Flutter Pizzeria

# Présentation

## L’application

Nous allons développer une application mobile permettant de commander des pizzas, boissons et autres.

## Les outils

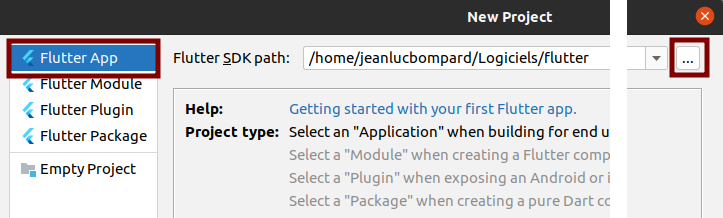
Nous allons utiliser Android Studio avec les SDK Flutter et Dart.

# Création du projet

## Création de la structure

Lancez Android Studio, choisissez “Create New Flutter Project”.

Sélectionnez “Flutter App” et le SDK Flutter (si celui-ci est vide) :



Cliquez sur suivant puis nommez votre projet “pizzeria” (le nom doit être en minuscule); choisissez aussi le répertoire de création.

**Attention : le chemin du répertoire ne doit pas contenir d’espaces, caractères accentués, caractères spéciaux, …**

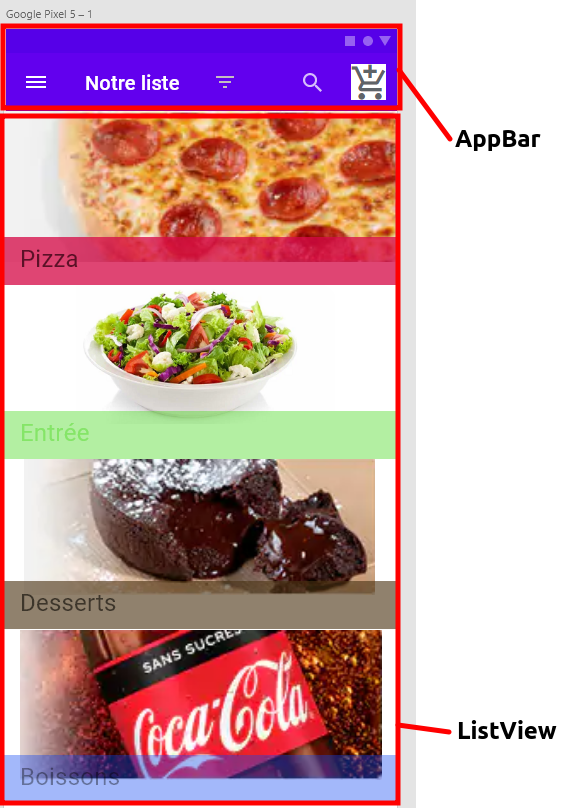
Cliquez sur “Finish”.

# 

## Design de la page principale

Avant toute chose, vous êtes obligé de faire le design du projet. Pour ce faire, j’ai utilisé Adobe XD mais comme vous allez le voir je ne suis pas (du tout) un designer.

L’écran principal devrait ressembler à cela :



Le Widget AppBar est déjà présent dans notre projet. Par contre, il nous faut créer le Widget ListView. L’UI de ce Widget ne sera pas modifié par l'utilisateur donc nous allons créer un Widget sans état.

## Création de la page principale

Ouvrez le fichier lib/main.dart.

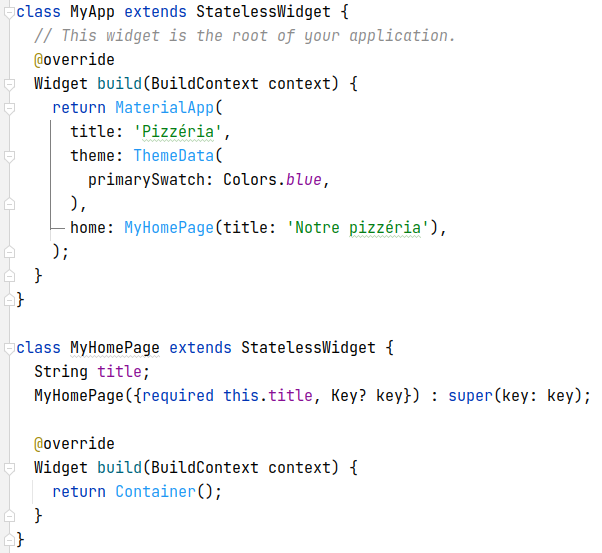
Supprimez les classes MyHomePage et \_MyHomePageState

En bas du fichier, créez un Widget sans état en entrant le texte “stless” puis entrée : nommez la classe MyHomePage.

