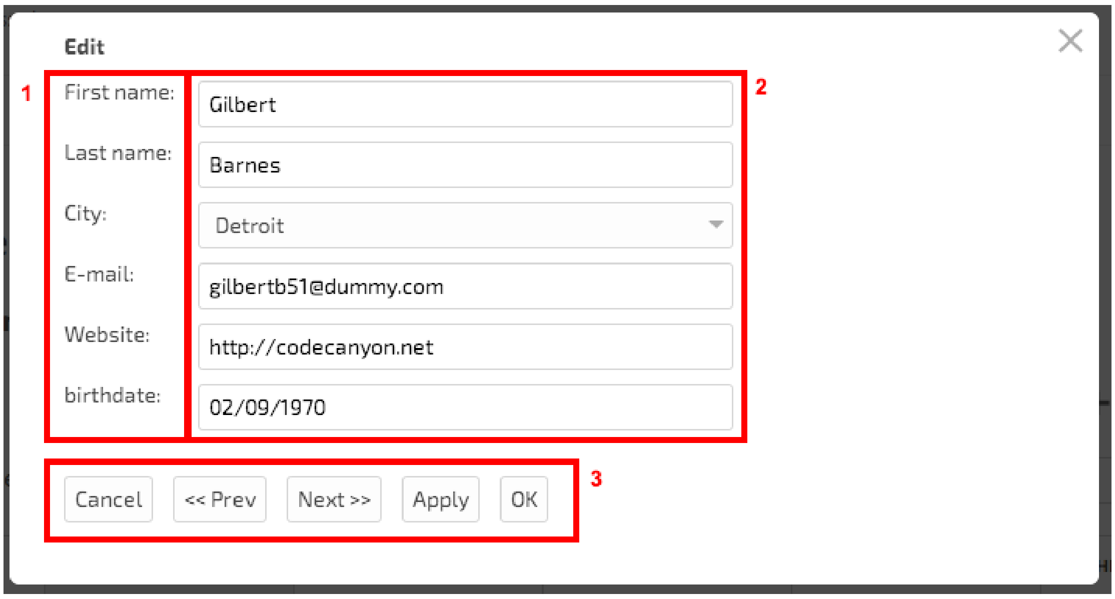


Also it will be possible to select rows in the table by clicking them:



FIRST NAME	LAST NAME	CITY	E-MAIL	WEBSITE	BIRTHDATE
Allan	Agnew	Chicago	allanagn@gmail.com	http://allanagnew.net	09/01/1994
Gilbert	Barnes	Detroit	gilbertb51@dumm...	http://codecanyon....	02/09/1970
William	Carter	Chicago	WilliamMCarter@j...	http://williamcarte...	05/01/1983
Beverly	Cole	Detroit	beverlycole23@det...	http://beverly23.org	02/09/1990
Theodore	Corwin	Chicago	theodore@gmail.c...	http://theodore.com	02/07/1987
Jacqueline	Driskell	Washington	jdrisk@gmail.com	http://yahoo.com	02/09/1936
Myrtle	Franklin	New York	myrtke124@gmail....	http://newyork2.com	05/08/1990
Reina	King	New York	ReinaTKing@rhyta....	http://PartySpecial....	06/04/1984
Michele	Lassiter	Illinois	michele123@asd.c...	http://blabla.com	06/01/1926
Ellsworth	Merritt	New York	ellsworth11@mail....	http://yahoo.com	05/08/1980

When a row is selected, or when you click on the “New” button, you will get the editor popup. The exact look of the popup will depend on the table that you created and configured:



First name:	Gilbert
Last name:	Barnes
City:	Detroit
E-mail:	gilbertb51@dummy.com
Website:	http://codecanyon.net
birthdate:	02/09/1970

1. **Column names** – in the left side you will see the column names. For each column a separate entry will be rendered.
2. **Editor inputs** – To the right of each column name the editor inputs will be rendered. Exact look of each input will depend on the configuration that you define for it. Please see documentation chapter 7 for detailed information and instructions on all possible input types and front-end table editing in general.
3. **Front-end editor navigation and control –** in this area users can see different buttons that allow the to navigate through table entries and control the editing process:
   - **Cancel**: revert all changes and close the editing gialog
   - **Prev, next**: navigate between the rows in the table. Also switches the pages of the table.
   - **Apply**: save the entry, but do not close the dialog.
   - **OK**: save the entry and close the editing dialog.

4. Creating tables from different data sources

This chapter contains detailed walkthrough tutorials for creating tables from different data sources: Excel file, CSV file, MySQL query, JSON, XML and serialized PHP arrays.

4.1 Creating a wpDataTable from Excel

This section gives a walkthrough tutorial on creating and rendering wpDataTables from Excel XLS and XLSX files.

One of the formats that can be used to create a wpDataTable is an Excel file. Please keep in mind that this format is fine only for small tables (up to a couple of hundreds of rows), since bigger ones would cause performance issues both on server and on the client sides.

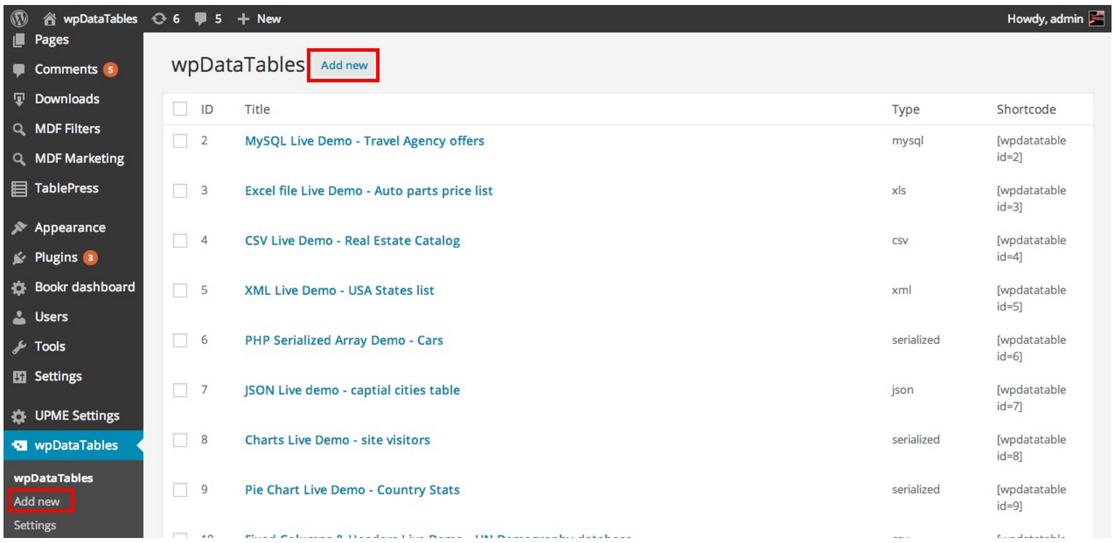
1. **Prepare the data**
   First thing to do for creating a wpDataTable is to prepare the data. In Excel case this means we need to prepare the Excel file. There are several things you need to keep in mind:
   - **First row values will be treated as headers**. So the cell count in the first row will be supposed to be the cell count for all rows.
   - **Merged cells aren't supported**. Neither in header part, or the table itself, vertically or horizontally joined cells won't be read by the plugin correctly.
   - **Images won't be imported**. When you import the Excel file, if you have images pasted in the file, they won't be read correctly by the library, so you would need to first upload images to some hosting, and paste just links to the images in the table. See documentation section 6.6 for more info about using image columns.
   - **Date, time and other specific formats should be formatted as plain text**. Excel uses some specific notation for storing

dates, time, and other specific formats. So please format these cells as “Text” before uploading the file to the server.

You can use almost any Excel file with wpDataTables. The example we will use in our tutorial can be downloaded here.

2. Create a new wpDataTable

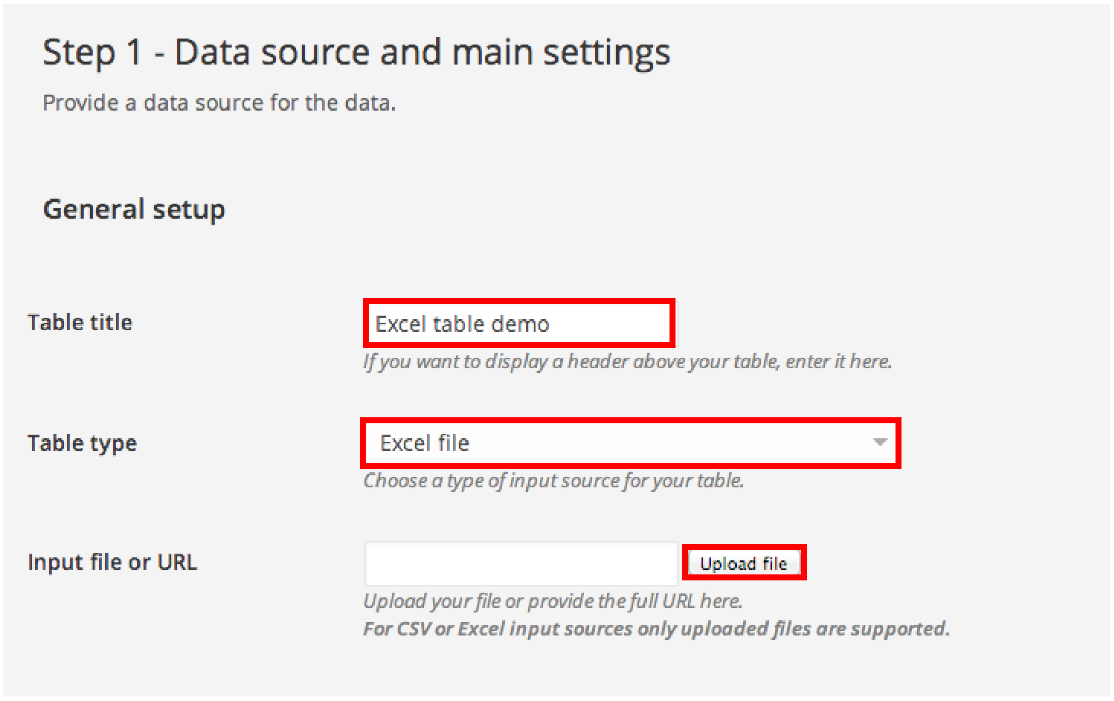
Click either one of the two “Add new” links in wpDataTables plugin’s admin menu.



ID	Title	Type	Shortcode
2	MySQL Live Demo - Travel Agency offers	mysql	[wpdatatable id=2]
3	Excel file Live Demo - Auto parts price list	xls	[wpdatatable id=3]
4	CSV Live Demo - Real Estate Catalog	csv	[wpdatatable id=4]
5	XML Live Demo - USA States list	xml	[wpdatatable id=5]
6	PHP Serialized Array Demo - Cars	serialized	[wpdatatable id=6]
7	JSON Live demo - captial cities table	json	[wpdatatable id=7]
8	Charts Live Demo - site visitors	serialized	[wpdatatable id=8]
9	Pie Chart Live Demo - Country Stats	serialized	[wpdatatable id=9]

3. Set the table title and provide the data source

Define the table title if you would like to see it on the page, set the table type as “Excel file”, and then click on “Upload file”:


Step 1 - Data source and main settings

Provide a data source for the data.

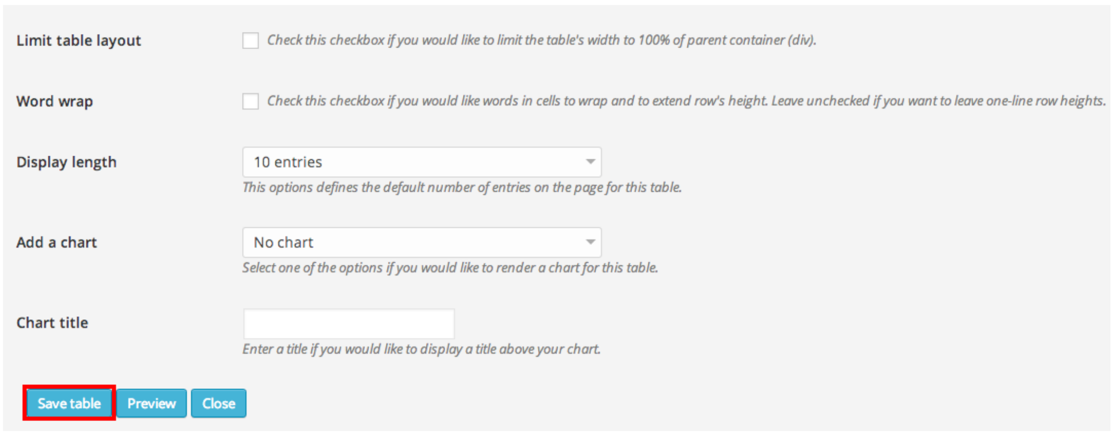
General setup

Table title	Excel table demo *If you want to display a header above your table, enter it here.*
Table type	Excel file *Choose a type of input source for your table.*
Input file or URL	Upload file *Upload your file or provide the full URL here.* ***For CSV or Excel input sources only uploaded files are supported.***

In the popup that appears you can drag&drop the file that you prepared, click on “Select Files” and find this file on your local computer, or find it in the WordPress Media Library if you have already uploaded it previously. Once you have chosen the file you need to make sure that the “File URL” option is selected (otherwise wpDataTables plugin won’t see the Excel file), so that the direct file URL will be shown in the “Link URL” input. Once you’ve done that you can click the “Insert into Post” button.

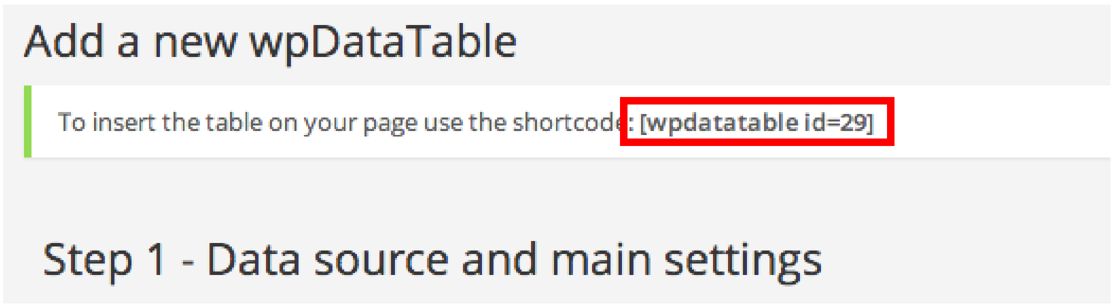
4. Set the additional table data and save the table

When the data source is provided you can define the additional table properties, such as enabling or disabling the advanced filter, making the table responsive, and so on (refer to the documentation sections 2.2, and chapter 5 to read about all these settings in detail). You can also define these properties later. Once you’re done you can click the “Save table”.


Limit table layout	☐ Check this checkbox if you would like to limit the table's width to 100% of parent container (div).
Word wrap	☐ Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave unchecked if you want to leave one-line row heights.
Display length	10 entries This options defines the default number of entries on the page for this table.
Add a chart	No chart Select one of the options if you would like to render a chart for this table.
Chart title	Enter a title if you would like to display a title above your chart.

5. **Copy the table shortcode and paste on the post or page.**

Once the table is saved you will get a success notification. Then the table shortcode will appear in the upper part of the table edit page:

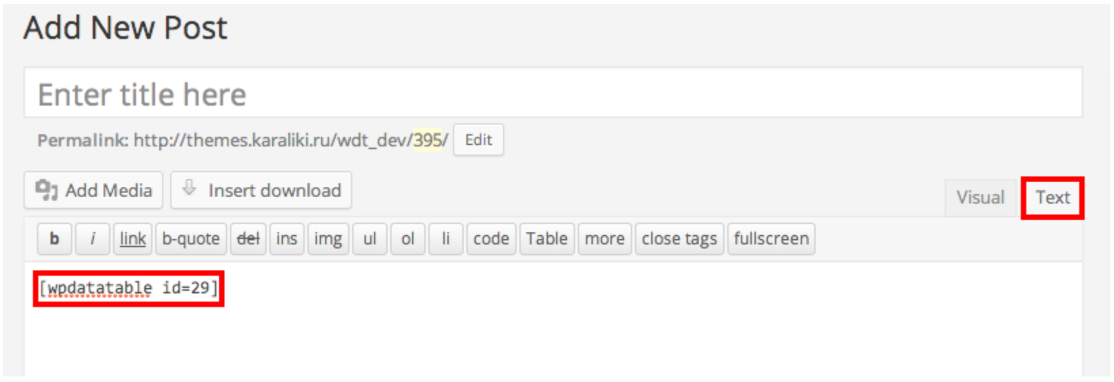

Add a new wpDataTable

To insert the table on your page use the shortcode: [wpdatatable id=29]

Step 1 - Data source and main settings

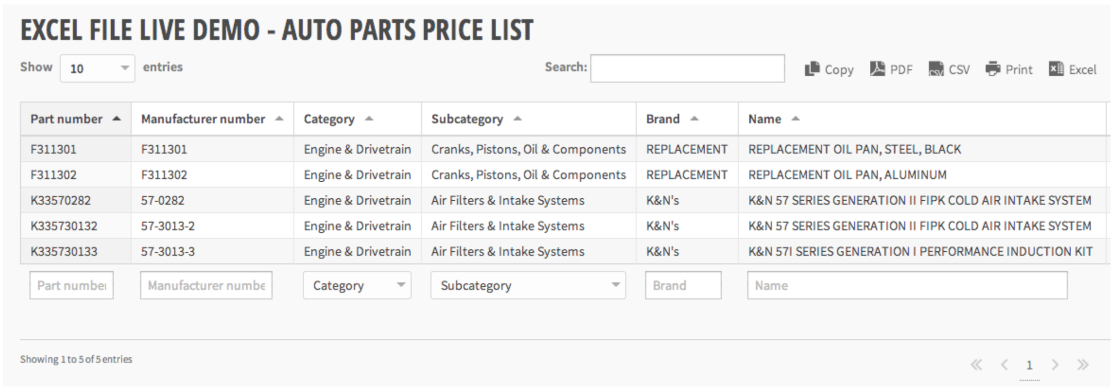
You can copy this shortcode and go to the post or page where you need the table.

There in the editor you can paste the shortcode. Please use the "Text" editor mode for this, since then you will be sure that no invalid markup will be pasted along with the shortcode:



```
[wpdatatable id=29]
```

Once you've done that you can publish the page and open it in the site front-end.


EXCEL FILE LIVE DEMO - AUTO PARTS PRICE LIST

Show 10 entries   Search: Copy PDF CSV Print Excel

Part number	Manufacturer number	Category	Subcategory	Brand	Name
F311301	F311301	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, STEEL, BLACK
F311302	F311302	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, ALUMINUM
K33570282	57-0282	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM
K335730132	57-3013-2	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM
K335730133	57-3013-3	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57I SERIES GENERATION I PERFORMANCE INDUCTION KIT
Part number	Manufacturer number	Category	Subcategory	Brand	Name

Showing 1 to 5 of 5 entries   « ‹ 1 › »

If you get some errors during this process please refer to the troubleshooting section (chapter 10) of this documentation, FAQ page on the site. If you don't find a solution there you can refer to our support system.

> **Notice for developers:** main logic for this feature is in the file: source/class.wpdatatable.php, method buildByExcel().

4.2 Creating a wpDataTable from CSV file

This section gives a walkthrough tutorial on creating and rendering wpDataTables from CSV (comma-separated-values) files.

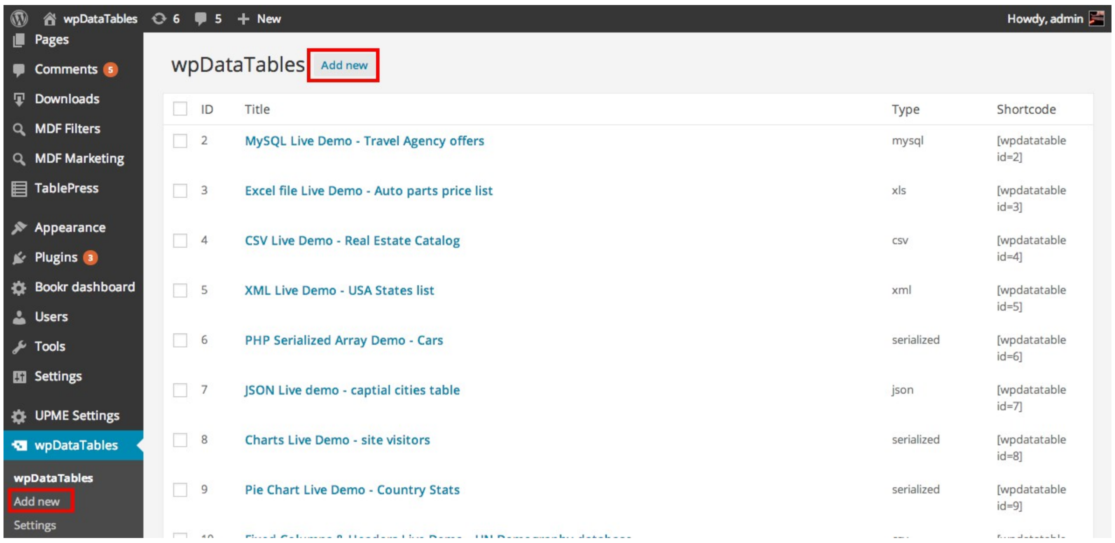
CSV is a common format, supported by most of the software that works with tables. Generally, it's just a text file, where each line represents a table row, and cell values are separated by commas or semicolumns (that's why the format is called CSV, comma-separated-values). It works faster the Excel file, since it's much smaller, so whenever possible I would recommend using it instead of XLS/XLSX (but MySQL is always better).

1. **Prepare the data**
   First thing to do is to prepare the CSV file for the wpDataTable plugin. There're not much limitations that you should keep in mind:
   - **First row values will be treated as headers.** Same as for Excel files, first row is the "main" row, which should represent the headers, and the column count. If other rows in the table will have less or more cells then the first one, this may cause trouble.
   - **Do not use non-standard delimiters.** This shouldn't be an issue in most of the cases. Some programs put non-standard separators in CSV file, which causes it to break, so please watch that you use comma or semicolumn for separating the cells, and line break for separating column rows.

You can use basically any CSV file for the table, then one that is used in the tutorial can be found here.

2. **Create a new wpDataTable**

Click either one of the two "Add new" links in wpDataTables plugin's admin menu.

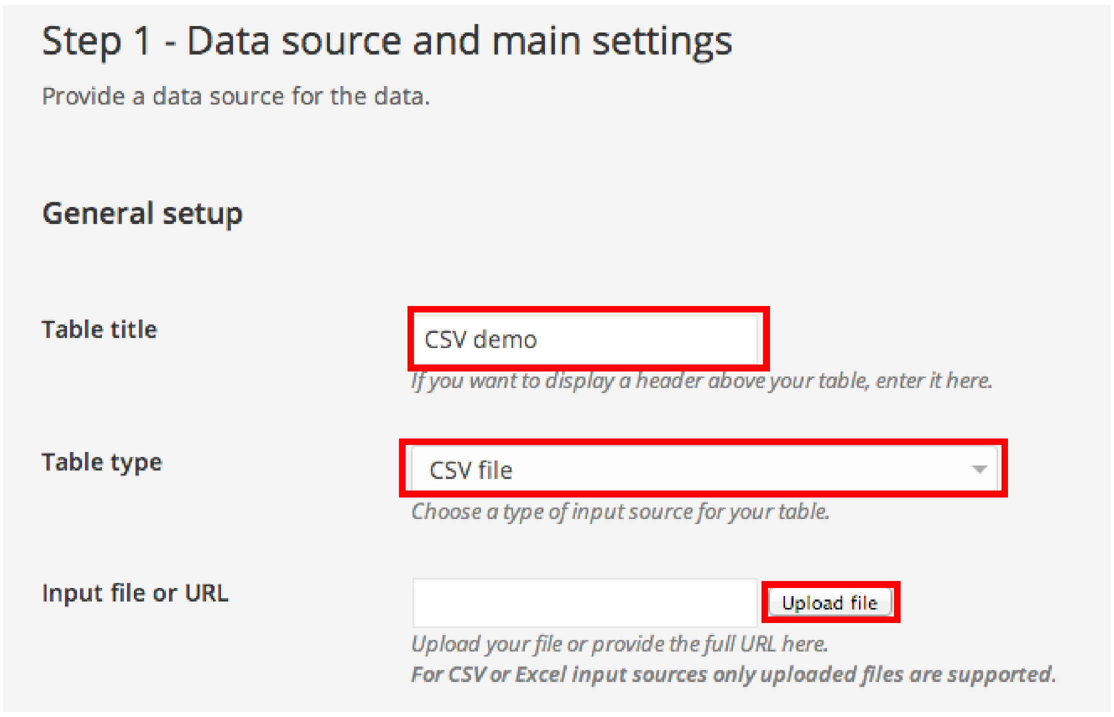


wpDataTables Add new

ID	Title	Type	Shortcode
2	MySQL Live Demo - Travel Agency offers	mysql	[wpdatatable id=2]
3	Excel file Live Demo - Auto parts price list	xls	[wpdatatable id=3]
4	CSV Live Demo - Real Estate Catalog	csv	[wpdatatable id=4]
5	XML Live Demo - USA States list	xml	[wpdatatable id=5]
6	PHP Serialized Array Demo - Cars	serialized	[wpdatatable id=6]
7	JSON Live demo - captial cities table	json	[wpdatatable id=7]
8	Charts Live Demo - site visitors	serialized	[wpdatatable id=8]
9	Pie Chart Live Demo - Country Stats	serialized	[wpdatatable id=9]

3. Set the table title and provide the data source

Define the table title if you would like to see it on the page, set the table type as "CSV file", and then click on "Upload file":



Step 1 - Data source and main settings

Provide a data source for the data.

General setup

Table title: CSV demo

If you want to display a header above your table, enter it here.

Table type: CSV file

Choose a type of input source for your table.

Input file or URL: Upload file

Upload your file or provide the full URL here.

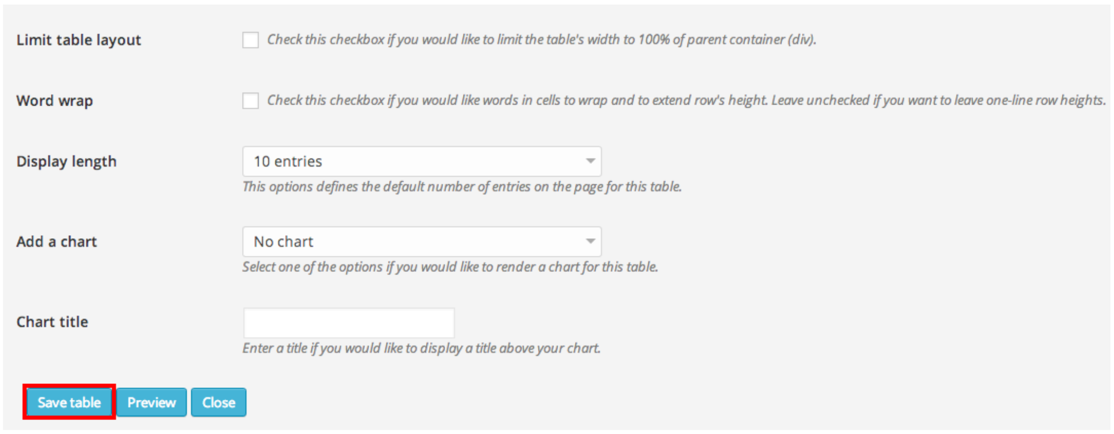
For CSV or Excel input sources only uploaded files are supported.

In the popup that appears you can drag&drop the file that you prepared, click on "Select Files" and find this file on your local computer, or find it in the WordPress Media Library if you have already uploaded it previously.

Once you have chosen the file you need to make sure that the "File URL" option is selected (otherwise wpDataTables plugin won't see the CSV file), so that the direct file URL will be shown in the "Link URL" input. Once you've done that you can click the "Insert into Post" button.

4. Set the additional table data and save the table

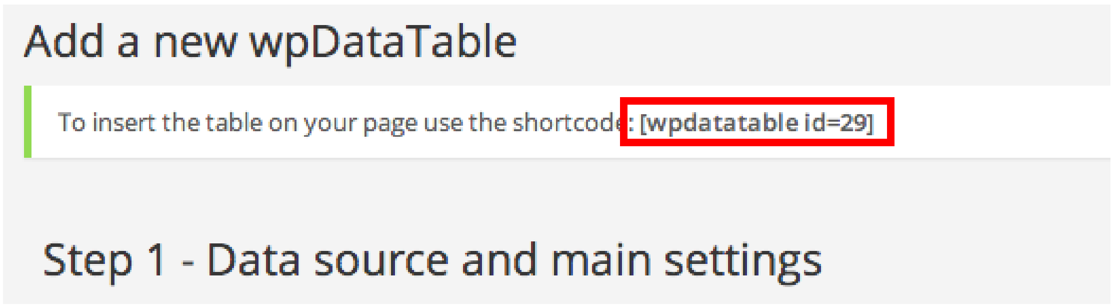
When the data source is provided you can define the additional table properties, such as enabling or disabling the advanced filter, making the table responsive, and so on (refer to the documentation sections 2.2, and chapter 5 to read about all these settings in detail). You can also define these properties later. Once you're done you can click the "Save table".



Limit table layout	☐ Check this checkbox if you would like to limit the table's width to 100% of parent container (div).
Word wrap	☐ Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave unchecked if you want to leave one-line row heights.
Display length	10 entries This options defines the default number of entries on the page for this table.
Add a chart	No chart Select one of the options if you would like to render a chart for this table.
Chart title	Enter a title if you would like to display a title above your chart.

5. Copy the table shortcode and paste on the post or page.

Once the table is saved you will get a success notification. Then the table shortcode will appear in the upper part of the table edit page:



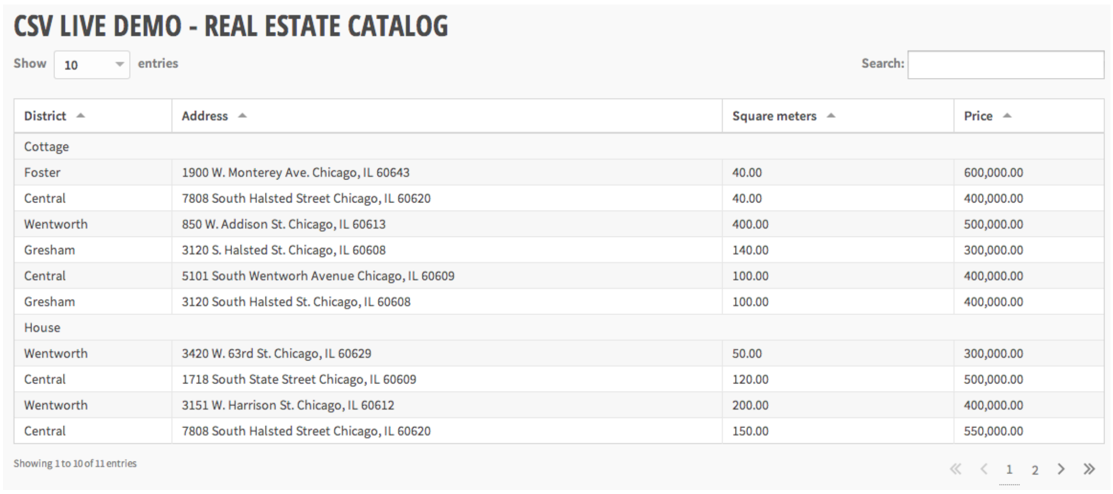
Add a new wpDataTable

To insert the table on your page use the shortcode: [wpdatatable id=29]

Step 1 - Data source and main settings

You can copy this shortcode and go to the post or page where you need the table.

There in the editor you can paste the shortcode. Please use the “Text” editor mode for this, since then you will be sure that no invalid markup will be pasted along with the shortcode:



CSV LIVE DEMO - REAL ESTATE CATALOG

Show 10 entries Search:

District	Address	Square meters	Price
Cottage			
Foster	1900 W. Monterey Ave. Chicago, IL 60643	40.00	600,000.00
Central	7808 South Halsted Street Chicago, IL 60620	40.00	400,000.00
Wentworth	850 W. Addison St. Chicago, IL 60613	400.00	500,000.00
Gresham	3120 S. Halsted St. Chicago, IL 60608	140.00	300,000.00
Central	5101 South Wentworh Avenue Chicago, IL 60609	100.00	400,000.00
Gresham	3120 South Halsted St. Chicago, IL 60608	100.00	400,000.00
House			
Wentworth	3420 W. 63rd St. Chicago, IL 60629	50.00	300,000.00
Central	1718 South State Street Chicago, IL 60609	120.00	500,000.00
Wentworth	3151 W. Harrison St. Chicago, IL 60612	200.00	400,000.00
Central	7808 South Halsted Street Chicago, IL 60620	150.00	550,000.00

Showing 1 to 10 of 11 entries

« ‹ 1 2 › »

If you get some errors during this process please refer to the troubleshooting section (chapter 10) of this documentation, FAQ page on the site. If you don’t find a solution there you can refer to our support system.

> **Notice for developers:** main logic for this feature is in the file: source/class.wpdatatable.php, method buildByExcel().

4.3 Creating a MySQL-based wpDataTable

This section gives a walkthrough tutorial on creating and rendering wpDataTables from MySQL table using a SQL query.

MySQL is probably the most commonly used database engine in the web. It is fast, it handles the data storing and querying on a convenient way, and

WordPress itself works based on MySQL, so it makes sense to also store your table data in MySQL engine.

The MySQL server and database for your tables can be the same as your WordPress installation is using, or a remote one (*make sure that it accepts remote connections in that case*). Please refer to documentation section 1.5 for details on how to set up a connection to a remote MySQL server.

1. **Prepare the data**
   Same as for other input types, first thing for MySQL-based tables to do is to prepare the table data for the wpDataTable plugin. This basically includes the following:
   - **Prepare the table structure and create the table on MySQL side.** You can use the same DB as WordPress is using, or a separate MySQL connection (see documentation section 1.5 for details on how to set up a connection to a remote MySQL server). You can use e.g. PHPMyAdmin software to easily create the table structure.
   - **Add at least several rows of data to the table on MySQL side.** wpDataTables cannot render empty tables (at least for now) so MySQL table should contain at least several rows of data so that it could be rendered on wpDataTables side.
   - **Prepare and test the MySQL query.** Before you actually use the MySQL query in wpDataTables it's better to test it first in some software, like PHPMyAdmin, to make sure that it does work and does return the data that you would like to see in the table. If you don't want to see all columns from a table, but just some of them, it's better to query only the ones that you need, instead of querying all and then hiding the unneeded ones – it is better for performance. E.g. "**SELECT name, year, surname FROM table**" instead of "**SELECT * FROM table**", if

“**table**” has many columns, and you only need to show these three.

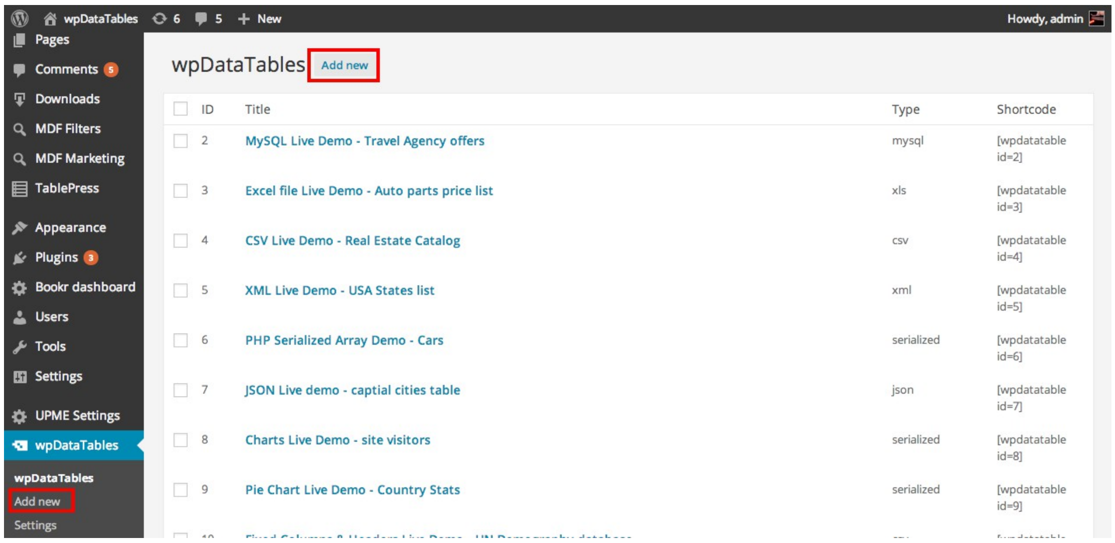
Please also keep in mind these limitations (we are working on avoiding them, but now please keep them in mind):

- **Please do not use spaces in the column names.** E.g. use ‘**column1**’ instead of ‘**column 1**’.
- **Please do not put a semicolon in the end of the query.**
- **Please do not use column names that can coincide with reserved MySQL words** (like DATE, BY, ORDER, GROUP, …).
- **Please do not use numeric column names** like 1, 22, etc.
- **Please do not use “ORDER BY” in the statement.** wpDataTables has its own sorting engine so it makes no sense to use MySQL’s sorting, since it will be overridden. Also server-side processing feature adds this part of statement automatically when users triggers the sorting on the front-end, and having it in initial statement may cause the table to crash.

You can use any MySQL query for your table. MySQL dump for the table that will be used in this example can be downloaded here.

2. Create a new wpDataTable

Click either one of the two “Add new” links in wpDataTables plugin’s admin menu.

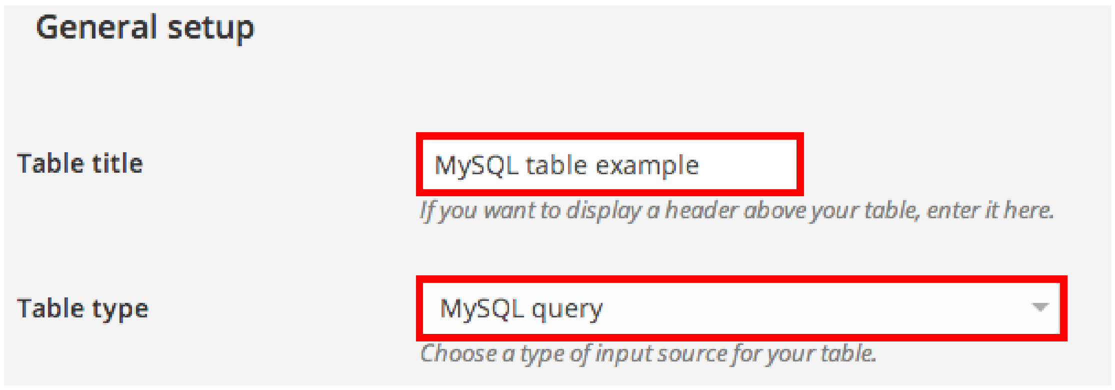


wpDataTables Add new

ID	Title	Type	Shortcode
2	MySQL Live Demo - Travel Agency offers	mysql	[wpdatatable id=2]
3	Excel file Live Demo - Auto parts price list	xls	[wpdatatable id=3]
4	CSV Live Demo - Real Estate Catalog	csv	[wpdatatable id=4]
5	XML Live Demo - USA States list	xml	[wpdatatable id=5]
6	PHP Serialized Array Demo - Cars	serialized	[wpdatatable id=6]
7	JSON Live demo - captial cities table	json	[wpdatatable id=7]
8	Charts Live Demo - site visitors	serialized	[wpdatatable id=8]
9	Pie Chart Live Demo - Country Stats	serialized	[wpdatatable id=9]

3. Set the table title and provide the data source

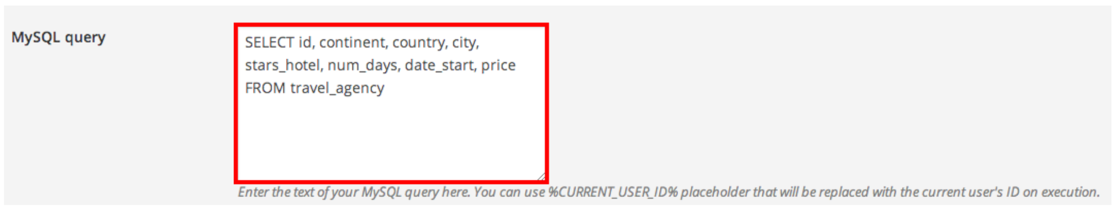
Define the table title if you would like to see it on the page, set the table type as “MySQL”:



General setup

Table title	MySQL table example *If you want to display a header above your table, enter it here.*
Table type	MySQL query *Choose a type of input source for your table.*

Then you can paste the MySQL query that will return the table data in the MySQL query input:



MySQL query

```
SELECT id, continent, country, city,
stars_hotel, num_days, date_start, price
FROM travel_agency
```

Enter the text of your MySQL query here. You can use %CURRENT_USER_ID% placeholder that will be replaced with the current user's ID on execution.

