AUTOBACKOFFICE

Technical documentation

ABO Version 4.1.0

Last Updated on August, 10th 2024

Contact us

Website	www.autobackoffice.com
YouTube	youtube.autobackoffice.com
Facebook	facebook.com/autobackoffice
Twitter	twitter.com/autobackoffice

Contents

What's new in 4.1.0

If you already use ABO, here you can find a list of all the features of the new version. Please feel free to refer to the full documentation for more details.

Compatibility with MySQL 8.0 and PHP 8.2

ABO is now compatible with MySQL 8.0 and PHP 8.2 too.

Before MySQL version 8.0, you would always declare for example an integer as *INT(11)* , but in MySQL 8.0 that declaration has been deprecated.

Just to clarify: it's always been a misunderstanding that the number represented the maximum number of digits of the INT, while it simply represented the number of digits to be displayed if the ZEROFILL option was enabled.

In MYSQL 8.0, this kind of declaration has been deprecated for all integer types.

ABO still interprets the zerofill number if it's there, otherwise will happily work without it.

Supported graphic files

Added WEBP files are now supported

Views

Added Supports for views, albeit not fully. Any view can be listed and customized as if it were a table.

Even if and when (rarely) possible, at the moment you cannot delete / update elements in a view.

Logs

ABO can now log all the select / insert / update / delete according to the selected tables.

Config Tables

Added Sort field: the user can drag&drop the rows and they will be sorted accordingly

Example: suppose you have a query in your website printing all the items according to a field called sortOrder: you simply tell ABO that "sortOrder" is your sort order field, and the user will be able to change the order with drag&drop!

Added CSS SQL Match / CSS Value: If a row in that table meets the criteria, that specific CSS value will be applied to the whole row while viewing it.

Example: you want to write in red all the inactive rows. You can write `active='no'` in *SQL Match*, and `color:red;` in *CSS Value*.

CAREFUL! If the *SQL Match* has a syntax error, ABO will not be able to properly show the table, so if you do not see the tables anymore in the client, check the *SQL Match* content

Added Start Sort: every table will be able to be sorted by default at the beginning by any field and in any direction.

Example: you have an "orders" table, usually you want to show the new orders first, so you simply set the order to be "id desc" at the beginning

Added Add/Edit/Erase level: if the login users are stored in a DB table, you can assign them privileges, and if they are not enough, they cannot add/edit/erase elements.

Picture upload

Now a picture can be grabbed by its URL. ABO will download it locally, upload it to the right folder and change the DB accordingly (there are still some cases, depending on the browser, in which the picture is not correctly refreshed, but the feature works perfectly).

Embedded HTML editor

Added Supports for "<a href>" links.

Tables

Added Button to force to show all the records, regardless of the pagination

Added It's now possible to print a number in front of each and every record, useful for remote debugging

Added You will be able to use an iPhone-like switch for binary yes/no DB values, with a displayed value too that can be different from the DB value.

Example: in the DB you have an “active” field with “Y,N” values. You can tell ABO to use that field as a switch with “Y” being ON, and “N” being OFF, plus you can also specify – if needed – what to print next to the switch, and it can be any string, like “YES!” for ‘Y’ and “NO!” for ‘N’.

Login passwords

Added Now the passwords can also be SHA-512 encoded

User pages

Added Search bar can be disabled

Default types

Added In case of 'time', a timepicker will be shown by default if not overridden

Customization

Added Debug switch: if enabled, you can see the table structure and run custom queries in the client

Default options

Added Show/edit password only if your ABOlevel is not the same or higher than the one you want to view/edit. Of course, passwords are masked ALSO in PDF/XLS/CSV exports.

ABO Server

Added In the table configuration, you will see a real time help with the *edit1_aux* and *view1_aux* options and their meanings

Added You can print your date field with the special formatting php_date, using the "date" parameters (e.g. "Y-M-d")

Added You can print ANY custom data from ANY custom field because you can provide your own PHP code to print your data

ABO Client

Added You can instruct ABO to let the user type something with autocomplete, getting the data from the same column they are editing

Added You can instruct ABO to treat any type (e.g. varchar or tinytext) as an Integer and a Float, with a maximum length too

Added The upload_dir path can now be customized with specific column values.

For example, you want to store the pictures in different folders according to the user, you can set upload_dir to ../img/{userId}/

Added The equivalent of a foreign key in ABO can now be printed with any custom query instead of a single field like in the past.

For example, you have a field called userId, in the past you would have linked it to the "name" field of the "users" table, and that was it. Now, you can write your own query to display whatever you want, e.g. CONCAT(`name`, ' ', `lastName`)

Added If properly configured, you will be able to use an iPhone-like switch for binary yes/no DB values.

Introduction

ABO has been written with simplicity in mind: you can start to use it in literally 30 seconds, and you can add more and more functionalities when you really need them.

Too many tools force you to configure everything even when you do not need all the functionalities: ABO is different. It grows only when you really need it to grow.

So, this documentation is just a reference: you are advised to look at the YouTube videos first, and then using this documentation to get faster access to some specific features.

Please use this link to access our YT channel:

https://youtube.autobackoffice.com

Or, in particular, to the 30 seconds installation:

https://www.youtube.com/watch?v=Se-sdBKloMo

The new features added from the last version will be marked with **4.1.0** .

ABO history

Most products come to life because people need a new solution to an old problem.

ABO, short for AutoBackOffice, is no exception.

We were using some available tools, some commercial, some free, but we reached a point at which, even the (expensive) commercial products didn't fit our needs anymore. We had a specific problem: we do not use WordPress to create websites, we create them specifically for a customer, from scratch. But then, we always had to create a backoffice in order to let our customer modify the database. The available tools fell into 2 categories: either too crude, both graphic-wise and letting final users full access to everything, or too complicated to use, create and update, both for us and for our final customers.

So, we did what all companies usually do: we created an internal tool, and ABO 1.0 was born. And we kept improving it, until we decided to release it to the public starting from version 2.0, and as everybody says... The rest is history :)

What it all means is that ABO is NOT optimized to run on a custom server, but 90% of ABO installations usually run on a shared hosting environment, in which all the standard packages are usually already installed and working.

We tried to use only standard libraries, even though most of the cPanel shared hosting nowadays let you configure which libraries to include, but in case you are wondering, if you don’t have them, you will need at least mysql and mbstring, and imagemagik if you plan to use the feature to transform PDF to JPG. If you have root access, these are the commands to update your PHP (at the moment of writing, 8.2 is the current PHP version):

apt-get install php8.2-mysql

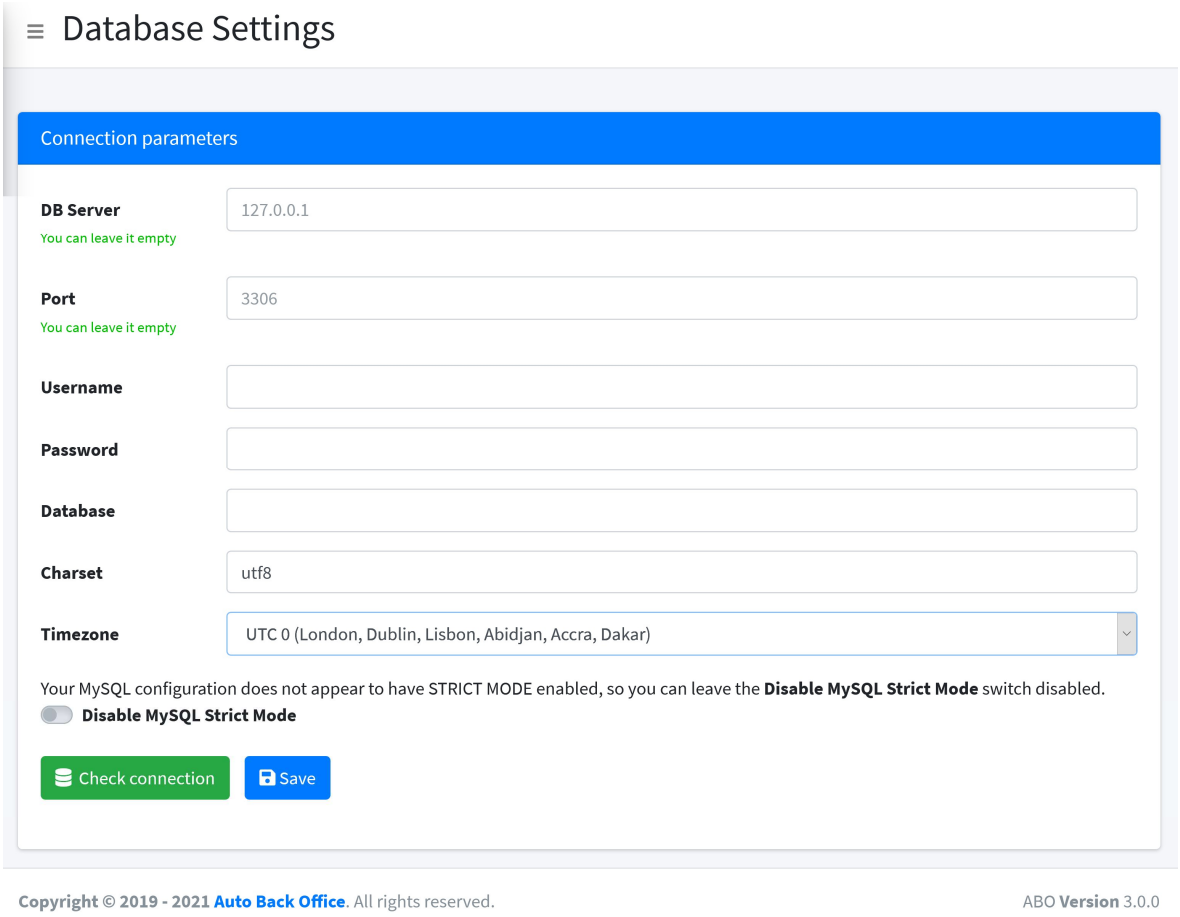
apt-get install php8.2-mbstring

systemctl restart apache2

DB Settings

YouTube video: https://www.youtube.com/watch?v=Bxokq0MYz4E&t=23s

In this page you must specify all the parameters for the MySQL connection.



