USERGUIDE
EMM

METRON-ENERGY MANAGEMENT MODULE (JOOL)

POWERED BY METRON

Laurent HANET

Version 1.03.6 - 15/05/2021

Table of contents

"EMM (JOOL)" is a program that allows the collection, organisation and processing of a large amount of energy-related data in order to derive useful information. All intellectual rights related to the software, its illustrations and documentation are the exclusive property of Dapesco.

Using EMM (JOOL)

This manual covers how to use EMM (JOOL) to create entities, counters and trees, inject useful properties, events and invoices, and develop relevant analyses and reports.

It follows on from the introductory manual "**Discovering EMM (JOOL)**" and will reuse many of the concepts introduced there. It is therefore advisable to master the concepts developed in the introductory manual before embarking on this advanced user manual.

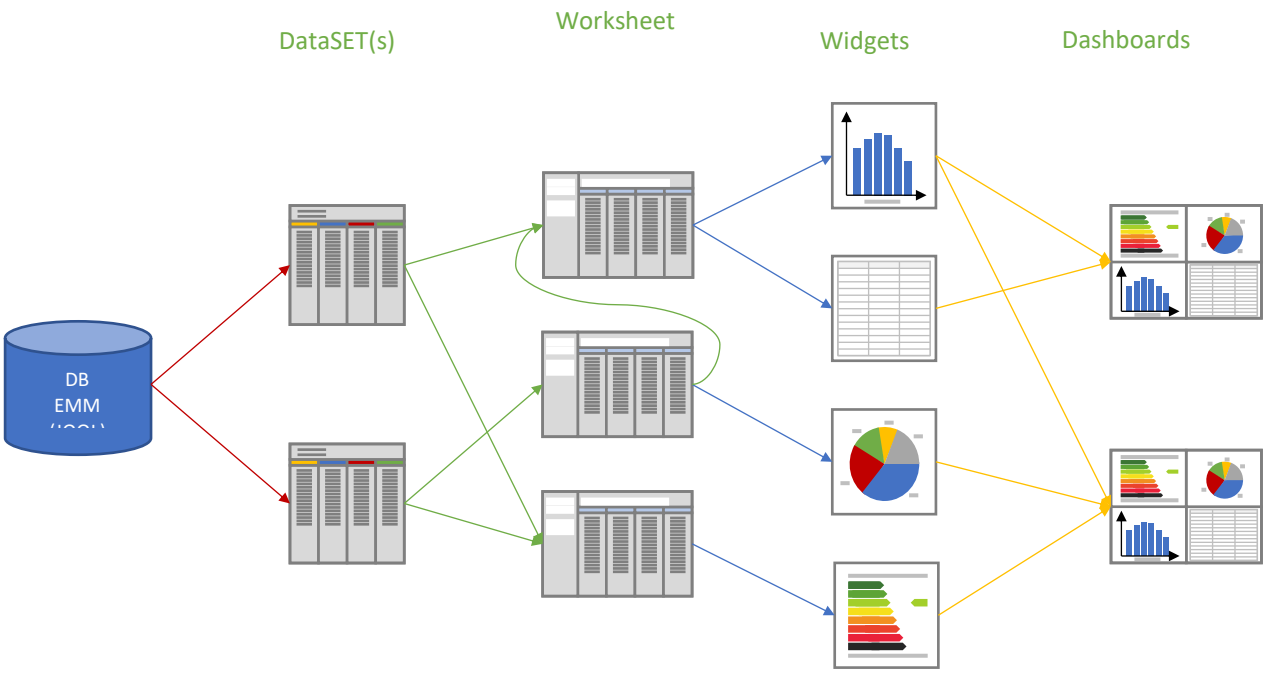
In the same vein, this manual discusses many tools that require the use of EMM (JOOL) syntax in order to function optimally. The "**EMM (JOOL) Syntax**" manual is therefore recommended reading in order to master all aspects of EMM (JOOL) detailed in this manual.

The informed reader will notice slight redundancies between these different manuals, but these will usually be on crucial points that need repeating.

1. General working structure in EMM (JOOL)

The working method in EMM (JOOL) is broken down into several levels, from the simple association of pre-built blocks to compose dashboards, to the definition of tables of values extracted from the database.

These steps can be summarised as follows:



The first level, the **DataSet** calculation, is an extraction of the database to retrieve the data profiles and/or properties stored there. Very few calculations are performed here, at most an aggregation of the data or a shift to shift them in time.

Worksheets then use these DataSets, the live DB, or other worksheets to perform all the preparatory calculations and formatting of the data.

Once the data tables have been finalised in the spreadsheets, the numerical results can be retrieved and displayed in the **Widgets**. Widgets are simple modules for displaying data from the spreadsheets, and no calculations are performed in them. **Widgets** can take many forms, such as graphs, gauges, energy labels, or simple data tables.

Finally, at the last level, it is possible to compose the widgets together to create **dashboards** or PDF reports sent by e-mail. Again, no calculations are performed at this level, it is simply a matter of composition and layout.

The advantage of subdividing the procedure in this way is that it is possible to split the rights on the basis of these levels, depending on the user's skills.

- Simple visualization of dashboards.
- Composition of own dashboards and reports based on existing widgets.
- Construction of widgets based on existing spreadsheets
- Creation/Modification of spreadsheets based on existing DataSets (advanced user)
- Creation of DataSets to extract useful information from the database.

2. Entity / Counter / Channel / Tree

A. Definitions and concepts

In EMM (JOOL), the surveyed area is divided into several object types.

An **entity** is an element of the tree structure that does not contain its own data flow. A site, a building or a CPE group can be an entity.
For ease of use, the term "**entity**" is often used (abusively) to encompass the entities themselves (sites, buildings, etc.) and the counting points.

A **metering point**, or **counter**, is an element of the tree structure on which values are measured, whether it be consumption or otherwise. It can be a physical counter, a temperature sensor, or even a calculated counter.

A **channel is** a part of a metering point that contains a data profile.
A **channel** is always included at a metering point but a metering point can have several **channels**, for example a dual time meter that would contain a peak and an off-peak channel.
The data profiles stored in the channels can consist of remotely collected data, manually collected data, data from invoices, or purely calculated data from other channels.
A **type is** a sub-category of an object, allowing it to be easily classified. Entities, meters, channels, logical links... all have associated **types**. For example, we can have entities of **type** "SITE", or channels

of **type** "HP" (for Peak Hours). It should be noted that certain **types** can be very generic, and one will then obtain meters of **type** "meter". Be careful in these cases not to confuse the terms used.

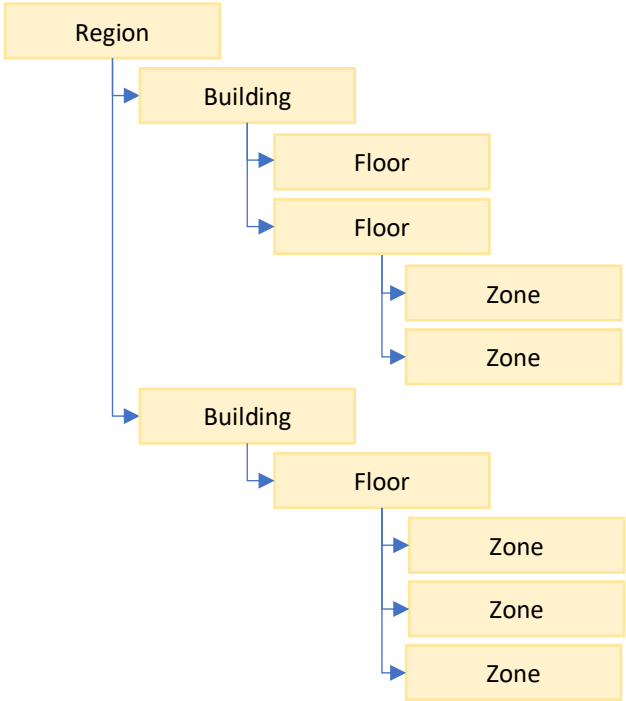
The **tree structure** is the structure containing and linking all the entities and count points in a client's database.
The **tree structure is** the actual structure, based on the kinship links between entities or counters, and the logical links of any kind that may connect entities and counters to each other.

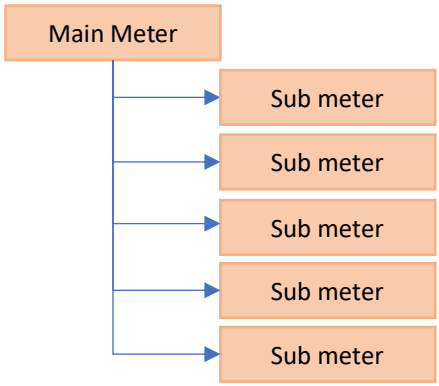
Views, on the other hand, are overlays of the tree structure, which can take different forms, defined by the EMM (JOOL) administrator. They will be used to customise the navigation of users by displaying entities and counters in an order of their choice, sometimes without any direct relation to the real tree structure.

B. Practical rules for implementation

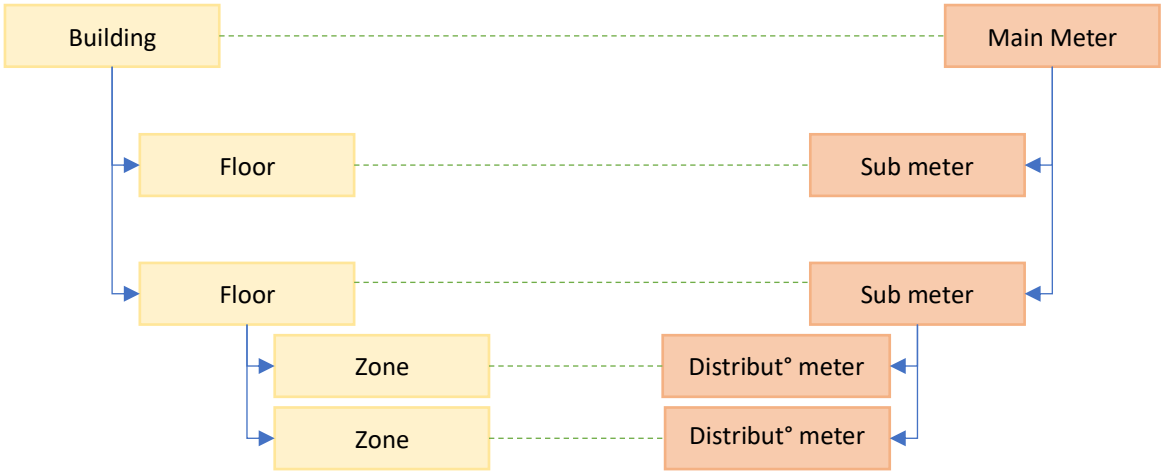
For the sake of standardisation and simplification, the tree structure in EMM (JOOL) should follow a few rules:

The links of relationship are only between elements of the same nature. Thus, an entity will only be related to entities, and a counter will only be related to counters. The interest is to be able to create a double tree structure, one linking the entities together, each entity containing its daughter entities (for example: a building will be the parent of the floors constituting it), and the other tree structure linking the counters together, thus constituting a counting plan, each sub-counter being the child of its associated main counter.

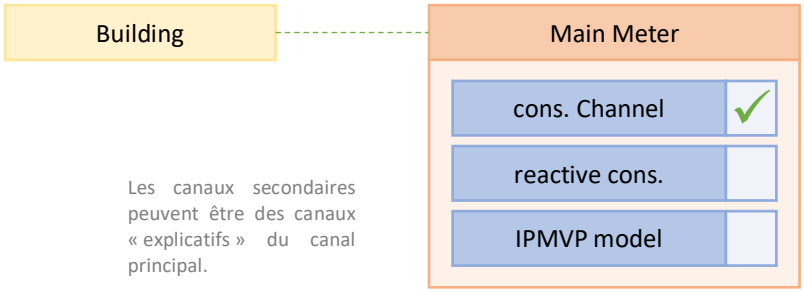


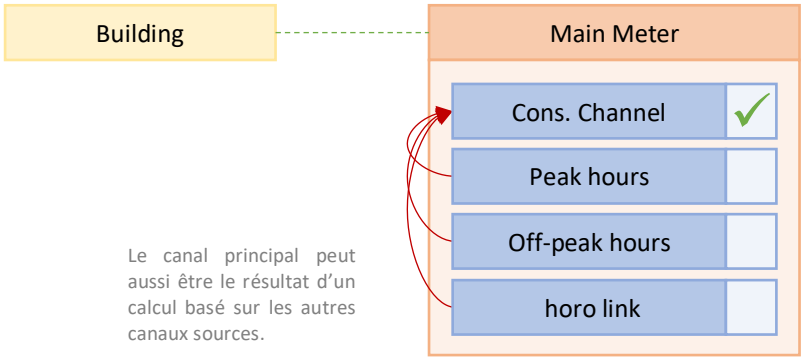


The entities and their corresponding meters are linked by logical links of type "ENTITY_METER", so that the consumption associated with an entity can be retrieved at any time by retrieving the data from the meter linked to it.



Each metering point contains a main measurement channel and possibly one or more associated channels. The idea is not to have one big metering point that groups all the channels in the building, but to have one meter per important measurement, and the explanatory channels included in that same metering point.

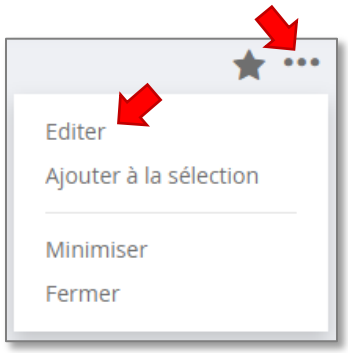




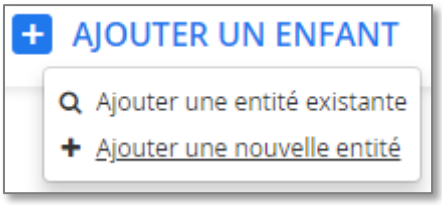
C. Manual creation of entities

In addition to the bulk import procedure (seen in another chapter), whether for entities themselves or for counters, it is possible to create a new entity/counter manually by going through the file of an existing entity.

- Open an existing entity record
- Switch to edit mode via the options at the top right of the form

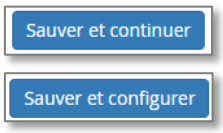


- Once in edit mode, click on the "Add a child" button and choose "Add a new entity".



- A creation window appears, allowing you to define the type, name and reference of the new entity.

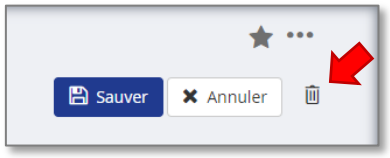
- You can then click on "Save and continue". The new entity will be created, and the active entity file will remain. If "Save and Configure" is chosen, the new entity will be created and its entity file will be opened in the workspace for configuration.



It is then possible to completely configure the new entity by editing its form to associate properties and links to it (it is also possible to cut the link with the entity that was used to create it, if necessary, and to recreate the useful links).

D. Manual deletion of entities

When you are in edit mode on an entity/counter record, you can also delete it by clicking on the small bin icon at the top right of the record.

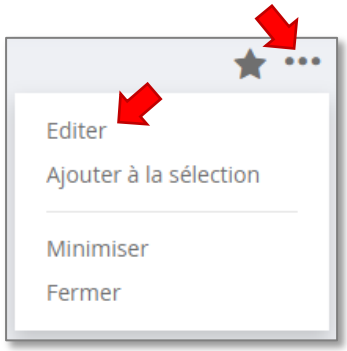


Warning: an erasure is irreversible. Always be careful not to delete anything inadvertently.

3. The entity sheet in edit mode

The record of an entity/counter is used to centralise and display all its characteristics. It is accessible from the channel selector (bottom left of the EMM (JOOL) interface) via the info button or via the search button by choosing a search result.

Once on the record of an entity, it is possible to switch to editing mode in order to modify all its characteristics, via the options at the top right of the record.



In edit mode, the entity sheet keeps the same general structure (tabs, frames...) but many options appear to encode the desired modifications.

Important: going into edit mode is only possible for users who have sufficient rights to the objects to be modified (see the expert's guide to rights management). In the same way, the modifications that it will be possible to make in edit mode will be limited for each user according to the rights that have been assigned to them. In the remainder of this manual, we will consider that the user connected is an administrator with full rights over the entire database.

A. General information

In edit mode, the general information of the entity (name, reference, type, icon...) becomes modifiable fields. Whatever the displayed tab of the entity record, the general information banner always remains visible at the top of the record.

a) Change name, reference and type

Simply click on the name or reference at the top left of the entity record to see a cursor appear to modify it.

A small planet icon next to the name opens a pop-up to encode the names in different languages if required.

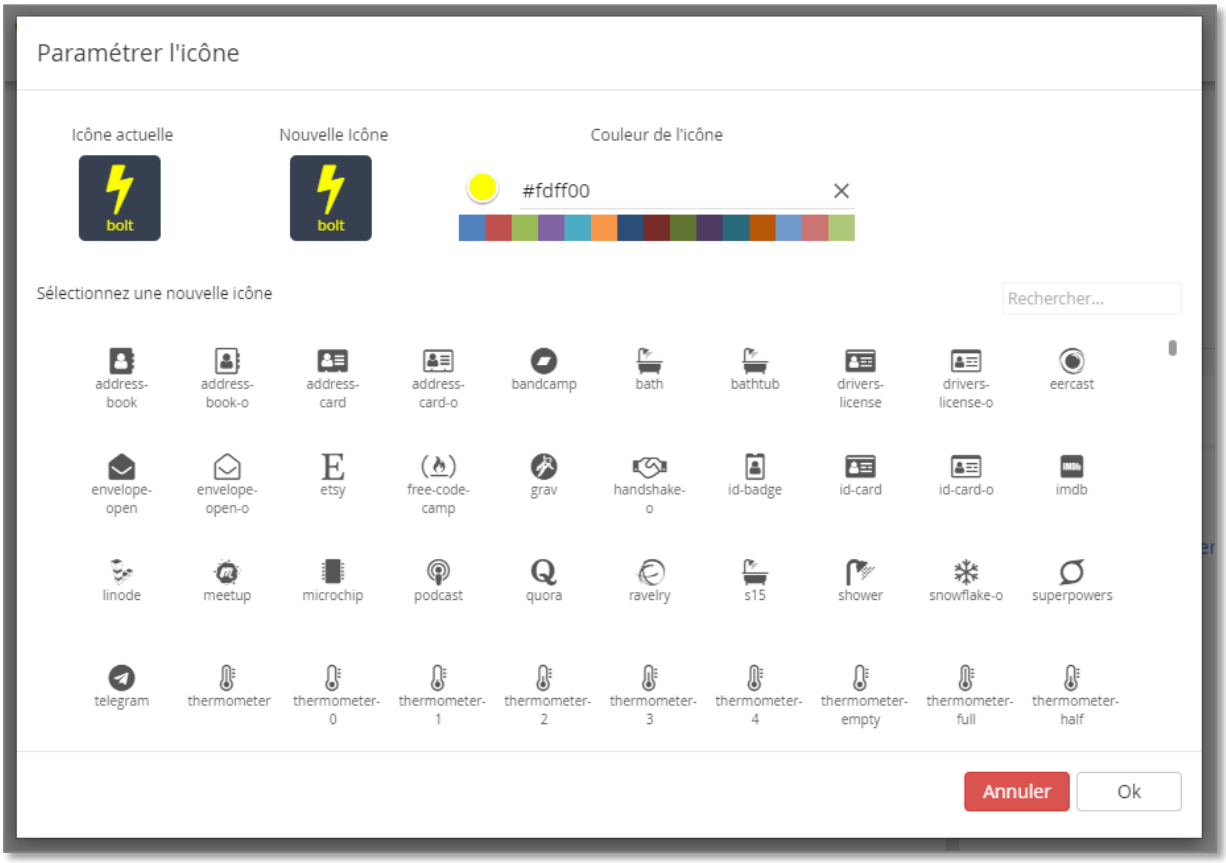
Note: In general, a small planet icon will always be found near the editable fields of multilingual texts to allow editing of texts in other supported languages.

In the same way, the entity type is now listed in a drop-down menu allowing to change its value among the other existing types in the database.

Warning: the type of an entity determines the list of property blocks to which it has access. Changing the type of an entity will therefore modify this list of blocks. In the case where a property block of an entity is filled with values and the entity loses access to this type of block following a type change, the values stored in the properties of the lost block will be lost.

b) Change the icon

Still in edit mode, the icon at the top of the form can be clicked to open the icon choice interface where you can change the design and colour.



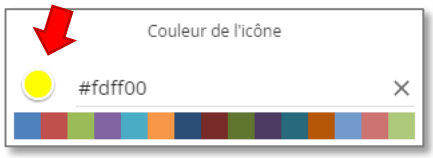
The choice of the new design will be made by clicking on the desired icon in the lower part of the pop-up (a search field is provided in the upper right-hand corner to filter the icons if necessary).

The choice of colour can be made :

- by clicking on one of the proposed colours under the colour field

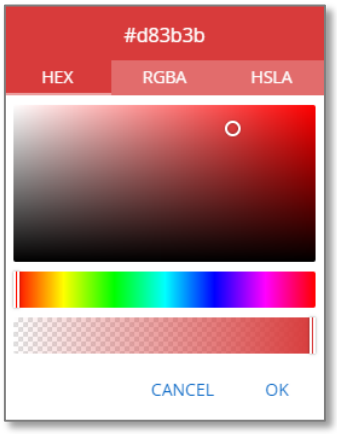
- by directly entering the hexadecimal code of the colour in the field to the right of the icons

- by clicking on the coloured disc, which will open a full palette for choosing a colour.



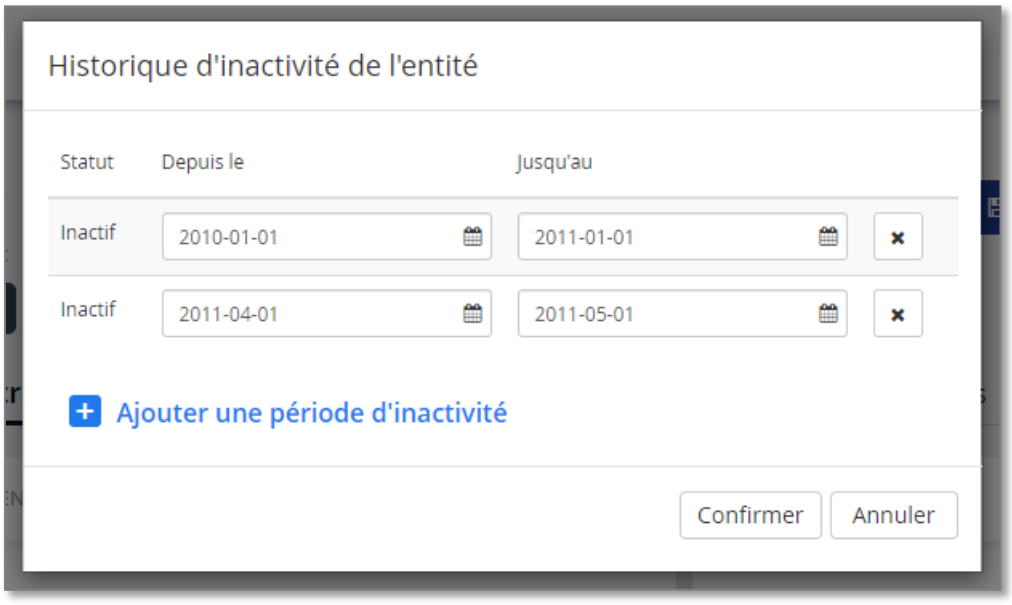
In the palette, the 3 buttons on top allow you to display and modify the colour code in hexadecimal, RGBA or HSLA form.

The choice of colour can also be made by selecting a hue in the colour bar, then finely choosing the saturation and luminosity in the colour block. The bar at the bottom of the palette allows you to define the transparency of the colour.



c) Activate / Deactivate the entity/counter

Also in the header of the entity/counter record, the Status button allows the counter to be activated or deactivated. This button will open a pop-up window to define the periods of inactivity of the counter, which can then be used in the analysis tools.

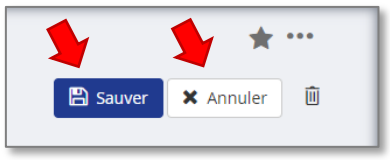


Inactivity periods are defined by their start and end dates, and it is possible to delete an inactivity period with the small cross to the right of the corresponding line.

Note that the button itself in the header of the entity/counter record will display "Active" or "Inactive" depending on the current status of the entity/counter (based on "now").

d) Save your changes

Once the desired changes have been made, you can Cancel or Save the changes made to the entity record via the "Cancel" or "Save" buttons at the top right of the entity record.



B. Structure" tab in edit mode

In the "Structure" tab, you can modify the entity's logical or parentage links.

a) Cutting an existing link

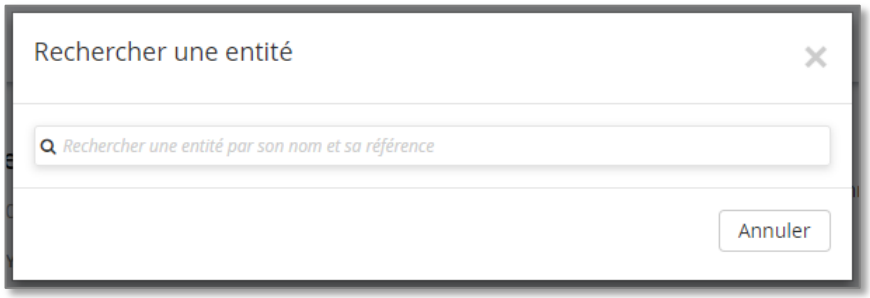
In edit mode, there will be an 'x' symbol to the right of each entity or counter present in the record. Clicking on the button of one of the entities will break the link between the entity in question and the entity whose record is open.

b) Changing parents

In edit mode, a "Change Parent" button appears at the top right of the parent entities frame.

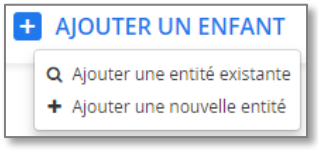
Clicking on it opens a window that will allow you to find the entity you want to take as parent.

The newly selected parent will then replace the current parent.



c) Add a child

In edit mode, an "Add Child" button appears at the top right of the child entities frame. This button will open a list with two choices: "Add an existing entity" or "Add a new entity".



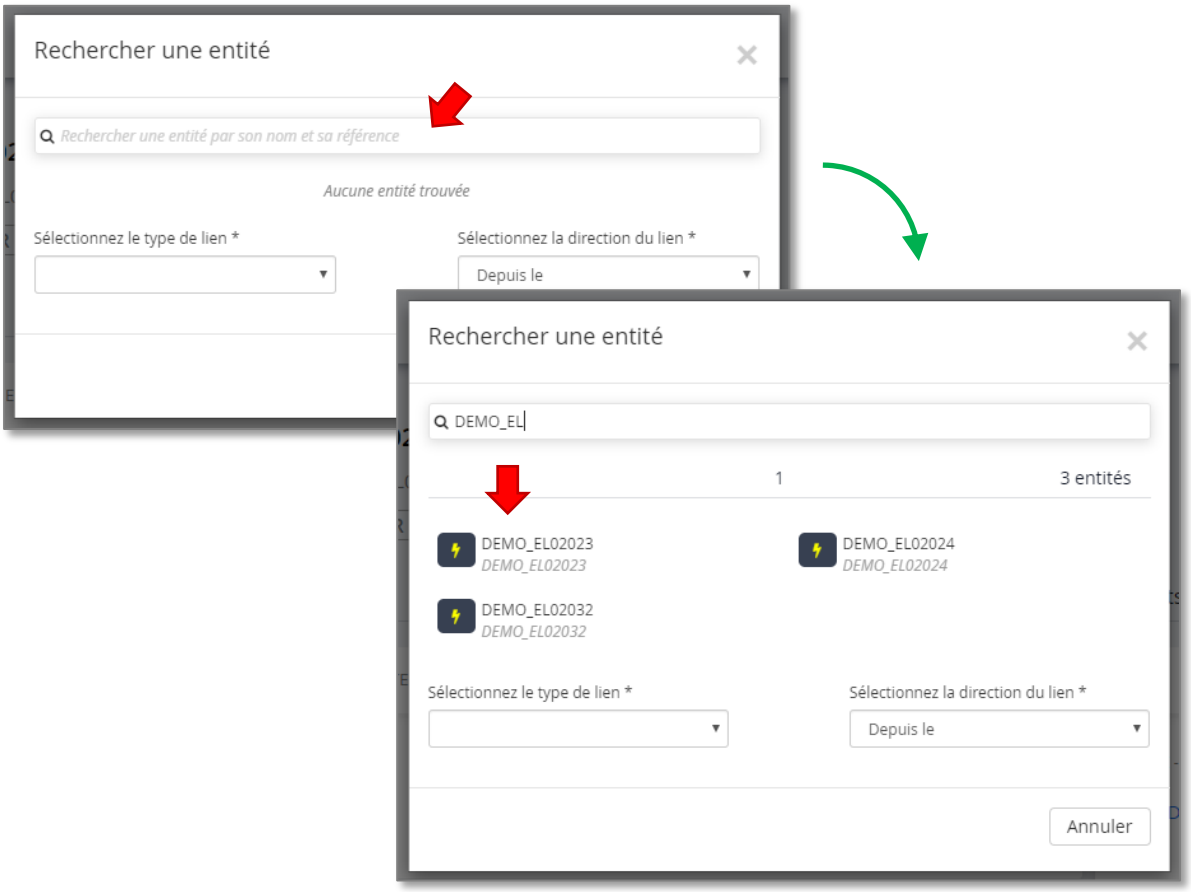
Adding an existing entity will open a selection window to find the desired entity, and the entity then chosen will be added as a child. This entity will then lose its current parent to the active entity, since an entity can only ever have one parent.

Adding a new entity will open another window. The type (from among the existing entity types and counters) and the name and reference of the new entity must be defined here.

The 'Save and Continue' button will save the new child and return to the entity record being edited, while the 'Save and Configure' button will save the new child and display its entity record for configuration.

d) Add a link

In edit mode, an "Add a link" button appears at the top right of the linked entities frame. This button opens a dialog box allowing to find an existing entity via a search field. Once the search has been carried out, you can select a result to add it as a link of the type chosen (link type selector at the bottom left) and in the direction chosen (selector at the bottom right).

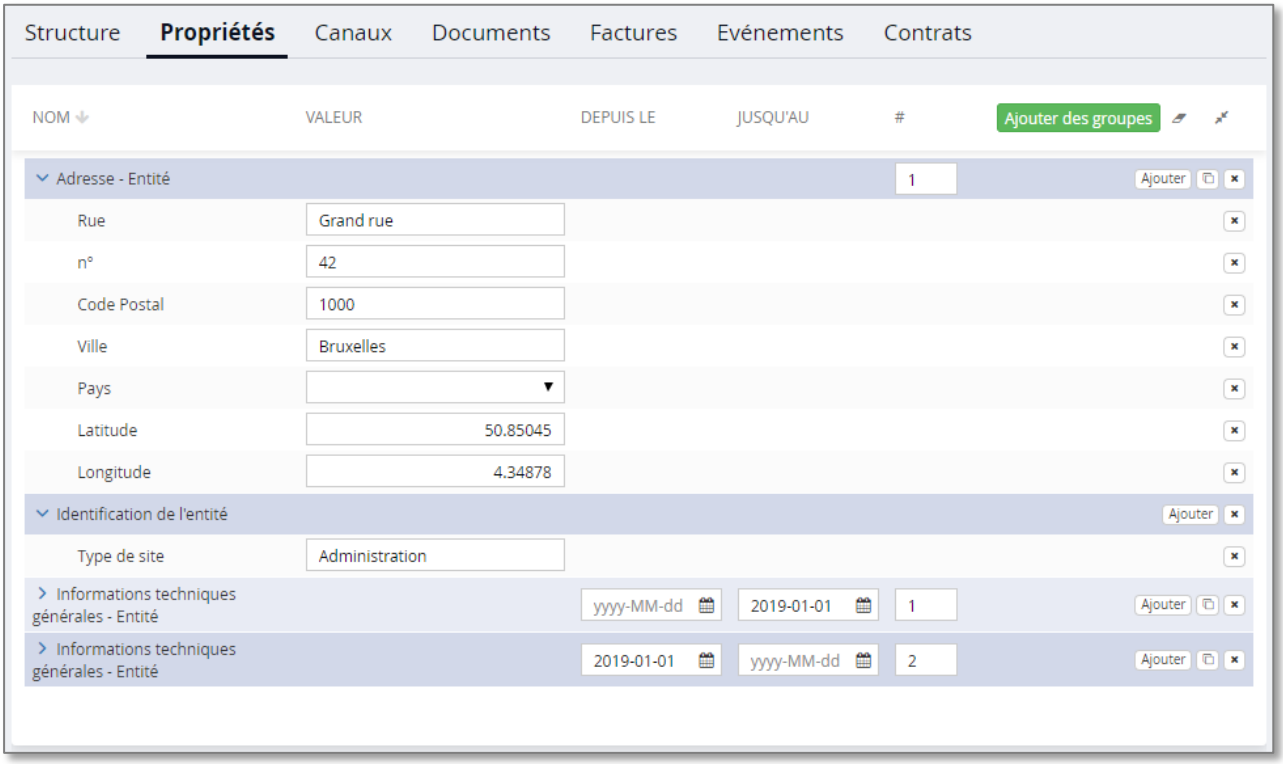


As regards the direction of the link to be created, if we are on the entity sheet and we want to link a counter to it, in the case of an ENTITY_METER link, we must choose "from".

If you are on the record of the object on side "1" of the link "1:N", you must therefore choose "from" to attach another object on side "N".

C. Properties" tab in edit mode

In edit mode, each property value becomes an editable field. In the same way, the start and end dates of historical blocks are editable (by entering the date manually or via a visual calendar), as are the IDs of multiple or historical blocks.



For ease of use, the button with the arrows at the top right allows you to expand all the property blocks or minimize them all at once.

If you move the mouse cursor over a property name or the title of the property block, a tooltip shows you the reference of the property or the block. This can be useful for retrieving the reference of a property that you want to use in a spreadsheet for example, without having to go and find it in the property configurator.

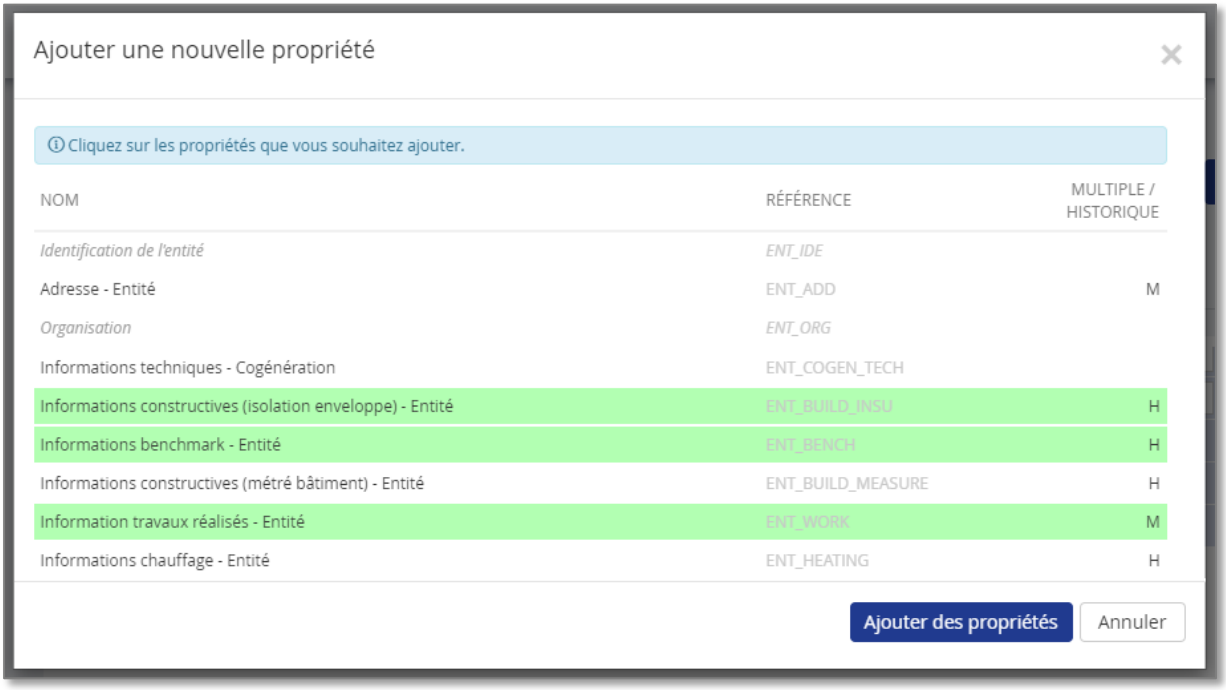
Note also that the coloured bands containing the titles of the historicised property blocks are in a lighter colour if they relate to a non-current period. Historical property blocks with currently valid values will have the same colour as non-historical blocks.

a) Adding property blocks

In edit mode, the "Add groups" button appears at the top right of the frame, allowing you to add a property block, among the

blocks preconfigured for this type of entity (by the EMM (JOOL) administrator).

This button opens a pop-up window containing the list of available property blocks for the entity type in question.



Unique blocks already associated with the entity are greyed out (a second instance cannot be added).

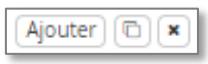
Multiple or historical blocks can be added in several copies and are therefore not greyed out.

The Multiple/Historical column may contain 'M' to indicate a multiple block, or 'H' for a historical block.

