

## TP7 – Formulaire de composition de pizza

### Introduction, Contexte et Objectifs

Nous savons maintenant mener à bien toutes les étapes qui permettent de créer un formulaire simple, mis en forme avec bootstrap, et récupérable selon un modèle de formulaire (IngredientForm, PizzaForm), qui permet une cohérence totale entre les modèles 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.

Inspirez-vous des formulaires PizzaForm et ingredientForm.

## Modification de la view pizza dans le fichier views.py

Modifiez la view pizza :

- récupérez tous les ingrédients par une requête `all()`, ils serviront à construire un select de formulaire,
- prenez cette liste d'ingrédients au contenu à transmettre au template, en plus de ce qui était déjà transmis.

## Modification du template pizza.html

Sous le tableau des ingrédients de la pizza, insérez un formulaire composé d'un select d'ingrédients (avec son label) et d'un input pour saisir la quantité. Vous noterez que ce formulaire est à la mode bootstrap, et que le select est construit dans une boucle `for` à partir de la liste des ingrédients calculée dans la view précédente. Le code peut être le suivant :

```
<h2>ajouter un ingrédient</h2>
<form action="/pizzas/{{pizza.idPizza}}/addIngredient/" method="post">
  {% csrf_token %}
  <div class="mb-3">
    <label for="id_ingredient" class="form-label">ingrédient </label>
    <select name="ingredient" class="form-select" required id="id_ingredient">
      <option selected disabled>choisissez</option>
      {% for ing in lesIng %}
      <option value="{{ing.idIngredient}}">{{ing.nomIngredient}}</option>
      {% endfor %}
    </select>
  </div>
  <div class="mb-3">
    <label for="id_quantite" class="form-label">quantité </label>
    <input type="text" name="quantite" class="form-control" maxlength="100" required id="id_quantite">
  </div>
  <button type="submit" class="btn btn-primary">Envoyer</button>
</form>
```

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 - Mozilla Firefox

détails d'une pizza x +

127.0.0.1:8000/pizzas/2/

Appli Pizza 2023 les pizzas créer une pizza les ingrédients créer un ingrédient

voici notre pizza

pizza napolitaine (prix : 14.80 €)

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

ajouter un ingrédient

ingrédient

choisissez l'ingrédient à ajouter

quantité

Envoyer

© Appli Pizza 2023

## Création de l'url

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

```
urls.py - webpizza - Visual Studio Code
```

```
File Edit Selection View Go Run Terminal Help
```

```
EXPLORER
```

```
webpizza > webpizza > urls.py > ...
```

```
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.formulaireIngrédient),
26     path('ingredients/create/', views.creerIngrédient),
27     path('pizzas/add/', views.formulairePizza),
28     path('pizzas/create/', views.creerPizza),
29     path('pizzas/<int:pizza_id>/addIngrédient/', views.ajouterIngrédientDansPizza),
30 ]
31
```

main\* 0 0 0 Ln 29, Col 46 Spaces: 4 UTF-8 LF Python 3.9.2 (env: venv)

## Création de la view ajouterIngredientDansPizza

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

- recupère le formulaire posté,
- si l'ingrédient est déjà présent dans la composition de la pizza, efface la ligne correspondante,
- crée la nouvelle instance de `Composition` pour l'ingrédient entrant dans la composition de la pizza, avec sa quantité,
- l'insère dans la base,
- recrée le tableau des ingrédients et prépare un nouveau formulaire d'ajout d'ingrédient.
- Rappelle le template `pizza.html`

C'est un peu compliqué (les valeurs des attributs de la composition créée), donc voici un exemple de code, commenté :

```
def ajouterIngredientDansPizza(request, pizza_id) :
    # récupération du formulaire posté
    formulaire = CompositionForm(request.POST)

    if formulaire.is_valid() :
        # récupération des données postées
        ing = formulaire.cleaned_data['ingredient']
        qte = formulaire.cleaned_data['quantite']
        piz = Pizza.objects.get(idPizza = pizza_id)
        compoPizza = Composition.objects.filter(pizza = pizza_id)
        lesIngredientsDeLaPizza = ((ligne.ingredient) for ligne in compoPizza)

        if ing in lesIngredientsDeLaPizza :
            compo = Composition.objects.filter(pizza = pizza_id, ingredient = ing)
            compo.delete()

        # création de la nouvelle instance de Composition et remplissage des attributs
        compo = Composition()
        compo.ingredient = ing
        compo.pizza = piz
        compo.quantite = qte
        # sauvegarde dans la base de la composition
        compo.save()

        # récupération de tous les ingrédients pour construire le futur select
        lesIngredients = Ingredient.objects.all()
        # actualisation des ingrédients entrant dans la composition de la pizza
        compoPizza = Composition.objects.filter(pizza = pizza_id)
        # on crée une liste dont chaque item contiendra l'identifiant de la composition (idComposition),
        # le nom de l'ingrédient et la quantité de l'ingrédient dans cette composition
        listeIngredients = []
        for ligneCompo in compoPizza :
            # on récupère l'Ingredient pour utiliser son nomIngredient
            ingredient = Ingredient.objects.get(idIngredient = ligneCompo.ingredient.idIngredient)
            listeIngredients.append(
                {"idComposition" : ligneCompo.idComposition, "nom" : ingredient.nomIngredient, "qte" : ligneCompo.quantite}
            )
        # on retourne l'emplacement du template, la pizza récupérée et la liste des ingrédients calculée ci-dessus
        return render(
            request,
            'applipizza/pizza.html',
            {"pizza" : piz, "liste" : listeIngredients, "lesIng" : lesIngredients}
        )
```

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