MicroSERVER UI

Description

1. Lecteurs ciblés

Cette documentation est à destination des utilisateurs de l'application MicroSERVER, permettant la visualisation de données collectées depuis des automates SIREA.

2. Présentation de l'application

2.1 Widgets du tableau de bord

Le tableau de bord contient toutes les données que vous jugez utiles à afficher en premier lieu. Ces données sont contenues dans des cadres appelés « widgets ». Il existe ainsi plusieurs widgets différents afin de visualiser au mieux les données sous différentes formes (tableau, graphe, diagramme…). Le tableau de bord est facilement personnalisable selon les besoins. On peut y ajouter, supprimer, ou déplacer facilement un widget.

2.2 Modes du tableau de bord

Le tableau de bord possède deux modes:

- le mode visualisation : c'est le mode par défaut. Il permet de visualiser l'ensemble des widgets avec leurs résultats.
- le mode édition : il permet de créer, modifier et bouger les widgets.

Pour accéder au mode visualisation quand on est dans le mode édition, il faut cliquer sur le bouton:

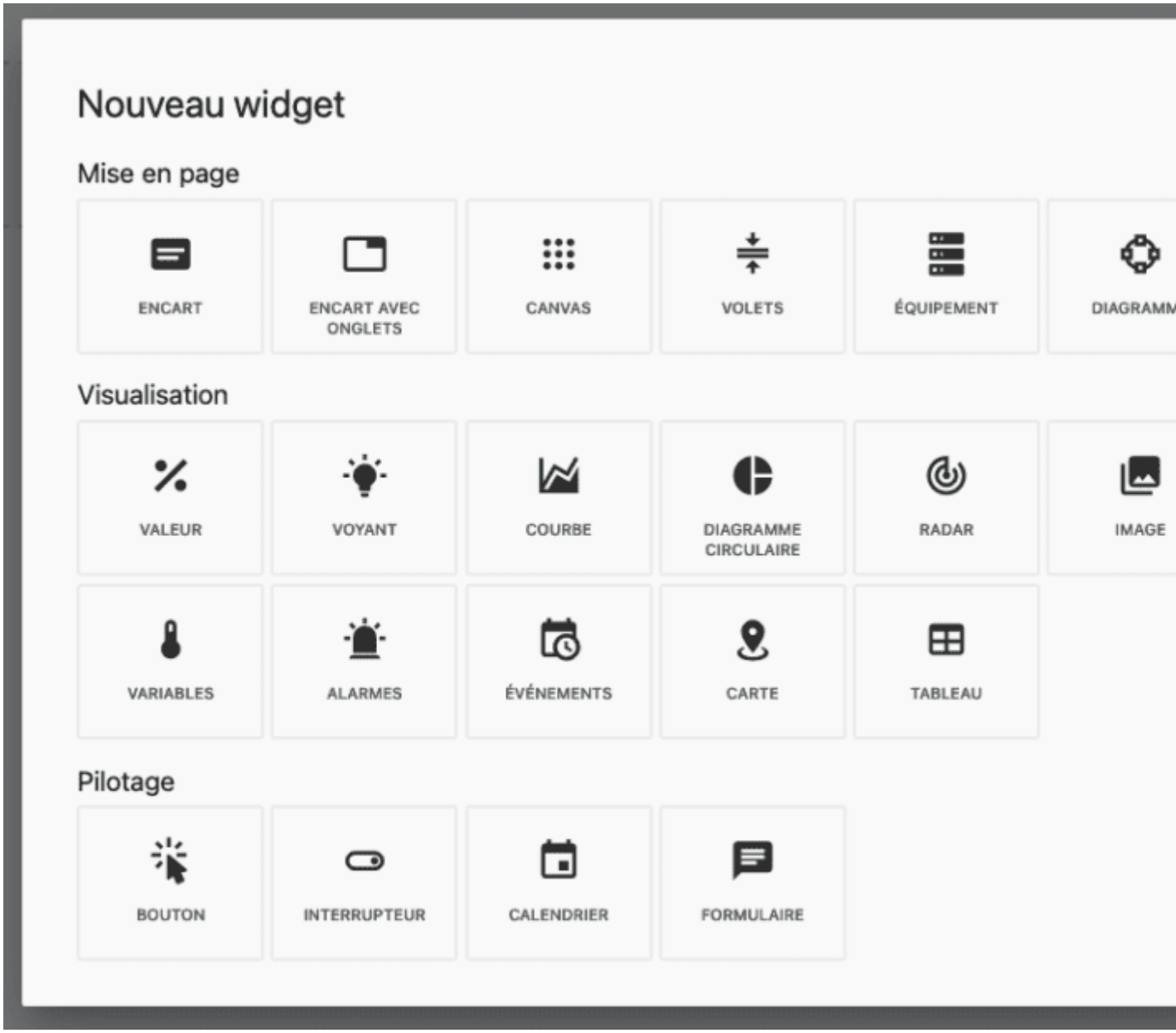
Pour accéder accède au mode édition à partir du mode visualisation, il faut cliquer sur le bouton:

Ces deux boutons se situent en bas à droite de l'écran.

2.3 Attribut d'un widget

2.3.1 Créer un widget

Pour ajouter un widget, il est nécessaire d'être dans le mode édition.
Cliquez ensuite sur le champ « + Ajouter un widget » en bas de la page.
Cette action ouvre une fenêtre *popup* permettant de créer un nouveau widget. Trois types de widgets sont proposés:



widgets microserver

- Mise en page
- Visualisation
- Pilotage

Le choix d'un widget ouvrira une nouvelle fenêtre *popup* sur la droite qui vous permettra de renseigner un certains nombres de champs. Les indications pour compléter cette fenêtre sont fournies dans **Widget Mise en page**, **Widget Données** et **Widgets Dédiés**. Cliquer sur "Enregistrer" pour confirmer la création. On passe alors dans le mode visualisation.

2.3.2 Déplacer un widget

Quand un widget est créé, on peut le déplacer sur le tableau de bord en cliquant sur cet icône. Elle se trouve en haut à droite du widget et est disponible seulement dans le mode édition.

2.3.3 Modifier le widget

Quand un widget est créé, on peut le modifier en cliquant sur cet icône. Elle se trouve en haut à droite du widget et est disponible seulement dans le mode édition.

2.3.4 Dupliquer le widget

Quand un widget est créé, on peut le dupliquer en cliquant sur cet icône. Elle se trouve en haut à droite du widget et est disponible seulement dans le mode édition. La copie est créé en dessous du widget actuel.

2.3.5 Supprimer le widget

Quand un widget est créé, on peut le supprimer en cliquant sur l'icône:
Cette icône se trouve en haut à droite du widget et est disponible seulement dans le mode édition.
Cette action est irréversible.

2.4 Catégorie de widgets : Mise en page

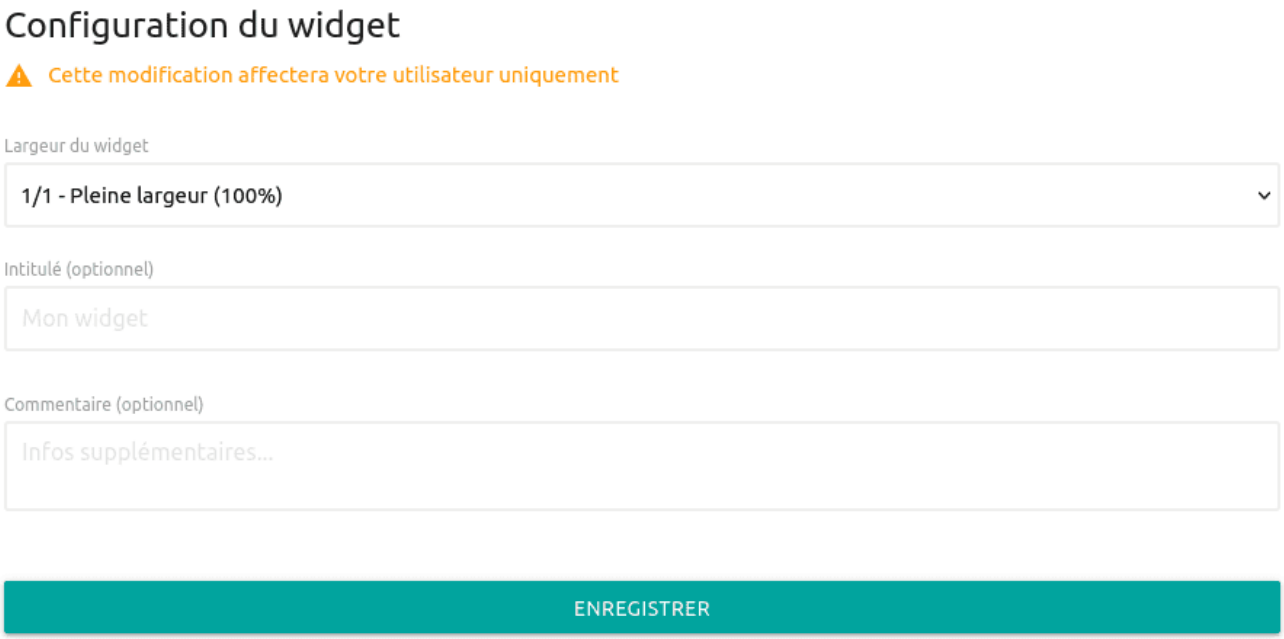
2.4.1 Widget Encart

Ce widget permet de créer un cadre (blanc) dans lequel on va pouvoir créer d'autres widgets. Le but est que l'encart serve de cadre dans lequel on va venir y insérer des widgets comme des photos. Dans l'encart, les widgets pourront changer de place.

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

On peut ajouter un Titre à l'encart et un commentaire en remplissant les deux formulaires optionnels.



Capture d'écran de la fenêtre *popup* de création/modification du widget Encart

2.4.2 Widget Encart avec onglets

Création et modification

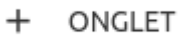
Ce widget s'inspire du widget encart. Il permet de créer un cadre (blanc) dans lequel on va pouvoir créer d'autres widgets à l'intérieur. Ces widgets seront rangés dans des onglets. Ainsi un encart pourra avoir 0, 1 ou plusieurs onglets.

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

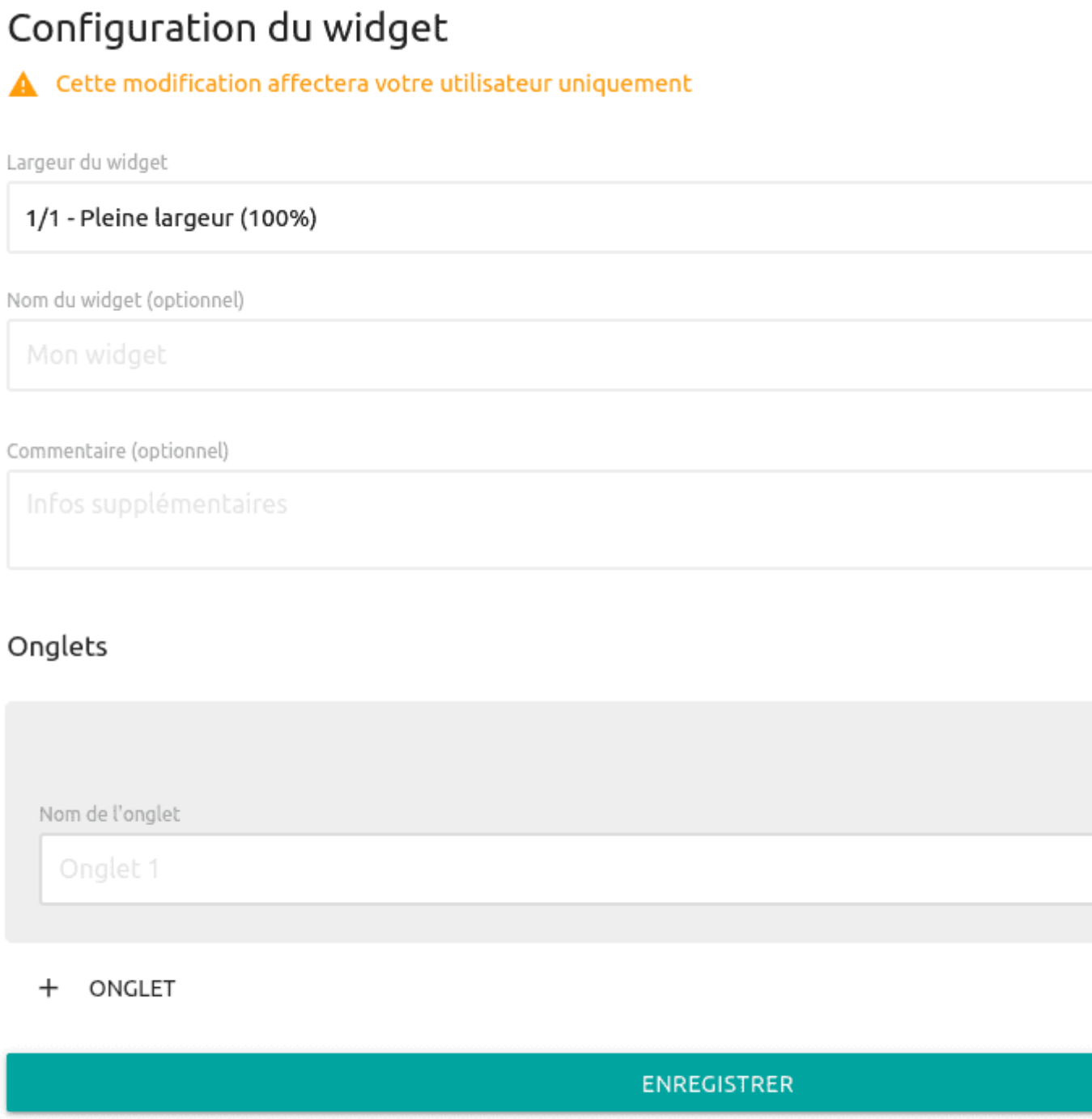
On peut ajouter un Titre à l'encart et un commentaire en remplissant les deux formulaires optionnels.

Pour créer un onglet, il suffit de renseigner le « Nom de l’onglet » voulu dans la partie « Onglets » dans le cadre gris.



Si on veut rajouter un nouvel onglet, cliquez sur
+ ONGLET.
Un nouveau cadre gris apparaît. Il est à remplir comme le précédent.

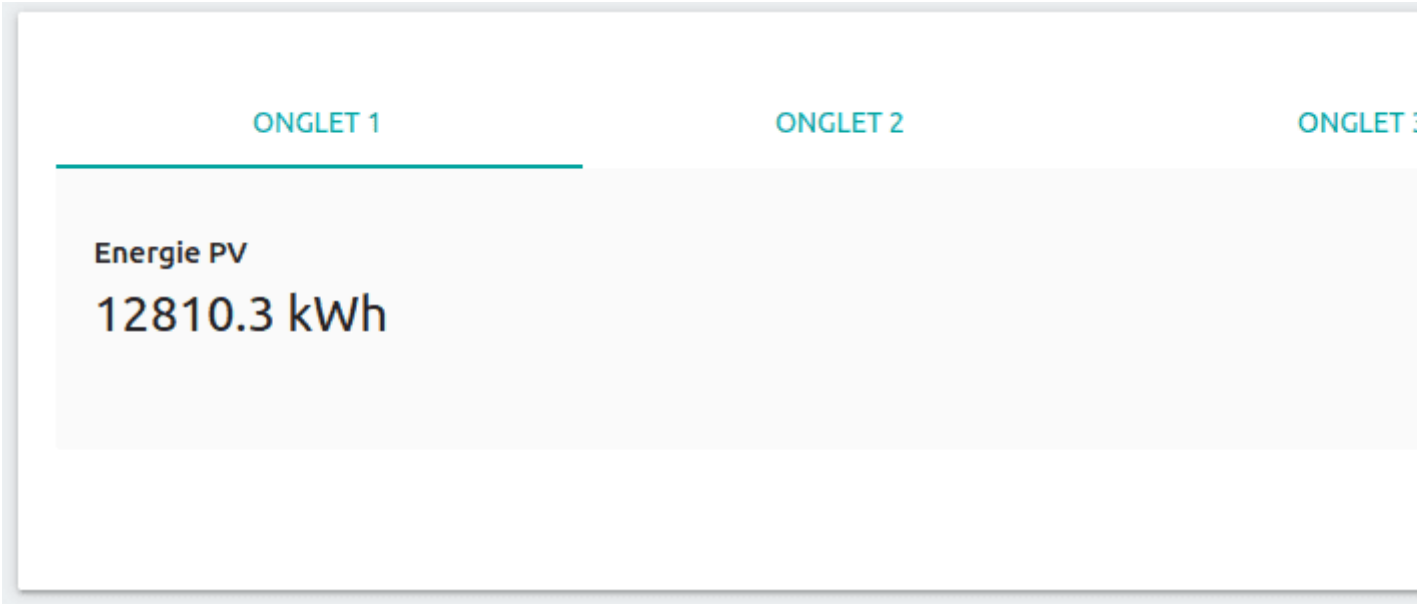
Si on veut supprimer un onglet, cliquez sur la croix en haut du cadre gris de l’onglet concerné. On ne peut supprimer un onglet que si il y en a plus de deux.



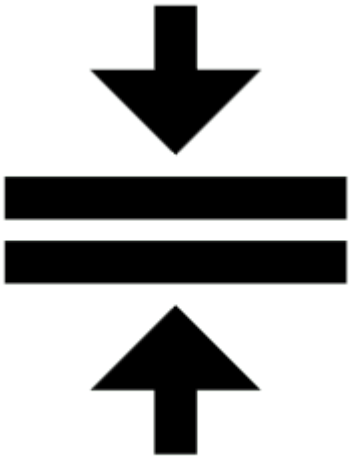
Capture d'écran de la fenêtre *popup* de création/modification du widget Encart avec Onglets

Navigation sur le widget

Dans le mode visualisation, on peut changer d'onglets en cliquant sur l'onglet désiré.



Capture d’écran d’un exemple de Widget Encart avec Onglets ayant 3 onglets (Onglet 1, Onglet 2 et Onglet 3)



2.4.3 Widget Volets

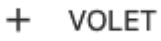
Création et modification

Ce widget s’inspire du widget encart avec onglet. Il permet de créer des espaces refermables qui vont permettre d’insérer d’autres widgets. Ces widgets seront rangés dans des volets. Ainsi un encart pourra avoir 0, 1 ou plusieurs volets.

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

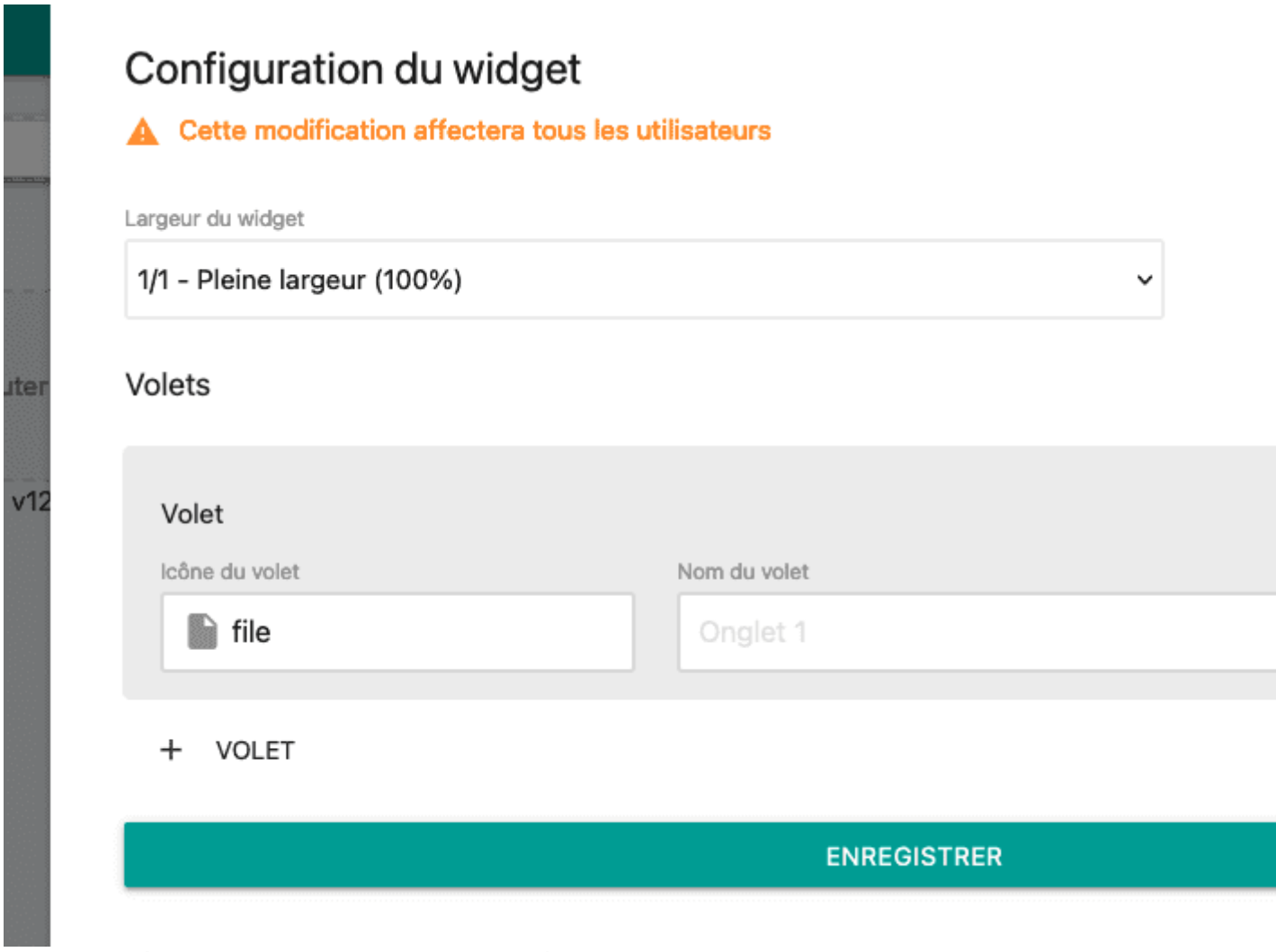
- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)
-

Pour créer un onglet, il suffit de renseigner le « Nom de l'onglet » voulu dans la partie « Onglets » dans le cadre gris.



Si on veut rajouter un nouveau volet, cliquez sur
+ VOLET. Un nouveau cadre gris apparaît. Il est à remplir comme le précédent.

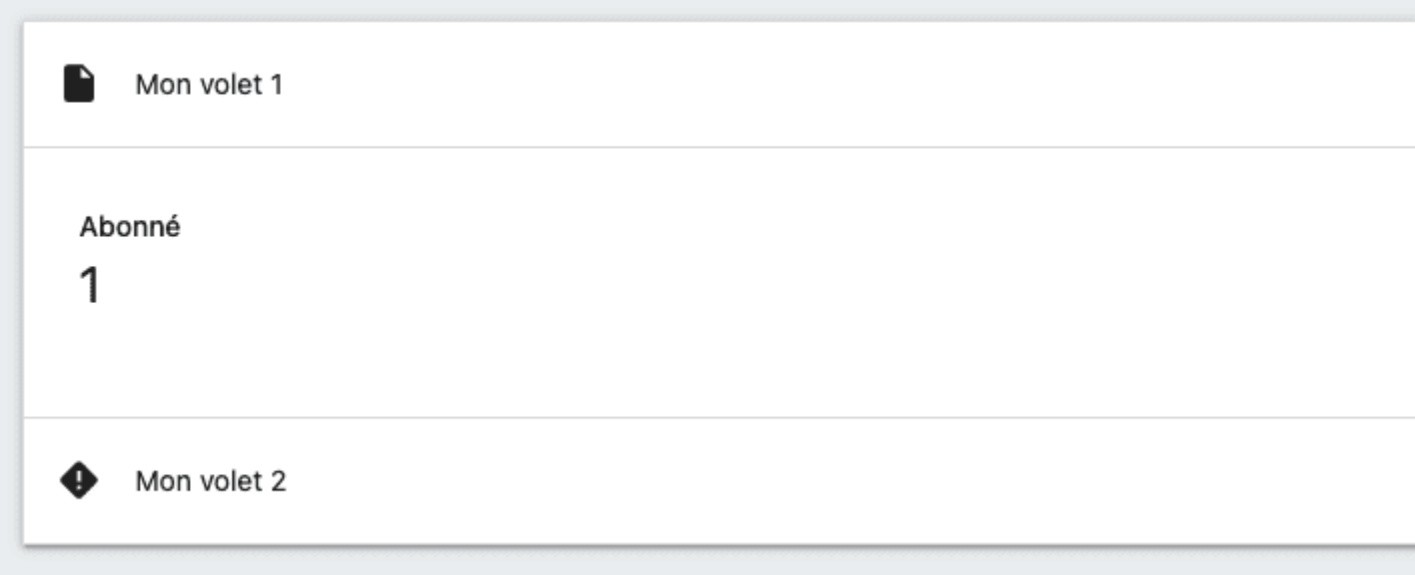
Si on veut supprimer un volet, cliquez sur la croix en haut du cadre gris de l'onglet concerné. On ne peut supprimer un volet que si il y en a plus de deux.



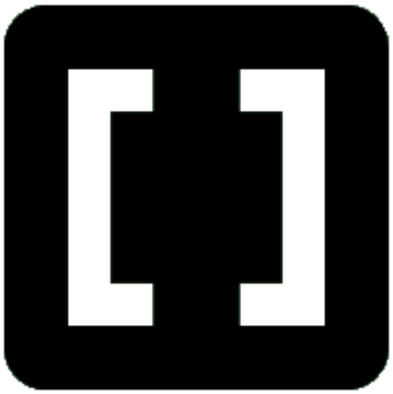
Capture d'écran de la fenêtre *popup* de création/modification du widget Volets

Navigation sur le widget

Dans le mode visualisation, on peut changer de volet en cliquant sur le volet désiré.



Capture d’écran de la fenêtre *popup* de création/modification du widget Volets



2.4.4 Widget Code

Ce widget permet de coder et de créer des fonctionnalités qui ne sont pas disponibles avec les autres widgets.