In this pop-up, one or more property blocks can be selected, which will then be added to the property sheet of the entity being edited.

b) Adding properties

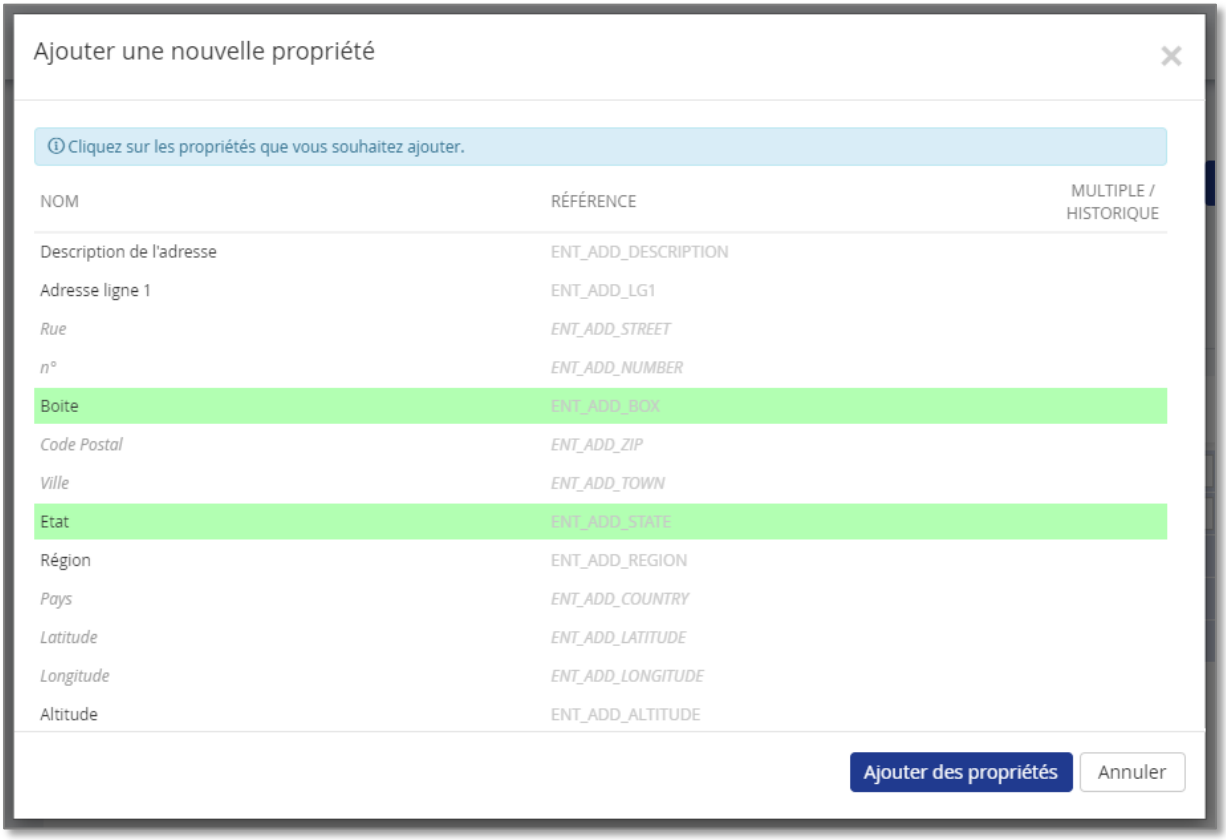
Within each property block, several buttons also appear to the right of the block name. The "Add" button allows you to add properties to the block in question.



A "Delete" button allows you to delete the entire block at once.

Between these two buttons appears a specific button for multiple or historicised blocks, allowing to duplicate this block of properties directly (without having to go through adding a block).

Clicking on the "Add" button will open the following popup, allowing you to choose the properties to be added to the existing block.

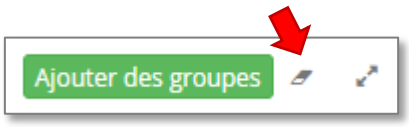


As for the blocks, the properties (unique in a block) already filled in on the entity are grayed out so as not to be duplicated. It is possible to choose one or more properties, which will then be added to the relevant block of the entity record being edited.

c) Cleaning up empty properties

In edit mode, when a new property block is created, all the properties associated with it are created in the form. Often, not all of them are filled in, and most of the time, many properties will remain empty.

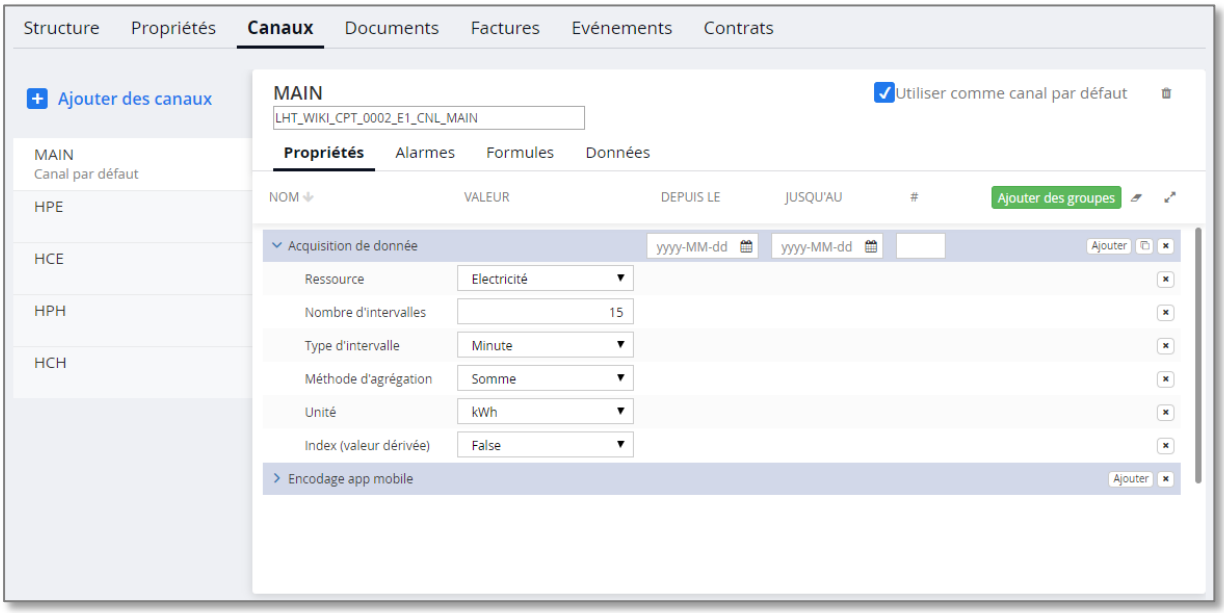
In order not to clutter up the entity/counter sheets, it is useful to be able to remove empty properties. This can be done manually by clicking on the small "x" button next to each property, or you can use the clean-up tool at the top right of the form.



It will of course always be possible to recover the properties thus removed by simply clicking on the add property button, seen in the previous paragraph.

D. Channels" tab in edit mode

The "**Properties**" sub-tab of the "Channels" tab in edit mode functions globally like the entity/counter properties tab, for each page associated with each source channel.



An additional button appears on the left, above the channel list, allowing you to create a new source channel associated with this counter. This button will then open a pop-up window allowing you to choose the type of source channel to create.

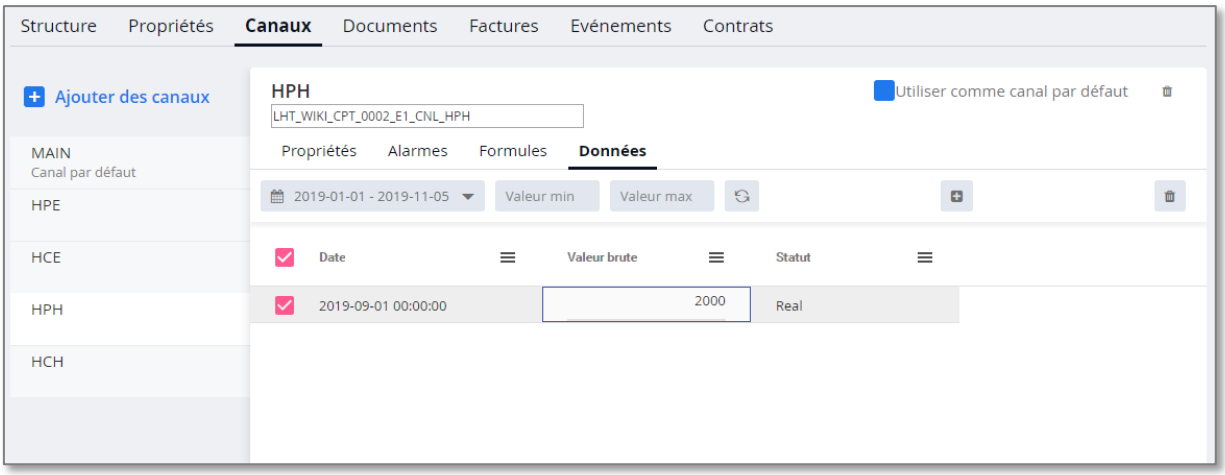
Also new compared to the "Properties" tab, two additional options appear at the top right of the form, allowing the active channel to be marked as the new default channel, and a small bin icon to delete the source channel if necessary.

In addition to these specificities, the interface for managing channel properties is perfectly similar to that for managing entity/counter properties.

The "**Alarms**" sub-tab works in the same way but is limited to containing the properties associated with the configuration of alarms on this channel.

The "**Formulas**" sub-tab, once switched to edit mode, allows you to modify existing formulas, adapt their periods, and add/delete formulas if necessary.

Finally, the "**Data**" sub-tab allows you to correct the data present in the channel. Compared to the normal (non-editing) version, the date and value fields are now editable.



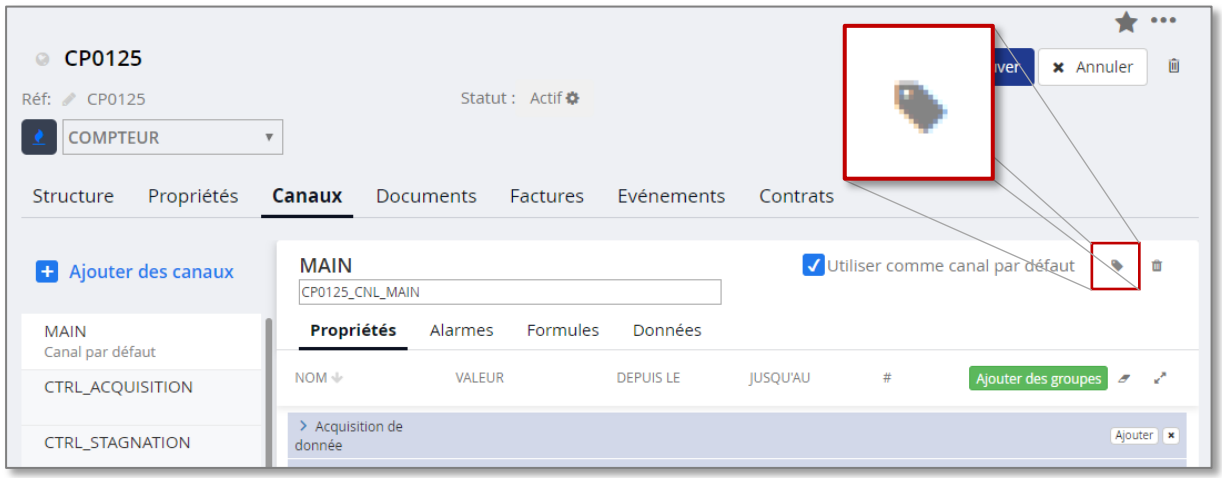
The value displayed in the "Status" column is not editable, but it will be adjusted according to the situation. "Real" indicates that the data is the one that has been entered or retrieved automatically by remote reading, and "Corrected" will indicate that this value has been corrected afterwards.

At the top right, the "+" and "Trash" buttons will respectively add a line of data, or delete the selected data (whose pink boxes on the far left are checked).

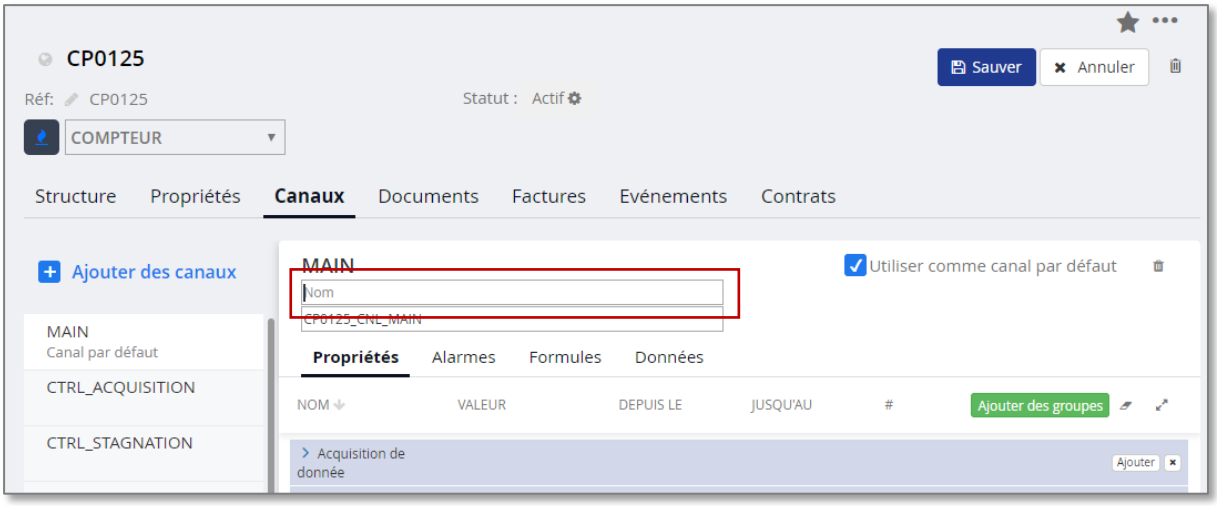
When a change has been made, "Save" and "Cancel" buttons will appear next to these "+" and "Trash" buttons so that you can save or cancel the current changes to the data.

a) Assign a name (optional) to a channel

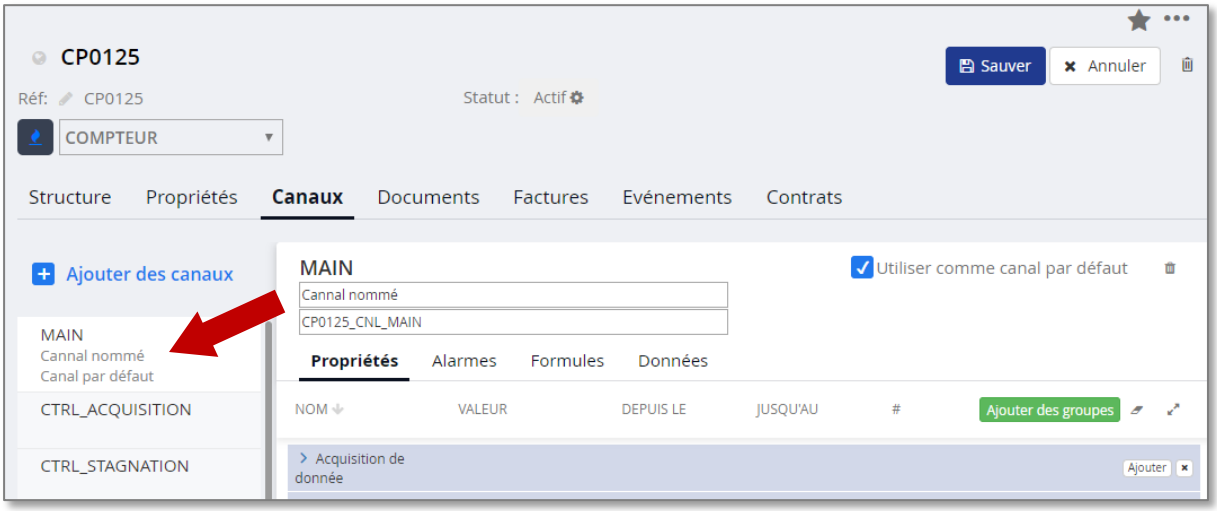
In edit mode, on the record of a channel that does not yet have a name, the following icon appears.



Clicking on this button brings up a new "name" field between the channel type and reference.

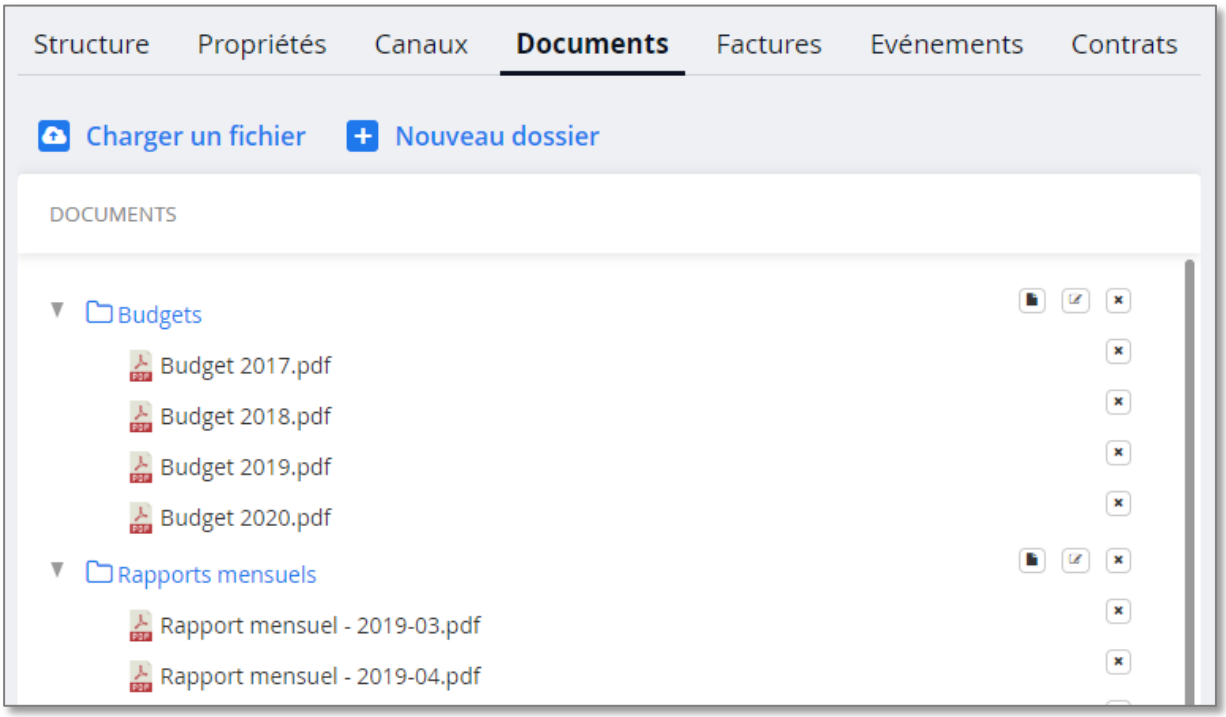


When filled in, this name will be displayed just below the type in the meter's channel list.



The main interest of this feature is to be able to easily differentiate between several channels that would have the same type in the channel list of a meter, without having to display them all one by one. These names will also come in handy in the mobile application to differentiate between different channels that may have the same type.

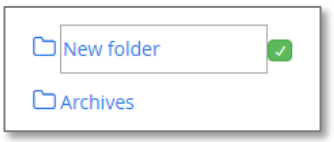
E. Documents" tab in edit mode



The "Upload File" button opens an upload interface allowing you to upload a file to the entity's root folder.

The "New Folder" button creates a new folder in the disk space associated with the entity. Note that it is not possible to nest multiple levels of folders within each other.

Once a new folder has been created, you can give it a name and then click on the little green button next to its name to confirm the creation of the folder.



To the right of each folder name, there are 3 buttons. The first one allows you to insert a file in the folder in question. The second allows you to rename the folder, and the last one allows you to delete the entire folder.

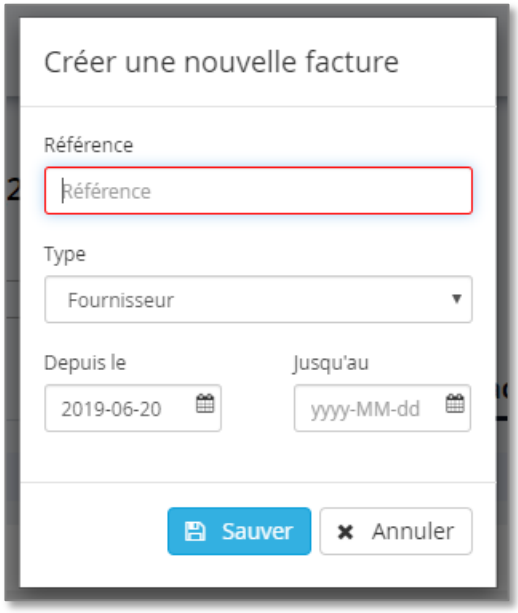
Finally, a button appears to the right of each file name, so that you can delete the file.

F. Invoices" tab in edit mode / Editing an invoice

The "Invoices" tab changes very little in edit mode. We just see the "Create a new invoice" button which opens a pop-up window allowing the creation of a new invoice associated to the active entity.

To create a new invoice, you will need to give it a reference, a type and start and end dates.

Once this information has been validated, you enter the newly created invoice file, in edit mode.



The invoice form in edit mode has exactly the same structure as the invoice form in view mode, except that all the fields can now be edited and property blocks or properties can be added to them, in exactly the same way as for all the property management interfaces in EMM (JOOL) (entity, channel, user, invoice, event properties, etc.)

G. Events" tab in edit mode / Editing an Event

The "Events" tab also changes very little in edit mode. We just see the "Create a new event" button which opens a pop-up allowing the creation of a new event associated with the active entity.

m^3

To create a new event, we must give it a reference (pre-filled by EMM (JOOL), given the very relative usefulness of references for events), a type (to be chosen among the existing types in a drop-down menu), a start date (mandatory) and an end date (optional).

Once this information has been validated, EMM (JOOL) creates the new event and opens its event form in edit mode.

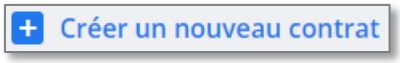
The event form in edit mode is similar to its version in view mode except that each of its fields are now editable, and properties and property blocks associated with this event can be added/modified/deleted using the usual property management interface.

Note: The property blocks available for each event type are limited to those that have been configured to be accessible to it. This configuration is the responsibility of the EMM (JOOL) administrator (see Expert User's Guide: "Administering EMM (JOOL)").

H. Contracts" tab in edit mode / Editing a Contract

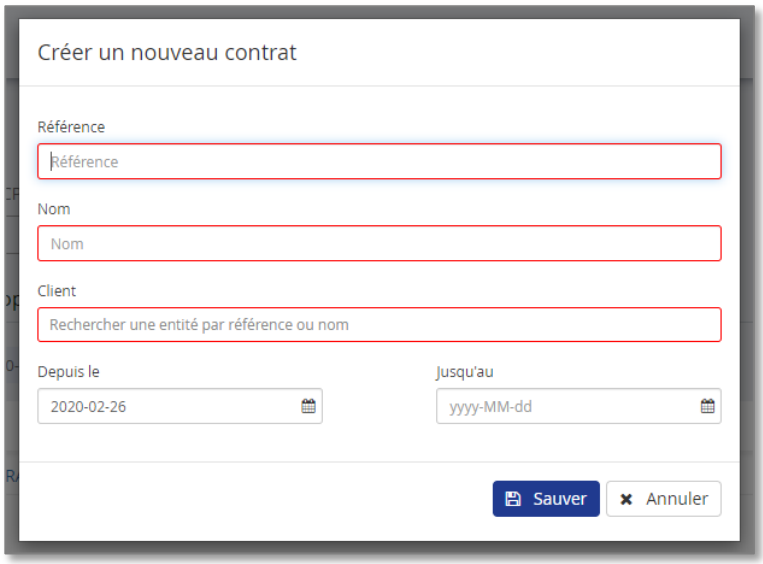
The creation and configuration of contracts is an advanced subject, and an expert's booklet is specifically dedicated to it. Here is only a quick summary of this functionality to allow advanced users to understand what it is about and possibly to modify some of the contractual information. For a better understanding of the details of how contracts work in EMM (JOOL), the expert's booklet "Tenants Module" is highly recommended.

There is also very little difference between the "Contracts" tab in view mode and in edit mode. The only differences are the appearance of a cross button at the end of each contract line, allowing the contract in question to be deleted, and the appearance of a "Create new contract" button which will open a pop-up allowing the creation of a contract linked to the open entity.



To create a new contract, it must be given a reference, a name, a start date (mandatory) and an end date (optional).

We also need to link it to an existing client (so we need to have pre-configured "client" entities in the database) and this is done via a search field where we can type in a few letters, and EMM (JOOL) will suggest the corresponding clients available in the database.



Once this information has been validated, EMM (JOOL) creates the new contract and opens its file in edit mode.

The contract record in edit mode is similar to the view version, except that new buttons appear to add/edit/delete rental units, allocation methods, and billing methods.

Attention: To create a contract correctly from the start, it is essential to follow these steps in order

0. The database must **first** contain at least one entity type that has been marked as "**Client**". Entity types are created by administrators, and on creation they have the option of marking them as "Clients" which can be associated with contracts.

0. If you want to add a billing method, its structure must be pre-created in the configurator by an administrator. A **billing method** is a block of specifically dedicated properties, which will then be injected into the contract records.

1. From the record of an entity, a counter, or even a customer, one must then switch to edit mode and **create a new contract** in the "Contracts" tab. It should be noted that if you start from the record of an entity marked "Customer" to create a contract, the "Customer" field of the creation pop-up will be pre-filled with the reference of the customer concerned.

2. Once the contract exists, it must be **associated with zones**. To do this, in the contract form in edit mode, click on the "Zones" button at the top of the form, and add the zones (Entities) that you want to be covered by the contract.

3. Once the fields have been defined, **a new rental unit** can be **added**. It is given a name and a reference and the corresponding tab appears in the lower part of the form, containing a box for allocation methods and another for billing methods.

4. In the allocation methods box, you can click on the "+" at the top right to **add an allocation method**.

a. The pop-up window that opens asks you to choose on which zone, and in this zone, on which counter(s) the method should be applied. If the "zone" field remains empty, it means that you have passed step 2 and that no zone is associated with the contract. To select several counters at once, you can hold down the "shift" key (to select a series of counters at once) or the "ctrl" key (to select several counters from the list).
b. Once the zones and counters have been chosen, the allocation method can be chosen from the pop-up menu, and depending on the method chosen, other fields will appear allowing the method to be configured (providing it with the parameters it needs).
c. The method can also be given start and end dates, bearing in mind that the validity periods will in any case be limited to the validity of the contract.

5. In the billing methods box, the "+" at the top right can be clicked to **add a billing method**. A drop-down menu is available, listing all pre-existing billing methods in the database (see step 0bis). Once a method has been chosen, new fields appear in the pop-up allowing the configuration of the property values associated with the chosen method.
   As with the allocation methods, validity periods can also be assigned to the billing methods, and these will also be truncated to the validity period of the contract itself.

4. Massive imports

The notion of massive import consists in a massive injection of data into EMM (JOOL) via a dedicated interface, rather than having to inject them one by one manually. This data can be new entities with their properties, updates of existing entities, links between entities, reading data, billing data...

Massive imports work in **insert-update**, in other words, each imported element must be identifiable (via its reference for an entity, via its id for an invoice...). EMM (JOOL) will then check if the identified element already exists, in which case, it will update this element with the new imported values, and if not, EMM (JOOL) will create a new element based on the imported properties.

The insert-update principle does not normally allow for bulk deletion via the import procedure. It will be possible to delete properties in an entity by importing an empty value, but it will not be possible to delete the entire entity via the bulk import.

A. General procedure

Bulk imports into EMM (JOOL) will require the creation of a **preparatory file, for** example via Excel, in order to have a table of data to inject into the bulk import interface. This preparatory file should contain one row per item to be imported, and one column per value to be imported on each item.
Some columns will be essential and will be used to identify the element to be created (or to update it if it already exists) and the others, optional, will simply contain the properties or data to be imported.

a) Preparatory file

The preparatory import file will therefore be of this type (a real import file may of course have many more lines...):

	A	B	C	D	E	F	G	H
1								
2		Reference	Name	Type	Parent	icon	icon-color	surface
3		SITE_001	Premier site	ENTITY		building	#ffffff	1200
4		SITE_002	Deuxième site	ENTITY		building	#ffffff	1500
5		SITE_003	Troisième site	ENTITY		building	#ffffff	450
6		ELEC_001	Compteur Elec principal	METER	SITE_001	lightning	#ffaa22	
7								

In this example, there is one row per entity, with a "Reference" column which will be used to identify the entity to be created/updated, and "Name", "Type", "Parent"... columns which contain the data to import.

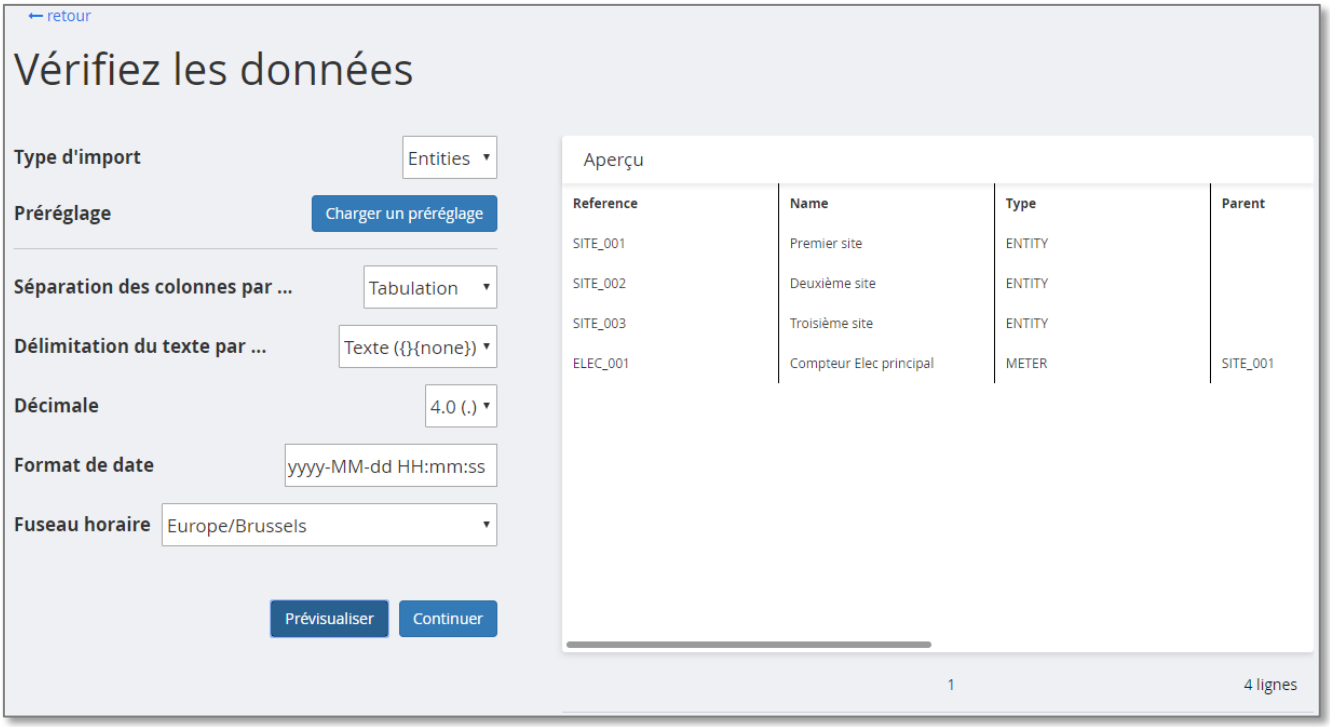
The EMM (JOOL) bulk import interface is accessible via the general import action button.
The following interface is then displayed:

The contents of the preparatory file must then be copied and pasted into the white zone of the interface. You can also drag and drop a CSV file directly onto this white area, or use the link at the bottom of the page to fetch an import file from somewhere on your hard drive.

Note: the display of imported data does not necessarily respect the alignment of the columns.

The "Continue" button then takes you to the next step.

b) Checking the data



This next page shows, on the left, a whole series of parameters to configure the import itself. You must therefore specify :

- The type of element to be imported: Entities, Source channels, Invoices, Data, Links between entities, users, events...

- The type of column separator in the import file
- The text delimiter in the import file format
- The decimal separator
- The date format (this is important for EMM (JOOL) to identify them correctly)
- The time zone

The "Load a preset" button allows you to reload a previously saved import configuration (so that you don't have to redo all the configuration work each time you import)

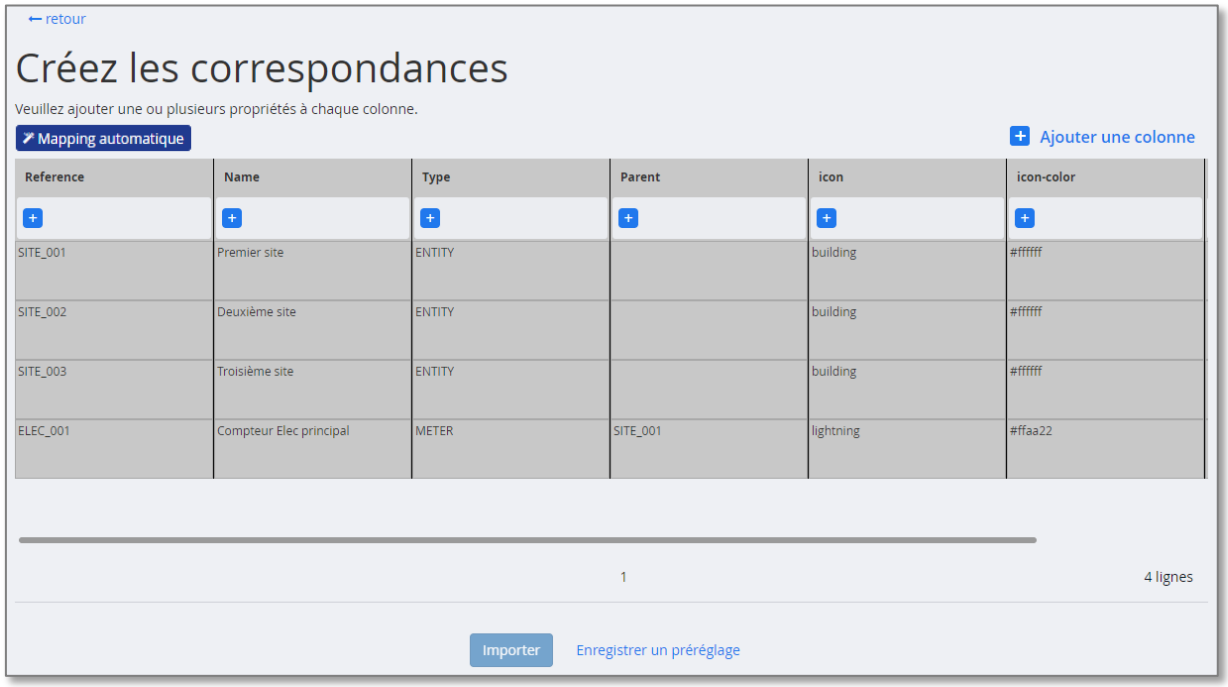
The "Preview" button displays the preparatory file in the right-hand area of the interface, formatted according to the chosen parameters.

Expert's tip: When previewing, it is important to check that each column has been identified by EMM (JOOL) as being in the expected format. The most common case is a date column identified as text because the declared date format is different from the one used in the preparatory file.

Once the parameters have been set, and the preview has been checked, the "Continue" button takes us to the next stage of the import.

c) Manual mapping

The mapping page includes all the columns of our preparatory file.



On this page, you can click on each box of the table to modify its content if necessary (last minute correction...). The box then changes colour to indicate that it has been manually modified.

Finally, it is on this page that one has the possibility to associate each column of the initial preparatory file with a field of the DB.

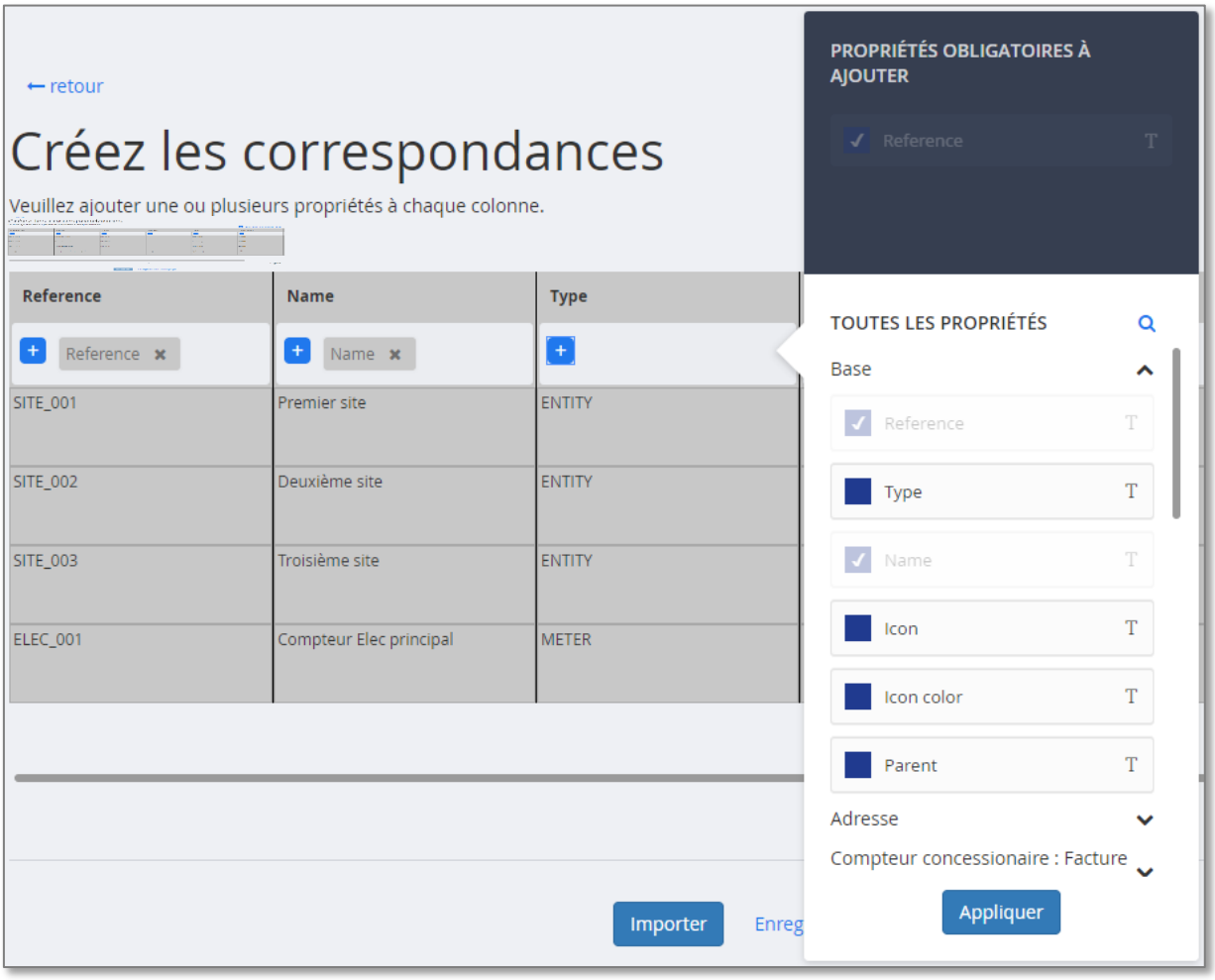
To do this, the small "+" buttons in each column will open the list of available fields (organised in blocks), and all you have to do is find the field to be associated with the column.

Some fields are compulsory (in order to be able to identify without error the element to be created/updated), and are found in the upper (and coloured) part of the list. The other fields are optional and are in the light part, at the bottom of the list.

In this list, you can then check the property to be associated with the column, and then move on to the next column.

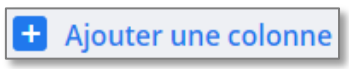
Properties already associated with a column are no longer reusable, and are greyed out in the list.

For each column already associated with a property, the name of the property will be found next to the column's "+" button, and this association can be removed if necessary.



Note: It is possible to associate the same column with several properties if necessary.

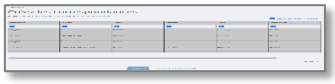
In the event that a column is forgotten in the preparatory file, a column can be created on the fly using the "Add a column" button. This button will generate an additional column in the import interface, as if it had been present in the preparatory file, and fill it with a constant value by default.



