Flutter Pizzeria - TD 5 - WebService

# Présentation

Nous récupérons l’ensemble des pizzas dans le widget “PizzaList”. Ce widget est avec état.

L’obtention des données se fera donc dans la méthode initState() grâce à une classe qui accède au Web Service.

La classe Pizza va convertir le JSON en objet.

Le but de ce TD est de créer un « pseudo web service » et de consommer ce web service via notre code Flutter.

## Mise en place du web service

Sur votre machine ou sur un serveur distant ou bien sur une machine virtuelle, vous allez créer un service web de type Apache (vous pouvez bien sûr créer un service nginx ou autre si vous préférez).

Suivant votre distribution, vous mettrez en œuvre :

* Windows : WAMP, XAMPP, EasyPHP, …
* Linux : LAMP
* Mac : XAMPP, MAMP

## Mise en place des répertoires

Dans le répertorie racine de votre serveur web, créez le répertoire « api » et dans ce répertoire, le répertoire « pizzas ». Dans le répertoire « pizzas », créez une page php nommée « index.php ».

*Remarque : vous êtes libre de créer le web service de votre choix (pour ma part, j’ai pris Spring Boot avec une base de données MySQL).*

Vous devriez accéder à la page php, en utilisant l’URL <http://localhost/api/pizzas> dans votre navigateur. Attention votre serveur web utilise peut-être un port différent.

Toujours dans le répertorie racine de votre serveur web, créez les répertoires « static/images/pizzas/ ». Dans le répertoire « pizzas », placez toutes les images de vos pizzas. Vous devriez voir les images de vos pizzas ; exemple, en utilisant l’URL <http://localhost/static/images/pizzas/pizza-bbq.jpg> dans votre navigateur.

## Exemple de page PHP pour les données JSON

L’URL suivante doit fonctionner : <http://localhost/api/pizzas> et donc afficher le JSON.

Exemple de code PHP :

<?php

*header*("Content-Type: application/json; charset=UTF-8");

$json = array();

$json[] = array(

'id' => 1,

'title' => 'Barbecue',

'garniture' => 'La garniture',

'image' => 'http://localhost/static/images/pizzas/pizza-bbq.jpg',

'price' => 6.5

);

$json[] = array(

'id' => 2,

'title' => 'Hawaï',

'garniture' => 'La garniture',

'image' => 'http://localhost/static/images/pizzas/pizza-hawai.jpg',

'price' => 9.5

);

$json[] = array(

'id' => 3,

'title' => 'Epinards',

'garniture' => 'La garniture',

'image' => 'http://localhost/static/images/pizzas/pizza-spinach.jpg',

'price' => 8.5

);

$json[] = array(

'id' => 4,

'title' => 'Végétarienne',

'garniture' => 'La garniture',

'image' => 'http://localhost/static/images/pizzas/pizza-vegetable.jpg',

'price' => 10.5

);

echo json\_encode($json) ;

# Le WebService

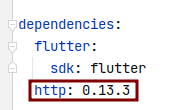
Rappel : Nous allons obtenir les données à partir d’un Web Service de type RESTful.

Voir aussi : [Fetch data from the internet](https://flutter.dev/docs/cookbook/networking/fetch-data)

Pour obtenir les données, nous allons utiliser un objet http pour contacter le WebService. Puis nous allons convertir les données JSON en objet Pizza.

## Le package http

Il faut ajouter le package http au projet. Ajouter le package au fichier pubspec.yaml :



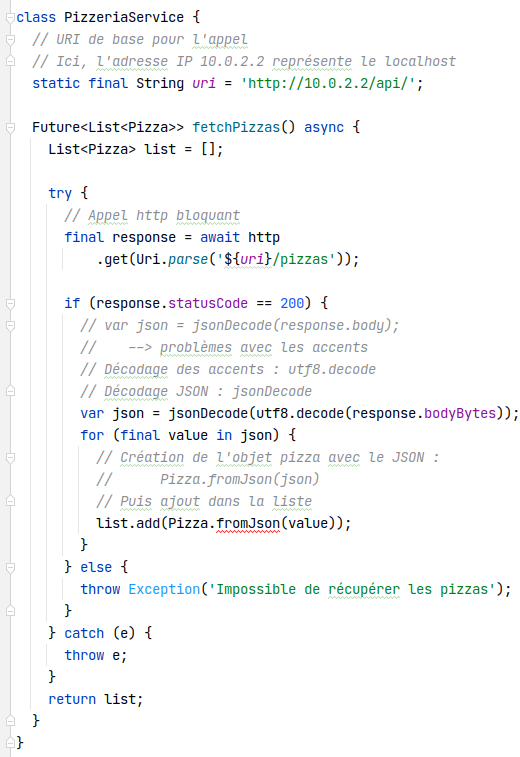
La version que vous devez utiliser est précisé ici : <https://pub.dev/packages/http>

### La classe PizzeriaService

Cette classe va nous permettre de contacter le Web Service.

Dans le répertoire “lib”, créer un répertoire “services”. Dans le répertoire “services”, créez un nouveau fichier nommé “pizzeria\_service”.

Le code sera le suivant :



**Important : L’adresse 10.0.2.2 représente le localhost de votre machine. Pour tester l’application, vous devez mettre en place un serveur web (Apache par exemple). Si votre serveur est situé sur une autre machine alors il faut indiquer son adresse IP ou son nom DNS.**

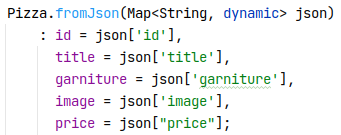
Une explication des appels est faite dans un des chapitres suivants.

### 

## La classe Pizza

Nous allons ajouter un constructeur nommé pour convertir le JSON en objet.

Le code sera le suivant :



Explications :

* Le JSON reçu est une map de chaîne de caractères. On récupère chaque élément grâce à sa clé. La conversion est faite automatiquement par Flutter.
* Remarque :
  + La valeur ‘1’ est converti en int
  + La valeur ‘1.0’ est convertie en double

### 

## La classe PizzaList

Nous allons donc modifier la méthode “initState()” pour obtenir les données d’après l’appel au Web Service. Le code sera le suivant :

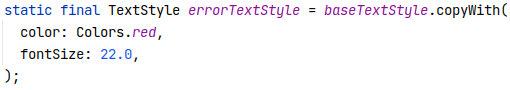


Explications :

* La classe FutureBuilder permet de construire un widget. Elle prend en paramètre au moins les 2 paramètres suivants :
  + future : un objet Future. Celui-ci est déclaré avec l’attribut late ce qui signifie que l’objet est initialisé plus tard. L’objet est obtenu dans la méthode initState
  + builder : méthode chargée de construire le widget. L’objet snapshot est le résultat de l’appel, il contient les données obtenues ou non (attribut data).

## La classe PizzeriaStyle

Il faut ajouter le style suivant :



## L’affichage des images des pizzas

Les images des pizzas sont donc maintenant chargées depuis une URL. Vous devez modifier tous les appels à Image.asset en Image.network, tout simplement !

## Les appels asynchrones

Voir le cours “Dart - Notions avancés” pour de plus amples explications.

# Résumé

Vous devez maitriser les éléments suivants :

* Obtenir des données d’après un service Web Restful : [Fetch data from the internet](https://flutter.dev/docs/cookbook/networking/fetch-data)