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

Le widget sera composé des trois encart suivants:

- HTML
- CSS
- Javascript

Un exemple de fonctionnalité à créer est l'affichage dynamique de la valeur d'une variable (nommé « X » de l'équipement d'index 1) et la possibilité de modifier cette valeur. Ci-dessous, le code associé à cette fonctionnalité:

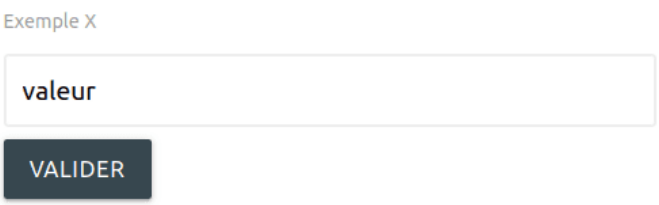
- HTML

```
<form id="form1" data-table="variables" data-mnemonic="X@1">
    <label class="form-label">Exemple X</label>
    <input type="text" data-field="rawValue" />
    <input type="button" value="Valider" class="btn btn-primary" onclick    ="
</form>
```

- JS

```
var submitForms = () => {
        $('#form1').submit();
    }
```

Le visuel final de cette fonctionnalité sera le suivant:



Capture d'écran du visuel pour l'exemple du widget code

2.5 Catégorie de widgets : Visualisation

2.5.1 Widget Valeur

Le widget valeur permet de visualiser la valeur d'une variable à un moment donné.

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

Pour choisir la variable souhaitée, il faut dans un premier temps sélectionner dans le menu déroulant, l'équipement et le nom de la variable.
Il y a aussi la possibilité de renommer la variable par un autre nom et de rajouter un commentaire.

Le champ « Type de valeur » permet de choisir quel type de valeur on souhaite avoir parmi trois options:

- si on veut une valeur ponctuelle (instantanée), cliquer sur « ponctuelle »
- si on veut une valeur moyennée, cliquer sur « moyenne ». De nouvelles informations de temporalité sont alors demandées. On a ainsi le choix de calculer la moyenne sur:

– une période statique. Un intervalle de temps avec une date de début et de fin est alors à renseigner
– une période calendaire. En fonction de la période choisie, si on clique sur précédente, l'intervalle s'étendra alors de l'heure, du jour, de la semaine, du mois ou de l'année précédente jusqu'à la date actuelle. En fonction de la période choisie, si on clique sur actuelle, alors la moyenne sera calculée sur l'heure, la journée, la semaine, le mois ou l'année actuelle. Par défaut la période est celle de la journée actuelle.
– une période glissante. En fonction de la durée N rentrée et de la période, l'intervalle de temps s'étendra sur les N dernière(s) heure(s), les N dernier(s) jour(s), les N dernière(s) semaine(s), les N dernier(s) mois ou les N dernière(s) année(s)

- si on veut un écart entre deux valeurs, cliquer sur « écart ». De nouvelles informations de temporalité sont alors demandées. On a ainsi le choix de calculer l'écart entre deux valeurs sur:

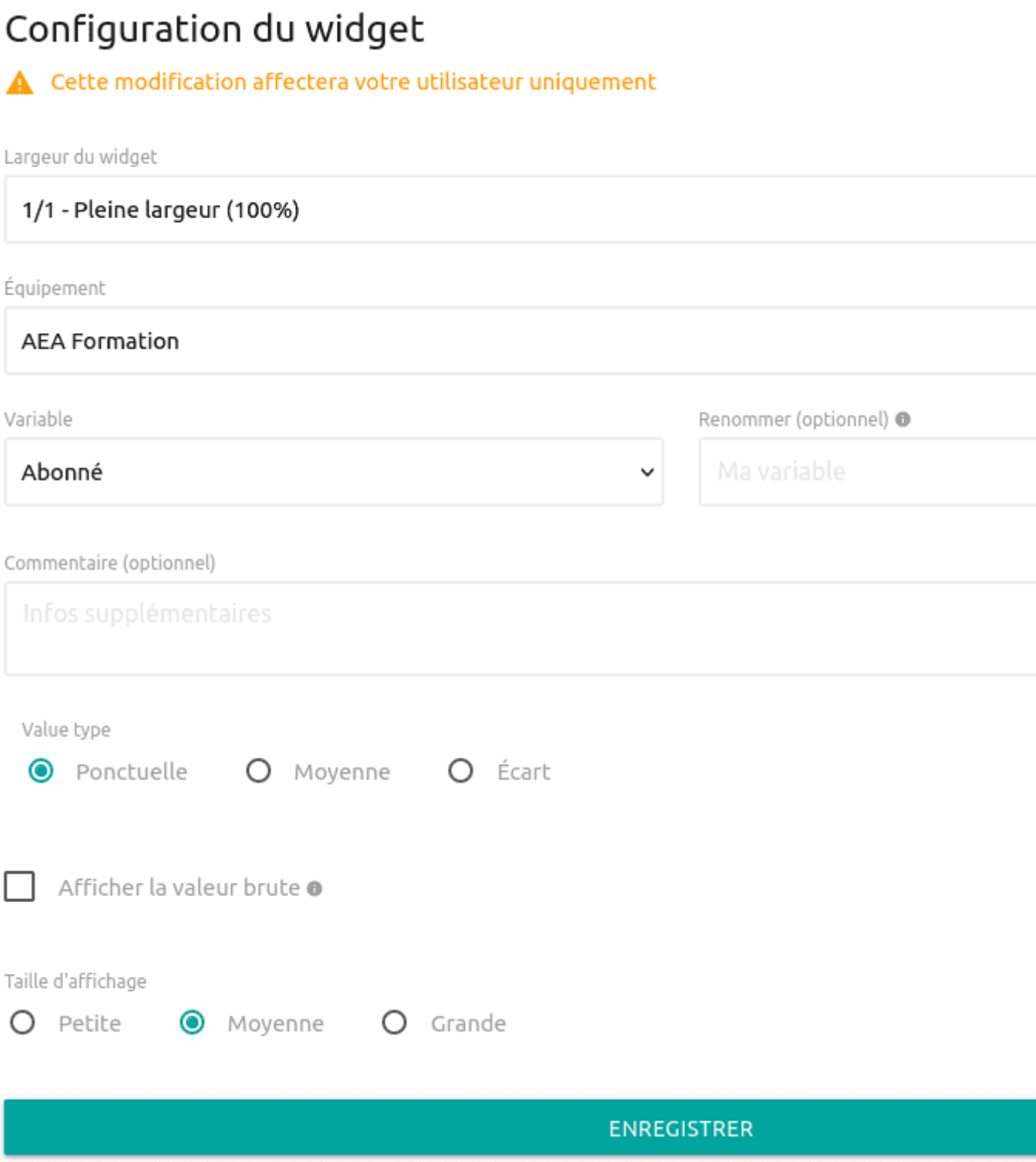
– une période statique. L'écart sera calculé entre la date de fin et la date de début renseignées.
– une période calendaire. En fonction de la période choisie, si on clique sur actuelle/précédente, alors l'écart sera calculé entre l'heure, la journée, la semaine, le mois ou l'année actuelle/précédente et la date actuelle. Par défaut, la date sera celle de la journée actuelle.
– une période glissante. L'écart sera alors calculé entre la date actuelle et la N dernière heure, le N dernier jour, la N dernière semaine, le N dernier mois ou la N dernière année

Par défaut, le graphe affichera une valeur ponctuelle.

En fonction des besoins, il est possible d'afficher la valeur brute en cochant « Afficher la valeur brute ». Néanmoins, l'unité ne sera pas affichée.

La taille de l'affichage est réglable puisque la taille de la police diminue si vous cliquez sur la taille petite. Elle augmente si vous cliquez sur « Moyenne » et elle est de taille maximale si vous cliquez sur

« Grande ». Par défaut la taille est cochée à « Moyenne ».



Capture d'écran de la fenêtre *popup* de création/modification du widget Valeur

2.5.2 Widget Voyant

Le widget voyant permet de visualiser si la valeur d'une variable respecte une/des condition(s) grâce à un voyant de couleur.

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

Pour choisir la variable souhaitée, il faut dans un premier temps sélectionner dans le menu déroulant, l'équipement et le nom de la variable. Il y a aussi la possibilité de renommer la variable par un autre nom et de rajouter un commentaire.

Le champ « Type de valeur » permet de choisir sur quel type de valeur la condition va être exécutée parmi trois options:

- si on désire que la condition soit sur une valeur ponctuelle (instantanée), cliquer sur « Ponctuelle »
- si on veut que la condition soit sur une valeur moyennée, cliquer sur « Moyenne ». De nouvelles informations de temporalité sont alors demandées. On a ainsi le choix de calculer la moyenne sur:

– une période statique. Un intervalle de temps avec une date de début et de fin est alors à renseigner
– une période calendaire. En fonction de la période choisie, si on clique sur précédente, l'intervalle s'étendra alors de l'heure, du jour, de la semaine, du mois ou de l'année précédente jusqu'à la date actuelle. En fonction de la période choisie, si on clique sur actuelle, alors la moyenne sera calculée sur l'heure, la journée, la semaine, le mois ou l'année actuelle. Par défaut la période est celle de la journée actuelle.
– une période glissante. En fonction de la durée N rentrée et de la période, la moyenne sera calculée sur les N dernière(s) heure(s), les N dernier(s) jour(s), les N dernière(s) semaine(s), les N dernier(s) mois ou les N dernière(s) année(s)

- si on désire que la condition soit sur un écart entre deux valeurs, cliquer sur « Écart ». De nouvelles informations de Temporalité sont alors demandées. On a ainsi le choix de calculer l'écart entre deux valeurs sur:

– une période statique. L'écart sera calculé entre la la date de fin et la date de début renseignés.
– une période calendaire. En fonction de la période choisie, si on clique sur actuelle/précédente, alors l'écart sera calculé entre l'heure, la journée, la semaine, le mois ou l'année actuelle/précédente et la date actuelle. Par défaut la date sera celle de la journée actuelle.
– une période glissante. L'écart sera alors calculé entre la date actuelle et la N dernière heure, le N dernier jour, la N dernière semaine, le N dernier mois ou la N dernière année

Par défaut, la condition sera sur une valeur ponctuelle.

Capture d'écran de la fenêtre *popup* de création/modification du widget Voyant

La partie « Conditions » sert à établir les conditions qui vont lever une alarme si elles sont validées.

La condition est construite de la manière suivante:

variable OPÉRATEUR_LOGIQUE valeur

La variable correspond à celle renseignée au-dessus.
Pour créer une condition, il faut compléter l'encart en gris en choisissant tout d'abord l'opérateur logique parmi:

- Est égale à (==)
- N'est pas égale à (!=)
- Est inférieure ou égale à (<=)
- Est strictement inférieure à (<)
- Est supérieure ou égale à (>=)
- Est strictement supérieure à (>)

L'opérateur logique par défaut est « Est égale à » (==).

Enfin, remplissez le champ « Valeur ». La valeur écrite dans cette case correspondra à la valeur dont la variable sera soit égale, non égale, inférieure ou égale, strictement inférieure, supérieure ou égale ou bien strictement supérieure.

En cliquant sur le champ « Couleur du voyant », il est possible de modifier la couleur du voyant à l'aide d'un sélectionneur de couleur. On peut aussi directement y écrire la valeur hexadécimale de la couleur souhaitée. Le voyant aura cette couleur tant que la condition n'est pas vérifiée. Par défaut la couleur est bleue (#276cb8).

Si la condition est vérifiée, un texte peut être affiché plutôt que la valeur de la variable. Le texte voulu doit alors être écrit dans le champ « Texte à afficher ». Si on laisse ce champ vide, c'est la valeur d'origine qui sera affichée.

En bas de l'encart gris à droite, le champ « Afficher la valeur brute » sert, si elle est cochée, à afficher la valeur brute de la variable sans unité.

En dessous de l'encart, la couleur du voyant quand la condition est vérifiée est modifiable en cliquant sur le champ correspondant. Par défaut, la couleur est rouge.

A droite, toujours si la condition est vérifiée, un texte peut être affiché plutôt que la valeur de la variable. Le texte voulu doit alors être écrit dans le champ « Texte à afficher ». Si on laisse ce champ vide, c'est la valeur d'origine qui sera affichée.

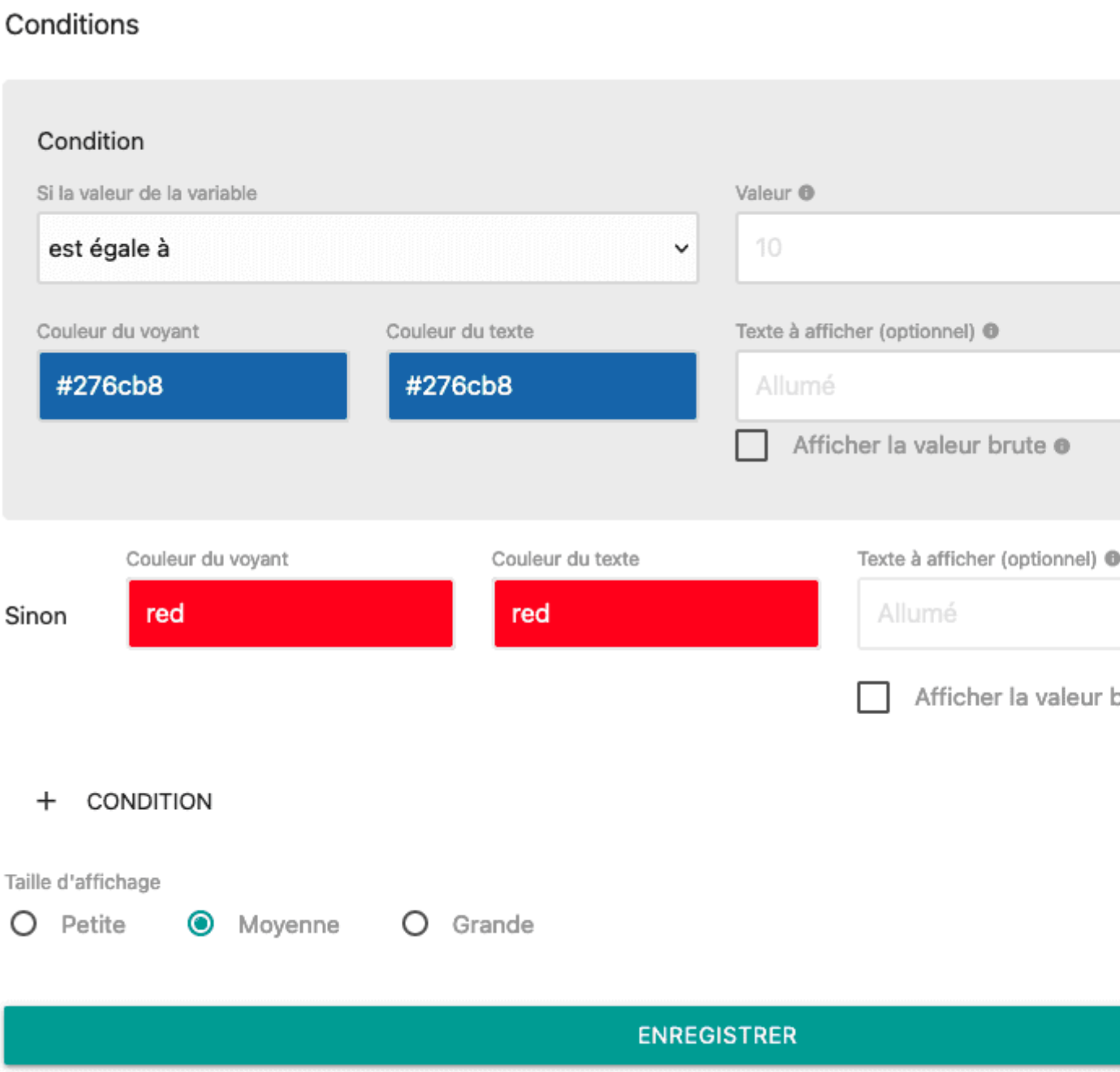
Le champ Afficher la valeur brute sert, si elle est cochée, à afficher la valeur brute de la variable sans unité.

Pour l'instant on a une seule condition. Si on désire rajouter d'autres conditions, cliquer sur « + CONDITION ». Un nouvel encart gris s'affiche alors. Il est à remplir comme le précédent explicité ci-dessus.

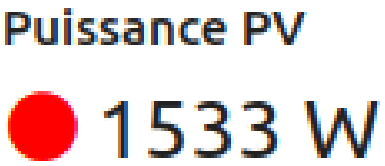
Ainsi le voyant changera de couleur lorsque *la condition1 ET la condition2* seront vraies. Si on a trois conditions, le voyant changera de couleur lorsque *la condition1 ET la condition2 ET la condition3* seront vraies, etc.

La taille de l'affichage est réglable puisque la taille de la police diminue si vous cliquez sur la taille petite. Elle augmente si vous cliquez sur « Moyenne » et elle est de taille maximale si vous cliquez sur

« Grande ». Par défaut la taille est cochée à « Moyenne ».



Capture d’écran de la partie « Condition » de la fenêtre popup de création/modification du widget Voyant



Exemple de widget Voyant

2.5.3 Widget Courbe

Le widget courbe permet d'accéder à une vue d'ensemble de l'évolution des valeurs d'une variable sous forme de courbe.
La fenêtre *popup* de ce widget se compose de trois onglets:

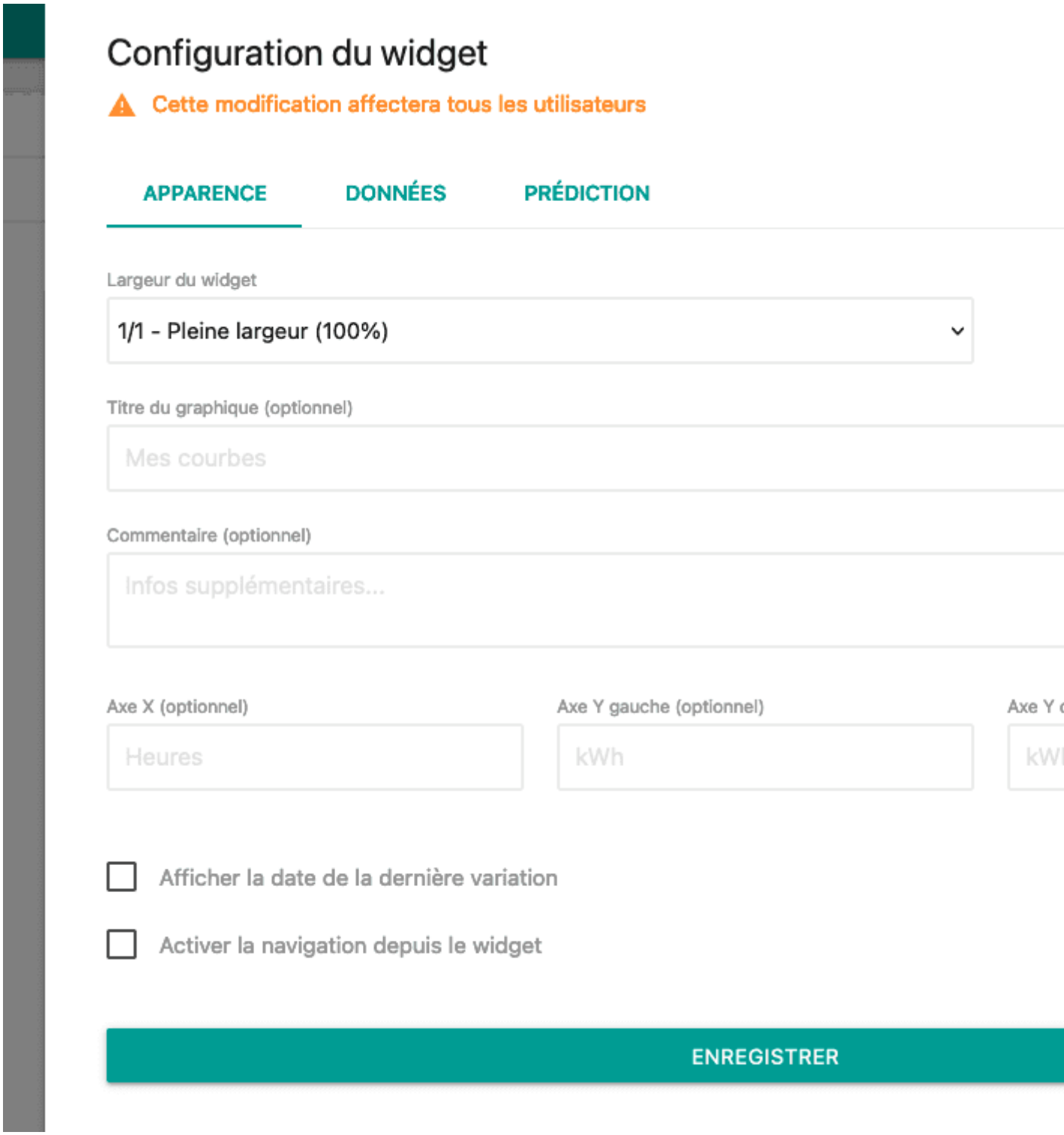
- l'onglet « Apparence »
- l'onglet « Données »
- l'onglet « Prédiction »

Onglet « Apparence »

Cet onglet permet de configurer l'aspect général du widget et son en-tête.
La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

On peut ajouter un titre au graphique et un commentaire en remplissant les deux formulaires optionnels.
En cochant, « Activer la navigation depuis le widget », on débloque les fonctionnalités de changement des dates de début et de fin sur le widget dans le mode visualisation.



Capture d'écran de l'onglet « Apparence » de la fenêtre popup de création/modification du widget Courbe

Onglet « Données »

Cet onglet permet de configurer les caractéristiques du graphe voulu.

La partie « Temporalité » permet de sélectionner le type de période souhaité parmi trois options disponibles:

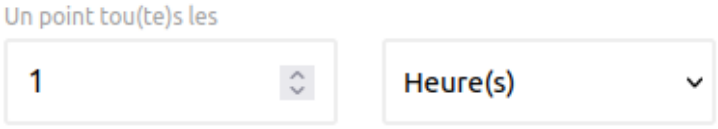
- « Période statique » définit un intervalle de temps d'affichage. Une date de début et de fin est à renseigner.

Capture d'écran de la partie « Période statique » de l'onglet « Données » de la fenêtre *popup* du widget Courbe

- « Période calendaire » définit un intervalle de temps d'affichage par rapport à la date actuelle ou à la date précédente. On peut choisir d'afficher les données de l'heure, du jour, de la semaine, du mois ou de l'année précédente ou actuelle en sélectionnant dans le menu déroulant la période souhaitée ainsi que le décalage. Par défaut la période est celle de la journée actuelle.

Capture d'écran de la partie « Période calendaire » de l'onglet « Données » de la fenêtre *popup* du widget Courbe

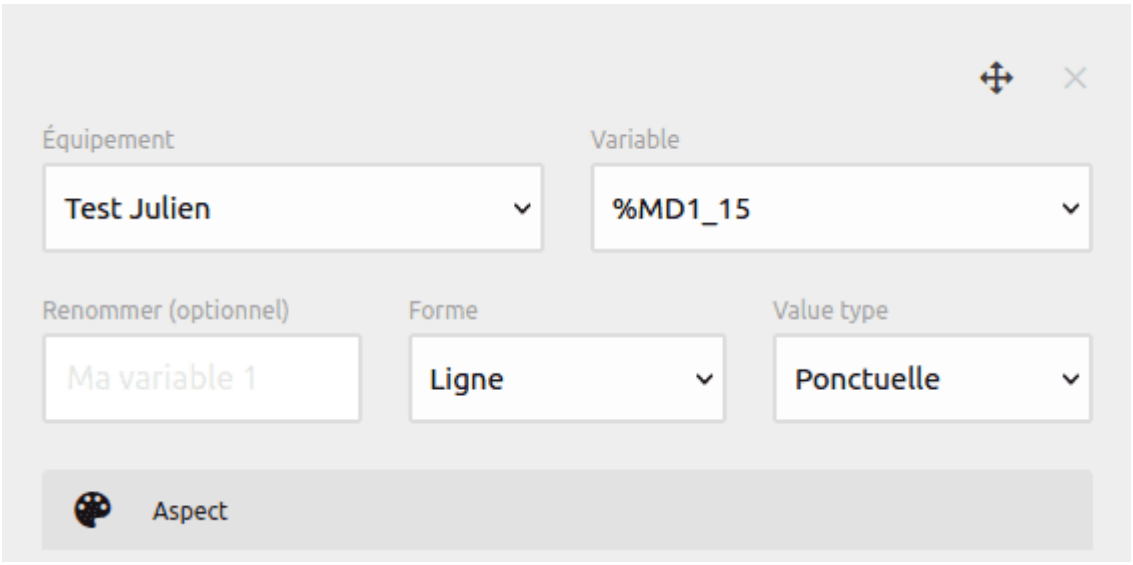
Après avoir choisi une période, on peut modifier la précision des points. Cette précision correspond à l'écart de temps entre deux points affichés sur le graphe. Par défaut, cet écart est d'une heure.



Capture d'écran de la partie « Précision » de l'onglet « Données » du widget Courbe

```
Avertissement
Plus on choisit une longue période avec une petite précision, plus le temps qu
```

La partie « Variables » permet de choisir les variables dont on souhaite que les valeurs soient affichées.
Par défaut, une seule variable est affichée sur le widget. Une variable et ses caractéristiques sont dans un encart gris.