Reminder: in the mapping window, you can manually modify the data of each cell, simply by clicking on it.

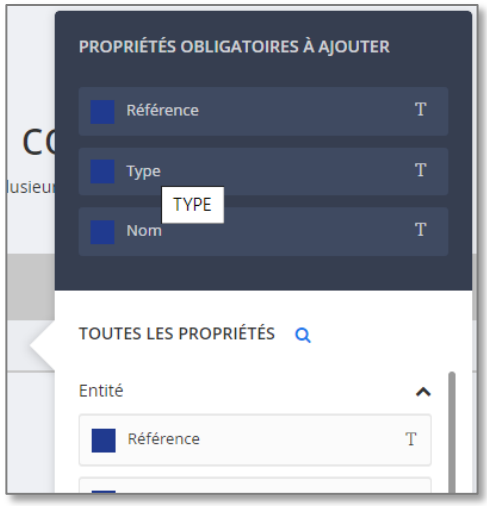
Once the mapping is complete, you can then save the complete configuration for future use, and thus avoid having to redo the same mapping operations each time you import. Finally, the "Import" button will perform the actual import, and will return a summary page of the import performed.

d) Automatic Mapping



If the preparatory file has been designed with the exact references of the properties to be mapped as column headings, the "Automatic Mapping" button at the top left of the page will allow the columns to be mapped automatically, without having to go over each column manually.

Expert advice: For automatic mapping to work best, each column in the preparatory file should be titled with the reference of the associated target property. For general information such as name, reference, icon... including unknown references, you can hover over the property name in the mapping list to get a tooltip that will tell you what to name the column.
In the example opposite, putting the cursor on "Type" displayed the "TYPE" tooltip confirming that the column in the preparatory Excel file should be called "TYPE" (here, it was not difficult to guess, it is more complicated with the colour of the icon for example...)



B. Special cases: multiple or historical blocks of properties

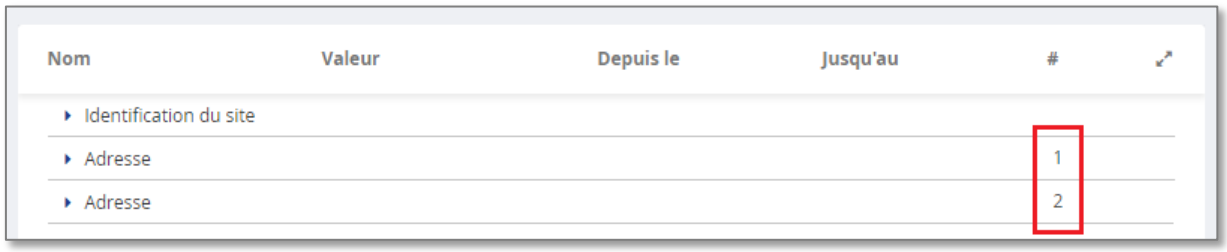
When importing a single property, the mapping is simple, you just have to check the desired property in the list. However, if you want to import a multiple or historicised block of properties, it gets a little more complicated.

a) Multiple blocks

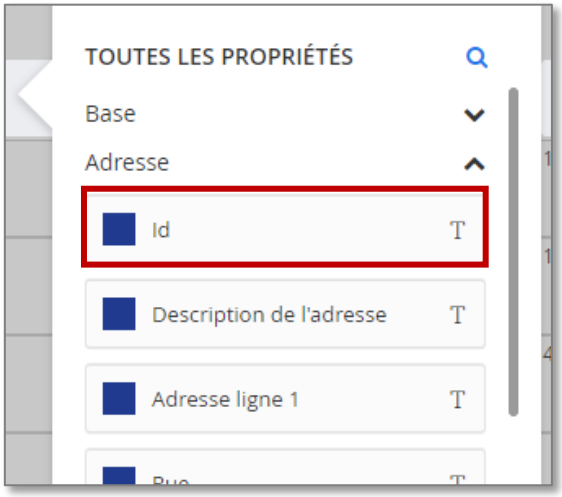
If we import properties into a **block** defined as **multiple**, a new instance of this property block will be created containing the new property values. Indeed, for EMM (JOOL), by default, these are new property values to be added to the list of values already present in the database. A new block is therefore created to receive them.

If the purpose of the import is to replace the values of an instance of an already existing block, EMM (JOOL) will have to be allowed to identify which of these instances should be modified. To do this, each instance of a multiple block has been given an ID so that it can be found among its peers.

To find the ID of a multiple block of properties, it is enough to recover it in the entity sheet, tab properties, the ID of each multiple block is on the right of its name in the list of properties.



For the import, we must therefore have an "ID" column in the preparatory file, which will contain the ID necessary to identify the instance of the multiple block to be imported, and which must be associated with the "ID" property of the block concerned.



A multiple block import can then be done in several ways:

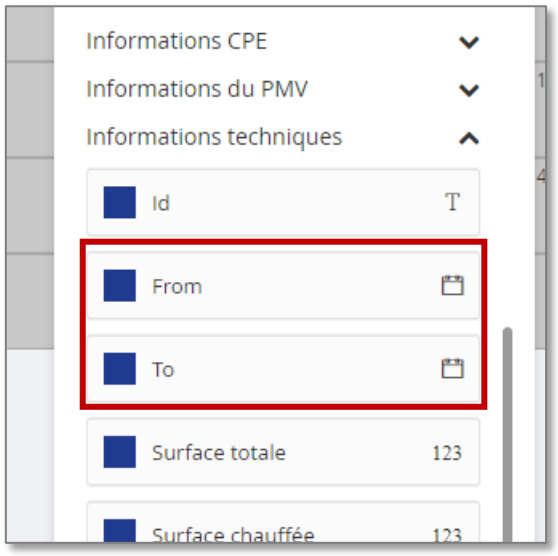
- **Without specifying an ID**: the imported block will be a new instance of the multiple block, with a default ID assigned by EMM (JOOL).
- **By specifying an ID that does not yet exist**: the imported block will be a new instance of this multiple block, but with the given ID as identifier.
- **By specifying an existing ID**: the imported block will be used to update the existing block with the specified ID.

Expert's note: If no existing instance of a newly imported multiple block has a numeric ID, and we import this new block without specifying an ID, then it will create this block as a new instance, and EMM (JOOL) will automatically assign it the ID = 1. If there are already one or more instances of this property block that have a numeric ID, EMM (JOOL) will find the largest existing numeric ID value and assign the next integer value to the new instance it creates.

b) Historical blocks

In the same way, it is also possible to import properties in **historical blocks** whose values can evolve over time. To do this, start and end dates can be added to these property blocks.
If no start date is given, the properties of the block are considered to be valid forever, and if no end date is given, they are considered to be valid forever.

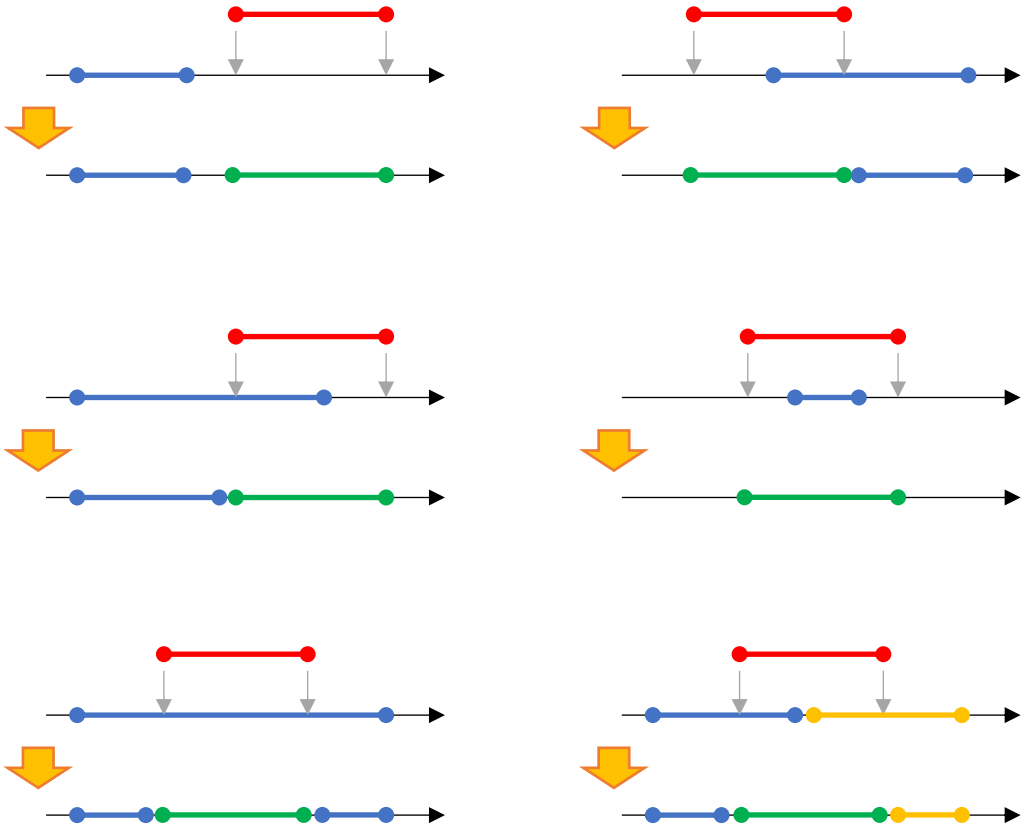
These historicised properties are therefore imported like normal properties, except that the preparatory file may contain a **From** column and a **To column** which delimit its validity in time.



When mapping columns for import, each block defined as historicised will contain two additional columns - **From** and **To** - which will allow the corresponding columns to be associated for import.

Caution: If an instance of a historicised property block is imported whose validity period conflicts with existing instances, EMM (JOOL) will overlap the existing blocks to bring in the new one. The new one then takes precedence over the pre-existing blocks. A pre-existing block that is completely overwritten by a new version will then be completely obliterated by the new one and disappear from the database.

Examples of imports of a historicised block in the presence of other pre-existing instances.



C. Types of imports

As mentioned above, each type of import has one or more mandatory properties without which the import cannot be done. These properties are most often necessary to identify the item to be imported, such as the reference of an entity or the ID of a multiple block.

Type of import	Mandatory properties	Reference for automatic mapping
Entities (actual or counted)	• Entity reference • Name of the entity • Type of entity	REFERENCE NAME TYPE
Source channels	• Source channel reference • Source channel type (Main, Default, Horo...) • Reference of the entity (counter) to which the channel should belong	REFERENCE TYPE ENTITYREFERENCE
Invoices	• Invoice reference • Type of invoice (Actual, Virtual, Proforma...) • From: start date of the invoice • To: date of end of billing • Reference of the invoiced entity	REFERENCE TYPE FROM TO ENTITYREFERENCE
Data	• Reference of the counter where to inject the data • Reference of the source channel where to inject the data • Date of data collection (end of measurement period) • Value of the data collected	ENTITY_REF SOURCE_REF DATE VALUE
Logical links between entities	• Entity from left to right (asymmetric links ! **1:N**) (side "1") • Right-hand entity to be linked (side "N") • Reference of the type of link to be used	LEFT_ENT_REF RIGHT_ENT_REF LINK_REF
User	• Login • Language code (e.g. FR) • Email • First name • Name	Login Language Mail First Name Last name
Events	• Entity reference • Event reference • Name of the event • Type of event • Start date of the event	ENTITY_REF EVENT_REF EVENT_NAME TYPE FROM
Contracts	• Contract reference • Name of the contract • Customer entity reference • Start date of the contract	CONTRACT_REF CONTRACT_NAME CONTRACT_CLIENT_ENTITY FROM

User rights	• Identifier • Function reference • Domain reference	USER_LOGIN ROLE_REFERENCE DOMAIN_REFERENCE

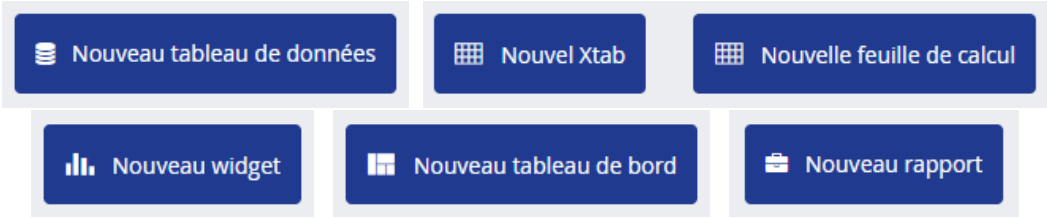
Note: **Parentage links** are special and are not treated as logical links between entities. The relationship will be imported via the properties when importing entities, the reference of the parent of an entity being considered as one of the properties of the daughter entity.

5. Tool editors: General

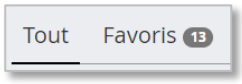
Whether opening the DataSet, spreadsheet, widget, dashboard or report editing page, the user is presented with a page listing all the objects in the chosen category, allowing them to choose an existing tool or request the creation of a new tool of that type.

At the top of the page, there is a search field that allows you to find an existing tool (dataset, spreadsheet, Xtab ... depending on the editor you are in) based on all or part of its name or reference.

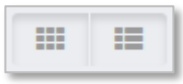
At the top right-hand corner, there are one or more buttons to create a new tool of the type concerned.



Finally, the page is divided into two tabs, allowing you to display all the existing tools, or to display only the favourite tools of the active user. The number next to the title of the "Favourites" tab indicates the number of tools in that tab.

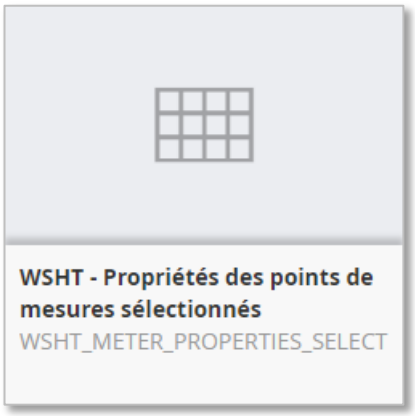


For DataSets and Spreadsheets, the actual list of tools can be displayed in two ways (choice via the following buttons in the top right-hand corner of the page)



A. Home - Tile view

The first option (and default view) is a tile view, with one tile per existing tool. Each tile looks like this:



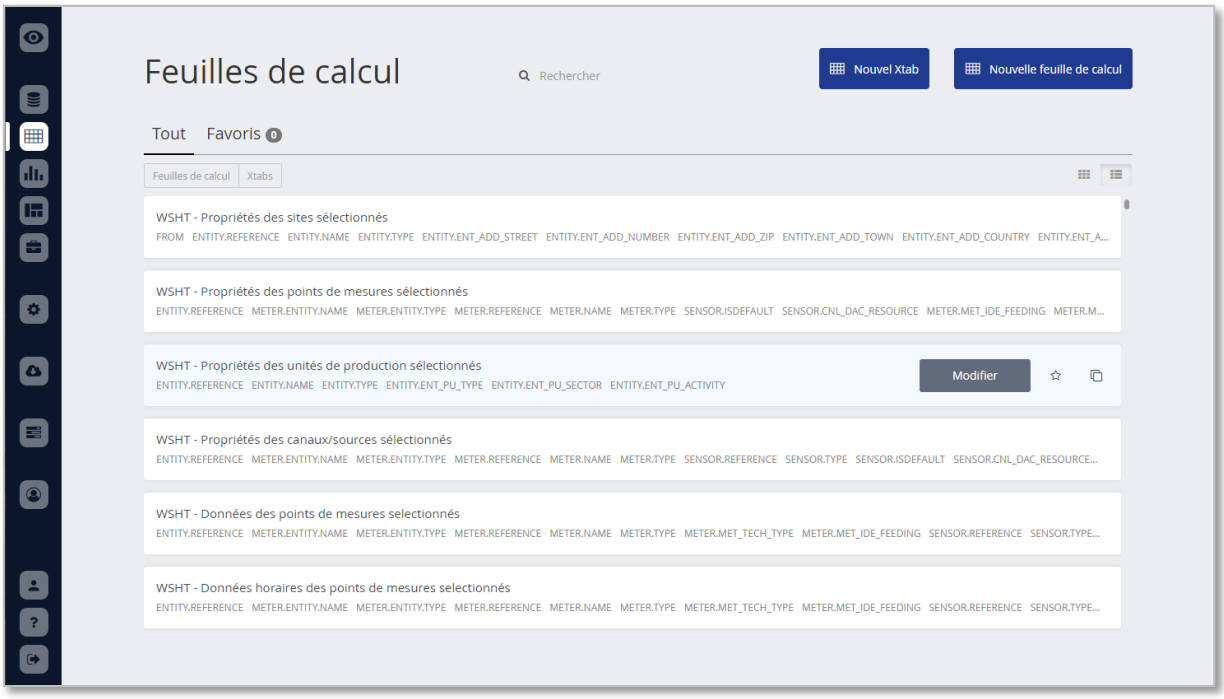
Normal tile

Mouse-over tile

The tile displays an icon, the name of the tool (in bold) and its reference (in grey)

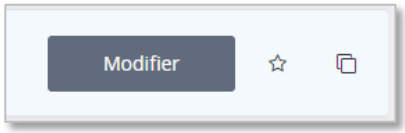
On mouse-over, a star appears (at the top of the tile) allowing the tool to be bookmarked, another icon allowing the tool to be cloned, an "Edit" button which opens the definition of this tool in the editor itself, and an explanatory text, which the creator of this sheet has written to describe the function of this tool.

B. Home - List view



The other view option will list the different tools in rows, displaying their names and the list of columns they contain.

When hovering over each line, the following buttons will appear, allowing you to edit the tool, bookmark it (star) or clone it (clone icon).

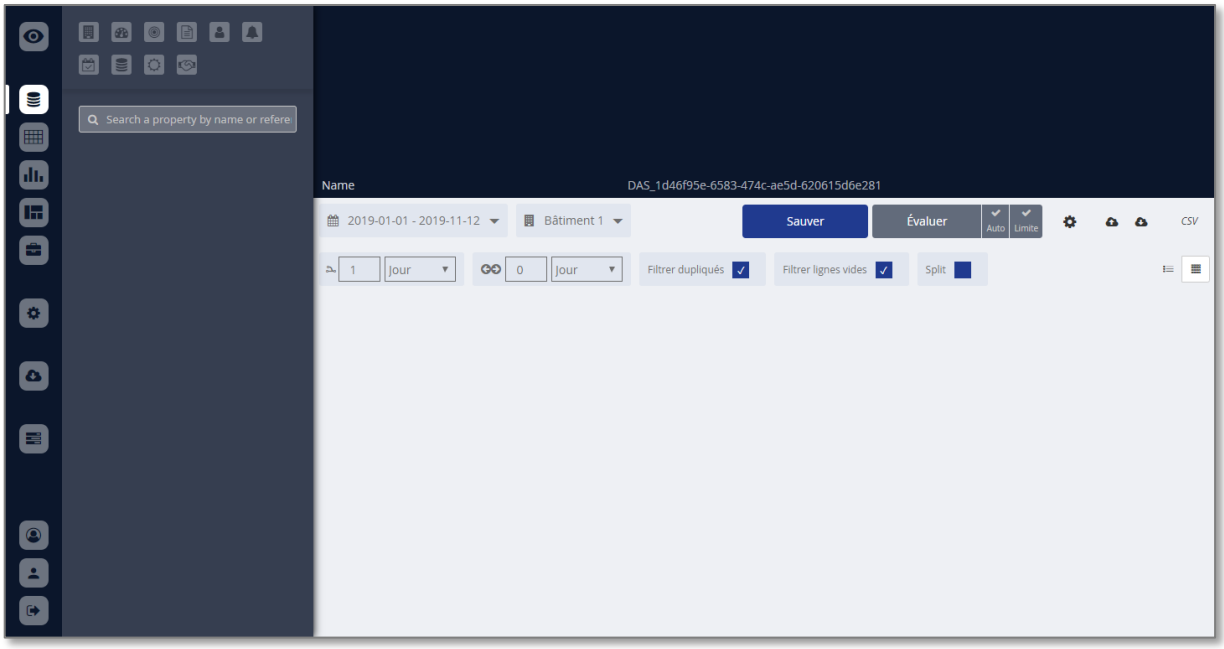


6. DataSet: extracting useful data

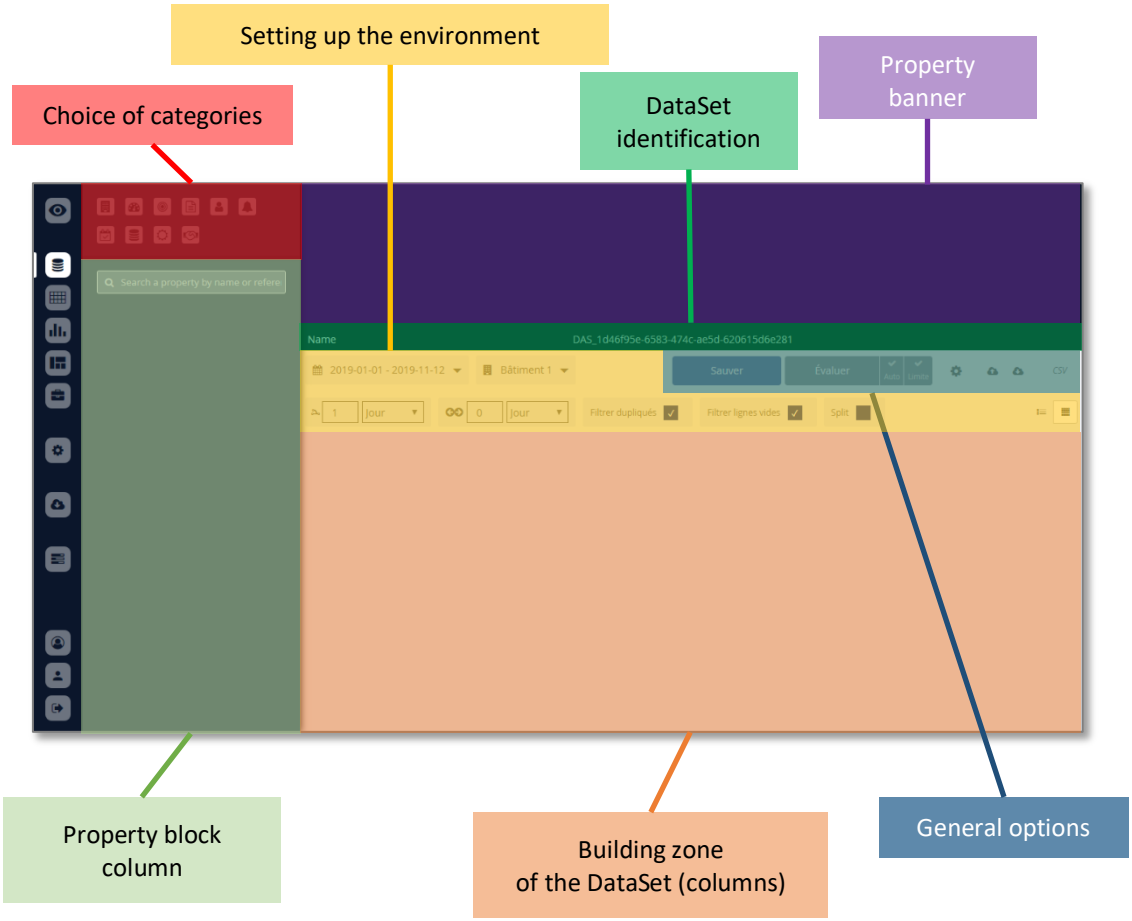
The first step in creating an analysis tool in EMM (JOOL) is the creation of a DataSet. This consists of an extraction of the database to retrieve all the information that will be useful for the following stages of the analysis.

This chapter covers the practicalities of creating a DataSet, but it can be supplemented by the expert's booklet "Creating DataSets", which goes into more detail on each of the rules for creating a clean and efficient DataSet.

The interface for creating DataSets looks like this:



This interface can be divided into several distinct areas:



A. Choice of object categories

The buttons in this area allow you to view all the existing property blocks in the database, grouped into object categories.

Each button will display the list of property blocks associated with the selected object type in the left-hand column of the interface.

Entities: the list of property blocks linked to the entities themselves. Reminder: an entity does not have a directly linked data profile.

Counters: the list of property blocks of the count points (counters) present in the DB.

Channels: the list of property blocks specific to the data channels inside the counters.

Invoices: the list of invoice property blocks.

Users: user property blocks.

Alarms: the property blocks associated with raised alarms.

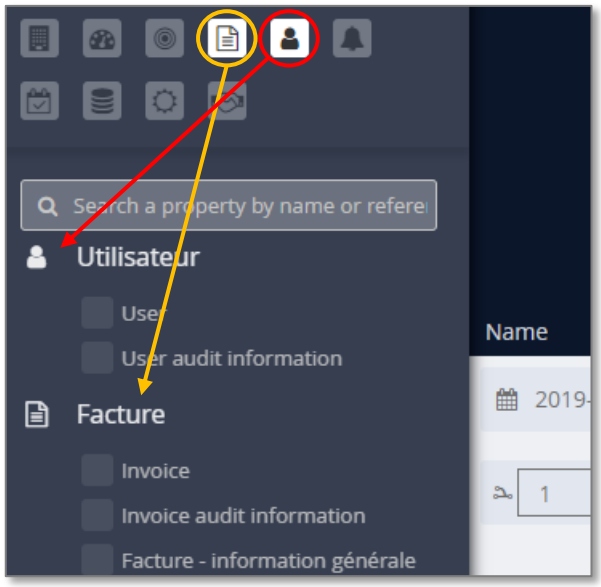
Events: the property blocks associated with events.

Data: the information blocks related to the data itself (data profiles).

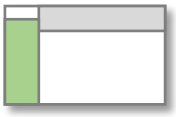
Meteo: the information available at the level of the existing weather entities.

Contracts: the properties related to objects of type contract in the DB.

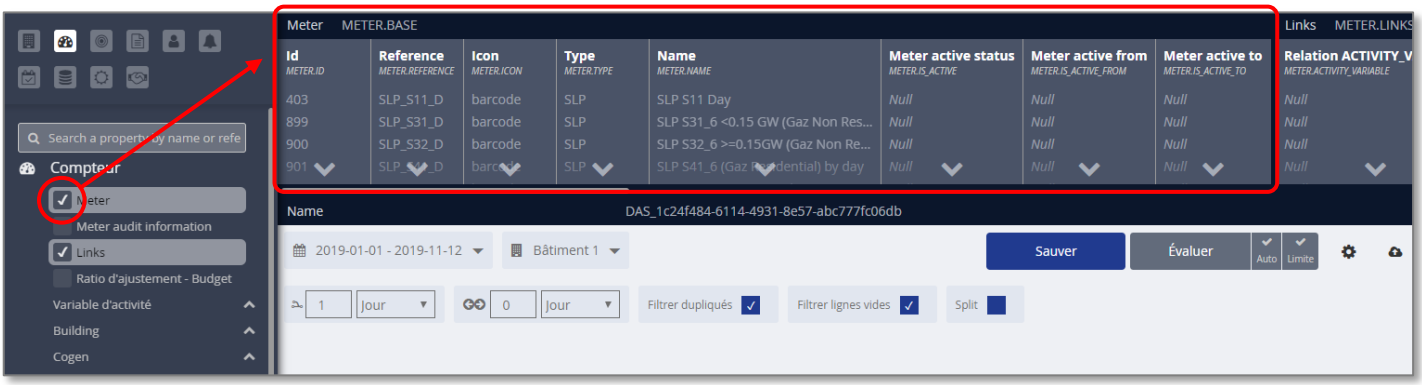
Selecting one or more of these categories will display all the property blocks associated with that category of objects in the left-hand column of the interface.



B. Choice of property blocks



Once the property blocks are displayed in the left-hand column, it is possible to tick one or more of these blocks, which will display the properties they contain in the top banner of the interface.



As mentioned above, property blocks are classified into groups according to the types of objects they concern. The same property block can therefore be found in several object types if these various types can use the block concerned.

Generic property blocks, or those not yet linked to any object type, will be listed outside the groups, at the top of the list.

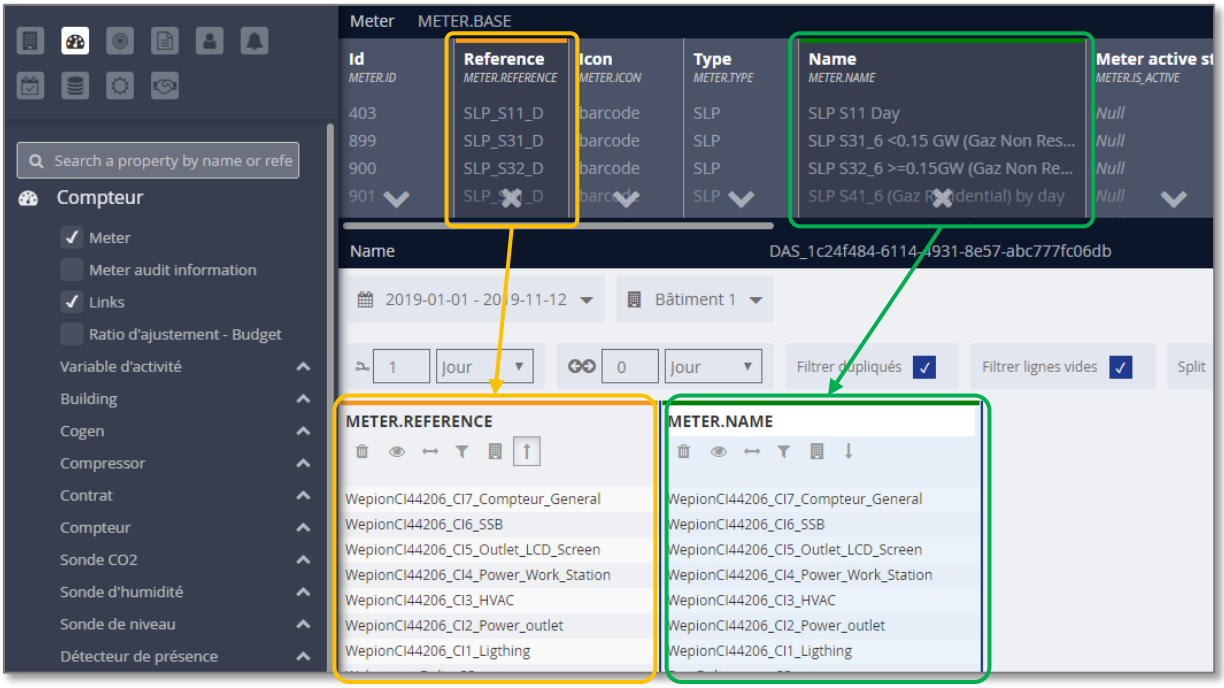
It is possible to display several categories in the block column, and by clicking on the name of one of these categories, it can be minimised if necessary for better readability.

C. Choice of properties

Once one or more property blocks have been selected in the left-hand column, the banner at the top of the interface displays the list of properties contained in the selected property blocks.

For each property, we see its name, its reference, and the first few values of that property, to get a quick idea of what to expect.

In this list of properties, we can then retrieve each property individually to make a column that we will add to the DataSet. Clicking on a property's box in the top banner will make that property appear in the body of the DataSet, in the main part of the interface.



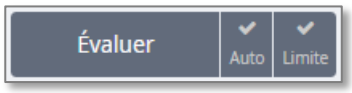
Properties added to the DataSet are given a colour to help find them quickly when editing DataSets. These colours have no structural impact, they are simply visual guides.

It is possible, if necessary, to add the same column several times in the DataSet. For example: we could have the same value profile aggregated in avg (average) in one column and in max (maximum) in the next column. The average of the data for the time slot and the maximum value reached in this time slot will therefore be recovered.

Once the columns have been selected, you can close the categories (left-hand column) to limit the number of available columns, visible in the top banner.

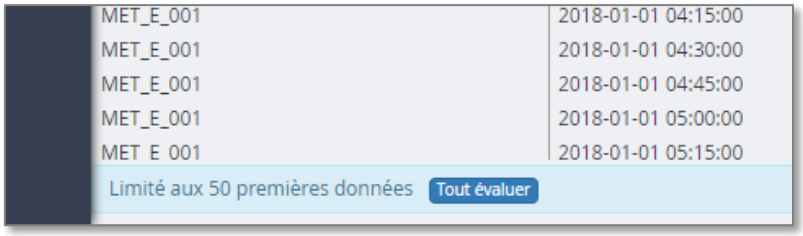
D. Column configuration

When the desired properties have been added as DataSet columns, they can then be evaluated to get an overview of the final result.

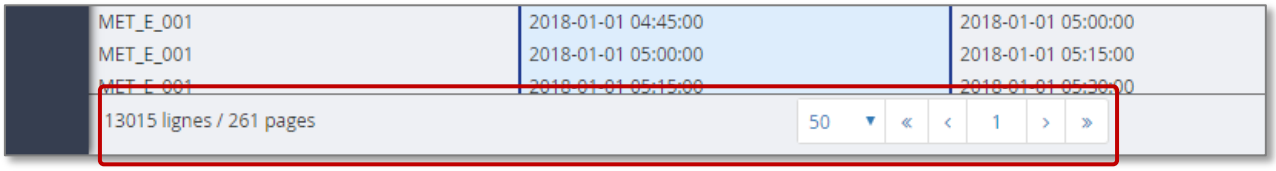


This is done via the "Evaluate" button in the general options area, at the top right of the columns. The small "Auto" box next to this button allows you to make the evaluations automatic each time the DataSet definition is modified. The small "Limit" box limits the calculation of the DataSet preview to the first 50 rows.

If a run has been performed on the first 50 values, it is then possible to click on the "Evaluate All" button at the bottom of the DataSet to force a calculation on all rows. The other possibility is to uncheck the small "Limit" box and re-evaluate the DataSet.



If too many lines are to be displayed, the DataSet will be displayed in several pages.



At the top of each column of the DataSet, you can find several things:

- An editable field that contains the column reference, automatically pre-populated according to the origin of the column.

- A series of icons allowing you to configure (sort, filter, etc.) the column.

Delete the column

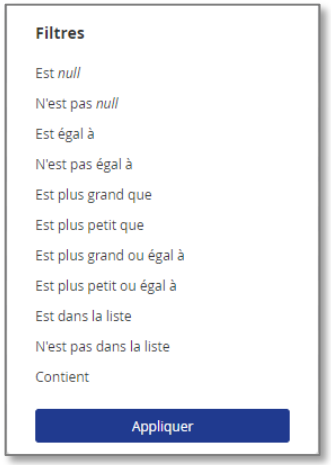
Display **provenance**: This button

- Reopens the category from which the column came if it had been closed (left column)
- Scrolls in the left-hand column to make the block containing the property associated with that column visible on the screen (if required).
- Checks the block concerned, which makes its properties reappear in the banner if they had been removed from it

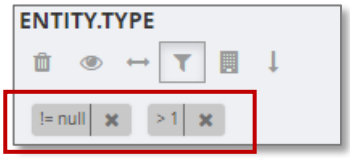
- Scroll sideways in the top banner so that the property in question is visible on the screen

This button is used as a **handle** to move the column within the DataSet. This is how you can rearrange the columns in a DataSet for proper formatting.

Opens a pop-up window to apply a **filter** to the column. This allows you to filter the rows of the DataSet based on the values contained in each of its columns if required.
The available filters are shown in the example on the right.
Filters requesting a value (e.g. "is equal to") will offer a field when activated to encode the required value.
It is also possible to activate several filters simultaneously if required.



Once a filter is activated, a small button will appear under the icons summarising the filter and allowing it to be removed.

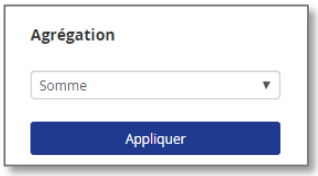


Sorts on the column. The sorting is chronological on a date column, ascending for numeric, and alphabetical for text.
A second click on this button will reverse the sorting, and a third click will remove the sorting.

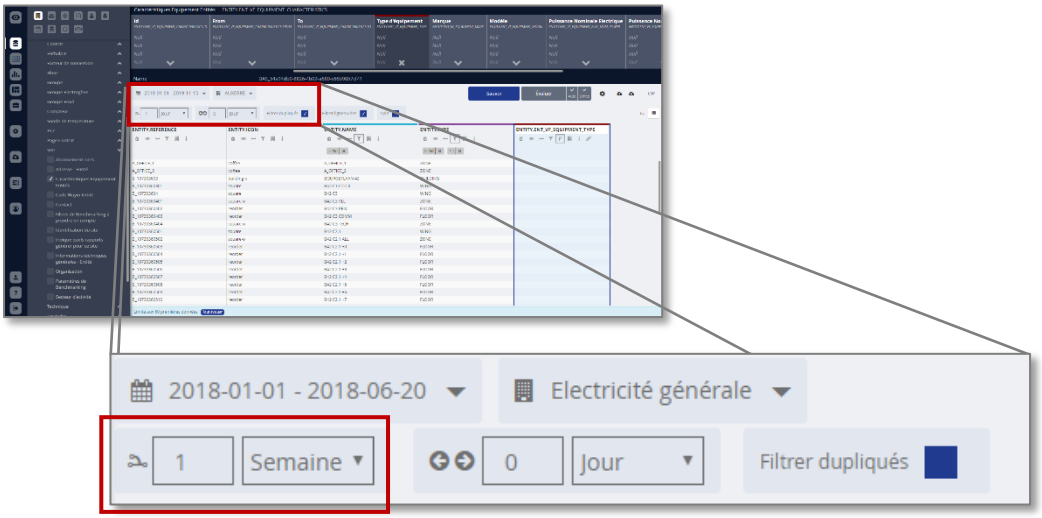
Only present on columns containing entities (references, names...), this button allows you to automatically filter the DataSet to limit it to the **active selection**.

Only present on columns containing dates, this button automatically filters the DataSet to the **active context**.
This button is a shortcut to add this (very common) filter without going through the filter pop-up.

Only present on numeric data columns, this button signals a possible **aggregation** on the values of the column. This button opens a pop-up window allowing you to choose the aggregation method to be applied from a drop-down menu.



Caution: If a numeric column is aggregated, all numeric columns must be aggregated as well. All numeric columns may have their own aggregation method, but they will all share the same time step, defined in the fields at the head of the table, just below the time context selector.



As mentioned above, it is possible to import several columns from the same property and aggregate them differently, to obtain for example a column with the average of the data per time slice, and another with its maximum on each time slice.

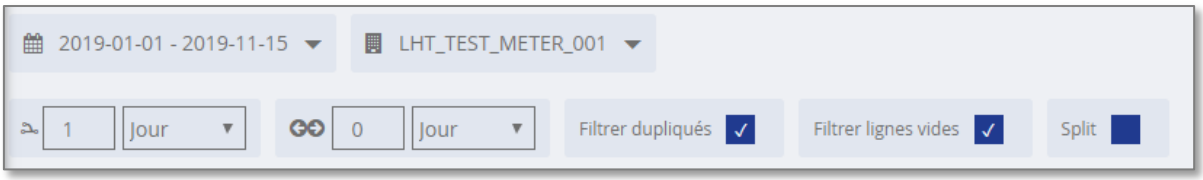
Present only on the property columns, this "**Xproperty**" button allows you to search for the property value not only on the item concerned, but also on the closest entities/counters.

