Flutter Pizzeria - TD 4

# Le panier

Nous avons tout mis en place pour pouvoir commander des pizzas. Nous allons donc faire le panier.

## Différentes stratégies

Le panier va être disponible sur toutes les pages. Il faut donc que l’objet Panier soit disponible depuis l’ensemble des autres objets. Pour cela, nous pouvons envisager différentes stratégies.

### Passage par paramètre

Le plus simple est de créer un objet panier dans la classe racine de notre application : la classe MyHomePage. L’objet sera ensuite transmis, via les constructeurs, à tous les autres objets.

### Le Design Pattern Singleton

Le Design Pattern Singleton peut être utilisé pour centraliser les données.

### Les Provider

Voir <https://flutter.dev/docs/development/data-and-backend/state-mgmt/simple>

Le package Provider permet d’implémenter le Design Pattern Observable. Il se base donc sur un objet pouvant notifier une liste d’objets d’un changement.

La classe Panier serait chargée de notifier tout changement aux autres classes de type UI.

### Le Design Pattern BloC (Business Logic Component)

Voir <https://bloclibrary.dev/#/gettingstarted>

Ce Design Pattern est basé sur les Streams et les événements pour échanger les données.

Il est plus difficile à appréhender.

## Stratégie retenue

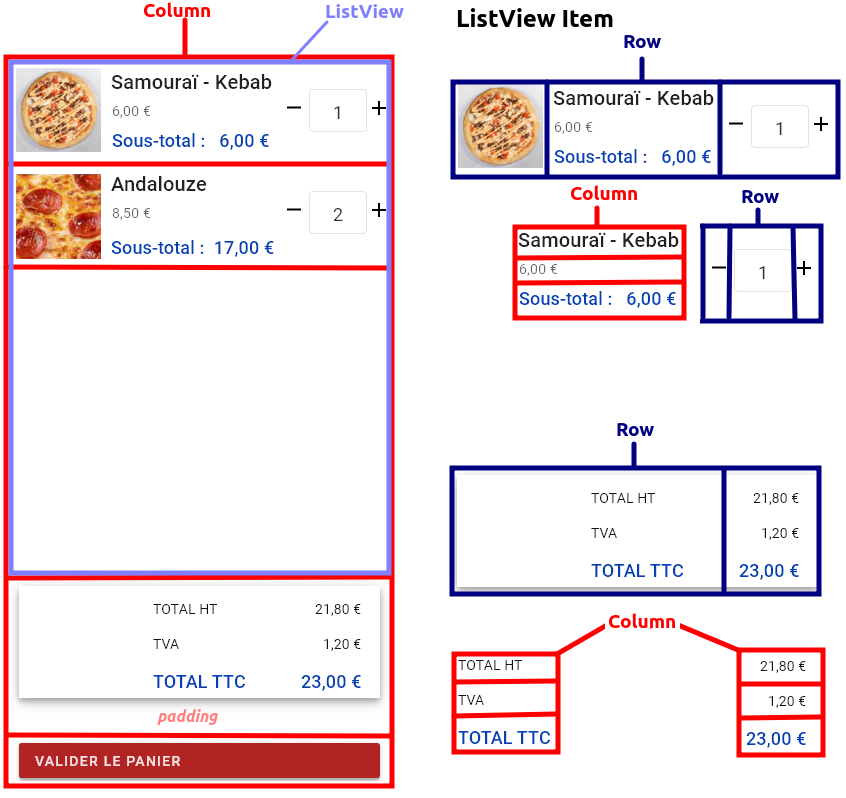
Dans un premier temps, nous allons prendre le passage par paramètre. Vous pourrez par la suite choisir les “Provider”.

# 

# La page Panier - UI

## Le design

Le design doit ressembler à ceci :



## Le code de départ

Comme c’est une nouvelle fenêtre, nous allons créer la classe dans le répertoire “ui”.

Dans ce répertoire, créez un nouveau fichier nommé “panier”.

Ce Widget va être un widget avec état car il est modifiable par l’utilisateur.

Dans le fichier vide, tapez “stful” puis entrée, nommez la classe “Panier”.

Comme c’est une nouvelle fenêtre, il nous faut ajouter les Widgets Scaffold et AppBar.

Nous allons afficher :

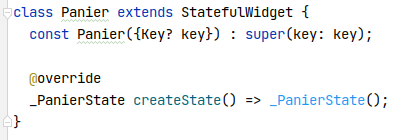
* la liste des pizzas commandées dans une ListView.
* le montant total HT, la TVA et le montant total TTC
* le bouton “Valider” pour finaliser la commande.

