

Sommaire

- [Sommaire](#)
- [Wekan - Fonctionnalités globales](#)
 - [Fonctionnalités basiques](#)
 - [Fonctionnalités Avancés](#)
- [Wekan - Fonctionnalités précises](#)
 - [REST API](#)
 - [Possibilités / exemples](#)
 - [Modèles](#)
 - [Exemple de langage d'une carte :](#)
 - [Les règles](#)
 - [Les webhooks](#)
 - [Un historique instantanée sur discord](#)
 - [Les droits et autorisation](#)
 - [Champs personnalisés](#)
 - [Envoie de mails](#)
 - [Conversion tableaux](#)
 - [1. Excel](#)
 - [2. HTML](#)
 - [3. CSV](#)
 - [4. Json](#)
- [Base de donnés -MangoDB-](#)
 - [Interface console](#)
 - [Interface graphique](#)
 - [Bug images](#)
- [Divers](#)
 - [Liens](#)
 - [Liens Wekan](#)
 - [Liens MongoDB](#)
 - [Liens divers :](#)
 - [A faire](#)
 - [Autres](#)
 - [Bug :](#)
 - [Bug images](#)
 - [Bug API :](#)
- [Parties supplémétaires -KeyCloak-](#)
 - [Authentification centralisée - OAuth2](#)
 - [Installation des librairies](#)
 - [Lancement d'un container de test](#)
- [Divers](#)
 - [Liens](#)
 - [Liens Keycloak](#)
 - [A faire](#)
- [Info Dives](#)
- [Divers - Perso](#)
 - [Connexion a keycloak sur docker](#)
 - [Connexion au formulaire sur docker](#)

Wekan - Fonctionnalités globales

Fonctionnalités basiques

- Création de différents tableaux
 - Créations de lignes et colonnes personnalisables
 - Listes personnalisables / visuelles et techniques
 - Limitations **WIP** (Taille maximal) douce ou dur
 - Tableau peut être mis en favori pour y accéder à partir de la barre principale
 - Personnalisation interne dans les *cartes / listes / tableaux*
 - (Rajout ou suppression de certaine option comme la date d'ajout, etc..)
 - Affichage du tableau sous forme de **calendrier, de couloires** et **de listes**.
-

- Visibilités des tableaux
 - Modification d'interface global de l'application
 - Modifications d'interface
 - Système de filtre avancée
 - Option de recherche
-

- **Annonces**
 - Qui est un message envoyé à tous les utilisateurs qui s'affiche à la création d'un tableau plus généralement **quand on ouvre un tableau pour la première fois**
 - **Champs personnalisés**
 - **Votes, planing poker**
 - Temps passer sur chaque "**cartes / projets**"
-

- **Modification des droits sur les comptes des utilisateurs**
 - Changement de mail
 - Changement d'identifiant
 - Suppression de compte par oui ou par non
 - Possibilité de **désactiver les comptes**, leurs **créations** ou encore de faire "**mot de passe oublier**"
 - Suppression de **possibilité de création de compte**
 - (Dans la zone inscription sa demandera alors un code d'inscription en plus)
 - Peut être remplacer par un **système d'invitation** a l'application accompagner d'une **invitation à différents tableau** si besoin ou d'autre système comme celui en place avec **Keycloak**.
-

- **Notifications par mail**
 - Exportation de tableaux
 - (Json / Excel / CSV / TSV / HTML)
-

- Ajout de taches
 - **Rajout d'image / pièces jointes / commentaire / lien / checklists / sous objectifs / description / étiquettes**
(Les parties de checklists peut être converties en carte)
 - Rajout de date "**Reçue**" / "**Début**" / "**A échéance**" / "**Fin**"
 - Visualisation de l'historique d'interaction
 - Rajout participants / Personne assigné

-
- Droits / autorisation
 - Droits **utilisateurs**
 - Droits **groupes**
 - Droits **organisations**
 - Droits **tableaux**
-

- On peut copier le lien d'une carte pour y accéder directement
 - (Les liens mis dans une carte sont mis en hyperliens)
-

Fonctionnalité version supérieure :

- Système de traduction manuelle
- Modifier les paramètres des tâches secondaires
- Mise à part le système de stockage le reste est plus au moins **similaire**.

WeKan® info	7.30.0
Meteor_version	2.14.1-beta.0
Node_version	14.21.4
MongoDB_version	6.0.13

Fonctionnalités Avancés

- Les *API*
 - La base de données **Mango**
 - Modèles de *cartes / listes / tableaux*
 - Les templates comprennent modification graphique et de contenu
 - Les *règles*
 - Elles permettent de faire une action automatique après un déclencheur
 - Les **webhook**
-
-

Wekan - Fonctionnalités précises

REST API

Possibilités / exemples

⚠ **Token nécessaire** ⚠

Pour récupérer un token utiliser les commandes de login ou en curl :

```
curl http[s]🙄/[lien]/users/login \  
-d "username=[nomUtilisateur]&password=[motDePasse]"
```

OU

```
-d "email=[monEmail]&password=[motDePasse]"
```

Exemple de résultats :

```
{ "id": "NodzpCXXXXXXMozdj", "token": "UVcRid9ropZlz9d8AXXXXXXXpUHJa6Jry99IJ", "tokenExpires": "2024-04-28T09:50:57.713Z" }
```

⚠ **Paramètres :** ⚠

```
username : Mettre le pseudo OU email : Mettre son email
password : Mettre son mot de passe
```

Possibilités d'utilisation des API *-(Testée sur PostMan)-* :
Les API ont une base très proche entre eux et en clair

⚠ **Les créations :** ⚠

Pour les créations on peut **les personnaliser sur beaucoup de points** en les mettant en paramètres.
Dont **l'identifiant utilisateur** qui peut être récupérer en même temps que le token avec l'API *login*.