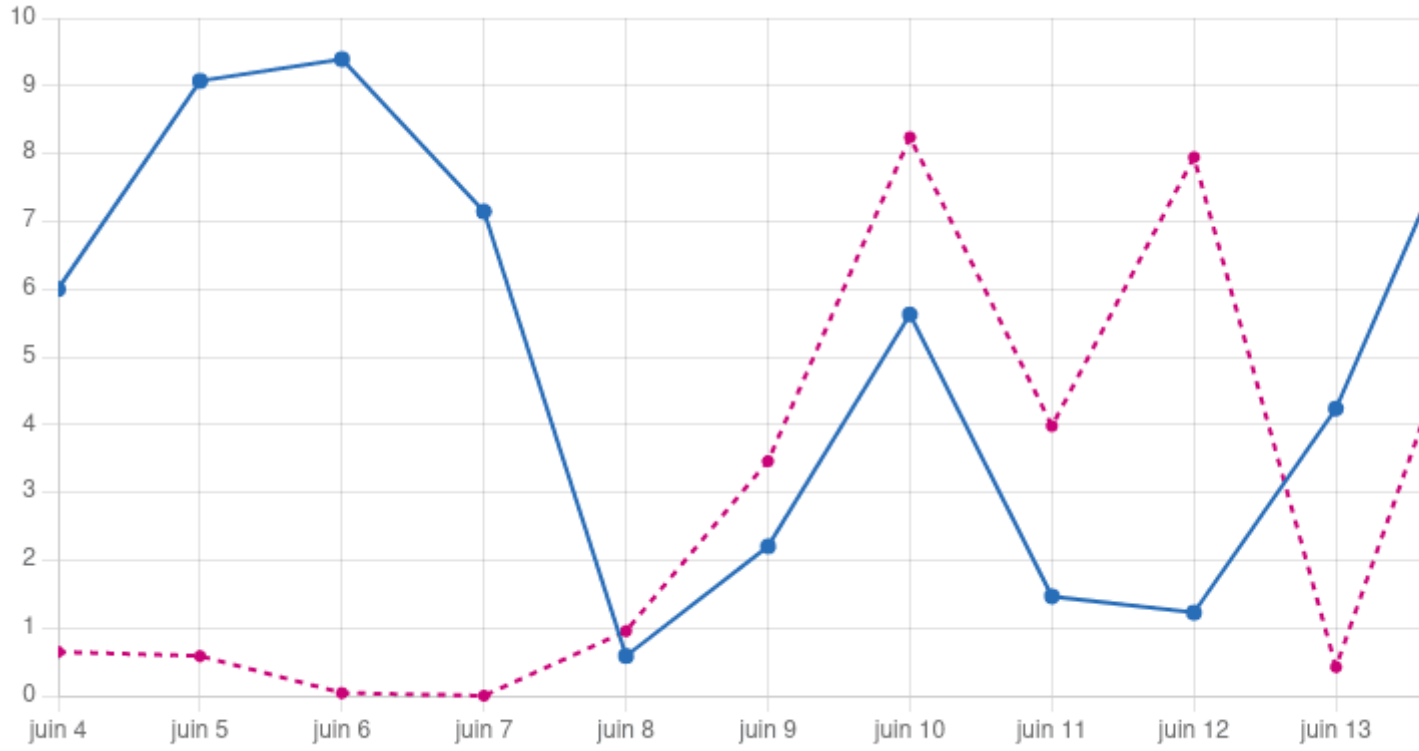


Capture d'écran d'un exemple de variable sélectionnée de la fenêtre *popup* du widget Courbe

Pour choisir la variable, sélectionner son équipement et son nom dans les menus déroulants prévus à cet effet. Il est aussi possible de renommer la variable.

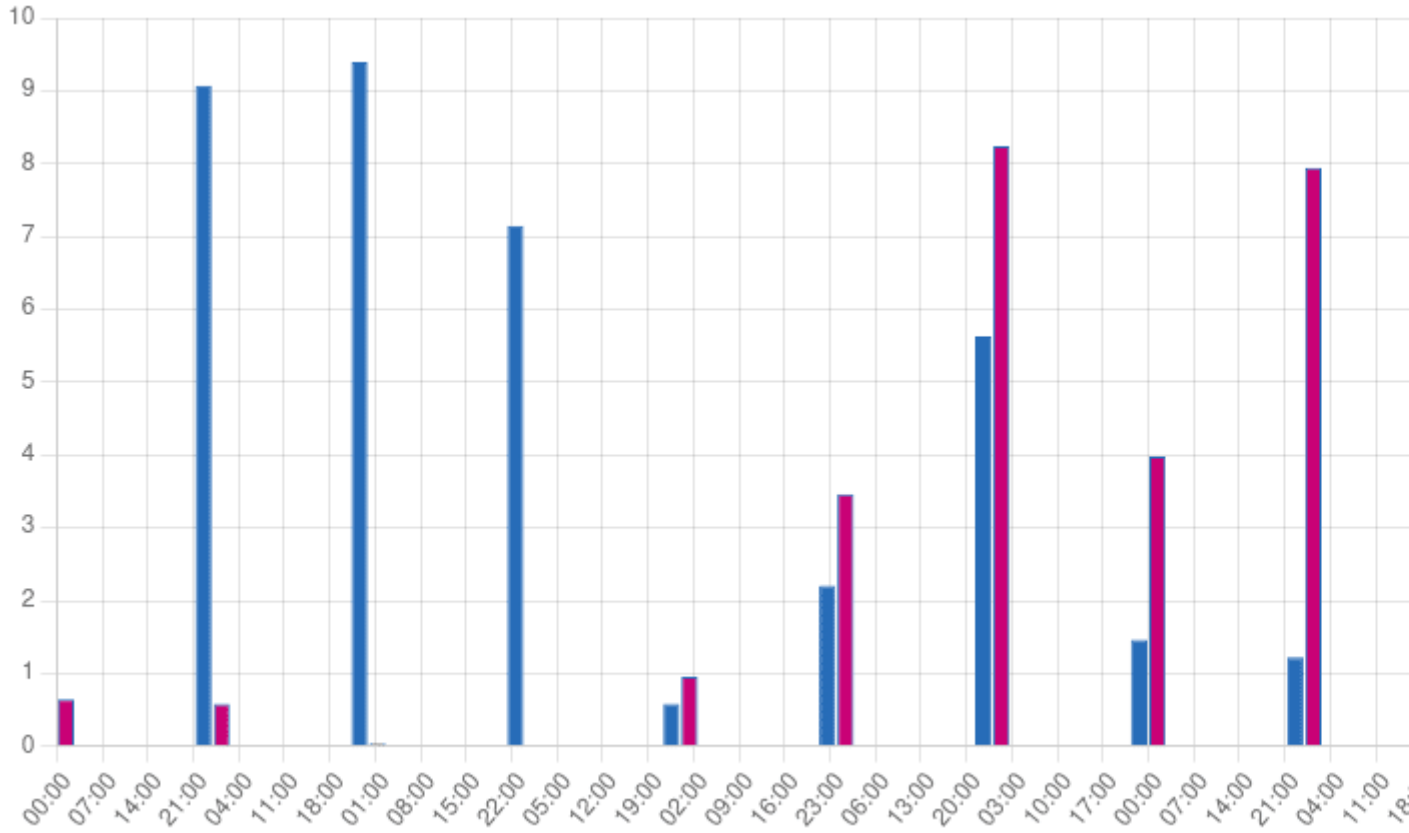
Selon les valeurs que l'on a et l'usage dont on veut en faire, deux formes de graphe sont disponibles dans le champ Forme:

- graphe en Ligne. Un exemple de graphe en ligne est représenté ci-dessous:



Capture d’écran d’un graphe en ligne (widget Courbe)

- graphe en Barre. Un exemple de graphe en barre est représenté ci-dessous:



Capture d'écran d'un graphe en barre (widget Courbe)

```
Avertissement
Il est possible de combiner graphe en ligne et graphe en barre mais il faut fa
```

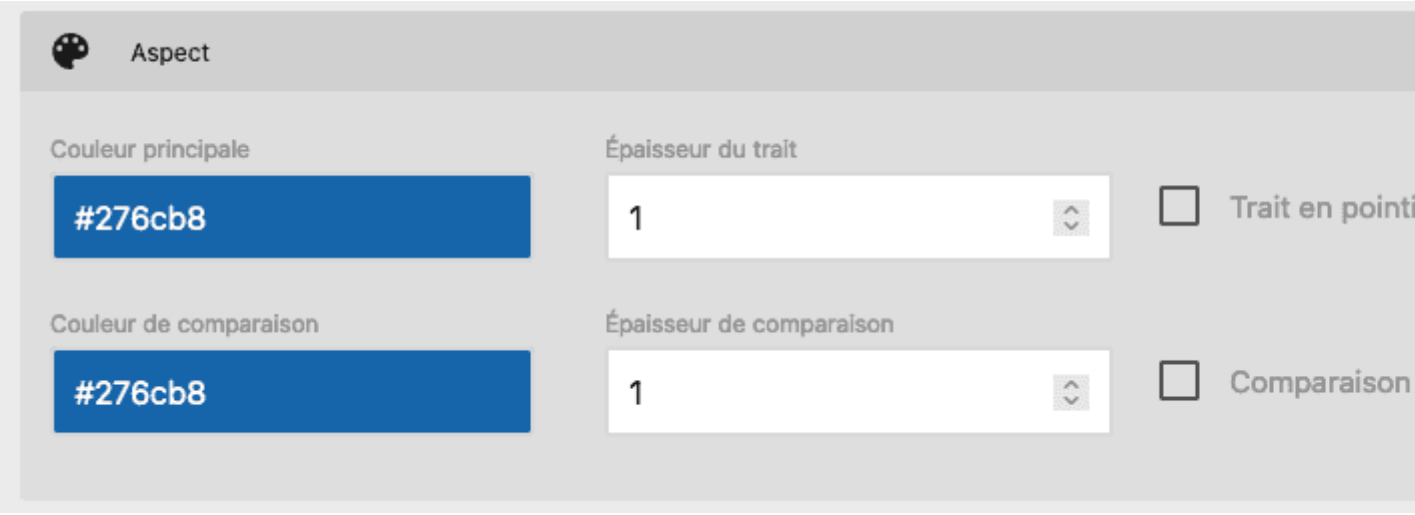
La partie « Type de valeur » permet de choisir quel type de valeur, on souhaite avoir pour chaque point parmi trois options:

- si on veut une valeur ponctuelle (valeur de l'instant), cliquer sur « ponctuelle »
- si on veut une valeur moyennée (moyenne calculée entre deux points), cliquer sur « moyenne »
- si on veut un écart entre deux valeurs (écart entre le point actuel et le point précédent), cliquer sur « écart ».

Par défaut, le graphe affichera des valeurs ponctuelles.

Pour finir on peut choisir l'aspect qu'auront les valeurs de la variable sur le graphe.
Si on a un graphe en ligne, la partie « Aspect » contient six caractéristiques modifiables:

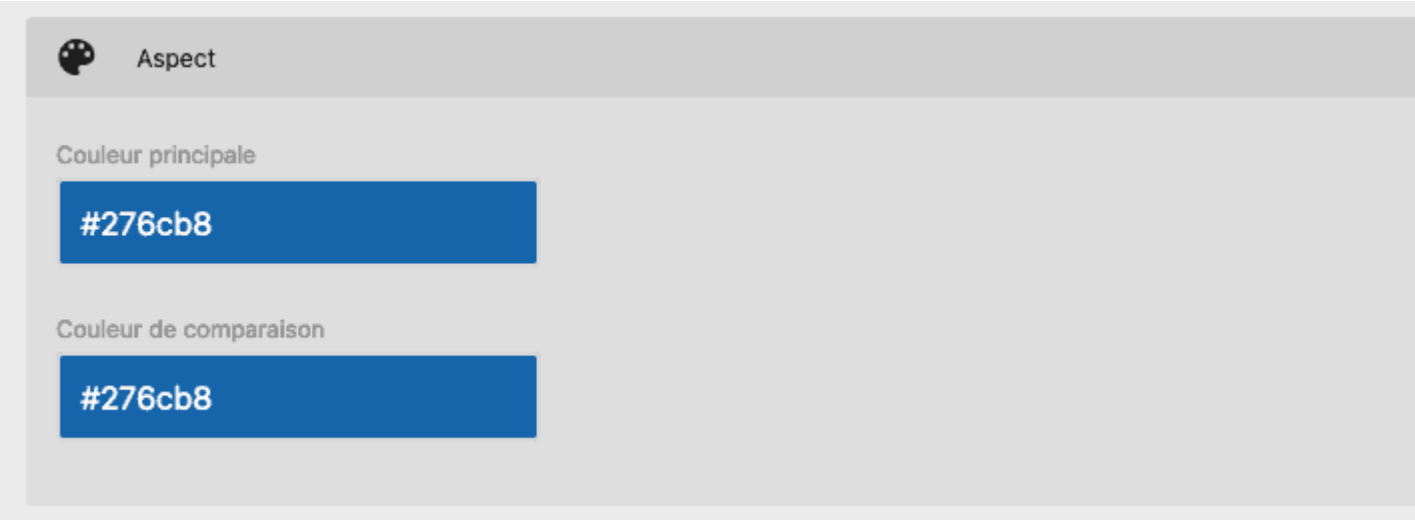
- la couleur principale. Si on clique sur ce champ, on peut modifier la couleur par une autre. La couleur par défaut est bleu (#276cb8).
- l'épaisseur du trait. Cette valeur va de 1 à 4 pour un trait fin vers épais. L'épaisseur par défaut est 1.
- le champ « Trait en pointillé ». Si ce champ est coché alors le trait de la courbe sera en pointillé. Par défaut le champ est décoché.
- les mêmes informations apparaissent pour configurer l'aspect de la courbe de comparaison



Capture d'écran de l'aspect d'une variable d'une ligne de l'onglet « Données » de la fenêtre *popup* de création/modification du widget Courbe

Si on a un graphe en barre, la partie « Aspect » contient trois caractéristiques modifiables:

- la couleur principale. Si on clique sur ce champ, on peut modifier la couleur par une autre. La couleur par défaut est bleu (#276cb8)
- les mêmes informations apparaissent pour configurer l'aspect de la barre de comparaison



Capture d'écran de l'aspect d'une variable d'un graphe barre de l'onglet « Données » de la fenêtre *popup* de création/modification du widget Courbe

```
Information
Il est à noter que la couleur en hexadécimal peut-être écrite dans les cases d
```

Onglet « Prédiction »

Cet onglet comporte une fonctionnalité avancée du widget courbe: la prédiction. Grâce au logiciel NeurEco développé par Adagos, on aura la possibilité de créer un modèle d'intelligence artificielle permettant de prédire les valeurs des variables demandées dans l'onglet « Données ». Ces valeurs dépendront des paramètres que l'on va renseigner dans l'onglet « Prédiction ».

Le bouton « Activer la prédiction » débloque les fonctionnalités de génération de modèle sur le widget ainsi que ces informations dans le mode visualisation. De plus, si le modèle a été généré alors les courbes et/ou les barres prédites seront affichées en plus des courbes réelles.

La partie « Données météorologiques » permet de sélectionner quels paramètres météorologiques on souhaite que notre modèle prennent en compte pour générer le modèle. Les paramètres sélectionnables sont:

- la radiation solaire
- la température réelle de l'air
- la couverture nuageuse
- l'humidité de l'air
- la vitesse du vent
- la direction du vent
- la durée d'ensoleillement
- la précipitation de pluie

Dans cette partie, on nous demande aussi de compléter le champ « Adresse postale » ainsi que les coordonnées géographiques avec la latitude et la longitude à laquelle les données on été récoltées.

La « Période d'apprentissage » permet de sélectionner la période sur laquelle l'algorithme va se baser pour apprendre et construire le modèle. Il y a 4 choix:

- « Aucune période définie ». La période s'étendra alors du premier jour ou il a commencé à y avoir des données jusqu'à maintenant.
- « Période statique » définit un intervalle de temps d'affichage. Une date de début et de fin est à renseigner

Capture d'écran de la partie « Période statique » de l'onglet « Prédiction » de la fenêtre *popup* du widget Courbe

- « Période calendaire » définit un intervalle de temps d'affichage par rapport à la date actuelle ou à la date précédente. On peut choisir d'afficher les données de l'heure, du jour, de la semaine, du mois ou de l'année précédente ou actuelle en sélectionnant dans le menu déroulant la période souhaitée ainsi que le décalage. Par défaut la période sera celle de la journée actuelle.

Capture d'écran de la partie « Période calendaire » de l'onglet « Prédiction » de la fenêtre *popup* du widget Courbe

- « Période glissante » définit un intervalle de temps des dernier(e)(s) heure(s)/ jour(s)/ semaine(s)/ mois jusqu'à la date actuelle. Pour cela, il suffit de renseigner une durée et un type de période dans le menu déroulant. Par défaut, la période s'étendra de hier à aujourd'hui

Capture d'écran de la partie « Période glissante » de l'onglet « Prédiction » de la fenêtre *popup* du widget Courbe

En dessous du choix de la période, on peut décider de la précision des points dans un menu déroulant. La précision est soit à l'heure, au jour, à la semaine ou au mois en sachant qu'il faut une certaines quantités de points pour que le modèle soit fiable. Cette quantité varie en fonctions des données que

l'on a. Par défaut la précision est à l'heure.

Capture d'écran de la précision dans l'onglet « Prédiction » de la fenêtre *popup* de création/modification du widget Courbe

La partie « Période de la prédiction » permet de sélectionner la période sur laquelle on souhaite avoir une prédiction. Pour cela, il suffit d'incrémenter et de décrémenter la durée et de choisir une période parmi prochaine(s) heure(s), prochain(s) jour(s), prochaine(s) semaine(s), prochain(s) mois, prochaine(s) année(s). Cette période va se rajouter à la suite de la période d'affichage que l'on a déjà défini dans l'onglet « Données ».

```
Avertissement
Le temps de génération d'un modèle peut prendre beaucoup de temps en fonction
```

Navigation sur le widget

Fonctionnalités générales

Une fois que l'on a cliqué sur le bouton « Enregistrer », on peut observer notre widget Courbe.

```
Information
Si on ne veut que visualiser les valeurs d'une ou de plusieurs variables, on n
```

Pour zoomer sur une partie du graphe, cliquez sur la zone du graphe que l'on veut agrandir.

Si le champ « Activer la navigation sur le widget » a été coché alors on peut modifier la période d'affichage de la courbe en changeant les dates des champs « Date de début » et « Date de fin » puis en cliquant sur » Appliquer ».

Capture d'écran de la navigation depuis un widget Courbe en mode visualisation

Sur le widget, il est possible de visualiser la date de dernière variation de la variable en bas à gauche du du widget.

Fonctionnalités propres à la prédiction

Si on a aussi rempli l'onglet « Prédiction », alors on va pouvoir dans un premier temps générer un modèle en cliquant sur le bouton « Générer un modèle »

La génération d'un modèle peut prendre un certains temps en fonction de la quantité de données qu'on lui donne en paramètres. On peut donc suivre l'avancée de la génération de ce modèle avec le barre d'avancement située en bas du widget. Plusieurs états d'avancée sont possibles:

- Envoi des données pour l'apprentissage
- Génération du modèle
- Apprentissage terminée

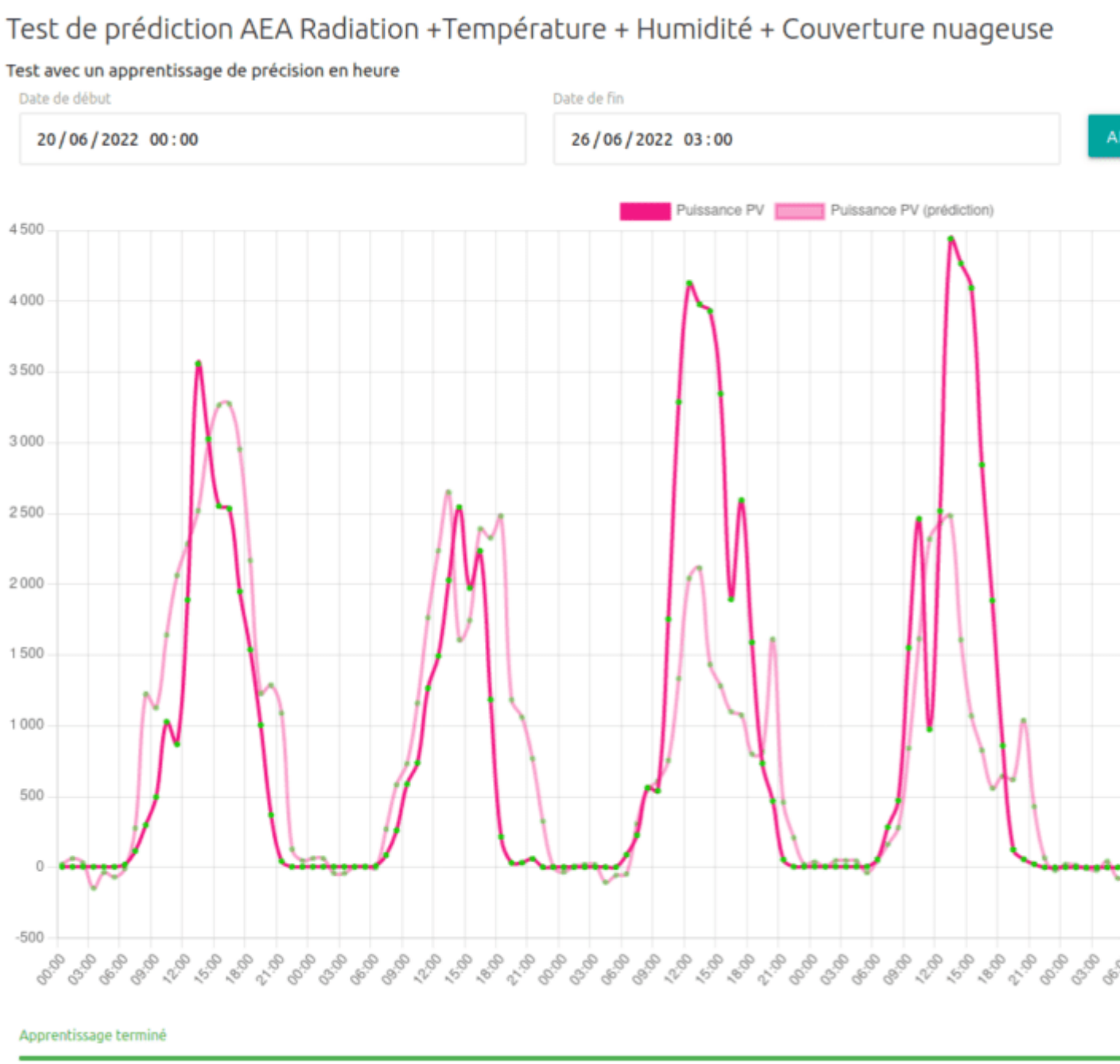
Si un modèle n'a pas encore généré, l'information « Apprentissage non démarré » est affichée

De plus, certaines erreurs peuvent être levées lors de la création du modèle:

- Erreur de communication avec le serveur de prédictions

La génération du modèle sera alors interrompu.

Lorsque la génération s'est déroulée correctement alors on va pouvoir observer que la courbe prédite s'affiche sur le widget (couleur plus claire que les courbe réelle). On peut alors vérifier si notre modèle est correct ou pas en fonction de si la courbe réelle et la courbe prédite se chevauchent (si la période est dans le passé).



Capture d'écran d'un graphe en ligne avec une courbe réelle (rose foncé) et une courbe prédite (rose pâle) (widget Courbe)

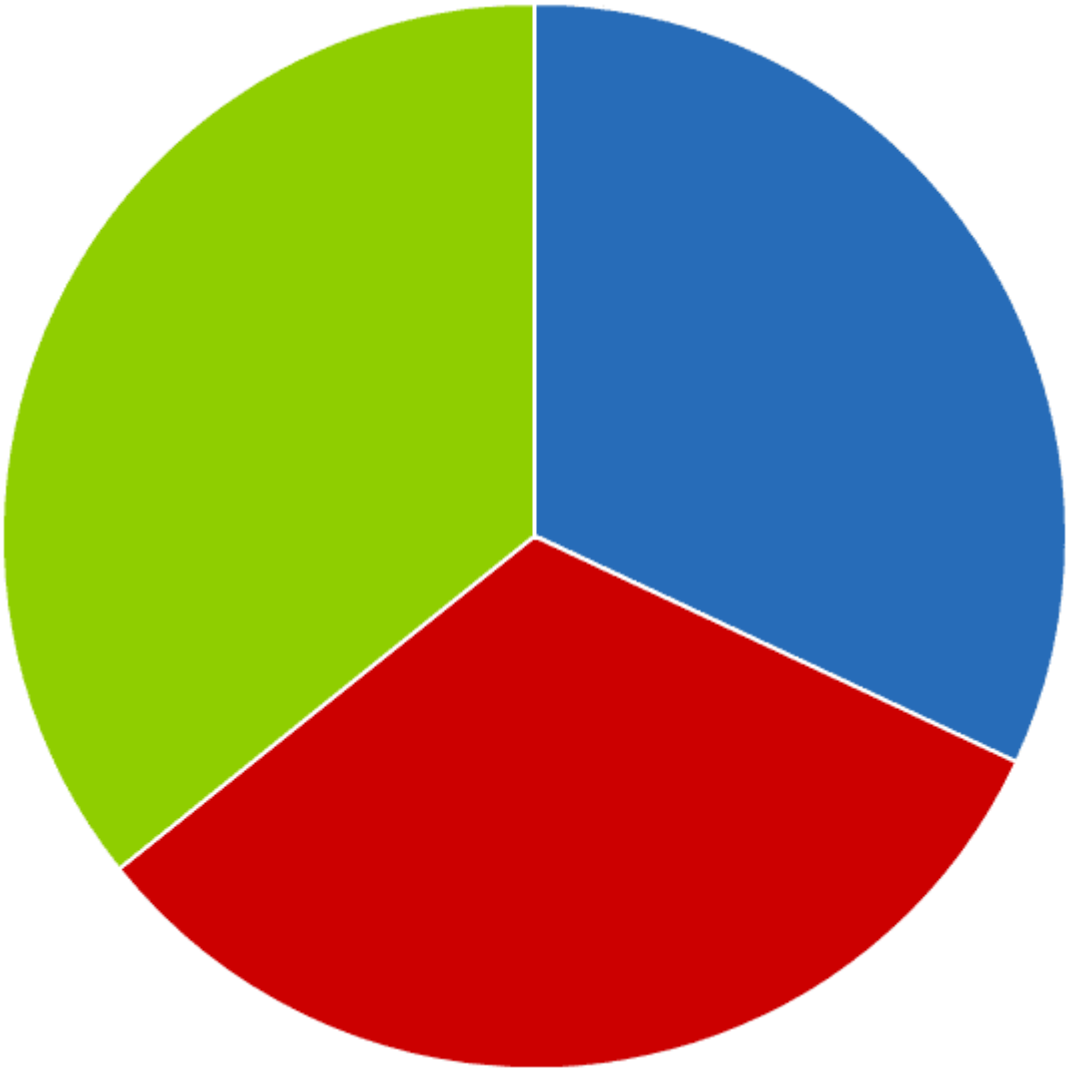
2.5.4 Widget Diagramme

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

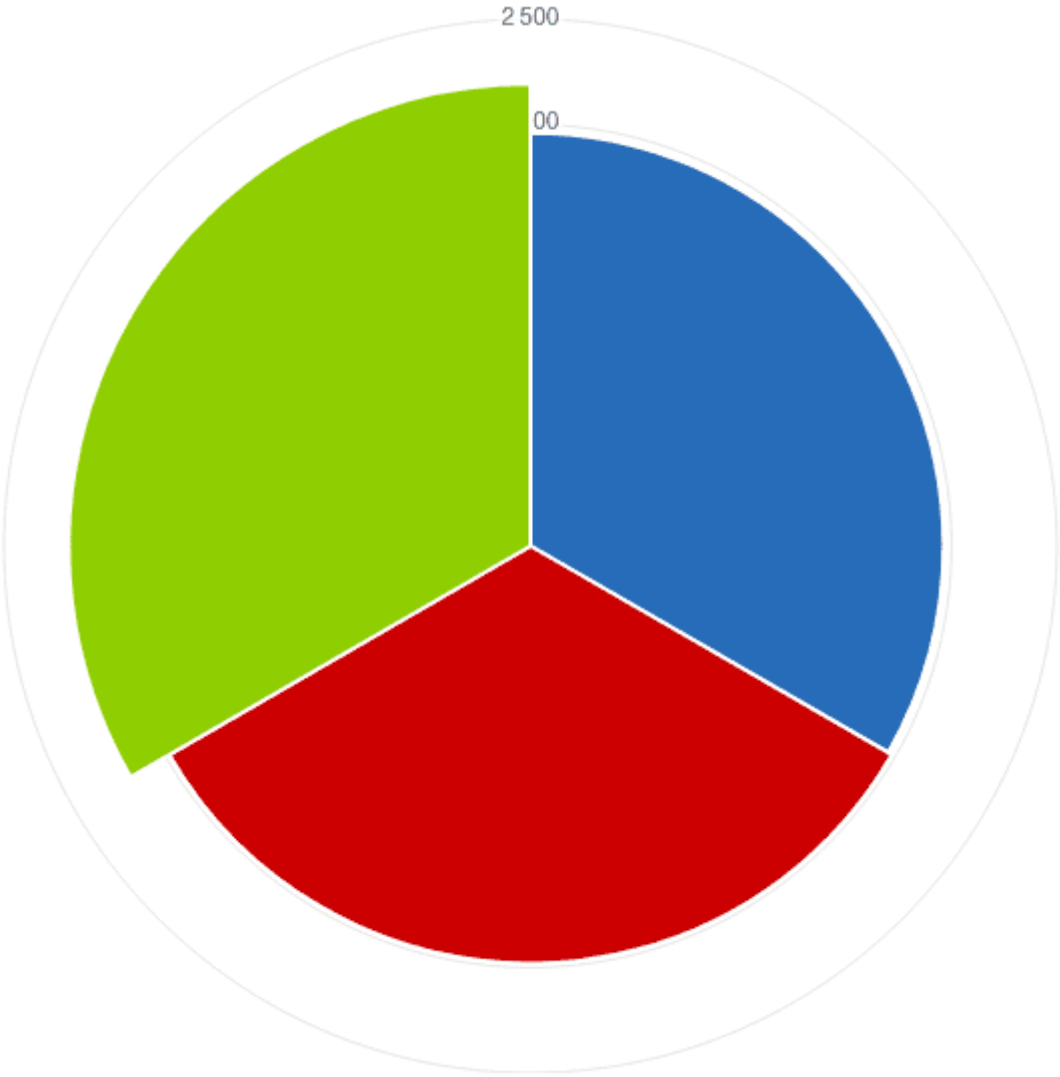
On peut ajouter un titre au graphique et un commentaire en remplissant les deux formulaires optionnels.

