

Sprints



Sprint 1 — "Mise en place du front statique et logique de base"



Objectif général

👉 Apprendre à manipuler les **views**, **templates**, **urls**, et la **gestion des fichiers statiques** dans Django à travers l'app `core`.



Durée estimée

📅 1 à 2 jours (selon ton rythme)



Tâches détaillées



Tâche 1 — Vérifier la configuration du projet

But : t'assurer que ton projet Django reconnaît bien les fichiers statiques et les templates.

Actions à faire :

- Vérifier dans `settings.py` :
 - que `APP_DIRS: True` est bien défini dans `TEMPLATES`
 - que `INSTALLED_APPS` contient bien `'apps.core'`
 - que `STATIC_URL` est configuré (`'static/'`)
- Créer le dossier `apps/core/templates/core/`
- Créer le dossier `apps/core/static/core/` (pour CSS, images, etc.)

Notions à manipuler :

- `BASE_DIR` , `TEMPLATES` , `STATIC_URL`
- Structure des dossiers Django

Livrable :

- Projet fonctionnel avec `runserver` sans erreur
-



Tâche 2 — Créer la page d'accueil (Home)

But : apprendre à lier une vue, une URL, et un template.

Actions à faire :

- Créer une vue `home_view(request)` dans `apps/core/views.py`
- Ajouter l'URL correspondante dans `apps/core/urls.py`
- Créer le fichier `home.html` dans `apps/core/templates/core/`
- Afficher un message simple : "Bienvenue sur KeepNotes"
- Lancer le serveur et tester sur `http://127.0.0.1:8000/`

Indices / Notions à utiliser :

- `HttpResponse` , `render(request, 'core/home.html')`
- URLs Django (`path()`)
- Fonction `views` (FBV)

Livable :

- Une page d'accueil affichée correctement via une vue
-



Tâche 3 — Créer une page "À propos"

But : pratiquer la création d'une deuxième page statique.

Actions à faire :

- Créer `about_view(request)`
- Créer le template `about.html`
- Ajouter une balise `<a>` dans `home.html` vers la page "À propos"
- Relier l'URL `/about/` à la vue correspondante

Indices / Notions à utiliser :

- Routage multiple dans `urls.py`
- Liens internes avec `{% url 'core:about' %}`

Livable :

- Navigation entre "Accueil" et "À propos"
-

Tâche 4 — Créer un `base.html`

But : mettre en place un template parent pour factoriser le code HTML.

Actions à faire :

- Créer `base.html` dans `core/templates/core/`
- Y définir :

```
<!DOCTYPE html>
<html>
<head>
  <title>{% block title %}KeepNotes{% endblock %}</title>
  <link rel="stylesheet" href="{% static 'core/style.css' %}">
</head>
<body>
  <header>
    <h1>KeepNotes</h1>
    <nav>
      <a href="{% url 'core:home' %}">Accueil</a>
      <a href="{% url 'core:about' %}">À propos</a>
    </nav>
  </header>
  <main>
    {% block content %}{% endblock %}
  </main>
</body>
</html>
```

- Modifier `home.html` et `about.html` pour étendre `base.html`

Indices / Notions :

- `{% extends 'core/base.html' %}`
- `{% block content %}`
- `{% load static %}`

Livrable :

- Les pages utilisent un template parent commun

🎨 Tâche 5 — Ajouter un fichier CSS statique

But : tester la gestion des fichiers statiques.

Actions à faire :

- Créer `apps/core/static/core/style.css`
- Y ajouter un style simple (couleur, marges, etc.)
- Charger le fichier dans `base.html` via `{% load static %}`

Notions à utiliser :

- `{% load static %}`
- `STATICFILES_DIRS` si nécessaire

Livable :

- Apparence visuelle minimale sur le site

✅ Tâche 6 — Personnaliser la page 404 (optionnelle)

But : découvrir comment Django gère les erreurs.

Actions à faire :

- Créer `404.html` dans `core/templates/core/`
- Simuler une erreur pour tester l'affichage

Notions à explorer :

- Gestion des erreurs personnalisées dans Django

Livable :

- Page 404 personnalisée

🏁 Livrables finaux du Sprint 1

N°	Livable	Description
1	Projet fonctionnel sans erreur	Toutes les apps bien configurées
2	Page d'accueil (<code>home.html</code>)	Affichée depuis une vue et un template
3	Page "À propos" (<code>about.html</code>)	Accessible via la navigation

N°	Livrable	Description
4	<code>base.html</code>	Template parent avec héritage
5	CSS statique fonctionnel	Fichier chargé via <code>{% static %}</code>
6	Navigation entre les pages	Liens fonctionnels entre home et about

Prochain Sprint (aperçu)

Sprint 2 – “Gestion des notes (CRUD)”

- Introduction des CBV (`ListView` , `DetailView` , `CreateView` , etc.)
- Formulaire et enregistrement des données
- Premiers templates dynamiques