Exemples, avec la création d'un nouveau tableau et tous ses paramètres.

⚠ **:xxxxxIdAuChoix** = id que l'on récupère avec le **GET /api/XXXXX** en fonction du tableau ⚠

⚠ ⚠ ⚠ **Si vous voulez utiliser les api sur un tableau privée il faut récupérer l'ID dans la base de données**
⚠ ⚠ ⚠

- **Login / register :**
Permet de s'authentifier

```
POST /users/login
-----
POST /users/register
```

--**Pour c'est deux la, aucune authentication est nécessaire**--

- **Boards :**
⚠ **:tableauIdAuChoix** = id que l'on récupère avec **GET /api/boards** ⚠ Les **GET /api/boards** affichent que les tableaux publics

Méthodes	Url	Description
GET	/api/boards	Permet d'afficher tous les tableaux publics.
GET	/api/boards/:tableauIdAuChoix	Permet d'afficher un tableau spécifique.
POST	/api/boards	Permet de créer un tableau.
DELETE	/api/boards/:tableauIdAuChoix	Permet de supprimer un tableau spécifique.
GET	/api/boards/:tableauIdAuChoix/export	Permet d'exporter un tableau spécifique.
PUT	/api/boards/:tableauIdAuChoix/labels	Permet d'ajouter un tag au tableau spécifique.

Méthodes	Url	Description
GET	/api/users/:utilisateurIdAuChoix/boards	Permet de voir les tableaux d'un utilisateur.

• **Checklists :**

Vous pouvez faire la même chose avec les checklist *item* en rajoutant
`/:checklistIdAuChoix/items/:itemIdAuChoix` pour crée / supprimer / afficher des objets de la checklist

Exemple :

```
GET /api/boards/:tableauIdAuChoix/cards/:carteIdAuChoix/checklists
↓
GET
/api/boards/:tableauIdAuChoix/cards/:carteIdAuChoix/checklists/:checklistIdAuChoix/items
/:itemIdAuChoix
```

Méthodes	Url	Description
GET	/api/boards/:tableauldAuChoix/cards/:carteldAuChoix/checklists	Donne la liste des checkbox attaché a une carte.
POST	/api/boards/:tableauldAuChoix/cards/:carteldAuChoix/checklists	Crée une checkbox dans une carte.
GET	/api/boards/:tableauldAuChoix/cards/:carteldAuChoix/checklists/:checklistIdAuChoix	Récupère une checklist.
DELETE	/api/boards/:tableauldAuChoix/cards/:carteldAuChoix/checklists/:checklistIdAuChoix	Supprime une checklist au choix.

• **CardComment :**

Méthodes	Url	Description
GET	/api/boards/:tableauldAuChoix/cards/:idDeLaCarte/comments	Affiche tout les commentaires sur la carte choisie.
POST	/api/boards/:tableauldAuChoix/cards/:idDeLaCarte/comments	Ajoute un commentaires sur la carte choisie.

Méthodes	Url	Description
POST	/api/boards/:tableauIdAuChoix/cards/:idDeLaCarte/comments/:commentaireIdAuChoix	Affiche un commentaire sur la carte choisie.
POST	/api/boards/:tableauIdAuChoix/cards/:idDeLaCarte/comments/:commentaireIdAuChoix	Supprime un commentaire sur la carte choisie.

• CustomFields :

Méthodes	Url	Description
GET	/api/boards/:tableauIdAuChoix/custom-fields	Donne la liste des champ customisée dans le tableau.
POST	/api/boards/:tableauIdAuChoix/custom-fields	Crée un champ customisée dans le tableau.
GET	/api/boards/:tableauIdAuChoix/custom-fields/:champCustomIdAuChoix	Affiche un champ customisée précis dans le tableau.
DELETE	/api/boards/:tableauIdAuChoix/custom-fields/:champCustomIdAuChoix	Supprimer un champ customisée précis dans le tableau.

• Integrations :

Méthodes	Url	Description
GET	/api/boards/api/boards/:tableauIdAuChoix/integrations/:unEntier	Donne une simple intégration au tableau.
PUT	/api/boards/api/boards/:tableauIdAuChoix/integrations/:unEntier	Permet de modifier les donnés d'intégrations.
DELETE	/api/boards/api/boards/:tableauIdAuChoix/integrations/:unEntier	Permet de supprimer une intégration.
DELETE	/api/boards/api/boards/:tableauIdAuChoix/integrations/:unEntier/activities	Permet de supprimer activités abonnés.
POST	/api/boards/api/boards/:tableauIdAuChoix/integrations/:unEntier/activities	Permet d'ajouter des activités abonnés.

• Lists :

Méthodes	Url	Description
GET	/api/boards/:tableauIdAuChoix/lists	Donne la liste des listes de ce tableau.
POST	/api/boards/:tableauIdAuChoix/lists	Ajoute une liste dans le tableau.
POST	/api/boards/:tableauIdAuChoix/lists/:listIdAuChoix	Donne la liste de ce tableau.

Méthodes	Url	Description
DELETE	/api/boards/:tableauIdAuChoix/lists/:listIdAuChoix	supprime une liste dans ce tableau.

Peut ressembler à sa :

DELETE <http://infcau104:8085/api/boards/aJCyxp4M9iMDeWsik/lists/4gnrDgcqRv2bL5bcs>

(En seule paramètre l'authentification avec le token)

• Cartes :

Méthodes	Url	Description
POST	/api/boards/:IdDuTableau/lists/:IdDeLaListe/cards	Ajoute une carte à une liste, un tableau et une colonne.
PUT	/api/boards/:IdDuTableau/lists/:DeIdDeLaListe/cards/:IdDeLaCarte	Met à jour une carte.
DELETE	/api/boards/:IdDuTableau/lists/:IdDeLaListe/cards/:IdDeLaCarte	Supprime une carte.

