

AUTOBACKOFFICE

Technical documentation

Last Updated on Feb. 10th, 2021

Contact us

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

Introduction

While reading this documentation could feel, at times, overwhelming, 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>

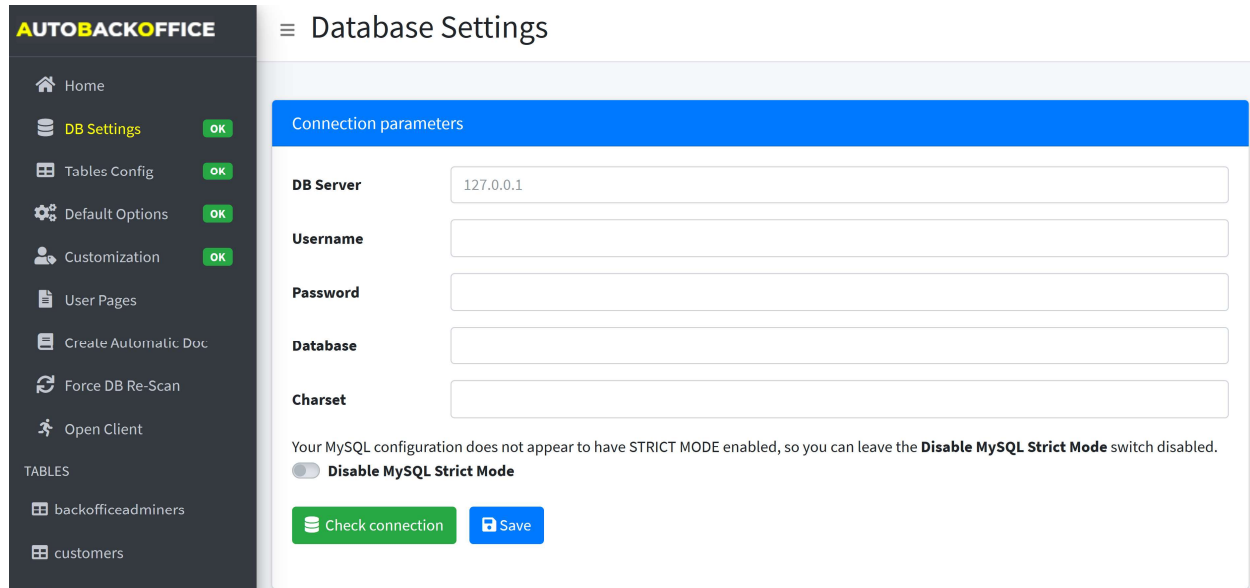
Or, in particular, to the 30 seconds installation:

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

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 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.

In case you have a non-standard port, please contact us to enable that option.

It is on our to-do list, but it is a low priority issue since no customer has ever asked to change it to this day.

Disable MySQL strict mode

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's 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

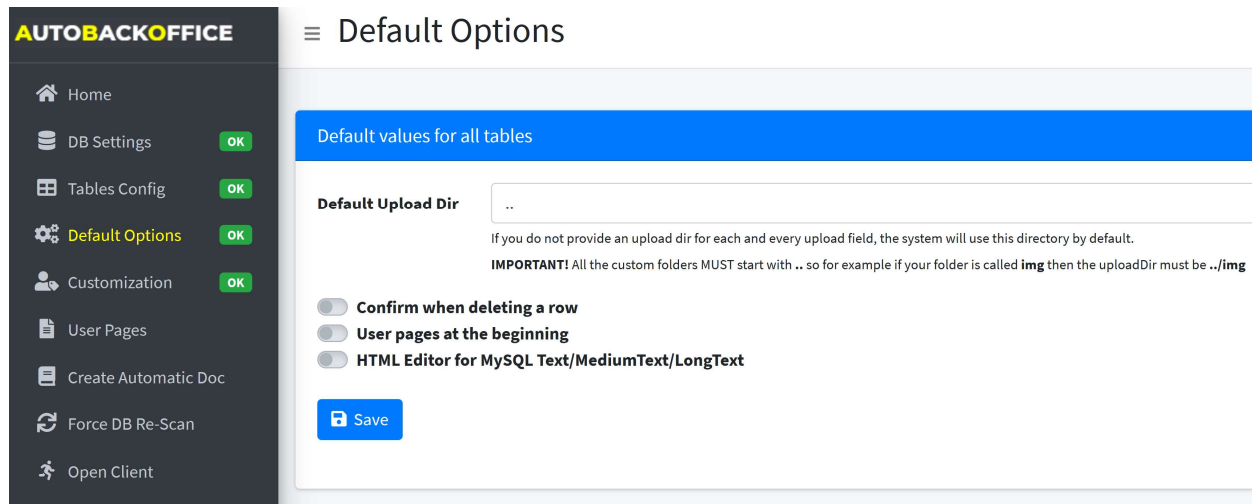
YouTube video: <https://www.youtube.com/watch?v=Bxokq0MYz4E&t=44s>