Ensuite, on peut sélectionner la forme du diagramme parmi les trois formes suivantes:



- camembert

- anneau



- polaire

Le champ « Type de valeur » permet de choisir quel type de valeur on souhaite que notre diagramme représente parmi trois options:

- si on veut des valeurs ponctuelles (instantanées), cliquer sur « Ponctuelle »
- si on veut des valeurs moyennées, cliquer sur « Moyenne ». De nouvelles informations de temporalité sont alors demandées. On a ainsi le choix de calculer les moyennes sur:

– une période statique. Un intervalle de temps avec une date de début et de fin est alors à renseigner
– une période calendaire. En fonction de la période choisie, si on clique sur précédente, l'intervalle s'étendra alors de l'heure, du jour, de la semaine, du mois ou de l'année précédente jusqu'à la date actuelle. En fonction de la période choisie, si on clique sur actuelle, alors la moyenne sera calculée sur l'heure, la journée, la semaine, le mois ou l'année actuelle. Par défaut la période est celle de la journée actuelle.

– une période glissante. En fonction de la durée N rentrée et de la période, l'intervalle de temps s'étendra sur les N dernière(s) heure(s), les N dernier(s) jour(s), les N dernière(s) semaine(s), les N dernier(s) mois ou les N dernière(s) année(s)

- si on veut un écart entre deux valeurs, cliquer sur « Écart ». De nouvelles informations de temporalité sont alors demandées. On a ainsi le choix de calculer l'écart entre deux valeurs sur:

– une période statique. L'écart sera calculé entre la la date de fin et la date de début renseignés.
– une période calendaire. En fonction de la période choisie, si on clique sur actuelle/précédente, alors l'écart sera calculé entre l'heure, la journée, la semaine, le mois ou l'année actuelle/précédente et la date actuelle. Par défaut la date sera celle de la journée actuelle.
– une période glissante. L'écart sera alors calculé entre la date actuelle et la N dernière heure, le N dernier jour, la N dernière semaine, le N dernier mois ou la N dernière année

Par défaut, des valeurs ponctuelles seront représentées sur le diagramme.

Pour que le diagramme soit construit, il a besoin de valeurs. Pour cela, il faut compléter la partie « Variables ».

Pour chaque variable, sélectionnez l'équipement auquel elle appartient ainsi que son nom dans les menus déroulants.

Si le nom de la variable ne convient pas, on peut le renommer en remplissant le champ « Renommer ».

La variable aura aussi une couleur sur le diagramme. Cette couleur est par défaut bleue (#276cb8) mais est modifiable en cliquant sur le champ « Couleur ». Un sélectionneur de couleur s'affiche.

```
Information
Il est à noter que la couleur en hexadécimal peut-être écrite dans les cases d
```

Pour ajouter une variable sur le diagramme, cliquez sur + Variable. Un nouvel encart apparaît. Il est à compléter de la même manière qu'explicitée précédemment.

2.5.5 Widget Radar

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)

- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

On peut ajouter un titre au graphique et un commentaire en remplissant les deux formulaires optionnels.

Le champ « Type de valeur » permet de choisir quel type de valeur on souhaite que notre radar représente parmi trois options:

- si on veut des valeurs ponctuelles (instantanées), cliquer sur « Ponctuelle »
- si on veut des valeurs moyennées, cliquer sur « Moyenne ». De nouvelles informations de temporalité sont alors demandées. On a ainsi le choix de calculer les moyennes sur:

– une période statique. Un intervalle de temps avec une date de début et de fin est alors à renseigner
– une période calendaire. En fonction de la période choisie, si on clique sur précédente, l'intervalle s'étendra alors de l'heure, du jour, de la semaine, du mois ou de l'année précédente jusqu'à la date actuelle. En fonction de la période choisie, si on clique sur actuelle, alors la moyenne sera calculée sur l'heure, la journée, la semaine, le mois ou l'année actuelle. Par défaut la période est celle de la journée actuelle
– une période glissante. En fonction de la durée N rentrée et de la période, l'intervalle de temps s'étendra sur les N dernière(s) heure(s), les N dernier(s) jour(s), les N dernière(s) semaine(s), les N dernier(s) mois ou les N dernière(s) année(s)

- si on veut un écart entre deux valeurs, cliquer sur « Écart ». De nouvelles informations de temporalité sont alors demandées. On a ainsi le choix de calculer l'écart entre deux valeurs sur:

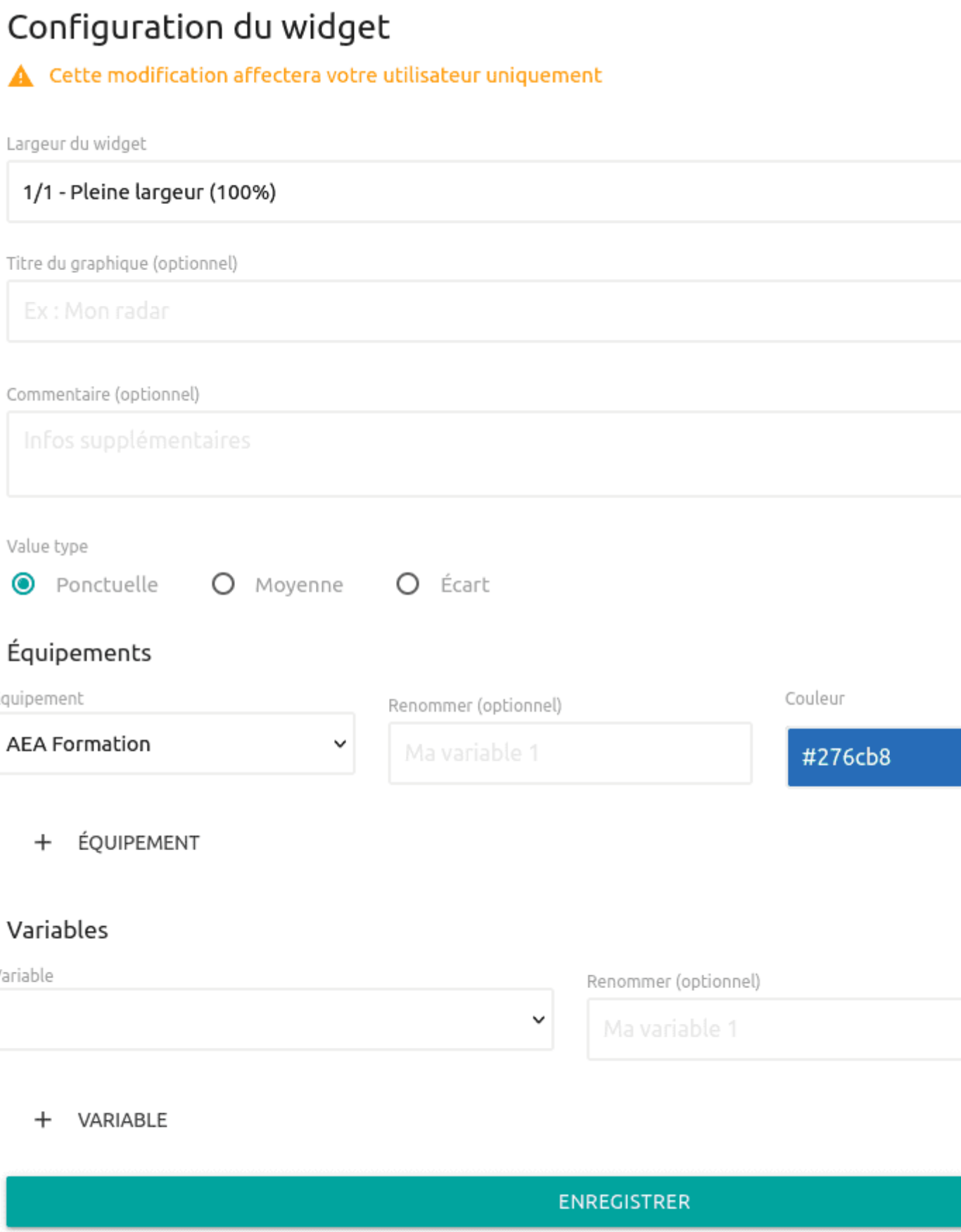
– une période statique. L'écart sera calculé entre la la date de fin et la date de début renseignés
– une période calendaire. En fonction de la période choisie, si on clique sur actuelle/précédente, alors l'écart sera calculé entre l'heure, la journée, la semaine, le mois ou l'année actuelle/précédente et la date actuelle. Par défaut la date sera celle de la journée actuelle
– une période glissante. L'écart sera alors calculé entre la date actuelle et la N dernière heure, le N dernier jour, la N dernière semaine, le N dernier mois ou la N dernière année

Par défaut, des valeurs ponctuelles seront représentées sur le diagramme.

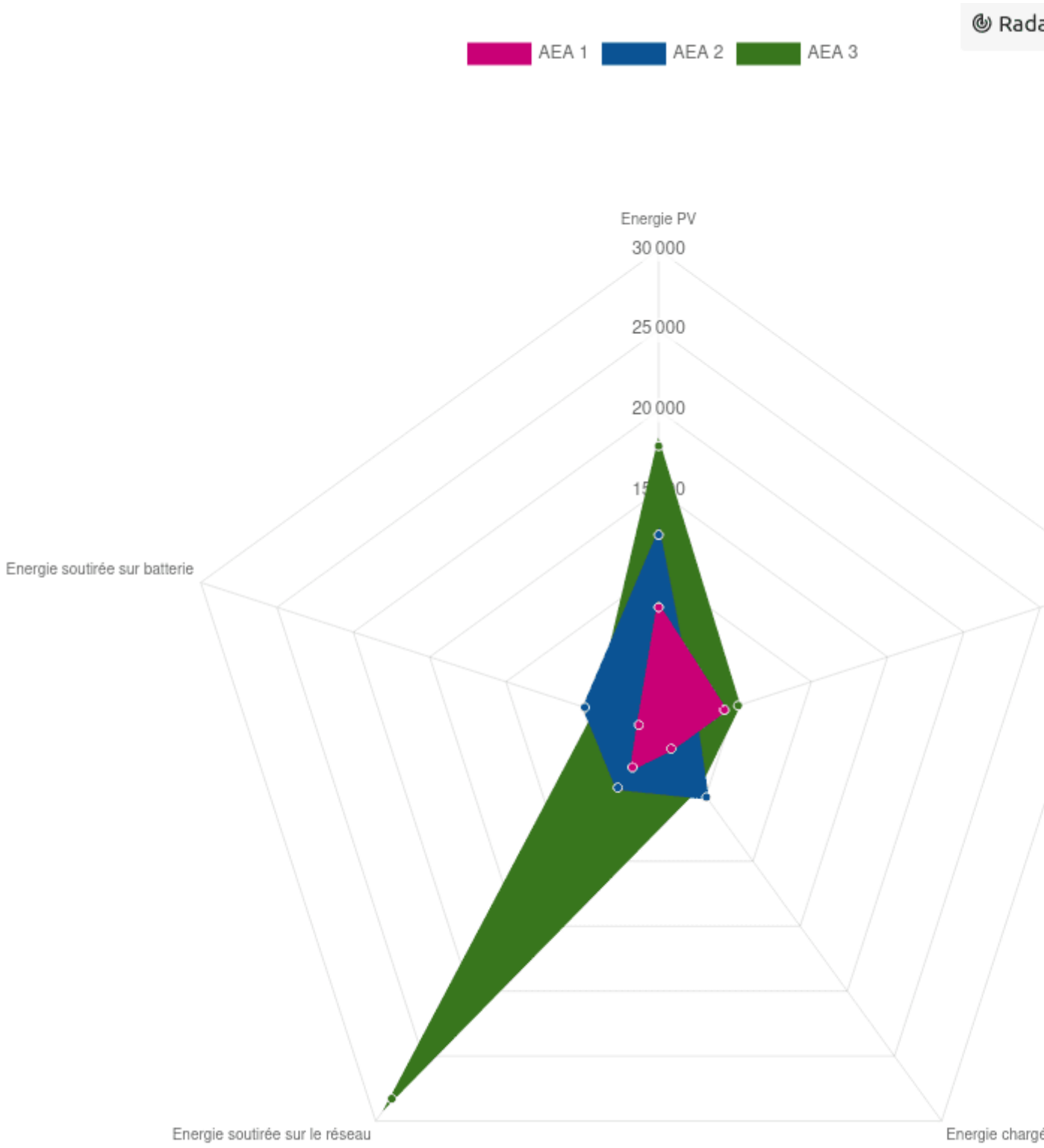
Les données que le widget va représenter vont être celles d'équipements. Dans la partie « Équipement », sélectionnez dans le menu déroulant du champ « Équipement », l'équipement souhaité.
On peut renommer son nom sur le widget en remplissant le champ « Renommer ». Cependant, ce remplissage est optionnel.
Pour ajouter un axe, cliquez sur + Équipement.

Les axes du diagramme vont être des variables. Dans la partie « Variable », sélectionnez dans le menu déroulant du champ « Variable », la variable souhaitée.
On peut renommer son nom sur le widget en remplissant le champ « Renommer ». Cependant, ce remplissage est optionnel.
Pour ajouter un axe, cliquez sur + Variable.

Information
Le diagramme en radar sert à représenter sur un plan en deux dimensions au moi


Configuration du widget
Cette modification affectera votre utilisateur uniquement
Largeur du widget
1/1 - Pleine largeur (100%)
Titre du graphique (optionnel)
Ex : Mon radar
Commentaire (optionnel)
Infos supplémentaires
Value type
Ponctuelle
Moyenne
Écart
Équipements
Équipement
AEA Formation
Renommer (optionnel)
Ma variable 1
Couleur
#276cb8
+ ÉQUIPEMENT
Variables
Variable
Renommer (optionnel)
Ma variable 1
+ VARIABLE
ENREGISTRER

Capture d’écran de la fenêtre *popup* de création/modification du widget Radar



Capture d’écran d’un exemple de diagramme Radar

2.5.6 Widget Tableau

Ce widget sert à créer et visualiser un tableau. Plusieurs types de tableaux sont déjà pré-configurés mais on a la possibilité de personnaliser nos propres tableaux en fonction des besoins.

Onglet « Apparence »

Cet onglet permet de configurer l'aspect général du widget et son en-tête.

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

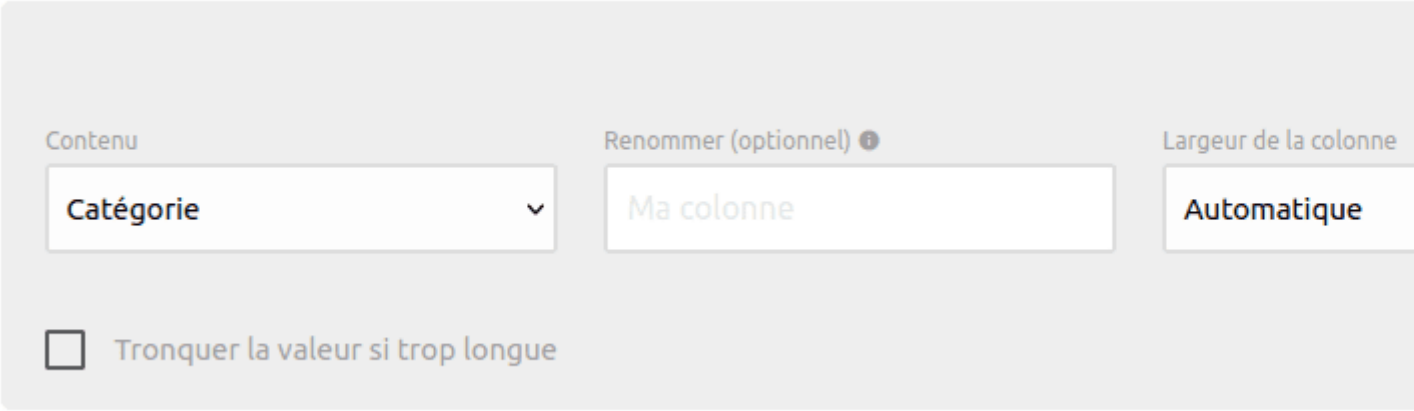
On peut ajouter un Titre au graphique et un commentaire en remplissant les deux formulaires optionnels.

Onglet « Données »

Le tableau peut afficher différentes informations en fonction du type de contenu voulu. En cliquant sur le champ « Type de contenu », on a le choix entre les informations sur:

- le calendrier
- les comptes utilisateurs
- les équipements
- les procédures
- les profils utilisateurs
- les reports
- les variables

Une fois que l'on a sélectionné le type de contenu, on observe les colonnes par défaut du tableau. Une colonne est représentée dans un encart gris suivant l'exemple:



Capture d'écran d'une colonne de l'onglet « Données » de la fenêtre *popup* du widget Tableau

Le nom de la colonne est écrit dans le champ « Contenu ». On peut sélectionner une autre colonne dans le menu déroulant.

```
Avertissement
Si on choisit un nouvelle colonne alors il se peut déjà qu'elle soit sélection
```

En remplissant, le champ « Renommer », on renomme le nom de la colonne. Cette action est optionnelle et par défaut le nom de la colonne sera celui inscrit dans le champ « Contenu ».

On a aussi la possibilité de choisir la taille de la largeur de la colonne. Par défaut, la taille de la colonne sera ajustée automatiquement mais on peut la changer en cliquant sur « Largeur de la Colonne ». Elle pourra être ainsi:

- petite (10%)
- moyenne (25%)
- grande (50%)

En cochant, « Tronquer la valeur si trop longue » alors le tableau va afficher des variables tronquées dont le nombre de chiffres significatifs dépendra de la taille du widget.

Pour supprimer la colonne, cliquez sur la croix rouge en haut à droite.

Pour changer l’ordre des colonnes, cliquez sur le bouton en haut à droite puis faites glisser la colonne vers l’endroit souhaité.

Pour ajouter une colonne, cliquez sur le bouton
+ COLONNE. Une nouveau encart gris à compléter apparaît.

Chaque type de contenu, il y a des colonnes par défaut et des colonnes additionnelles disponibles (se référer au tableau ci-dessous).

Type de contenu	Colonnes par défaut	Colonnes disponibles
Calendrier	– Catégorie – Équipement – Variable – Commentaire – Date de début – Date de fin – Valeur de début – Valeur de retour – Occurrence	– Catégorie de la variable
Comptes utilisateurs	– Intitulé – Commentaire – Profil	– Activé – Format régional – Adresse email – Numéro de téléphone – Filtre zones – Filtres équipements – Période de disponibilité – Période d’indisponibilité
Équipements	– Index – Type – Intitulé – Zones – Etat de la com. – Alarmes	– Activé – Commentaire – Timeout (en secondes) – Délai de rafraîchissement (en secondes) – Protocole de com. – Port de com. – Numéro d’esclaves

Type de contenu	Colonnes par défaut	Colonnes disponibles
Procédures	– Intitulé – Commentaire – Variables associées – Exécuter au démarrage – Exécuter périodiquement	– Jours de la semaine – Jours du mois – Mois – Heures – Minutes – Secondes – Code
Profils utilisateurs	– Intitulé – Commentaire – Peut administrer le système – Peut gérer les utilisateurs	– Filtre catégories – Peut rafraîchir les équipements – Peut forcer les variables – Peut acquitter les alarmes
Reports	– Intitulé – Commentaire – Type	– Destinataire – Envoyé sur déclenchement d’alarme – Sujet sur déclenchement d’alarme – Message sur déclenchement d’alarme – Envoyer sur arrêt d’alarme – Sujet sur arrêt d’alarme – Message sur arrêt d’alarme – Filtre zones – Filtre équipements – Filtre catégories – Filtre types de variables

Type de contenu	Colonnes par défaut	Colonnes disponibles
Variables	– Équipement – Adresse – Mnémonique – Libellé – Valeur – Dernière variation	– Type – Registre – Commentaire – Interne – Publique – Forçable – Priorité – Valeur de forçage min. – Valeur de forçage max. – Catégories – Format – Précision – Valeur brute min. – Valeur brute max. – Valeur mise à l’échelle min. – Valeur mise à l’échelle max. – Générer des alarmes – Libellé des alarmes – Condition de l’alarme – Générer des événements – Libellé des événements – Condition de l’événement – Historiser les valeurs – Moyenner la courbe – Délai entre deux points (en secondes) – Hystérésis

Tableau des différentes colonnes pour chaque contenu

```
Information
Pour créer le tableau que l’on souhaite, il faut modifier les colonnes par déf
```

Onglet « Tri & filtre »

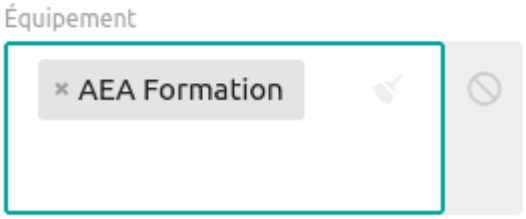
Tri par défaut

Pour visualiser la tableau, il est nécessaire de sélectionner une colonne (dans le menu déroulant Colonne) dont les éléments seront triés soit par ordre ascendant ou descendant (dans le menu déroulant Ordre).

Filtre par défaut

Si on souhaite visualiser des éléments précis, on peut filtrer en sélectionnant le ou les attributs de filtrage d’une colonne. Il est tout à fait possible de filtrer sur une ou plusieurs colonnes.

On peut supprimer un attribut en cliquant sur la croix à sa gauche (sur l'attribut même)(par exemple pour supprimer l'attribut AEA Formation, cliquez sur la croix à sa gauche).



Pour exclure un filtre sur une colonne, cliquez sur le bouton

La case de la colonne est alors entourée en rouge

Pour vider un filtre de tous ses attributs, cliquez sur l'icône du balai à droite dans la case filtrée.

Pour chaque type de contenu, on peut filtrer sur différentes colonnes (se référer au tableau ci-dessous).

Contenu	Filtre sur la/les colonnes
Calendrier	– Équipement – Variable – Catégorie

Contenu	Filtre sur la/les colonnes
Comptes utilisateurs	– Intitulé – Activé – Commentaire – Profil – Format régional – Adresse email – Numéro de téléphone – Filtre zones – Filtres équipements
Équipements	– Index – Activé – Type – Intitulé – Commentaire – Zones – Timeout (en secondes) – Délai de rafraîchissement (en secondes) – Protocole de com. – Port de com. – Numéro d’esclaves
Procédures	– Intitulé – Commentaire – Variables associées – Exécuter au démarrage – Exécuter périodiquement – Jours de la semaine – Jours du mois – Mois – Heures – Minutes – Secondes – Code
Profils utilisateurs	– Intitulé – Commentaire – Filtre catégories – Peut administrer le système – Peut gérer les utilisateurs – Peut rafraîchir les équipements – Peut forcer les variables – Peut acquitter les alarmes

Contenu	Filtre sur la/les colonnes
Reports	– Intitulé – Commentaire – Type – Destinataire – Envoyé sur déclenchement d'alarme – Sujet sur déclenchement d'alarme – Message sur déclenchement d'alarme – Envoyer sur arrêt d'alarme – Sujet sur arrêt d'alarme – Message sur arrêt d'alarme – Filtre zones – Filtre équipements – Filtre catégories – Filtre types de variables
Variables	– Équipement – Type – Mnémonique – Libellé – Catégorie – Valeur

Tableau des colonnes sur lesquelles on peut filtrer en fonction du type de contenu

2.5.7 Widget Alarmes

Ce widget sert à afficher la liste des alarmes qui se sont levées. Ces alarmes peuvent toujours être levées ou bien éteintes. Elles peuvent être aussi acquittées.

Onglet « Apparence »

Cet onglet permet de configurer l'aspect général du widget et son en-tête.

