Tablesorter

This page documents the implementation of the jQuery Tablesorter plugin within Tiki. Tablesorter was applied more extensively to PluginFancytable in Tiki 11, and then to the user table at tiki-adminusers.php in Tiki 12.

The method used for PluginFancytable differs from the one described here - the plugin implementation will be changed to be incorporated into the one described here so that Tablesorter can be more easily applied to other plugin tables.

Wishlist

Open

Rating	Subject	Submitted by	Importance	Easy to solve?	Priority	Category	Volunteered to solve	LastModif	Comments
● (0)	Plugin alias &/or plugin trackerlist fails to load in some page in dev.t.o (Uncaught TypeError: Cannot read property 'childNodes' of null)	Xavier de Pedro	8	5	40	• Community projects • Dogfood on a *.tiki.org site • Regression		2017-08-14	0
● (0)	Tablesorter regression: date range filters are applied instead of just preselected compared to 15.x	Xavier de Pedro	7	5	35	• Regression • Consistency		2017-01-04	1 lindon-01 Jan 17
● (0)	TableSorter header from bug tables disappeared when using plugin alias (wishes - plugin trackerlist)	Xavier de Pedro	6	5	30	• Community projects • Dogfood on a *.tiki.org site • Regression		2018-05-24	0

Rating	Subject	Submitted by	Importance	Easy to solve?	Priority	Category	Volunteered to solve	LastModif	Comments
● (0)	Navbar overlapping tablesorter heading row (col titles, filter and col select icons)	Xavier de Pedro	3	7	21	• Usability • Dogfood on a *.tiki.org site • Conflict of two features (each works well independently)		2018-12-11	1 Chealer9-11 Dec 18
● (0)	jQuery tablesorter pager at the bottom does not show initially	Philippe Cloutier	3	5	15	• Bug		2019-06-18	0
● (0)	Fancytable datafunction fault in ver 18, okay in ver 15	Robin Squance	4		0	• Usability		2018-04-23	0

Pending

Rating	Subject	Submitted by	Importance	Easy to solve?	Priority	Category	Volunteered to solve	LastModif	Comments
◑ (0)	Tablesorter doesn't allow to sort on tracker item status	Xavier de Pedro	4	5	20	• Error • Consistency • Conflict of two features (each works well independently)		2016-07-22	0

Rating	Subject	Submitted by	Importance	Easy to solve?	Priority	Category	Volunteered to solve	LastModif	Comments
(0)	Tablesorter doesn't allow to filter on Tracker item status	Xavier de Pedro	5	5	25	• Error • Consistency • Conflict of two features (each works well independently)		2016-07-22	0

Closed

[+]

community members interested in Trackers/Tracker dev.

Login	Full Name	LastModif

No records found

Description

Tablesorter is a jQuery plugin used to sort and filter table columns and also apply pagination. Where tables represent rows of data from a Tiki database table, the pagination, filtering and sorting can be applied on the server side and displayed through Ajax. The benefit of using Tablesorter is (hopefully) quicker filtering, sorting and paginating without needing to refresh the entire page.

Tablesorter is housed on github at https://github.com/Mottie/tablesorter. Documentation can be found at http://mottie.github.io/tablesorter/docs/.

Implementation

The jQuery external files for Tablesorter are maintained in /vendor/jquery/plugins/tablesorter through Composer. The additional Tiki-specific code used to generate the needed Tablesorter jQuery code to load into the header is in /lib/core/Table.

Here is a brief outline of how the Tiki portion of the code works when initiated through a static call:

- Call the the Table_Factory class as follows:
  ```
  Table_Factory::build('users', $usersettings);
  ```
- Table_Factory will call itself to build the settings for the specified table (the users table at tiki-adminusers.php in this example) by calling Table_Settings_Users
- These table settings are included in a call to the Table_Manager class made by Table_Factory
- The Table_Manager class calls the Table_Code_Manager class to generrate the jQuery code based on

the table settings
- The Table_Code_Manager class will call various other the Table_Code classes (such as Table_Code_Pager) to create the various sections of the jQuery code and then put the sections together to create the final complete code
- Table_Manager then loads the complete jQuery code into the header

How to Apply to a Table in Tiki

Below are the key steps to implementing Tablesorter for a table in Tiki. The users table at tiki-adminusers.php will be used as an example.

1. Add a file to generate the table settings in /lib/core/Table/Settings. See /lib/core/Table/Settings/Users.php for an example and /lib/core/Table/Settings/Abstract.php for more information on possible settings
    - The naming conventions for /lib/core needs to be followed in order for classes to properly autoload:
        - If the new table is the listing of Tiki groups, then the file should be called /lib/core/Groups.php and the class will need to be:
        ```
        class Table_Settings_Groups extends Table_Settings_Abstract
        ```

2. Prepare the smarty template (or HTML)
    - Add an id to the table element as follows. In this example, the id "usertable" was added to tiki-adminusers.tpl
    ```
    <table id="usertable" class="normal">
    ```
    This id is used in the table settings file
    - Make sure the table has a thead and tbody element
    - Some parts of the page may need to be hidden, for example filtering fields and pagination controls that will be replaced with the Tablesorter functionality
    - Any self_link smarty functions used in the header (previous method of sorting columns) will automatically eliminated through jQuery so no changes to the smarty template are needed for this
    - The reset sort and reset filter buttons are also added through jQuery, so no changes to the smarty template are needed for this

3. Prepare the php file where the data for the table is pulled from the database - tiki-adminusers.php in this example
    This file will be called twice when first invoked. The first time, before the Tablesorter code is loaded into the header, we need to skip the call to the database and invoke Table_Factory instead (although the total records in the database is needed for this call). Once the jQuery is loaded into the header, the file will be called again through Ajax - this time the database call should be made. See below for an example:
    ```
    //////////////this part was added for Tablesorter////////////////////////////////
    //this variable means that Tablesorter will be applied
    ```

```
$tsOn = $prefs['disableJavascript'] == 'n' && $prefs['feature_jquery_tablesorter'] ==
'y' ? true : false;
//this should be a smarty variable in case some parts of the smarty template need to be
hidded or altered when applying Tablesorter
$smarty->assign('ts', $tsOn);
//if $_REQUEST['tsAjax'] is true, that means this is an ajax call from Tablesorter
$tsAjax = isset($_REQUEST['tsAjax']) && $_REQUEST['tsAjax'] ? true : false;

//do the database call if this is the Tablesorter Ajax call or if Tablesorter won't be
used
if ($tsAjax || !$tsOn) {
///////////////end of first part added for
Tablesorter////////////////////////////////////
        $users = $userlib->get_users(
                $offset,
                $numrows,
                $sort_mode,
                $find,
                $initial,
                true,
                $filterGroup,
                $filterEmail,
                !empty($_REQUEST['filterEmailNotConfirmed']),
                !empty($_REQUEST['filterNotValidated']),
                !empty($_REQUEST['filterNeverLoggedIn'])
        );
/////////////this part was added for Tablesorter////////////////////////////////////
//if Tablesorter will be used, invoke it before getting rows from the database
} elseif($tsOn) {
    //the total number of rows is needed for the Tablesorter pager to work properly
        $users['cant'] = $userlib->count_users('');
        $users['data'] = $users['cant'] > 0 ? true : false;
        Table_Factory::build('users', array('total' => $users['cant']));
}
///////////////end of second part added for
Tablesorter////////////////////////////////////
```