TP5 – Vue et template de détail d'une pizza

Introduction, Contexte et Objectifs

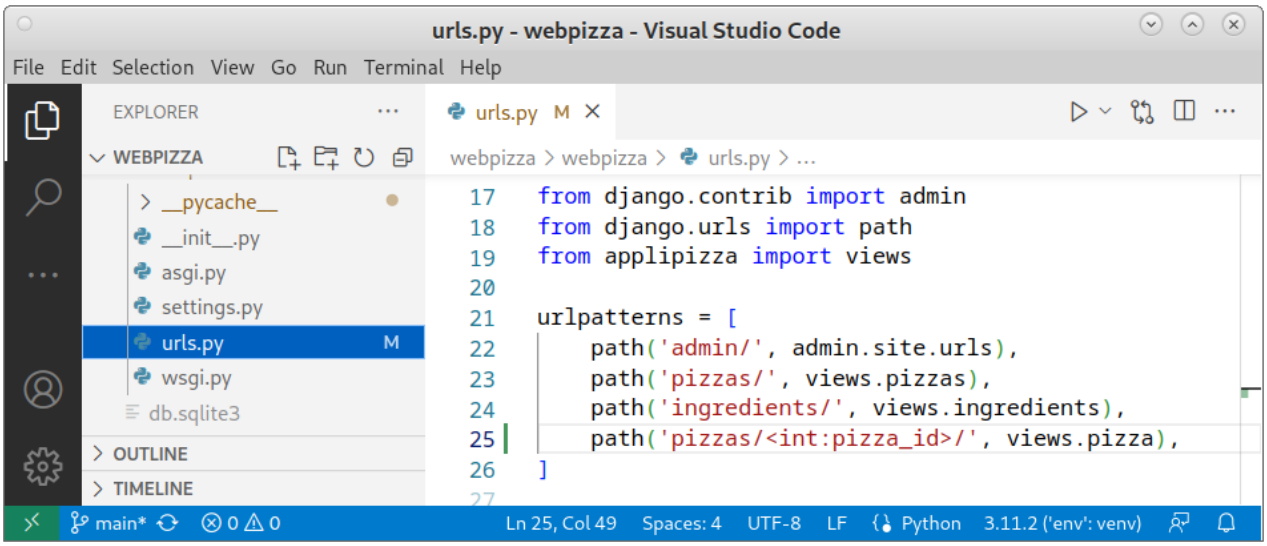
Les premiers templates, améliorées par le gabarit base.html, font des listes des pizzas et d'ingrédients. Nous allons agrémenter notre projet d'un nouveau template qui donnera les détails sur une pizza. Nous mettrons aussi en place l'url vers ce template, ainsi que le lien vers ce template dans la liste globale des pizzas.