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

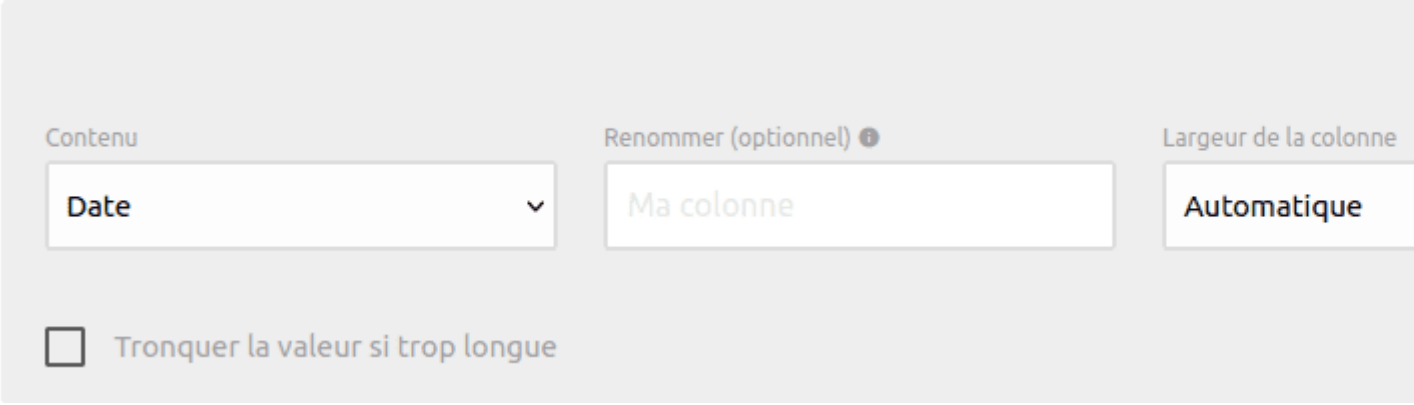
- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

On peut ajouter un nom au widget et un commentaire en remplissant les deux formulaires optionnels.

En cochant, « Activer le filtre depuis le widget », on débloque les fonctionnalités de filtrage des alarmes sur le widget depuis le mode visualisation.

Onglet « Données »

La liste des alarmes va être présentée sous forme d'un tableau modulable.
Un tableau possède plusieurs colonnes.
Une colonne est représentée dans un encart gris suivant l'exemple suivant:



Capture d'écran d'un exemple de colonne dans l'onglet « Données » de la fenêtre *popup* de création/modification du widget Alarme

Le nom de la colonne est écrit dans le champ « Contenu ». On peut sélectionner une autre colonne dans son menu déroulant.

```
Avertissement
Si on choisit un nouvelle colonne alors il se peut déjà qu'elle soit sélection
```

Le tableau par défaut sera composé de quatre colonnes dont les noms par ordre sont:

- Date
- Équipement
- Variable
- Alarme

Une cinquième colonne (Catégorie de la variable) est disponible dans le menu déroulant du champ « Contenu ».

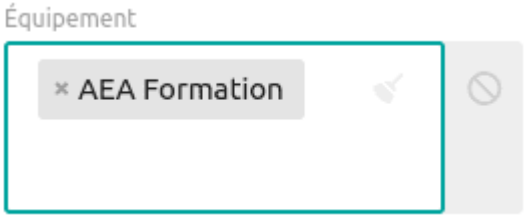
En remplissant, le champ « Renommer », on renomme le nom de la colonne. Cette action est optionnelle et par défaut le nom de la colonne sera celui inscrit dans le champ « Contenu ».

On a aussi la possibilité de choisir la taille de la largeur de la colonne. Par défaut la taille de la colonne sera ajustée automatiquement mais on peut la changer en cliquant sur « Largeur de la Colonne ». Elle pourra être ainsi:

- petite (10%)

- moyenne (25%)
- grande (50%)

En cochant, « Tronquer la valeur si trop longue » alors le tableau va afficher des variables tronquées dont le nombre de chiffre significatif dépendra de la taille du widget.



Pour supprimer la colonne, cliquez sur la croix rouge en haut à droite.

Pour changer l’ordre des colonnes, cliquez sur l’icône à quatre flèches en haut à droite puis faites glisser la colonne vers l’endroit souhaité.

Pour ajouter une colonne, cliquez sur le bouton
+ COLONNE en bas. Une nouveau encart gris à compléter apparaît.

Onglet « Tri & filtre »

Tri par défaut

Pour visualiser la tableau, il est nécessaire de sélectionner une colonne (dans le menu déroulant Colonne) dont les éléments seront triés soit par ordre ascendant ou descendant (dans le menu déroulant Ordre).

On a le choix de trier selon cinq colonnes:

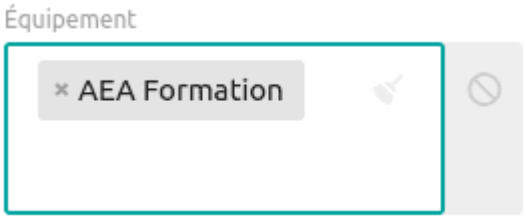
- Alarme
- Catégories de la variable
- Date
- Équipement

- Variable

Filtre par défaut

Si on souhaite visualiser des éléments précis, on peut filtrer en sélectionnant le ou les attributs de filtrage d'une colonne. Il est tout à fait possible de filtrer sur plusieurs colonnes.

On peut supprimer un attribut en cliquant sur la croix à sa gauche (sur l'attribut même)(par exemple pour supprimer l'attribut AEA Formation, cliquez sur la croix à sa gauche).



Le filtre sur une colonne en particulier peut-être mis en place en non. Pour exclure un filtre sur une colonne, cliquez sur le bouton suivant:

La case de la colonne est alors entourée en rouge.

Pour vider un filtre de tous ses attributs, cliquez sur l'icône du balai à droite dans la case filtrée.

Pour le type de contenu calendrier, on peut filtrer sur la/les colonne(s) qui constitue(nt) notre tableau. Ces colonnes sont les mêmes que celles déclarées précédemment dans l'onglet « Données ».

En plus de réaliser un filtrage sur les colonnes, on peut filtrer les alarmes sur une période précise dans la partie « Date ».

Quatre choix de période s'offre à nous:

- « Aucune période définie »: toutes les alarmes vont être affichées.

- « Période statique »: on doit choisir la date de début et de fin délimitant la période. En cliquant sur la case Date de début ou Date de fin, on a la possibilité de rentrer la date en l'écrivant ou bien en la sélectionnant grâce au sélectionneur de date affiché.

Capture d'écran de la partie « Période statique » de l'onglet « Filtre » de la fenêtre *popup* du widget Alarme

- « Période calendaire »: définit un intervalle de temps d'affichage par rapport à la date actuelle ou à la date précédente. On peut choisir d'afficher les alarmes de l'heure, du jour, de la semaine, du mois ou de l'année précédente ou actuelle en sélectionnant dans le menu déroulant la période souhaitée ainsi que le décalage. Par défaut la période est celle de la journée actuelle.

Capture d'écran de la partie « Période calendaire » de l'onglet « Filtre » de la fenêtre *popup* du widget Alarme

- « Période glissante »: définit un intervalle de temps des dernier(e)(s) heure(s)/ jour(s)/ semaine(s)/ mois jusqu'à la date actuelle. Pour cela, il suffit de renseigner une durée et un type de période dans le menu déroulant. Par défaut, la période s'étend de hier à aujourd'hui.

Capture d'écran de la partie « Période glissante » de l'onglet « Filtre » de la fenêtre *popup* du widget Alarme

Navigation sur le widget

Filtrage en mode visualisation

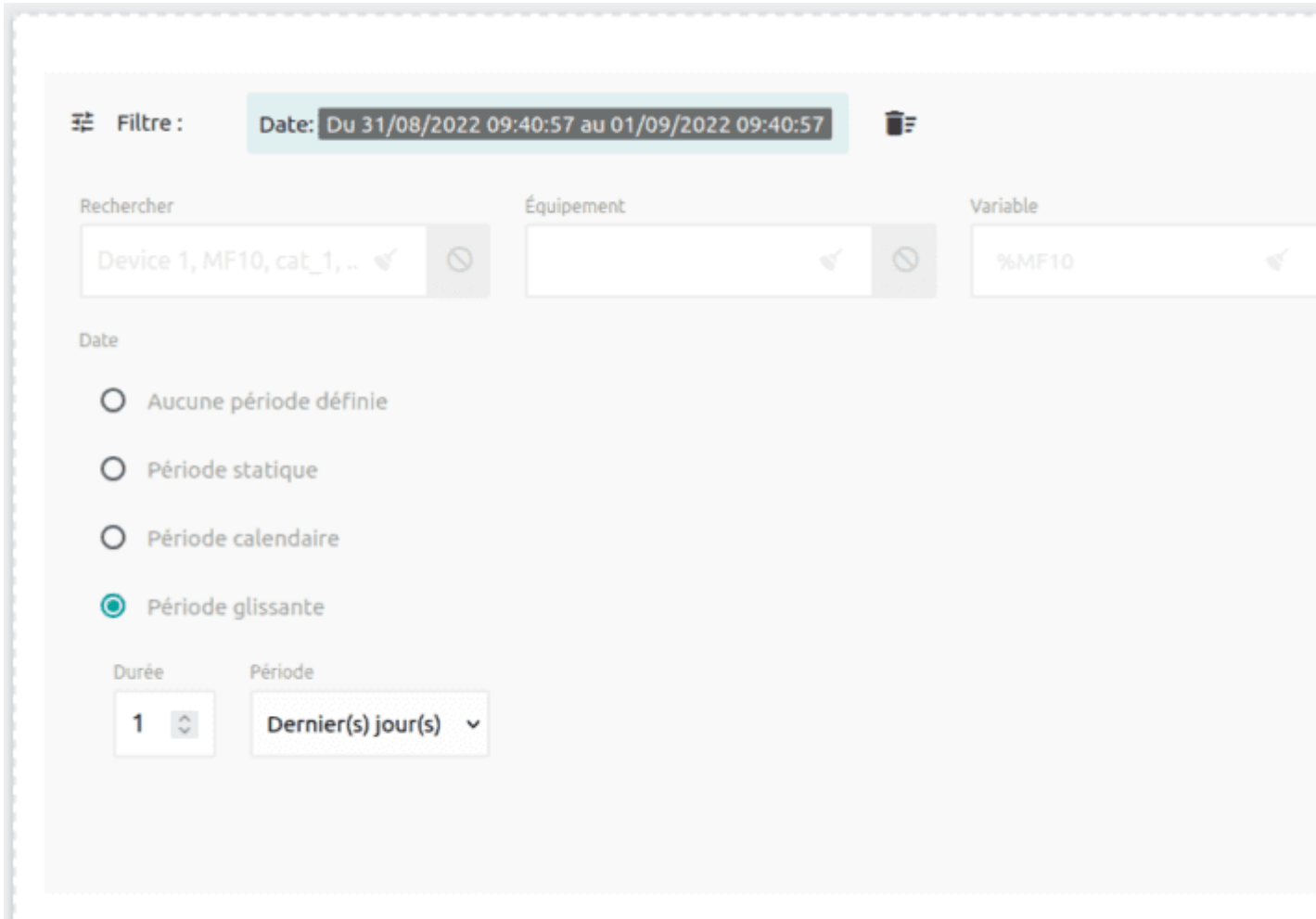
Une fois le widget créé, si on a coché « Activer le filtre depuis le widget » alors on peut directement sur le widget filtrer les alarmes en cliquant sur « Filtre » depuis le mode visualisation. On peut filtrer les alarmes sur les colonnes Équipement, Variable et Catégorie.

Si on recherche une ou des alarme(s) précise(s) alors on peut écrire un de leur mot-clé dans le champ « Rechercher » pour que seules les alarmes concernées par ce mot-clé s'affichent.

La période peut aussi être modifié dans la partie « Date ». Les mêmes champs que précédemment sont disponibles.

Une fois le filtrage réalisé, cliquez sur « Appliquer ».

Si on veut ré-initialiser les champs (pour qu'ils soit tous vierges à nouveau), cliquez sur le bouton « Ré-initialisé les champs ».



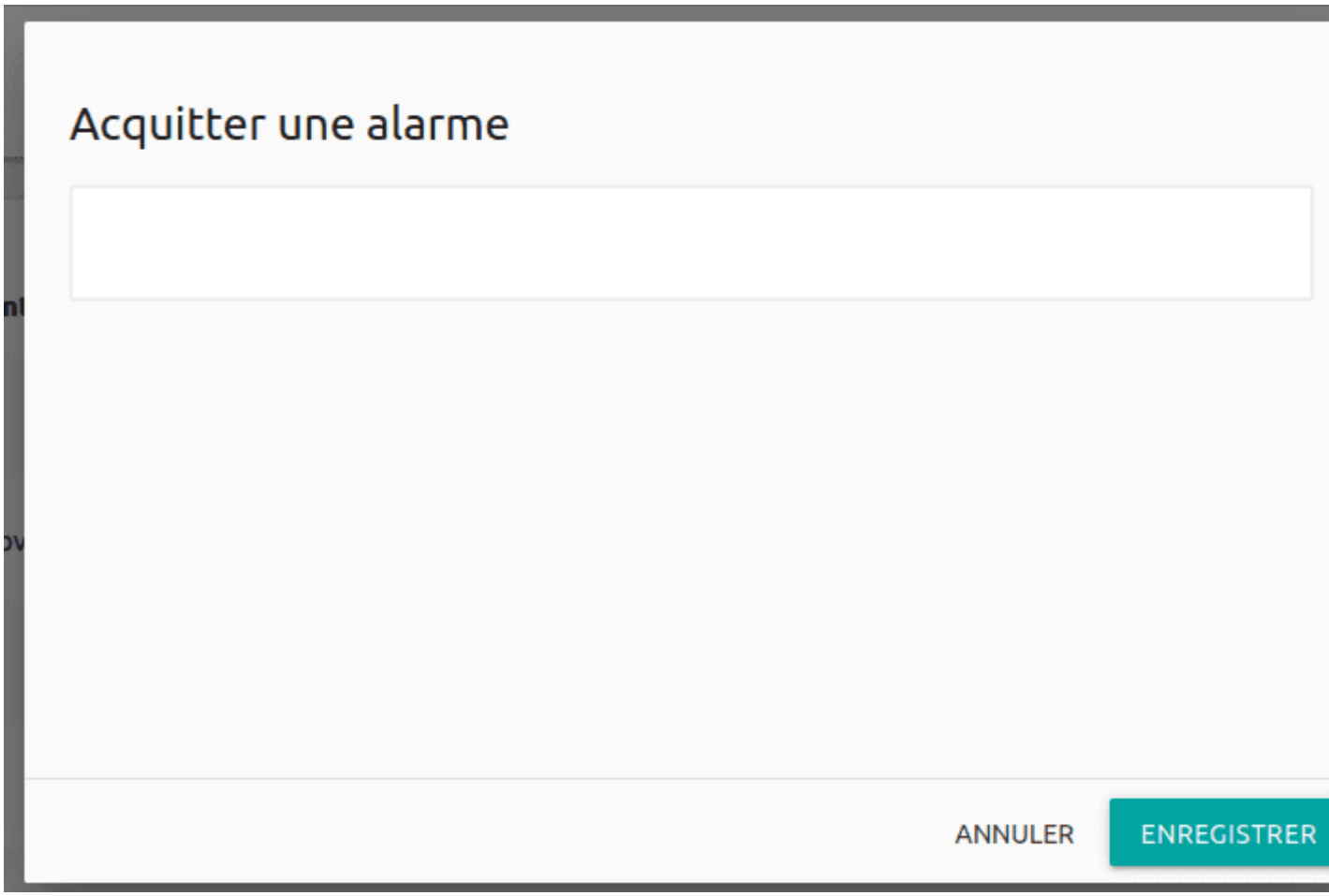
Capture d'écran du filtrage des événements depuis le widget Alarme

Acquitter une alarme

Acquitter une alarme signifie que l'on a bien remarqué qu'il y avait une alarme. En aucun cas l'alarme va s'éteindre à la suite d'un acquittement. Il faut régler le problème que soulève l'alarme pour qu'elle s'éteigne. L'acquittement est un pense-bête.

Ainsi, quand une alarme se lève, on a la possibilité de l'acquitter en cliquant sur l'icône:

Un message Acquitter apparaît, cliquez y dessus. Une fenêtre *popup* s'ouvre. On peut y écrire un message (par exemple, « Problème à régler d'ici 2 jours »). Cliquez ensuite sur « Enregistrer ».
Si on s'est trompé et que l'on ne veut pas acquitter l'alarme, il faut cliquer sur « Annuler ».



Capture d'écran de la fenêtre *popup* pour acquitter une alarme dans le widget Alarme

Tri des colonnes

Date

On peut changer le tri des colonnes, en cliquant sur les flèches haut et bas à coté de la colonne sur laquelle on veut que le tri s'effectue. Par exemple, sur la capture d'écran, on observe les flèches haut et bas à droite de la colonne nommée « Date ».

2.5.8 Widget Événements

Ce widget sert à afficher la liste des événements qui se sont levés sur une certaine période.

Onglet « Apparence »

Cet onglet permet de configurer l'aspect général du widget et son en-tête.

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

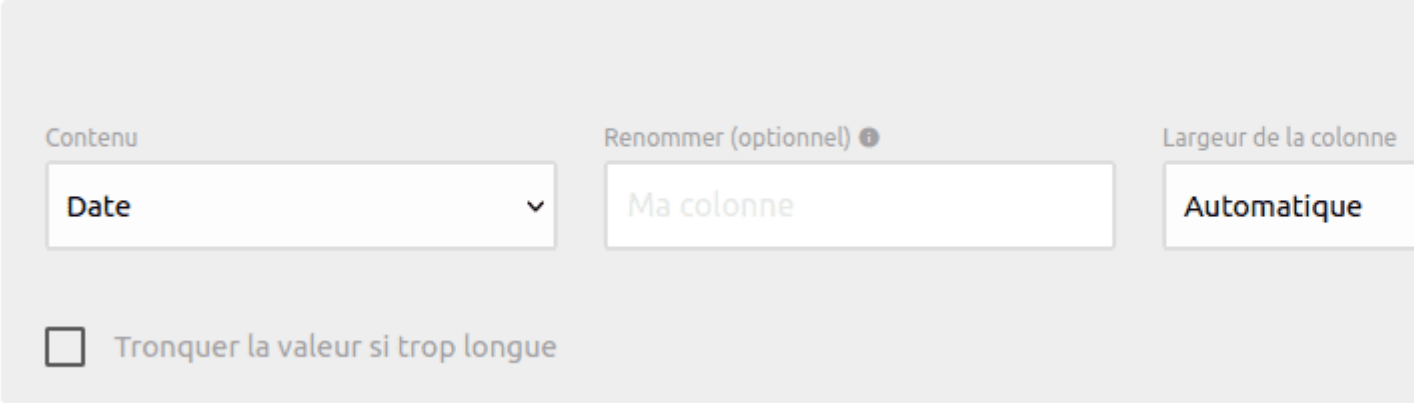
- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

On peut ajouter un nom au widget et un commentaire en remplissant les deux formulaires optionnels.

En cochant, « Activer le filtre depuis le widget » on débloque les fonctionnalités de filtrage des événements sur le widget depuis le mode visualisation.

Onglet « Données »

La liste des événements va être présentée sous forme d'un tableau modulable.
Un tableau possède plusieurs colonnes.
Une colonne est représentée dans un encart gris suivant l'exemple:



Capture d'écran d'un exemple de colonne dans l'onglet « Données » de la fenêtre *popup* de création/modification du widget Événement

Le nom de la colonne est dans le champ « Contenu ». On peut sélectionner une autre colonne dans son menu déroulant.

```
Avertissement
Attention, si on choisi un nouvelle colonne alors il se peut déjà qu'elle soit
```

Le tableau par défaut sera composé de 5 colonnes dont les noms par ordre sont:

- Date
- Équipement
- Variable
- Libellé
- Action

Une sixième colonne (Catégorie de la variable) est disponible dans le menu déroulant du champ « Contenu ».

En remplissant, la case « Renommer », on renomme le nom de la colonne. Cette action est optionnelle et par défaut le nom de la colonne sera celui inscrit dans le champ « Contenu ».

On a aussi la possibilité de choisir la taille de la largeur de la colonne. Par défaut la taille de la colonne sera ajustée automatiquement mais on peut la changer en cliquant sur « Largeur de la Colonne ». Elle pourra être ainsi:

- petite (10%)
- moyenne (25%)
- grande (50%)

En cochant, « Tronquer la valeur si trop longue » alors le tableau va afficher des variables tronquées dont le nombre de chiffre significatif dépendra de la taille du widget.

Pour supprimer la colonne, cliquez sur la croix rouge en haut à droite.

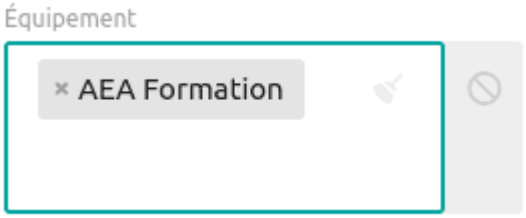
Pour changer l'ordre des colonnes, cliquez sur l'icône à quatre flèches en haut à droite puis faites glisser la colonne vers l'endroit souhaité.

Pour ajouter une colonne, cliquez sur le bouton
+ COLONNE. Une nouveau encart gris à compléter apparaît.

Onglet « Filtre »

Filtre par défaut

Si on souhaite visualiser des éléments précis, on peut filtrer en sélectionnant le ou les attributs de filtrage d'une colonne. Il est tout à fait possible de filtrer sur plusieurs colonnes.



On peut supprimer un attribut en cliquant sur la croix à sa gauche (sur l'attribut même)(par exemple pour supprimer l'attribut AEA Formation, cliquez sur la croix à sa gauche).

Le filtre sur une colonne en particulier peut-être mis en place en non. Pour exclure un filtre sur une colonne, cliquez sur le bouton suivant:

La case de la colonne est alors entourée en rouge

Pour vider un filtre de tous ses attributs, cliquez sur l'icône du balai à droite dans la champ filtré.

Pour le type de contenu calendrier, on peut filtrer sur les colonnes:

- Équipement
- Variable
- Catégorie.

En plus de réaliser un filtrage sur les colonnes, on peut filtrer les évènement sur une période précise dans la partie « Date ».

Quatre choix de période s'offre à nous:

- « Aucune période définie »: toutes les alarmes vont être affichées.
- « Période statique »: on doit choisir la date de début et de fin délimitant la période. En cliquant sur le champ « Date de début » ou « Date de fin », on la possibilité de rentrer la date en l'écrivant au format jj-mm-aaaa hh:mm (jour/mois/année heure:minute) ou bien en la sélectionnant grâce au sélectionneur de date qui s'affiche.

Capture d'écran de la partie « Période statique » de l'onglet « Filtre » de la fenêtre *popup* du widget Événement

- « Période calendaire »: définit un intervalle de temps d'affichage par rapport à la date actuelle ou à la date précédente. On peut choisir d'afficher les événements de l'heure, du jour, de la semaine,

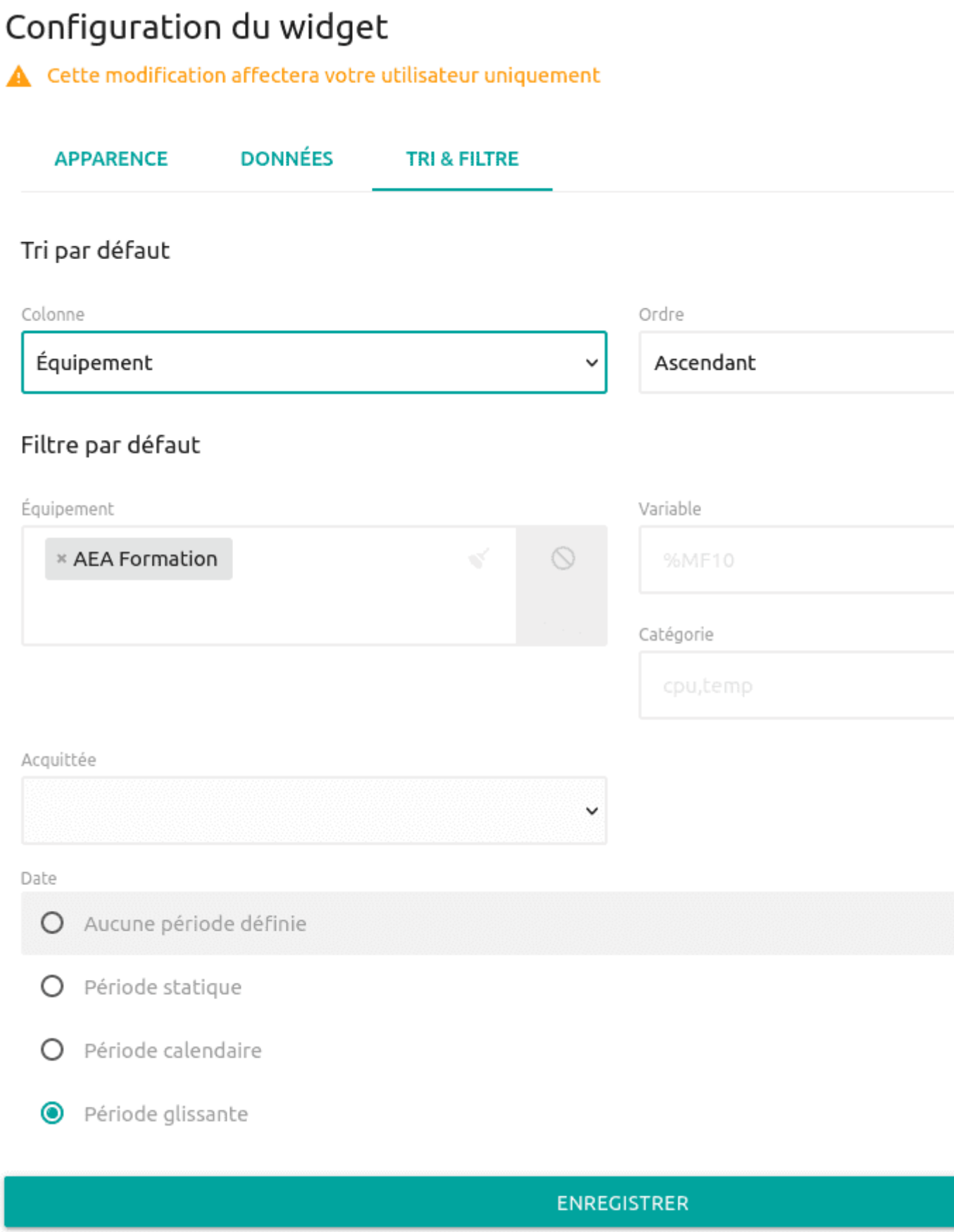
du mois ou de l’année précédente ou actuelle en sélectionnant dans le menu déroulant la période souhaitée ainsi que le décalage. Par défaut la période est celle de la journée actuelle.

