

Conduite et gestion de projet

Saisie d'un protocole thérapeutique pour un
épileptique

MI3App

MI2A Management

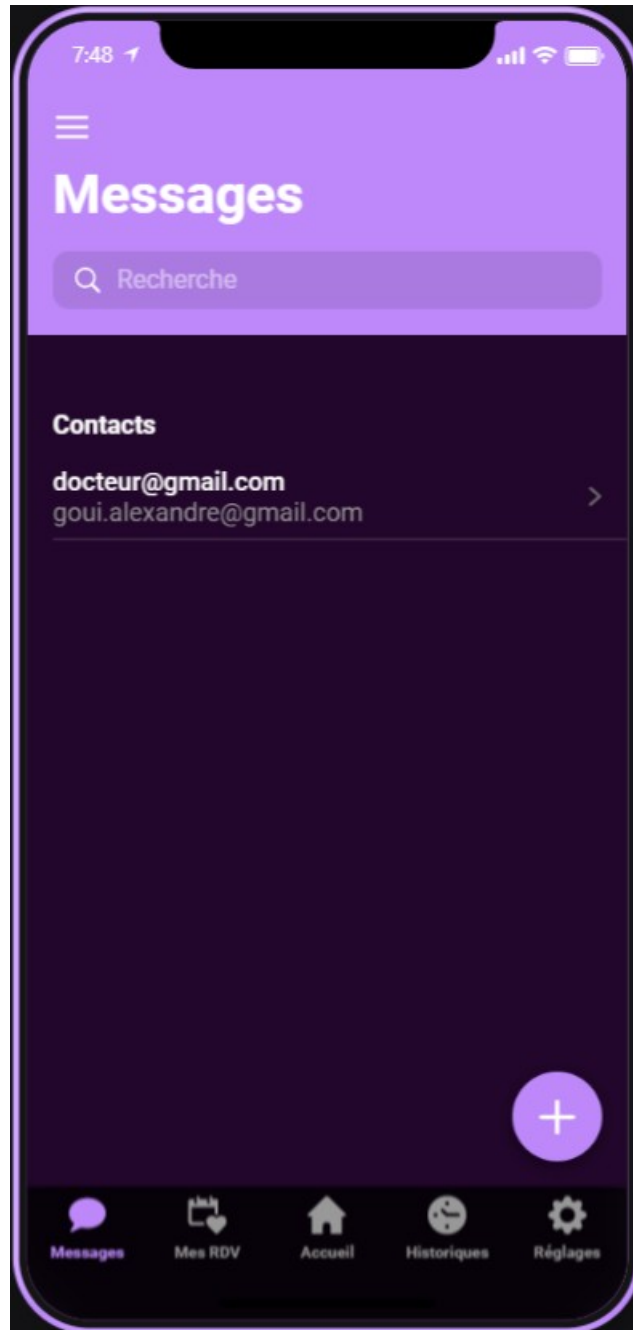
Alexandre GOUJ
Michel KADDOUH
Alexis CHANDARA
Ismael DHKISSI

2020 – 2021

Sommaire

Messages	3
Mes RDV	5
Accueil	6
Historiques	9
Réglages	12
Notification	14

Messages



Afin que deux personnes puissent communiquer de façon privée, il suffit de créer, dans le Google Sheets, un page ayant la colonne de l'**utilisateur 1**, l'**utilisateur 2** et un **Row ID** pour chaque ligne.

ChatUserUno	ChatUserDeux	Row ID
exemple@gmail.com	exemple@gmail.com	6DIM037rRh-2WN8huZ2
goui.alexandre@gmail.com	docteur@gmail.com	hv0-WNsfsOO2LZh1UJI

ChatContact[tab] RDV[tab] 1 Accueil[tab]

Maintenant, sur GlideApps, on crée une **Inline List** visible par l'**utilisateur 1** ou l'**utilisateur 2**, chaque élément correspondra à une discussion entre l'utilisateur 1 et l'utilisateur 2, on y ajoute une section **Comments** visible pour ces deux-là et c'est fini.

Mes RDV

Accueil

Historiques

Réglages

MENU

Tchat

Les médicaments

Bilan cartographique

SCREEN DATA

Comments Row ID

Rich Text bg_color

Aucun commentaire

Ajouter un commentaire

VISIBILITY

Show component when ChatUserUno is signed-in user

or ChatUserDeux is signed-in user

Add Condition

Mes RDV



Pour cet onglet, rien de difficile, on utilise seulement les fonctionnalités de GlideApps.