Basically, you must provide the data for these two PHP commands:

```
$conn = @new mysqli($cfg["localhost"], $cfg["username"], $cfg["password"], $cfg["database"]);
$conn->set_charset($cfg["charset"]);
```

ABO can use a DB stored on another server but, most of the times, the 3306 port (MySQL) is not open for security reasons.

If you cannot connect to an external DB, you must ask the server administrator to enable the port 3306 for you. If it is a shared hosting, then they will probably not be able to do it just for you.

3.0.0 You can now select a specific port, other than 3306.

3.0.0 You can select a time zone for both PHP and MySQL, so that all date/time functions will refer to it

Disable MySQL strict mode

In a nutshell: don't touch it if everything works. If you have problems when you INSERT your records, then enabling this feature will probably help. Let's see why.

If you have some fields with NOT NULL values, and the default is not defined, MySQL usually assigns to these fields some implicit values (e.g. all the integers are set to 0).

Unfortunately, this is not true in some specific MySQL versions, in which if the strict mode is enabled, then MySQL requires you to set a specific value if it is declared NOT NULL.

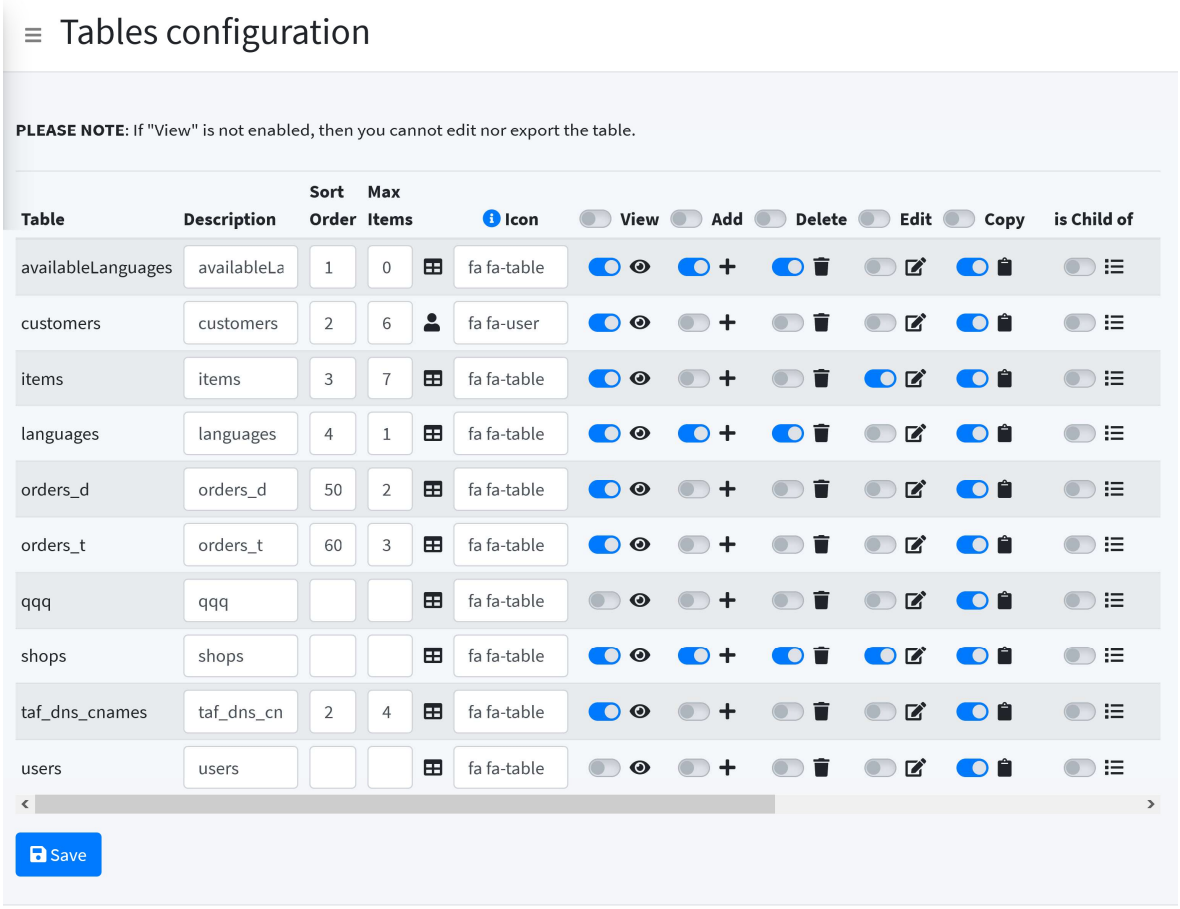
In this case, when ABO tries to create / clone / change a value, if there is no value, ABO doesn't know what to insert in the query, and MySQL stops ABO from proceeding.

There are four different solutions to this problem:

1. Remove the NOT NULL definition from the field
2. Set a default value for the field in the table definition
3. Remove the STRICT_ALL_TABLES and/or the STRICT_TRANS_TABLES from the MySQL sql_mode variable from the MySQL configuration
4. Turn on the switch called "Disable MySQL Strict Mode". This will reset the sql_mode configuration variable at each and every ABO DB connection.

Tables Config

YouTube video: https://www.youtube.com/watch?v=Bxokq0MYz4E&t=44s



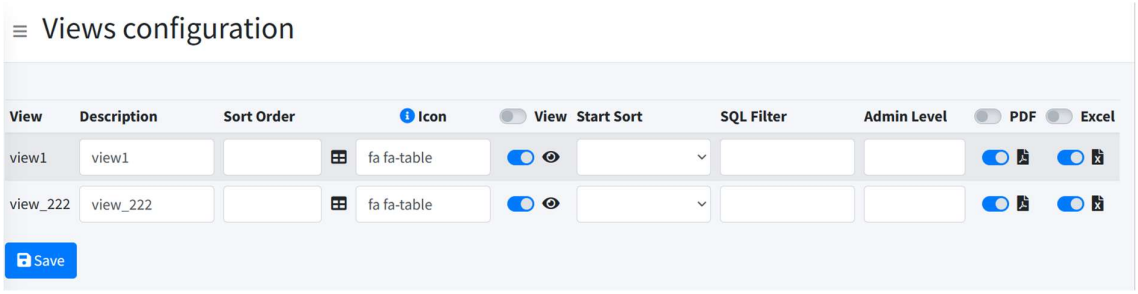
Here you can configure all the DB tables from a general point of view. For each table you can specify:

- **Description**: it's the "pretty name" that will be shown to the final user
- 3.0.0 **Sort Order**: It is a number which represents the order in which the tables will be shown to the ABO client. If there is no number inserted, the tables will be shown in their natural order
- 3.0.0 **Max Items**: If there is a number different from zero, the ABO client will be blocked from inserting more than the number of records specified. 3.1.0 If **Multi User Mode** is enabled, then every user will have this number as the maximum items to be inserted. If you want, you can specify a different maximum value for each user. Please refer to the specific section for more information
- **Fa icon**: it's the font awesome icon used near the table name. You can simply type it in real time, and it will be updated, or if you do not remember their names, you can click on the "info" icon and you will be redirected to the Font Awesome website
- **View**: the final user will be able to view the page
- **Add**: the final user will be able to add a record to the table
- **Delete**: the final user will be able to delete a record from the table
- **Edit**: the final user will be able to edit a record in the table
- 3.0.0 **Copy**: the final user will be able to copy the content of the row to the clipboard, with either TAB or COMMA to separate the fields, according to the default setting