• Users :

Méthodes	Url	Description
POST	/users/register	Enregistre un nouvel utilisateur.
POST	/api/users	Crée un nouvel utilisateur.
PUT	/api/users/:id	Désactive un utilisateur existant.
PUT	/api/users/:id	Active un utilisateur existant.
PUT	/api/users/:id	L'administrateur prend "la propriété".
DELETE	/api/users/:id	Supprime un utilisateur existant.
GET	/api/users/:id	Obtient les informations d'un utilisateur.
GET	/api/users	Affiche tous les utilisateurs.
GET	/api/user	Obtient les informations de l'utilisateur connecté.

• Swimlanes :

Méthodes	Url	Description
GET	/api/boards/:IdDuTableau/swimlanes	Donne la liste des listes du tableau.
POST	/api/boards/:IdDuTableau/swimlanes	Ajoute une liste dans le tableau.
GET	/api/boards/:IdDuTableau/swimlanes/:swimlaneIdAuChoix	Donne un liste au choix dans le tableau.
DELETE	/api/boards/:IdDuTableau/swimlanes/:swimlaneIdAuChoix	Supprime une liste au choix dans le tableau.

• Et d'autres...

• Les permissions :

Permet d'ajouter des permissions sur un tableau pour un utilisateur

```
POST /api/boards/:tableauIdAuChoix/members/:utilisateurIdAuChoix
```

⚠ **Uniquement l'administrateur peut utiliser (la plupart) des API REST** ⚠

Pour plus de détails aller voir les REST API de Wekan [sur leur forum](#) ou sur leur [github](#).

-(Avec des exemples de codes en *JavaScript, Shell, HTTP, Node.js, Ruby, Python, Java* et *Go* [sur leur forum](#))-

Modèles

Modèles existant : **cartes / listes / tableaux**

- On peut créer des templates en cliquant sur son **profile** -> **Modèles** et le crée à son besoin.
Une fois crée un tableau à partir de ce modèle en cliquant sur **Ajouter un tableau** -> **Modèles** -> **puis entrer le nom du modèle voulu** ensuite il suffit de le crée.
- On peut faire la même dans le tableau avec les **cartes / listes** en procédent de la même manière **Touche pour crée** -> **Modèle** -> **Mettre le nom** -> **Valider**.
- On en ajouter un a partir d'un objet (*carte, table, etc..*) déjà existant a partir du bouton "**link**" (*Disponible sur les versions supérieure uniquement*)
- Les tableau crée / carte / liste sont unique dans le même tableau et peuvent en avoir des différentes...
- Les modèles copie **les couleurs / le contenu**

⚠ **Les modèles ne peuvent que être crée par les utilisateurs classics** ⚠

⚠ **Les connexion via id euroméropole ne fonctionne pas** ⚠

- Les templates sont très efficace pour **automatiser** la création de sections, que ce soit sur le plan **visuel ou technique**, en se basant sur des **cartes, listes** ou **tableaux** déjà existants.
- Ces modèles **simplifient le processus** en générant automatiquement les éléments nécessaires, rendent l'utilisation plus **pratique et rapide** pour la mise en place de contenus structurés et cohérents.
- En exploitant cette fonctionnalité, il devient possible **d'optimiser le flux de travail**, de **garantir la conformité** aux normes établies et de **gagner du temps** dans la conception et la mise en page.

En version supérieur il est possible de créer un objet à partir d'un autre existant sur un autre tableau sans passer par les templates.

Plus d'information avec cette [vidéo explicative](#) ou cette [page Github Officiel](#).

⚠ **Templates = modèles** ⚠

Exemple de langage d'une carte :

```
[
  {
    "title": "Titre de la première carte",
```

```
"description": "Description de la première carte"
},
{
  "title": "Titre de la seconde carte",
  "description": "Description de la deuxième carte"
},
{
  "title": "Titre de la dernière carte",
  "description": "Description de la dernière carte"
}
]
```

Les règles

Elles permettent de faire une action automatique après un déclencheur.

Par exemples crée une colonne après qu'une carte avec un certain nom soit créé.

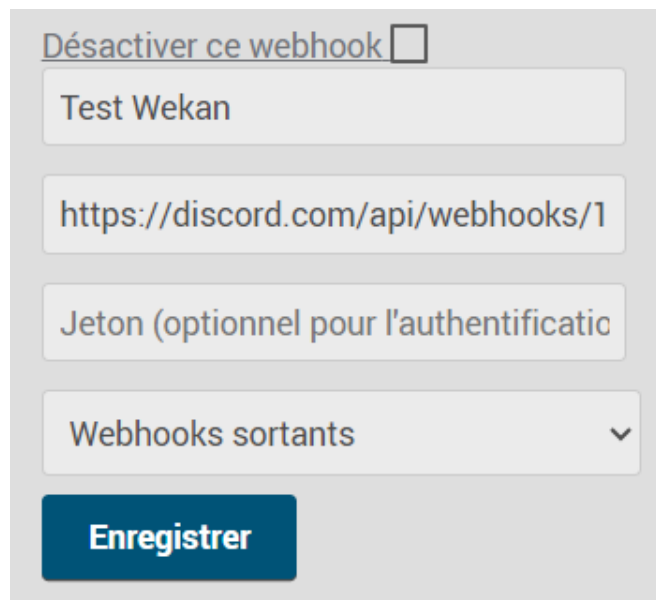
Les webhooks

Permettent de crée différentes fonctionnalités dynamiques

Un historique instantanée sur discord

Pour se faire crée un Webhook sur discord, récupérer son URL et le mettre dans **Profile -> Panneau d'administration -> Webhooks globaux**

Une fois le lien mis sa donne :

The image shows a web form for configuring a webhook. At the top, there is a link "Désactiver ce webhook" followed by a checkbox. Below this is a text input field containing "Test Wekan". The next field is a text input for the URL, containing "https://discord.com/api/webhooks/1". Below that is a text input for the token, containing "Jeton (optionnel pour l'authentificatio". The final field is a dropdown menu labeled "Webhooks sortants" with a downward arrow. At the bottom of the form is a blue button labeled "Enregistrer".

