

23/06/2023

Fiche explicative de GRIST

Fonctionnement / Exemples

Table des matières

Table des matières	2
I. Introduction - contexte	3
II. Présentation de GRIST	3
III. Fonctions basiques.....	4
- Fonctions type Excel / Formule	4
- Système de vue	4
- Type des colonnes	7
IV. Création de vue customisé / exemples.....	8
1. <i>Création du code des vues</i>	8
2. <i>Création de la vue sur GRIST</i>	9
3. <i>Configuration de la vue</i>	10
I. Ajout du lien et droit d'accès	10
II. Configuration de widget	10
III. Assignment des variables	10
4. <i>Vus customisé – exemples --</i>	11
- Carte	11
- Formulaire / carte cantine Type	12
- Bloc Note / Clip Board	13
- Lecteur vidéo	14
- Embedding GRIST	15
V. Sources et liens	16



I. Introduction - contexte

Notre projet consiste à rechercher et analyser des solutions libres telles que **OnlyOffice** et **Grist** pour *remplacer les produits éditeurs traditionnels* tels que le **pack Office** comprenant **Word, Excel, Access**, et autres. L'objectif est de *trouver des alternatives open-source* offrant des fonctionnalités similaires, voire supérieures, tout en préservant la flexibilité, la sécurité et l'accessibilité.

Dans notre recherche, nous évaluerons la convivialité, l'adaptabilité aux besoins des utilisateurs et l'intégration de ces solutions dans les environnements de travail existants. Des critères tels que la compatibilité des formats de fichiers, la sécurité des données, la collaboration en temps réel et l'évolutivité seront pris en compte. Nous fournirons une évaluation complète de chaque solution, en mettant en évidence leurs avantages et leurs limitations, afin de recommander la meilleure alternative pour remplacer le **pack Office** dans notre contexte spécifique.

Ce projet vise à promouvoir l'utilisation de solutions libres, favorisant la liberté de choix, la collaboration ouverte et le développement communautaire. En adoptant des alternatives open-source, nous contribuerons à encourager l'innovation et à assurer la pérennité de nos outils bureautiques. De plus, nous garantirons la compatibilité avec les formats standardisés et offrirons une expérience utilisateur améliorée.

II. Présentation de GRIST

GRIST est une application innovante qui offre une solution de remplacement à **Microsoft Access**, une base de données largement utilisée. Conçue pour être conviviale et puissante, **GRIST** permet aux utilisateurs de créer, gérer et organiser efficacement leurs données, tout en offrant une interface intuitive et flexible tout en étant libre.

Grâce à ses fonctionnalités avancées, **GRIST** simplifie le processus de création de formulaires, de requêtes et de rapports, offrant ainsi une meilleure expérience utilisateur. Que ce soit pour une utilisation personnelle ou professionnelle, **GRIST** fournit un environnement fiable et sécurisé pour stocker et manipuler les informations, facilitant ainsi la gestion des données de manière efficace et accessible.

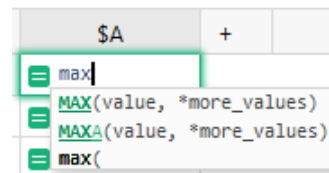
III. Fonctions basiques

- Fonctions type Excel / Formule

Dans GRIST nous avons des fonctions qui sont similaire à Excel comme « **Min** » / « **Max** » / « **Médiane** » / etc...

Les formules que l'on peut rajouter sont fait à base de code *Python* et permettent donc un éventail de choix plus élevé qu'avec Accès.

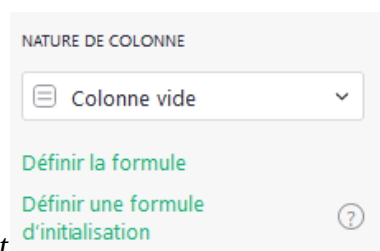
Pour en rajouter il y a différentes façons de procéder, on peut simplement écrire dans une des cases de la colonne « = » suivit de notre formule en incluant les données.



On peut le faire à partir de l'onglet « **Colonne** » vers le centre de la cliquant sur « **Définir la formule** » puis il suffit de l'écrire.

Dessous on peut rajouter une « **Formule d'initialisation** » qui permet de faire la même chose, elle se déclenchent dans certains cas, et valeur calculée comme donnée.