- **Is Child Of**: this is used to simulate a foreign key. It will ask for the table and the two fields of the relationship. In our data example, suppose we have an invoice_head and invoce_details, this option will let you simulate a query like
  SELECT invoice_details.*
  FROM invoice_details, invoice_head
  WHERE invoice_head.id = invoice_details.invoiceId AND invoice_head.id = XXX
- 4.0.0 **Start Sort:** It will show ALL the possible fields, both ASC and DESC, so that you can force a starting sort order for that specific table
- 4.0.0 **Sort Field:** You can pick any integer field to act as a sort field, and the user will be able to drag&drop that row to change its order value
- 3.0.0 **SQL Filter:** A fixed SQL "WHERE" string attached to each and every SELECT. Suppose you want the final user to only see the data with *id<10*, then you can simply type it here.
- 4.1.0 **CSS SQL Match:** You can write a SQL WHERE clause, so that if a record has that value, then the whole row will use the CSS Style defined in the next field, for example *id=2*
- 4.1.0 **CSS Value:** This CSS style will be applied to all the rows satisfying the condition of the previous field "CSS SQL Match", for example *color:red;*
- 3.0.0 **Admin Level:** If the login type is "From Table", then every user can have a level assigned to him/her, and if their level is not bigger or equal to this value, the user will not be able to interact with this table. An example is the login table itself: you want the admin to be able to add/delete users, but you do not want all the users to see/modify this table
- 4.0.0 **Add/Edit/Erase Level:** If login is *From Table* and every user has a level, every user will be able to add/edit/erase items only if their level is high enough
- 3.0.0 **Multi User Field:** If this table must be handled as a multi user table, here you must specify the field of the table to be used as the user field "virtual" foreign key. See the corresponding explanation at the end of this document to have a more detailed explanation
- **PDF**: the final user will be able to generate a PDF of the data in the table
- **Excel**: the final user will be able to generate a CSV file from the data in the table

4.0.0 Views Config

YouTube video: Not available



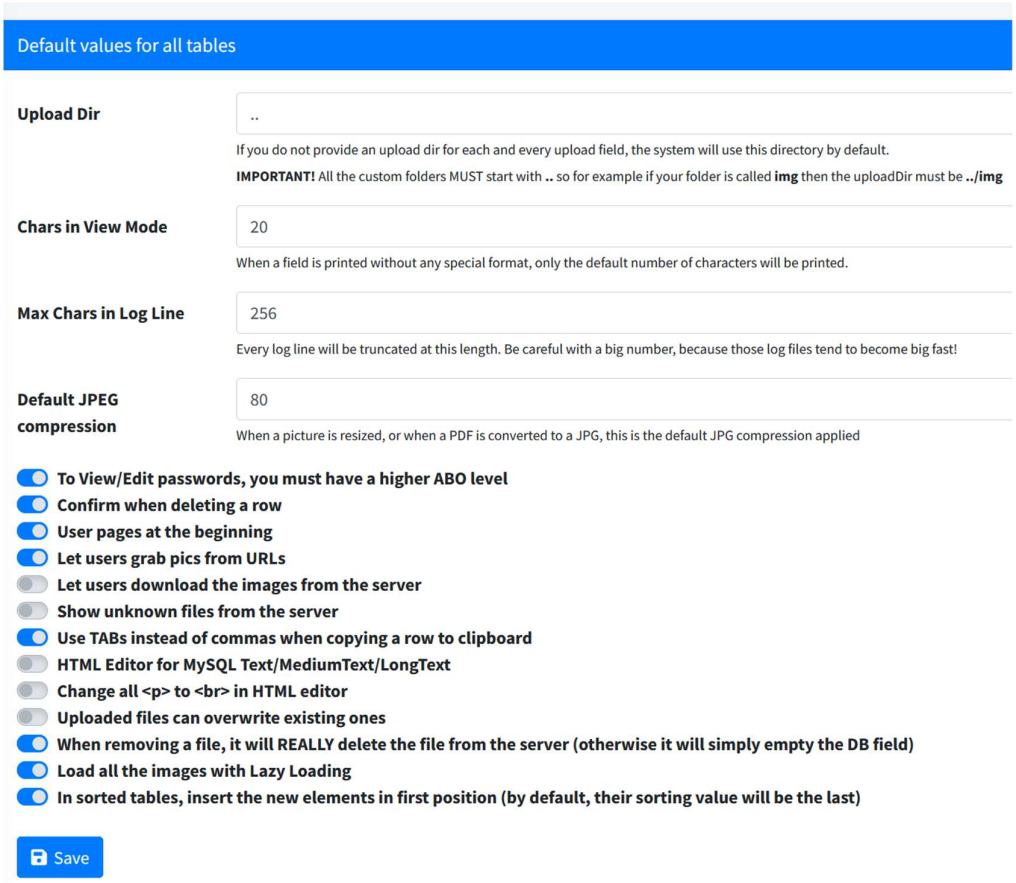
Here you can configure all the DB views from a general point of view. For each view you can specify:

- **Description**: it's the "pretty name" that will be shown to the final user
- **Sort Order**: It is a number which represents the order in which the tables will be shown to the ABO client. If there is no number inserted, the tables will be shown in their natural order
- **Fa icon**: it's the font awesome icon used near the table name. You can simply type it in real time, and it will be updated, or if you do not remember their names, you can click on the "info" icon and you will be redirected to the Font Awesome website
- **View**: the final user will be able to view the page
- **Start Sort:** It will show ALL the possible fields, both ASC and DESC, so that you can force a starting sort order for that specific view
- **SQL Filter:** A fixed SQL "WHERE" string attached to each and every SELECT. Suppose you want the final user to only see the data with *id<10*, then you can simply type it here.
- **Admin Level:** If the login type is "From Table", then every user can have a level assigned to him/her, and if their level is not bigger or equal to this value, the user will not be able to interact with this table. An example is the login table itself: you want the admin to be able to add/delete users, but you do not want all the users to see/modify this view
- **PDF**: the final user will be able to generate a PDF of the data in the view
- **Excel**: the final user will be able to generate a CSV file from the data in the view

View support is still experimental. In particular, without a primary index, ABO cannot show a single record as if it were a table. You will only be able to see all of the records together.

Default Options

YouTube video: https://www.youtube.com/watch?v=Bxokq0MYz4E&t=120s



Upload Dir: since you are supposed to install ABO at the same level of your index.php file, ABO will never upload the files to its own directory, but it would start from the upper level (that is, "**..**").

3.1.0 **Chars in View Mode**: by default, if a field is displayed without any specific format, ABO will print the first 20 characters. This parameter will override this default number, letting you choose how many characters will be displayed.

4.0.0 **Max chars in Log Line**: by default, every log line is limited to 256 characters to avoid creating huge log files. If needed, this value can be increased here.

3.2.0 **Default JPEG compression**: when uploading a JPG file (or a PDF to be converted to JPG) the default compression rate is 80, but this value can be modified here.

4.0.0 **Let users grab pics from URLs**: If this option is enabled, the users will be able to grab a picture giving an URL. ABO will download the picture, will upload it to the correct folder and update the database. This feature is useful is e.g. there is an old website from which pictures must be downloaded, re-uploaded and the DB synched. It's way fast to copy/paste the URL and let ABO do the rest. WARNING This feature, while fully working, doesn't correctly refresh the image sometimes.

4.0.0 **Show unknown files from the server**: If this option is enabled, the users will be able to pick any file from the server, even unknown type ones.

Confirm when deleting a row: If ON, an alert will be shown before deleting a row

2.3.0 **User pages at the beginning**: by default the user pages are listed AFTER the DB tables. If you use this switch, they will be shown BEFORE the DB tables

3.1.0 **Let users download the images from the server**: When you are allowed to UPLOAD an image and you select "From Server", then you will also be able to download all the images from the server to your local computer

3.1.0 **Show unknown files from the server**: When you are allowed to UPLOAD an image and you select "From Server", then if there are files which are NOT recognized as images, if this option is enabled, you can also download these unknown files from the server to your local computer

3.0.0 **Use TABs instead of commas**: If you enable the "copy" function in the Tables Config, then here you can choose between using TABs or commas when copying a row to the clipboard

HTML Editor: By default, all the Text/Medium Text/Long Text DB fields are edited using a HTML TEXTAREA. If you enable this switch, the quill HTML editor will be used (https://quilljs.com/) **4.0.0** Added the "link" (a href) option, and better re-arranged some icons

3.1.0 **Change all <p> to
 in the HTML editor**: Depending on how you setup your main website and/or your CSS, sometimes you do not want to embed all the text in paragraphs, but you simply want to go to the next line when a paragraph is over. By default, the HTML editor that ABO is using (Quill - https://quilljs.com/) wraps all text in **<p>** tags. If you select this option, all the **<p>** tags will be replaced with **
**. The best way to understand if this option helps you, is to simply modify the text using the HTML editor, and see what looks correct on your website.

3.2.0 **Uploaded file can overwrite existing ones**: If enabled, uploaded file can overwrite existing ones. By default, you cannot overwrite those.