Note: if you would like to use current user's ID in the query (e.g. to show records from some table only for the current user) you can use the placeholder **%CURRENT_USER_ID%** - it will be replaced with the actual ID value of the currently logged in user at the runtime moment. So e.g. this:

SELECT * FROM yourtable WHERE user_id = %CURRENT_USER_ID%

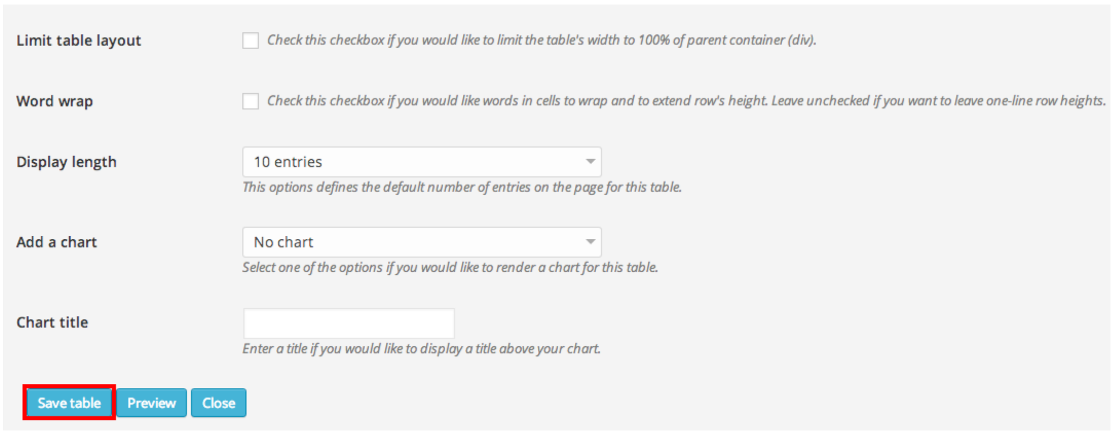
will be sent to MySQL as

SELECT * FROM yourtable WHERE user_id = 1

(if current user has ID = 1).

4. Set the additional table data and save the table

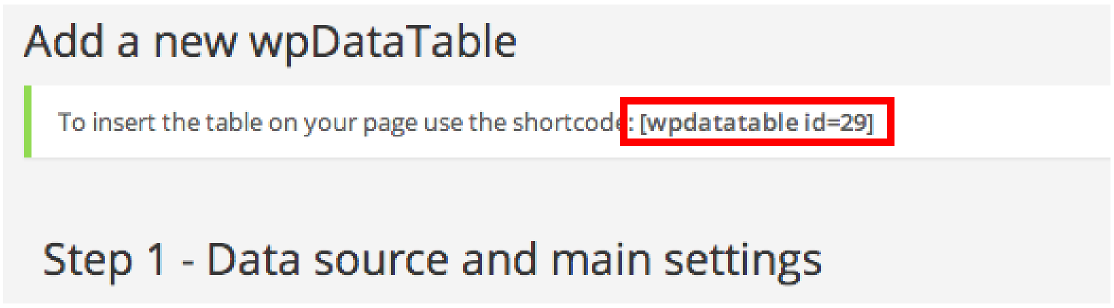
When the data source is provided you can define the additional table properties, such as enabling or disabling the advanced filter, making the table responsive, and so on (refer to the documentation sections 2.2, and chapter 5 to read about all these settings in detail). You can also define these properties later. Once you're done you can click the "Save table".



Limit table layout	☐ *Check this checkbox if you would like to limit the table's width to 100% of parent container (div).*
Word wrap	☐ *Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave unchecked if you want to leave one-line row heights.*
Display length	10 entries *This options defines the default number of entries on the page for this table.*
Add a chart	No chart *Select one of the options if you would like to render a chart for this table.*
Chart title	*Enter a title if you would like to display a title above your chart.*

5. **Copy the table shortcode and paste on the post or page.**

Once the table is saved you will get a success notification. Then the table shortcode will appear in the upper part of the table edit page:

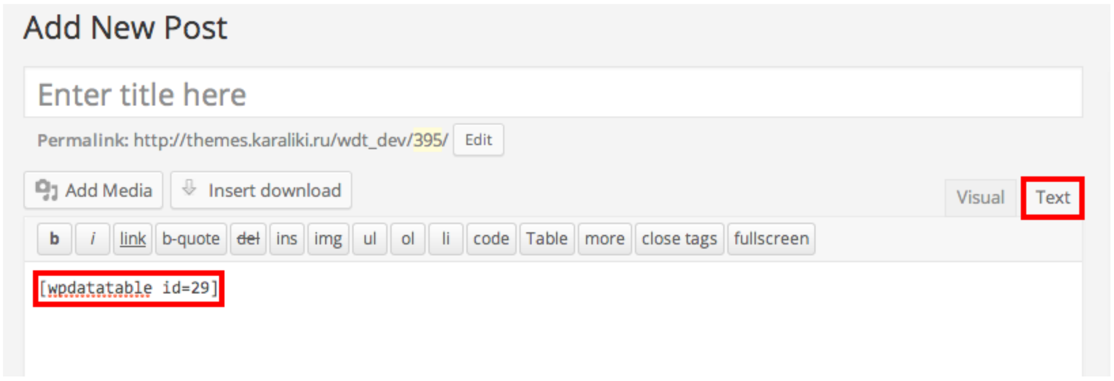

Add a new wpDataTable

To insert the table on your page use the shortcode: [wpdatatable id=29]

Step 1 - Data source and main settings

You can copy this shortcode and go to the post or page where you need the table.

There in the editor you can paste the shortcode. Please use the "Text" editor mode for this, since then you will be sure that no invalid markup will be pasted along with the shortcode:

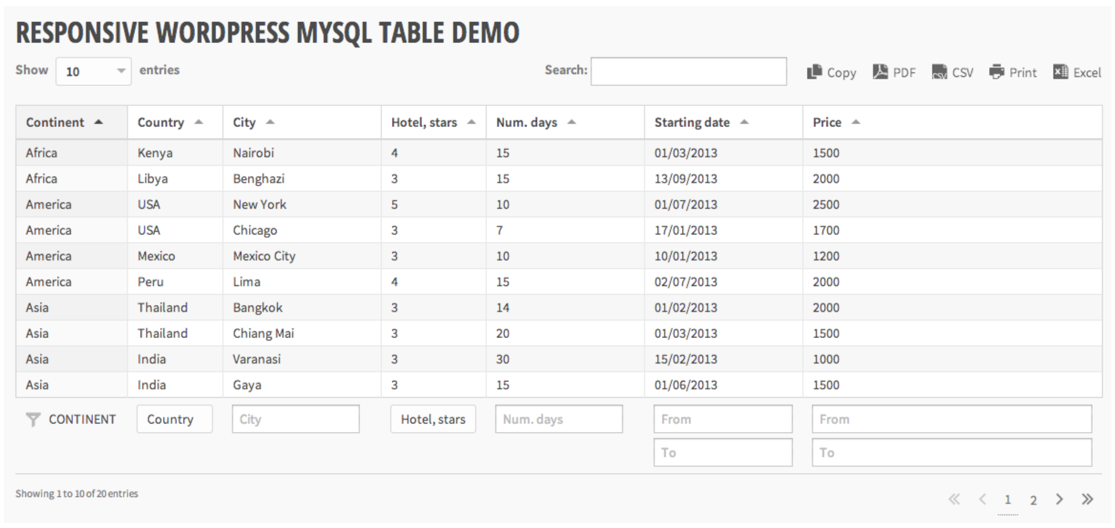

Add New Post

Enter title here

Permalink: http://themes.karaliki.ru/wdt_dev/395/

```
[wpdatatable id=29]
```

Once you've done that you can publish the page and open it in the site front-end.



RESPONSIVE WORDPRESS MYSQL TABLE DEMO

Show 10 entries Search: Copy PDF CSV Print Excel

Continent	Country	City	Hotel, stars	Num. days	Starting date	Price
Africa	Kenya	Nairobi	4	15	01/03/2013	1500
Africa	Libya	Benghazi	3	15	13/09/2013	2000
America	USA	New York	5	10	01/07/2013	2500
America	USA	Chicago	3	7	17/01/2013	1700
America	Mexico	Mexico City	3	10	10/01/2013	1200
America	Peru	Lima	4	15	02/07/2013	2000
Asia	Thailand	Bangkok	3	14	01/02/2013	2000
Asia	Thailand	Chiang Mai	3	20	01/03/2013	1500
Asia	India	Varanasi	3	30	15/02/2013	1000
Asia	India	Gaya	3	15	01/06/2013	1500
CONTINENT	Country	City	Hotel, stars	Num. days	From / To	From / To

Showing 1 to 10 of 20 entries

« ‹ 1 2 › »

If you get some errors during this process please refer to the troubleshooting section (chapter 10) of this documentation, FAQ page on the site. If you don't find a solution there you can refer to our support system.

> **Notice for developers:** main logic for this feature is in the file: source/class.wpdatatable.php, method queryBasedConstruct().

4.4 Creating MySQL-based wpDataTables with server-side processing.

This section gives a brief overview on creating wpDataTables with server-side processing feature enabled.

Server-side processing feature delegates all the sorting, pagination, filtering and other data processing routines to MySQL engine, and only several rows are fetched from MySQL server via AJAX-request each time you open a page, sort by some column, filter by some column, etc. Please read full details about the server-side processing feature in documentation section 5.1.

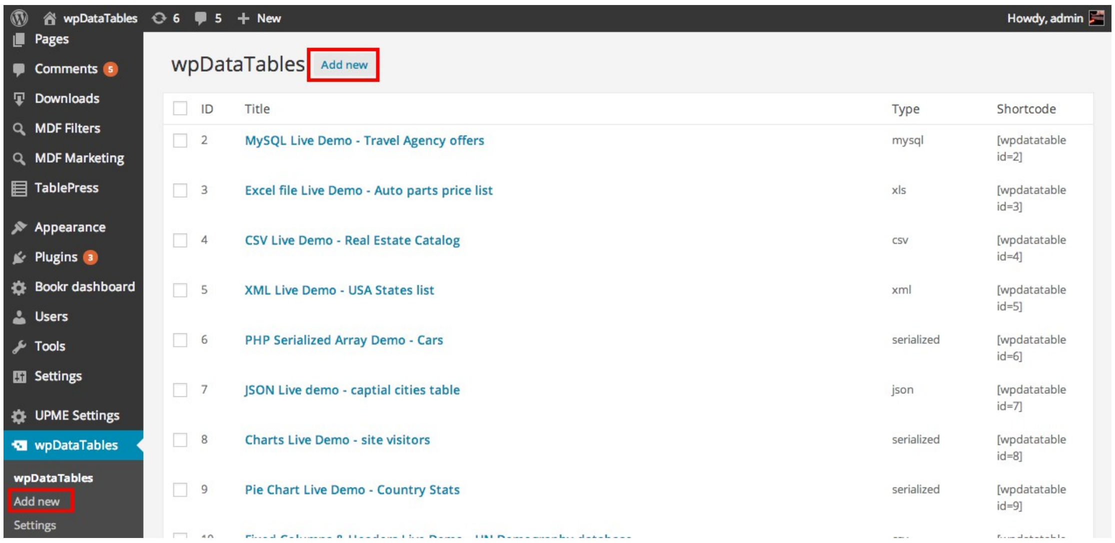
1. **Prepare the data**

Everything that was mentioned in section 4.3 of the documentation (for MySQL tables without server-side processing feature) is the same. Please read this section for details on how to prepare MySQL data for wpDataTables.

You can use any MySQL query for your table. MySQL dump for the table that will be used in this example can be downloaded here.

2. Create a new wpDataTable

Click either one of the two "Add new" links in wpDataTables plugin's admin menu.



	ID	Title	Type	Shortcode
☐	2	MySQL Live Demo - Travel Agency offers	mysql	[wpdatatable id=2]
☐	3	Excel file Live Demo - Auto parts price list	xls	[wpdatatable id=3]
☐	4	CSV Live Demo - Real Estate Catalog	csv	[wpdatatable id=4]
☐	5	XML Live Demo - USA States list	xml	[wpdatatable id=5]
☐	6	PHP Serialized Array Demo - Cars	serialized	[wpdatatable id=6]
☐	7	JSON Live demo - captial cities table	json	[wpdatatable id=7]
☐	8	Charts Live Demo - site visitors	serialized	[wpdatatable id=8]
☐	9	Pie Chart Live Demo - Country Stats	serialized	[wpdatatable id=9]

3. Set the table title and provide the data source

Define the table title if you would like to see it on the page, set the table type as "MySQL":

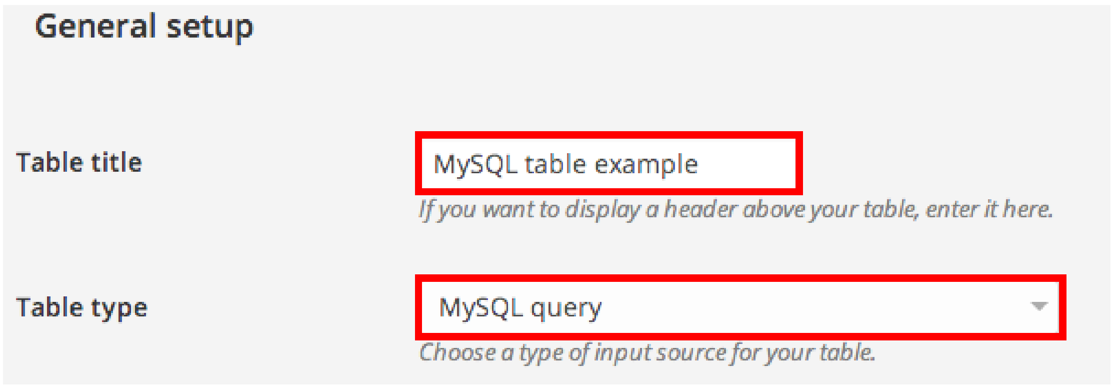
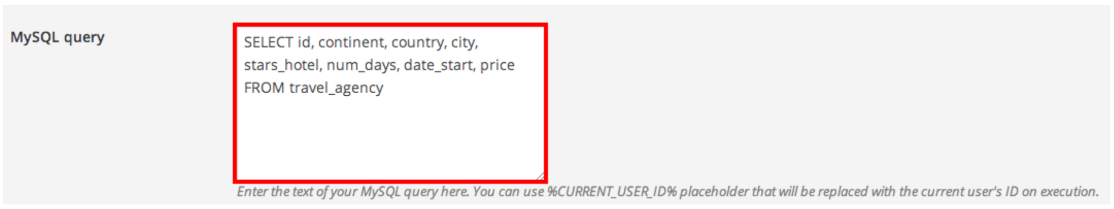

General setup

Table title	MySQL table example *If you want to display a header above your table, enter it here.*
Table type	MySQL query *Choose a type of input source for your table.*

Then you can paste the MySQL query that will return the table data in the MySQL query input:


MySQL query

```
SELECT id, continent, country, city,
stars_hotel, num_days, date_start, price
FROM travel_agency
```

Enter the text of your MySQL query here. You can use %CURRENT_USER_ID% placeholder that will be replaced with the current user's ID on execution.

Note: if you would like to use current user's ID in the query (e.g. to show records from some table only for the current user) you can use the placeholder **%CURRENT_USER_ID%** - it will be replaced with the actual ID value of the currently logged in user at the runtime moment. So e.g. this:

SELECT * FROM yourtable WHERE user_id = %CURRENT_USER_ID%

will be sent to MySQL as

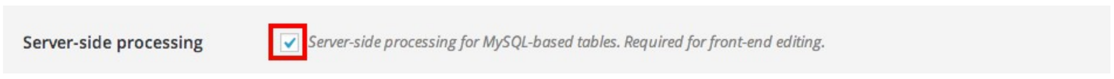
SELECT * FROM yourtable WHERE user_id = 1

(if current user has ID = 1).

4. Set the additional table data and save the table

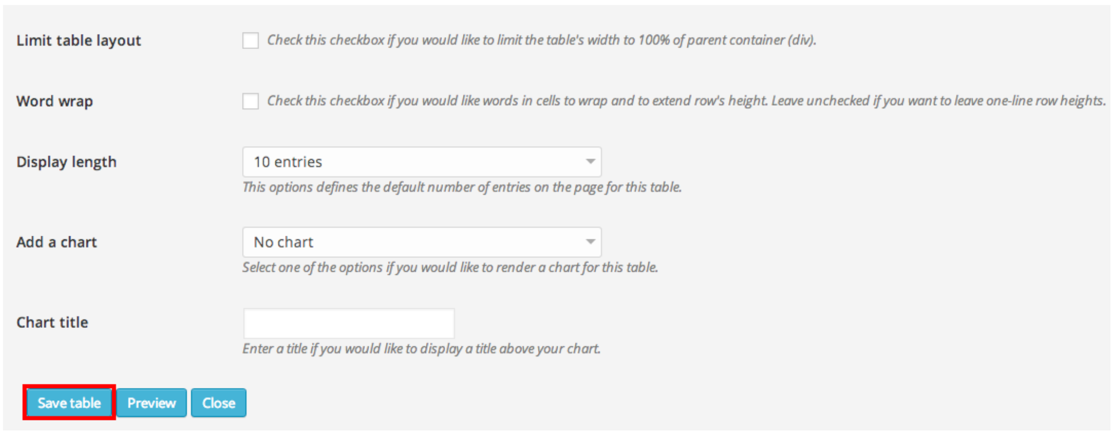
When the data source is provided you can define the additional table properties, such as enabling or disabling the advanced filter, making the table responsive, and so on (refer to the documentation sections 2.2, and chapter 5 to read about all these settings in detail). You can also define these properties later.

The main key part for tables with server-side processing is to tick the checkbox "Server-side processing" for this table which will actually tell the plugin to use the feature.



Server-side processing ✓ *Server-side processing for MySQL-based tables. Required for front-end editing.*

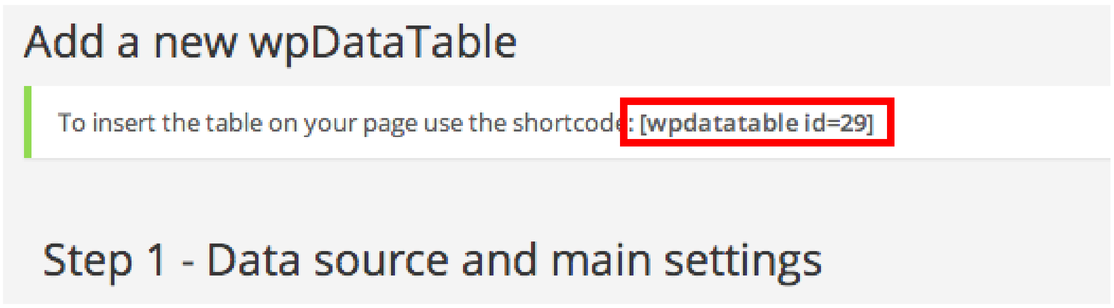
Once you're done you can click the "Save table".



Limit table layout	☐ *Check this checkbox if you would like to limit the table's width to 100% of parent container (div).*
Word wrap	☐ *Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave unchecked if you want to leave one-line row heights.*
Display length	10 entries *This options defines the default number of entries on the page for this table.*
Add a chart	No chart *Select one of the options if you would like to render a chart for this table.*
Chart title	 *Enter a title if you would like to display a title above your chart.*

5. Copy the table shortcode and paste on the post or page.

Once the table is saved you will get a success notification. Then the table shortcode will appear in the upper part of the table edit page:

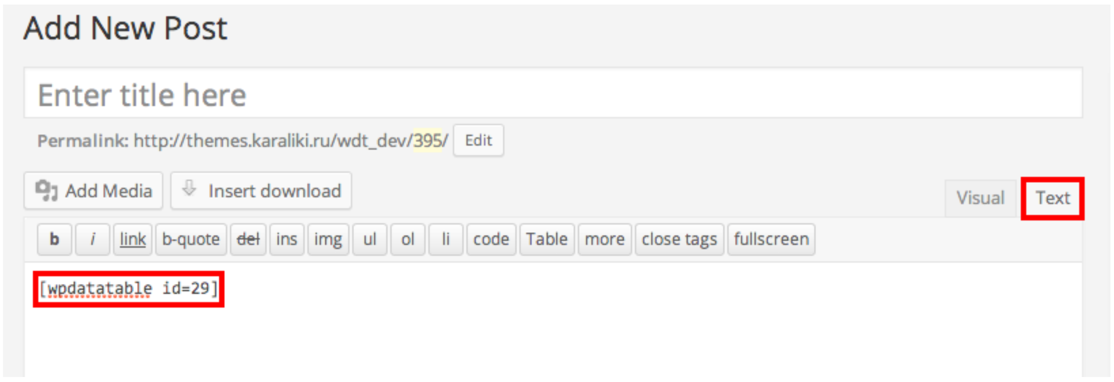

Add a new wpDataTable

To insert the table on your page use the shortcode: [wpdatatable id=29]

Step 1 - Data source and main settings

You can copy this shortcode and go to the post or page where you need the table.

There in the editor you can paste the shortcode. Please use the “Text” editor mode for this, since then you will be sure that no invalid markup will be pasted along with the shortcode:

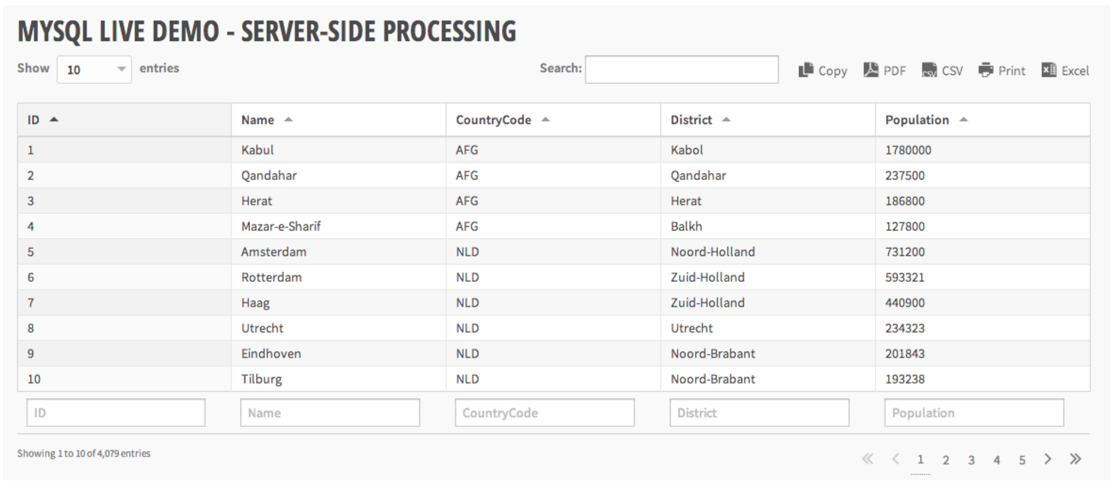

Add New Post

Enter title here

Permalink: http://themes.karaliki.ru/wdt_dev/395/

```
[wpdatatable id=29]
```

Once you’ve done that you can publish the page and open it in the site front-end. The only difference from “usual” wpDataTable is that every time you switch the page of the table, or sort, or filter by some value, an AJAX request to the server will be sent, and preloader overlay will be rendered on the table while waiting for response.



MYSQL LIVE DEMO - SERVER-SIDE PROCESSING

Show 10 entries

Search:

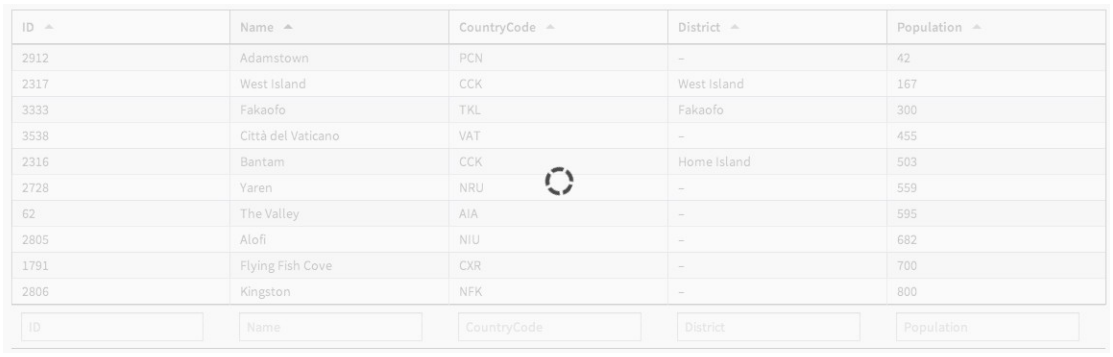
Copy PDF CSV Print Excel

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabol	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e-Sharif	AFG	Balkh	127800
5	Amsterdam	NLD	Noord-Holland	731200
6	Rotterdam	NLD	Zuid-Holland	593321
7	Haag	NLD	Zuid-Holland	440900
8	Utrecht	NLD	Utrecht	234323
9	Eindhoven	NLD	Noord-Brabant	201843
10	Tilburg	NLD	Noord-Brabant	193238
ID	Name	CountryCode	District	Population

Showing 1 to 10 of 4,079 entries

« ‹ 1 2 3 4 5 › »

Preloader:



ID	Name	CountryCode	District	Population
2912	Adamstown	PCN	–	42
2317	West Island	CCK	West Island	167
3333	Fakaofo	TKL	Fakaofo	300
3538	Città del Vaticano	VAT	–	455
2316	Bantam	CCK	Home Island	503
2728	Yaren	NRU	–	559
62	The Valley	AIA	–	595
2805	Alofi	NIU	–	682
1791	Flying Fish Cove	CXR	–	700
2806	Kingston	NFK	–	800
ID	Name	CountryCode	District	Population

If you get some errors during this process please refer to the troubleshooting section (chapter 10) of this documentation, FAQ page on the site. If you don't find a solution there you can refer to our support system.

> **Notice for developers:** main logic for this feature is in the file: source/class.wpdatatable.php, method queryBasedConstruct(); the AJAX action ('wdt_get_ajax_data') is in the file controllers/wdt_ajax_actions.php,

4.5 Creating a table from JSON input

This section gives a walkthrough tutorial on creating and rendering wpDataTables from JSON data source.

JSON (JavaScript Object Notation) is a convenient format that a lot of web services support. If your system can return data in JSON format you can easily use wpDataTables plugin as a realtime visualisation tool for JSON tables in WordPress.

JSON input file is reformatted and passed to wpDataTables render engine.

1. **Prepare the data.**

Since wpDataTables is working with table data, you would need to make sure that you return the JSON data in a specific format. Basically, it should be an array of objects, where each object should have the same number and structure of properties. For example:

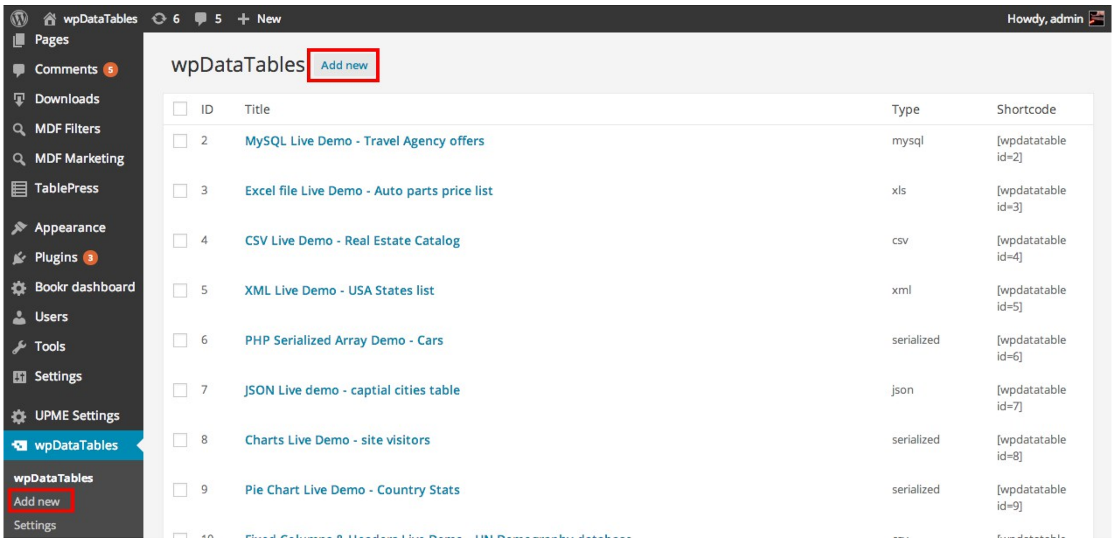
```
[
        {
                ‘name’: ‘Peter’,
                ‘age’: 24
        },
        {
                ‘name’: ‘John’,
                ‘age’: 37
        },
        …
]
```

And so on.

You can use any data that you want in your JSON, we will use this file in the example.

2. Create a new wpDataTable

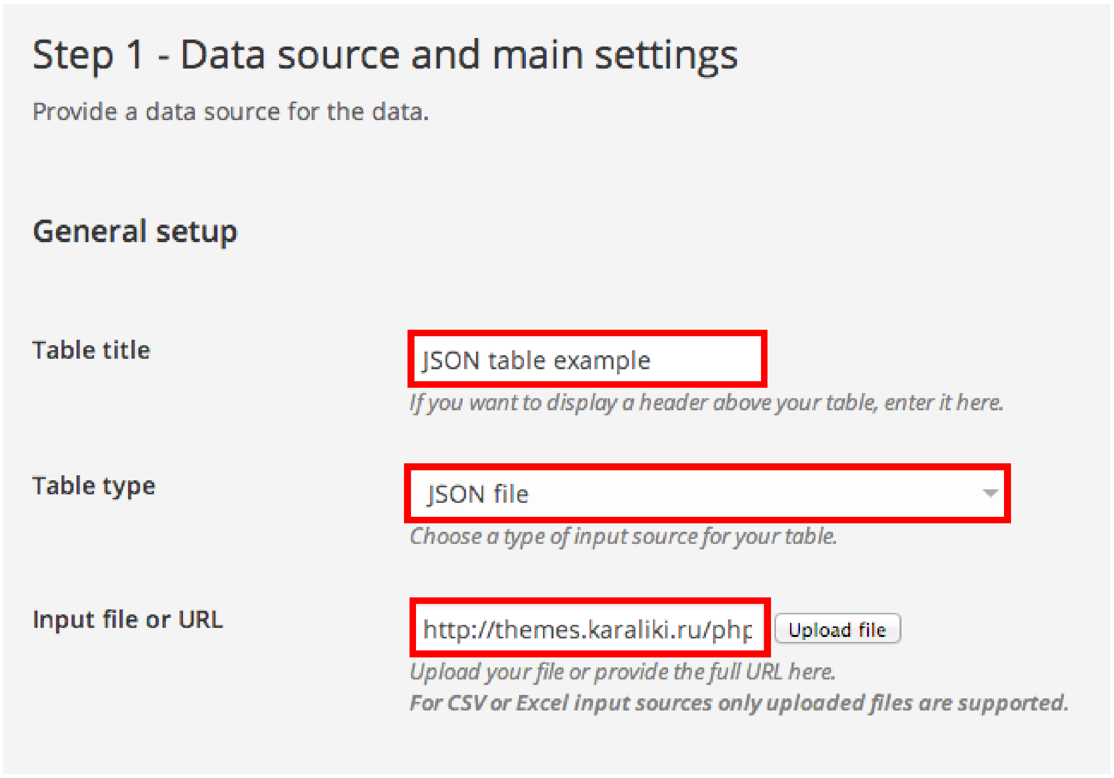
Click either one of the two “Add new” links in wpDataTables plugin’s admin menu.



ID	Title	Type	Shortcode
2	MySQL Live Demo - Travel Agency offers	mysql	[wpdatatable id=2]
3	Excel file Live Demo - Auto parts price list	xls	[wpdatatable id=3]
4	CSV Live Demo - Real Estate Catalog	csv	[wpdatatable id=4]
5	XML Live Demo - USA States list	xml	[wpdatatable id=5]
6	PHP Serialized Array Demo - Cars	serialized	[wpdatatable id=6]
7	JSON Live demo - captial cities table	json	[wpdatatable id=7]
8	Charts Live Demo - site visitors	serialized	[wpdatatable id=8]
9	Pie Chart Live Demo - Country Stats	serialized	[wpdatatable id=9]

3. Set the table title and provide the data source

Define the table title if you would like to see it on the page, set the table type as “JSON file”, then paste the URL of your file to the “Input file or URL” input:


Step 1 - Data source and main settings

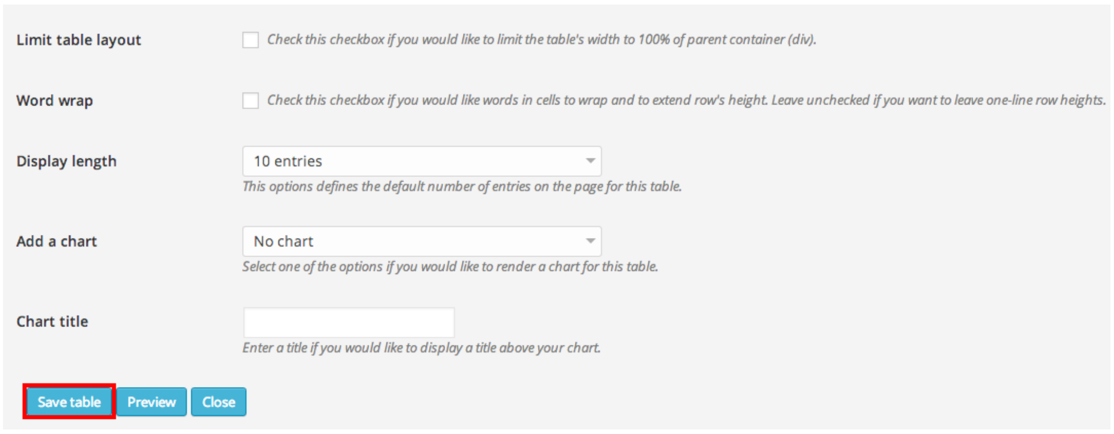
Provide a data source for the data.

General setup

Table title	JSON table example *If you want to display a header above your table, enter it here.*
Table type	JSON file *Choose a type of input source for your table.*
Input file or URL	http://themes.karaliki.ru/php *Upload your file or provide the full URL here.* ***For CSV or Excel input sources only uploaded files are supported.***

4. Set the additional table data and save the table

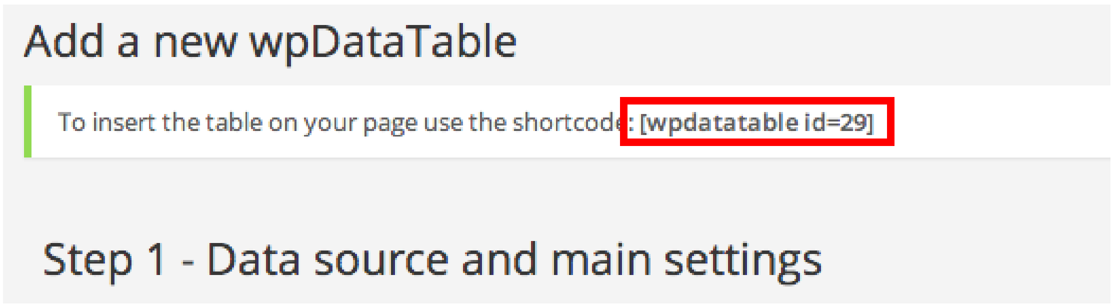
When the data source is provided you can define the additional table properties, such as enabling or disabling the advanced filter, making the table responsive, and so on (refer to the documentation sections 2.2, and chapter 5 to read about all these settings in detail). You can also define these properties later. Once you're done you can click the "Save table".



Limit table layout	☐ *Check this checkbox if you would like to limit the table's width to 100% of parent container (div).*
Word wrap	☐ *Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave unchecked if you want to leave one-line row heights.*
Display length	10 entries *This options defines the default number of entries on the page for this table.*
Add a chart	No chart *Select one of the options if you would like to render a chart for this table.*
Chart title	*Enter a title if you would like to display a title above your chart.*

5. Copy the table shortcode and paste on the post or page.

Once the table is saved you will get a success notification. Then the table shortcode will appear in the upper part of the table edit page:

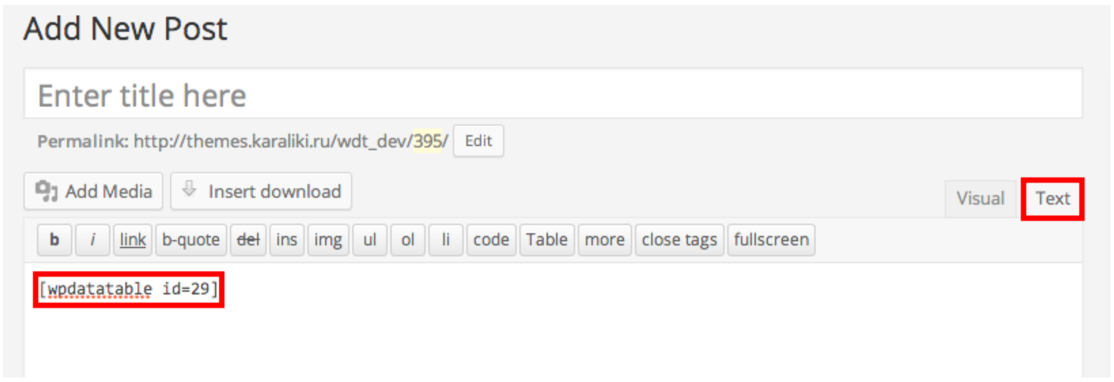

Add a new wpDataTable

To insert the table on your page use the shortcode: [wpdatatable id=29]

Step 1 - Data source and main settings

You can copy this shortcode and go to the post or page where you need the table.

There in the editor you can paste the shortcode. Please use the "Text" editor mode for this, since then you will be sure that no invalid markup will be pasted along with the shortcode:

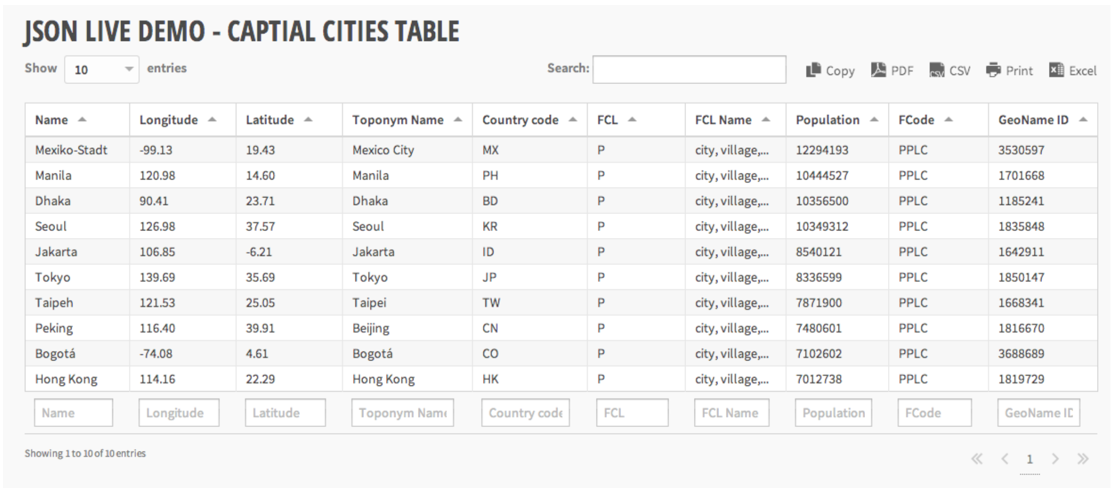

Add New Post

Enter title here

Permalink: http://themes.karaliki.ru/wdt_dev/395/ Edit

```
[wpdatatable id=29]
```

Once you've done that you can publish the page and open it in the site front-end.


