Projet 8 - Créez une plate-forme pour amateurs de Nutella

Parcours OpenClassrooms - Développeur d'application Python

Étudiant : Guillaume OJARDIAS Mentor : Erwan KERIBIN
Mentor évaluateur : Babacar SYLLA

Présentation

Démarche de création

Bilan du projet

Présentation

Objectif

- Trouver en un clic un produit de substitution

Cahier des charges

- Fonctionnalités :
  - Avoir un compte utilisateur
  - Sauvegarder des substituts dans ses favoris
  - Faire une recherche dès la page d'accueil
- Design :
  - Esquisses de disposition des pages
  - Charte graphique
  - Utilisation du template Creative proposé par StartBootstrap

Résultat final

- http://projet-8.ojardias.io

Démarche de création

Étapes du projet

- Création du projet Django
- Mise en place de le gestion des utilisateurs => `User` personnalisé
- Création des modèles de données (`Product` et `Category`)
- Insertion des produits dans la BDD (commande `init_db`)
- Mise en place du système de recherche des substituts
- Ajout de la fonctionnalité de mise en favoris
- Ajout des propositions lors de la recherche

Code source

- Python + Django
- GitHub -> Pur-Beurre

Arborescence du projet

```
|- config                  #- Configuration du projet
|- homepage                ##
|- openfoodfacts            #
|- product                  # Applications
|- search                   #
|- users                   ##
|- static                  #- Fichiers statiques
|- templates               #- Templates généraux
|- tests                   ##
    |- homepage             #
    |- openfoodfacts        #
    |- product              # Tests
    |- search               #
    |- users               ##
```

Algorithme de recherche d'un substitut

- Filtre pour le calcul des catégories en commun

```
q = Q(categories__in=product.categories.all()) & Q(
    nutriscore_grade__lt=product.nutriscore_grade))
```

- Requête pour obtenir les substituts

```
substitutes = (Product.objects.annotate(
    common_categories=Count("categories", filter=q))
        .order_by("-common_categories", "nutriscore_grade")
        .exclude(code=product.code)
        .exclude(name=product.name)[:30])
```

Tests sur la vue auto_completion de l'application search (1/3) :

- Mock de la class Product :

```
class MockProduct:
    def __init__(self, products, substitutes=""):
        # Mock de l'attribut "objects"
        self.objects = MockProductManager(
            products, substitutes
        )
```

Tests sur la vue auto_completion de l'application search (2/3) :

- Mock de la class ProductManager :

```
class MockProductManager:
    def __init__(self, products, substitutes):
        self.products = products
        self.substitutes = substitutes

    { ··· }

    # Mock la méthode
    def get_products_by_name(self, *args, **kwargs):
        return self.products
```

Tests sur la vue auto_completion de l'application search (3/3) :

- Utilisation de MockProduct :

```
def test_auto_completion_return_json(self):
    # Instantiation du Mock
    mock_product = MockProduct(self.products)
        .objects.get_products_by_name
    # Patch du Mock
    with patch("product.models.Product", mock_product):
    # Appel de la vue
    response = self.client.post(
        reverse("search:auto_completion"),
        data={"name": "nut"},
    )
    self.assertEqual(response.status_code, 200)
    self.assertTrue(loads(response.content)
        .get("products_names"))
```

Bilan du projet

Découpage du projet

- Perfectible
  - Changer le nom de `homepage`
  - Regrouper les templates
- Penser son découpage en amont
  - Changement en cours difficile
  - Facilité avec l'expérience

Tests

- **Coverage** est un très bon outil
  - Utile pour mettre en relief les zones non testées
  - Attention à avoir un esprit critique
- Le **Test Driven Development**
  - Indispensable
  - Difficile pour un débutant
  - Viens avec le temps et l'expérience

Django

- Nombreux outils disponibles *out of the box*
  - Users
  - Admin
  - API (non utilisé sur ce projet)
- Part de *magie* appréciée par certains
- Très puissant pour une utilisation poussée

Fin

- Merci pour votre attention