

## TP15 – Le panier

### Introduction, Contexte et Objectifs

Dans cette nouvelle série de TP nous mettons en place le panier d'achats. Nous pouvons partir sur le modèle suivant :

- une ligne de commande concerne une pizza, pour une certaine quantité et un certain prix (le prix n'est pas seulement la quantité fois le prix de la pizza, car le prix d'une certaine pizza peut varier dans le temps), et appartient à une commande,
- une commande est initiée à une certaine date, par un utilisateur (instance de PizzaUser), avec un prix total et un booléen qui dit si elle est payée.

En fait une commande non payée (il n'y en aura qu'une seule) constitue le panier, et dès que la commande est payée, le panier est vidé. Le panier n'est donc constitué que des lignes de la commande pas encore payée (éventuellement vide).

Nous allons créer une application applipanier, mettre en place les modèles, opérer les migrations.

Ensuite nous mettrons en place toutes les fonctionnalités du panier.

Même si on ne le précise pas, pensez à sécuriser les actions et les views !

## L'application applipanier et ses modèles

1. Créez l'application applipanier comme on l'a fait pour applipizza et applicompte. N'oubliez pas de l'inclure dans INSTALLED\_APPS.
2. Créez les deux modèles Commande et LigneCommande comme décrits dans l'introduction.
3. Mettez en place la migration et vérifiez que la base de données contient les nouvelles tables.

## Affichage du panier

1. Pour tester l'affichage du panier, créez un panier directement dans sqlitebrowser :

Table : applipanier\_commande

	idCommande	dateCommande	payee	prix	pizzauser_id
	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	2023-07-19	0	45.90	9

Table : applipanier\_lignecommande

	idLigneCommande	quantite	prix	commande_id	pizza_id
	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	2	31.1	1	1
2	2	1	14.8	1	2

Vous noterez que c'est un panier puisque la commande n'est pas payée. Ce panier contient 2 achats :

- 2 pizzas d'identifiant 1,
- 1 pizza d'identifiant 2.

Ici, c'est le client d'identifiant 9 qui a rempli ce panier.

2. On code ici la view `afficherPanier(request)` :

a. On récupère le `PizzaUser` :

```
user = PizzaUser.objects.get(id = request.user.id)
```

b. Ensuite on récupère les commandes non payées de ce client (ceci renvoie un ensemble qui contient 0 ou 1 commande) :

```
lesCommandesNonPayees = Commande.objects.filter(pizzauser = user).filter(payee = False)
```

c. Si cet ensemble n'est pas vide, alors le panier est l'élément d'indice 0 de cet ensemble, et les lignes du panier sont les entrées de la table `LigneCommande` correspondant à l'identifiant `idCommande` du panier :

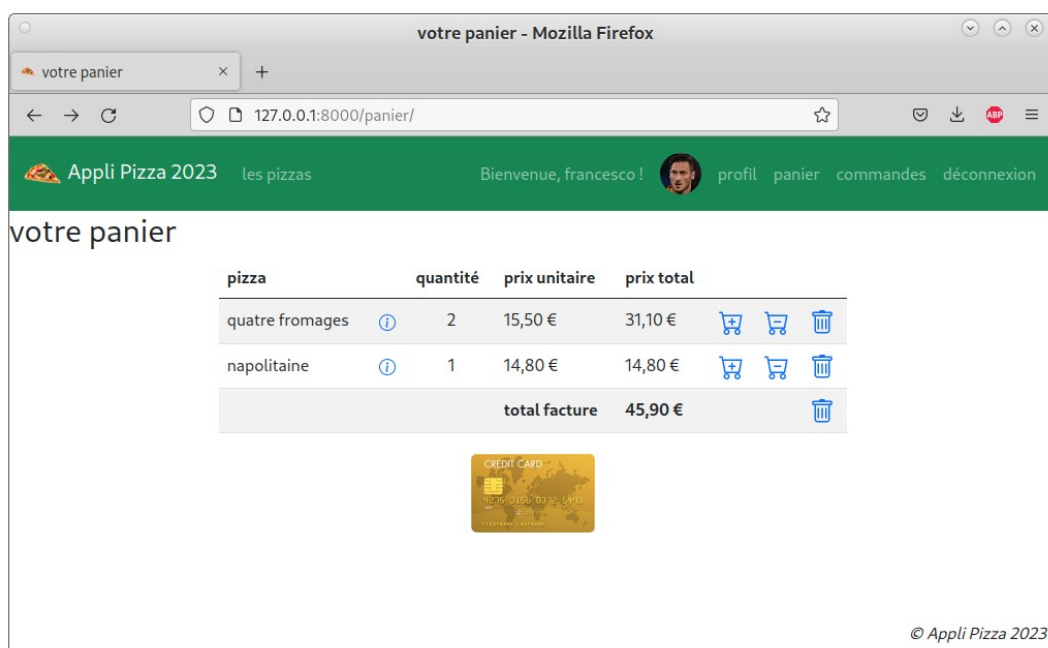
```
panier = lesCommandesNonPayees[0]
```

```
lesLignesDuPanier = LigneCommande.objects.filter(commande_id = panier.idCommande)
```

d. Sinon, `panier` et `lesLignesDuPanier` sont mises à `None`.

e. Enfin, on retourne le `render` correspondant au template `panier.html` (pas encore écrit), avec pour contenu le `user`, le panier et les lignes du panier.

