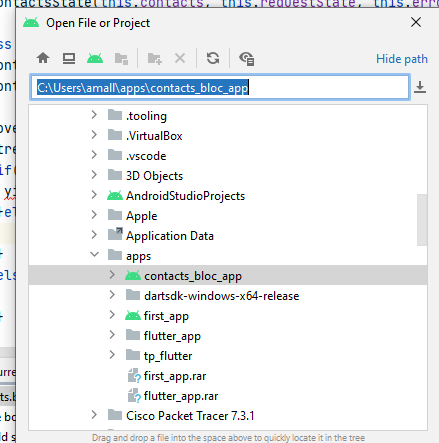
**Compte-Rendu :Projet Développement Mobile**

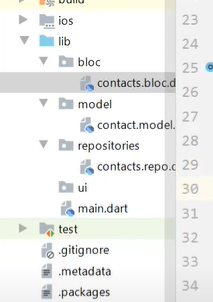
**Part 10 - Développement Mobile Flutter - Blocs - GET\_IT DI - Contacts Messages App - Part 1**

**1erephase:**

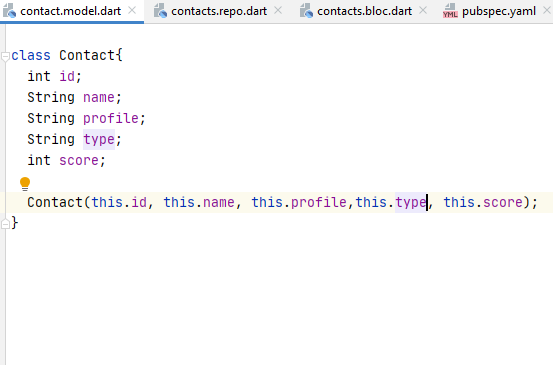
Dans le dossier apps j’ai pu ajouter le fichier android :



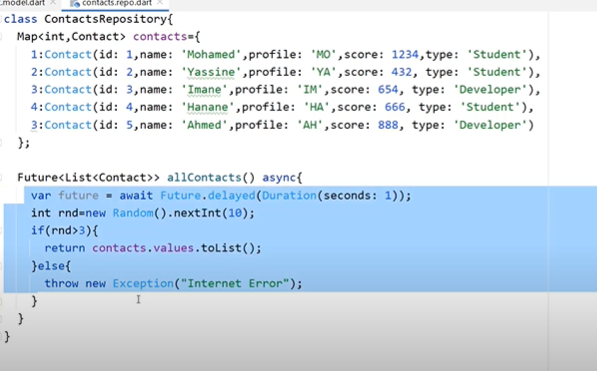
* Ensuite dans le dossier lib j’ai ajouter plusieurs package pour pouvoir après mieux les géré ;

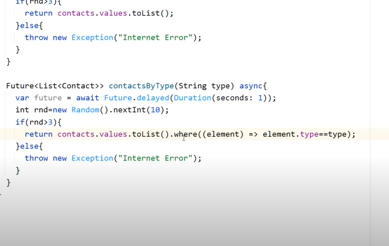


* Dans le dossier model il y a la partie variable dans la class Contacts

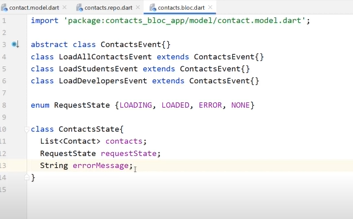


* Puis dans le fichier contacts.rep.dart j’ai crée une liste de contacte



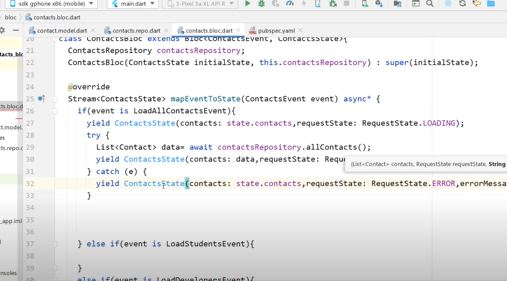


* Dans la partie bloc j’ai créé un fichier contacts.bloc.dart j’ai créé une class abstrait ou il y’a trois évènement avec leur paramètre ,j’ai eu besoin de mon state ou je vais gère une liste de contacts également pour savoir l’état ;