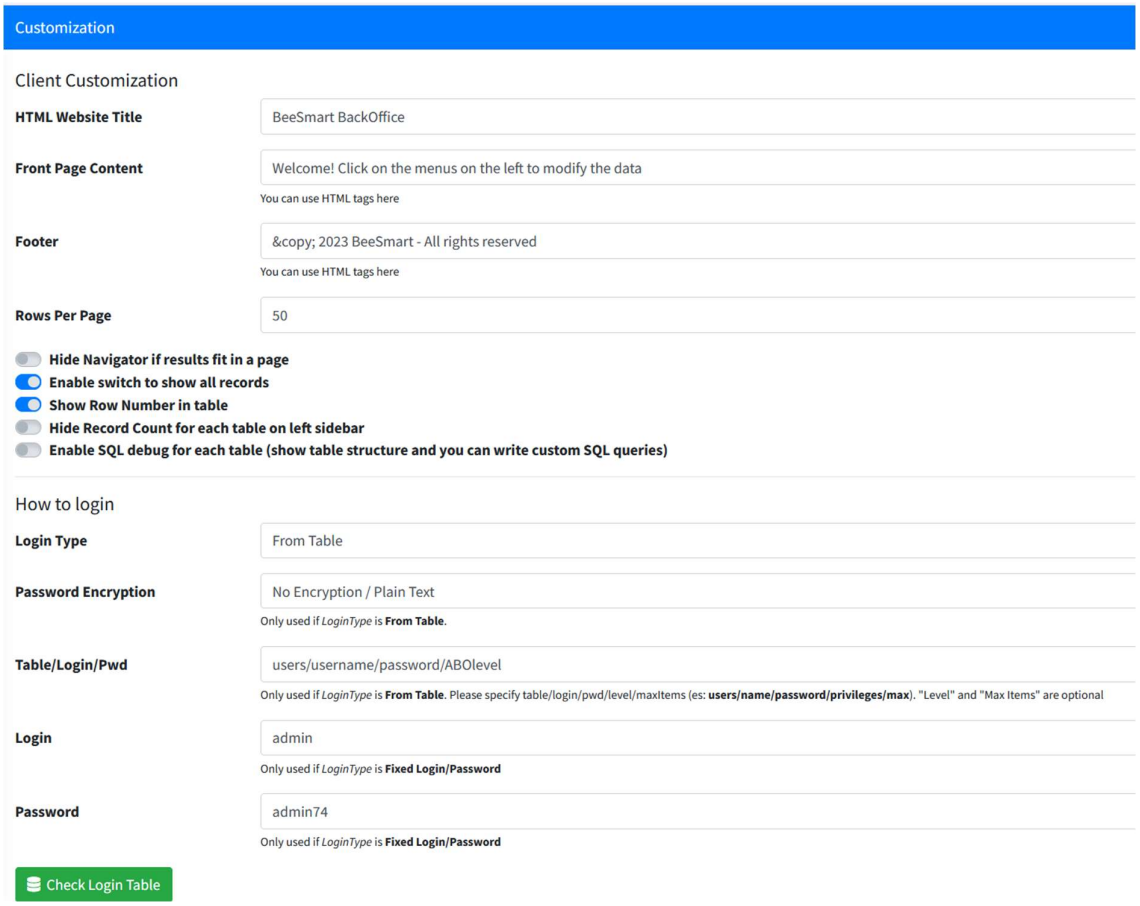
3.2.0 **When removing a file, it will actually be deleted**: When you handle files and/or images, if you delete them from a field, the field is simply emptied. If this option is enabled, the file gets actually deleted from the server. By default, no file is ever deleted.

3.2.0 **Load all images with lazy loading**: If this flag is set, abo client will add the lazy loading attribute to all the displayed images.

4.0.0 **In sorted tables, insert the new elements in first position (by default, their sorting value will be the last)**: As it says, when you have sorted tables and you add an element, by default that element will go to the last position. If this flag is set, the newly inserted element will be the first one in the sorting.

Customization

YouTube video: https://www.youtube.com/watch?v=Bxokq0MYz4E&t=157s



HTML Website title: It's the <title> HTML property of the client

Front Page Content: What the final user will see in the client right pane. You can also use HTML tags. It's basically a customizable welcome screen for the final user

Footer: What the final user will see in ABO's footer

Rows per page: The default number of paginated rows in the client

3.0.0 **Hide Navigator:** If the results fit in a page, do not show the navigator

4.0.0 **Enable switch to show all records:** The user will be able to force ABO to show all the records in a table, regardless of the pagination

4.0.0 **Show Row number in table:** It will show a counter at the beginning of each record, starting from 1. This is useful in case of remote assistance without the possibility of sharing the screen. Especially when the client cannot see the primary index (which is very common), this number can be used to identify a record

3.0.0 **Hide record count:** Usually, ABO shows how many records are available in each table on the sidebar. In case of big tables and a lot of them, this could slow down the client a bit, so this feature can be turned off.

4.1.0 **Enable SQL Debug:** If enabled, at the top of each table in ABO Client you will see the table structure and you will be able to execute ANY query with the same privileges of the user that you configured in the "DB Options" page.

Login Type: It can be none, fixed, from table:

- **None**: it won't require any password. While we suggest you should never use this option, there are cases (private use, intranet not connected to the internet, and so on) in which you could use it for a simplified access
- **Fixed**: hardcoded value for login/password.
- **From table**: You will have to pick a table with two corresponding fields for login and password. Of course, you should then hide this table to avoid people reading it

Table/Login/Pwd/ 3.0.0 **Level /** 3.1.0 **MaxItems**: In case you select "from table" from the previous option, then you should specify your table, login and password field here, separated by a slash. For example: usersTable/name/pwd. There is a possibility to specify even a fourth field, with the privileges of the user (so that he/she can only see the tables as specified in Tables Config). If **MultiUserMode** is enabled, then you can specify a field for the maximum number of items for that specific user

4.0.0 In case the login type is "From Table", then the passwords can be either plain or SHA-512 encoded

Login/Password: In case of a hardcoded login, you will have to fill the data here

3.0.0 **Allow PDF to JPG conversion**: If this option is enabled, in case of an upload of a picture in PDF, it will automatically be converted into a JPG. This option requires ImageMagick, which is usually installed with each version of PHP, but just to be sure, check that it's working on your server. If ABO cannot find it installed, it will show a warning, but it will let you use this function anyway.

Favico for the client: The favico in the ABO client. It can be uploaded ex-novo or it can be chosen from a picture already on the server. Usually, the favico is a square PNG with a transparent color, but you are free to choose whatever you like. Try to use at least a 128x128 pixel resolution image.

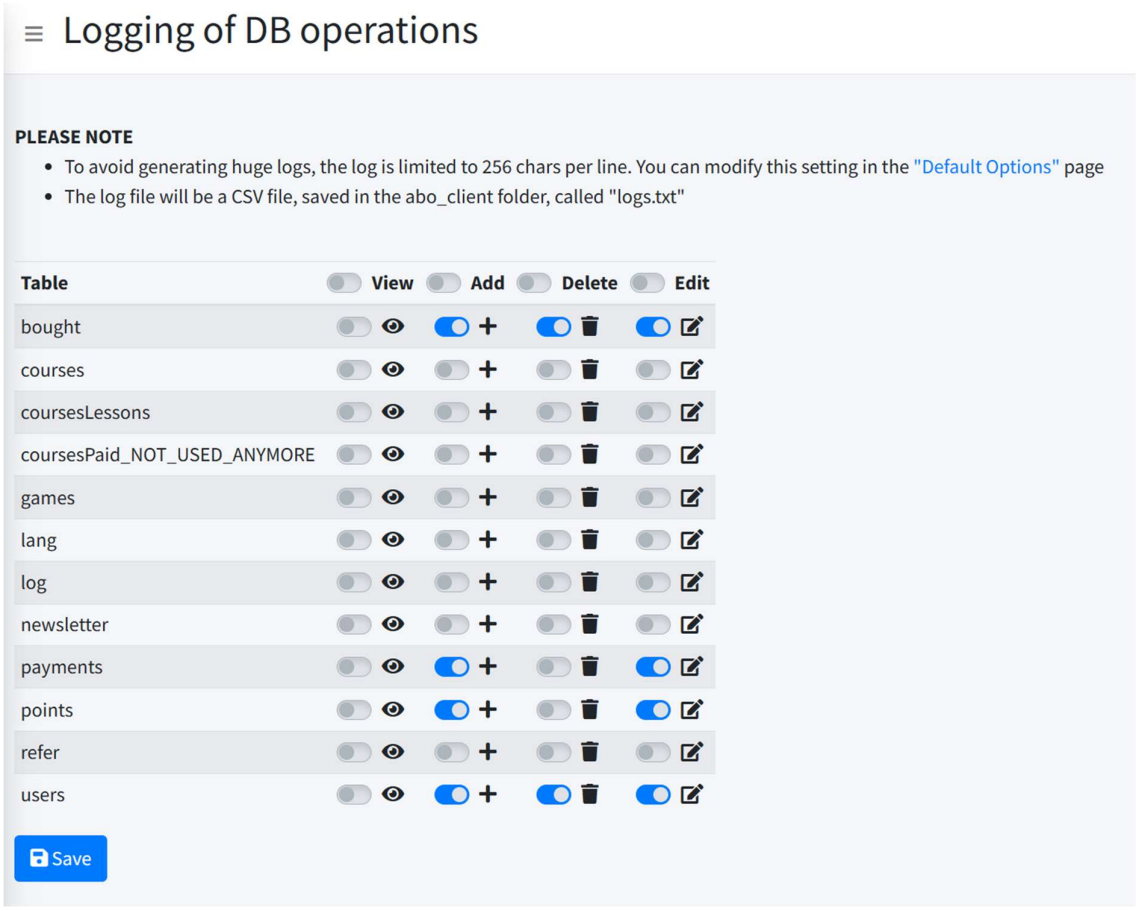
Logo for login page: The logo shown on the ABO login page. It can be uploaded ex-novo or it can be chosen from a picture already on the server. Try to make it at least 512 pixels wide.

Logo for backoffice: The upper left corner logo for the ABO client. It can be uploaded ex-novo, or it can be chosen from a picture already on the server. The standard resolution is 200x28 but, of course, any picture with a suitable aspect ratio will do.

3.0.0 **Enable Multiuser:** With this option enabled AND the **Login Type** set to **From Table**, you can handle the client with multiple users. Please refer to the specific section for more information

4.0.0 Logs Page

YouTube video: Not available at the time of writing. Please subscribe to the YT channel.