Modification du fichier urls.py

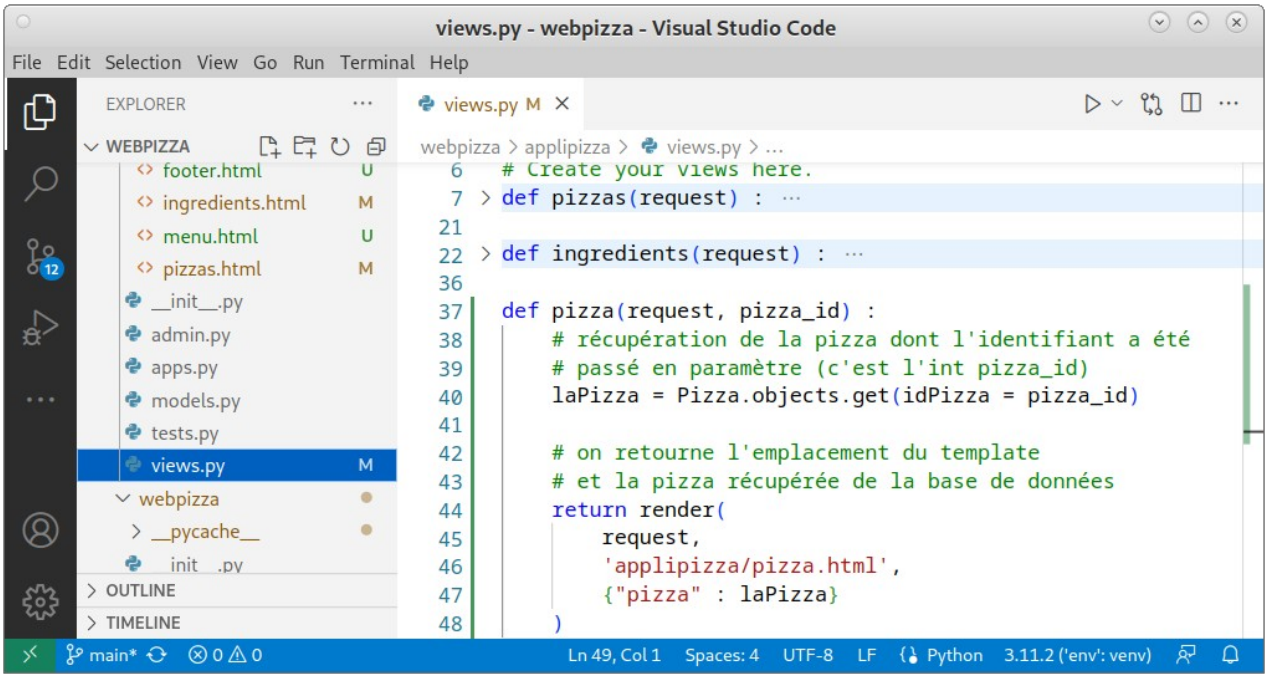
1. Nous allons créer une url générique pour le détail d'une pizza. Nous pourrons l'appeler ainsi :

   http://127.0.0.1:8000/pizzas/1/
   http://127.0.0.1:8000/pizzas/2/

   Dans le fichier urls.py, ajoutez le path de la dernière ligne (il contient un int variable nommé pizza_id), path qui envoie vers une nouvelle vue « pizza », pas encore codée dans le fichier applipizza/views.py.



2. Créez dans applipizza/views.py une view **pizza** dont le code est le suivant. Vous remarquerez qu'elle utilise un paramètre supplémentaire pizza_id qui correspond à la variable du path.



Notez la requête pour récupérer LA bonne pizza

laPizza = Pizza.objects.get(idPizza = pizza_id)

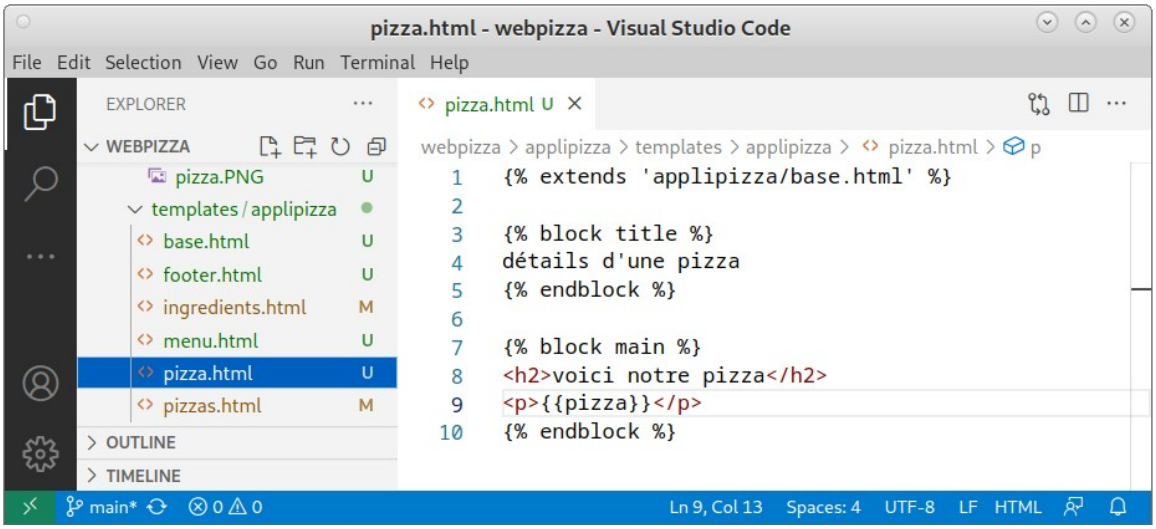
a. la méthode get qui récupère un élément de la base en fonction du critère qui suit en paramètres,