A ne surtout pas oublier de rendre les colonnes visibles que ceux qui les ajoutent (ajouter une colonne **Owner** qui prend comme valeur l'e-mail de l'utilisateur).

Accueil



Premièrement, pour pouvoir afficher les médicaments choisis de l'utilisateur, on ajoute une colonne **Traitements** qui prend juste deux valeurs : VRAI ou FAUX.

Ensuite, on crée une colonne **Relation** sur **GlideApps** qui relie la colonne **Traitements** et les médicaments choisis par l'utilisateur.

The first screenshot shows the 'Traitements' column configuration in GlideApps. The 'Label' is set to '... à prendre'. The 'Column Type' is set to 'Relation'. The 'CONFIGURATION' section shows 'Traitements ...' selected, with 'Traitements(à cocher)' and 'Actuellemen' as options. The 'Match multiple' checkbox is checked. The second screenshot shows the resulting data table with columns: Row ID, Actuellemen, Nom du traitement, and Description. The 'Actuellemen' column is highlighted with a red box and labeled 'Colonne spécifique à l'utilisateur'.

Row ID	Actuellemen	Nom du traitement	Description
GYa7alaySG-pANcsypnV	<input type="checkbox"/>	Dépakine Chrono 500mg	Ce médicament
8-VH.D6GY2q748Hlwz7	<input type="checkbox"/>	Lalala	C'est utile, faut p
E8RHJJtWSpOvV7w3bQ	<input type="checkbox"/>	Witamin	C'est utile, faut p
62JRueiVSlqZP6QAsX4G	<input type="checkbox"/>	Finifini	C'est utile, faut p
pIlqdBdLRkCgQfBGVfrVE	<input type="checkbox"/>	Poipoi	C'est utile, faut p
5Rz5fcG7TsiEK5i-9lw-Ig	<input checked="" type="checkbox"/>	Bonne nuit	C'est utile, faut p

Deuxièmement, pour afficher le nombre de crise(s) par mois, on a besoin d'une colonne qui prend comme valeur la date du mois et de l'année, et une copie des valeurs de la page qui calcule le nombre de crise(s) par mois pour chaque utilisateur, sur une autre page de la **Google Sheets**.

Il nous était obligé de le faire ainsi car il n'y avait pas moyen de le faire avec des colonnes **Relations** de **GlideApps**, en effet, les colonnes **Relations** ne permettent qu'une seule configuration, hors nous en avons besoin de plusieurs : Owner, la date du mois.

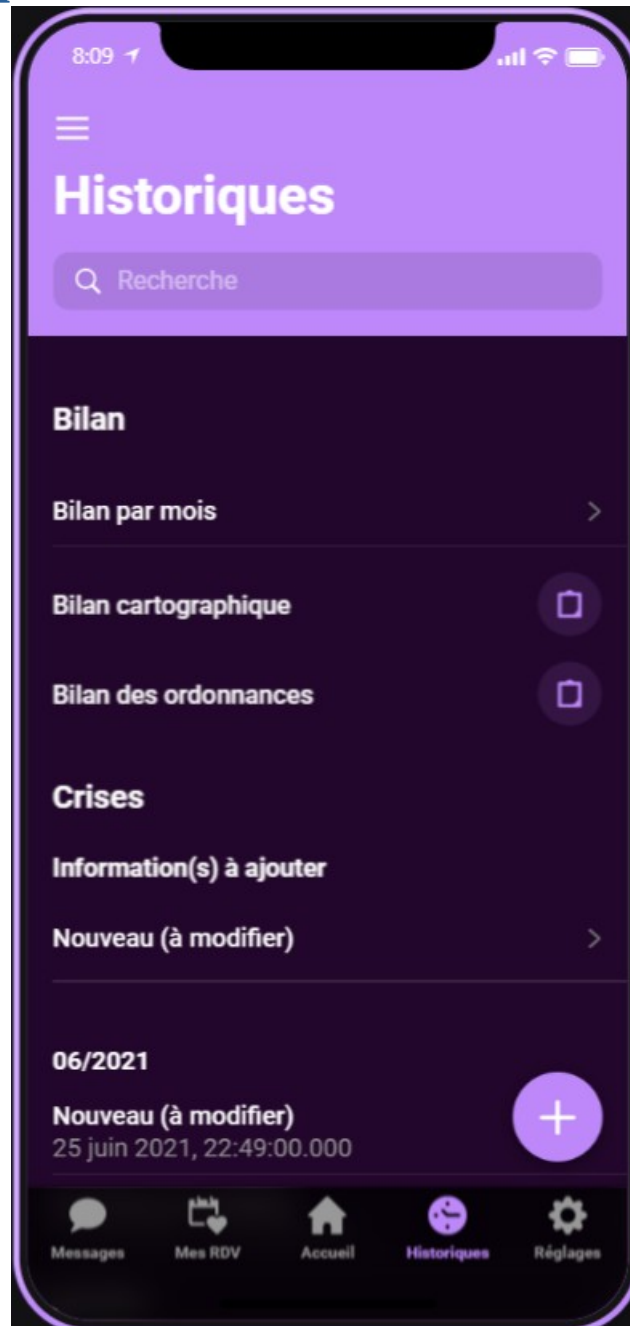
Et dernièrement, on va afficher quelques actualités en relation avec l'épilepsie. Pour cela, on crée une colonne **Date** dans le Google Sheets, qui prend comme valeur la date actuelle. Ensuite, on crée une colonne **Relation** sur **GlideApps** qui relie cette colonne avec celle d'une autre page.

