Flutter Pizzeria – TD 2

# La liste des pizzas

Pour notre liste de pizzas, nous allons créer les classes Pizza et PizzaData.

## La classe Pizza

Cette classe est notre modèle et comporte peu d’attribut.

Dans le répertoire “models, créer la classe Pizza :

* click droit sur le répertoire
* choisissez New | Dart File
* Nommez le fichier “pizza”

La classe doit ressembler à ceci :

Une image contenant texte

Description générée automatiquement

## La classe PizzaData

Cette classe va être une classe utilitaire pour obtenir des pizzas. Créez la classe PizzaData :

* click droit sur le répertoire “models”
* choisissez New | Dart File
* Nommez le fichier “pizza\_data”

La classe doit ressembler à ceci :

Une image contenant texte

Description générée automatiquement

## L’affichage de la liste des pizzas

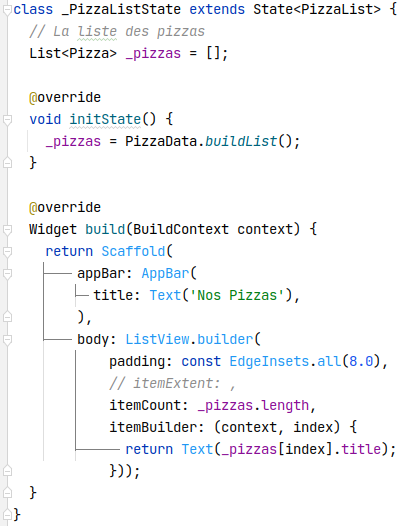
Comme c’est une nouvelle fenêtre, nous allons créer un répertoire nommé “ui” (dans le répertoire lib).

Dans ce répertoire, créez un nouveau fichier nommé “pizza\_list”.

Ce Widget va être un widget avec état car, plus tard, ses éléments seront modifiés avec l’interaction utilisateur.

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

Cette classe va afficher une liste de pizzas. Le code sera le suivant (extrait) :



Explications :

* Nous avons toujours le Widget AppBar
* Le Widget Stateful permet d’initialiser les valeurs de départ avec la méthode surchargée “initState”.
* La ListView est construite comme précédemment avec un padding en plus. Nous fixeront l’attribut itemExtent plus tard.

Il nous faut maintenant passer du menu à la liste des pizzas lorsque l’on clique sur le menu Pizzas.

# La navigation

La navigation d’une fenêtre à une autre se fait grâce au Widget [Navigator](https://flutter.dev/docs/cookbook/navigation/navigation-basics).

Rappel, exemple de navigation :

Navigator.*push*(

context,

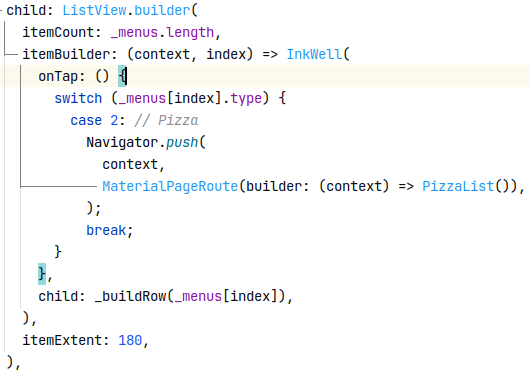
MaterialPageRoute(builder: (context) => AWidget()),

);

Pour aller du menu à la liste des pizzas, il faut que l’utilisateur clique sur le menu Pizza. La gestion d’un “click” se fait avec différents Widget dont [GestureDetector](https://flutter.dev/docs/cookbook/gestures/handling-taps) ou [InkWell](https://flutter.dev/docs/cookbook/gestures/ripples).

Nous allons utiliser InkWell lorsque l’utilisateur clique sur un menu.