b. le critère donné : l'attribut idPizza doit avoir la même valeur que l'int pizza_id récupéré par la view pizza.

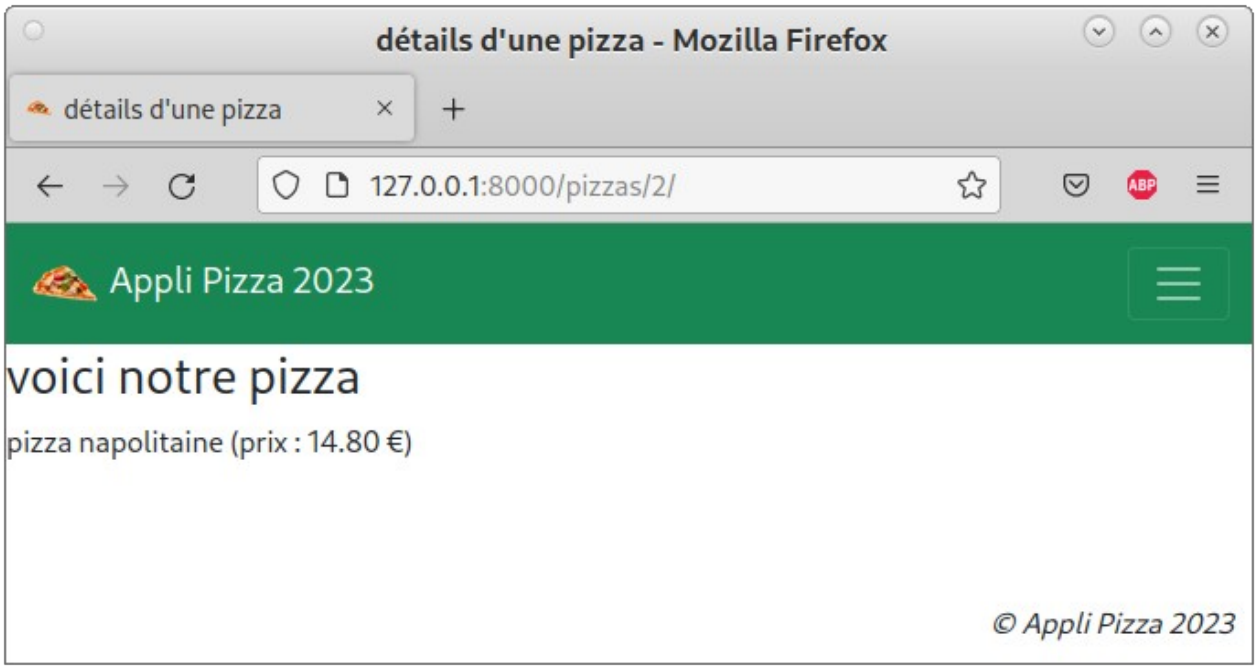
Notez aussi, comme précédemment, la façon de transmettre au template l'information concernant cette pizza (sous forme de dictionnaire dans le render).

3. Créez maintenant le template pizza.html avec le code suivant. Analysez ce code, vous y retrouverez des éléments déjà vus. Vous remarquerez en particulier l'utilisation de la variable de gabarit dans l'affichage {{pizza}} .