Ainsi toute les modifications seront automatiquement envoyées sur *discord* avec toute les informations, du profile utilisée à l'emplacement et l'action fait.

--Dans cette situation mettre le Webhook en sortie--

Exemples, sa se fait sous forme de :

Matys FREYERMUTH created card "Réunion test Wekan" to list "Liste Test Wekan" at swimlane "Default" at board "Test WebHooks again"
<https://sf-wekan-staging.homol.strasbourg.cloud/b/x6FfixyXK9SE2bPCb/test-webhooks-again/KGyqHHnGqJHDKzTJt>

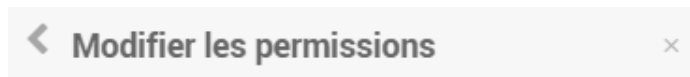
Les droits et autorisation

- Différents types de droits des utilisateurs avec leurs groupes :
 - **Ajustables avec les droits individuelles**
 - **Les organisation**
 - **Les équipes**
 - **Les globaux** (*public ou privée*)

(Également directement dans le tableau en fonction en rajoutant des utilisateurs, des organisations ou encore des équipes pour donner des droits particulier directement sur chaque tableau, ou simplement en mettant le tableau public ou privée qui sera afficher / cacher pour tous les utilisateurs sauf le créateur et ceux mentionner)

- Possibilité de vérifier les utilisateurs, mettre des droits customisé, etc..

Exemple des droits d'utilisateurs qui sont assignables dans un tableau :



Admin

Peut voir et éditer les cartes, supprimer des participants et changer les paramètres du tableau.

Normal ✓

Peut voir et modifier les cartes. Ne peut pas changer les paramètres.

Aucun commentaire

Ne peut pas voir les commentaires et les activités.

Commentaire uniquement

Ne peut que commenter des cartes.

Travailleur

Peut seulement déplacer des cartes, s'assigner à une carte et la commenter.

⚠ ⚠ ⚠ **Les tableau en public peuvent être accéder par TOUT LE MONDE qui possède le lien - donc à éviter**
⚠ ⚠ ⚠

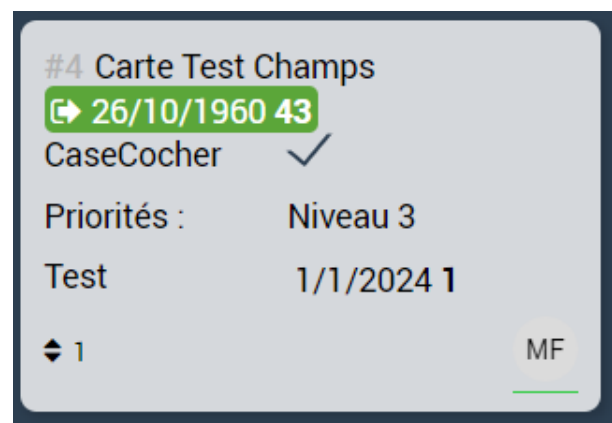
Si on désactive les tableaux publics et qu'il y en a de crée, on ne pourra plus y accéder.

--Les utilisateur qui vont sur un tableau public peuvent lire les données mais pas interagir avec--

Champs personnalisés

Les champs personnalisés permettent de modifier les cartes afin de rajouter différentes choses à l'intérieures comme des **textes, nombres, dates, liste de choix, devises, case à cochers et des modèles de chaînes** en les affichant directement sur la carte ou les afficher en cliquant dessus et tout ça en plus des champs classiques.

Exemples avec des champs (**liste de choix, case à cocher et date**) :



--A compléter - Détailler plus les manière de procéder ici--

Envoie de mails

[Page de dev pour les mails](#)

Permettra d'envoyer des notification / invitations par mail / historique (Via SMTP ?).

