

Compte Rendu

1. Introduction

Je travaille dans une entreprise internationale qui a des filiales dans plusieurs pays. Pour les **comptables**, j'avais pour mission de retourner le taux de change dans différentes devises en **temps réel**. Une fois l'information récupérée, je devais la **formater** et la mettre dans un fichier texte. Pour ensuite pouvoir l'intégrer dans l'**ERP** de l'entreprise. Pas de restrictions de technologie, je pouvais faire comme je le souhaitais.

2. Solution

Je me suis donc renseigné sur les différentes technologies qui pouvaient m'aider à réaliser cette tâche. J'ai vu qu'il y avait pas mal d'**API** déjà existantes pour **récupérer les taux de change**. Mais les API avaient, pour la plupart du temps, un **free trial limité**. C'était donc une solution **payante**. J'ai donc cherché une autre solution. Je me suis rendu compte que je pouvais récupérer les taux de change en utilisant une sorte de "**scraping**" (ce n'est pas vraiment du scrapping) de la page web.

3. Technologies Utilisées

Pour ce projet, j'ai utilisé les technologies suivantes :

- **Node.js** : Environnement d'exécution JavaScript
- **Playwright** : Bibliothèque d'automatisation de navigateur web
- **fs** : Module Node.js natif pour la manipulation des fichiers
- **path** : Module Node.js natif pour la gestion des chemins de fichiers

4. Quelques images

Windows PowerShell

Windows PowerShell

(index)	0	1	2	3	4	5	6	7
10								
0								
1	'1,0000'	'1,0365'	'163,1900'	'0,8289'	'0,9319'	'1,4919'	'10,3398'	'7,5627'
02'	'107,2704'							
2	'0,9648'	'1,0000'	'157,4480'	'0,7998'	'0,8990'	'1,4400'	'10,0672'	'7,2971'
69'	'103,4962'							
3	'0,0061'	'0,0064'	'1,0000'	'0,0051'	'0,0057'	'0,0091'	'0,0639'	'0,0463'
32'	'0,6573'							
4	'1,2065'	'1,2507'	'196,9000'	'1,0000'	'1,1243'	'1,8005'	'12,5879'	'9,1242'
99'	'129,4107'							
5	'1,0733'	'1,1125'	'175,1900'	'0,8897'	'1,0000'	'1,6018'	'11,1995'	'8,1178'
70'	'115,1365'							
6	'0,6700'	'0,6945'	'109,3524'	'0,5554'	'0,6243'	'1,0000'	'6,9919'	'5,0680'
28'	'71,8799'							
7	'0,0958'	'0,0993'	'15,6404'	'0,0794'	'0,0893'	'0,1430'	'1,0000'	'0,7249'
23'	'10,2808'							
8	'0,1322'	'0,1370'	'21,5772'	'0,1096'	'0,1232'	'0,1973'	'1,3796'	'1,0000'
98'	'14,1832'							

exchange_rates.txt

20/12/2024 15:07:43

Taux de Change au Comptant

Devise	EUR	USD	JPY	GBP	CHF	CAD	MAD	CNY	THB	ILS	RUB
EUR	1,0000	1,0365	163,1900	0,8289	0,9319	1,4919	10,3398	7,5627	35,8354	3,7802	107,2704
USD	0,9648	1,0000	157,4480	0,7998	0,8990	1,4400	10,0672	7,2971	34,5735	3,6469	103,4962
JPY	0,0061	0,0064	1,0000	0,0051	0,0057	0,0091	0,0639	0,0463	0,2195	0,0232	0,6573
GBP	1,2065	1,2507	196,9000	1,0000	1,1243	1,8005	12,5879	9,1242	43,2135	4,5599	129,4107
CHF	1,0733	1,1125	175,1900	0,8897	1,0000	1,6018	11,1995	8,1178	38,4470	4,0570	115,1365
CAD	0,6700	0,6945	109,3524	0,5554	0,6243	1,0000	6,9919	5,0680	24,0025	2,5328	71,8799
MAD	0,0958	0,0993	15,6404	0,0794	0,0893	0,1430	1,0000	0,7249	3,4330	0,3623	10,2808
CNY	0,1322	0,1370	21,5772	0,1096	0,1232	0,1973	1,3796	1,0000	4,7361	0,4998	14,1832
THB	0,0279	0,0289	4,5559	0,0231	0,0260	0,0417	0,2913	0,2111	1,0000	0,1055	2,9947
ILS	0,2646	0,2742	43,1754	0,2193	0,2465	0,3948	2,7606	2,0010	9,4769	1,0000	28,3802
RUB	0,0093	0,0097	1,5213	0,0077	0,0087	0,0139	0,0973	0,0705	0,3339	0,0352	1,0000