Utile pour stocker l'horodatage ou l'auteur d'un nouvel enregistrement, des données, et plus encore.



Penser à renommer les colonnes et les table que vous allez / avait créer pour mieux se repérer !

Exemple de formules :

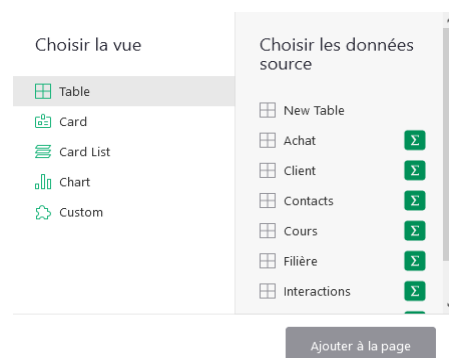
*Ce code permet de faire office de « **Clé unique** »*

```
( = $MaColonne and len(MaTable.lookupRecords(MaColonne=$MaColonne)) > 1)
```

PYTHON

- Système de vue

La création de vue permet de crée de rajouter une visuelle de **table dynamique**¹, **des cartes**², **des listes**², **des diagrammes et schéma**³ et **des vue customisée** (Voir [vue customisé](#)).



1. La création de vue de type « **Table** » permet de mettre un visuelle (*dynamique ou non*) sur la page présente, pour se faire dans l'onglet création de la vue il suffit de sélectionner les données sources et des choisir les données qui seront utilisée (*Via widgets ou choix manuelle*).

II CLIENTS

	client_id	nom	prénom	Sexe	adresse	code_postal	ville	téléphone	commande	+
1	1	TURMO	Gérard	H	24 rue Marchal	67000	Strasbourg	753101010	🔗 Clients[1]	
2	2	JONARD	Marie	F	1 rue de Madrid	67200	Strasbourg	656303030	🔗 Clients[2]	
3	3	VERDIER	Jean- Claude	H	31 bdl Claude	67150	Erstein	755211212	🔗 Clients[5]	
4	4	ALAIN	Morgon	F	28 rue Marchal	67000	Strasbourg	754214512	🔗 Clients[5]	
5	5	MULER	Thomas	H	11 rue de Ma...	67200	Strasbourg	766548725	🔗 Clients[2]	
6	6	KHAN	Irfan	H	3 bdl Claude	67150	Erstein	756542454	🔗 Clients[6]	
7	7	LEBLOND	Rémi	H	6 rue foret noire	67202	Strasbourg	4528868	🔗 Clients[7]	
8									🔗	

PRODUITS

	pro_id	pro_nom	quantité	\$prix unitaire	\$total	clients	+
1	4557	clé USB	30	54	1620	🔗 Clients[4]	
2						🔗	

2. Pour les vues de type « **Card** » et « **Card List** » le fonctionnement est le même. Les « **Card** » permettent de visualiser les données tout comme au-dessus mais sous forme de tableau.

La différence entre les deux est que « **Card** » affiche les résultats en fonction de la ligne sélectionner, alors que « **Card List** » affiche toute la table.

II CATEGORIE Card List

cat_id

1

cat_lab

clinet_id

🔗 Categorie[2]

commande

🔗 Categorie[1]

ROW 2

cat_id

2

cat_lab

clinet_id

🔗

commande

🔗 Categorie[2]

ROW 3

cat_id

3

cat_lab

clinet_id

🔗

commande

🔗 Categorie[6]

ROW 4

cat_id

4

cat_lab

clinet_id

🔗 Categorie[3]

commande

🔗 Categorie[7]

ROW 5

cat_id

5

cat_lab

clinet_id

🔗 Categorie[2]

commande

🔗 Categorie[6]

ROW 6

COMMANDES Card

1 OF 1

com_id

14

com_num

454545

com_date

18/08/2022

prix de commande

224

note

Client

🔗 4

prod

🔗 24444424

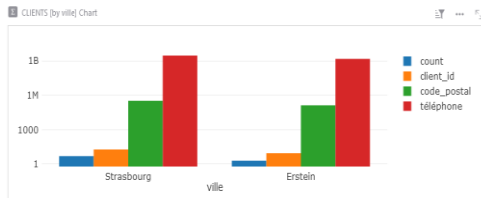
pro_id

🔗 Commandes[4]