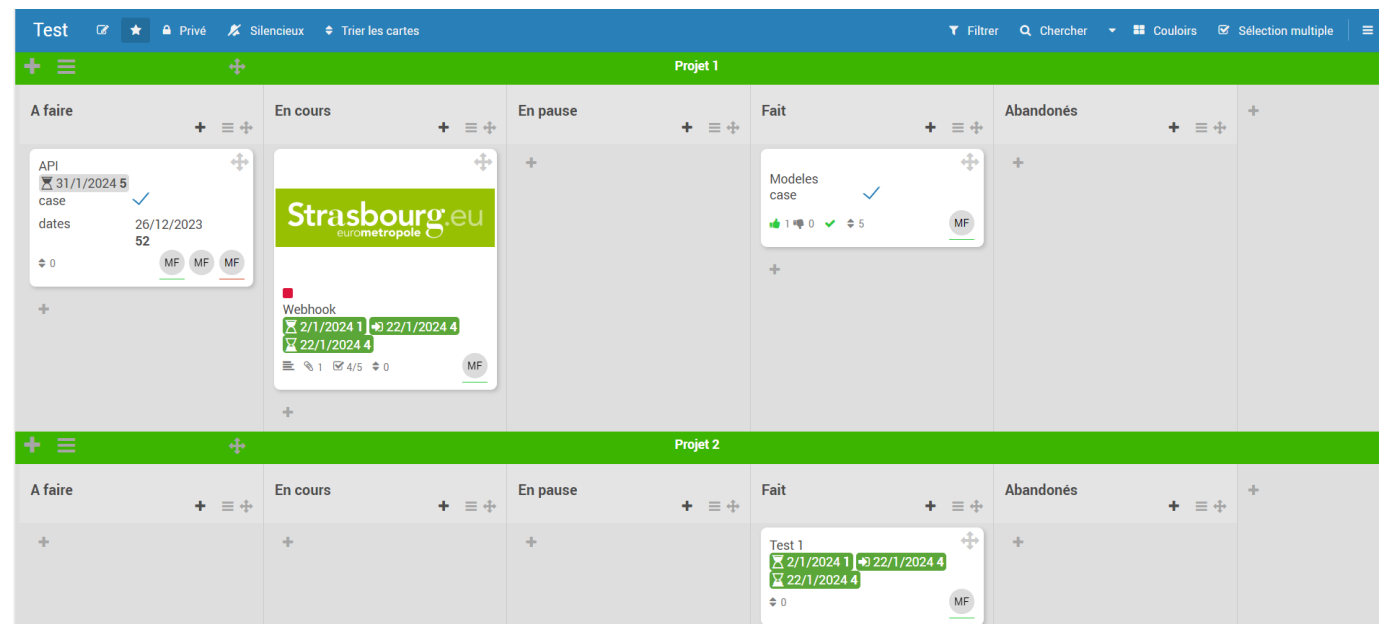
-Passer par un serveur mail ?-

Plus de détails sur [le github wekan pour les mails](#) ou pour plus de détails : [celui-la](#).

--<https://github.com/wekan/univention/issues/6>--

Conversion tableaux

1. Excel



↓↓↓

1	Test											
2												
3	Description											
4												
5	Created at			Modified at			Members			custatg50056		
6	2024/01/22 10:06:00			2024/01/22 10:43:20								
7	Number	Title	Description	Parent card	Owner	Created at	Last modified at	Received	Start	Due	End	List
8	1	Cartes			custatg50056	2024/01/22 10:06:35	2024/01/22 10:11:57					Listes
9	2	Cartes			custatg50056	2024/01/22 10:06:38	2024/01/22 10:11:57					Listes
10	3	Test 1			custatg50056	2024/01/22 10:09:07	2024/01/22 10:11:57					Fait
11	4	API			custatg50056	2024/01/22 10:15:15	2024/01/22 10:35:57	2023/03/26 10:09:00	2024/01/22 10:09:00	2024/01/22 16:00:00	2024/01/22 10:09:00	A faire
12	5	Webhook	Description test		custatg50056	2024/01/22 10:37:51	2024/01/22 10:42:40	2024/01/01 11:00:00	2024/01/22 10:37:00	2024/01/22 16:00:00	2024/01/22 10:38:00	En cours
13	6	Modèles			custatg50056	2024/01/22 10:39:07	2024/01/22 10:42:29					Fait
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												
32												
33												
34												
35												
36												
37												
38												
39												
40												
41												

Informations non affiché / sauvegardée après conversion :

- Les règles
- Champs personnalisés
- Historique d'activités
- Les paramètres visuelles (Couleur des listes, cartes, etc..)
- Paramètres de priorités
- Checklists
- Pièces jointes
- Votes
- Planning poker

2. HTML

En conversion HTML uniquement le visuelle de la page est conserver (À l'exception de quelques éléments comme des icônes) sans possibilité d'interagir avec les carte / listes / etc..

3. CSV

Format difficile à lire à l'œil (Utilisé pour les exportation sur un autre support)

Exemple :

	A	B	C	D	E	F	G	H	I
1	Title,"Description", "List", "Swimlane", "Owner", "Requested By", "Assigned By", "Members", "Assignee", "Labels", "Start", "Due",								
2	Installer **meteo-suivi**,"", "Cloturé", "Applications", "cus\970595", " ", " ", " ", " ", " ", " ", " ", " ", " ", "false", "0", "2023-08-18T16:16:3								
3	Tester Teleport,"", "Idées", "Infrastructure", "cus\970595", " ", " ", " ", " ", " ", " ", " ", " ", " ", "false", "0", "2023-08-18T16:14:56+00:00", "								
4	Installer un cluster de test sur NKE,"", "Cloturé", "Infrastructure", "cus\970595", " ", " ", " ", " ", " ", " ", " ", " ", " ", "false", "0", "2023-08-1								
5	Mettre au point une procédure de remontage du cluster,"Remonter rapidement une copie d'un cluster existant, pour test c								
6	Installer AWX,"", "Fait", "Applications", "cus\970595", " ", " ", " ", " ", " ", " ", " ", " ", " ", "false", "0", "2023-08-02T14:21:2								
7	Tester ArgoCD,"", "Fait", "Infrastructure", "cus\970595", " ", " ", " ", " ", " ", " ", " ", " ", " ", "false", "0",								
8	Tester la solution de sauvegarde Kasten,"", "A faire", "Infrastructure", "cus\970595", " ", " ", " ", " ", " ", " ", " ", " ", " ", "Installation-red Procédure-c								

Même résultats pour les deux options **CSV**

4. Json

Format difficile à lire à l'œil (Utilisé pour les exportation sur un autre support)

