**Descripción de la API Elegida: REST COUNTRIES**

1. **Visión General**

La API **REST Countries** es un servicio gratuito y de código abierto que proporciona datos detallados sobre todos los países del mundo. Está basada en principios RESTful y ofrece endpoints para consultar información geográfica, demográfica, económica y cultural de los países. Originalmente inspirada en el proyecto de, **Fayder Florez** [**https://github.com/apilayer/restcountries/**](https://github.com/apilayer/restcountries/)**)** actualmente es mantenida por **apilayer**, una empresa líder en microservicios API.

1. **Características Principales**

* **Acceso Gratuito:** No requiere autenticación ni API Key para uso básico.
* **Datos Actualizados:** Incluye información como nombres, capitales, monedas, idiomas, regiones, áreas, poblaciones, y más.
* **Formato de Respuesta:** Datos estructurados en formato **JSON**.
* **Filtrado de Campos:** Permite especificar campos deseados con el parámetro **?fields=** para reducir la carga de datos.
* **Compatibilidad con Expresiones Regulares:** Soporta búsquedas avanzadas con patrones de texto.
* **Soporte para CORS:** Facilita su uso en aplicaciones web frontend.
* **Lenguajes y Monedas:** Incluye códigos ISO 639-1 (idiomas) y ISO 4217 (monedas).
* **Licencia:** Distribuida bajo la **Mozilla Public License 2.0** (MPL 2.0), permitiendo uso comercial y modificación con atribución.

1. **Puntos de Acceso (Endpoints)**

La API ofrece múltiples endpoints para consultas específicas. Algunos destacan por su utilidad directa:

|  |  |  |
| --- | --- | --- |
| ENDPOINT | DESCRIPCION | EJEMPLO |
| /v3.1/all | Devuelve todos los países con todos sus datos. | [**https://restcountries.com/v3.1/all**](https://restcountries.com/v3.1/all) |
| /v3.1/name/{name} | Búsqueda por nombre común u oficial del país (coincidencia parcial). | [**https://restcountries.com/v3.1/name/france**](https://restcountries.com/v3.1/name/france) |
| /v3.1/capital/{capital} | Búsqueda por ciudad capital. | [**https://restcountries.com/v3.1/capital/paris**](https://restcountries.com/v3.1/capital/paris) |
| /v3.1/alpha/{code} | Búsqueda por códigos ISO (2 letras, 3 letras, numérico o CIOC). | [**https://restcountries.com/v3.1/alpha/co**](https://restcountries.com/v3.1/alpha/co)  **(Colombia)** |
| /v3.1/currency/{currency} | Búsqueda por moneda (código ISO 4217). | [**https://restcountries.com/v3.1/currency/cop**](https://restcountries.com/v3.1/currency/cop)  **(COP, Colombia)** |
| /v3.1/lang/{language} | Búsqueda por idioma (código ISO 639-1 o nombre). | [**https://restcountries.com/v3.1/lang/spanish**](https://restcountries.com/v3.1/lang/spanish) |
| /v3.1/region/{region} | Búsqueda por región (ej.: Europa, África). | [**https://restcountries.com/v3.1/region/europe**](https://restcountries.com/v3.1/region/europe) |
| /v3.1/subregion/{subregion} | Búsqueda por subregión (ej.: "Northern Europe"). | [**https://restcountries.com/v3.1/subregion/Northern**](https://restcountries.com/v3.1/subregion/Northern)  **Europe** |
| /v3.1/translation/{translation} | Búsqueda por nombre traducido (ej.: "Germany" en inglés). | [**https://restcountries.com/v3.1/translation/germany**](https://restcountries.com/v3.1/translation/germany) |

1. **Estructura de los Datos Devueltos**

La API devuelve objetos JSON con atributos clave. Ejemplo de respuesta para Colombia:

[

{

"name": {

"common": "Colombia",

"official": "República de Colombia",

"nativeName": {

"spa": {

"official": "República de Colombia",

"common": "Colombia"

}

}

},

"tld": [".co"],

"cca2": "CO",

"ccn3": "170",

"cca3": "COL",

"cioc": "COL",

"independent” : true,

"status” : "officially-assigned",

"unMember": true,

"currencies": {

"COP": {

"name": "Peso colombiano",

"symbol": "$"

}

},

"capital": ["Bogotá"],

"region": "Americas",

"subregion” : "South America",

"languages” : {

"spa” : "Spanish"