Capture d’écran de la partie « Période calendaire » de l’onglet « Filtre » de la fenêtre *popup* du widget Événement

- « Période glissante »: définit un intervalle de temps des dernier(e)(s) heure(s)/ jour(s)/ semaine(s)/ mois jusqu’à la date actuelle. Pour cela, il suffit de renseigner une durée et un type de période dans le menu déroulant. Par défaut, la période s’étend de hier à aujourd’hui.

Capture d’écran de la partie « Période glissante » de l’onglet « Filtre » de la fenêtre *popup* du widget Événement

```
Information
Il est à noter que les événements sont triés par ordre chronologique.
```


Configuration du widget
Cette modification affectera votre utilisateur uniquement
APPARENCE
DONNÉES
TRI & FILTRE
Tri par défaut
Colonne
Équipement
Ordre
Ascendant
Filtre par défaut
Équipement
AEA Formation
Variable
%MF10
Catégorie
cpu,temp
Acquittée
Date
Aucune période définie
Période statique
Période calendaire
Période glissante
ENREGISTRER

Capture d’écran de l’onglet « Filtre » de la fenêtre *popup* de création/modification du widget Événement

Navigation sur le widget

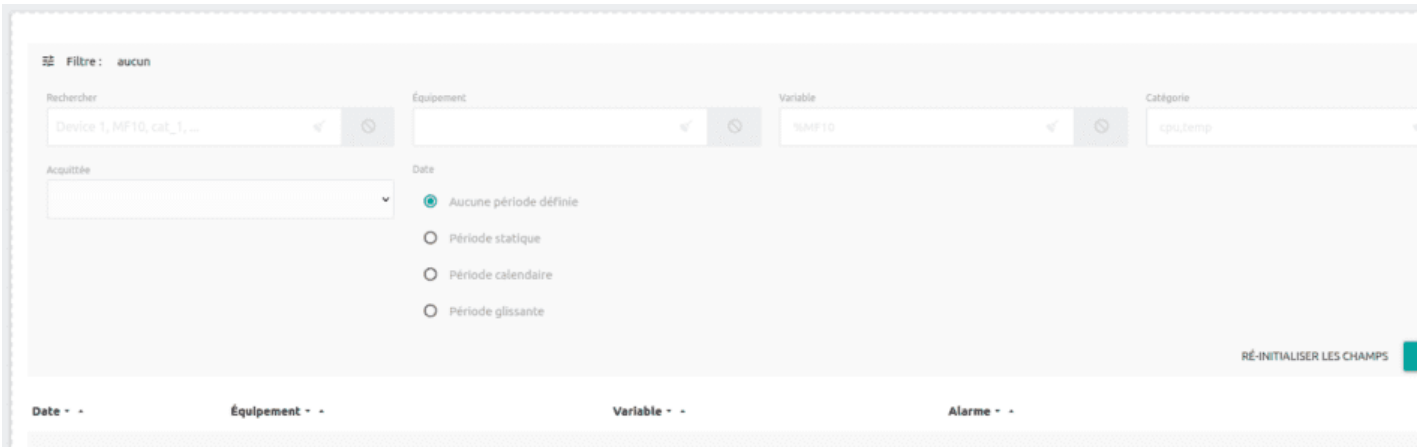
Une fois le widget créé, si on a coché « Activer le filtre depuis le widget » alors on peut directement sur le widget filtrer les événement en cliquant sur « Filtre ». On peut filtrer sur les colonnes Équipement, Variable et/ou Catégorie.

Si on recherche un ou des événement(s) précis alors on peut écrire un de leur mot-clé dans le champ « Rechercher » pour que seuls les évènements concernés par ce mot-clé s’affichent.

La période peut aussi être modifiée dans la partie « Date ». Les mêmes champs que précédemment sont disponibles.

Une fois le filtrage réalisé, cliquez sur « Appliquer ».

Si on veut ré-initialiser les champs (pour qu’ils soient tous vierges à nouveau), cliquez sur « Ré-initialisé les champs ».



Capture d’écran du filtrage des événements depuis le widget Évènement

2.6 Catégorie de widgets : Pilotage

2.6.1 Widget Bouton

Ce widget permet de créer un bouton qui tant qu'il est actif va modifier la valeur d'une variable.

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

Un bouton est lié à une variable. Choisissez ainsi l'équipement dans le menu déroulant des « Équipements » et le nom de la variable souhaitée dans le menu déroulant « Variable ».

Quand le bouton sera activé, la variable va prendre une nouvelle valeur. Cette nouvelle valeur est à écrire dans le champ du même nom. Seules les variables autorisées à être forcées sont sélectionnables.

On a la possibilité d'ajouter un nom au bouton. Ce nouvel intitulé est à écrire dans le champ « Libellé » mais cette démarche est optionnelle. Ce nouveau nom se placera à droite sur le bouton.

A sa gauche, le bouton est marqué d'une icône. Par défaut, l'icône affiché sera celle d'un fichier. On peut modifier cette icône, en en sélectionnant une autre en cliquant sur le champ « Icône ». Cette démarche est optionnelle.

La couleur de fond du bouton par défaut est bleu (#01a49e). On peut la changer facilement en cliquant sur le champ « Couleur du fond ». Un sélectionneur de couleur apparaît. Cette démarche est optionnelle.

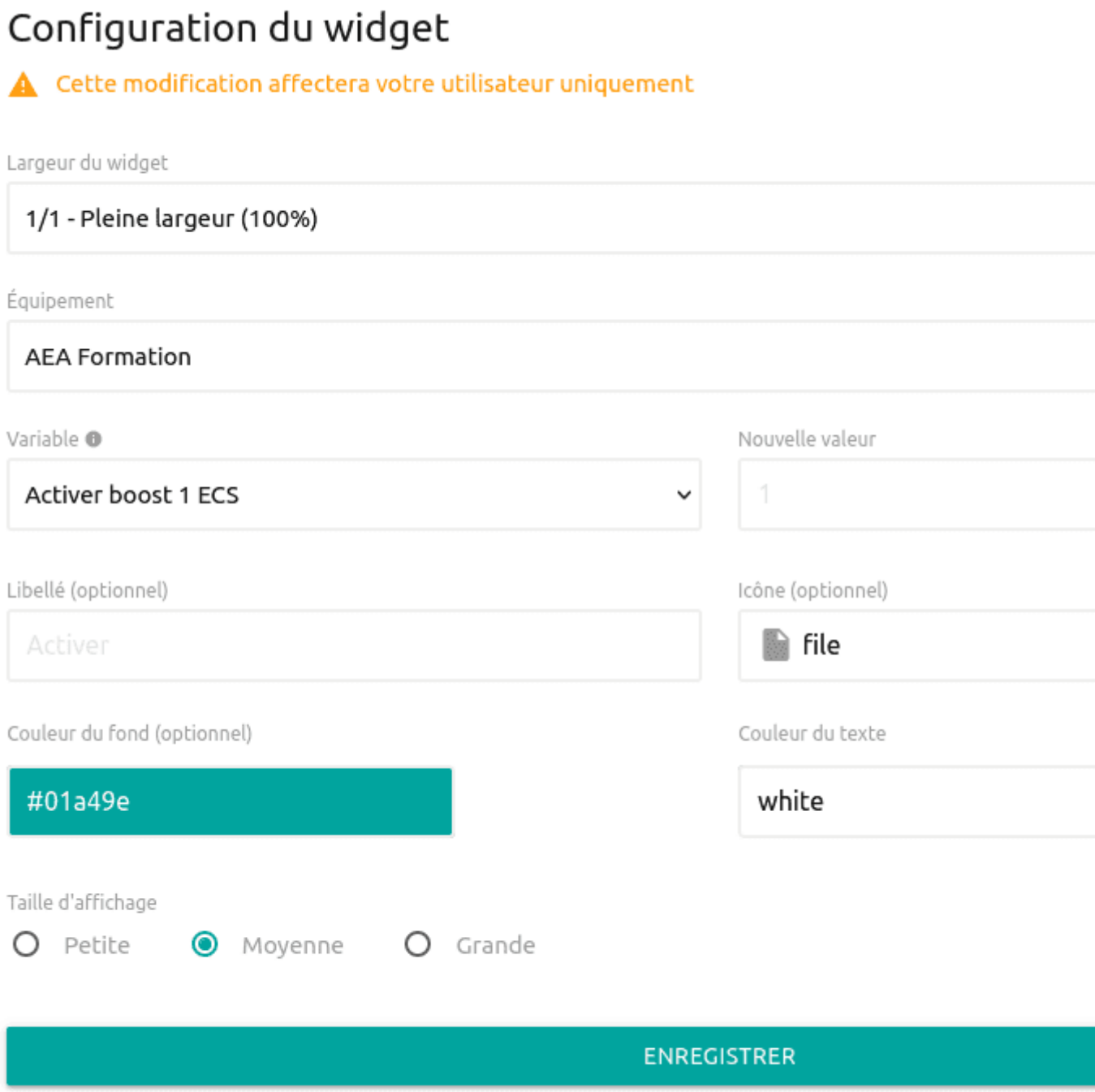
On peut aussi changer la couleur du texte en cliquant sur le champ « Couleur du texte ». Un sélectionneur de couleur apparaît alors permettant de faire notre choix parmi un grand nombre de couleurs.

```
Information
Il est à noter que la couleur en hexadécimal peut-être écrite dans les cases d
```

La taille de l’affichage du bouton est réglable:

- le bouton sera petit si vous cochez « Petite »
- le bouton sera de taille normale si vous cochez « Moyenne »
- le bouton sera de grande taille si vous cochez « Grande »

Par défaut la taille est cochée à moyenne.



Capture d'écran de la fenêtre *popup* de création/modification du widget Bouton

2.6.2 Widget Interrupteur

Ce widget permet de créer un interrupteur qui lorsqu'il est allumé (état ON) modifie la valeur d'une variable. Lorsque le bouton est éteint, la variable aura la valeur de son état OFF.

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

Un interrupteur est lié à une variable. Choisissez ainsi l'équipement dans le menu déroulant des « Équipements » et le nom de la variable souhaitée dans le menu déroulant « Variable ».

Quand l'interrupteur est éteint (état OFF), la variable prendra une valeur par défaut. Cette nouvelle valeur est à écrire dans le champ « Valeur de l'état OFF ». Seules les variables autorisées à être forcées sont sélectionnables.
On a la possibilité de donner un intitulé à cet état. Cet intitulé est à écrire dans le champs « Libellé de l'état OFF » mais cette démarche est optionnelle.
De plus, pour être plus visible, il est possible de colorer le bouton lorsqu'il est dans cet état en choisissant une couleur dans le sélectionneur de couleurs du champs « Couleur de l'état OFF ». La couleur de fond du bouton par défaut est gris (#f1f1f1). Cette démarche est optionnelle aussi.

Quand l'interrupteur est allumé (état ON), la variable prendra une valeur par défaut. Cette nouvelle valeur est à inscrire dans le champ « Valeur de l'état ON ». Seules les variables autorisées à être forcées sont sélectionnables.
On a la possibilité de donner un intitulé à cet état. Cet intitulé est à écrire dans le champs « Libellé de l'état ON » mais cette démarche est optionnelle.
De plus, pour être plus visible, il est possible de colorer le bouton lorsqu'il est dans cet état en choisissant une couleur dans le sélectionneur de couleurs du champs « Couleur de l'état ON ». La

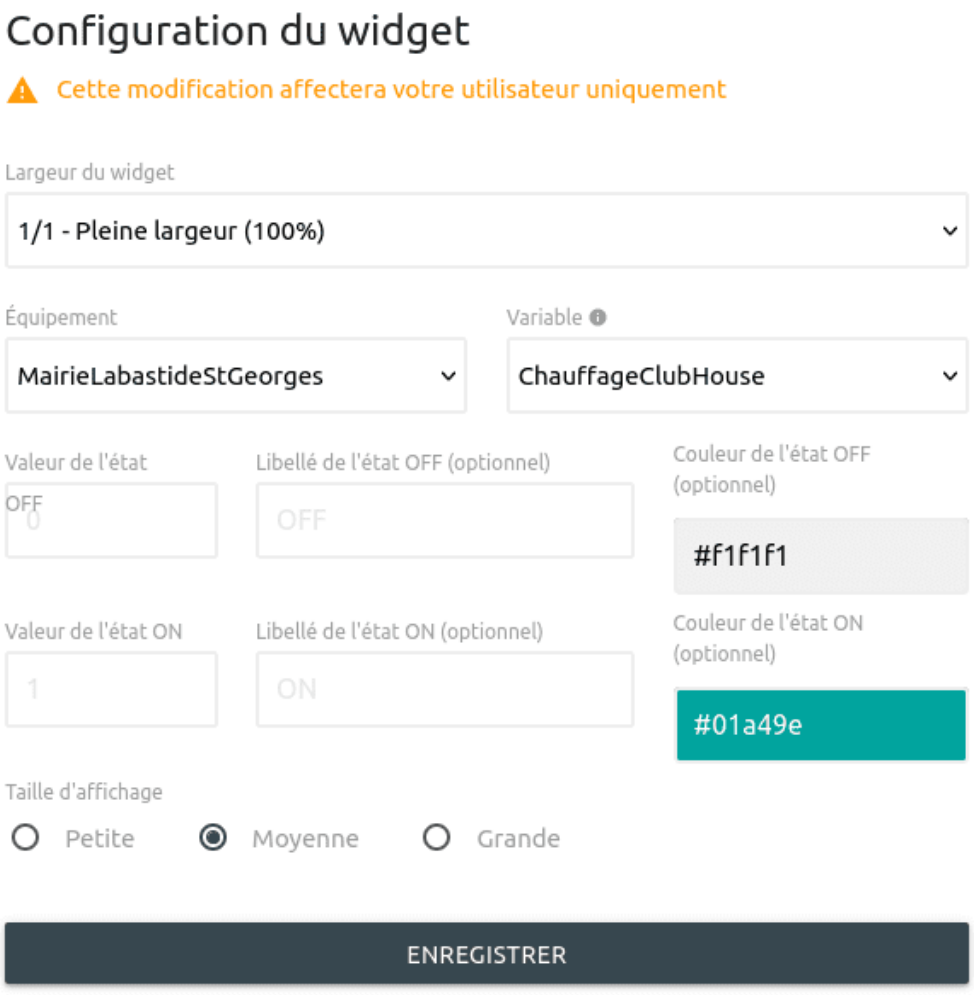
couleur de fond du bouton par défaut est bleu (#01a49e). Cette démarche est optionnelle aussi.

```
Information
Il est à noter que la couleur en hexadécimal peut-être écrite dans les cases d
```

La taille de l'affichage du bouton est réglable:

- le bouton sera petit si vous cochez « Petite »
- le bouton sera de taille normale si vous cochez « Moyenne »
- le bouton sera de grande taille si vous cochez « Grande »

Par défaut la taille est cochée à moyenne.



Capture d'écran de la fenêtre *popup* de création/modification du widget Interrupteur

2.6.3 Widget Calendrier

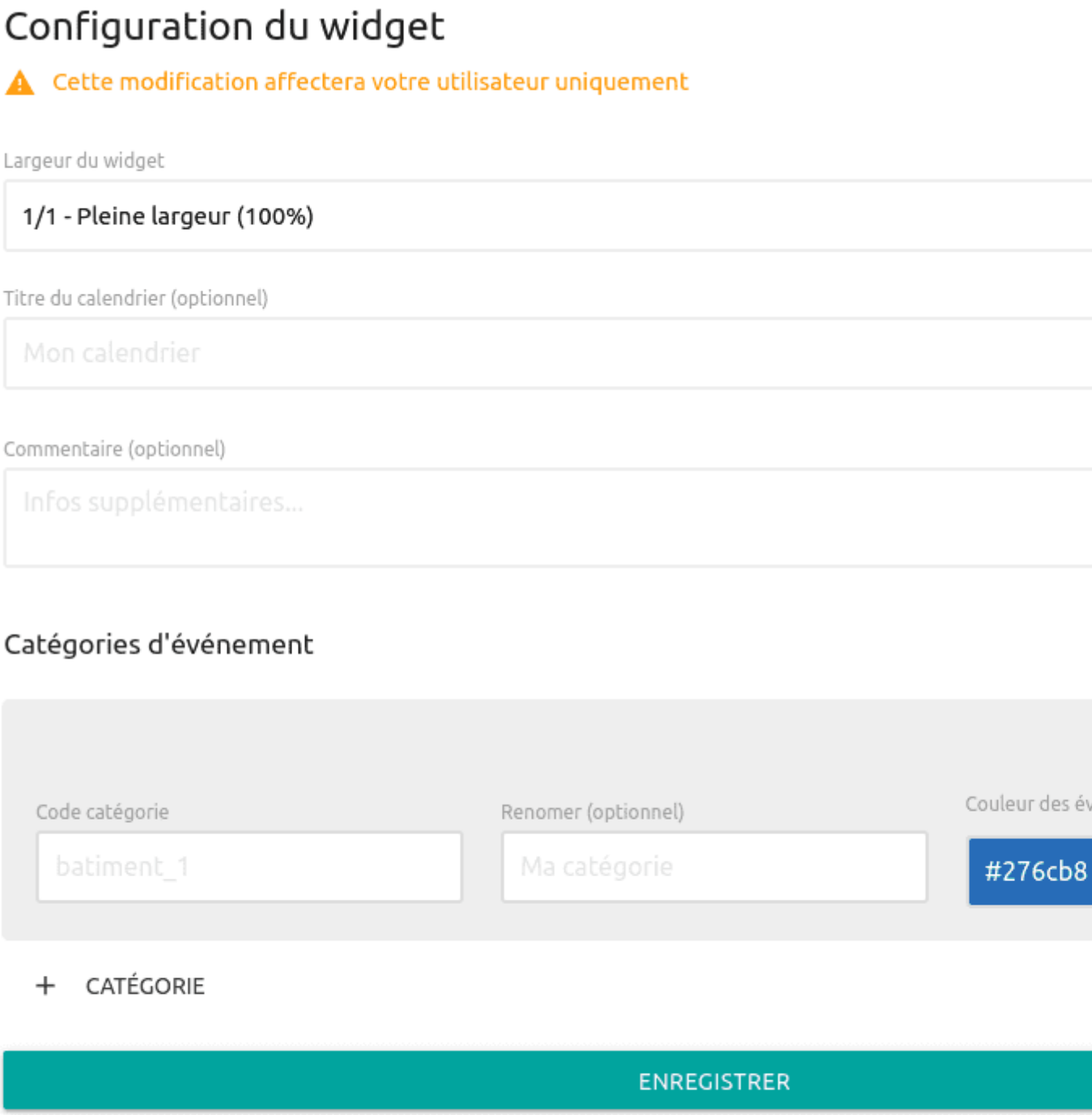
Ce widget sert à afficher un calendrier sur une journée, sur une semaine ou sur un mois.
A l'aide du calendrier, on va pouvoir déclarer des événements.

La largeur du widget est par défaut à 1/1 – Pleine largeur (100 %). Il est possible de la diminuer en sélectionnant une autre largeur parmi la liste suivante:

- ¼ – Quart de largeur (25%)
- ? – Tiers de largeur (33%)
- ½ – Demi largeur (50%)
- ? – Tiers de largeur (67%)
- ¾ – Trois quarts de largeur (75%)

On peut ajouter un Titre au calendrier et un commentaire en remplissant les deux formulaires optionnels.

Un événement appartient à une catégorie. Pour créer une catégorie sur le calendrier, il suffit de renseigner le code de la catégorie voulu dans « Code_catégorie » dans la partie « Catégories d'évènements ». Chaque catégorie sera dans un cadre gris.
On peut renommer ce code par un autre nom dans la case « Renommer(optionnel) ». Tous les événements appartenant à cette catégorie seront représentées par une couleur. Cette dernière est modifiable en cliquant sur le champ « Couleur des événements ». Par défaut elle est bleue (#276cb8).
Si on veut rajouter une nouvelle catégorie, cliquez sur + CATEGORIE. Un nouveau cadre gris apparaît. Il est à remplir comme le précédent.



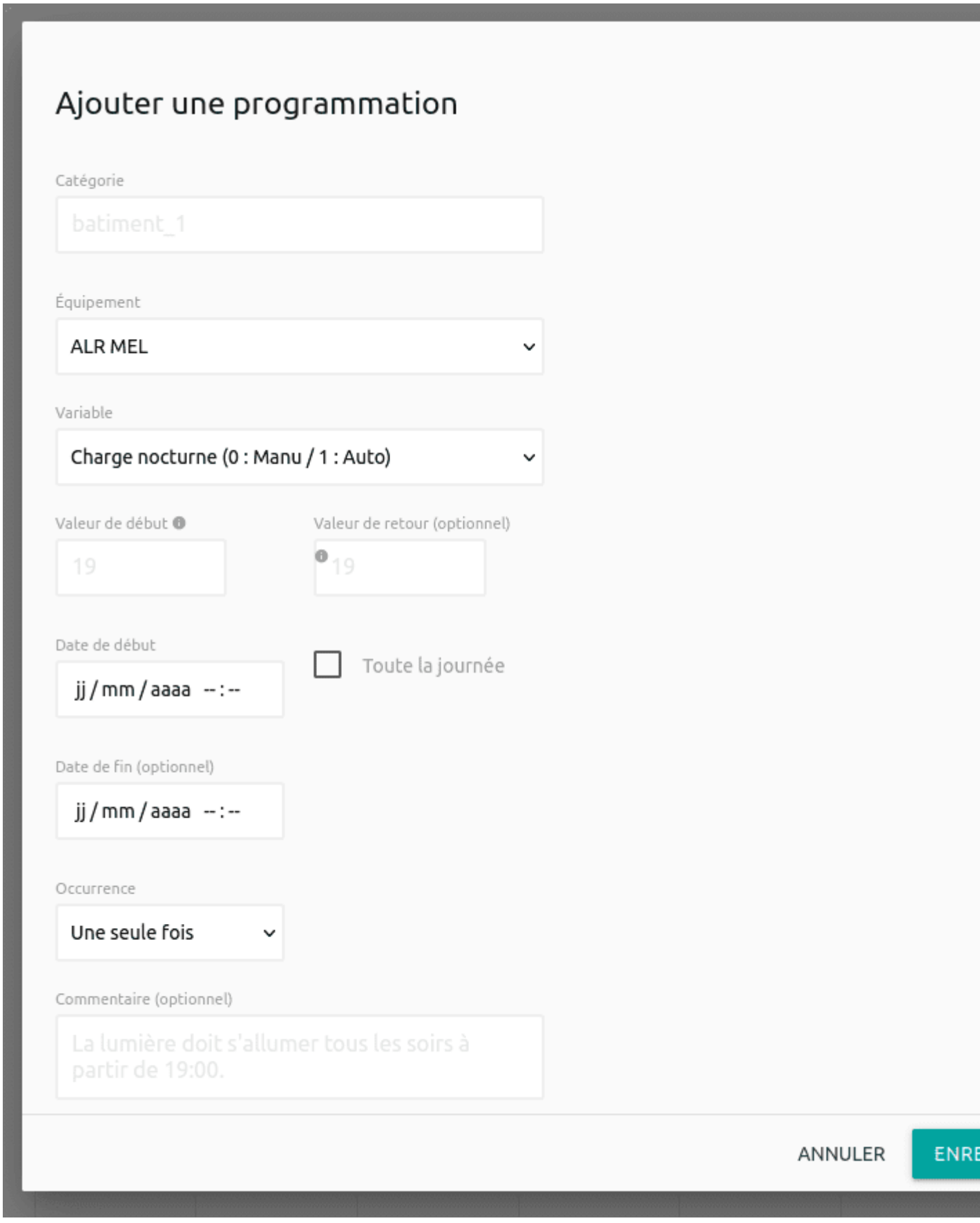
Capture d'écran de l'onglet « Apparence » de la fenêtre *popup* de création/modification du widget Calendrier

Navigation sur le widget

En mode visualisation, plusieurs fonctionnalités s’offrent à nous.

Programmation d’un événement

Pour programmer un nouvel événement, cliquez sur le bouton « Programmer » en haut à droite. La fenêtre *popup* suivante s’ouvre:


Ajouter une programmation
Catégorie
batiment_1
Équipement
ALR MEL
Variable
Charge nocturne (0 : Manu / 1 : Auto)
Valeur de début
19
Valeur de retour (optionnel)
19
Date de début
jj / mm / aaaa --:--
Toute la journée
Date de fin (optionnel)
jj / mm / aaaa --:--
Occurrence
Une seule fois
Commentaire (optionnel)
La lumière doit s'allumer tous les soirs à partir de 19:00.
ANNULER
ENRE

Capture d'écran de la fenêtre *popup* de programmation d'un événement sur le widget Calendrier

L'événement que l'on va créer peut appartenir à une catégorie. Remplissez alors le champ « Catégorie » avec le nom de la catégorie voulu ou bien sélectionnez la catégorie parmi celles proposées. Tous les événements de la même catégorie auront ainsi la même couleur. Il faut savoir que si on ne remplit pas cette case alors l'événement sera classé dans aucune catégorie mais sera colorié tout de même par une couleur. Tous les événements n'ayant pas de catégorie auront la même couleur.

Un événement est levé sur une variable. Choisissez donc l'équipement dans le menu déroulant des « Équipements » et le nom de la variable souhaitée dans le menu déroulant « Variable ».

Lorsqu'un événement se produit, la variable va avoir une nouvelle valeur. Par exemple, 1 pour signifier l'allumage de l'éclairage. Cette valeur sera écrite dans le champ Valeur de début. Il est à noter que si la valeur est une chaîne de caractères alors elle est à mettre entre guillemets (“….”).

Lorsque l'événement est terminé, on peut choisir d'affecter une nouvelle valeur à la variable. Cette valeur sera renseignée dans le champ « Valeur de retour ». Elle est optionnelle et si on laisse cette case vide alors la valeur de la variable sera la même que celle avant l'événement.