Exemple :

```
{ "_format": "wekan-board-1.0.0", "_id": "CBFcruQarQ7sDEc6n", "title": "Test", "permission": "private", "sort": 104, "slug": "test", "archived": false, "createdAt": "2024-01-22T10:05:59.722Z", "modifiedAt": "2024-01-22T12:12:51.629Z", "members": [{ "userId": "ixBf4JxkdvQLiqE5S", "isAdmin": true, "isActive": true, "isNoComments": false, "isCommentOnly": false, "isWorker": false }], "color": "belize", "subtasksDefaultBoardId": null, "subtasksDefaultListId": null, "dateSettingsDefaultBoardId": null, "dateSettingsDefaultListId": null, "allowsSubtasks": true, ..... }
```

⚠ *L'exportation servirait plus d'archives que pour une réel utilisation quotidienne* ⚠ ⚠ *La conversion peut créer des erreurs de chargement de css, il faut juste attendre pour que la conversion se passe correctement* ⚠

Base de données -MongoDB-

Interface console

Pour se connecter a **MongoDB** se connecter au container mongo avec cette commande :

```
docker exec -ti 71fb sh
```

Ensuite, lancer le client de consultation avec la commande suivante et afficher les tables de la BD "**Wekan**" :

```
mongosh
show dbs --Optionnel--
use wekan
show tables
```

Interface graphique

Il est possible d'installer une interface graphique [la MongoDB Compass](#), téléchargeable [ici](#).

Une fois télécharger il faut se connecter à la base de données en configurant les bons paramètres (*Différents moyen d'accès sont possible ici on est passé sur le serveur wekan en ligne pour le stockage*)

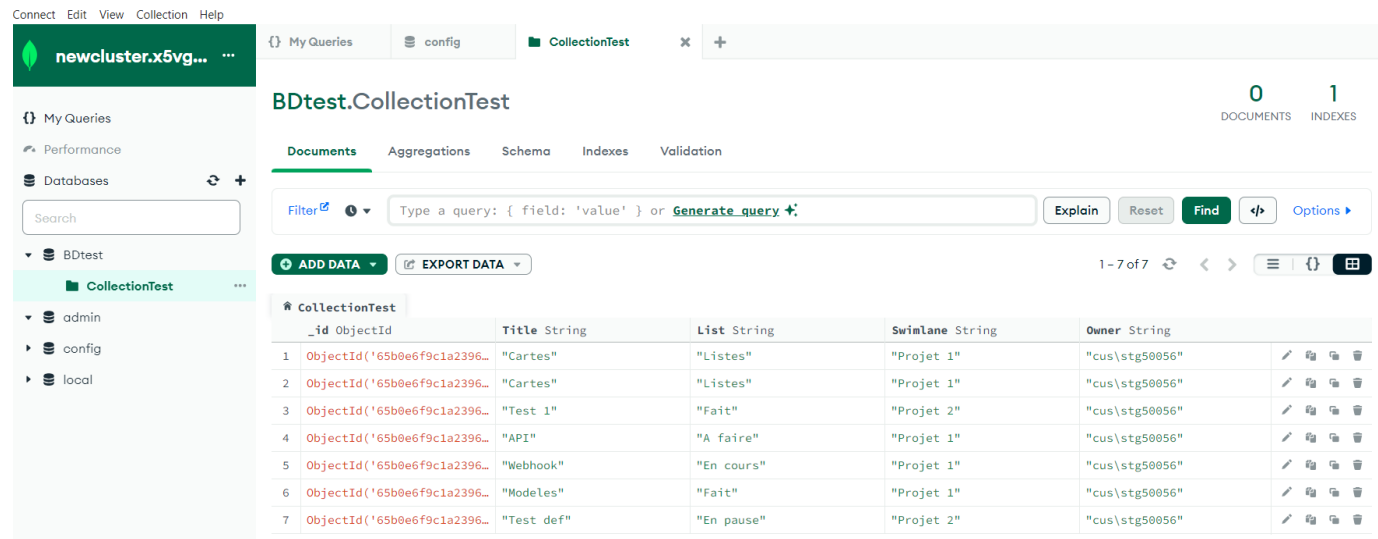
Exemple du code utilisé pour l'analyse (Pas de paramètres nécessaire pour celui-là) :

```
mongodb+srv://admin2:admin2@newcluster.x5vghfa.mongodb.net/
```

```
//admin2:admin2@
```

```
//nomUtilisateur:motDePasse@
```

Voici un exemple du résultat après connexion (*Les données d'une table wekan on été utilisé pour le test*) :



- Une fois sur l'interface graphique il est possible de **créer / modifier / supprimer** / etc.. comme une base de données classique.
- Il est également possible d'afficher **différents type d'affichage** ou encore afficher des **statistiques, des schéma**, etc..
- Tout en état dynamique, c'est à dire **les id se modifient dynamiquement** en si on en supprime un avant tout comme **les tables et colonnes** qui sont sur le même principe.
- Compass est limité dans le nombre de données qu'il peut afficher simultanément par rapport au temps et à la taille mais **possible de le modifier grâce aux paramètres "MaxTimeMs"** (Le temps max pendant laquelle l'application va essayer de charger les documents en millisecondes) et **"Limit"** (La nombre de documents qui vont être chargés).
- Possibilité de mettre des **règles**, qui permettront **d'automatiser certaines actions**.
- Sur **MongoDB** vous pouvez visualiser toutes les informations de sa page wekan en plus des ID et d'autres informations utiles, on peut également les modifier librement.
- ⚠ **MongoDB ne supporte pas les bureaux virtuels** ⚠