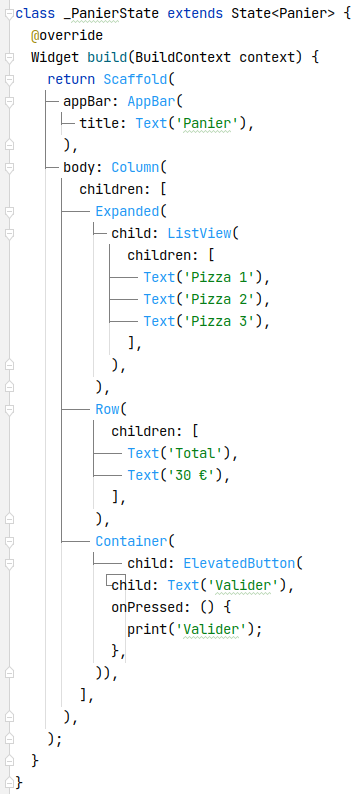
Contraintes UI :

* Les totaux et le bouton “Valider” auront une hauteur fixe.
* La ListView doit prendre toute la place restante.

Dans un premier temps, nous allons faire un début d’UI en mettant des valeurs en dur.

Le code sera le suivant :





Évidemment cela n’est pas très “User Friendly" mais nous l’arrangerons par la suite.

# 

# La navigation

Pour aller vers le panier, l’utilisateur va cliquer sur une icône que nous allons placer dans tous les widget AppBar.

Si nous devons modifier tous les widget AppBar alors nous allons créer notre propre Appbar.

## Une AppBar personnalisée

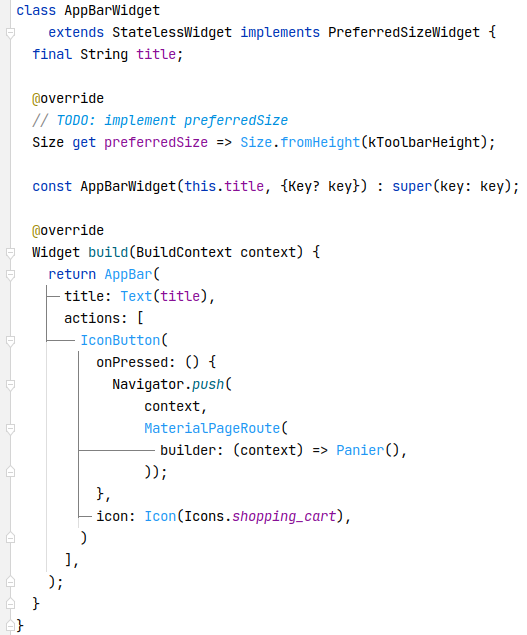
Dans le répertoire “ui/share”, créez un nouveau fichier nommé “appbar\_widget”. La classe sera sans état.

Dans le fichier vide, tapez “stless” puis entrée, nommez la classe “AppBarWidget”.

Cette classe va afficher AppBar avec un icône, cliquable, pour le panier. Elle reçoit un titre et recevra plus tard l’objet panier.

Lorsque l’utilisateur clique sur le panier, il est redirigé vers l’UI panier.

Le code sera le suivant :



Explication :

La classe implémente “PreferredSizeWidget”. Nous devons l'implémenter car nous allons utiliser l’AppBarWidget dans le widget Scaffold (attribut “appBar”).

Or l’attribut “appBar” du widget Scaffold exige que le widget associé hérite de “PreferredSizeWidget” pour pouvoir fixer la hauteur. Ce qui est fait en surchargeant le getter “preferredSize”.

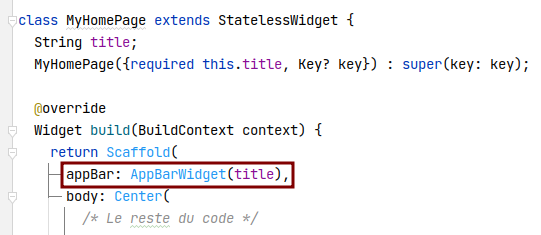
## Intégration de l’AppBar personnalisée

Il nous faut maintenant intégrer cette AppBar dans les autres pages.

### La page principale

Nous allons remplacer l’AppBar “classique” par notre AppBar personnalisée.

Il faut modifier la méthode build de la classe MyHomePage comme ceci :



### La liste des pizzas et le détail d’une pizza

Vous devez faire la même chose pour la classe “PizzaList” et la classe “PizzaDetails”.

# 

# La classe Cart

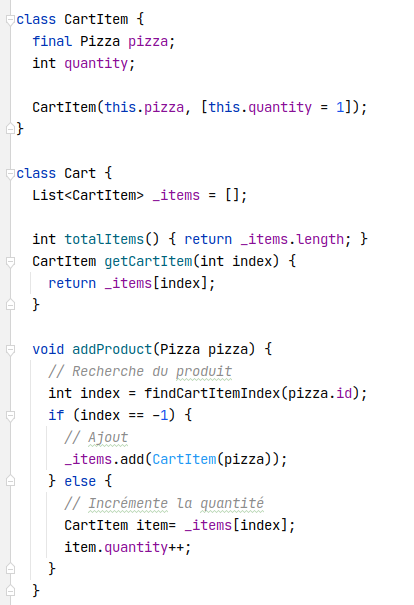
Le panier affiche la liste de pizzas commandées. Nous allons donc créer une classe “Cart” qui va gérer nos commandes.