If, for example, we have a list of meters and we want to retrieve the surface area they are used to heat, we have a problem because this surface area property is not stored directly on the meter but on the site associated with it, and this via a path that can sometimes be complicated.

Retrieving the property from the DataSet, as well as using the JQL word ".properties" will not return anything. On the other hand, the JQL word ".xproperties" will search for the requested value on nearby entities. This is also what the DataSet does if this box has been checked.

E. Formatting options

Just above the DataSet columns is an area containing fields for setting several options:



- A **temporal context**: Definable via a pop-up temporal selector, this context will not be saved with the DataSet, it is only used to set the situation for the visualisation of the DataSet results when it is created.

- An **entity selection**: Definable via an entity selection pop-up similar to the one used in viewer mode, this selection is used to evaluate the DataSet in the creation interface, it will not be saved with the DataSet either.

Below these selectors is an area for defining an aggregation time step, which will be common to all aggregated columns in the DataSet.

These fields will be used when one or more data profiles are retrieved from the DataSet columns. The aggregation fields will be associated with the aggregation method choices entered in each profile column to form consistent aggregations.

In the choices available for this time step, one can also find an "Automatic" option which will define the aggregation time step dynamically according to the total duration of the observation period.

- If the duration of the time selection exceeds 500 days: 1 year
- If the duration of the time selection exceeds 300 : 1 month
- If the duration of the time selection exceeds 10 days: 1 day
- If the duration of the time selection exceeds 1 day: 1 hour
- Otherwise: No aggregation

It is this type of aggregation that will make it possible to use graphs in "linerange" mode to display an average trend and a variation zone (see chapter on graphical widgets)

Two other fields allow you to define a time shift of profiles. This is where you can, for example, retrieve last week's data and join it to another DataSet retrieving this week's data, and then join the two based on the date.

Next, a checkbox allows you to filter out duplicate data in the DataSet. This filter will remove rows that are completely duplicated, thus having the same data for each column in the DataSet.

Another checkbox allows you to remove completely empty rows.

Finally, the "Split" box can only be used in the case of DataSets containing invoices, and it will apply a split to the dataSet, cutting them off at the beginning of each time step defined in the fields used for aggregation.

F. General options

In this area, the "Save" button allows you to save the definition of the DataSet, and the "Evaluate" button allows you to force its evaluation, to check that it returns what it is supposed to. The small "Auto" box next to the "Evaluate" button indicates whether you want the evaluation to be done automatically without having to ask each time.

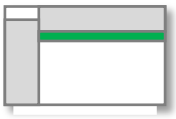
The 'Delete' button is only displayed if you are working in a DataSet that has already been saved, and it allows you to delete it if necessary.

The "Edit Properties" button, in the form of a gear, opens a pop-up window allowing you to modify the reference, name and description of the DataSet.

As a reminder, the **reference** of an object in EMM (JOOL) is the unique identifier, used by the syntax to access it. It must absolutely be unique. The **name** of an object is a more humanly understandable identifier, which may not be unique, and which will be used by the user to identify the object in question. When available, the **description** is a small free text allowing to add some useful information associated with the object.

Finally, the cloud symbols allow you to save the DataSet definition to a file for later use (whether as a back-up, or for copying to another DB for example) or to load a previous backup to your hard drive.

G. DataSet identification



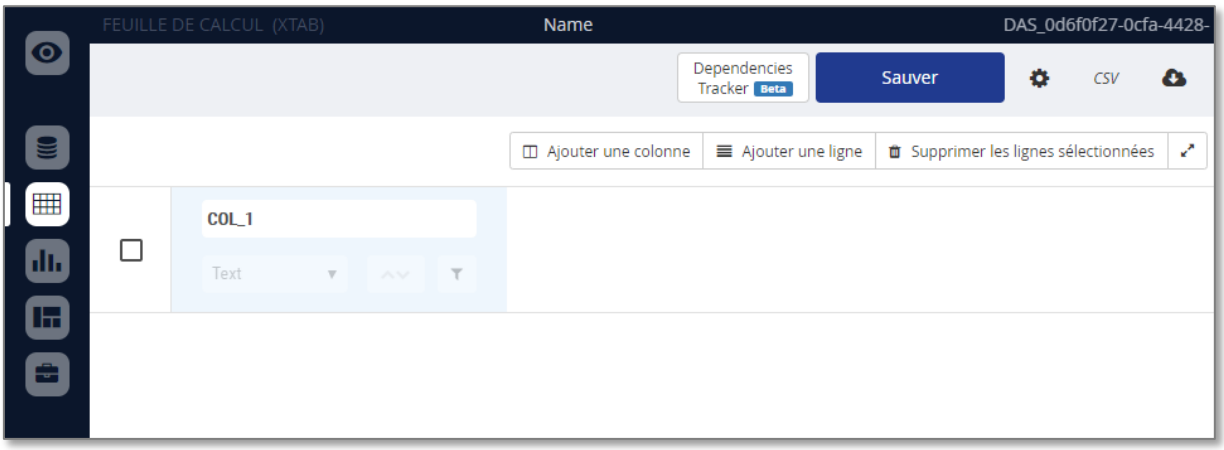
Just below the properties banner, a thin area contains the name and reference of the DataSet. The name is an editable field directly in the interface page, but the reference can only be changed via the 'Edit Properties' button in the previous section.

7. WorkSheet: Preparatory calculations

A. Static sheet (Xtab)

A static sheet (or Xtab) is a table of fixed, uncalculated data that can be used in other analyses. This kind of static sheet can contain price data for example, or values of fixed coefficients, used by many sites...

The interface for creating Xtabs has the following structure:



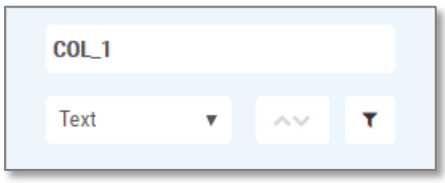
The thin dark banner at the top of the page contains the (directly editable) name and reference of the open Xtab.

At the top right is a button to save the XTab as is, a small gear wheel to edit its identifiers (name, reference and description), a button to export the XTab as a CSV file, and a final button to import data massively into the XTab.

a) Manual construction of an Xtab

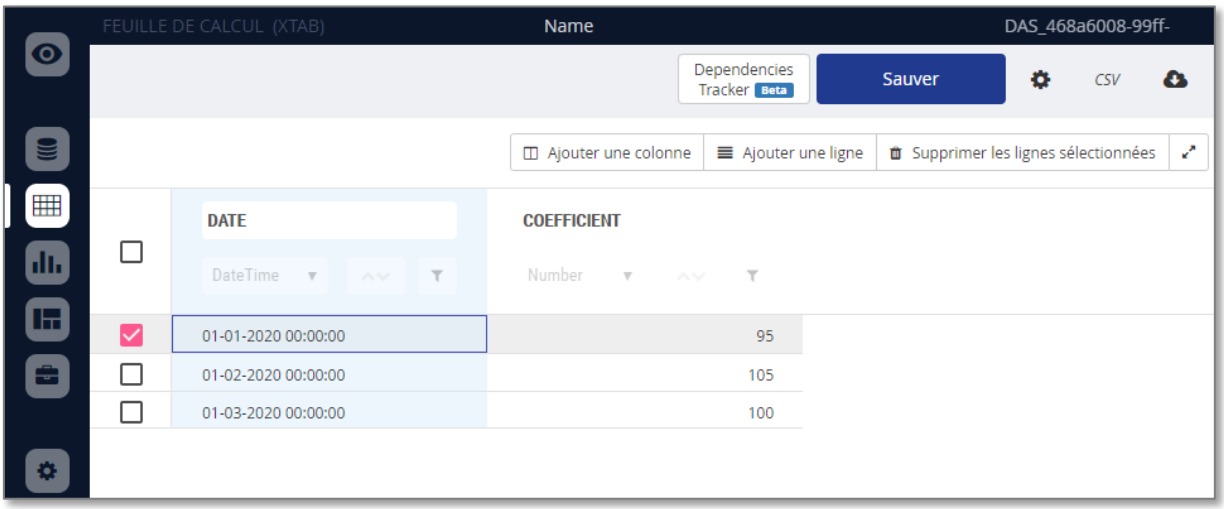
Just below these main options at the top right of the page are buttons for creating rows or columns, and for deleting selected rows.

We generally start by creating the columns, thus defining the structure of the Xtab. For each column, we can then define its name (which will be used to access its contents) but also the type of object that the column will contain (Text, Numeric, Date, Boolean...).



It should also be noted that each column will have a button to sort on the values in the column, and a funnel button, which brings together a range of filter, sort and column formatting options.

When the columns are ready, you can then start adding rows and filling in the Xtab values. This is done by adding rows, then entering each box and typing in the desired value. In the case of dates, a calendar will open to choose the desired date, and for other types of value, simply type it in.

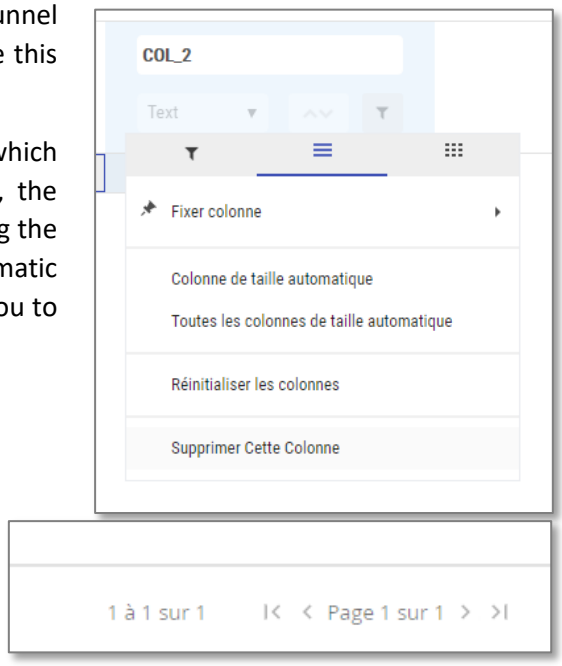


Once the Xtab is filled with its values, it can be saved and used in EMM (JOOL) syntax.

Note that a column of checkboxes is located on the left-hand side of the Xtab. It is therefore possible to check one or more rows to select them and then delete them en masse with the "Delete selected rows" button.

If you want to delete a column, you can go to the funnel button of the column and choose the option "Delete this column".

This button is divided into three sub-tabs, the first of which allows you to apply filters to the values displayed, the second of which contains the display options for fixing the column to one side of the table, or giving it an automatic width, as well as deleting it, and the last tab allows you to hide the display of certain columns if required.



Finally, at the bottom of the interface window, there are, as always, a few buttons to manage a multi-page display of the Xtab.

b) Mass import of content into an XTab

In addition to manual data entry, it is also possible to import data into XTab from a preparation file on a massive scale.

The import procedure is similar to that used for bulk imports, with a large area where you can copy and paste the data table (or drag and drop a CSV file, or search for a file on your hard drive)

Once the data is in the import area, you go to the next step where you can specify the column separators, the decimal separators, the date formats... of the import file, then finally execute the import.

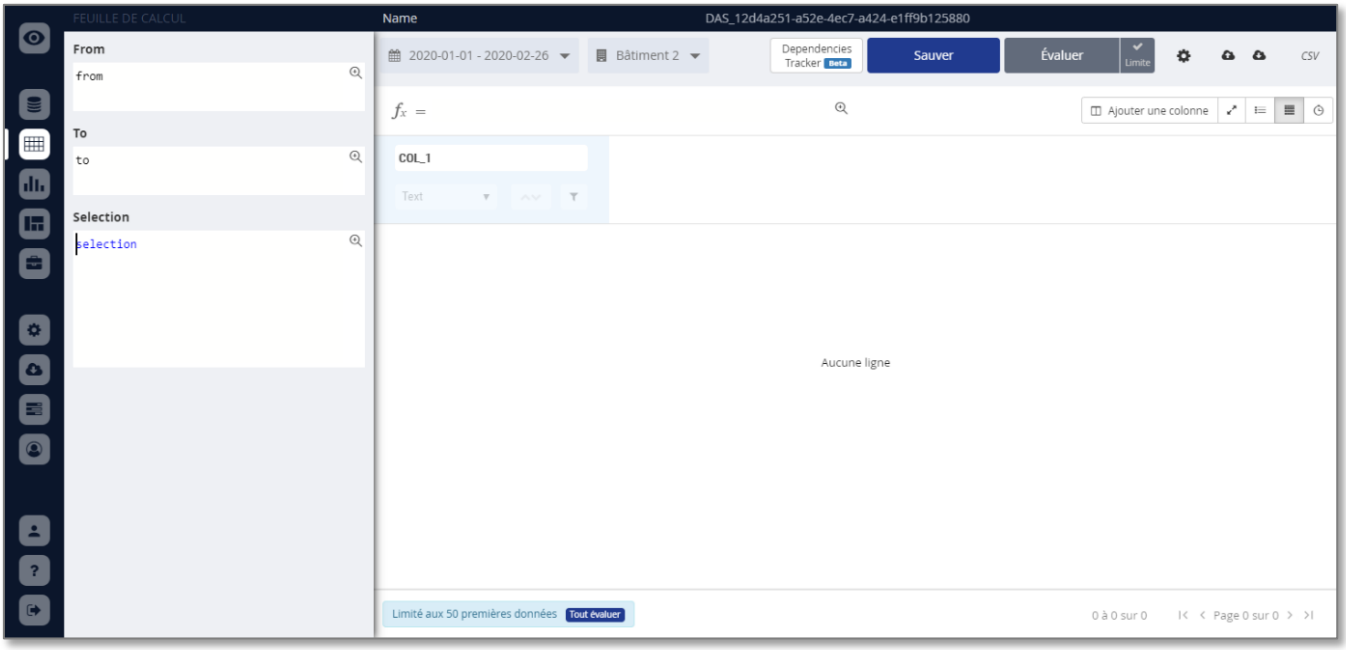
Note: Here, there is no need to map the source file columns to the property references. The import retrieves the entire import file, including the column structure, their references, and the values of each cell. EMM (JOOL) will attempt to identify the types of each column based on their contents, but it is important to check this before saving the XTab.

B. Dynamic spreadsheet

Dynamic worksheets are built on the basis of data from DataSets, other worksheets (dynamic or XTabs) or directly from the database information.

They perform calculations on this data to build up useful and prepared information for the widgets' final use.

The worksheet creation interface has the following structure:



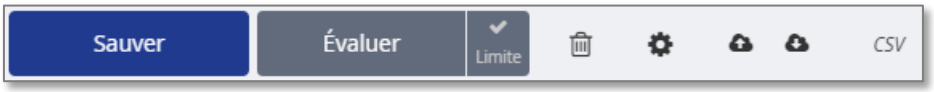
As with XTabs, there is the thin banner at the top of the page containing the (editable) name and reference of the worksheet.

A magnifying glass icon can be found throughout the fields in this editor. This opens a pop-up window containing a text editor that makes it easier to write formulas in EMM (JOOL) language. This text editor gives us more space than the limited size fields on the page.

This section of the user manual covers the practicalities of creating worksheets, but it can be supplemented by the expert's booklet "Creating Worksheets", which goes into more detail on each of the useful rules for creating a clean and efficient worksheet.

a) Evaluate / Save / Modify properties / Delete

At the top right are a few buttons to perform general actions:



Sauver	Saves the worksheet definition in its current state

	Starts a recalculation of the sheet. The "Limite" checkbox forces a limitation of the calculation to the first 50 lines. Unchecking this box will allow a complete calculation.
	This button is only displayed if you are working in a worksheet that has already been saved, and allows you to delete it.
	Opens a pop-up window allowing you to modify the worksheet identifiers (Reference, Name, Description)
	Save the worksheet definition in its current state for later use (back-up, transfer to another DB...)
	Loads a previously saved worksheet definition. This definition will overwrite the current configuration of the open worksheet.
	Allows you to export the displayed result of the worksheet in CSV format. Be careful, this button does not restart a calculation, it only exports the results already displayed. This means that if a modification has been made in the definition but the calculation has not been restarted, the old results will be exported. The same applies to the 50 row limit. If a test has been done on 50 rows, an export will just return these 50 rows. Do not forget to run a complete calculation before exporting, otherwise you will only get partial results

b) Time and entity selectors

At the top of the window, next to the general buttons, there are selectors (entity selector and time selector) for choosing the time context and the selection of entities to be analysed. These selectors are similar to those that can be found in the DataSet editor or even in the dashboards in visualisation mode.

This information (context and active selection) will not be saved with the worksheet definition, it is only used to visualise results when creating the worksheet. It is during the execution of a worksheet that it should be provided with the actual context and selection to work with.

c) Temporal context

At the top of the left-hand section of this interface are the "FROM" and "TO" fields. These fields should contain formulas (in EMM (JOOL) syntax) that provide the actual time context for the spreadsheet, the time period over which the calculations should be performed.

- This temporal context can be fixed with "hard" dates (e.g. the case of consumption calculation for a fixed reference year).

- It can also be based on the **now** keyword which represents the immediate date/time at the time of execution.

- It can also simply reflect the active context (chosen by the user in the temporal selectors) by using the keywords **from** and **to**, defining here respectively the beginning and the end of this active context.

- Finally, it can take these dates and/or keywords and modify them via EMM (JOOL) syntax to obtain a specific working time context (e.g. **now.synchro(1; "day")** will return the start of the day of the date of the spreadsheet execution).

Examples (EMM (JOOL) syntax):

FROM: from

TO: to

→ The time context will be precisely that selected in the time selector.

FROM: now.synchro(1; "month").add(-1; "month")

TO: now.synchro(1; "month")

→ The time context will be last month. We start from the current date, synchronise to the beginning of the month, and remove one month, and as the end date, we take the current date, and synchronise to the beginning of the month.

Once these fields are filled in, the **from** and **to** keywords are redefined and will now reflect the results of these "FROM" and "TO" fields for the rest of the spreadsheet (and no longer the from/to of the active context).

d) Selection

Under the "FROM" and "TO" fields, you will find the "SELECTION" box which will allow you to define (in EMM (JOOL) syntax) the list of objects to be calculated.

- This box can contain a formula based on a list of hard entities, such as (@REF_1 ;@REF_2), in the case of fixed scope analyses.

- It can also be based on the **selection** keyword, which refers to the active selection, the one made by the user in the entity selectors.

- It can also start from the keyword **all**, which represents all entities, or from a #TYPE, which represents the list of entities of type "TYPE", whatever the type chosen.

- It can also contain a formula that calls on one or more existing DataSets or worksheets, in order to retrieve their results and use them in its calculations.

- Finally, it can contain any list of values or formula producing a list of values, such as a sequence of dates generated with the "range" function.

Whatever its form, the "SELECTION" box will return a collection of objects (entities, counters, properties, data, numerical values, table rows, etc.) which will constitute the scope of work on which the analysis will focus.

The result table in the spreadsheet (main area, right) will contain one row for each object in the selection, and for each of these rows, the keyword "item" (in EMM (JOOL) syntax) will designate the object in question, attached to the row.

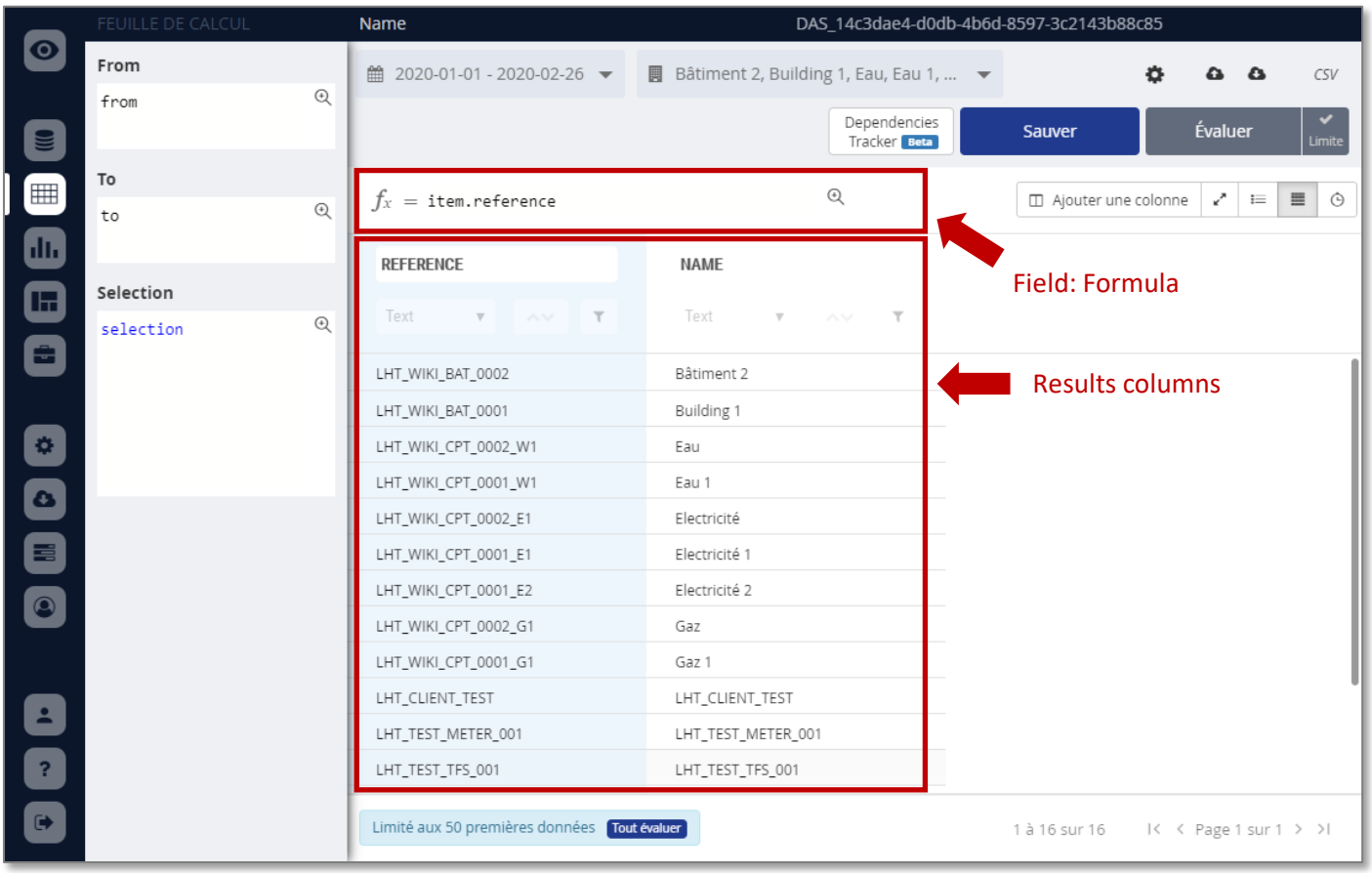
e) Creation of columns

Firstly, the 'Add Column' button at the top right of the main editor area allows you to create the columns that will make up the worksheet.

Then, for the configuration of the columns, the editor proposes two visualization modes to create the columns. You can switch from one mode to the other using the buttons at the top right of the window.

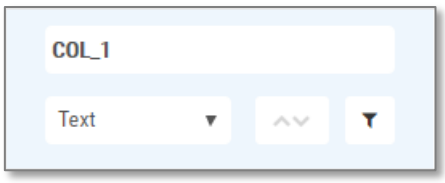
Table mode

The "Table" mode is the one that will be selected by default when opening a worksheet. Structurally, it looks like this:

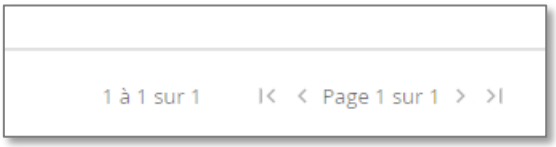


The field at the top of the page displays the formula for the currently selected column. You can select a column by clicking on it, and it will then take on a slightly bluish tint. As elsewhere, the magnifying glass is used to open an editor and have more space to write your formula.

The header of each column contains several fields allowing you to define its name, the type of data it should contain (text, numeric, date...) and to sort or filter the rows according to the values in the column.



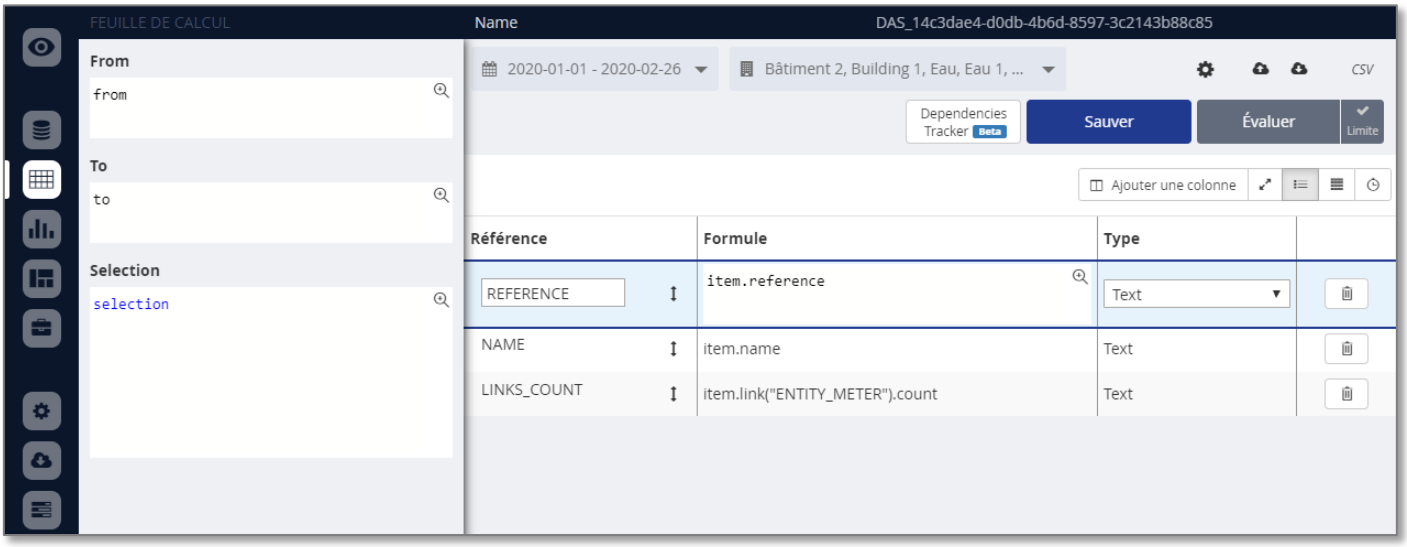
At the bottom of the page, a "Rate All" button removes the constraint of the first 50 lines (the same effect as removing the tick in the "Rate" button), and at the bottom right, there is a way to navigate through the results if they require more than one page to be displayed.



The advantage of Table mode is that it allows you to easily view the results of the worksheet construction, even in the middle of its elaboration. The disadvantage is that you can only view one formula at a time, switching from one column to another.

List" mode

By switching the display to "List" mode, the structure is reorganised and the formulas of all the columns are listed one below the other. Each row in this mode corresponds to a column in the final worksheet, and we can configure them all at once before launching an execution (which will put us back into "Table" mode)



A definition row will have a field displaying the column reference, a small button with double arrows, which allows you to enter and move it to change the order of the columns (this is the only way to move columns in a worksheet), then a field displaying the column formula, followed by a drop-down menu allowing you to choose the type of values the column will contain and finally a button in the form of a dustbin to delete the row.

The advantage of the "List" mode is that you see all the formulas at once, and it is easier to change them en masse when creating a worksheet. The disadvantage is that you don't see the results of the calculation, which is why an evaluation switches you back to "Table" mode

Contents of the formulas

In the formula fields of each column, EMM (JOOL) syntax can be used to retrieve the desired information and display it in the worksheet columns. For more details on the language used in EMM (JOOL), see the Syntax manual, which covers in detail all the features available in EMM (JOOL).

Statistical evaluation tool

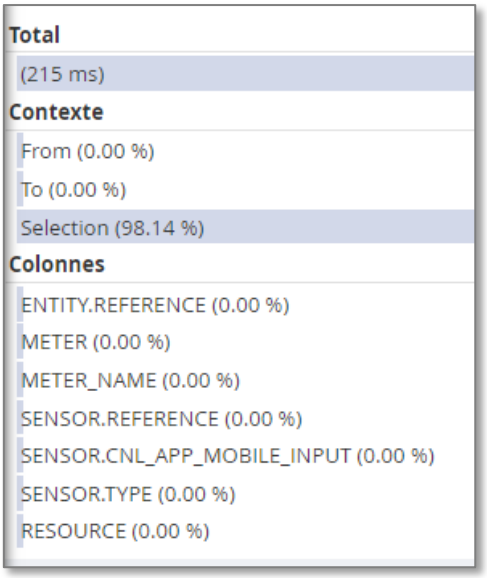
The last button in the spreadsheet editor, at the top right of the main area, representing a stopwatch, takes you to a page showing the calculation times of the different components of the spreadsheet.

On this page, we find the total calculation time of the worksheet (in milliseconds), then the evaluation times of the context components (From, To, Selection ... which, being formulas, can take time to calculate), and finally a list of the columns of the worksheet, with each time the percentage of total calculation time that the column cost to execute.

This tool is particularly useful when designing a spreadsheet as it allows you to quickly see which part is the most time-consuming, and it is possible to test different ways of writing formulas, objectively measuring their performance during an execution.

Note that this page will only have information when at least one run of the spreadsheet has been performed. The times shown are for the last run performed, and these times may obviously vary slightly from one run to another.

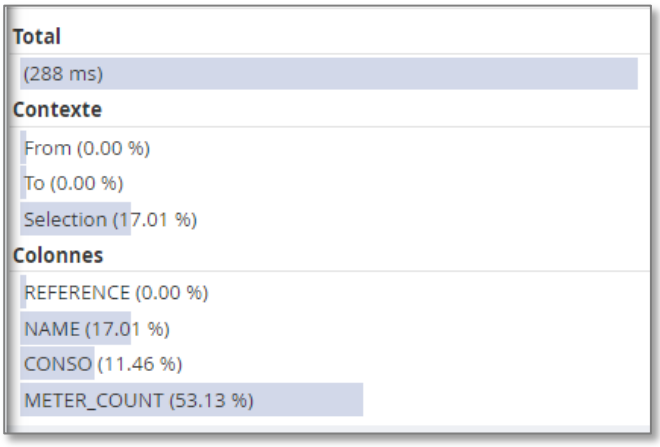
Example:



This sheet took 215ms to run, and almost all of the calculation time was caused by the selection.

As this spreadsheet was created to retrieve a DataSet and copy all the columns, these results are quite consistent.

Example:



In this example, the selection calculation takes 17% of the total calculation time, the REFERENCE column is almost instantaneous, but the NAME column alone takes 17% of the time. It may therefore be relevant to check whether the formula for this column is optimal, or even whether this column is really necessary.

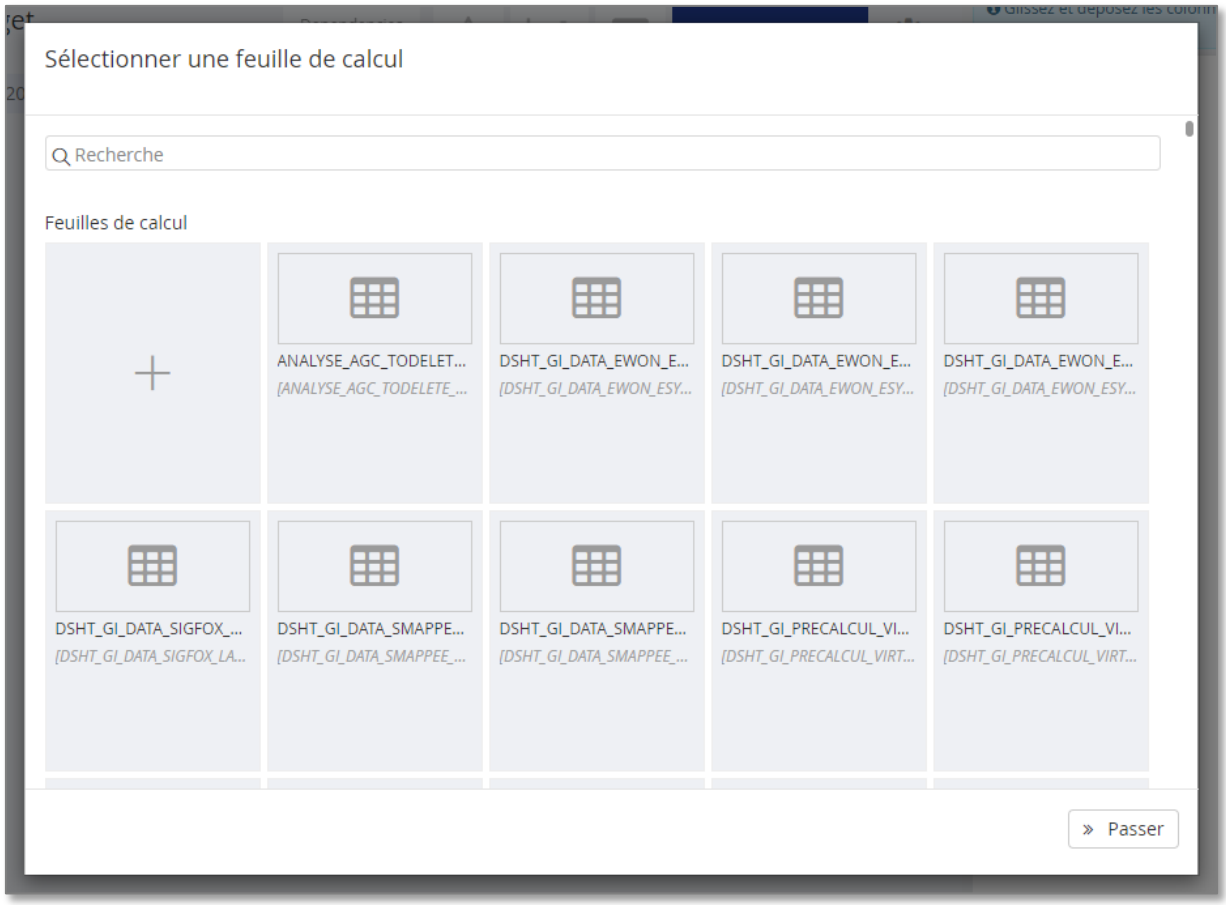
The same applies to the "METER_COUNT" column which takes up more than half of the total calculation time of the sheet.

8. Widgets : Display of results

A widget is a tool for formatting and displaying data, based on a pre-existing worksheet.
A widget cannot generally contain any formulas, it simply displays the data from a DataSet in a chosen format.

When you open the Widget Editor, you can either open a pre-existing widget or create a new one ("New widget" button at the top right of the Widget Editor home window).

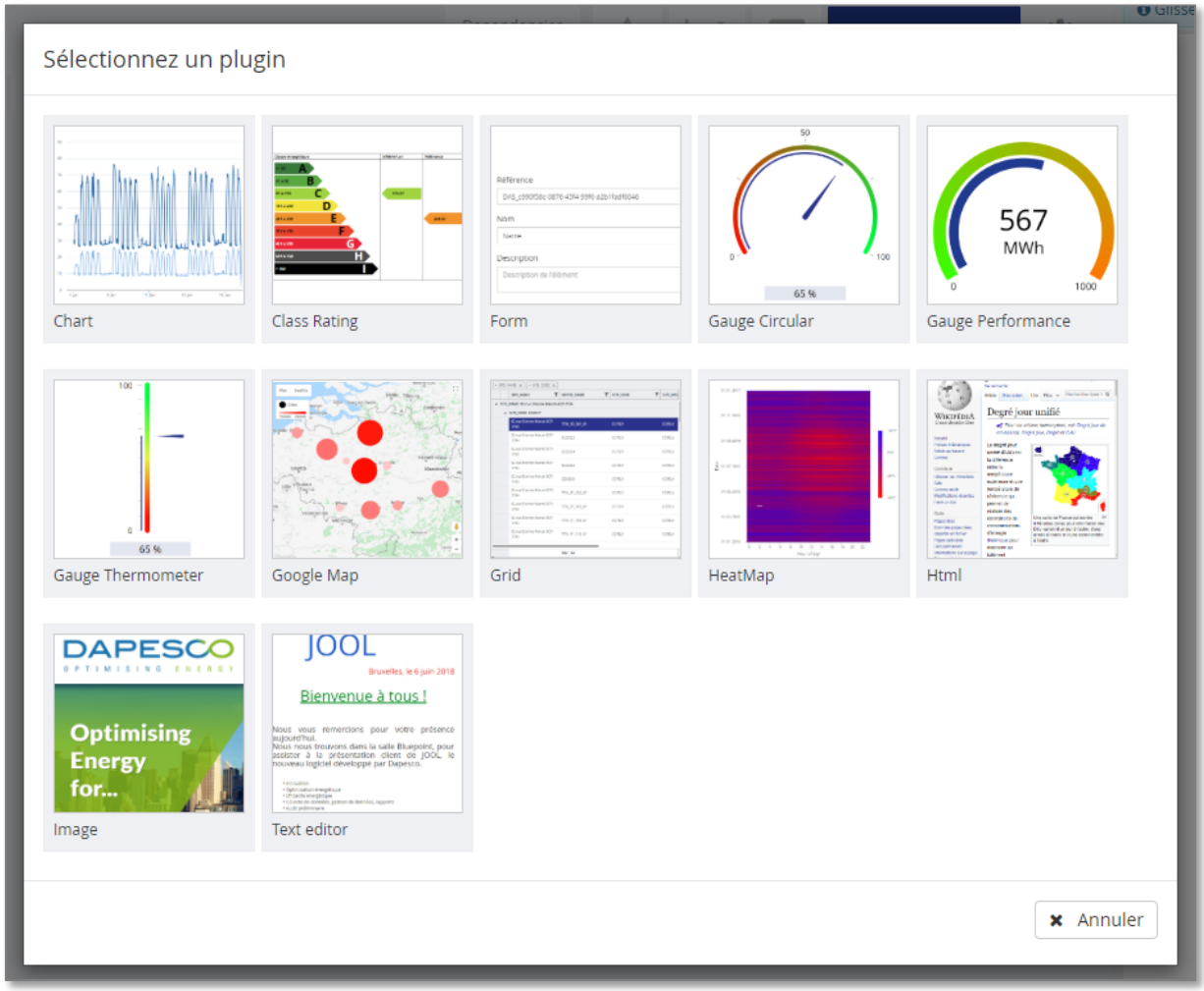
In this case, EMM (JOOL) opens the list of existing spreadsheets, so that one can choose which **spreadsheet** the widget will be based on.



A search field allows you to filter the list, and each tile contains an icon, the name, reference and description of an existing worksheet. Hovering the mouse over the text of one of these three pieces of information will bring up a tooltip with the same text, which can be very useful if the text is truncated due to lack of space.