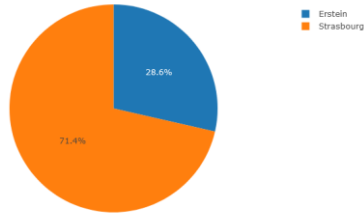
3. Il est possible de créer des schémas et des diagrammes à l'aide de la vue « **Chart** », dont la façon de le crée est tout aussi similaire et possèdent de nombreux type différents :

--- Schéma fait sur la même table ---

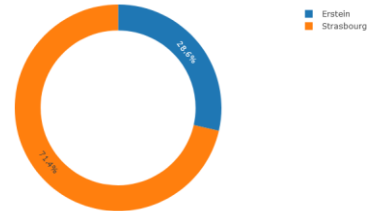
Bar Chart :



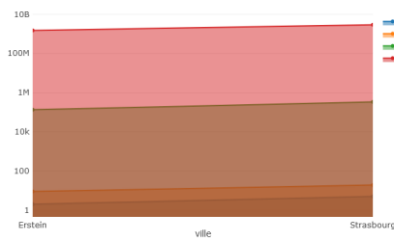
Pie Chart :



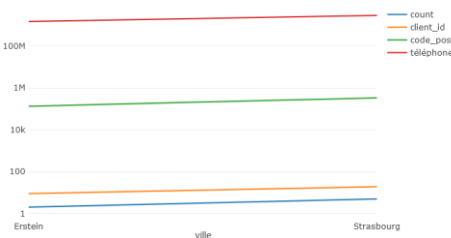
Donut Chart :



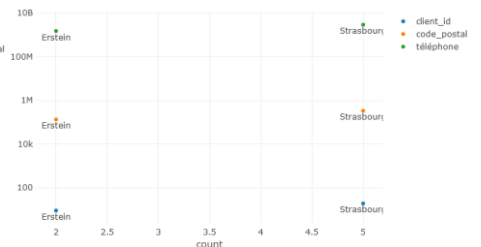
Area Chart :



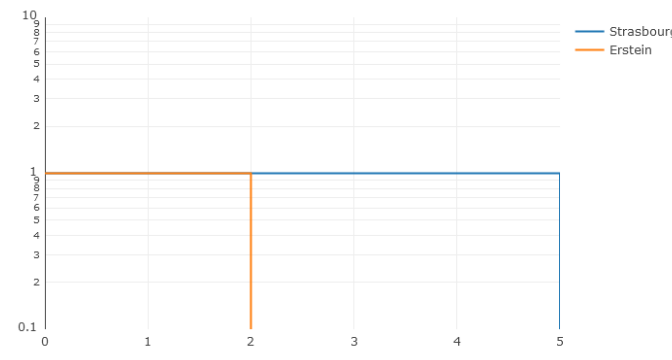
Line Chart :



Scatter Plot :



Kaplan-Meier Plot :



- Type des colonnes

Quand on sélectionne une colonne, il y a l'onglet « Type de colonne »,
Qui permet de choisir le type de valeur qui composera la colonne sélectionnée.

TYPE DE COLONNE

Any

Any	: Permet de stocker des lettres, des chiffres, etc...
Text	: Permet d'écrire n'importe quoi sous forme de texte.
Numeric	: Permet d'écrire des nombres et chiffres entier et décimal.
Integer	: Permet d'écrire des nombres et chiffres entier.
Toggle	: Permet de crée un système de vrai ou faux sous différentes formes : - Une case à cocher : ☐ - Via un texte : (<i>Soit Vrai / Soit Faux</i>) - Un Toggle : ☐
Date	: Permet uniquement d'écrire du texte sous forme de date : 2023-06-06
DateTime	: Permet d'afficher la même chose que date mais basé sur les heures par zone géographique.
Choice	: Permet de crée différent onglet de choix mais en mettre 1 par case.
ChoiceList	: Permet de crée différent onglet de choix mais en mettre plusieurs par case.
Reference	: Permet d'afficher les données d'une colonne d'une autre page (1 par case).
Référence List	: Permet d'afficher les données d'une colonne d'une autre page (Plusieurs par case).
Attachement	: Permet de stocker des URL.

In-Person

In-Person

Phone

Email

To-Do

IV. Création de vue customisé / exemples

1. Création du code des vues

Le code est en **Html** et **JavaScript** voir un rajout de **Css** et de **Json** pour les plus complexes.

