

Prueba técnica: **Web Scrapers**

Esta prueba consiste en implementar el backend de una aplicación que permita gestionar procesos que consulten **periódicamente** valores en una página web, para este caso [CoinMarketCap](https://coinmarketcap.com). A estos procesos los llamaremos *scrapers*, y consisten en un hilo o tarea que extrae un valor dentro de un HTML (como el que se obtiene al hacer un GET a <https://coinmarketcap.com>) con una frecuencia dada. **Para esto debes implementar una API que permita ver, añadir, eliminar y editar la frecuencia de un scraper.**

Para visualizar los *scrapers* creados, ya se desarrolló¹ una vista simple que permite verlos en una tabla. A continuación un ejemplo de esta vista.

ID	Moneda	Último valor leído	Fecha última actualización	Frecuencia	Fecha de creación
1	Dogecoin	\$ 0.003181	16:11:30 27/08/20	25	12:16:37 27/08/20
3	Bitcoin	\$ 11275.9	16:11:30 27/08/20	60	12:53:47 27/08/20

Ejemplo de la vista mostrando 2 procesos de *scrapers*.


A continuación, una captura de pantalla de CoinMarketCap con la columna **Price** resaltada (puedes obtener el valor de cualquier otra parte del sitio, mientras sea leyendo el HTML² y no usando su API).

¹ Esta vista ya está implementada (no debes desarrollarla) y funcionará siempre y cuando se cumplan las definiciones de la API requerida.




² Una forma de obtener el HTML como string es utilizando el valor de `content` de un response. Por ejemplo:

```
python -c "import requests;
print(requests.get('https://coinmarketcap.com/').content)"
```

Cryptocurrencies: 5,670 • Markets: 22,685 • Market Cap: \$260,490,062,774 • 24h Vol: \$56,756,249,3

 **CoinMarketCap** Rankings Tools

Top 100 Cryptocurrencies by

Cryptocurrencies ▾ Exchanges ▾ Watchlist				
Rank	Name	Market Cap	Price	Volume (24h)
1	 Bitcoin	\$168,709,865,689	\$9,159.53	\$16,351,411,436
2	 Ethereum	\$25,304,155,930	\$226.81	\$6,518,422,105
3	 Tether	\$9,200,173,582	\$1.00	\$19,954,671,212

Una base de datos SQLite simple debe guardar los *scrapers* creados. A continuación una propuesta para este modelo:

Columna	Tipo
id	Integer
created_at	DateTime
currency	Characters
frequency	Integer

Se entregó junto a este documento un proyecto basado en **Django** sobre el cual esperamos que implementes tu solución. Podrás ver un *README* con la definición de la API que debes implementar.

Objetivos a implementar:

- Permitir **crear** *scrapers* para consultar a *coinmarketcap.com* los precios de las monedas: Bitcoin, Ethereum, Tether, etc.
- Permitir **editar** la frecuencia de un *scraper*.
- Permitir **eliminar** un *scraper*.
- Validar que no se puedan crear varios *scraper* para la misma moneda.

Requisitos:

- No agregar más dependencias a las ya definidas.
- Entregar la solución a la prueba en a lo más **3 días**, enviando/compartiendo un repositorio en Git a equipo@rgcorp.cl con el título **Solución a prueba backend**.

Disclaimer: RG Corp SpA reconoce los derechos de Propiedad Intelectual del candidato sobre los códigos desarrollados durante el proceso de postulación, reconociendo que este es un desarrollo absolutamente original, de manera que la empresa no podrá hacer uso de los mismos bajo ninguna circunstancia.