

■ Django : Modules (Apps) – Créer et intégrer

Structure type d'un projet Django

- Projet (merchex/) : dossier racine qui contient tout le code du projet.
- Sous-dossier de configuration (merchex/ à l'intérieur) : settings.py, urls.py, etc.
- Application (listings/) : un module fonctionnel réutilisable (ex. gestion des annonces).

1) Créer une application (module) : listings

Commande (Windows/cmd) :

```
python manage.py startapp listings
```

2) Installer l'app dans le projet

Édite merchex/settings.py et ajoute 'listings' dans INSTALLED_APPS :

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'listings', # ■ notre module  
]
```

3) Première vue (contrôleur minimal)

Dans listings/views.py :

```
from django.http import HttpResponse  
  
def home(request):  
    return HttpResponse("Listings – module intégré ■")
```

4) Routes de l'app (urls.py de l'app)

Crée listings/urls.py :

```
from django.urls import path  
from .views import home  
  
urlpatterns = [  
    path('', home, name='listings_home'),  
]
```

5) Brancher l'app dans les URLs du projet

Dans merchex/urls.py :

```
from django.contrib import admin  
from django.urls import path, include # ■ include  
  
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('listings/', include('listings.urls')), # ■ branche le module  
    path('', include('listings.urls')), # optionnel : page d'accueil = listings  
]
```

6) Migrations (si modèles plus tard)

```
python manage.py makemigrations  
python manage.py migrate
```

7) Lancer le serveur

```
python manage.py runserver
```

■ Résultat

- Ton module listings est créé, installé et accessible via /listings/.
- Tu peux maintenant ajouter des modèles (models.py), formulaires, templates et fichiers statiques propres à ce module.