Un événement sera levé pendant une période de temps définie ayant une date de début et une date de fin. La date de début est à renseignée dans le champ « Date de début » sous la format jj/mm/aaaa hh:mm (jour/mois/année heure:minute). En cliquant sur ce champ, on a la possibilité de choisir une date grâce à un sélectionneur de date.
En cochant la case « Toute la journée », l'événement sera levé la journée entière de la date de début rentrée.
Pour que l'événement se termine, rentrez une date de fin dans le champ « Date de fin » sous le format jj/mm/aaaa hh:mm (jour/mois/année heure:minute). En cliquant sur ce champ, on a la possibilité de choisir une date grâce à un sélectionneur de date. Le remplissage de cette case est optionnel. Si on n'indique aucune valeur de date de fin alors l'événement sera levé de la date de début jusqu'à une durée indéterminée.

Par défaut, l'événement apparaîtra une seule fois. Mais si vous souhaitez rendre cet événement récurrent, dans le champ « Occurrence » sélectionnez une nouvelle occurrence parmi:

- chaque jour
- chaque semaine
- chaque mois
- chaque année

Enfin, l'événement peut-être estampillé d'un commentaire écrit dans le champ « Commentaire ».

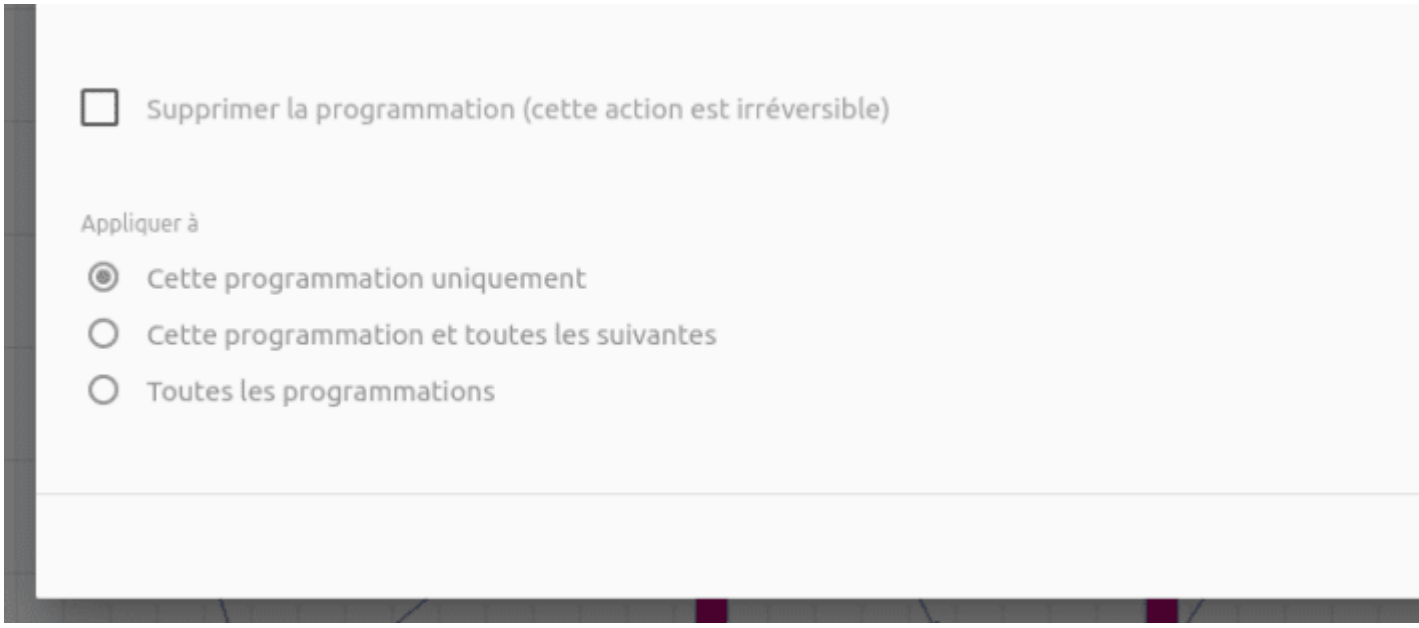
Enfin, pour créer la programmation de l'événement cliquer sur le bouton « Enregistrer »

Modification de la programmation d'un événement

Une autre fonctionnalité du calendrier est la modification d'événement. Pour modifier la programmation d'un événement, cliquez sur l'événement voulu sur le calendrier. Une fenêtre *popup* nommé « Modifier une programmation » s'ouvre. Les mêmes champs que pour la création d'un événement sont affichés. Pour modifier l'événement, il suffit de changer les champs voulus (vous référez à la partie « Programmation d'un événement » pour les informations de ces champs).

Dans cette fenêtre, il est possible de supprimer l'événement, en cochant « Supprimer la programmation ». Cette action est irréversible. Une fois cochée, on peut indiquer quelle programmation on souhaite supprimer:

- soit uniquement l'événement que l'on a sélectionné
- soit cet événement et les suivants c'est à dire que si l'événement est un événement récurrent alors les suivants ainsi que celui ci vont être supprimés
- soit tous les événements passés et futurs ainsi que celui-ci vont être supprimés



Capture d'écran de la partie suppression d'un événement de la fenêtre *popup* de modification d'un événement du widget Calendrier

Cliquer à nouveau sur le bouton « Enregistrer » pour sauvegarder les modifications.

```
Information
Si l'événement modifié est un événement récurrent alors la modification sera a
```

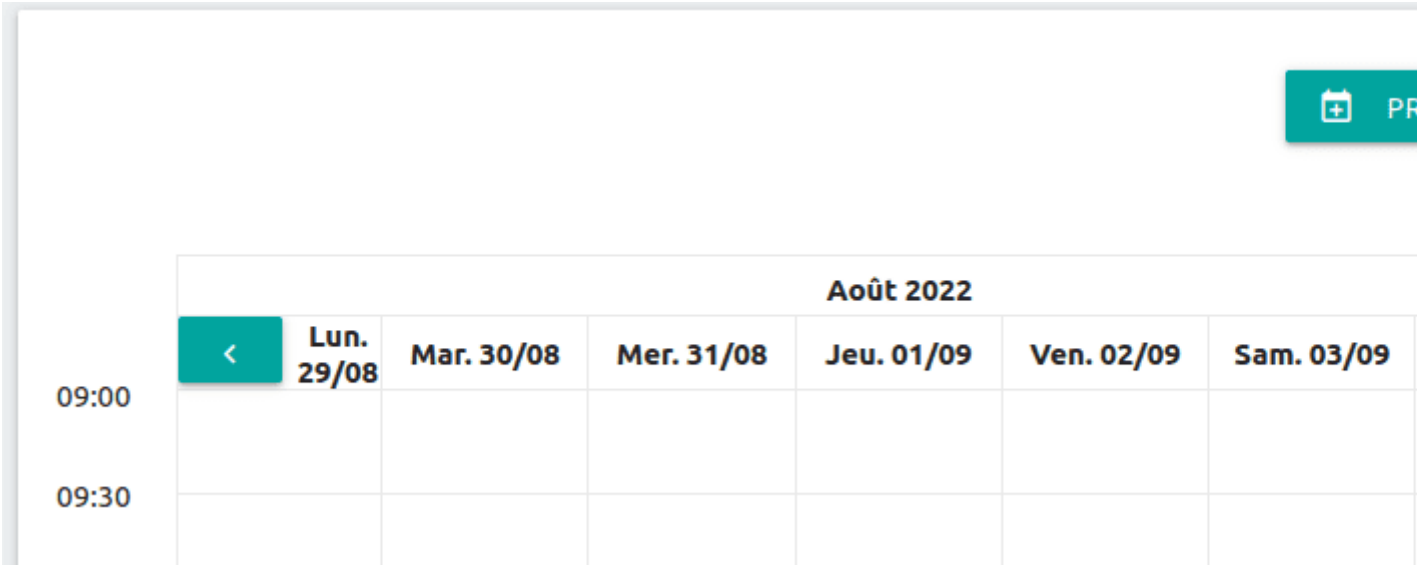
Filtrage de événements

Une autre fonctionnalité du calendrier est le filtrage des événements. En haut à droite, on peut faire apparaître/disparaître une catégorie d'événements en cliquant sur le nom de la catégorie. On peut faire apparaître/disparaître autant de catégories que l'on veut.

Visualisation jour, semaine ou mois

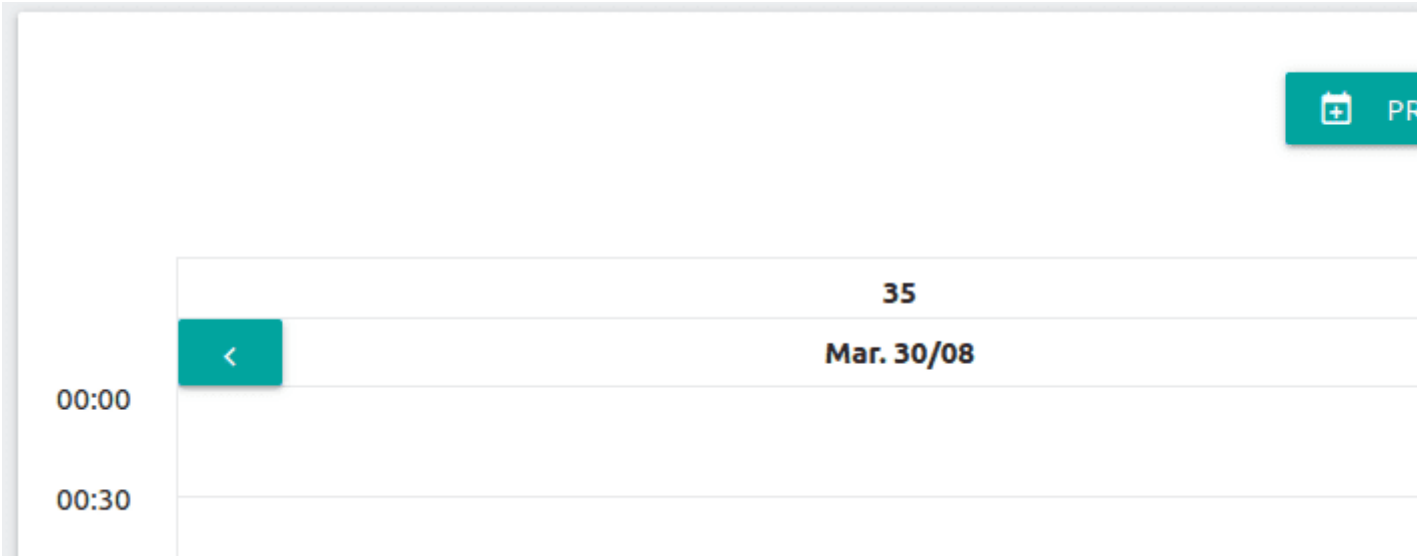
Trois types de visualisation des événements sont possibles.

La vue par défaut est la vue par semaine.



Capture d'écran d'une vue semaine sur le widget Calendrier

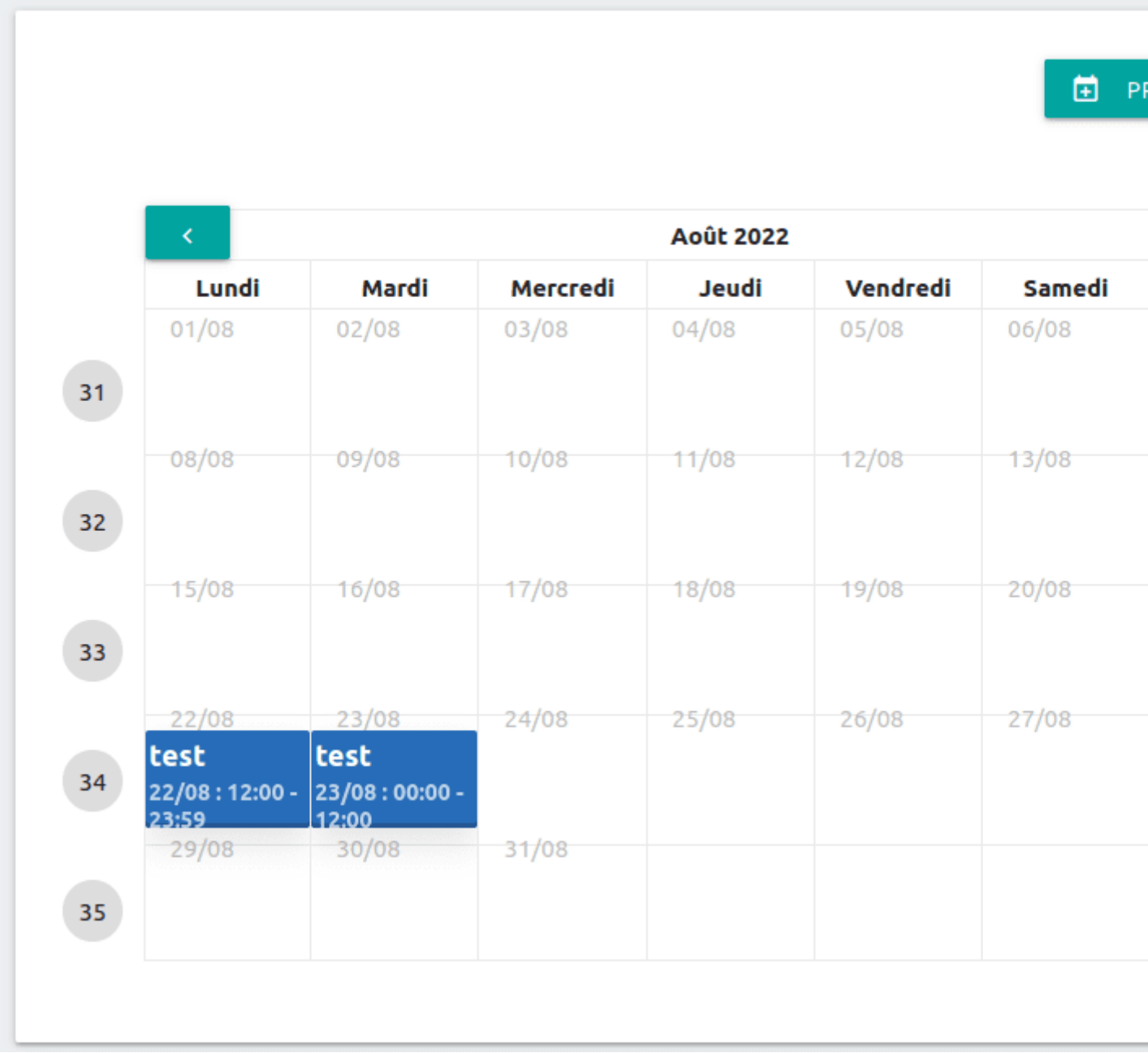
Pour visualiser plus précisément, les événements d'une journée, cliquez sur le date voulue (par exemple, Mar. 30/08).



Capture d'écran d'une vue jour sur le widget Calendrier

Pour repasser, sur une visualisation à la semaine depuis une visualisation à la journée, cliquez sur le numéro de la semaine au dessus de la date (par exemple ici 35).

Pour visualiser les événements sur un mois entier, il faut d'abord avoir une vue par semaine, puis cliquez sur le mois au dessus des dates(par exemple sur notre capture d'écran Août 2022).



Capture d'écran d'une vue mois sur le widget Calendrier

Pour revenir sur une vue semaine, il suffit de cliquer sur le numéro de la semaine que l'on souhaite sur la gauche (sur la capture d'écran ci-dessus, les numéro de semaines disponibles sont 31, 32, 33, 34 et 35).

On peut passer d'une journée à une autre, d'une semaine à une autre ou d'un mois à un autre en cliquant sur les flèches vertes en haut à gauche et à droite.

3. Présentation de l'API

3.1 Généralités

Ce document détaille les requêtes HTTP permettant à l'utilisateur d'interagir avec MicroSERVER au travers de fonctions, fournissant ainsi une interface de communication entre le serveur Web et d'autres services externes.

```
Information
La documentation n'informe que sur les tables "devices" et "variables" et sur
```

3.2 Fonctions

3.2.1 Headers

Tout d'abord, chaque requête aura un paramètre 'headers'. Afin de bien s'authentifier sur l'API, on a besoin d'une clé API (api_key). Cette clé sera par la suite utilisée dans la requête en l'appelant de cette manière:

```
headers={'Authorization': 'Bearer {api_key}'.format(api_key=API_KEY)}
```

3.2.2 Méthode GET

Grace à l'API de MicroSERVER, il est possible d'obtenir des informations sur les tables:

- les variables (variables)
- les équipements (devices)

Il est aussi possible d'obtenir des informations sur des données historisées:

- les événements (events)
- les alarmes (alarms)
- les valeurs historisées (plot-values)
- les compteurs historisés (diff-values)
- les données historiques brutes (log-entries)

```
Information
Les données sont remontées sous format JSON. Les exemples ont été réalisés en
```

3.2.2.1 Variables

Pour obtenir la liste des variables, la requête HTTP sera:

```
/variables
```

Cette requête peut recevoir plusieurs paramètres supplémentaires afin d'avoir des informations plus précises. Les paramètres qui ne prendront qu'une valeur seront identifiés par *param=VALUE* et les paramètres qui peuvent prendre une liste de valeurs seront identifiés par *param[]=VALUES*:

Paramètre	Description
limit=VALUE	Nombre d'éléments (VALUE) à renvoyer au maximum
sort=VALUE(,asc/desc) ou sort[]=VALUES(,asc/desc)	Tri sur VALUE de manière ascendante (,asc) ou descendante (,desc). Il est possible de trier sur plusieurs champs VALUES
field=VALUE ou field[] =VALUES	Filtrage sur le champs VALUE ou les champs VALUES voulus
search=VALUE	Renvoie uniquement les éléments dont n'importe quel champ contient le texte VALUE
search~x=VALUE	Renvoie uniquement les éléments dont n'importe quel champ ne contient pas le texte VALUE
cursor=VALUE	Valeur du champ 'cursor' renvoyé lors de la dernière requête avec le paramètre *limit=VALUE* défini afin d'obtenir la suite des éléments
champ=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur égale à la valeur VALUE spécifiée.
champ~x=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n'ont pas une valeur égale à la valeur VALUE spécifiée.
champ~s=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur contenant la valeur VALUE spécifiée.
champ~sx=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n'ont pas une valeur contenant la valeur VALUE spécifiée.
champ~i=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur se situant dans l'intervalle spécifié sous la forme '[' ou ']' suivi de {valeur-min}, puis ';', puis {valeur-max}, puis ']' ou '['.
champ~ix=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur ne se situant pas dans l'intervalle spécifié sous la forme '[' ou ']' suivi de {valeur-min}, puis ';', puis {valeur-max}, puis ']' ou '['.

Tableau sur les paramètres d'une requête GET variables

Pour les variables, la liste des champs possibles est:

Nom du champs	Description
id	Identifiant de la variable
deviceId	Identifiant de l'équipement
device	Nom de l'équipement
displayLabel	Valeur de le la représentation de la variable

address	Adresse de la variable
type	Type de la variable
intRegister	Adresse de la variable
priority	Priorité de la variable
mnemonic	Mnémonique de la variable
label	Libellé de la variable
comment	Commentaire
isInternal	Booléen indiquant si la variable est interne ou pas
isPublic	Booléen indiquant si la variable est publique ou pas
isWriteable	Booléen indiquant si la variable peut être écrite ou pas
minWriteValue	Valeur minimale que peut prendre la variable
maxWriteValue	Valeur maximale que peut prendre la variable
categories	Catégories dont fait partie la variable
format	Format de la variable
log	Booléen indiquant si le changement de valeur historise l'ancienne valeur
floatPrecision	Entier indiquant le nombre de chiffre significatif après la virgule. 0 si la variable n'est pas un décimale.
unit	Unité de la variable. Si aucune unité alors » est retourné.
minRemoteScaleValue	Valeur minimale brute
maxRemoteScaleValue	Valeur maximale brute
minLocalScaleValue	Valeur minimale lors de la mise à l'échelle
maxLocalScaleValue	Valeur maximale lors de la mise à l'échelle
inhibitionUser	Utilisateur ayant inhibé l'alarme
inhibitionStartTime	Date de début de l'inhibition
inhibitionStopTime	Date de fin de l'inhibition
hasAlarm	Booléen autorisant la gestion des alarmes
alarmLabel	Libellé de l'alarme
alarmCondition	Condition de l'alarme
hasEvent	Booléen autorisant la gestion des événements
eventLabel	Libellé de l'événement
eventCondition	Condition de l'événement
hasPlot	Booléen autorisant l'historisation des valeurs
plotIsAveraged	Booléen autorisant de moyenner la courbe
plotMinDelay	Délai entre deux points en secondes sur la courbe
plotMinDiff	Hystérésis de la variable
snmpOID	Adresse SNMP de la variable
snmpType	Type SNMP de la variable
rawValue	Valeur actuelle de la variable

displayValue	Valeur formatée de la variable

Tableau sur les champs disponibles pour une variable

Par défaut, toutes les valeurs seront retournées avec tous les champs dont le tri sera par ordre chronologique de création de variables.

Un exemple de requête HTTP est:

```
https://192.168.1.223/variables?limit=2&field[]=id&field[]=mnemonic
```

Le code écrit en python sera le suivant:

```
query = 'variables'
query = {'limit': 2, 'field[]': ['id', 'mnemonic']}
req = requests.get(SERVER_URI + query, params=param, headers={
        'Authorization': 'Bearer {api_key}'.format(api_key=API_KEY)
    })
if req.status_code == 200:
    print('OK')
    data = req.json()
    print(data)
else:
    print('ERR:', req.json())
```

La réponse à cette requête est:

```
{'items': [{'id': '6', 'mnemonic': 'ALM_COM@1'}, {'id': '7', 'mnemonic': 'BUSY

Information
Dans la réponse à la requête on s'aperçoit qu'il y a une clé 'cursor' en plus
```

3.2.2.2 Équipements

Pour obtenir la liste des équipements, la requête HTTP sera:

```
/devices
```

Cette requête peut recevoir plusieurs paramètres supplémentaires afin d'avoir des informations plus précises. Les paramètres qui ne prendront qu'une valeur seront identifiés par *param=VALUE* et les paramètres qui peuvent prendre une liste de valeurs seront identifiés par *param[]=VALUES*:

Paramètre	**Description**
limit=VALUE	Nombre d'éléments (VALUE) à renvoyer au maximum
sort=VALUE(,asc/desc) ou sort[]=VALUES(,asc/desc)	Tri sur VALUE de manière ascendante (,asc) ou descendante (,desc). Il est possible de trier sur plusieurs champs VALUES
field=VALUE ou field[] =VALUES	Filtrage sur le champs VALUE ou les champs VALUES voulus

search=VALUE	Renvoie uniquement les éléments dont n'importe quel champ contient le texte VALUE
search~x=VALUE	Renvoie uniquement les éléments dont n'importe quel champ ne contient pas le texte VALUE
cursor=VALUE	Valeur du champ 'cursor' renvoyé lors de la dernière requête avec le paramètre *limit=VALUE* défini afin d'obtenir la suite des éléments
champ=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur égale à la valeur VALUE spécifiée.
champ~x=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n'ont pas une valeur égale à la valeur VALUE spécifiée.
champ~s=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur contenant la valeur VALUE spécifiée.
champ~sx=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n'ont pas une valeur contenant la valeur VALUE spécifiée.
champ~i=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur se situant dans l'intervalle spécifié sous la forme '[' ou ']' suivi de {valeur-min}, puis ';', puis {valeur-max}, puis ']' ou '['.
champ~ix=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur ne se situant pas dans l'intervalle spécifié sous la forme '[' ou ']' suivi de {valeur-min}, puis ';', puis {valeur-max}, puis ']' ou '['.

Tableau sur les paramètres d'une requête GET devices

Pour les équipements, la liste des champs possibles est:

Nom du champs	**Description**
id	Identifiant de l'équipement
isActive	Booléen indiquant si l'équipement est actif (1) ou en erreur (0)
type	Type de l'équipement
name	Nom de l'équipement
comment	Commentaire
displayLabel	Valeur de le la représentation de l'équipement
zones	Zone(s) de l'équipement
purgeDelay	Délai de purge des historiques (en jours)
timeout	Timeout (en secondes)
refreshDelay	Délai de rafraîchissement (en secondes)
comProtocol	Protocole de communication utilisé
comPort	Port de communication utilisé
slaveNumber	Numéro d'esclave. None pour aucun.
password	Mot de passe de l'équipement

comState	État de la communication
alarms	Nombre d'alarmes actives
isEditable	Booléen indiquant si l'équipement est modifiable par un utilisateur

Tableau sur les champs disponibles pour un équipement

Par défaut, toutes les valeurs seront retournées avec tous les champs dont le tri sera par ordre chronologique de création d'équipements.

Un exemple de requête HTTP est:

```
https://192.168.1.223/devices?limit=2&field[]=id&field[]=isActive&field[]=type
```

Le code écrit en python sera le suivant:

```
query = 'devices'
param = {'limit': 2, 'field[]': ['id', 'isActive', 'type']}
req = requests.get(SERVER_URI + query, params=param, headers={
        'Authorization': 'Bearer {api_key}'.format(api_key=API_KEY)
    })
if req.status_code == 200:
    print('OK')
    data = req.json()
    print(data)
else:
    print('ERR:', req.json())
```

La réponse à cette requête est:

```
{'items': [{'id': '1', 'isActive': True, 'type': 'AEH10000'}, {'id': '2', 'isA
```

3.2.2.3 Alarmes

Pour obtenir la liste des alarmes, la requête HTTP sera:

```
/alarms
```

Cette requête peut recevoir plusieurs paramètres supplémentaires afin d'avoir des informations plus précises. Les paramètres qui ne prendront qu'une valeur seront identifiés par *param=VALUE* et les paramètres qui peuvent prendre une liste de valeurs seront identifiés par *param[]=VALUES*:

Paramètre	**Description**
limit=VALUE	Nombre d'éléments (VALUE) à renvoyer au maximum
sort=VALUE(,asc/desc) ou sort[]=VALUES(,asc/desc)	Tri sur VALUE de manière ascendante (,asc) ou descendante (,desc). Il est possible de trier sur plusieurs champs VALUES
field=VALUE ou field[] =VALUES	Filtrage sur le champs VALUE ou les champs VALUES voulus

search=VALUE	Renvoie uniquement les éléments dont n'importe quel champ contient le texte VALUE
search~x=VALUE	Renvoie uniquement les éléments dont n'importe quel champ ne contient pas le texte VALUE
cursor=VALUE	Valeur du champ 'cursor' renvoyé lors de la dernière requête avec le paramètre *limit=VALUE* défini afin d'obtenir la suite des éléments
champ=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur égale à la valeur VALUE spécifiée.
champ~x=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n'ont pas une valeur égale à la valeur VALUE spécifiée.
champ~s=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur contenant la valeur VALUE spécifiée.
champ~sx=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n'ont pas une valeur contenant la valeur VALUE spécifiée.
champ~i=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur se situant dans l'intervalle spécifié sous la forme '[' ou ']' suivi de {valeur-min}, puis ';', puis {valeur-max}, puis ']' ou '['.
champ~ix=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur ne se situant pas dans l'intervalle spécifié sous la forme '[' ou ']' suivi de {valeur-min}, puis ';', puis {valeur-max}, puis ']' ou '['.

