

Concepts fondamentaux :

Routage explicite

Le routage explicite dans Django se fait via le fichier `urls.py`.

Django adopte un routage explicite, c'est-à-dire que chaque route doit être déclarée manuellement dans un fichier Python, sans détection automatique des endpoints.

(Pour rappel : Un endpoint (ou point de terminaison) est une adresse d'accès à une ressource ou une fonctionnalité d'une application, autrement dit un endpoint = une URL + une méthode HTTP")

Le routage explicite favorise la lisibilité et la maintenance à grande échelle.

Il permet aussi de **découper le routage par application**, ce qui améliore la modularité.

Le routage explicite renforce la **séparation des responsabilités** :

- le contrôleur logique (vue) est dans `views.py`
- la déclaration du chemin d'accès est dans [urls.py](#)

Exemple :

```
# projet/urls.py
from django.urls import path
from app.views import liste_articles, detail_article

urlpatterns = [
    path('articles/', liste_articles, name='liste_articles'),
    path('articles/<int:id>', detail_article, name='detail_article'),
]
```

Les formulaires (Forms)

Les formulaires Django permettent de récupérer, valider et traiter des données utilisateur de manière sécurisée et cohérente.

Ils constituent un pont entre :

- les templates (affichage HTML du formulaire),
- les vues (logique de traitement),
- et les modèles (enregistrement des données).

Deux types de formulaires principaux

forms.Form

→ Pour créer un formulaire manuel, indépendant d'un modèle.

Exemple :

```
1 from django import forms
2
3 class ContactForm(forms.Form):
4     nom = forms.CharField(max_length=50)
5     email = forms.EmailField()
6     message = forms.CharField(widget=forms.Textarea)
7 |
```

forms.ModelForm

→ Pour créer un formulaire lié directement à un modèle de la base de données.

Exemple :

```
1 from django import forms
2 from .models import Article
3
4 class ArticleForm(forms.ModelForm):
5     class Meta:
6         model = Article
7         fields = ['titre', 'contenu']
8
```

Cycle de vie d'un formulaire Django

1. Affichage du formulaire dans le template :

```
4 <form method="POST">
5     {% csrf_token %}
6     {{ form.as_p }}
7     <button type="submit">Envoyer</button>
8 </form>
9
```

2. Soumission : Django reçoit les données via une requête POST.

3. Validation automatique :

- Les champs obligatoires, formats d'e-mail, tailles, etc. sont vérifiés.
- Si le formulaire est valide → on peut enregistrer les données.
- Sinon → les erreurs sont renvoyées au template.

4. Traitement dans la vue :

```
1 from django.shortcuts import render, redirect
2 from .forms import ArticleForm
3
4 def ajouter_article(request):
5     if request.method == "POST":
6         form = ArticleForm(request.POST)
7         if form.is_valid():
8             form.save()
9             return redirect('liste_articles')
10    else:
11        form = ArticleForm()
12    return render(request, 'ajouter_article.html', {'form': form})
13
```