Quand on veut créer une vue customisée il faut retenir qu'il y a **3 fonction très importantes**, ils seront utilisés la majorité du temps dans le votre fichier JavaScript :

```
grist.ready();
grist.onRecord(function (record) {
  // Quand le curseur bouge.
});
grist.onRecords(function (record) {
  // Quand les données dans la Table change.
});
grist.onOptions(function (options, interaction) {
  // Quand la configuration change.
});
```

JAVA SCRIPT

Et pour pouvoir choisir les colonnes utilisé dans GRIST la ligne sera utilisé :

```
grist.ready({columns: ['NomColonne', 'NomColonne2']});
```

Pour la création d'un panneau d'option :

```
// Stock une valeur simple.
await grist.setOption('color', '#FF0000');

// Stock un objets complexe en Json.
await grist.setOption('settings', {lines: 10, skipFirst: true});

// Lis les anciennes données sauvegardées.
const color = await grist.getOption('color');

// Réinitialise les options.
await grist.clearOptions();

// Prend et remplace les options.
await grist.getOptions();
await grist.setOptions({...});
```

JAVA SCRIPT

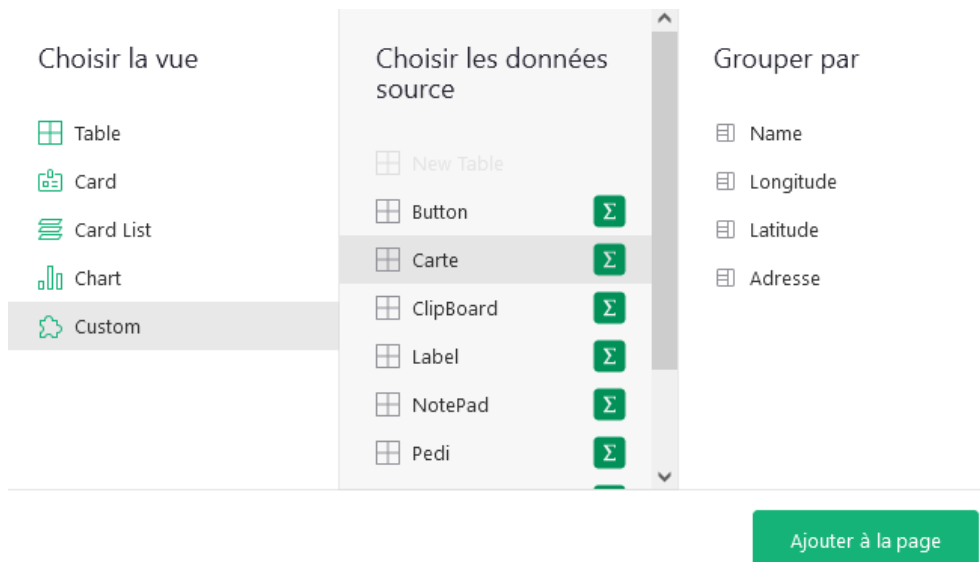
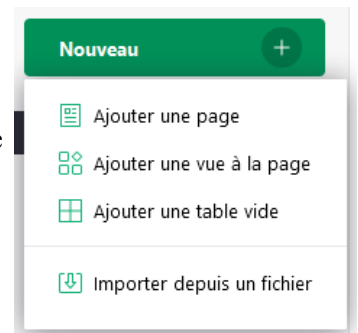
Il y a de nombreuses façons de créer un widget personnalisé, pour en créer un plus complexe et complet aller directement sur : <https://bureautique-libre.strasbourg.eu/documentation.html>

--- Pour voir d'autres exemples voir : [Vues customisé](#) ---

2. Création de la vue sur GRIST

De retour sur GRIST, pour ajouter la vue customisé le procédé est similaire aux vues classique, il faut cliquer sur le bouton « **Nouveau** » en haut à droite de la page puis sur « **Ajouter une vue à la page** ».

Ensuite sur la page de choix des vues, sélectionné « **Custom** » vous avez le choix de sélectionner les données que vous voudrez utiliser, vous pourrez choisir manuellement celle qui seront utilisées ou tout simplement prendre le Widgets pour tous les prendre.