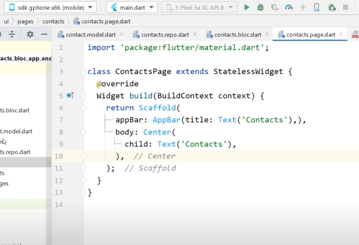




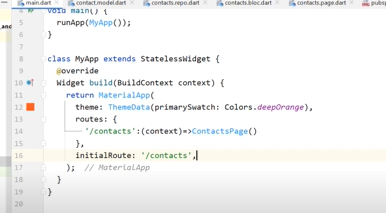
* Ensuite j’ai récupéré la liste des contacts depuis la méthode



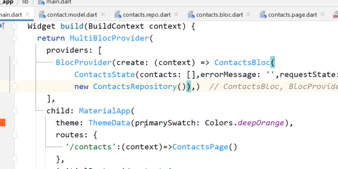
* Dans la partie ui j’ai créé une package dans lequel il y a deux dossier contacts et message



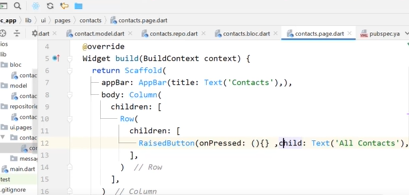
* Je suis passe a main.dart ou j’ai fait la classe Myapp



* Après ceci j’ai ajouté les éléments au fur et mesure, puis j’ai utilisé Bloc Provider MultipBlocProvider



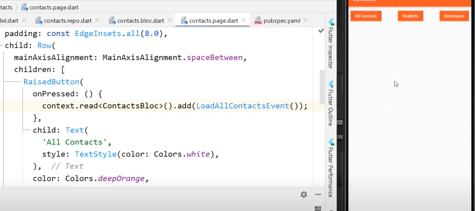
* Dans contacts.page j’ai ajoute un classe ContactsPage pour créer le contenue :



Pour les boutons :



* Puis depuis l’objet context pour émettre un évènement dans les bouton



* Puis j’ai fait appeler à BlocBuilder



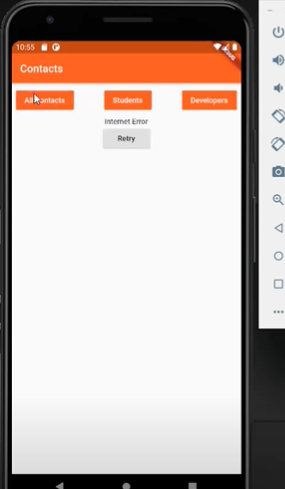
* J’ai retourné CircularProgramIndiacator ou j’ai fait un if :



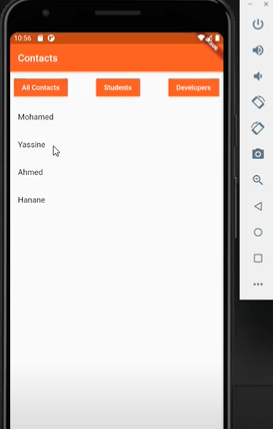
* Dans ce cas-là on va recharge la liste des contacts avec LOADED



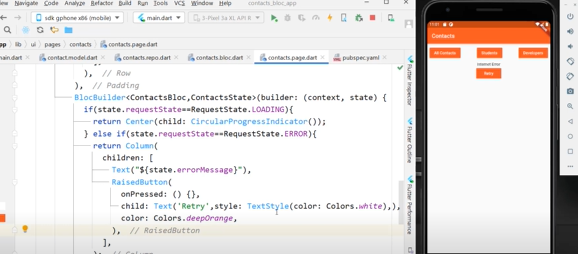
* Voici l’aperçue :



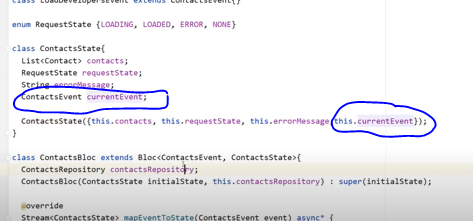
* Sachant que notre liste et dans un élément colonne d’où il faut l’enveloppe avec expanded