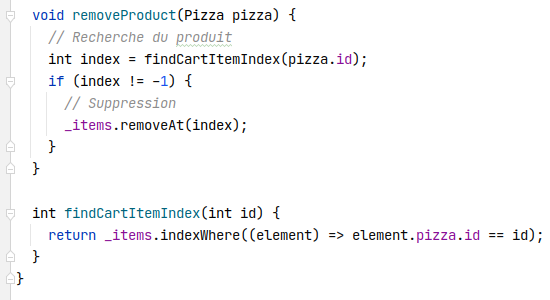
Cette classe va avoir :

* une liste d’objet CartItem
* la classe CartItem aura pour attribut une pizza et la quantité commandé
* une méthode pour ajouter une pizza
* une méthode pour supprimer une pizza
* plus des méthodes de calcul
* et des accesseurs

Dans le répertoire “models”, créer le fichier “cart”.

Le code sera le suivant :





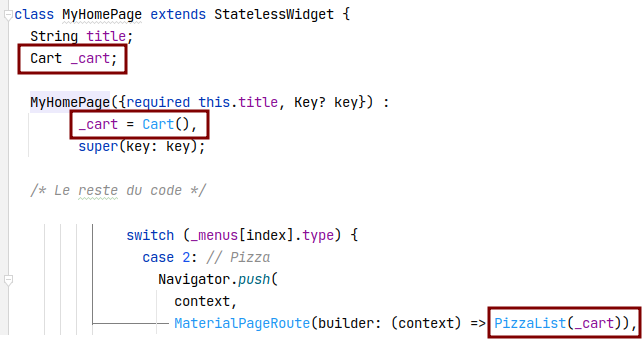
Nous compléterons cette classe si besoin.

# Utilisation de la classe Cart

Nous allons donc prendre la stratégie “Passage par paramètre”. Il faut donc créer un objet Cart dans le widget de base “MyHomePage”.

## Le widget principal MyHomePage

Il faut modifier la classe MyHomePage pour ajouter l’objet “cart” et le passer aux widgets enfants.



## Le widget PizzaList

La classe PizzalList va donc avoir un objet Cart. Cet objet sera transmis aux widgets PizzaDetails et BuyButtonWidget.

De plus, le widget BuyButtonWidget doit savoir quelle est la pizza en cours, il faut donc le lui transmettre.

Vous devez modifier le code ainsi :



Explication :

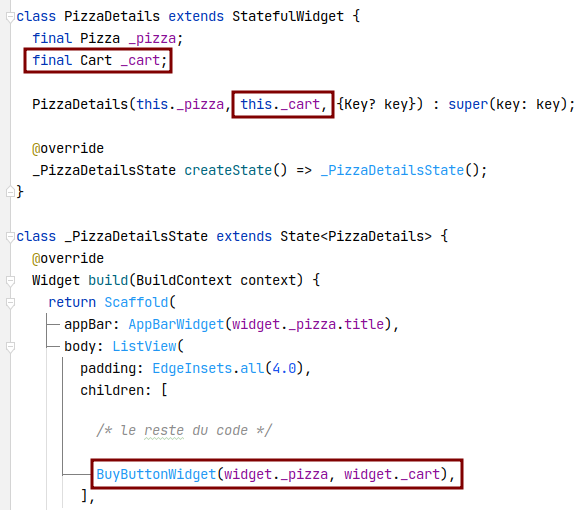
Le getter “widget” de la classe \_PizzaListState permet d’identifier l’objet “PizzaList”. \_PizzaListState

## 

## Le widget PizzaDetails

La classe PizzalDetails va aussi avoir un objet Cart. Cet objet sera transmis au widget BuyButtonWidget.

Vous devez modifier le code ainsi :



## 

## Le widget BuyButtonWidget

Ce widget a un bouton permettant de commander (méthode onTap) une pizza. Nous allons ajouter la pizza en cours et le panier puis implémenter la commande d’une pizza.

Le code sera le suivant :



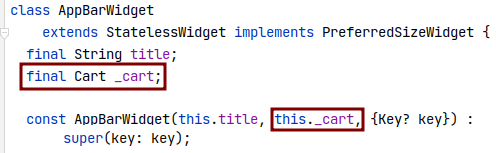
## Le widget AppBarWidget

Ce widget permet d’afficher le panier. Il faut donc que le widget ait l’objet Cart pour le transmettre à la page Panier.