Quelques réglages :

* Ajouter un attribut “title” de type texte à la classe MyHomePage
* Changer les titres, de la classe MyApp, en “Pizzéria” et “Notre pizzéria”.

Le code devrait ressembler à ceci :



# 

# La page principale

La page principale est donc constituée de la liste du menu.

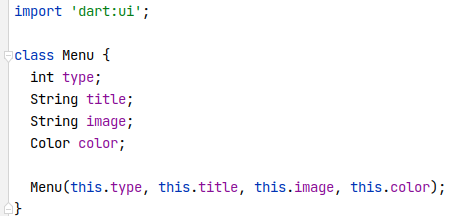
Nous allons créer “en dur”, dans la classe “myHomePage”, la liste du menu. Cette liste est constituée de menu, il faut donc créer la classe menu.

## La classe Menu

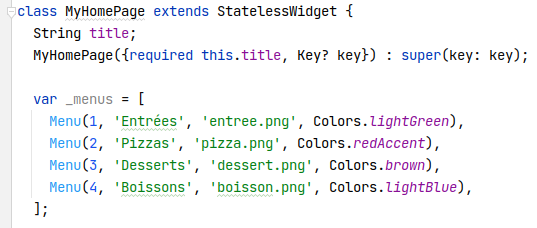
Créer un répertoire “models” dans le répertoire “lib”. Dans ce répertoire “models, créer la classe Menu :

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

La classe doit ressembler à ceci :



Dans la classe MyHomePage, ajoutez une liste de menu :

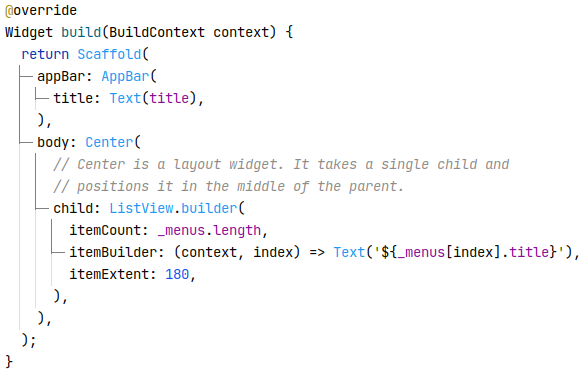


## 

## La liste des menus

Pour afficher une liste, nous pouvons utiliser le Widget [ListView](https://api.flutter.dev/flutter/widgets/ListView-class.html).

Nous allons modifier la méthode “build” pour afficher notre ListView :

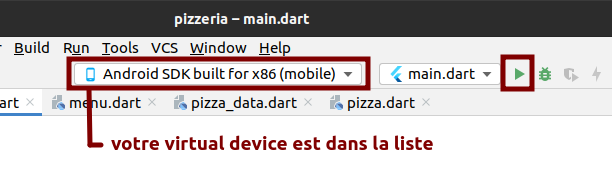
￼

Un peu d’explication :

* Le constructeur ListView.builder permet de construire la ListView d’après un objet List.
* itemCount : indique le nombre d’éléments à afficher
* itemExtent : indique la hauteur d’un élément. Il est conseillé de l’utiliser sinon Flutter recalcule la hauteur pour chaque élément de la liste
* itemBuilder : permet de construire un élément d’après son indice (variable index). Nous affichons ici un texte simple.

## Testons

Si vous ne l’avez pas déjà fait, créez un téléphone virtuel avec AVD Manager. Sélectionnez-le dans la liste et lancez l’application :



Le rendu n’est pas joli mais il est fonctionnel.