3. Configuration de la vue

----Pour la configuration de la vue l'exemple utilisé est celui de la [Carte](#) à voir ci-dessous----

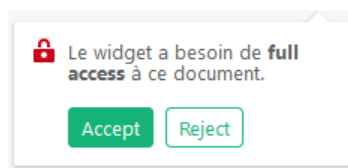
I. Ajout du lien et droit d'accès

Une fois la vue customisée créée il est nécessaire de la configurer afin de pouvoir l'utiliser, pour se faire il faut accéder à l'onglet en cliquant sur la vue.

Une fois dessus vous devriez copier l'URL récupérer plus tôt et le coller dans la case « **Personnalisé** », un affichage devrait commencer à apparaître (*Il sera différent en fonction du contenu du code*).

Une fois fait vous devrez avoir un message qui apparaît :

Qui demanderas l'accès qui lui est requis pour un fonctionnement optimal, il est donc nécessaire de lui donner.



De ce fait toujours leurs donner les droits demandés !

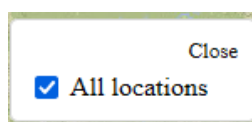
II. Configuration de widget

Certaines vues customisées possèdent un onglet de configuration, pour y accéder il suffit de cliquer sur le bouton « **Ouvrir la configuration** » qui est juste sous le niveau d'accès vu au-dessus.

Ouvrir la configuration

Quand on clique dessus un onglet avec les configurations disponibles apparaît sur notre vue, on peut interagir avec les options que l'on avait créées dessus.

(Cette partie doit être faite lors de la création du code)



III. Assignment des variables

L'assignation des variables est également une partie qui a été créé préalablement dans le code donc toute les vues n'en n'ont pas forcément.

L'onglet se situant juste sous l'onglet « **Configuration** » ou sous le niveau d'accès s'il n'y en a pas, elle permet donc d'assigner certaines variables qui vont être utilisé en sélectionnant la colonne par rapport à la ligne sélectionner via l'emplacement de curseur.

4. Vues customisé – exemples --



Vous trouverez plus de lien sur <https://bureautique-libre.strasbourg.eu/> !

Pour voir le fonctionnement des vues customisé : [Création de vue customisé / exemples](#) !

- Carte

URL de cette exemple : <https://bureautique-libre.strasbourg.eu/Templates/map/>

Après avoir créé cette vue et mis le lien / les droits il est nécessaire de la configurer via les variables de configuration qui seront *pour cette exemple les seules valeurs affichées*.

Dans la variable assigner les colonnes correspondantes aux valeurs que vous voudrez afficher. (Ils ne sont pas tous obligatoire)

Nécessite un accès complet au document !

Et voici un résultat possible :

CARTE

	Name	Longitude	Latitude	Adresse	+
1	Alakurtti	30.749	67.234		
2	Libya	24	23		
3	Strasbourg	7.7521113	48.5734053		
4	Calcutta	84.223	20.34		
5					

CARTE Custom

Vue Trier et Filtrer Données sou

TITRE DE LA VUE
CARTE Custom
[Modifier la vue](#)

PERSONNALISÉ
<https://grist-template.strasbourg.eu>

ACCESS LEVEL
Lire les données source s...
[Ouvrir la configuration](#)
[En savoir plus sur les vues personnalisées](#)

NAME
Name

LONGITUDE
Longitude

LATITUDE
Latitude

GEOCODE (facultatif)
Choisir une colonne de ty...

ADDRESS (facultatif)
Adresse

GEOCODED ADDRESS (facultatif)
Choisir une colonne de ty...

- **Formulaire / carte cantine Type**

- Bloc Note / Clip Board

URL de cette exemple (**Bloc Note**) : <https://bureautique-libre.strasbourg.eu/Templates/notepad/>

URL de cette exemple (**Clip Board**) : <https://bureautique-libre.strasbourg.eu/Templates/clipboard/>

Après avoir créé cette vue et mis le lien / les droits il est nécessaire de la configurer via les variables de configuration qui seront *pour cette exemple les seules valeurs affichées également*.

Ces deux Widgets sont extrêmement similaires, dans là on peut mettre qu'une colonne, qui sera celle dans laquelle les données seront affichées et utilisables.

- Dans le cas du **bloc note**, vous pourrez avec lui afficher du texte / code avec des habillages (**Gras / Italique / etc...**), faire une présentation complète.
- Pour le **Clip Board**, il vous permet quand vous avez une grande quantité de texte dans une case de la visualiser proprement et la copier en intégralité.

Nécessite un accès complet au document !