Modifiez le code de la construction de la liste des menus (fichier main.dart, méthode “build” de la classe “MyHomePage”) comme ceci :



Explications :

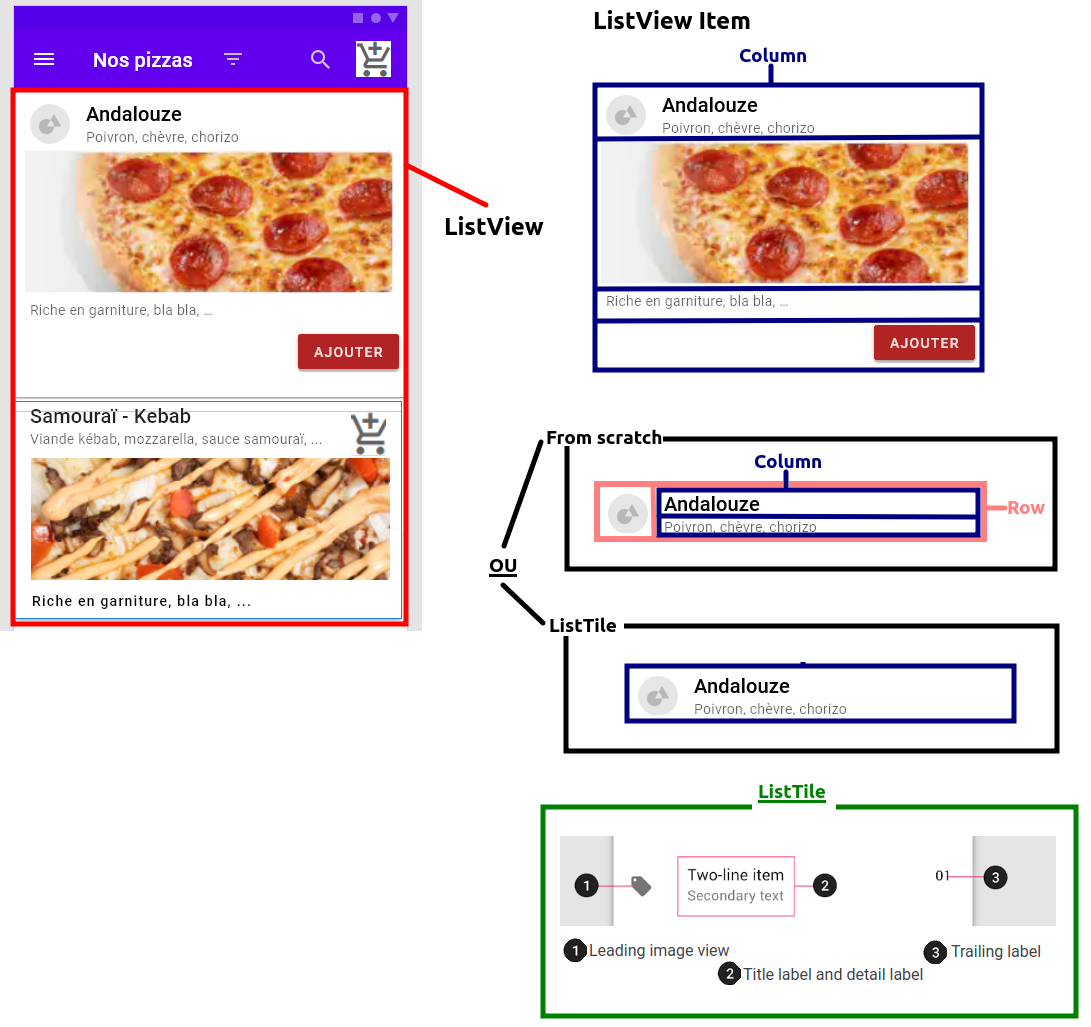
* itemBuilder ne construit plus l’élément. La construction de l’élément est encapsulée dans le Widget InkWell.
* Le Widget InkWell a la méthode “onTap” qui réagit au clic de l'utilisateur
* Au clic, l’application va vers la fenêtre PizzaList

Testez !!

# La liste des pizzas (suite)

La liste des pizzas affiche uniquement le nom de la pizza. Nous allons afficher plusieurs informations dont l’image de la pizza.

## Proposition de design



## Le design retenu

Pour un élément, nous allons utiliser un Widget Column avec comme enfants un Widget ListTile, une image, une description et un bouton.

## L’implémentation

Nous allons modifier la méthode « build » de la classe PizzaList.