The image shows the process of setting up a GlideApp column. The top part shows a Google Sheet with the formula `=AUJOURDHUI()` in a cell, which is highlighted with a red box. Below this, a table shows the data for the 'Date' column, with the value 'dimanche 18 avril 2021' highlighted in a red box. The bottom part shows the GlideApp interface with the 'Date' column configuration. The 'Label' is set to 'Today's News', the 'Column Type' is 'Relation', and the 'CONFIGURATION' section shows the 'Aujourd'hui' column selected, with the 'Actus(→ Accueil)' column selected for the 'Date' field. The 'Match multiple' checkbox is checked.

	B	C	D
Etat	noMoreUsed(forNow)	Aujourd'hui	CurrentHour
AI	noMoreUsed(forNow)	dimanche 18 avril 2021	19:42
UX	noMoreUsed(forNow)	dimanche 18 avril 2021	19:42

Row ID	Date	Actualité	Description	bg_color
BoHs5rMQQdq2sLDpyT	dimanche 18 avril 2021	https://drive.google.com/file/d/1...	Les bons gestes	<pre>
a.lfGpyswR5WVfDmRZW	dimanche 18 avril 2021	https://www.youtube.com/watch...	Une crise d'épilepsie vue	<pre>

Historiques



Cet onglet sert à plusieurs fonctionnalités de l'application, premièrement répertorier les crise(s), avoir un bilan du nombre de crise(s) par mois, un bilan sur le nombre total de crises et un bilan des ordonnances de l'utilisateur.

→ Pour répertorier les crises de l'utilisateur, on a eu plusieurs actions à mener

Tout d'abord, on récupère les crises détectées par l'application faite par le groupe « **la team qui ne répond jamais** » à l'adresse suivante :

<https://docs.google.com/spreadsheets/d/1I7W-Fo4P2LnWUysDLfkXIAUoMdZo4GruNYtqzOlcYpI/edit#gid=1273572047>

Ensuite, on récupère ces données sur une autre **Google Sheets**

(<https://docs.google.com/spreadsheets/d/1kA90NXfcnW4QN-EhDSNyFZfUJfX0TIFC6tHMsIfNBd8/edit#gid=0>), puis on traite les données afin qu'ils puissent être utilisés par GlideApps.

Et enfin, on les récupère sur notre **Google Sheets** principale, utilisé pour l'application **MI3App** de **GlideApps**.

Pour les données ajoutées par l'application, nous avons converti la date, car le format utilisé n'est pas très visuel (ex. : **<2021-06-25T22:49:00.000Z>**), en un meilleur visuel (en gardant l'exemple précédent, **<25 juin 2021, 22:49:00.000>**)

2021-06-25T22:49:00.000Z	25 juin 2021	22:49:00.000Z	22:49:00.000	25 juin 2021, 22:49:00.000
2021-06-19T23:26:00.000Z				19 juin 2021, 23:26:00.000
2020-08-15T23:43:00.000Z				15 août 2020, 23:43:00.000
2021-04-18T00:29:00.000Z	18 avril 2021	00:29:00.000Z	00:29:00.000	18 avril 2021, 00:29:00.000

→ Pour le bilan du nombre de crise(s) par mois

Pour obtenir le nombre de crise(s) par mois, nous avons créé une nouvelle page sur notre **Google Sheets**, puis on a effectué une requête sur la page stockant les crises.