### 

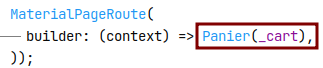
### Modification du widget

Il faut ajouter un objet Cart :



Problème : l’application ne compile plus car vous venez d’ajouter un paramètre à la classe AppBarWidget. Corrigez cela dans les fichiers concernés (voir Dart Analysis).

Et le passer à la page Panier :



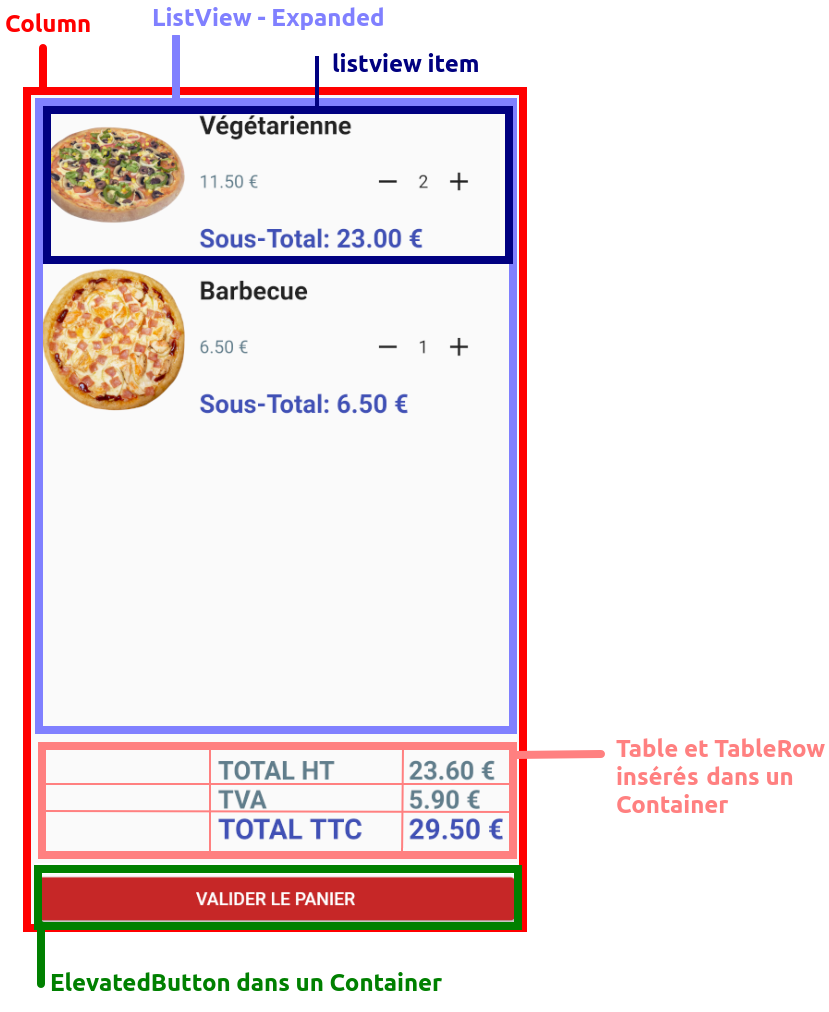
## 

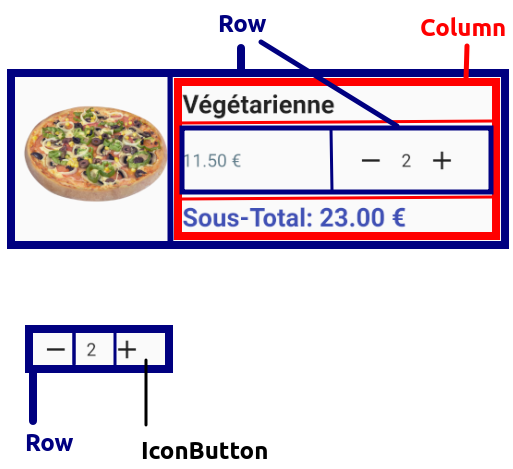
# La page Panier - UI (suite)

N’oublions pas que le but de tout ceci est d’afficher la liste des pizzas commandées.

## Le design définitif

Le design définitif sera le suivant :

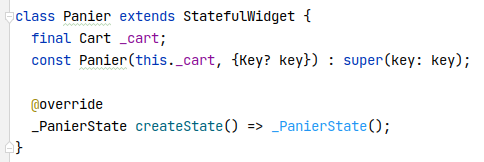




## Un début de code

Vous êtes un peu plus habitué à coder en Flutter donc il faut faire le widget Panier à partir du design.

Cependant, voici un début de code :





A vous de jouer 😊

# Résumé

Vous devez maitriser les éléments suivants :

* Utilisation de [Table et TableRow](https://api.flutter.dev/flutter/widgets/Table-class.html)