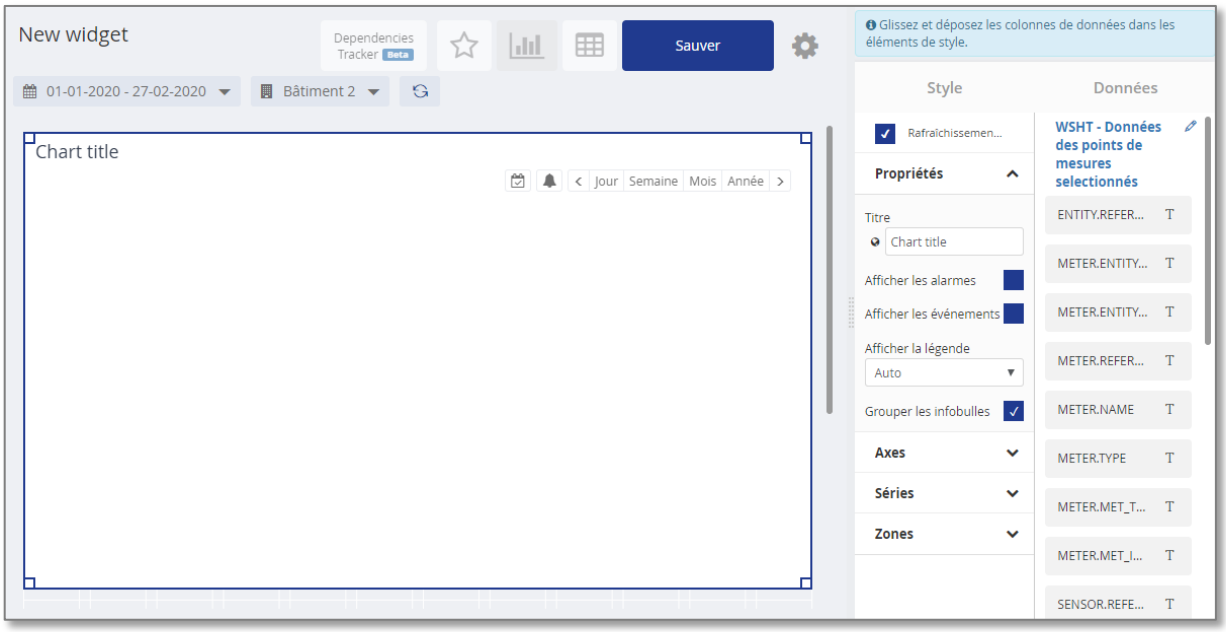
If you want to build a widget that is not based on a worksheet, you can choose the 'Skip' button at the bottom right of the page to skip the choice screen and move on to building the widget without having to choose a source worksheet.

Once the worksheet to be used has been selected, EMM (JOOL) will ask for the **type of Widget to be** created.



Widgets can be of various forms: chart, label, form, various gauges, maps, data tables, matrix graphs, HTML reports...

Once the source worksheet and the type of widget have been chosen, we enter the widget **creation interface** itself. These will all have a similar structure, regardless of the type of widget chosen (here the chart interface as an example):



At the top left is a field (editable) displaying the name of the Widget.

Just below this are the time and entity selectors, which allow the result of the widget's configuration to be viewed during its creation, based on a realistic context and selection. As usual, these fields will not be saved with the definition of the widget, and only serve to allow the visualisation of results to facilitate creation.

To the right of the selectors are a few buttons:

Recalculates the result of the widget after a configuration change

Add/remove widget from the favourites list

Change the type of widget being created (back to the widget type selection screen).

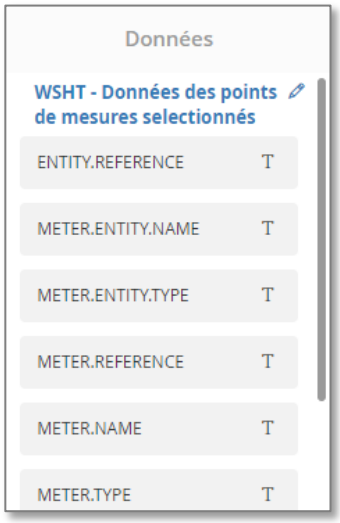
Change the source worksheet for this widget (opens the worksheet selection screen).

Save the active widget, with its current characteristics (name, reference, description, definition...)

Open a window displaying the widget identification parameters (Reference, Name, Description)

(Optional) Only appears if you are working on an existing chart, this button allows you to delete the widget.

At the far right of the interface window is the "Data" column, which lists all the columns in the associated worksheet.



At the top of the column, we have the name of this worksheet (If needed, we can click on this name to open the worksheet for editing, a "back to widget" button will then be available in the worksheet editing interface to go back to the widget definition).

Below, there is a grey block for each column of the spreadsheet, displaying the column reference and an icon indicating the type of data contained in the column (Text, Date, Numeric...)

Each of these grey blocks can be used as a source for a property of the widget. To do this, drag & drop the grey block to the area to be configured in the "Style" column.

To the left of this column is the "Style" column, which will contain all the parameters to be configured for the widget being created. This column will change its composition according to the type of widget to be created (chart, label, gauge, grid...). The following chapters detail these parameters for each of the existing widget types.

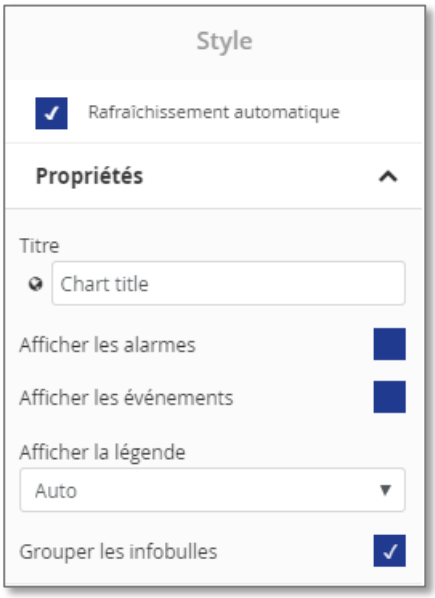
A. Chart

In the case of a 'Graph' widget, the configuration column (Style) consists of the following sections

a) Properties

This section contains the general properties of the widget. In the case of a chart, you can :

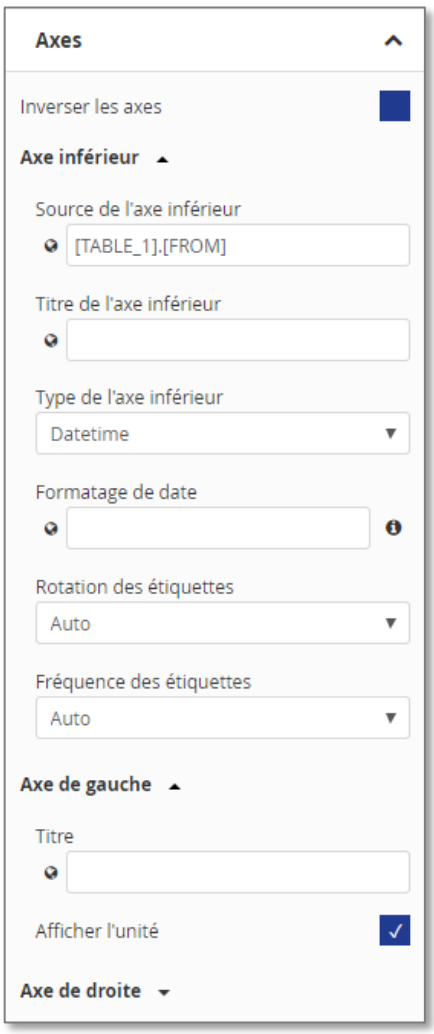
- Define a (multilingual) title for the graphic
- Check or uncheck the display of alarms on the graph
- Check or uncheck the display of events on the graph
- Set the default display of the graph legend (right, bottom, or not displayed)
- Choose whether to have one tooltip per track on the graph, or a grouped tooltip for all tracks at once.



At the top of the "Style" column, there is also an option to disable the automatic refresh. In this case, the graph will be prevented from reloading each time its definition is modified. This can be useful in the case of graphs based on complicated spreadsheets requiring a high execution time. This will avoid untimely loading times each time a configuration field is defined.

b) Axes

This section allows you to configure the axes of the graph.



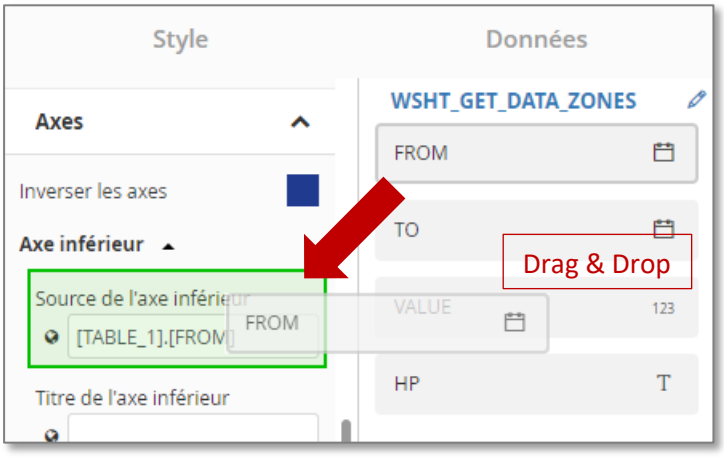
The first option is a checkbox allowing you to invert the X-Y axes if you want horizontal histograms instead of vertical ones, for example.

The lower axis requires a series of values to serve as the X-axis in the graph. This series of values can be a series of dates or something else (e.g. categories, to display a histogram graph).

The set of values that will be used as the X axis should be one of the columns in the widget's source spreadsheet. To do this, drag & drop one of the column names (grey block) from the "Data" column to the "Source of the lower axis" box.

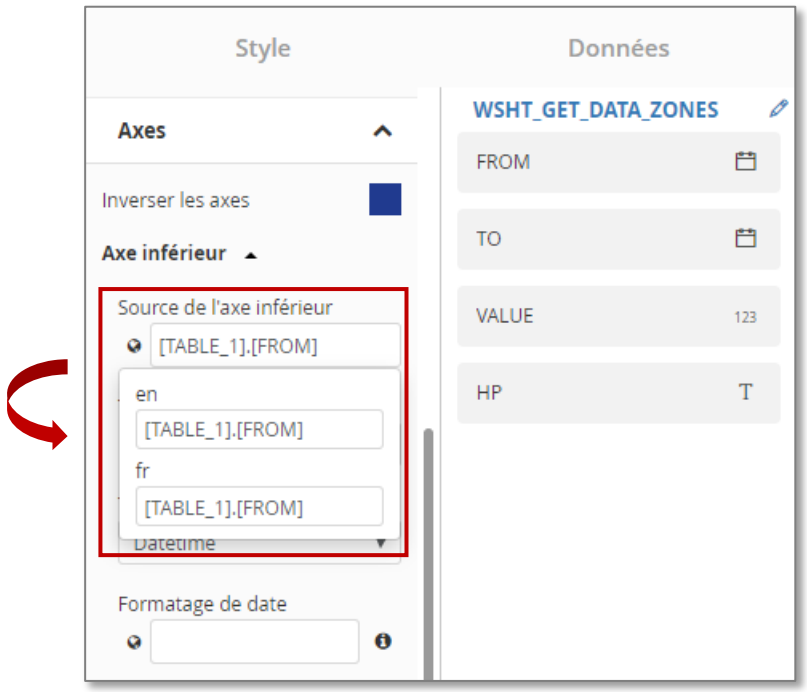
During drag & drop, the areas eligible to receive a source column turn green.

Dragging a grey block from the 'Data' column into an area of the 'Style' column will then associate the graph property with the column in the source spreadsheet. The title of the lower axis can be simple text.



Warning: Some fields in the "Style" column are potentially multilingual (such as the "Source of the lower axis" field). These fields can be used directly like the others, in which case the associated column will be used for all languages, but if you want to associate different columns according to the language of the connected user (or of the recipient in the case of sending emails by automatic tasks), you can use the small planet available next to the field.
Clicking on this planet will open the list of available languages, then a different column can be dragged into each language field if required.

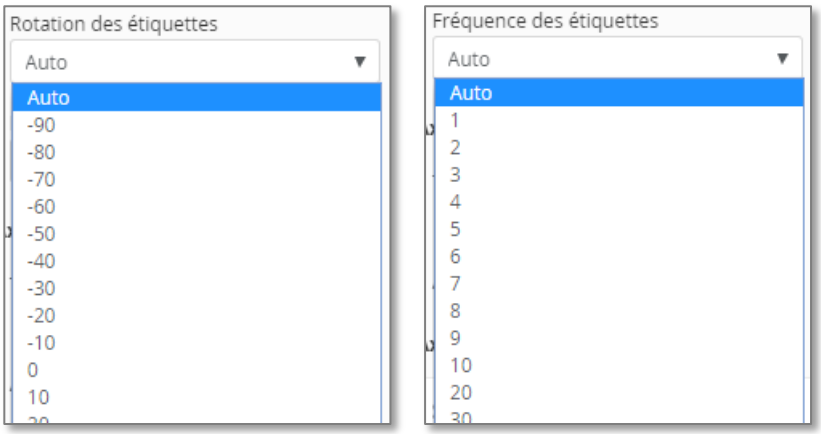
The graph will then adapt and fetch the right column according to the language configured for the connected user (or for the recipient).



The "**Type of lower axis**" field is multiple choice and allows you to indicate whether the graph will display time series (DateTime) or data by category (category).

In the case of a series of dates (DateTime), a multilingual "**Date Formatting**" field will appear, allowing the user to choose the display format of the dates of the axis according to the language of the user. This formatting will be defined by means of the classic time formatting codes "dd MM yyyyy ...". A reminder is available by clicking on the small "info" logo next to the field.

The "**Label Rotation**" field allows you to force a tilt to the values displayed below the lower axis. This field is a drop-down menu, and the angles are calculated from zero (horizontal) in the positive clockwise direction and in the negative trigonometric direction.

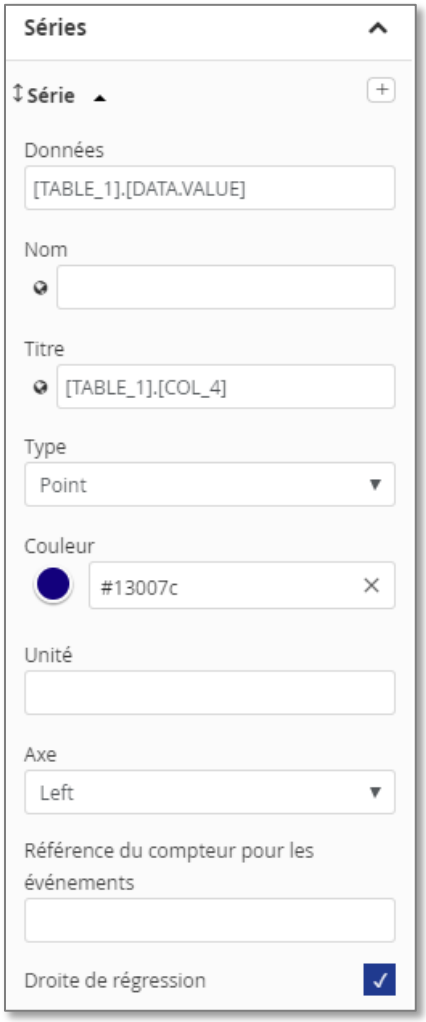


Finally, the "**Label frequency**" field allows you to display a value under all the scales or to do so one scale out of N, "N" being the value entered in this field.

The other axes (left and right) only have options for the title that can be displayed (multilingual text) and whether or not to display the unit on this axis.

c) Series

This is where you can configure the actual numerical series that you want to display in the graph.



At the top right, the small "**+**" button allows you to add an additional series. This will create a new parameter block below the first one, allowing the new series to be configured.

In the parameters of each series, there is a "**Data**" field which will host a column from the source spreadsheet (via a drag & drop similar to that of the lower axis seen above). This column must obviously contain numerical data in order to be displayed in the graph.

The "**Name**" field (multilingual) is used to identify the data set. This name will only be visible in the configuration interface and will have no impact on the display of the graph.

The "**Title**" field (multilingual) will be displayed in the graph legend. This field can contain a hard text or a column from the source sheet.

The **type of graph** is multiple choice and allows to choose the appearance of this series in the graph (line, points, histograms, superimposed histograms...)

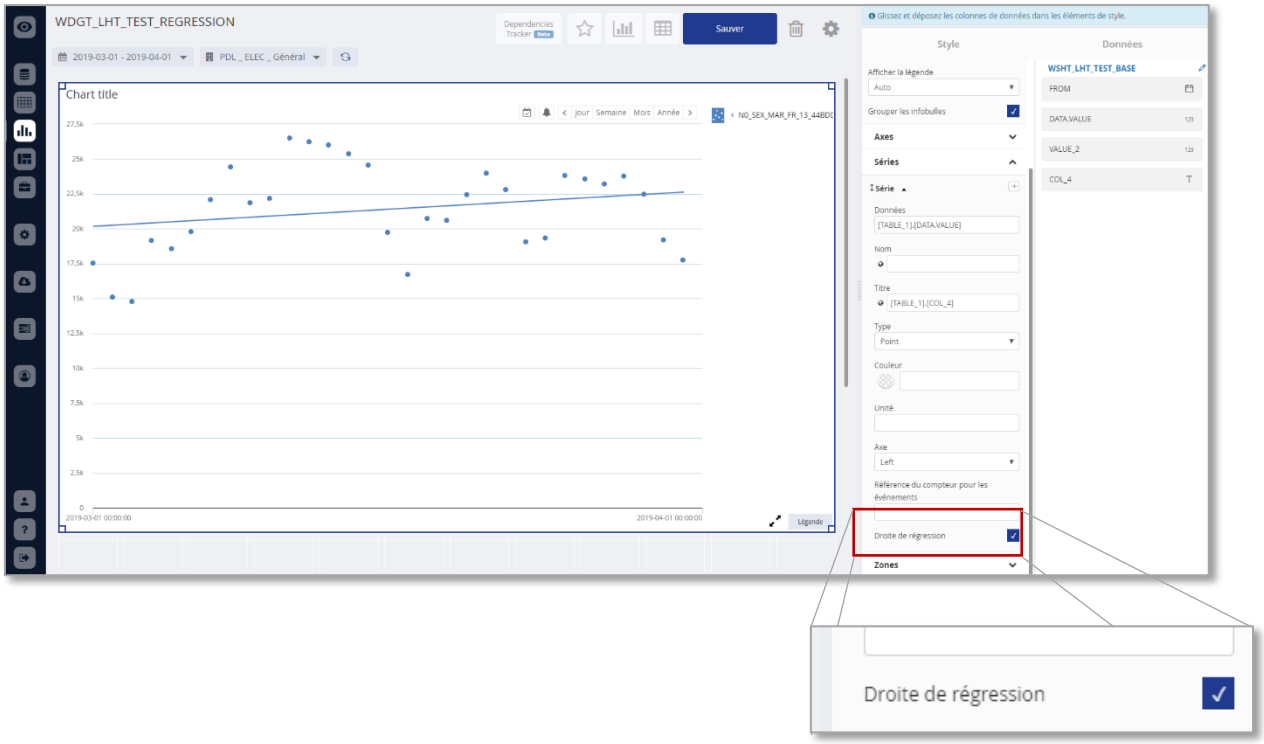
You can then choose the **colour** of the series using a palette or by directly entering the hexadecimal code of the colour in the corresponding field (e.g.: #ff0000 = Red).

The **unit** indicates in which unit the series is displayed.

The "**Axis**" field allows you to define whether the series is displayed based on the left or right axis in the graph.

The following field allows you to indicate the **reference of the counter** to which the alarms or events to be displayed are associated. This field is only useful if the "display alarms" or "display events" box at the top of the "Style" column has been checked.
The fact is that the active selection when generating a widget contains entities or counters. Alarms are stored not on the counters, but on the channels in those counters. Without any prior indication, EMM (JOOL) would have to retrieve the alarms of all the channels attached to the selected counter(s) and display them all, which can be overloaded and quickly unreadable. The solution is to tell EMM (JOOL) where it should go to retrieve the events you want to display, which is done via this field in the "Style" column.

The last field is a checkbox to indicate whether to display a linear **regression line** to the series. This regression line will automatically take on the same colour as the series of values to which it relates.



d) Special case: the sub-series

If you want to display several series in the same graph, you just have to define several series in the widget, and associate to each series definition, a column of the source spreadsheet which will contain the data of this series. This has the advantage of allowing you to configure each series as you wish, the first in blue points, the second in red histograms...

On the other hand, this requires the construction of a source spreadsheet with one column per series, which may require several joins between preparatory sheets, and which may cause problems in case of series with different time steps.

Date	Série 1	Série 2	Série 3
01/01/2019	50	12	7
02/01/2019	51	11	8
03/01/2019	49	10	9
04/01/2019	45	12	8
05/01/2019	55	13	9
06/01/2019	39	12	7
07/01/2019	42	11	10
08/01/2019	45	12	6
09/01/2019	50	10	6

Another possibility is to create a single series definition, but which will contain data from several profiles. (see example opposite)

This will require a source sheet with a date column and a value column, and that these columns contain all the data for each profile in succession.

Then, to be able to differentiate the origins of each piece of data, an additional column will be needed which will contain the name or reference of the original counter.

This column will therefore contain as many different values as there are source profiles in the spreadsheet.

A single series definition can then be created in the graphical widget, by putting the "Date" column on the horizontal axis, and the "Value" column in the series, then dragging the "Series" column into the "Title" field of the series definition.

EMM (JOOL) will then detect that it has several title values, and it will create, based on this single definition, a different sub-series per value in the "Series" column.

If no colour is defined for the definition of this macro-series, each of the sub-series will be assigned a colour according to a pre-defined order in EMM (JOOL). If a colour has been assigned, it will be applied to all the sub-series (not recommended for reasons of legibility)

Date	Valeur	Série
01/01/2019	50	Série 1
02/01/2019	51	Série 1
03/01/2019	49	Série 1
04/01/2019	45	Série 1
05/01/2019	55	Série 1
06/01/2019	39	Série 1
07/01/2019	42	Série 1
08/01/2019	45	Série 1
09/01/2019	50	Série 1
01/01/2019	12	Série 2
02/01/2019	11	Série 2
03/01/2019	10	Série 2
04/01/2019	12	Série 2
05/01/2019	13	Série 2
06/01/2019	12	Série 2
07/01/2019	11	Série 2
08/01/2019	12	Série 2
09/01/2019	10	Série 2
01/01/2019	7	Série 3
02/01/2019	8	Série 3
03/01/2019	9	Série 3
04/01/2019	8	Série 3
05/01/2019	9	Série 3
06/01/2019	7	Série 3
07/01/2019	10	Série 3
08/01/2019	6	Série 3
09/01/2019	6	Série 3

This has the advantage that the construction of the source sheet is extremely simple, and if several profiles have different time steps, there will be no problem to concatenate them into one set of values. (whereas in the previous case, the join between tables based on the date column could cause problems if the time steps were too different)

The only drawback is that the form of all the subseries will automatically be the same. It is impossible to say that the first one should be in points and the second one in histograms for example.

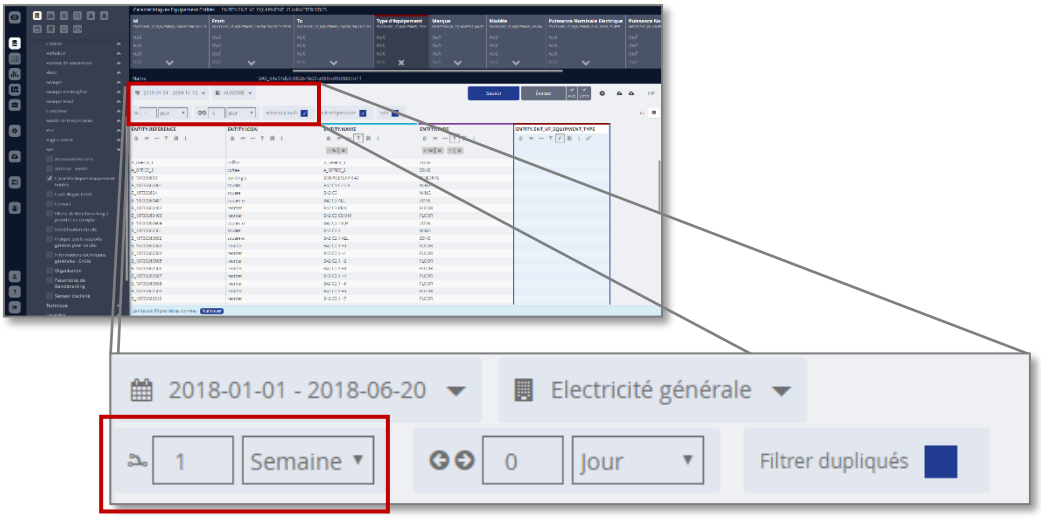
If the "Regression line" box is ticked, a line will be calculated for each sub-series, and will be coloured for ease of reading.

e) Special case: "Linerange" type series

The "Linerange" type of series is special in that it does not display the data itself, but rather its general trend and the area in which the values vary. The general trend is in fact a graph representing the aggregated values according to a self-adjusting time step, depending on the total size of the observation period.

In order to have an auto-adaptive graph, we can have as a source a Worksheet based on a DataSet aggregated with an automatic time step. (see earlier, chapter on DataSet formatting). Note that it is also possible to perform this kind of aggregation in JQL language directly in a worksheet or in a formula.

Reminder



In the choices available for this time step, we therefore take the "Automatic" time step which will define the aggregation time step dynamically according to the total duration of the observation period.

- If the duration of the time selection exceeds 500 days: 1 year
- If the duration of the time selection exceeds 300 : 1 month
- If the duration of the time selection exceeds 10 days: 1 day
- If the duration of the time selection exceeds 1 day: 1 hour
- Otherwise: No aggregation

In the source DataSet, several columns containing the following information will be needed:

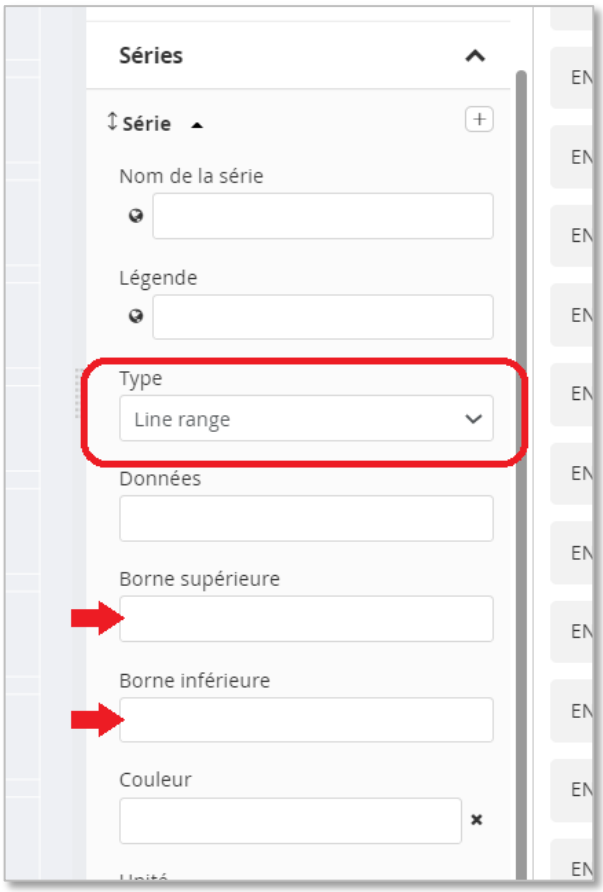
- **Dates and Data**: the data to be displayed, aggregated according to the most relevant method (e.g. sum for consumption)

- **Lower bound**: the same data, but this time aggregated to the minimum value (thus returning the minimum value over the time interval)

- **Upper bound**: again the same data, but this time aggregated to the maximum value (thus returning the maximum value over the time interval)

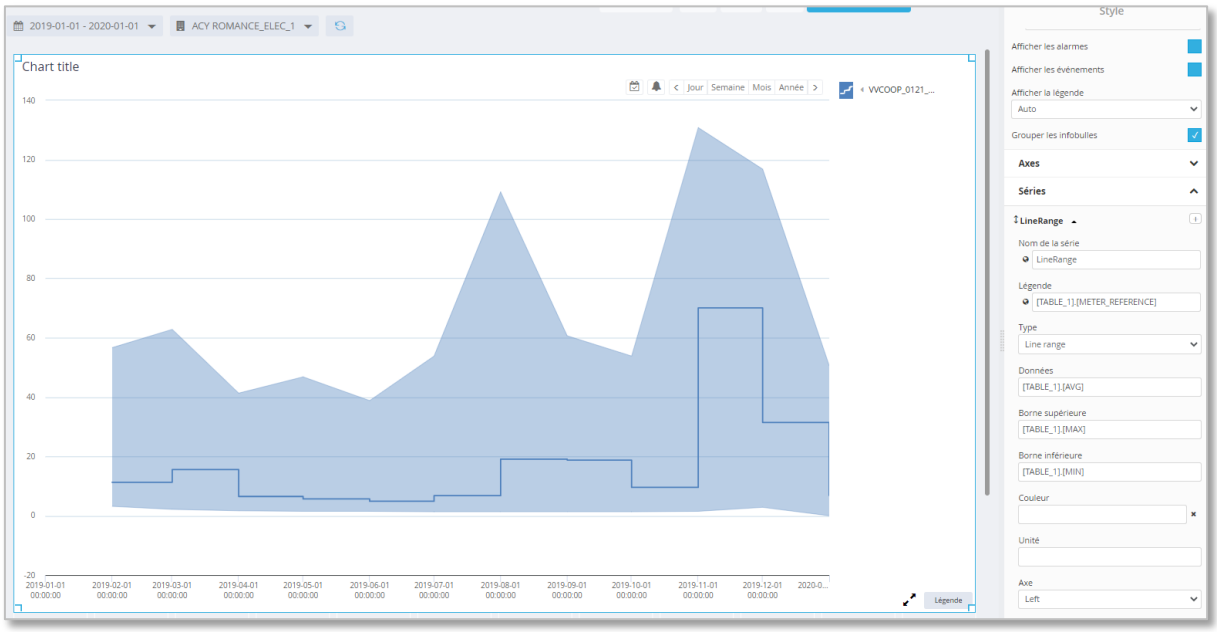
Once in the widget (after passing through a worksheet which recovers and possibly transforms the data if necessary), we can then choose the type of series "Linerange". By choosing this option, two additional fields will appear in the editor: "Upper boundary" and "Lower boundary".

By dragging these columns into the corresponding fields of the widget editor, a graph will be obtained with a curve auto-fitted to the observation period, and a coloured area around this curve, indicating the bounds of the values taken in each interval.

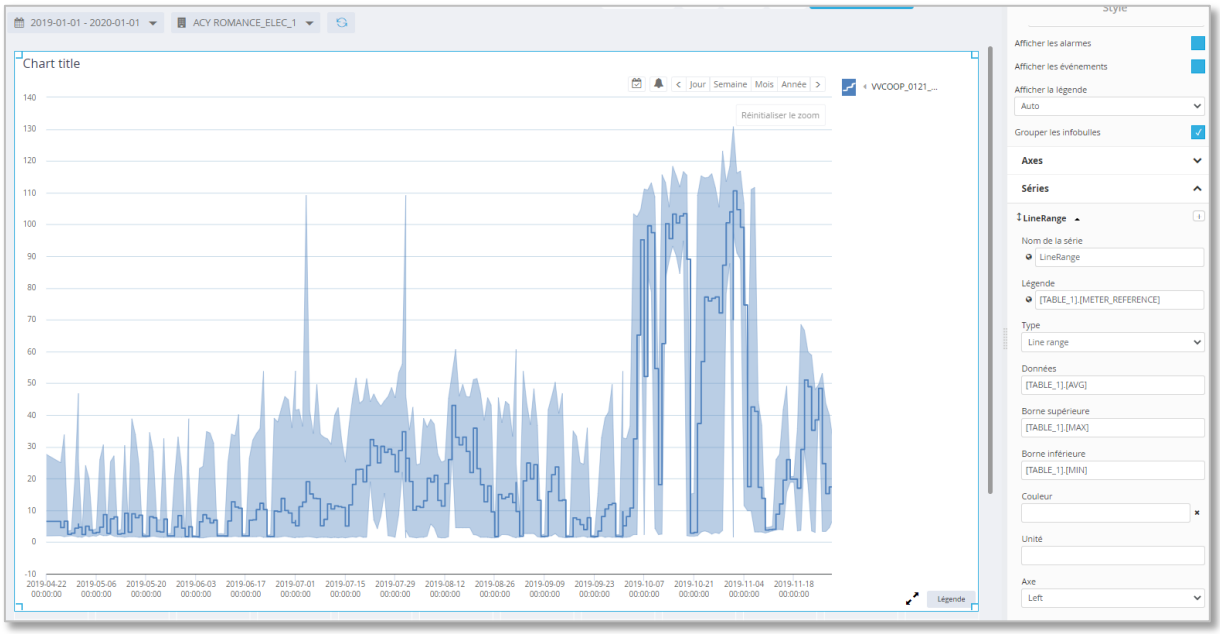


If the graph is zoomed in, the data will be recalculated and if the period passes certain thresholds, the automatic aggregation will adapt to the situation.

Example with a period of one year -> monthly aggregation



The same graph, zoomed to show a few months -> Switch to daily aggregation



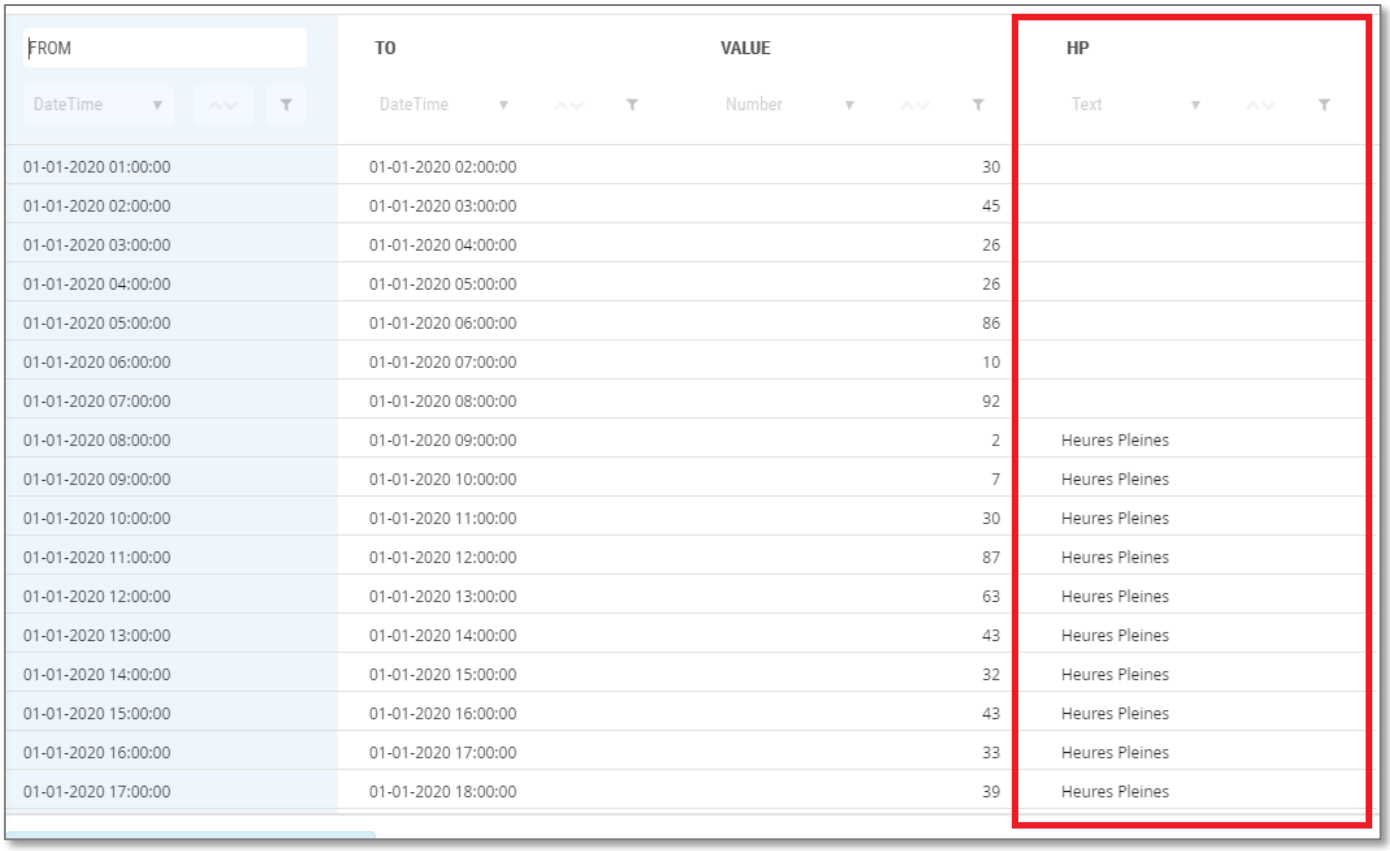
...and the same graph, but zoomed in on a few days (<10 days) -> switch to hourly aggregation

f) Zones

In addition to displaying series of values, it is also possible to display series of background colour zones in the chart. To do this, open the "Zones" section of the "Style" column.

To make this work, the source worksheet will need to contain a column prepared for use as a series of fields. This column will be filled with boxes either empty or with a given value, and it is the presence of the value in a box that will cause the corresponding time slice (From-To) to be coloured.

In the example below, the source worksheet has a column "HP" which contains the text "Peak hours" if the time slot has an hour number between 8 and 17. The off-peak time slots do not contain any information.