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
- **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
- **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
- **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 invoice_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

Default Options

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



The screenshot shows the 'Default Options' configuration page in the AUTOBACKOFFICE application. On the left is a dark sidebar with a menu containing: Home, DB Settings (OK), Tables Config (OK), Default Options (OK), Customization (OK), User Pages, Create Automatic Doc, Force DB Re-Scan, and Open Client. The main content area is titled 'Default Options' and features a blue header 'Default values for all tables'. Below this, the 'Default Upload Dir' is set to '..'. A note explains that if no upload dir is provided, the system will use this directory by default, and an important warning states that custom folders must start with '..'. Three toggle switches are present: 'Confirm when deleting a row' (disabled), 'User pages at the beginning' (disabled), and 'HTML Editor for MySQL Text/MediumText/LongText' (disabled). A blue 'Save' button is at the bottom.

Default 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, “..”).

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

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

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/>)

Customization

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

AUTOBACKOFFICE

Home

DB Settings **OK**

Tables Config **OK**

Default Options **OK**

Customization **OK**

User Pages

Create Automatic Doc

Force DB Re-Scan

Open Client

Custom parameters

Customization

HTML Website Title

Front Page Content Welcome to the backoffice! Please use the column on the left to start modifying the data, thank you!

Footer

Login Type No Password

Table/Login/Pwd

Only used if LoginType is **From Table**. Please specify table/login/pwd (es: **users/name/password**)

Login

Only used if LoginType is **Fixed Login/Password**

Password

Only used if LoginType is **Fixed Login/Password**

Rows Per Page 20

Favicon for the client

Please use a PNG picture with a size of at least 128x128

Choose file Browse From server

Logo for Login Page

Choose file Browse From server

Logo for Backoffice

Choose file Browse From server

Check Login Table Save

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

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
- 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: 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

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

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

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.

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.

User Pages

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

Available since 2.3.0

AUTOBACKOFFICE ≡ Custom User Pages

You can leave empty rows if you feel so. ABO will only show the ones with the "Page Name" filled in.
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¶m1=10¶mExample2=123`
If you want to erase a page, simply empty the "name" field and save again

fa icon	Page Name	PHP Page / URL	Description	Open New
fa fa-table				<input type="checkbox"/>
fa fa-table				<input type="checkbox"/>
fa fa-table				<input type="checkbox"/>
fa fa-table				<input type="checkbox"/>
fa fa-table				<input type="checkbox"/>

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.

