



## **(E2) Compte rendu de projet : Automatisation de la collecte horlogère Breguet et enrichissement par l'intelligence artificielle**

### **Table des matières**

<b>Introduction.....</b>	<b>2</b>
<b>1. Contexte et méthode actuelle de travail .....</b>	<b>2</b>
<b>2. Mise en place d'une solution automatisée par scraping.....</b>	<b>3</b>
<b>3. Enrichissement automatique des données via l'API OpenAI.....</b>	<b>4</b>
<b>4. Résultats obtenus.....</b>	<b>5</b>
<b>Conclusion .....</b>	<b>5</b>

# Introduction

Le client souhaite automatiser la collecte des informations sur les montres de la marque Breguet disponibles sur la plateforme EveryWatch.com. Aujourd'hui, ce travail est effectué manuellement, ce qui mobilise un temps considérable pour un résultat parfois incomplet ou sujet à erreurs.

L'objectif est donc de mettre en place une solution fiable, évolutive et simple à utiliser, permettant de récupérer automatiquement les données essentielles (modèle, référence, estimation, visuel, etc.), tout en préparant à terme une exploitation enrichie et intelligente de ces données.

Ce document a pour but de présenter une analyse stratégique des options disponibles, en trois volets :

- Un état des lieux des méthodes actuelles utilisées par le client,
- Une proposition de solution d'automatisation par scraping web,
- Un enrichissement possible via l'intelligence artificielle, en particulier l'API d'OpenAI, pour structurer et fiabiliser les données extraites.

## 1. Contexte et méthode actuelle de travail

Actuellement, la collecte d'informations sur les montres Breguet visibles sur le site *EveryWatch.com* est réalisée **manuellement**. Elle repose sur une navigation visuelle du site et une saisie à la main dans un tableau Excel, ligne par ligne. Ce processus, bien que rigoureux, est **extrêmement chronophage**. Il nécessite une attention constante pour éviter les erreurs de transcription, les doublons ou les oublis.

Ce mode de fonctionnement pose plusieurs **limites structurelles** :

- **Temps passé important**, surtout lorsque le volume de montres augmente.
- **Risques d'erreurs humaines**, qui peuvent affecter la fiabilité du fichier final.
- **Aucune scalabilité possible** : l'effort reste le même, que l'on doive relever 20 montres ou 200.
- **Peu de possibilité d'analyse ou d'exploitation poussée**, car les données sont limitées et souvent non normalisées.

Dans ce contexte, il est légitime d'envisager une **solution automatisée**, capable de fiabiliser la collecte, de gagner en productivité, et de poser les bases d'une exploitation plus intelligente des données.

## 2. Mise en place d'une solution automatisée par scraping

Afin de répondre au besoin d'automatiser la collecte d'informations sur les montres Breguet, une première solution a été développée à l'aide du langage **Python**. Elle repose sur un découpage en deux modules complémentaires, pour garantir une meilleure organisation et une facilité de maintenance.

- Le fichier `login.py` permet de gérer la **connexion au site EveryWatch.com**, en utilisant la bibliothèque Playwright. Il initialise le navigateur, assure une navigation fluide, et applique les éventuelles configurations nécessaires pour accéder à l'ensemble des contenus (cookies, fermeture des fenêtres contextuelles, etc.).
- Le second fichier, `re_logic_bis2.py`, constitue le **cœur du scraper**. Il contient la logique permettant de :
  - Ouvrir la page des montres disponibles sur le site.
  - Faire défiler automatiquement la page afin de charger l'ensemble des modèles affichés dynamiquement.
  - Identifier et filtrer les montres **portant la mention "Breguet"**.
  - Extraire pour chaque montre les **informations principales visibles** : nom du modèle, référence, estimation de prix, image, lien vers la fiche produit.
  - Regrouper l'ensemble des résultats dans un fichier **Excel structuré**, généré automatiquement.

Cette solution permet déjà de **diviser considérablement le temps de travail** et d'éliminer les erreurs liées à la saisie manuelle. Elle constitue une première étape solide vers une automatisation efficace.

Néanmoins, elle présente une limite importante : **certaines informations clés ne sont pas accessibles directement** depuis les cartes affichées sur la page principale. Pour les obtenir, il est nécessaire d'ouvrir chaque fiche produit, puis d'**analyser des blocs de texte libre** souvent non structurés.