5. Description du Script

Le script fonctionne de la manière suivante :

1. Initialisation :
- Configuration de Playwright avec Chrome en mode headless
 - Configuration des chemins de fichiers
 - Définition des devises à surveiller (EUR, USD, JPY, GBP, CHF, etc.)
2. Récupération des données :
- Connexion au site Boursorama (<https://www.boursorama.com/bourse/devises/taux-de-change/>)
 - Attente du chargement du tableau des taux
 - Extraction des données via les sélecteurs CSS
3. Traitement des données :
- Formatage des taux de change en tableau
 - Ajout des en-têtes avec les devises
 - Alignement et mise en forme des données

4. Export :

- Génération d'un fichier texte formaté (exchange_rates.txt)
- Horodatage des données
- Mise à jour automatique toutes les 2 minutes

5. Fonctionnement du Projet

Le projet s'exécute de manière automatisée avec les caractéristiques suivantes :

5.1 Automatisation

- Exécution automatique toutes les 2 minutes
- Mode headless (sans interface graphique)
- Surveillance continue des taux de change

5.2 Format de Sortie

- Fichier texte structuré (exchange_rates.txt)
- Format tabulaire avec alignement des colonnes
- Horodatage de chaque mise à jour

5.3 Maintenance

- Code modulaire et bien structuré
- Gestion des erreurs intégrée
- Configuration facile à modifier (devises, intervalles)

Conclusion

Cette solution permet de :

- Réduire les coûts en évitant les API payantes
- Automatiser complètement le processus
- Garantir une mise à jour en temps réel des taux de change
- Faciliter l'intégration avec l'ERP existant

La solution est évolutive et peut être adaptée pour suivre d'autres devises ou modifier le format de sortie selon les besoins.

7. Extraits de Code Commentés

7.1 Initialisation et Imports

```
// Import des modules nécessaires
const { chromium } = require("playwright"); // Pour l'automatisation du navigateur
const fs = require("fs"); // Pour la manipulation des fichiers
const path = require("path"); // Pour la gestion des chemins
```

7.2 Configuration des Devises

```
// Configuration des en-têtes avec les devises suivies
const headers = [
  "EUR",
  "USD",
  "JPY",
  "GBP",
  "CHF",
  "CAD",
  "MAD",
  "CNY",
  "THB",
  "ILS",
  "RUB",
];

// Tableau complet des devises avec espace pour l'alignement
const currencies = [
  " ",
  "EUR",
  "USD",
  "JPY",
  "GBP",
  "CHF",
  "CAD",
  "MAD",
  "CNY",
  "THB",
  "ILS",
  "RUB",
];
```

7.3 Extraction des Données

```
// Fonction principale de récupération des taux
const fetchExchangeRates = async () => {
  const browser = await chromium.launch({ headless: true });
  const page = await browser.newPage();

  // Navigation vers la page des taux de change
  await page.goto("https://www.boursorama.com/bourse/devises/taux-de-change/");
  await page.waitForSelector(".c-table__body");

  // Extraction des données du tableau
  const exchangeRates = await page.$$eval(".c-table__row", (rows) => {
    return rows.map((row) => {
      const cells = Array.from(row.querySelectorAll("td")).map((cell) =>
        cell.innerText.trim()
      );
    });
  });
```

```
        return cells;
    });
});
};
```

7.4 Formatage et Export

```
// Formatage des données en tableau aligné
const formattedExchangeRates = exchangeRates
    .map((row, index) =>
        [currencies[index], ...row]
        .map((cell) => (cell ? cell.padEnd(8) : " ").padEnd(8)))
        .join(" | ")
    )
    .join("\n");

// Création du contenu final avec horodatage
const formattedHeaders = ["Devise", ...headers]
    .map((h) => h.padEnd(8))
    .join(" | ");

const content = `${new Date().toLocaleString()}
\nTaux de Change au Comptant\n
${formattedHeaders}\n${formattedExchangeRates}\n\n`;

// Écriture dans le fichier
const filePath = path.join(__dirname, "exchange_rates.txt");
fs.writeFileSync(filePath, content);
```

7.5 Automatisation

```
// Fonction de démarrage avec intervalle
const startFetching = async () => {
    await fetchExchangeRates(); // Première exécution
    setInterval(fetchExchangeRates, 120000); // Toutes les 2 minutes
};

// Lancement du programme
startFetching();
```

Ces extraits de code montrent les points clés de l'application :

- L'utilisation de Playwright pour l'automatisation du navigateur
- La structure de données pour les devises
- L'extraction et le formatage des données
- L'export dans un fichier texte
- L'automatisation avec des intervalles réguliers