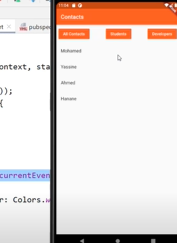
* Ensuit j’ai change le style du bouton :



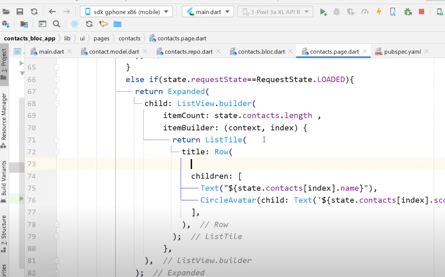
* Quand il y a erreur c’est qu’il faut relance l’évènement j’ai eu besoin d’ajouter dans le state contatsEvent currentEvent comme ça quand on va gère ici la sortie

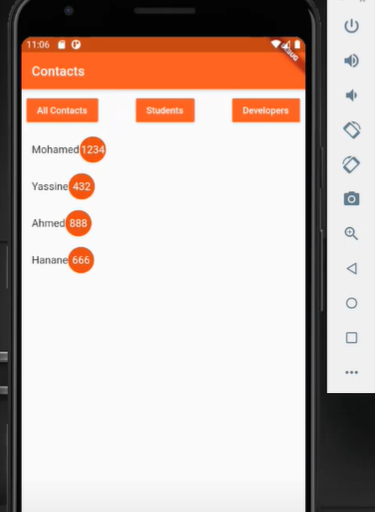


* Voilà comme ça j’ai pu règle le problème du message d’erreur

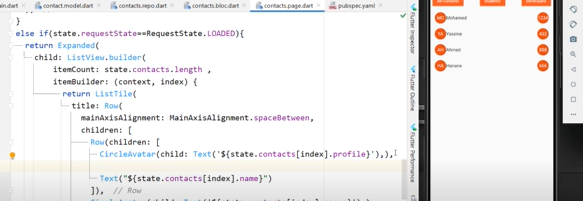


* Puis j’ai complété la liste des contacte plus un rond de score :

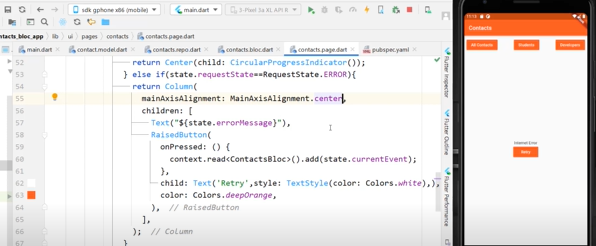




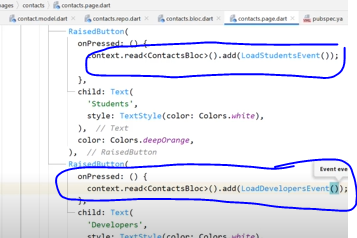
* Pour regroupe les éléments :



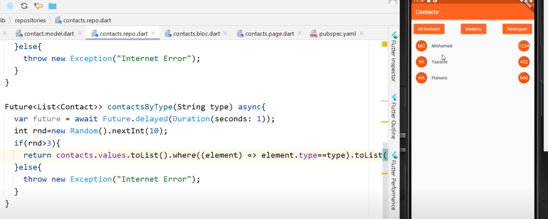
* J’ai enveloppé column par Expanded pour centre le message d’erreur

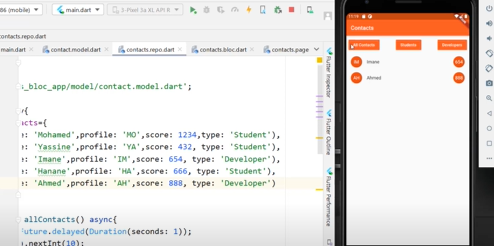


* Maitenant pour charge la liste des Students avec LoadStudentsEvent() et developer avec LoadDevelopersEvents :

