

# Test d'Intégration API Frontend-Backend

---

## Configuration effectuée

### ☒ Services API mis à jour

- `productService.ts` : Adapté aux endpoints Django réels
- `categoryService.ts` : Nouveau service pour les catégories
- Configuration des endpoints dans `api.ts`

### ☒ Types TypeScript synchronisés

- Interface `Product` mise à jour selon le modèle Django
- Interface `Category` ajoutée
- Interface `ProductImage` ajoutée
- Filtres `ProductFilters` adaptés

### ☒ Hooks personnalisés créés

- `useProducts` : Gestion d'état des produits
- `useCategories` : Gestion d'état des catégories
- `useProduct` : Gestion d'état d'un produit individuel

### ☒ Page de test créée

- Route `/test-api` pour tester les endpoints
- Affichage des produits, catégories, et produits en vedette

## Comment tester

### 1. Démarrer le backend Django

```
cd Backend
python manage.py runserver
```

### 2. Démarrer le frontend React

```
cd Frontend
npm run dev
```

### 3. Accéder à la page de test

Ouvrir : `http://localhost:5173/test-api`

## Points à vérifier

### ☒ Endpoints testés

- `GET /api/products/` - Liste des produits
- `GET /api/products/?is_featured=true` - Produits en vedette
- `GET /api/categories/` - Liste des catégories
- `GET /api/categories/nested/` - Catégories imbriquées

### ☒ Données affichées

- ☐ Liste des produits avec images
- ☐ Informations produit (nom, slug, prix, catégorie)
- ☐ Liste des catégories actives
- ☐ Structure imbriquée des catégories
- ☐ Produits en vedette

## CORS à configurer

Si erreurs CORS, ajouter au Django `settings.py` :

```
CORS_ALLOWED_ORIGINS = [  
    "http://localhost:5173",  
    "http://127.0.0.1:5173",  
]
```

## Problèmes identifiés

### Composants legacy

- `InfiniteProductScroll` utilise encore l'ancienne interface
- `FloatingProductsGrid` nécessite mise à jour
- Solution temporaire : Fonction de conversion dans `productUtils.ts`

### TODO - Prochaines étapes

1. Mettre à jour tous les composants pour nouveaux types
2. Migrer HomePage vers vraies données API
3. Adapter ProductsPage et ProductDetailPage
4. Configurer gestion d'erreurs et loading states
5. Ajouter filtres avancés dans ProductsPage

## URLs de test importantes

- Page de test API : `http://localhost:5173/test-api`
- API Produits : `http://localhost:8000/api/products/`
- API Catégories : `http://localhost:8000/api/categories/nested/`

- Admin Django : <http://localhost:8000/admin/>