

DOCUMENTATION TECHNIQUE

EXPORT HEKTOR → APIMO

Client : Réseau Bonaparte

Auteur : Baptiste Fort

Date : Décembre 2025

Version : 1.0



Sommaire

1. Contexte du projet	Page 3
2. Problématique technique	Page 3
3. Solution développée	Page 4
4. Architecture technique	Page 5
5. Processus d'extraction	Page 6
6. Injection API Apimo	Page 7
7. Résultats & Statistiques	Page 8
8. Maintenance & Évolutions	Page 8

1. Contexte du Projet

Le réseau **Bonaparte Art de Vivre** gère un portefeuille immobilier important réparti sur deux plateformes distinctes :

- **Hektor** : Logiciel historique contenant 468 biens
- **Apimo** : Nouvelle plateforme de référence

L'objectif stratégique est de **centraliser 100% du portefeuille sur Apimo** pour unifier la gestion et optimiser les workflows commerciaux.

2. Problématique Technique

⚠ Obstacle majeur

Hektor ne propose aucune API d'export. La plateforme n'offre pas de fonctionnalité d'extraction massive des données. Aucune documentation technique n'est disponible pour accéder programmatiquement aux informations.

Options envisagées :

✗ **Export CSV natif**
Non disponible dans Hektor

✗ **API REST officielle**
Hektor refuse de fournir un accès API

✓ **Web Scraping automatisé**
Solution retenue : simulation comportement humain

3. Solution Développée

Approche : Simulation Comportement Humain

La solution consiste à piloter un navigateur Chrome réel via **Python + Selenium**, simulant les actions d'un utilisateur humain pour extraire les données de chaque bien.

Stack Technique :



Techniques Anti-Détection

Pour éviter d'être bloqué par les systèmes anti-bot de Hektor :

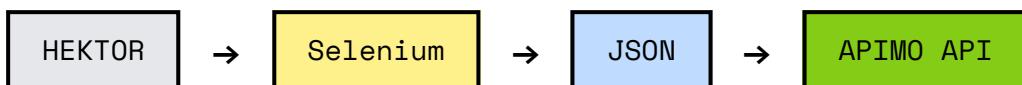
- **Délais aléatoires** entre 2 et 5 secondes entre chaque action
- **Simulation de scroll** progressif sur les pages
- **Mouvements de souris** naturels avant les clics
- **User-Agent rotatif** pour varier les signatures

4. Architecture Technique

```
# Structure du projet

hektor-apimo-export/
├── main.py # Point d'entrée
├── scraper/
│   ├── hektor.py # Extraction Hektor
│   └── parser.py # Parsing HTML
├── api/
│   └── apimo.py # Client API Apimo
├── models/
│   └── property.py # Modèle de données
├── utils/
│   ├── anti_detect.py # Comportement humain
│   └── logger.py # Logging
└── output/
    └── report.html # Rapport migration
```

Flux de données :



5. Processus d'Extraction

1

Authentification Hektor

Connexion automatique avec gestion des sessions et cookies.

```
driver.get("https://hektor.immo/login")
driver.find_element(By.ID, "email").send_keys(credentials.email)
time.sleep(random.uniform(1, 3))
driver.find_element(By.ID, "password").send_keys(credentials.password)
```

2

Navigation & Extraction

Parcours de chaque fiche bien et extraction des données.

```
for property_id in property_list:
    driver.get(f"/properties/{property_id}")
    html = driver.page_source
    data = parse_property(html) # BeautifulSoup
    properties.append(data)
```

3

Données Extraites

Titre	Prix	Surface
Photos	DPE/GES	GPS
Pièces	Chambres	Description

6. Injection API Apimo

Authentification API

```
# Configuration Apimo API
APIMO_API_URL = "https://api.apimo.pro"
APIMO_PROVIDER = "xxxxx"
APIMO_TOKEN = "Bearer {access_token}"

headers = {
    "Authorization": APIMO_TOKEN,
    "Content-Type": "application/json"
}
```

Création de bien

```
POST /agencies/{agency_id}/properties

{
    "category": 1, // Vente
    "type": 2, // Appartement
    "price": 450000,
    "area": { "value": 85 },
    "rooms": 4,
    "bedrooms": 2,
    "city": { "zipcode": "75008" }
}

// Response: 201 Created
```

7. Résultats & Statistiques

468

BIENS MIGRÉS

100%

TAUX RÉUSSITE

~4h

TEMPS EXÉCUTION

8. Maintenance & Évolutions

Recommandations :

1. Conserver les credentials Hektor pour migrations futures
2. Rapport HTML disponible pour vérification des données
3. Script réutilisable pour nouveaux biens si nécessaire

Mission terminée avec succès le 15/12/2025