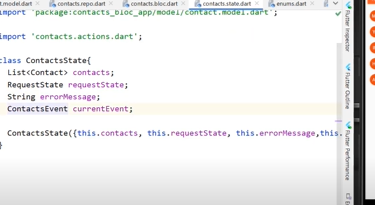


* Au niveau du bloc j’ai juste fait un copie colle et j’ai modifie juste le nom LoadallContacts par LoadStudentsEvent et l’autre aussi
* Voici Pour Student et developer faudra ajouter .to liste dans le dossier contacts.repo.list:

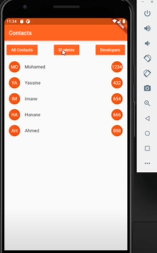
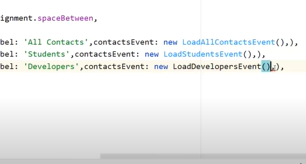


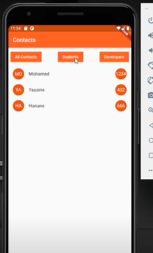


* Dans le bloc j’ai regroupé chaque chose dans un fichier sépare : contacts.action.dart et contacts.state.dart

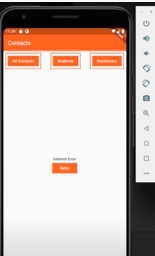


* Puis après pour plus passe par un bouton pour accéder autre j’ai créer un fichier ou je relier avec bar Button

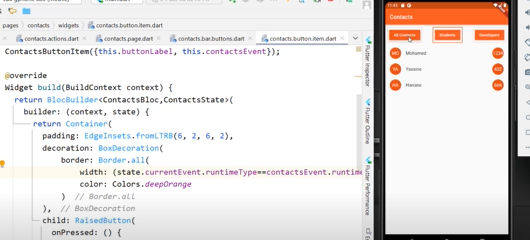




* Comme désigne J’ai mis un container



* Avec runtime s’as permis que quand j’appuis sur un bouton que la bordure soit encadre que sur lui :



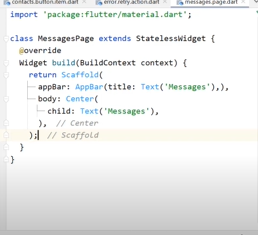
* Voila sa c’était pour la liste des contacts ;

¨Pour résume ;Maintenant j’ai un exemple d’application qui es mieux structure dans laquelle j’ai essaye de sépare la partie logique ui

* Ou on trouve la partie package model ou j’ai créé la class Contact
* Dans repository ou j’ai géré les donnes dans collection de type Map et j’ai créé des méthodes des liste de contacts par type, aussi j’ai utilisé une temporisation avec Future.delayed
* Dans la partie Bloc sous trois fichiers : ou j’ai créé des actions, des States et des blocs
* Pour la partie enum ou j’ai fait appeler à RequestState LOADING ,LOADED,ERROR, et NONE
* Ensuite je suis passé a ui :
* Ou il y a des pages ; PageContacts
* Dans la méthode **main** j’ai fait appeler à la dépendance **GETIT** qui permet l’injection et de dépendance base sur flutter

**Part 11 - Développement Mobile Flutter - Blocs - GET\_IT DI - Contacts Messages App - Part 2**

* Apres avoir passé de la page contacte maintenant je suis passe à la page message .



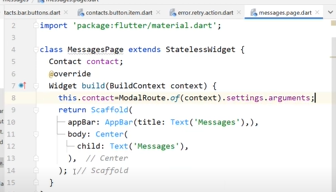
* Dans main j’ai ajouté une Route



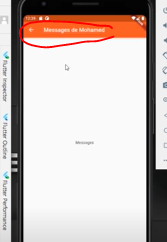
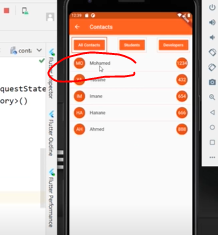
* Dans contacte liste j’ajoute onTap :



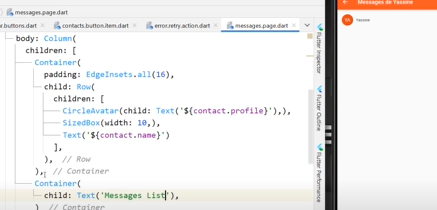
* Dans la page message j’ajouter contacte comme ça pour faire appeler ou je récupère les arguments qui a dans la route