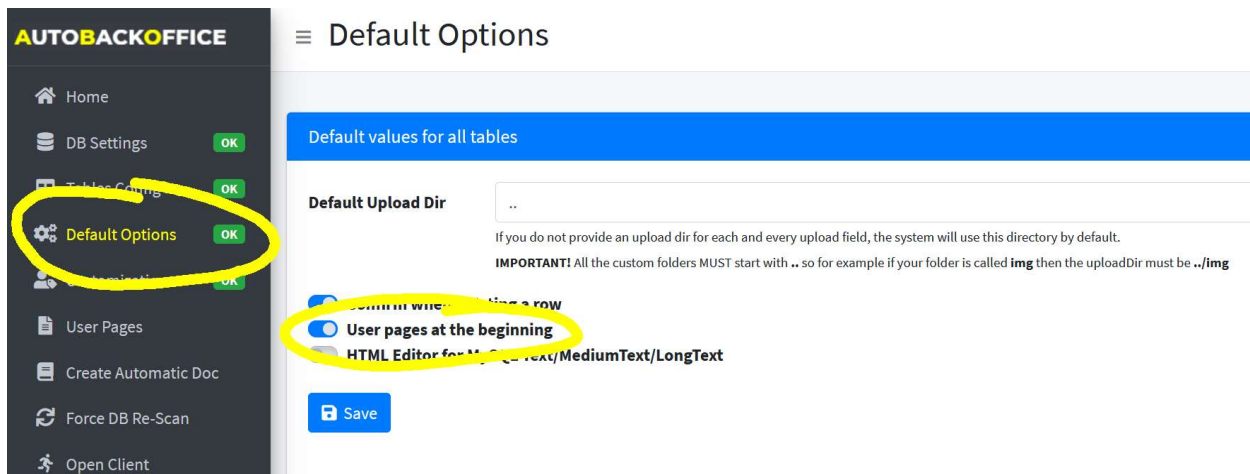
With these user pages, you can do exactly this: adding custom pages to the ABO client.

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¶m1=10¶mExample2=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.
- **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 `Description`). 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¶m1=10¶mExample2=123`

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” Bootstrap 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.

The screenshot shows the AUTOBACKOFFICE web application. On the left is a dark sidebar with the logo 'AUTOBACKOFFICE' at the top. Below the logo are several menu items: 'Home' (house icon), 'DB Settings' (database icon), 'Tables Config' (table icon), 'Default Options' (gear icon), 'Customization' (person icon), 'User Pages' (document icon), 'Create Automatic Doc' (document icon), 'Force DB Re-Scan' (refresh icon), and 'Open Client' (person icon). The 'Force DB Re-Scan' option is highlighted. The main content area has a white background. At the top, there is a blue header with a hamburger menu icon and the text 'Force DB Refresh'. Below this is a blue bar with the text 'DB Refresh Report'. The main content area contains the following text: 'DELETED tables / columns' followed by 'Nothing has been deleted from the last DB version.'; 'ADDED tables / columns' followed by 'Column **details (tinytext)** from table **items** has been added'; and 'COLUMNS which changed type' followed by 'No columns changed type.' At the bottom of this section is a blue button labeled 'Download Report'.

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.