JSON LIVE DEMO - CAPTIAL CITIES TABLE

Name	Longitude	Latitude	Toponym Name	Country code	FCL	FCL Name	Population	FCode	GeoName ID
Mexiko-Stadt	-99.13	19.43	Mexico City	MX	P	city, village,...	12294193	PPLC	3530597
Manila	120.98	14.60	Manila	PH	P	city, village,...	10444527	PPLC	1701668
Dhaka	90.41	23.71	Dhaka	BD	P	city, village,...	10356500	PPLC	1185241
Seoul	126.98	37.57	Seoul	KR	P	city, village,...	10349312	PPLC	1835848
Jakarta	106.85	-6.21	Jakarta	ID	P	city, village,...	8540121	PPLC	1642911
Tokyo	139.69	35.69	Tokyo	JP	P	city, village,...	8336599	PPLC	1850147
Taipeh	121.53	25.05	Taipei	TW	P	city, village,...	7871900	PPLC	1668341
Peking	116.40	39.91	Beijing	CN	P	city, village,...	7480601	PPLC	1816670
Bogotá	-74.08	4.61	Bogotá	CO	P	city, village,...	7102602	PPLC	3688689
Hong Kong	114.16	22.29	Hong Kong	HK	P	city, village,...	7012738	PPLC	1819729

Showing 1 to 10 of 10 entries

If you get some errors during this process please refer to the troubleshooting section (chapter 10) of this documentation, FAQ page on the site. If you don't find a solution there you can refer to our support system.

> **Notice for developers:** main logic for this feature is in the file:
> source/class.wpdatatable.php, method jsonBasedConstruct(),

4.6 Creating a table from XML input

This section gives a walkthrough tutorial on creating and rendering wpDataTables from an XML data source.

XML (eXtensible Markup Language) is for many years a standard of cross-platform and cross-system formats of data exchange.

If you need to print table data from XML source (e.g. if your system can output data as an XML web service) you can use wpDataTables plugin based to publish XML tables in WordPress. XML input file is reformatted and passed to wpDataTables render engine.

1. **Prepare the data.**

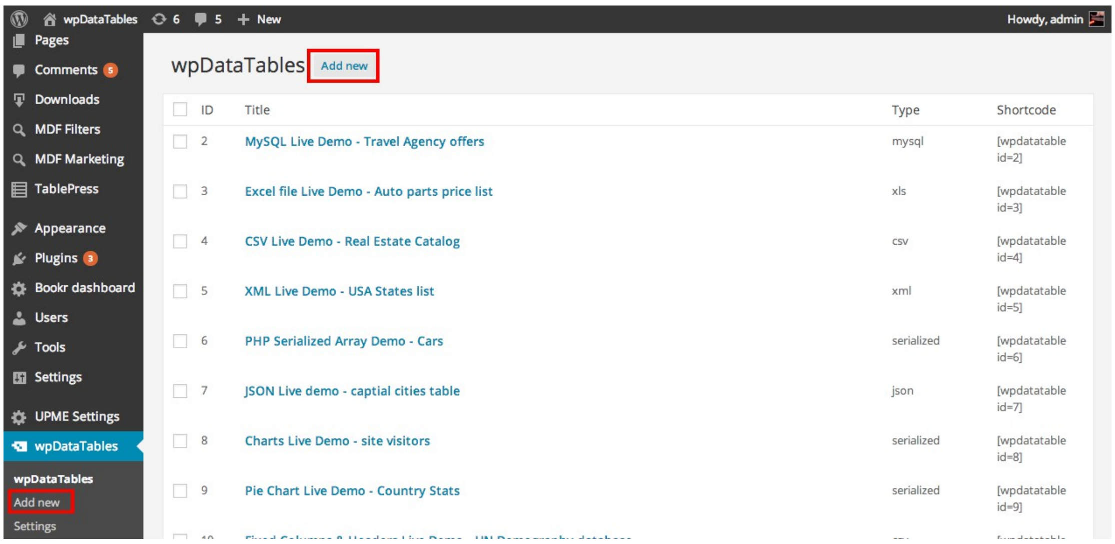
Since wpDataTables is working with table data, you would need to make sure that you return the XML data in a specific format. Basically, the XML file should describe array of elements, where each element should have the same number and structure of properties. For example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<people>
        <person name=”Peter” age=”24” />
        <person name=”John” age=”27” />
</people>
```

So, you need to keep in mind, that multi-level XML files will not be parsed correctly.
You can use any data that you need in your XML file, if you would like to see our example file you can download it here.

2. Create a new wpDataTable

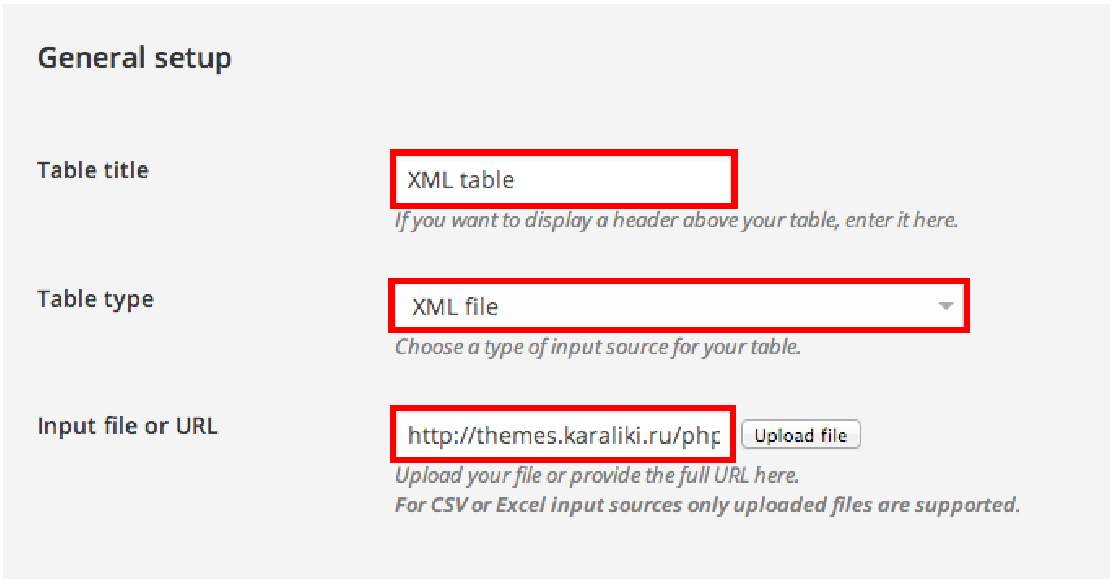
Click either one of the two “Add new” links in wpDataTables plugin’s admin menu.



ID	Title	Type	Shortcode
2	MySQL Live Demo - Travel Agency offers	mysql	[wpdatatable id=2]
3	Excel file Live Demo - Auto parts price list	xls	[wpdatatable id=3]
4	CSV Live Demo - Real Estate Catalog	csv	[wpdatatable id=4]
5	XML Live Demo - USA States list	xml	[wpdatatable id=5]
6	PHP Serialized Array Demo - Cars	serialized	[wpdatatable id=6]
7	JSON Live demo - captial cities table	json	[wpdatatable id=7]
8	Charts Live Demo - site visitors	serialized	[wpdatatable id=8]
9	Pie Chart Live Demo - Country Stats	serialized	[wpdatatable id=9]

3. Set the table title and provide the data source

Define the table title if you would like to see it on the page, set the table type as “XML file”, then paste the URL of your file to the “Input file or URL” input:

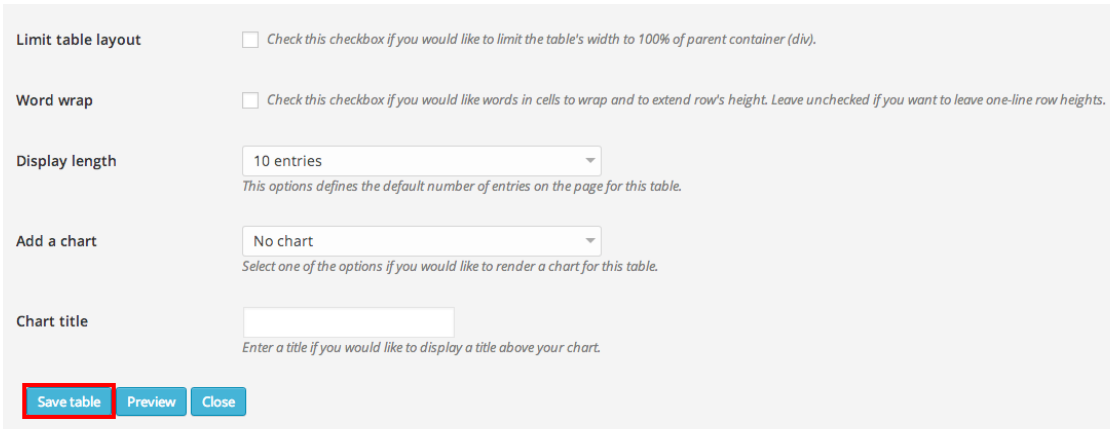


General setup

Table title	XML table *If you want to display a header above your table, enter it here.*
Table type	XML file *Choose a type of input source for your table.*
Input file or URL	http://themes.karaliki.ru/php *Upload your file or provide the full URL here.* ***For CSV or Excel input sources only uploaded files are supported.***

4. Set the additional table data and save the table

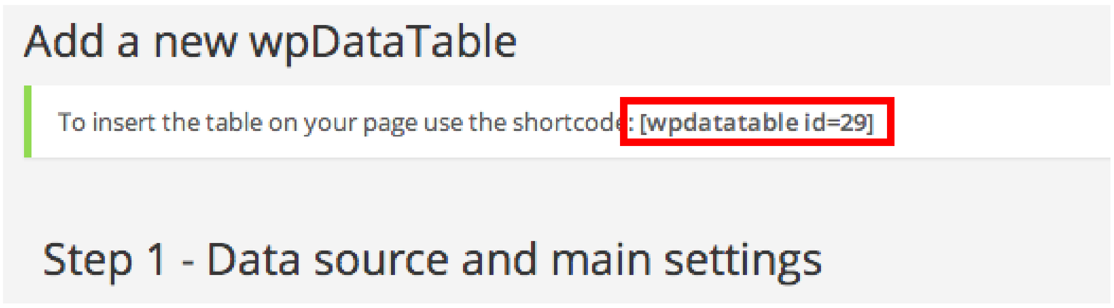
When the data source is provided you can define the additional table properties, such as enabling or disabling the advanced filter, making the table responsive, and so on (refer to the documentation sections 2.2, and chapter 5 to read about all these settings in detail). You can also define these properties later. Once you’re done you can click the “Save table”.



Limit table layout	☐ *Check this checkbox if you would like to limit the table's width to 100% of parent container (div).*
Word wrap	☐ *Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave unchecked if you want to leave one-line row heights.*
Display length	10 entries *This options defines the default number of entries on the page for this table.*
Add a chart	No chart *Select one of the options if you would like to render a chart for this table.*
Chart title	*Enter a title if you would like to display a title above your chart.*

5. **Copy the table shortcode and paste on the post or page.**

Once the table is saved you will get a success notification. Then the table shortcode will appear in the upper part of the table edit page:



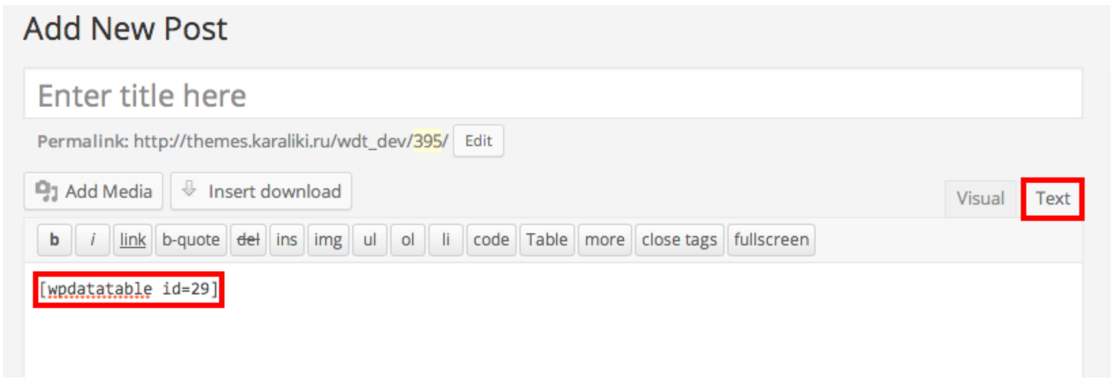
Add a new wpDataTable

To insert the table on your page use the shortcode: [wpdatatable id=29]

Step 1 - Data source and main settings

You can copy this shortcode and go to the post or page where you need the table.

There in the editor you can paste the shortcode. Please use the "Text" editor mode for this, since then you will be sure that no invalid markup will be pasted along with the shortcode:



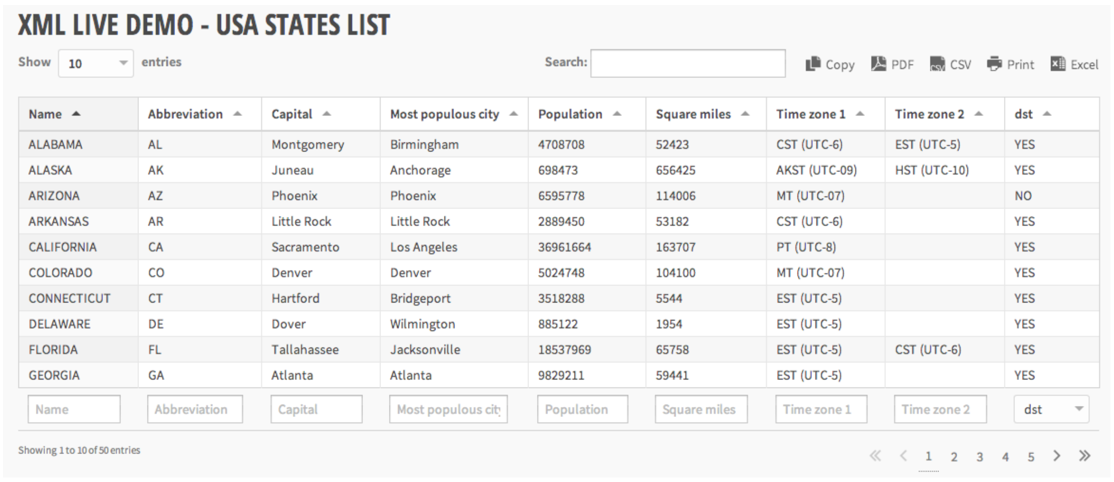
Add New Post

Enter title here

Permalink: http://themes.karaliki.ru/wdt_dev/395/ Edit

```
[wpdatatable id=29]
```

Once you've done that you can publish the page and open it in the site front-end.



XML LIVE DEMO - USA STATES LIST

Name	Abbreviation	Capital	Most populous city	Population	Square miles	Time zone 1	Time zone 2	dst
ALABAMA	AL	Montgomery	Birmingham	4708708	52423	CST (UTC-6)	EST (UTC-5)	YES
ALASKA	AK	Juneau	Anchorage	698473	656425	AKST (UTC-09)	HST (UTC-10)	YES
ARIZONA	AZ	Phoenix	Phoenix	6595778	114006	MT (UTC-07)		NO
ARKANSAS	AR	Little Rock	Little Rock	2889450	53182	CST (UTC-6)		YES
CALIFORNIA	CA	Sacramento	Los Angeles	36961664	163707	PT (UTC-8)		YES
COLORADO	CO	Denver	Denver	5024748	104100	MT (UTC-07)		YES
CONNECTICUT	CT	Hartford	Bridgeport	3518288	5544	EST (UTC-5)		YES
DELAWARE	DE	Dover	Wilmington	885122	1954	EST (UTC-5)		YES
FLORIDA	FL	Tallahassee	Jacksonville	18537969	65758	EST (UTC-5)	CST (UTC-6)	YES
GEORGIA	GA	Atlanta	Atlanta	9829211	59441	EST (UTC-5)		YES
Name	Abbreviation	Capital	Most populous cit	Population	Square miles	Time zone 1	Time zone 2	dst

Showing 1 to 10 of 50 entries

If you get some errors during this process please refer to the troubleshooting section (chapter 10) of this documentation, FAQ page on the site. If you don't find a solution there you can refer to our support system.

> **Notice for developers:** main logic for this feature is in the file:
> source/class.wpdatatable.php, method XMLBasedConstruct(),

4.7 Creating a table from serialized PHP array

This section gives a walkthrough tutorial on creating and rendering wpDataTables from a serialized PHP array.

Sometimes you cannot get the table from some data source directly, e.g. because XML web service returns data for many cities, and you would like to show the data only for your city; or because you need to do some calculations that MySQL can't do. In this case you can use PHP as an "adapter" for your data.

You may use this approach if you want to prepare the data for tables from some external sources manually: e.g. if you receive the data from several queries, and then restructure it in PHP. This way you can always have actual data in the table on your website, since data itself isn't stored in the database, and always is received from the input PHP file.

1. **Prepare the data**

   First you would need to prepare the data on PHP side in an array, then echo the serialized version of it, something like:

   ```
   <?php
           $yourArray = methodThatReturnsData();
           echo serialize($yourArray);
   ?>
   ```

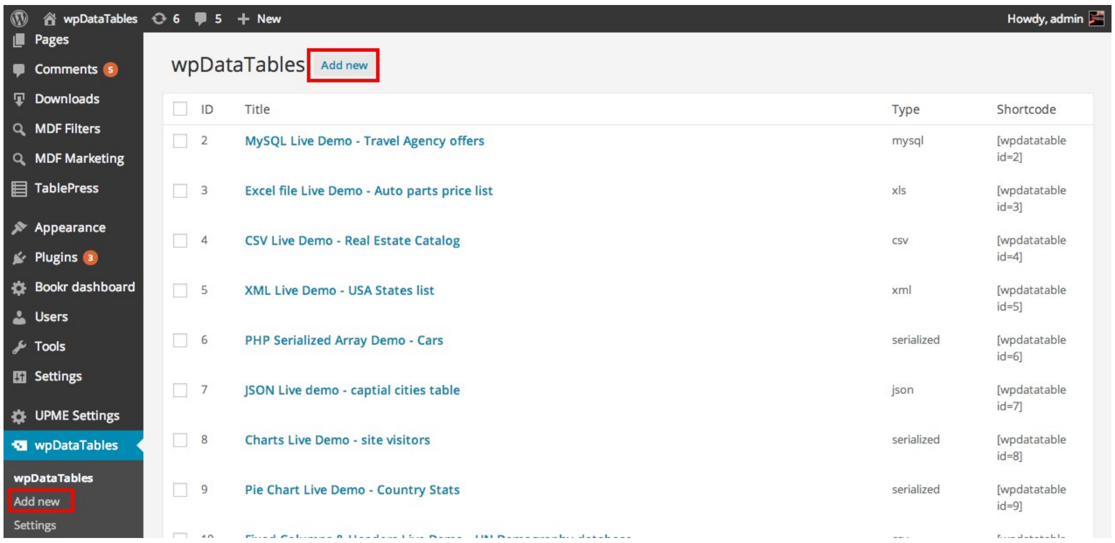
   Please keep in mind that array should have 2 dimensions, the top level should be number-indexed, and represent rows; the second level should have string keys, that would represent column headers, and the values would represent cell values.

   All the array entries of the second level should have the same structure (array value count and key names), otherwise you will get an exception.

   You can use any PHP file from any host that echoes out a serialized array, we will use the file from this link.

2. **Create a new wpDataTable**

Click either one of the two "Add new" links in wpDataTables plugin's admin menu.

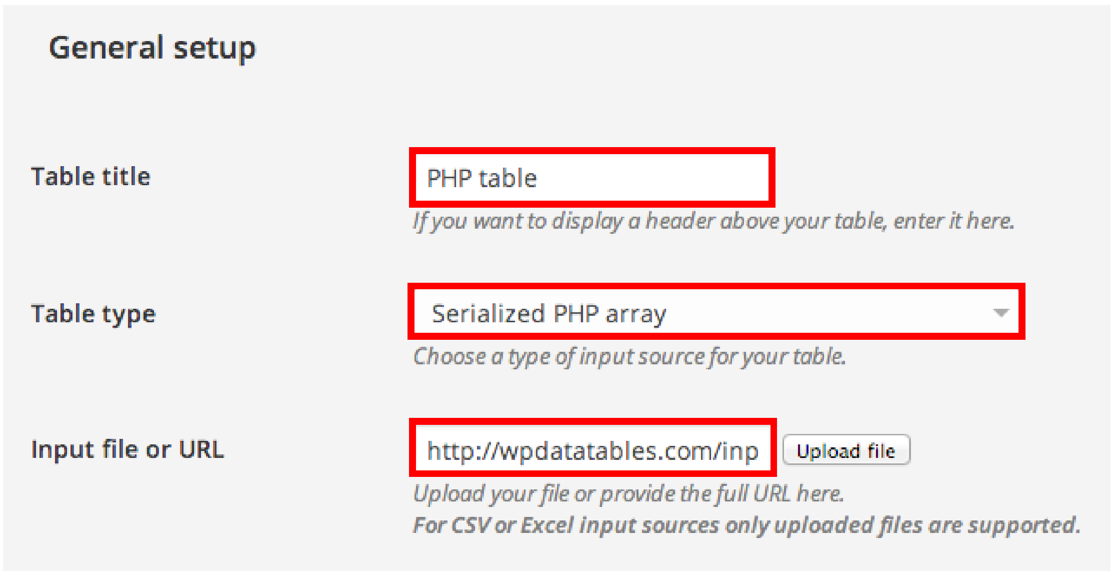


wpDataTables Add new

ID	Title	Type	Shortcode
2	MySQL Live Demo - Travel Agency offers	mysql	[wpdatatable id=2]
3	Excel file Live Demo - Auto parts price list	xls	[wpdatatable id=3]
4	CSV Live Demo - Real Estate Catalog	csv	[wpdatatable id=4]
5	XML Live Demo - USA States list	xml	[wpdatatable id=5]
6	PHP Serialized Array Demo - Cars	serialized	[wpdatatable id=6]
7	JSON Live demo - captial cities table	json	[wpdatatable id=7]
8	Charts Live Demo - site visitors	serialized	[wpdatatable id=8]
9	Pie Chart Live Demo - Country Stats	serialized	[wpdatatable id=9]

3. Set the table title and provide the data source

Define the table title if you would like to see it on the page, set the table type as "Serialized PHP array", then paste the URL of the PHP file to the "Input file or URL" input:

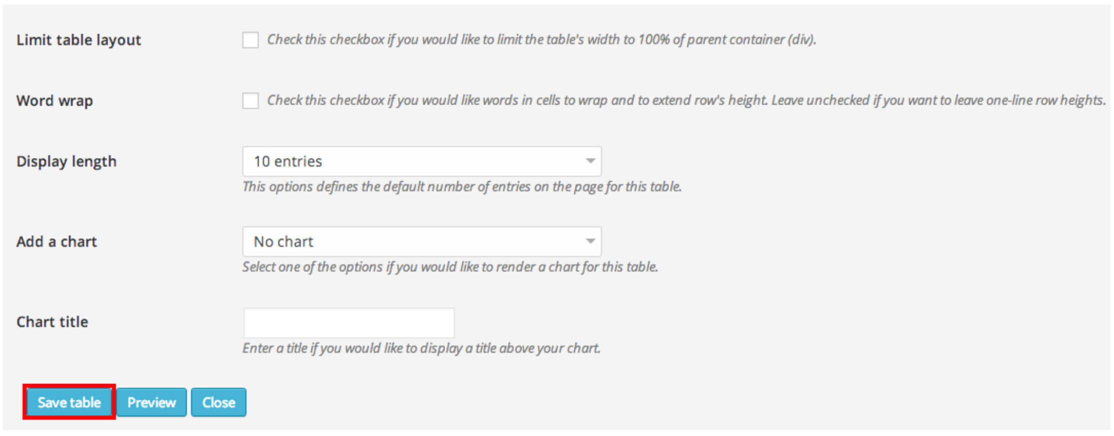


General setup

Table title	PHP table *If you want to display a header above your table, enter it here.*
Table type	Serialized PHP array *Choose a type of input source for your table.*
Input file or URL	http://wpdatatables.com/inp [Upload file] *Upload your file or provide the full URL here.* *For CSV or Excel input sources only uploaded files are supported.*

4. Set the additional table data and save the table

When the data source is provided you can define the additional table properties, such as enabling or disabling the advanced filter, making the table responsive, and so on (refer to the documentation sections 2.2, and chapter 5 to read about all these settings in detail). You can also define these properties later. Once you're done you can click the "Save table".



Limit table layout — Check this checkbox if you would like to limit the table's width to 100% of parent container (div).

Word wrap — Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave unchecked if you want to leave one-line row heights.

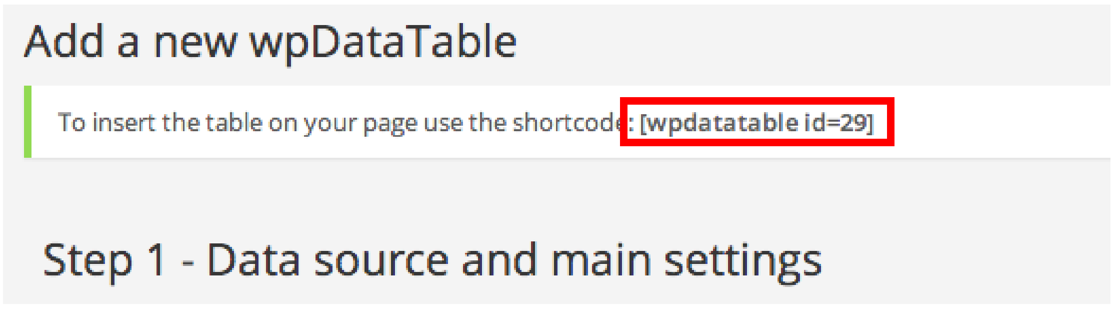
Display length — 10 entries — This options defines the default number of entries on the page for this table.

Add a chart — No chart — Select one of the options if you would like to render a chart for this table.

Chart title — Enter a title if you would like to display a title above your chart.

5. **Copy the table shortcode and paste on the post or page.**

Once the table is saved you will get a success notification. Then the table shortcode will appear in the upper part of the table edit page:



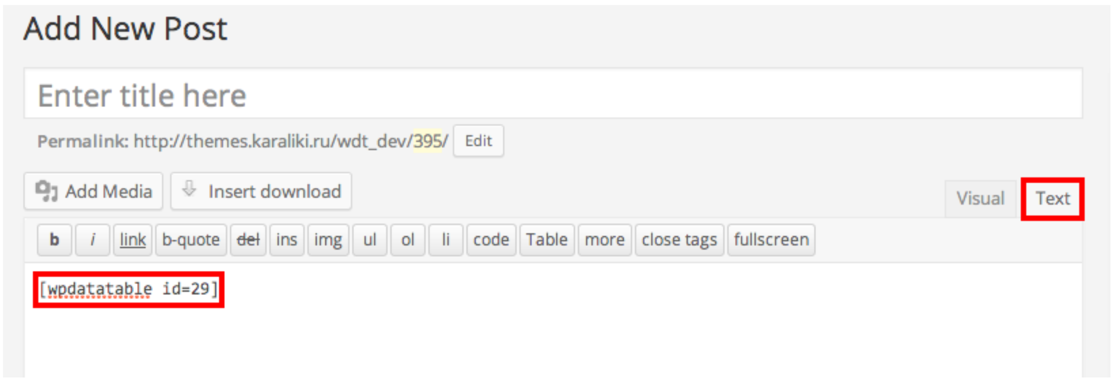
Add a new wpDataTable

To insert the table on your page use the shortcode: [wpdatatable id=29]

Step 1 - Data source and main settings

You can copy this shortcode and go to the post or page where you need the table.

There in the editor you can paste the shortcode. Please use the "Text" editor mode for this, since then you will be sure that no invalid markup will be pasted along with the shortcode:

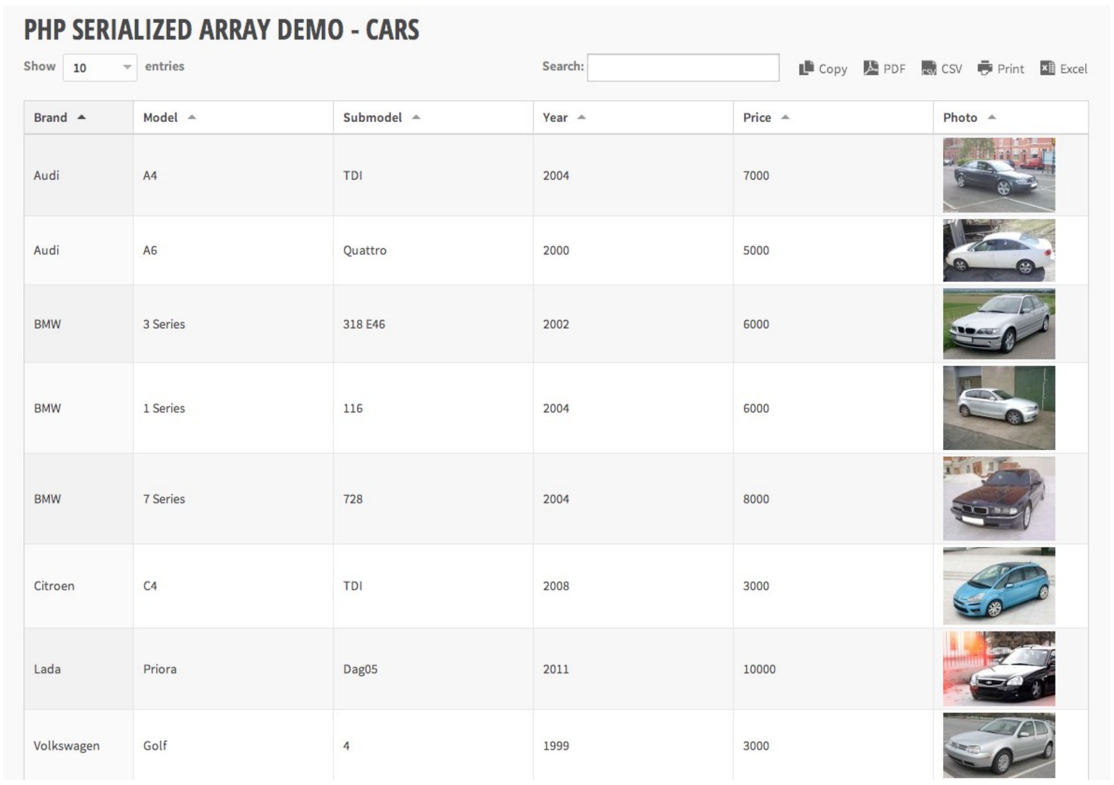

Add New Post

Enter title here

Permalink: http://themes.karaliki.ru/wdt_dev/395/ Edit