## Amélioration de la liste du menu

Une image étant plus parlante qu’un titre, nous allons ajouter des images.

### Les images

Créez l'arborescence de répertoires nommés “assets/images/menus” à la racine du projet.

Copiez les images de menu (depuis 360Learning).

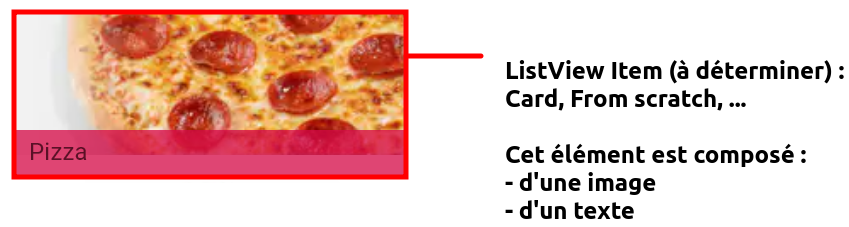
Pour pouvoir utiliser des images dans Flutter, il faut les déclarer dans le fichier “pubspec.yaml” comme ceci (dans la section flutter) :

flutter:  
uses-material-design: true  
 assets:  
 - assets/images/menus/

A chaque modification de ce fichier, vous devez exécuter la commande ”flutter pub get”.

### Un élément de la liste

Au niveau design, cela doit ressembler à ceci :



Nous allons déléguer la création d’un élément à une méthode nommée “buildRow”. Nous allons créer, dans un premier temps, un Widget Column avec une image et un texte :



Explications :

* L’image, étant trop grande, déborde de l’item. Il faut donc la mettre dans un Container avec une hauteur fixe.
* Une image est chargée, depuis un fichier, avec le constructeur Image.asset('chemin/vers/image'). L’image va occuper tout l’espace dans la largeur

C’est mieux mais ce n’est pas encore ça ;)

Problèmes : on a dû fixer une taille pour l’image et le texte est trop petit.

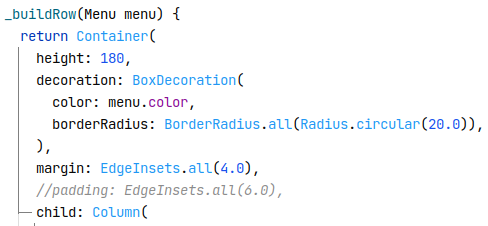
Une petite amélioration :



Explications :

* Le texte a une hauteur fixe avec un style défini (fonte : famille, taille, …)
* L’image doit donc occuper tout le reste de l’espace libre. Le Widget Expanded le fait pour nous.

C’est mieux mais nous voulons comme couleur de fond celle du menu, que les bords soient arrondis et qu’il y est un espace entre les éléments :



Explications :

* Le Widget Column est mis dans un Widget Container pour pouvoir définir la hauteur,
* les marges : margin: EdgeInsets.all(4.0),
* et les bords arrondis : decoration: BoxDecoration(color: menu.color, **borderRadius: BorderRadius.all(Radius.circular(20.0))**,),

Nous pouvons maintenant nous occuper de notre liste de pizzas.

# Résumé

Vous devez maitriser les éléments suivants :

* Fichier [pubspec.yaml](https://docs.flutter.dev/development/tools/pubspec)
* Utilisation de [MaterialApp](https://api.flutter.dev/flutter/material/MaterialApp-class.html)
* Utilisation de [Scaffold](https://api.flutter.dev/flutter/material/Scaffold-class.html)
* Utilisation de [ListView.builder](https://api.flutter.dev/flutter/widgets/ListView-class.html)
* Utilisation de [Column](https://api.flutter.dev/flutter/widgets/Column-class.html), voir aussi : <https://docs.flutter.dev/development/ui/layout>
* Utilisation de [Image.asset](https://docs.flutter.dev/development/ui/assets-and-images)
* Utilisation de [Expanded](https://api.flutter.dev/flutter/widgets/Expanded-class.html)