

## # Scraper de Tarifas de Hoteles en Booking.com

Este proyecto proporciona un scraper automatizado para extraer la tarifa más baja de hoteles específicos en Booking.com para los próximos 60 días y generar reportes diarios en formato Excel y CSV.

### ## Características

- \* Extracción de tarifas de hoteles a partir de una lista de URLs de Booking.com.
- \* Cálculo dinámico de las fechas para los próximos 60 días a partir de la fecha de ejecución.
- \* Identificación de la tarifa más baja por día para cada hotel.
- \* Generación de reportes en formato Excel (.xlsx) y CSV.
- \* Configuración para automatización diaria mediante `cron`.

### ## Requisitos Previos

Para ejecutar este scraper, necesitarás tener instalado lo siguiente:

- \* **Python 3.x**
- \* **pip** (gestor de paquetes de Python)
- \* Una cuenta en **Apify** y tu **Apify API Token**.

### ## Instalación

1. **Clonar el repositorio (si aplica) o descargar los archivos del proyecto.**
2. **Instalar las dependencias de Python:**

Abre tu terminal y ejecuta el siguiente comando:

```
```bash
pip install apify-client pandas openpyxl
```
```

- \* `apify-client`: Para interactuar con la API de Apify.
- \* `pandas`: Para la manipulación de datos y la generación de archivos Excel/CSV.
- \* `openpyxl`: Es una dependencia de `pandas` para escribir archivos `.xlsx`.

### ## Configuración

1. **Obtener tu Apify API Token:**

Si aún no tienes una cuenta en Apify, regístrate en [Apify.com](https://apify.com/). Una vez que hayas iniciado sesión, ve a la sección de **Integrations settings** en tu consola de Apify para encontrar tu token de API personal.

2. **Configurar el script `apify_scraper.py`:**

Abre el archivo `apify_scraper.py` con un editor de texto y realiza los siguientes cambios:

- \* **Reemplaza `YOUR_APIFY_API_TOKEN`:**

Busca la línea:

```
```python
APIFY_API_TOKEN = "YOUR_APIFY_API_TOKEN"
```
```

Y reemplázala con tu token de API real de Apify:

```
```python
APIFY_API_TOKEN = "tu_token_api_real_aqui"
```
```

- \* **Añadir las URLs de tus hoteles:**

Busca la sección `hotel_urls`:

```
```python
hotel_urls = [
    "https://www.booking.com/hotel/es/barcelo-sants.es.html",
    # Agrega más URLs de hoteles aquí
]
```
```

Reemplaza la URL de ejemplo y añade las URLs de los 5 hoteles específicos de Booking.com que desees monitorear. Asegúrate de que cada URL esté entre comillas dobles y separada por comas.

### ## Uso

Para ejecutar el scraper manualmente, abre tu terminal, navega hasta el directorio donde guardaste `apify_scraper.py` y ejecuta:

```
```bash
python3 apify_scraper.py
```
```

```
'''

El script imprimirá mensajes de progreso en la consola y, al finalizar, generará dos archivos en el mismo directorio:
```

```
* `tarifas_hoteles_YYYYMMDD.xlsx`
* `tarifas_hoteles_YYYYMMDD.csv`
```

Donde `YYYYMMDD` será la fecha de ejecución del script (ej., `tarifas\_hoteles\_20250620.xlsx`).

```
## Automatización Diaria (usando `cron`)
```

Para que el scraper se ejecute automáticamente cada día, puedes configurarlo usando `cron` en sistemas Linux (incluido el entorno de sandbox).

#### 1. \*\*Hacer el script ejecutable:\*\*

Abre tu terminal y ejecuta:

```
```bash
chmod +x /home/ubuntu/apify_scraper.py
```
```

#### 2. \*\*Editar el crontab:\*\*

En la terminal, ejecuta:

```
```bash
crontab -e
```
```

Si es la primera vez que editas el crontab, se te pedirá que elijas un editor. `nano` es una buena opción para principiantes.

#### 3. \*\*Añadir la tarea programada:\*\*

Dentro del editor, añade la siguiente línea al final del archivo. Esta línea ejecutará el script todos los días a la 1:00 AM (puedes ajustar la hora cambiando los primeros dos números):

```
'''
0 1 * * * /usr/bin/python3 /home/ubuntu/apify_scraper.py >> /home/ubuntu/scraper_log.log 2>&1
'''
```

```
* `0 1 * * *`: Ejecuta a la 1:00 AM todos los días.
* `/usr/bin/python3`: Ruta al intérprete de Python 3.
* `/home/ubuntu/apify_scraper.py`: Ruta completa a tu script.
* `>> /home/ubuntu/scraper_log.log 2>&1`: Redirige la salida y los errores a un archivo de registro para depuración.
```

#### 4. \*\*Guardar y salir:\*\*

\* Si usas `nano`: Presiona `Ctrl+O`, luego `Enter`, y finalmente `Ctrl+X`.

```
## Estructura de los Archivos de Salida
```

Los archivos Excel y CSV contendrán las siguientes columnas:

```
* **Hotel Name**: Nombre del hotel.
* **URL**: URL del hotel en Booking.com.
* **YYYY-MM-DD**: Columnas para cada uno de los próximos 60 días, mostrando la tarifa más baja encontrada para ese día.
Si no se encuentra una tarifa, se mostrará
```

"N/A".

```
## Solución de Problemas Comunes
```

```
* **`ERROR: Por favor, reemplaza 'YOUR_APIFY_API_TOKEN' con tu token real de Apify...`**: Asegúrate de haber actualizado la variable `APIFY_API_TOKEN` en `apify_scraper.py` con tu token real.
* **No se generan archivos de salida**: Verifica que las URLs de los hoteles en `hotel_urls` sean correctas y que el token de Apify sea válido. Revisa el archivo `scraper_log.log` para posibles errores si la automatización está configurada.
* **Precios incorrectos o faltantes**: La lógica de extracción de precios asume que la API de Apify devuelve los precios de las habitaciones y las fechas de check-in/check-out. Si la estructura de datos de la API cambia o si Booking.com no proporciona la información de precios para ciertas fechas, el script podrá no encontrar los datos. Revisa la salida de la API en la consola de Apify para depurar.
* **Problemas con `cron`**: Asegúrate de que la ruta al intérprete de Python (`/usr/bin/python3`) y al script (`/home/ubuntu/apify_scraper.py`) sean correctas. Revisa el archivo `scraper_log.log` para ver los errores de ejecución del `cron`.
```

```
## Contacto
```

Si tienes alguna pregunta o necesitas ayuda adicional, no dudes en contactarme.