

# Concepts fondamentaux : Système d'applications

Chaque app est une **unité logique et fonctionnelle**, regroupant ses propres :

- modèles (base de données),
- vues (logique métier),
- templates (interface),
- tests,
- fichiers de configuration (urls.py, admin.py, etc.).

Autrement dit, une *app* Django, c'est un **mini-module web complet**, que tu peux développer, tester et réutiliser ailleurs.

Chaque application peut être **autonome** :

Tu peux la copier-coller dans un autre projet Django sans tout casser, tant qu'elle est référencée dans `INSTALLED_APPS`.

Aspect	Apport
<b>Modularité</b>	Travail en équipe : chaque groupe peut travailler sur une app.
<b>Réutilisabilité</b>	Une app peut être extraite et réutilisée ailleurs (ex. "blog", "auth").
<b>Maintenance</b>	On peut tester et déployer une app indépendamment.
<b>Séparation des responsabilités</b>	Logique claire entre le projet global et les modules locaux.

## Les Tests (tests unitaires et d'intégration)

Chaque app Django inclut généralement un fichier tests.py (ou un dossier tests/) où tu peux écrire tes tests automatisés.

Ces tests permettent de vérifier :

- que les modèles enregistrent et récupèrent bien les données,
- que les vues renvoient les bonnes réponses HTTP,
- que les templates s'affichent correctement,
- que les formulaires et les URLs fonctionnent comme prévu.

**Exemple :**

```
tests.py
1 from django.test import TestCase
2 from .models import Article
3
4 class ArticleModelTest(TestCase):
5     def test_creation_article(self):
6         article = Article.objects.create(titre="Test", contenu="Ceci est un test")
7         self.assertEqual(article.titre, "Test")
8
```

**Exécution des tests :**

```
>>> python manage.py test
```

→ Django détecte automatiquement tous les fichiers de test dans les apps et exécute les scénarios.

## Limites à connaître

- Trop d'apps = risque de dépendances croisées (importations circulaires, logique dupliquée).
- Certaines configurations globales (authentification, templates, middleware) peuvent lier fortement les apps entre elles.
- Les tests mal structurés ou redondants peuvent ralentir le développement s'ils ne sont pas bien ciblés.