Starting with version 4.0.0, ABO supports a basic form of logging of every operation performed by the client. While the logging capabilities could seem basic, they are very granular.

In fact, you can choose to log any select / insert / delete / update operation for each and every table independently.

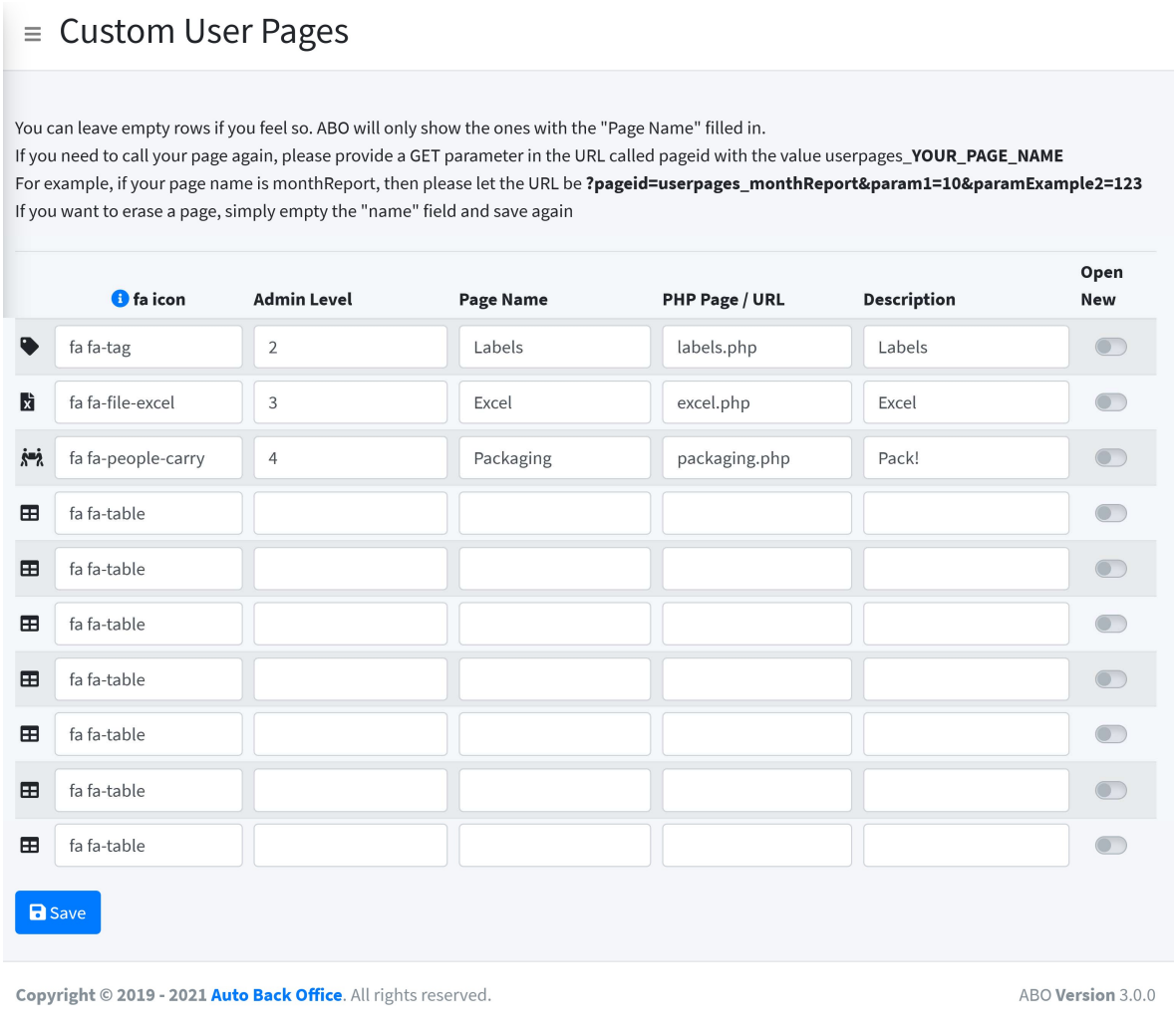
By default, the log string is limited to 256 chars to avoid huge log files, but this can be modified in the “Default Options” page.

The log file itself is called *logs.txt*, it’s stored under the *abo_client* folder, and it’s a CSV file, easily imported e.g. in Excel.

User Pages

YouTube video: Not available at the time of writing. Please subscribe to the YT channel.

Available since 2.3.0



This option lets you create custom PHP user pages, to be shown on the client.

Sometimes, letting your user to interact with the database is not enough. Maybe you want to generate a specific report, or you want to show a graph, or you simply want to embed other functions/pages, not directly related to the DB manipulation.

With these user pages, you can do exactly this: adding custom pages to the ABO client.

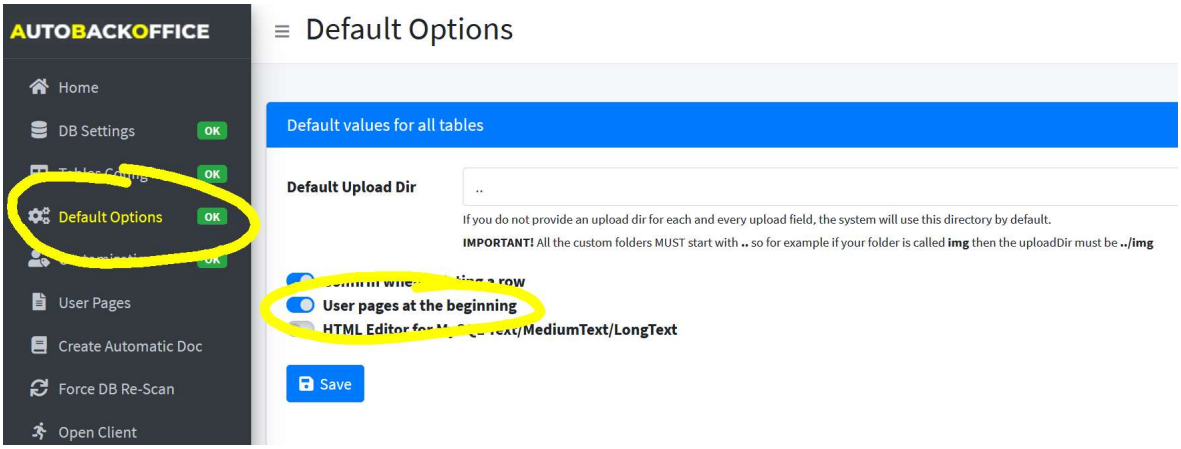
Of course, you are in control: if you want to mix and match data from the DB with other sources, feel free to do it! You can write any kind of PHP page that you wish!

A page can simply be a link to another website, or it can point to a PHP script on the ABO client folder. If it is a PHP script, keep in mind that:

- All the output will be shown on the right pane
- You have jQuery, Bootstrap, SweetAlert2 and Font Awesome available

- If you need to call your page again, please provide a GET parameter in the URL called `pageid` with the value `userpages_YOUR_PAGE_NAME.` For example, if your page name is `monthReport`, then please let the URL be
  `?pageid=userpages_monthReport&param1=10&paramExample2=123`
  To avoid caching issues with some web servers/clients, if your content is dynamic, please append to the URL a random value, to make it different every time. In PHP, you can use e.g. `uniqid()`, while in JS you can use `Math.random()`

By default, in the ABO client the custom user pages will be shown after the DB tables, but you can force them to be shown before the DB tables by changing the corresponding switch in the "Default Options" ABO server page as shown below.



For each page, you can choose:

- **The Font Awesome icon**: as per the regular tables, you can immediately start typing the name of the icon and it will be updated in real time (e.g. fa fa-eye). The default icon is "fa fa-table". As usual, if you click on the "i" icon, you will be able to open the Font Awesome website to get a list of all the available icons.
- 3.0.0 **Admin Level**: If the Login Type is "From Table", then you can also specify a privilege level so that only the users with the same or greater privilege can access it
- **Page Name**: This is your internal name of the page, eventually used if you want to call your page again from within your page and/or another one (as explained before, and as it will be explained again below). There cannot be any spaces in the name. The name must be unique in the "User Pages". Please be aware that ABO does not make any checks in that regard, so if you use a DUPLICATE name, ABO will not be able to distinguish between the two.
- **PHP Page / URL**: This is either the PHP file name, or the final website. If it is a website, please include the corresponding protocol (e.g.: https://www.google.com). If it is a page, you can ALSO include a valid path on your server (e.g.: scripts/mypage1.php), otherwise the page will have to be stored in the same "abo client" directory.
- **Open new**: If it is switched on, a new page/tab will be open when the user clicks on it (basically, the link will be `<a target="_blank" href="your link">Description</a>`
  If it is switched off, the page will open on the right pane. All the OUTPUT of your page will be shown on the right pane.
  Usually, if you need to open it in a new tab, it means it is a different website. If you need to

show it into the right pane, it is an internal page of your backoffice.
If you need to call your internal page again, please provide a GET parameter in the URL called `pageid` with the value `userpages_`**`YOUR_PAGE_NAME`**
For example, if your page name is `monthReport`, then please let the URL be
`?pageid=userpages_monthReport&param1=10&paramExample2=123`

- **4.0.0** **Hide SearchBar**: If it is switched on, the user page will NOT show the search bar on top (and in fact this should be the standard behavior, since at the moment there is no way to implement a search in the user pages)

User Page Example

This is an example of an external page called `page2`.