3. On code maintenant le template `panier.html`. Voici ce à quoi cela doit ressembler :



Les éléments importants de ce template :

- a. le tableau : il est construit à partir des lignes de la variable de gabarit `lesLignesDuPanier`, dans une boucle `for`,
- b. Les liens d'information : ils renvoient sur la page de détail de la pizza en question,
- c. Le lien pour vider tout le panier,
- d. Les liens pour augmenter ou diminuer les quantités d'une pizza du tableau. Si la quantité passe à 0, la ligne est supprimée.
- e. Les totaux des lignes et le total global s'actualisent à toute action.

En autonomie, mettez tous ces éléments en place. Les liens doivent tous être écrits, même si un clic sur un lien déclenche pour le moment une erreur. Ces liens peuvent être de la forme :

- `href="/pizzas/{{ligne.pizza.idPizza}}/"`
- `href="/pizzas/{{ligne.pizza_id}}/buy/"`
- `href="/cart/{{ligne.pizza_id}}/decrease/"`
- `href="/cart/{{ligne.pizza_id}}/delete/"`
- `href="/cart/delete/"`
- `href="/cart/pay/"`

Les urls et les views seront vus dans les questions suivantes.

## Ajouter une pizza au panier

1. Pour ajouter une pizza au panier :

Dans la liste des pizzas on a déjà les liens de la forme

```
href="/pizzas/{ {piz.idPizza} }/buy/"
```

Créez l'url :

```
path('pizzas/<int:pizza_id>/buy/', views.ajouterPizzaAuPanier),
```

2. Créez la view

```
ajouterPizzaAuPanier(request, pizza_id)
```

Vous respecterez l'algorithme suivant :

- a. on récupère le PizzaUser,
- b. on récupère la Pizza à ajouter,
- c. on récupère le panier actuel, au besoin on le crée,
- d. on ajoute au prix du panier celui de la pizza, et on sauvegarde le panier,
- e. on récupère les lignes de ce panier, puis la ligne du panier concernant cette pizza,
- f. si une telle ligne existe, on augmente sa quantité de 1, son prix du prix de la pizza, et on sauvegarde cette ligne
- g. sinon, on la crée, on affecte à son attribut pizza la pizza ajoutée, à sa quantité la valeur 1, à son prix le prix de la pizza, à son attribut commande le panier en cours, et on sauvegarde la ligne,
- h. Ensuite on rerécupère les lignes du panier (on actualise),
- i. ensuite, on retourne le render avec comme template panier.html, et comme contenu les lignes du panier, le panier et le user,

3. Vérifiez que l'action d'achat sur la page des pizzas actualise bien le panier.

4. Si votre template panier.html présente le bon lien, le clic sur l'icône



doit aussi augmenter la quantité pour cette pizza. Vérifiez.

## Supprimer une ligne du panier

1. Pour supprimer une ligne du panier :  
Dans le tableau du panier on a déjà les liens de la forme

`href="/cart/{{ligne.pizza_id}}/delete/"`

Créez l'url :

`path('cart/<int:pizza_id>/delete/', views.retirerDuPanier),`

2. Créez la view

`retirerDuPanier(request, pizza_id)`

Vous respecterez l'algorithme suivant :

- a. on récupère le PizzaUser,
  - b. on récupère le panier actuel,
  - c. on récupère la pizza à retirer,
  - d. on récupère la ligne du panier qui concerne cette pizza,
  - e. on récupère la quantité de cette ligne,
  - f. on supprime la ligne,
  - g. on modifie le prix total du panier en lui retirant la quantité récupérée précédemment multipliée par le prix de la pizza supprimée,
  - h. on sauvegarde le panier,
  - i. on rerécupère les lignes du panier,
  - j. s'il n'y en a plus aucune, on supprime le panier, on le met à None, ainsi que lesLignesDuPanier,
  - k. ensuite, on retourne le render avec comme template panier.html, et comme contenu les lignes du panier, le panier et le user.
3. Vérifiez que l'action de suppression d'une ligne du panier actualise bien le panier (y compris quand cela vide complètement le panier).

## Vider le panier

1. Pour vider le panier, c'est très simple :  
Dans le tableau du panier on a déjà le lien de la forme

`href="/cart/delete/"`

Créez l'url :

```
path('cart/delete/', views.viderPanier),
```

2. Créez la view

```
viderPanier(request)
```

Vous respecterez l'algorithme suivant :

- a. on récupère le PizzaUser,
- b. on récupère le panier actuel,
- c. on supprime le panier, ce qui, par les contraintes en CASCADE, va supprimer les lignes de ce panier,
- d. on met le panier à None, ainsi que lesLignesDuPanier,
- e. ensuite, on retourne le render avec comme template panier.html, et comme contenu les lignes du panier, le panier et le user.