```
[wpdatatable id=29]
```

Once you've done that you can publish the page and open it in the site front-end


PHP SERIALIZED ARRAY DEMO - CARS

Brand	Model	Submodel	Year	Price	Photo
Audi	A4	TDI	2004	7000	
Audi	A6	Quattro	2000	5000	
BMW	3 Series	318 E46	2002	6000	
BMW	1 Series	116	2004	6000	
BMW	7 Series	728	2004	8000	
Citroen	C4	TDI	2008	3000	
Lada	Priora	Dag05	2011	10000	
Volkswagen	Golf	4	1999	3000	

If you get some errors during this process please refer to the troubleshooting section (chapter 10) of this documentation, FAQ page on the site. If you don't find a solution there you can refer to our support system.

> **Notice for developers:** main logic for this feature is in the file:
> source/class.wpdatatable.php, method arrayBasedConstruct(),

4.8 Shortcode attributes

This section gives a quick overview on different shortcode attributes that you can use

wpDataTables shortcode has several additional attributes that may come in handy:

- **show_only_chart:** you can set this to true in these cases when you don't want to render the table, and only the chart. So shortcode can look like this: [wpdatatable id=3 show_only_chart=true]
- **var1, var2, var3:** these three shortcode variables can be later used as placeholder values (if you want to use one wpDataTable in several posts with a little modification, e.g. predefined value of the filter for some column). You can use these variables like this:

[wpdatatable id=4 var1=4 var2="test"]

4.9 Placeholders

This section gives a quick overview on placeholders that you can use in default filter values or MySQL query.

Placeholders are predefined templates that are replaced with some actual values at the execution time. E.g. you can use this if you want a MySQL query to return only rows for currently logged in user.
Placeholders are supported in MySQL query, and in "Default values" input. Default values are used in advanced filter, and in front-end editor.

- **%CURRENT_USER_ID%** - replaced with the currently logged in user's ID (if the user is logged in). When you are using this in MySQL query when building the table please make sure that there are values in the table for the current user.

- **%VAR1%, %VAR2%, %VAR3%** - replaces with the shortcode attributes, e.g. if you have a shortcode like:

  [wpdatatable id=22 var1=41]

  %VAR1% will be replaced with **41** at the moment of execution.

5. Table features

This chapter contains detailed explanations, descriptions and overview of different table features of wpDataTables.

5.1 Server-side processing.

This section gives a detailed explanation of what exactly is the server-side processing feature; how it works, when you should and shouldn't use it.

One of the main benefits of MySQL DB engine is fast data processing, which means that when you need to sort whole table by one of the columns, or filter by some value, it will be done fast even when it contains several hundreds, or thousands or even millions of records.

What is the key difference between "usual" wpDataTables and wpDataTables with server-side processing turned on? When should the first and the second option be used?

- **"Usual" wpDataTables** (the ones that do not have server-side processing feature enabled) fetch all the rows output whole table to the front-end at once, then split it to pages, using Javascript; all data processing (page switching, sorting, filtering) is done on client's computer by Javascript. This works great for relatively small tables: once the table gets bigger, the page also gets larger, which means that the page generation and load time will also increase; also each sorting, filtering or page switching will take more time. So, for larger data sets "usual" wpDataTables will work slow, and for really large ones (10 000 rows and more) it can crash the page completely.

- **wpDataTables with server-side processing enabled** fetch only the amount of rows needed on the page at this exact moment. By default it equals to the number of rows that administrator defines for the table in the "Display length" setting, then the front-user can change it. E.g. if the table is configured to show 10 rows by default, only 10 rows would be queried from MySQL; and when the user switches to the next page, sorts the table by some column, filters by some column, an AJAX-request is sent to the server, data is processed by MySQL server, and 10 rows is returned again. This may slow down smaller tables, but for larger tables this may be the only option.

There's no "official" rule or, since it very much depends on the hosting, bandwidth, and the device of the front-end user, but I'd say, in general, file-based (Excel, CSV, XML, JSON) tables could be used for tables that don't have more then 1000 cells (columns * rows), MySQL table type can be used for any limits; but server-side processing feature is mandatory for data sets larger then 3000-4000 rows.

We will not go through the wpDataTable creation process once again, since it is completely same as for MySQL-based tables without server-side processing feature; please refer to documentation section 4.3 to see in detail what is the procedure to create a MySQL-based wpDataTable. But please keep in mind **these important limitations that this feature has:**

- **Queries with inner or left joins, or conditions will not work correctly with server-side processing feature enabled, when pasted directly to wpDataTables**. E.g. this will work fine: "**SELECT * FROM table**", and this can cause trouble: "**SELECT t1.*, t2.* FROM t1 INNER JOIN t2 ON t1.id=t2.some_id WHERE t1.price < 2000**" This is because tables with server-side processing feature do automatic parsing of the query, and form new queries for sorting, filtering, etc., and with joins or complicated conditions, this parsing will not work. **To avoid this** please prepare a MySQL view (a stored query), which will return the data that you need, call it e.g. "**view1**" and then build a wpDataTabled based on a simple query like "**SELECT * FROM view1**". You can read more about MySQL views here.

- **Charts will not work correctly with tables that have server-side processing enabled.**
- **Row grouping** feature can also cause trouble when combined with server-side processing.

Additionally I would like to repeat once again the limitations that the plugin has for the MySQL tables and queries, because ignoring this can cause problems with server-side processing:

- **Please do not use spaces in the column names.** E.g. use '**column1**' instead of '**column 1**'.
- **Please do not put a semicolon in the end of the query.**
- **Please do not use column names that can coincide with reserved MySQL words** (like DATE, BY, ORDER, GROUP, ...).
- **Please do not use numeric column names** like 1, 22, etc.
- **Please do not use "ORDER BY" in the statement.** wpDataTables has its own sorting engine so it makes no sense to use MySQL's sorting, since it will be overridden. Also server-side processing feature adds this part of statement automatically when users triggers the sorting on the front-end, and having it in initial statement may cause the table to crash.

5.2 Responsiveness

This section gives a detailed overview on the responsive mode for the tables, and an instruction how to use it.

Responsiveness (in web design) is a feature that allows presenting the same content differently depending on the device that is used to view it. E.g. the same web site can be presented differently on a laptop, mobile phone or tablet. Most of WordPress themes support responsiveness, so we also added this feature to wpDataTables.

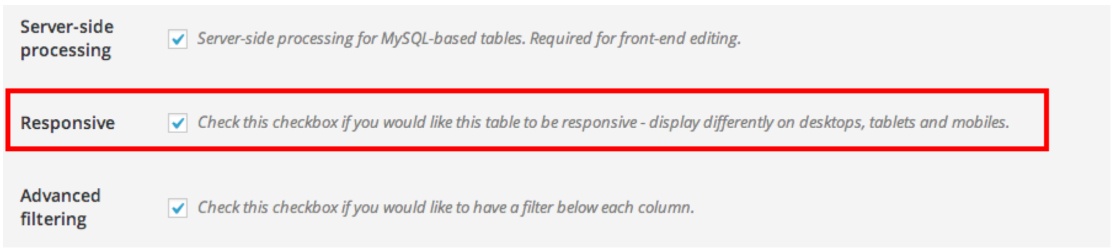
Any wpDataTable can be responsive, it doesn't matter which data source do you use – MySQL or Excel, or whatever else. The responsiveness is a front-end feature, so it's applied when the table is already rendered.

In wpDataTables you can define yourself which columns do you want to be visible or hidden on tablets and/or mobiles. The hidden columns data will be still available for the users that would like to see it in a dropdown.

Instruction for creating a responsive wpDataTable:

1. **Create a wpDataTable**. Create a table using any of the available input sources. You can refer to documentation chapter 4 to see how to create wpDataTables based on different input streams.

2. **Set the responsive checkbox in table properties.** On the table edit page, in "Additional properties" of the table find the "Responsive" checkbox and check it:

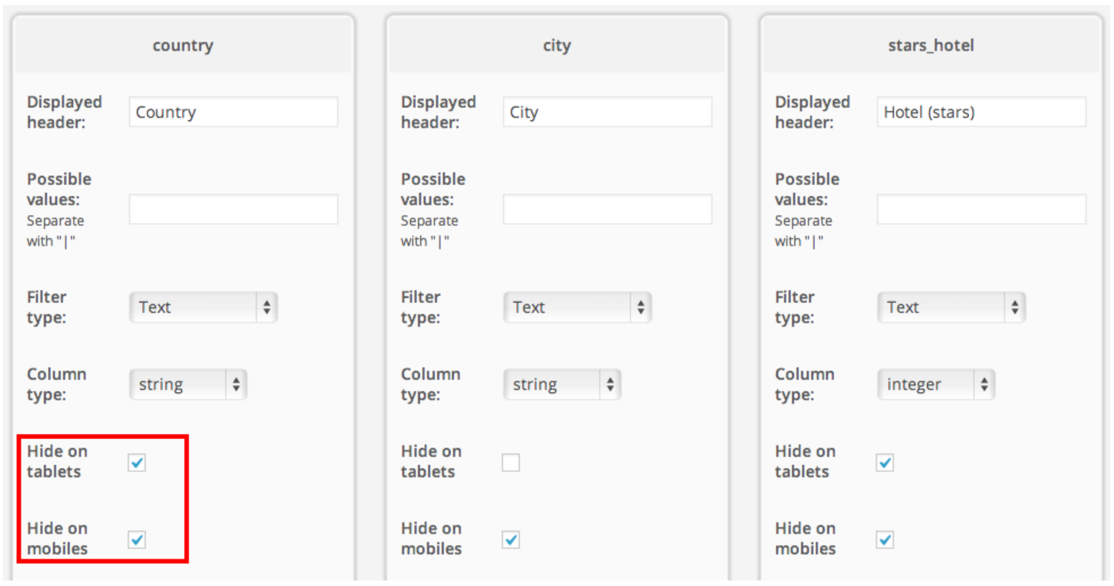


Server-side processing	✓ *Server-side processing for MySQL-based tables. Required for front-end editing.*
Responsive	✓ *Check this checkbox if you would like this table to be responsive - display differently on desktops, tablets and mobiles.*
Advanced filtering	✓ *Check this checkbox if you would like to have a filter below each column.*

Then click “Save table”.

3. **For each column define whether it will be hidden on mobiles and tablets**. In column properties for each column you will get two checkboxes:

- **“Hide on tablets”:** checking this checkbox will mean that the cells of this column won’t be visible on tablets; and the data from there will be available only after clicking on “+” button which will appear on the left of each row.
- **“Hide on mobiles”:** checking this checkbox will make the cells from this column invisible on mobile phone screens, but the data will be available after clicking on the “+” button which will be on the left of each row.

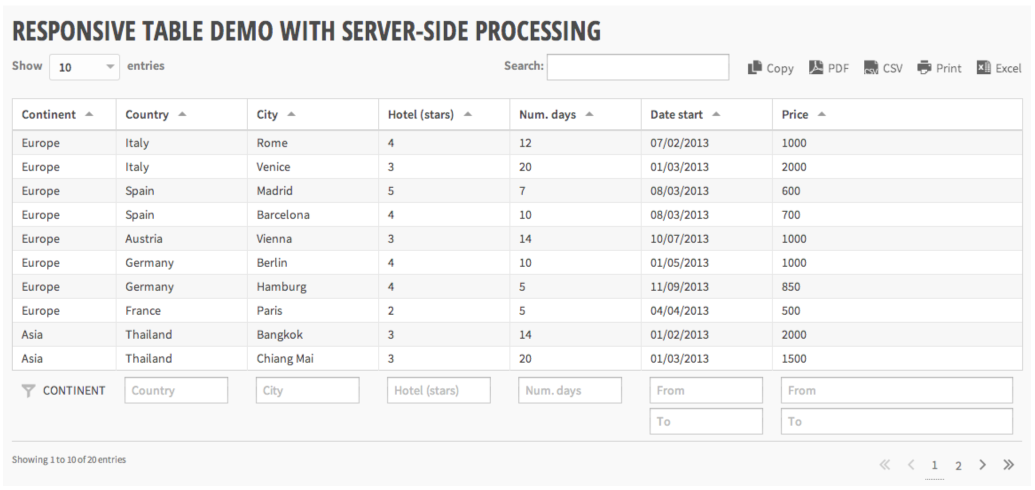


	country	city	stars_hotel
Displayed header:	Country	City	Hotel (stars)
Possible values: Separate with "|"			
Filter type:	Text	Text	Text
Column type:	string	string	integer
Hide on tablets	✓		✓
Hide on mobiles	✓	✓	✓

Do not forget to leave at least one column visible both for mobiles and tablets!

When you configure which columns will be visible and which will be hidden on different types of devices you can save the table, publish the shortcode in some post or page, and open the page in your site frontend. Then you can resize your browser's window to see the responsive mode in action:

Full-size (Deskop):

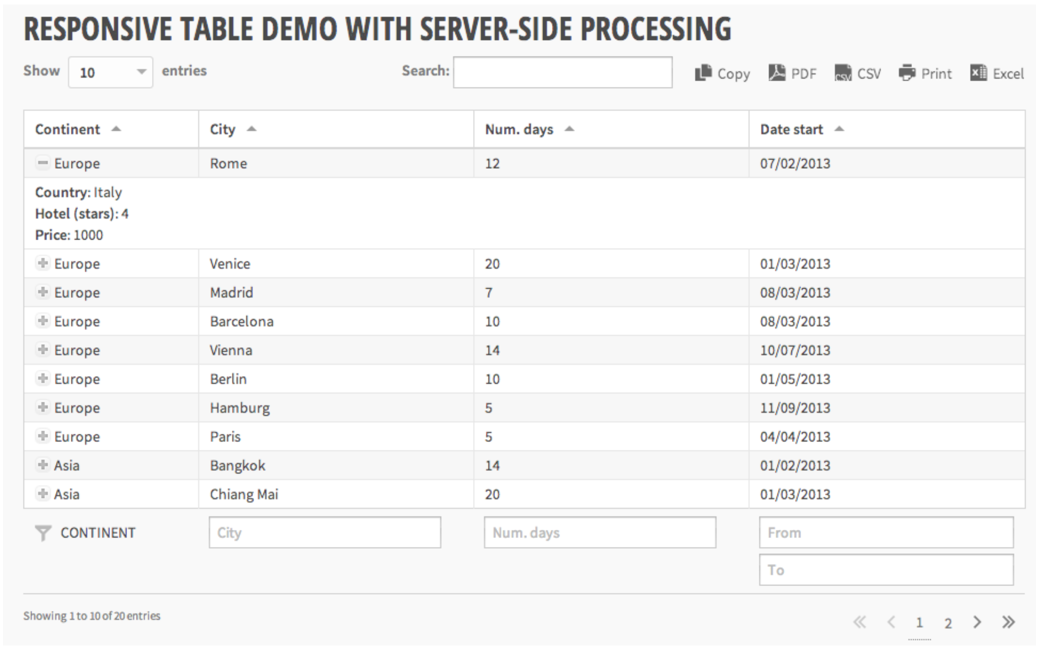


RESPONSIVE TABLE DEMO WITH SERVER-SIDE PROCESSING

Continent	Country	City	Hotel (stars)	Num. days	Date start	Price
Europe	Italy	Rome	4	12	07/02/2013	1000
Europe	Italy	Venice	3	20	01/03/2013	2000
Europe	Spain	Madrid	5	7	08/03/2013	600
Europe	Spain	Barcelona	4	10	08/03/2013	700
Europe	Austria	Vienna	3	14	10/07/2013	1000
Europe	Germany	Berlin	4	10	01/05/2013	1000
Europe	Germany	Hamburg	4	5	11/09/2013	850
Europe	France	Paris	2	5	04/04/2013	500
Asia	Thailand	Bangkok	3	14	01/02/2013	2000
Asia	Thailand	Chiang Mai	3	20	01/03/2013	1500
CONTINENT	Country	City	Hotel (stars)	Num. days	From / To	From / To

Showing 1 to 10 of 20 entries

Tablet size: note that some of the cells are hidden, but available in a dropdown when we click on the expand button.

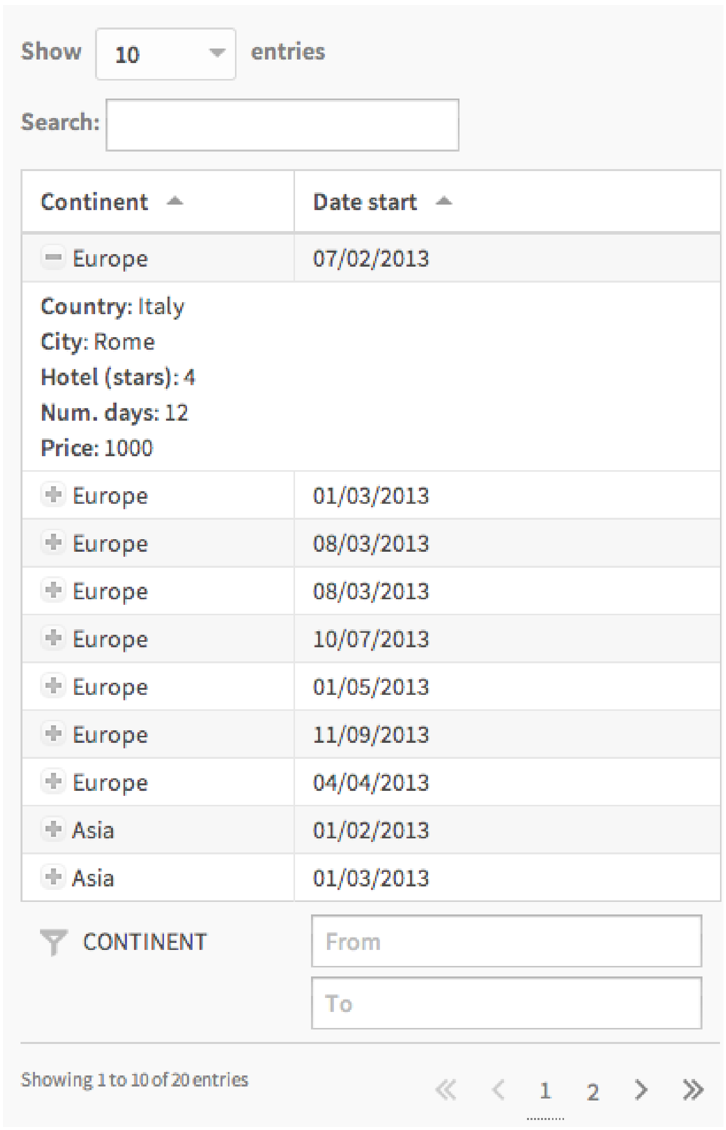


RESPONSIVE TABLE DEMO WITH SERVER-SIDE PROCESSING

Continent	City	Num. days	Date start
− Europe	Rome	12	07/02/2013
Country: Italy **Hotel (stars)**: 4 **Price**: 1000			
+ Europe	Venice	20	01/03/2013
+ Europe	Madrid	7	08/03/2013
+ Europe	Barcelona	10	08/03/2013
+ Europe	Vienna	14	10/07/2013
+ Europe	Berlin	10	01/05/2013
+ Europe	Hamburg	5	11/09/2013
+ Europe	Paris	5	04/04/2013
+ Asia	Bangkok	14	01/02/2013
+ Asia	Chiang Mai	20	01/03/2013
CONTINENT	City	Num. days	From / To

Showing 1 to 10 of 20 entries

Mobile size: note that only some of the cells are visible for all of the rows, but all the others are available in a dropdown when we click on the expand button.



Continent	Date start
− Europe	07/02/2013
Country: Italy City: Rome Hotel (stars): 4 Num. days: 12 Price: 1000	
+ Europe	01/03/2013
+ Europe	08/03/2013
+ Europe	08/03/2013
+ Europe	10/07/2013
+ Europe	01/05/2013
+ Europe	11/09/2013
+ Europe	04/04/2013
+ Asia	01/02/2013
+ Asia	01/03/2013

Notice for developers: main logic for this feature is in JS files: assets/js/responsive/datatables.responsive.js, assets/js/responsive/lodash.min.js, assets/js/wpdatatables/wpdatatables.js (the minified version is used by default, so you would need to include the non-minified one if you would like to make changes)

5.3 Advanced filtering, filtering in a form or a widget

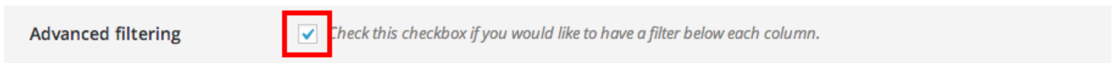
This section gives a detailed overview on the advanced filtering feature. A detailed instruction on the usage is given, and different filter types and algorithms are explained.

Advanced filtering is a special feature of wpDataTables which allows front-end table users to filter the data in the tables by the values of each individual column; e.g. get only the records where the price is between two values; or name contains some substring. It is possible to combine multiple advanced filters. This makes it possible to use wpDataTables as a filterable catalog. Advanced filter provides multiple input types with different behavior for end user convenience.

One more thing to notice is that filtering logic is combining, so you can filter by several columns at once, building complicated queries like "*Get all the rows where the price is between 1000 and 5000 and starting date is between 01/01/2014 and 01/03/2015, and the continent is either Asia or Africa*".

1. **Enabling advanced filter.**

To enable advanced filter for a wpDataTable you need to tick the "Advanced filtering" checkbox in "Additional settings" of Step 1 (it is pre-selected by default).

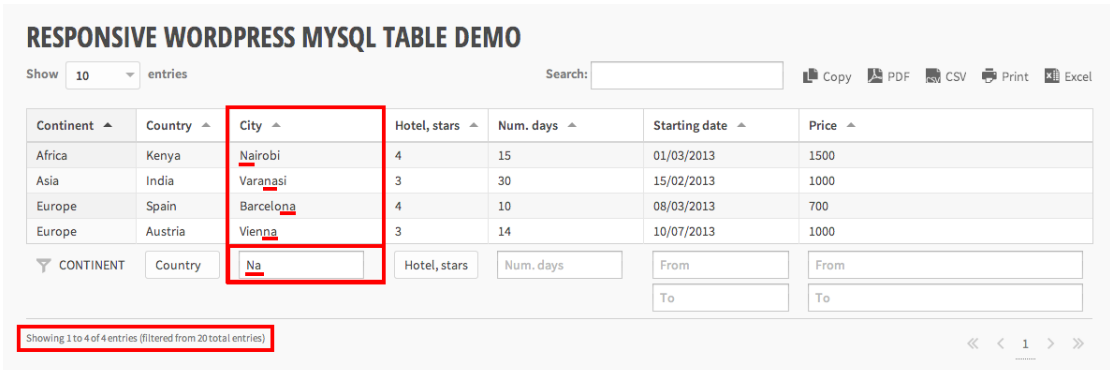


Advanced filtering ✓ Check this checkbox if you would like to have a filter below each column.

When this checkbox is checked, it will become possible to also define a filter type for each of the columns in Step 2 of the table edit page. Different filter types produce different filtering behavior, so now we will go through all of these in detail.

2. Text filter

This filter type is a basic one: it will perform the search based on the fragment of text that you enter. Only the rows, where the text in the cell of this column contain the entered text will be shown in the table:



RESPONSIVE WORDPRESS MYSQL TABLE DEMO

Show 10 entries Search:

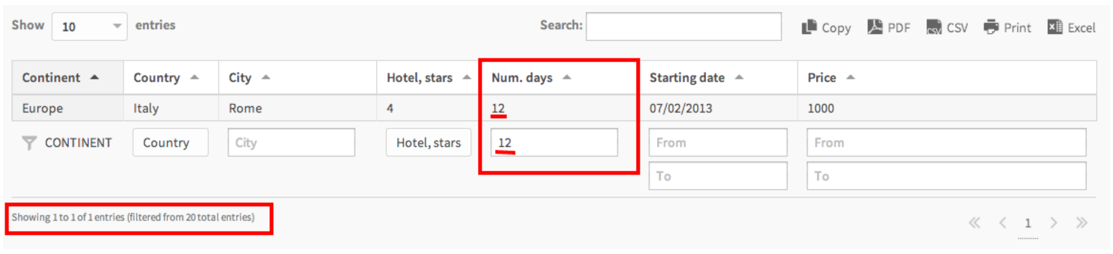
Continent	Country	City	Hotel, stars	Num. days	Starting date	Price
Africa	Kenya	Nairobi	4	15	01/03/2013	1500
Asia	India	Varanasi	3	30	15/02/2013	1000
Europe	Spain	Barcelona	4	10	08/03/2013	700
Europe	Austria	Vienna	3	14	10/07/2013	1000
CONTINENT	Country	Na	Hotel, stars	Num. days	From / To	From / To

Showing 1 to 4 of 4 entries (filtered from 20 total entries)

In this example the “City” column has a text filter type assigned. When we enter the “**na**” fragment, only the rows that contain this fragment in the city names are shown (**Na**irobi, Vara**na**si, Vien**na**, Barcelo**na**), all other rows are filtered out.

3. Number filter

This is another basic filter type. When some number will be entered in the filter, only the rows, where this column has exactly the same number, will be displayed:



Show 10 entries Search: Copy PDF CSV Print Excel

Continent	Country	City	Hotel, stars	Num. days	Starting date	Price
Europe	Italy	Rome	4	12	07/02/2013	1000
CONTINENT	Country	City	Hotel, stars	12	From / To	From / To

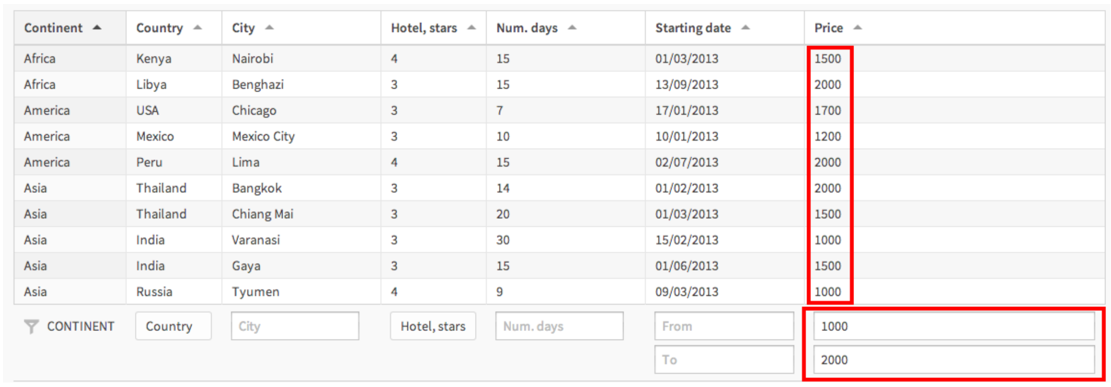
Showing 1 to 1 of 1 entries (filtered from 20 total entries)

1

E. g. in this example "Num. days." Column has a number filter type, "12" is entered, and only one row is shown, because only this row has "12" value of this cell.

4. Number range filter

This filter type should be used when you want table users to be able to filter the table by the values of some column not strictly, but within some ranges of numbers: e.g. values between 3 and 7, or just "greater then 3", and so on. It will output two inputs: "from" and "to":

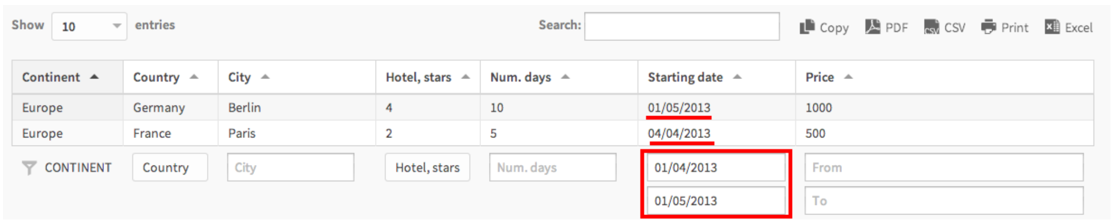


Continent	Country	City	Hotel, stars	Num. days	Starting date	Price
Africa	Kenya	Nairobi	4	15	01/03/2013	1500
Africa	Libya	Benghazi	3	15	13/09/2013	2000
America	USA	Chicago	3	7	17/01/2013	1700
America	Mexico	Mexico City	3	10	10/01/2013	1200
America	Peru	Lima	4	15	02/07/2013	2000
Asia	Thailand	Bangkok	3	14	01/02/2013	2000
Asia	Thailand	Chiang Mai	3	20	01/03/2013	1500
Asia	India	Varanasi	3	30	15/02/2013	1000
Asia	India	Gaya	3	15	01/06/2013	1500
Asia	Russia	Tyumen	4	9	09/03/2013	1000
CONTINENT	Country	City	Hotel, stars	Num. days	From / To	1000 / 2000

In this example the "Price" column has the number range filter enabled. 1000 is put as the lower limit, and 2000 as the higher limit; so only the rows that have the price values between these two limits are shown (limit values are included as well).

5. Date range filter

Date range filter type should be used only when the column has a date type, and when you would like your users to be able to filter the table data by some ranges of dates, e.g. “later then first june”, or “between 1st may and 15th of august”, etc.:



Show 10 entries Search: Copy PDF CSV Print Excel

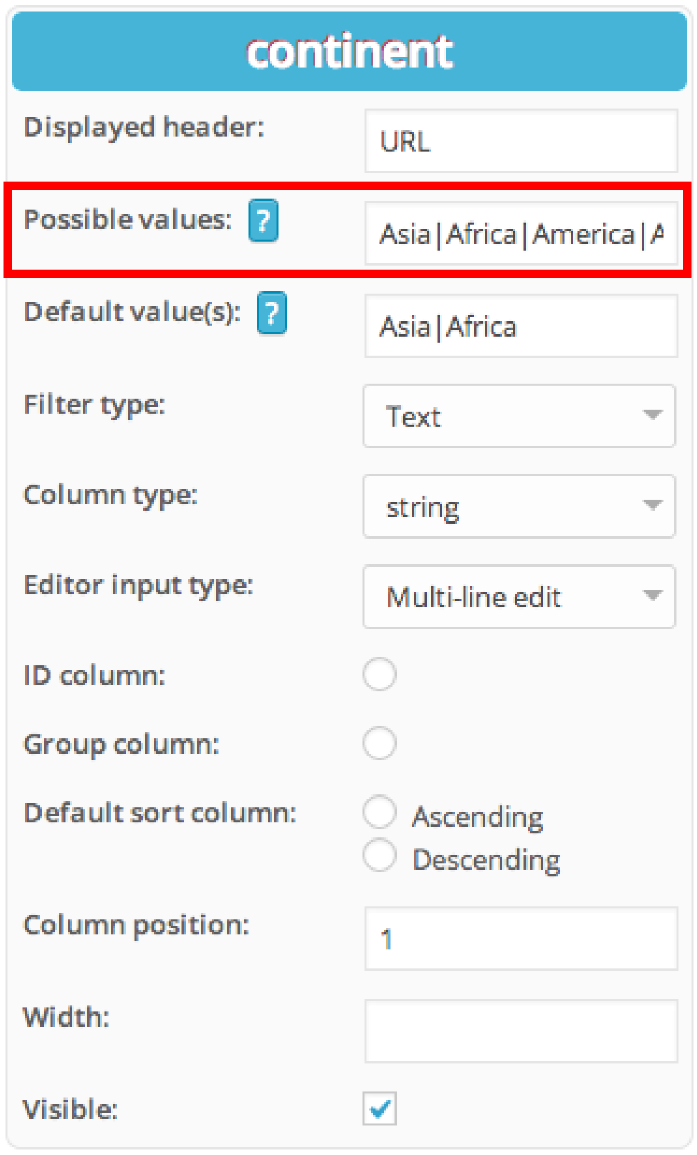
Continent	Country	City	Hotel, stars	Num. days	Starting date	Price
Europe	Germany	Berlin	4	10	01/05/2013	1000
Europe	France	Paris	2	5	04/04/2013	500
CONTINENT	Country	City	Hotel, stars	Num. days	01/04/2013 01/05/2013	From To

In this example the “Starting date” column has a Date Range filter, where the “From” value is set to 01/04/2013, and the “To” value to “01/05/2013”. So only the rows, where cells in the starting date column have the date value between these 2 are shown.

6. Select box filter

Selectbox filter type is one of the most convenient filters, it will allow the front-end users to select one of the pre-defined values for some of the columns, and the table data will be filtered based on this value.

For tables which do not have server-side processing feature enabled the possible values for this filter will be pre-filled automatically from the column “in the order of appearance”, i.e. it will pick all unique values from the cells in this column. For tables which do have server-side processing enabled you would need to define possible values, separating them with a “|” sign:



continent

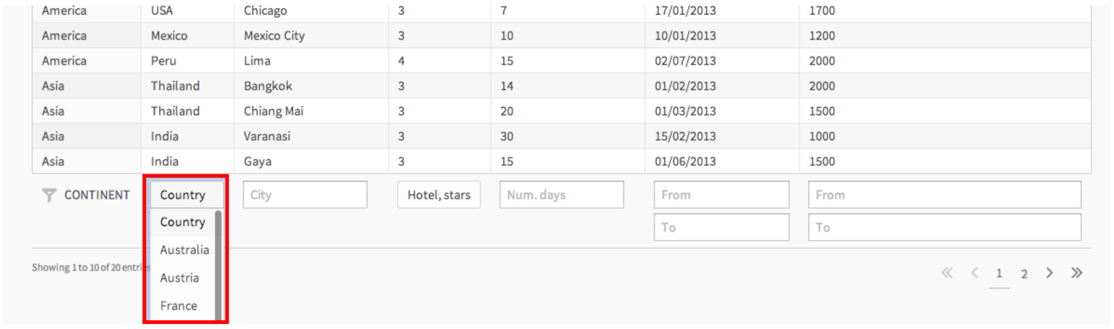
Displayed header:	URL
Possible values:	Asia|Africa|America|A
Default value(s):	Asia|Africa
Filter type:	Text
Column type:	string
Editor input type:	Multi-line edit
ID column:	○
Group column:	○
Default sort column:	○ Ascending ○ Descending
Column position:	1
Width:	
Visible:	✓

E.g. you if you need to filter by "Fruits" column, you can put in this input field:

Apple|Peach|Lemon|Orange

You can also override the default generated values with this input field, if you want them to display in order different from order of appearance in table.

The filter is rendered on the front-end as a dropdown selectbox:



America	USA	Chicago	3	7	17/01/2013	1700
America	Mexico	Mexico City	3	10	10/01/2013	1200
America	Peru	Lima	4	15	02/07/2013	2000
Asia	Thailand	Bangkok	3	14	01/02/2013	2000
Asia	Thailand	Chiang Mai	3	20	01/03/2013	1500
Asia	India	Varanasi	3	30	15/02/2013	1000
Asia	India	Gaya	3	15	01/06/2013	1500

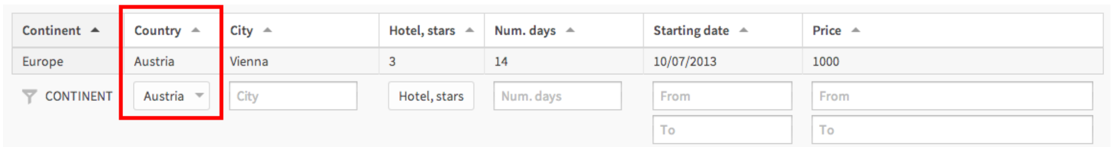
CONTINENT | Country | City | Hotel, stars | Num. days | From / To | From / To

Country
Australia
Austria
France

Showing 1 to 10 of 20 entri

« ‹ 1 2 › »

When user chooses one of the values in the dropdown, the table is filtered, and only the rows that contain the selected value in the column will be displayed:



Continent ▲	Country ▲	City ▲	Hotel, stars ▲	Num. days ▲	Starting date ▲	Price ▲
Europe	Austria	Vienna	3	14	10/07/2013	1000

CONTINENT | Austria | City | Hotel, stars | Num. days | From / To | From / To

7. Checkbox filter

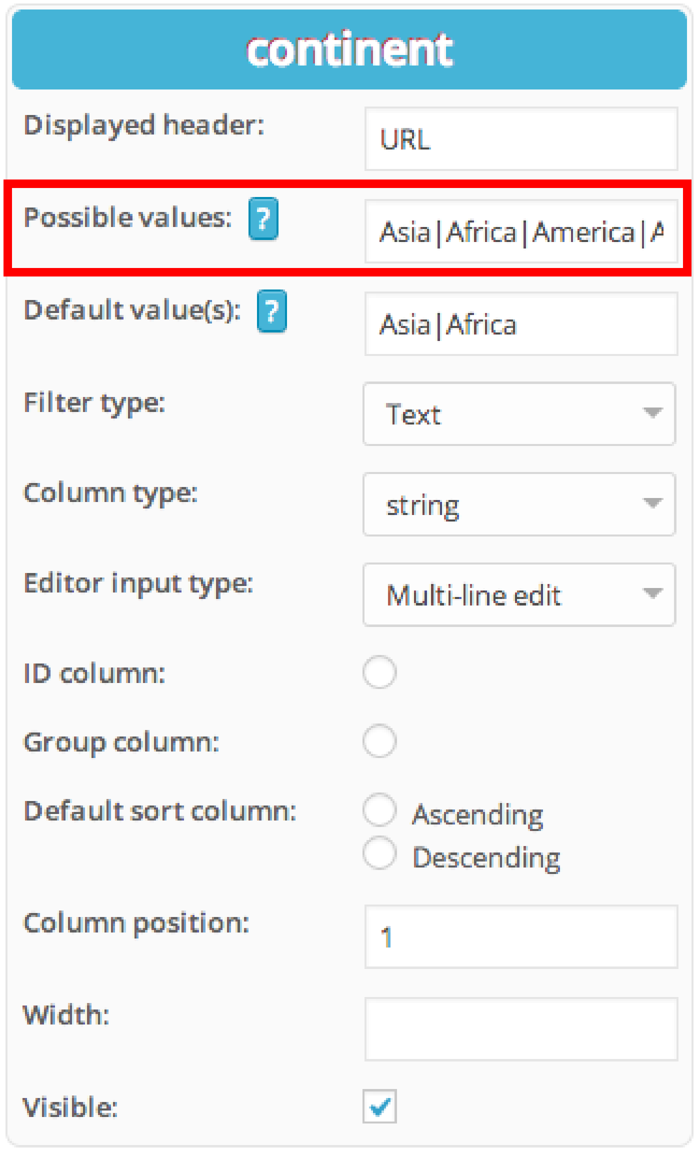
Checkbox filter is a convenient way to filter a table by more then one value of a column. If you want the users to be available to filter the rows in the table with logic like "Show all the rows where the continent is either Asia or Africa", so that you would define exact options, this is the right option for you.

For tables which do not have server-side processing feature enabled the possible values for this filter will be pre-filled automatically from the column "in the order of appearance", i.e. it will pick all unique values from the cells in this column. For tables which do have server-side processing enabled you would need to define possible values, separating them with a

“|” sign, same as for the selectbox filter type (See pt. 6 from this list for the details).

You can also override the default generated values with this input field, if you want them to display in order different from order of appearance in table.

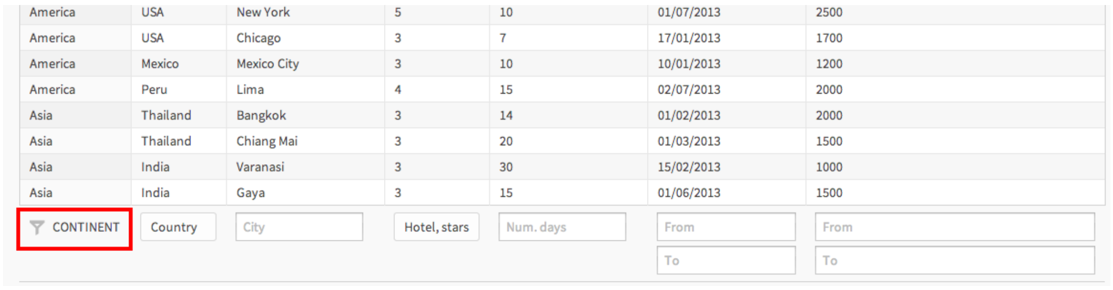
To use this filter you would need to set “Fiter type” setting as “Checkbox”, and then optionally define the possible values (but you must do it if you use server-side processing feature for this table):



continent

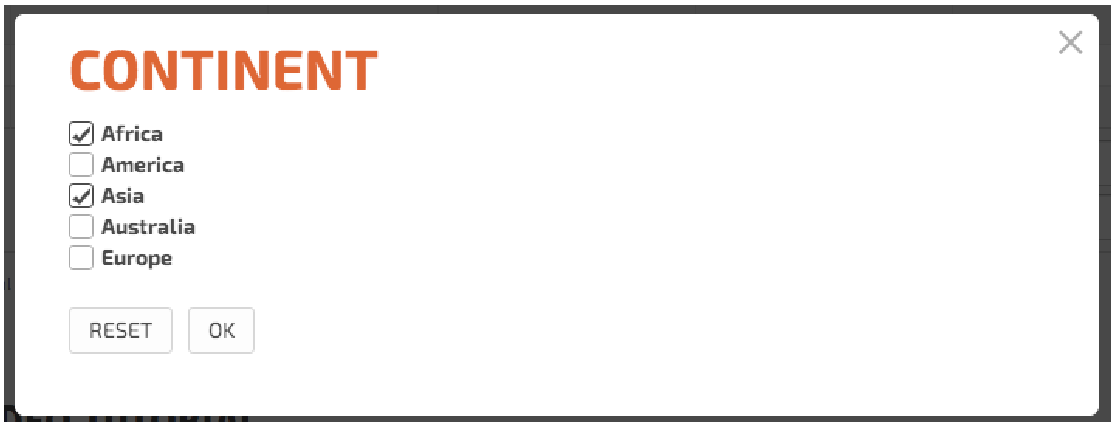
Displayed header:	URL
Possible values: ?	Asia | Africa | America | A
Default value(s): ?	Asia | Africa
Filter type:	Text
Column type:	string
Editor input type:	Multi-line edit
ID column:	○
Group column:	○
Default sort column:	○ Ascending ○ Descending
Column position:	1
Width:	
Visible:	✓

Then on the front-end users will get a button under this column:



America	USA	New York	5	10	01/07/2013	2500
America	USA	Chicago	3	7	17/01/2013	1700
America	Mexico	Mexico City	3	10	10/01/2013	1200
America	Peru	Lima	4	15	02/07/2013	2000
Asia	Thailand	Bangkok	3	14	01/02/2013	2000
Asia	Thailand	Chiang Mai	3	20	01/03/2013	1500
Asia	India	Varanasi	3	30	15/02/2013	1000