It will print a button that, when pressed, will call ITSELF AGAIN using `pageid=userpages_page2` adding also `param1`, and the value of `param1` will be increased at every call.

At the bottom, it will create a row divided in 3 columns using the “col-md-4” Boostrap class.

Note:

- The usage of Bootstrap classes: no need to include and/or declare it
- The button to call the page again with the `pageid` parameter, followed by any other custom parameter.

```
<?php
  $v = @$_GET["param1"]; // sanitize it in real life :)
?>
<a  class='btn btn-primary'
    href='?pageid=userpages_page2&param1=<?=++$v?>'>
    Increment
</a>
<h1>param1 = <?=$v?></h1>
<div class=row>
  <div class=col-md-4>
    col1
  </div>
  <div class=col-md-4>
    col2
  </div>
  <div class=col-md-4>
    col3
  </div>
</div>
```

Create Automatic Doc

YouTube video: https://www.youtube.com/watch?v=Bxokq0MYz4E&t=638s

This option is only available in the Professional / Enterprise edition, not in the free version.

It automatically generates a PDF with 2 different sections: in the first one, it will detail all the tables and all the fields with a description of what you can do with them in plain English.

The second part is a technical documentation of the MySQL DB, with all the tables and the corresponding type.

While this has an intrinsic value, especially in this day and age it's always a good sign to deliver to your customer some PRINTED documentation, but writing a document is always time consuming, nobody wants to do that, and it's prone to mistakes.

With ABO, all is done for you with a single click, using as much data as possible from your configuration files.

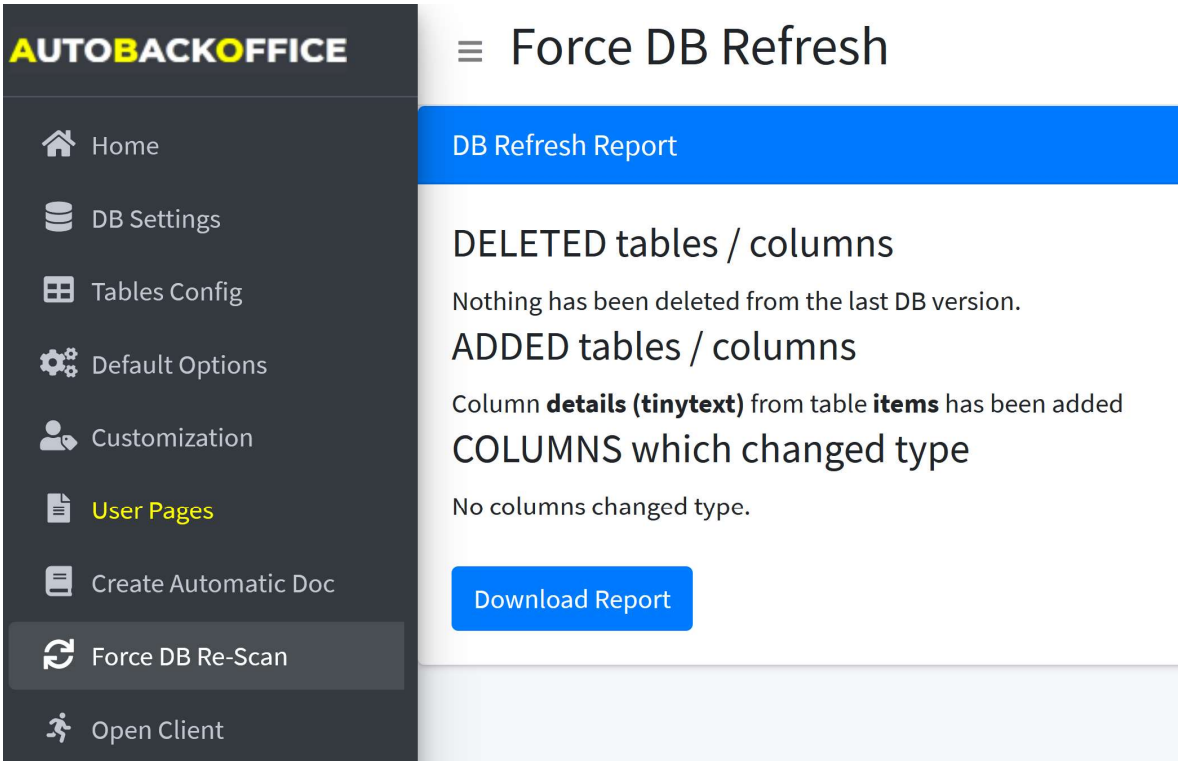
For example, the first page will be printed with the login logo and the HTML Website title; the footer of the page will be taken from the Footer customization, but in particular the field description and help will be taken from your table details.

If no detail is available, ABO will write something inferring it from the MySQL type, e.g. if it's a TINYTEXT value, ABO will say that you can write a string up to 127 characters long, while if it's an INT(11) it will say that the final user can enter an integer value with at most 11 digits.

Force DB Re-Scan

YouTube video: https://www.youtube.com/watch?v=Bxokq0MYz4E&t=687s

This is one of the BEST options of ABO, probably the option that makes you save THE MOST TIME compared with all the competitors.



Suppose you need to add a "details" field to a table as a short string. You add the column to the table, and now you should update your backoffice. What do you do, usually? You need to open your configuration file, find the table/field, force the type, compile/rescan the data, save it, upload to the website, and so many other operations, right?

Welcome to ABO: **you click ONCE on "Force DB Re-Scan" and you are DONE**. Done. Nothing else to do.

ABO will also generate a report of what has been added, deleted, or changed.

Sure, ABO won't know how you want your user to view / edit that field, but if it's an integer, it will be treated as an integer; if it's a short string, it will be treated as a short string.

Basically, with another backoffice generator, every time you change the DB, it's always a headache.

With ABO, one click will solve all your problems.

Since 4.0.0, also the **views** are supported.

Open Client

YouTube video: https://www.youtube.com/watch?v=Bxokq0MYz4E&t=315s

This option will simply open the “abo_client” folder on your website, so you can check it in real time

Defaults for viewing / editing values

YouTube video: Not available

One of the best ABO features is that you do not really need to configure ABO client: if a DB field is specific enough, ABO will use some specific ways to display or to edit that field, without you having to customize anything. That's why, most of the times, once you enable a table, you do not need to do anything else and it will simply work.

This is a list of all the view / edit behaviors of ABO according to the MySQL types, without any custom configuration. Of course, you can always force them to behave in a custom way

Field Type	View Behavior	Edit Behavior
enum	Show the value. If customized with aliases, you will see them instead of the actual values	Combobox with all the values. If NULL is possible, an empty row will be shown. If customized with aliases, you will see them instead of the actual values
Any "bit" values	Decimal value	Can only press 0 or 1 keys
Timestamp Datetime	Date / time value	Open a datetime picker
GEOMETRY POINT LINESTRING POLYGON MULTIPOINT MULTILINESTRING MULTIPOLYGON GEOMETRYCOLLECTION	Text representation	Textarea
BINARY VARBINARY BLOB TINYBLOB MEDIUMBLOB LONGBLOB	First 20 values, base 64 encoded	Textarea
TEXT MEDIUMTEXT LONGTEXT	First 20 characters	Textarea
date	Date value	Date picker
time	Time value	**4.0.0** Time picker
tinytext	First 20 characters	Text, limited to 127 chars
Decimal Int Tinyint Smallint Mediumint Bigint	Number	Only digits, limited to the max defined number
Char varchar	First 20 characters	Up to the defined length
set	Show the value	Combobox with all the available values

Current DB tables

YouTube video: https://www.youtube.com/watch?v=Bxokq0MYz4E&t=343s

For every table, you will have a list of all the available fields with the MySQL type.

Table **items**

Table items

Field Name	MySQL Type	Status
id	int(11)	Hidden
name	tinytext	Read Only
qty	tinytext	Writable
price	tinytext	Writable
picture1	tinytext	Writable
picture2	tinytext	Writable
picture3	tinytext	Writable
notes	text	Writable
varchar	varchar(50)	Writable
lastQtyUpdate	timestamp	Writable
available	enum('Y','N')	Writable
erased	enum('Y','N')	Writable
size	tinytext	Writable

With each field, you can perform 6 different actions:

- View the ABO configuration for that field, by pressing on the field name
- Edit the ABO configuration for that field, by pressing on the cog icon

- Hide that field by pressing the eye icon
- 3.0.0 Making the field read-only by pressing the slashed-eye icon
- 3.0.0 Making the field writable by pressing the edit icon
- Copy the format of a field, to be pasted on another field

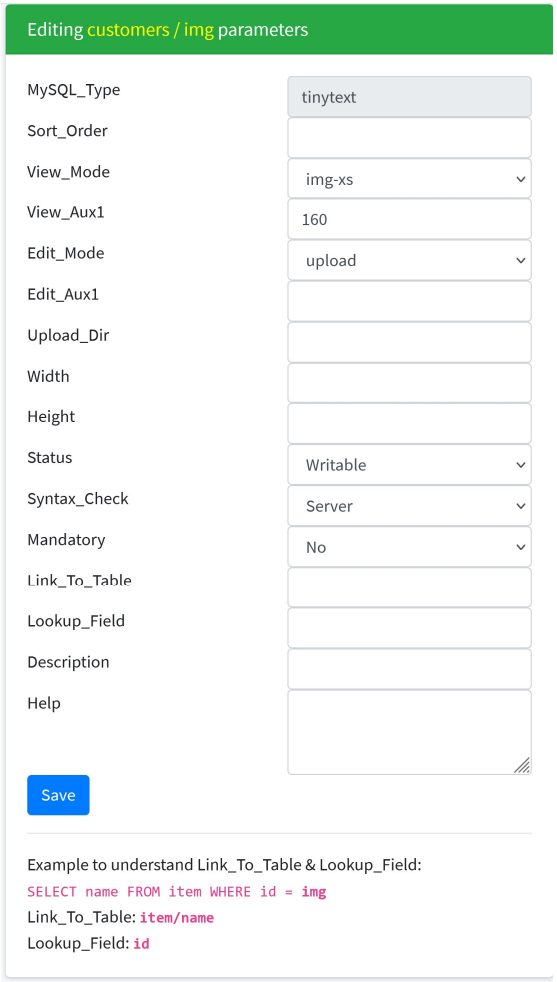
Copy Format

This is one of the most useful ABO features, that as far as we know, it's not available in any other backoffice generator.