La requête consiste à récupérer l'utilisateur de la crise, la date du mois et de l'année, le nombre de crises et le mois et l'année (en un format différent).

B2									
1	Bilan	Owner	Date	Nb crises	ForSortingPurpose				
2	Bilan par mois			count					
3	Bilan par mois	17zrami@gmail.com	avril 2021	2	04/2021				
4	Bilan par mois	exemple@gmail.com		1					
5		goui.alexandre@gmail.com	#VALUE!	1	#VALUE!				
6		goui.alexandre@gmail.com	août 2020	1	08/2020				
7		goui.alexandre@gmail.com	avril 2021	1	04/2021				
8		goui.alexandre@gmail.com	juin 2021	2	06/2021				

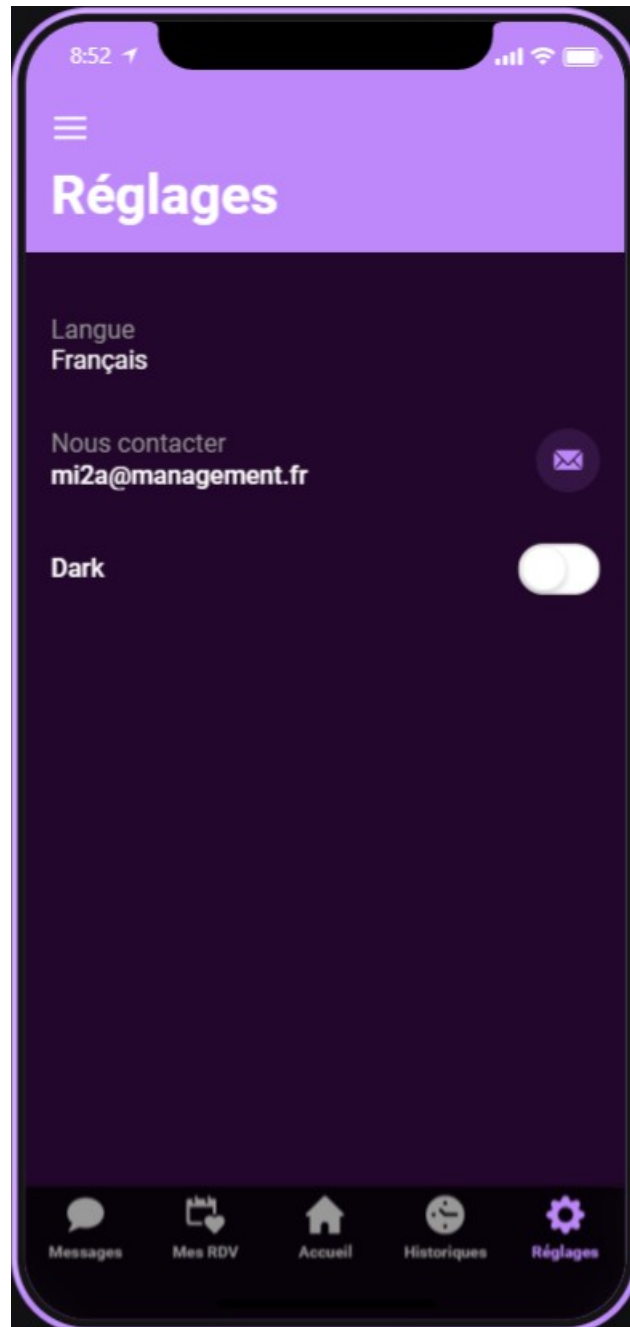
→ Pour le bilan sur le nombre total de crises

Nous avons uniquement utilisé les fonctionnalités de GlideApps.

→ Pour le bilan des ordonnances

Pour le bilan des ordonnances, nous avons créé deux pages sur notre **Google Sheets** : une qui va se remplir lorsque l'utilisateur ajoute des médicaments à son ordonnance, et l'autre qui va obtenir une copie de la première mais uniquement les valeurs uniques pour ne pas avoir de doublons.