Asia	India	Gaya	3	15	01/06/2013	1500
CONTINENT	Country	City	Hotel, stars	Num. days	From To	From To

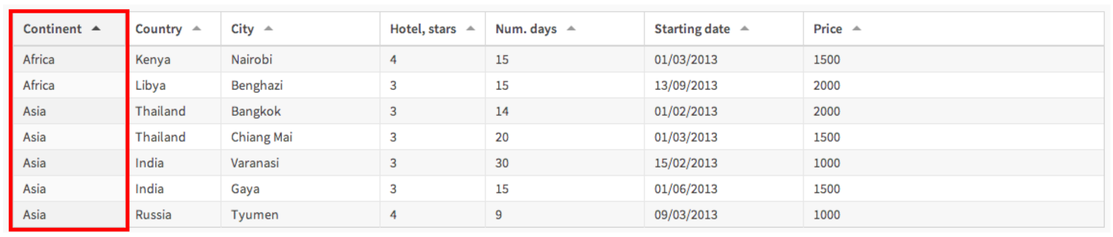
When users click on this button a pop-up appears, where all the possible values of this column are listed with checkboxes:



CONTINENT

- [x] Africa
- [ ] America
- [x] Asia
- [ ] Australia
- [ ] Europe

When user ticks any of the checkboxes the table will be rendered simultaneously, and only the rows that contain the selected values in the column will be displayed:

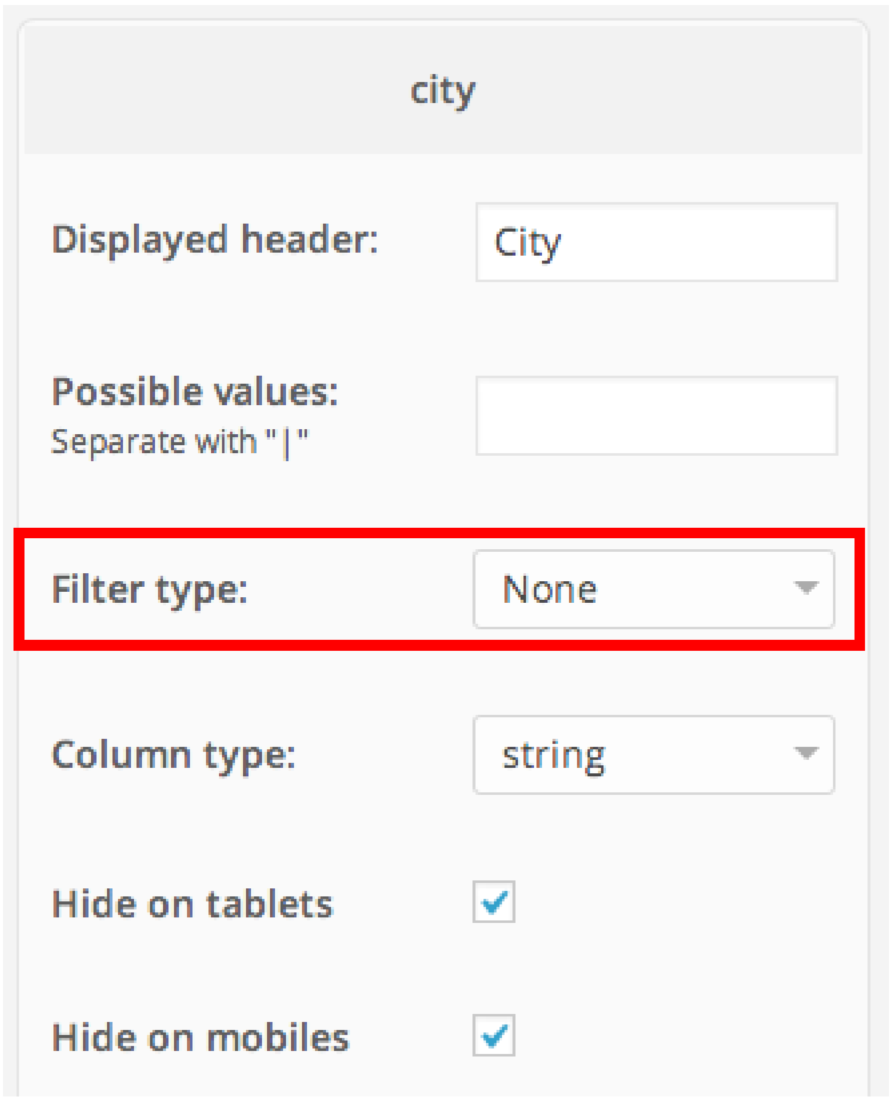


Continent	Country	City	Hotel, stars	Num. days	Starting date	Price
Africa	Kenya	Nairobi	4	15	01/03/2013	1500
Africa	Libya	Benghazi	3	15	13/09/2013	2000
Asia	Thailand	Bangkok	3	14	01/02/2013	2000
Asia	Thailand	Chiang Mai	3	20	01/03/2013	1500
Asia	India	Varanasi	3	30	15/02/2013	1000
Asia	India	Gaya	3	15	01/06/2013	1500
Asia	Russia	Tyumen	4	9	09/03/2013	1000

"**OK**" button closes the popup, and "**Reset**" button unticks all the checkboxes.

8. Disabling the filter for one column

Sometimes you don't want to allow your users to filter the table by all of the columns, some columns are less significant, and it makes sense to hide advanced filter for them. For this you can just define the filter type as "None" for these columns:



city

Displayed header: City

Possible values:
Separate with "|"

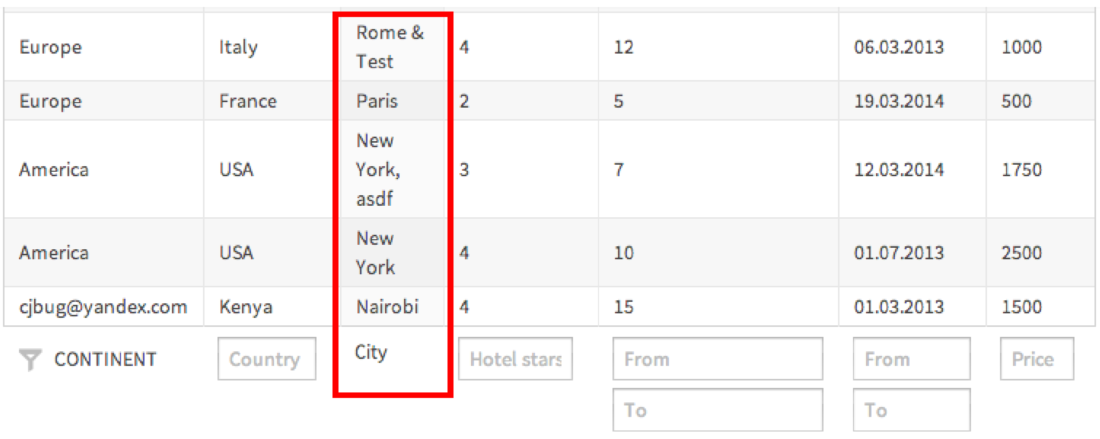
Filter type: None

Column type: string

Hide on tablets ✓

Hide on mobiles ✓

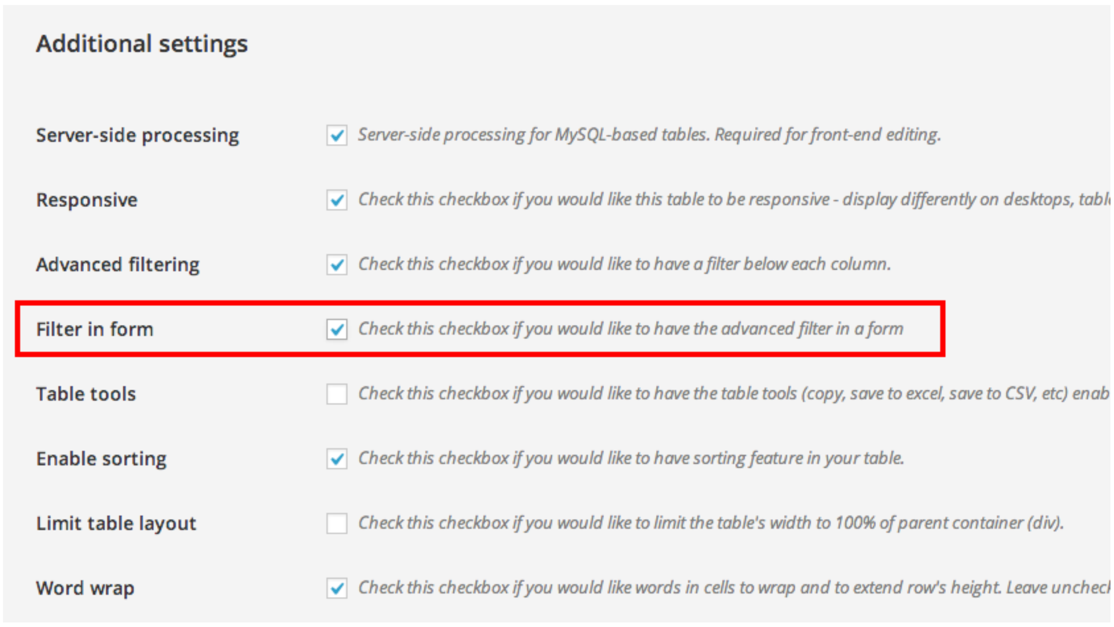
Then this column will not have a filter on the front-end:



Europe	Italy	Rome & Test	4	12	06.03.2013	1000
Europe	France	Paris	2	5	19.03.2014	500
America	USA	New York, asdf	3	7	12.03.2014	1750
America	USA	New York	4	10	01.07.2013	2500
cjbug@yandex.com	Kenya	Nairobi	4	15	01.03.2013	1500
CONTINENT	Country	City	Hotel stars	From / To	From / To	Price

9. Advanced filter in a form

Having an advanced filter below each column is convenient, but not for all cases; if there is a lot of columns, or if you would like to filter by invisible columns it makes sense to have the filter separated from the table itself. This is supported by wpDataTables with the “filter in form” feature. To use it you just need to set the “Filter in form” checkbox in the “Additional settings” of step 1 of the table edit page:

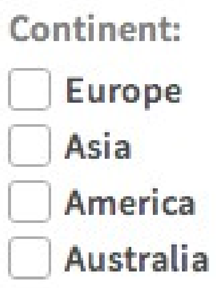


Additional settings

Server-side processing	✓ Server-side processing for MySQL-based tables. Required for front-end editing.
Responsive	✓ Check this checkbox if you would like this table to be responsive - display differently on desktops, tabl
Advanced filtering	✓ Check this checkbox if you would like to have a filter below each column.
Filter in form	✓ Check this checkbox if you would like to have the advanced filter in a form
Table tools	☐ Check this checkbox if you would like to have the table tools (copy, save to excel, save to CSV, etc) enab
Enable sorting	✓ Check this checkbox if you would like to have sorting feature in your table.
Limit table layout	☐ Check this checkbox if you would like to limit the table's width to 100% of parent container (div).
Word wrap	✓ Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave uncheck

All the other settings for filters are the same. Please keep in mind just a couple of things:

- **Checkbox filter will not appear in a popup, but all checkboxes will be displayed in a form**:

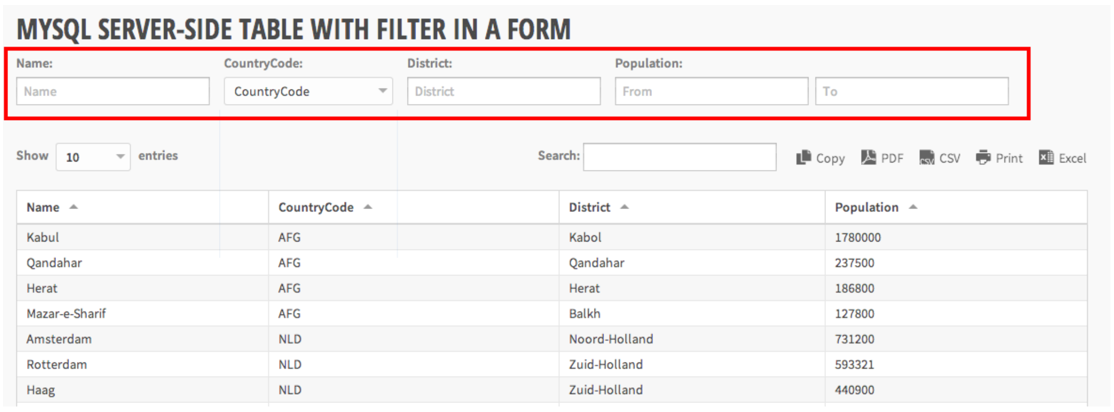


Continent:

☐ Europe
☐ Asia
☐ America
☐ Australia

- **You can also define filter types for invisible columns.** The columns will not be visible in the table, but the filtering block will appear in the filtering form. So you can extend the filtering functionality without overloading the table – just keep in mind that having too many invisible columns is not good for performance.

When this checkbox is checked, and the table is saved, front-end look of the table will be changed: a filtering form will appear above the table:


MYSQL SERVER-SIDE TABLE WITH FILTER IN A FORM

Name:	CountryCode:	District:	Population:	
Name	CountryCode	District	From	To

Show 10 entries   Search: Copy PDF CSV Print Excel

Name	CountryCode	District	Population
Kabul	AFG	Kabol	1780000
Qandahar	AFG	Qandahar	237500
Herat	AFG	Herat	186800
Mazar-e-Sharif	AFG	Balkh	127800
Amsterdam	NLD	Noord-Holland	731200
Rotterdam	NLD	Zuid-Holland	593321
Haag	NLD	Zuid-Holland	440900

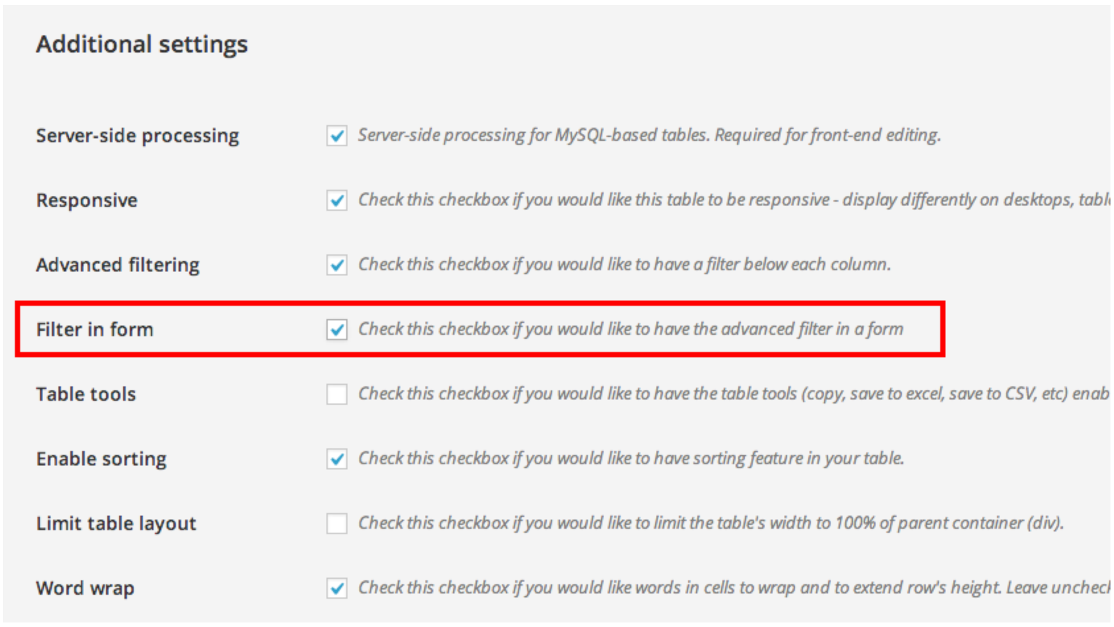
The logic of the filters is the same as for the usual look of the advanced filter, when the filters are rendered in table's footer.

10. Advanced filter in a widget (in WordPress sidebar)

WordPress has a built-in functionality of widgets and sidebars, so wpDataTables uses it for the filtering feature: this way you can define yourself the region of the page where to render the filtering form.

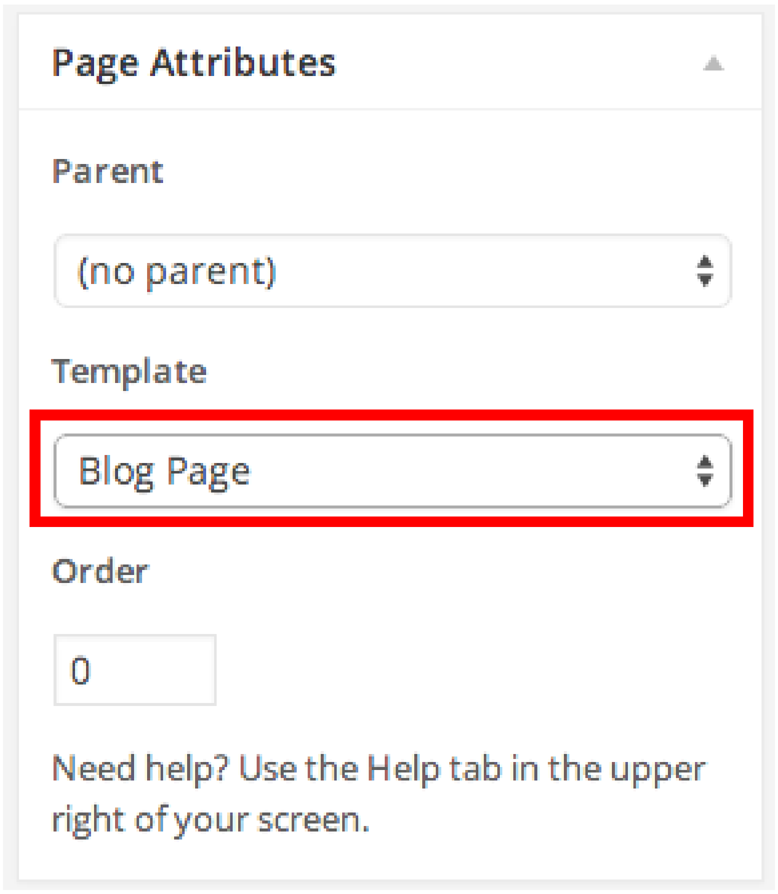
To use it you need to follow these rules:

- Tick the checkbox "Filter in form" on the table settings:


Additional settings

Server-side processing	✓	Server-side processing for MySQL-based tables. Required for front-end editing.
Responsive	✓	Check this checkbox if you would like this table to be responsive - display differently on desktops, tabl
Advanced filtering	✓	Check this checkbox if you would like to have a filter below each column.
Filter in form	✓	Check this checkbox if you would like to have the advanced filter in a form
Table tools	☐	Check this checkbox if you would like to have the table tools (copy, save to excel, save to CSV, etc) enab
Enable sorting	✓	Check this checkbox if you would like to have sorting feature in your table.
Limit table layout	☐	Check this checkbox if you would like to limit the table's width to 100% of parent container (div).
Word wrap	✓	Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave unchecl

- Use the shortcode in the page or post, which has some of the registered sidebars defined in the template (the title of the template depends on the theme you're using):


Page Attributes

Parent

(no parent)

Template

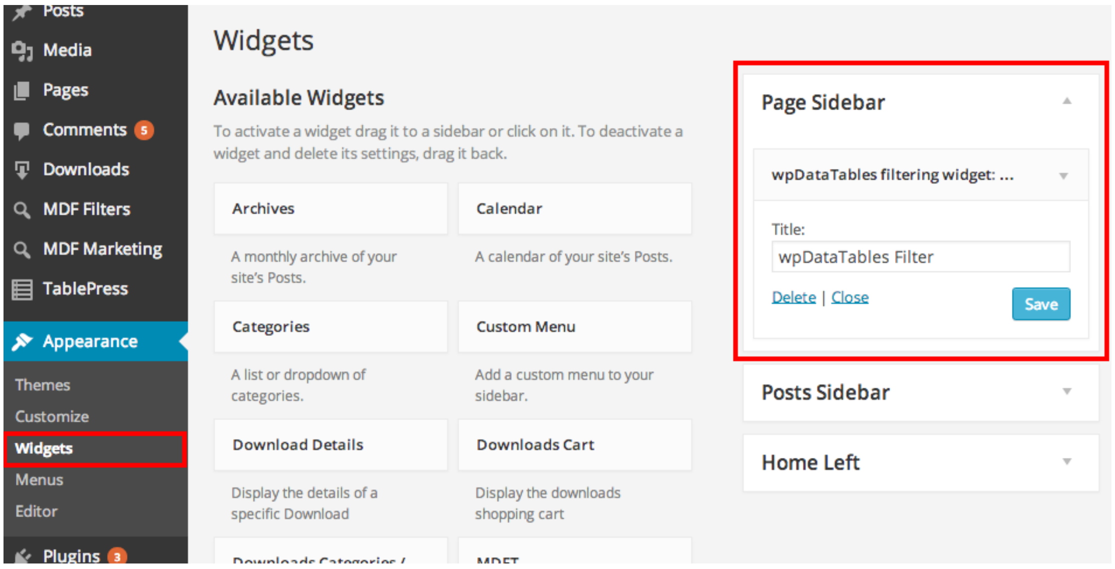
Blog Page

Order

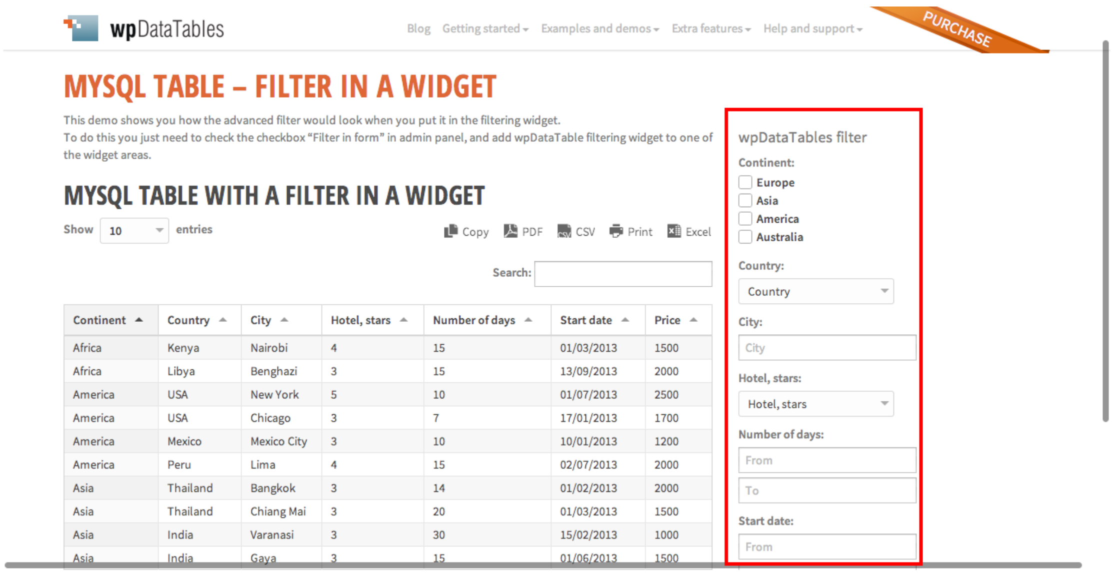
0

Need help? Use the Help tab in the upper right of your screen.

- In the "**Appearance**" -> "**Widgets**" area of the admin panel drag the "wpDataTables filtering widget" widget to the sidebar block of the page that you used to paste the shortcode. You can define there the title for this widget:


Posts
Media
Pages
Comments 5
Downloads
MDF Filters
MDF Marketing
TablePress
Appearance
Themes
Customize
Widgets
Menus
Editor
Plugins 3
Widgets
Available Widgets
To activate a widget drag it to a sidebar or click on it. To deactivate a widget and delete its settings, drag it back.
Archives
A monthly archive of your site's Posts.
Calendar
A calendar of your site's Posts.
Categories
A list or dropdown of categories.
Custom Menu
Add a custom menu to your sidebar.
Download Details
Display the details of a specific Download
Downloads Cart
Display the downloads shopping cart
Page Sidebar
wpDataTables filtering widget: ...
Title:
wpDataTables Filter
Delete | Close
Save
Posts Sidebar
Home Left

When this is done, the filter will be moved to the sidebar (or other region that you have chosen for it):



MYSQL TABLE – FILTER IN A WIDGET

This demo shows you how the advanced filter would look when you put it in the filtering widget.
To do this you just need to check the checkbox "Filter in form" in admin panel, and add wpDataTable filtering widget to one of the widget areas.

MYSQL TABLE WITH A FILTER IN A WIDGET

Show 10 entries

Copy PDF CSV Print Excel

Search:

Continent	Country	City	Hotel, stars	Number of days	Start date	Price
Africa	Kenya	Nairobi	4	15	01/03/2013	1500
Africa	Libya	Benghazi	3	15	13/09/2013	2000
America	USA	New York	5	10	01/07/2013	2500
America	USA	Chicago	3	7	17/01/2013	1700
America	Mexico	Mexico City	3	10	10/01/2013	1200
America	Peru	Lima	4	15	02/07/2013	2000
Asia	Thailand	Bangkok	3	14	01/02/2013	2000
Asia	Thailand	Chiang Mai	3	20	01/03/2013	1500
Asia	India	Varanasi	3	30	15/02/2013	1000
Asia	India	Gaya	3	15	01/06/2013	1500

wpDataTables filter

Continent:
- Europe
- Asia
- America
- Australia

Country: Country

City: City

Hotel, stars: Hotel, stars

Number of days: From To

Start date: From

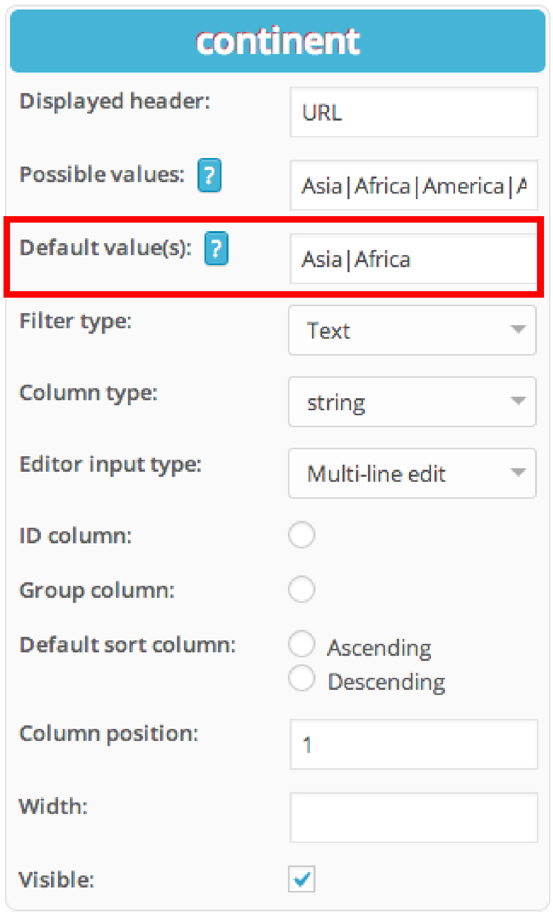
Advanced filter in a widget also has a "Reset filters" button which can be used to refresh the filter set.

> **Notice for developers:** main logic for this feature is in JS file: assets/js/jquery-datatables/jquery.dataTables.columnFilter.js, the JSON description of the columns for the filtering JS is generated in source/class.wpdatatable.php,

method getColumnFilterDefs(). The widget is initialized in source/class.filterwidget.php

10. Predefined default filter values

If you want to load the page with some predefined settings for the filters you can make use of the “Default values” input. Values from this input will be used as the predefined filter values, and also in the front-end editor if the table is editable. For the checkbox filters you can predefine several preselected values, separating them with a “|”:


continent

Displayed header:	URL
Possible values:	Asia|Africa|America|A
Default value(s):	Asia|Africa
Filter type:	Text
Column type:	string
Editor input type:	Multi-line edit
ID column:	○
Group column:	○
Default sort column:	○ Ascending ○ Descending
Column position:	1
Width:	
Visible:	✓

You can use placeholders (%CURRENT_USER_ID%, %VAR1%, %VAR2%, %VAR3%) in this input. See documentation section 4.9 for more information on placeholders.

11. Defining filter values in the URL

You can also predefine filter values for the table in the URL, which may come in handy when you would like to e.g. send your client a deeplink with some info from your catalog already filtered for him.
The URL key that will be parsed by the wpDataTables is "**wdtColumnFilter[KEY]**" where KEY is either a zero-based numerical index of the column you would like to filter, or a initial header of the column (header or the CSV or Excel table, or MySQL column header for MySQL-based column header). E.g.

?wdt_column_filter[1]=Asia&wdt_column_filter[country]=Thailand

The values parsed from this URL will be used as default filter values.

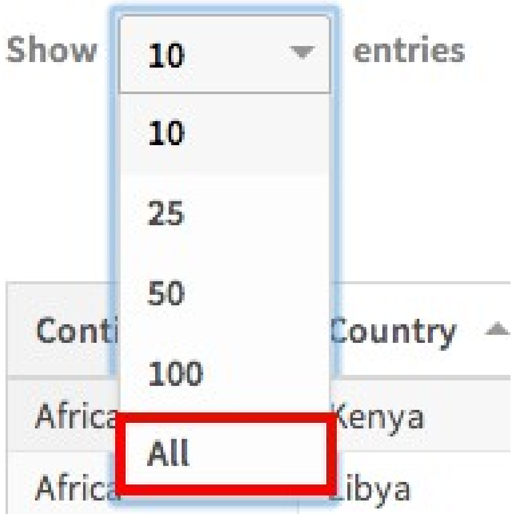
5.4 Table tools

This section gives a detailed overview on the "Table Tools" feature. A detailed instruction on the usage is given, and an explanation on how different buttons of this block work.

Table Tools is a 3rd party library, based on the TableTools plugin for DataTables. It is working in Adobe Flash on client side, which results in a number of limitations and possible problems in use; first thing I would like to mention these key limitations of this feature:

- Table Tools will not work on mobiles.
- Table Tools will not work without Flash player support.

- Excel file, generated by the TableTools, is not a “real” Excel file: it is a CSV file, with *.xls* extension. Most of the Excel installations would read it normally, but sometimes it can cause problems.
- **For tables with server-side processing only the visible rows will be exported to any format.** This means that if you want the whole table to be exported (or copied to clipboard), you would need to choose “Show All” feature in the “show ... entries” block, and then do the export:

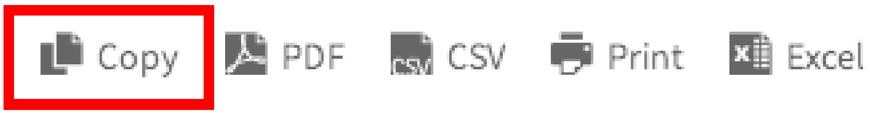


- **Generated PDF is not customizable**. Unfortunately the library does not allow “fine-tuning” of the generated PDF, so font, size, and other properties cannot be changed.
- **Other front-end plugins cannot be combined with Table Tools.** E.g. if you have row grouping enabled for the table, or some customizations you did yourself, they wouldn’t be reflected in the generated Excel, CSV or PDF document, neither in the clipboard.

Now we can go through all the available buttons and the functions.

1. **Copy**

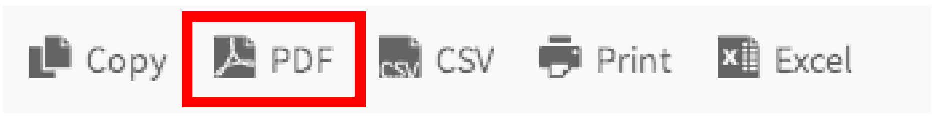
Copy button triggers the "copy to clipboard" function, so it copies all the rows that exist in the page to your OS clipboard and you can later paste it in some other software.

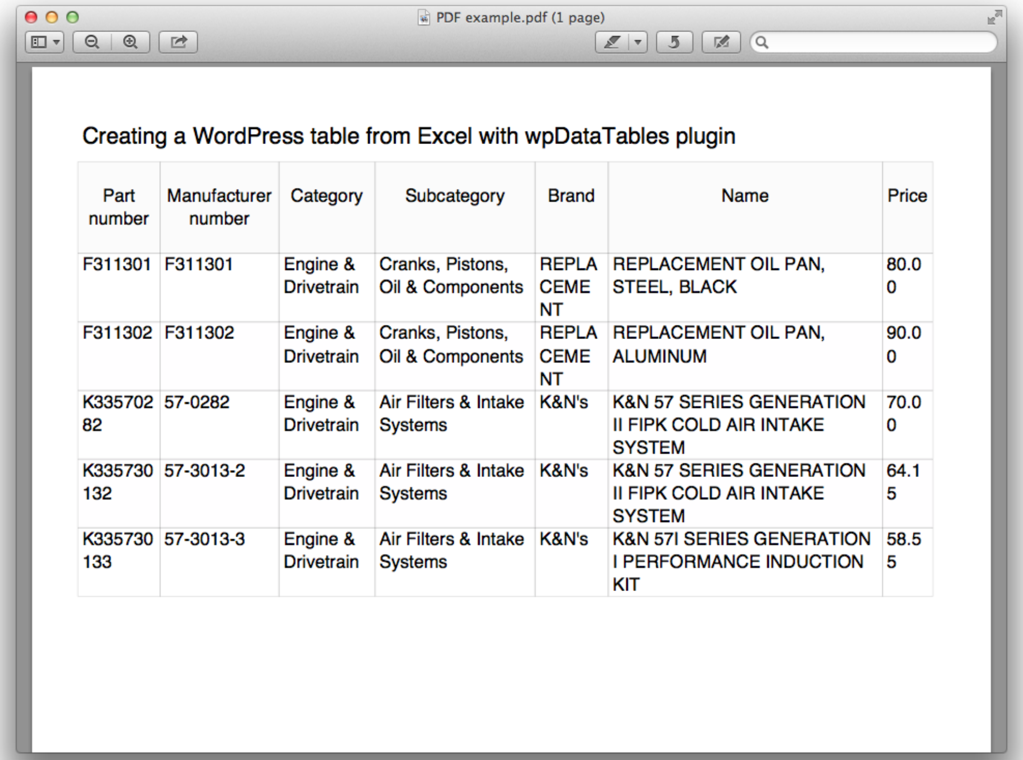


Once you click it, you get a popup with notification on how many rows were copied to the clipboard, e.g. "Copied 20 rows to the clipboard".
All the rows are copied to the clipboard, including headers, tab-separated; you can paste it then to e.g. MS Excel.
For tables with server-side processing enabled, this copies only the visible columns.

2. **PDF**

"PDF" button generates a new PDF document out of the table, and calls the browser's "Save as ..." dialog so you could choose a path for your file. It is a very basic PDF with the page title, and the table with simple formatting:





PDF example.pdf (1 page)

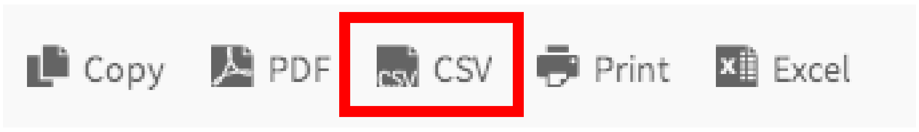
Creating a WordPress table from Excel with wpDataTables plugin

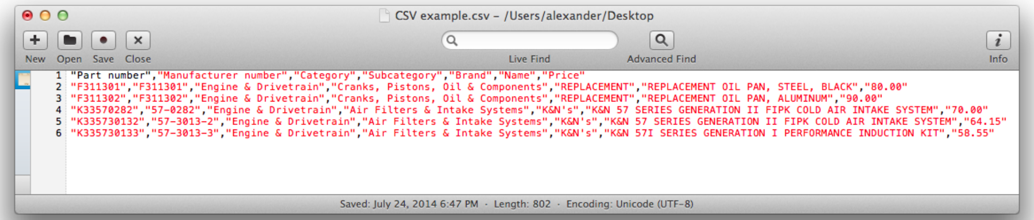
Part number	Manufacturer number	Category	Subcategory	Brand	Name	Price
F311301	F311301	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, STEEL, BLACK	80.00
F311302	F311302	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, ALUMINUM	90.00
K33570282	57-0282	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM	70.00
K335730132	57-3013-2	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM	64.15
K335730133	57-3013-3	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57I SERIES GENERATION I PERFORMANCE INDUCTION KIT	58.55

For tables with server-side processing enabled, PDF will contain only the visible columns.

3. CSV

“CSV” button generates a new file in “Comma-Separated-Values” format out of the table, and calls the browser’s “Save as …” dialog so you could choose a path for your file. All the values from your table, including the headers will be exported. The cells will be separated by commas, rows will be separated by line breaks:





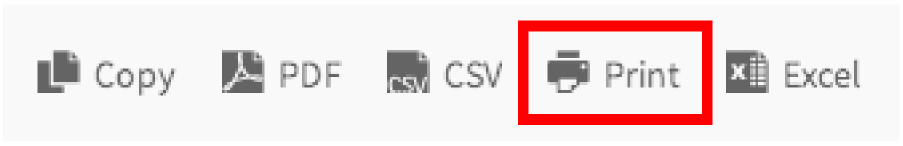
```
"Part number","Manufacturer number","Category","Subcategory","Brand","Name","Price"
"F311301","F311301","Engine & Drivetrain","Cranks, Pistons, Oil & Components","REPLACEMENT","REPLACEMENT OIL PAN, STEEL, BLACK","80.00"
"F311302","F311302","Engine & Drivetrain","Cranks, Pistons, Oil & Components","REPLACEMENT","REPLACEMENT OIL PAN, ALUMINUM","90.00"
"K33570282","57-0282","Engine & Drivetrain","Air Filters & Intake Systems","K&N's","K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM","70.00"
"K335730132","57-3013-2","Engine & Drivetrain","Air Filters & Intake Systems","K&N's","K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM","64.15"
"K335730133","57-3013-3","Engine & Drivetrain","Air Filters & Intake Systems","K&N's","K&N 57I SERIES GENERATION I PERFORMANCE INDUCTION KIT","58.55"
```

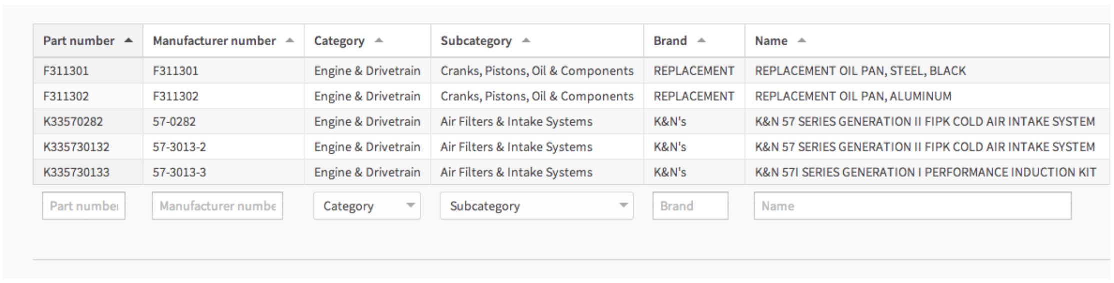
Saved: July 24, 2014 6:47 PM · Length: 802 · Encoding: Unicode (UTF-8)

For tables with server-side processing enabled, CSV file will contain only the visible columns.

4. Print

“Print” button uses jQuery functionality to hide all the elements on the page, except for the table. For some themes it can look weird, but for most of these it will work correctly, and make the view of the page “printable”:





Part number	Manufacturer number	Category	Subcategory	Brand	Name
F311301	F311301	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, STEEL, BLACK
F311302	F311302	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, ALUMINUM
K33570282	57-0282	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM
K335730132	57-3013-2	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM
K335730133	57-3013-3	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57I SERIES GENERATION I PERFORMANCE INDUCTION KIT
Part numbe	Manufacturer numbe	Category	Subcategory	Brand	Name

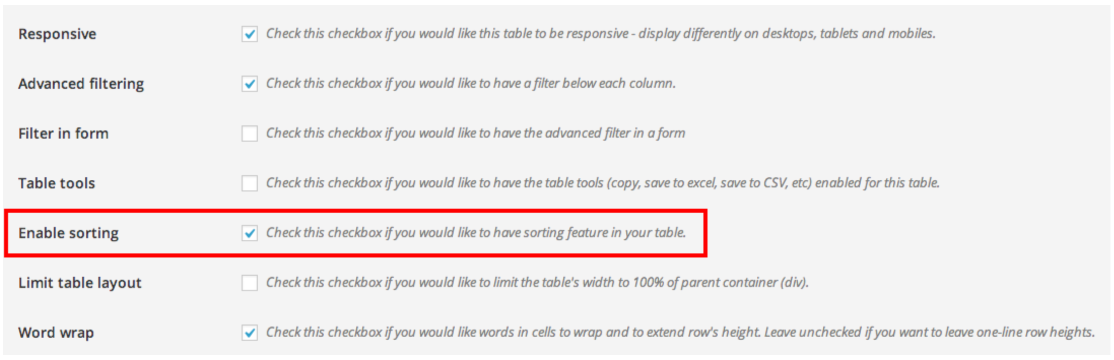
5. **Excel**

“Excel” button generates a new CSV file, but saves is with “XLS” extension, so that MS Excel would recognize it as an Excel file. See pt. 3 of this list for details.

5.5 Sorting

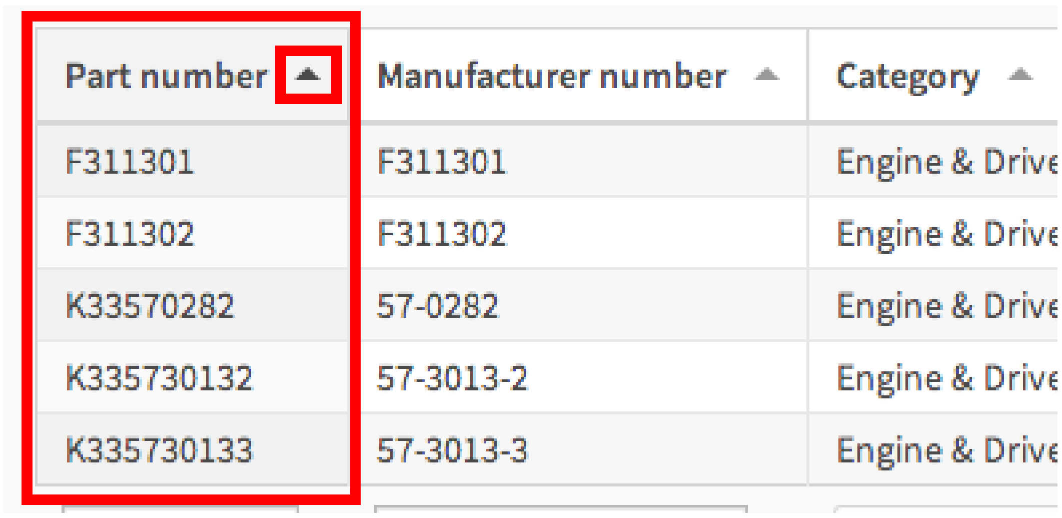
This section gives a detailed overview on the sorting table feature.

Tables rendered by the wpDataTables plugin are sortable by default; it means that the content of the table can be ordered by the values of one of the columns. The ordering rules depend on the column type: e.g. string columns will be ordered by alphabetical rules, integer or float by arithmetic rules, and so on. See chapter 6 of this documentation for detailed info on different column types, and sorting rules.
To enable sorting (or to disable, since it is enabled for all columns by default) you need to use the “**Enable sorting**” checkbox in the “Additional settings” section of the Step 1 in the table editing page.



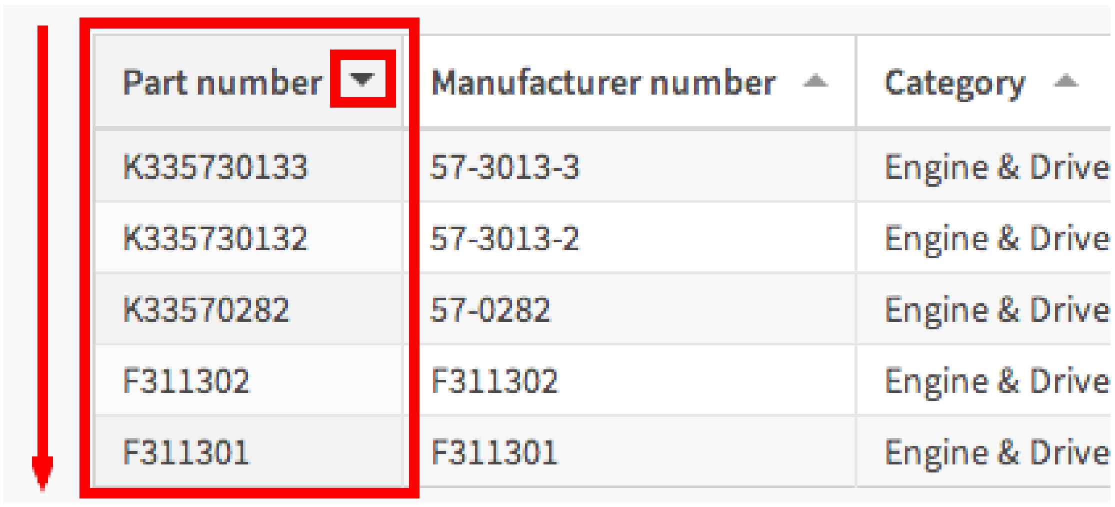
Responsive	✓	*Check this checkbox if you would like this table to be responsive - display differently on desktops, tablets and mobiles.*
Advanced filtering	✓	*Check this checkbox if you would like to have a filter below each column.*
Filter in form	☐	*Check this checkbox if you would like to have the advanced filter in a form*
Table tools	☐	*Check this checkbox if you would like to have the table tools (copy, save to excel, save to CSV, etc) enabled for this table.*
Enable sorting	✓	*Check this checkbox if you would like to have sorting feature in your table.*
Limit table layout	☐	*Check this checkbox if you would like to limit the table's width to 100% of parent container (div).*
Word wrap	✓	*Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave unchecked if you want to leave one-line row heights.*

When the sorting is enabled, each column header will get “interactive”: it will work as a button, and when you click it, you will sort the table by this column in ascending (increasing order), which will be indicated by the highlighting of the column (means that it is used for sorting), and by the triangle showing to the right of the header turning black. When triangle is showing "up" it means that the table is sorted in ascending order – from "smaller" to "bigger" value:



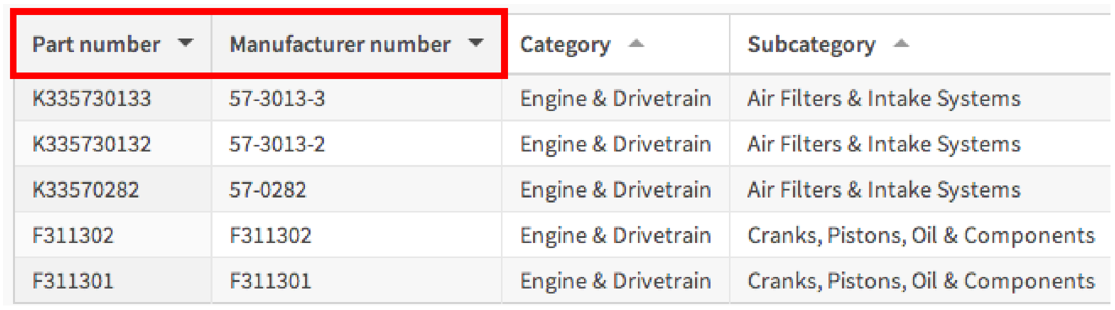
Part number ▲	Manufacturer number ▲	Category ▲
F311301	F311301	Engine & Drive
F311302	F311302	Engine & Drive
K33570282	57-0282	Engine & Drive
K335730132	57-3013-2	Engine & Drive
K335730133	57-3013-3	Engine & Drive

When you click the same header again, the triangle will turn "down", which means that the sorting is done in descending order from "bigger" to "smaller" value:



Part number ▼	Manufacturer number ▲	Category ▲
K335730133	57-3013-3	Engine & Drive
K335730132	57-3013-2	Engine & Drive
K33570282	57-0282	Engine & Drive
F311302	F311302	Engine & Drive
F311301	F311301	Engine & Drive

If you sort by one of the columns, then press “Shift” button on the keyboard, and, while holding it, click on another header, it will be used as a second sorting column (first the ordering will be done based on the first column, and then based on the second column):



Part number ▼	Manufacturer number ▼	Category ▲	Subcategory ▲
K335730133	57-3013-3	Engine & Drivetrain	Air Filters & Intake Systems
K335730132	57-3013-2	Engine & Drivetrain	Air Filters & Intake Systems
K33570282	57-0282	Engine & Drivetrain	Air Filters & Intake Systems
F311302	F311302	Engine & Drivetrain	Cranks, Pistons, Oil & Components
F311301	F311301	Engine & Drivetrain	Cranks, Pistons, Oil & Components

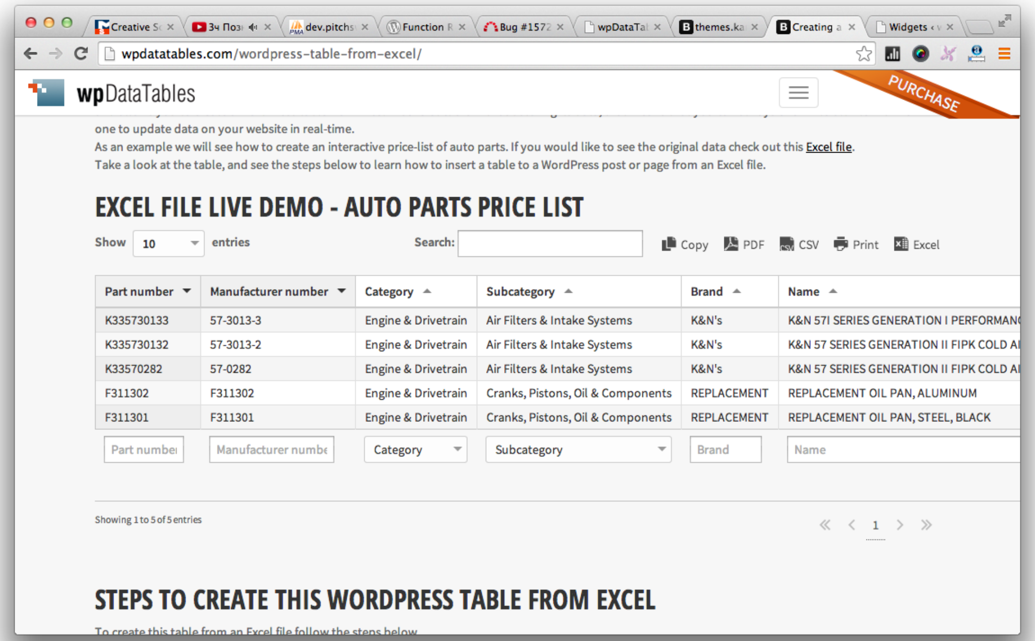
The secondary sorting means that when the values of the first columns are the same for a number of rows, you can also use some other column values to get the order that you want.
If the sorting is enabled, one of the columns will be used for ordering when you open the page (“default sorting column”). If you don’t provide it manually, the first column will be used for ordering; if you want to use a different one, you can define it in the step 2 of the table edit page: each column has an input “**Default sorting column**” with 2 radio buttons – “Ascending” and “Descending”. If you enable the radio button for one of the columns, this column will be used for ordering when front-end users first open the page.

5.6 Table layout

This section gives an explanation of the “limit table layout” feature.

Different tables require different layouts on the page: some tables have a lot of columns, and need more width, some tables have several columns, but long content in the cells, and it makes sense to wrap the words, but limit the table width to keep it same with the page.

By default the table isn't limited in the width, and the cells content is printed in one line. It makes sense when the table has a lot of columns, but even when the table has only several columns it can get wider then the screen, and the horizontal scrollbar appears:



one to update data on your website in real-time.
As an example we will see how to create an interactive price-list of auto parts. If you would like to see the original data check out this Excel file.
Take a look at the table, and see the steps below to learn how to insert a table to a WordPress post or page from an Excel file.

EXCEL FILE LIVE DEMO - AUTO PARTS PRICE LIST

Show 10 entries Search: Copy PDF CSV Print Excel

Part number	Manufacturer number	Category	Subcategory	Brand	Name
K335730133	57-3013-3	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57I SERIES GENERATION I PERFORMAN
K335730132	57-3013-2	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD A
K33570282	57-0282	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD A
F311302	F311302	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, ALUMINUM
F311301	F311301	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, STEEL, BLACK
