

Cahier des Charges :

Partie 1 : Solution de Web Scraping pour les Annonces en Tunisie

1. Contexte et Objectifs

L'objectif de ce projet est de développer une solution de scraping permettant d'extraire les annonces publiées en Tunisie durant les mois de janvier et février 2025. Les annonces doivent porter sur un thème bien déterminé de votre choix (**immobilier**, véhicule, ...). Les données collectées devront être stockées et accessibles via une API REST.

2. Périmètre du Projet

- **Sites cibles :**
 - Tunisie Annonce (<https://www.tunisie-annonce.com/>)
 - Un ou plusieurs autres sites immobiliers selon le choix des étudiants.
- **Types d'annonces :** Vente et location (appartements, maisons, terrains, locaux commerciaux, etc.).
- **Format des données :** CSV, JSON ou Excel.
- **Livraison sous forme d'une API :** Une API REST exposant les données collectées.
- **Versionnement :** Utilisation de **Git/GitHub** pour le suivi du code.

3. Spécifications Techniques

3.1 Scraping des données

- Utilisation de **Selenium**, **BeautifulSoup** ou **Scrapy** pour l'extraction des données.
- Extraction des **informations clés** (exemple annonces immobilières) :
 - **Titre de l'annonce** vous trouverez le titre à côté de [Réf:
 - **Prix**
 - **Type de bien** (Appartement, maison, terrain, local commercial, etc.) **Nature**(Location, Terrain, Vente, Bureaux & Commerces)
 - **Localisation** (Ville, quartier) **Région**
 - **Superficie** **Surface**
 - **Description** de l'annonce **Texte**
 - **Contact du vendeur** (si disponible) si vous trouvez un numéro de 8 chiffres dans **Texte**
 - **Date de publication** **Modifiée**
 - **Lien vers l'annonce** y'a des liens dans **Texte annonce**

3.2 Stockage des données

- Sauvegarde des données sous format **CSV**, **JSON** ou **Excel**.
- Option de stockage en base de données **SQLite** ou PostgreSQL (bonus) ou mongodb.

3.3 Développement de l'API REST

- Développement avec **Flask ou FastAPI**.
- Endpoints de l'API :
 - GET /annonces → Retourne toutes les annonces collectées.
 - POST /scrape → Lance une nouvelle session de scraping.

3.4 Versionnement et Documentation

- **Code versionné sur GitHub** avec un fichier README expliquant :
 - L'installation et l'exécution de l'application.
 - La description des endpoints de l'API.
 - Les dépendances nécessaires (requirements.txt).

5. Livrables Attendues

- **Code source sur un dépôt GitHub.**
- **Script de scraping** permettant d'extraire les données.
- **Données extraites** en CSV, JSON ou Excel.
- **API REST opérationnelle** permettant d'extraire et d'accéder aux données.
- **Documentation** expliquant le fonctionnement de la solution.

6. Évaluation

L'évaluation du projet sera basée sur :

1. **Qualité du code** (modularité, clarté, respect des bonnes pratiques).
2. **Fonctionnalité de l'API** (respect des endpoints définis, filtres).
3. **Utilisation efficace de Git/GitHub** (commits clairs, gestion de versions).
4. **Présentation du projet** (explication des choix techniques, démonstration de l'API).