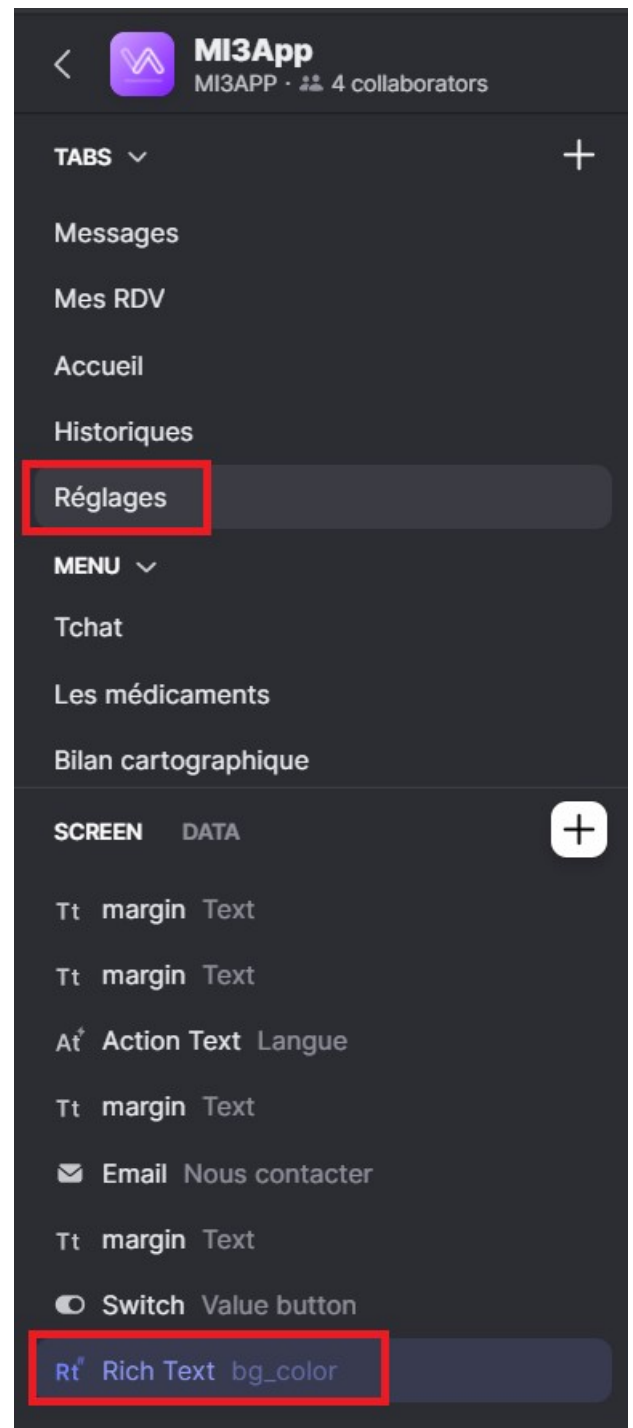
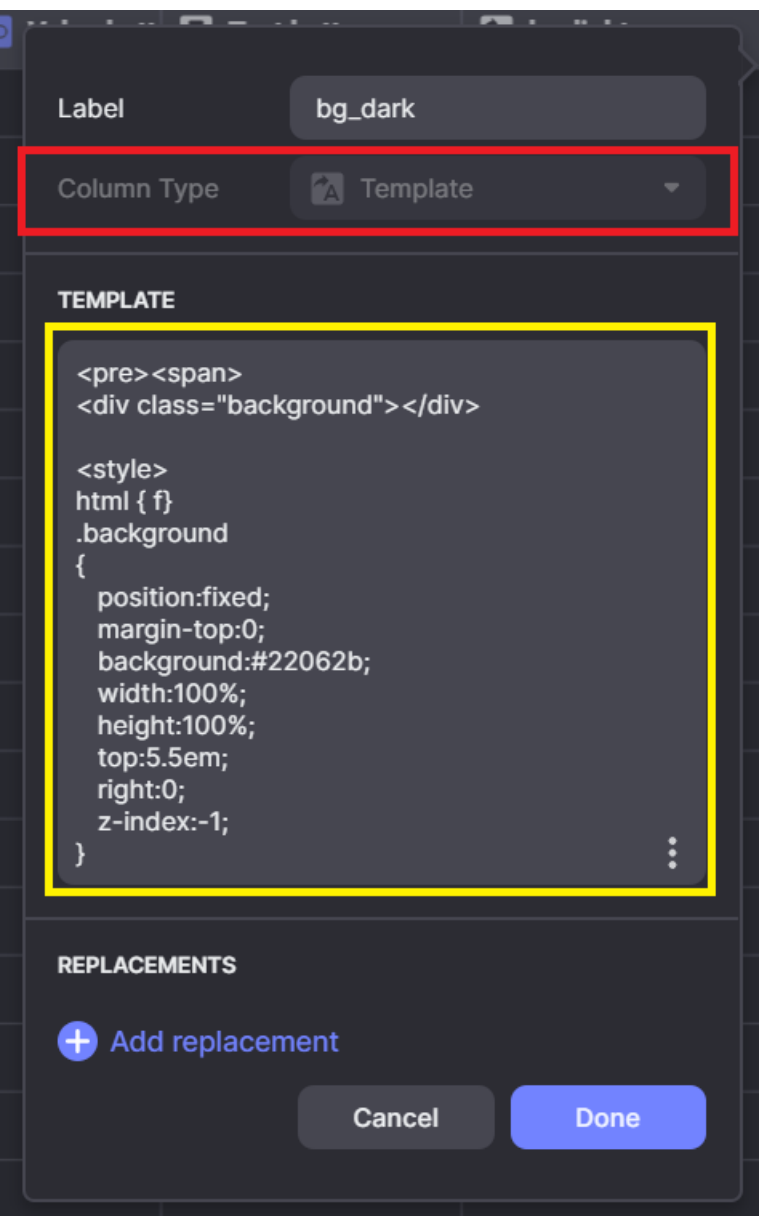
Réglages