Part numbe	Manufacturer numbe	Category	Subcategory	Brand	Name

Showing 1 to 5 of 5 entries

STEPS TO CREATE THIS WORDPRESS TABLE FROM EXCEL

To create this table from an Excel file follow the steps below

To limit the table width to the parent container width (so it wouldn't be wider then the page) you need to tick the checkbox "Limit table layout" in the "Additional settings" section of the step 1 on table edit page:

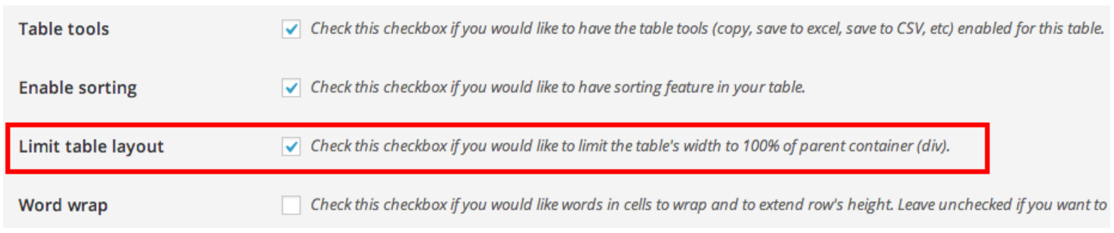
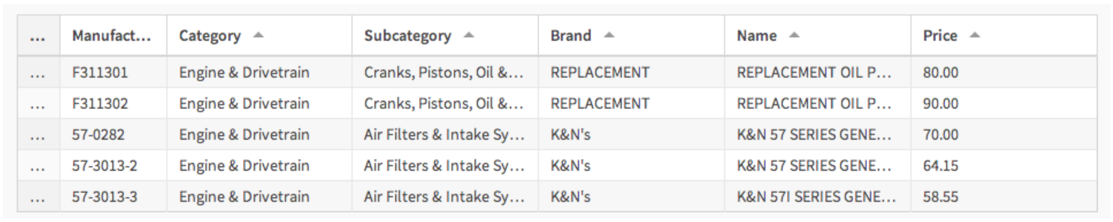


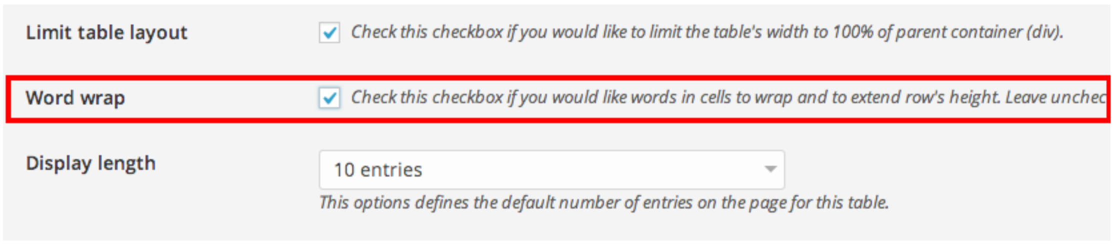
Table tools	✓ *Check this checkbox if you would like to have the table tools (copy, save to excel, save to CSV, etc) enabled for this table.*
Enable sorting	✓ *Check this checkbox if you would like to have sorting feature in your table.*
Limit table layout	✓ *Check this checkbox if you would like to limit the table's width to 100% of parent container (div).*
Word wrap	☐ *Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave unchecked if you want to*

When this is done, and the table is saved, it will be limited in width to fit the page:

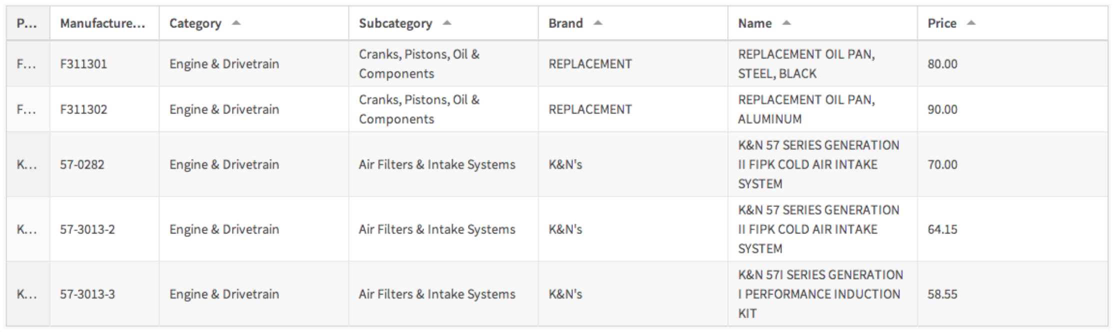


…	Manufact…	Category	Subcategory	Brand	Name	Price
…	F311301	Engine & Drivetrain	Cranks, Pistons, Oil &…	REPLACEMENT	REPLACEMENT OIL P…	80.00
…	F311302	Engine & Drivetrain	Cranks, Pistons, Oil &…	REPLACEMENT	REPLACEMENT OIL P…	90.00
…	57-0282	Engine & Drivetrain	Air Filters & Intake Sy…	K&N's	K&N 57 SERIES GENE…	70.00
…	57-3013-2	Engine & Drivetrain	Air Filters & Intake Sy…	K&N's	K&N 57 SERIES GENE…	64.15
…	57-3013-3	Engine & Drivetrain	Air Filters & Intake Sy…	K&N's	K&N 57I SERIES GENE…	58.55

This makes the table fit the screen, but now if the cell content is too wide, it gets cut with a "…" – to avoid this you need to tick the "Word Wrap" checkbox. This will make the cell content to break into several lines:

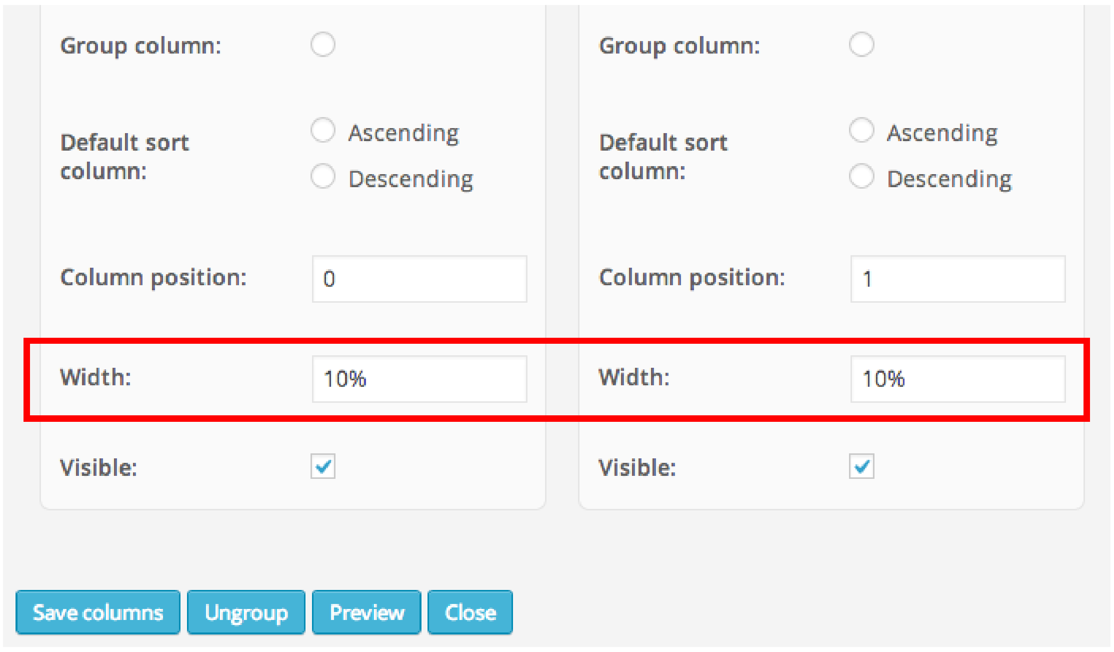


Limit table layout	✓ *Check this checkbox if you would like to limit the table's width to 100% of parent container (div).*
Word wrap	✓ *Check this checkbox if you would like words in cells to wrap and to extend row's height. Leave unchec*
Display length	10 entries *This options defines the default number of entries on the page for this table.*



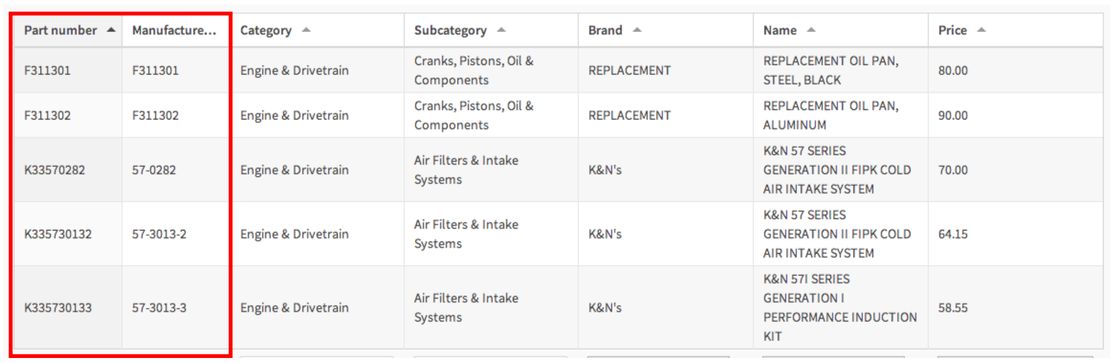
P…	Manufacture…	Category	Subcategory	Brand	Name	Price
F…	F311301	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, STEEL, BLACK	80.00
F…	F311302	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, ALUMINUM	90.00
K…	57-0282	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM	70.00
K…	57-3013-2	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM	64.15
K…	57-3013-3	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57I SERIES GENERATION I PERFORMANCE INDUCTION KIT	58.55

When the "limit table layout" option is enabled you can also define the columns widths (**please note that they will be ignored until this checkbox is ticked**). You can define them either in percent, or in pixels. E.g. if we set the widths for first two columns in this example table in percent:



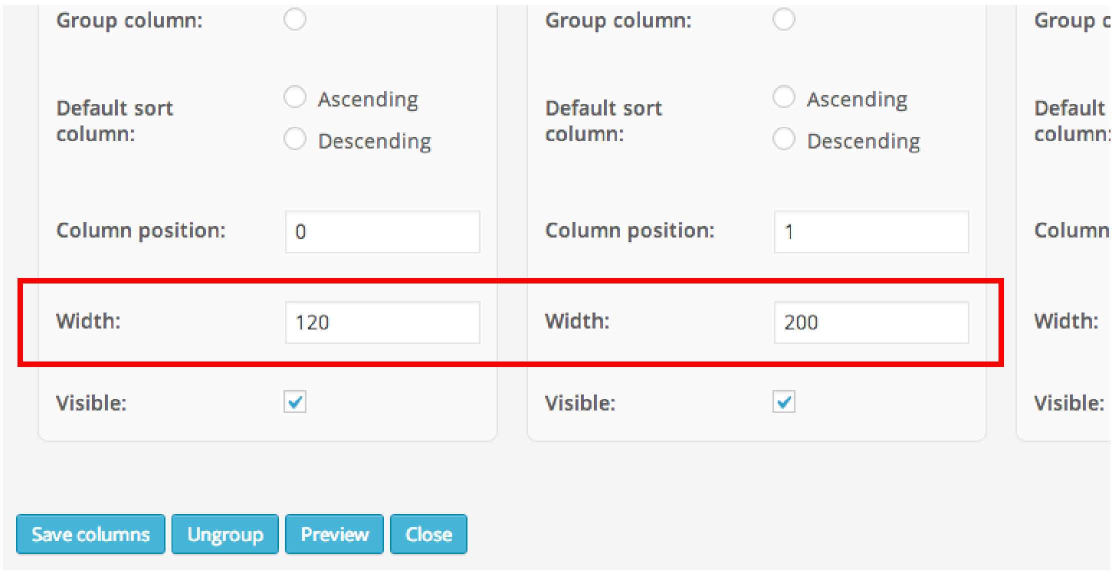
Group column:	○	Group column:	○
Default sort column:	○ Ascending ○ Descending	Default sort column:	○ Ascending ○ Descending
Column position:	0	Column position:	1
Width:	10%	Width:	10%
Visible:	✓	Visible:	✓

They get wider:

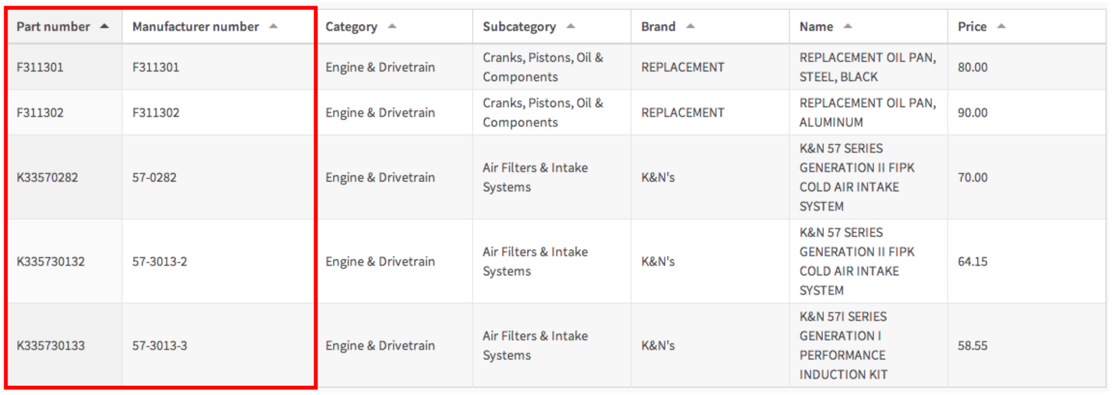


Part number	Manufacture...	Category	Subcategory	Brand	Name	Price
F311301	F311301	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, STEEL, BLACK	80.00
F311302	F311302	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, ALUMINUM	90.00
K33570282	57-0282	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM	70.00
K335730132	57-3013-2	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM	64.15
K335730133	57-3013-3	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57I SERIES GENERATION I PERFORMANCE INDUCTION KIT	58.55

Also we can do the same in pixels (without the "px"):



Group column:	○	Group column:	○	Group c
Default sort column:	○ Ascending ○ Descending	Default sort column:	○ Ascending ○ Descending	Default column:
Column position:	0	Column position:	1	Column
Width:	120	Width:	200	Width:
Visible:	✓	Visible:	✓	Visible:



Part number	Manufacturer number	Category	Subcategory	Brand	Name	Price
F311301	F311301	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, STEEL, BLACK	80.00
F311302	F311302	Engine & Drivetrain	Cranks, Pistons, Oil & Components	REPLACEMENT	REPLACEMENT OIL PAN, ALUMINUM	90.00
K33570282	57-0282	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM	70.00
K335730132	57-3013-2	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57 SERIES GENERATION II FIPK COLD AIR INTAKE SYSTEM	64.15
K335730133	57-3013-3	Engine & Drivetrain	Air Filters & Intake Systems	K&N's	K&N 57I SERIES GENERATION I PERFORMANCE INDUCTION KIT	58.55

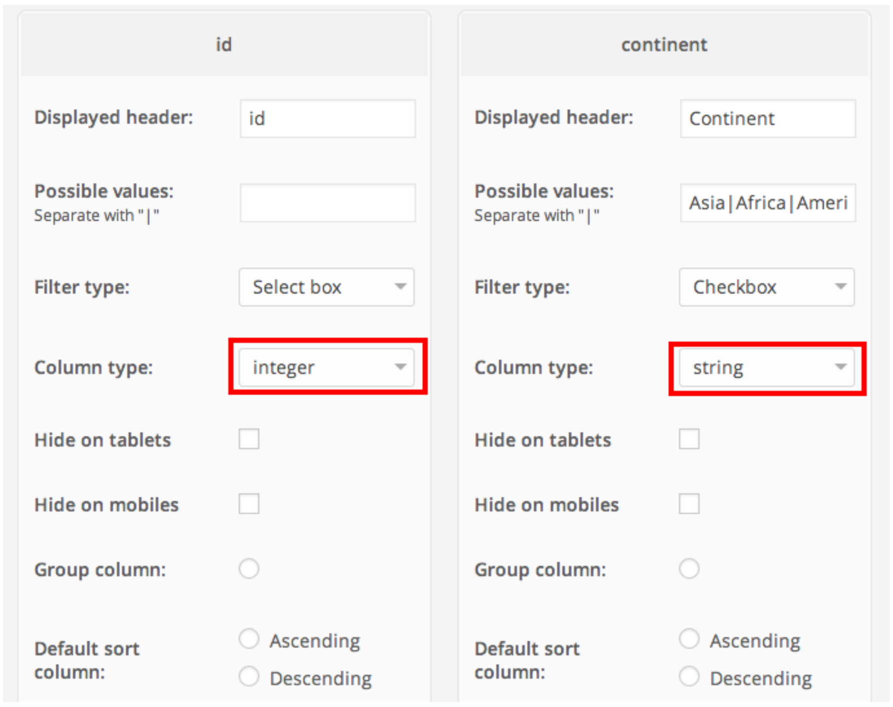
Notice for developers: main logic for this feature is in JS files: assets/js/jquery-datatables/TableTools.min.js and assets/js/wpdatatables/wpdatatables.js (minified version is used by default, so you would need to link the non-minified one in source/class.wpdatatable.php if you would like to change it).

6. Column types and column features

This chapter contains detailed explanations, descriptions and overview of different column types, and column features. The key differences between different column types are explained

In wpDataTables each column can have its own type: this is done because the sorting, or filtering rules are not the same for different data types. For example, sorting dates and sorting texts has completely different logic, this is why different column types are introduced.

wpDataTables has a built-in autodetect engine, so the column types are detected when the data source is first read, in most of the cases it is done correctly, but you can re-define the column type manually using the “Column type” selectbox, which appears for each column in the table edit page:



	id	continent
Displayed header:	id	Continent
Possible values: Separate with "|"		Asia|Africa|Ameri
Filter type:	Select box	Checkbox
Column type:	integer	string
Hide on tablets	☐	☐
Hide on mobiles	☐	☐
Group column:	○	○
Default sort column:	○ Ascending ○ Descending	○ Ascending ○ Descending

Please note the key difference between **Column type, Filter type and Editor input type** selectboxes:

- **Column type:** defines the rendering rules, and the sorting rules.
- **Filter type:** defines the look and behavior of the advanced filter for the column.

- **Editor input type:** defines the look and behavior of the element used in the editor popup (applicable only for editable MySQL tables).

Let's go through each column type and different column features in detail.

6.1 String (text) columns

This section gives an overview on the string (default text) columns.

String columns are the default and the most basic column types of wpDataTables. You can put any content in them, it will be rendered "as is", so if you e.g. want to put some specific HTML (buttons, iframes, or anything else) you can generate it in the data source, and it will be rendered on the front-end.

- **Rendering rules**: the only thing to note is that newline characters in the input data source will be translated to "<br/>" tag in the front-end.
- **Sorting rules:** alphabetical**.**
- **Applicable filter types:** text, selectbox, checkbox.
- **Issues to note:** if you use some HTML inside of string columns, sorting and filtering by this column can work weird (since it will apply the sorting rules also to HTML).

> **Notice for developers:** rendering rules for this column types are in the file: source/class.wpdatacolumn.string.php, sorting rules (for tables without server-side feature) are built-int to DataTables jQuery plugin.

6.2 Integer columns

This section gives an overview on the integer numeric columns.

Integer columns are used when you need to render numeric values. Even if in the input data source you have non-integer values for this column (not just numbers, or numbers with some decimal part), if you define a column as integer, it will cast and render all the values as integers – cut all non-numerical chars, and the decimal part.

- **Rendering rules**: none, value is casted as a number as rendered "as is".
- **Sorting rules:** arithmetical.
- **Applicable filter types:** text (will also work), number, number range, selectbox, checkbox.
- **Issues to note:** none.

> **Notice for developers:** rendering rules for this column types are in the file: source/class.wpdatacolumn.integer.php, sorting rules (for tables without server-side feature) are built-int to DataTables jQuery plugin.

6.3 Float columns

This section gives an overview on the floating numeric columns (for numbers with decimal part).

Float column type should be used when you need to render numeric values with a decimal part. Even non-float content type from the input will be casted as a number, and render with 2 numbers after the comma.

- **Rendering rules**: value is casted as a number, and rendered with 2 numbers after the comma.
- **Sorting rules:** arithmetical.
- **Applicable filter types:** text (will also work), number, number range, selectbox, checkbox.
- **Issues to note:** none.

> **Notice for developers:** rendering rules for this column types are in the file: source/class.wpdatacolumn.formatnum.php, sorting rules (for tables without server-side feature) are in assets/js/php-datatables/wpdatatables.funcs.min.js

6.4 Date columns

This section gives an overview on the date columns.

Date columns are used to render the date values in the format that you specify on the plugin settings page. If the content in the input data source cannot be casted as a date, PHP will reset the date to "01/01/1970" and this date will be rendered.

- **Rendering rules**: value is casted as a date, then formatted using the format defined in the plugin settings.
- **Sorting rules:** date.
- **Applicable filter types:** date.
- **Issues to note:**
  - When working with Excel files please set the format of Date columns as plain text; since PHPExcel library that parses Excel files does not parse the Excel date format correctly and this results in wrong rendering of the dates (usually they all get reset to 01/01/1970).
  - When you're working with MySQL please cast timestamps, and datetime columns to just dates to avoid problems.

> **Notice for developers:** rendering rules for this column types are in the file: source/class.wpdatacolumn.date.php, sorting rules (for tables without server-side feature) are in assets/js/php-datatables/ wpdatatables.funcs.min.js

6.5 URL link columns

This section gives an overview on the URL link columns.

URL link columns are used to render the hyperlinks which can be clicked to open pages within your website or on some external websites.

- **Rendering rules**: If the content of the cells contains a combination of "two sticks" - **||** - then everything to the left of this combination is used as an URL address for the link, and everything to the right as displayed text. I.e., this:

  http://wpdatatables.com||Check out wpDataTables

  will be rendered as this:

  Check out wpDataTables

  If the "||" combination is not found in the content, whole content will be used both for the link and as the display value. I.e., this:

  http://wpdatatables.com

  will be rendered as this:

  http://wpdatatables.com

- **Sorting rules:** alphabetical.
- **Applicable filter types:** text.
- **Issues to note:** sometimes some issues may appear when you're sorting on URL columns with complicated HTML structure.

> **Notice for developers:** rendering rules for this column types are in the file: source/class.wpdatacolumn.link.php, sorting rules (for tables without server-side feature) are built-in to DataTables jQuery library.

6.6 E-mail link columns

This section gives an overview on the E-mail link columns.

E-mail link columns are used to render the hyperlinks which can be clicked to open default mail application to write messages.

- **Rendering rules**: If the content of the cells contains a combination of "two sticks" - **||** - then everything to the left of this combination is used as an E-mail address for the link, and everything to the right as displayed text. I.e., this:

user@domain.com||Write me an e-mail

will be rendered as this:

Write me an e-mail

If the "||" combination is not found in the content, whole content will be used both for the link and as the display value. I.e., this:

user@domain.com

will be rendered as this:

user@domain.com

- **Sorting rules:** alphabetical.
- **Applicable filter types:** text.
- **Issues to note:** sometimes some issues may appear when you're sorting on E-mail columns with complicated HTML structure.

Notice for developers: rendering rules for this column types are in the file: source/class.wpdatacolumn.email.php, sorting rules (for tables without server-side feature) are built-in to DataTables jQuery library.

6.7 Image columns

This section gives an overview on the image columns.

Image link columns are used to render images in the table, either "full-size", or as a link from a thumbnail to the full size.

- **Rendering rules**: If the content of the cells contains a combination of "two sticks" - **||** - then everything to the left of this combination is used as a link to the full image size, and everything to the right as a link to the thumbnail:

http://dmn.com/fullsize.png|| http://dmn.com/thumb.png

If the "||" combination is not found in the content, whole content will be used as a link to a full-size image.

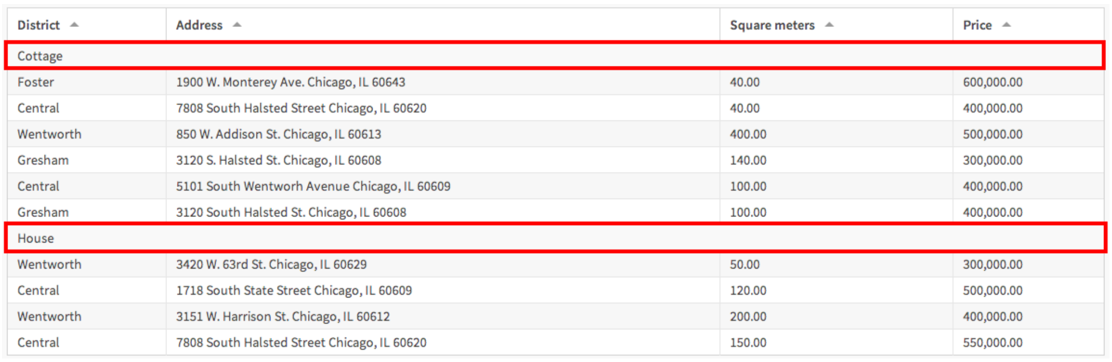
- **Sorting rules:** alphabetical.
- **Applicable filter types:** text.
- **Issues to note:**
  - Sometimes some issues may appear when you're sorting on image columns since it's done alphabetically and alphabetic sorting of HTML doesn't make much sense.
  - **Please note that the plugin does not create thumbnails** (at least for the moment), so you would need to take care of this yourself.
  - **If you would like to open full-size images in lightbox,** you can use some plugin for that, e.g. WP Lightbox 2

> **Notice for developers:** rendering rules for this column types are in the file: source/class.wpdatacolumn.image.php, sorting rules (for tables without server-side feature) are built-in to DataTables jQuery library.

6.8 Row grouping

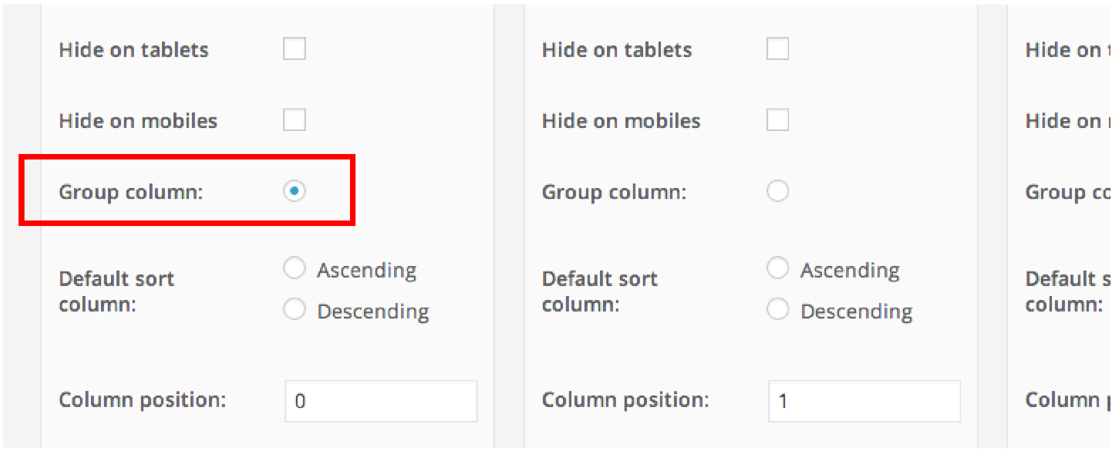
This section gives an overview on the row grouping feature.

If one column of your table contains similar values for many rows, maybe it makes sense to use it as a "group column": it means that it will not be rendered as a column, but its values will be used to group the rows; each group will be marked with one row with joined cells above, containing the group value:



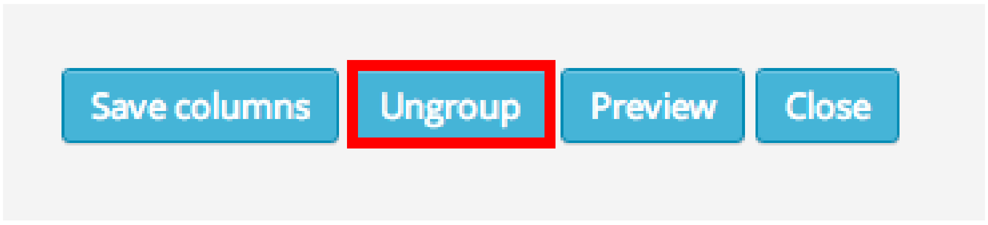
District	Address	Square meters	Price
Cottage			
Foster	1900 W. Monterey Ave. Chicago, IL 60643	40.00	600,000.00
Central	7808 South Halsted Street Chicago, IL 60620	40.00	400,000.00
Wentworth	850 W. Addison St. Chicago, IL 60613	400.00	500,000.00
Gresham	3120 S. Halsted St. Chicago, IL 60608	140.00	300,000.00
Central	5101 South Wentworh Avenue Chicago, IL 60609	100.00	400,000.00
Gresham	3120 South Halsted St. Chicago, IL 60608	100.00	400,000.00
House			
Wentworth	3420 W. 63rd St. Chicago, IL 60629	50.00	300,000.00
Central	1718 South State Street Chicago, IL 60609	120.00	500,000.00
Wentworth	3151 W. Harrison St. Chicago, IL 60612	200.00	400,000.00
Central	7808 South Halsted Street Chicago, IL 60620	150.00	550,000.00

To use this feature you need to tick the "Group column" radio button for one of the columns on the page edit page:



Hide on tablets ☐	Hide on tablets ☐	Hide on t
Hide on mobiles ☐	Hide on mobiles ☐	Hide on m
Group column: ●	Group column: ○	Group co
Default sort column: ○ Ascending ○ Descending	Default sort column: ○ Ascending ○ Descending	Default s column:
Column position: 0	Column position: 1	Column p

If you ticked this checkbox accidentally, you can disable it by pressing the "Ungroup" button in the bottom:

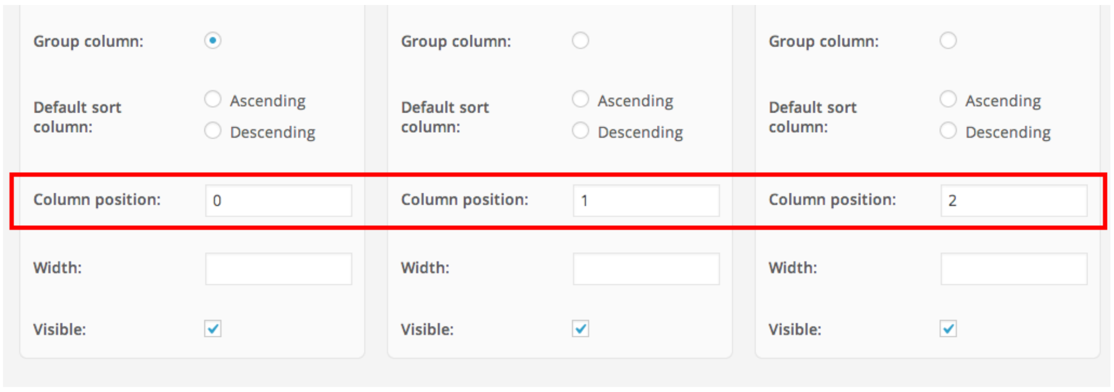


6.9 Column reordering

This section gives an overview on how you can change the order of columns.

If you would like to see the columns in the front-end in an order, different then the one you have in your data source, you can easily do it by just dragging and dropping the column blocks on the table edit page in the order that you want (don’t forget to click the “Save columns” button afterwards).

Other option, which is less convenient, is to define the column ordering manually in the “Column position” input fields:

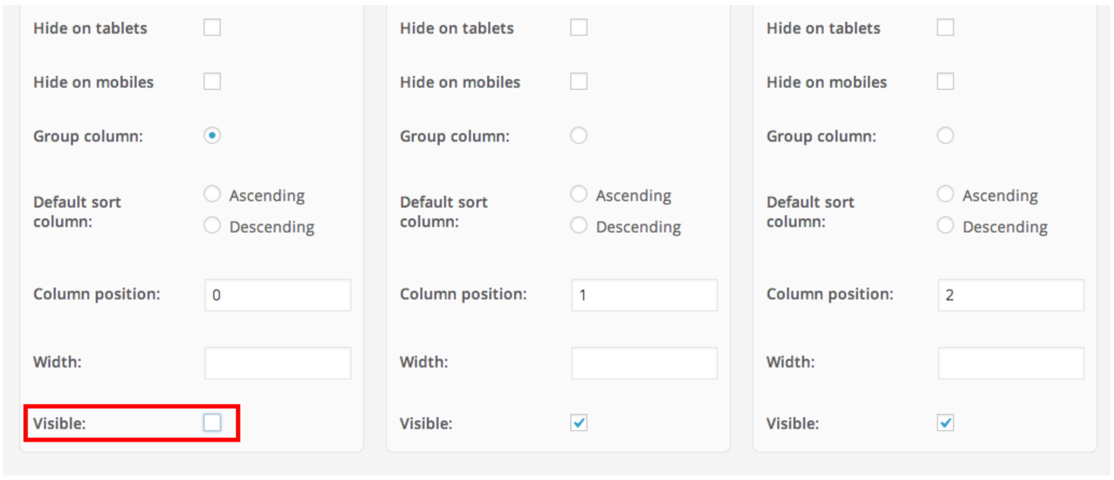


Group column:	●	Group column:	○	Group column:	○
Default sort column:	○ Ascending ○ Descending	Default sort column:	○ Ascending ○ Descending	Default sort column:	○ Ascending ○ Descending
Column position:	0	Column position:	1	Column position:	2
Width:		Width:		Width:	
Visible:	✓	Visible:	✓	Visible:	✓

6.10 Column visibility

This section gives an overview on how you can change the columns visibility.

If some of the columns that you have in the input data source shouldn’t be visible to the front-end users, you can easily hide them, just by unticking the “Visible” checkbox:



Hide on tablets	☐	Hide on tablets	☐	Hide on tablets	☐
Hide on mobiles	☐	Hide on mobiles	☐	Hide on mobiles	☐
Group column:	●	Group column:	○	Group column:	○
Default sort column:	○ Ascending ○ Descending	Default sort column:	○ Ascending ○ Descending	Default sort column:	○ Ascending ○ Descending
Column position:	0	Column position:	1	Column position:	2
Width:		Width:		Width:	
Visible:	☐	Visible:	✓	Visible:	✓

Please note several things about column visibility:

- **Invisible columns are still passed to the front-end**. It means that if there are a lot of invisible columns, they will load the page without need. So if you don't need the data in the front-end at all, it's better to delete these columns in the data source when possible.
- **Invisible columns can be used for filtering**. If the advanced filter is enabled, and is rendered in the form, or a widget, the filter can also use the data from invisible columns. Please see documentation section 5.3 for details.

7. Front-end editing

This chapter contains detailed tutorials on creating editable wpDataTables, different editor input types and limitations of this feauture.

In many cases you would like not only to display the table values to the front-end users of your site, but also allow them to edit these values: e.g. displaying and managing orders, or a bugtracker, or anything else. wpDataTables allows front-end editing for MySQL-based tables.

7.1 Creating editable tables.

This section gives an overview on how you can create editable wpDataTables.

To create an editable table you would need to follow these steps:

1. **Create a table on MySQL side**

First thing to do is to create the table on MySQL side. The easiest way to do this is to use some MySQL data manager like PHPMyAdmin, MySQL Workbench, SQLYog, or similar. Main requirement is that the table should have an ID field, which should be an autoincrement integer.
If you need an example for the CREATE SQL statement for a table, that can be editable, you can use this one:

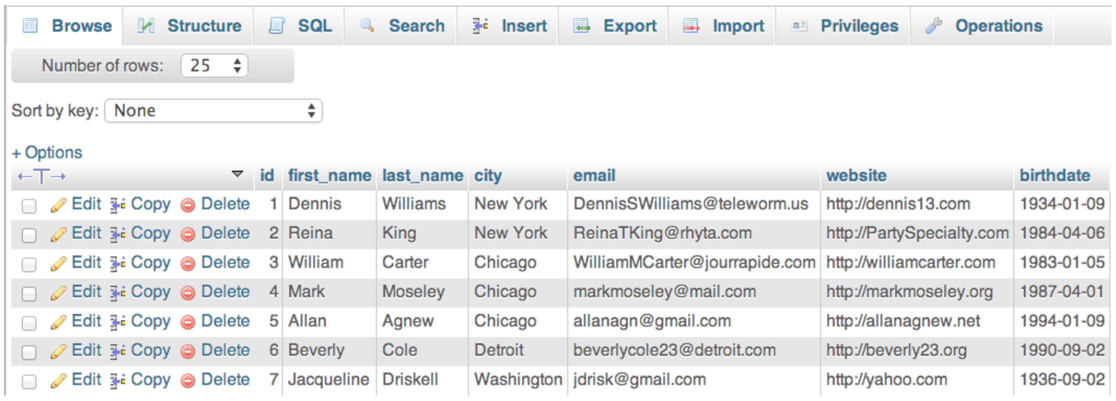
```
CREATE TABLE users_base (
  id int(11) NOT NULL auto_increment,
  first_name varchar(255) NOT NULL,
  last_name varchar(255) NOT NULL,
  city varchar(255) NOT NULL,
  email varchar(255) NOT NULL,
  website varchar(255) NOT NULL,
  birthdate date NOT NULL,
  PRIMARY KEY  (id)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8_general_ci;
```

Here you can see that the first field is an ID field, which is an integer auto increment. This is important, because this field will be used to identify the row from the front-end.

We are currently working on a Table Constructor extension for wpDataTables so this step won't be necessary in future, but it is for now.

2. Add at least a couple of rows to the table on the front-end

wpDataTables can generate tables from MySQL only when there is some data returned by the query. So you would need to pre-fill the data with some values, and add some rows to the dataset. The easiest way to do so is again to use some MySQL manager like PHPMyAdmin:



←T→ ▼	id	first_name	last_name	city	email	website	birthdate
Edit Copy Delete	1	Dennis	Williams	New York	DennisSWilliams@teleworm.us	http://dennis13.com	1934-01-09
Edit Copy Delete	2	Reina	King	New York	ReinaTKing@rhyta.com	http://PartySpecialty.com	1984-04-06
Edit Copy Delete	3	William	Carter	Chicago	WilliamMCarter@jourrapide.com	http://williamcarter.com	1983-01-05
Edit Copy Delete	4	Mark	Moseley	Chicago	markmoseley@mail.com	http://markmoseley.org	1987-04-01
Edit Copy Delete	5	Allan	Agnew	Chicago	allanagn@gmail.com	http://allanagnew.net	1994-01-09
Edit Copy Delete	6	Beverly	Cole	Detroit	beverlycole23@detroit.com	http://beverly23.org	1990-09-02
Edit Copy Delete	7	Jacqueline	Driskell	Washington	jdrisk@gmail.com	http://yahoo.com	1936-09-02

We are currently working on a Table Constructor extension for wpDataTables so this step won't be necessary in future, but it is for now.

3. Create a MySQL-based wpDataTable based on the table you just prepared

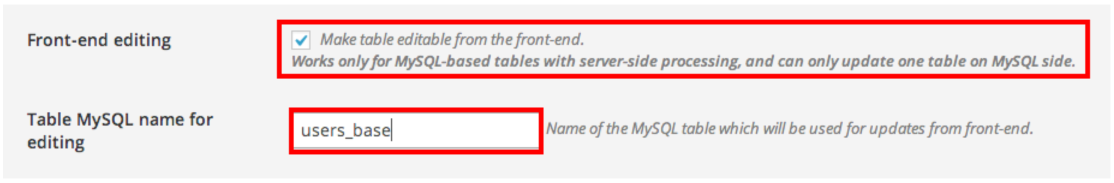
When the initial data is prepared you would need to create a wpDataTables with server-side processing enabled based on this table.

Please refer to documentation section 4.4 to see how to do this. For our example dataset the query would be simple:

```
SELECT * FROM users_base;
```

4. **Enable front-end editing feature and provide the MySQL table name**

When you configure all the settings which are common for "usual" MySQL-based wpDataTable you would need to tick the checkbox "Front-end editing" to enable the editing feature, and then provide the MySQL table name that should be used for editing:

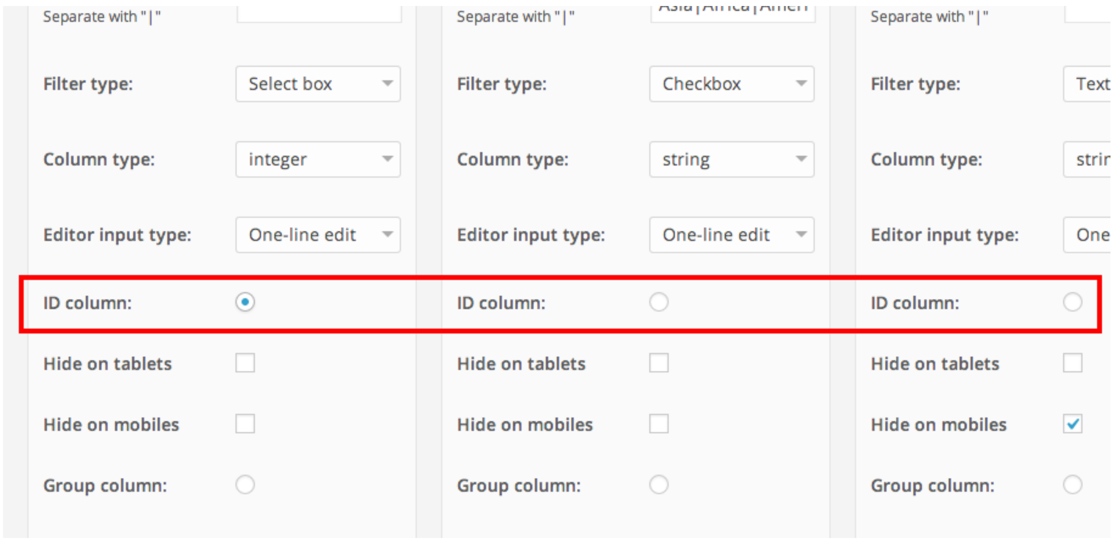


Front-end editing	✓ *Make table editable from the front-end.* *Works only for MySQL-based tables with server-side processing, and can only update one table on MySQL side.*
Table MySQL name for editing	users_base *Name of the MySQL table which will be used for updates from front-end.*

It is necessary to provide the name once again for a couple of reasons: first of all it is not always possible to parse it from the query; and secondly – sometimes you would like to send the changes to a different table rather from the one used in the query.
After you define these settings don't forget to press the "Save table" button.

5. **Set an ID column for front-end editing**

When you save the wpDataTable after configuring it as editable, each column block will get a radio "**ID column**". Please set it as enabled for the column that was defined as an autoincrement integer

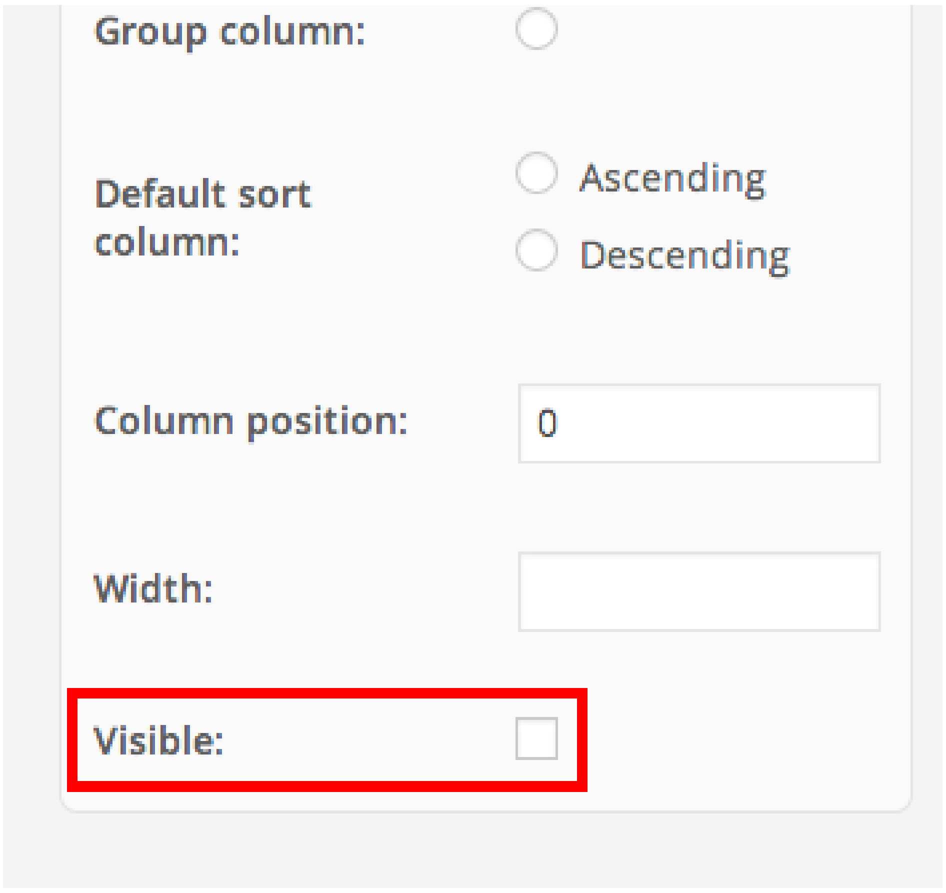


Separate with "|"		Separate with "|"	Asia | Africa | Americ	Separate with "|"	
Filter type:	Select box	Filter type:	Checkbox	Filter type:	Text
Column type:	integer	Column type:	string	Column type:	strir
Editor input type:	One-line edit	Editor input type:	One-line edit	Editor input type:	One
ID column:	●	ID column:	○	ID column:	○
Hide on tablets	☐	Hide on tablets	☐	Hide on tablets	☐
Hide on mobiles	☐	Hide on mobiles	☐	Hide on mobiles	✓
Group column:	○	Group column:	○	Group column:	○

This will tell the front-end editor to use this column as a unique identifier of the row on MySQL side.

Please note: If you use a column with non-unique values as an ID column, it might happen that when you save the settings on the front-end, not only one column will be edited, but multiple columns.

It makes sense to make the ID columns invisible, not to confuse the users since IDs are usually not human-readable:

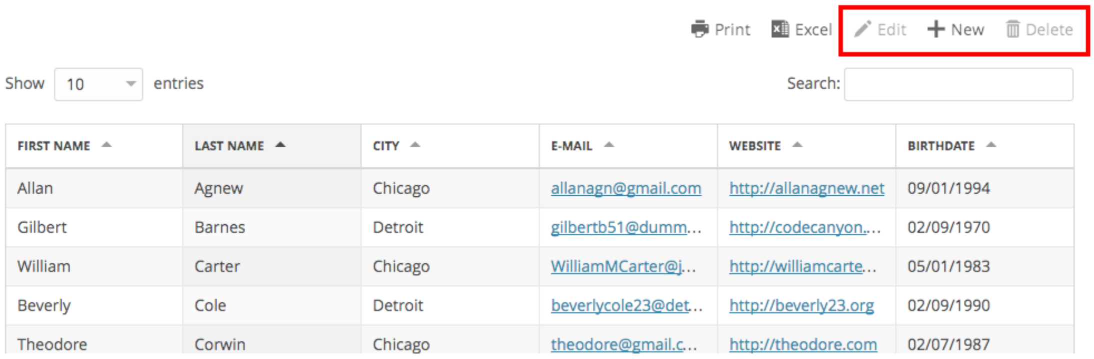


Group column:	○
Default sort column:	○ Ascending ○ Descending
Column position:	0
Width:	
Visible:	☐

ID will be still passed to the front-end, and used to identify the row, but will not appear in the table.

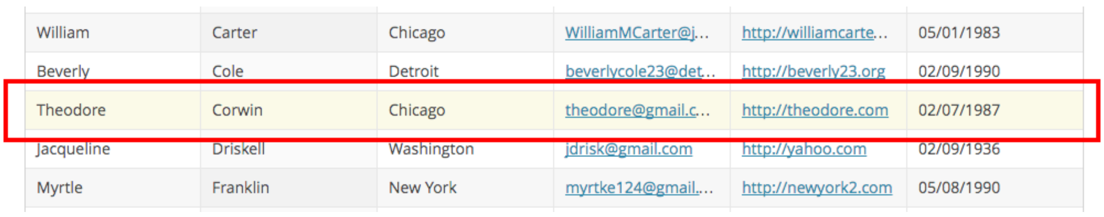
6. Save the columns and open the table in the front-end.

After you set up the ID column you can save the columns, paste the shortcode in some post or page, and then open it on your site's front-end. You will then see that there are three new buttons in the TableTools section of the table (or just three if you didn't enable TableTools for this wpDataTable):



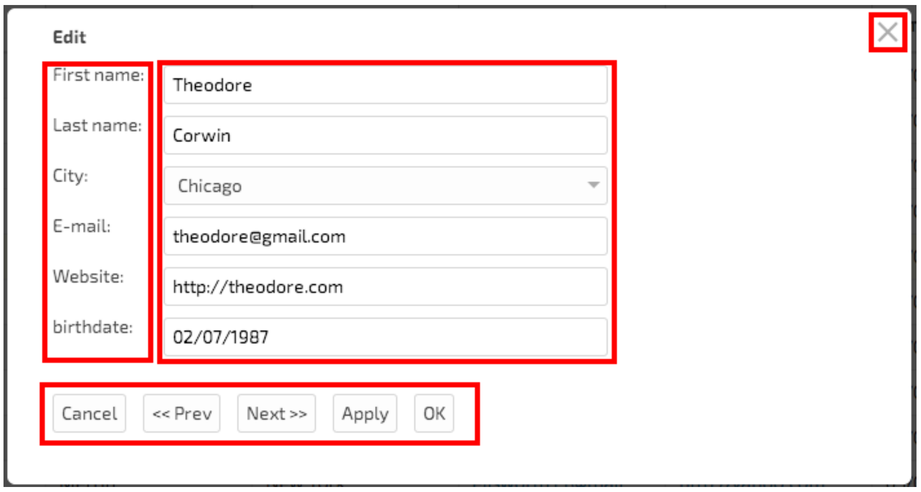
FIRST NAME	LAST NAME	CITY	E-MAIL	WEBSITE	BIRTHDATE
Allan	Agnew	Chicago	allanagn@gmail.com	http://allanagnew.net	09/01/1994
Gilbert	Barnes	Detroit	gilbertb51@dumm...	http://codecanyon....	02/09/1970
William	Carter	Chicago	WilliamMCarter@j...	http://williamcarte...	05/01/1983
Beverly	Cole	Detroit	beverlycole23@det...	http://beverly23.org	02/09/1990
Theodore	Corwin	Chicago	theodore@gmail.c...	http://theodore.com	02/07/1987

Also it becomes possible to select a row in the table by clicking it, it will become highlighted (you can change the highlight color along with other table colors in the table settings page):



William	Carter	Chicago	WilliamMCarter@j...	http://williamcarte...	05/01/1983
Beverly	Cole	Detroit	beverlycole23@det...	http://beverly23.org	02/09/1990
Theodore	Corwin	Chicago	theodore@gmail.c...	http://theodore.com	02/07/1987
Jacqueline	Driskell	Washington	jdrisk@gmail.com	http://yahoo.com	02/09/1936
Myrtle	Franklin	New York	myrtke124@gmail....	http://newyork2.com	05/08/1990

When you click on the “Add new”, or when you select one of the rows, you will see the front-end editor popup. The popup is responsive, so it will work correctly on mobile and tablet devices as well:



Edit

First name:	Theodore
Last name:	Corwin
City:	Chicago
E-mail:	theodore@gmail.com
Website:	http://theodore.com
birthdate:	02/07/1987

- In the left side of the editor popup you will see the names of the columns;
- On the right side of the editor popup you sill see the editor inputs for the cells;
- On the bottom side you will see the control and navigation buttons:
    - **Cancel** – discards all changes and closes the editor popup;
    - **<< Prev** – selects the previous row of the table (if not the first one is selected), if necessary switches to the previous page, and puts the data from this row in the front-end editor. Changes will be discarded, if they weren't saved.
    - **Next >>** - selects the next row of the table (if not the last one is selected), if necessary switches the table to the next page, and puts the data from this row to the front-end editor inputs. Changes will be discarded, if they weren't saved.
    - **Apply** – saves the data from the editor in the current row, but doesn't close the popup.
    - **OK** – saves the data from the editor in the current row, and closes the editor popup.
- **Right top "X" button** is equal to "Cancel" – discards all unsaved changes and closes the popup.

The input types in the editor and the validation rules will be applied accordingly to the input type specified in the column settings. See documentation section 7.2 for details on all possible editor input types.

7.2 Limitations of front-end editing

This section gives an overview on the editing limitations and things that front-end editing doesn't support.

Since wpDataTables is not exactly a data manager, its front-end editing feature has certain limitations. Please keep them in mind:

- **Only one MySQL table can be edited at a time**. Queries from multiple tables with joins cannot be used for editable feature, since SQL UPDATE and INSERT statements are generated automatically, and for now there is no way to update multiple tables.
- **Front-end editor will not work correctly with row grouping** or other table front-end features.
- **Server-side processing MUST be turned on for front-end editing** - it will turn on automatically if you forget to enable it.
- **Only MySQL tables can be edited** – I know this is kind of obvious, but after a number of requests I had to put this here as well.

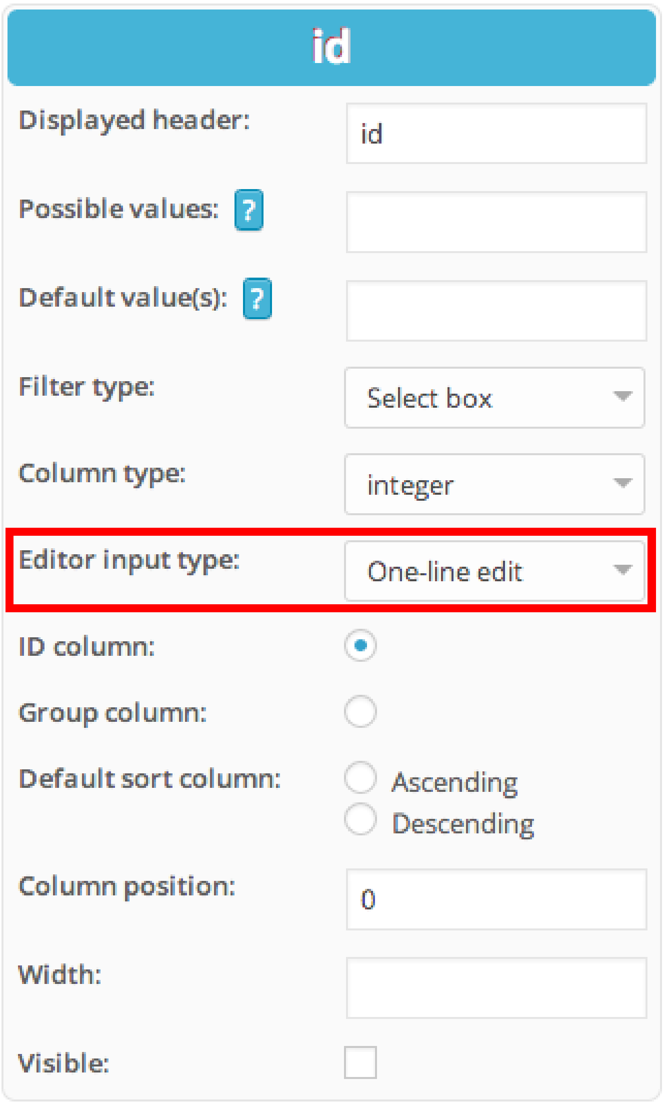
Also all limitations that apply to tables with server-side feature enabled are applicable here. Let's repeat these once again:

- **Charts will not work correctly with tables that have server-side processing enabled.**
- **Please do not use spaces in the column names.** E.g. use '**column1**' instead of '**column 1**'.
- **Please do not put a semicolon in the end of the query.**
- **Please do not use column names that can coincide with reserved MySQL words** (like DATE, BY, ORDER, GROUP, ...).
- **Please do not use numeric column names** like 1, 22, etc.
- **Please do not use "ORDER BY" in the statement.**

7.3 Editor input types

This section gives a detailed overview on each of the editor input types, use cases.

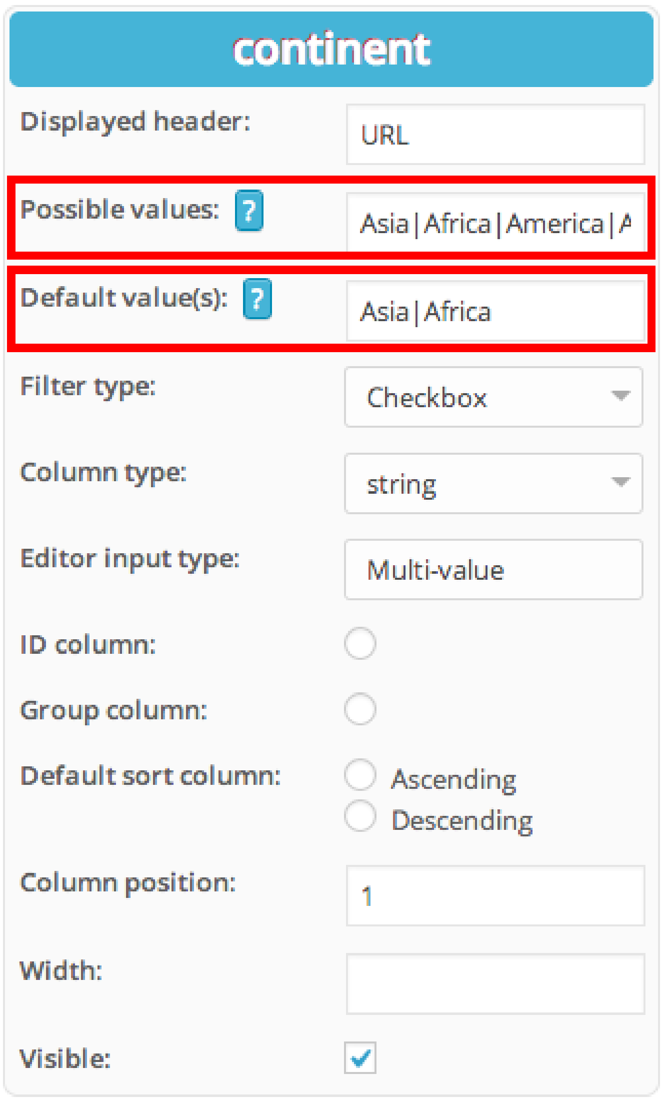
Same as for the advanced filtering, you can define the look of the input elements for each column. This setting will appear in the column block when you set the wpDataTable as editable on the table edit page:



id

Displayed header:	id
Possible values: ?	
Default value(s): ?	
Filter type:	Select box
Column type:	integer
Editor input type:	One-line edit
ID column:	●
Group column:	○
Default sort column:	○ Ascending ○ Descending
Column position:	0
Width:	
Visible:	☐

If you want to have some predefined value for the input you can use the “Default value(s)” input for this. For multi-value selectboxes you can define several default values, separating them with a “|”:

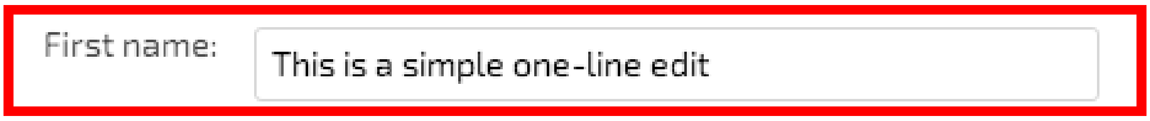

continent

Displayed header:	URL
Possible values:	Asia | Africa | America | A
Default value(s):	Asia | Africa
Filter type:	Checkbox
Column type:	string
Editor input type:	Multi-value
ID column:	○
Group column:	○
Default sort column:	○ Ascending ○ Descending
Column position:	1
Width:	
Visible:	✓

Let’s go through each editor input type in detail.

1. One-line edit

This editor type will render a simple one-line text input. Users can provide anything they need in this field:


First name: This is a simple one-line edit

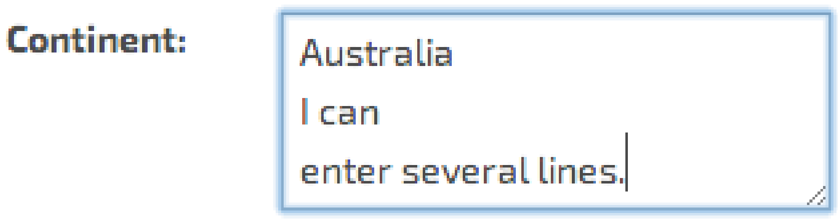
Applicable column types: on wpDataTables side any (but usually string), on MySQL side VARCHAR, or TEXT.

Issues to note: the only thing to note here is that if you use this edit on some MySQL column which is not VARCHAR or TEXT, and users type in some text, it will not be saved.

Validation rules: none.

2. Multi-line edit

This editor type will render a textarea, where users can type in long texts, and do line breaks:


Continent:

Australia
I can
enter several lines.

When these cells will be rendered in the front-end they will also have these line breaks that user typed in.

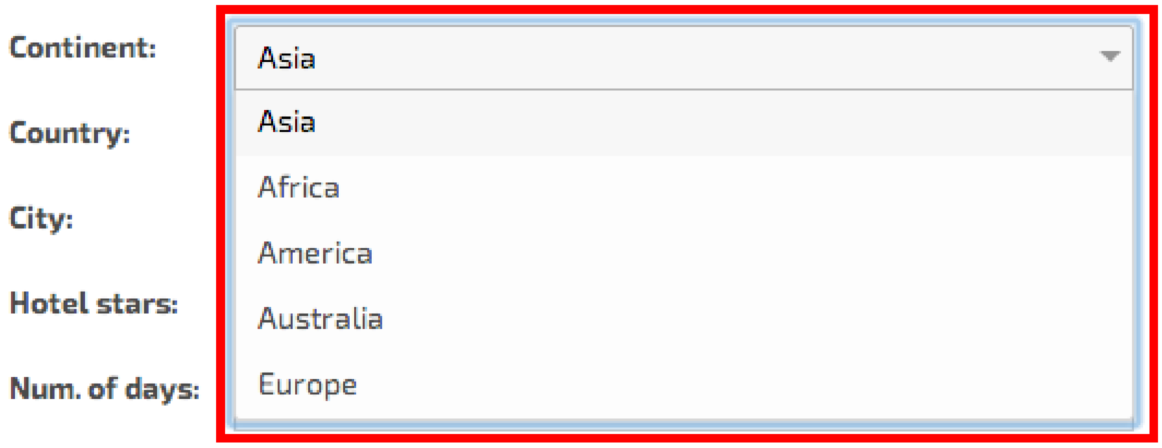
Applicable column types: on wpDataTables side any (but usually string), on MySQL side VARCHAR, or TEXT.

Issues to note: the only thing to note here is that if you use this edit on some MySQL column which is not VARCHAR or TEXT, and users type in some text, it will not be saved.

Validation rules: none.

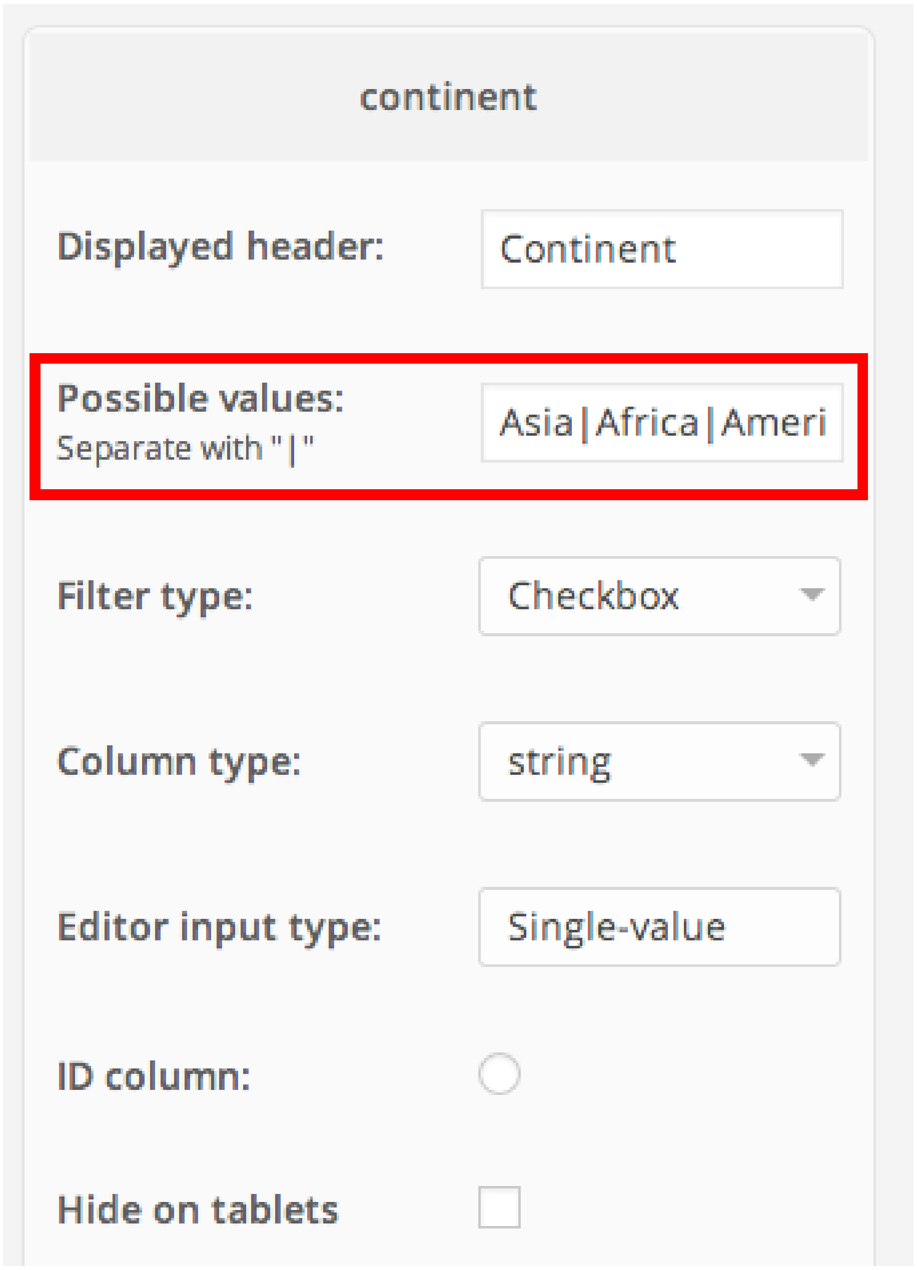
3. Single-value selectbox

This editor input type will render a dropdown selectbox, where front-end users could select one value out of several possible:



Applicable column types: on wpDataTables side any (but usually string), on MySQL side VARCHAR, TEXT, ENUM. Also can be used for numerical values.

Issues to note: In order for this input to work you must define the possible values of the column in the “Possible values” input of the column settings block on the table edit page, separated with a “|”:

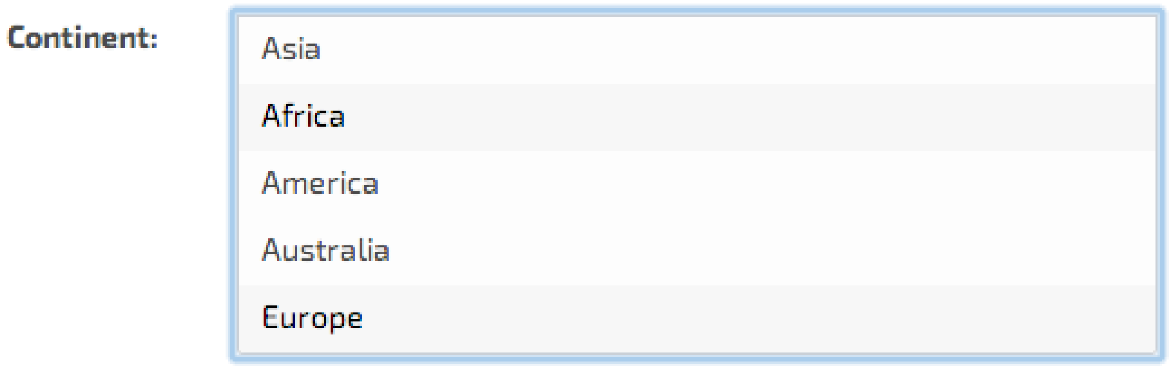

continent

Displayed header:	Continent
Possible values: Separate with "|"	Asia|Africa|Ameri
Filter type:	Checkbox
Column type:	string
Editor input type:	Single-value
ID column:	○
Hide on tablets	☐

Validation rules: none.

4. **Multiple-value selectbox**

This editor input type will render a selectbox, where front-end users could select several values out the defined:



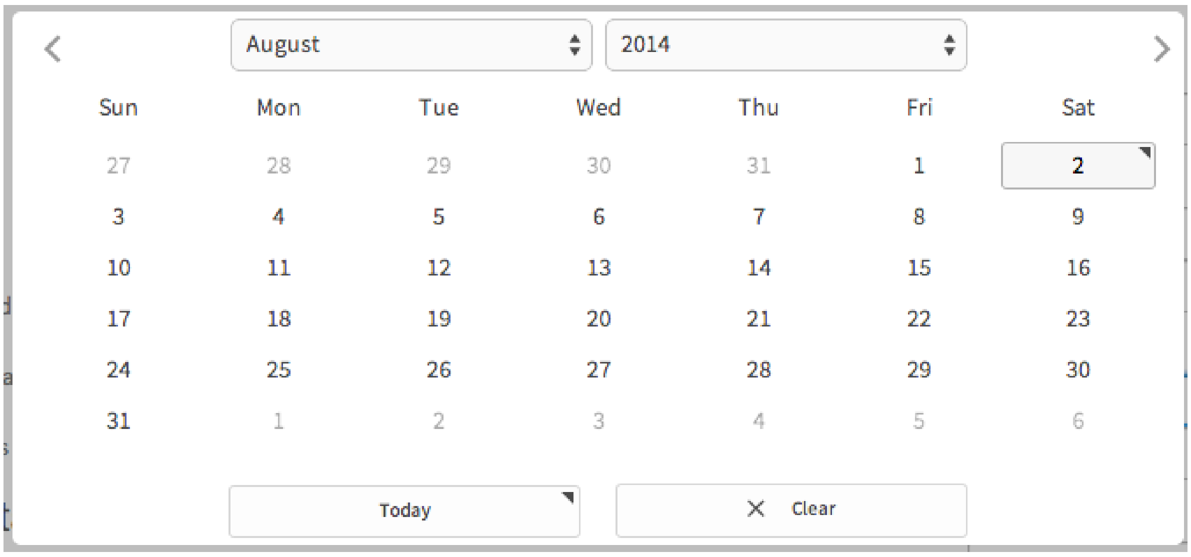
Applicable column types: on wpDataTables side any (but usually string), on MySQL side VARCHAR, TEXT, ENUM. Also can be used for numerical values.

Issues to note: In order for this input to work you must define the possible values of the column in the “Possible values” input of the column settings block on the table edit page, separated with a “|”.

Validation rules: none.

5. Date

This editor input type will render an input, which will show datepicker щт click. In this datepicker front-end users can define a date, which will then be pasted in a format that you provided on the settings page:



The datepicker popup is responsive, which means that it will be looking fine also on mobiles and tablets.

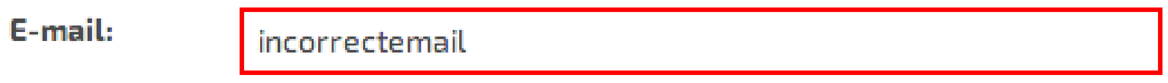
Applicable column types: on wpDataTables side - date, on MySQL side DATE, TIMESTAMP. Also can be used for numerical values.

Issues to note: On some configurations you'd need to play around with column format on MySQL side. If you find some problems with some of the date formats please let us know.

Validation rules: none.

6. E-mail link

This editor type will render a simple one-line text input, which will be validated as an e-mail when the user tries to save the entry. If not a valid e-mail is specified it will highlight the input field as incorrect, and the entry will not be saved.



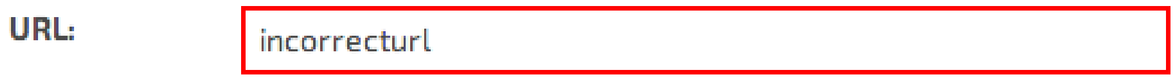
Applicable column types: on wpDataTables side any (but usually string), on MySQL side VARCHAR, or TEXT.

Issues to note: the only thing to note here is that if you use this edit on some MySQL column which is not VARCHAR or TEXT, and users type in an e-mail, it will not be saved.

Validation rules: the entered data must be a valid e-mail.

7. URL link

This editor type will render a simple one-line text input, which will be validated as an URL link when the user tries to save the entry. If not a valid URL is specified it will highlight the input field as incorrect, and the entry will not be saved.



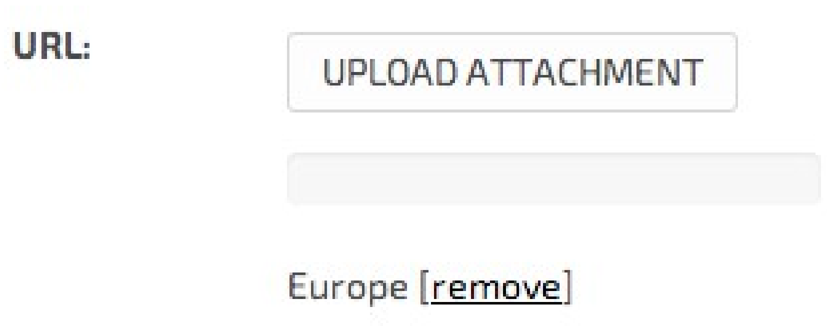
Applicable column types: on wpDataTables side any (but usually string), on MySQL side VARCHAR, or TEXT.

Issues to note: the only thing to note here is that if you use this edit on some MySQL column which is not VARCHAR or TEXT, and users type in an e-mail, it will not be saved.

Validation rules: the entered data must be a valid URL.

8. Attachment

This editor type will render a file manager block: “Upload attachment” button, an upload progressbar, and (if the file was already uploaded) a remove link:



Using this block your users will be able to upload an attachment, which will be then saved in the default “wp-uploads” WordPress upload folder (usually it’s also sorted by years and months inside). wpDataTables will store the link to the file in the MySQL table. Also progressbar will allow the users to see the upload progress, and if the file is already existing, users will be able to remove it using the “remove” link.

The column of the table will contain the download links to attachments.

Applicable column types: on wpDataTables side – URL link, on MySQL side VARCHAR, or TEXT.

Issues to note:

- Column type for this column **must** be “URL link” or it won’t work.
- Please make sure that wp-uploads folder allows writing.
- Files should not be bigger then the server allows (UPLOAD_MAX_FILESIZE)

Validation rules: none.

Notice for developers: PHP logic for editing the tables is mostly in the file: wpdatatables.php, methods: **wdt_save_table_frontend()** -adding and editing values, **wdt_upload_file() –** processing attachment uploads, **wdt_delete_table_row()** – deleting rows, **wdt_delete_uploaded_file()** – deleting attachments. JS logic is in the file assets/js/wpdatatables/wpdatatables.js, template of the editor is in the file templates/plain_html_table.inc.php

8. Charts

This chapter contains detailed tutorials on using charts in wpDataTables, different chart types, adding series to the charts, and the limitations of this feature.

One of the side features of wpDataTables is the chart functionality. Charts are rendered by Google Chart Tools, and you can use any data source to render them, but please keep in mind that this is only a side feature, and only some basic charts can be rendered; for something complicated you would need to implement some custom solution.

8.1 Using charts.

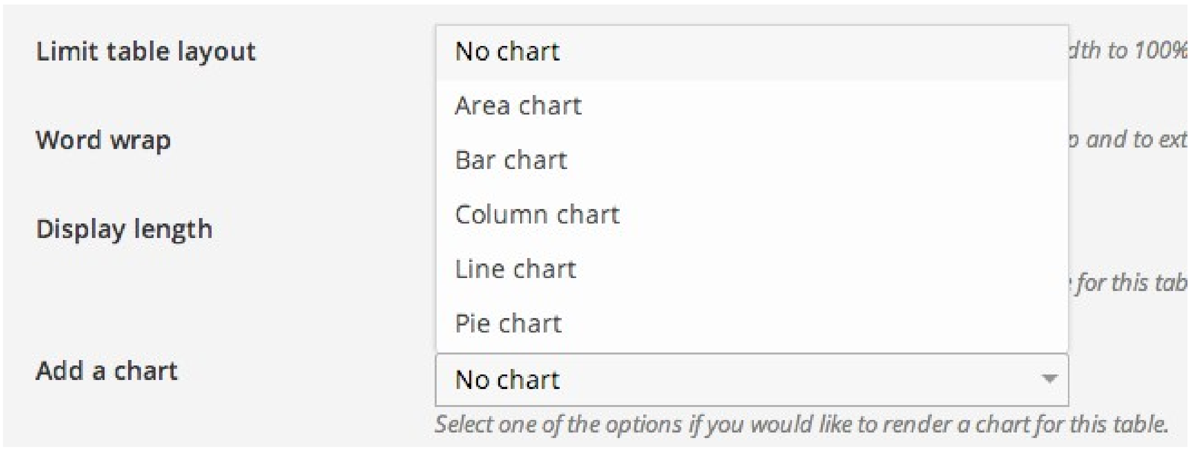
This section gives a dfetailed tutorial on using charts with wpDataTables.

One of the side features of wpDataTables is the chart functionality. Charts are rendered by Google Chart Tools, and you can use any data source to render them, but please keep in mind that this is only a side feature, and only some basic charts can be rendered; for something complicated you would need to implement some custom solution.

To render a chart you would need to follow these steps:

1. **First you need to build a wpDataTable** based on any of the input data sources. See documentation chapter 4 for detailed tutorials for all data sources.

2. **Select a chart type** in the Additional settings section of step 1 on the table edit page:


Limit table layout

Word wrap

Display length

Add a chart: No chart

Select one of the options if you would like to render a chart for this table.

Please see documentation section 8.2 for a detailed explanation of each chart type.

3. **Optionally define a chart title, which will appear above the chart:**

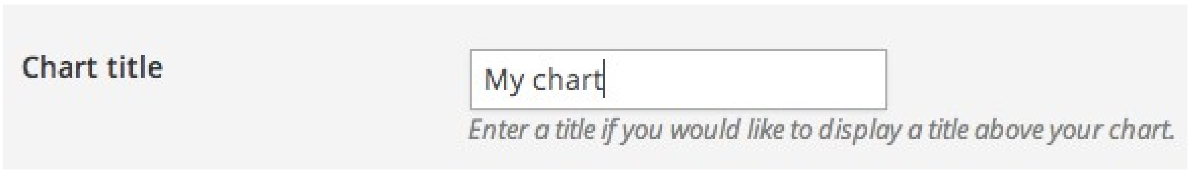
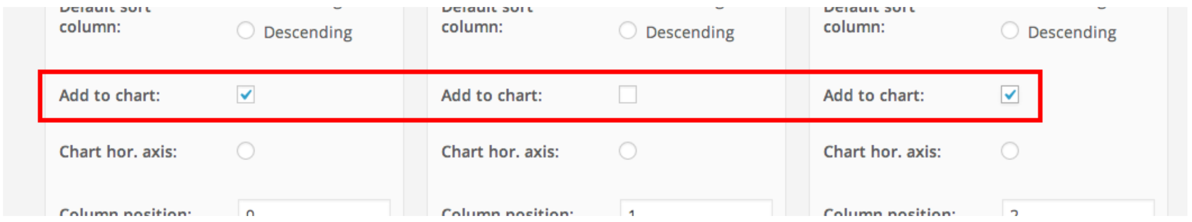


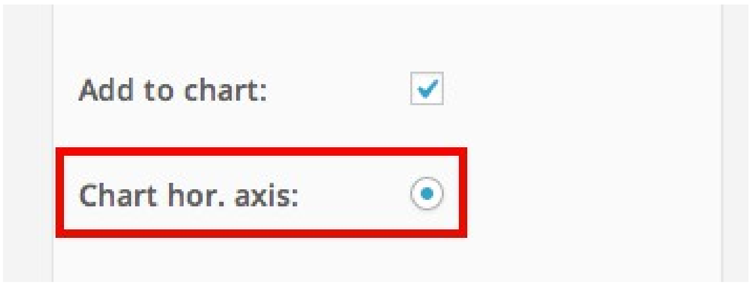
Chart title: My chart

Enter a title if you would like to display a title above your chart.

4. **Choose the columns that you would like to use in the chart.** Each selected column will be used as chart series:


Default sort column: Descending	Default sort column: Descending	Default sort column: Descending
Add to chart: ✓	Add to chart: ☐	Add to chart: ✓
Chart hor. axis: ○	Chart hor. axis: ○	Chart hor. axis: ○
Column position: 0	Column position: 1	Column position: 2

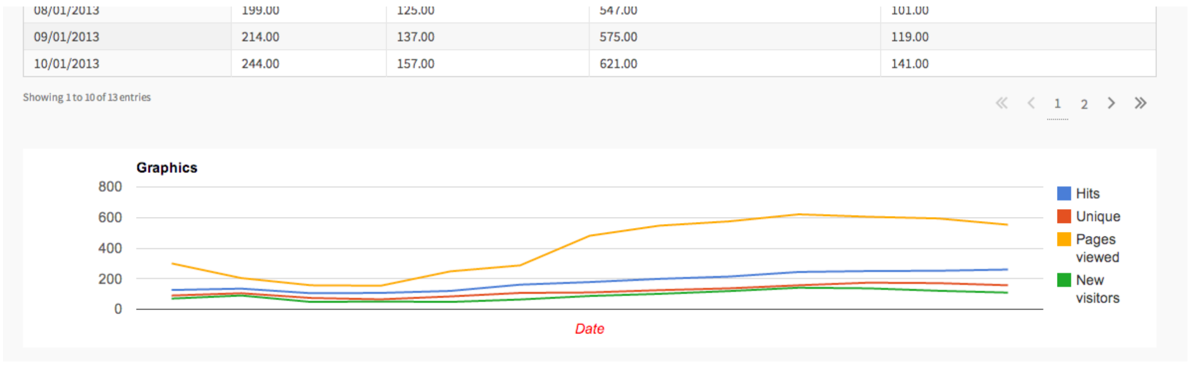
5. **Define the column which values will be used as horizontal axis labels.** Use the radio button in one of the column settings block.



Add to chart: [✓]

Chart hor. axis: ●

6. **Save and publish the table.** After these settings are done you can save the table, the columns, and generate the shortcode. Then you can publish it in a post or page, and see the rendered chart:



08/01/2013	199.00	125.00	547.00	101.00
09/01/2013	214.00	137.00	575.00	119.00
10/01/2013	244.00	157.00	621.00	141.00

Showing 1 to 10 of 13 entries

If you would like to render only the chart on the page, without the table, you can put a "**show_only_chart=true**" block in the table shortcode, like:

[wpdatatable id=1 show_only_chart=true]

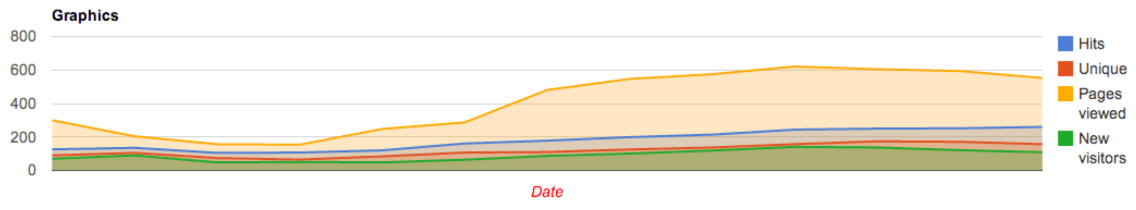
Then only the chart will be rendered on the page.

8.2 Chart types

This section gives an overview on different chart types that wpDataTables support.

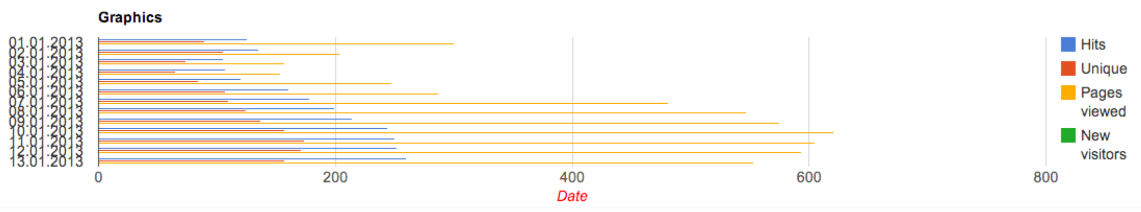
1. **Area chart.**

Area chart renders each series as a colored area which is outlined by the column values graph:



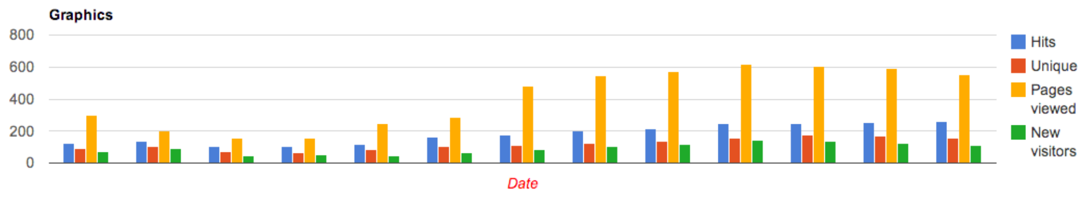
2. **Bar chart.**

Bar chart renders each series as a number or bars representing the column values which are stacked from the left. In this chart type the horizontal axis is rendered as vertical axis and the opposite:



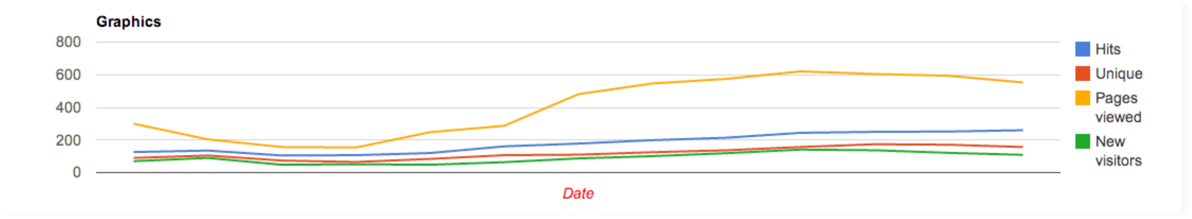
3. **Column chart.**

Column chart renders each series as a number or columns, which represent the values from the table:



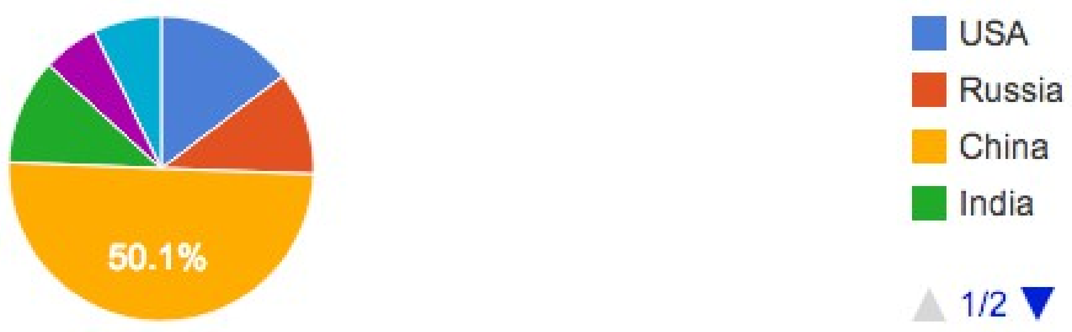
4. Line chart.

Line chart renders each series as a line which connects the dots, representing table values:



5. Pie chart.

Pie chart renders each series as a “pie slice”:



8.3 Chart limitations

This section covers the limitations that the chart feature has.

There are certain limitations for the chart functionality. Mostly these limitations are because of the fact that the chart is using the same data source as the table.

1. **Charts will not work with server-side processing feature.** Unfortunately only the data of the first several rows will be represented in the chart.
2. **Data in the charts will not be filtered when you filter the table.**
3. **Multiple charts per one table are not supported.**

> **Notice for developers:** PHP logic for editing the tables is in the file: source/class.wpdatatable.php, methods: **addChartSeries(), setChartType(), getChartType(), setChartTitle(), getChartTitle(), setChartVerticalAxis(), setChartHorizontalAxis(), renderChart(), printChart()**. JS logic is using Google hosted JS files, template of the charts is in the file templates/chart_js_template.inc.php

9. How to ...

This chapter contains answers to most commonly asked questions.

9.1 How to hide "Show ... entries" block?

The easiest way to do this is to put this CSS block somewhere in your theme's CSS file, or in inline CSS:

```
div.dataTables_length { display: none !important; }
```

9.2 How to hide "Showing ... of ..." block?

The easiest way to do this is to put this CSS block somewhere in your theme's CSS file, or in inline CSS:

```
div.dataTables_info { display: none !important; }
```

9.3 How to hide the global search block?

The easiest way to do this is to put this CSS block somewhere in your theme's CSS file, or in inline CSS:

```
div.dataTables_filter { display: none !important; }
```

9.4 How to hide one of the Table Tools buttons?

For this you would need to open the file: source/class.wpdatatable.php, and find this block:

```
if($this->tableToolsEnabled()){
        $obj->dataTableParams->oTableTools = array(
                'sSwfPath'    =>    PDT_JS_PATH.'jquery-datatables/media/swf/copy_cvs_xls_pdf.swf',
                'aButtons' => array(
                        array('sExtends' => 'xls', 'sFileName' => '*.xls'),
                        array('sExtends' => 'print')
                        )
                );
                if(!$this->isEditable()){
                        $obj->dataTableParams->oTableTools['aButtons'][] = 'csv';
                        $obj->dataTableParams->oTableTools['aButtons'][] = 'pdf';
                        $obj->dataTableParams->oTableTools['aButtons'][] = 'copy';
                }
}
```

(Should be somewhere around lines #2638-2650).

To remove "Copy" button remove this line:

```
$obj->dataTableParams->oTableTools['aButtons'][] = 'copy';
```

To remove "PDF" button remove this line:

```
$obj->dataTableParams->oTableTools['aButtons'][] = 'pdf';
```

To remove "CSV" button remove this line:

```
$obj->dataTableParams->oTableTools['aButtons'][] = 'csv';
```

To remove “Print” button remove this line:

```
array('sExtends' => 'print')
```

To remove “Excel” button remove this line:

```
array('sExtends' => 'xls', 'sFileName' => '*.xls'),
```

9.5 How to disable sorting for one of the columns?

Please add this line to your page’s inline Javascript block:

```
jQuery(function(){
	wpDataTables.table_0.fnSettings().aoColumns[3].bSortable = false;
});
```

This example disables sorting for the fourth column of the first table on the page.

In this block of code you would need to replace the “0” in the **table_0** variable with some other index if you have more then one table on the page (they are indexed starting from zero). E.g. for the third table this should be **table_2**. Also please replace the [3] with the column index, which is also zero-based (e.g. for column number 5 this should be [4]).

9.6 How to filter by invisible columns?

Please see documentation section 5.3

9.7 How to change the position of filtering form for “filter in form” feature?

For this you would need to modify the template which is in the file templates/plain_html_table.inc.

You can move this block:

```
<?php if($table->getFilteringForm()) { ?>
<div class="wpDataTables wpDataTablesFilter">
        <div id="filterBox_<?php echo $table->getId()?>" class="wpDataTableFilterBox">
          <?php foreach( $table->getColumns() as $key=>$column) { ?>
                <?php if($column->getFilterType()->type != 'null') { ?>
                <div  class="wpDataTableFilterSection"  id="<?php  echo  $table->getId().'_'.
$key.'_filter' ?>_sections">
                        <label><?php echo $column->getHeader() ?>:</label>
                                <div id="<?php echo $table->getId().'_'.$key.'_filter' ?>"></div>
                        </div>
                        <?php } ?>
          <?php } ?>
        </div>
</div>
<?php } ?>
```

below the table, or somewhere else where you need it. Or maybe you can just use the widget, see documentation section 5.3 for examples

9.8 How to insert images in columns?

The plugin can render images, and thumbnails which are linked to images. See documentation section 6.7 for all the details.

9.9 How to disable opening links in a popup?

For this you can modify the file source/class.wpdatacolumn.link.php and remove all the occurences of:

```
target='_blank'
```

9.10 How to show only chart without a table?

Please see documentation section 8.1 for the details.

9.11 How to render a table via PHP, not the shortcoed

For this you can call the function directly from PHP like this:

wdt_output_table($table_id);

Where **$table_id** is the table identifier from the shortcode.

10. Troubleshooting

This chapter contains answers on main questions and problems that users have.

11.1 After I create the table I see it on the page without styling, filters, sorting, pagination and buttons

Generally, it is always because of Javascript not executing correctly. Possible reasons:

1. PHP version.

Most frequent reason is that your host has old PHP version. Can you please try updating to PHP 5.4 (should not be a problem). If this is not an option please open the file:source/class.wpdatatable.php Then replace the line

```
return json_encode($obj, JSON_HEX_APOS || JSON_HEX_QUOT || JSON_HEX_TAG);
```

with just

```
return json_encode($obj);
```

But updating PHP would be better.

2. “wpautop” filter

Your theme, or some of the plugins you’re using, are applying the “wpautop” filter to the generated content, and this is ruining the javascript. To check this you can open the source code of the generated page with a table, browse to the position to where the table should be inserted and look at the generated script block. If it looks something like:

```
<p><script></p> <p>($(function(){</p> <p>var oDt_table_0</p> ...
```

Another way how you can check if this is your case by opening the JS developer console (e.g. in Chrome - View->Developer->Javascript console) and checking the reported errors. In case of this problem you will see something like

"Unrecognized token: '<'".

If that's your case then "wpautop" auto-formatting filter is the reason of problems.

You can read more about the "wpautop" filter in WordPress Codex: http://codex.wordpress.org/Function_Reference/wpautop

Generally, several possible solutions for this are possible (you can read about both in the article in Codex):

Check if your theme is allowing to use unformatted content data or excerpt data. Many themes use [raw][/raw] shortcodes for that, but in case of your theme this may be different.

Since usually it's the theme's fault, and if you checked and there are no shortcodes or settings, which can disable the 'wpautop' filter, you can try the following approach: open the theme's directory, and do "search by text" inside it - look for the string "wpautop". Wherever you find that this filter is applied to the "the_content", or "the_excerpt" - just comment it out.

If it's not the theme's fault - find out which plugin is causing this - try to disable plugins one by one.

If you don't want to cut this filter from theme, or plugin, there's a way to insert wpDataTable on the page without a shortcode. You can create a special WP template for the page, and insert the table you generated with a function call:

```
<?php echo wdt_output_table( $table_id ) ?>
```

This way the filter won't be applied.

Add another plugin, which will remove it.

If nothing else helps you can use a "hammer" solution - you can open the file wp-includes/formatting.php of your WP installation, and find the wpautop() function, which is usually on line #188. There on the first line of the function just add:

```
return $pee;
```

This way filter will just return unformatted string.

3. Problem with jQuery library

Some themes (including some versions of the default twenty twelve theme) insert jQuery library in the footer of the page. Is some extreme cases jQuery library is either not included in the theme, or is included several times (e.g. by the theme itself and then by the plugin). To check it you can look through the source code for the "jquery" string. The javascript for jQuery library should be inserted in the header in order for wpDataTables to work, and shouldn't be overwritten by other plugins. You can check if this is your case by opening the JS developer console (e.g. in Chrome - View->Developer->Javascript console) and checking the reported errors. In case of this problem you will see something like "**Undefined function: $()**".

Solution for the Twenty Twelve theme: open the theme's functions file wp-content/themes/twentytwelve/functions.php, and find thefunction twentytwelve_scripts_styles(). There right after the first line global $wp_styles insert this:

```
wp_enqueue_script('jquery');
```

This is enough to make the plugin working!

11.2 After I upload the CSV or Excel file and try to create a table I get an error “File does not exist”.

Some WordPress installations have an option of inserting a "parametrized" URL of the file instead of direct URL. This way when you upload a "file.csv" you get in the URL something likehttp://mysite.com/file.php?id=123 in the URL field instead of http://mysite.com/wp-content/file.csv

These installations have a radiobutton which you can use to switch between kind of URLs inserted - please change it to use the direct file URL and this should solve the problem.

11.3 When I upload the CSV/Excel file and try to generate the table nothing happens

1. **Malformed file.** Some packages save the CSV/Excel files in specific format which cannot be recognized by the plugin, or with some specific column separator. Try to re-save it in some different software (MS Excel, Google Docs, etc).

2. **Too large file**. Too large files can't be processed by the plugin - actually this is not plugin's restriction, but the memory limit allowed to be used by PHP scripts gets exhausted.

You can increase the memory limit in php.ini, if your server allows this, something like:

memory_limit = 128M

Or in the wp-config.php file.

If you didn't find your problem here you can contact us through wpdatatables.com. Please provide:

- Detailed description of the problem;
- Javascript error log, related to the plugin;
- PHP error log, related to the plugin.

Thank you!

12. Information for developers

This chapter contains some information and guidelines for developers that would like to customize wpDataTables for their needs.

12.1 Actions

This section gives an overview on WP actions that are used in wpDataTables.

wpdatatables_before_get_table_metadata($table_id)

This action is executed before the table metadata is fetched from the database (table settings, link to the file or the SQL query). You can use it to call some function that you need, or set some global variable.

wpdatatables_before_get_columns_metadata($table_id)

This action is executed before the columns metadata is fetched from the database (column settings). You can use it to call some function that you need, or set some global variable.

wpdatatables_before_render_table($table_id)

This action is executed before the table has starting to render (either by shortcode or without it).

wpdatatables_after_save_settings()

This action is executed after wpDataTables saves the settings data (after user presses “Save settings” in the backend settings page).

wpdatatables_after_save_table($table_id)

This action is executed after wpDataTables saves the table data (after user presses “Save table” in the backend settings page).

wpdatatables_after_save_columns()

This action is executed after wpDataTables saves the column data (after user presses “Save table” in the backend settings page).

wpdatatables_settings_page()

This action is executed when the settings page is open in the backend

wpdatatables_addnew_page()

This action is executed when user opens the “Add new wpDataTable” in the backend

wpdatatables_browse_page()

This action is executed when user opens the browse page.

wpdatatables_try_generate_table($type, $content)

This action is executed when a new table is first saved: wpDataTables tries to generate the table based on the provided data, and creates the columns and tables in the databased based on the results.
$type is the table type (xls, csv, mysql, json, xml, php).
$content is either a link to the data source file, or MySQL query.

wpdatatables_before_create_columns($table, $table_id)

This action is executed before the column data is saved to the database (when the Save columns button is clicked in the admin panel).
$table is the WPDataTable object (class source/class.wpdatatable.php).
$table_id is the ID of the table description in MySQL.

wpdatatables_after_insert_column($column, $table_id)

This action is executed after the new column is inserted in the database (when the columns are first generated by the plugin).
$column is the column object (source/class.wpdatacolumn.php and its children)
$table_id is the identifier of the table from MySQL

wpdatatables_admin_styles()

This action is executed when the CSS files for the wpDataTables admin panel are enqueued.

wpdatatables_admin_scripts()

This action is executed when the JS files for the admin panel are enqueued.

wpdatatables_get_ajax_data($table_id)

This action is executed before AJAX data is fetched for the server-side processing tables. $table_id is the table identifier from the MySQL table.

wpdatatable_before_render_chart($table_id)

This action is executed before chart starts to render. $table_id is the table identifier from the MySQL table.

wpdatatables_after_frontent_edit_row($formdata, $row_id_value, $table_id)

This action is executed after editing action was applied. $formdata is the array of keys and values for editing, $row_id_value is the ID field of the row in MySQL table that was edited, $table_id is the table identifier from the MySQL plugin table (wp_wpdatatables)

wpdatatables_before_upload_file()

This action is called when the frontend file uploader is used, before the upload is processed.

wpdatatables_before_delete_row($row_id, $table_id)

This action is called before a row is deleted from the frontend editor. $row_id is the value of ID column, $table_id is the table ID from wp_wpdatatables table.

wpdatatables_before_delete_file ($row_id, $table_id)

This action is called before an uploaded file is deleted from the frontend editor. $row_id is the value of ID column, $table_id is the table ID from wp_wpdatatables table.

wpdatatables_before_filtering_form($table_id);

This action is called before the filtering form is rendered on the page. $table_id is the table ID from wp_wpdatatables table.

wpdatatables_after_filtering_form($table_id);

This action is called after the filtering form is rendered on the page. $table_id is the table ID from wp_wpdatatables table.

wpdatatables_before_table($table_id)

This action is called before the table itself is rendered on the page. $table_id is the table ID from wp_wpdatatables table.

wpdatatables_after_table($table_id)

This action is called after the table is rendered on the page. $table_id is the table ID from wp_wpdatatables table.

wpdatatables_before_header($table_id)

This action is called before the table header TR element is rendered on the page. $table_id is the table ID from wp_wpdatatables table.

wpdatatables_after_header($table_id)

This action is called after the table header TR element is rendered on the page. $table_id is the table ID from wp_wpdatatables table.

wpdatatables_before_footer($table_id)

This action is called before the table footer TR element is rendered on the page. $table_id is the table ID from wp_wpdatatables table.

wpdatatables_after_footer($table_id)

This action is called after the table footer TR element is rendered on the page. $table_id is the table ID from wp_wpdatatables table.

wpdatatables_before_first_row($table_id)

This action is called before first table row is rendered on the page. $table_id is the table ID from wp_wpdatatables table.

wpdatatables_before_row($table_id, $row_index)

This action is called before the table row is rendered on the page. $table_id is the table ID from wp_wpdatatables table, $row_index is a zero-based index of the row.

wpdatatables_after_row($table_id, $row_index)

This action is called after the table row is rendered on the page. $table_id is the table ID from wp_wpdatatables table, $row_index is a zero-based index of the row.

wpdatatables_before_editor_dialog($table_id)

This action is called before the table editor dialog (only for editable tables) is rendered on the page. $table_id is the table ID from wp_wpdatatables table.

wpdatatables_after_editor_dialog($table_id)

This action is called after the table editor dialog (only for editable tables) is rendered on the page. $table_id is the table ID from wp_wpdatatables table.

12.2 Filters

This section gives an overview on WP filters that are used in wpDataTables.

wpdatatables_filter_table_metadata($table_metadata, $table_id)

This filter is applied to the table metadata returned from the DB (table settings, link to the data source file or MySQL query). $table_metadata is an array of values. $table_id is the id from database.

wpdatatables_filter_columns_metadata($columns_metadata, $table_id)

This filter is applied to the columns metadata returned from the DB (table settings, link to the data source file or MySQL query). $table_metadata is an array of values. $table_id is the table ID from the database.

wpdatatables_filter_rendered_table($table_content, $table_id)

This filter is applied to the rendered table HTML. $table ID is the table identifier from the database.

wpdatatables_get_system_fonts($system_fonts)

This filter is applied to the array of fonts available for the style generator in the wpDataTables settings page.

wpdatatables_before_save_settings($_POST)

This filter is applied to the contents of the $_POST array when "Save settings" button is pressed in the admin panel, before the actual saving is done.

wpdatatables_filter_browse_page($browse_page)

This filter is applied to the HTML content of the Admin browse page before it is rendered. $browse_page is the default HTML content.

wpdatatables_filter_new_table_page($new_table_page)

This filter is applied to the HTML content of the admin "New table" page before it is rendered. $new_table_page is the default HTML content

wpdatatables_filter_edit_page($edit_table_page);

This filter is applied to the HTML content of the admin "Edit table" page before it is rendered. $edit_table_page is the default HTML content

wpdatatables_filter_settings_page($settings_page)

This filter is applied to the HTML content of the admin settings page before it is rendered. $settings_page is the default HTML content.

wpdatatables_before_save_table($_POST)

This filter is applied to the content of the $_POST array when the “Save table” button is pressed in the admin panel, before the actual saving is done.

wpdatatables_before_save_columns($_POST)

This filter is applied to the content of the $_POST array when the “Save columns” button is pressed in the admin panel, before the actual saving is done.

wpdatatables_try_generate_table_result ($result)

This filter is applied to the result of the initial table which is generated when a new wpDataTable is first saved (this method tries to generate a table, and prepares the metadata in case of success).

wpdatatables_filter_column_before_save($column, $table_id)

This filter is applied to the $column object (source/class.wpdatacolumn.php and its children) before it is saved to the database. $table_id is the table identifier from MySQL.

wpdatatables_filter_update_column_array($update_array, $table_id, $column)

This filter is applied to the array which is inserted in the database (by $wpdb->update()) when the columns are first generated and saved for a new table.

$update_array is an array of keys and values with column settings which will be saved to the database.

$table_id is the table identifier from the database.

$column is the column object (source/class.wpdatacolumn.php and its children).

wpdatatables_filter_insert_column_array($insert_array, $table_id, $column)

This filter is applied to the array which is inserted in the database (by $wpdb->insert()) when the columns are first generated and saved for a new table.

$insert_array is an array of keys and values which will be inserted to the database.

$table_id is the table identifier from the database.

$column is the column object (source/class.wpdatacolumn.php and its children).

wpdatatables_filter_insert_table_array($insert_array)

This filter is applied to the array which will be sent to the $wpdb->insert() method when creating and saving a new table. $insert_array is an associative array.

wpdatatables_filter_update_table_array($update_array, $table_id)

This filter will be applied to the array which will be sent to the $wpdb->update() method when updating table settings. $update_array is an associative array, $table_id is the table identifier from the DB.

wpdatatables_filter_browse_tables($all_tables)

This filter is applied to the array of tables returned from MySQL when the existing tables are fetched from the database.

wpdatatables_filter_mysql_query($query, $table_id)

This filter is applied to the MySQL query before it is sent to MySQL server. $query is the query text, $table_id is the table identifier from the MySQL table (wp_wpdatatables).

wpdatatables_filter_json($json, $table_id)

This filter is applied to the JSON string before building a table based on it. $json is the JSON string itself, $table_id is the table identifier from the MySQL table (wp_wpdatatables).

wpdatatables_filter_simplexml($simplexml_object, $table_id)

This filter is applied to the parsed XML (SimpleXML library is used for parsing the XML files), a SimpleXML object, before creating a table based on it. $simplexml_object is the SimpleXML object, $table_id is the table identifier from the MySQL table (wp_wpdatatables).

wpdatatables_filter_excel_array($parsed_array, $table_id, $file_path)

This filter is applied to the array which is generated based on the Excel file. First each excel file is parsed to the array, which you can later modify with this filter. $parsed_array is the array which is parsed out of the Excel file, $table_id is the table id entifier from the MySQL table (wp_wpdatatables), $file_path is the path to Excel file.

wpdatatables_link_to_skin_css($skin_css_url, $table_id)

This filter is applied to the URL of Skin CSS file before it is inserted in the page. $skin_css_url is the URL to the current skin CSS file, $table_id is the table identifier from the MySQL table (wp_wpdatatables)

wpdatatables_filter_chart($chart_js_content , $series_headers, $series_values, $table_id)

This filter is applied to the string with JS contents which render the charts. JS is using Google Charts library, so you can override or extend it as you like. $chart_js_content is the JS string which will be inserted to the page, $series_headers is the array of headers for the charts series,

$series_values is an array of arrays of values for each series, $table_id is the table identifier from the MySQL table (wp_wpdatatables)

wpdatatables_filter_table_description($object, $table_id);

This filter is applied to an object (PHP's StdObj), which is later printed as a table's data attribute (json_encoded), and used to initialize the DataTables instance, and to describe other table settings and parameters that are used by the frontend JS library. $object is the PHP StdObj which contains all the necessary fields for initializing the table, , $table_id is the table identifier from the MySQL table (wp_wpdatatables)

wpdatatables_filter_style_block($style_block_html, $table_id)

This filter is applied to the <style></style> HTML block that is generated based on the settings that users provide in the plugin settings admin page. $style_block_html is the rendered HTML, $table_id is the table identifier from the MySQL table (wp_wpdatatables).

wpdatatables_filter_frontend_formdata($formdata, $table_id)

This filter is applied to the data sent from the front-end editable table. $formdata contains the keys and values that will be inserted or updated in the table, $table_id is the table identifier from the MySQL table (wp_wpdatatables).

wpdatatables_allow_edit_table($allow_edit, $allowed_editor_roles, $table_id)

This filter is applied to the results of check which defines if current user can or can't edit the table (only for editable tables). The default logic is: if editor roles aren't defined for the table, anyone can edit it; if they are then only the users that have one of these roles can edit it, but you can override the logic using this filter. $allow_edit is the Boolean value returned by the original check. $allow_editor_roles is an array of role names that are allowed to edit the table, $table_id is the table identifier from the MySQL table (wp_wpdatatables).

wpdatatables_filter_cell_val($val)

This filter is applied to the value of the cell before it is returned to the front-end. $val is the cell value

wpdatatables_filter_column_js_definition($jsDef, $columnHeader)

This filter is applied to the object that will be later passed to the front-end DataTables library in JSON format with column properties. You can use it to assign some specific classes, or something else. $jsDef is the current object, $columnHeader is the original column header (e.g. MySQL column name for MySQL-based columns).

wpdatatables_filter_date_cell($formattedValue)

This filter is applied to the value of a date cell before it is returned to the front-end. $formattedValue is the value of the cell.

wpdatatables_filter_email_cell($formattedValue)

This filter is applied to the value of an e-mail column before it is passed to the front-end. $formattedValue is the value of an e-mail cell.

wpdatatables_filter_int_cell($formattedValue)

This filter is applied to the value of an integerCd .. column before it is passed to the front-end. $formattedValue is the value of the cell.

wpdatatables_filter_float_cell ($formattedValue)

This filter is applied to the value of a float column before it is passed to the front-end. $formattedValue if the value of the cell.

wpdatatables_filter_image_cell($formattedValue)

This filter is applied to the value of an image column before it is passed to the front-end. $formattedValue is the value of the cell.

wpdatatables_filter_link_cell($formattedValue)

This filter is applied to the value of an URL link column before it is passed to the front-end. $formattedValue is the value of the cell.

wpdatatables_filter_string_cell($formattedValue)

This filter is applied to the value of a string column before it is passed to the front-end. $formattedValue is the value of the cell.

wpdatatables_filter_column_classes($classes, $columnHeader)

This filter is applied to the set of CSS classes of the column before it is passed to the front-end. $classes is the string with CSS classes (space separated), $columnHeader is the original header of the column (e.g. MySQL table column name in case of MySQL-based table).

wpdatatables_filter_table_classes($classes, $table_id)

This filter is applied to the set of CSS classes of the table before it is passed to the front-end. $classes is the string with CSS classes (space separated), $table_id is the table identifier from the MySQL table (wp_wpdatatables).

wpdatatables_before_generate_constructed_table_name($table_name)

This filter is applied to the name which will be used as a MySQL table name when it is generated by a Table constructor. $table_name is the default name generated by the plugin (wp_wpdatatable_{index}).

wpdatatables_filter_table_title($table_title, $table_id)

This filter is applied to the title of the table before it is displayed in the front-end. If you want to change it dynamically, or add some extra content, you can use it.

Credits

I would like to thank the following people, whose work made wpDataTables possible:

- **Allan Jardine** – author of DataTables jQuery plugin and the TableTools library.
- **Jovan Popovic** – initial author of the ColumnFilter plugin and the Row grouping plugin.
- **Seen Sai Yang** – author of the idea for table responsiveness.
- **Ilya Makarov** – author of the remodal responsive popup.
- **Ben Plum** – author of the FormStone GUI library, parts of which are used in wpDataTables plugin.
- **Sebastian Tschan** – author of the jQuery upload plugin.
- **Amsul** – author of the Pickadate datepicker.
- **Milos Roksandic –** our support manager, and a contributor to wpDataTables codebase
- **Vladica Bibeskovic –** our support manager, and a contributor to wpDataTables codebase

Thank you for reading and I hope you will enjoy using wpDataTables!