Et voici des résultats possibles :

The image displays two screenshots of the Grist application interface. The top screenshot shows the 'NOTEPAD' widget, which is a text editor with a dark background and a light-colored border. It contains a table with three columns: 'A', 'B', and 'C'. The 'A' column contains the text 'Vue bloc note indépendante', 'temp', and 'temp'. The 'B' column is empty. The 'C' column contains the text 'temp'. The bottom screenshot shows the 'CLIPBOARD' widget, which is a text editor with a light background and a light-colored border. It contains a table with three columns: 'A', 'B', and 'C'. The 'A' column contains the text 'Test de texte ...', 'Test pour test', and 'Test pour test'. The 'B' column contains the text 'Test pour test', 'Test pour test', and 'Test pour test'. The 'C' column contains the text 'Test pour test', 'Test pour test', and 'Test pour test'. The 'CLIPBOARD' widget also features a green 'Copy Text' button at the top right.

- Lecteur vidéo

URL de cette exemple (**Lecteur vidéo**) : <https://bureautique-libre.strasbourg.eu/Templates/rendervideo/>

Le lecteur vidéo possède le même système de configuration que les précédents, il suffit d'entrer la colonne possédant les vidéos à visualiser qui seront sous forme d'URL YouTube (https://www.youtube.com/watch?v=zNLHX_ezY50).

Une fois renseigné il vous suffit de sélectionner la ligne ayant le lien voulu puis la vidéo apparaîtra dans la vue prêtée à être regardée.

Nécessite un accès de lecture seule !

Et voici un résultat possible :

The screenshot displays the VIDEOREADER interface. On the left, a table lists video entries:

	Video Name	URL
1	Custom Wideget	https://www.youtube.com/watch?v=zNLHX_ezY50
2	RandomVideo	https://www.youtube.com/watch?v=mVtGHknLnwk
3		

On the right, a sidebar contains configuration options:

- TITRE DE LA VUE**: VIDEOREADER Custom
- PERSONNALISÉ**: <https://sl-grist-tmpl-staging.homx>
- ACCESS LEVEL**: Lire les données source sélé...
- URL**: URL

The main area shows a video player with a preview of a video titled 'Château de Bellerive'.

- Embedding GRIST



Un Embedding vous permet d'accéder / de visualiser / de modifier une table d'un autre fichier sans devoir changer de page à chaque fois qu'une modification est nécessaire.

URL de cette exemple (**Embedding GRIST**) : <https://bureautique-libre.strasbourg.eu/Templates/Embedding/>

L'Embedding GRIST possède le même système de configuration que les précédents, il suffit d'entrer la colonne possédant les vidéos à visualiser qui seront sous forme d'URL de tableau GRIST par exemple : (<https://grist.strasbourg.eu/wZvrn73sL7G3/Untitled-document>).

Une fois renseigné il vous suffit de sélectionner la ligne ayant le lien voulu puis le tableau apparaîtra et vous pourrez le modifier d'un autre fichier.

Nécessite un accès complet !

Et voici un résultat possible :

The screenshot displays the Grist application interface. The main table, titled 'TABLE_TEST', has columns: 'ActionButt...', 'Fonctionne ?', and 'Contenu'. It contains six rows, with rows 2, 4, and 5 containing links to Grist documents. Below this, a detailed view of 'TABLE1' is shown, with columns A, B, and C. Row 1 contains the text 'Se tableau est stocké sur un autre fichier'. The right sidebar contains configuration options for the view, including 'TITRE DE LA VUE' (set to 'TABLE_TEST Custom'), 'PERSONNALISÉ' (set to a specific URL), 'ACCESS LEVEL' (set to 'Accès complet au document'), and 'TEXT COLUMN' (set to 'ActionButton').

V. Sources et liens

- Lien vers la page GRIST : <https://grist.strasbourg.eu/>
- Lien vers le support de GRIST : <https://support.getgrist.com/>
- Lien vers documentation Embedding : <https://support.getgrist.com/embedding/>
- Lien vers nos Templates : <https://bureautique-libre.strasbourg.eu/widgets.html>
- Lien vers notre site E-SUITE : <https://bureautique-libre.strasbourg.eu/>
- Lien vers salon d'aide Watcha : <https://watcha.strasbourg.eu/app/#/room/#entraide:watcha.strasbourg.eu>