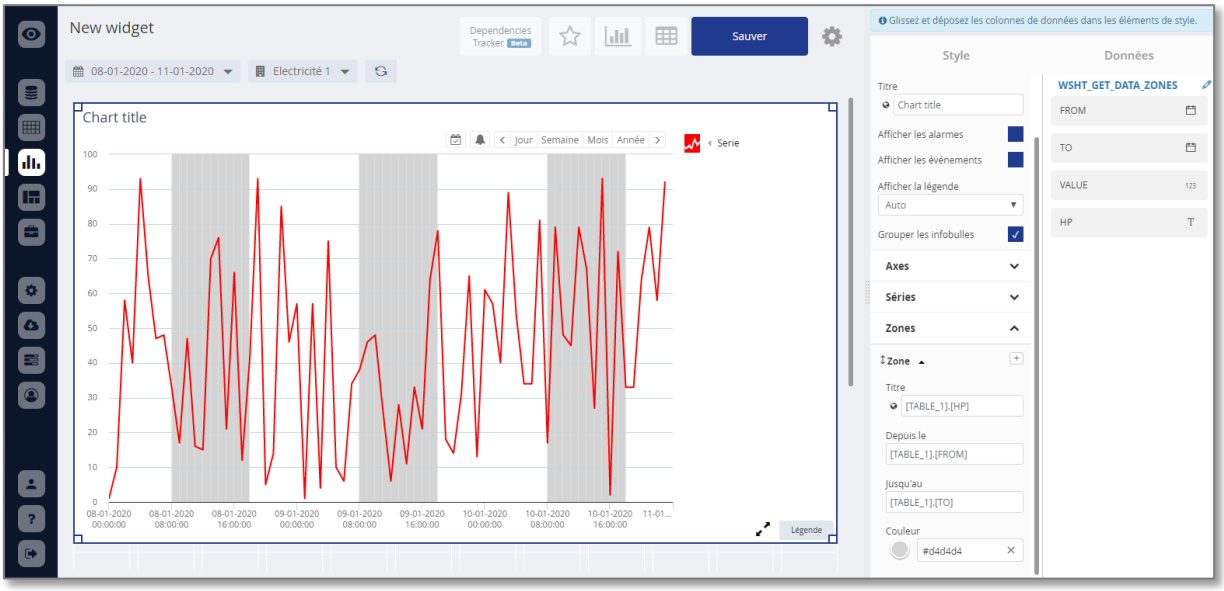
FROM	TO	VALUE	HP
DateTime	DateTime	Number	Text
01-01-2020 01:00:00	01-01-2020 02:00:00	30	
01-01-2020 02:00:00	01-01-2020 03:00:00	45	
01-01-2020 03:00:00	01-01-2020 04:00:00	26	
01-01-2020 04:00:00	01-01-2020 05:00:00	26	
01-01-2020 05:00:00	01-01-2020 06:00:00	86	
01-01-2020 06:00:00	01-01-2020 07:00:00	10	
01-01-2020 07:00:00	01-01-2020 08:00:00	92	
01-01-2020 08:00:00	01-01-2020 09:00:00	2	Heures Pleines
01-01-2020 09:00:00	01-01-2020 10:00:00	7	Heures Pleines
01-01-2020 10:00:00	01-01-2020 11:00:00	30	Heures Pleines
01-01-2020 11:00:00	01-01-2020 12:00:00	87	Heures Pleines
01-01-2020 12:00:00	01-01-2020 13:00:00	63	Heures Pleines
01-01-2020 13:00:00	01-01-2020 14:00:00	43	Heures Pleines
01-01-2020 14:00:00	01-01-2020 15:00:00	32	Heures Pleines
01-01-2020 15:00:00	01-01-2020 16:00:00	43	Heures Pleines
01-01-2020 16:00:00	01-01-2020 17:00:00	33	Heures Pleines
01-01-2020 17:00:00	01-01-2020 18:00:00	39	Heures Pleines

The idea is that every time slice that contains something in its HP column will be coloured, and the slices where HP is empty will keep a white background.

Once this source worksheet is prepared, we can create the widget, and in the "Zones" section, we must configure the following fields:

- **Title**: will be associated with the HP column of the example, the one that contains or not a text to indicate whether or not to colour the graph background.
- **From** and **To**: will be associated with the "From" and "To" columns, indicating the beginning and end of each time slot.
- **Colour**: choice of the colour to be applied to the graph background for the marked time slices.

The result will be something like this (as in the example above).

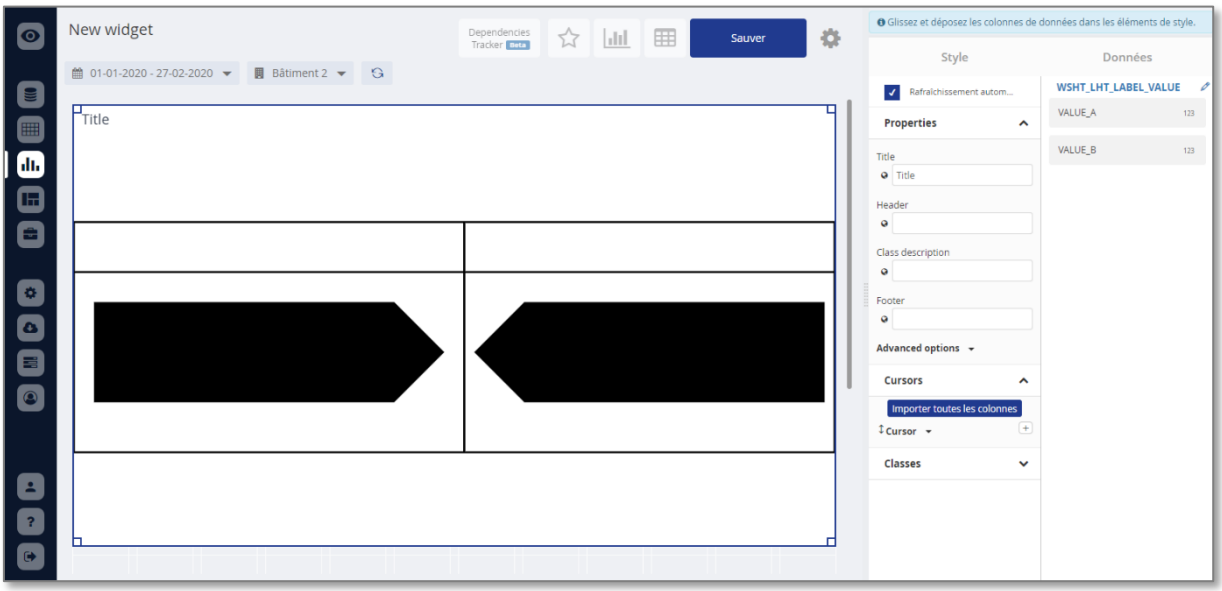


Each time slot where the HP column contains text has a coloured background in the requested colour. Here we have all the full hours with a light grey background.

B. Label (Class Rating)

Based on the Energy Label principle, these displays can show value classes and related sliders.

The interface for creating this type of widget is therefore similar to that for creating graphics. Only the "Style" column changes in composition, as well as the final result displayed, of course.

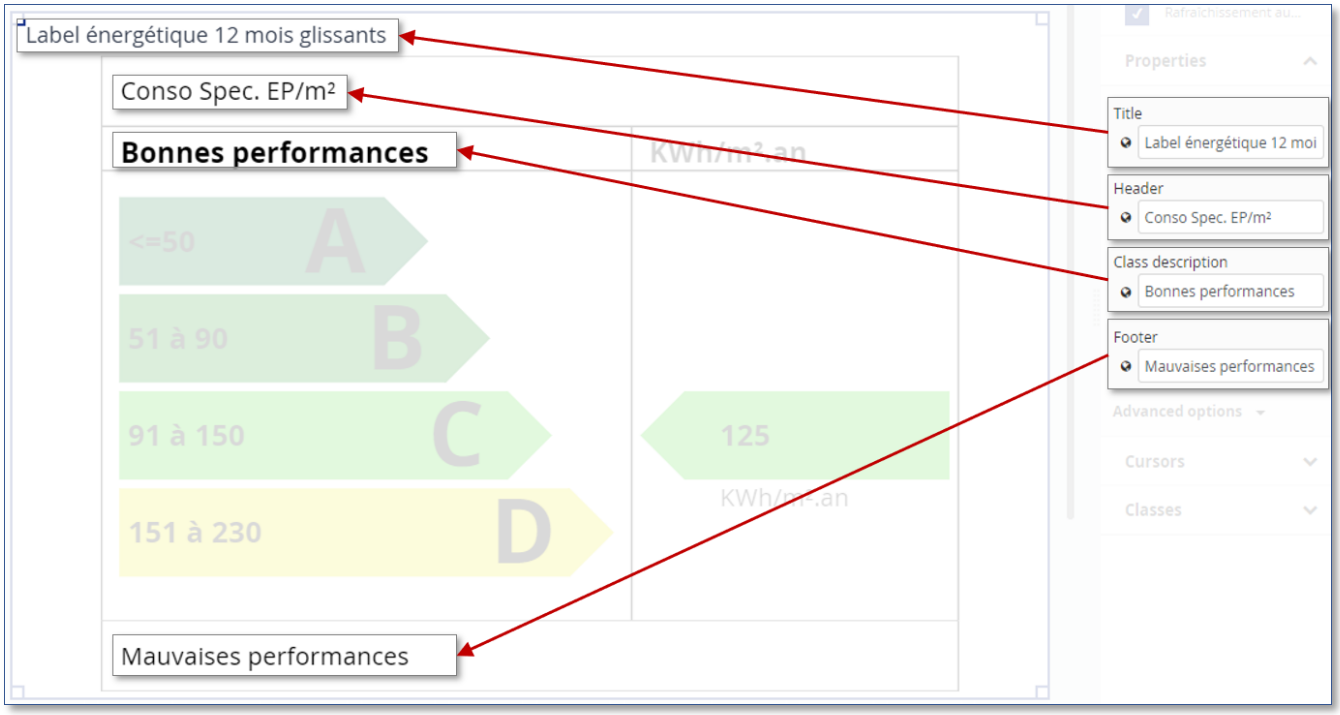


In the "Style" column, we find the following sections:

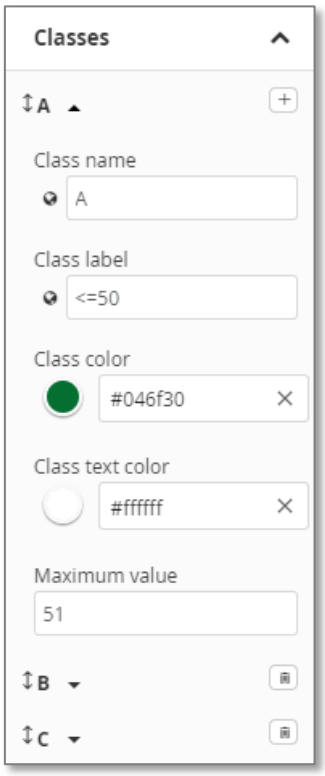
a) Properties

This section contains the general properties of the widget. In the case of a label, it contains the following properties:

- **Title:** the title of the widget
- **Header**: a header to be put at the very top of the label. If the property remains empty, the box disappears completely from the label display.
- **Class description:** the title to put at the top of the class column.
- **Footer**: a footer that appears at the bottom of the widget. If this property is left empty, the footer box will disappear completely from the display.



b) Classes



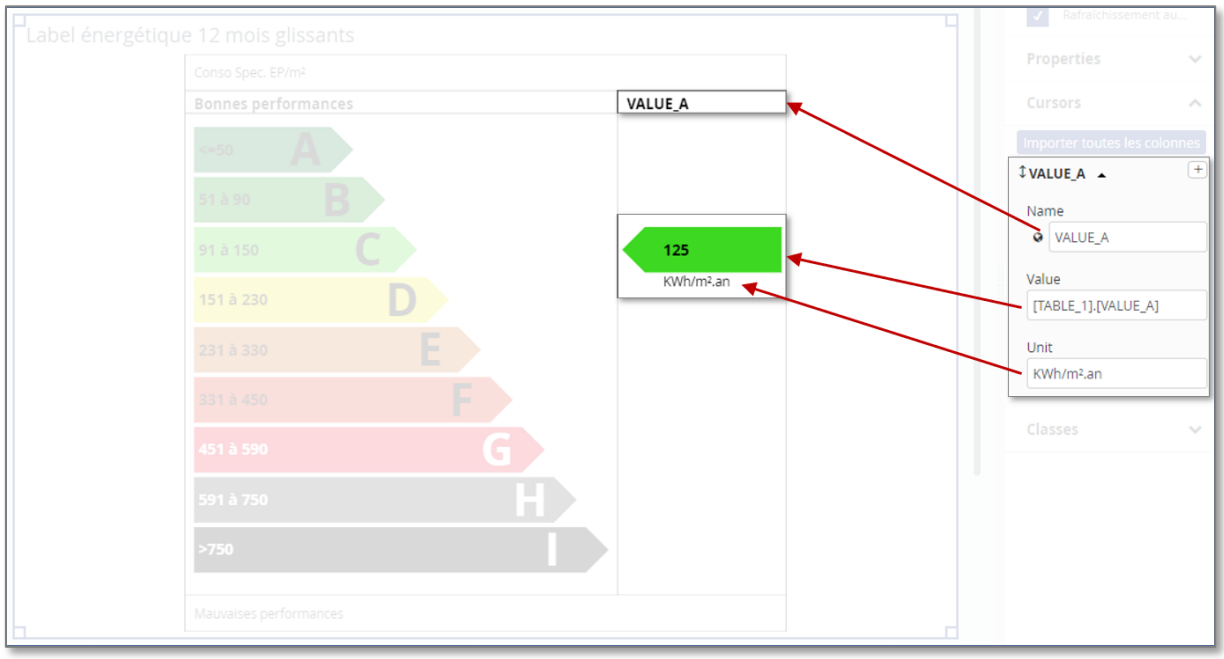
At the bottom of the "Style" column list is the "Classes" section, where you can define each of the label's classes individually.

Each class will be defined by :

- **Class name** (multilingual): the name of the class, usually a letter, which will be displayed on its coloured banner.
- **Class label** (multilingual): the class legend, a text explaining its condition.
- **Class color**: the colour of the associated colour strip, which can be chosen using a palette (click on the colour circle) or by directly entering the hexadecimal code of the colour in the field provided.
- **Class text color**: the colour of the text displayed on the colour banner
- **Maximum value**: the maximum value allowed in the class in question. The minimum value is the maximum limit of the previous class.

c) Cursors

Finally, in this section, we can define the cursors to be displayed on the label. As for the series in a chart, we can add cursors via the small "+" button at the top right of this section.



Each cursor will be defined by 3 properties:

- **Name**: Its name, which will be visible at the top of the column where it will be displayed

- **Value**: Its value, defined via a column in the source sheet (assigned via drag & drop, as for numerical series in charts)

- **Unit** (optional): The unit of its value, which will appear below the cursor.

At the top of this section, there is also an "import all columns" button, which serves as a shortcut to manually import all the columns in the source sheet, each into a different cursor. This option can be useful if there are several cursors to import. It will then be sufficient to delete the unnecessary cursors via the small "delete" buttons next to their names.

C. Form

The "Form" widget can be used to automatically generate forms for encoding reports, which can then be opened as dashboards directly in EMM (JOOL), or sent by email so that users can encode their reports from outside EMM (JOOL), in kiosk mode.

Like all other widgets, they are based on a worksheet that will retrieve the meters and channels to be read.

a) Source spreadsheet

This spreadsheet must contain the following columns:

- **METER:** the reference of the meter to be read
- **SENSOR:** the reference for the channel to be surveyed
- **SENSOR.TYPE :** the type of channel to be measured

These columns are essential for the widget to know which channels to read and in which meters they are located.

In addition, when creating a new form, a default HTML template is created for the display of the form page, and this template uses certain information that is useful to add to the source spreadsheet.

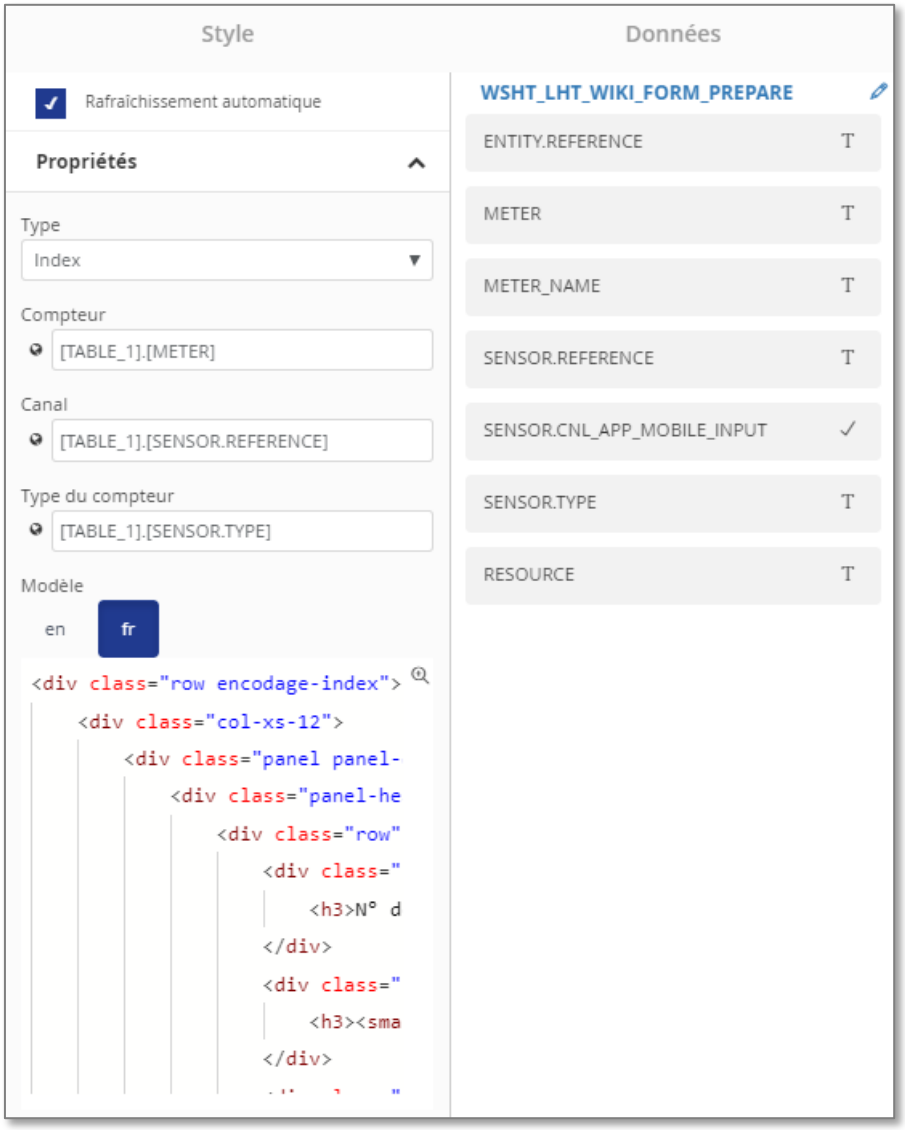
- **METER:** the reference of the counter to be read, also structurally necessary, but it is explicitly this column name that is found in the template.
- **METER_NAME:** the name of the meter
- **RESOURCE:** the (translated) resource measured by the meter

The names given above are those used by the HTML template. For quick implementation, these names should be given to the corresponding columns in the spreadsheet.

b) Building the widget

As with all widgets, the configuration interface allows you to drag and drop columns into useful fields. Here we have three essential fields:

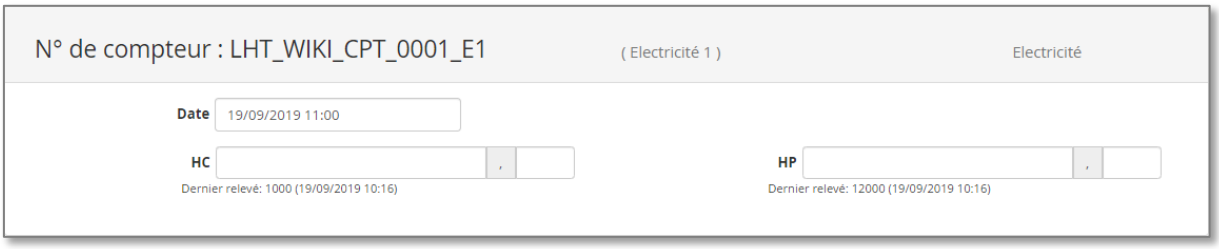
- **Counter**: which must be associated with the column giving the reference of the counter concerned (METER)
- **Channel**: the column giving the reference of the channels to be surveyed
- **Type**: the column returning the type of the channel, to serve as a label for each feed field



The "Type" drop-down menu at the top of the column currently only contains "Index", meaning that the forms can currently only be used to retrieve readings (indexes or consos in fact), but in the near future, other functionalities will make it possible to create forms that retrieve properties, comments... and this field will be used to activate the correct information retrieval mechanism.

c) Default HTML template

The default HTML template defines the form that the block for each counter will take in the encoding form.



The header contains the reference and name of the counter and the resource it measures. Below this header, there is an encoding area which displays

- **The reading date:** by default, the current date is displayed, but it can be changed by clicking on the date field. This will open a small calendar allowing you to choose the actual date of reading.

- For each channel to be read on this counter, there will then be a field for entering the **value read**. To the left of the field, we find the type of channel concerned, and under the field, we have information on the last encoding (value and date)

In the case of a counter where several channels are to be encoded, there will be several fields in this encoding area.

The form widget as a whole will contain as many reading blocks as there are meters to be read, all on the same HTML template. At the very bottom of this form, a "Send" button will upload the data entered into EMM (JOOL). A small green message at the bottom right of the screen will indicate if the data has been uploaded.

Note: the form does not update automatically, a refresh of the page (F5) is useful to see the newly encoded values appear.

d) Editing the HTML template

For advanced users with some knowledge of HTML, it is possible to modify the default template of the form. This already allows you to modify the column names listed above, but also to modify the appearance of the headers of each counter. (Note: the encoding area is not customisable, only the header)

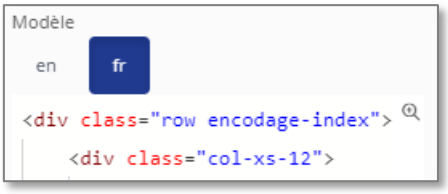
In this model, the structure of the header is found, which can be modified as desired. The creation of CSS classes is possible, as is the use of EMM (JOOL) syntax to create conditional displays.

For example, if the channels of a counter have already been read less than 30 days ago, the header will appear in green instead of the default grey. To do this, the last encoding dates will have to be retrieved, which can be done either directly in the HTML template code, or retrieved from a column in the source spreadsheet if it contains a column returning this date.

Columns can therefore be added to the source spreadsheet to help customise the headings of the relief blocks.

In the HTML template for the form, you will find the @@form@@ statement. This indicates where EMM (JOOL) should insert the encoding area, which will contain the date field and the various relief fields.

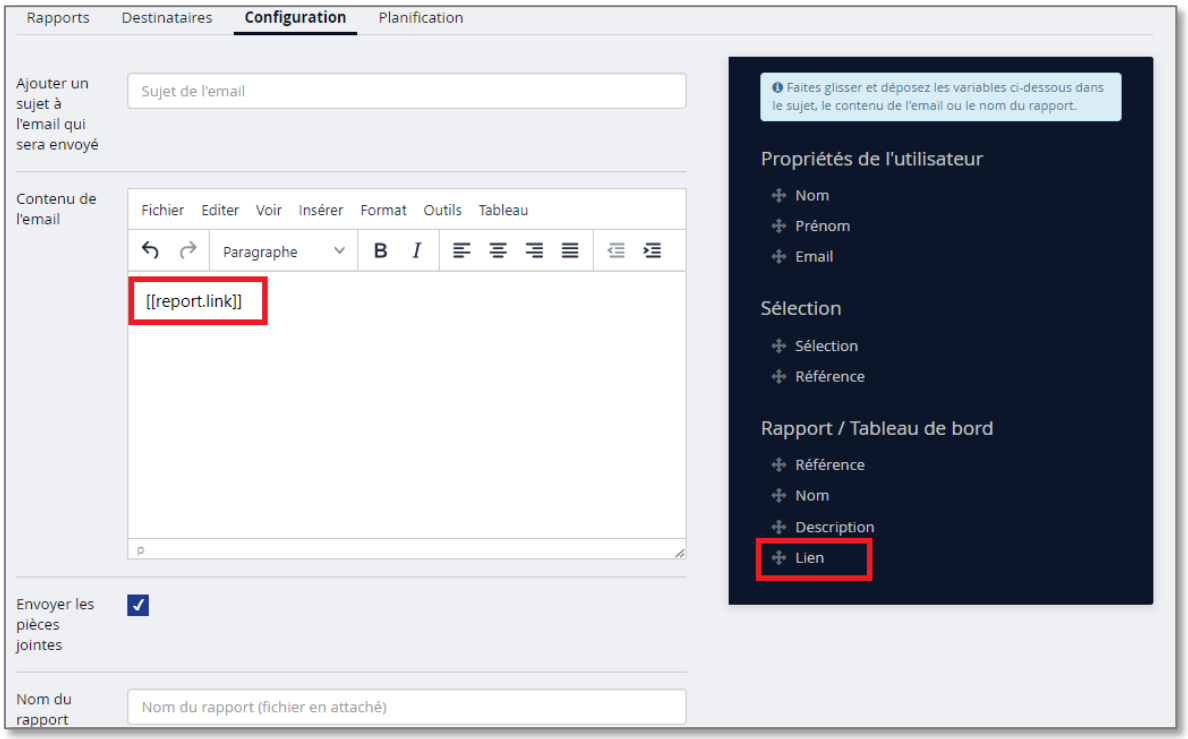
Note: The HTML template can be defined in different languages so that the titles are adapted to the language of the recipient. To see the different templates in the different languages, you can use the en or fr buttons to switch between them.



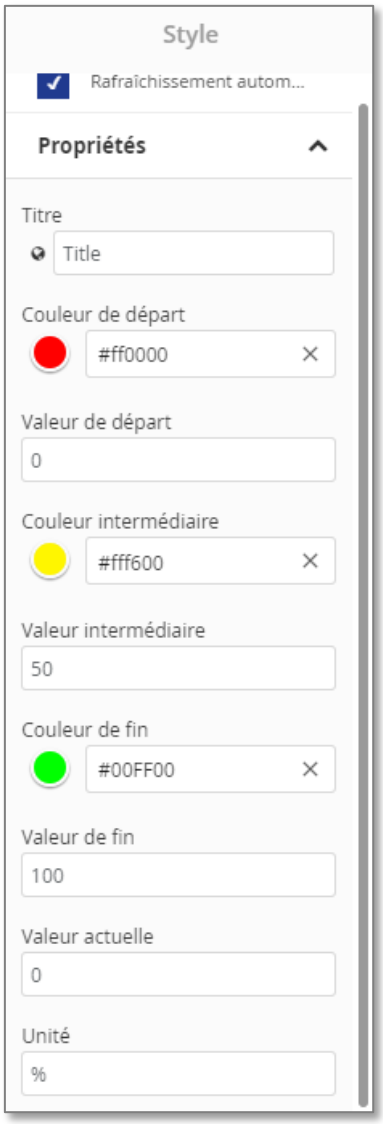
e) Sending the encoding form

When configuring report sending tasks, a "link" can now be dragged from the right hand area to the content of the email. This will add the line [[report.link]] which will inject a URL link to the report in question into the email.

In the case of an encoding form, this will therefore link to the web page generated on the fly displaying the encoding form which can then be filled in by the recipient of the email.

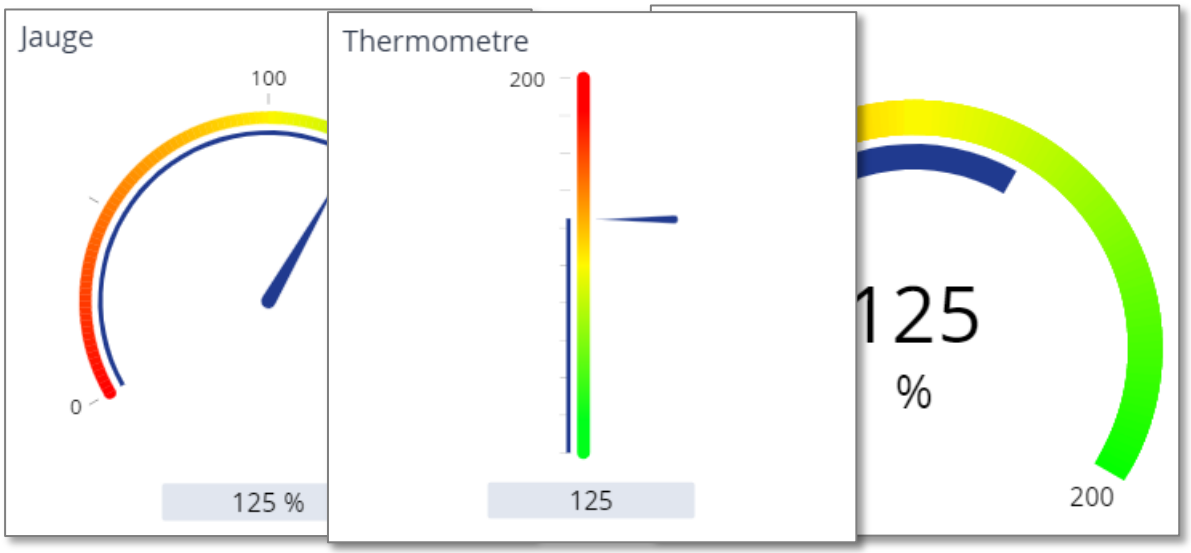


D. Gauge (Circular, Performance or Thermometer)



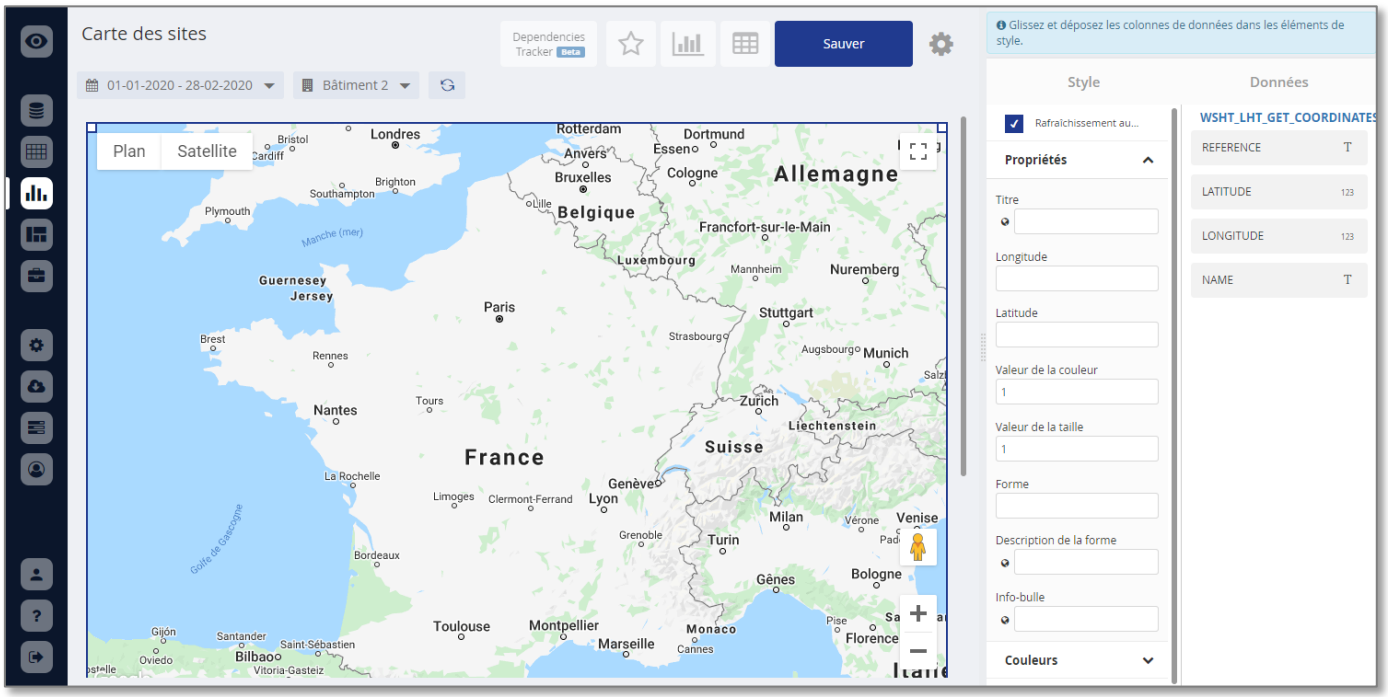
Gauges, whether circular, performance or thermometer, have the same configuration interface. Their "Style" column contains a single "Properties" section where you can define

- **Title** (multilingual): the title of the widget
- **Start colour**: the bottom colour of the scale, configurable via a palette or directly via the hexadecimal colour code.
- **Starting value**: the numerical value associated with the bottom of the scale. This will be the minimum value considered in the gauge, and can either be hard-coded or associated with a column in the source spreadsheet.
- **Intermediate colour/value:** similar to the starting colour/value, but for the middle of the gauge.
- **Colour/End value**: same as above, but for the end of the gauge.
- **Current value**: the value to be displayed in the gauge. This value is usually taken from one of the columns in the source precalculation sheet.
- **Unit** (optional): the unit of the displayed value.


Jauge
100
0
125 %
Thermometre
200
125
125
%
200

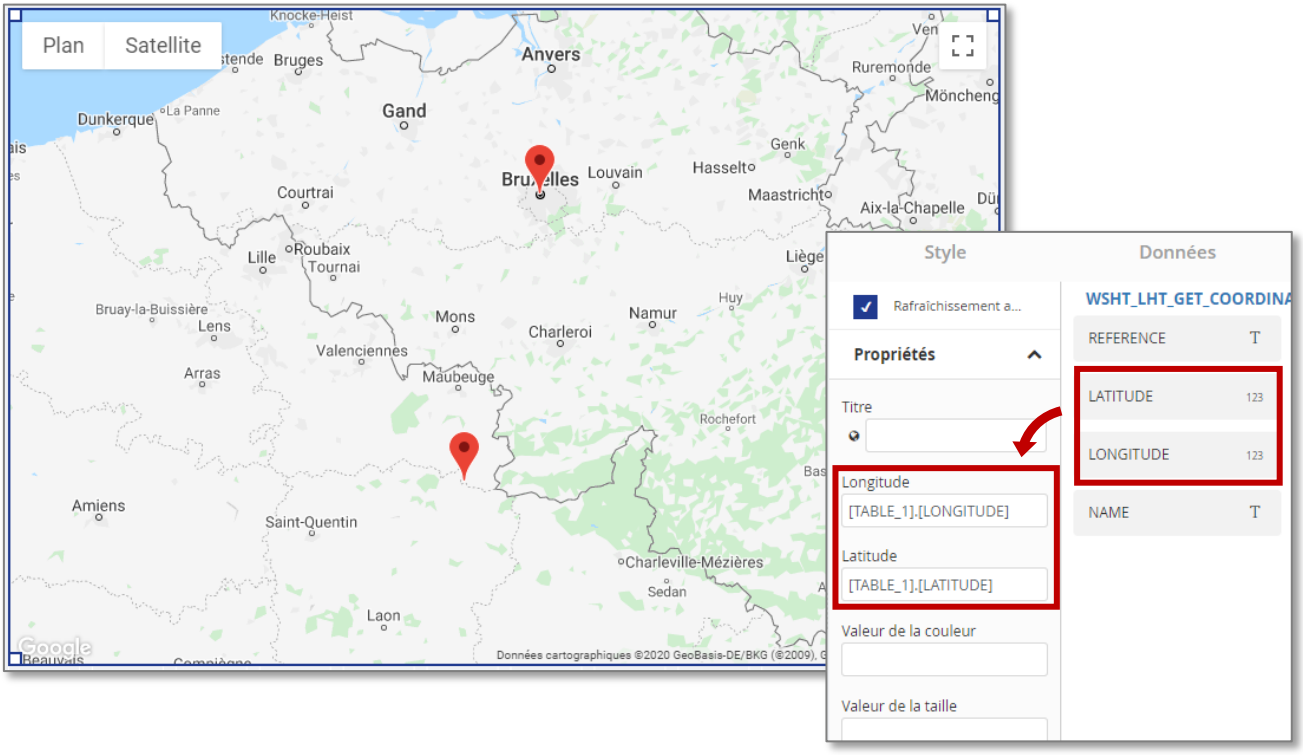
E. Map (Google map)

The EMM (JOOL) map widget allows you to locate entities/counters on a world map, using the Latitude and Longitude properties that will need to be encoded in the entities to be displayed. This widget looks like this:



The source worksheet will obviously have to contain two columns with the latitudes and longitudes of the entities to be displayed. These columns will be dragged over the Latitude and Longitude fields of the "Style" column.

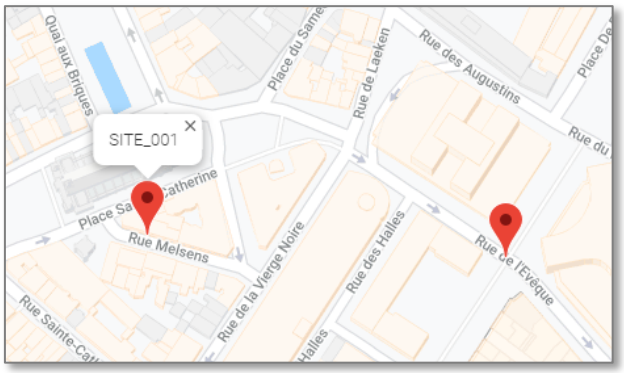
This first basic configuration already allows you to create a simple map:

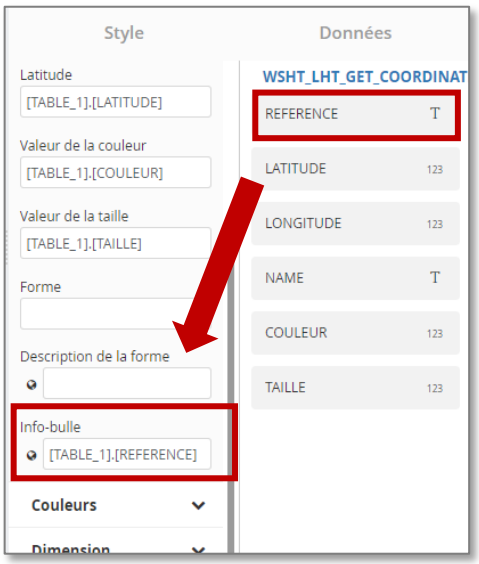


a) Tooltips

In addition to placing a marker on the map, it can be assigned a tooltip. The text that will appear in the tooltip is defined in the "Tooltip" field in the "Style" column of the widget editor, and it can be filled in hard with constant text for everyone (not very interesting), or it can be associated with a column from the source worksheet.

If, for example, we associate the column "REFERENCE" with the field "Tooltip", we will obtain for each marker on the map with a tooltip containing the reference of the site concerned.





A more complete caption (text) can of course be prepared in the source worksheet column, to link to a tooltip.

Note: the text in the tooltip can be written in HTML. It will then be interpreted dynamically in the tooltip, which makes it possible to make line breaks, to put text in bold or in colour if necessary.

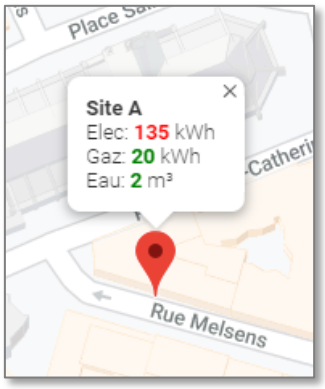
Advanced example: with the text below injected into the "tooltip" field, we obtain the result shown opposite (this text would be constructed in one of the columns of the source worksheet).

< b>Site A< br>

Elec: < span style="color:red">135 kWh< br>

Gas: < span style="color:green">20 kWh< br>

Water: < span style="color:green">2 m³

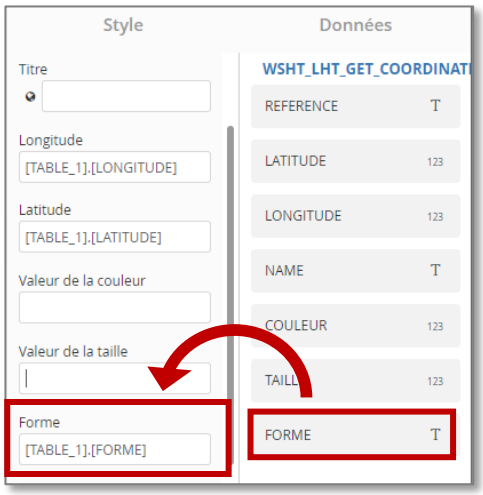


Note: The tooltip is a multilingual field. It can therefore be assigned different columns in the source worksheet depending on the language of the user concerned.

b) Shape of the marker

In order to further customise the map, each marker can also be given a shape of your choice. To do this, we must use the "Shape" field in the "Style" column, and provide it with a letter which will identify an existing shape in EMM (JOOL). This letter can be hard-coded in the box, or taken from a column in the source worksheet if you want to have different shapes for different sites.