Pour cet onglet, nous avons utilisé uniquement les fonctionnalités de **GlideApps**, sauf pour l'option de la couleur du background de l'application (Dark ou Light mode).

Afin de pouvoir changer la couleur du background de l'application, nous avons dû ajouter du code CSS. Celui-ci consiste à créer un block qui prend toute la taille de l'écran à une couleur définie.

Une chose importante, la colonne créée doit être de type **Template** et pour pouvoir faire comprendre **GlideApps** que c'est un code, nous l'ajoutons en tant que **Rich Text**.



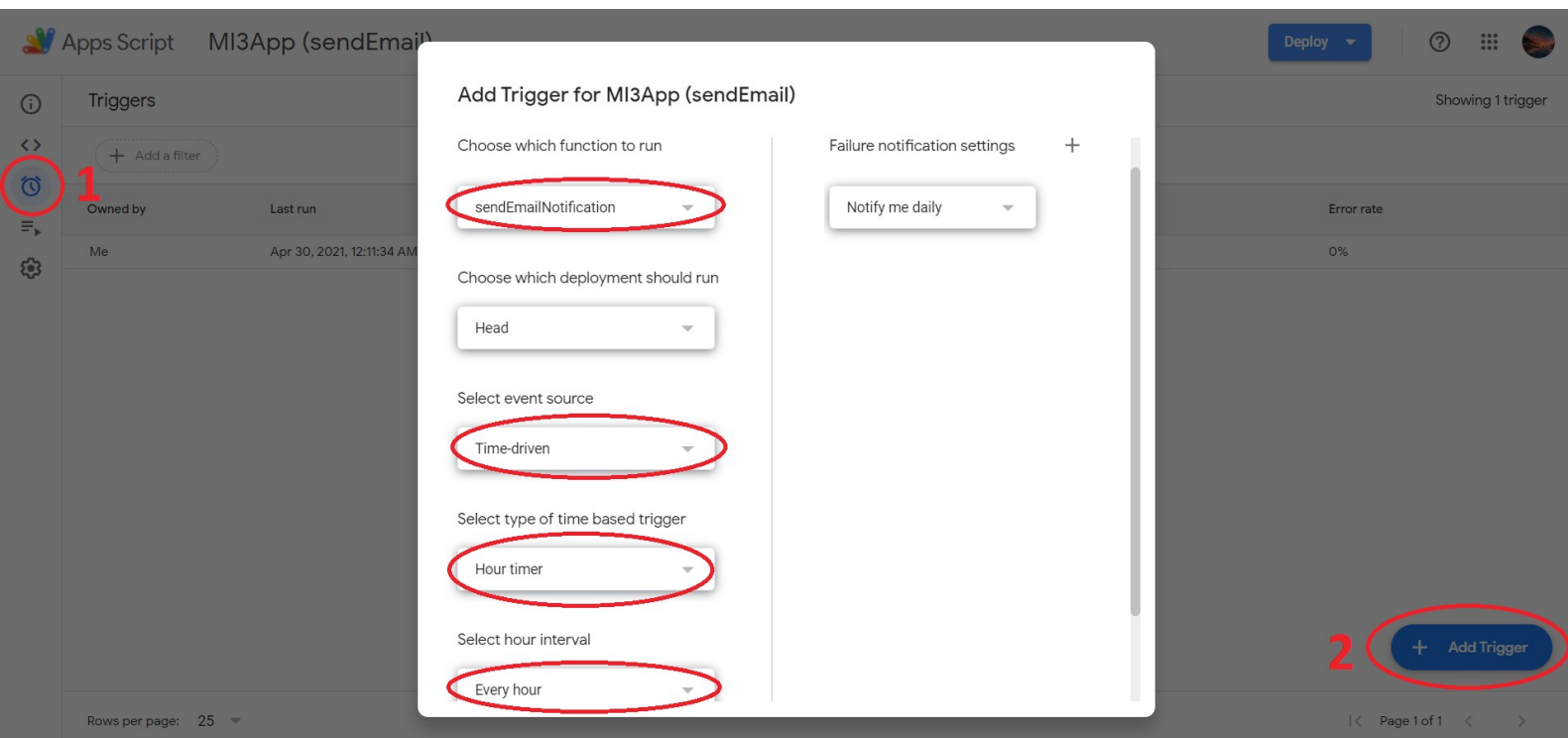
Notification

Afin de pouvoir être notifié lorsque l'utilisateur doit prendre un médicament, nous avons créé un script qui utilise les valeurs de la page du bilan des ordonnances.

Ensuite, nous avons créé un **Trigger** sur **Google Apps Script** en *Time-based*, il s'activera toutes les heures afin d'envoyer un email selon conditions.

```
1 function sendEmailNotification()
2 {
3   var ss = SpreadsheetApp.getActiveSpreadsheet() ;
4   var sheet = ss.getSheetByName('BilanOrdonnance[tab]');
5
6   var subject = "[MI3App] Il est l'heure de prendre votre médicament !" ;
7   var message ;
8
9   var toSend ;
10  var email ;
11  var matinMidiSoir ;
12  var currentHour ;
13
14
15  var medoc ;
16  var nbPrise ;
17  var methodePrise ;
18
19
20  for (var i=2 ; i<14 ; i++)
21  {
22    toSend = sheet.getRange( i,12 ).getValue() ;
23
24    if ( toSend == 'yes' )
25    {
26      email = sheet.getRange( i,1 ).getValue() ;
27      matinMidiSoir = sheet.getRange( i,6 ).getValue() ;
28      currentHour = sheet.getRange( i,11 ).getValue() ;
29
