

# TP7 – Formulaire de composition de pizza

## Contexte et Objectifs

Nous savons maintenant mener à bien toutes les étapes qui permettent de créer un formulaire simple (pizza, ingrédient) à partir du `modelForm` qui permet une cohérence totale entre les classes et les tables de données.

Pour les compositions, nous allons créer un formulaire de même type, qui proposera une liste (type « select ») d'ingrédients pouvant entrer dans la composition d'une pizza.

Ce formulaire ne fera pas l'objet d'une page à part, comme pour la création d'une pizza ou d'un ingrédient, mais sera directement intégré à la page de détail d'une pizza, sous le tableau des ingrédients de la pizza. C'est plus simple car nous verrons ainsi évoluer en direct le tableau des ingrédients.

## Création d'un nouveau formulaire

Dans le fichier `forms.py`, créez une nouvelle classe de formulaire :

a. son nom est `CompositionForm`, il hérite de `ModelForm`

b. ses deux champs sont :

- `ingredient`, qui donnera un select dont les choix seront tirés de la table `ingrédients`,
- `quantite`, dont la valeur sera une chaîne de caractères à remplir par l'utilisateur.

```
forms.py - webpizza - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
webpizza
  <> formulaireCreationPiz...
  <> formulaireModificatio...
  <> formulaireModificatio...
  <> ingredients.html
  <> pizza.html
  > OUTLINE
  > TIMELINE

forms.py
12     model = Pizza
13     fields = ['nomPizza', 'prix']
14
15     class CompositionForm(ModelForm) :
16         class Meta :
17             model = Composition
18             fields = ['ingredient', 'quantite']
19
20
```

## Modification de la vue formulairePizza dans le fichier views.py

Modifiez la vue formulairePizza :

- créez une instance de CompositionForm
- intégrez ce formulaire dans le contenu à transmettre au template

```
views.py - webpizza - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
webpizza
  <> formulairePizza.h...
  <> ingredients.html
  <> pizza.html
  <> pizzas.html
  <> traitementFormul...
  <> traitementFormul...
  <> __init__.py
  <> admin.py
  <> apps.py
  <> forms.py
  <> models.py
  <> tests.py
  <> views.py
  > webpizza
    > __pycache__
    <> __init__.py
    <> asgi.py
    <> settings.py
    <> urls.py
    <> wsgi.py
  > OUTLINE
  > TIMELINE

views.py
39 def pizza(request, pizza_id) :
40
41     # le formulaire d'ajout d'un ingrédient
42     formulaire = CompositionForm()
43
44     # récupération de la pizza dont l'idPizza a été passé en
45     # paramètre dans l'url définie dans urls.py
46     laPizza = Pizza.objects.get(idPizza = pizza_id)
47
48     # récupération des ingrédients entrant dans la composition de la pizza
49     compo = Composition.objects.filter(pizza = pizza_id)
50
51     # la liste des ingrédients, avec nom et quantité
52     listeIngredients = []
53
54     for c in compo :
55         ing = Ingredient.objects.get(idIngredient = c.ingredient.idIngredient)
56         listeIngredients.append({"nom" : ing.nomIngredient, "qte" : c.quantite})
57
58     # on retourne l'emplacement de la vue et la pizza récupérée
59     return render(
60         request,
61         "applipizza/pizza.html",
62         {"pizza" : laPizza, "liste" : listeIngredients, "form" : formulaire}
63     )
64
```

## Modification du template pizza.html

Sous le tableau des ingrédients de la pizza, insérez une balise formulaire qui affichera celui créé par la vue formulairePizza. Ce formulaire aura pour action la chaîne suivante :

```
<form action="/pizzas/{pizza.idPizza}/addIngredient/" method="post">
```

Pour le moment, l'url décrite et la vue de traitement du formulaire ne sont pas écrites. Néanmoins, vous devriez avoir un rendu comme :

détails d'une pizza – Chromium

détails d'une pizza x +

127.0.0.1:8000/pizzas/2/

les pizzas les ingrédients créer une pizza créer un ingrédient

### voici notre pizza

- pizza napolitaine (prix = 14.80 €)

### les 5 ingrédients de la pizza napolitaine

ingrédient	quantité	action
tomate	200 grammes	<a href="#">supprimer</a>
mozzarella	200 grammes	<a href="#">supprimer</a>
basilic	à volonté	<a href="#">supprimer</a>
viande hâchée	200 grammes	<a href="#">supprimer</a>
crème fraîche	2 cuillères à soupe	<a href="#">supprimer</a>

### ajouter un ingrédient

L'ingrédient :