Example: The letter "n" represents a circle and the number "6" represents a pointed triangle. If you arrange for the source worksheet to contain a column with "6" for sites and "n" for weather stations, you can use this column in the "Shape" field and on the map you will have a triangle to indicate the weather stations and a circle to indicate the sites.

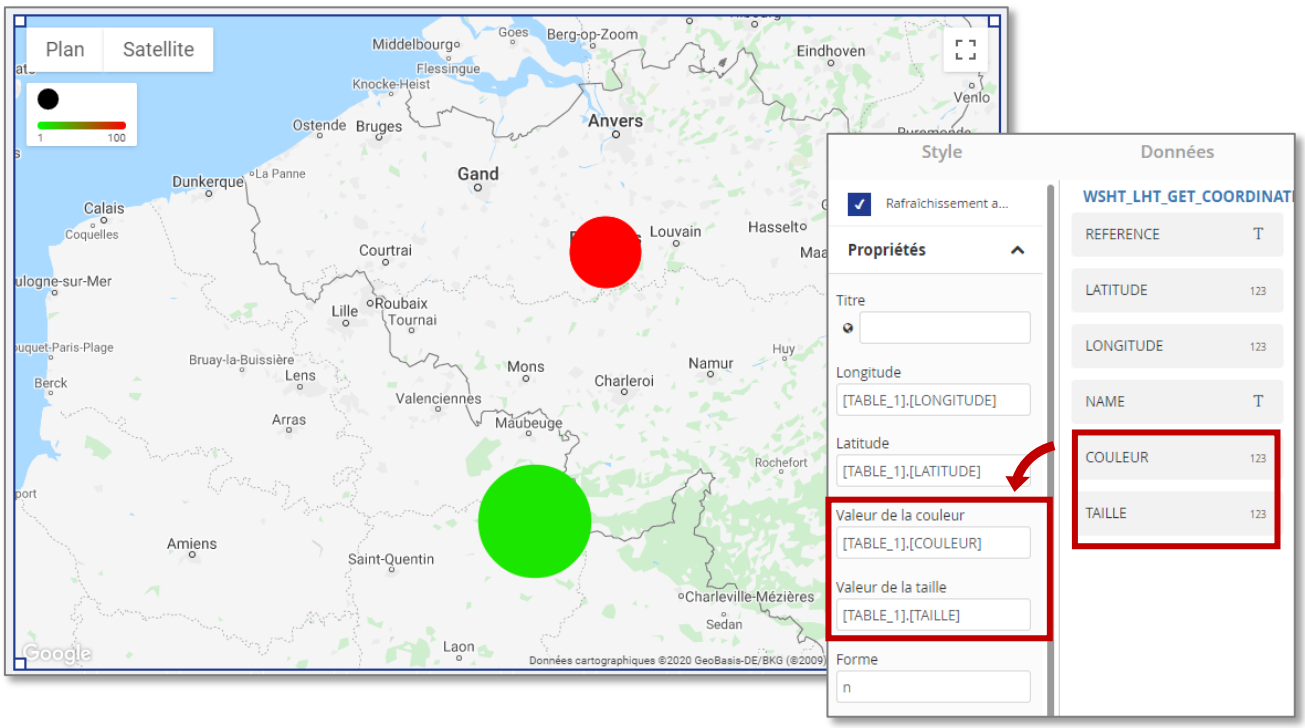


c) Marker size and colour

Finally, it is possible to enhance the map by specifying a **size** and **colour** for the positioning markers, which can vary according to values given in the source worksheet.

For example, we could have dots on the map indicating sites, in green if they have low consumption per square metre and in red if they do not (going through the whole range of colour gradations between the two for the different values of consumption/m²), and we could associate the size of the marker with the surface area of the site concerned. We would then have large green dots for large sites with good energy performance, and small red dots for small sites with poor surface consumption (and all possible mixes of size and colour according to these values).

The values used to define the size and colour of each point must therefore be taken from the source worksheet and dragged onto the "Colour value" and "Size value" fields in the "Style" column



By default, the colour and size scales are set to green to red, and 1 to 40 pixels, both for property values ranging from 1 to 100. These scales can of course be changed by opening the 'Colours' and 'Dimensions' sections of the 'Style' column of the widget.

Colour scale

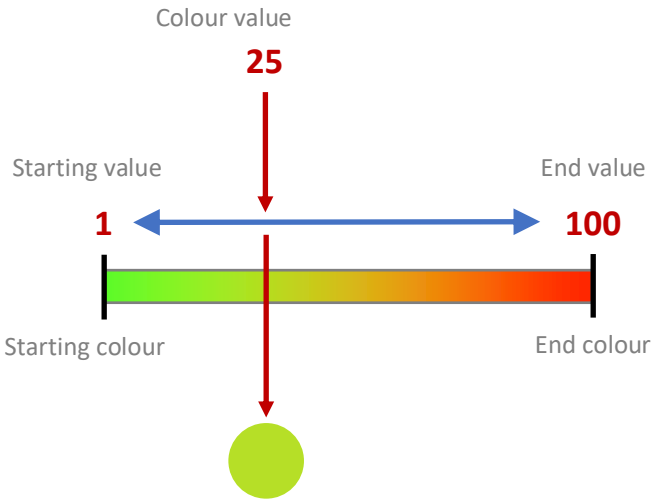
In the "Colours" section, the colour scale of the markers that will be displayed on the map can be defined.

This is done by specifying two extreme colours for the scale: "**Start colour**" and "**End colour**", and then associating these colours with (numerical) start and end values.

The start and end values define the range of numerical values that the measured property will be able to traverse and the closer this value is to the start value, the closer the marker will be to the start colour (and vice versa).

In the example opposite, a value of 1 will correspond to a green marker, a value of 100 will correspond to a red marker, and a value of 50 will correspond to half the gradient of the colour scale between green and red.

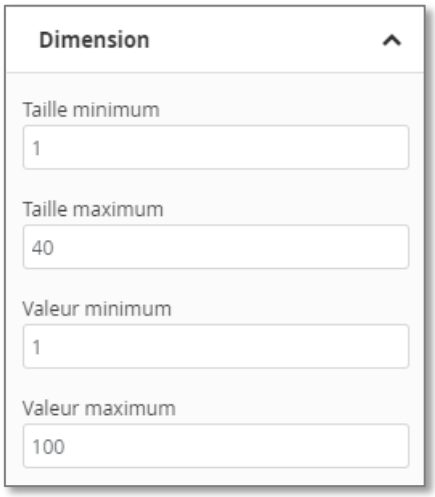
Example: A value of 25 will be placed on the scale from 1 to 100, and its colour will therefore correspond to a green at the quarter of the scale from green to red.



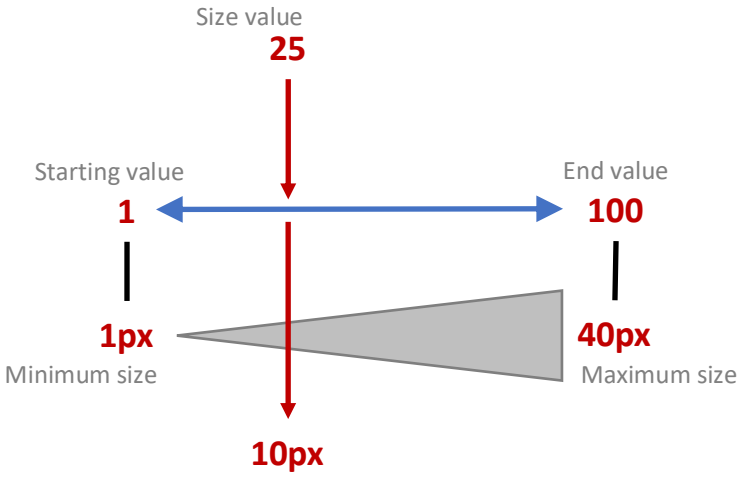
Size scale

Following the same principle as for colours, a starting size (in pixels) and an ending size are defined, and extreme values of the scale are associated with these sizes.

The value that will then be used to determine the size of the point to be displayed will be positioned on this size scale and will be assigned a size in pixels proportional to its position on the scale.



Example: A value of 25 will be placed on the scale from 1 to 100, and its colour will therefore correspond to a green at the quarter of the scale from green to red.



F. Grid data

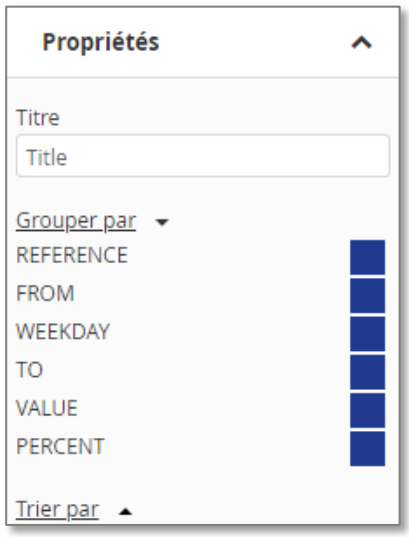
A grid is a widget that displays an array of values, pre-calculated in a spreadsheet, as a report or dashboard component.

The creation interface is similar to the other widgets, and the "Style" column contains two sections:

- **Properties**: section containing options for viewing the table
- **Columns**: list of columns to be displayed, based on the data present in the source spreadsheet.

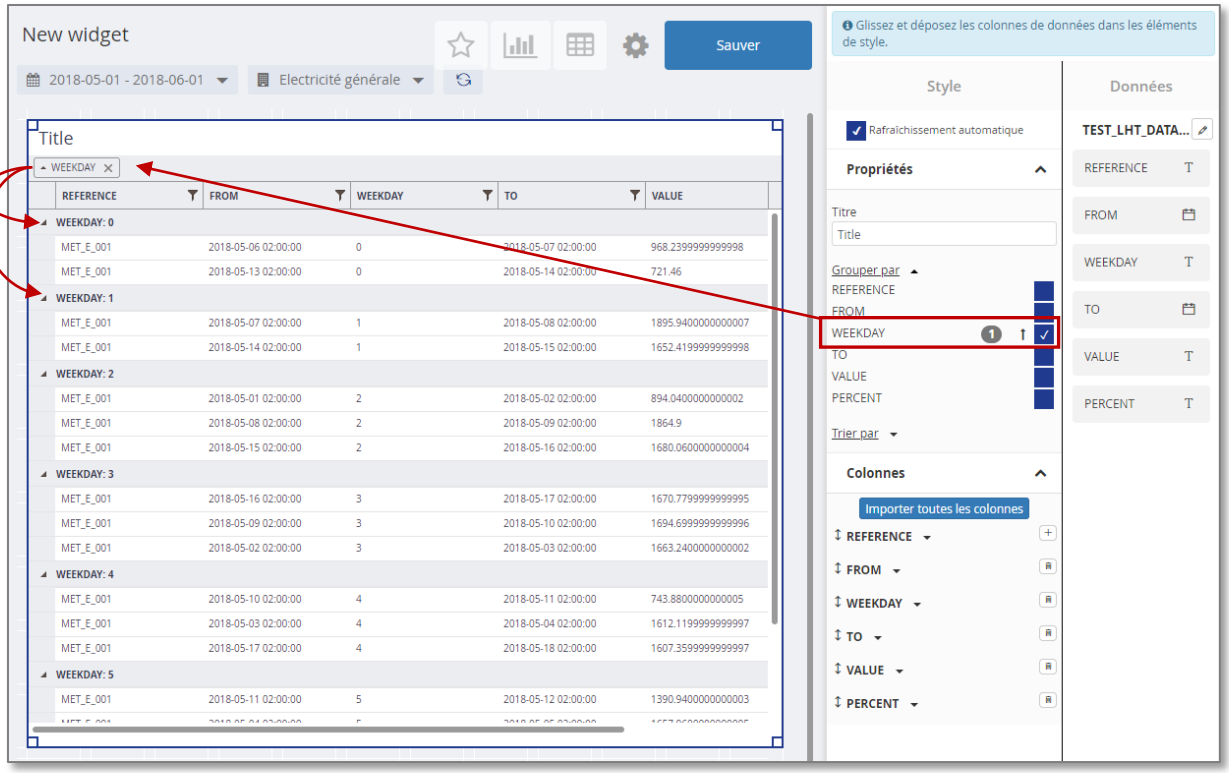
a) Properties

In the "**Properties**" section, there is the usual field containing the title of the Widget, and then two areas listing all the columns in the grid, and allowing you to request a grouping or a filter on certain columns.

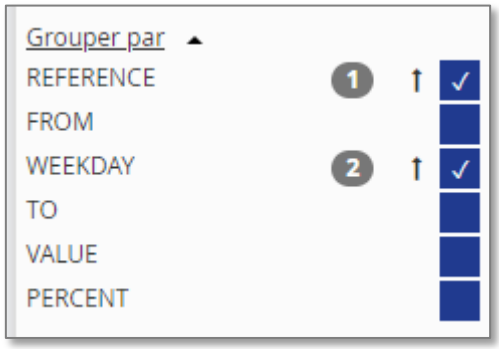


When a column is **grouped** according to a column (by checking the box associated with its column name), the table is reorganised to display one block per value of the chosen column, and in each block, all the rows of the table with this value in the chosen column will be found.

Result of a grouping :



It is also possible to make several consecutive sub-groupings by ticking other boxes. In this case, the groups will be made in cascade; the rows will be distributed according to the column of the first grouping, then, in each grouping, a sub-grouping will be made on the basis of the second grouped column...



In the "Style" column, the numbers that appear next to the checked boxes indicate the order of grouping, and the small arrow is a button that allows you to choose the order in which the groups should appear (in ascending or descending order...)

The "**Sort by**" section is similar and allows you to sort the columns, so that the rows can be sorted in any order you wish.

Several sortings can be performed in cascade, with a second sorting allowing the rows that would have had the same value in the first sorting to be sorted.

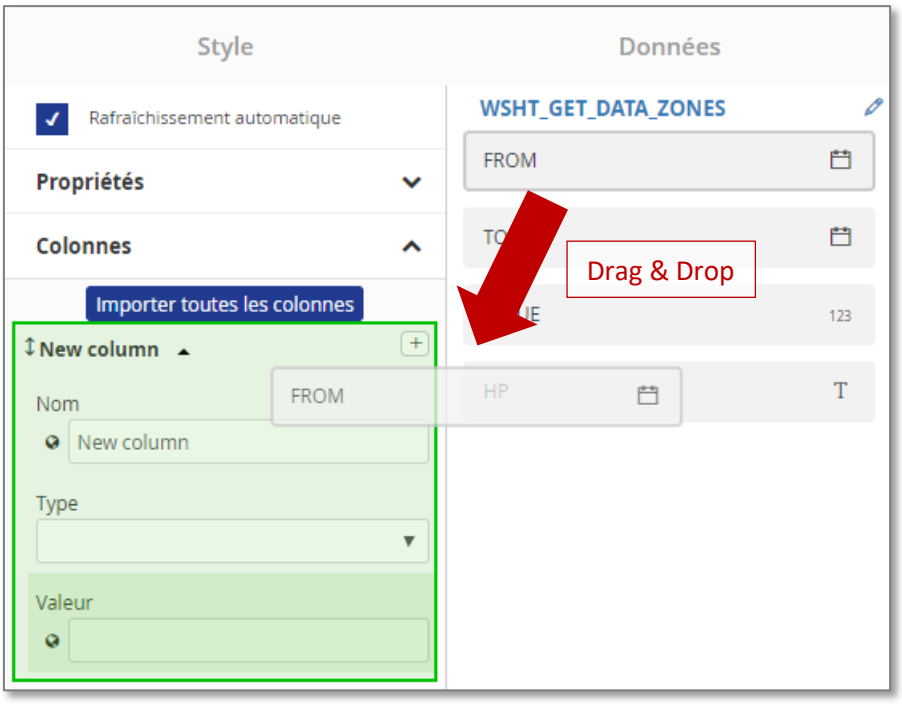
You will then find the arrow allowing you to choose the ascending or descending order of the values, as well as the number indicating the order to be applied for the sorting cascade.

b) Columns

Still in the "Style" column, in the "**Columns**" section, you can define the columns that should be retrieved from the source spreadsheet and displayed in the table (grid). It is on the basis of this list of configured columns that the above column lists are constructed (for grouping and sorting).

An "import all columns" button allows you to directly retrieve all the columns from the source worksheet, instead of configuring them all manually.

Manually, the small "+" button adds a column, and a drag & drop of the column name from the "Data" column (on the right, listing all the columns in the source worksheet) to the definition of the column to be created will add the desired column from the source worksheet into the presentation grid.



The values for each field are filled in automatically at the time of drag & drop, but can be changed manually if required.

Depending on the type of values contained in the created column, a series of additional fields will be available to complete the configuration.

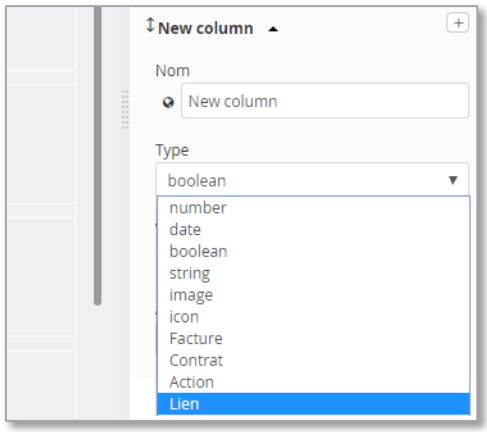
- For dates, a multilingual "Date formatting" field allows you to specify in which format the date should be displayed in the table, possibly depending on the language of the logged-in user or the report recipient.
- For numerical values, a field allows you to define the formatting of the numbers (thousands separators, decimal separator...)
- For all types, an aggregation field is available, allowing you to ask EMM (JOOL) to aggregate the column according to one of the methods allowed by the type of value (number for texts, sum/average/min/max... for numbers...)

Again, each column will have a small 'trash' button to the right of its name in the 'Style' column, to delete it if necessary.

In the "Style" column, to the left of the name of each configured column, there is a double arrow which can be used to enter the column to move it around the grid. The grid columns are displayed from left to right in the order in which the columns were configured (i.e. from top to bottom in the "Style" column)

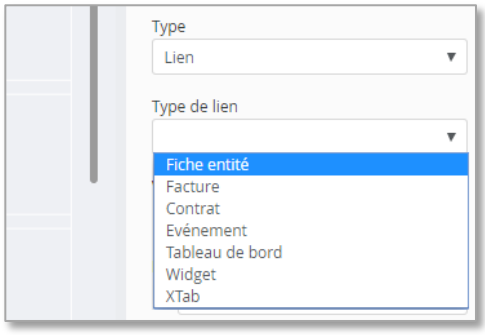
c) Links

In the columns of the grid, in addition to the classic types (numeric, date, boolean, text...) it is possible to choose the "Link" type which allows to create a dynamic link in the grid, which will send to the object linked to the click.



The following object types can be linked:

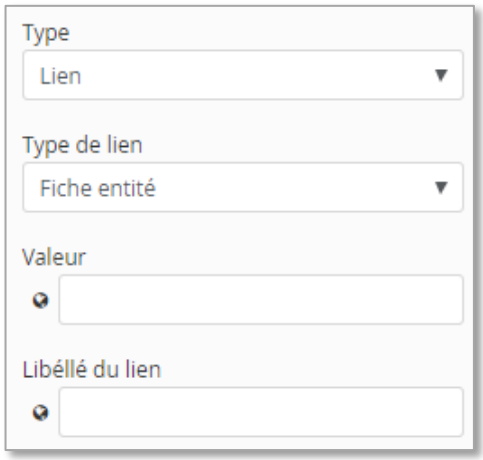
- Entity sheet
- Invoice
- Contract
- Event
- Dashboard
- Widget
- XTab



Once a link type has been chosen, two or three fields appear to define the object to be linked.

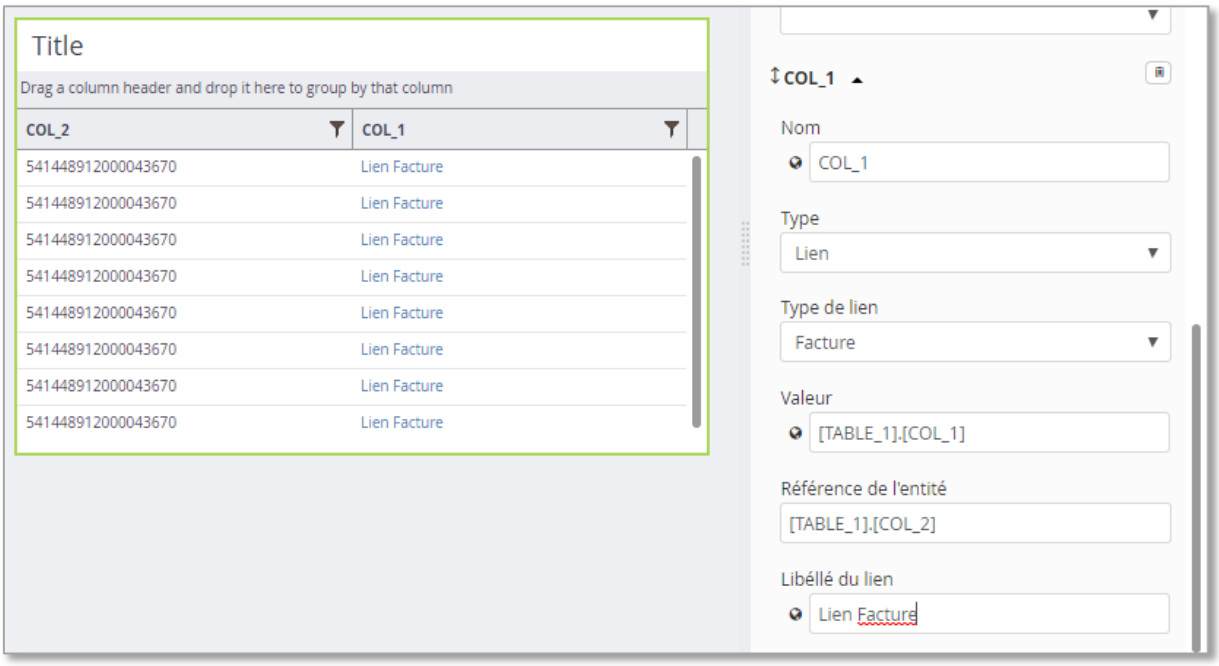
The "**Value**" field will receive the reference of the item to be linked (in hardcopy or via a column of the source worksheet) and the "**Link label**" field will receive the text to be displayed on the link (again in hardcopy or via a worksheet column).

In the case of a link to an invoice, a third field will appear to identify the reference of the **entity** to which the invoice is attached.

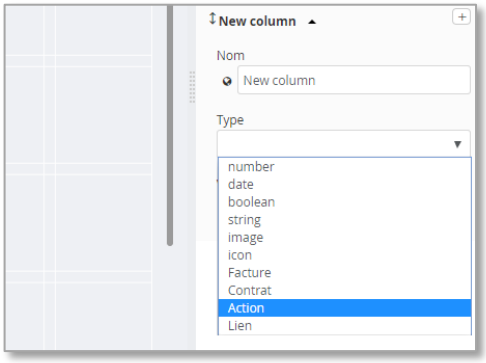


When all the fields are correctly configured, the grid will return dynamic links in the "**link**" column which, in view mode, will open the linked object.

Below is an example of a grid listing the invoices, with the "COL_1" column of the source worksheet containing the invoice reference and the "COL_2" column containing the counter reference. In this example, the clickable link will say "Invoice Link" (here, in hard copy)



d) Actions

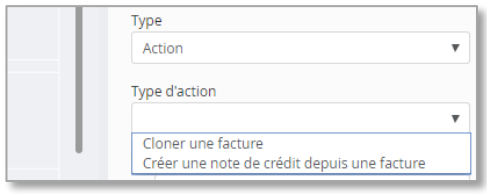


The "**Action**" type allows you to create a button in the grid, which will perform a defined action on demand.

The actions currently implemented are the following:

- **Clone an invoice**: on the basis of a given invoice, creates an identical copy of it (except for a "_CLONE" suffix to its reference) and opens the record of this new invoice so that the user can edit it.

- **Create a credit note** from an invoice: based on a given invoice, creates a credit note with the information of the selected invoice, and opens the credit note file for the user to edit.



When one of these actions is chosen, new fields will appear to configure the action precisely.

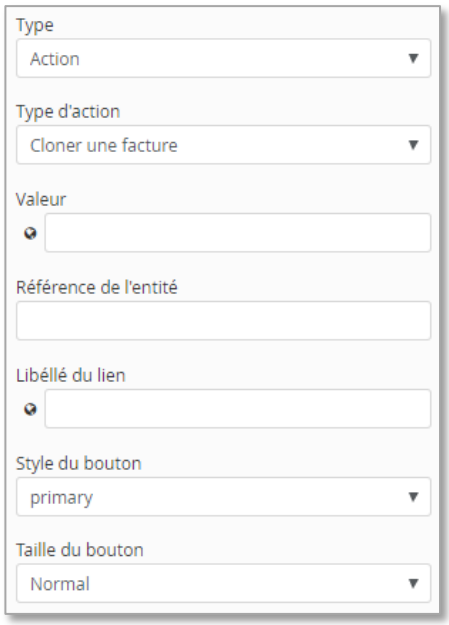
- **Value**: must contain the reference of the invoice used as a basis for the action (the two current actions are based on an invoice). Once again, this reference can be hardcoded, or retrieved from a column of the source worksheet.

- **Entity reference**: will contain the reference of the counter to which the invoice is attached.

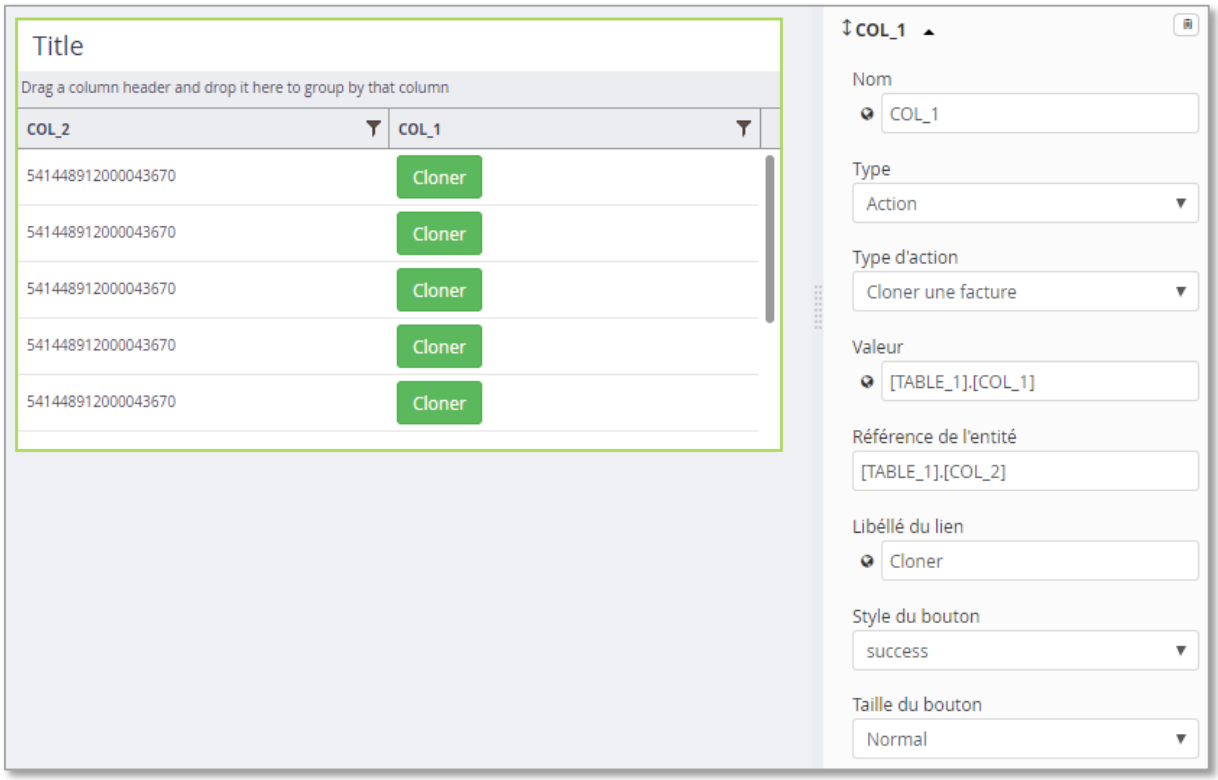
- **Link label**: contains the text that will be displayed on the action button in the grid.

- **Button Style**: Opens a drop-down menu to choose one of the predefined colours for the button.

- **Button size**: allows you to choose between different sizes for the button.



When all the fields are correctly configured, the grid will return buttons in the "**action**" column of the chosen colour and size, which will allow the user to perform the requested action directly from the grid displayed in viewer mode.



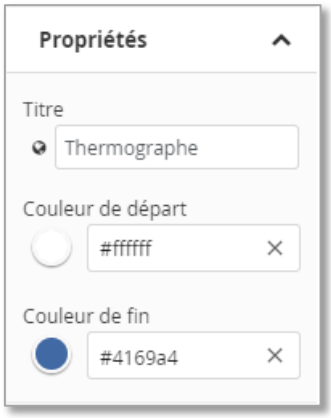
G. Matrix graph (HeatMap)

A matrix graph (also sometimes called a "thermograph") is a graph that places the 24 hours of the day on the X-axis and the list of days in the chosen context on the Y-axis. The graph then receives a data profile and for each value of this profile, it displays a coloured box at the corresponding date and time, the colour of the box being chosen on a pre-configured scale according to the value of the profile.

This type of graph allows energy companies to quickly detect certain information. Here, for example,

we can see that weekends have a less intense hue, so we can deduce that the meter consumes less at weekends. Similarly, we can see that most of the activity takes place between 6 and 4pm every day.

The configuration of this type of widget is relatively simple. You start by giving a title and configuring the colour scale to be used in the "Properties" section of the "Style" column of the widget.



The start colour will be the lowest value of the profile in the chosen context, and the end colour will be the highest value achieved by the profile in the chosen context.

Note: This means that if a disproportionately high value interferes with a data profile, the whole graph will be the lowest colour and only that high-coloured peak will be seen.

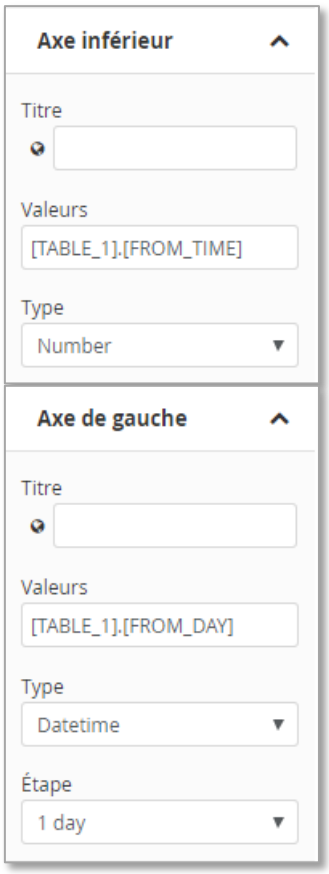
The X and Y axes can then be prepared.

Each axis can receive a (multilingual) title either from a hard-coded value in the field or from a column in the source woorksheet.

The inner axis (X) will receive the series of time-of-day values from a column of the source worksheet.

The left (Y) axis will receive the series of date values from the context, from another column of the source worksheet.

Finally, the "Data" section allows you to associate the column of the source worksheet containing the numerical data to be displayed with the "Values" field, which will then place the data on the graph.



H. HTML page

In a report or a dashboard, it is possible to inject HTML code that will be interpreted by the EMM (JOOL) display. The "HTML Page" widget allows you to prepare this code.

Its "Style" column contains two possibilities

- **Source**: Allows you to indicate a URL address, which will point to an external site. This site will then be opened in the widget window, inside the dashboard or the report. Note that EMM (JOOL) is a secure site (httpS), so sites opened through this type of widget must also be in https or they will not be displayed.

- **Content**: This window may contain HTML code, which will be interpreted by the browser when viewed in EMM (JOOL). All standard HTML5 tags are tolerated in this editor, in addition to EMM (JOOL) tags for including calculated data in the page.

In the case of HTML code built in EMM (JOOL), it is possible to retrieve information from the database and inject it into the widget. To do this, EMM (JOOL) code can be integrated directly using the {{ ... }} tag, which indicates to the HTML editor that it is EMM (JOOL) code to be calculated. Between these tags, we can use the usual EMM (JOOL) syntax, to retrieve references, properties, consumptions...

Example: the following code can be entered in an HTML report

```
The consumption of your site is {{ selection.data.sum }} kWh
```

This code will return the following text when interpreted by EMM (JOOL)

The consumption of your site is 1527 kWh

If there is little to insert, it is therefore possible to calculate the necessary values on the fly, calculating each value to be displayed at the time of display.

However, if the calculations are a little more complex and a large number of results need to be displayed, this method could take a long time to calculate.

Example: The following code...

```
January: {{ worksheet("WSHT_COMPUTE_VALUE"). column("A"). sum }} < br>

February: {{ worksheet("WSHT_COMPUTE_VALUE"). column("B"). sum }} < br>

March: {{ worksheet("WSHT_COMPUTE_VALUE"). column("C"). sum }} < br>
```

...will return the following text when interpreted by EMM (JOOL)

January: 100

February: 110

March: 105

In this example, we need 3 values, all calculated by a worksheet, but in practice, we call this worksheet 3 times in the code, so EMM (JOOL) will execute it 3 times. In a case where the worksheet is complex and takes a long time to compute, and we have a large number of calls to retrieve many values, this will explode the overall computation time of the HTML widget.

One way to deal with this problem is to use the associated spreadsheet as the source of the HTML widget.

a) Using the source worksheet in an HTML widget

In the context of HTML widgets, it is not possible to drag columns from the source worksheet as with other widgets, but it is possible to access its contents via the augmented HTML code (with the EMM (JOOL) {{ }} tags).

To do this, the @@itemtemplate@@ tag should be used

This new EMM (JOOL) tag is used to open (and close) a loop in the HTML code, whose code will be repeated sequentially for each of the objects in the selection. The keyword "item" can therefore be used inside these tags, and it will represent the object that caused the iteration at each iteration of the loop.

In our particular case, the widget's source worksheet is then considered to be a collection of rows, and 'item' will represent each row of the worksheet in turn.

Example: Let's imagine a source worksheet that would return the following table of data

SITE	METER	CONSO	RESOURCE
Site Nord	Electricité Nord	125	Electricité
Site Sud	Electricité Sud	132	Electricité
Siège central	Electricité Central	98	Electricité

If the HTML code in the widget is as follows:

Summary of consumption

@@itemtemplate@@

{{ item.column("SITE") }} : {{ item.column("METER") }} - {{ item.column("CONSO") }}< br>

@@itemtemplate@@

The result will be as follows:

Summary of consumption

Northern site: Electricité Nord - 125

South site: Electricité Sud - 132

Headquarters: Electricité Central - 98

So we looped over each row of the source worksheet, to recreate a new row in the HTML widget, and at each iteration, we retrieved the values of the row from the worksheet via the expression "item.column("REFERENCE")".

With this approach, it is of course possible to develop more complete HTML widgets, taking advantage of the HTML language and EMM (JOOL) syntax, by creating conditional HTML styles based on EMM (JOOL) values, or by using the HTML code of tables to display tables of data structured in HTML from the rows of the source worksheet.

Advanced example: Based on the same source worksheet as the previous example...

SITE	METER	CONSO	RESOURCE
Site Nord	Electricité Nord	125	Electricité
Site Sud	Electricité Sud	132	Electricité
Siège central	Electricité Central	98	Electricité

The following code could be created:

Consumption summary for the region {{ selection.name }}< br>

< table>

< tr>< th>Site</th>< th>Counter</th>< th>Consumer</th></tr>

```
@@itemtemplate@@
< tr>
        < td> {{ item.column("SITE") }}</td>
        < td> {{ item.column("METER") }}</td>
        < td> {{ item.column("CONSO") }}</td>
</tr>
@@itemtemplate@@
< /table>
```

The result, when interpreted by the HTML widget, will be as follows:

Summary of consumption for the Metropolitan France region

Site	Compteur	Conso
Site Nord	Electricité Nord	125
Site Sud	Electricité Sud	132
Siège central	Electricité Central	98

The rendering can obviously vary depending on the CSS styles applied to the HTML, but it is therefore possible to recreate a data table from a worksheet, in an HTML widget.

The "selection.name" code in the first line is out of the loop, so it can't be based on the "item" keyword, but on the "selection" keyword, and retrieve the name of the region that would be in active selection when the widget is launched.

Out of the loop again, we start the HTML table (<table><tr>...) and build its row of column headings (not to be repeated at each iteration, so well to leave out of the loop).

Finally, in the loop, we inject the code of a complete line, using the keyword "item" to work on each line, and retrieving the components of the source worksheet via the ". column" function.

This method, which uses the widget's source worksheet to retrieve its information, is much more efficient than recalling the same worksheet several times in the body of the code (as was done in the previous chapter). Indeed, one only calculates the source worksheet once, which is much more economical in terms of calculation time.

However, this method is limited to relatively simple cases in the sense that only one loop is possible (the @@itemtemplate@@ tag opens a loop and its repetition closes the loop... no nested loops possible). For more complex constructions, we will have to use an object of type "HTML Report" prepared in a dedicated editor, and then integrate it into the HTML widget.

b) Calling an external HTML report

The creation of an "HTML Report" object is a detailed subject, specifically developed in the EMM (JOOL) syntax manual. The fact is that in the HTML Report Editor, it is possible to combine HTML syntax and

EMM (JOOL) syntax to create HTML objects, which can then be included in other reports via the following syntax

Syntax :

htmlreport("RefRapportHTML"; selection; from; to; "LANG"; selection_substitution)

- "RefRapportHTML" will be the reference of the "HTML Report" object that we want to generate.
- selection (optional) is the selection you want to use for the generation of the HTML report
- from; to (optional) represent the time context for which to generate the HTML report
- "LANG" (optional) is the code of the language in which to generate the report (for example: "FR")
- selection_substitution (optional) is another way to pass the selection to the report

If neither selection nor context is provided when calling this function (these three parameters are optional), the generating selection and context will be the active selection and context.

If no language code is provided, the language of the logged-in user will be used to generate the report.

If a selection is specified as the second argument (necessarily composed of a list of entities/counters) the keyword "selection" present in the selection box of the sub-report will be replaced by the list of entities/counters.