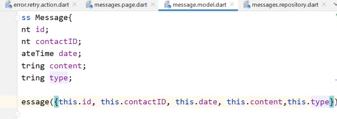
* Ensuite j’ai sélection un contacte pour accède a la partie message



* Dans messages on ajoute une liste de message reçue



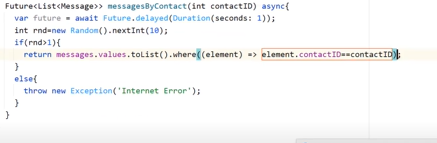
* Dans model j’ai créé une class



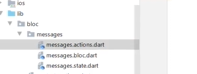
* Puis dans repository ou on créer une class messageRepository pour créer des données



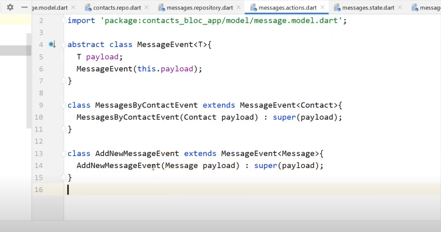
* Créer la méthode Future pour retourne une liste de message



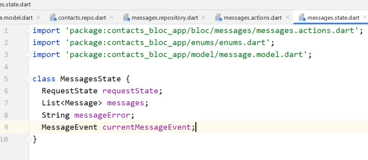
* Après je suis passe à bloc : ou j’ai créer des sous package de message
* Action
* Bloc
* State



* Dans action j’ai créé une class abstrait ou il y’a trois évènement avec leur paramètre, j’ai eu besoin de mon state ou je vais gère une liste de contacts également pour savoir l’état ;



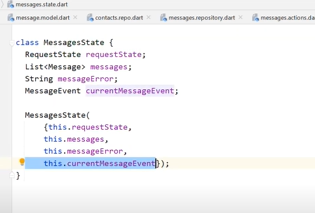
* Dans State :



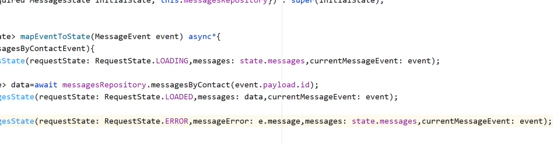
* Dans Bloc ou initial state est obligatoire  :



* Puis l’ajout de currentMessege







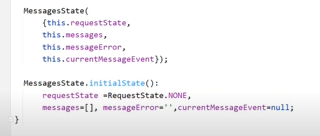
* Dans la partie ui j’affiche la liste des contacts avec le service GetIt



* Pour affiche le message



* Pour initialise le state je l’ai créé dans le state :



* En retournr dans main je remplace null par MessageState.InitialState

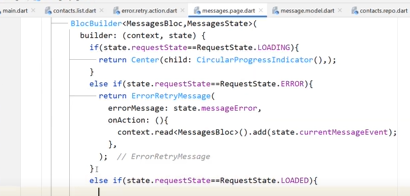


* Dans messagePage j’affiche la liste des messages il va falloir émettre l’évènement

Avec **MessagesByContactEvent**



* J’ai fait appele a BlocBuilder ou il y a une listeview

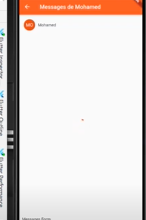


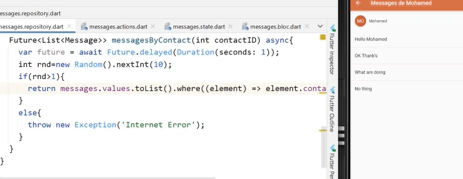


* Le else suivant retourne un container
* L’expanded doit être en dehors de blocbuilder