C'est à ce niveau que l'intégration d'une **intelligence artificielle de traitement du langage naturel** (NLP) devient pertinente. Elle permettrait de **comprendre et extraire automatiquement** ces données techniques à partir des descriptions longues, en simulant une lecture humaine. C'est l'objet de la troisième phase de ce projet.

### 3. Enrichissement automatique des données via l'API OpenAI

Pour compléter les informations obtenues par scraping, une troisième étape a été mise en œuvre : **l'analyse des textes longs issus des fiches détaillées**. Ces descriptions, souvent riches en données techniques (diamètre, matière, type de mouvement, numéros de série, etc.), ne sont pas toujours présentées sous une forme structurée exploitable automatiquement.

Afin d'en extraire les informations pertinentes sans intervention humaine, nous avons intégré l'**API d'OpenAI**, qui permet d'interpréter ces blocs de texte libre à l'aide d'un modèle de traitement du langage naturel (LLM – Large Language Model).

Concrètement, l'outil envoie chaque description produit à l'API, avec un **prompt spécifique conçu pour identifier et extraire les éléments techniques horlogers** les plus couramment recherchés (taille du boîtier, matériau, mouvement, référence, présence d'accessoires, etc.). Les réponses sont ensuite structurées automatiquement dans un tableau Excel.

#### Pourquoi OpenAI ?

Le choix d'OpenAI repose sur plusieurs facteurs :

- **Fiabilité des résultats** : les modèles d'OpenAI sont parmi les plus performants pour l'analyse de texte complexe et semi-structuré.
- **Souplesse du paramétrage** : l'outil permet d'adapter finement les instructions (prompts) selon les besoins du client.
- **Facilité d'intégration** : l'API est facilement utilisable en Python, ce qui permet une continuité fluide avec le reste du script.

#### Coût de l'API

Le coût d'utilisation de l'API OpenAI dépend du volume de texte à traiter. À titre d'exemple, pour le modèle **gpt-4o** utilisé ici :

- Le traitement d'un lot de 100 fiches produit représente environ **1 à 2 euros** en coût d'appel API.
- Le coût reste **entièrement proportionnel à l'usage**, et peut être maîtrisé via des seuils ou des quotas.

Ce coût très faible rapporté au gain de temps et de fiabilité en fait une **solution à forte valeur ajoutée**, parfaitement adaptée à un usage ponctuel ou régulier dans le cadre d'un suivi horloger.

## 4. Résultats obtenus

La solution développée a permis d'automatiser l'extraction et l'enrichissement des données pour **4 800 montres** issues du site *EveryWatch.com*. Ce traitement, qui aurait nécessité plusieurs jours en manuel, est désormais exécuté en **moins d'une seconde** (0,86 s), grâce à l'utilisation combinée de Playwright et d'un moteur d'enrichissement intelligent basé sur l'API OpenAI.

À titre de comparaison, la saisie manuelle d'une fiche complète (environ 10 à 12 champs) prenait en moyenne **3 minutes par montre**, soit un total de **240 heures** de travail pour 4 800 références. Ce volume est désormais traité **en un clic**, avec une précision constante et sans fatigue.

L'enrichissement des fiches techniques via l'**API OpenAI** (1 appel par montre) permet d'extraire automatiquement des informations complexes à partir des descriptions longues : année de production, matériau, calibre, numéros de série, etc. Le coût moyen par appel est estimé à **0,004 €**, soit un **coût total de 19,20 €** pour l'ensemble des 4 800 montres.

En résumé :

- **Temps humain économisé** : ~240 heures (soit environ 6 semaines de travail à temps partiel)
- **Économie en coût humain** (base 20 €/h) : **≈ 4 800 €**
- **Coût total de l'automatisation (scraping + OpenAI)** : **< 20 €**

Le retour sur investissement est donc **immédiat et massif**, tout en garantissant une qualité et une standardisation des données difficilement atteignables en traitement manuel.

## Conclusion

Ce projet marque une étape importante dans la modernisation des processus de collecte et de structuration des données horlogères. En remplaçant un traitement manuel long, coûteux et peu scalable par une solution automatisée et enrichie par l'intelligence artificielle, nous avons démontré qu'il est possible d'allier **gain de temps**, **fiabilité des données** et **réduction des coûts**.

L'architecture mise en place reste ouverte à de futurs développements, notamment en intégrant des analyses visuelles (via IA de vision), des systèmes de détection de doublons ou des exports vers une base de données dynamique. Elle constitue ainsi une **base robuste et évolutive** pour construire une veille horlogère intelligente, rapide et durable.