3. Vérifiez que l'action de suppression du panier le vide bien.

## Retirer une pizza du panier

1. Pour retirer une pizza du panier :  
Dans le tableau du panier on a déjà le lien de la forme

`href="/cart/{ligne.pizza_id}/decrease/"`

Créez l'url :

```
path('cart/<int:pizza_id>/decrease/', views.retirerUnePizzaDuPanier),
```

## 2. Créez la view

### `retirerUnePizzaDuPanier(request, pizza_id)`

Vous respecterez l'algorithme suivant :

- a. on récupère le `PizzaUser`,
- b. on récupère le panier actuel,
- c. on récupère la pizza à retirer,
- d. on récupère la ligne du panier qui concerne cette pizza,
- e. on récupère la quantité de cette ligne,
- f. on modifie le prix total du panier en lui retirant le prix de la pizza supprimée,
- g. si la quantité récupérée est 1, on supprime la ligne
- h. sinon on diminue la quantité de 1, on diminue le prix de la ligne du prix de la pizza, et on sauvegarde la ligne,
- i. on sauvegarde le panier,
- j. on rerécupère les lignes du panier,
- k. s'il n'y en a plus aucune, on supprime le panier, on le met à `None`, ainsi que `lesLignesDuPanier`,
- l. ensuite, on retourne le `render` avec comme template `panier.html`, et comme contenu les lignes du panier, le panier et le user.

3. Vérifiez que l'action de retrait d'une pizza du panier met correctement à jour le panier.

## Valider (payer) le panier

1. Pour valider le panier (le payer) :  
Dans le tableau du panier on a déjà le lien de la forme

`href="/cart/pay/"`

Créez l'url :

`path('cart/pay/', views.payerPanier),`

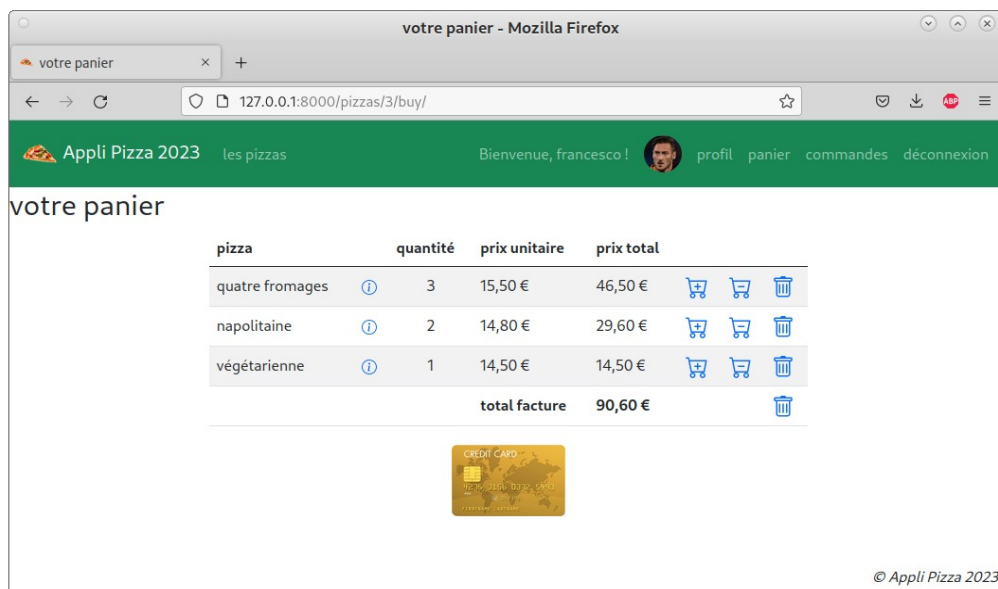


## 2. Créez la view

`payerPanier(request)`

Vous respecterez l'algorithme suivant :

- on récupère le `PizzaUser`,
- on récupère le panier actuel,
- on passe l'attribut "payee" du panier à `True`,
- on sauvegarde le panier,
- ensuite, on retourne le `render` avec comme template `avisPaielement.html`, et comme contenu le panier et le user.

3. Créez le template `avisPaielement.html` qui informe le client, comme dans les captures suivantes :

## 4. Testez l'action de validation du panier.


## Commander une pizza à partir de la page de détails

Pour terminer ce TP, faites en sorte que la page de détails d'une pizza affiche un bouton qui permet de l'acheter. Ce bouton ne sera affiché que si l'utilisateur est un client connecté.

détails d'une pizza - Mozilla Firefox


détails d'une pizza x +

← → ↻ 127.0.0.1:8000/pizzas/2/ ☆

🍕 Appli Pizza 2023 les pizzas Bienvenue, francesco !  profil panier commandes déconnexion

### voici notre pizza

pizza napolitaine (prix : 14.80 €)



ajouter au panier

### les 5 ingrédients de la pizza napolitaine

ingrédient	quantité
tomate	200 grammes
mozzarella	200 grammes
basilic	à volonté
viande hachée	200 grammes
crème fraîche	2 cuillérées à soupe

© Appli Pizza 2023