Suppose you want to configure an image: you have to specify that it will be an image, then the dimension while viewed, the final folder of the upload, and several other parameters. Now, suppose you have 10 images: you have to repeat this process 10 times, losing time and risking making a mistake.

With ABO, you can simply press the copy icon and "paste the format" on another field. Not only that: ABO will correctly show ONLY the fields with a DIFFERENT format, so it will be faster to check the fields which have still to be modified. Furthermore, if a field has a different MySQL type, a question mark will appear next to the field, because usually you do not want to paste a format if the MySQL type is different.

All parameters



MySQL_Type: it's read-only, it shows the original MySQL type

3.0.0 **Sort_Order**: ABO Client will sort the fields according to this number. If no number is entered, ABO client will show the fields in the same order that they appear in the DB

4.0.0 If you change the *View_Mode* and *Edit_Mode*, now you will receive a real time hint/help on the possible values for *edit1_aux* and *view1_aux*

View_Mode: it can be one of the following:

- **4.0.0** **Custom.php**: it lets you print the field data in any custom format you want. It requires 3 steps:
  1) create a folder called *view* under *abo_client*
  2) create a file called *tableName.field.php* under the *view* folder
  3) the variable *$data* will contain the field value, and ABO will print its value at the end

  For example, suppose you have a table called *customers* and a field called *name*, and you want to write that name with the first capital letter. You will need to:
  1) create the folder *view* under abo_client if it doesn't exist yet
  2) create a file called *customers.name.php* and put it under *abo_client/view*
  3) That file will contain **<?php $data = ucwords($data) ?>**

  IMPORTANT! ABO will echo htmlentities($data), so there is no need to do it yourself!
- **3.1.0** **print_N_chars**: It will display the first "N" characters of the text field according to the "View_Aux1" field. If there are more, ABO Client will print **[…]** at the end.
- **4.0.0** **php_date:** It will print the value according to the *view_aux1* value as if it would be printed in PHP using the "date" command (e.g. `Y/M/d`)
- **4.1.0** **switchOnOff:** It It will display an iPhone-style switch depending on the values in *Edit_Aux1* field. The format is *yes,no* meaning the first value will turn the switch on, the second will turn it off. You can also print a text next to the switch, described in *View_Aux1*, in the same *yes,no* format, so that the user can also have a text explanation next to the switch. Please note that while the values in *Edit_Aux1* **MUST MATCH** the values in the DB (and this is usually used in ENUM('Y','N') fields), the values in *View_Aux1* are totally arbitrary.
- **Full text:** It will display the whole field. By default, ABO only shows the first 20 chars.
- **Img:** It will display the image at the native resolution
- **Img-xs:** It will display the image with a forced width of 32px
- **Img-s:** It will display the image with a forced width of 128px
- **Img-l:** It will display the image with a forced width of 512px
- **3.1.0** **img-custom**: It will display the image with a forced width of the pixels specified in the "View_Aux1" field.
- **Base64_full:** It will display the whole field encoded in base 64
- **Base64_cut:** It will display the first 20 bytes of the field encoded in base 64
- **3.1.0** **color_picker**: It will display the color both in Hexadecimal (#80ff00) and using a border of the same color, like this: #80ff00

3.1.0 **View_aux1**: Used to provide additional parameters to the view.

- **3.1.0** If **View_Mode** is *img-custom*, this represents the width in pixel of the image

- 3.1.0 If **Edit_Mode** is file and if *View_aux1* is **download**, then the user can also download back the file
- 4.1.0 If **Edit_Mode** is switchOnOff**,** this is a text next to the switch in the same *yes,no* format, so that the user can also have a text explanation next to the switch.
- 3.2.0 If the MySQL type is "enum", then you can specify the corresponding strings to be printed instead. Suppose you have `ENUM('fa fa-users', 'fa fa-phone')`, you can specify here **User Icon, Phone Icon** to display a "pretty print" of those enum values. Or, suppose you have `ENUM('Y','N','H')`, you can specify **Yes, No, Hidden.** The values must be separated by commas, and quotes/double quotes are not supported yet.

Edit_Mode: it can be one of the following:

- **Textarea:** A HTML textarea will be used (3.2.0 col/rows can be forced using "Edit_aux1", see below)
- **Html:** A WYSISYG HTML editor will be used (https://www.quilljs.com until 4.0.0)
- **Upload:** Select this one if you want to be able to upload an image
- **File:** Select this one if you want to be able to upload a generic file
- **Tinytext:** It will be treated as a generic, single-line text
- **switchOnOff:** as already described in "View_Mode"
- 4.0.0 **Autocomplete**: it creates an autocomplete dropdown as soon as the user types something. What ABO does is a "SELECT DISTINCT" from that specific field and uses it for the dropdown values
- 4.0.0 **Integer / float**: it gets the maximum length from *edit_aux1*. ABO does it automatically for all the corresponding types, but in this case you can force it to be integer/float even if e.g. the field is tinytext.
- 3.1.0 **color_picker**: It will display the system "Color Palette" window so that you can either manually pick a color, or write the corresponding Hexadecimal code (this depends on the browser and/or OS implementation – the "input type=color" command is used)

3.1.0 **Edit_aux1**: Used to provide additional parameters for the edit operation.

- 3.1.0 If **Edit_Mode** is *upload*, this represents the FORCED NAME of the uploaded filename
- 3.1.0 If **Edit_Mode** is *file*, this represents the FORCED NAME of the uploaded filename
- 3.2.0 If **Edit_Mode** is *textarea,* you can write *row,col* (example: 10,5) to specify the rows and cols of the textarea. If you specify those values, the textarea WILL NOT be styled by bootstrap
- 4.1.0 If **Edit_Mode** is *switchOnOff,* you must write the two values in the DB that corresponds to the on/off state of the switch. So, for example, if the DB field is an ENUM('Y','N'), you must enter *Y,N*

Upload_dir: the final folder in which the file/image will be uploaded. Keep in mind that this folder is relative to the abo_client folder, so USUALLY if you are using ABO as a website backoffice, this field should ALWAYS start with "../" to go back to your website root folder 4.0.0 if in the path you specify a field name in curly brackets, ABO will replace it with the current value. For example: *../img/{customerId}/* ABO will use the "customerId" field to load from/save to that folder. This is useful to keep for example pictures separated in different folder for different categories

3.1.0 **Width**: Prior to 3.1.0, this was the minimum width of the uploaded image. Any image smaller than this were to be discarded. From 3.1.0 on, ANY image will be accepted and adapted to be AT LEAST this size. If the aspect ratio doesn't match, the picture will be centrally zoomed and cropped

to fit into the Width/Height restrains. **3.2.0** If the file type is "**Upload**", then this is the maximum size in bytes of the uploaded file. It's checked locally first, and then also on the server.

3.1.0 **Height**: Prior to 3.1.0, this was the minimum height of the uploaded image. Any image smaller than this were to be discarded. From 3.1.0 on, ANY image will be accepted and adapted to be AT LEAST this size. If the aspect ratio doesn't match, the picture will be centrally zoomed and cropped to fit into the Width/Height restrains.

Status: Can be set to hidden, read only or writable

Syntax_check: not used (at least until 3.0.0)

Mandatory: not used (at least until 3.0.0)

Description: the name displayed in ABO client, and also in the automatic PDF generation

Help: A text displayed next to the field in ABO client, and also in the automatic PDF generation

Link_to_table and **Lookup_field**: These two parameters are used together to simulate a foreign key. Let's see an example:

```
SELECT name FROM item WHERE id = id
```
Link_To_Table: **`item/name`**
Lookup_Field: **`id`**

As you can see, **Link_to_table** contains the two connected tables and **Lookup_Field** contains the "link field" from the second table.

4.0.0 If link_to_table contains a double quote after the table name, ABO will use that following string as a MySQL query. For example, if *Link_To_Table* equals to *users/"CONCAT(`name`,' ',`lastName`)* , then the final user will be able to see not only the *name* field, but the name followed by a space followed by the last name.

