APIs

Protocolos que permiten a diferentes aplicaciones o sistemas comunicarse entre sí. Definen cómo se pueden solicitar y compartir datos de fuentes externas.

Librería request: herramienta de Python para hacer peticiones a una API para obtener información.

Cómo se usa:

```
import requests # Traemos la librería para usarla.

# Hacemos la petición a un API de ejemplo.
response = requests.get('https://api.example.com/weather?city=Madrid') # Esta direcció

# Mostramos la respuesta que nos devuelve el API.
print(response.json())
```

1. Traemos la librería para usarla.
   import requests

2. Hacemos la petición a un API, la dirección es el endpoint.
   response = requests.get('https://api.example.com/weather?city=Madrid')

3. Mostramos la respuesta que nos devuelve el API en formato JSON. Leer como si fuera un diccionario.
   print(response.json())

4. Se puede conocer el **estado de una petición**, el API devuelve una respuesta con los datos solicitados y un código que indica cómo fue la solicitud:
   print(response.status_code)

```
import requests # No olvidar traer la librería!

# Hacemos una petición al API
response = requests.get('https://api.example.com/weather?city=Madrid')

# Verificamos si la petición fue exitosa (código 200)
if response.status_code == 200:
    # Mostramos los datos del API
    print(response.json())
else:
    # Mostramos un mensaje si hubo un error
    print(f"Error en la petición. Código de estado: {response.status_code}")
```

Métodos HTTP: acciones que se pueden realizar en un endpoint.

- Get: obtener datos.
- Post: enviar datos para crear nuevos recursos.
- Put: actualizar recursos existentes.
- Delete: eliminar algo.

Headers y Params:

- **Headers:** etiquetas que agregan información adicional, ayudan a describir lo que solicitas o envías.
  La API **balldontlie** especifica que necesitas api key (autorización) en la solicitud.
- **Params:** next_cursor datos a partir de una página. per_page tantos elementos por página.

```
response = requests.get(f"http://api.balldontlie.io/v1/players", # Nuestra pagir Copiar d
        headers={'Authorization': tu_api_key}, # Aquí hay un header, que es necesario
        params={'next_cursor': pagina, 'per_page': por_pagina}) # Aquí un params...
```

Solicitudes y respuestas: las **solicitudes** se hacen a través de métodos HTTP para interactuar con una API. Las **respuestas** contienen los datos solicitados y si la solicitud fue exitosa. **Códigos de respuestas**:

CÓDIGO	VALOR CÓDIGO	SIGNIFICADO	EXPLICACIÓN
1XX	Respuesta correcta	Uso infrecuente	
2XX	Éxito	Petición procesada	
	200	**OK**	**Petición exitosa**
	201	Creado	Se ha creado lo solicitado por petición
	202	Aceptado	Procesando petición
	204	Sin contenido	Petición procesada, no hay contenido para devolver
3XX	Redirección	A otra URL	
4XX	Error durante la petición		
	401	Petición incorrecta	No se ha procesado correctamente por el servidor
	402	Sin autorización	Debe identificarse
	403	Prohibido	El servidor no permite por falta de permisos
	404	**No encontrado**	**Lo solicitado no existe**
5XX	**Error del servidor**	**No da respuesta**	
	501	Error interno	El servidor no pude procesar peticiones
	503	No disponible	Exceso de peticiones o fuera de servicio

Endpoints: URLs específicas a las que puedes acceder a través de APIs para Realizar operaciones específicas.

Autenticación: verifica la identidad del usuario o aplicación que realiza la solicitud a la API. **Mecanismos de autenticación:** tokens de acceso, claves de API.

Rate limiting: algunas APIs limitan la cantidad de solicitudes de un usuario en un tiempo.

Formato de datos: proporcionan datos en varios formatos como JSON o XML.

Documentación: obligatorio leerla para conocer información sobre endpoints disponibles, parámetros de solicitud y respuestas esperadas.

Ética y legalidad: respetar las políticas de uso de las APIs.