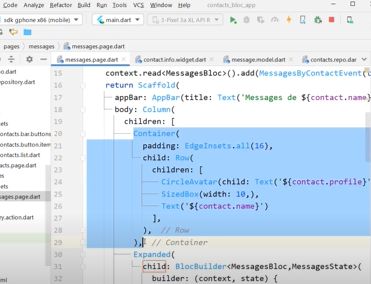


* Je choisie un contacte il génère la page message

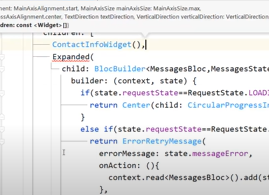




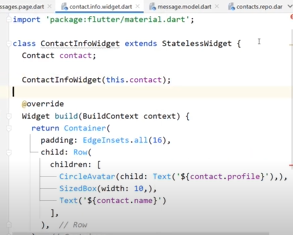
* Changement de désigne ; pour cela j’ai créé un nouveau package widget pour les informations sur le message :
* Dans la page message :



Remplace par :



* Dans widget :



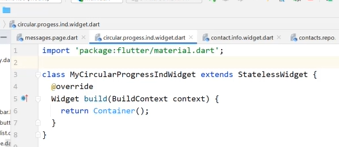
* Pour cela j’ai ajouté a Contacte Widget :



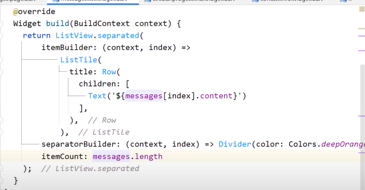
* Pour celui-là j’ai créé des widgets séparer :

* Voilà son widget



* Dans les widgets j’ajoute une liste pour cela on effectue les mêmes étapes :

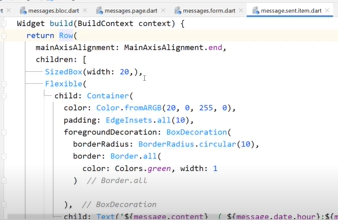


* Le composant dans la page :

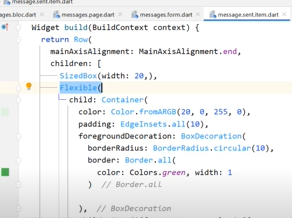


* Ensuite il y a eu pour la forme aussi pour ceci message item qui va permettre de mieux gère entre les message reçu et envoyé

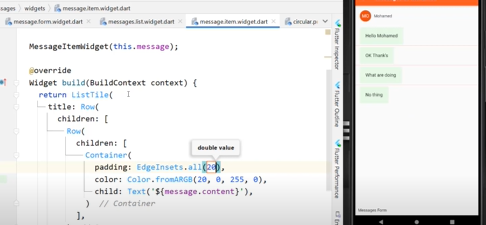
Pour les envoyés :



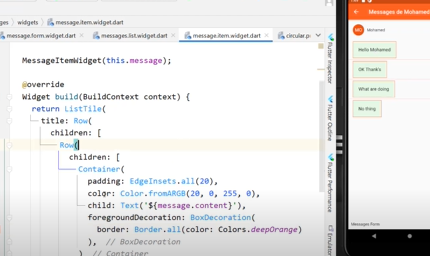
**Flexible** permet si le texte est trop long de l’ajuste



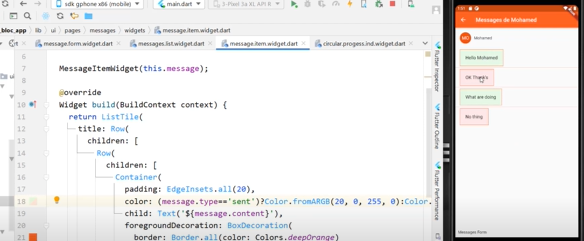
* Ensuite pour donne plus un aspect de conversation j’ai mis un CONTAINER



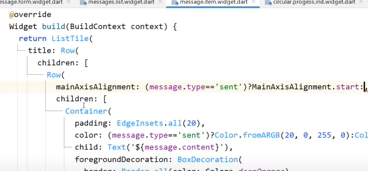
* Pour un peu plus de décoration j’ai fait appeler a foregroundDecoration

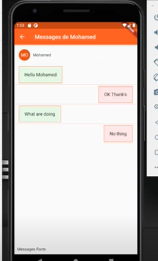


* Encore pour plus de désigne j’ai mis ceux qui sont envoyé avec une couleur et ceux reçu d’une autre :