4.0.0 If *Mandatory* equals to "No", then the field can also be 0 (and it will print '---' with NULL value). If you want to customize the "0" value, you can put in *view_aux1* what you want to print, and in *edit_aux1* the value you want to set (usually it's 0 or NULL, but you are free to set any value you want)

3.0.0 Multi User management

YouTube video: Not available at the time of writing. Please subscribe to the YT channel.

Available since 3.0.0

Suppose you have several sales agents, each of them needing to handle their own customers in the DB.

Of course, you cannot let them see, modify or delete somebody else's customer.

Suppose you have these 2 tables:

CUSTOMER
id
name
salesAgentId

AGENT
id
name
password

How can you use ABO to let them only handle their respective customers?

1. In the "Customization" page, set Login Table to "From Table", and Table/Login/Pwd to agent/name/password
2. In the "Customization" page again, enable the MultiUser option (at the bottom of the page)
3. In the "Tables Config", for the table "Customer" select "salesAgentId" from the MultiUser field.

That's it!

From now on, when an agent logins into the system, ABO will handle all of it automatically.

Suppose the agent id is 30:

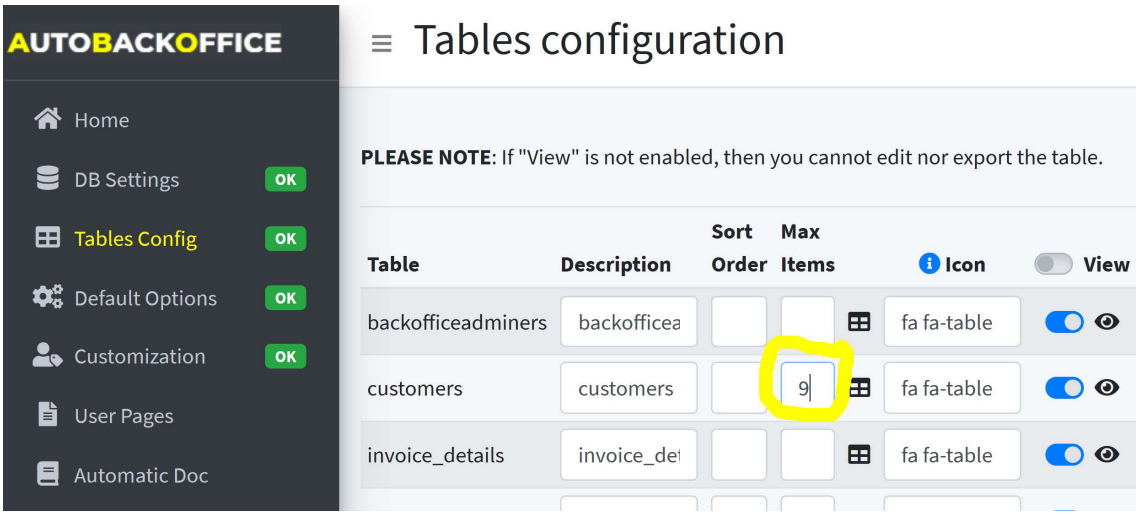
- In case of VIEWING the customer table, ABO will only show the owned customers (basically, adding to the SELECT something like "WHERE salesAgentId=30"
- In case of ADDING a new customer, automatically (and without showing it to the final user) adding the salesAgentId field with the value 30.

3.1.0 Max Items in case of Multi User

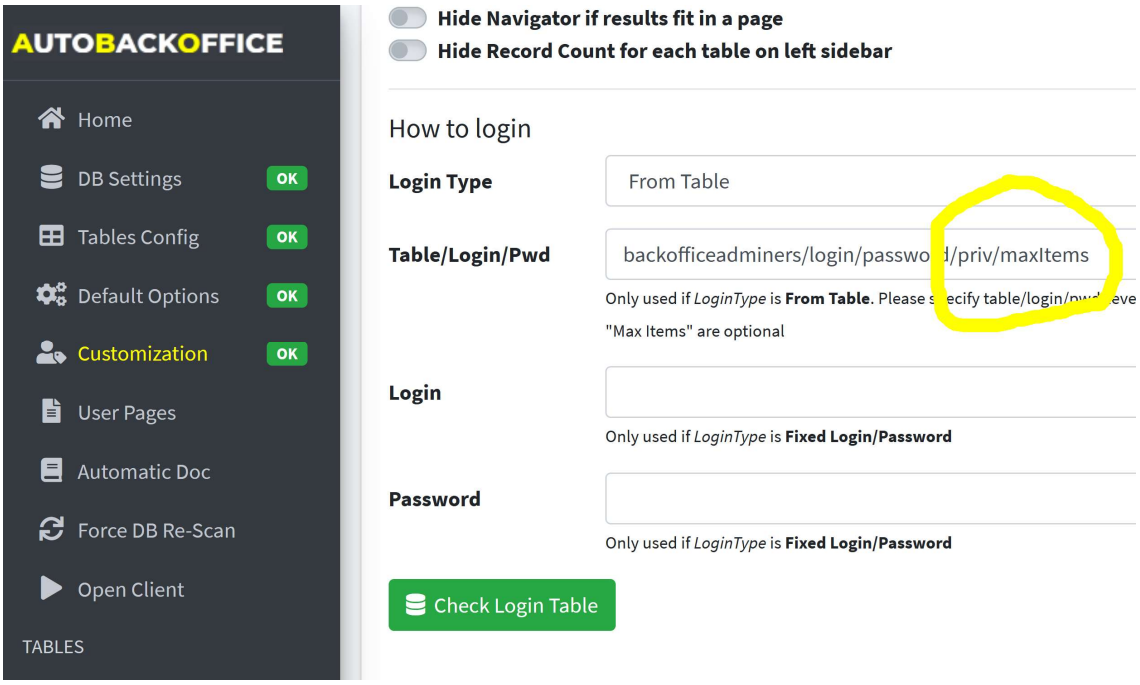
YouTube video: Not available at the time of writing. Please subscribe to the YT channel.

Suppose you have an eCommerce portal in which each shop wants to upload several items. Depending on how much they pay, they can upload a different number of items.

If you want ALL of them to be able to upload – let's say – a maximum of 9 items, then in the **Tables Config** you can simply write "9" in the **Max Items** box, like this:



If you want to be able to specify a maximum number of items FOR EACH SHOP, then you can link a specific field of the login table to the max items in the **Customization** page:



ABO Support

Depending on the version that you use, you are entitled to different levels of support.

FREE VERSION

You have access to:

- This documentation of course
- Installation guide at https://www.autobackoffice.com/index.php?pageid=installation
- YouTube videos at https://youtube.autobackoffice.com
- FAQs at https://www.autobackoffice.com/index.php?pageid=faqs
- Known Issues at https://www.autobackoffice.com/index.php?pageid=knownIssues
- For commercial inquiries only https://facebook.com/autobackoffice or https://twitter.com/autobackoffice

PROFESSIONAL EDITION

All of the above, plus:

- Ticket access, with an answer within 24/48h at https://support.autobackoffice.com

ENTERPRISE EDITION

All of the above, plus:

- Direct email support, with an answer within 24h at support@autobackoffice.com

Implementation of a new feature

If you need a special feature implemented just for you, we adopt this simple policy:

1. Since it would be too time-consuming to keep several versions/subversions for all the different customers, we will implement that feature only if we deem it useful for all our customers
2. We will send a quote for adding the feature
3. If accepted and paid for, we will implement it
4. We will generate a new version for ALL our paying customers with that feature available

Basically, you would pay for the feature for everybody else, but our quote will be more of a token than a real full price of the development, and it will be done specifically for you first.

The price will also change depending on the type of license that you have. If you have the free version, we will request that you buy the professional or the enterprise version first.

Fixed Bugs from previous versions

- In some specific instances, if viewmode was FullView and it was empty, then you couldn’t edit it.
- In some specific instances, priority level was not considered when editing a row.
- If sortOrder was enabled, when editing a single field, you could ALSO see the sortOrder.
- When notification of edit came after saving, it was showing the NAME of the field and NOT the "description" (the "new name" assigned in ABO, eg field is longName, description is "Name on Receipt", the notification showed longName)
- When renaming an uploaded file because name existed, it stopped if there were already 10 existing files. Now it basically tries until there is an available name
- SERVER: single quotes on description of table name prevented to load all the tables config (not fully sanitized)
- login from table only worked with “login” table name.
- If edit is disabled, it can be enabled again in "VIEW ROW"
- If sortdragdrop is enabled in a table, it's ALWAYS enabled
- Edit_mode forced to “html”, if default was not html, then it was not working
- Edit single field with priority and login from table not working in some specific cases
- In the upload image, the jpg compression value was not handled correctly sometimes
- After a DB refresh, in some cases the table names on the left contained some garbage characters

What if something goes wrong

I DO NOT SEE MY TABLES ANYMORE IN THE CLIENT

Check that there are no syntax errors in *SQL Match*. If it's not empty, please remove any text from it and open the client again. If it's working, then there is a syntax error in your *SQL Match*.

I MADE A (WRONG) MODIFICATION TO THE TABLES CONFIGURATION AND I WANT TO GO BACK

Every time you modify the tables configuration, ABO automatically creates a backup in the form of conf.abo.timestamp. Simply erase conf.abo and replace it with the latest conf.abo.timestamp