Bug images

Une image stocker dans MongoDB :

```

_id: ObjectId('65b372208934c16e7999075b')
length: 781002
chunkSize: 261120
uploadDate: 2024-01-26T08:49:36.708+00:00
filename: "images-of-nature-4.jpg"
contentType: "image/jpeg"
▼ metadata: Object
  boardId: "mGYfYBroL7mn8BnNM"
  swimlaneId: "wDBrY3j jc7MJiJq2o"
  listId: "Hv2X463PmERRHwwMw"
  cardId: "2SK7wFAy8miQmJLoj"
  versionName: "original"
  fileId: "65b36f991e84d072942e305f"

```

- Les images disparaissent après une up de version (-Transition entre la **6.09** et la **6.10** & une fonction dans **Meteore files 2.2.1**-) **Dev Post de la version 6.10**
 - <https://github.com/wekan/wekan/pull/4638>
 - <https://github.com/wekan/wekan/issues/4629>

Attachement et **attachement.files** sont les images / fichier mis après la version 6.10 Et **attachement.files** stocke également les ancienne en plus des images de la nouvelles version

Pour rajouter les ancienne images il faudrait donc crée dans **attachement** et **attachement.files**.

Correction apres **Meteor 2.7.1**.

Solution autre -1- :

-Voir plus ici-

```

# Backing up the database
docker stop wekan-app
docker exec wekan-db mongodump
docker cp wekan-db:dump /some/nice/place
# Stopping the service
docker-compose stop
# Removing the databases persistent file storage
docker volume rm wekan-db
# Replacing the docker-compose.yml with a prepared file with everything needed for
version 7.09
cp docker-compose-7.09.yml docker-compose.yml
# Starting the database and restoring the dump made earlier
docker-compose up wekan-db -d
docker cp /some/nice/place wekan-db:/dump
docker exec wekan-db mongorestore --drop --db wekan /dump
# Starting wekan itself
docker-compose up -d

```

Solution autre -2- :

-Voir plus ici-

Divers

Liens

Liens Wekan

- [Github Wekan](#)
- [Github Wekan API](#)
- [Github Wekan API 2](#)
- [Github Wekan Support Questions](#)
- [Wekan Strasbourg](#)

Liens MongoDB

- [MongoDB Documentation - Compass](#)
- [MongoDB Gestionnaire distant - Compass](#)
- [Compass Download](#)

-
- [Page a s'inspirer](#)
 - [Wekan cloud](#)
 - [Wekan mail test](#)
-

Liens divers :

- [Cours BD Mango -Open Classroom-](#)
-

A faire

- ☒ Analyse des fonctionnalités
 - ☒ Basics
 - ☒ Complexes
- Utilisation des fonctionnalités basics
 - ☒ Création des tableaux
 - ☒ Personnalisations des tableaux
 - ☒ Modification des tableaux
 - ☒ Modification esthétique et graphique des tableaux
 - ☒ Etc...
- Les modèles *-Problème rencontrée-*
 - ☒ Les cartes
 - ☒ Les listes
 - ☒ Les tableaux
- ☒ Les règles *-Base-*
- Les API
 - ☒ Basics
 - ☒ Login / register
 - ☒ Boards

- ☒ Checklists
 - ☒ CardComment
 - ☒ CustomFields
 - ☒ Integrations
 - ☒ Lists
 - ☒ Cartes
 - ☒ Swimlanes
- ☒ Complexes
 - ☒ Users
 - ☒ Les permissions
- ☒ Bug - commande POST
- Les Webhook
 - ☒ Basics -Dont test discord-
 - ☒ Complexes
- Droits / autorisation
 - ☒ Individuelles
 - ☒ Organisation
 - ☒ Équipes
 - ☒ En fonction du tableau
- ☒ Champs personnalisés -Base-
- ☒ Analyse des exports
- ☐ Fonctionnalité d'envoi de mails
- ☒ Analyse interface texte CLI de Wekan
 - Interface inexistante
- Analyse de MongoDB
 - ☒ Interface graphique
 - ☒ Système de fonctionnement
 - ☒ Bug images changement de version
- ☒ Fiche présentation / documentations
- Progression global
 - ☒ 25 %
 - ☒ 50 %
 - ☒ 75 %
 - ☒ 99%

Autres

Versions utilisée :

WeKan ® Version	5.99.0
Version de Meteor	2.2
Version de Node	12.22.10
Version de MongoDB	6.0.12

Versions utilisée 2 :

WeKan ® Version	6.10.0
Version de Meteor	2.2
Version de Node	14.19.0
Version de MongoDB	6.0.13

Raccourcies Wekan :

 Raccourcis clavier

Afficher/Masquer la barre latérale du tableau

w

Filtrer mes cartes

q

Afficher/Masquer la barre latérale des filtres

f

Afficher/Masquer la barre latérale de recherche

/

Retirer tous les filtres

x

Afficher cette liste de raccourcis

?

Fermer la boîte de dialogue

ESC

Auto-complétion des participants

@

Affecter cette carte à vous-même

SPACE

Archiver la carte

c

Bug :

- Les templates disparaissent
 - Uniquement possible de créer des modèles en ayant un compte à authentification classique
- Les images disparaissent après une up de version (-Transition entre la **6.09** et la **6.10** & une fonction dans Meteor files **2.2.1-**) **Dev Post de la version 6.10**
 - <https://github.com/wekan/wekan/pull/4638>

- <https://github.com/wekan/wekan/issues/4629>
- Possibilité de modification de la base de données pour régler sa (*Voir ici*)
- API création non fonctionnel avec POST (*Internal Server Error*)