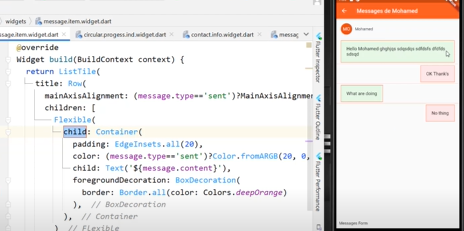


* Ensuite je suis passe a l’alignement tous ce qui es sent a droit et reçue à gauche pour cela j’utilise

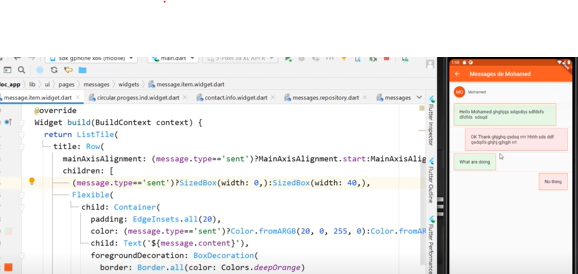


* Quand le texte déborde on utilise Flexible pour mieux gére le surplus du text puis j’ai utilise sizeBox pour l’espace

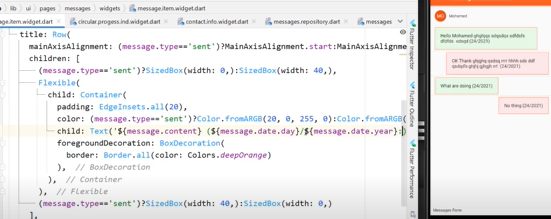


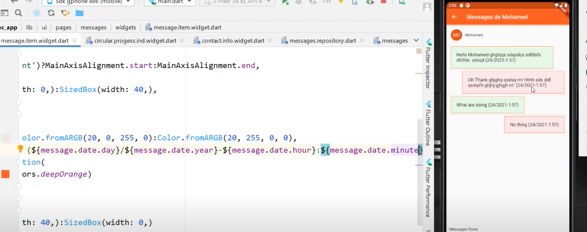


* J’ai aussi fait la meme chose pour le sent

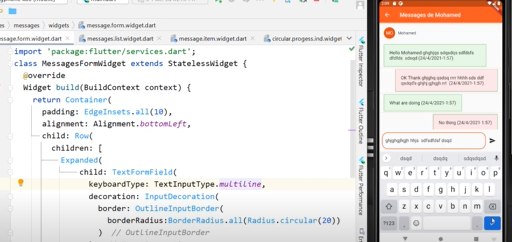


* En plus du text j’ai ajouter la date du message et aussi l’heure

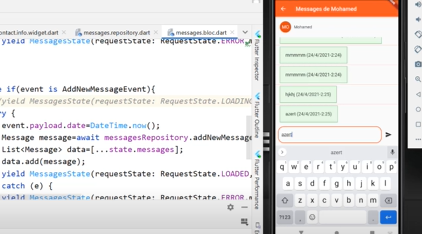
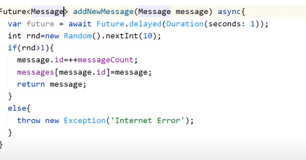




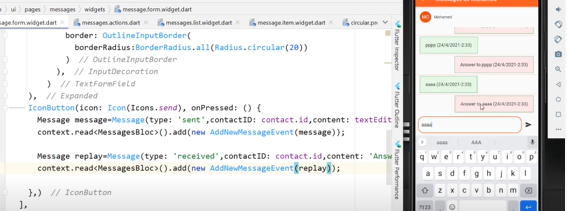
* Pour passe un message :la barre a message



* Puis j’ai traite le message au niveau des blocs



* Auto-received :



* Pour cette partie j’ai pas utilise le **statefullWidget** mais les **stateLessWidget** et grâce au service **get it** j’ai pu sépare la logique présentation avec la logique métier et il y a la communication entre la logique **ui** et la partie métier qui se fait **asyc**

**Part 13 - Développement Mobile Flutter - Blocs - GET\_IT DI - Contacts Messages App - Part 4**