Tableau sur les paramètres d'une requête GET alarms

Nom du champs	**Description**
id	Identifiant de l'alarme
time	Timestamp auquel l'alarme s'est levée
variableId	Identifiant de la variable
variable	Nom de la variable
variableCategories	Catégories de la variable
deviceId	Identifiant de l'équipement
device	Nom de l'équipement
label	Libellé de l'alarme
isAck	Booléen indiquant si l'alarme a été acquittée ou non
ackTime	Timestamp auquel l'alarme a été acquittée
ackUser	Nom de l'utilisateur qui a acquitté l'alarme
ackComment	Commentaire laissé lors de l'acquittement de l'alarme

Tableau sur les champs disponibles pour une alarme

Par défaut, toutes les valeurs seront retournées avec tous les champs dont le tri sera par ordre chronologique d'apparition d'alarmes.

Un exemple de requête HTTP est:

```
https://192.168.1.223/alarms?limit=2&sort=time,desc&field[]=id&field[]=time
```

Le code écrit en python sera le suivant:

```
query = 'alarms'
param = {'sort': 'time,desc', 'limit': 2, 'isActive': True, 'field[]': ['id',
req = requests.get(SERVER_URI + query, params=param, headers={
        'Authorization': 'Bearer {api_key}'.format(api_key=API_KEY)
    })
if req.status_code == 200:
    print('OK')
    data = req.json()
    print(data)
else:
    print('ERR:', req.json())
```

La réponse à cette requête est:

```
{'items': [{'id': '270953256', 'time': 1660202056.43676}, {'id': '270953231',
```

3.2.2.4 Événements

Pour obtenir la liste des événements, la requête HTTP sera:

```
/events
```

Cette requête peut recevoir plusieurs paramètres supplémentaires afin d'avoir des informations plus précises. Les paramètres qui ne prendront qu'une valeur seront identifiés par *param=VALUE* et les paramètres qui peuvent prendre une liste de valeurs seront identifiés par *param[]=VALUES*:

Paramètre	Description
limit=VALUE	Nombre d'éléments (VALUE) à renvoyer au maximum
sort=VALUE(,asc/desc) ou sort[]=VALUES(,asc/desc)	Tri sur VALUE de manière ascendante (,asc) ou descendante (,desc). Il est possible de trier sur plusieurs champs VALUES
field=VALUE ou field[] =VALUES	Filtrage sur le champs VALUE ou les champs VALUES voulus
search=VALUE	Renvoie uniquement les éléments dont n'importe quel champ contient le texte VALUE
search~x=VALUE	Renvoie uniquement les éléments dont n'importe quel champ ne contient pas le texte VALUE
cursor=VALUE	Valeur du champ 'cursor' renvoyé lors de la dernière requête avec le paramètre *limit=VALUE* défini afin d'obtenir la suite des éléments

champ=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur égale à la valeur VALUE spécifiée.
champ~x=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n'ont pas une valeur égale à la valeur VALUE spécifiée.
champ~s=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur contenant la valeur VALUE spécifiée.
champ~sx=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n'ont pas une valeur contenant la valeur VALUE spécifiée.
champ~i=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur se situant dans l'intervalle spécifié sous la forme '[' ou ']' suivi de {valeur-min}, puis ';', puis {valeur-max}, puis ']' ou '['.
champ~ix=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur ne se situant pas dans l'intervalle spécifié sous la forme '[' ou ']' suivi de {valeur-min}, puis ';', puis {valeur-max}, puis ']' ou '['.

Tableau sur les paramètres d'une requête GET events

Nom du champs	**Description**
time	Timestamp d'apparition de l'évènement
deviceId	Identifiant de l'équipement
device	Nom de l'équipement
variableId	Identifiant de la variable
variable	Nom de la variable
variableCategories	Catégories de la variable
label	Libellé de l'événement
action	Action de l'événement (passage de telle valeur à telle valeur)

Tableau sur les champs disponibles pour un événement

Par défaut, toutes les valeurs seront retournées avec tous les champs dont le tri sera par ordre chronologique d'apparition d'événements.

Un exemple de requête HTTP est:

```
https://192.168.1.223/events?limit=2&field[]=variableId&field[]=time
```

Le code écrit en python sera le suivant:

```
query = 'events'
param = {'limit': 2, 'field[]': ['variableId', 'time']}
req = requests.get(SERVER_URI + query, params=param, headers={
        'Authorization': 'Bearer {api_key}'.format(api_key=API_KEY)
    })
if req.status_code == 200:
    print('OK')
```

```
    data = req.json()
    print(data)
else:
    print('ERR:', req.json())
```

La réponse à cette requête est:

```
{'items': [{'variableId': '62238', 'time': -1755053102}, {'variableId': '62268
```

3.2.2.5 Valeurs historisées

Pour obtenir la liste des valeurs historisées, la requête HTTP sera:

```
/plot-values
```

Cette requête peut recevoir plusieurs paramètres supplémentaires afin d'avoir des informations plus précises. Les paramètres qui ne prendront qu'une valeur seront identifiés par *param=VALUE* et les paramètres qui peuvent prendre une liste de valeurs seront identifiés par *param[]=VALUES*:

Paramètre	Description
limit=VALUE	Nombre d'éléments (VALUE) à renvoyer au maximum
sort=VALUE(,asc ou sort[]=VALUES	Tri sur VALUE de manière ascendante (,asc) ou descendante (,desc). Il est possible de trier sur plusieurs champs VALUES
field=VALUE ou field[]=VALUES	Filtrage sur le champs VALUE ou les champs VALUES voulus
search=VALUE	Renvoie uniquement les éléments dont n'importe quel champ contient le texte VALUE
search~x=VALUE	Renvoie uniquement les éléments dont n'importe quel champ ne contient pas le texte VALUE
cursor=VALUE	Valeur du champ 'cursor' renvoyé lors de la dernière requête avec le paramètre *limit=VALUE* défini afin d'obtenir la suite des éléments
avgValues=VALUE	Renvoie des valeurs moyennée (VALUE=1) ou pas (VALUE=0). Par défaut, les valeurs ne sont pas moyennées.
nbValues=VALUE	Renvoie VALUE valeurs avec un pas de temps égal entre les valeurs
champ=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur égale à la valeur VALUE spécifiée.
champ~x=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n'ont pas une valeur égale à la valeur VALUE spécifiée.
champ~s=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur contenant la valeur VALUE spécifiée.
champ~sx=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n'ont pas une valeur contenant la valeur VALUE spécifiée.

champ~i=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur se situant dans l'intervalle spécifié sous la forme '[' ou ']' suivi de {valeur-min}, puis ';', puis {valeur-max}, puis ']' ou '['.
champ~ix=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur ne se situant pas dans l'intervalle spécifié sous la forme '[' ou ']' suivi de {valeur-min}, puis ';', puis {valeur-max}, puis ']' ou '['.

Tableau sur les paramètres d'une requête GET plot-values

```
Note
Il est important de savoir qu'il est indispensable de mettre le paramètre vari
Il est à noter aussi que l'on ne peut réaliser une requête plot-values que sur
```

Nom du champs	Description
time	Timestamp de stockage de la valeur
rawValue	Valeur brute de la variable
displayValue	Valeur tronquée de la variable

Tableau sur les champs disponibles pour une valeur historisée

Un exemple de requête HTTP est:

```
https://192.168.1.223/plot-values?variableID=19212&time~i=[1634680800; 1653516
```

Le code écrit en python sera le suivant:

```
query = 'plot-values'
param = {'variableId': 19212, 'time~i': '[1634680800; 1653516000[', 'nbValues'
req = requests.get(SERVER_URI + query, params=param, headers={
        'Authorization': 'Bearer {api_key}'.format(api_key=API_KEY)
    })
if req.status_code == 200:
    print('OK')
    data = req.json()
    print(data)
else:
    print('ERR:', req.json())
```

La réponse à cette requête est:

```
{"items":[{"time":1638447840,"rawValue":2823.39497961264,"displayValue":"2823.
"cursor":"{\"variableId\":\"19212\",\"time~i\":\"[1634680800; 1653516000[\",\"
```

3.2.2.6 Compteurs historisés

Pour obtenir la liste des compteurs historisés, la requête HTTP sera:

```
/diff-values
```

Cette requête peut recevoir plusieurs paramètres supplémentaires afin d'avoir des informations plus précises. Les paramètres qui ne prendront qu'une valeur seront identifié par *param=VALUE* et les paramètres qui peuvent prendre une liste de valeurs seront identifiés par *param[]=VALUES*:

limit=VALUE	Nombre d'éléments (VALUE) à renvoyer au maximum
sort=VALUE(,asc/desc) ou sort[]=VALUES(,asc/desc)	Tri sur VALUE de manière ascendante (,asc) ou descendante (,desc). Il est possible de trier sur plusieurs champs VALUES
field=VALUE ou field[] =VALUES	Filtrage sur le champs VALUE ou les champs VALUES voulus
search=VALUE	Renvoie uniquement les éléments dont n'importe quel champ contient le texte VALUE
search~x=VALUE	Renvoie uniquement les éléments dont n'importe quel champ ne contient pas le texte VALUE
cursor=VALUE	Valeur du champ 'cursor' renvoyé lors de la dernière requête avec le paramètre *limit=VALUE* défini afin d'obtenir la suite des éléments
avgValues=VALUE	Renvoie des valeurs moyennée (VALUE=1) ou pas (VALUE=0). Par défaut, les valeurs ne sont pas moyennées.
nbValues=VALUE	Renvoie VALUE valeurs avec un pas de temps égal entre les valeurs
champ=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur égale à la valeur VALUE spécifiée.
champ~x=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n'ont pas une valeur égale à la valeur VALUE spécifiée.
champ~s=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur contenant la valeur VALUEspécifiée.
champ~sx=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n'ont pas une valeur contenant la valeur VALUE spécifiée.
champ~i=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur se situant dans l'intervalle spécifié sous la forme '[' ou ']' suivi de {valeur-min}, puis ';', puis {valeur-max}, puis ']' ou '['.
champ~ix=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur ne se situant pas dans l'intervalle spécifié sous la forme '[' ou ']' suivi de {valeur-min}, puis ';', puis {valeur-max}, puis ']' ou '['.

Tableau sur les paramètres d'une requête GET diff-values

```
Note
Il est important de savoir qu'il est indispensable de mettre le paramètre vari
```

Nom du champs	Description
time	Timestamp de stockage de la valeur
rawValue	Valeur brute de la variable

displayValue	Valeur tronquée de la variable

Tableau sur les champs disponibles pour un compteur historisé

Un exemple de requête HTTP est:

```
https://192.168.1.223/diff-values?variableId=19212&nbValues=4&time~i=[16346808
```

Le code écrit en python sera le suivant:

```
query = 'diff-values'
param= {'variableId': 19212, 'time~i': '[1634680800; 1653516000[', 'nbValues':
req = requests.get(SERVER_URI + query, params=param, headers={
        'Authorization': 'Bearer {api_key}'.format(api_key=API_KEY)
    })
if req.status_code == 200:
    print('OK')
    data = req.json()
    print(data)
else:
    print('ERR:', req.json())
```

La réponse à cette requête est:

```
{'items': [{'time': 1640959200, 'rawValue': 545510, 'displayValue': '545510'},
```

3.2.2.7 Données historiques brutes

Pour obtenir la liste des historiques brutes, la requête HTTP sera:

```
/log-entries
```

```
Avertissement
Cette requête est à utiliser avec précaution car elle permet notamment d'envoy
```

Cette requête peut recevoir plusieurs paramètres supplémentaires afin d'avoir des informations plus précises. Les paramètres qui ne prendront qu'une valeur seront identifiés par *param=VALUE* et les paramètres qui peuvent prendre une liste de valeurs seront identifiés par *param[]=VALUES*:

limit=VALUE	Nombre d'éléments (VALUE) à renvoyer au maximum
sort=VALUE(,asc/desc) ou sort[]=VALUES(,asc/desc)	Tri sur VALUE de manière ascendante (,asc) ou descendante (,desc). Il est possible de trier sur plusieurs champs VALUES
field=VALUE ou field[] =VALUES	Filtrage sur le champs VALUE ou les champs VALUES voulus
search=VALUE	Renvoie uniquement les éléments dont n'importe quel champ contient le texte VALUE
search~x=VALUE	Renvoie uniquement les éléments dont n'importe quel champ ne contient pas le texte VALUE

cursor=VALUE	Valeur du champ ‘cursor’ renvoyé lors de la dernière requête avec le paramètre *limit=VALUE* défini afin d’obtenir la suite des éléments
champ=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur égale à la valeur VALUE spécifiée.
champ~x=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n’ont pas une valeur égale à la valeur VALUE spécifiée.
champ~s=VALUE	Renvoie uniquement les éléments dont les champs spécifiés ont une valeur contenant la valeur VALUE spécifiée.
champ~sx=VALUE	Renvoie uniquement les éléments dont les champs spécifiés n’ont pas une valeur contenant la valeur VALUE spécifiée.
champ~i=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur se situant dans l’intervalle spécifié sous la forme ‘[‘ ou ‘]’ suivi de {valeur-min}, puis ‘;’, puis {valeur-max}, puis ‘]’ ou ‘[‘.
champ~ix=VALUE	Renvoie uniquement les éléments dont les champs spécifiés contiennent une valeur ne se situant pas dans l’intervalle spécifié sous la forme ‘[‘ ou ‘]’ suivi de {valeur-min}, puis ‘;’, puis {valeur-max}, puis ‘]’ ou ‘[‘.

Tableau sur les paramètres d’une requête GET log_entries

Nom du champs	**Description**
id	Identifiant de la valeur stockée
time	Timestamp de stockage de la valeur
variableId	Identifiant de la variable
variable	Nom de la variable
deviceId	Identifiant de l’équipement
device	Nom de l’équipement
type	Type de la valeur stockée
value	Valeur de la variable à ce timestamp

Tableau sur les champs disponibles pour une donnée historique brute

Un exemple de requête HTTP est:

```
https://192.168.1.223/log-entries?variableId=19212&time~i=[1634680800; 1653516
```

Le code écrit en python sera le suivant:

```
query = 'log-entries'
param= {'variableId': 19212, 'time~i': '[1634680800; 1653516000[', 'limit':2}
req = requests.get(SERVER_URI + query, params=param, headers={
        'Authorization': 'Bearer {api_key}'.format(api_key=API_KEY)
    })
if req.status_code == 200:
```

```
    print('OK')
    data = req.json()
    print(data)
else:
    print('ERR:', req.json())
```

La réponse à cette requête est:

```
{'items': [{'id': '35441586', 'time': 1634710517, 'variableId': '19212', 'vari
```

3.2.3 Méthode POST

Cette méthode est utilisée pour créer une nouvelle entrée dans une table. Pour cela, il faut choisir la table que l'on veut modifier puis ensuite définir les paires clés/valeurs de l'élément que l'on veut créer.

3.2.3.1 Variables

Pour créer une nouvelle variable, la requête HTTP sera:

```
/variables
```

Cette requête sera suivie d'un dictionaire dont la clé 'item' contient elle -même un dictionnaire contenant les clés/valeurs de certains champs de la variable:

```
{'item': {'deviceId': ..., 'device': ..., 'displayLabel': ..., ...}
```

Ainsi, on peut renseigner avec des valeurs les clés suivantes:

Nom du champs	Description
deviceId	Identifiant de l'équipement
device	Nom de l'équipement
displayLabel	Valeur de le la représentation de la variable
address	Adresse de la variable
type	Type de la variable
intRegister	Adresse de la variable
priority	Priorité de la variable
mnemonic	Mnémonique de la variable
label	Libellé de la variable
comment	Commentaire
isInternal	Booléen indiquant si la variable est interne ou pas
isPublic	Booléen indiquant si la variable est publique ou pas
isWriteable	Booléen indiquant si la variable peut être écrite ou pas
minWriteValue	Valeur minimale que peut prendre la variable
maxWriteValue	Valeur maximale que peut prendre la variable
categories	Catégories dont fait partie la variable

format	Format de la variable
floatPrecision	Entier indiquant le nombre de chiffre significatif après la virgule. 0 si la variable n'est pas un décimale.
unit	Unité de la variable. Si aucune unité alors » est retourné.
minRemoteScaleValue	Valeur minimale brute
maxRemoteScaleValue	Valeur maximale brute
minLocalScaleValue	Valeur minimale lors de la mise à l'échelle
maxLocalScaleValue	Valeur maximale lors de la mise à l'échelle
inhibitionUser	Utilisateur ayant inhibé l'alarme
inhibitionStartTime	Date de début de l'inhibition
inhibitionStopTime	Date de fin de l'inhibition
hasAlarm	Booléen autorisant la gestion des alarmes
alarmLabel	Libellé de l'alarme
alarmCondition	Condition de l'alarme
hasEvent	Booléen autorisant la gestion des événements
eventLabel	Libellé de l'événement
eventCondition	Condition de l'événement
hasPlot	Booléen autorisant l'historisation des valeurs
plotIsAveraged	Booléen autorisant de moyenner la courbe
plotMinDelay	Délai entre deux points en secondes sur la courbe
plotMinDiff	Hystérésis de la variable
snmpOID	Adresse SNMP de la variable
snmpType	Type SNMP de la variable
rawValue	Valeur actuelle de la variable
displayValue	Valeur formatée de la variable

Tableau sur les champs disponibles pour une variable

```
Information
Rien de nous oblige à remplir tous les champs pour créer une variable et il n'
```

Un exemple de requête HTTP est:

```
https://192.168.1.223/mserver/ui/api/v1/variables
```

avec comme paramètres:

```
{'item': {'label': 'temperature', 'deviceId': 2}
```

Le code écrit en python sera le suivant:

```
query = 'variables'
data = {'item' : json.dumps(
    {
        'label': 'temperature',
```

```
        'deviceId': 2}, sort_keys=True)}
req = requests.post(SERVER_URI + query, data=data, headers={
        'Authorization': 'Bearer {api_key}'.format(api_key=API_KEY)
    })
print(req.json())
```

La variable une fois créée, la sortie (req.json()) affiche l'identifiant de la variable dans la base de données (champ id).

3.2.3.2 Equipements

Pour créer une nouveau équipement, la requête HTTP sera:

```
/devices
```

Cette requête sera suivie d'un dictionaire dont la clé 'item' contient elle -même un dictionnaire contenant les clés/valeurs de certains champs de la variable:

```
{'item': {'isActive': ..., 'type': ..., 'name': ..., ...}
```

Ainsi, on peut renseigner avec des valeurs les clés suivantes:

Nom du champs	Description
isActive	Booléen indiquant si l'équipement est actif (1) ou en erreur (0)
type	Type de l'équipement
name	Nom de l'équipement
comment	Commentaire
displayLabel	Valeur de le la représentation de l'équipement
zones	Zone(s) de l'équipement
purgeDelay	Délai de purge des historiques (en jours)
timeout	Timeout (en secondes)
refreshDelay	Délai de rafraîchissement (en secondes)
comProtocol	Protocole de communication utilisé
comPort	Port de communication utilisé
slaveNumber	Numéro d'esclave. None pour aucun.
password	Mot de passe de l'équipement
comState	État de la communication
alarms	Nombre d'alarmes actives
isEditable	Booléen indiquant si l'équipement est modifiable par un utilisateur

Tableau sur les champs disponibles pour un équipement

```
Information
Rien de nous oblige à remplir tous les champs pour créer un équipement et il n
```

Un exemple de requête HTTP est:

```
https://192.168.1.223/mserver/ui/api/v1/variables
```

avec comme paramètres:

```
{'item': {'label': 'temperature', 'deviceId': 2}
```

Le code écrit en python sera le suivant:

```
query = 'devices'
data =  {'item' : json.dumps(
    {
        'name': 'Equipement_test',
        'comment': 'Ceci est un test'}, sort_keys=True)}
req = requests.post(SERVER_URI + query, data=data, headers={
        'Authorization': 'Bearer {api_key}'.format(api_key=API_KEY)
    })
print(req.json())
```

L'équipement une fois créé, la sortie (req.json()) affiche l'identifiant de l'équipement dans la base de données (champ id).

3.2.4 Méthode PATCH

Cette méthode est utilisée pour modifier un élément dans une table.

3.2.4.1 Variables

Pour modifier une variable, la requête HTTP sera:

```
/variables/{id}
```

Avec 'id' correspondant à l'index de la variable dans la base de données. Cette requête sera suivie d'un dictionaire dont la clé 'item' contient elle -même un dictionnaire contenant les clés/valeurs de certains champs de la variable:

```
'item={'deviceId': ..., 'device': ..., 'displayLabel': ..., ...}'
```

Ainsi, on peut renseigner ou modifier avec des valeurs les clés suivantes:

Nom du champs	Description
deviceId	Identifiant de l'équipement
device	Nom de l'équipement
displayLabel	Valeur de le la représentation de la variable
address	Adresse de la variable
type	Type de la variable
intRegister	Adresse de la variable
priority	Priorité de la variable
mnemonic	Mnémonique de la variable

label	Libellé de la variable
comment	Commentaire
log	Booléen indiquant si la modification de la variable lève un évènement ou non
isInternal	Booléen indiquant si la variable est interne ou pas
isPublic	Booléen indiquant si la variable est publique ou pas
isWriteable	Booléen indiquant si la variable peut être écrite ou pas
minWriteValue	Valeur minimale que peut prendre la variable
maxWriteValue	Valeur maximale que peut prendre la variable
categories	Catégories dont fait partie la variable
format	Format de la variable
floatPrecision	Entier indiquant le nombre de chiffre significatif après la virgule. 0 si la variable n'est pas un décimale.
unit	Unité de la variable. Si aucune unité alors » est retourné.
minRemoteScaleValue	Valeur minimale brute
maxRemoteScaleValue	Valeur maximale brute
minLocalScaleValue	Valeur minimale lors de la mise à l'échelle
maxLocalScaleValue	Valeur maximale lors de la mise à l'échelle
inhibitionUser	Utilisateur ayant inhibé l'alarme
inhibitionStartTime	Date de début de l'inhibition
inhibitionStopTime	Date de fin de l'inhibition
hasAlarm	Booléen autorisant la gestion des alarmes
alarmLabel	Libellé de l'alarme
alarmCondition	Condition de l'alarme
hasEvent	Booléen autorisant la gestion des événements
eventLabel	Libellé de l'événement
eventCondition	Condition de l'événement
hasPlot	Booléen autorisant l'historisation des valeurs
plotIsAveraged	Booléen autorisant de moyenner la courbe
plotMinDelay	Délai entre deux points en secondes sur la courbe
plotMinDiff	Hystérésis de la variable
snmpOID	Adresse SNMP de la variable
snmpType	Type SNMP de la variable
rawValue	Valeur actuelle de la variable
displayValue	Valeur formatée de la variable

Tableau sur les champs disponibles pour une variable

Un exemple de requête HTTP est:

```
https://192.168.1.223/mserver/ui/api/v1/variables
```

avec comme paramètres:

```
'item={'label': 'temperature', 'deviceId': 2}'
```

Le code écrit en python sera le suivant:

```
query = 'variables'
data =  {'item' : json.dumps(
    {
        'label': 'temperature',
        'deviceId': 2}, sort_keys=True)}
req = requests.patch(SERVER_URI + query, data=data, headers={
        'Authorization': 'Bearer {api_key}'.format(api_key=API_KEY)
    })
print(req.json())
```

La variable une fois modifiée, la sortie (req.json()) affiche None.

```
Information
Pour les variables de type chaîne de caractères (MS), il est indispensable d'e
Par exemple:
data = {'item' : json.dumps(
    {
        'label': 'url',
        'rawValue': json.dumps('https://test.fr')}, sort_keys=True)}
```

3.2.4.2 Equipements

Pour modifier un équipement, la requête HTTP sera:

```
/devices/{id}
```

Avec ‘id’ correspondant à l’index de l’équipement dans la base de données. Cette requête sera suivie d’un dictionaire dont la clé ‘item’ contient elle -même un dictionnaire contenant les clés/valeurs de certains champs de la variable:

```
'item={'isActive': ..., 'type': ..., 'name': ..., ...}'
```

Ainsi, on peut renseigner avec des valeurs les clés suivantes:

Nom du champs	Description
isActive	Booléen indiquant si l’équipement est actif (1) ou en erreur (0)
type	Type de l’équipement
name	Nom de l’équipement
comment	Commentaire
displayLabel	Valeur de le la représentation de l’équipement
zones	Zone(s) de l’équipement
purgeDelay	Délai de purge des historiques (en jours)
timeout	Timeout (en secondes)

refreshDelay	Délai de rafraîchissement (en secondes)
comProtocol	Protocole de communication utilisé
comPort	Port de communication utilisé
slaveNumber	Numéro d'esclave. None pour aucun.
password	Mot de passe de l'équipement
comState	État de la communication
alarms	Nombre d'alarmes actives
isEditable	Booléen indiquant si l'équipement est modifiable par un utilisateur

Tableau sur les champs disponibles pour un équipement

Un exemple de requête HTTP est:

```
https://192.168.1.223/mserver/ui/api/v1/variables
```

avec comme paramètres:

```
'item={'label': 'temperature', 'deviceId': 2}'
```

Le code écrit en python sera le suivant:

```
query = 'devices'
data = {'item' : json.dumps(
    {
        'name': 'Equipement_test',
        'comment': 'Ceci est un test'}, sort_keys=True)}
req = requests.patch(SERVER_URI + query, data=data, headers={
        'Authorization': 'Bearer {api_key}'.format(api_key=API_KEY)
    })
print(req.json())
```

L'équipement une fois modifié, la sortie (req.json()) affiche None.

3.2.5 Méthode DELETE

Cette méthode est utilisée pour supprimer un élément d'une table.

3.2.5.1 Variables

Pour supprimer une variable d'une table, la requête HTTP sera:

```
/variables/{id}
```

Avec 'id' correspondant à l'index de la variable dans la base de données. Il n'y a besoin d'aucun paramètre.

3.2.5.1 Equipements

Pour supprimer une variable d'une table, la requête HTTP sera:

`/devices/{id}`

Avec ‘id’ correspondant à l’index de la variable dans la base de données. Il n’y a besoin d’aucun paramètre.