},

"translations” : {

"deu” : {"official” : "Republik Kolumbien", "common” : "Kolumbien"},

"fra” : {"official” : "République de Colombie", "common” : "Colombie"}

},

"population” : 50882891,

"area” : 1141748,

"gini” : {"2019” : 51.3},

"timezones” : ["UTC−05 :00"],

"borders": ["BRA", "ECU", "PAN", "PER", "VEN"],

"flag” : "https://restcountries.eu/data/col.svg",

"regionalBlocs” : [

{

"acronym” : "PA",

"name” : "Pacific Alliance",

"otherNames": ["Alianza del Pacífico"]

},

{

"acronym": "USAN",

"name” : "Union of South American Nations",

"otherAcronyms": ["UNASUR", "UNASUL", "UZAN"],

"otherNames": ["Unión de Naciones Suramericanas", "South American Union"]

}

]

}

]

1. **Uso en Python**

La API se integra fácilmente con Python usando la biblioteca requests. Ejemplo básico:

import requests

# Endpoint para obtener todos los países con campos específicos

url = "https://restcountries.com/v3.1/all?fields=name,capital,currencies,region,population"

try:

respuesta = requests.get(url)

# Verificar si la respuesta es exitosa (código 200)

respuesta.raise\_for\_status()

# Convertir a JSON

datos = respuesta.json()

# Mostrar datos de los primeros 5 países

for pais in datos[:5]:

print(f"Nombre: {pais['name']['common']}")

print(f"Capital: {pais.get('capital', ['N/A'])[0]}")

print(f"Región: {pais.get('region', 'N/A')}")

print(f"Población: {pais.get('population', 0)}")

print("-" \* 40)

except requests.exceptions.RequestException as e:

print(f"Error en la solicitud: {e}")

**Salida Ejemplo:**

Nombre: Colombia

Capital: Bogotá

Región: Americas

Población: 50882891

----------------------------------------

Nombre: Brazil

Capital: Brasília

Región: Americas

Población: 212559417

----------------------------------------

Nombre: France

Capital: Paris

Región: Europe

Población: 67902648

----------------------------------------

1. **Integración con Funcionalidades del Curso**

La API se alinea perfectamente con los objetivos pedagógicos del curso de programación básica en Python:

1. **Interacción con APIs (Fase III):**

* Uso de **requests.get()** para solicitudes HTTP y manejo de errores con **try-except.**
* Procesamiento de respuestas JSON con **response.json()**.

1. **Manipulación de Datos (Fase II):**

* Extracción de campos específicos (ej.: **nombre**, **población**) con acceso seguro usando **.get(clave, valor\_predeterminado)**.
* Uso de listas por comprensión para filtrar y transformar datos.

1. **Análisis Estadístico (Fase III):**

* Extracción de valores numéricos (ej.: **population**, **area**) para calcular estadísticas con **statistics**.
* Visualización de datos con **matplotlib** (ej.: gráficos de barras para población).

1. **Manejo de Archivos (Fase III):**

* Exportación de datos a Excel con **pandas** o JSON con **json.dump()**.

1. **Expresiones Regulares (Fase III):**

* Uso de **re.search()** para filtrar países por patrones en nombres o idiomas.

1. **Casos de Uso Real**

**Casos de Uso Reales**

* **Spotify International Pricing Index:** Ajuste de precios según región y moneda.
* **Gorillaz:** Integración en proyectos multimedia.
* **Xero:** Gestión financiera con datos geográficos.
* **Spotify:** Adaptación de contenido según idioma y región.

1. **Errores y Buenas Prácticas**

**Errores Comunes:**

1. Sin conexión a Internet: Manejar excepciones con **try-except**.
2. Respuesta no válida: Validar **response.status\_code == 200**.
3. Campos inexistentes: Usar **.get(clave, valor\_predeterminado)**.

**Buenas Prácticas:**

* Filtrar Campos: Reducir tráfico usando **?fields=**.
* Validación de Datos: Verificar valores nulos o inválidos.
* Documentación: Usar **docstrings** para explicar funciones.
* Límites de Tasa (Rate Limiting): Aunque no se especifican explícitamente, el proyecto recibe millones de visitas diarias, lo que podría requerir donaciones para mantener la estabilidad.

1. **Limitaciones y Consideraciones**

* No Soporta Operadores Lógicos: No permite búsquedas complejas como "países en Europa y con moneda EUR".
* Formato de Datos: Algunos campos (ej.: **borders**, **languages**) requieren procesamiento adicional para ser legibles.

1. **Conclusión**

La **API REST Countries** es una herramienta versátil y accesible para integrar datos geográficos y demográficos en aplicaciones de cualquier tipo. Su combinación de facilidad de uso, datos exhaustivos y comunidad activa la convierte en una opción ideal para desarrolladores principiantes y avanzados. La integración con Python, como se muestra en el código del curso, permite resolver problemas reales de análisis de datos, automatización y visualización, siguiendo principios de Modularización y buenas prácticas de programación.