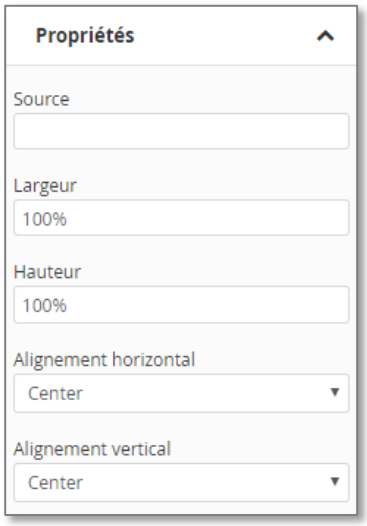
If a substitution selection is indicated in the sixth argument (free type, you can have an array, a grouping, etc.), this substitution selection will replace the entire selection box of the sub-report, thus replacing any formula that may have been indicated there.

(more details and examples in the syntax manual detailing all the subtleties of the JQL language)

This syntax can be used in the body of HTML reports to include sub-reports, but also directly in HTML widgets to retrieve an HTML report and inject it into the widget frame, which can then be used as a component of a dashboard.

I. Image

This type of widget is quite basic. It simply allows you to retrieve an image from the database (or online) and display it. This can be useful for customising a dashboard or a report with the client's logo for example.



The properties of this widget ("Style" column) allow you to define the source of the image, its dimensions and its alignment.

Note that the source and dimensions can be hard-coded (as for a client logo, for example), but they can also come from the source spreadsheet.

In this case, the spreadsheet will need to return an image file name in one of these columns, which will be dragged to the "Source" field in the "Style" column.

The image management interface is still under development. For the time being, if you need to use an image in a widget, please contact your EMM (JOOL) administrator so that they can upload the image and provide you with the url link to use. (xxx)

J. Text box

This type of widget allows you to insert a text box in a report or a dashboard. The interface of the widget is extremely simple as the "Style" column is simply an "Open Editor" button allowing you to enter the text you want.

This text editor contains a whole series of styles, such as bolding, italicising, colouring, etc.

In addition to this, this editor also allows you to insert EMM (JOOL) code to make its content dynamic. The EMM (JOOL) code should be enclosed in double braces

Normal text, **{{ EMM (JOOL) code returning text }}**, continuation of normal text.

This allows you to inject information into the text, such as the reference or the name of the selected entity... (Useful, for example, for making report titles, with the name of the site at the top of the page, in a format of your choice (colour, size, etc.))

9. Dashboards and reports

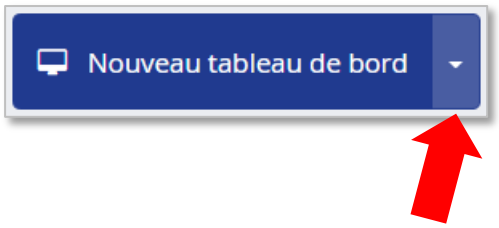
Based on the existing widgets, it is possible to create composites to make dashboards.
The dashboard editor is available via its icon in the action bar.

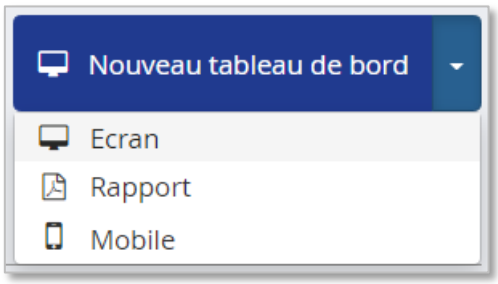
There are 3 types of dashboards:

- **Screen**: a display intended for one screen and which cannot be scrolled down. If there are too many things to display on a screen, the creator of the board can however create additional tabs to slide in his widgets.
- **Report**: display for printing as a PDF.

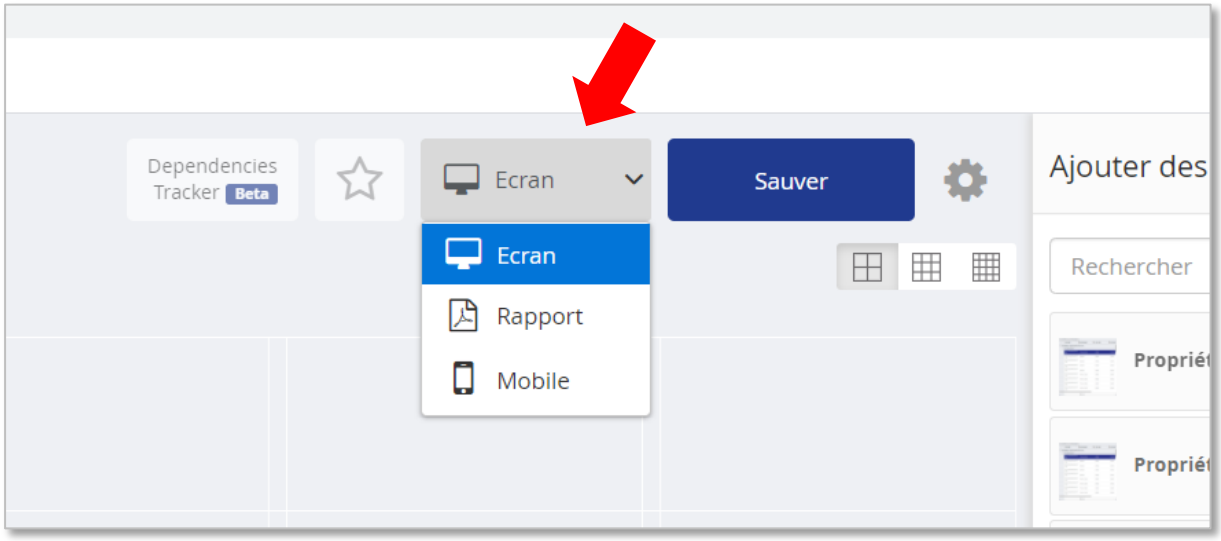
- **Mobile**: Specific display for the mobile application.

The type of dashboard to be created can be chosen directly when creating a new dashboard on the homepage of the editor.



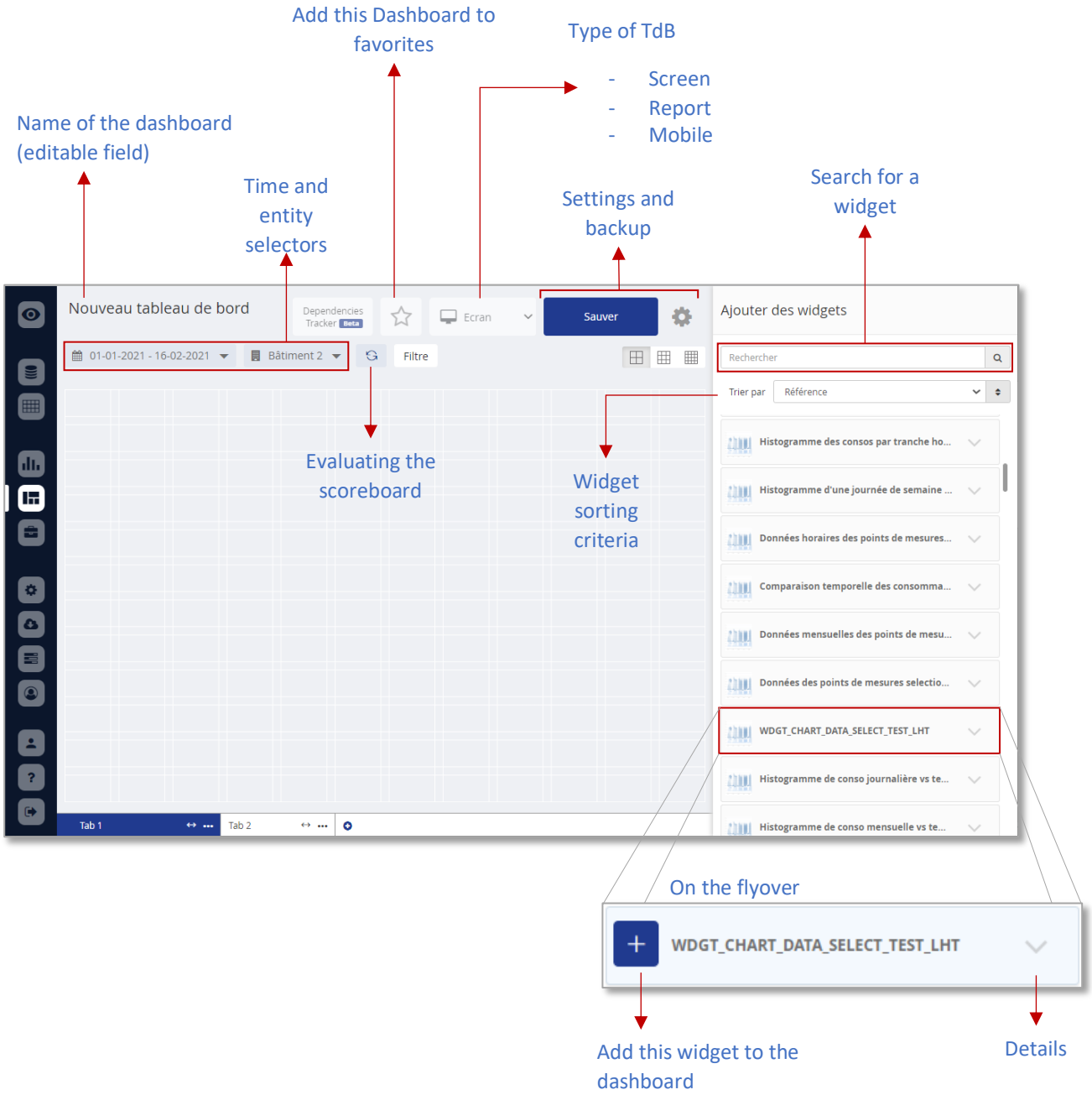


It can also be modified afterwards directly in the editor via this drop-down button at the top right of the screen.

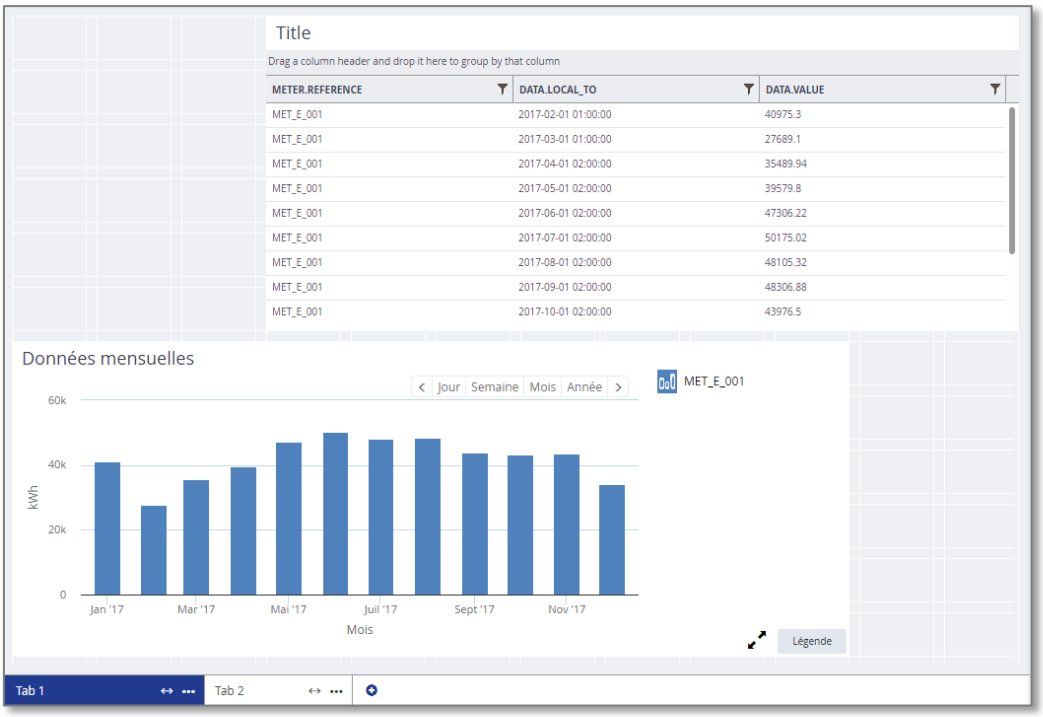


General

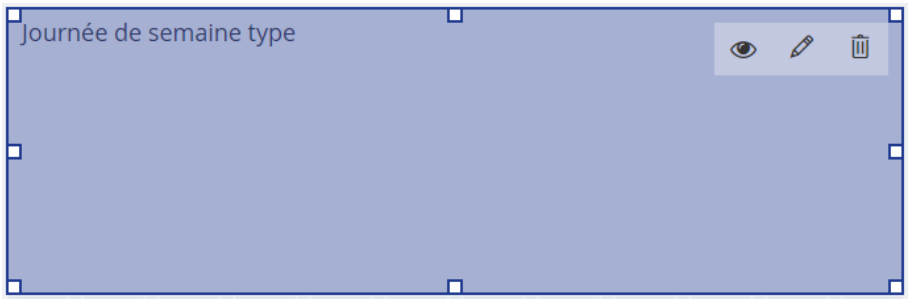
The interface for creating dashboards is as follows (shown here in screen mode):



The right-hand side of the screen displays the list of widgets available for the dashboard. When you have found the one you are looking for, you can add it to the panel by clicking on the "+" button that appears on the hover.



Clicking on a widget selects it (it is coloured blue), and it can then be set up as desired for display in the dashboard.



By clicking on it, it can be moved to the positioning grid of the dashboard.

The small white squares on the edges of the widget allow it to be resized.

The buttons at the top right allow you to

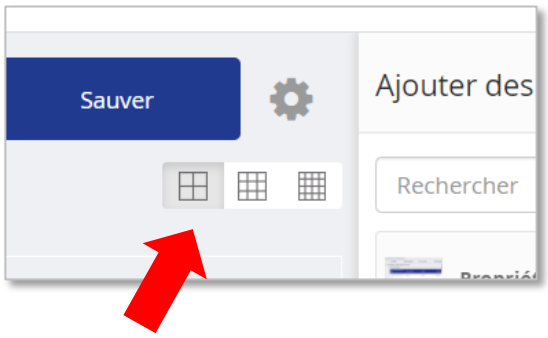
- **find** the widget in the list of widgets (right column)
- **edit** the chosen widget (you then leave the ToB editor to open the widget editor)
- **remove** the widget from the dashboard

As with Datasets, Worksheets and Widgets, the entity and context selectors at the top left of the page allow you to simulate the result of the dashboard based on a selection and context, but they will not be saved in the dashboard definition.

Screen mode - specifics

The screen mode is therefore intended to be displayed on a screen, whether it is a manager's computer or a screen in a public space. It cannot be scrolled, but it can contain HTML widgets which can be scrolled if necessary.

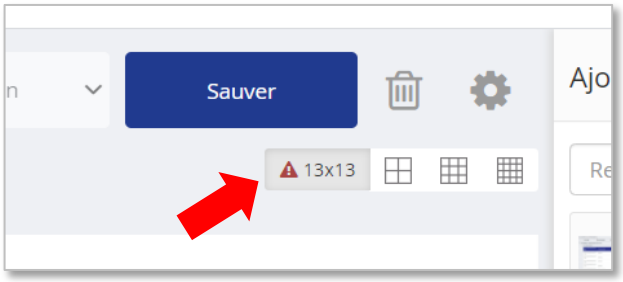
In screen mode, the format is limited to a grid of 12x12 cells, or 24x24 or 36x36. This format can be changed using the following buttons, again at the top right of the screen.



Each cell of the grid has the same ratio as the screen (logical since we have the same number of rows and columns), and it is not possible to create tables with numbers of columns outside the preset formats.

Note: some old dashboards, created before this grid format restriction, have not been modified and are still supported in their current formats ("legacy" mode), but it will no longer be possible to modify this format separately to migrate it to one of the 3 standard formats.

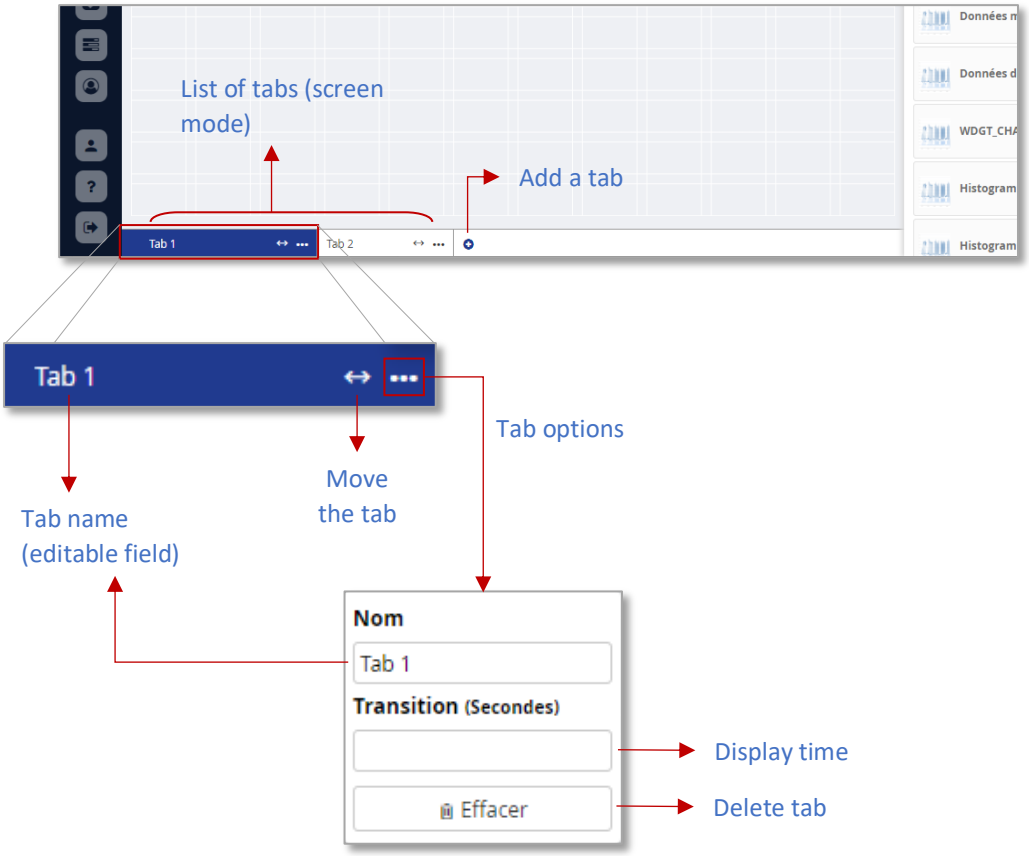
In this case, an additional box will appear at the top right, indicating the current format (example below with a 13x13), indicating that this dashboard is in a non-standard format.



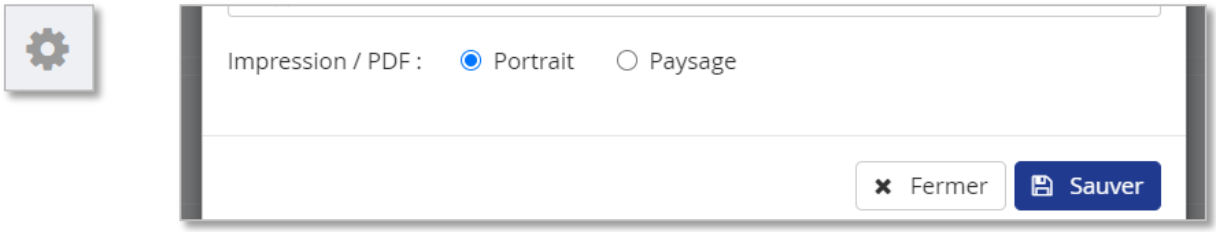
It will then be possible to choose between the current format or the 3 standard formats, but not to create another non-standard format (note: if you switch to a standard format, you will not be able to return to the initial non-standard format)

At the bottom left of the editor in screen mode, there is a list of tabs and buttons to create new ones or move them.

In the options of a tab, you can define its name as well as the display time of this tab, in case the dashboard is displayed via a kiosk link.



When such a dashboard is exported as a PDF from EMM (JOOL)'s viewer mode, it will be printed in landscape orientation to best fit the screen format. It is however possible to force a PDF printout to landscape orientation by going to the properties menu and changing the orientation at the bottom of the pop-up window.

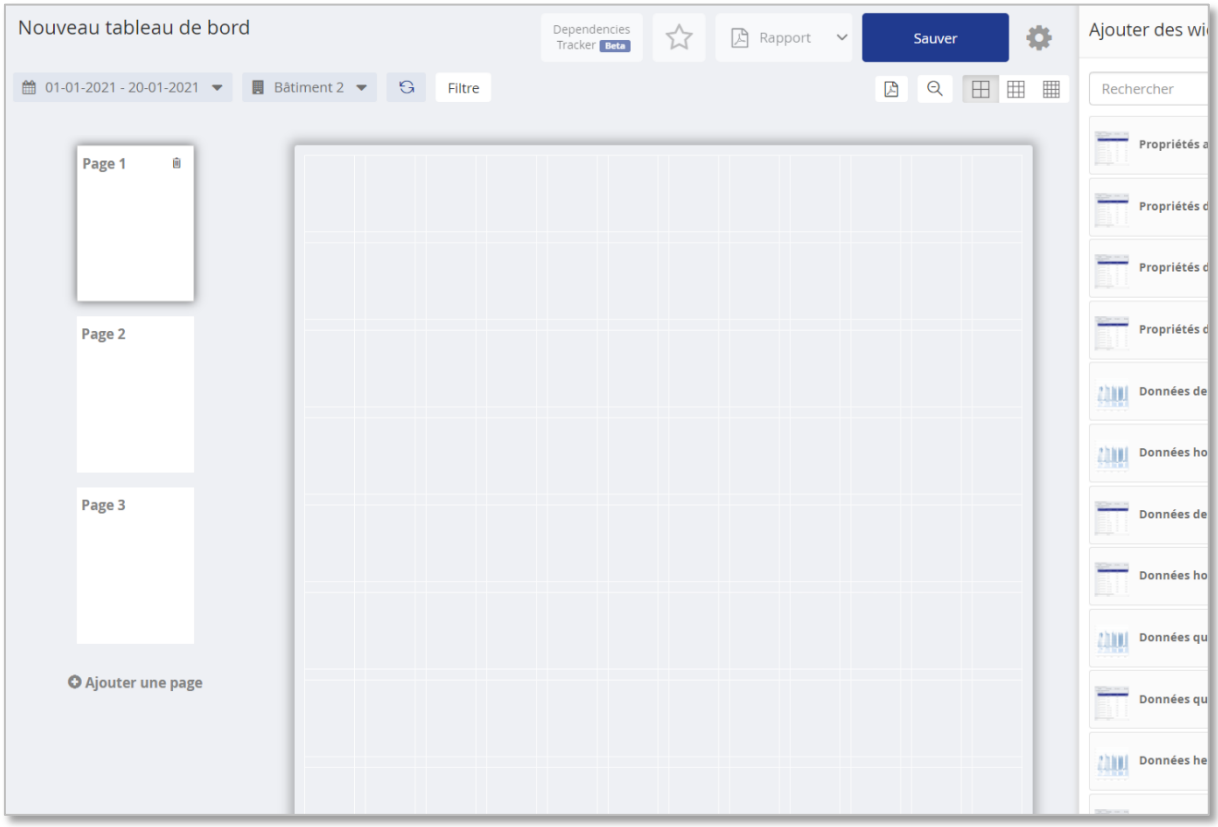


In this pop-up window, you can define the transition and refresh times in case this dashboard is exported via a kiosk link, for example for public display.

Report mode - specifics

This mode is intended for A4 display to prepare reports for printing as PDFs. This does not prevent them from being displayed in the EMM (JOOL) menus for on-screen viewing, but they will retain their A4 ratio even when displayed on screen.

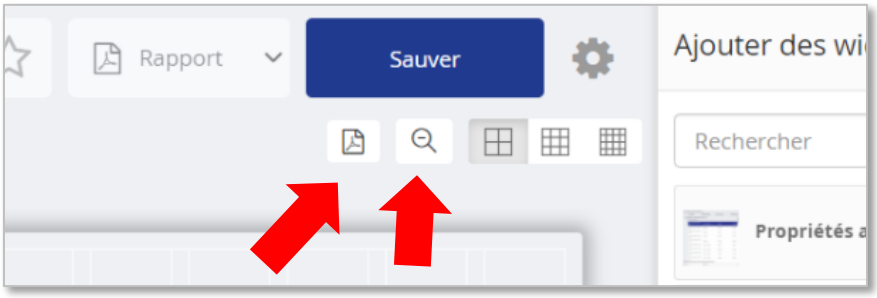
The editor of this type of dashboard has a list of thumbnails on the left of the screen, allowing you to manipulate the pages and to compose them page by page, as you would do tab by tab in screen mode. It is possible to add or delete pages, but also to move them (via drag&drop) to reorganise a report for example.



In the same way as for the screen mode, each page in report mode can be broken down into grids of 12x12, 24x24 or 36x36 cells, and the same principle of "historical" format applies.

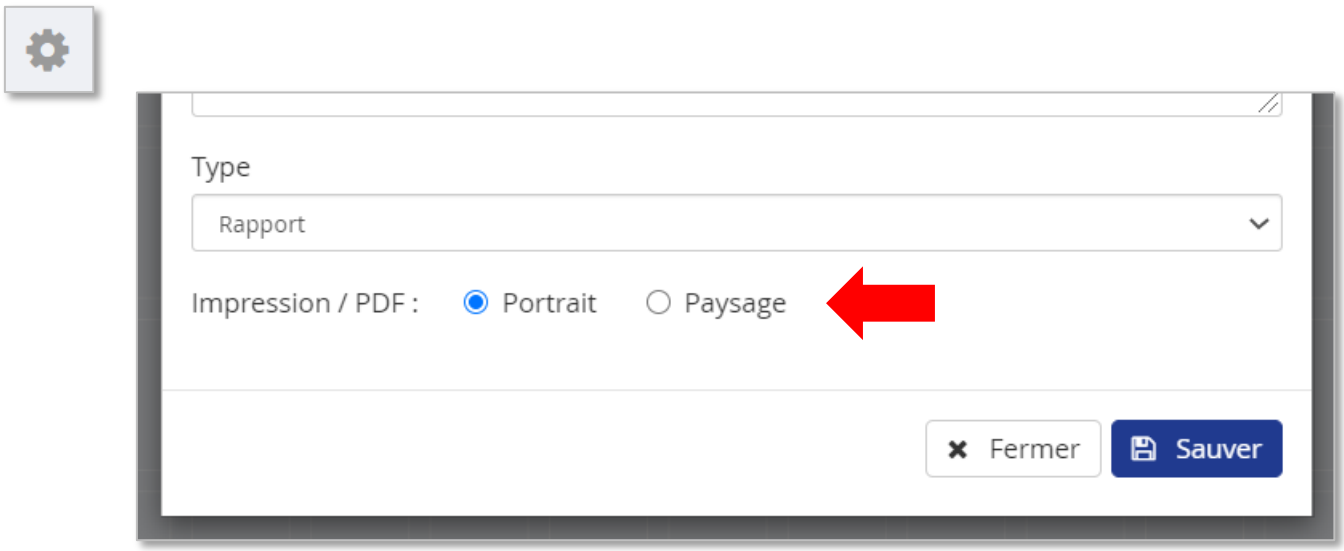
In the upper right-hand corner of the screen, in addition to the format buttons, there is also a magnifying glass button which allows you to zoom in on the page to display it at screen width or to return it to full page (this has no impact on the composition of the dashboard, we are just talking about display in the editor)

There is also a button to change the orientation of the report to landscape or portrait.



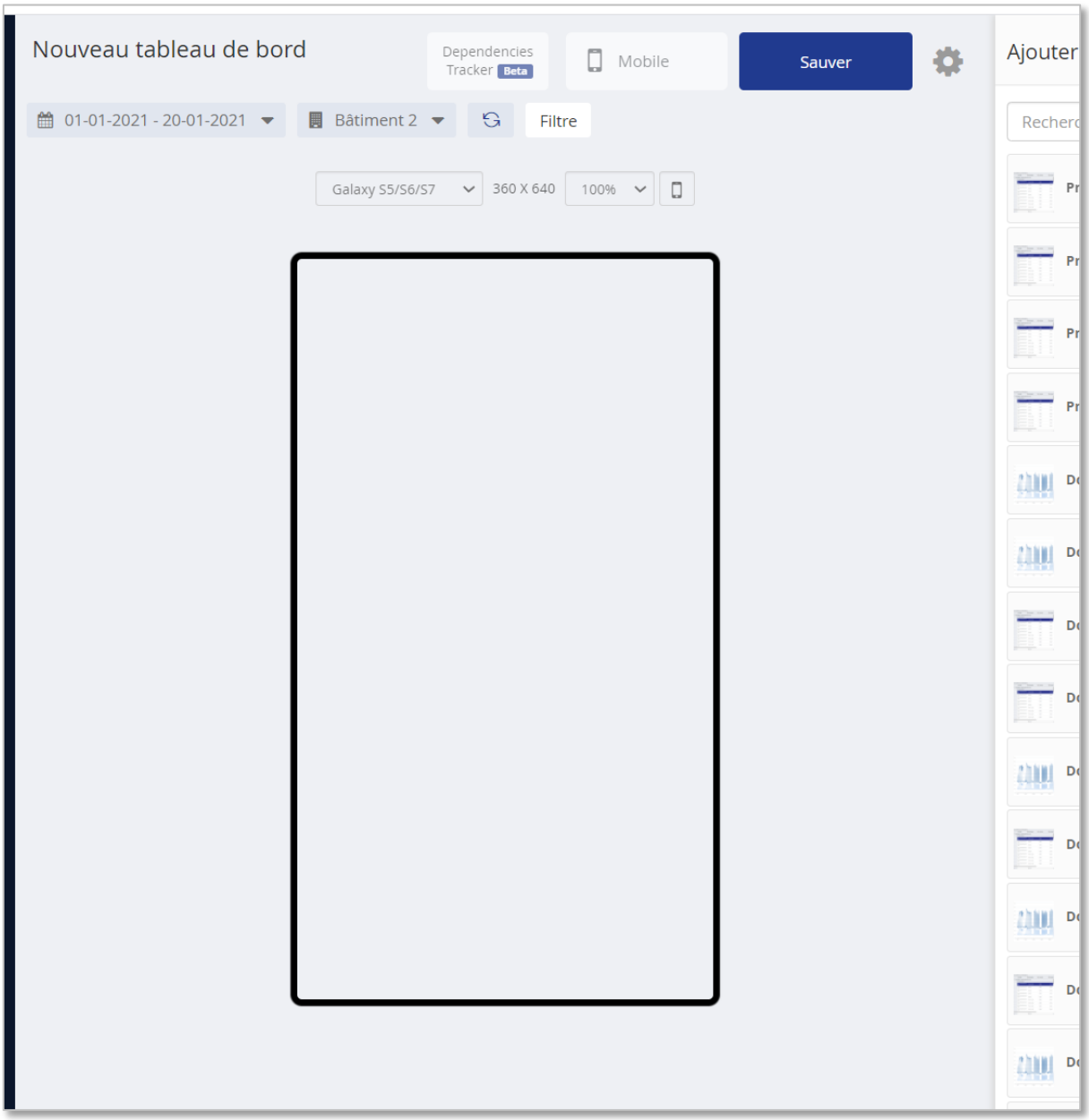
Note: This orientation applies to the whole report at once, not page by page.

As with the screen mode, this print orientation can also be set in the dashboard properties pop-up.

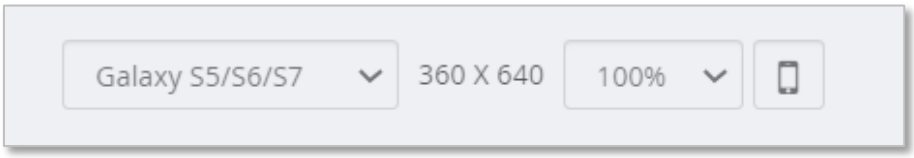


Mobile mode - specifics

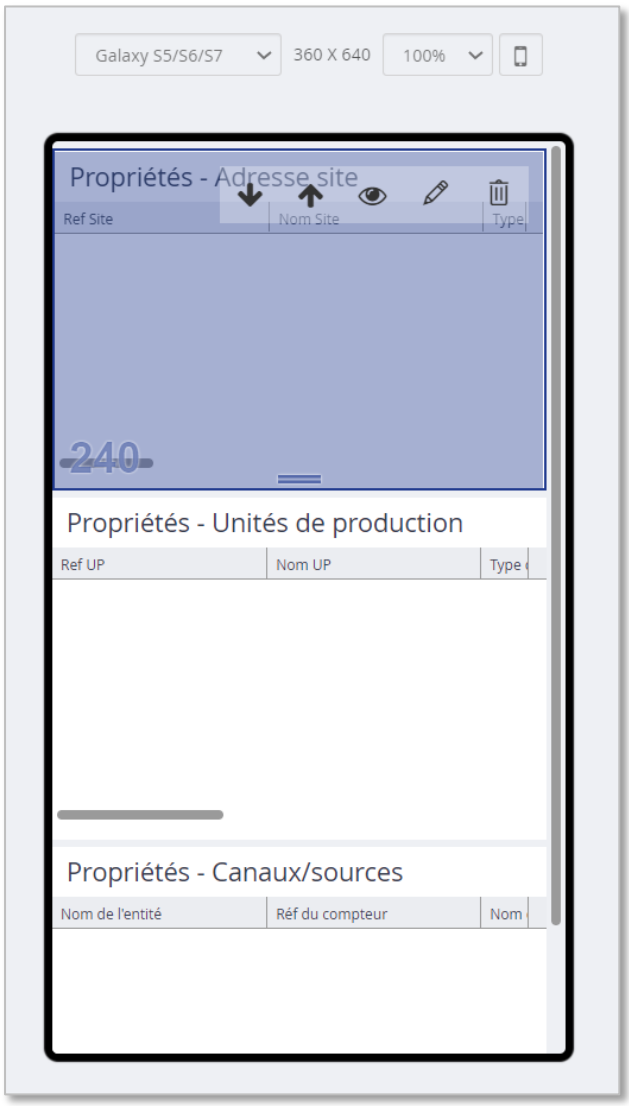
Decoupled from "traditional" dashboards, mobile dashboards are a type of dashboard in their own right.



At the top of the editor, a series of options allow you to preview the result on different types of laptops, different resolutions, and even to see how it looks on a horizontal laptop.



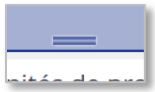
Once the widgets are added to the dashboard, they are simply displayed one below the other, without any further layout.



When a widget is selected, the arrow buttons that appear at the top right of the widget allow the widget to be moved up or down one place in the list of displayed widgets.

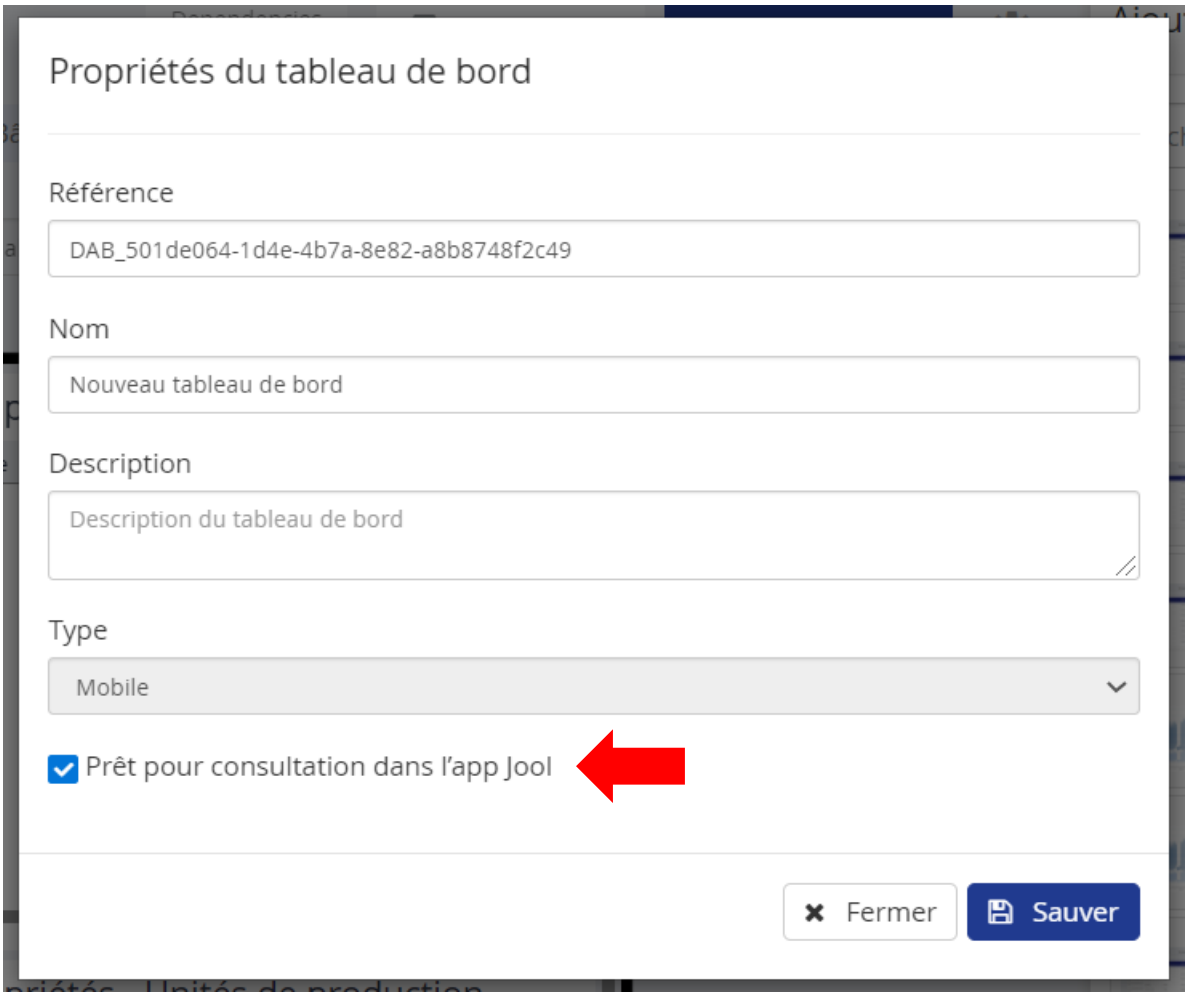
The number displayed at the bottom left (240 in the example opposite), indicates the number of pixels in height assigned to this widget.

This height can be changed in steps of 20 pixels by moving vertically the two-line slider that appeared at the bottom of the selected widget



Note: when switching from another mode to a mobile mode, it is impossible to return to another mode. Indeed, the layout of the other modes is lost if you switch to mobile, and the return is not planned for the moment.

In the dashboard properties pop-up, a checkbox has been added for the mobile type. This checkbox indicates whether the dashboard is ready to be displayed on the mobile application or whether it is under development. This can avoid incomplete dashboards.


Propriétés du tableau de bord
Référence
DAB_501de064-1d4e-4b7a-8e82-a8b8748f2c49
Nom
Nouveau tableau de bord
Description
Description du tableau de bord
Type
Mobile
Prêt pour consultation dans l'app Jool
Fermer
Sauver