Il faut donc avoir :

* Un Widget Column avec pour enfants :
  + Un ListTile (avec le titre de la pizza)
  + L’image de la pizza
  + La description de la pizza
  + Un bouton « Commander »

Il nous faut donc ajouter les images des pizzas.

**Important :**

Récupérez les images sur MLB pour les mettre dans le dossier « assets/images/pizzas ». Ajoutez le répertoire dans le fichier « pubspec.yaml » (n’oubliez pas de faire un flutter pub get).

Il faut modifier la construction d’un élément de la liste pour le faire dans une méthode « buildRow » comme ceci :

Une image contenant texte

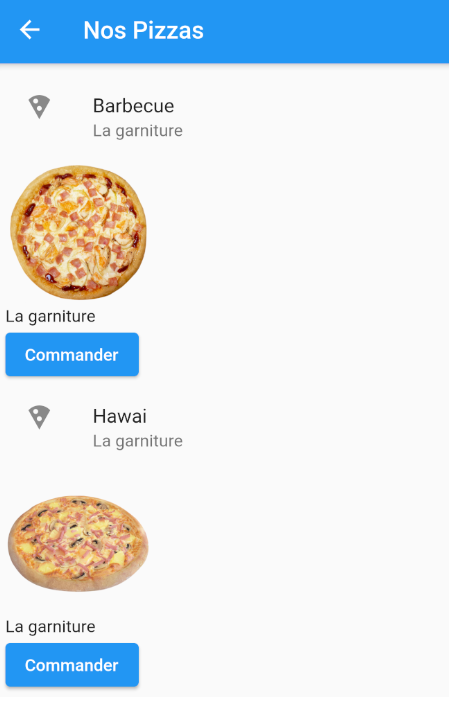
Description générée automatiquement

La méthode « \_buildRow » va donc avoir un Widget Column avec comme enfants un Widget ListTile, une image, une description et un bouton :



Testez !!

Vous devriez avoir ce résultat :



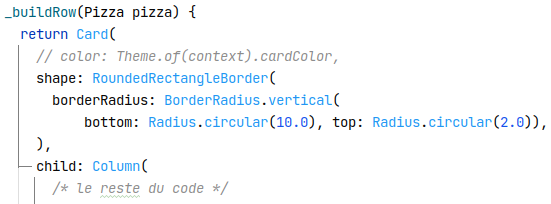
Ce n’est pas trop mal mais il manque la mise en page.

## Un peu d’amélioration graphique

Nous allons mettre en forme l’affiche d’une pizza. Les modifications vont porter sur la méthode buildRow.

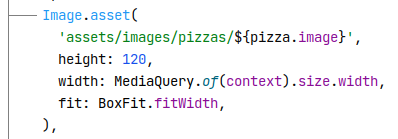
### Le Widget Card

Nous allons utiliser un Widget Card pour faire les bords arrondis (en bas) :



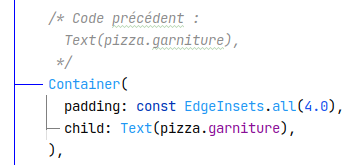
### L’image de la pizza

La pizza est affichée en entier. Nous voulons l’afficher sur toute la largeur en tronquant l’image si nécessaire. Nous allons utiliser l’objet “MediaQuery” pour calculer la largeur disponible :



### L’affichage de la garniture

Nous allons mettre du padding pour ne pas coller le texte :



### L’affichage du bouton “Commander”

Le fond du bouton est bleu et le bouton est placé à gauche. Or nous le voulons avec un fond rouge placé à droite avec un icône d’un panier :



Testez !!

# Résumé

Vous devez maîtriser les éléments suivants :

* Utilisation de [StateFul et StateLess Widget](https://docs.flutter.dev/development/ui/interactive)
* Utilisation de padding, margin avec [EdgeInset](https://api.flutter.dev/flutter/painting/EdgeInsets-class.html) : <https://www.fluttercampus.com/guide/144/how-to-add-padding-margin-on-text-widget-in-flutter/> et <https://www.tutorialkart.com/flutter/flutter-padding/>
* Utilisation de la [navigation](https://docs.flutter.dev/cookbook/navigation/navigation-basics)