Open Client

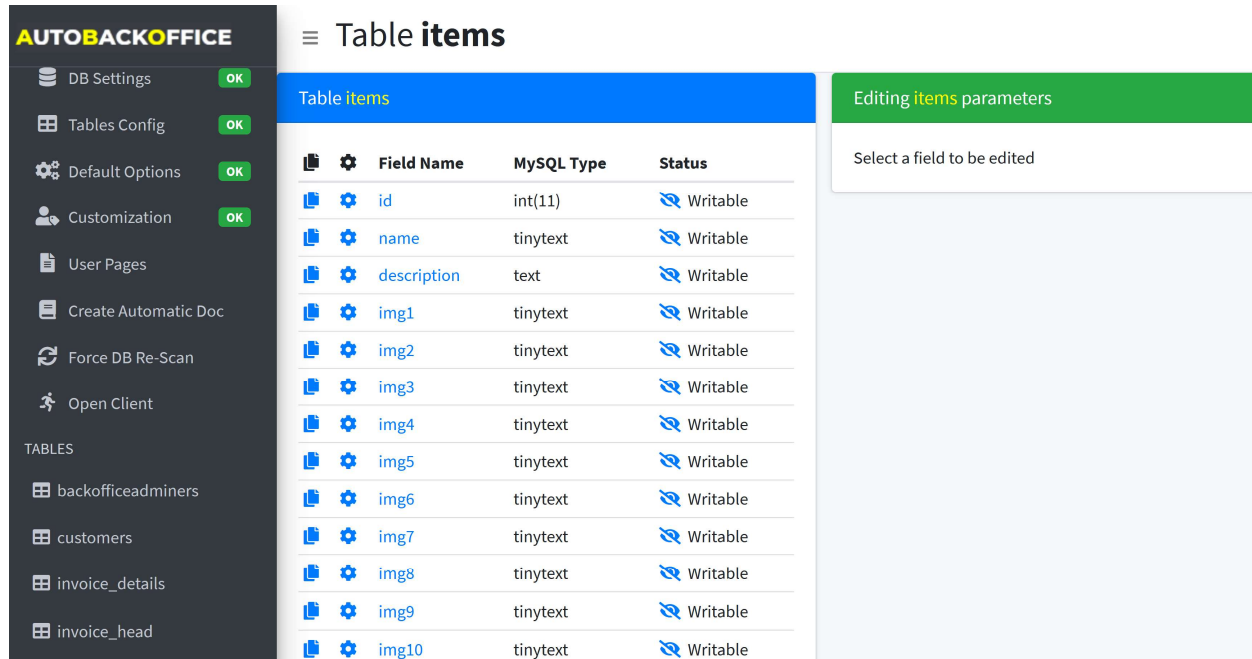
YouTube video: <https://www.youtube.com/watch?v=Bxokq0MYz4E&t=315s>

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

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.



AUTOBACKOFFICE

DB Settings OK

Tables Config OK

Default Options OK

Customization OK

User Pages

Create Automatic Doc

Force DB Re-Scan

Open Client

TABLES

- backofficeadmins
- customers
- invoice_details
- invoice_head

Table items

	Field Name	MySQL Type	Status
	id	int(11)	Writable
	name	tinytext	Writable
	description	text	Writable
	img1	tinytext	Writable
	img2	tinytext	Writable
	img3	tinytext	Writable
	img4	tinytext	Writable
	img5	tinytext	Writable
	img6	tinytext	Writable
	img7	tinytext	Writable
	img8	tinytext	Writable
	img9	tinytext	Writable
	img10	tinytext	Writable

Editing items parameters

Select a field to be edited

With each field, you can perform 4 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 on the eye 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

Table **items**

		Field Name	MySQL Type	Status
		id	int(11)	Writable
		name	tinytext	Writable
		description	text	Writable
		img1	tinytext	Writable
		img2	tinytext	Writable
		img3	tinytext	Writable
		img4	tinytext	Writable
		img5	tinytext	Writable
		img6	tinytext	Writable
		img7	tinytext	Writable
		img8	tinytext	Writable
		img9	tinytext	Writable
		img10	tinytext	Writable
		newField	tinytext	Writable
		details	tinytext	Writable

Editing **items** / id parameters

MySQL_Type

int(11)

View_Mode

Edit_Mode

Upload_Dir

Width

Height

Status

Writeable

Syntax_Check

Server

Mandatory

No

Link_To_Table

Lookup_Field

Description

Help

Save

Example to understand Link_To_Table & Lookup_Field:

```
SELECT name FROM item WHERE id = id
```

Link_To_Table:

item/name

Lookup_Field:

id

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

View_Mode: it can be one of the following:

- **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
- **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

Edit_Mode: it can be one of the following:

- **Textarea:** A HTML textarea will be used
- **Html:** A WYSISYG HTML editor will be used (<https://www.quilljs.com> until 2.3.0)
- **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

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

Width: This is the minimum width of the uploaded image. Any image smaller than this will be discarded

Height: This is the minimum height of the uploaded image. Any image smaller than this will be discarded

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

Syntax_check: not used (at least until 2.3.0)

Mandatory: not used (at least until 2.3.0)

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.