Cette variable pizza est celle qui a été passée dans le dictionnaire de la view précédente.

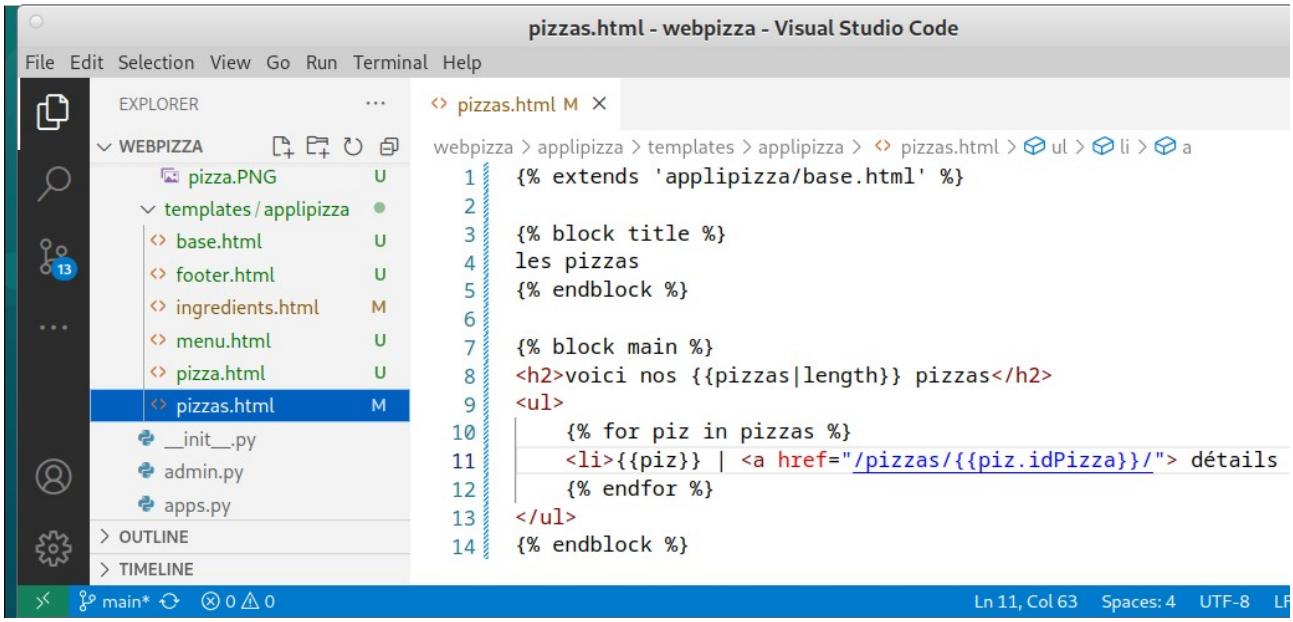


4. Essayez maintenant les deux url du 1. Vous devriez obtenir un résultat de ce type :



5. Nous allons maintenant ajouter, dans la page pizzas.html, les liens de détail menant à la page relative à chaque pizza.