Bug images

Une image stocker dans MongoDB :

```
_id: ObjectId('65b372208934c16e7999075b')
length: 781002
chunkSize: 261120
uploadDate: 2024-01-26T08:49:36.708+00:00
filename: "images-of-nature-4.jpg"
contentType: "image/jpeg"
▼ metadata: Object
  boardId: "mGYfYBroL7mn8BnNM"
  swimlaneId: "wDBrY3jjc7MJiJq2o"
  listId: "Hv2X463PmERRHwwMw"
  cardId: "2SK7wFAy8miQmJLoj"
  versionName: "original"
  fileId: "65b36f991e84d072942e305f"
```

- Les images disparaissent après une up de version (-Transition entre la **6.09** et la **6.10** & une fonction dans **Meteor files 2.2.1**-) **Dev Post de la version 6.10**
 - <https://github.com/wekan/wekan/pull/4638>
 - <https://github.com/wekan/wekan/issues/4629>

Attachement et **attachement.files** sont les images / fichier mis après la version 6.10 Et **attachement.files** stocke également les ancienne en plus des images de la nouvelles version

Pour rajouter les ancienne images il faudrait donc crée dans **attachement** et **attachement.files**.

Correction apres **Meteor 2.7.1**.

Solution autre -1- :

-Voir plus ici-

```
# Backing up the database
docker stop wekan-app
docker exec wekan-db mongodump
docker cp wekan-db:dump /some/nice/place
# Stopping the service
docker-compose stop
# Removing the databases persistent file storage
docker volume rm wekan-db
# Replacing the docker-compose.yml with a prepared file with everything needed for
version 7.09
cp docker-compose-7.09.yml docker-compose.yml
# Starting the database and restoring the dump made earlier
```

```
docker-compose up wekandb -d
docker cp /some/nice/place wekan-db:/dump
docker exec wekan-db mongorestore --drop --db wekan /dump
# Starting wekan itself
docker-compose up -d
```

Solution autre -2- :
[-Voir plus ici-](#)

Bug API :

Bug	Cause	Solution
API POST ne fonctionne pas	La version est trop récente, le bug est présent	donc passer a une version supérieure afin de supprimer le bug

Versions utilisée (début du bug) :

WeKan ® Version	6.11.0
Version de Meteor	2.2
Version de Node	14.19.0
Version de MongoDB	6.0.13

Pour mettre toujours la dernier version de Wekan :

```
image: ghcr.io/wekan/wekan:latest
```

Parties supplémentaires -KeyCloak-

[Liste de librairies Client et Serveur pour PHP](#)

Authentification centralisée - OAuth2

Le but de cette application est de tester le fonctionnement d'une application **PHP** qui utiliserait le protocole **OAuth2** pour s'authentifier auprès du serveur **KeyCloak**.

Installation des librairies

Utilisation de **composer** :

```
docker run --rm -ti -v `pwd`::/app composer bash
```

Déclaration des librairies à l'aide de la commande **require**.

Exemples :

- **composer search xxxx** pour rechercher une librairie
- **composer require xxxx/xxxx** pour installer une librairie xxx/xxx et l'ajouter au fichier **composer.json**

Lancement d'un container de test

```
docker build .  
docker run --rm -ti -p 8086:80 <id image>
```

Penser a faire des rafraîchissements

```
git pull
```

Divers

- Keycloak = <http://infcau104:8091/>
- Page PHP = <http://infcau104:8086/>

Liens

Liens Keycloak

- [Templates Keycloak](#)
- [GitHub stevenmaguire](#)
- [Module OpenId](#)
- [Module OpenId Github](#)
- [Facile Manager](#)

A faire

- ☒ Analyse de Keycloak
 - ☒ Création d'une page de connexion Eurométropole
- ☒ **Configurer un nouveau client OAuth dans KeyCloak :**
 - ☒ Aller dans la section "Clients" et créer un nouveau client pour l'application PHP.
 - ☒ Configurer les paramètres du client.
- ☒ **Choisir une librairie PHP pour OAuth2 :**
 - ☒ Installer la librairie choisie avec Composer.
- ☒ **Mettre en place le code PHP :**
 - ☒ Créer un fichier PHP pour gérer le flux d'authentification OAuth2.
 - ☒ Configurer le client OAuth2 avec les informations client de KeyCloak.
 - ☒ Implémenter la logique pour rediriger l'utilisateur vers le formulaire d'authentification KeyCloak.
- ☒ **Intégrer le formulaire KeyCloak :**
 - ☒ Utiliser les styles CSS pour personnaliser le formulaire KeyCloak.
 - ☐ Accéder au formulaire KeyCloak à l'URL appropriée.
- ☒ **Gérer les résultats de l'authentification :**

- ☒ Gérer la redirection après une authentification réussie.
 - ☒ Afficher une page différente en cas de succès.
 - ☒ Gérer le cas d'échec en affichant un message approprié.
 - ☒ Fiche présentation / documentations
 - Progression global
 - ☒ 25 %
 - ☒ 50 %
 - ☒ 75 %
 - ☐ 99%
-
-
-

Info Dives

Arboréscences :

-  conf
-  css
-  documents
-  error
-  js
-  result
-  index.php
-  login.php

Sur infcau104, voir plus.. -temp-

Divers - Perso

Connexion a keycloak sur docker

```
ssh -nom-@infcau104
cd ../../var/data/keycloak
docker-compose up
```

Connexion au formulaire sur docker

-Aller ici-

```
ssh -nom-@infcau104
cd ../../var/data/authentication-centralisee/Formulaire_Oth0
```

-Suite ici-

Aller sur la page Keycloak infcau :

- <http://infcau104:8091>
- Connexion admin avec :

- **Id** : *admin*
 - **Psw** : *password*
- Connexion Client pour test avec :
 - **Id** : *testclient*
 - **Psw** : *password*

FREYERMUTH Matys