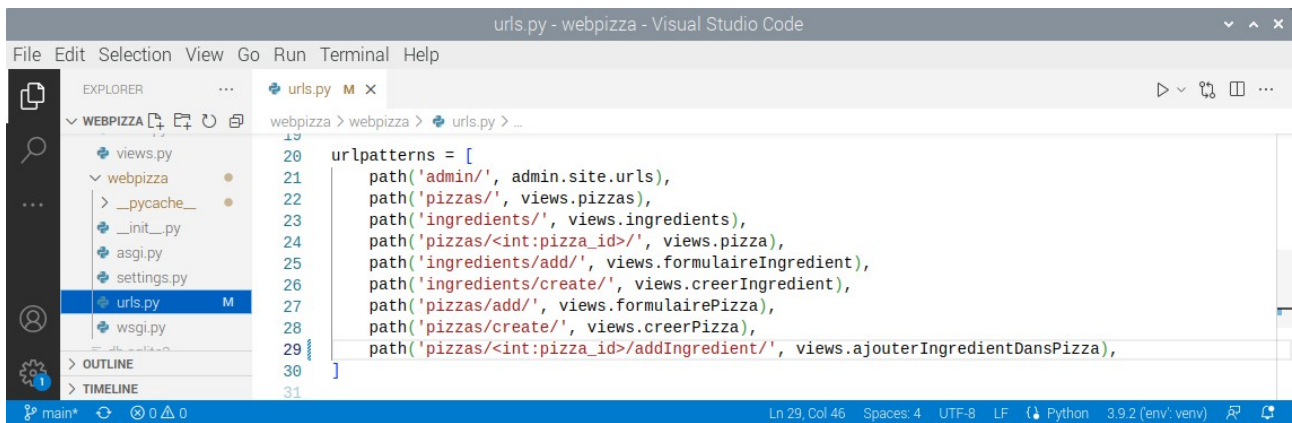
La quantité :

Envoyer

Appli Pizza 2023

## Création de l'url

Modifiez le fichier urls.py pour renseigner le nouveau path :



```
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('pizzas/', views.pizzas),
23     path('ingredients/', views.ingredients),
24     path('pizzas/<int:pizza_id>', views.pizza),
25     path('ingredients/add/', views.formulaireIngredient),
26     path('ingredients/create/', views.creerIngredient),
27     path('pizzas/add/', views.formulairePizza),
28     path('pizzas/create/', views.creerPizza),
29     path('pizzas/<int:pizza_id>/addIngredient/', views.ajouterIngredientDansPizza),
30 ]
31
```

## Création de la vue ajouterIngredientDansPizza

Ajoutez la « vue » ajouterIngredientDansPizza dans le fichier views.py. Cette vue ressemble beaucoup à la vue pizza. Elle :

- recupère le formulaire posté,
- crée la nouvelle instance de Composition pour l'ingrédient entrant dans la composition de la pizza, avec sa quantité,
- l'insère dans la base,
- recréé le tableau des ingrédients et un nouveau formulaire d'ajout d'ingrédient.

Comme c'est un peu compliqué (en particulier les valeurs des attributs de la composition créée), voici un exemple de code :

```
views.py - webpizza - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
webpizza
├── env
├── webpizza
├── applipizza
│   ├── __pycache__
│   ├── migrations
│   ├── static
│   └── templates/a...
│       ├── base.html
│       ├── formulaireIngredi...
│       ├── formulairePizza.h...
│       ├── ingredients.html
│       ├── pizza.html
│       ├── pizzas.html
│       ├── traitementFormul...
│       └── traitementFormul...
├── __init__.py
├── admin.py
├── apps.py
├── forms.py
├── models.py
├── tests.py
├── views.py
├── webpizza
│   ├── __pycache__
│   ├── __init__.py
│   ├── asgi.py
│   ├── settings.py
│   ├── urls.py
│   ├── wsgi.py
│   ├── db.sqlite3
│   ├── manage.py
│   ├── .gitignore
│   └── README.md
└── OUTLINE
    └── TIMELINE

urls.py M
views.py M X

webpizza > applipizza > views.py > ajouterIngredientDansPizza

152
153 def ajouterIngredientDansPizza(request, pizza_id) :
154
155     # récupération du formulaire posté
156     formulaire = CompositionForm(request.POST)
157
158     if formulaire.is_valid() :
159         # récupération des données postées
160         idIng = formulaire.cleaned_data['ingredient']
161         qte = formulaire.cleaned_data['quantite']
162
163         # création de la nouvelle instance de Composition
164         # et remplissage des attributs
165         compo = Composition()
166         compo.ingredient = Ingredient.objects.get(idIngredient = idIng)
167         compo.pizza = Pizza.objects.get(idPizza = pizza_id)
168         compo.quantite = qte
169
170         # sauvegarde dans la base de la composition
171         compo.save()
172
173     # le nouveau formulaire d'ajout d'un ingrédient
174     formulaire = CompositionForm()
175
176     # récupération de la pizza dont l'idPizza a été passé en
177     # paramètre dans l'url définie dans urls.py
178     laPizza = Pizza.objects.get(idPizza = pizza_id)
179
180     # récupération des ingrédients entrant dans la composition de la pizza
181     compo = Composition.objects.filter(pizza = pizza_id)
182
183     # la liste des ingrédients, avec nom et quantité
184     listeIngredients = []
185
186     for c in compo :
187         ing = Ingredient.objects.get(idIngredient = c.ingredient.idIngredient)
188         listeIngredients.append({"nom" : ing.nomIngredient, "qte" : c.quantite})
189
190     # on retourne l'emplacement de la vue et la pizza récupérée
191     return render(
192         request,
193         "applipizza/pizza.html",
194         {"pizza" : laPizza, "liste" : listeIngredients, "form" : formulaire}
195     )
```

Vérifiez que tout fonctionne, et que le tableau s'enrichit au fur et à mesure.