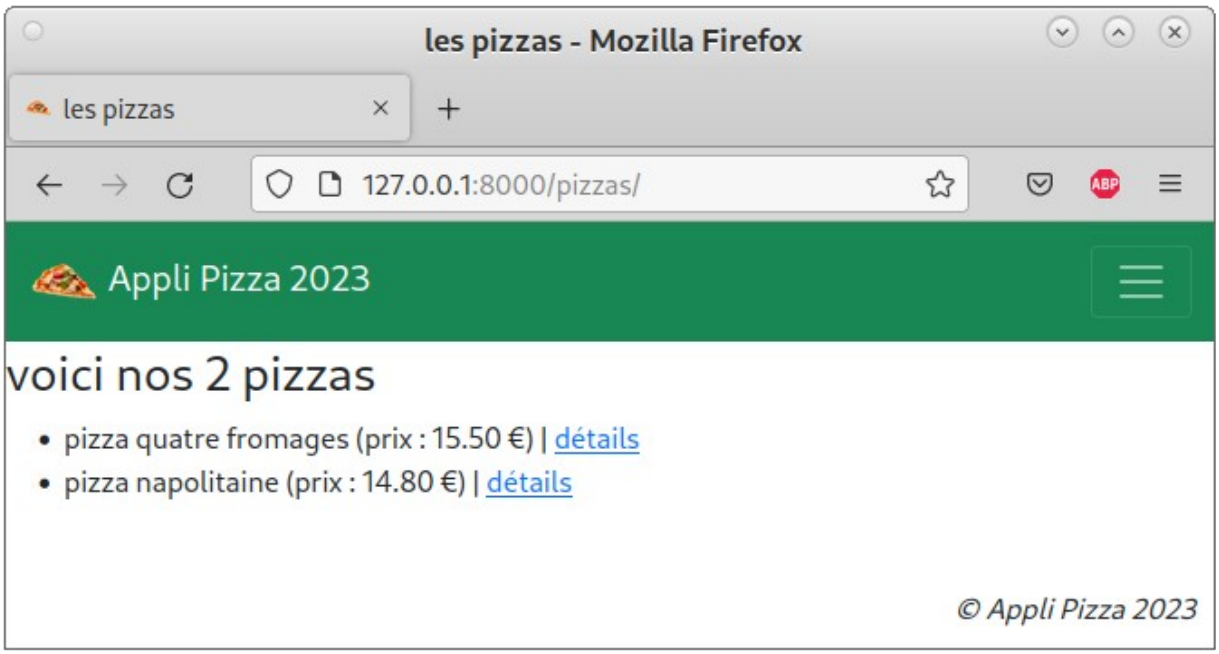
   Changez le code pour intégrer ces liens :



6. Vérifiez l'effet. Pour le moment, cela ne présente pas vraiment d'intérêt, puisque la vue de détail n'apporte rien de plus que les renseignements déjà donnés par la page pizzas.html.

   Mais cela va bientôt changer, puisque vous allez compléter cette vue de détail d'une pizza en listant les ingrédients entrant dans sa composition.

   Rendu actuel de la page pizzas.html :



Amélioration du template détail d'une pizza

Cette partie est à faire en autonomie.

Vous devrez améliorer la view d'une pizza, définie dans views.py, pour qu'elle donne aussi la liste des ingrédients de la pizza, avec leur quantité.

Cette amélioration demandera de faire une requête sur la base de données, en récupérant de la table Composition toutes les entrées pour lesquelles l'identifiant de la pizza correspond à celui passé en argument (pizza_id).

Cette requête utilise la méthode « filter » :

```
compo = Composition.objects.filter(pizza = pizza_id)
```

- pizza désigne la clé étrangère de la classe Composition,
- pizza_id est le paramètre passé dans l'url.

Puis vous devrez construire une liste des ingrédients, que vous transmettrez au template par l'intermédiaire du dictionnaire python dans lequel il n'y a pour le moment que la pizza à afficher.

Voici en gros les étapes de la view que vous pouvez valider :

```
# récupération de la pizza dont l'identifiant a été passé
# en paramètre (c'est l'int pizza_id)

# récupération des ingrédients entrant dans la composition de la pizza

# on crée un dictionnaire dont chaque item sera lui-même
# un dictionnaire contenant :
# - l'identifiant de la composition (idComposition)
# - le nom de l'ingrédient
# - la quantité de l'ingrédient dans cette composition

# on retourne l'emplacement du template, la pizza récupérée de la base
# et la liste des ingrédients calculée ci-dessus
```

Enfin, au niveau du template pizza.html, vous devrez présenter les ingrédients proprement, par exemple sous forme d'un tableau en utilisant les classes bootstrap.

Voici un exemple de rendu :