30      methodePrise = sheet.getRange( i,7 ).getValue() ;
31      nbPrise = sheet.getRange( i,8 ).getValue() ;
32      medoc = sheet.getRange( i,3 ).getValue() ;
33
34      if ( matinMidiSoir == "Matin" || matinMidiSoir == "matin" )
35      {
36        if ( currentHour == "08" )
37        {
38          message = "Bonjour !\n\n" ;
39          message = message + "N'oubliez pas de prendre : \n\t"
40          message = message + nbPrise + "x " + medoc + " " + methodePrise ;
41          message = message + "\n\nPassez une bonne journée !"
42
43          MailApp.sendEmail(email, subject, message) ;
44        }
45      }
46
47      else if ( matinMidiSoir == "Midi" || matinMidiSoir == "midi" )
48      {
49        if ( currentHour == "12" )
50        {
51          message = "Bonjour !\n\n" ;
52          message = message + "N'oubliez pas de prendre : \n\t"
53          message = message + nbPrise + "x " + medoc + " " + methodePrise ;
54          message = message + "\n\nPassez une bonne journée !"
55
56          MailApp.sendEmail(email, subject, message) ;
57        }
58      }
59
60      else if ( matinMidiSoir == "Soir" || matinMidiSoir == "soir" )
61      {
62        if ( currentHour == "19" )
63        {
64          message = "Bonjour !\n\n" ;
65          message = message + "N'oubliez pas de prendre : \n\t"
66          message = message + nbPrise + "x " + medoc + " " + methodePrise ;
67          message = message + "\n\nPassez une bonne journée !"
68
69          MailApp.sendEmail(email, subject, message) ;
70        }
71      }
72    }
73  }
```

Script Google Apps Script



Création d'un Trigger Google Apps Script