

Economía y Ciencia de los Datos

Uso de APIs

Carlos Alvarado & Pablo González

29 de agosto de 2022

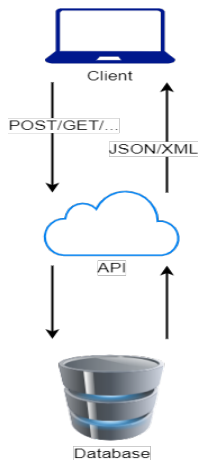
Bases necesarias y motivación

¿Qué es una API?

El término API alude a una Interfaz de Programación de Aplicaciones y se refiere a las funciones, métodos y procesos que brinda una aplicación/software para que sea empleada por otro programa informático (similar a un contrato de servicio entre dos aplicaciones).

Aspectos generales:

- Servicio entregado por la aplicación.
(i.e., proveedor de los datos)
- Suelen requerir registro previo a su aplicación.
- Existe documentación y condiciones de uso.
- Output generalmente en formato JSON.



¿Para qué podríamos usar una?

Como científicos de datos, la APIs nos proveeran de fuentes de información generalmente actualizadas y en un formato estructurado. Estos "servicios" suelen incluir documentación, haciendo su uso más fácil y, generalmente, con posibilidad de hacer consultas en caso de encontrar un problema.

Entre algunos ejemplos de APIs interesantes para el curso destacan:

- BDE de Banco Central:
https://si3.bcentral.cl/estadisticas/Principal1/web_services/index.htm
- Banco Mundial:
<https://datahelpdesk.worldbank.org/knowledgebase/topics/125589>
- TMDb: <https://www.themoviedb.org/documentation/api>
- Reddit: <https://www.reddit.com/dev/api/>
- Twitter: <https://developer.twitter.com/en/docs/twitter-api>

Conectándonos a una API (via Python)

Requisitos

Una vez generamos una cuenta para utilizar la API de un sitio web, el siguiente paso correspondería a conectarnos.

En esta ocasión utilizaremos los siguientes módulos:

- **json**: Permite codificar y decodificar datos en formato json (notación de objetos de JavaScript)
- **requests**: Nos permite transmitir datos en la web via protocolo http.

Es importante notar que si bien la metodología tiende a ser similar para cada sitio, antes de conectarse a cualquier API es importante leer su documentación.

Ejemplo: Conexión a Banco Central

Código de ejemplo para solicitar una serie desde BDE

```
def bde_data( serie, name, freq, first_date = None, last_date = None,
             user = '#usuario', password = '#contraseña' ):

    base_url = "https://si3.bcentral.cl/SieteRestWS/SieteRestWS.ashx"
    url = f"{base_url}?user={user}&pass={password}" + \
          f"{'&firstdate=' + str(first_date) if first_date else ''}" + \
          f"{'&lastdate=' + str(last_date) if last_date else ''}" + \
          f"&timeseries={serie}&function=GetSeries"

    response = requests.get(url).json()['Series']['Obs']

    temp = pd.DataFrame( response ).iloc[:, :-1] \
           .astype({ 'value': float }) \
           .rename( columns = { 'indexDateString': 'periodo', 'value': name })

    temp[ 'periodo' ] = pd.to_datetime( temp[ 'periodo' ], format = '%d-%m-%Y' ) \
                       .dt.to_period( 'D' ).dt.strftime( freq )

    return temp.set_index( 'periodo' )
```

- Usuario y contraseña deben ser solicitados.
- El segmento relacionado con la frecuencia no es necesario, pero puede servir si posteriormente se quieren calcular estadísticos a nivel trimestral / anual.

Ejemplo: The Movie Database

Código de ejemplo para solicitar datos desde TMDB

```
def obtener_peliculas(self, pid:str, min_cal:float, votes:int)->list:
    """
    Obtiene información de créditos (película) de la persona cuya id es entregada.
    Esto ocurre unicamente si la película cumple con una calificación y cantidad de votos mínimas.
    """

    conn = http.client.HTTPSConnection("api.themoviedb.org")
    conn.request("GET", f"{self.base_url}/person/{pid}/movie_credits?api_key={self.api_key}&language=en-US")
    respuesta, data = conn.getresponse(), []

    if respuesta.status == 200:

        cases = json.loads( respuesta.read().decode('utf-8') )['cast']

        # Ocasionalmente las películas no tienen calificación media
        for i in cases:
            if 'vote_average' in i:
                if i['vote_average'] >= min_cal and i['vote_count'] >= votes:
                    data.append(i)
            else:
                print(f'La extracción de películas terminó con el siguiente error {respuesta.status}')

    conn.close()
    return data
```

Algunos aspectos relevantes a mencionar son:

- Posibilidad de filtrar por diversos aspectos (incluyendo géneros y años).
- En caso de buscar construir redes, tener cuidado con los filtros (existen muchas películas).
- Notar que no siempre se entregan todos los campos.

Pongámoslo en práctica

Ir a Notebook

